



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Πληροφορική»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Ανάπτυξη Διαδικτυακής Εφαρμογής Ηλεκτρονικού Καταστήματος Βιβλιοπωλείου Web Application Online Bookstore
Όνοματεπώνυμο Φοιτητή	ΑΓΓΕΛΟΣ ΛΑΒΔΑΣ
Πατρώνυμο	ΣΤΕΦΑΝΟΣ
Αριθμός Μητρώου	ΜΠΠΛ18041
Επιβλέπων	ΕΥΘΥΜΙΟΣ ΑΛΕΠΗΣ, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης **Δεκέμβριος 2022**

Τριμελής Εξεταστική Επιτροπή

Ευθύμιος Αλέπης
Αναπληρωτής Καθηγητής

Μαρία Βίρβου
Καθηγήτρια

Ευάγγελος Σακκόπουλος
Αναπληρωτής Καθηγητής

Ευχαριστίες

Η παρούσα μεταπτυχιακή εργασία με θέμα «Ανάπτυξη Διαδικτυακής Εφαρμογής Ηλεκτρονικού Βιβλιοπωλείου », εκπονήθηκε το 2022 από τον φοιτητή ΛΑΒΔΑ ΑΓΓΕΛΟ του Μεταπτυχιακού Προγράμματος «Πληροφορική» της Σχολής Τεχνολογιών Πληροφορικής και Επικοινωνιών, του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς.

Μετά το πέρας της εκπόνησής της διατριβής, νιώθω την ανάγκη να εκφράσω τις ευχαριστίες μου προς τα άτομα που συνέβαλαν στην ολοκλήρωση της, παρέχοντάς μου είτε γνώση, είτε ηθική υποστήριξη.

Για την υλοποίηση της διπλωματικής αυτής εργασίας θέλω να ευχαριστήσω τον επιβλέπων Αναπληρωτή Καθηγητή του τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς κ. Ευθύμιο Αλέπη, για την ανάθεση του θέματος ,τον καθοδηγητικό και συμβουλευτικό του ρόλο σε όλη την διάρκεια υλοποίησης του Project.

Επιπρόσθετα ,θέλω να εκφράσω τις ευχαριστίες και την ευγνωμοσύνη μου σε όλους τους καθηγητές του προγράμματος μεταπτυχιακών σπουδών «Πληροφορική » του Πανεπιστημίου Πειραιώς για τις γνώσεις και τις πολύτιμες εμπειρίες που απέκτησα στο κλάδο της Πληροφορικής.

Τέλος, δεν θα μπορούσα να μην αναφερθώ στην οικογένειά μου και στους φίλους μου, για την κατανόηση, την συμπαράσταση και την ενθάρρυνση που μου παρείχαν καθ' όλη τη διάρκεια του μεταπτυχιακού προγράμματος και ιδιαίτερα στη διεκπεραίωση της εργασίας αυτής .

Περιεχόμενα

Ευχαριστίες	4
Περιεχόμενα	5
Περίληψη	7
Abstraction	7
Κεφάλαιο 1	8
1.1 Αντικείμενο διπλωματικής εργασίας	8
1.2 Σύνοψη διπλωματικής εργασίας	9
Κεφάλαιο 2 - Τεχνολογίες και Αρχιτεκτονική Σχεδίαση	10
2.1 Περιβάλλον Ανάπτυξης	10
2.2 Εισαγωγή ASP.NET Framework	10
2.2.1 .NET Framework	11
2.2.2 .NET Core	12
2.2.3 Διαφορές ανάμεσα σε .NET Framework και .NET Core	13
2.3 Γλώσσα προγραμματισμού C#	14
2.4 Αρχιτεκτονική MVC (Model -View -Controller)	15
2.4.1 Τρέχουσες τάσεις του MVC	15
2.4.2 Αναλυτικά τα μέρη του MVC Framework	16
2.4.3 Πλεονεκτήματα Αρχιτεκτονικής MVC	18
2.5 Τεχνολογίες διεπαφής χρήστη (front-end)	19
2.5.1 Html	19
2.5.2 Css (Cascading Style Sheets)	20
2.5.3 Bootstrap	20
2.5.4 JavaScript	21
2.5.5 jQuery βιβλιοθήκη	22
2.5.6 Σύνταξη Razor για ASP.NET MVC	22
2.6 Σχεδιασμός της Βάσης Δεδομένων	23
2.6.1 Βάση Δεδομένων Microsoft SQL Server	23
2.6.2 Entity Framework – Διαχείριση Δεδομένων	23
2.6.3 Χαρακτηριστικά του Entity Framework	24

2.6.4 Προσεγγίσεις για τη δημιουργία των Entities στο Entity Framework	25
2.6.5 Αποθήκευση και ανάκτηση δεδομένων -LINQ	26
Κεφάλαιο 3 – Ανάλυση Απαιτήσεων.....	28
3.1 UML (Unified Modeling Language)	29
3.2 Διάγραμμα Περιπτώσεων Χρήσης (Use Case)	33
3.3 Διάγραμμα Κλάσεων (Class)	35
3.4 Διάγραμμα Δραστηριότητας (Activity)	36
3.5 Διάγραμμα Ακολουθίας (Sequence).....	38
3.6 Διάγραμμα Καταστάσεων (State machine)	40
Κεφάλαιο 4 - Δομικά χαρακτηριστικά της εφαρμογής	41
4.1 Controllers	43
4.2 Models.....	70
4.3 Views.....	75
4.4 ViewModels	81
4.5 Data και appsettings	83
Κεφάλαιο 5 - Παρουσίαση λειτουργιών της εφαρμογής	87
5.1 Αρχική Σελίδα (HomePage)	87
5.2 Εγγραφή Πελάτη (Customer Register)	90
5.3 Είσοδος Πελάτη - Login Customer	92
5.4 Δημιουργία Παραγγελίας.....	98
5.5 Είσοδος Διαχειριστή (Admin Login) & Λειτουργίες Διαχειριστή.....	103
5.5.1 Αναζήτηση Παραγγελιών Πελατών	106
5.5.2 Απόθεμα Βιβλίων	110
5.5.3 Διαθέσιμες Αποθήκες Βιβλιοπωλείου	113
5.5.4 Πωλήσεις Βιβλίων	115
5.5.5 Παραγγελίες βιβλίων σε Προμηθευτή	116
Κεφάλαιο 6 – Συμπεράσματα & Μελλοντικές Βελτιώσεις-Επεκτάσεις.....	117
Παράρτημα	119
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	126

Περίληψη

Η παρούσα διατριβή πραγματεύεται την δημιουργία διαδικτυακής ιστοσελίδας ενός ηλεκτρονικού καταστήματος βιβλιοπωλείου (e-Bookstore). Μέσω της ιστοσελίδας αυτής, ο κάθε πελάτης θα μπορεί να αναζητεί το βιβλίο που τον ενδιαφέρει, να βλέπει λεπτομέρειες που αφορούν αυτό και τέλος να προχωρά στην ηλεκτρονική παραγγελιά του, εφόσον αυτό είναι διαθέσιμο. Από την άλλη πλευρά, ο διαχειριστής του καταστήματος θα μπορεί να λαμβάνει τις παραγγελίες των πελατών, να τις εγκρίνει και να ενημερώνεται σε πραγματικό χρόνο για το απόθεμα του κάθε βιβλίου που βρίσκεται στις διάφορες αποθήκες που κατέχει το κατάστημα. Έκτος των παραπάνω, η διαδικτυακή εφαρμογή ενσωματώνει τις 2 παρακάτω λειτουργίες.

Η Πρώτη λειτουργία σχετίζεται με την δυνατότητα έκδοσης και εξαγωγής/λήψης του φορολογικού παραστατικού πώλησης σε αρχείο .pdf τόσο από την πλευρά του πελάτη όσο και από την πλευρά του διαχειριστή. Το παραστατικό θα είναι διαθέσιμο στο προφίλ του πελάτη μετά την έγκριση και οριστικοποίηση της παραγγελιάς από τον διαχειριστή. Τα φορολογικά παραστατικά πώλησης που χρησιμοποιήθηκαν στην εργασία είναι δύο (2) και συγκεκριμένα είναι η Απόδειξη Λιανικής Πώλησης και το Τιμολόγιο -Δελτίο Αποστολής.

Η Δεύτερη λειτουργία αφορά στη δυνατότητα του διαχειριστή να λαμβάνει πληροφορίες σχετικά με τις πωλήσεις του καταστήματος, οι οποίες θα τον βοηθήσουν στην ανατροφοδότηση του (όπως για παράδειγμα ποια βιβλία έχουν πραγματοποιήσει τις μεγαλύτερες πωλήσεις και μέσα σε ποιο χρονικό διάστημα). Η ανάπτυξη της εφαρμογής πραγματοποιήθηκε με χρήση τεχνολογιών ASP.NET.Core MVC, SQL Server και Entity Framework.

Abstraction

This postgraduate dissertation discusses the creation of an online bookstore website (e-Bookstore). Through this website, each customer will be able to search for the book, he/she is interested in, see details about it and finally proceed with his/her online order, if it is available. On the other hand, the store administrator will be able to receive customer orders, approve them and be informed in real time about the stock of each book located in the various warehouses owned by the store. Apart from the above, the online application integrates the following 2 functions.

The First function deals with the capability of issuing and exporting/downloading the sales tax document in a .pdf file both from the customer's and the administrator's side. The document will be available in the customer's profile after the order has been approved and finalized by the administrator. The sales tax documents used in the dissertation are two (2), the Retail Receipt and the Invoice-Delivery Note.

The second function refers to the capability of the administrator to receive information about the sales of the store, which will help him in his feedback. (such as which books have been best sellers and within which time period). The application was developed using ASP.NET.Core MVC, SQL Server and Entity Framework technologies.

Κεφάλαιο 1

1.1 Αντικείμενο διπλωματικής εργασίας

Η τεχνολογία τα τελευταία χρόνια εξελίσσεται με ραγδαίους ρυθμούς ασκώντας πολύ μεγάλη επίδραση σε όλους τους τομείς της ζωής μας. Ένα από τους πρώτους τομείς που υιοθέτησε εργαλεία της πληροφορικής και της τεχνολογίας δεν είναι άλλος από τον τομέα του ηλεκτρονικού εμπορίου. Η “είσοδος” της τεχνολογίας στην καθημερινότητα μας έχει αλλάξει κατά πολύ τον τρόπο με τον οποίο αλληλοεπιδρούμε με τον κόσμο γύρω μας και το εμπόριο δε θα μπορούσε να αποτελεί εξαίρεση στον κανόνα. Ταυτόχρονα, η άνοδος και η επικράτηση του ηλεκτρονικού εμπορίου έχει αλλάξει τις συνήθειες μας σαν καταναλωτές, ενώ η πρόσβαση σε παγκόσμια καταστήματα από το κινητό μας έχει επίσης αλλάξει τον τρόπο με τον οποίο πια αντιμετωπίζουμε τα φυσικά καταστήματα, τα οποία δεν μπορούν να ανταγωνιστούν τα αντίστοιχα ηλεκτρονικά.

Αντικείμενο της διατριβής αυτής αποτελεί η δημιουργία μίας διαδικτυακής εφαρμογής ηλεκτρονικού καταστήματος βιβλιοπωλείου. Πρόκειται για ανάπτυξη μιας Full-Stack εφαρμογής ιστού καθώς περιλαμβάνει τόσο το back-end κομμάτι όσο και το front-end. Επιλέχθηκε η αρχιτεκτονική Model View Controller (MVC) η οποία χρησιμοποιείται για τη δημιουργία περιβαλλόντων αλληλεπίδρασης χρήστη. Το MVC έχει υιοθετηθεί ευρέως ως αρχιτεκτονική για ανάπτυξη Web εφαρμογών σε όλες τις βασικές γλώσσες προγραμματισμού, ενώ έχουν αναπτυχθεί πολυάριθμα frameworks που επιβάλλουν το μοτίβο αυτό.

Στην εφαρμογή υπάρχουν διαφορετικοί ρόλοι χρηστών με αντίστοιχα δικαιώματα, συγκεκριμένα υπάρχουν δύο και είναι ο χρήστης με δικαιώματα διαχειριστή και ο χρήστης-πελάτης. Η εφαρμογή θα καταγράφει και θα αποθηκεύει σε μία βάση δεδομένων τα στοιχεία των χρηστών πελατών και τις παραγγελίες που έχουν πραγματοποιήσει. Μέσω της εφαρμογής ο πελάτης και συγκεκριμένα μέσα από το λογαριασμό του έχει τη δυνατότητα να παρακολουθεί το status των παραγγελιών του και όταν αυτές εγκριθούν θα μπορεί να λαμβάνει σε αρχείο pdf το αντίστοιχο φορολογικό παραστατικό.

Από την άλλη πλευρά, η εφαρμογή δίνει τη δυνατότητα στον χρήστη-διαχειριστή να παρακολουθεί το απόθεμα των βιβλίων, να διαχειρίζεται τις παραγγελίες των πελατών του και να λαμβάνει σημαντικές πληροφορίες σχετικά με το πλήθος των πωλήσεων και τα έσοδα που αποφέρουν αυτές. Ακόμα ο διαχειριστής μπορεί να αυξάνει το απόθεμα των βιβλίων και να υποβάλλει αιτήματα παραγγελίας βιβλίων προς τους Προμηθευτές.

Στόχος είναι η ανάπτυξη μιας εφαρμογής η οποία θα παρέχει αξιοπιστία στο χρήστη. Δηλαδή να μπορεί ο χρήστης να εμπιστευτεί την εφαρμογή από το αρχικό στάδιο να λαμβάνει ακριβείς πληροφορίες για τα βιβλία του καταστήματος που τον ενδιαφέρουν μέχρι το στάδιο της πραγματοποίησης των παραγγελιών και των αγορών του. Επιπλέον, η εφαρμογή αυτή πρέπει να είναι εύχρηστη και φιλική για τον χρήστη, να μπορεί δηλαδή να την χρησιμοποιήσει με ευκολία και να εξυπηρετεί τον σκοπό του άμεσα.

Με σκοπό τα παραπάνω, η κύρια τεχνολογία που χρησιμοποιήθηκε για την ανάπτυξη της ήταν το Framework ASP.NET Core 3.1 της Microsoft, σε συνδυασμό με το Entity Framework για τη δημιουργία των οντοτήτων, ενώ τα δεδομένα θα αποθηκεύονται σε μία βάση δεδομένων τύπου Microsoft SQL Server. Το εργαλείο προγραμματισμού που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής είναι το Visual Studio.

1.2 Σύνοψη διπλωματικής εργασίας

Ακολουθεί μία σύντομη επισκόπηση του παρόντος εγγράφου.

Η παρούσα εργασία αποτελείται από έξι κεφάλαια.

Στο **πρώτο κεφάλαιο** περιγράφεται το αντικείμενο της εργασίας.

Στο **δεύτερο κεφάλαιο** παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

Στο **τρίτο κεφάλαιο** γίνεται αναφορά στην ανάλυση των απαιτήσεων σε συνδυασμό με τη χρήση UML διαγραμμάτων.

Στο **τέταρτο κεφάλαιο** περιγράφονται τα δομικά χαρακτηριστικά της εφαρμογής, αναλύοντας τα μέρη της εφαρμογής και τον κώδικα που αυτά απαρτίζονται .

Στο **πέμπτο** παρουσιάζονται λεπτομερώς οι λειτουργίες της εφαρμογής.

Στο **έκτο και τελευταίο κεφάλαιο**, καταγράφονται τα συμπεράσματα και οι προοπτικές βελτίωσης της υλοποιηθείσας εφαρμογής.

Στην συνέχεια ακολουθεί παράρτημα όπου εμφανίζεται το περιεχόμενο των πινάκων της βάσης δεδομένων της διαδικτυακής εφαρμογής.

Κεφάλαιο 2 - Τεχνολογίες και Αρχιτεκτονική Σχεδίαση

2.1 Περιβάλλον Ανάπτυξης

Για την ανάπτυξη της εφαρμογής χρησιμοποιήθηκε το Microsoft Visual Studio και συγκεκριμένα η έκδοση 2022 Enterprise.

Το Visual Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) που δημιουργήθηκε από την Microsoft. Χρησιμοποιείται για την ανάπτυξη προγραμμάτων ηλεκτρονικών υπολογιστών, καθώς και ιστοσελίδων, εφαρμογών ιστού, υπηρεσιών ιστού και εφαρμογών για κινητά. Το Visual Studio χρησιμοποιεί πλατφόρμες ανάπτυξης λογισμικού της Microsoft όπως τα Windows API, τα Windows Form, το Windows Presentation Foundation, το Windows Store και το Microsoft Silverlight. Μπορεί να παράγει τόσο εγγενή κώδικα όσο και διαχειριζόμενο κώδικα.

Το Visual Studio περιλαμβάνει ένα πρόγραμμα επεξεργασίας κώδικα το οποίο υποστηρίζει το IntelliSense (το στοιχείο ολοκλήρωσης κώδικα) καθώς και το refactoring κώδικα. Το ολοκληρωμένο πρόγραμμα εντοπισμού σφαλμάτων λειτουργεί τόσο ως πρόγραμμα εντοπισμού σφαλμάτων σε επίπεδο πηγής όσο και ως εργαλείο εντοπισμού σφαλμάτων σε επίπεδο μηχανής. Άλλα ενσωματωμένα εργαλεία περιλαμβάνουν έναν προφίλ κώδικα, σχεδιαστή μορφών για την κατασκευή εφαρμογών GUI, σχεδιαστή ιστοσελίδων, σχεδιαστή τάξεων και σχεδιαστή σχήματος βάσης δεδομένων. Αποδέχεται plug-ins που βελτιώνουν τη λειτουργικότητα σχεδόν σε όλα τα επίπεδα - συμπεριλαμβανομένης της προσθήκης υποστήριξης για συστήματα ελέγχου πηγής (όπως το Subversion και το Git) και την προσθήκη νέων εργαλείων όπως οι εκδότες και οι οπτικοί σχεδιαστές για συγκεκριμένες γλώσσες ή ομάδες εργαλείων για άλλες πτυχές της ανάπτυξης λογισμικού (όπως ο πελάτης Team Foundation Server: Team Explorer).

Το Visual Studio υποστηρίζει 36 διαφορετικές γλώσσες προγραμματισμού και επιτρέπει στον επεξεργαστή κειμένου και στον εντοπισμό σφαλμάτων να υποστηρίζει (σε διαφορετικούς βαθμούς) σχεδόν οποιαδήποτε γλώσσα προγραμματισμού, υπό την προϋπόθεση ότι υπάρχει μια συγκεκριμένη γλώσσα. Οι ενσωματωμένες γλώσσες περιλαμβάνουν C, C ++, C ++ / CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML και CSS. Υποστήριξη για άλλες γλώσσες όπως Python, Ruby, Node.js και μεταξύ άλλων είναι διαθέσιμη μέσω plug-ins. Τα Java (και J #) υποστηρίχθηκαν στο παρελθόν.

2.2 Εισαγωγή ASP.NET Framework

Το ASP.NET είναι ένα πλαίσιο εφαρμογής Web που σχεδιάστηκε και αναπτύχθηκε από τη Microsoft. Είναι Framework ανοιχτού κώδικα, αποτελεί υποσύνολο του .NET Framework και είναι διάδοχος του κλασικού ASP (Active Server Pages). Το ASP.NET είναι ένα πλαίσιο εφαρμογών ιστού ανοιχτού κώδικα, από την πλευρά του διακομιστή, που επιτρέπει στους προγραμματιστές να δημιουργούν δυναμικούς ιστότοπους, εφαρμογές Ιστού και υπηρεσίες Ιστού. Επιτρέπει επίσης τη σύνταξη προγραμμάτων χρησιμοποιώντας οποιαδήποτε γλώσσα υποστήριξης .NET.

Η τελευταία τεχνολογία ASP.NET ονομάζεται **ASP.NET Core**. Είναι μια νέα υλοποίηση του ASP.NET ως αρθρωτό πλαίσιο με άλλα πλαίσια όπως το Entity Framework.

Το ASP.NET παρέχει μια σειρά από μοντέλα προγραμματισμού για την ανάπτυξη εφαρμογών. Μερικές από αυτές είναι οι εξής.

- **ASP.NET MVC** – Επιτρέπει τη δημιουργία ιστοσελίδων σύμφωνα με το μοτίβο σχεδίασης μοντέλου, προβολής και ελεγκτή
- **ASP.NET Web Pages** – Επιτρέπει την προσθήκη δυναμικού κώδικα και πρόσβασης δεδομένων απευθείας μέσα στη σήμανση HTML
- **ASP.NET Web Forms** – Επιτρέπει τη δημιουργία αρθρωτών σελίδων εκτός στοιχείων με συμβάντα διεπαφής χρήστη
- **ASP.NET Web API** – Επιτρέπει την ανάπτυξη Web API πάνω από το πλαίσιο .NET

Για την υλοποίηση της εφαρμογής χρησιμοποιήθηκε το μοντέλο ASP.NET MVC .

2.2.1 .NET Framework

Το .NET Framework χρησιμοποιείται για την ανάπτυξη εφαρμογών που έχουν ως βάση τις φόρμες , το διαδίκτυο και τις υπηρεσίες Web. Χρησιμοποιείται για την ανάπτυξη τόσο desktop εφαρμογών όσο και εφαρμογών που βασίζονται σε διακομιστή. Υπάρχει μια ποικιλία γλωσσών προγραμματισμού διαθέσιμες στην πλατφόρμα .NET , με τις κυριότερες να είναι η VB.Net και η C#. Χρησιμοποιείται για τη δημιουργία εφαρμογών για Windows, κινητές συσκευές, διαδικτυακές(Web) κ.λπ. Παρέχει πολλές λειτουργίες και υποστηρίζει επίσης βιομηχανικά πρότυπα.

Η κύρια διαφορά μεταξύ **.NET** και **ASP.NET** είναι ότι το .NET είναι ένα πλαίσιο λογισμικού που επιτρέπει την ανάπτυξη και εκτέλεση εφαρμογών ενώ το ASP.NET το **ASP.NET είναι μέρος του πλαισίου .NET** που χρησιμοποιείται για τη δημιουργία δυναμικών ιστοσελίδων, εφαρμογών ιστού και υπηρεσιών ιστού.

Το .NET Framework στοχεύει διάφορες πλατφόρμες, συμπεριλαμβανομένων κινητών, ενσωματωμένων συστημάτων, λειτουργικών συστημάτων και πρόσθετων προγραμμάτων συνδεδεμένων στα προγράμματα περιήγησης ιστού(web browser plugins). Επιπλέον, παρέχει διαλειτουργικότητα με πολλές γλώσσες προγραμματισμού, βελτιώνει την ασφάλεια, παρέχει έναν αυτόματο συλλέκτη σκουπιδιών(automatic garbage collector), παρέχει εργαλεία για ανάπτυξη, εκτέλεση και πολλές άλλες υπηρεσίες.

2.2.2 .NET Core

Το .NET Core είναι μια νέα έκδοση του .NET Framework. Είναι μια δωρεάν ανοιχτού κώδικα γενικής χρήσης πλατφόρμα ανάπτυξης εφαρμογών λογισμικού που διατηρεί η Microsoft.

Είναι ένα cross-platform framework που σημαίνει ότι λειτουργεί με Windows, Mac OS X και Linux. Το .NET Core Framework μπορεί να χρησιμοποιηθεί για τη δημιουργία μιας ποικιλίας εφαρμογών, συμπεριλαμβανομένων κινητών, desktop, Online, cloud, IoT(Internet of Things), μηχανικής μάθησης, μικροϋπηρεσιών, παιχνιδιών, κι άλλων. Το .NET Core έχει γραφτεί από την αρχή για να γίνει ένα αρθρωτό, ελαφρύ, γρήγορο και cross-platform Framework. Έχει ανανεωθεί για να κάνει το .NET γρήγορο, επεκτάσιμο και σύγχρονο. Το .NET Core είναι μια από τις μεγάλες συνεισφορές της Microsoft και κυκλοφορεί υπό την άδεια MIT.

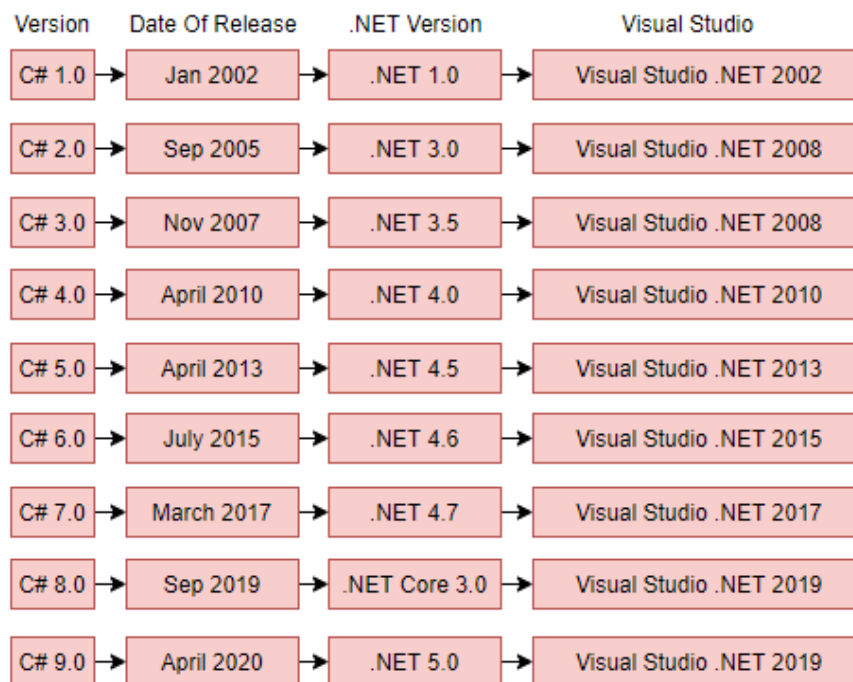
Στην επόμενη ενότητα ακολουθεί συγκριτικός πίνακας με τα χαρακτηριστικά του .NET Core και του .NET Framework.

2.2.3 Διαφορές ανάμεσα σε .NET Framework και .NET Core

	.NET Core	.NET Framework
Open Source	Net Core είναι ανοικτού κώδικα	Μερικά τμήματα του είναι ανοικτού κώδικα
Compatibility	Δημιουργήθηκε με την αρχή "build once, run anywhere" που σημαίνει ότι είναι συμβατό σε Windows, Linux & IOS	Είναι συμβατό με Windows
Application Models	Το .Net Core δεν υποστηρίζει την ανάπτυξη desktop εφαρμογών και εστιάζει περισσότερο στον διαδίκτυο (Web), τα Windows Mobile και το κατάστημα των Windows	Το Net Framework χρησιμοποιείται για την ανάπτυξη τόσο desktop εφαρμογών όσο και διαδικτυακών εφαρμογών, ενώ υποστηρίζει Windows φόρμες και εφαρμογές WPF (Windows Presentation Foundation).
Performance and Scalability	Το NET Core προσφέρει υψηλή απόδοση και επεκτασιμότητα.	Το Net Framework είναι λιγότερο αποτελεσματικό σε σύγκριση με το .Net Core όσον αφορά την απόδοση και την επεκτασιμότητα των εφαρμογών.
Android Development	Το .NET Core είναι συμβατό με πλατφόρμες εφαρμογών για κινητές συσκευές ανοικτού κώδικα, π.χ. Xamarin, μέσω της .NET Standard Library. Οι προγραμματιστές χρησιμοποιούν τα εργαλεία του Xamarin για να διαμορφώσουν την εφαρμογή για κινητά για συγκεκριμένες κινητές συσκευές όπως iOS, Android και τηλέφωνα Windows.	Το NET Framework δεν υποστηρίζει κανένα framework για την ανάπτυξη εφαρμογών που να αφορούν κινητές συσκευές
CLI Tools	Το .NET Core παρέχει ελαφριούς editors και εργαλεία γραμμής εντολών για όλες τις υποστηριζόμενες πλατφόρμες.	Το .Net Framework είναι βαρύ για τη διεπαφή γραμμής εντολών (CLI) και οι προγραμματιστές προτιμούν να εργάζονται με ελαφριά διεπαφή γραμμής εντολών.
Security	.NET Core δεν παρέχει λειτουργίες όπως το Code Access Security	.NET Framework περιλαμβάνει εξ αρχής τη λειτουργία Code Access Security

2.3 Γλώσσα προγραμματισμού C#

Η γλώσσα προγραμματισμού C# προφέρεται ως "C-Sharp". Είναι μια αντικειμενοστραφή γλώσσα προγραμματισμού που παρέχεται από τη Microsoft και εκτελείται σε περιβάλλον .Net Framework. Ο Anders Hejlsberg είναι γνωστός ως ο ιδρυτής της γλώσσας C#. Η C# έχει εξελιχθεί πολύ από την πρώτη της κυκλοφορία που ήταν το έτος 2002. Πρωτοεμφανίστηκε με την έκδοση .NET Framework 1.0 και κατά τη συγγραφή της παρούσας εργασίας, η τρέχουσα έκδοση της C# είναι η 11.0 .



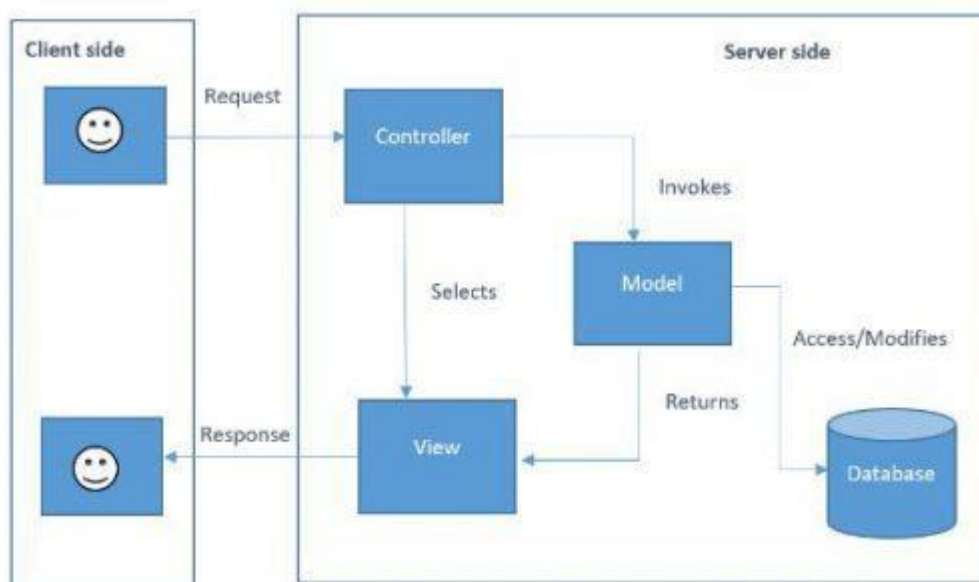
Εικόνα 1

Βασίζεται στις γλώσσες προγραμματισμού C++ και Java, αλλά έχει πολλές πρόσθετες επεκτάσεις που χρησιμοποιούνται για την εκτέλεση της προσέγγισης προσανατολισμένης σε τμηματικό προγραμματισμό. Παρακάτω παρατίθενται τα είδη των εφαρμογών που μπορούν να δημιουργηθούν με χρήση της γλώσσας προγραμματισμού C# .

- Desktop application – Desktop εφαρμογές με γραφικό περιβάλλον (κουμπιά, μπάρα πλοήγησης κλπ.)
- Desktop console apps which respond to the end user through a command line console-εφαρμογές κονσόλας με χρήση γραμμής εντολών
- Web applications – εφαρμογές ιστού
- Applications that can talk to databases and web services -εφαρμογές που επικοινωνούν με βάσεις δεδομένων και με υπηρεσίες διαδικτύου
- Applications that run on mobile devices – εφαρμογές που μπορούν να λειτουργήσουν σε κινητές συσκευές

2.4 Αρχιτεκτονική MVC (Model -View -Controller)

Το πλαίσιο Model-View-Controller (MVC) είναι ένα αρχιτεκτονικό μοτίβο που διαχωρίζει μια εφαρμογή σε τρία κύρια λογικά στοιχεία Model, View και Controller. Εξ ου και η συντομογραφία MVC. Κάθε στοιχείο αρχιτεκτονικής έχει δημιουργηθεί για να χειρίζεται ένα συγκεκριμένο τμήμα ανάπτυξης μιας εφαρμογής. Το MVC διαχωρίζει την επιχειρηματική λογική και το επίπεδο παρουσίασης. Χρησιμοποιήθηκε παραδοσιακά για γραφικό περιβάλλον χρήστη (GUIs) σε επιτραπέζιους (desktop) υπολογιστές. Σήμερα, η αρχιτεκτονική MVC στην τεχνολογία Web έχει γίνει δημοφιλής για τον σχεδιασμό εφαρμογών Ιστού καθώς και για εφαρμογές κινητών συσκευών.



Εικόνα 2

2.4.1 Τρέχουσες τάσεις του MVC

Οι ιστότοποι που χρησιμοποιούμε σήμερα είναι απίστευτα πολύπλοκες εφαρμογές οι οποίες έχουν αναπτυχθεί από χιλιάδες προγραμματιστές τα τελευταία 20 χρόνια, από απλές σελίδες HTML με μερικό κώδικα CSS έως πολύπλοκες εφαρμογές οι οποίες απαιτούν μεγάλη προσπάθεια. Για να απλοποιήσουν τον κώδικα και να διευκολύνουν την εργασία τους, οι προγραμματιστές χρησιμοποιούν διάφορα μοτίβα για να σχεδιάσουν τα έργα τους.

Το MVC, γνωστό και ως (Model View Controller), είναι το πιο δημοφιλές από αυτά τα Framework. Χρησιμοποιείται για να χωρίσει μια μεγάλη εφαρμογή σε μικρότερα τμήματα. Κάθε τμήμα του έργου έχει σχεδιαστεί για να απεικονίζει τη δική του λειτουργία. Η αρχιτεκτονική MVC γίνεται τόσο δημοφιλής που πλέον τα περισσότερα δημοφιλή Framework ακολουθούν το μοτίβο σχεδιασμού

MVC για την ανάπτυξη των εφαρμογών. Μερικά από τα δημοφιλή Frameworks που ακολουθούν το πρότυπο MVC Design είναι :

- ✓ **JAVA Frameworks:** Sprint, Spring Boot.
- ✓ **Python Framework:** Django.
- ✓ **NodeJS (JavaScript):** ExpressJS.
- ✓ **PHP Framework:** Cake PHP, Phalcon, PHPixie.
- ✓ **Ruby:** Ruby on Rails.
- ✓ **Microsoft.NET:** ASP.net MVC.
- ✓ **Perl:** Catalyst. etc.

2.4.2 Αναλυτικά τα μέρη του MVC Framework

Σε γενικές γραμμές γνωρίζουμε ότι το MVC αποτελείται από 3 στοιχεία. Model, View και Controller. Όμως υπάρχει ακόμα ένα κύριο συστατικό, το οποίο είναι εξίσου σημαντικό για να λειτουργήσει το Framework. Και αυτό είναι το router . Άρα μπορούμε να πούμε ότι υπάρχουν συνολικά 4 στοιχεία του MVC.

1. View – Where the user interacts , το οποίο αλληλοεπιδρά με τον χρήστη
2. Router – Handles the Request where to process. Χειρίζεται το αίτημα για επεξεργασία
3. Controller – Process the request and send the response to view. Επεξεργάζεται το αίτημα και απαντά στο View
4. Model – Middleware that handles the database operations. Ως ενδιάμεσο λογισμικό χειρίζεται τις λειτουργίες της βάσης δεδομένων

➤ Model

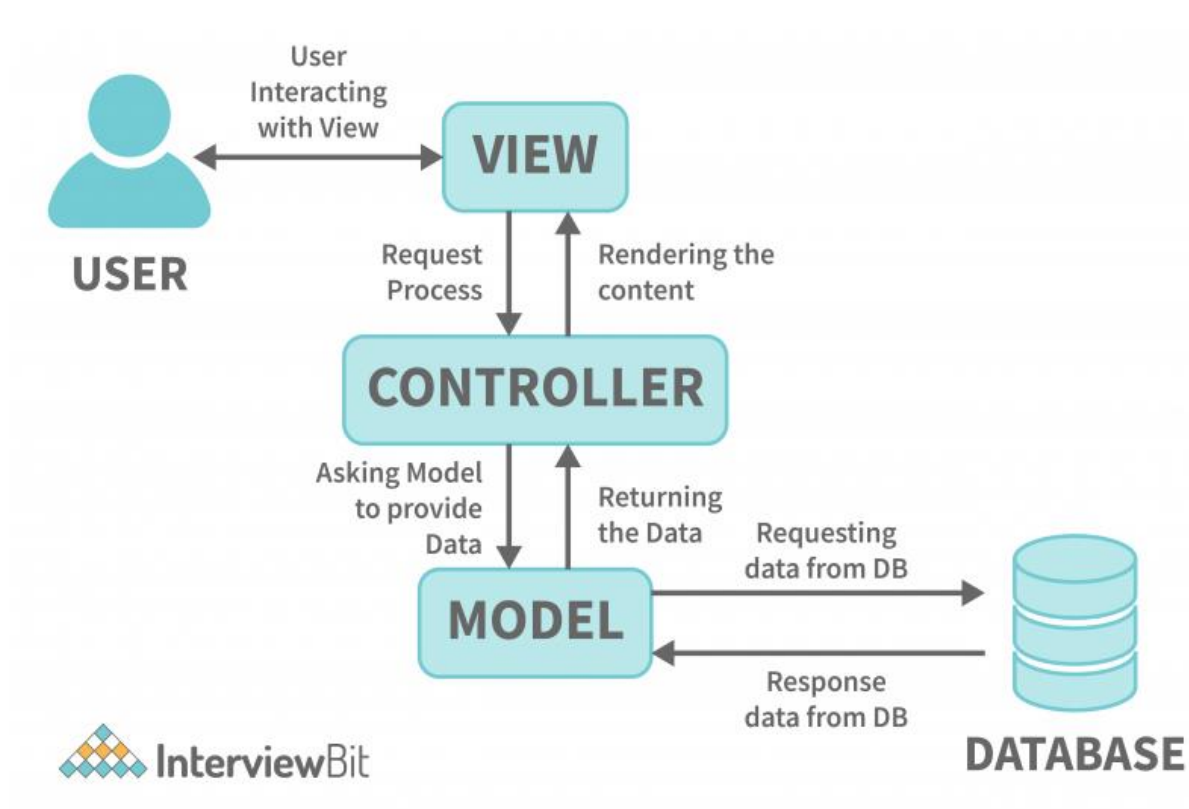
Ένα στοιχείο μοντέλου αποθηκεύει δεδομένα και λογική. Για παράδειγμα, ένα αντικείμενο Controller θα ανακτήσει πληροφορίες πελάτη από μια βάση δεδομένων. Τα δεδομένα μεταφέρονται μεταξύ των στοιχείων του ελεγκτή ή μεταξύ των στοιχείων της επιχειρηματικής λογικής. Αυτό χειρίζεται δεδομένα και τα στέλνει πίσω στη βάση δεδομένων ή χρησιμοποιείται επίσης για την προβολή και απεικόνιση των ίδιων πληροφοριών. Επιπλέον, απαντά στα αιτήματα των views και δέχεται οδηγίες από τον ελεγκτή που του επιτρέπουν να ενημερώνεται μόνο του. Είναι επίσης το χαμηλότερο επίπεδο του προτύπου που είναι υπεύθυνο για τη διατήρηση των δεδομένων.

➤ View

Ουσιαστικά, η προβολή αντιπροσωπεύει τον τρόπο με τον οποίο παρουσιάζονται τα δεδομένα στην εφαρμογή. Το στοιχείο View(προβολή) χρησιμοποιείται για τη λογική του User Interface (UI) της εφαρμογής. Οι προβολές δημιουργούνται με βάση τα δεδομένα που συλλέγονται από το μοντέλο. Ζητώντας πληροφορίες από το μοντέλο, παρουσιάζεται στον χρήστη η τελική οθόνη (UI). Εκτός από την προβολή των δεδομένων όπως γραφήματα, διαγράμματα και πίνακες, η προβολή εμφανίζει επίσης δεδομένα από άλλες πηγές. Όλα τα στοιχεία διεπαφής χρήστη, όπως πλαίσια κειμένου, αναπτυσσόμενα μενού κ.λπ. θα εμφανίζονται σε κάθε προβολή.

➤ Controller

Οι ελεγκτές είναι εκείνα τα στοιχεία μιας εφαρμογής που χειρίζονται την αλληλεπίδραση με τον χρήστη. Η είσοδος του χρήστη ερμηνεύεται από τον ελεγκτή, προκαλώντας αλλαγή του μοντέλου και της προβολής με βάση τις πληροφορίες που λαμβάνει. Επικοινωνώντας ο ελεγκτής (controller) με το αντίστοιχο view, ένας χρήστης μπορεί να αλλάξει την εμφάνιση της προβολής (για παράδειγμα, να κάνει κύλιση σε ένα έγγραφο) και να ενημερώσει την κατάσταση του συνδεδεμένου μοντέλου (για παράδειγμα, αποθήκευση ενός εγγράφου).



Εικόνα 3

➤ MVC Routing

Στο MVC, η δρομολόγηση είναι μια διαδικασία αντιστοίχισης του αιτήματος του προγράμματος περιήγησης στην ενέργεια του ελεγκτή και επιστροφής απόκρισης. Κάθε εφαρμογή MVC έχει προεπιλεγμένη δρομολόγηση για τον προεπιλεγμένο HomeController. Μπορούμε να ορίσουμε προσαρμοσμένη δρομολόγηση για τον κάθε νέο ελεγκτή που δημιουργούμε.

2.4.3 Πλεονεκτήματα Αρχιτεκτονικής MVC

- Οργανώνει εφαρμογές Web μεγάλου μεγέθους

Καθώς υπάρχει διαχωρισμός του κώδικα μεταξύ των τριών επιπέδων, γίνεται εξαιρετικά εύκολο να διαιρεθεί και να οργανωθεί η λογική εφαρμογών ιστού σε εφαρμογές μεγάλης κλίμακας (τις οποίες απαιτείται να διαχειρίζονται μεγάλες ομάδες προγραμματιστών). Το κύριο πλεονέκτημα της χρήσης τέτοιων πρακτικών κώδικα είναι ότι βοηθά στην εύρεση συγκεκριμένων τμημάτων κώδικα γρήγορα και επιτρέπει την εύκολη προσθήκη νέων λειτουργιών.

- Εύκολα τροποποιήσιμο

Η χρήση της μεθοδολογίας MVC επιτρέπει την εύκολη τροποποίηση ολόκληρης της εφαρμογής. Η προσθήκη/ενημέρωση του νέου τύπου προβολών απλοποιείται στο μοτίβο MVC (καθώς μια ενότητα είναι ανεξάρτητη από τις άλλες ενότητες). Έτσι, τυχόν αλλαγές σε ένα συγκεκριμένο τμήμα της εφαρμογής δεν θα επηρεάσουν ποτέ ολόκληρη την αρχιτεκτονική. Αυτό, με τη σειρά του, θα συμβάλει στην αύξηση της ευελιξίας και της επεκτασιμότητας της εφαρμογής.

- Ταχύτερη Διαδικασία Ανάπτυξης

Καθώς υπάρχει διαχωρισμός του κώδικα μεταξύ των τριών επιπέδων, η ανάπτυξη εφαρμογών Ιστού χρησιμοποιώντας το μοντέλο MVC επιτρέπει σε έναν προγραμματιστή να εργαστεί σε μια συγκεκριμένη ενότητα (ας πούμε, την προβολή) ενώ ένας άλλος μπορεί να εργαστεί σε οποιοδήποτε άλλο τμήμα (ας πούμε, τον ελεγκτή) ταυτόχρονα. Αυτό επιτρέπει την εύκολη εφαρμογή της επιχειρηματικής λογικής καθώς βοηθά στην τετραπλή επιτάχυνση της διαδικασίας ανάπτυξης. Έχει παρατηρηθεί ότι σε σύγκριση με άλλα μοντέλα ανάπτυξης, το μοντέλο MVC καταλήγει να εμφανίζει υψηλότερες ταχύτητες ανάπτυξης (έως και τρεις φορές).

- Υποστηρίζει ασύγχρονη επίκληση μεθόδου (AMI)

Δεδομένου ότι η αρχιτεκτονική MVC λειτουργεί καλά με τη JavaScript και τα πλαίσια της, δεν αποτελεί έκπληξη το γεγονός ότι υποστηρίζει επίσης τη χρήση της Ασύγχρονης Επίκλησης Μεθόδου (AMI), επιτρέποντας στους προγραμματιστές να δημιουργούν εφαρμογές ιστού με ταχύτερη φόρτωση. Σημαίνει ότι οι εφαρμογές MVC μπορούν να λειτουργούν ακόμη και με αρχεία PDF, προγράμματα περιήγησης για συγκεκριμένους ιστότοπους, καθώς και για desktop widgets.

- Εύκολος σχεδιασμός και συντήρηση

Το παράδειγμα MVC είναι χρήσιμο κατά την αρχική φάση σχεδιασμού της εφαρμογής, επειδή δίνει στον προγραμματιστή μια περίληψη για το πώς να τακτοποιήσει τις ιδέες του σε πραγματικό κώδικα. Είναι επίσης ένα εξαιρετικό εργαλείο που βοηθά στον περιορισμό της αντιγραφής κώδικα και επιτρέπει την εύκολη συντήρηση της εφαρμογής.

- Επιστρέφει δεδομένα χωρίς μορφοποίηση

Επιστρέφοντας μη μορφοποιημένα δεδομένα, το πλαίσιο MVC μας δίνει τη δυνατότητα να δημιουργήσουμε τη δική μας μηχανή προβολής. Για παράδειγμα, οποιοσδήποτε τύπος δεδομένων μπορεί να μορφοποιηθεί χρησιμοποιώντας HTML, αλλά με το πλαίσιο MVC, μπορούμε επίσης να μορφοποιήσουμε τα δεδομένα χρησιμοποιώντας το πρόγραμμα προβολής Macromedia Flash ή

Dream. Είναι χρήσιμο για τους προγραμματιστές επειδή τα ίδια στοιχεία μπορούν να επαναχρησιμοποιηθούν με οποιαδήποτε διεπαφή.

- Πολλαπλές προβολές

Στην αρχιτεκτονική MVC, η ανάπτυξη διαφορετικών στοιχείων προβολής για το στοιχείο του μοντέλου μας είναι εύκολα εφικτή. Μας δίνει τη δυνατότητα να αναπτύξουμε διαφορετικά στοιχεία προβολής, περιορίζοντας έτσι την αντιγραφή κώδικα καθώς διαχωρίζει τα δεδομένα και την επιχειρηματική λογική.

- Πλατφόρμα φιλική προς το SEO (Search Engine Optimization)

Η πλατφόρμα MVC υποστηρίζει σε μεγάλο βαθμό την ανάπτυξη διαδικτυακών εφαρμογών φιλικών προς το SEO. Για να δημιουργήσουμε περισσότερες επισκέψεις από μια συγκεκριμένη εφαρμογή, το MVC παρέχει έναν εύκολο τρόπο ανάπτυξης URL που είναι φιλικές προς το SEO RESTful.

Ο σχεδιασμός ιστοσελίδων SEO σχεδιάζει και αναπτύσσει ιστοσελίδες για να τους κάνει φιλικούς προς τις μηχανές αναζήτησης. Το να κάνουμε έναν ιστότοπο φιλικό προς το SEO σημαίνει ότι η Google και άλλες μηχανές αναζήτησης μπορούν να ανιχνεύουν αποτελεσματικά κάθε σελίδα στον ιστότοπο, να ερμηνεύουν αποτελεσματικά το περιεχόμενο και να το καταχωρούν στη βάση δεδομένων τους.

- Υποστηρίζει TDD (test-driven development)

Ένα σημαντικό πλεονέκτημα του μοτίβου MVC είναι ότι απλοποιεί κατά πολύ τη διαδικασία δοκιμής. Διευκολύνει τον εντοπισμό σφαλμάτων σε εφαρμογές μεγάλης κλίμακας, καθώς πολλαπλά επίπεδα ορίζονται δομικά και γράφονται σωστά στην εφαρμογή. Έτσι, η ανάπτυξη μιας εφαρμογής πραγματοποιείται απρόσκοπτα με χρήση τμηματικών δοκιμών .

Έτσι, το μοτίβο σχεδιασμού MVC είναι μια εξαιρετική προσέγγιση για την ανάπτυξη εφαρμογών Ιστού. Σήμερα, υπάρχουν αρκετές επιχειρήσεις που επιλέγουν την ανάπτυξη τέτοιων εφαρμογών με βάση την αρχιτεκτονική MVC.

2.5 Τεχνολογίες διεπαφής χρήστη (front-end)

Οι τεχνολογίες που αφορούν τη λειτουργία του UI (User Interface) δηλαδή το μέρος του προγράμματος που αλληλοεπιδρά ο χρήστης.

Είναι HTML5 (Hyper Text Markup Language), CSS3 (Cascading Style Sheets) και JavaScript.

2.5.1 Html

Η HTML είναι το ακρωνύμιο Hyper Text Markup Language δηλαδή Γλώσσα Χαρακτηρισμού Υπερ-Κειμένου, το οποίο χρησιμοποιείται για τη ανάπτυξη ιστοσελίδων και εφαρμογών ιστού. Η Html είναι η κύρια γλώσσα σήμανσης για τις ιστοσελίδες, και τα στοιχεία της είναι τα βασικά δομικά στοιχεία των ιστοσελίδων.

2.5.2 Csst (Cascading Style Sheets)

Το **CSS** σημαίνει Cascading Style Sheets. Είναι η γλώσσα για την περιγραφή της παρουσίασης των ιστοσελίδων, συμπεριλαμβανομένων των χρωμάτων, της διάταξης και των γραμματοσειρών, καθιστώντας έτσι τις ιστοσελίδες εμφανίσιμες στους χρήστες.

Η CSS είναι μια γλώσσα που καθορίζει τη σχεδίαση και τη διάταξη των ιστοσελίδων. Με άλλα λόγια, το CSS ελέγχει πώς φαίνονται οι ιστοσελίδες όταν φορτώνονται σε ένα πρόγραμμα περιήγησης. Ονομάζουμε αυτό το σχέδιο και τη διάταξη «στυλ» της σελίδας. Η CSS είναι η τυπική γλώσσα για το στυλ και συνήθως λειτουργεί σε συνδυασμό με την HTML .

Πλεονεκτήματα χρήσης CSS:

Λιγότερη χρήση κώδικα: Οι προγραμματιστές μπορούν να χρησιμοποιήσουν το CSS για να εφαρμόσουν το ίδιο στυλ σε πολλές σελίδες και στοιχεία σελίδας σε έναν ιστότοπο, εξοικονομώντας τεράστιο χρόνο και μειώνοντας την πιθανότητα σφαλμάτων. Η τροποποίηση ενός στυλ σε ολόκληρο τον ιστότοπο απαιτεί αλλαγή μόνο ενός αποσπάσματος κώδικα.

Περισσότερες επιλογές στυλ: Μπορούμε να κάνουμε πολλά με το CSS, πολύ περισσότερα από όσα επέτρεπε το αρχικό σύστημα HTML. Με σαφές όραμα, τεχνογνωσία CSS και λίγη υπομονή, μπορούμε να προσαρμόσουμε έναν ιστότοπο ακριβώς στις προτιμήσεις μας.

Τυποποίηση: Δεδομένου ότι η CSS είναι η ενιαία γλώσσα για το στυλ ιστοσελίδων, ένας προγραμματιστής ή σχεδιαστής μπορεί να κατανοήσει το στυλ οποιουδήποτε ιστότοπου κοιτάζοντας τα αρχεία CSS.

Καλύτερη απόδοση: Το CSS μειώνει την ποσότητα του επαναλαμβανόμενου κώδικα στυλ. Λιγότερος κώδικας σημαίνει μικρότερα αρχεία και μικρότερα αρχεία σημαίνουν ταχύτερους χρόνους φόρτωσης σελίδας.

2.5.3 Bootstrap

Το Bootstrap είναι ένα ισχυρό πλαίσιο εργασίας διεπαφών (front-end Framework) το οποίο χρησιμοποιείται για την ταχύτερη και ευκολότερη ανάπτυξη ιστοσελίδων και διαδικτυακών εφαρμογών. Είναι δωρεάν και ανοιχτού κώδικα. Βασίζεται στις τεχνολογίες HTML, CSS, JavaScript και είναι συμβατό με όλους τους φυλλομετρητές. Επιλέχτηκε για την υλοποίηση της εφαρμογής γιατί είναι κατάλληλο για νέους προγραμματιστές αφού είναι πολύ εύκολο στην εγκατάσταση και στη χρήση. Προσφέρει ένα σταθερό και ευέλικτο σύστημα διάταξης (grid system) το οποίο είναι αποκρίσιμο (responsive) σε όλες τις διαστάσεις του φυλλομετρητή. Συνεπώς, το Bootstrap είναι κατάλληλο για ανάπτυξη εφαρμογών φιλικών προς τα κινητά (mobile phones) και τις ταμπλέτες (tablets). Περιλαμβάνει πλήθος από συστατικά ιστού (Web components), όπως κουμπιά, μπάρες, φόρμες, dropdown και είναι επεκτάσιμο με τη βοήθεια των JavaScript plug-ins που διαθέτει. Παρέχει πολλές κλάσεις CSS οι οποίες μπορούν να χρησιμοποιηθούν σε πίνακες, φόρμες, κουμπιά, εικόνες και άλλα στοιχεία ώστε να διευκολύνουν σε μεγάλο βαθμό την παραγωγή των ζητούμενων αποτελεσμάτων. Τέλος, έχει πολύ καλό εγχειρίδιο (documentation) και μια μεγάλη και δυναμική κοινότητα χρηστών.

2.5.4 JavaScript

Η JavaScript είναι η πιο δημοφιλής γλώσσα προγραμματισμού και χρησιμοποιείται από προγραμματιστές σε όλο τον κόσμο για τη δημιουργία δυναμικού και διαδραστικού περιεχομένου Ιστού. Είναι μια ερμηνευμένη γλώσσα προγραμματισμού με αντικειμενοστραφείς δυνατότητες.

Η JavaScript είναι μια γλώσσα προγραμματισμού που βασίζεται σε κείμενο και χρησιμοποιείται τόσο από την πλευρά του πελάτη όσο και από την πλευρά του διακομιστή και μας επιτρέπει να κάνουμε τις ιστοσελίδες διαδραστικές. Όπου η HTML και η CSS είναι γλώσσες που δίνουν δομή και στυλ σε ιστοσελίδες, η JavaScript δίνει στις ιστοσελίδες διαδραστικά στοιχεία που προσελκύουν έναν χρήστη. Συνηθισμένα παραδείγματα JavaScript που συναντάμε καθημερινά περιλαμβάνουν το πλαίσιο αναζήτησης στο **Amazon**, ένα βίντεο ανακεφαλαίωσης ειδήσεων ενσωματωμένο στους **The New York Times** ή την ανανέωση της ροής σας στο **Twitter**.

Πριν από τη JavaScript, οι ιστοσελίδες κατασκευάζονταν μόνο με HTML και CSS. Το HTML και το CSS είναι ικανά να δημιουργούν μόνο στατικές σελίδες που μπορούν να διαμορφωθούν με χρήση υπερσυνδέσμων. Η ενσωμάτωση JavaScript βελτιώνει την εμπειρία χρήστη της ιστοσελίδας μετατρέποντάς την από στατική σελίδα σε δυναμική. Ανακεφαλαιώνοντας, θα μπορούσαμε να πούμε ότι η JavaScript προσθέτει συμπεριφορά σε ιστοσελίδες.

Μερικοί από τους πιο δημοφιλείς ιστότοπους που είναι κατασκευασμένοι με JavaScript, είναι οι **Google**, **YouTube** και **Facebook**.

Η JavaScript επιτρέπει στους προγραμματιστές να εφαρμόσουν λειτουργίες όπως:

- Εμφάνιση και απόκρυψη μενού ή πληροφοριών
- Προσθήκη hover εφέ
- Δημιουργία γκαλερί εικόνων σε μορφή καρουζέλ
- Μεγέθυνση ή σμίκρυνση μιας εικόνας
- Αναπαραγωγή ήχου ή βίντεο σε μια ιστοσελίδα
- Προσθήκη κινούμενων εικόνων
- Δημιουργία αναπτυσσόμενων μενού και μενού σε hamburger-style

Με την εκτενή συλλογή πλαισίων της JavaScript, οι προγραμματιστές μπορούν να δημιουργήσουν αποτελεσματικά εφαρμογές για κινητά και τον ιστό. Τα πλαίσια είναι βιβλιοθήκες προγραμμένου κώδικα JavaScript που χρησιμοποιούν οι προγραμματιστές για τυπικές λειτουργίες. Μπορούμε να σκεφτούμε ένα πλαίσιο JavaScript σαν ένα σχεδιάγραμμα.

Τα πιο δημοφιλή πλαίσια εφαρμογών JavaScript είναι :

- **React**: μια συλλογή JavaScript που δημιουργήθηκε για τη δημιουργία διεπαφών χρήστη για εφαρμογές Web.
- **React Native**: μια συλλογή που επιτρέπει στους προγραμματιστές να δημιουργούν εφαρμογές για κινητά με JavaScript.
- **Node.js**: μια συλλογή που επιτρέπει αμφίδρομη επικοινωνία με διακομιστές για την ανταλλαγή δεδομένων.

Το **Facebook** δημιούργησε τα **React** και **React Native**, τα οποία τροφοδοτούν αυτήν τη στιγμή μερικές από τις πιο ισχυρές εφαρμογές κοινωνικών μέσων στον κόσμο. Το **React** χρησιμοποιείται από ιστότοπους όπως το **Uber**, το **Pinterest**, το **Netflix**, το **Instagram**, το **Amazon**, το **Twitter**, το

Udemy και άλλα. Facebook, Instagram, Shopify, Words with Friends, Skype, Pinterest, Discord, Tesla και Uber Eats χρησιμοποιούν το React Native.

2.5.5 jQuery βιβλιοθήκη

Ως προγραμματιστές ιστού, συναντάμε HTML, CSS και JavaScript αρκετά νωρίς. Καθώς οι σελίδες μας γίνονται πιο περίπλοκες, η απαιτούμενη χρήση της JavaScript γίνεται πιο πολύπλοκη στη σύνταξη της. Εκεί μπαίνουν οι βιβλιοθήκες. Μια βιβλιοθήκη JavaScript είναι ένα αρχείο που περιέχει λειτουργίες για την εκτέλεση κοινών εργασιών, την επαναχρησιμοποίηση κώδικα και την απλοποίηση του κώδικα JavaScript συνολικά.

Το jQuery είναι μια από τις πιο δημοφιλείς βιβλιοθήκες JavaScript . Το jQuery διευκολύνει την ανάπτυξη ιστού ξεπερνώντας όλα τα «πράγματα» που κάνουν την JavaScript δύσκολη στη χρήση. Με αυτό, μπορούμε να καλέσουμε απλές μεθόδους αντί να ξαναγράψουμε μπλοκ εργασιών.

Είναι μια ελαφριά βιβλιοθήκη JavaScript ανοιχτού κώδικα που μας βοηθά να δημιουργήσουμε διαδραστικές ιστοσελίδες με κινούμενα σχέδια, οπτικά εφέ και προηγμένες λειτουργίες. Είναι η πιο δημοφιλής βιβλιοθήκη JavaScript, που χρησιμοποιείται από εκατομμύρια ιστότοπους παγκοσμίως.

Το μόντο του jQuery είναι «γράψτε λιγότερα, κάντε περισσότερα», και αυτό ακριβώς κάνει. Το jQuery είναι ένα επαναχρησιμοποιήσιμο κομμάτι κώδικα που μας δίνει τη δυνατότητα να κάνουμε περισσότερα με τη JavaScript.

Το jQuery αναλαμβάνει κοινές εργασίες JavaScript και τις τοποθετεί σε μεθόδους. Στη συνέχεια οι προγραμματιστές, αντί να γράφουν όλο αυτόν τον κώδικα με το χέρι, μπορούν απλώς να καλέσουν αυτές τις μεθόδους, και αυτό αναλαμβάνει για τα όλα τα υπόλοιπα. Αυτές οι συντομεύσεις επιτρέπουν στους χρήστες να γράφουν καλύτερο κώδικα και να είναι πιο παραγωγικοί.

Σύμφωνα με πολλές εκτιμήσεις ,περισσότερο από το 50% όλων των ιστοτόπων με Javascript χρησιμοποιούν την βιβλιοθήκη jQuery .

Τα κύρια χαρακτηριστικά του jQuery περιλαμβάνουν:

- Χειρισμό events
- Χειρισμό DOM (Document Object Model)
- Κινούμενα σχέδια και εφέ
- Πλαίσιο AJAX

2.5.6 Σύνταξη Razor για ASP.NET MVC

Το Razor είναι μια σύνταξη σήμανσης για την ενσωμάτωση .NET κώδικα σε ιστοσελίδες. Η σύνταξη Razor αποτελείται από σήμανση Razor, C# και HTML. Τα αρχεία που περιέχουν Razor έχουν γενικά κατάληξη αρχείου σε .cshtml. Η σύνταξη του Razor είναι παρόμοια με τις μηχανές προτύπων διαφόρων πλαισίων εφαρμογών μίας σελίδας JavaScript (SPA), όπως Angular, React, VueJs και Svelte.

Το Razor υποστηρίζει C# και χρησιμοποιεί το σύμβολο @ για τη μετάβαση από HTML σε C#. Το Razor επεξεργάζεται expressions C# και τις προβάλλει σε HTML.

Το Razor είναι μια απλή σύνταξη προγραμματισμού για την ενσωμάτωση κώδικα πλευράς διακομιστή σε ιστοσελίδες.

Με λίγα λόγια το Razor μας δίνει την δυνατότητα να ενσωματώσουμε κώδικα Visual Basic και C # σε ιστοσελίδες.

Όταν ο διακομιστής διαβάζει την σελίδα, εκτελεί πρώτα τον κώδικα Razor, πριν στείλει τη σελίδα HTML στο πρόγραμμα περιήγησης. Ο κώδικας που εκτελείται στον διακομιστή μπορεί να εκτελέσει εργασίες που δεν μπορούν να γίνουν στο πρόγραμμα περιήγησης, για παράδειγμα πρόσβαση σε μια βάση δεδομένων διακομιστή. Ο κώδικας διακομιστή μπορεί να δημιουργήσει δυναμικό περιεχόμενο HTML αμέσως, προτού σταλεί στο πρόγραμμα περιήγησης. Παρατηρώντας το πρόγραμμα περιήγησης, το HTML που δημιουργείται από τον κώδικα διακομιστή δεν διαφέρει από το στατικό περιεχόμενο HTML.

2.6 Σχεδιασμός της Βάσης Δεδομένων

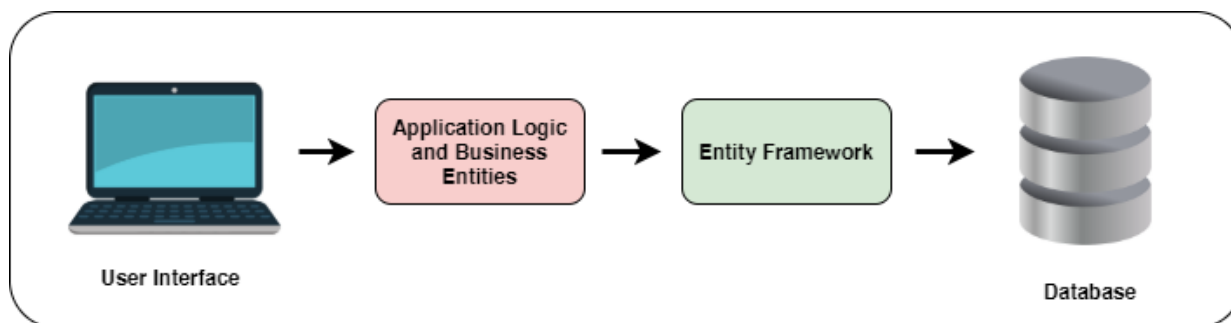
2.6.1 Βάση Δεδομένων Microsoft SQL Server

Ο Microsoft SQL Server είναι ένα από τα κύρια συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων στην αγορά που εξυπηρετεί ένα ευρύ φάσμα εφαρμογών λογισμικού για επιχειρηματική ευφυΐα και ανάλυση σε εταιρικά περιβάλλοντα. Βασισμένο στη γλώσσα Transact-SQL, ενσωματώνει ένα σύνολο επεκτάσεων προγραμματισμού τυπικής γλώσσας και η εφαρμογή του είναι διαθέσιμη για χρήση τόσο σε premises όσο και στο cloud. Ο Microsoft SQL Server είναι ιδανικός για την αποθήκευση όλων των επιθυμητών πληροφοριών σε σχεσιακές βάσεις δεδομένων, καθώς και για τη διαχείριση τέτοιων δεδομένων χωρίς προβλήματα, χάρη στην οπτική του διεπαφή, τις επιλογές και τα εργαλεία που διαθέτει. Αυτό είναι ζωτικής σημασίας, ειδικά για ιστότοπους που έχουν τη δυνατότητα εγγραφής χρηστών για σύνδεση.

2.6.2 Entity Framework – Διαχείριση Δεδομένων

Entity Framework είναι ένα ανοικτού κώδικα πλαίσιο (framework) για .NET εφαρμογές που δημιουργήθηκε από την Microsoft. Δίνει τη δυνατότητα στους προγραμματιστές να δουλεύουν με data χρησιμοποιώντας αντικείμενα (objects) που προέρχονται από ειδικές κλάσεις χωρίς να χρειαστεί να δώσουν ιδιαίτερη προσοχή σε ποια στήλη του πίνακα ή σε ποιον πίνακα της βάσης δεδομένων θα αποθηκευτούν τα δεδομένα. Με αυτό το Framework, οι προγραμματιστές μπορούν να εργαστούν σε υψηλότερο επίπεδο αφαίρεσης όταν χειρίζονται τα data και μπορούν να δημιουργούν εφαρμογές με χρήση λιγότερου κώδικα σε σχέση με τον παραδοσιακό τρόπο ανάπτυξης εφαρμογών.

Το Entity Framework βρίσκεται μεταξύ των Επιχειρηματικών Οντοτήτων και της Βάσης Δεδομένων. Το Εννοιολογικό του μοντέλο, μας επιτρέπει να αναζητούμε πληροφορίες από τη βάση δεδομένων και επιτρέπει την εύκολη δημιουργία αντικειμένων για τον χειρισμό των δεδομένων.



Εικόνα 4

Για την υλοποίηση του project χρησιμοποιήθηκε η έκδοση Entity Framework Core 3.1 . Χαρακτηριστικό της έκδοσης Core είναι ότι έχει συμβατότητα με τα λειτουργικά συστήματα Windows, Mac και Linux σε αντίθεση με παλαιότερες εκδόσεις οι οποίες λειτουργούσαν μόνο σε Windows. Επιπρόσθετα η έκδοση Core έχει πρόσβαση σε πολλά διαφορετικά είδη βάσεων μέσω βιβλιοθηκών(plug-in libraries) που καλούνται πάροχοι (providers) . Συγκεκριμένα , ο Nuget-Package ο οποίος εφόσον εγκατασταθεί ,είναι συμβατός με βάσεις PostgreSQL, SQL Server, MySQL ,SQLite, SQL Compact, Oracle DB, Azure Cosmos DB , Jet (Microsoft Access).

2.6.3 Χαρακτηριστικά του Entity Framework

Παρακάτω καταγράφονται σημαντικά χαρακτηριστικά του Entity Framework:

- **Μοντελοποίηση (Modelling)**

Επιτρέπει τη δημιουργία Μοντέλων Οντοτήτων που μπορούν να στέλνουν και να λαμβάνουν διαφορετικού τύπου δεδομένα.

- **Cross-Platform**

Το Entity Framework υποστηρίζει cross-platform που του επιτρέπει να τρέχει σε Mac, Windows και Linux.

- **Υποβολή ερωτημάτων (Querying)**

Επιτρέπει στους προγραμματιστές να υποβάλλουν query για να ανακτήσουν δεδομένα από τη βάση δεδομένων.

- **Αποθήκευση (Saving)**

Με τη χρήση της συνάρτησης SaveChanges() γίνεται εκτέλεση των εντολών INSERT, UPDATE και DELETE στη βάση δεδομένων.

- **Παραμετροποίηση (Configuration)**

Το Entity Framework επιτρέπει στον προγραμματιστή να διαμορφώσει το μοντέλο του Entity Framework. Το Fluent API μπορεί επίσης να χρησιμοποιηθεί για να κάνει override των προεπιλεγμένων κανόνων.

- **Χρήση Migration**

Για τη δημιουργία ή τη διαχείριση του σχήματος δεδομένων, το Entity Framework παρέχει ένα σύνολο εντολών migration που μπορούν να εκτελεστούν στη γραμμή εντολών.

- **Change Tracking**

Επιτρέπει στο χρήστη να παρακολουθεί όλες τις αλλαγές που συνέβησαν στις τιμές των ιδιοτήτων.

- **Build-in-convention**

Ακολουθεί ενσωματωμένες συνθήκες και διαθέτει ενσωματωμένους προεπιλεγμένους κανόνες που διαμορφώνουν αυτόματα το Entity Framework.

- **Caching**

Η αποθήκευση στην κρυφή μνήμη βοηθά στην αύξηση της ταχύτητας αναζήτησης δεδομένων, καθώς το Entity Framework περιλαμβάνει το πρώτο επίπεδο αποθήκευσης στην κρυφή μνήμη

- **Concurrency**

Το Entity Framework χρησιμοποιεί το Optimistic Concurrency από προεπιλογή για την προστασία των αλλαγών αντικατάστασης(Overwriting changes).

2.6.4 Προσεγγίσεις για τη δημιουργία των Entities στο Entity Framework

- **Model First**

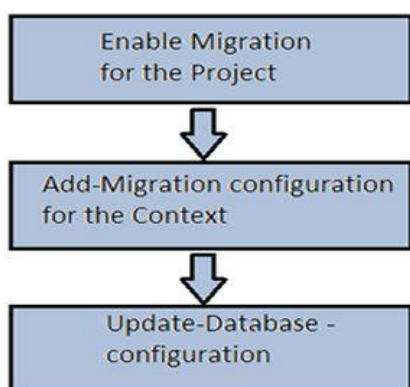
Στην προσέγγιση **Model First**, δημιουργούμε πρώτα το Entity Data Model (EDM) και μετά παράγεται η βάση δεδομένων από αυτό. Συνήθως δημιουργούμε ένα κενό EDM χρησιμοποιώντας τον Entity Data Model Wizard του Visual Studio, καθορίζουμε τις οντότητες και τις σχέσεις τους στο Visual Studio και στη συνέχεια, δημιουργείται η βάση δεδομένων από αυτό το καθορισμένο μοντέλο. Μπορούμε εύκολα να δημιουργήσουμε οντότητες και να ορίσουμε τις σχέσεις και τις συσχετίσεις τους χρησιμοποιώντας τον Designer του Visual Studio. Μπορούμε επίσης να καθορίσουμε την ιδιότητα κλειδιού (Key) και τους τύπους δεδομένων για τις ιδιότητες των οντοτήτων χρησιμοποιώντας τον Designer. Μπορούμε να χρησιμοποιήσουμε μερικές κλάσεις για να εφαρμόσουμε πρόσθετες δυνατότητες στις οντότητές μας.

- **Database First**

Μπορούμε να χρησιμοποιήσουμε την προσέγγιση **Database First** εάν η βάση δεδομένων έχει ήδη σχεδιαστεί και είναι έτοιμη. Σε αυτήν την προσέγγιση, το Entity Data Model (EDM) δημιουργείται από την υποκείμενη βάση δεδομένων. Χρησιμοποιώντας την προσέγγιση Database First, τα αρχεία edmx στο Visual Studio IDE παράγονται από τη βάση δεδομένων. Μη αυτόματες αλλαγές στη βάση δεδομένων ενημερώνουν πάντα το EDM (για παράδειγμα, εάν αλλάξει το σχήμα της υποκείμενης βάσης δεδομένων). Για να το κάνουμε αυτό, απλώς ενημερώνουμε το EDM από τη βάση δεδομένων στο Visual Studio IDE.

- Code First

Η προσέγγιση **Code First** μας δίνει τη δυνατότητα να δημιουργήσουμε ένα μοντέλο οντοτήτων και τη σχέση τους χρησιμοποιώντας κλάσεις και στη συνέχεια να δημιουργήσουμε τη βάση δεδομένων από αυτές τις κλάσεις. Μας δίνει τη δυνατότητα να εργαζόμαστε με το Entity Framework με αντικειμενοστραφή τρόπο. Εδώ δεν χρειάζεται να ανησυχούμε για τη δομή της βάσης δεδομένων. Όπως αναφέρθηκε προηγουμένως, πρέπει να δημιουργήσουμε κλάσεις που να αντιπροσωπεύουν τους πίνακες της βάσης δεδομένων. Αφού οριστούν οι οντότητες και καθοριστούν οι ιδιότητες και οι συσχετίσεις, χρησιμοποιώντας συγκεκριμένες εντολές στον Package Manager, παράγεται το μοντέλο και η βάση δεδομένων. Ανοίγοντας τον Package Manager Console στο Visual Studio και με τη χρήση συγκεκριμένων εντολών (enable/add/update) migration δίνεται η δυνατότητα να γίνουν αλλαγές στο μοντέλο και στη συνέχεια να μεταφερθούν αυτές οι αλλαγές στο σχήμα της βάσης δεδομένων.



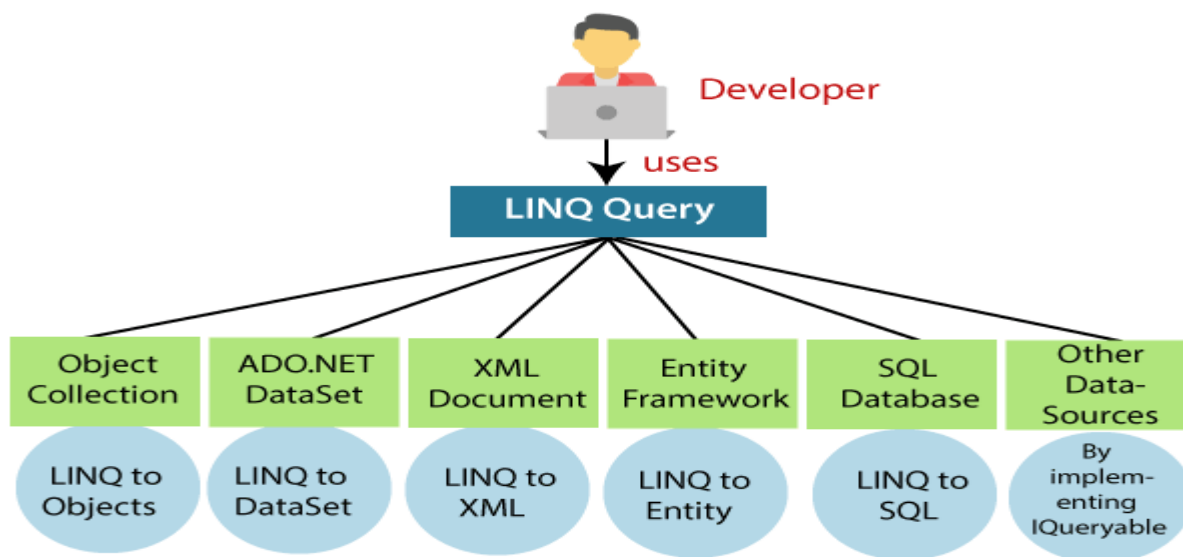
Εικόνα 5

Για την δημιουργία των Entities της εφαρμογής, χρησιμοποιήθηκε η προσέγγιση **Code First**.

2.6.5 Αποθήκευση και ανάκτηση δεδομένων -LINQ

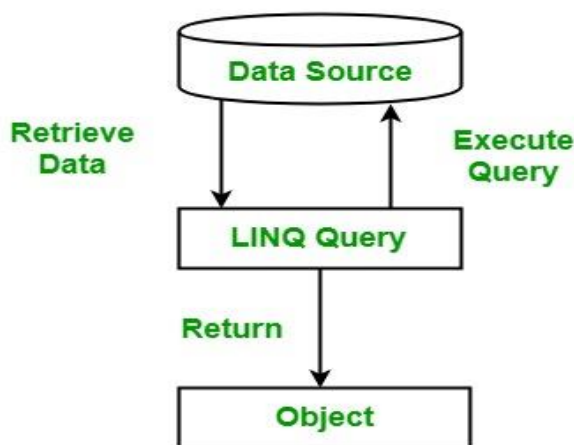
Το Language-Integrated-Query (LINQ) είναι ένα ισχυρό σύνολο τεχνολογιών που βασίζεται στην δυνατότητα ενσωμάτωσης των ερωτημάτων απευθείας στη γλώσσα C#. Τα ερωτήματα LINQ είναι η πρώτης τάξεως κατασκευή γλώσσας στο C# .NET, όπως και οι κλάσεις, οι μέθοδοι, τα συμβάντα. Το LINQ παρέχει μια συνεπή εμπειρία ερωτήματος σε αντικείμενα ερωτημάτων (LINQ σε αντικείμενα), σχεσιακές βάσεις δεδομένων (LINQ σε SQL) και XML (LINQ σε XML).

Το LINQ (Language Integrated Query) είναι ομοιόμορφη σύνταξη ερωτημάτων (query syntax) σε C# και VB.NET για ανάκτηση δεδομένων από διαφορετικές πηγές και μορφές. Είναι ενσωματωμένο σε C# ή VB, εξαλείφοντας έτσι την αναντιστοιχία μεταξύ γλωσσών προγραμματισμού και βάσεων δεδομένων, καθώς παρέχει μια ενιαία διεπαφή ερωτήματος για διαφορετικούς τύπους πηγών δεδομένων.



Εικόνα 6

Για παράδειγμα, η SQL είναι μια δομημένη γλώσσα ερωτημάτων που χρησιμοποιείται για την αποθήκευση και την ανάκτηση δεδομένων από μια βάση δεδομένων. Με τον ίδιο τρόπο, το **LINQ** είναι μια δομημένη σύνταξη ερωτημάτων ενσωματωμένη σε C# και VB.NET για ανάκτηση δεδομένων από διαφορετικούς τύπους πηγών δεδομένων όπως συλλογές, ADO.Net DataSet, Έγγραφα XML, υπηρεσία Web και MS SQL Server και άλλες βάσεις δεδομένων.



Εικόνα 7

Τα ερωτήματα LINQ επιστρέφουν αποτελέσματα ως αντικείμενα. Μας δίνει τη δυνατότητα να χρησιμοποιούμε αντικειμενοστραφή προσέγγιση στο σύνολο αποτελεσμάτων και να μην ανησυχούμε για τη μετατροπή διαφορετικών μορφών αποτελεσμάτων σε αντικείμενα.

Κεφάλαιο 3 – Ανάλυση Απαιτήσεων

Παρατίθενται οι λειτουργικές απαιτήσεις σε επίπεδο χρήστη :

Ρόλος : Χρήστης με δικαιώματα πελάτη

Ο χρήστης-πελάτης μπορεί να πλοηγηθεί στην ιστοσελίδα του βιβλιοπωλείου για να δει τα διαθέσιμα βιβλία.

Ο χρήστης-πελάτης μπορεί να πλοηγηθεί στην ιστοσελίδα του βιβλιοπωλείου και να αναζητήσει βιβλία με κριτήριο αναζήτησης την κατηγορία βιβλίου.

Ο χρήστης-πελάτης έχει τη δυνατότητα να εγγραφεί ως πελάτης του ηλεκτρονικού βιβλιοπωλείου.

Ο χρήστης-πελάτης έχει τη δυνατότητα να συνδεθεί στην διαδικτυακή εφαρμογή του βιβλιοπωλείου.

Ο χρήστης-πελάτης μπορεί να μεταβάλλει τόσο τον κωδικό του όσο και τα στοιχεία διεύθυνσης του στο λογαριασμό του .

Ο χρήστης-πελάτης μπορεί να δει αναλυτικά στοιχεία του βιβλίου εφόσον έχει πιστοποιηθεί, δηλαδή έχει εισάγει Username και Password.

Ο χρήστης-πελάτης μπορεί να προσθέσει οποιοδήποτε βιβλίο στο καλάθι ή να αφαιρέσει εφόσον έχει αρχικώς πιστοποιηθεί, δηλαδή έχει εισάγει Username και Password.

Ο χρήστης-πελάτης μπορεί να πραγματοποιήσει παραγγελία, εφόσον έχει αρχικώς πιστοποιηθεί, δηλαδή έχει εισάγει Username και Password.

Ο χρήστης-πελάτης μπορεί να λάβει το φορολογικό παραστατικό της παραγγελίας του, εφόσον η παραγγελία του έχει εγκριθεί από τον χρήστη-διαχειριστή του ηλεκτρονικού βιβλιοπωλείου.

Ρόλος : Χρήστης με δικαιώματα Διαχειριστή(Admin)

Ο χρήστης-διαχειριστής μπορεί να συνδεθεί(Log In) με ειδικούς κωδικούς ρόλου διαχειριστή.

Ο χρήστης-διαχειριστής ενημερώνεται από την πλατφόρμα της διαδικτυακής εφαρμογής για το σύνολο των μηνιαίων και ετήσιων εσόδων των βιβλίων που έχουν πωληθεί και για το πλήθος των παραγγελιών που είναι σε αναμονή έγκρισης .Τα μηνιαία έσοδα από τις πωλήσεις βιβλίων αναπαρίστανται σε γράφημα .

Ο χρήστης-διαχειριστής μπορεί να ελέγξει το ιστορικό των παραγγελιών των πελατών είτε αυτές βρίσκονται σε κατάσταση pending είτε σε κατάσταση confirmed.

Ο χρήστης-διαχειριστής μπορεί να εγκρίνει τις παραγγελίες των πελάτων και να εκτυπώσει το αντίστοιχο φορολογικό παραστατικό.

Ο χρήστης-διαχειριστής μπορεί να παρακολουθεί το απόθεμα του κάθε βιβλίου και σε ποια αποθήκη υφίσταται το απόθεμα αυτό.

Ο χρήστης-διαχειριστής ενημερώνεται με γράφημα για το σύνολο των εσόδων ανά μήνα.

Ο χρήστης-διαχειριστής μπορεί να λαμβάνει αναλυτικές πληροφορίες για τις αποθήκες του ηλεκτρονικού καταστήματος και ποιο απόθεμα ανά βιβλίο διατηρούν αυτές.

Ο χρήστης-διαχειριστής έχει την δυνατότητα να λαμβάνει λεπτομέρειες για το ποια βιβλία έχουν τις περισσότερες πωλήσεις, ποιο ύψος εσόδων έχουν αποφέρει αυτά και ποια η ποσότητα που έχει πωληθεί.

Ο χρήστης-διαχειριστής έχει την δυνατότητα να αιτηθεί παραγγελία βιβλίου στον προμηθευτή για ποσότητα βιβλίων που χρειάζεται.

Ο χρήστης-διαχειριστής έχει την δυνατότητα να παρακολουθήσει το ιστορικό των αιτήσεων στους προμηθευτές των βιβλίων και την κατάσταση των αιτήσεων αυτών.

Ο χρήστης-διαχειριστής έχει την δυνατότητα να αυξήσει το απόθεμα του βιβλίου σε μία συγκεκριμένη αποθήκη.

Μη-λειτουργικές απαιτήσεις :

Περιγράφουν ιδιότητες του συστήματος που συνήθως εκφράζονται βάσει χαρακτηριστικών της μορφής:

- Απόδοση (Performance)
- Χρησιμότητα (Usability)
- Ασφάλεια (Security)
- Νομιμότητα (Legislative)
- Ιδιωτικότητα (Privacy)

Με άλλα λόγια: περιγράφουν το πώς (ή το πόσο καλά) το σύστημα θα υποστηρίξει τις λειτουργικές απαιτήσεις .

Μπορούμε να τις θεωρήσουμε ως «**περιορισμούς**» που περιορίζουν τους τρόπους με τους οποίους θα μπορούσαμε να πραγματοποιήσουμε τις λειτουργικές απαιτήσεις. Παρακάτω αναφέρονται κάποιες μη λειτουργικές απαιτήσεις

- το σύστημα πρέπει να λειτουργεί αδιάλειπτα
- τα προσωπικά στοιχεία των πελατών πρέπει να προστατεύονται.
- ο χρήστης με δικαιώματα διαχειριστή μπορεί να έχει πρόσβαση στη βάση δεδομένων και μπορεί να την τροποποιεί.
- ο σχεδιασμός του συστήματος να είναι τέτοιος που να είναι φιλικός προς τον χρήστη
- το σύστημα πρέπει να επεξεργάζεται δεδομένα εντός ενός ελάχιστου χρόνου απόκρισης

3.1 UML (Unified Modeling Language)

Η Ενοποιημένη Γλώσσα Μοντελοποίησης (Unified Modeling Language, UML) είναι μία γραφική γλώσσα γενικού σκοπού, η οποία χρησιμοποιείται για τον προσδιορισμό, οπτικοποίηση, ανάπτυξη και τεκμηρίωση των κατασκευασμάτων (artifacts) ενός συστήματος λογισμικού.

Η UML είναι μία γλώσσα μοντελοποίησης (σύνολο από διαγράμματα). Δεν είναι γλώσσα προγραμματισμού .

Η UML χρησιμοποιείται :

- με οποιαδήποτε διαδικασία,
- σε οποιοδήποτε στάδιο ανάπτυξης
- για οποιαδήποτε τεχνολογική πλατφόρμα.

Η UML αποτυπώνει τόσο τη στατική δομή, όσο και τη δυναμική συμπεριφορά ενός συστήματος. Ένα αντικειμενοστραφές σύστημα μοντελοποιείται ως μία συλλογή αντικειμένων που αλληλοεπιδρούν για την εκτέλεση μιας λειτουργίας η οποία είναι τελικά αξιοποιήσιμη από τον χρήστη του συστήματος. Η **στατική δομή** καθορίζει τα είδη των αντικειμένων που είναι σημαντικά για το σύστημα καθώς και τις συσχετίσεις μεταξύ τους. Η **δυναμική συμπεριφορά** προσδιορίζει την εξέλιξη των αντικειμένων σε σχέση με τον χρόνο και την επικοινωνία μεταξύ τους.

Μοντελοποίηση

- Ένα μοντέλο είναι μία αναπαράσταση σε ένα συγκεκριμένο μέσο, κάποιας οντότητας στο ίδιο ή σε άλλο μέσο.
- Το μοντέλο αναπαριστά τα σημαντικά (από μία ορισμένη οπτική γωνία), στοιχεία της οντότητας που μοντελοποιείται, απλοποιώντας ή παραλείποντας τα υπόλοιπα.
- Το μέσο επάνω στο οποίο αναπτύσσεται το μοντέλο είναι βολικό για επεξεργασία.

Πλεονεκτήματα ενός μοντέλου :

- Ακριβής καθορισμός των απαιτήσεων έτσι ώστε όλοι οι εμπλεκόμενοι να τις κατανοούν με κοινό τρόπο. Μοντέλα ενός συστήματος λογισμικού κατασκευάζονται ώστε οι αναλυτές, προγραμματιστές, χειριστές έργων, πελάτες και τελικοί χρήστες να κατανοήσουν το σύστημα από την οπτική γωνία που τους ενδιαφέρει.
- Μελέτη του τρόπου σχεδίασης. Το μοντέλο ενός συστήματος λογισμικού παρέχει τη δυνατότητα διερεύνησης διαφόρων αρχιτεκτονικών, με οικονομικό τρόπο, αρκετά πριν από την κωδικοποίηση.
- Παραγωγή χρήσιμων πρωτοτύπων. Το μοντέλο ενός συστήματος λογισμικού μπορεί να χρησιμοποιηθεί για την παραγωγή πρωτοτύπων τα οποία με τη σειρά τους θα αξιοποιηθούν για τη διερεύνηση των απαιτήσεων.
- Διαχείριση περίπλοκων συστημάτων. Τα σύγχρονα συστήματα λογισμικού χαρακτηρίζονται από υψηλή πολυπλοκότητα όσον αφορά το μέγεθος τους, τον αριθμό των λειτουργιών τους, τον αριθμό των μονάδων που τα απαρτίζουν, το μέγεθος της ομάδας ανάπτυξης . Ένα μοντέλο μπορεί να συμβάλει στην αντιμετώπιση της πολυπλοκότητας εισάγοντας διάφορα επίπεδα αφαίρεσης, αποκρύπτοντας τον τεράστιο όγκο των λεπτομερειών.

Τα διαγράμματα UML που περιγράφουν τη στατική δομή του συστήματος είναι :

- Διάγραμμα κλάσεων (Class diagram)
- Διάγραμμα αντικειμένων (Object diagram)
- Διάγραμμα στοιχείων (Component diagram)
- Διάγραμμα διάταξης (Deployment diagram)

Τα διαγράμματα UML που περιγράφουν τη δυναμική δομή του συστήματος είναι :

- Διάγραμμα Περιπτώσεων Χρήσης (Use case diagram)
- Διάγραμμα Δραστηριοτήτων (Activity diagram)
- Διάγραμμα Ακολουθίας (Sequence diagram)
- Διάγραμμα Συνεργασίας (Collaboration diagram)
- Διάγραμμα Καταστάσεων (Statechart diagram)

Αναπτύχθηκαν κάποια από τα παραπάνω διαγράμματα για την ανάπτυξη της διαδικτυακής εφαρμογής. Συγκεκριμένα παρατίθενται τα διαγράμματα Περιπτώσεων Χρήσης, Κλάσεων, Δραστηριοτήτων, Ακολουθίας και Καταστάσεων

Διαγράμματα περιπτώσεων χρήσης (use cases diagrams) : Οι περιπτώσεις χρήσης περιγράφουν την δυναμική συμπεριφορά του συστήματος . Δείχνουν το πως οι χειριστές χρησιμοποιούνε το σύστημα .Αποτελούν το σημείο αναφοράς μια και χρησιμοποιούνται για να καθοδηγήσουν την ανάπτυξη του συστήματος σε όλες τις φάσεις ανάπτυξης.

Τα διαγράμματα περιπτώσεων χρήσης περιέχουν :

- Περιπτώσεις χρήσης (use cases)
- Χειριστές (actors)
- Συσχετίσεις

Είναι το βασικότερο είδος διαγράμματος της UML

Διαγράμματα κλάσεων (class diagrams)

Ένα διάγραμμα κλάσεων δείχνει την στατική δομή των κλάσεων του συστήματος και των σχέσεων μεταξύ τους . Ένα διάγραμμα κλάσεων συνήθως αποτελείται από:

- Κλάσεις (classes)
- Διαπροσωπίες (interfaces)
- Συνεργασίες (collaborations)
- Συσχετίσεις (relationships)

Το μοντέλο κλάσεων χρησιμοποιείται για :

Κατά την διάρκεια της ανάλυσης για να περιγράψει τις λειτουργικές απαιτήσεις (functional requirements)

Κατά τη διάρκεια του σχεδιασμού για να περιγράψει :

- το λεξιλόγιο του συστήματος (system's vocabulary)
- τις συνεργασίες (collaborations)
- το λογικό σχήμα της βάσης δεδομένων (logical database schema)

Διαγράμματα δραστηριοτήτων (activity diagrams): Ένα διάγραμμα δραστηριοτήτων περιγράφει την ροή των εργασιών μέσα στο σύστημα . Ένα διάγραμμα δραστηριοτήτων βασικά περιέχει:

- Δραστηριότητες (activities)
- Ενέργειες (actions)
- Μεταβάσεις (transitions)

Τα διαγράμματα δραστηριοτήτων χρησιμοποιούνται για την περιγραφή:

- μιας περίπτωσης χρήσης
- ροών εργασιών (workflows)
- διεργασιών (processes)
- Επιχειρηματικών διαδικασιών

Διαγράμματα ακολουθίας (sequence diagrams) : Τα διαγράμματα ακολουθίας παρουσιάζουν την αλληλεπίδραση των αντικειμένων μέσω της ανταλλαγής μηνυμάτων. Δίνουν έμφαση στη χρονική αλληλουχία των μηνυμάτων .

Ένα διάγραμμα ακολουθίας περιέχει :

- Χειριστές
- Αντικείμενα
- Μηνύματα που ανταλλάσσουν τα αντικείμενα

Τα διαγράμματα ακολουθίας χρησιμοποιούνται για να περιγράψουν τον κύκλο ζωής των αντικειμένων. Τα διαγράμματα ακολουθίας είναι σημασιολογικά ισοδύναμα με τα διαγράμματα συνεργασίας

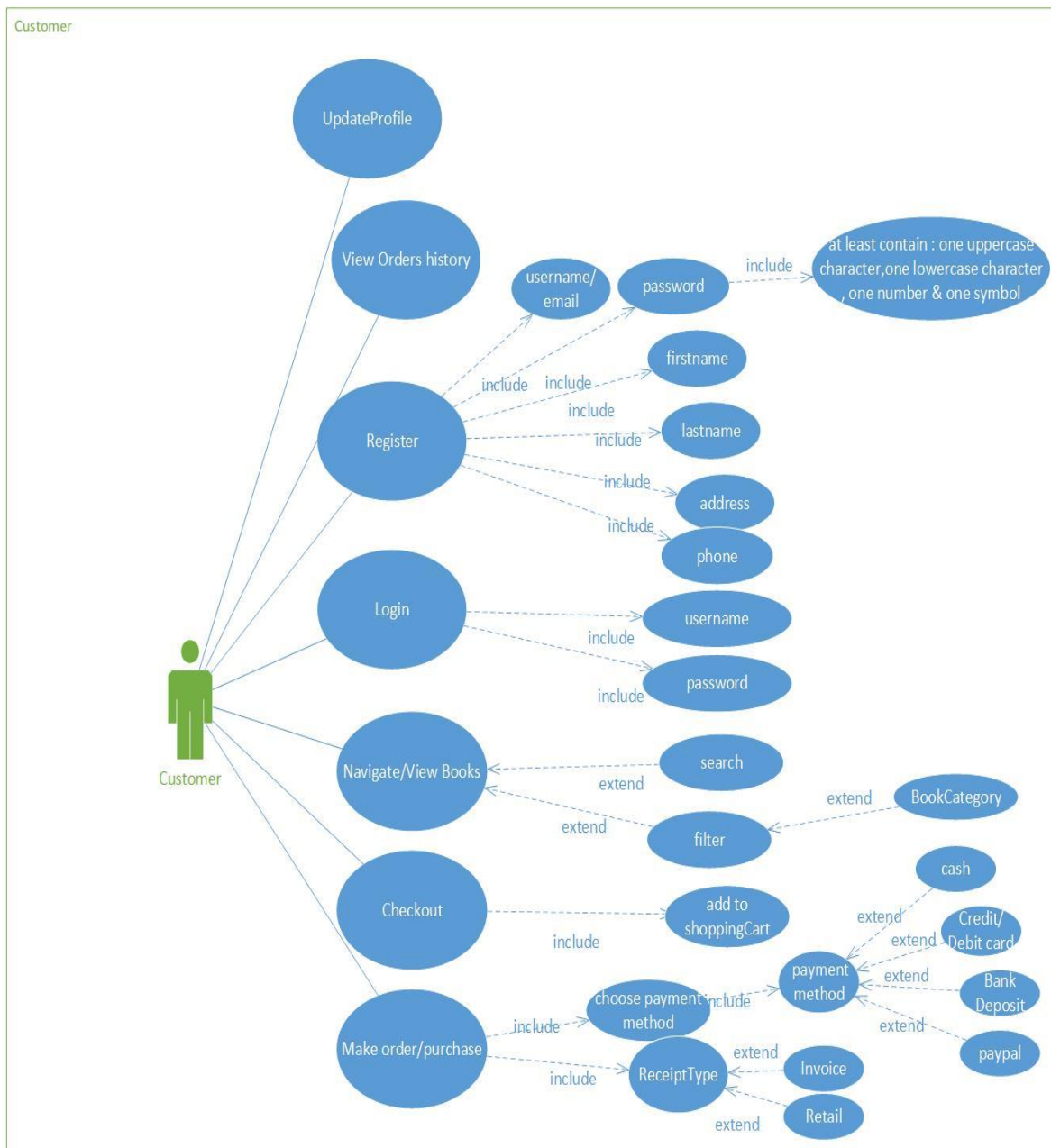
Διαγράμματα καταστάσεων (Statechart diagrams): Ένα διάγραμμα καταστάσεων περιγράφει μία μηχανή καταστάσεων δίνοντας έμφαση στην ροή του ελέγχου από κατάσταση σε κατάσταση. Μια μηχανή καταστάσεων προσδιορίζει :

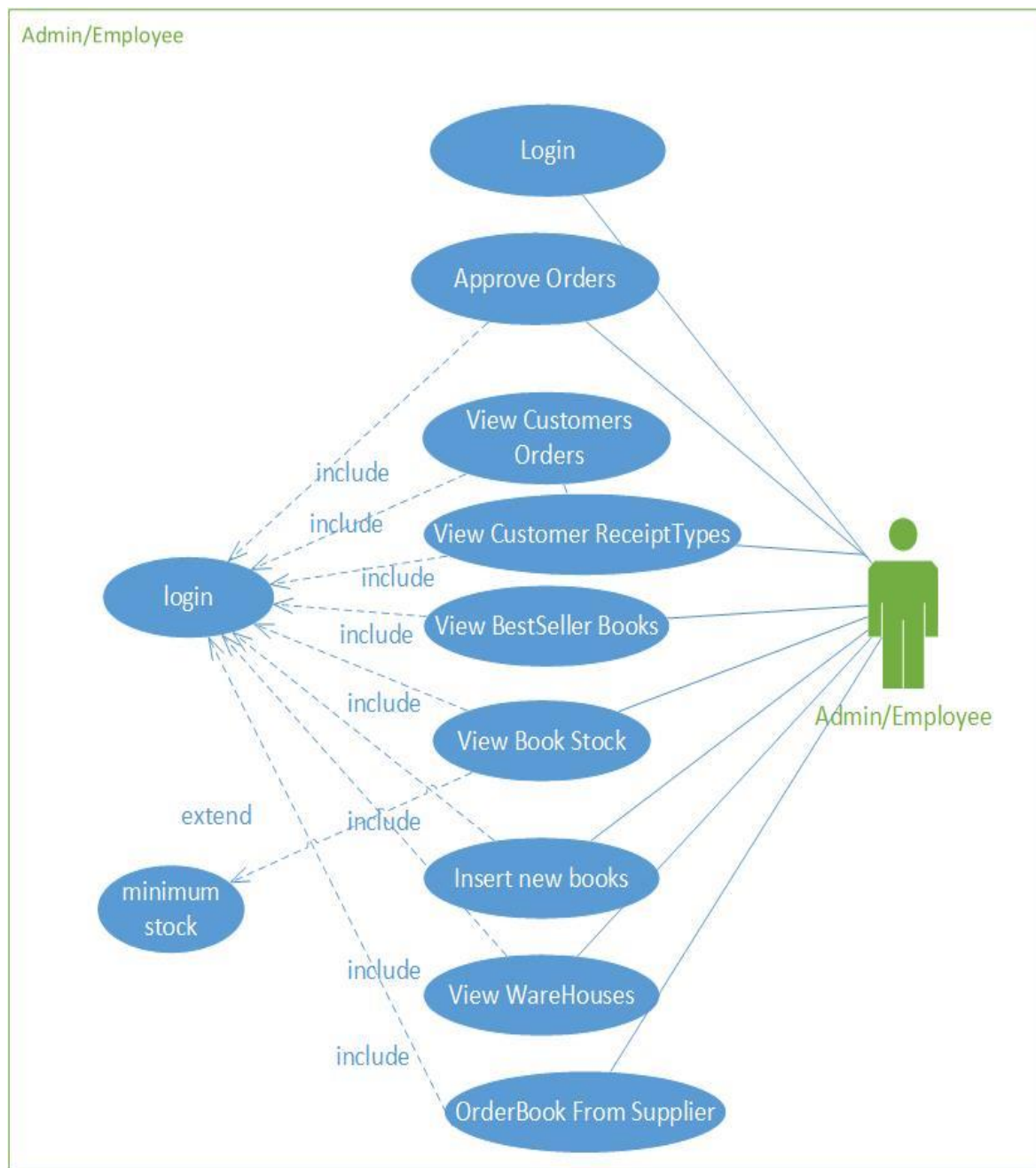
- Τις καταστάσεις (states) που μπορεί να βρεθεί ένα αντικείμενο
- Τα γεγονότα (events) στα οποία αντιδρά ένα αντικείμενο
- Την απόκριση (response) του αντικείμενου στα γεγονότα

Ένα διάγραμμα καταστάσεων περιέχει Καταστάσεις (states) και Μεταβάσεις (transitions) .

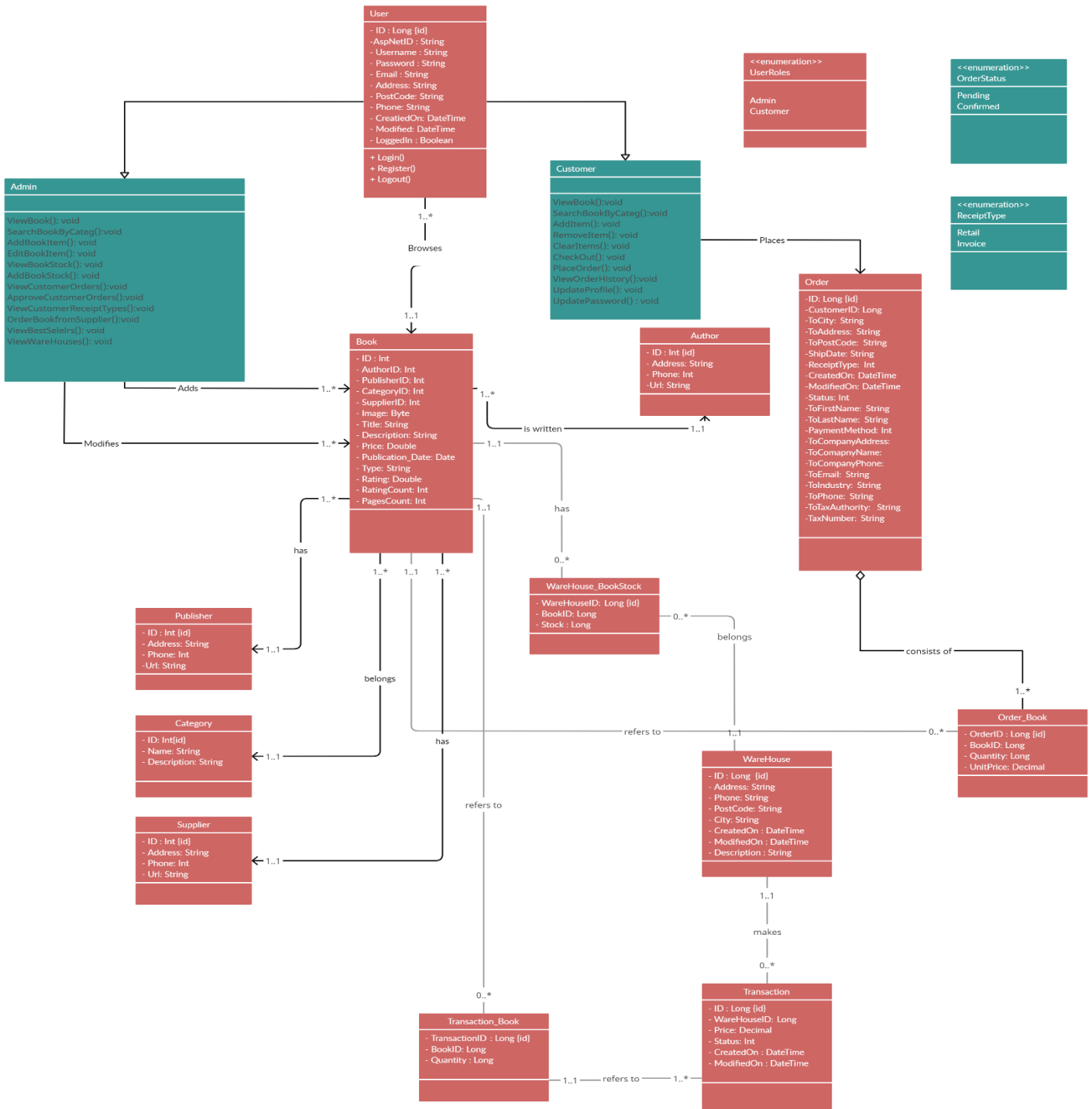
Ένα διάγραμμα καταστάσεων χρησιμοποιείται για την περιγραφή ενεργών (reactive) αντικειμένων.

3.2 Διάγραμμα Περιπτώσεων Χρήσης (Use Case)



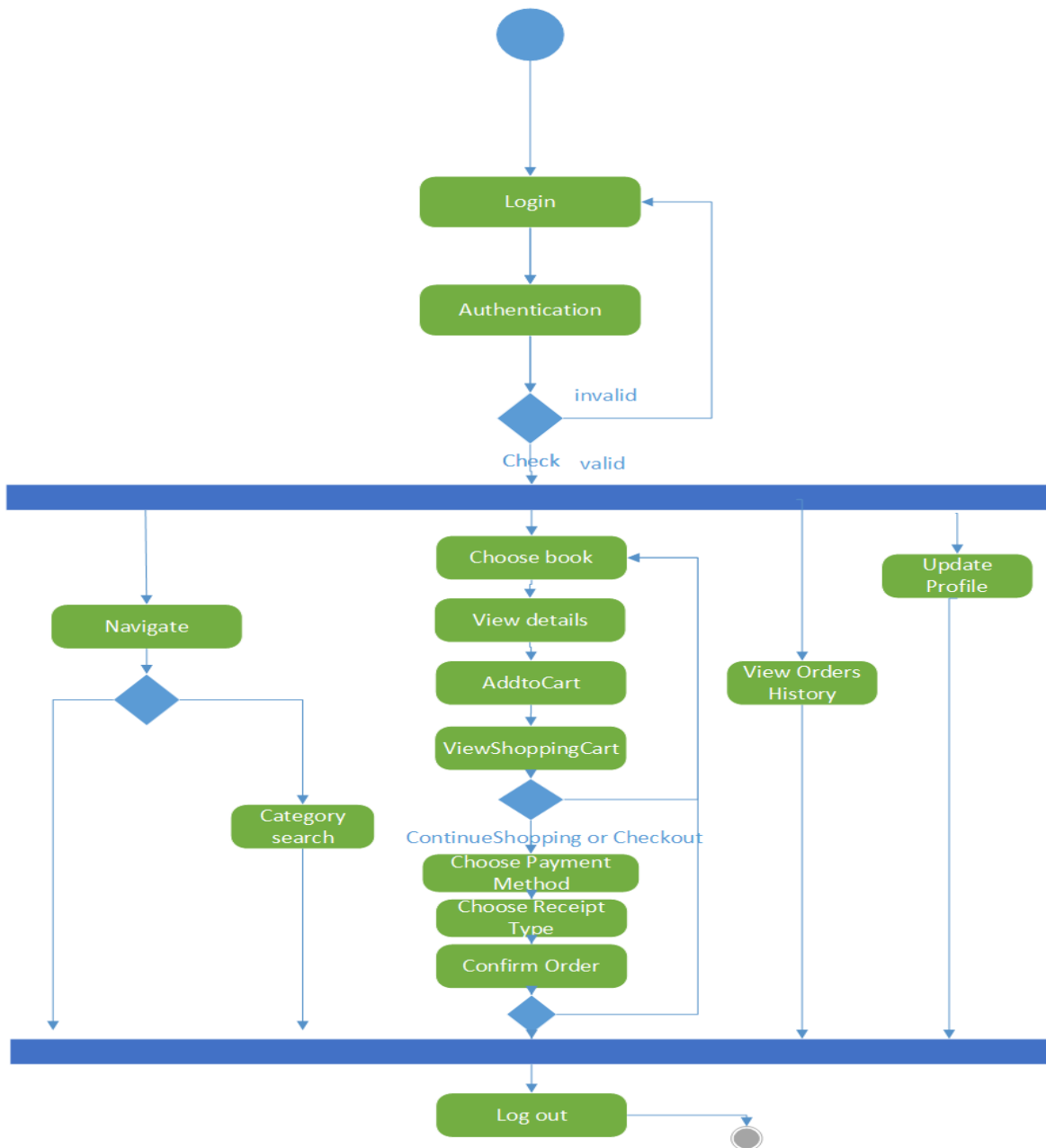


3.3 Διάγραμμα Κλάσεων (Class)

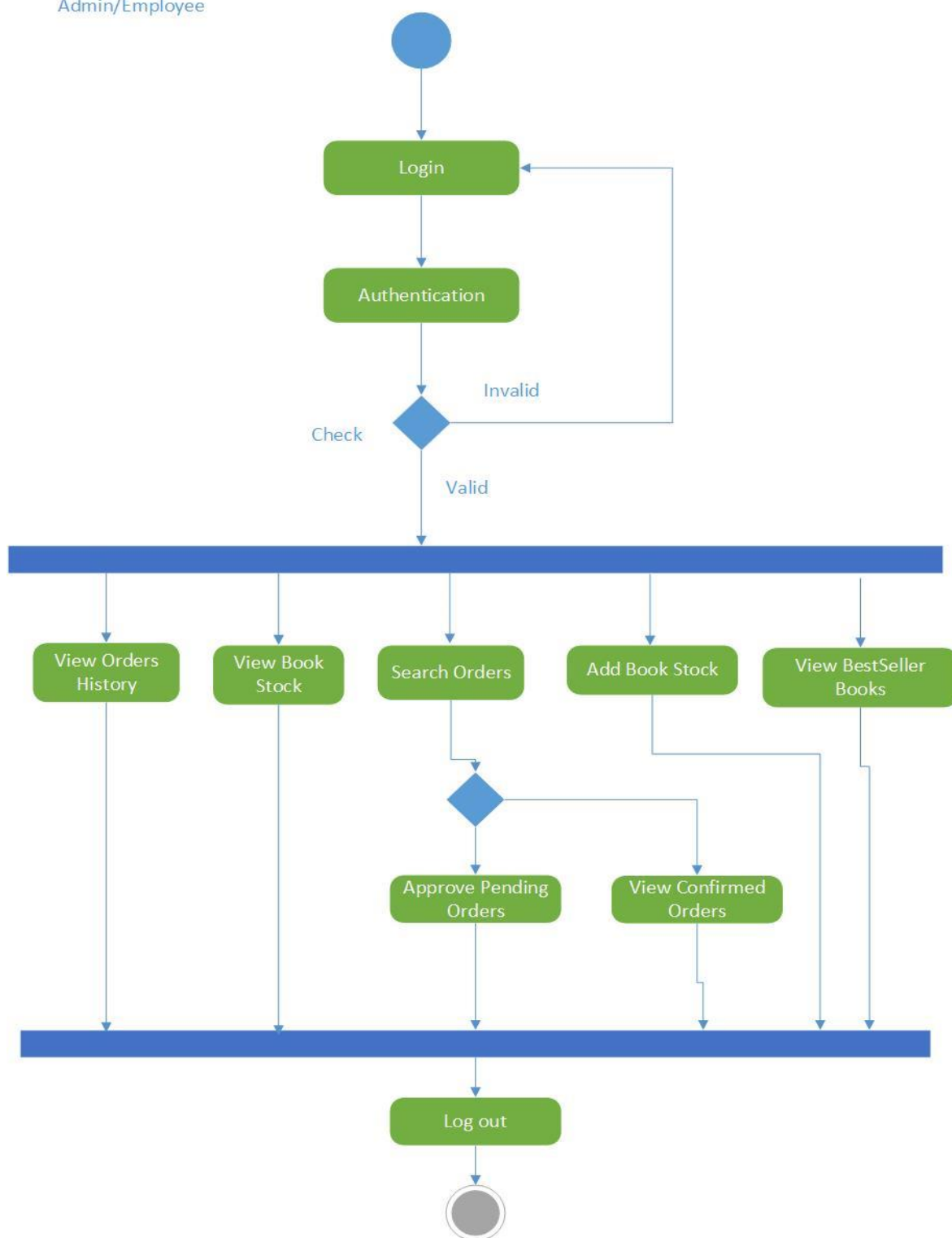


3.4 Διάγραμμα Δραστηριότητας (Activity)

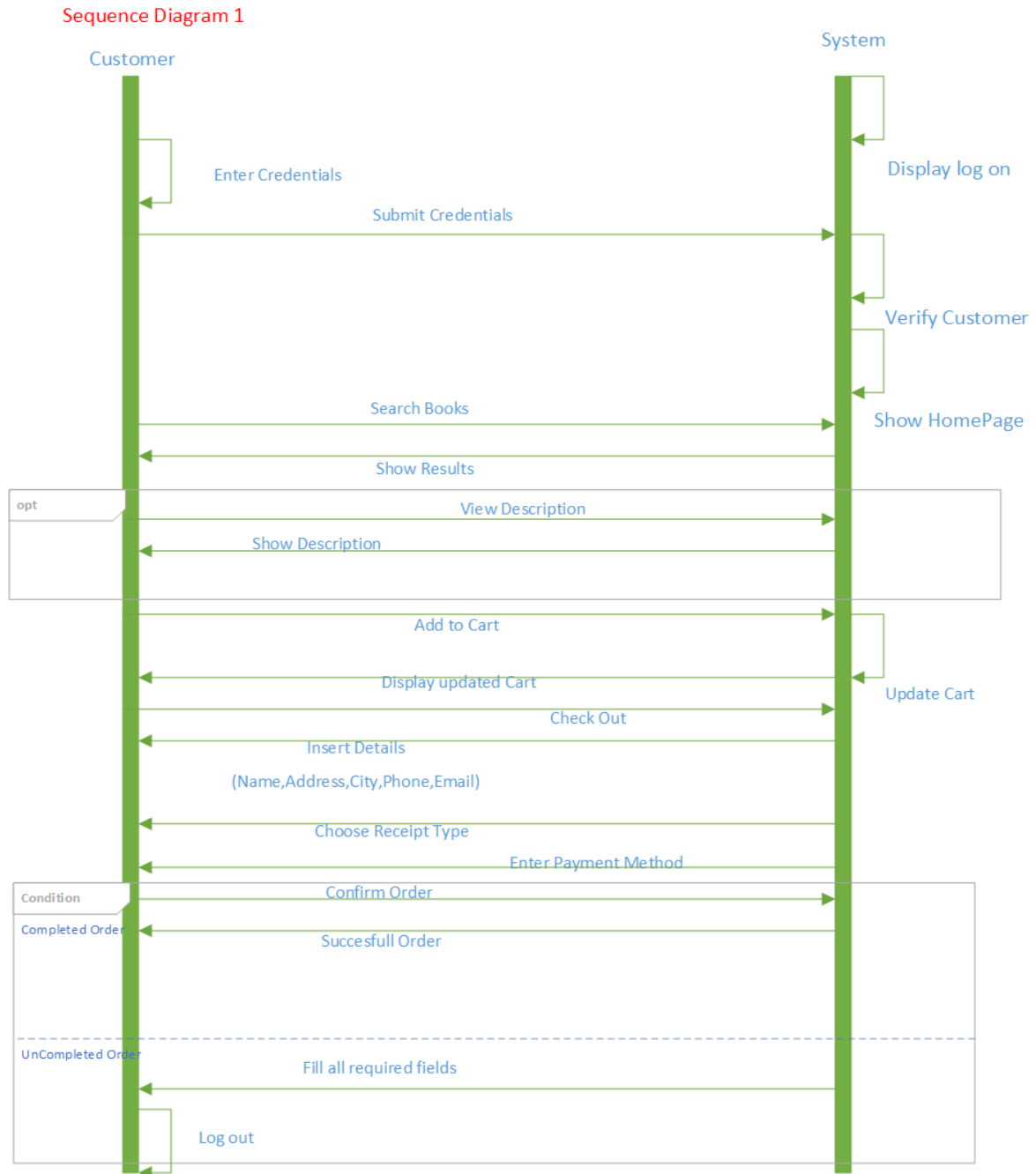
Activity Diagram for Customer/User



Activity Diagram for Admin/Employee



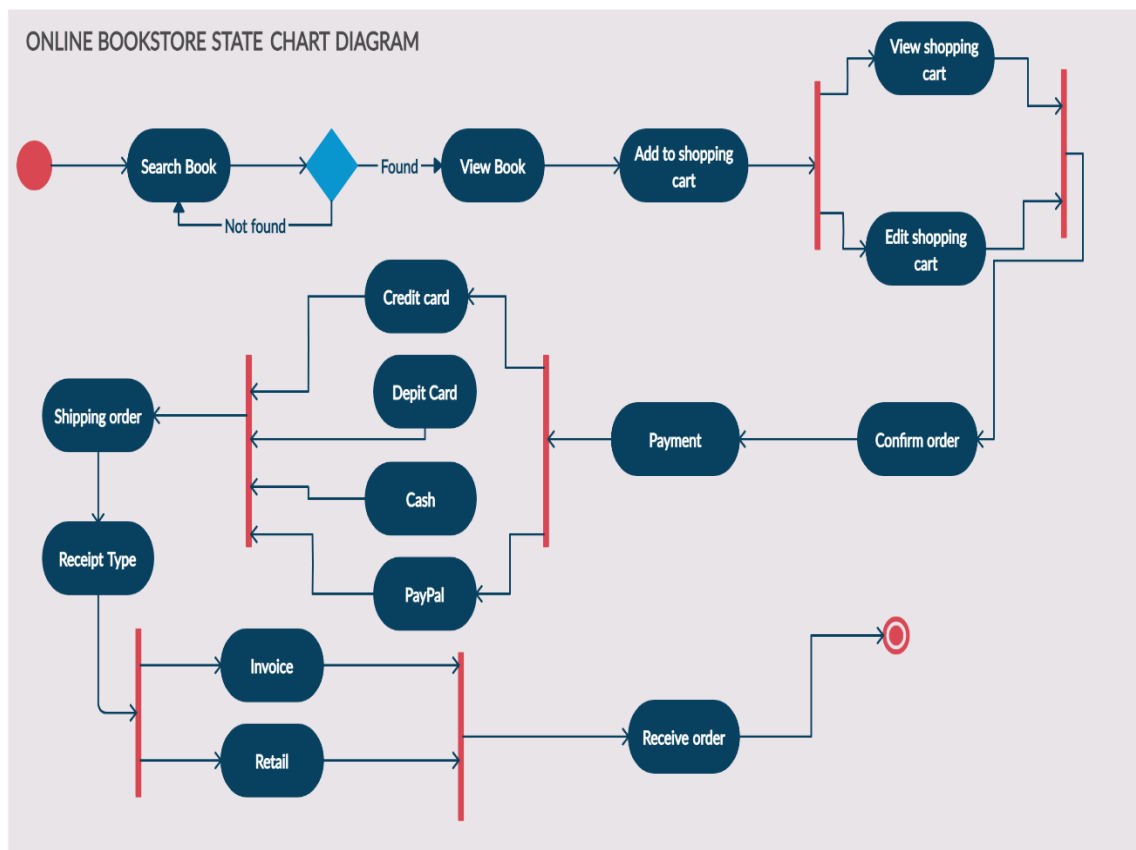
3.5 Διάγραμμα Ακολουθίας (Sequence)



Sequence Diagram version 2



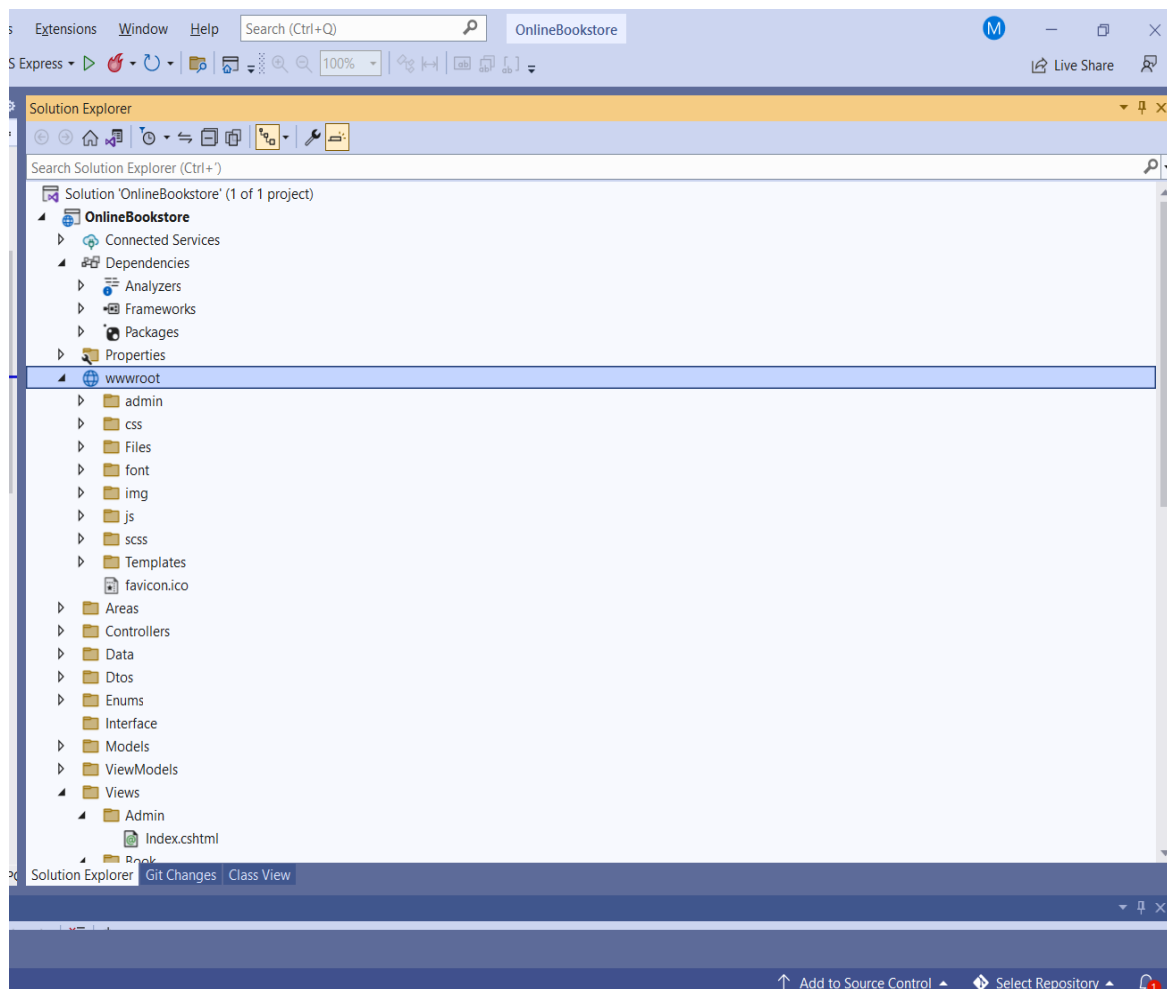
3.6 Διάγραμμα Καταστάσεων (State machine)



Για τον σχεδιασμό των διαγραμμάτων, χρησιμοποιήθηκε το πρόγραμμα Microsoft Visio και η διαδικτυακή εφαρμογή της Creately.com.

Κεφάλαιο 4 - Δομικά χαρακτηριστικά της εφαρμογής

Ο φάκελος **wwwroot** περιλαμβάνει τους υπό-φακέλους `css`, `files`, `fonts`, `img`, `js`, `templates` οι οποίοι χρησιμοποιούν στοιχεία του UI (user interface) της ιστοσελίδας.

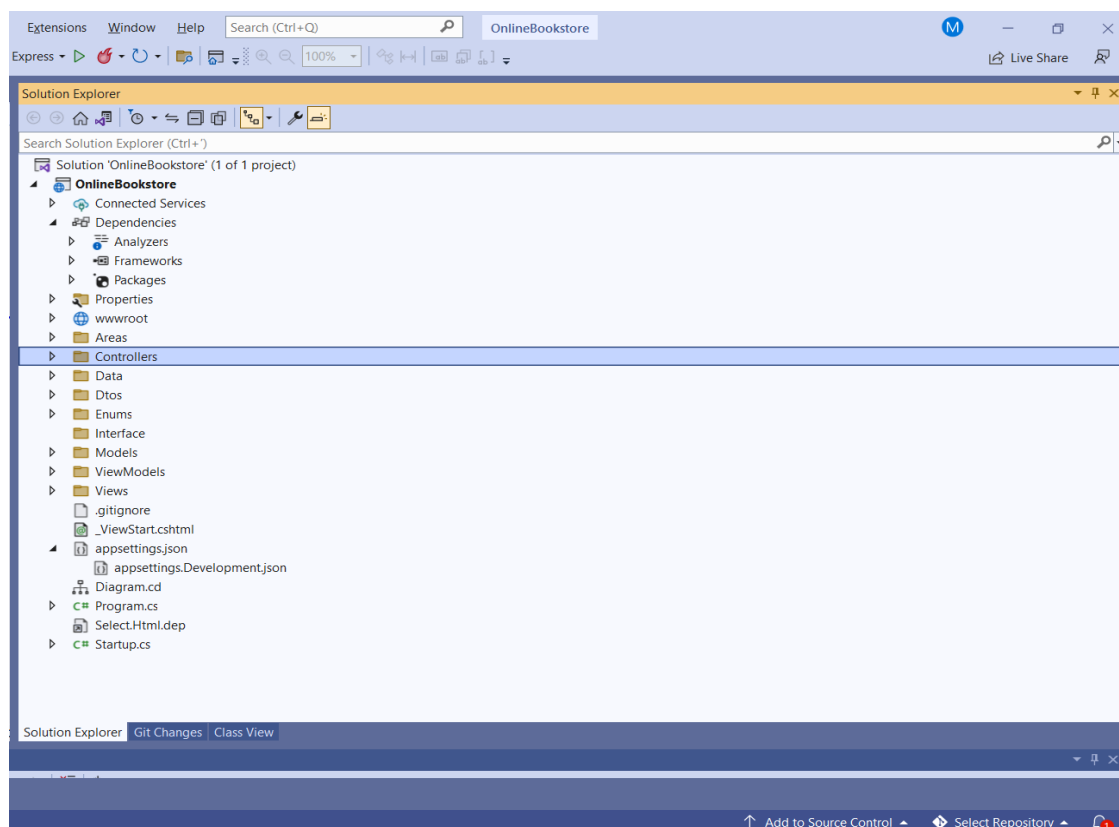


Εικόνα 8

- Ο φάκελος **css** περιλαμβάνει αρχεία `.css` που σχετίζονται με την μορφοποίηση του κώδικα `html` της ιστοσελίδας.
- Ο φάκελος **js** περιέχει αρχεία `.js` τα οποία έχουν κώδικα `javascript`. Εδώ εμπεριέχονται και οι βιβλιοθήκες `jQuery` και `bootstrap`.
- Ο φάκελος **img** περιέχει εικόνες που χρησιμοποιούνται κατά την εκκίνηση και λειτουργία της εφαρμογής.
- Ο φάκελος **files** περιέχει αρχεία `pdf`. Συγκεκριμένα είναι τα `output` αρχεία κατά την ολοκλήρωση μιας παραγγελίας από έναν πελάτη, είτε αυτό αφορά Απόδειξη λιανικής πώλησης (`retail`) είτε Τιμολόγιο Δελτίο Αποστολής (`Invoice`).
- Ο φάκελος **fonts** περιέχει τις γραμματοσειρές που θα χρησιμοποιηθούν στο πρόγραμμα.

- Ο φάκελος **templates** περιλαμβάνει το format των παραστατικών δηλαδή ο σχεδιασμός της εκτυπώσιμης μορφής αυτών σε αρχεία .pdf τόσο για την απόδειξη λιανικής πώλησης όσο και του τιμολογίου .

Όπως είχαμε αναφέρει και στο προηγούμενο κεφάλαιο, η αρχιτεκτονική MVC, αποτελείται από τα στοιχεία Model, View και Controller.



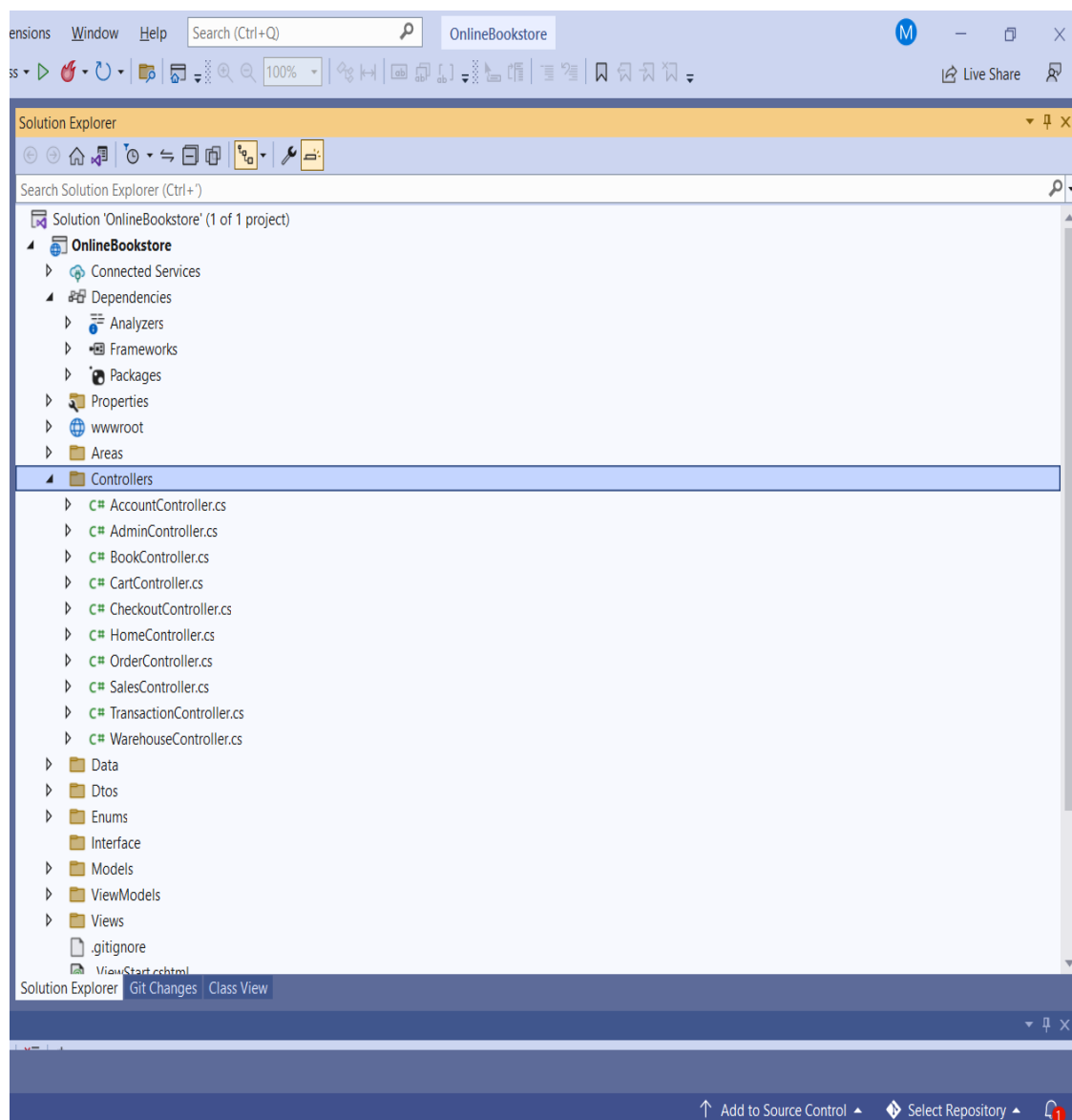
Εικόνα 9

Όπως παρατηρούμε στην παραπάνω εικόνα, υπάρχουν οι τρεις φάκελοι που ονομάζονται Μοντέλα (Models), Προβολές (Views) και Ελεγκτές (Controller). Επιπρόσθετα υπάρχει και ένας φάκελος που ονομάζεται ViewModels.

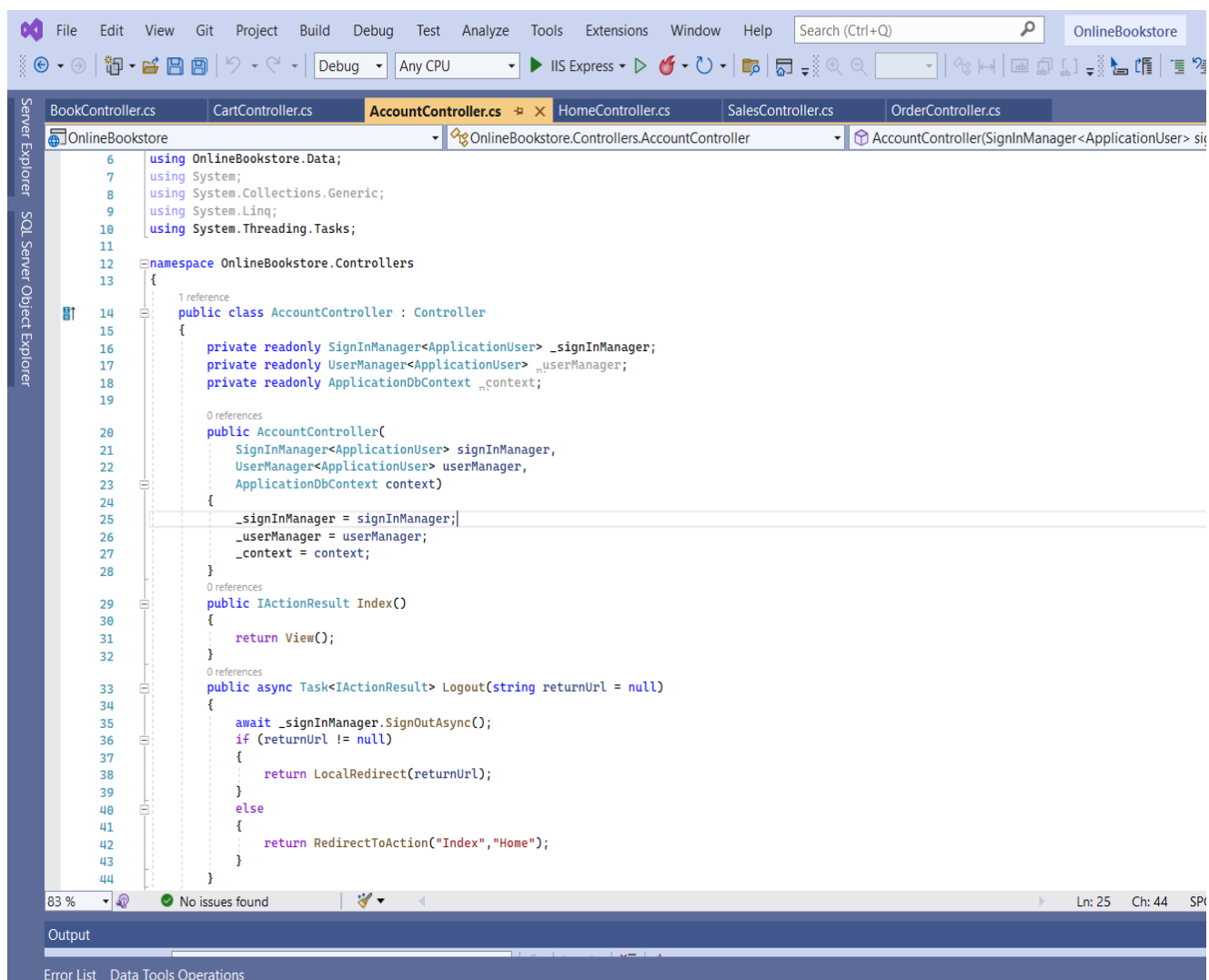
Οι Controllers λειτουργούν ως διεπαφή μεταξύ των στοιχείων του μοντέλου και της προβολής. Στην εργασία χρησιμοποιήθηκαν 10 controllers, όπως φαίνεται και παρακάτω. Ο HomeController είναι προεπιλεγμένος ελεγκτής για την εφαρμογή.

4.1 Controllers

Σε κάθε ελεγκτή βρίσκονται συγκεντρωμένες κάποιες μέθοδοι που εκτελούν ενέργειες (ActionResult) και αντιστοιχούν σε μία όψη (View), η οποία εμφανίζεται στη σελίδα του φυλλομετρητή.



Εικόνα 10 Controllers



```
6 using OnlineBookstore.Data;
7 using System;
8 using System.Collections.Generic;
9 using System.Linq;
10 using System.Threading.Tasks;
11
12 namespace OnlineBookstore.Controllers
13 {
14     public class AccountController : Controller
15     {
16         private readonly SignInManager<ApplicationUser> _signInManager;
17         private readonly UserManager<ApplicationUser> _userManager;
18         private readonly ApplicationDbContext _context;
19
20         0 references
21         public AccountController(
22             SignInManager<ApplicationUser> signInManager,
23             UserManager<ApplicationUser> userManager,
24             ApplicationDbContext context)
25         {
26             _signInManager = signInManager;
27             _userManager = userManager;
28             _context = context;
29         }
30
31         0 references
32         public IActionResult Index()
33         {
34             return View();
35         }
36
37         0 references
38         public async Task<IActionResult> Logout(string returnUrl = null)
39         {
40             await _signInManager.SignOutAsync();
41             if (returnUrl != null)
42             {
43                 return LocalRedirect(returnUrl);
44             }
45             else
46             {
47                 return RedirectToAction("Index", "Home");
48             }
49         }
50     }
51 }
```

Εικόνα 11 Account Controller

Ο Γενικός controller χρησιμοποιείται για να γίνεται η αποσύνδεση από την εφαρμογή και η ανακατεύθυνση στην αρχική οθόνη μετά το Log out.


```

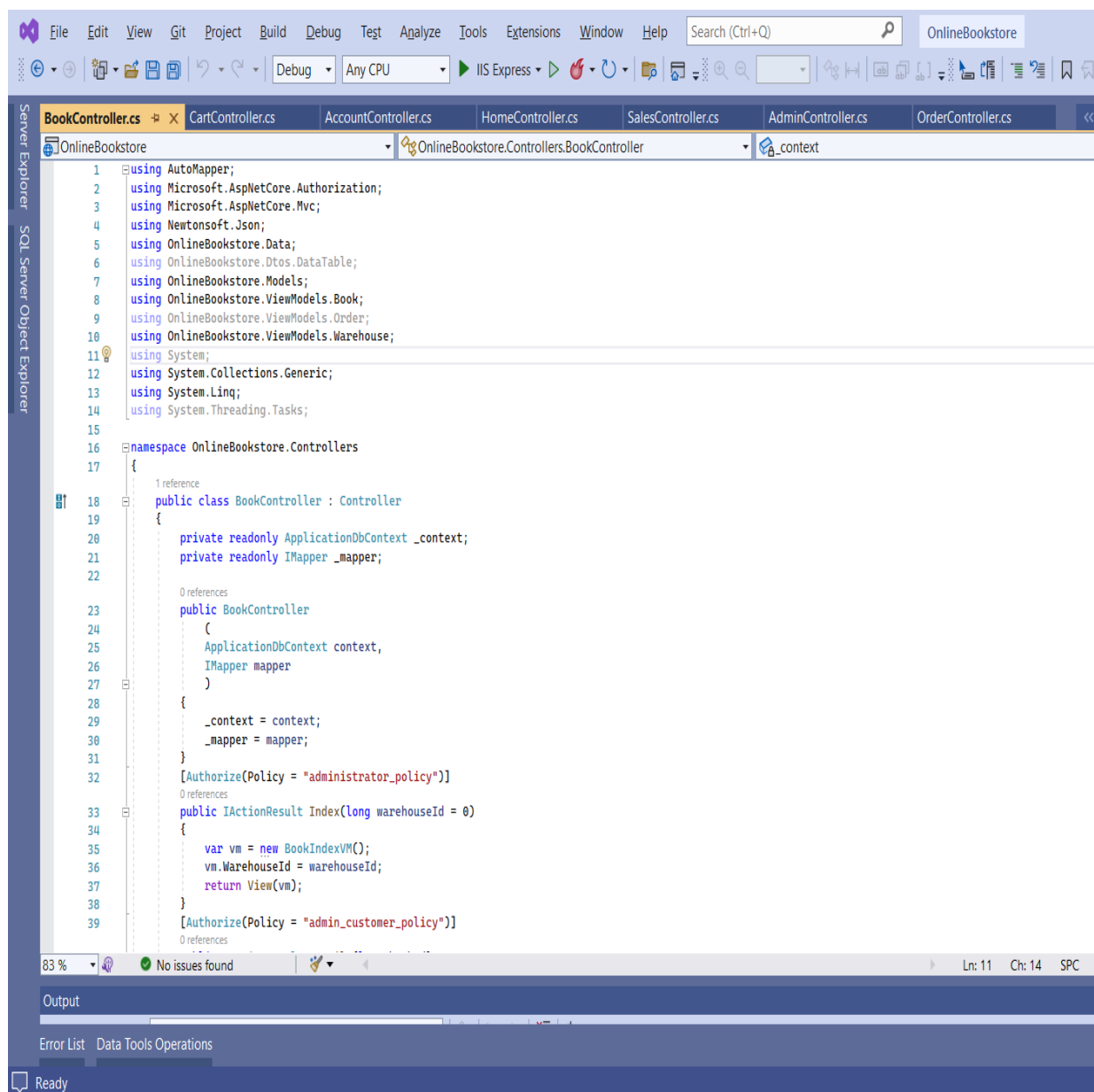
3  using OnlineBookstore.Data;
4  using OnlineBookstore.Enums;
5  using OnlineBookstore.ViewModels.Admin;
6  using System;
7  using System.Collections.Generic;
8  using System.Linq;
9  using System.Threading.Tasks;
10
11 namespace OnlineBookstore.Controllers
12 {
13     [Authorize(Policy = "administrator_policy")]
14     public class AdminController : Controller
15     {
16         private readonly ApplicationDbContext _context;
17
18         public AdminController(
19             ApplicationDbContext context
20         )
21         {
22             _context = context;
23         }
24
25         public IActionResult Index()
26         {
27             var vm = new AdminIndexVM();
28
29             var sales =
30                 (from bo in _context.BookOrders
31                  join b in _context.Books on bo.BookId equals b.Id
32                  join o in _context.Orders on bo.OrderId equals o.Id
33                  where o.CreatedOn.Year == DateTime.Now.Year && o.Status == (int)OrderStatuses.Confirmed
34                  group bo by o.CreatedOn.Month into g
35                  select new { Month = g.Key, BookSales = g.Sum(bo => bo.Quantity * bo.UnitPrice), Quantity = g.Sum(bo => bo.Quantity) }).ToList();
36
37             var salesArray = new int[12] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
38             foreach (var sale in sales)
39             {
40                 salesArray[sale.Month] = (int)sale.BookSales;
41             }
42             vm.SalesArray = salesArray;
43             vm.PendingOrders = _context.Orders.Where(x => x.Status == (int)OrderStatuses.Pending).Count();
44             vm.MonthlyEarnings = ((Long)sales.Sum(x => x.BookSales))/DateTime.Now.Month;
45             vm.AnnualEarnings = ((Long)sales.Sum(x=>x.BookSales));
46
47             return View(vm);
48         }

```

Εικόνα 12 AdminController

Το action Index φορτώνει τα δεδομένα της κεντρικής σελίδας του Admin (<https://localhost/Admin>). Το παρακάτω query φέρνει τις πωλήσεις των βιβλίων ανά μήνα για το τρέχον ημερολογιακό έτος. Δηλαδή για κάθε επιτυχή παραγγελία πολλαπλασιάζουμε την ποσότητα του κάθε βιβλίου της παραγγελίας με την τιμή του για να πάρουμε το κόστος της παραγγελίας και αθροίζοντας τα κόστη των παραγγελιών ανά μήνα παίρνουμε τις συνολικές πωλήσεις ανά μήνα και τις εμφανίζουμε στο κεντρικό διάγραμμα της σελίδας του Διαχειριστή.

Επιπρόσθετα υπολογίζουμε τις συνολικές πωλήσεις για όλους τους μήνες του τρέχοντος έτους. Υπολογίζουμε επίσης το μέσο όρο των πωλήσεων μέχρι τον τρέχον μήνα από την αρχή του έτους (Monthly Earnings). Και τέλος δείχνουμε και τον αριθμό των παραγγελιών που είναι σε εκκρεμότητα (Pending Orders).



Εικόνα 13 BookController_1

```

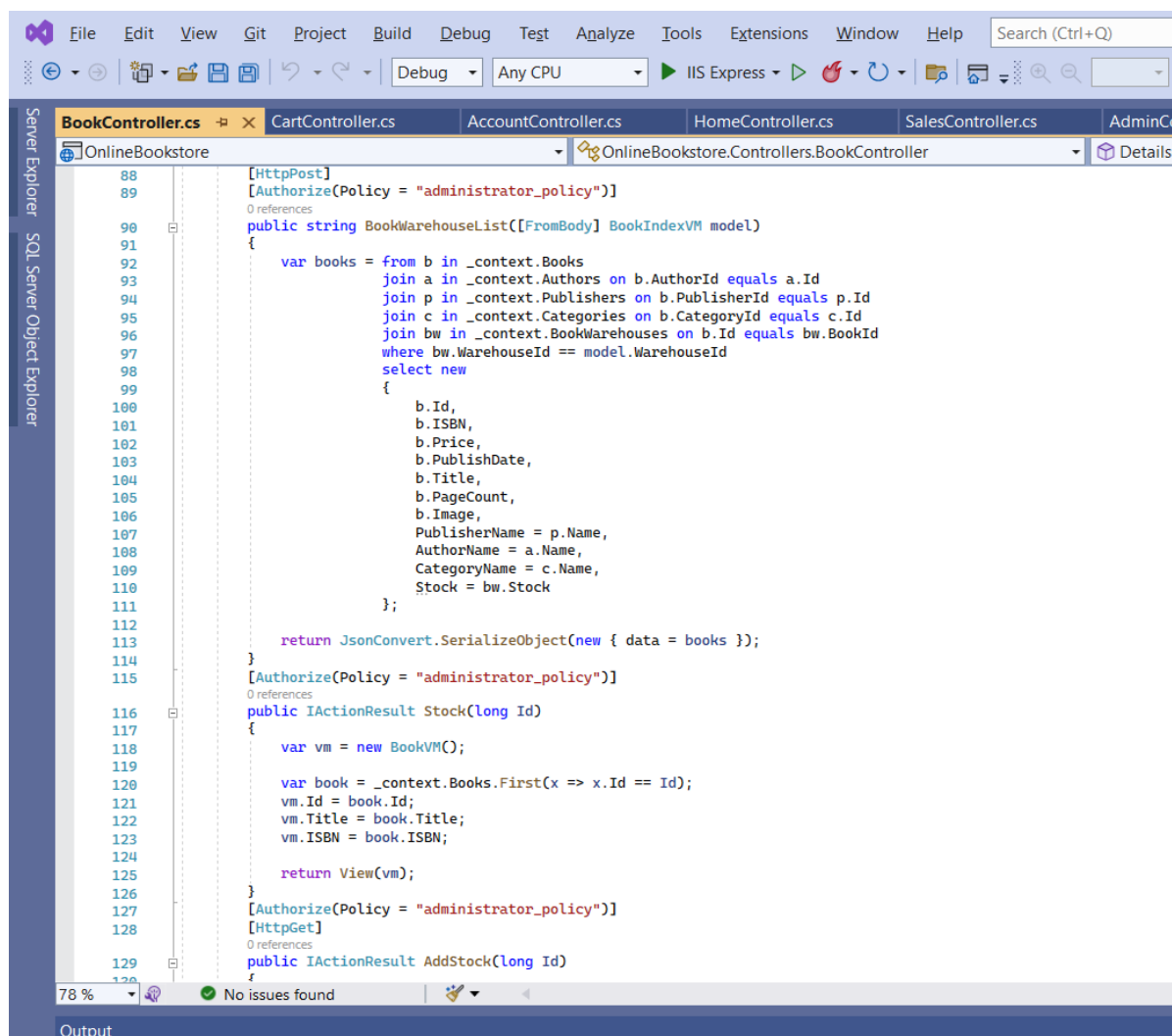
39 [Authorize(Policy = "admin_customer_policy")]
40 public IActionResult Details(long bookId)
41 {
42     var vm = new BookDetailsVM();
43     var book = _context.Books.Find(bookId);
44     vm.Book = _mapper.Map<BookVM>(book);
45     vm.Book.AuthorName = _context.Authors.Find(book.AuthorId).Name;
46     vm.Book.PublisherName = _context.Publishers.Find(book.PublisherId).Name;
47     vm.Book.CategoryName = _context.Categories.Find(book.CategoryId).Name;
48     vm.Book.IsInStock = _context.BookWarehouses.Any(x => x.BookId == book.Id && x.Stock > 0);
49     return View(vm);
50 }
51 [HttpPost]
52 [Authorize(Policy = "administrator_policy")]
53 public string BookList([FromBody] BookIndexVM model)
54 {
55     var books = from b in _context.Books
56                 join a in _context.Authors on b.AuthorId equals a.Id
57                 join p in _context.Publishers on b.PublisherId equals p.Id
58                 join c in _context.Categories on b.CategoryId equals c.Id
59                 join sub in (
60                     from bw in _context.BookWarehouses
61                     join b in _context.Books on bw.BookId equals b.Id
62                     join w in _context.Warehouses on bw.WarehouseId equals w.Id
63                     group bw by b.Id into g
64                     select new
65                     {
66                         Stock = g.Sum(x => x.Stock),
67                         Id = g.Key
68                     }
69                 ) on b.Id equals sub.Id into subs
70                 from s in subs.DefaultIfEmpty()
71                 select new
72                 {
73                     b.Id,
74                     b.ISBN,
75                     b.Price,
76                     b.PublishDate,
77                     b.Title,
78                     b.PageCount,
79                     b.Image,
80                     PublisherName = p.Name,
81                     AuthorName = a.Name,
82                     CategoryName = c.Name,
83                     Stock = s.Stock
84                 };
85     return JsonConvert.SerializeObject(new { data = books });
86 }

```

Εικόνα 14 BookController_2

Πατώντας την εικόνα ενός βιβλίου καλείται αυτό το action Details που φορτώνει την πληροφορία του βιβλίου, δηλαδή τίτλος βιβλίου, τιμή, περιγραφή, κατηγορία, συγγραφέα, εκδότη και εμφανίζεται εάν είναι σε απόθεμα ή όχι. Εάν δεν υπάρχει σε απόθεμα εμφανίζουμε στο χρήστη την κατάσταση **Sold Out** πάνω δεξιά στην εικόνα του βιβλίου.

Booklist-Όταν ανοίγει η σελίδα με τη λίστα των βιβλίων στο Admin Panel , καλείται το παρακάτω αίτημα . Το query αυτό φέρνει την πληροφορία των βιβλίων (ISBN, τιμή , εκδότη, συγγραφέα, κατηγορία, τίτλο βιβλίου ,αριθμό σελίδων κλπ.) αλλά και το απόθεμα των βιβλίων (για κάθε βιβλίο αθροίζει το απόθεμα σε όλα τις αποθήκες (Warehouses)που έχει το συγκεκριμένο βιβλίο)



```
88 [HttpPost]
89 [Authorize(Policy = "administrator_policy")]
90 public string BookWarehouseList([FromBody] BookIndexVM model)
91 {
92     var books = from b in _context.Books
93                 join a in _context.Authors on b.AuthorId equals a.Id
94                 join p in _context.Publishers on b.PublisherId equals p.Id
95                 join c in _context.Categories on b.CategoryId equals c.Id
96                 join bw in _context.BookWarehouses on b.Id equals bw.BookId
97                 where bw.WarehouseId == model.WarehouseId
98                 select new
99                 {
100                     b.Id,
101                     b.ISBN,
102                     b.Price,
103                     b.PublishDate,
104                     b.Title,
105                     b.PageCount,
106                     b.Image,
107                     PublisherName = p.Name,
108                     AuthorName = a.Name,
109                     CategoryName = c.Name,
110                     Stock = bw.Stock
111                 };
112
113     return JsonConvert.SerializeObject(new { data = books });
114 }
115 [Authorize(Policy = "administrator_policy")]
116 public IActionResult Stock(long Id)
117 {
118     var vm = new BookVM();
119
120     var book = _context.Books.First(x => x.Id == Id);
121     vm.Id = book.Id;
122     vm.Title = book.Title;
123     vm.ISBN = book.ISBN;
124
125     return View(vm);
126 }
127 [Authorize(Policy = "administrator_policy")]
128 [HttpGet]
129 public IActionResult AddStock(long Id)
130 {
```

Εικόνα 15 BookController_3

Το action Stock καλείται όταν κλικάρουμε το κουμπί Stock για κάποιο βιβλίο στην οθόνη με τα βιβλία και πραγματοποιεί ανακατεύθυνση στην οθόνη με τα αποθέματα της κάθε αποθήκης για το συγκεκριμένο βιβλίο.

```

127 [Authorize(Policy = "administrator_policy")]
128 [HttpGet]
129 public IActionResult AddStock(Long Id)
130 {
131     var vm = new AddStockVM();
132     var warehouses = _context.Warehouses.ToList();
133     var vmWarehouses = _mapper.Map<List<WarehouseVM>>(warehouses);
134     var book = _context.Books.First(x => x.Id == Id);
135     vm.Book.Id = book.Id;
136     vm.Book.Title = book.Title;
137     vm.Book.ISBN = book.ISBN;
138
139     return View(vm);
140 }
141 [Authorize(Policy = "administrator_policy")]
142 [HttpPost]
143 public IActionResult AddStock(AddStockVM vm)
144 {
145     var warehouses = _context.Warehouses.ToList();
146     var vmWarehouses = _mapper.Map<List<WarehouseVM>>(warehouses);
147     var book = _context.Books.First(x => x.Id == vm.Book.Id);
148     vm.Book.Id = book.Id;
149     vm.Book.Title = book.Title;
150     vm.Book.ISBN = book.ISBN;
151     var exists = _context.BookWarehouses.Any(x => x.WarehouseId == vm.WarehouseId && x.BookId == book.Id);
152     if (!exists)
153     {
154         _context.BookWarehouses.Add(new BookWarehouse()
155         {
156             BookId = book.Id,
157             Stock = vm.Stock,
158             WarehouseId = vm.WarehouseId
159         });
160     }
161     else
162     {
163         var bookWarehouse = _context.BookWarehouses.First(x => x.BookId == book.Id && x.WarehouseId == vm.WarehouseId);
164         bookWarehouse.Stock += vm.Stock;
165         _context.BookWarehouses.Update(bookWarehouse);
166     }
167
168     _context.SaveChanges();
169     return RedirectToAction("Stock", "Book", new { Id = book.Id });
170 }

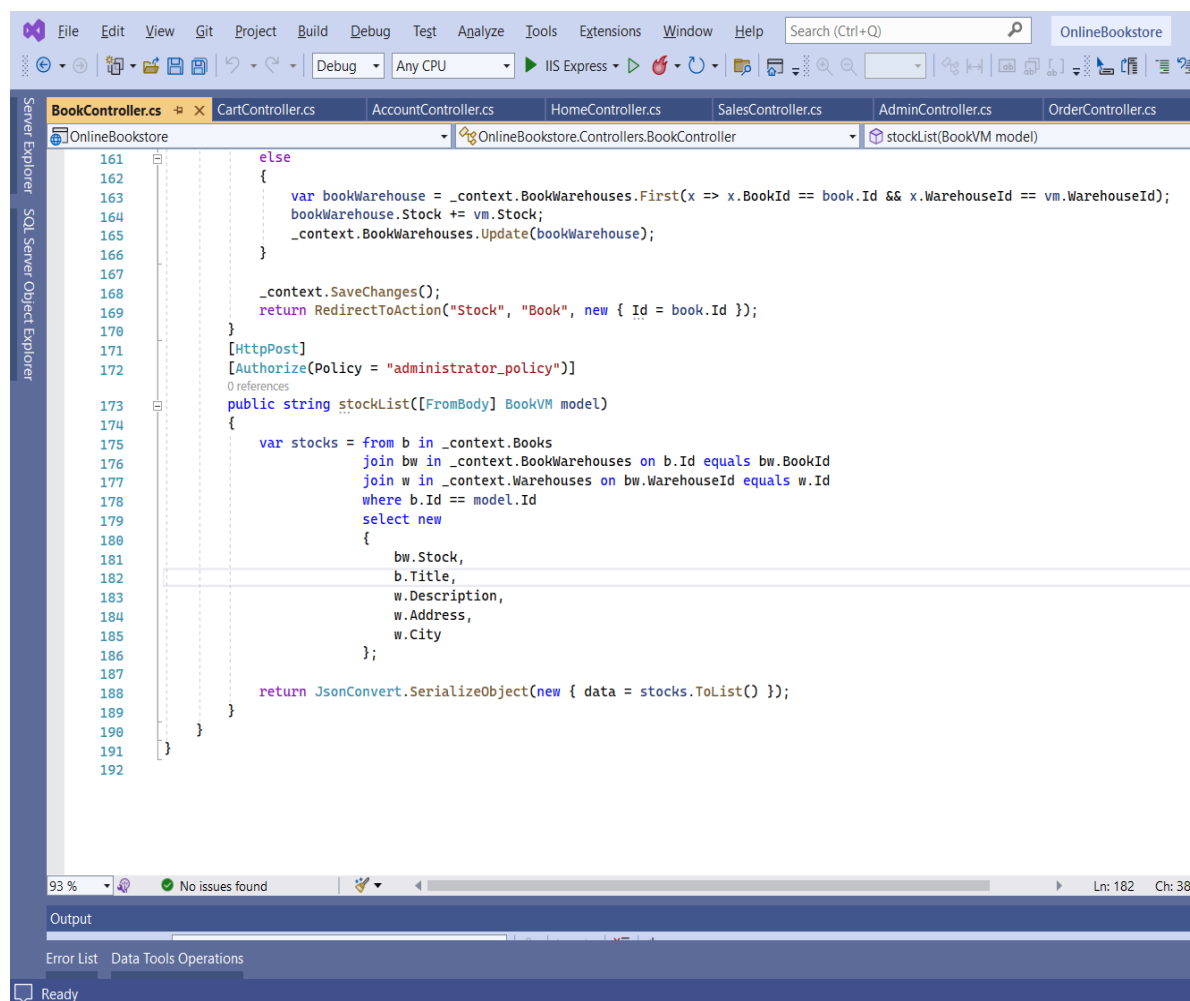
```

Εικόνα 16 BookController_4

Το 1ο action AddStock καλείται όταν πατήσουμε το κουμπί add Stock και παίρνει από την βάση την πληροφορία του βιβλίου (id , τίτλο, ISBN)και τη λίστα με τις αποθήκες.

Στο 2ο action AddStock ,πατώντας το κουμπί Add στην οθόνη AddStock, προσθέτουμε απόθεμα σε κάποιο βιβλίο και γίνεται υποβολή (submit) η φόρμα και καλείται το παρακάτω action. Ο ενδιαμέσος πίνακας BookWarehouses κρατάει την πληροφορία αποθέματος ανά βιβλίο και ανά αποθήκη. Αν υπάρχει εγγραφή στον πίνακα για το συγκεκριμένο βιβλίο κάνουμε Update τον πίνακα με το απόθεμα που βάλαμε (BookWarehouses Update).

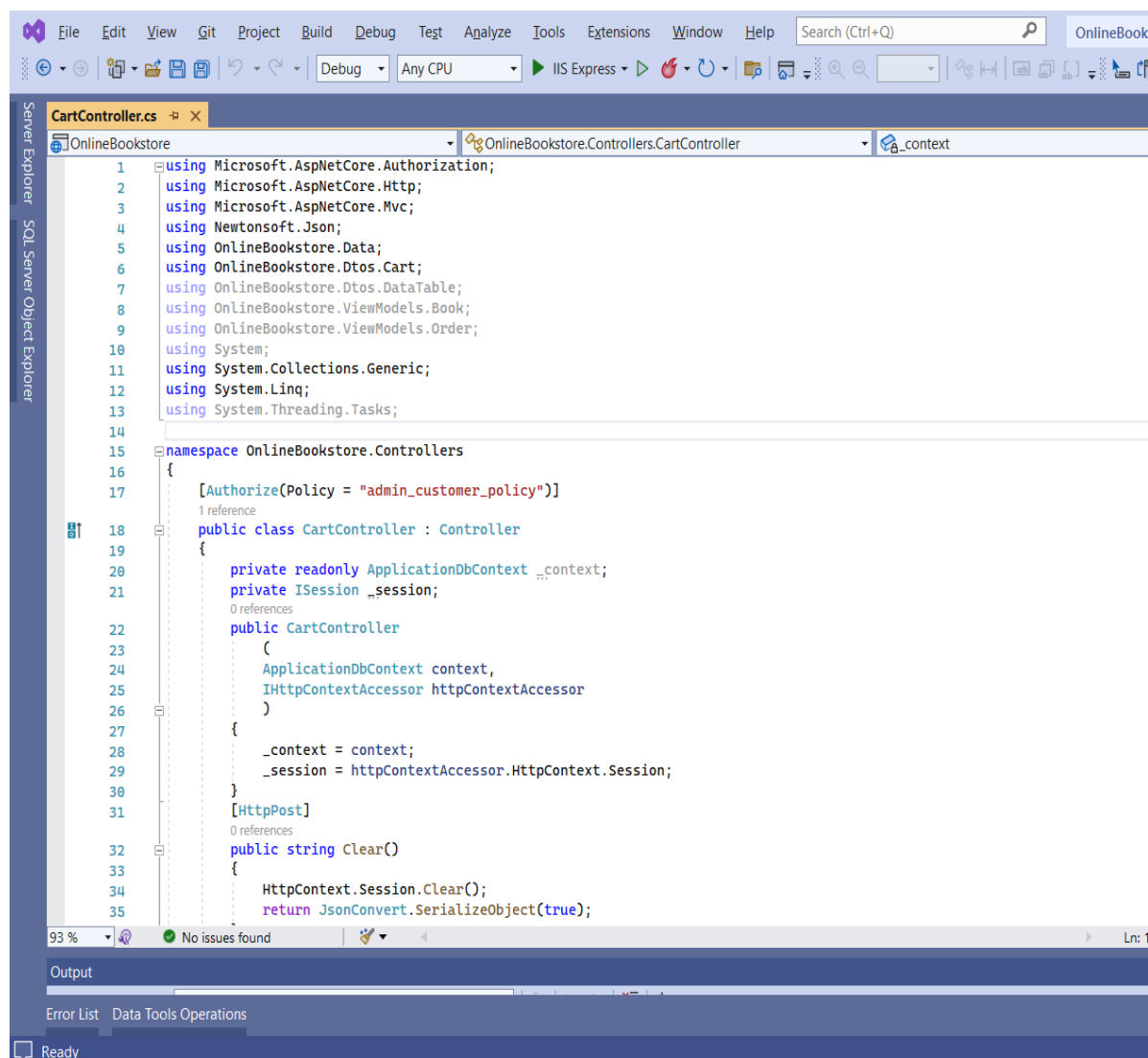
Αν δεν υπάρχει εγγραφή κάνουμε Insert (_context.BookWarehouses.Add).



```
161         else
162         {
163             var bookWarehouse = _context.BookWarehouses.First(x => x.BookId == book.Id && x.WarehouseId == vm.WarehouseId);
164             bookWarehouse.Stock += vm.Stock;
165             _context.BookWarehouses.Update(bookWarehouse);
166         }
167
168         _context.SaveChanges();
169         return RedirectToAction("Stock", "Book", new { Id = book.Id });
170     }
171     [HttpPost]
172     [Authorize(Policy = "administrator_policy")]
173     public string stockList([FromBody] BookVM model)
174     {
175         var stocks = from b in _context.Books
176                     join bw in _context.BookWarehouses on b.Id equals bw.BookId
177                     join w in _context.Warehouses on bw.WarehouseId equals w.Id
178                     where b.Id == model.Id
179                     select new
180                     {
181                         bw.Stock,
182                         b.Title,
183                         w.Description,
184                         w.Address,
185                         w.City
186                     };
187
188         return JsonConvert.SerializeObject(new { data = stocks.ToList() });
189     }
190 }
191
192 }
```

Εικόνα 17 BookController_5

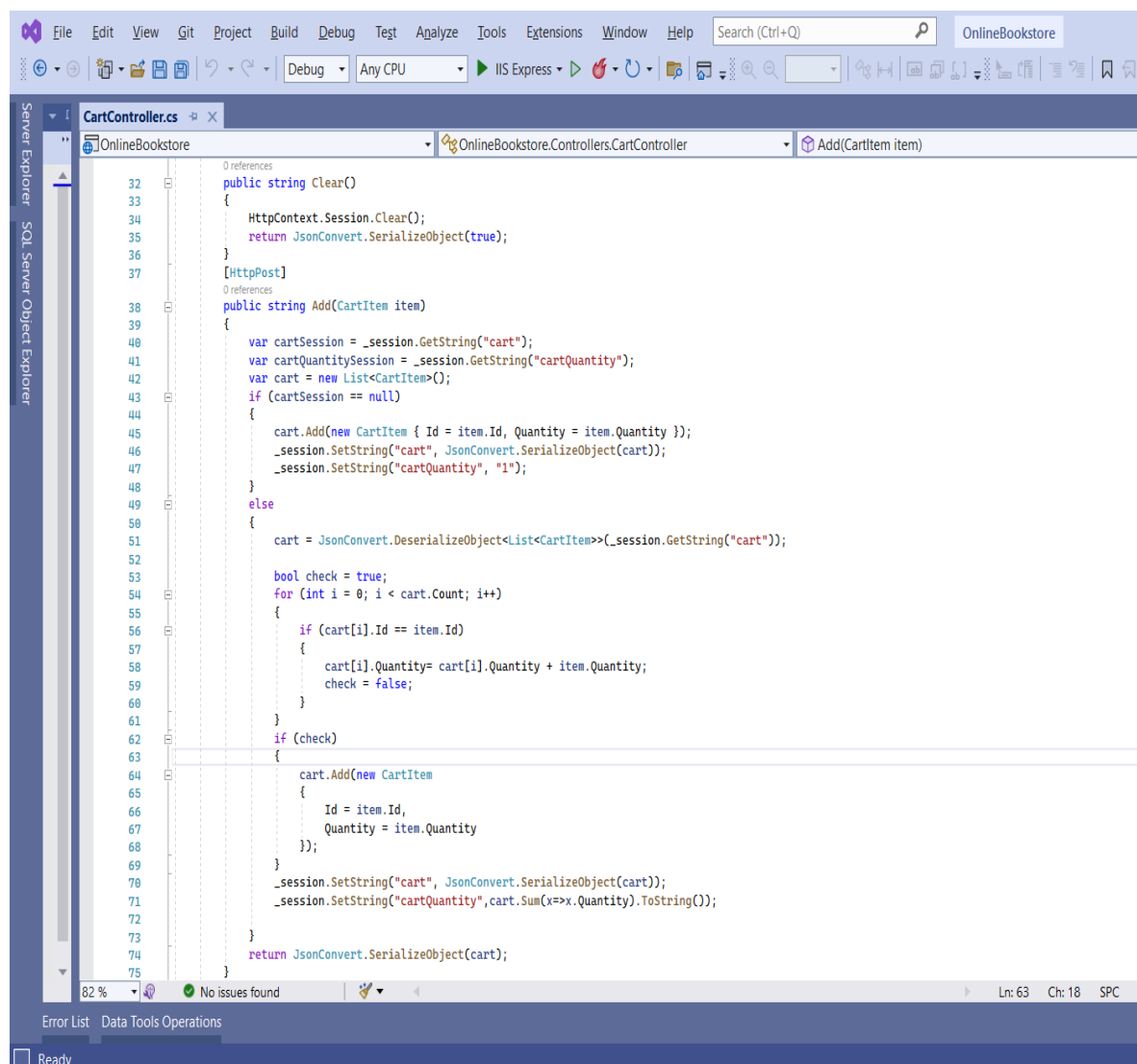
Στην οθόνη με τα αποθέματα ενός βιβλίου, καλείται αυτό το Request και μας φέρνει όλες τις αποθήκες και το απόθεμα για το συγκεκριμένο βιβλίο.



```
1 using Microsoft.AspNetCore.Authorization;
2 using Microsoft.AspNetCore.Http;
3 using Microsoft.AspNetCore.Mvc;
4 using Newtonsoft.Json;
5 using OnlineBookstore.Data;
6 using OnlineBookstore.Dtos.Cart;
7 using OnlineBookstore.Dtos.DataTable;
8 using OnlineBookstore.ViewModels.Book;
9 using OnlineBookstore.ViewModels.Order;
10 using System;
11 using System.Collections.Generic;
12 using System.Linq;
13 using System.Threading.Tasks;
14
15 namespace OnlineBookstore.Controllers
16 {
17     [Authorize(Policy = "admin_customer_policy")]
18     public class CartController : Controller
19     {
20         private readonly ApplicationDbContext _context;
21         private ISession _session;
22         public CartController
23         (
24             ApplicationDbContext context,
25             IHttpContextAccessor httpContextAccessor
26         )
27         {
28             _context = context;
29             _session = httpContextAccessor.HttpContext.Session;
30         }
31         [HttpPost]
32         public string Clear()
33         {
34             HttpContext.Session.Clear();
35             return JsonConvert.SerializeObject(true);
36         }
37     }
38 }
```

Εικόνα 18 CartController_1

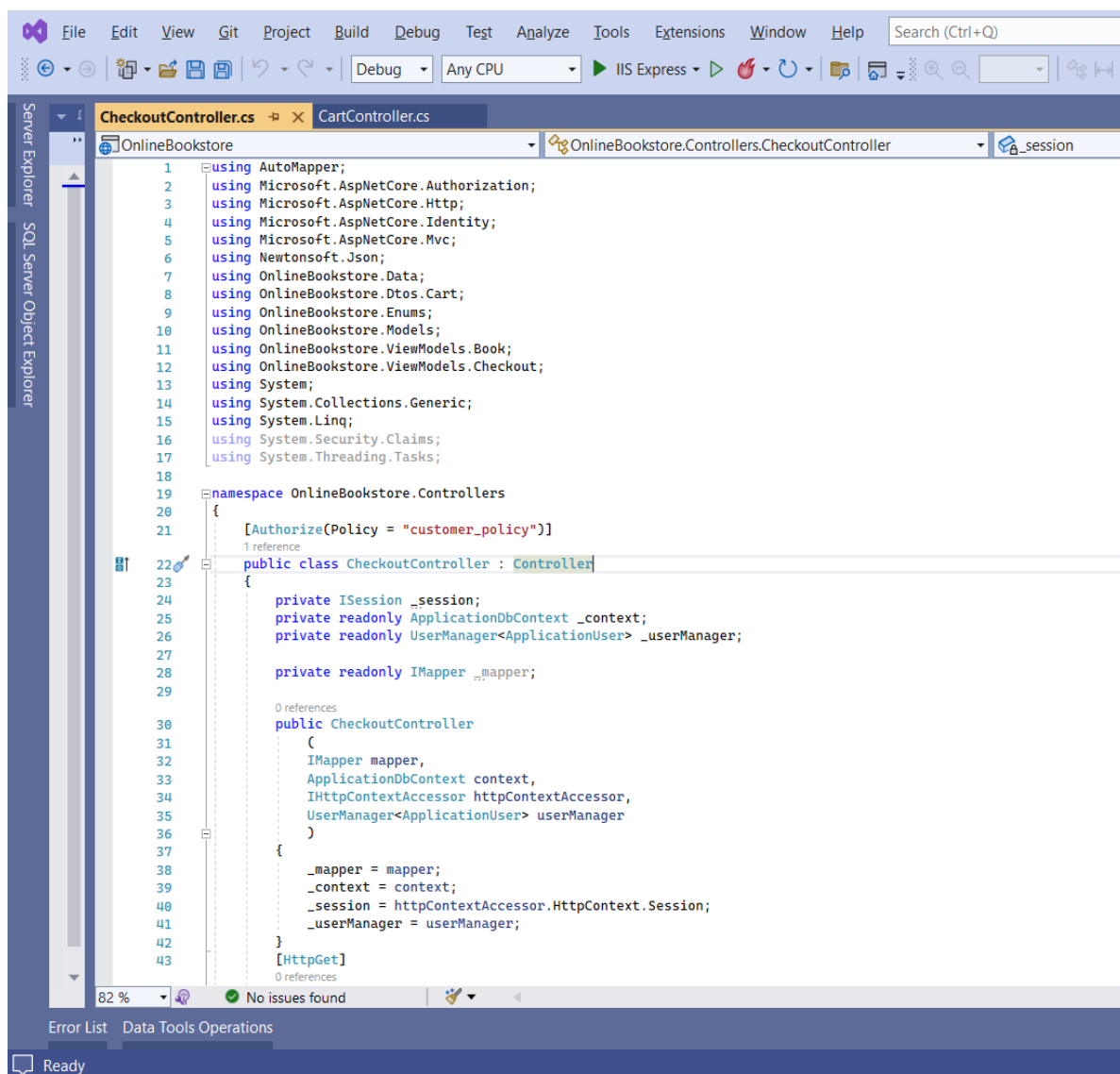
Η μέθοδος Clear καθαρίζει το Session που κρατάει τα δεδομένα του καλαθιού.



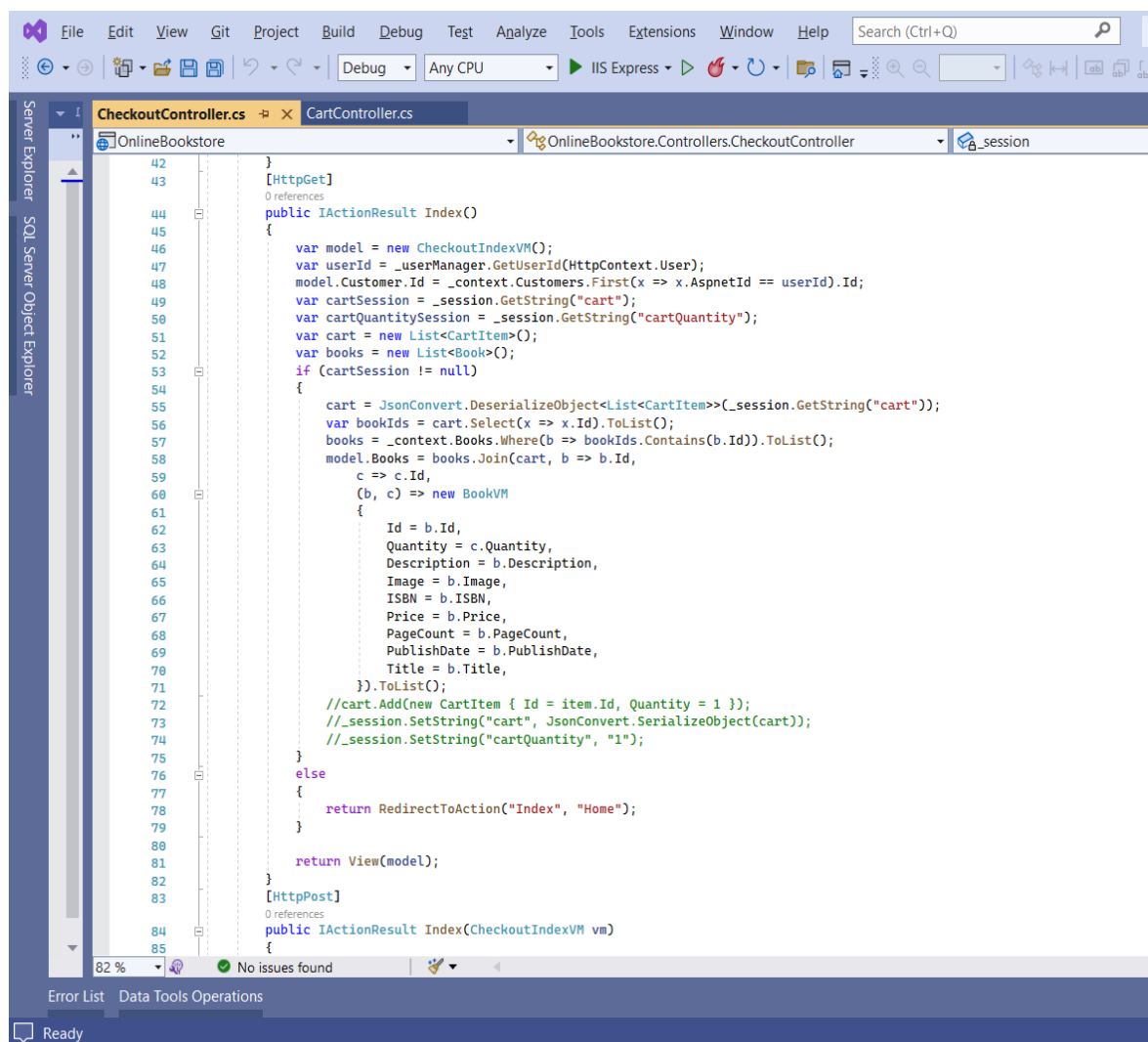
```
32 public string Clear()
33 {
34     HttpContext.Session.Clear();
35     return JsonConvert.SerializeObject(true);
36 }
37 [HttpPost]
38 public string Add(CartItem item)
39 {
40     var cartSession = _session.GetString("cart");
41     var cartQuantitySession = _session.GetString("cartQuantity");
42     var cart = new List<CartItem>();
43     if (cartSession == null)
44     {
45         cart.Add(new CartItem { Id = item.Id, Quantity = item.Quantity });
46         _session.SetString("cart", JsonConvert.SerializeObject(cart));
47         _session.SetString("cartQuantity", "1");
48     }
49     else
50     {
51         cart = JsonConvert.DeserializeObject<List<CartItem>>(_session.GetString("cart"));
52
53         bool check = true;
54         for (int i = 0; i < cart.Count; i++)
55         {
56             if (cart[i].Id == item.Id)
57             {
58                 cart[i].Quantity = cart[i].Quantity + item.Quantity;
59                 check = false;
60             }
61         }
62         if (check)
63         {
64             cart.Add(new CartItem
65             {
66                 Id = item.Id,
67                 Quantity = item.Quantity
68             });
69         }
70         _session.SetString("cart", JsonConvert.SerializeObject(cart));
71         _session.SetString("cartQuantity", cart.Sum(x=>x.Quantity).ToString());
72     }
73     return JsonConvert.SerializeObject(cart);
74 }
75 }
```

Εικόνα 19 CartController_2

Η συνάρτηση Add αποθηκεύει στο Session την πληροφορία των βιβλίων που προσθέτει ο πελάτης στο καλάθι δηλαδή κρατάει το Id του βιβλίου και την ποσότητα που βάζει ο πελάτης .



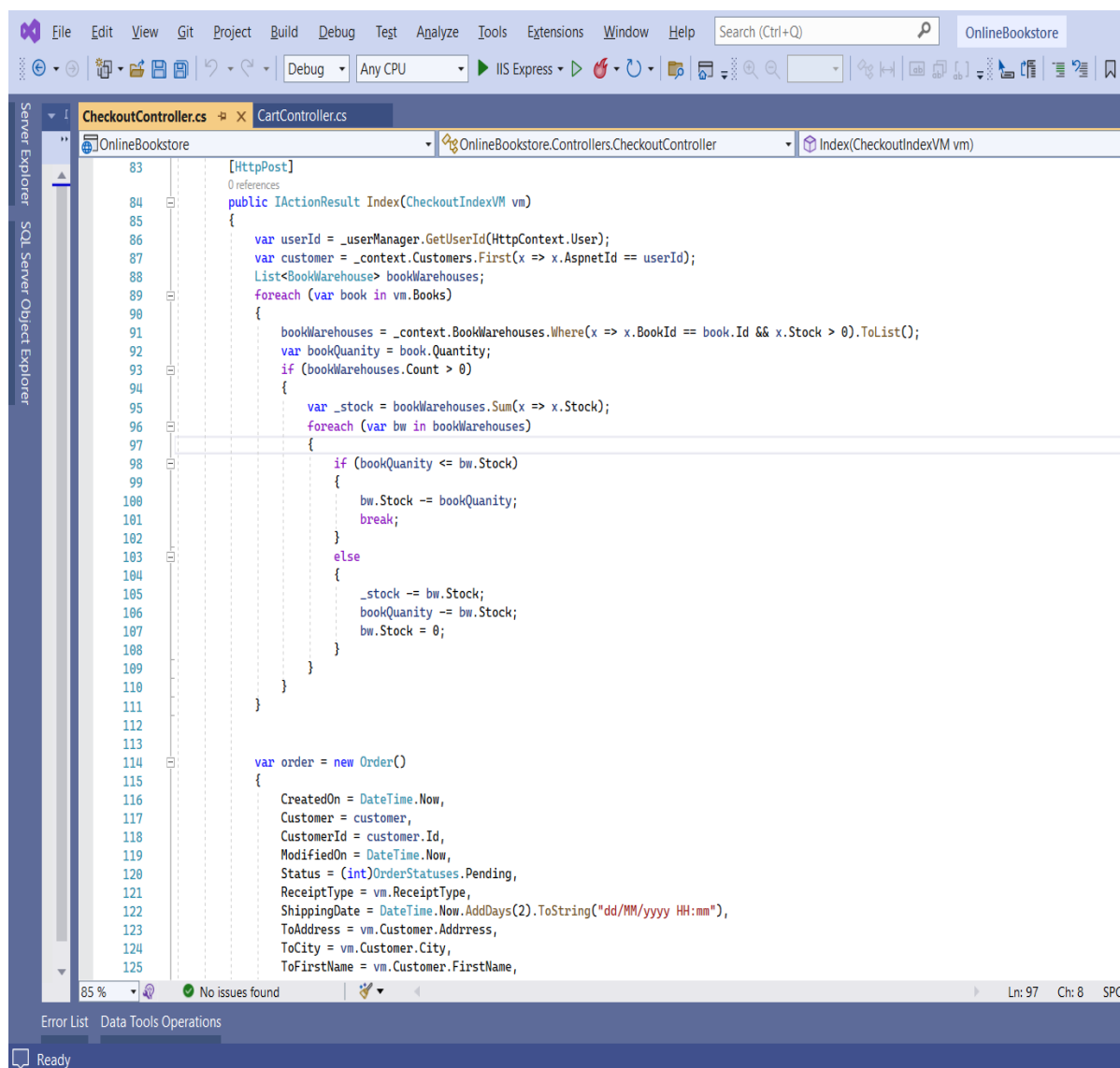
Εικόνα 20 CheckoutController_1



```
42 }
43 [HttpGet]
44 public IActionResult Index()
45 {
46     var model = new CheckoutIndexVM();
47     var userId = _userManager.GetUserId(HttpContext.User);
48     model.Customer.Id = _context.Customers.First(x => x.AspNetId == userId).Id;
49     var cartSession = _session.GetString("cart");
50     var cartQuantitySession = _session.GetString("cartQuantity");
51     var cart = new List<CartItem>();
52     var books = new List<Book>();
53     if (cartSession != null)
54     {
55         cart = JsonConvert.DeserializeObject<List<CartItem>>(_session.GetString("cart"));
56         var bookIds = cart.Select(x => x.Id).ToList();
57         books = _context.Books.Where(b => bookIds.Contains(b.Id)).ToList();
58         model.Books = books.Join(cart, b => b.Id,
59             c => c.Id,
60             (b, c) => new BookVM
61             {
62                 Id = b.Id,
63                 Quantity = c.Quantity,
64                 Description = b.Description,
65                 Image = b.Image,
66                 ISBN = b.ISBN,
67                 Price = b.Price,
68                 PageCount = b.PageCount,
69                 PublishDate = b.PublishDate,
70                 Title = b.Title,
71             }).ToList();
72         //cart.Add(new CartItem { Id = item.Id, Quantity = 1 });
73         //_session.SetString("cart", JsonConvert.SerializeObject(cart));
74         //_session.SetString("cartQuantity", "1");
75     }
76     else
77     {
78         return RedirectToAction("Index", "Home");
79     }
80     return View(model);
81 }
82 [HttpPost]
83 public IActionResult Index(CheckoutIndexVM vm)
84 {
85 }
```

Εικόνα 21 CheckoutController_2

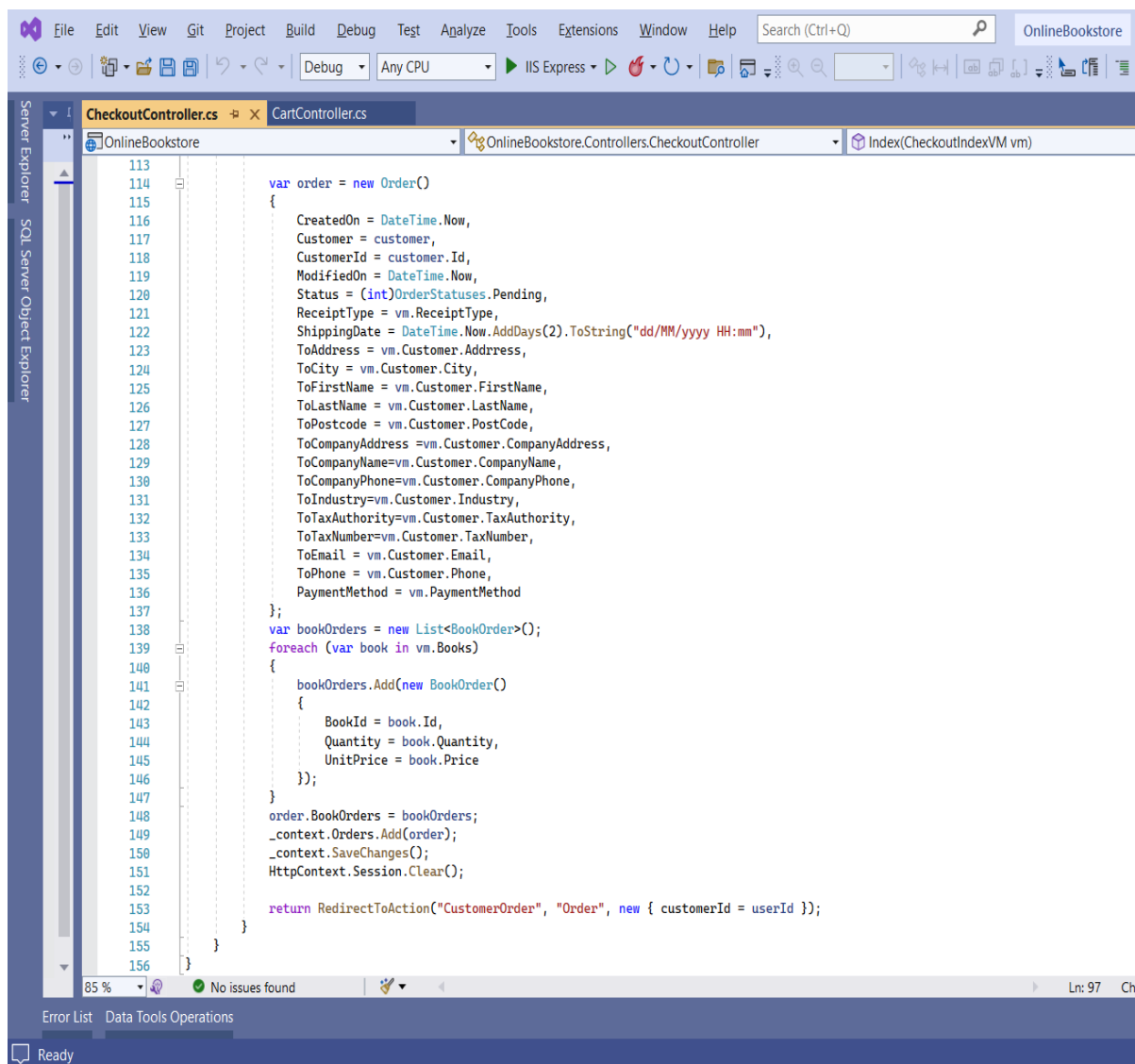
Το Action Index φορτώνει όλη την πληροφορία που χρειάζεται η Checkout σελίδα. Το query διαβάζει τα βιβλία που έχουν προστεθεί στο καλάθι μέσω του Session και παίρνει τα δεδομένα των βιβλίων αυτών.



```
83 [HttpPost]
84 public IActionResult Index(CheckoutIndexVM vm)
85 {
86     var userId = _userManager.GetUserId(HttpContext.User);
87     var customer = _context.Customers.First(x => x.AspNetId == userId);
88     List<BookWarehouse> bookWarehouses;
89     foreach (var book in vm.Books)
90     {
91         bookWarehouses = _context.BookWarehouses.Where(x => x.BookId == book.Id && x.Stock > 0).ToList();
92         var bookQuantity = book.Quantity;
93         if (bookWarehouses.Count > 0)
94         {
95             var _stock = bookWarehouses.Sum(x => x.Stock);
96             foreach (var bw in bookWarehouses)
97             {
98                 if (bookQuantity <= bw.Stock)
99                 {
100                     bw.Stock -= bookQuantity;
101                     break;
102                 }
103                 else
104                 {
105                     _stock -= bw.Stock;
106                     bookQuantity -= bw.Stock;
107                     bw.Stock = 0;
108                 }
109             }
110         }
111     }
112
113     var order = new Order()
114     {
115         CreatedOn = DateTime.Now,
116         Customer = customer,
117         CustomerId = customer.Id,
118         ModifiedOn = DateTime.Now,
119         Status = (int)OrderStatuses.Pending,
120         ReceiptType = vm.ReceiptType,
121         ShippingDate = DateTime.Now.AddDays(2).ToString("dd/MM/yyyy HH:mm"),
122         ToAddress = vm.Customer.Address,
123         ToCity = vm.Customer.City,
124         ToFirstName = vm.Customer.FirstName,
```

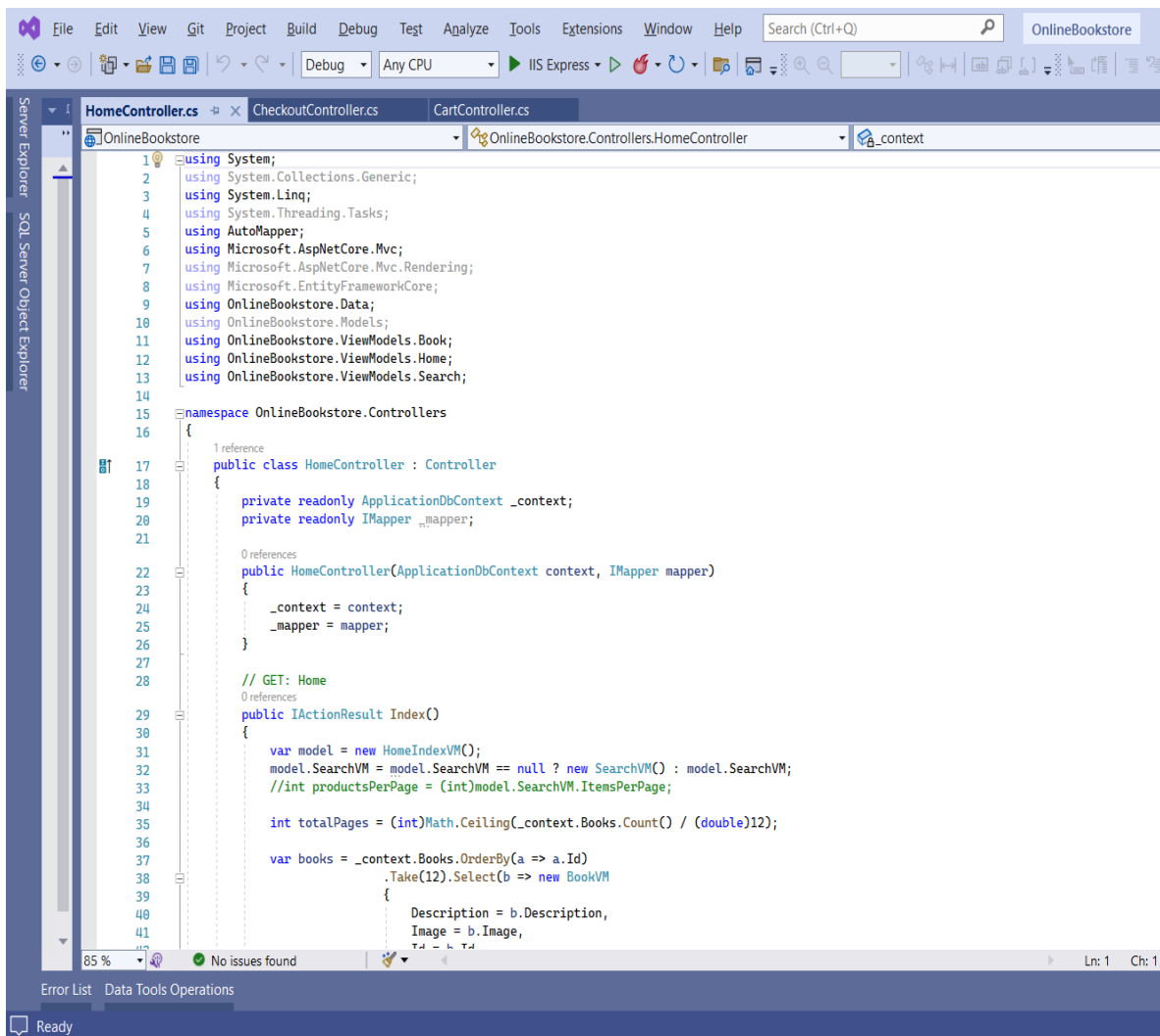
Εικόνα 22 CheckoutController_3

Το παραπάνω action καλείται όταν ο χρήστης πατήσει το κουμπί ContinueToCheckout. Αρχικά μειώνει το απόθεμα των βιβλίων που έκανε Checkout ο πελάτης. Κατά δεύτερον αποθηκεύει την παραγγελία στη βάση με όλα τα στοιχεία που συμπλήρωσε ο χρήστης στο Checkout καθώς και τα βιβλία της παραγγελίας, την ποσότητα τους και την τιμή μονάδας τους την συγκεκριμένη στιγμή.

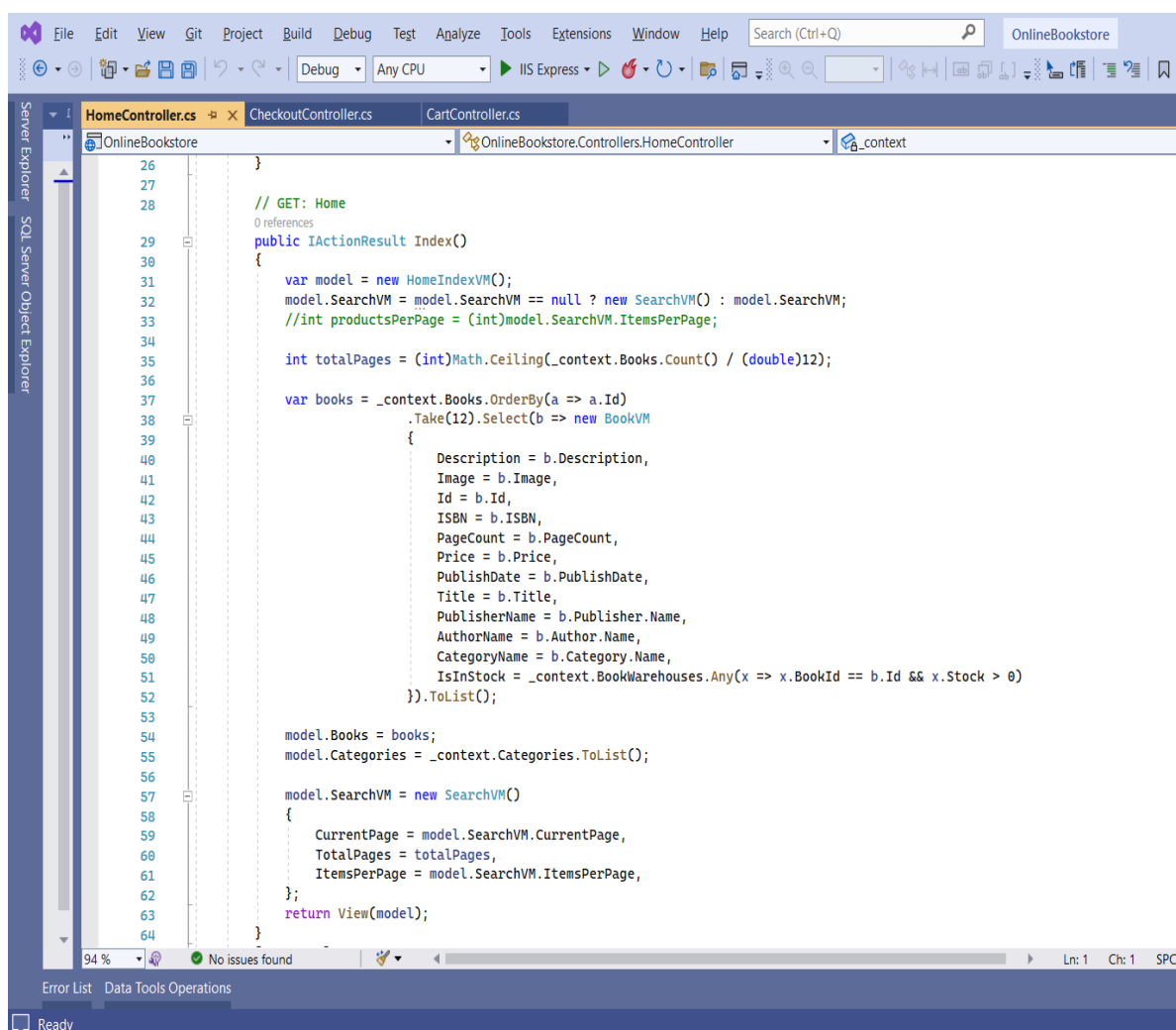


```
113
114
115
116     var order = new Order()
117     {
118         CreatedOn = DateTime.Now,
119         Customer = customer,
120         CustomerId = customer.Id,
121         ModifiedOn = DateTime.Now,
122         Status = (int)OrderStatuses.Pending,
123         ReceiptType = vm.ReceiptType,
124         ShippingDate = DateTime.Now.AddDays(2).ToString("dd/MM/yyyy HH:mm"),
125         ToAddress = vm.Customer.Address,
126         ToCity = vm.Customer.City,
127         ToFirstName = vm.Customer.FirstName,
128         ToLastName = vm.Customer.LastName,
129         ToPostcode = vm.Customer.PostCode,
130         ToCompanyAddress = vm.Customer.CompanyAddress,
131         ToCompanyName = vm.Customer.CompanyName,
132         ToCompanyPhone = vm.Customer.CompanyPhone,
133         ToIndustry = vm.Customer.Industry,
134         ToTaxAuthority = vm.Customer.TaxAuthority,
135         ToTaxNumber = vm.Customer.TaxNumber,
136         ToEmail = vm.Customer.Email,
137         ToPhone = vm.Customer.Phone,
138         PaymentMethod = vm.PaymentMethod
139     };
140
141     var bookOrders = new List<BookOrder>();
142     foreach (var book in vm.Books)
143     {
144         bookOrders.Add(new BookOrder()
145         {
146             BookId = book.Id,
147             Quantity = book.Quantity,
148             UnitPrice = book.Price
149         });
150     }
151     order.BookOrders = bookOrders;
152     _context.Orders.Add(order);
153     _context.SaveChanges();
154     HttpContext.Session.Clear();
155
156     return RedirectToAction("CustomerOrder", "Order", new { customerId = userId });
157 }
```

Εικόνα 23 CheckoutController_4



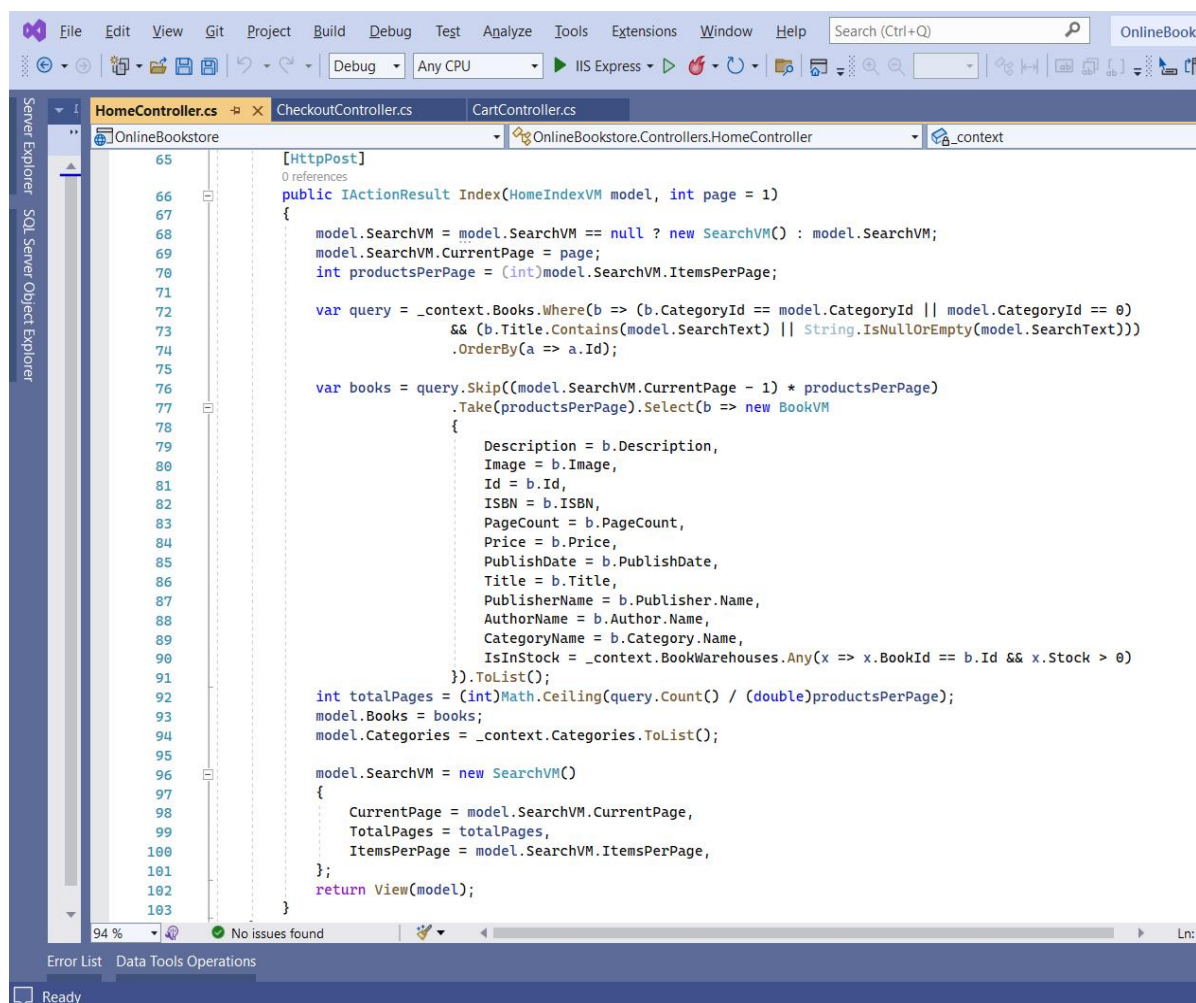
Εικόνα 24 HomeController_1



```
26     }
27
28     // GET: Home
29     public IActionResult Index()
30     {
31         var model = new HomeIndexVM();
32         model.SearchVM = model.SearchVM == null ? new SearchVM() : model.SearchVM;
33         //int productsPerPage = (int)model.SearchVM.ItemsPerPage;
34
35         int totalPages = (int)Math.Ceiling(_context.Books.Count() / (double)12);
36
37         var books = _context.Books.OrderBy(a => a.Id)
38             .Take(12).Select(b => new BookVM
39             {
40                 Description = b.Description,
41                 Image = b.Image,
42                 Id = b.Id,
43                 ISBN = b.ISBN,
44                 PageCount = b.PageCount,
45                 Price = b.Price,
46                 PublishDate = b.PublishDate,
47                 Title = b.Title,
48                 PublisherName = b.Publisher.Name,
49                 AuthorName = b.Author.Name,
50                 CategoryName = b.Category.Name,
51                 IsInStock = _context.BookWarehouses.Any(x => x.BookId == b.Id && x.Stock > 0)
52             }).ToList();
53
54         model.Books = books;
55         model.Categories = _context.Categories.ToList();
56
57         model.SearchVM = new SearchVM()
58         {
59             CurrentPage = model.SearchVM.CurrentPage,
60             TotalPages = totalPages,
61             ItemsPerPage = model.SearchVM.ItemsPerPage,
62         };
63         return View(model);
64     }
```

Εικόνα 25 HomeController_2

Το παραπάνω καλείται για να φορτώσει την αρχική σελίδα. Κάνει κλήση στη βάση και το αποτέλεσμα του query θα εμφανίσει τα 12 πρώτα βιβλία.



```
[HttpPost]
public IActionResult Index(HomeIndexVM model, int page = 1)
{
    model.SearchVM = model.SearchVM == null ? new SearchVM() : model.SearchVM;
    model.SearchVM.CurrentPage = page;
    int productsPerPage = (int)model.SearchVM.ItemsPerPage;

    var query = _context.Books.Where(b => (b.CategoryId == model.CategoryId || model.CategoryId == 0)
        && (b.Title.Contains(model.SearchText) || String.IsNullOrEmpty(model.SearchText)))
        .OrderBy(a => a.Id);

    var books = query.Skip((model.SearchVM.CurrentPage - 1) * productsPerPage)
        .Take(productsPerPage).Select(b => new BookVM
        {
            Description = b.Description,
            Image = b.Image,
            Id = b.Id,
            ISBN = b.ISBN,
            PageCount = b.PageCount,
            Price = b.Price,
            PublishDate = b.PublishDate,
            Title = b.Title,
            PublisherName = b.Publisher.Name,
            AuthorName = b.Author.Name,
            CategoryName = b.Category.Name,
            IsInStock = _context.BookWarehouses.Any(x => x.BookId == b.Id && x.Stock > 0)
        }).ToList();

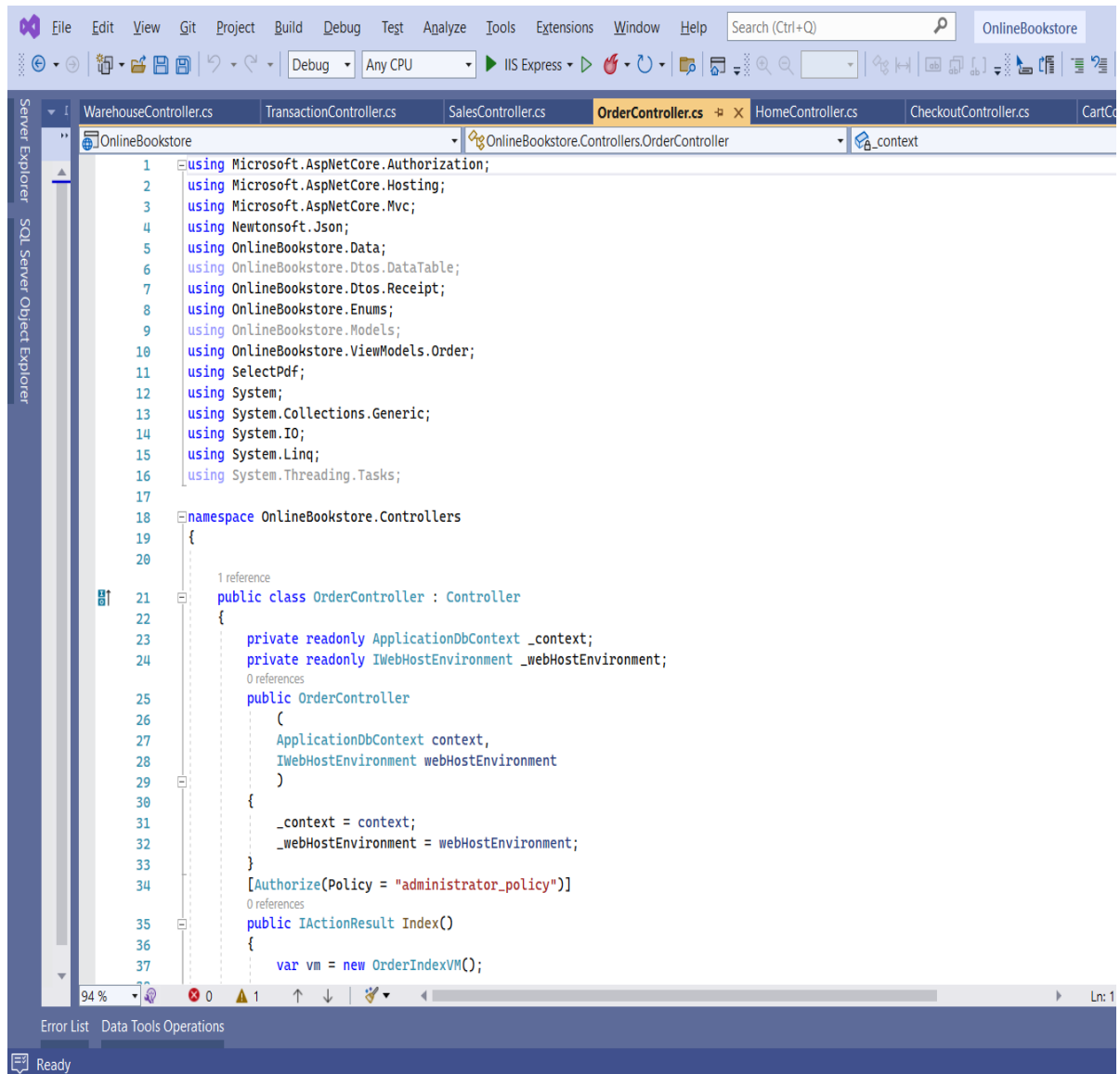
    int totalPages = (int)Math.Ceiling(query.Count() / (double)productsPerPage);
    model.Books = books;
    model.Categories = _context.Categories.ToList();

    model.SearchVM = new SearchVM()
    {
        CurrentPage = model.SearchVM.CurrentPage,
        TotalPages = totalPages,
        ItemsPerPage = model.SearchVM.ItemsPerPage,
    };
    return View(model);
}
```

Εικόνα 26 HomeController_3

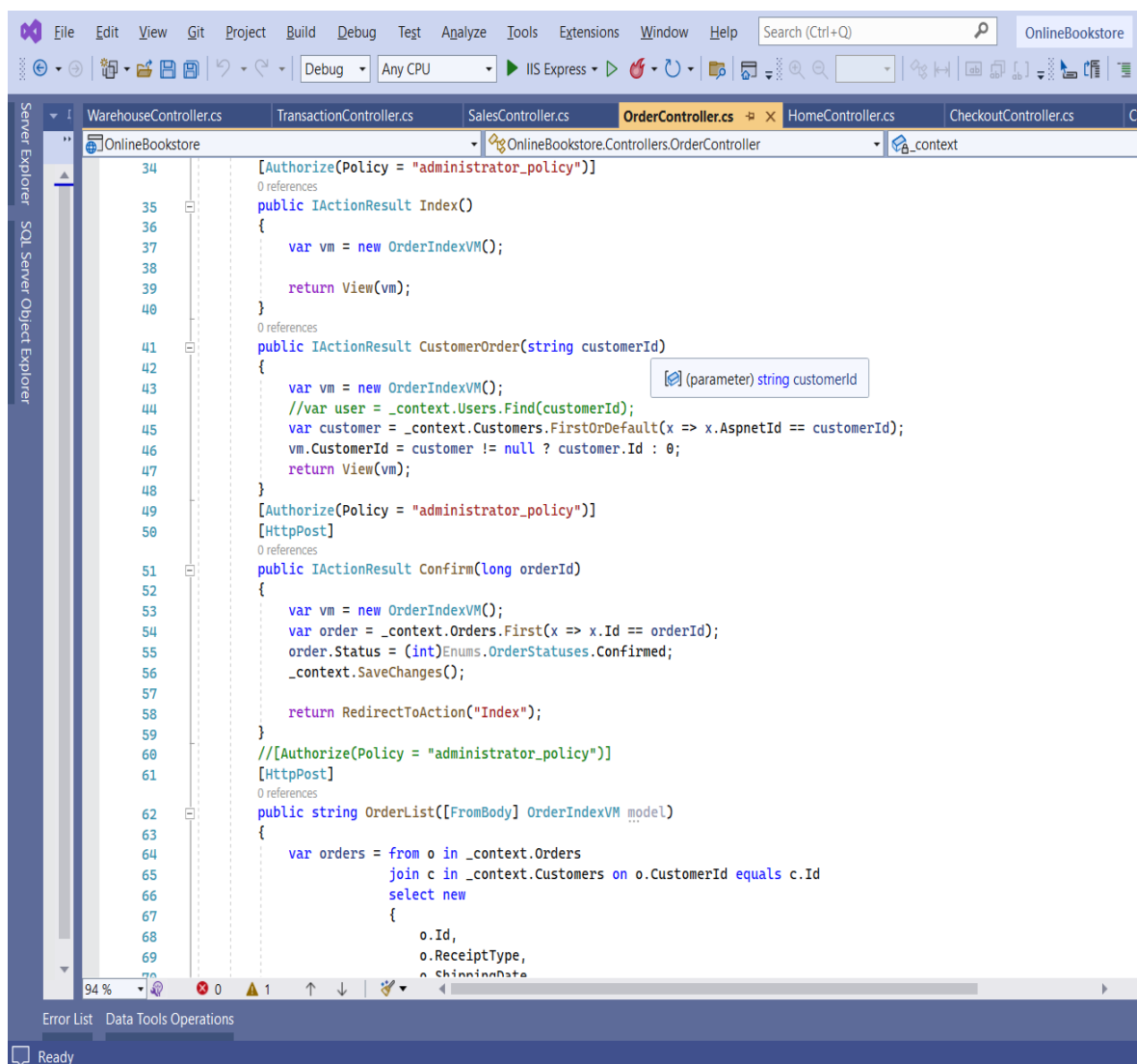
Το παραπάνω action Index καλείται όταν ο χρήστης κλικάρει το Search κουμπί στην αρχική οθόνη ή όταν πατήσει την σελίδα στο κάτω μέρος. Το query επιστρέφει τα 12 βιβλία της σελίδας που έχει επιλέξει και της κατηγορίας που έχει επιλέξει.

Επιπρόσθετα ελέγχει εάν ο τίτλος του βιβλίου περιέχει τη λέξη που βάζει στο Search πεδίο.



```
1 using Microsoft.AspNetCore.Authorization;
2 using Microsoft.AspNetCore.Hosting;
3 using Microsoft.AspNetCore.Mvc;
4 using Newtonsoft.Json;
5 using OnlineBookstore.Data;
6 using OnlineBookstore.Dtos.DataTable;
7 using OnlineBookstore.Dtos.Receipt;
8 using OnlineBookstore.Enums;
9 using OnlineBookstore.Models;
10 using OnlineBookstore.ViewModels.Order;
11 using SelectPdf;
12 using System;
13 using System.Collections.Generic;
14 using System.IO;
15 using System.Linq;
16 using System.Threading.Tasks;
17
18 namespace OnlineBookstore.Controllers
19 {
20
21     1 reference
22     public class OrderController : Controller
23     {
24         private readonly ApplicationDbContext _context;
25         private readonly IWebHostEnvironment _webHostEnvironment;
26         0 references
27         public OrderController
28         (
29             ApplicationDbContext context,
30             IWebHostEnvironment webHostEnvironment
31         )
32         {
33             _context = context;
34             _webHostEnvironment = webHostEnvironment;
35         }
36         [Authorize(Policy = "administrator_policy")]
37         0 references
38         public IActionResult Index()
39         {
40             var vm = new OrderIndexVM();
41         }
42     }
43 }
```

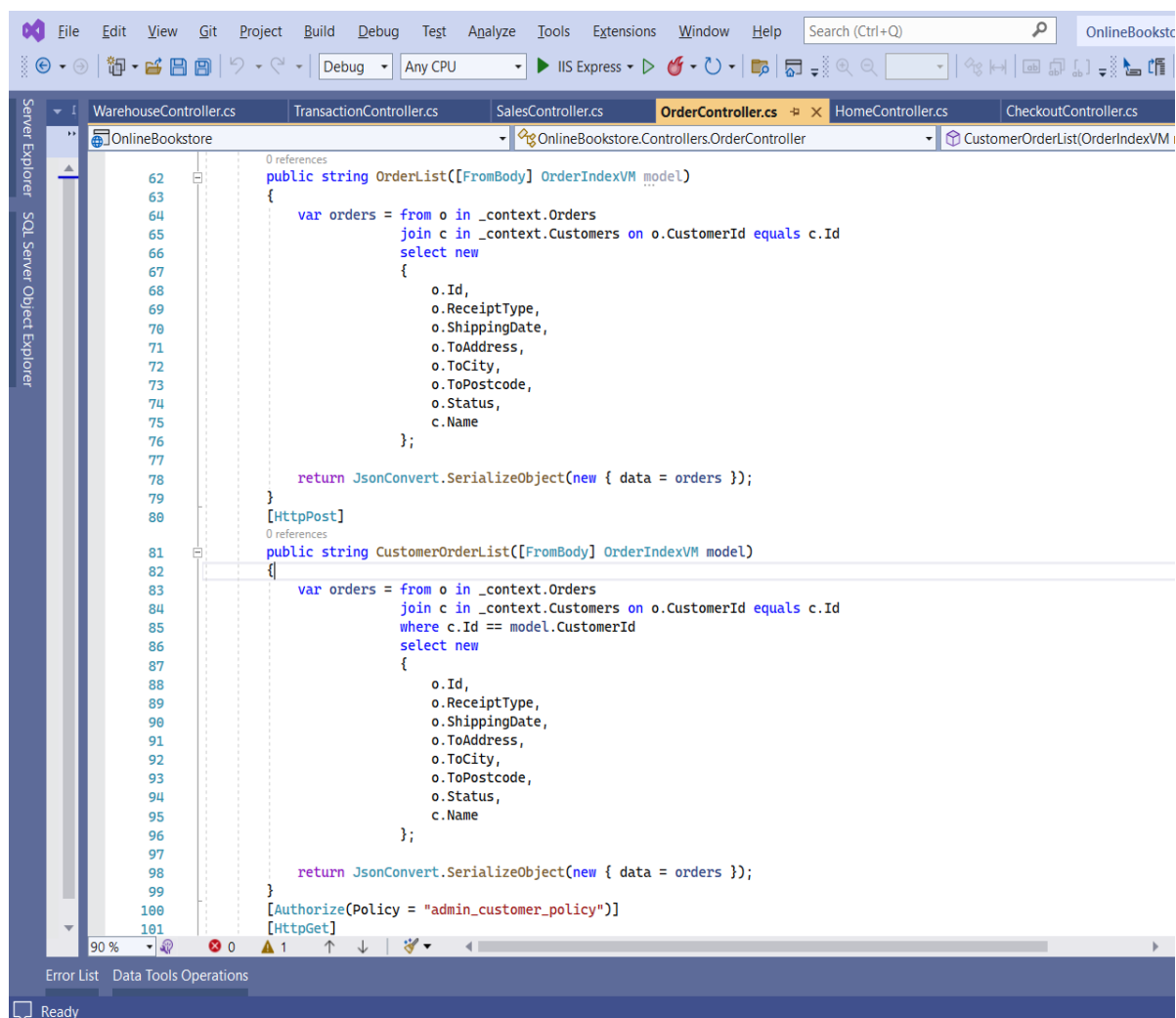
Εικόνα 27 OrderController_1



```
34 [Authorize(Policy = "administrator_policy")]
35 0 references
36 public IActionResult Index()
37 {
38     var vm = new OrderIndexVM();
39
40     return View(vm);
41 }
42 0 references
43 public IActionResult CustomerOrder(string customerId)
44 {
45     var vm = new OrderIndexVM();
46     //var user = _context.Users.Find(customerId);
47     var customer = _context.Customers.FirstOrDefault(x => x.AspNetId == customerId);
48     vm.CustomerId = customer != null ? customer.Id : 0;
49     return View(vm);
50 }
51 [Authorize(Policy = "administrator_policy")]
52 [HttpPost]
53 0 references
54 public IActionResult Confirm(long orderId)
55 {
56     var vm = new OrderIndexVM();
57     var order = _context.Orders.First(x => x.Id == orderId);
58     order.Status = (int)Enums.OrderStatuses.Confirmed;
59     _context.SaveChanges();
60
61     return RedirectToAction("Index");
62 }
63 // [Authorize(Policy = "administrator_policy")]
64 [HttpPost]
65 0 references
66 public string OrderList([FromBody] OrderIndexVM model)
67 {
68     var orders = from o in _context.Orders
69                 join c in _context.Customers on o.CustomerId equals c.Id
70                 select new
71                 {
72                     o.Id,
73                     o.ReceiptType,
74                     o.ShippingDate
75                 }
76 }
```

Εικόνα 28 OrderController_2

Το Action Confirm μεταβάλλει το status της παραγγελίας σε confirmed .



```
62 public string OrderList([FromBody] OrderIndexVM model)
63 {
64     var orders = from o in _context.Orders
65                 join c in _context.Customers on o.CustomerId equals c.Id
66                 select new
67                 {
68                     o.Id,
69                     o.ReceiptType,
70                     o.ShippingDate,
71                     o.ToAddress,
72                     o.ToCity,
73                     o.ToPostcode,
74                     o.Status,
75                     c.Name
76                 };
77     return JsonConvert.SerializeObject(new { data = orders });
78 }
79 [HttpPost]
80
81 public string CustomerOrderList([FromBody] OrderIndexVM model)
82 {
83     var orders = from o in _context.Orders
84                 join c in _context.Customers on o.CustomerId equals c.Id
85                 where c.Id == model.CustomerId
86                 select new
87                 {
88                     o.Id,
89                     o.ReceiptType,
90                     o.ShippingDate,
91                     o.ToAddress,
92                     o.ToCity,
93                     o.ToPostcode,
94                     o.Status,
95                     c.Name
96                 };
97     return JsonConvert.SerializeObject(new { data = orders });
98 }
99 [Authorize(Policy = "admin_customer_policy")]
100 [HttpGet]
101
```

Εικόνα 29 OrderController_3

Η μέθοδος OrderList τρέχοντας το κατάλληλο query στη βάση, δημιουργεί τη λίστα όλων των παραγγελιών.

Η μέθοδος CustomerOrderList ,τρέχοντας το κατάλληλο query στη βάση , δημιουργεί τη λίστα όλων των παραγγελιών ενός πελάτη.

```

100 [Authorize(Policy = "admin_customer_policy")]
101 [HttpGet]
102 public IActionResult PrintReceipt(long orderId)
103 {
104
105     var receipt =
106     (from o in _context.Orders
107     join c in _context.Customers on o.CustomerId equals c.Id
108     join sub in (from bo in _context.BookOrders
109                 join o in _context.Orders on bo.OrderId equals o.Id
110                 group bo by (new { o.Id }) into g
111                 select new { g.Key.Id, TotalPrice = g.Sum(bo => bo.Quantity * bo.UnitPrice), TotalQuantity = g.Sum(bo => bo.Quantity) })
112     on o.Id equals sub.Id
113     where o.Id == orderId
114     select new ReceiptDto()
115     {
116         CustomerAddress = o.ToAddress,
117         CustomerEmail = o.ToEmail,
118         CustomerName = o.ToFirstName + " " + o.ToLastName,
119         CustomerPhone = o.ToPhone,
120         CompanyName = o.ToCompanyName,
121         CompanyAddress = o.ToCompanyAddress,
122         CompanyPhone = o.ToCompanyPhone,
123         Industry = o.ToIndustry,
124         TaxAuthority = o.ToTaxAuthority,
125         TaxNumber = o.ToTaxNumber,
126         CustomerCode = c.AspNetId,
127         CreateDate = o.CreatedOn,
128         ReceiptType = o.ReceiptType,
129         ReceiptNumber = o.ReceiptType == (int)ReceiptTypes.Retail ? "ΑΠΛΗ" + o.Id.ToString() : "ΤΔΑΑ" + o.Id.ToString(),
130         ReceiptSeries = o.Id.ToString(),
131         TotalPriceWithVAT = sub.TotalPrice,
132         TotalPriceWithoutVAT = Math.Round(sub.TotalPrice * (decimal)(1 - 0.06), 2),
133         TotalVAT = Math.Round(sub.TotalPrice * (decimal)0.06, 2),
134         PaymentMethod = o.PaymentMethod
135     }).First();
136
137     var products = _context.BookOrders
138     .Join(_context.Orders, bo => bo.OrderId,
139     o => o.Id,
140     (bo, o) => new { bo, o })
141     .Join(_context.Customers, a => a.CustomerId,
142     c => c.Id,
143     (a, c) => new { a, c })
144     .Join(_context.Books, ab => ab.a.bo.BookId,
145     b => b.Id,

```

Εικόνα 30 OrderController_4

Στο Action PrintReceipt υπάρχουν 2 queries.

Το πρώτο query receipt ανακτά από τη βάση, όλα τα στοιχεία της παραγγελίας του πελάτη και υπολογίζει την συνολική αξία της παραγγελίας και τη συνολική ποσότητα βιβλίων. Το δεύτερο επιστρέφει την πληροφορία των βιβλίων της παραγγελίας. Τιμή μονάδας με/χωρίς ΦΠΑ, συνολική τιμή ανά κωδικό με/χωρίς ΦΠΑ (αν έχει πάρει παραπάνω αντίτυπα ενός βιβλίου)

```

145         b => b.Id,
146         (ab, b) => new { ab, b })
147         .Where(x => x.ab.a.o.Id == orderId)
148         .Select(x => new ReceiptProductDto()
149         {
150             Code = x.b.ISBN,
151             Description = x.b.Title,
152             Quantity = x.ab.a.bo.Quantity,
153             UnitPriceWithVAT = x.ab.a.bo.UnitPrice,
154             UnitPriceWithoutVAT = Math.Round(x.ab.a.bo.UnitPrice * (decimal)(1 - 0.06), 2),
155             TotalPriceWithVAT = Math.Round(x.ab.a.bo.Quantity * x.ab.a.bo.UnitPrice, 2),
156             TotalPriceWithoutVAT = Math.Round(x.ab.a.bo.Quantity * x.ab.a.bo.UnitPrice * (decimal)(1 - 0.06), 2)
157         }).ToList();
158
159         var webRootPath = _webHostEnvironment.WebRootPath;
160         var path = Path.Combine(webRootPath, "Templates\\RetailReceipt.txt");
161
162         if (receipt.ReceiptType == (int)ReceiptTypes.Invoice)
163         {
164             path = Path.Combine(webRootPath, "Templates\\InvoiceReceipt.txt");
165         }
166
167         var paymentMethod = "Credit Card";
168         if (receipt.PaymentMethod == (int)PaymentMethods.Debit) paymentMethod = "Debit Card";
169         if (receipt.PaymentMethod == (int)PaymentMethods.Paypal) paymentMethod = "Paypal";
170
171         var html = System.IO.File.ReadAllText(path);
172         html = html.Replace("[CUSTOMERCODE]", receipt.CustomerCode);
173         html = html.Replace("[COMPANYNAME]", receipt.CompanyName);
174         html = html.Replace("[COMPANYADDRESS]", receipt.CompanyAddress);
175         html = html.Replace("[COMPANYPHONE]", receipt.CompanyPhone);
176         html = html.Replace("[COMPANYEMAIL]", receipt.CustomerEmail);
177         html = html.Replace("[TAXNUMBER]", receipt.TaxNumber);
178         html = html.Replace("[TAXAUTHORITY]", receipt.TaxAuthority);
179         html = html.Replace("[Industry]", receipt.Industry);
180         html = html.Replace("[CUSTOMEREMAIL]", receipt.CustomerEmail);
181         html = html.Replace("[CUSTOMERPHONE]", receipt.CustomerPhone);
182         html = html.Replace("[CUSTOMERNAME]", receipt.CustomerName);
183         html = html.Replace("[CUSTOMERADDRESS]", receipt.CustomerAddress);
184         html = html.Replace("[CREATEDATE]", receipt.CreateDate.ToString("dd/MM/yyyy HH:mm"));
185         html = html.Replace("[ReceiptSeries]", receipt.ReceiptSeries);
186         html = html.Replace("[ReceiptNumber]", receipt.ReceiptNumber);
187         html = html.Replace("[PAYMENTMETHOD]", paymentMethod);
188         html = html.Replace("[TOTAL_PRICE_WITHOUT_VAT]", receipt.TotalPriceWithoutVAT.ToString("0.00"));
189         html = html.Replace("[TOTAL_VAT]", receipt.TotalVAT.ToString("0.00"));
190         html = html.Replace("[TOTAL_PRICE]", receipt.TotalPriceWithVAT.ToString("0.00"));
191         var html_products = System.IO.File.ReadAllText(Path.Combine(webRootPath, "Templates\\Products.txt"));
192         var temp_html_products = "";

```

Εικόνα 31 OrderController_5

Τέλος ο κώδικας διαβάζει τα 2 Templates αρχεία txt (που περιέχουν κώδικα html) για τις αποδείξεις και αντικαθιστά την παραπάνω πληροφορία στα placeholders.

```

var html_products = System.IO.File.ReadAllText(Path.Combine(webRootPath, "Templates\\Products.txt"));
var temp_html_products = "";
foreach (var product in products)
{
    temp_html_products += html_products;
    temp_html_products = temp_html_products.Replace("[PRODUCT_CODE]", product.Code);
    temp_html_products = temp_html_products.Replace("[PRODUCT_DESCRIPTION]", product.Description);
    temp_html_products = temp_html_products.Replace("[PRODUCT_QUANTITY]", product.Quantity.ToString("0.00"));
    temp_html_products = temp_html_products.Replace("[UNIT_PRICE_WITHOUT_VAT]", product.UnitPriceWithoutVAT.ToString("0.00"));
    temp_html_products = temp_html_products.Replace("[UNIT_PRICE_WITH_VAT]", product.UnitPriceWithVAT.ToString("0.00"));
    temp_html_products = temp_html_products.Replace("[TOTAL_UNIT_PRICE_WITHOUT_VAT]", product.TotalPriceWithoutVAT.ToString("0.00"));
    temp_html_products = temp_html_products.Replace("[TOTAL_UNIT_PRICE_WITH_VAT]", product.TotalPriceWithVAT.ToString("0.00"));
}
html = html.Replace("[PRODUCTS]", temp_html_products);
byte[] pdf = null;
PdfDocument doc = new PdfDocument();
string urlFooter = String.Empty;
string urlReceipts = String.Empty;
var filename = "Receipt_" + orderId.ToString();
urlReceipts = Path.Combine(webRootPath, "Files\\Receipts\\") + filename + ".pdf";
PdfHtmlSection footerHtml = null;
HtmlToPdf converter = new HtmlToPdf();

// footer settings
converter.Options.DisplayFooter = true;
converter.Footer.DisplayOnFirstPage = true;
converter.Footer.DisplayOnOddPages = true;
converter.Footer.DisplayOnEvenPages = true;
converter.Footer.Height = 100;

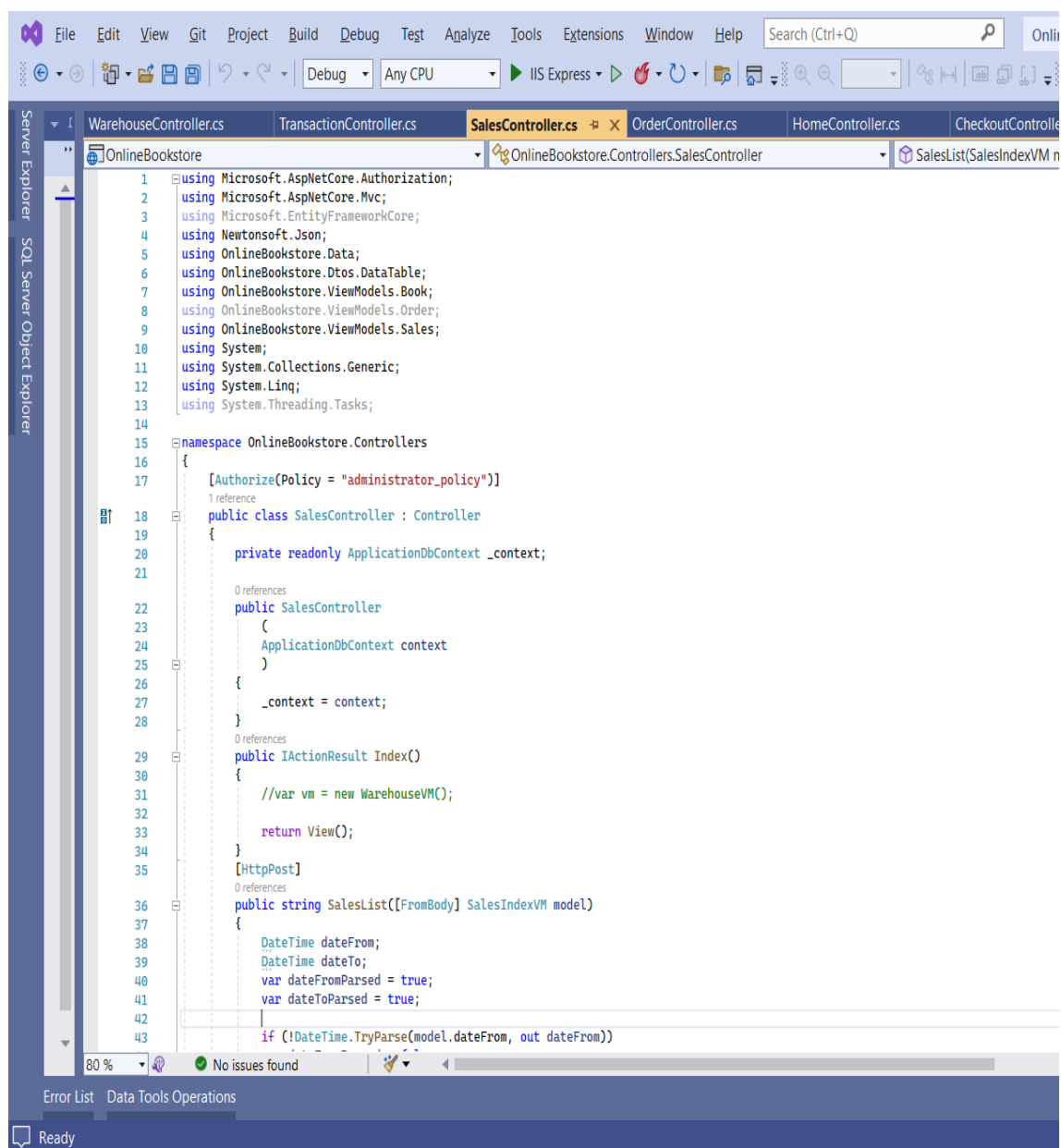
doc = converter.ConvertHtmlString(html, "");
doc.Save(urlReceipts);

using (FileStream fs = new FileStream(urlReceipts, FileMode.Open, FileAccess.Read))
{
    BinaryReader br = new BinaryReader(fs);
    long numBytes = new FileInfo(urlReceipts).Length;
    pdf = br.ReadBytes((int)numBytes);
    doc.Close();
    return File(pdf, "application/pdf");
}

```

Εικόνα 32 OrderController_6

Τα html αρχεία που πλέον έχουν την πληροφορία της παραγγελίας, με τη χρήση pdfDocument και FileStream μετατρέπονται σε PDF και γίνονται download στον browser.



```
1 using Microsoft.AspNetCore.Authorization;
2 using Microsoft.AspNetCore.Mvc;
3 using Microsoft.EntityFrameworkCore;
4 using Newtonsoft.Json;
5 using OnlineBookstore.Data;
6 using OnlineBookstore.Dtos.DataTable;
7 using OnlineBookstore.ViewModels.Book;
8 using OnlineBookstore.ViewModels.Order;
9 using OnlineBookstore.ViewModels.Sales;
10 using System;
11 using System.Collections.Generic;
12 using System.Linq;
13 using System.Threading.Tasks;
14
15 namespace OnlineBookstore.Controllers
16 {
17     [Authorize(Policy = "administrator_policy")]
18     public class SalesController : Controller
19     {
20         private readonly ApplicationDbContext _context;
21
22         public SalesController
23         (
24             ApplicationDbContext context
25         )
26         {
27             _context = context;
28         }
29         public IActionResult Index()
30         {
31             //var vm = new WarehouseVM();
32
33             return View();
34         }
35         [HttpPost]
36         public string SalesList([FromBody] SalesIndexVM model)
37         {
38             DateTime dateFrom;
39             DateTime dateTo;
40             var dateFromParsed = true;
41             var dateToParsed = true;
42
43             if (!DateTime.TryParse(model.dateFrom, out dateFrom))
```

Εικόνα 33 SalesController_1

Η SalesList μας επιστρέφει τη λίστα με τις πωλήσεις βιβλίων μεταξύ 2 ημερομηνιών που θα εισάγει ο χρήστης. Για κάθε βιβλίο που έχει πωληθεί μεταξύ των 2 ημερομηνιών, πολλαπλασιάζουμε την ποσότητα του βιβλίου με την τιμή του για να πάρουμε το κόστος της παραγγελίας και αθροίζοντας τα κόστη των παραγγελιών παίρνουμε τις συνολικές πωλήσεις του βιβλίου.

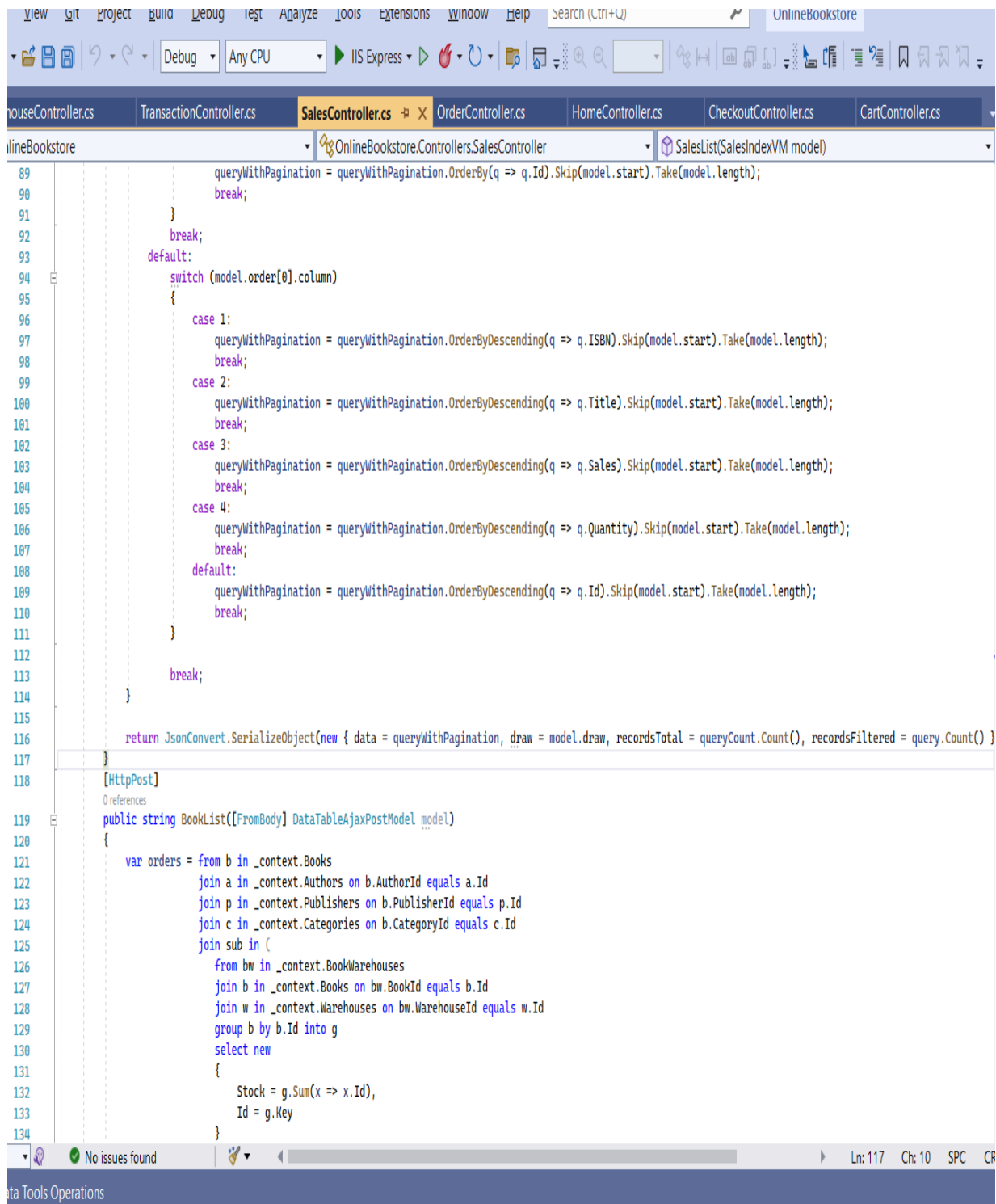
```

43     if (!DateTime.TryParse(model.dateFrom, out dateFrom))
44         dateFromParsed = false;
45
46     if (!DateTime.TryParse(model.dateTo, out dateTo))
47         dateToParsed = false;
48
49     var query =
50     (from b in _context.Books
51     join sub in (from bo in _context.BookOrders
52                 join b in _context.Books on bo.BookId equals b.Id
53                 join o in _context.Orders on bo.OrderId equals o.Id
54                 where (dateFromParsed && dateToParsed && o.CreatedOn >= dateFrom && o.CreatedOn <= dateTo) || (!dateFromParsed || !dateToParsed)
55                 group bo by (new { b.Id, b.ISBN }) into g
56                 select new { g.Key.Id, g.Key.ISBN, BookSales = g.Sum(bo => bo.Quantity * bo.UnitPrice), Quantity = g.Sum(bo => bo.Quantity) })
57     on b.Id equals sub.Id
58     orderby model.order[0].column
59     select new { Id = b.Id, ISBN = b.ISBN, Title = b.Title, Sales = sub.BookSales, Quantity = sub.Quantity });
60
61     var queryCount =
62     (from b in _context.Books
63     join sub in (from bo in _context.BookOrders
64                 join b in _context.Books on bo.BookId equals b.Id
65                 join o in _context.Orders on bo.OrderId equals o.Id
66                 group bo by (new { b.Id, b.ISBN }) into g
67                 select new { g.Key.Id})
68     on b.Id equals sub.Id
69     select new { });
70     var queryWithPaging = query;
71     switch (model.order[0].dir)
72     {
73     case "asc":
74         switch (model.order[0].column)
75         {
76         case 1:
77             queryWithPaging = queryWithPaging.OrderBy(q => q.ISBN).Skip(model.start).Take(model.length);
78             break;
79         case 2:
80             queryWithPaging = queryWithPaging.OrderBy(q => q.Title).Skip(model.start).Take(model.length);
81             break;
82         case 3:
83             queryWithPaging = queryWithPaging.OrderBy(q => q.Sales).Skip(model.start).Take(model.length);
84             break;
85         case 4:
86             queryWithPaging = queryWithPaging.OrderBy(q => q.Quantity).Skip(model.start).Take(model.length);
87             break;
88         default:
89             queryWithPaging = queryWithPaging.OrderBy(q => q.Id).Skip(model.start).Take(model.length);

```

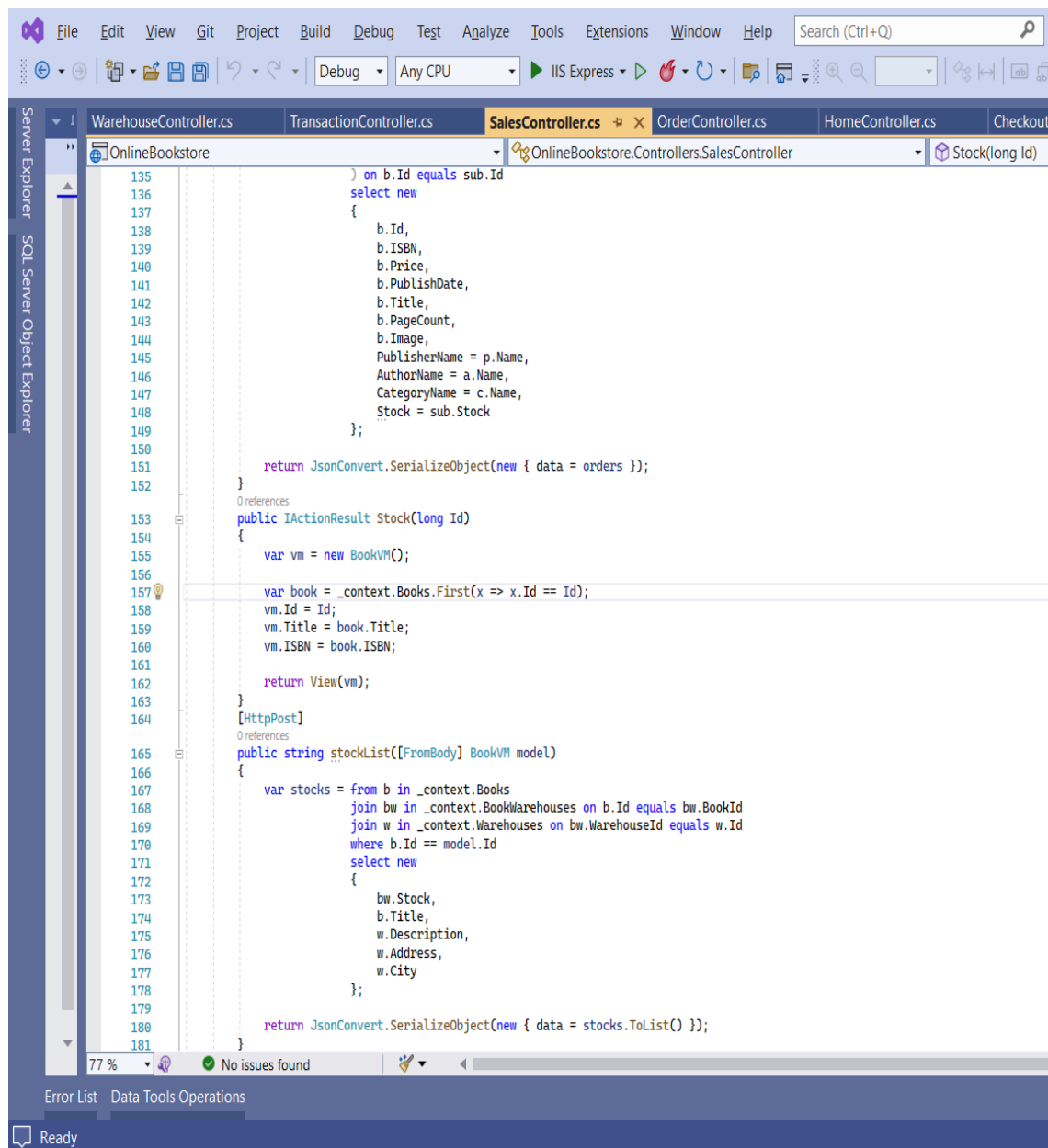
Εικόνα 34 SalesController_2

Επίσης γίνεται και ταξινόμηση με χρήση (Order by) των πωλήσεων (όταν ο χρήστης κλικάρει κάποια στήλη/κολώνα του datatable).



```
89         queryWithPagination = queryWithPagination.OrderBy(q => q.Id).Skip(model.start).Take(model.length);
90         break;
91     }
92     break;
93     default:
94         switch (model.order[0].column)
95         {
96             case 1:
97                 queryWithPagination = queryWithPagination.OrderByDescending(q => q.ISBN).Skip(model.start).Take(model.length);
98                 break;
99             case 2:
100                queryWithPagination = queryWithPagination.OrderByDescending(q => q.Title).Skip(model.start).Take(model.length);
101                break;
102             case 3:
103                queryWithPagination = queryWithPagination.OrderByDescending(q => q.Sales).Skip(model.start).Take(model.length);
104                break;
105             case 4:
106                queryWithPagination = queryWithPagination.OrderByDescending(q => q.Quantity).Skip(model.start).Take(model.length);
107                break;
108             default:
109                queryWithPagination = queryWithPagination.OrderByDescending(q => q.Id).Skip(model.start).Take(model.length);
110                break;
111         }
112     }
113     break;
114 }
115
116     return JsonConvert.SerializeObject(new { data = queryWithPagination, draw = model.draw, recordsTotal = queryCount.Count(), recordsFiltered = query.Count() });
117 }
118
119 [HttpPost]
120 0 references
121 public string BookList([FromBody] DataTableAjaxPostModel model)
122 {
123     var orders = from b in _context.Books
124                 join a in _context.Authors on b.AuthorId equals a.Id
125                 join p in _context.Publishers on b.PublisherId equals p.Id
126                 join c in _context.Categories on b.CategoryId equals c.Id
127                 join sub in (
128                     from bw in _context.BookWarehouses
129                     join b in _context.Books on bw.BookId equals b.Id
130                     join w in _context.Warehouses on bw.WarehouseId equals w.Id
131                     group b by b.Id into g
132                     select new
133                     {
134                         Stock = g.Sum(x => x.Id),
135                         Id = g.Key
136                     }
137                 ) on b.Id equals sub.Id
138     into g
139     select new
140     {
141         Title = b.Title,
142         ISBN = b.ISBN,
143         Author = a.Name,
144         Publisher = p.Name,
145         Category = c.Name,
146         Stock = g.Stock,
147         Id = g.Key
148     }
149     .OrderByDescending(o => o.Stock)
150     .Skip(model.start)
151     .Take(model.length);
152 }
```

Εικόνα 35 SalesController_3

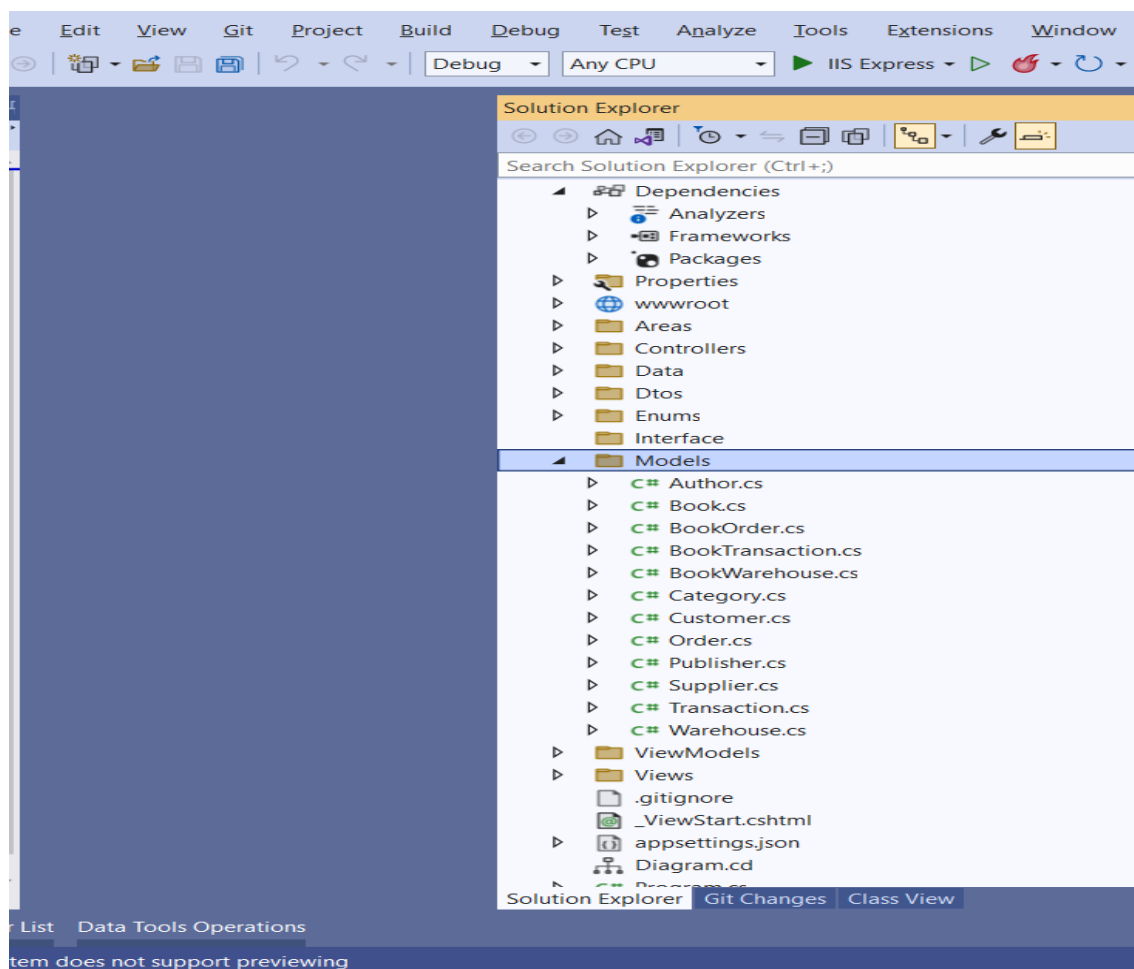


Εικόνα 36 SalesController_4

Η stockList με το αντίστοιχο query επιστρέφει τη λίστα με τις αποθήκες και το απόθεμα για το συγκεκριμένο βιβλίο.

4.2 Models

Στην αρχιτεκτονική MVC, το M σημαίνει Model και το Model είναι μια κανονική κλάση C#. Το μοντέλο είναι υπεύθυνο για το χειρισμό των δεδομένων και την επιχειρηματική λογική. Το μοντέλο αντιπροσωπεύει το σχήμα των δεδομένων και είναι υπεύθυνο για το χειρισμό αλλαγών που σχετίζονται με τη βάση δεδομένων.



Εικόνα 37

Ο φάκελος Models σε μια εφαρμογή τύπου MVC, περιέχει όλες τις κλάσεις που χρησιμοποιεί η εφαρμογή για να περάσει οποιαδήποτε πληροφορία, από ένα action σε ένα View ή ακόμα και σε άλλο action. Αυτά τα αντικείμενα μπορεί να είναι πληροφορία από τη βάση δεδομένων της εφαρμογής, ή ακόμα και απλά αντικείμενα που περιέχουν άλλα αντικείμενα από απλές κλάσεις.

Για τα Models σε ένα ASP.NET Web Application, χρησιμοποιείται πλέον το Entity Framework, το οποίο δημιουργεί τη βάση δεδομένων της εφαρμογής.

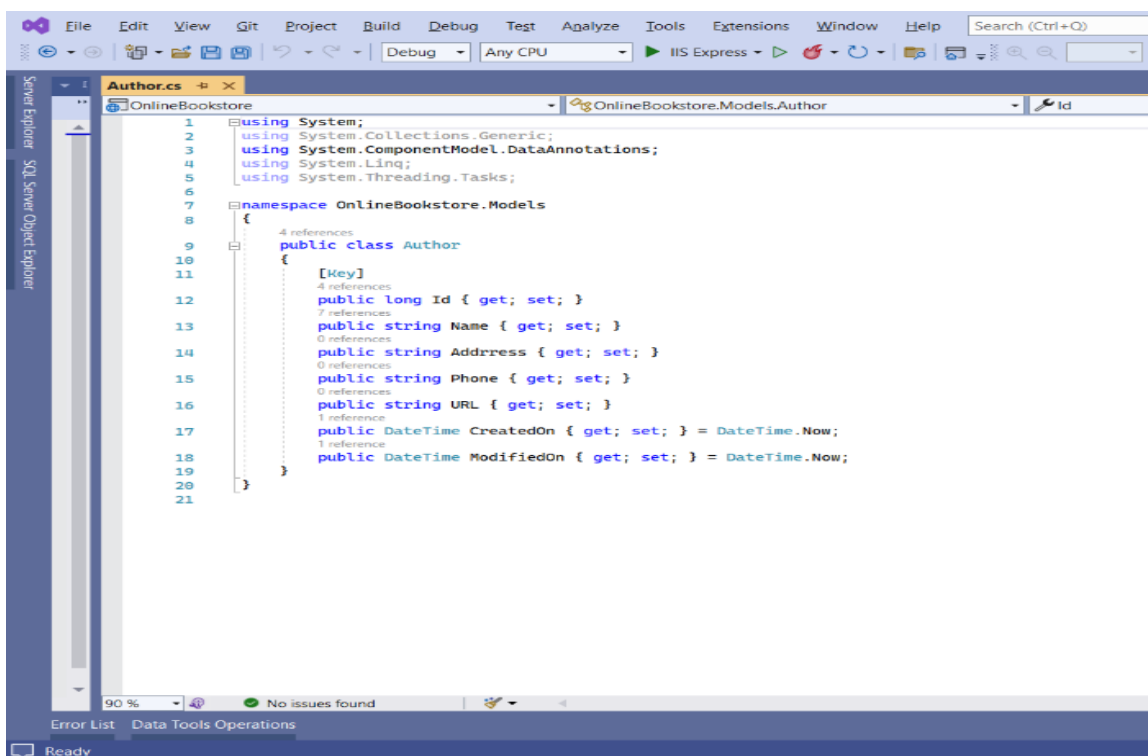
Για την υλοποίηση την εφαρμογής, όπως προαναφέρθηκε στο προηγούμενο κεφάλαιο, χρησιμοποιήθηκε η προσέγγιση Code First.

Με αυτήν, ο προγραμματιστής χρειάζεται να ορίσει μόνο τις κλάσεις με C# όπως αυτός θέλει να είναι στη βάση δεδομένων, και μετά να ορίσει με λεπτομέρεια ότι αυτές πρέπει να μεταφραστούν σε tables. Η μετάφραση αυτή πραγματοποιείται αυτόματα από το Entity Framework. Για την υλοποίηση της πλατφόρμας χρησιμοποιήθηκε το Code First workflow, καθώς το Entity Framework μπορεί αρκετά εύκολα να δημιουργήσει τα tables και τις μεταξύ τους συνδέσεις. Παρακάτω παρατίθενται κάποιες από τις κλάσεις της εφαρμογής.

Μέσω της Code First προσέγγισης γίνεται αρκετά εύκολη η δουλειά των προγραμματιστών, που δεν θέλουν να κάνουν από την αρχή όλα τα tables της βάσης δεδομένων μαζί με όλα τα constraints τους. Πλέον μπορούν απλά να γράφουν μια κλάση και μέσω του Entity Framework να μεταφραστούν σε tables της βάσης.

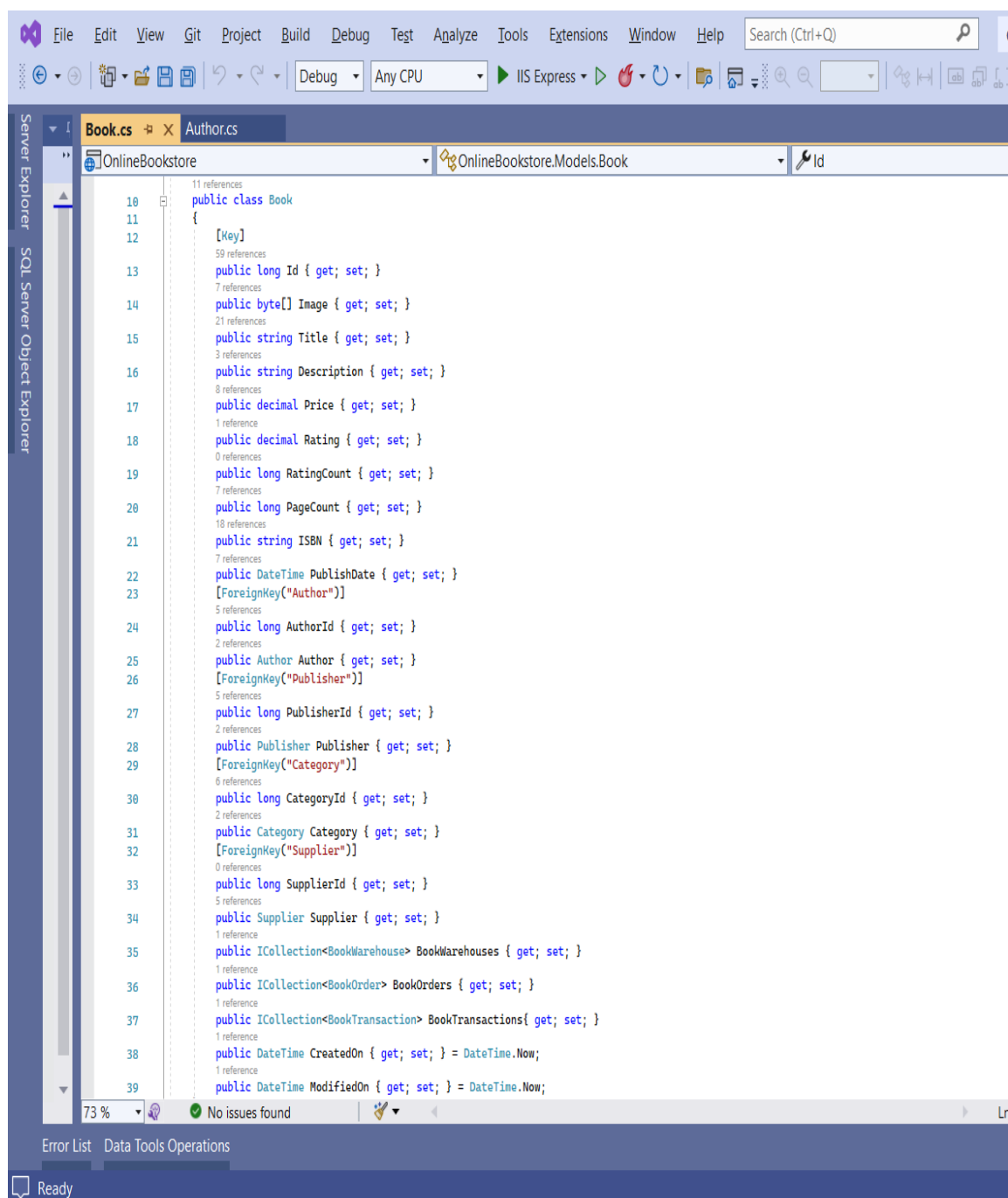
Το Entity Framework όμως, απαιτεί άλλο ένα βήμα για να δημιουργήσει το table και αυτό είναι το λεγόμενο **migration**. Συγκεκριμένα χρησιμοποιήθηκαν τα Code First Migrations, εφόσον χρησιμοποιήσαμε την Code First προσέγγιση. Τα Migrations χρησιμοποιούνται όταν θέλει ο προγραμματιστής να αλλάξει την βάση δεδομένων, είτε αλλάζοντας κάποια από τα στοιχεία των πινάκων, είτε προσθέτοντας ένα καινούργιο, ή ακόμα και να κάνει Populate κάποιον πίνακα με κάποιες τιμές.

Παρακάτω παρατίθενται ενδεικτικά οι 3 κλάσεις από τις συνολικά 12 που χρησιμοποιήθηκαν στην εφαρμογή. Πρόκειται για την κλάση Author που σχετίζεται με τα στοιχεία του συγγραφέα του βιβλίου, την κλάση Book που εμπεριέχει τα χαρακτηριστικά του βιβλίου και την κλάση BookOrder περιλαμβάνει πληροφορίες για τις παραγγελίες.



```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel.DataAnnotations;
4 using System.Linq;
5 using System.Threading.Tasks;
6
7 namespace OnlineBookstore.Models
8 {
9     4 references
10     public class Author
11     {
12         [Key]
13         4 references
14         public long Id { get; set; }
15         7 references
16         public string Name { get; set; }
17         0 references
18         public string Address { get; set; }
19         0 references
20         public string Phone { get; set; }
21         0 references
22         public string URL { get; set; }
23         1 reference
24         public DateTime CreatedOn { get; set; } = DateTime.Now;
25         1 reference
26         public DateTime ModifiedOn { get; set; } = DateTime.Now;
27     }
28 }
```

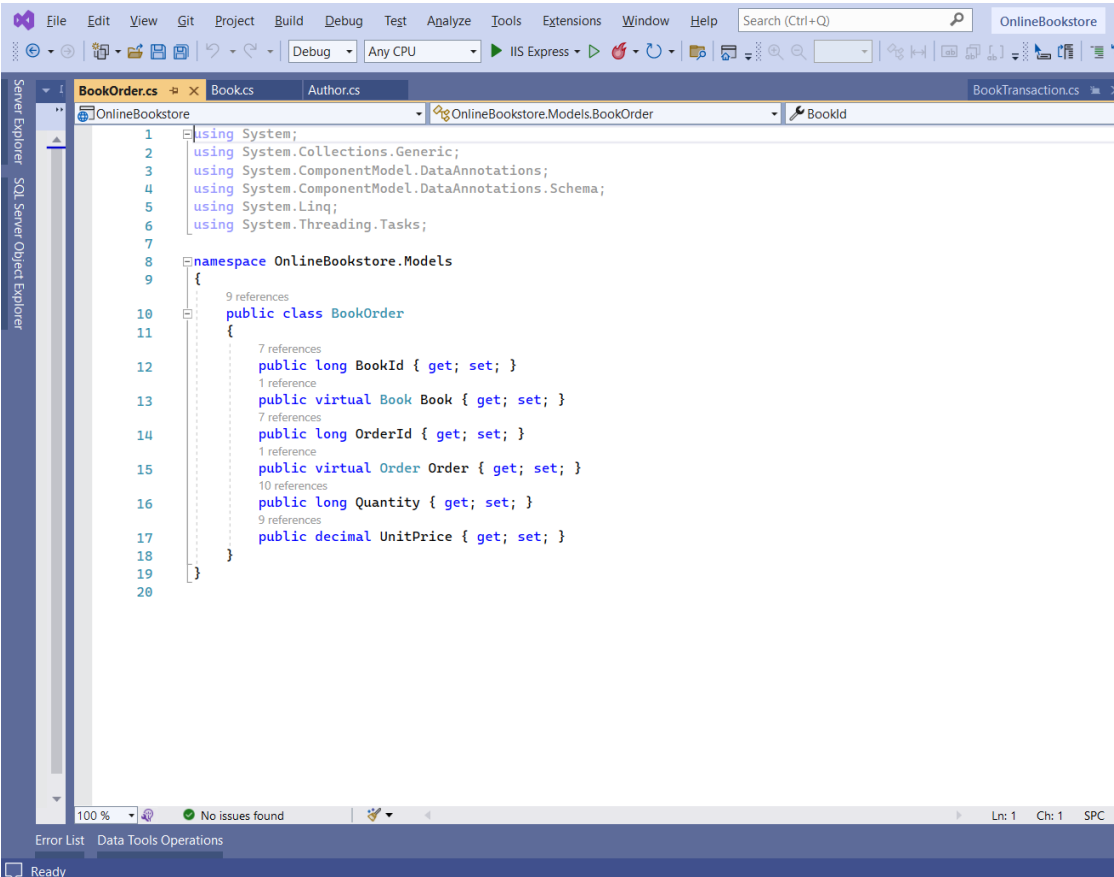
Εικόνα 38 Author



Εικόνα 39 Book

Οι σημάνσεις που εμφανίζονται εντός παρενθέσεων [] **πάνω** από τα attributes των κλάσεων ονομάζονται **data annotations**. Αυτές εφαρμόζονται στα χαρακτηριστικά μιας κλάσης οντότητας για να καθορίσουν μια ιδιότητα κλειδιού σε μία οντότητα και να κάνουν την αντίστοιχη στήλη Πρωτεύον

κλειδί (Primary Key) στη βάση δεδομένων. Η ένδειξη [key] στο Id σημαίνει ότι το συγκεκριμένο attribute είναι πρωτεύον κλειδί στο πίνακα Book . Η ένδειξη [Foreign Key] στο AuthorId σημαίνει ότι το συγκεκριμένο attribute είναι ξένο κλειδί για αυτόν πίνακα . Με αυτό τον τρόπο γίνεται κατανοητό πως συνδέονται οι πίνακες οντότητες μεταξύ τους. Επιπρόσθετα, μέσα στις κλάσεις , αποτυπώνεται με συγκεκριμένο τρόπο σύνταξης, η σχέση πολλαπλότητας μεταξύ των οντοτήτων . Στην παραπάνω κλάση Book , στη σειρά 25 αναγράφεται “ **public author author** ” . Με αυτό τον τρόπο εκφράζεται η σχέση **ένα προς ένα** μεταξύ της κλάσης Book και της κλάσης Author , που σημαίνει ότι ένα βιβλίο μπορεί να έχει γραφτεί μόνον από έναν συγγραφέα. Στη σειρά 36 ,στο κώδικα της ίδιας κλάσης αναγράφεται “**public ICollection<BookOrder>BookOrders**”. Με αυτόν τον τρόπο εκφράζεται η σχέση **ένα προς πολλά**, μεταξύ των κλάσεων-οντοτήτων Book και BookOrder, που σημαίνει ότι ένα βιβλίο μπορεί να έχει συμπεριληφθεί σε πολλές διαφορετικές παραγγελίες ή πολλές παραγγελίες περιλαμβάνουν το συγκεκριμένο βιβλίο.



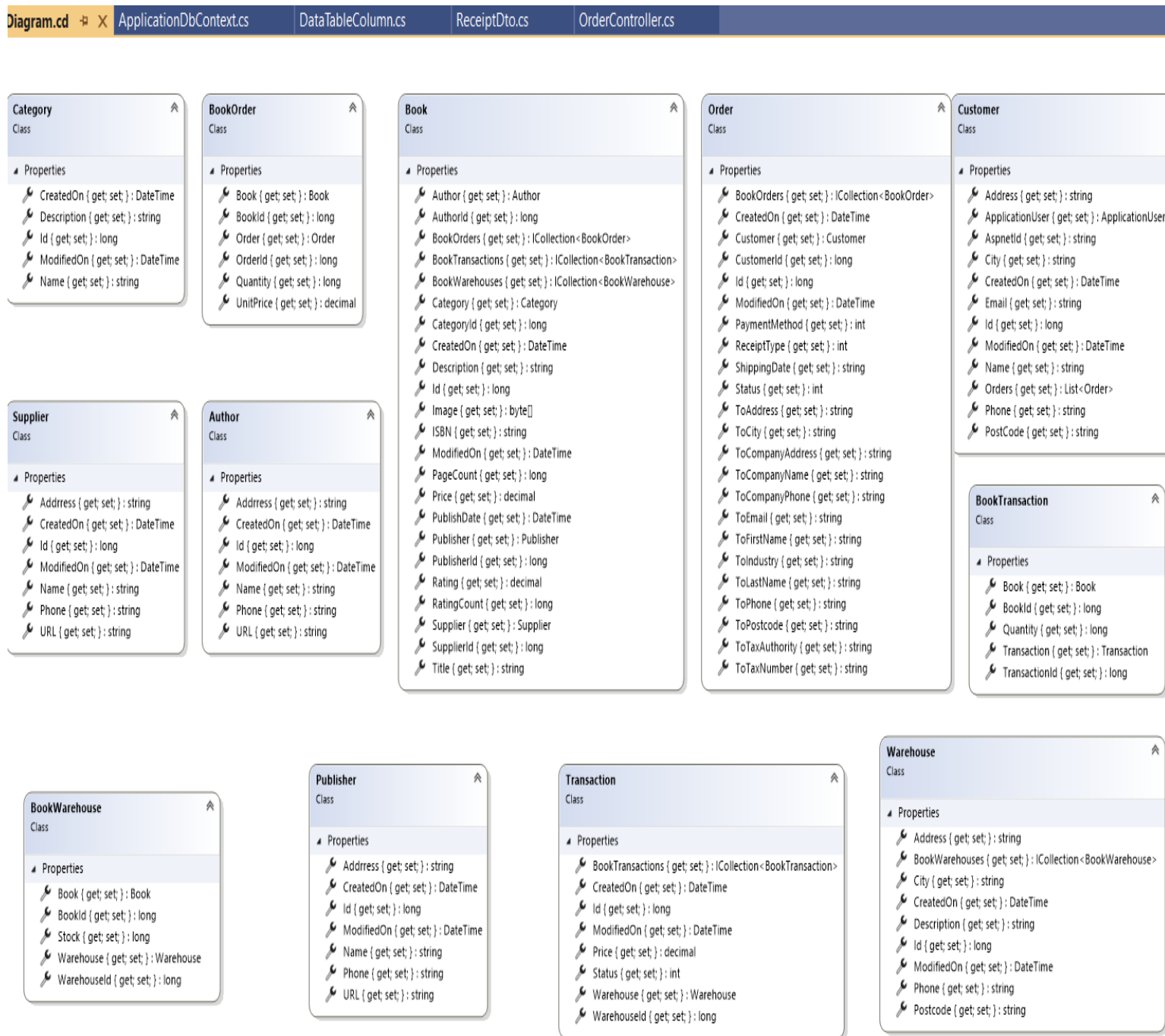
```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel.DataAnnotations;
4  using System.ComponentModel.DataAnnotations.Schema;
5  using System.Linq;
6  using System.Threading.Tasks;
7
8  namespace OnlineBookstore.Models
9  {
10     9 references
11     public class BookOrder
12     {
13         7 references
14         public long BookId { get; set; }
15         1 reference
16         public virtual Book Book { get; set; }
17         7 references
18         public long OrderId { get; set; }
19         1 reference
20         public virtual Order Order { get; set; }
21         10 references
22         public long Quantity { get; set; }
23         9 references
24         public decimal UnitPrice { get; set; }
25     }
26 }

```

Εικόνα 40 BookOrder

Στο παρακάτω πίνακα εμφανίζεται το σύνολο των 12 Model κλάσεων της εφαρμογής.



Εικόνα 41 Model Classes

4.3 Views

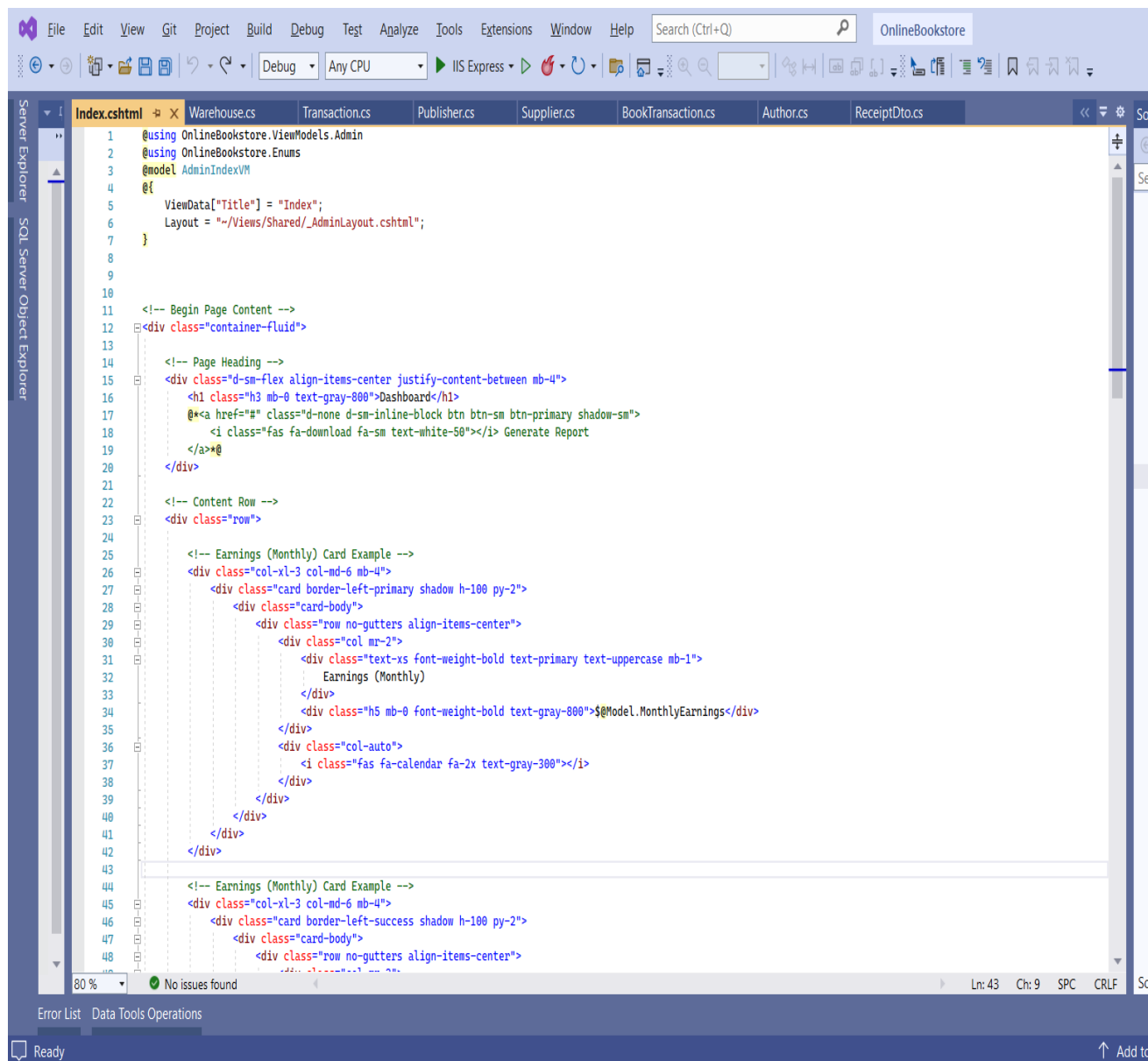
Ο φάκελος Views αποτελεί το στοιχείο Προβολή χρησιμοποιείται για όλη τη λειτουργία του UI της εφαρμογής.

Τα Views σε μια ASP.NET MVC εφαρμογή, είναι ουσιαστικά οι ιστοσελίδες που παρουσιάζονται στον χρήστη για να αλληλοεπιδράσει με αυτές. Ένα View περιέχει ένα συντακτικό που η Microsoft το ονομάζει Razor. Το συντακτικό αυτό είναι υπεύθυνο για την παρουσίαση της σελίδας στον χρήστη.

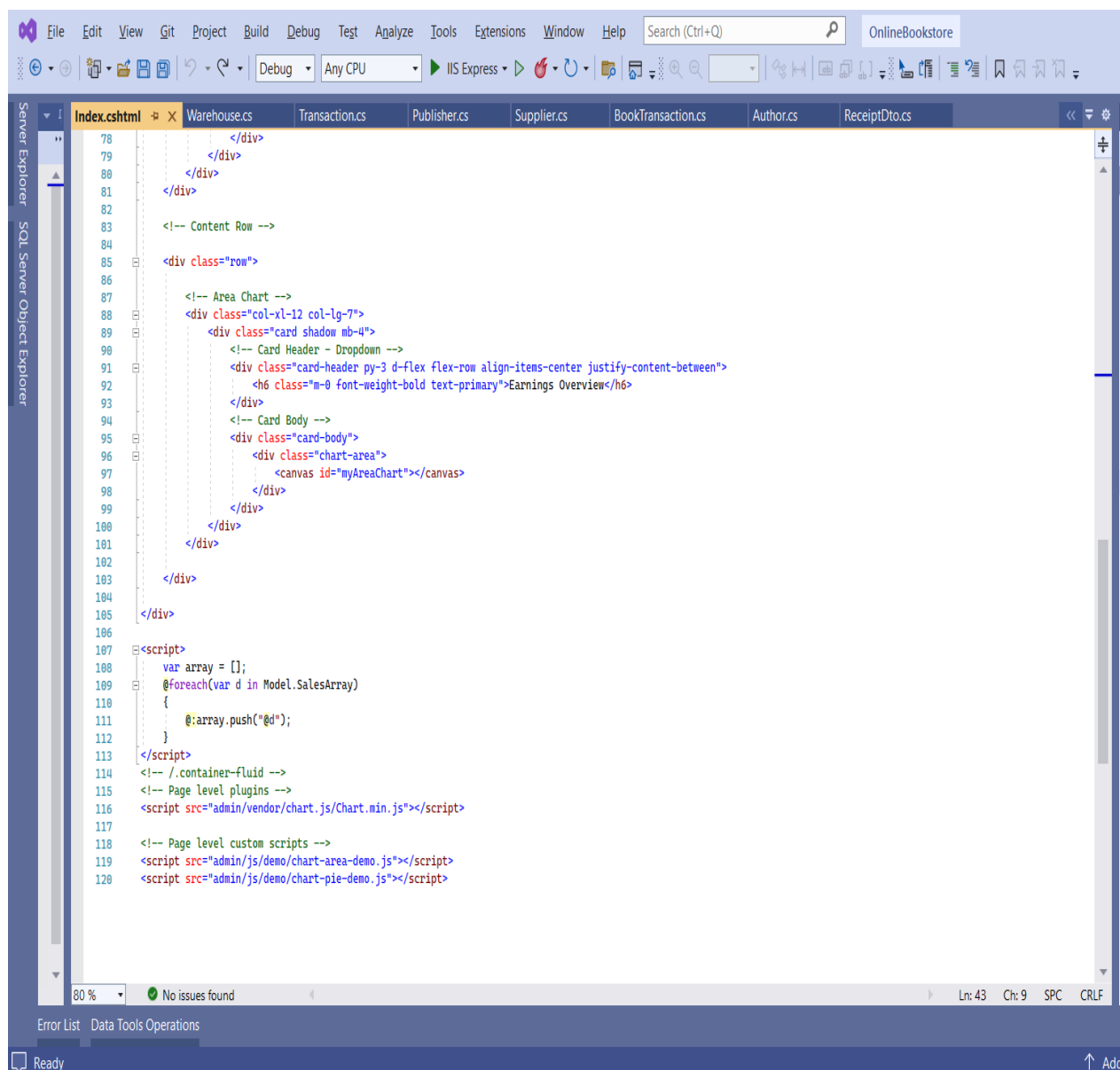
Το Razor περιέχει ένα συνδυασμό από HTML, jQuery και C#, έτσι ώστε να μπορεί εύκολα ο προγραμματιστής να διαμορφώσει όπως θέλει αυτός τη σελίδα, και ταυτόχρονα να πάρει την πληροφορία που θέλει από τον Controller πολύ απλά χρησιμοποιώντας C#. Τα αρχεία Views παράγονται ως αρχεία .cshtml.

Η default γλώσσα στο Razor syntax είναι η HTML οπότε μπορούμε εύκολα να χρησιμοποιήσουμε ότι ξέρουμε από αυτή τη γλώσσα. Αν όμως θέλει ο προγραμματιστής να χρησιμοποιήσει C# το μόνο που χρειάζεται να κάνει είναι να βάλει το σύμβολο @ μπροστά από μια έκφραση και να γράψει την λογική που θέλει. Το Razor syntax μάλιστα, υποστηρίζει πλήρως και if statements καθώς και loops, αλλά και απλό κώδικα σε C#. Ακόμα ο προγραμματιστής έχει τη δυνατότητα να συνδυάσει λογική C# με HTML tags για να έχει διαδραστικότητα η σελίδα του.

Παρατίθενται παρακάτω ενδεικτικά κάττοια από τα Views της εφαρμογής.



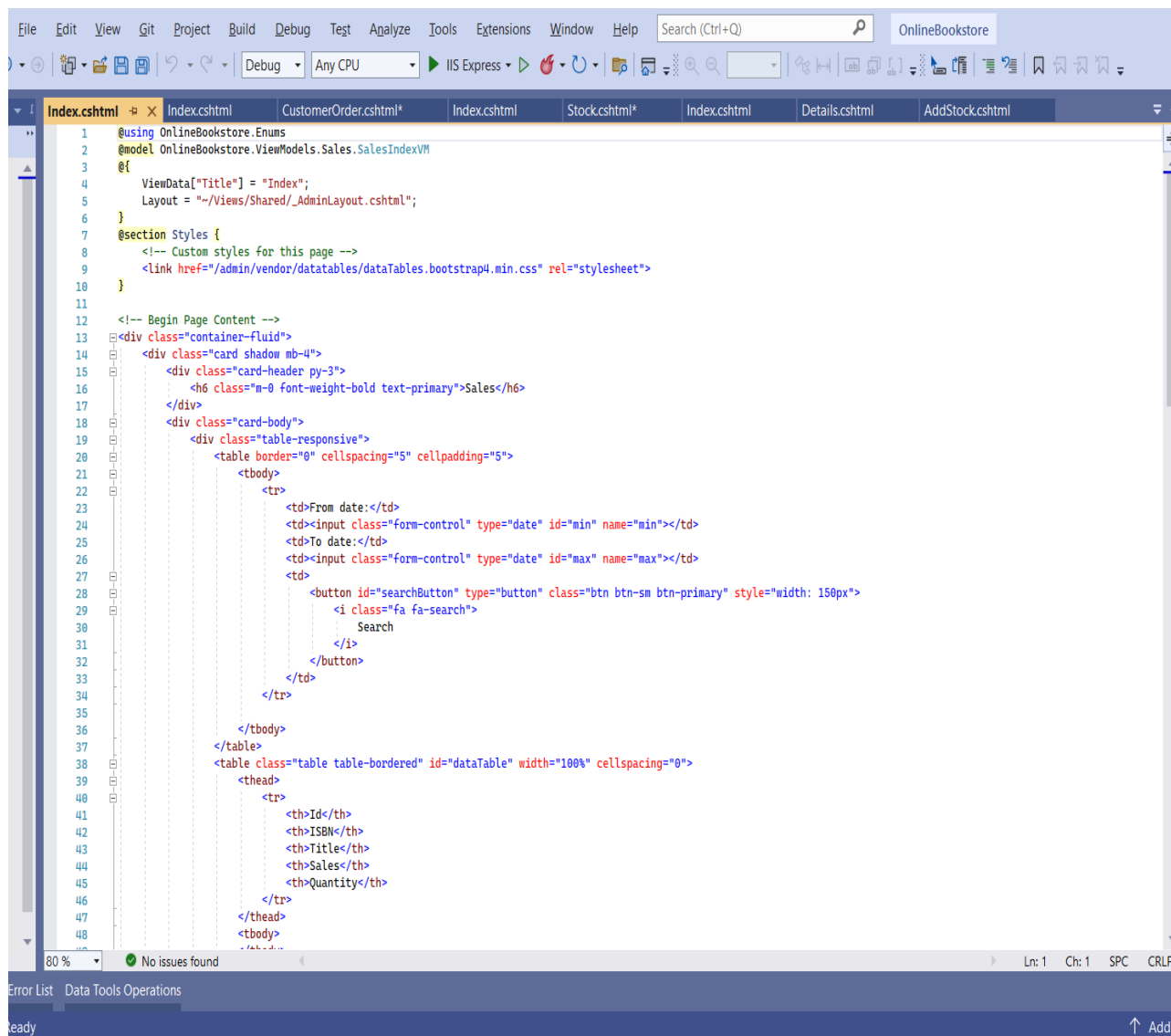
Εικόνα 42 Admin View 1



Εικόνα 44 Admin View 3

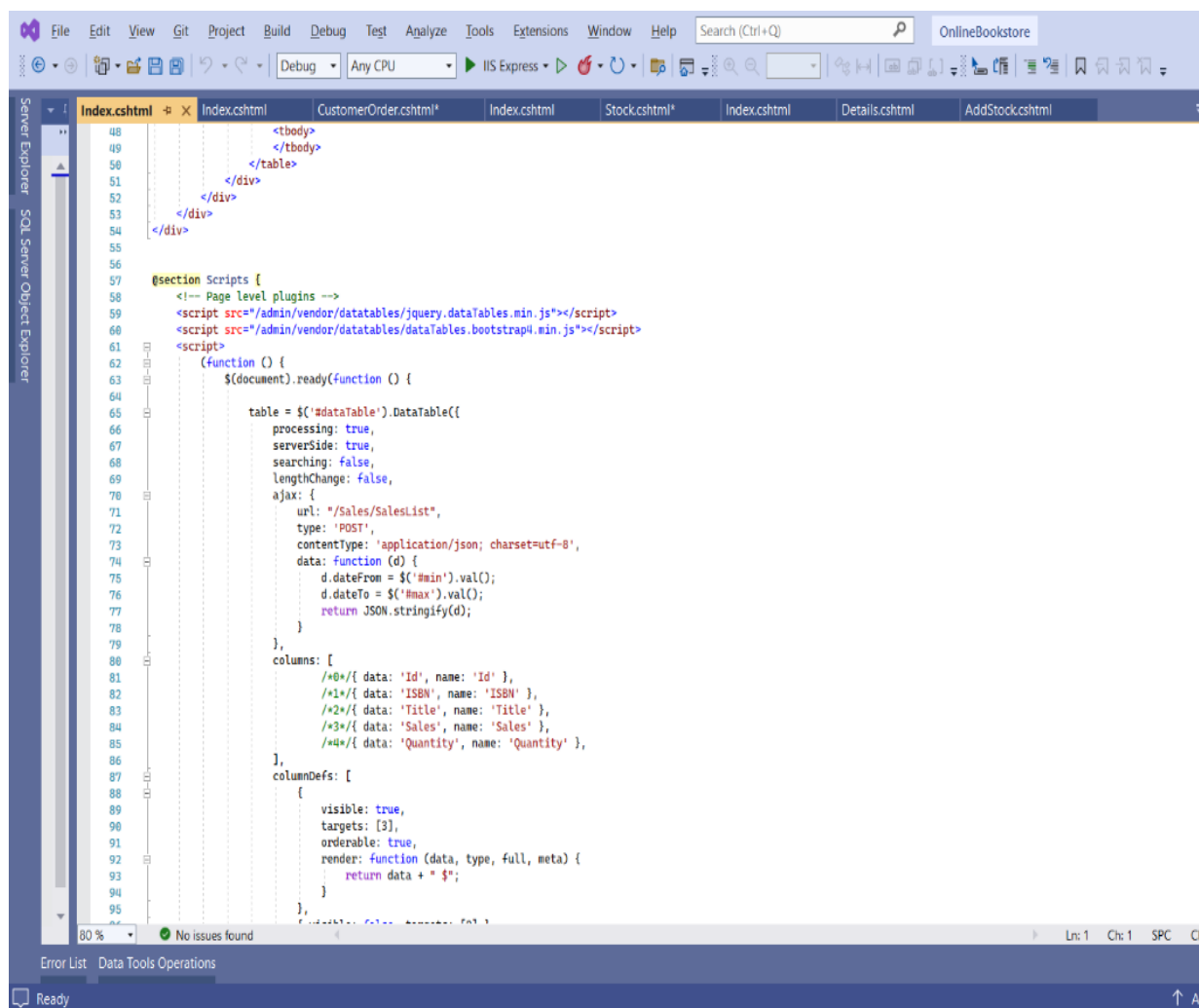
Στις παραπάνω εικόνες , εμφανίζεται ο κώδικας του πάνελ του διαχειριστή (Admin). Το panel του διαχειριστή δείχνει τα μηνιαία/ετήσια κέρδη , τις παραγγελίες σε αναμονή έγκρισης και το γράφημα των εσόδων ανά μήνα.

Στις παραπάνω εικόνες , εμφανίζεται ο κώδικας layout της επιλογής Sales του διαχειριστή (Admin). Πατώντας από την πλευρική μπάρα το κουμπί sales ,εμφανίζεται πίνακας με τις πωλήσεις ανά βιβλίο. Τα δεδομένα αυτά αντλούνται από το directory /Sales/SalesList. Περιέχει ακόμα τη φόρμα για αναζήτηση πωλήσεων μεταξύ των ημερομηνιών.

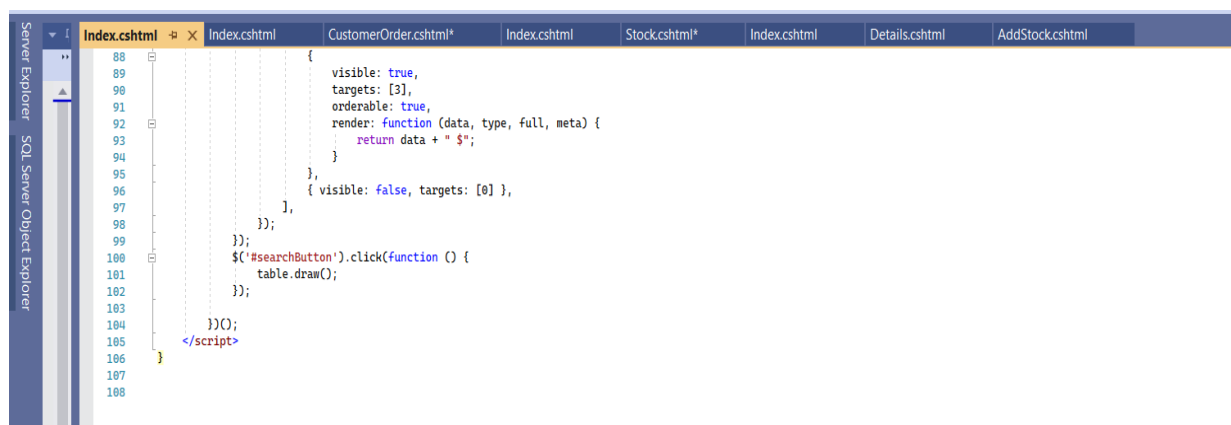


```
1 @using OnlineBookstore.Enums
2 @model OnlineBookstore.ViewModels.Sales.SalesIndexVM
3 @{}
4 ViewData["Title"] = "Index";
5 Layout = "~/Views/Shared/_AdminLayout.cshtml";
6 }
7 @section Styles {
8 <!-- Custom styles for this page -->
9 <link href="/admin/vendor/datatables/dataTables.bootstrap4.min.css" rel="stylesheet">
10 }
11
12 <!-- Begin Page Content -->
13 <div class="container-fluid">
14 <div class="card shadow mb-4">
15 <div class="card-header py-3">
16 <h6 class="m-0 font-weight-bold text-primary">Sales</h6>
17 </div>
18 <div class="card-body">
19 <div class="table-responsive">
20 <table border="0" cellspacing="5" cellpadding="5">
21 <tbody>
22 <tr>
23 <td>From date:</td>
24 <td><input class="form-control" type="date" id="min" name="min"></td>
25 <td>To date:</td>
26 <td><input class="form-control" type="date" id="max" name="max"></td>
27 <td>
28 <button id="searchButton" type="button" class="btn btn-sm btn-primary" style="width: 150px">
29 <i class="fa fa-search">
30 Search
31 </i>
32 </button>
33 </td>
34 </tr>
35 </tbody>
36 </table>
37 <table class="table table-bordered" id="dataTable" width="100%" cellspacing="0">
38 <thead>
39 <tr>
40 <th>Id</th>
41 <th>ISBN</th>
42 <th>Title</th>
43 <th>Sales</th>
44 <th>Quantity</th>
45 </tr>
46 </thead>
47 <tbody>
48 <tr>
```

Εικόνα 45 AdminSales Index View 1



Εικόνα 46 AdminSales Index View 2

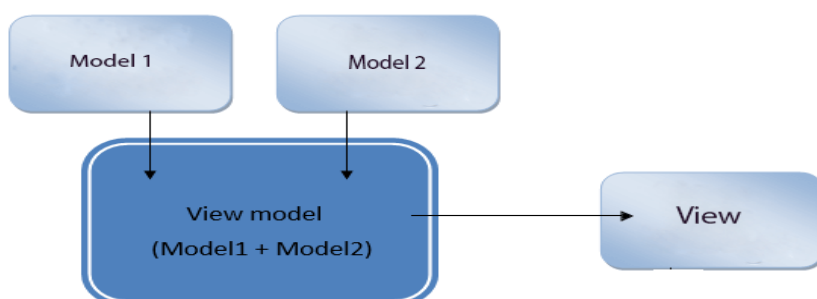


Εικόνα 47 AdminSales Index View 3

4.4 ViewModels

Το ViewModel μας παρέχει τον κατάλληλο διαχωρισμό προβλημάτων (SoC – Separation of Concerns). Η προβολή χρειάζεται να αποδίδει μόνο το μεμονωμένο αντικείμενο ViewModel και υπάρχει ένας συγκεκριμένος σκοπός για κάθε πτυχή της εφαρμογής, πράγμα που σημαίνει ότι η εφαρμογή θα είναι πιο οργανωμένη στον κώδικα. Χρησιμοποιώντας το ViewModel, χειριζόμαστε τον κώδικα ξεχωριστά σε ένα συγκεκριμένο σημείο και μακριά από την προβολή και τον ελεγκτή. Έτσι, αυτό μας παρέχει ένα καλύτερο SoC και κάνει τον κώδικά μας πιο maintainable. Κάνει επίσης τον κώδικά μας πιο ελέγξιμο, επειδή η δοκιμή μονάδων σημαίνει δοκιμή των μικρών μονάδων και είναι ευκολότερο να δοκιμαστεί εάν διαχωρίσουμε τον χειρισμό της λογικής σε ένα ξεχωριστό ViewModel.

Συνήθως χρησιμοποιούμε το ViewModel όταν ένα Model object δεν περιλαμβάνει όλα τα δεδομένα που ένα View χρειάζεται. Με άλλα λόγια, μερικές φορές απαιτείται η εμφάνιση δεδομένων από πολλές οντότητες σε προβολή που προέρχεται από διαφορετικές κατηγορίες μοντέλων δεδομένων.



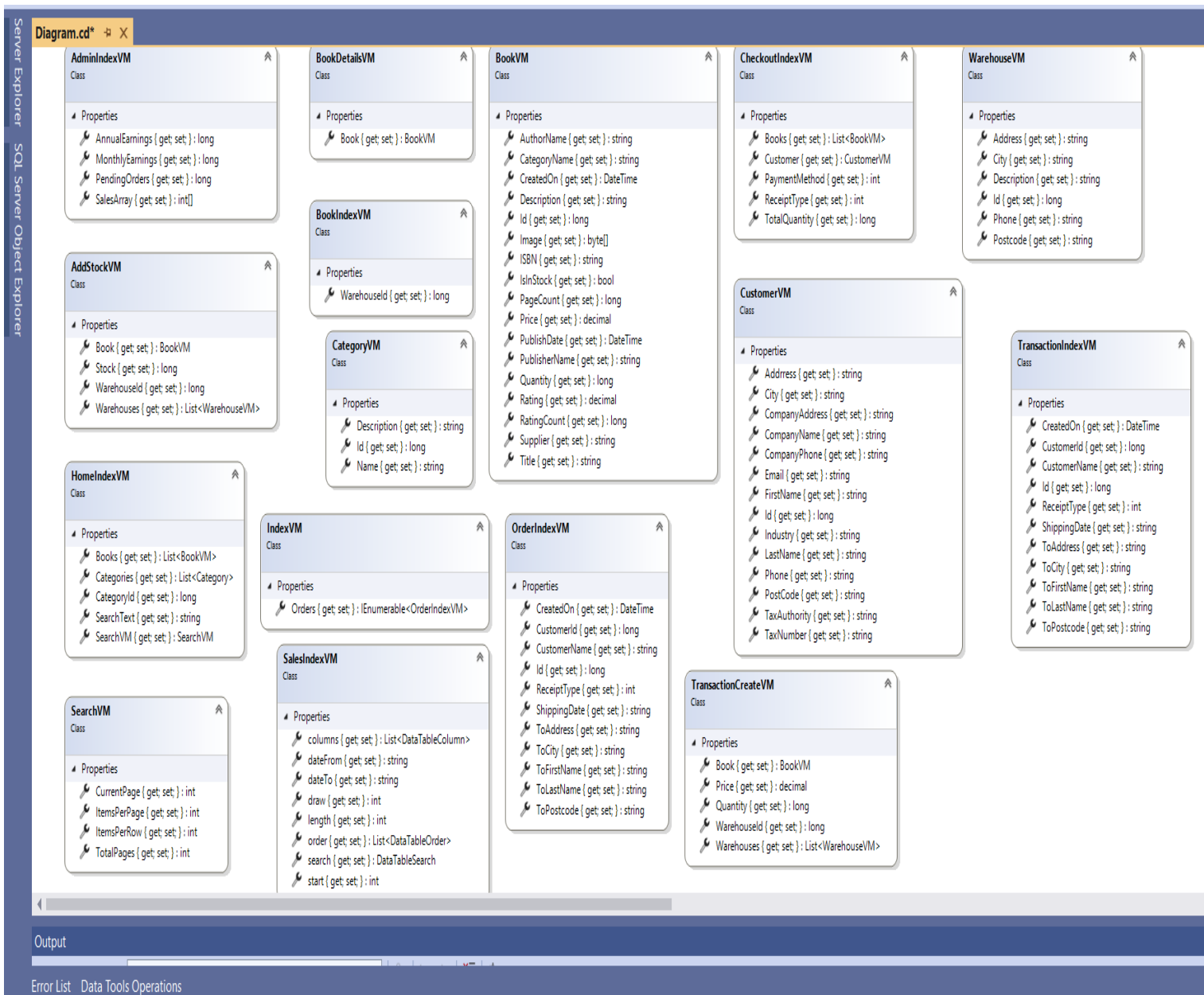
Εικόνα 48

Το ViewModel μπορεί να χρησιμοποιηθεί για :

- Διαχείριση ή δημιουργία αναπτυσσόμενων λιστών για μια συγκεκριμένη οντότητα
- Δημιουργία προβολής Master-Details σε ιστότοπους που βασίζονται σε δεδομένα
- Χρησιμοποιείται για την εμφάνιση καλαθιών αγορών σε ιστότοπους
- Χρησιμοποιείται στη σελιδοποίηση
- Χρησιμοποιείται σε έναν ιστότοπο για την εμφάνιση γραφικού στοιχείου προφίλ χρήστη
- Χρησιμοποιείται για την κατασκευή Πινάκων ελέγχου(Dashboard) οι οποίοι ενσωματώνουν πολλά δεδομένα σε ένα μέρος

Για την υλοποίηση της εργασίας, χρησιμοποιήθηκε το ViewModel για μερικούς από τους παραπάνω λόγους.

Στον παρακάτω πίνακα εμφανίζονται το σύνολο των ViewModels (VM) κλάσεων που χρησιμοποιήθηκαν στην εφαρμογή.



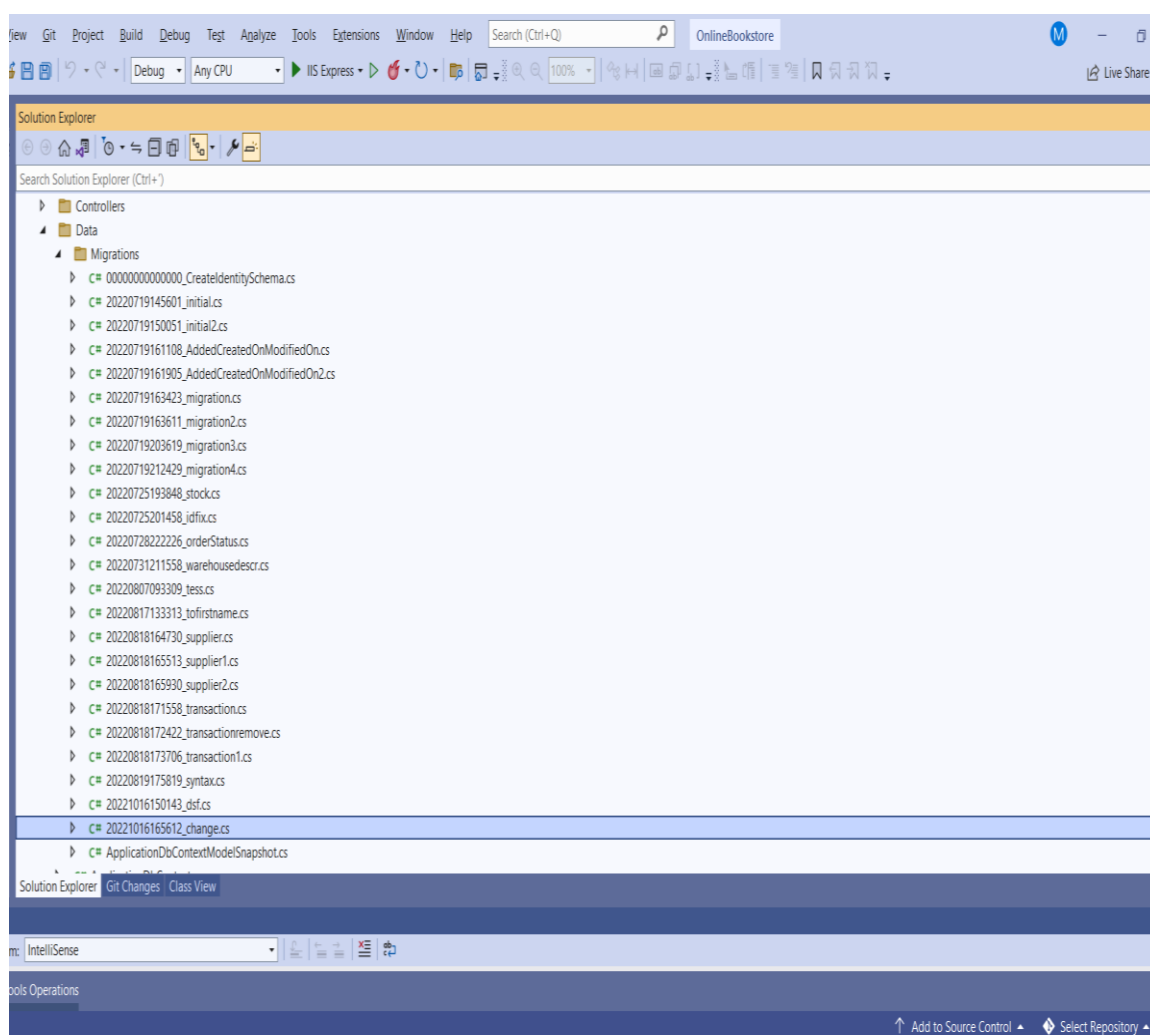
Εικόνα 49

4.5 Data και appsettings

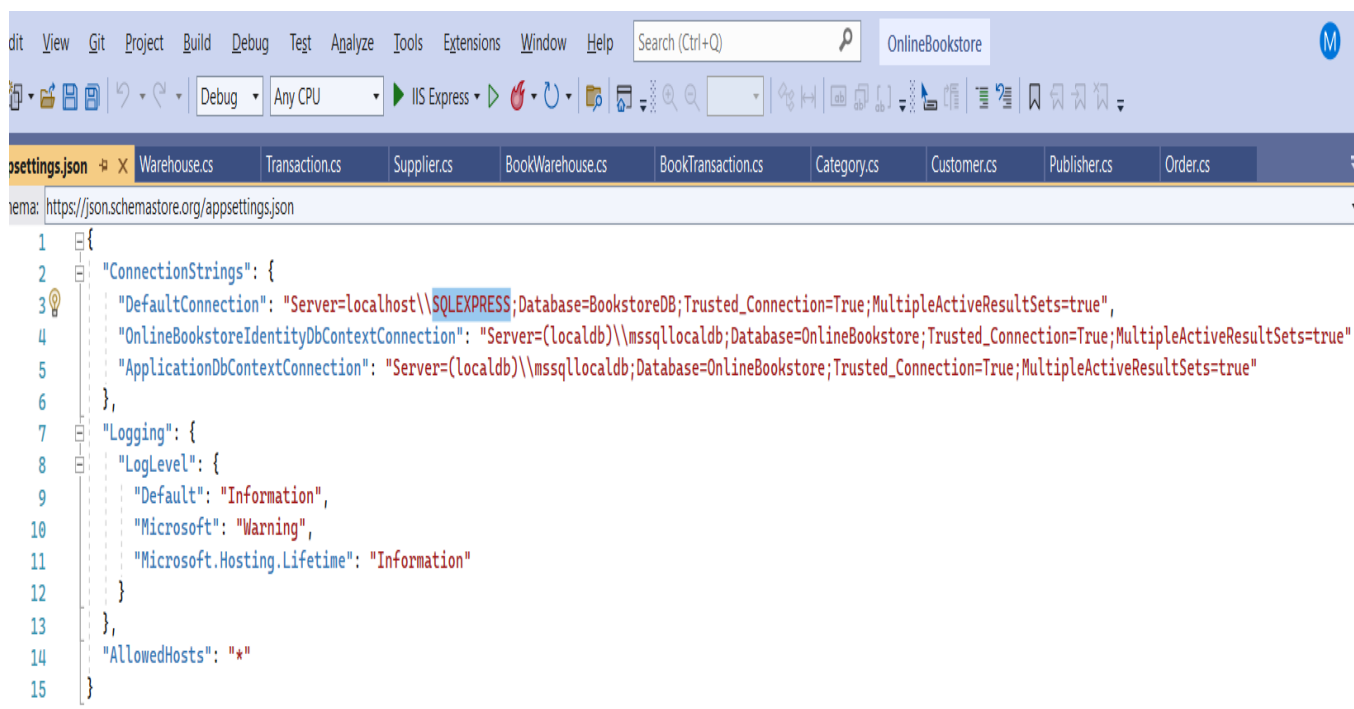
Ο φάκελος migration που βρίσκεται μέσα στον φάκελο Data, περιλαμβάνει όλες τις μεταβολές που έχουν γίνει στο σχήμα της βάσης.

Όπως αναφέραμε και στο προηγούμενο κεφάλαιο, για τη δημιουργία των Entities του Entity Framework, χρησιμοποιήθηκε η προσέγγιση Code First.

Επιλέγοντας την Code First, για να κάνουμε οποιαδήποτε μεταβολή στο σχήμα της βάσης, απαιτείται να χρησιμοποιήσουμε τον Package Manager Console. Εκκινώντας τον Package Manager Console, εκτελούμε αρχικά την εντολή enable-migration, στην συνέχεια εκτελούμε την εντολή add-migration και τέλος την εντολή update-migration. Παρατηρούμε παρακάτω όλο το ιστορικό των migrations, το οποίο καταχωρείται στον φάκελο Data.



Εικόνα 50

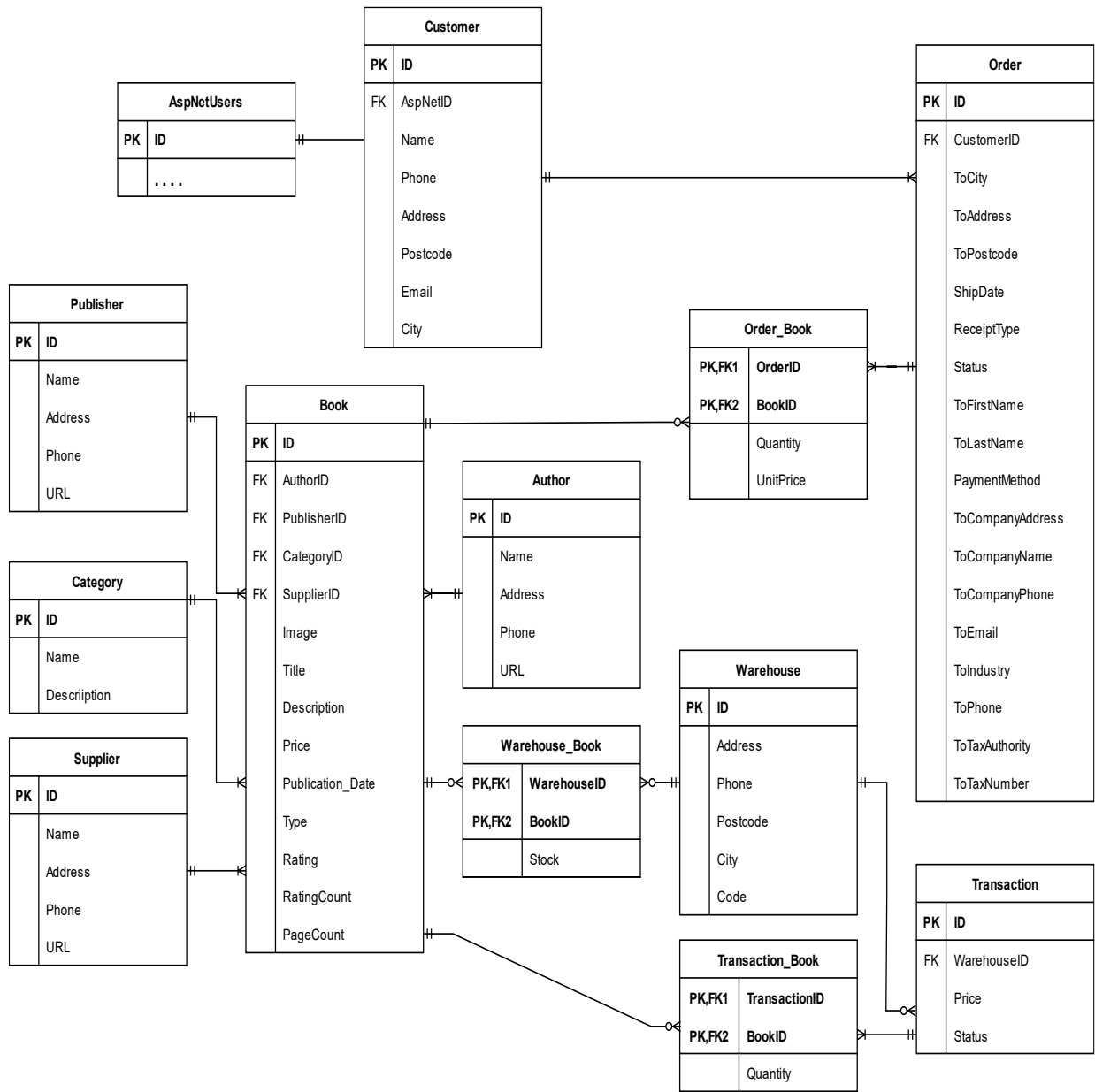


```
1 {
2   "ConnectionStrings": {
3     "DefaultConnection": "Server=localhost\\SQLEXPRESS;Database=BookstoreDB;Trusted_Connection=True;MultipleActiveResultSets=true",
4     "OnlineBookstoreIdentityDbContextConnection": "Server=(localdb)\\mssqllocaldb;Database=OnLineBookstore;Trusted_Connection=True;MultipleActiveResultSets=true",
5     "ApplicationDbContextConnection": "Server=(localdb)\\mssqllocaldb;Database=OnLineBookstore;Trusted_Connection=True;MultipleActiveResultSets=true"
6   },
7   "Logging": {
8     "LogLevel": {
9       "Default": "Information",
10      "Microsoft": "Warning",
11      "Microsoft.Hosting.Lifetime": "Information"
12    }
13  },
14  "AllowedHosts": "*"
15 }
```

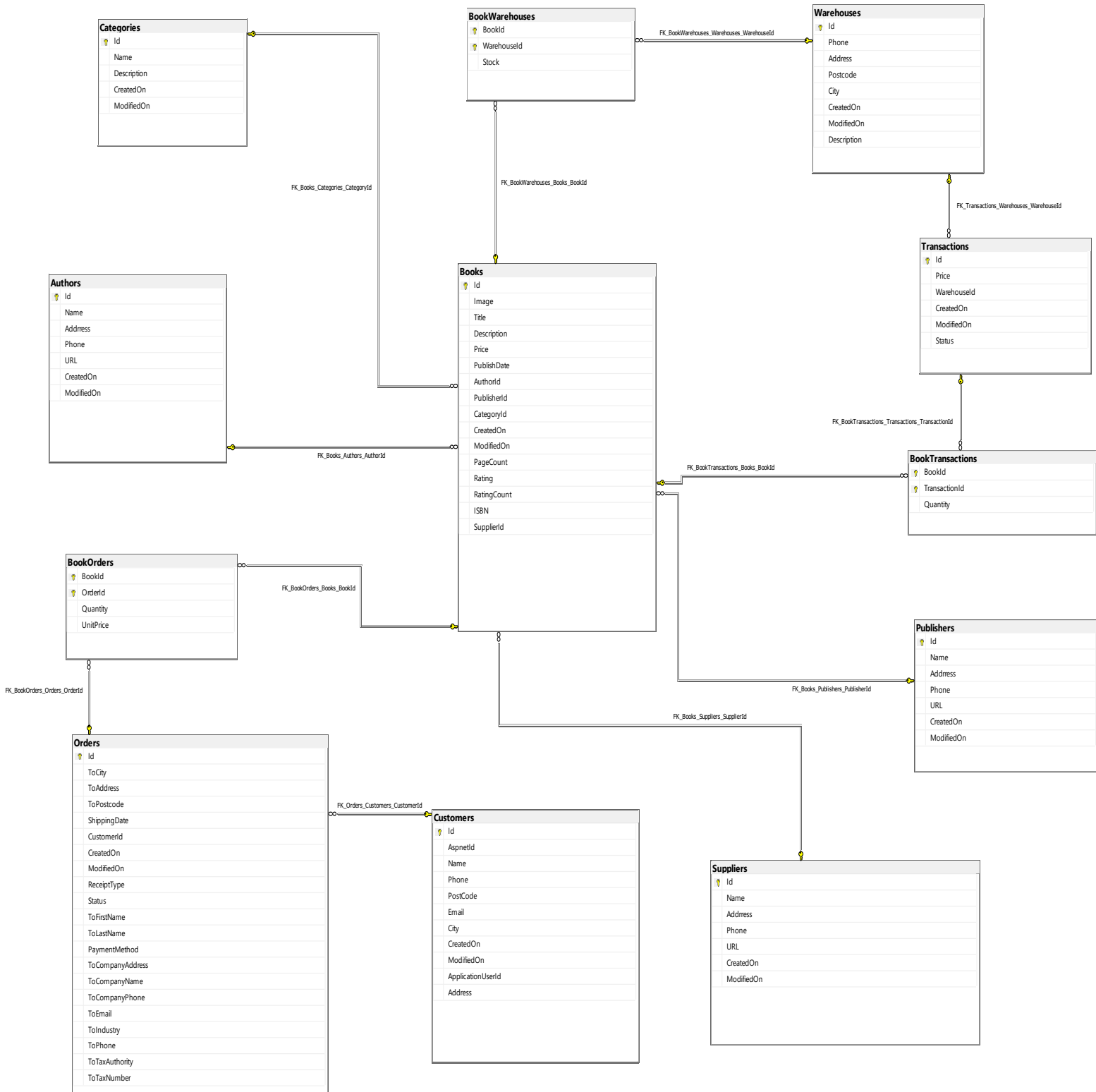
Εικόνα 51

Στο αρχείο app.settings εμφανίζονται τα connection strings της εφαρμογής με την βάση δεδομένων. Συγκεκριμένα , έχει επιλεγεί SQLEXPRESS και το όνομα της επιλεγμένης βάσης είναι BookstoreDB .

Στις παρακάτω εικόνες παρατίθενται το ERD και οι πίνακες της βάσης δεδομένων.



Εικόνα 52 ERD



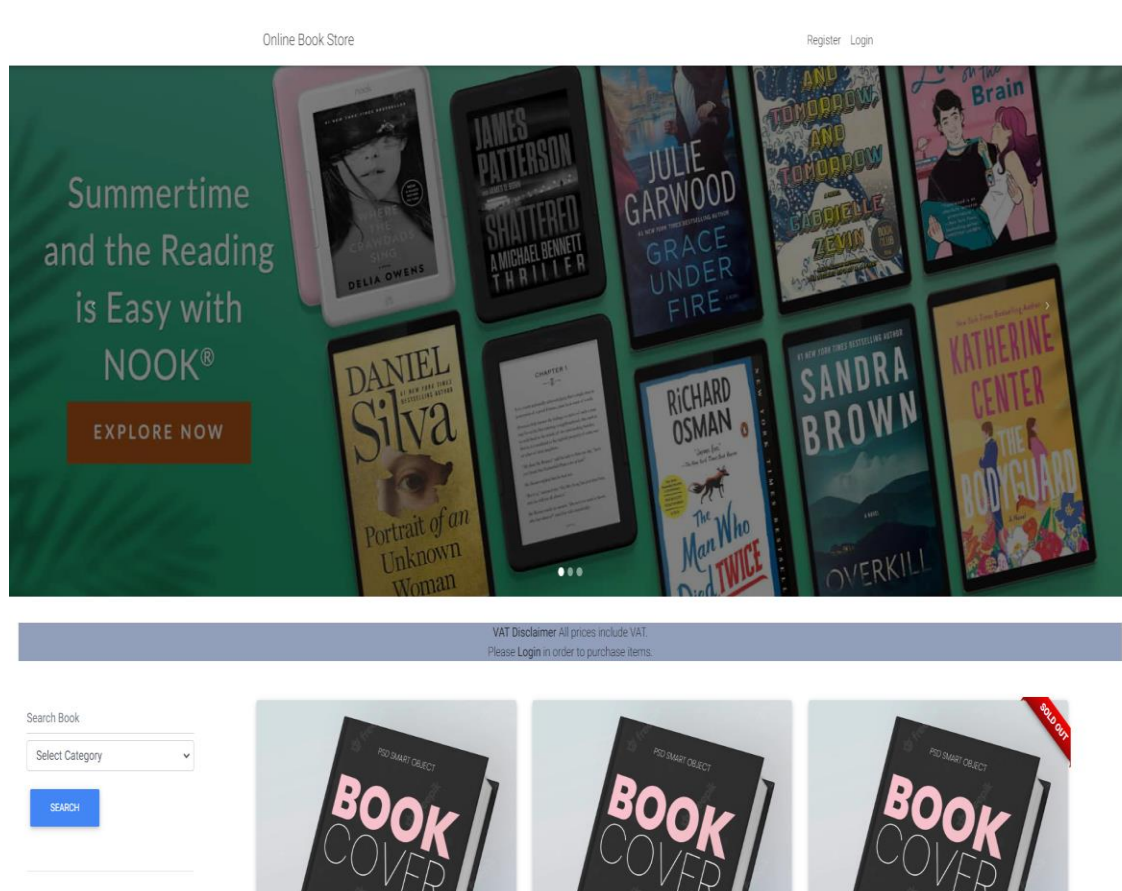
Εικόνα 43 Πίνακες Βάσης Δεδομένων

Ανάπτυξη Διαδικτυακής Εφαρμογής Ηλεκτρονικού Καταστήματος Βιβλιοπωλείου

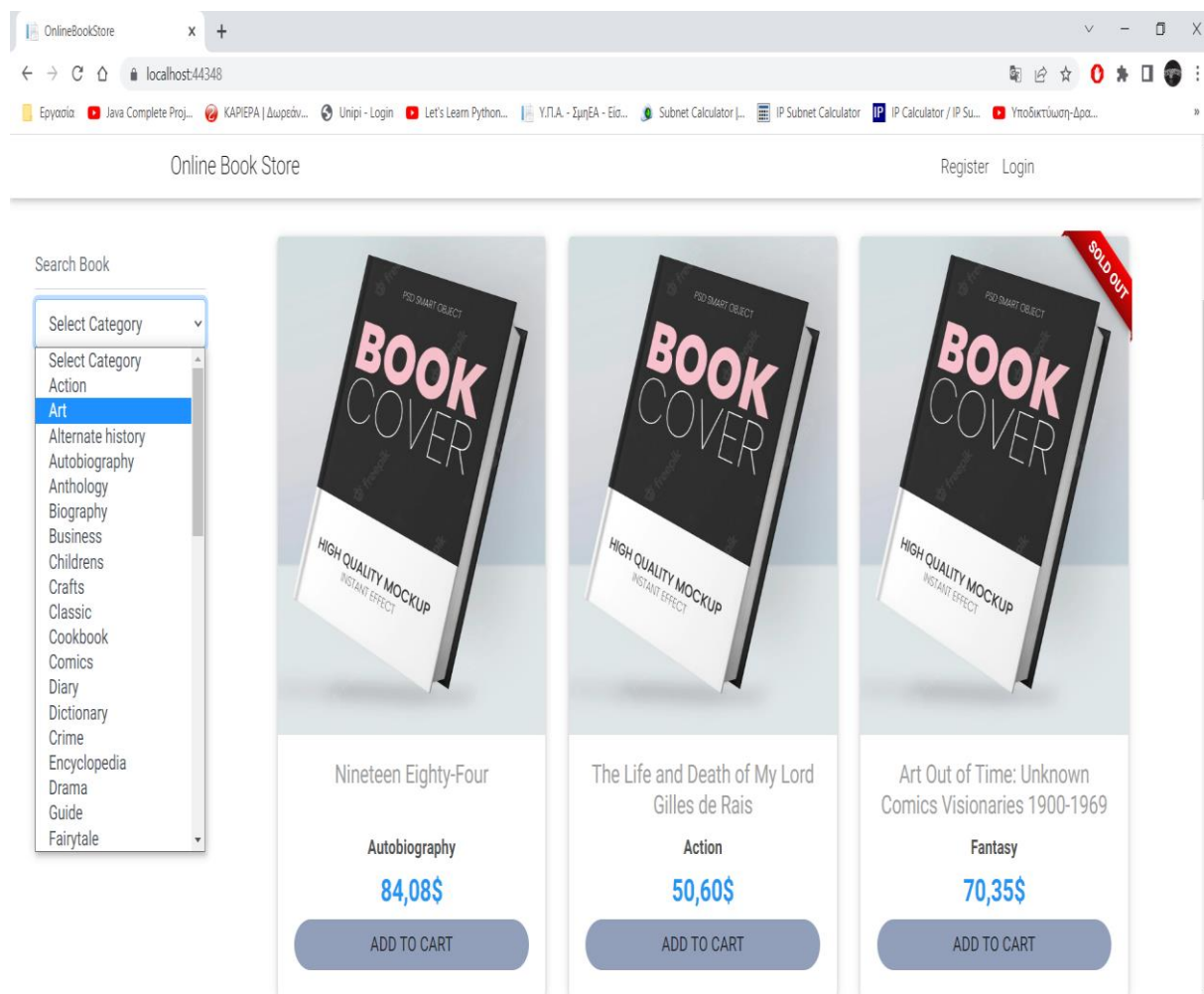
Κεφάλαιο 5 - Παρουσίαση λειτουργιών της εφαρμογής

5.1 Αρχική Σελίδα (HomePage)

Στην παρακάτω απεικονίζεται η αρχική σελίδα . Στο πάνω μέρος εμφανίζονται οι επιλογές Register και Login . Για να πραγματοποιήσει μια παραγγελία ένας πελάτης αρχικά πρέπει να εγγραφεί και να δημιουργήσει λογαριασμό. Σε περίπτωση που έχει ήδη δημιουργήσει λογαριασμό, θα πραγματοποιήσει login. Όπως παρατηρούμε στην εικόνα 54 , ένας χρήστης χωρίς να πραγματοποιήσει login, μπορεί να πλοηγηθεί στην ιστοσελίδα και να δει τους τίτλους και τις τιμές των βιβλίων χωρίς να έχει την δυνατότητα να τα τοποθετήσει στο καλάθι και να προχωρήσει σε παραγγελία. Δίνεται επίσης η δυνατότητα αναζήτησης βιβλίου ανά κατηγορία.



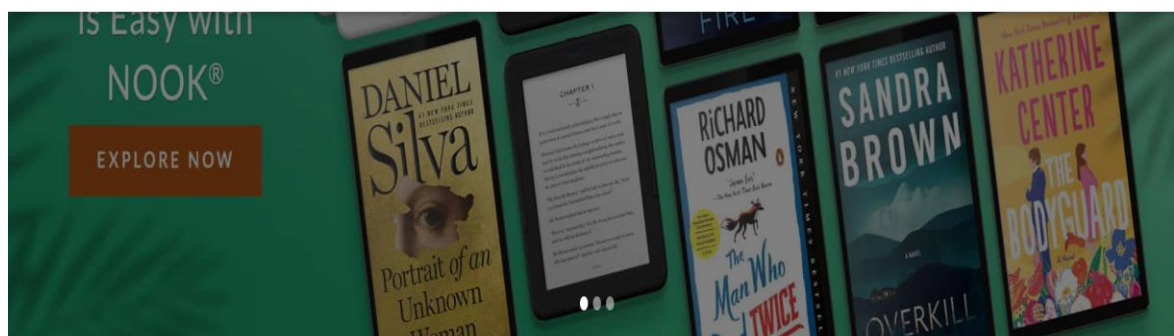
Εικόνα 44



Εικόνα 55

Σε περίπτωση που ο χρήστης δεν έχει πραγματοποιήσει είσοδο (login) , προσπαθήσει να κλικάρει πάνω σε ένα βιβλίο ανακατευθύνεται στην ιστοσελίδα της σύνδεσης.(login)

Online Book Store

[Register](#) [Login](#)

Log in

Use a local account to log in.

Email

Password

Remember me?

LOG IN

[Forgot your password?](#)

[Register as a new user](#)

Use another service to log in.

There are no external authentication services configured. See [this article](#) for details on setting up this ASP.NET application to support logging in via external services.

Εικόνα 45

5.2 Εγγραφή Πελάτη (Customer Register)

Online Book Store Register Login

Register

Create a new account.

Email

Password

Confirm password

Phone number

Address

Post Code

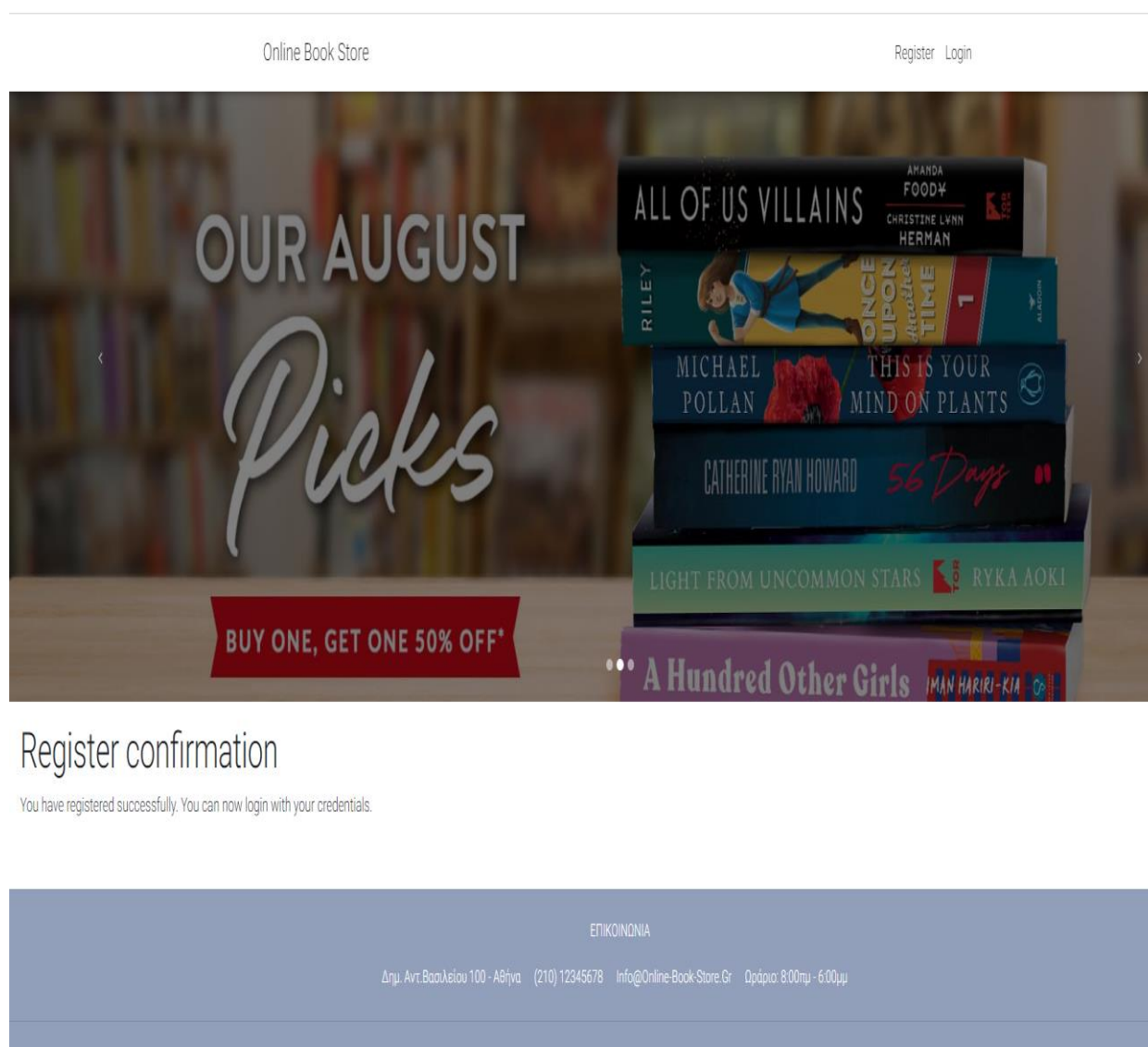
City

Use another service to register.

There are no external authentication services configured. See [this article](#) for details on setting up this ASPNET application to support logging in via external services.

Εικόνα 46

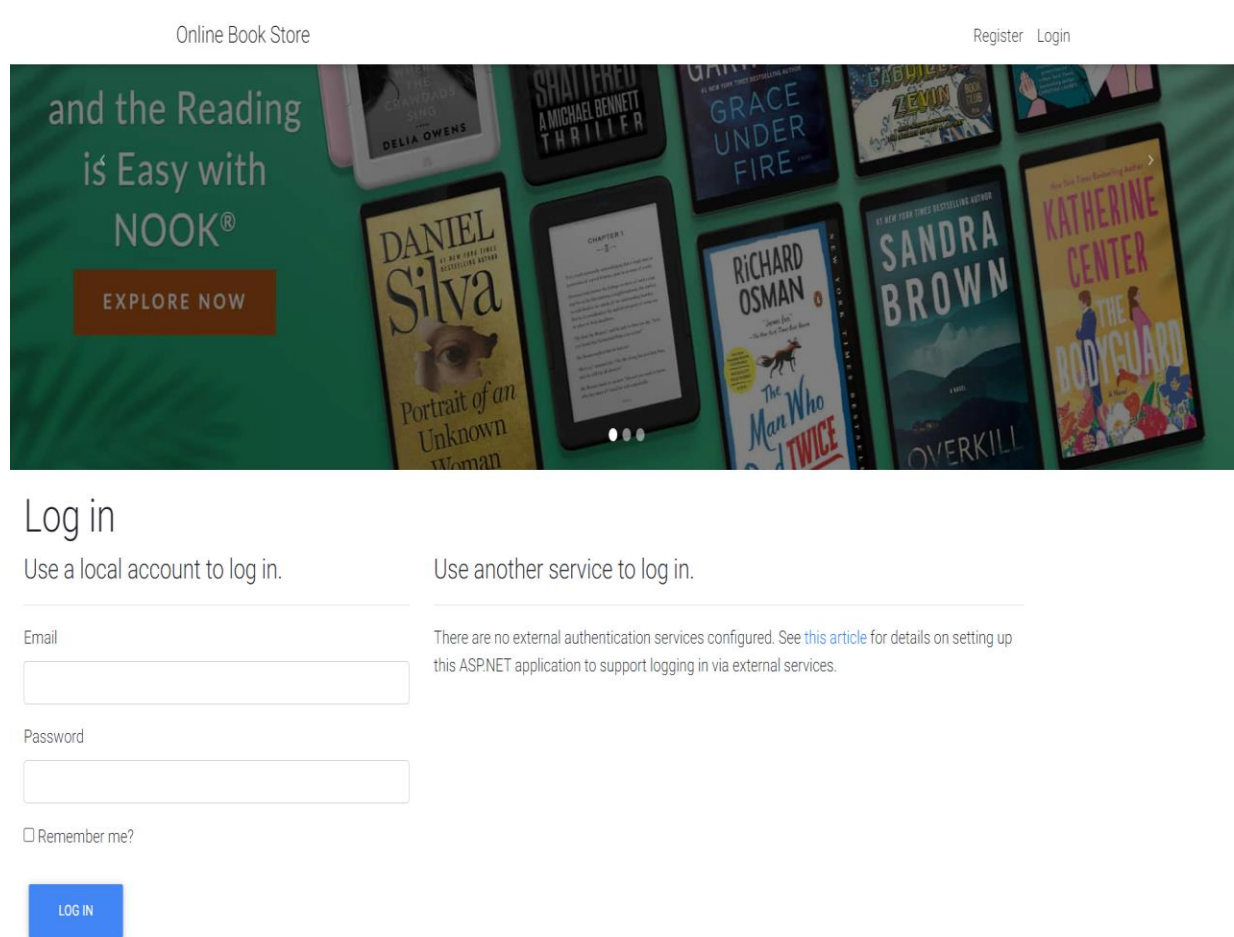
Αφού συμπληρώσουμε τα απαραίτητα στοιχεία στα πεδία της εγγραφής ,πατώντας το κουμπί Register , ολοκληρώνεται η εγγραφή και εμφανίζεται μήνυμα επιβεβαίωσης επιτυχούς εγγραφής που βρίσκεται στην εικόνα 58.



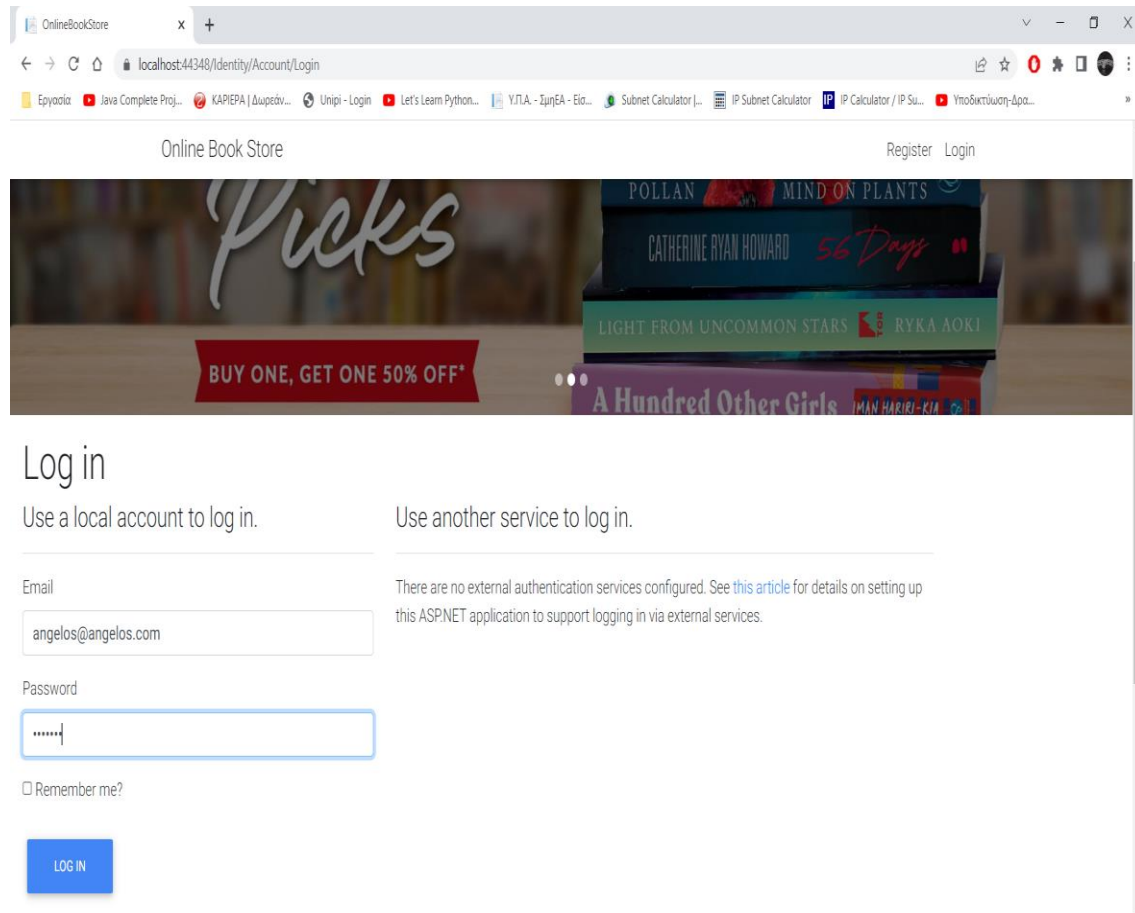
Εικόνα 47

5.3 Είσοδος Πελάτη - Login Customer

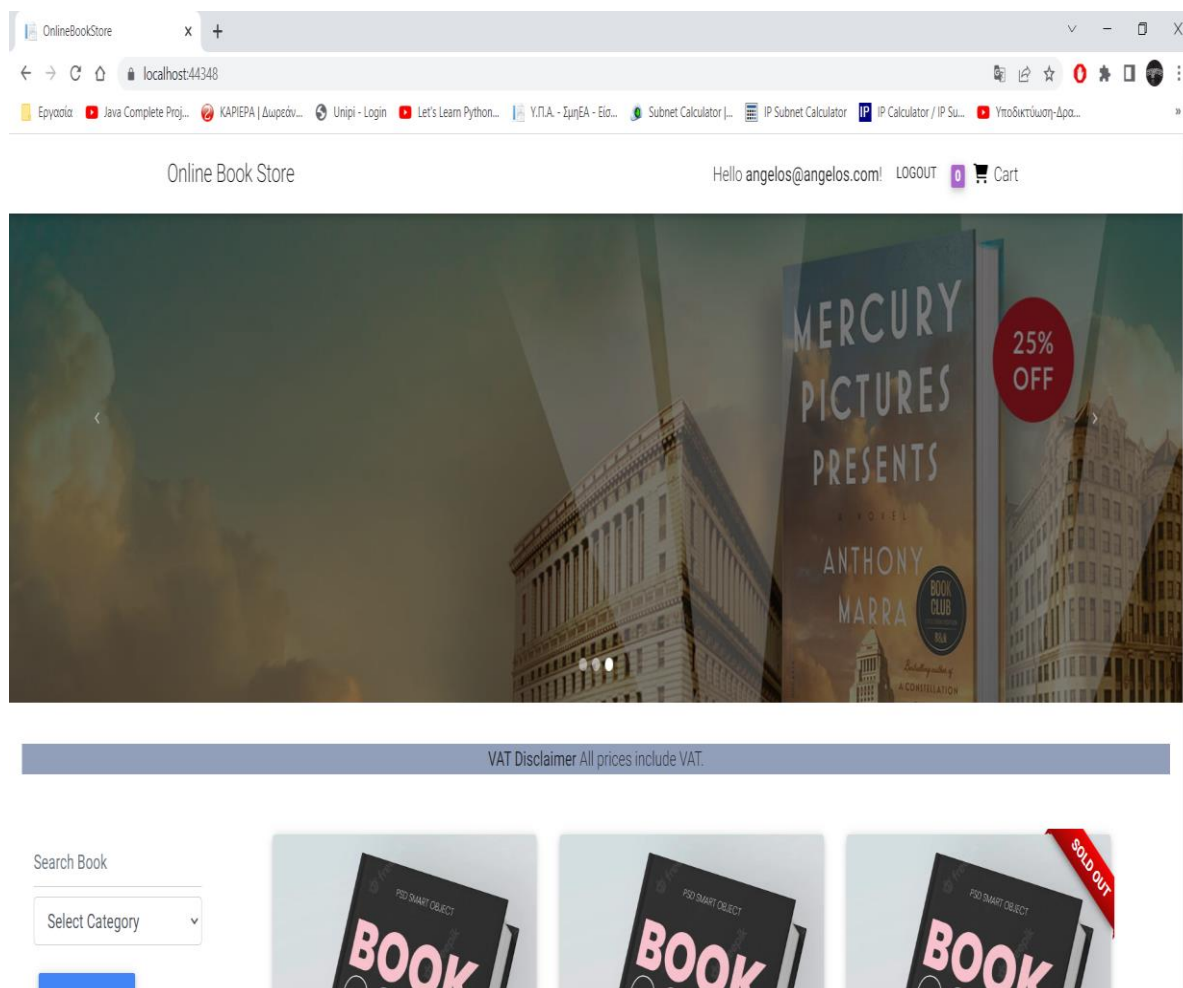
Κλικάρουμε το κουμπί **Login** και μας εμφανίζεται η παρακάτω εικόνα.



Εικόνα 48



Εικόνα 49



Εικόνα 50

Ο χρήστης αντιλαμβάνεται ότι έχει συνδεθεί επιτυχώς, καθώς παρατηρεί στο πάνω δεξιό μέρος της ιστοσελίδας να εμφανίζεται το **username** του, το οποίο στη παρούσα εφαρμογή είναι το **email** του και επίσης εμφανίζεται και η επιλογή **αποσύνδεση (LOGOUT)**. Επιπρόσθετα, εμφανίζεται μία αρίθμηση αριστερά από το σήμα του καλαθιού (Cart), το οποίο του δείχνει το πλήθος των βιβλίων που έχουν τοποθετηθεί στο καλάθι.

Ο χρήστης-πελάτης μπορεί να πλοηγηθεί στην ιστοσελίδα για να δει τα διαθέσιμα βιβλία όπως εμφανίζονται αρχικά δηλαδή χωρίς κάποιο κριτήριο επιλογής είτε επιλέγοντας να του εμφανιστούν ανά κατηγορία. Στη περίπτωση που υπάρχει η ετικέτα **SoldOut**, στην πάνω δεξιά εικόνα του βιβλίου, σημαίνει ότι το βιβλίο δεν είναι διαθέσιμο, το κουμπί Add to Cart είναι απενεργοποιημένο και η ετικέτα του κουμπιού εμφανίζεται με γκρι χρώμα. Αντίθετα τα βιβλία που έχουν απόθεμα, μπορούν να προστεθούν στο καλάθι και η ετικέτα του κουμπιού έχει χρώμα λευκό.

Ο χρήστης-πελάτης πατώντας κλικ πάνω στο email του, στο πάνω δεξιό μέρος της ιστοσελίδας μπορεί ανά πάσα στιγμή είτε να επικαιροποιήσει τα στοιχεία του προφίλ του είτε να αλλάξει το κωδικό του. Στο ίδιο μενού που εμφανίζεται και η επιλογή **My Orders**, με την οποία ο χρήστης βλέπει το ιστορικό των παραγγελιών του είτε αυτές είναι ολοκληρωμένες («**Confirmed**»)

είτε σε στάδιο αναμονής έγκρισης («**Pending**») από τον διαχειριστή. Οι λειτουργίες παρουσιάζονται στις παρακάτω εικόνες.

Online Book Store

Hello angelos@angelos.com! LOGOUT Cart

Manage your account

Change your account settings

- Profile
- Email
- Password
- My Orders

Profile

Username

Name

Phone number

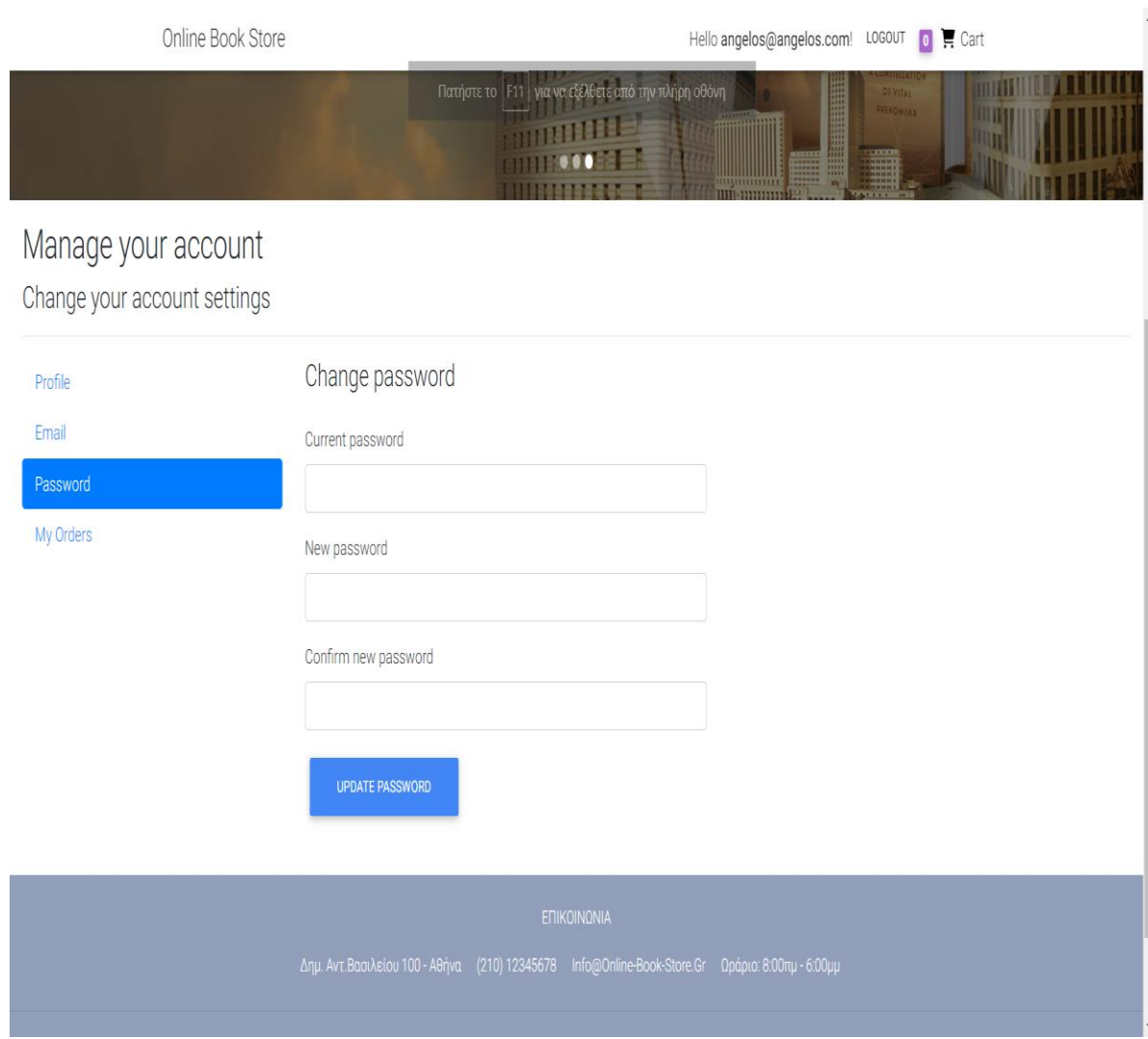
Address

Post Code

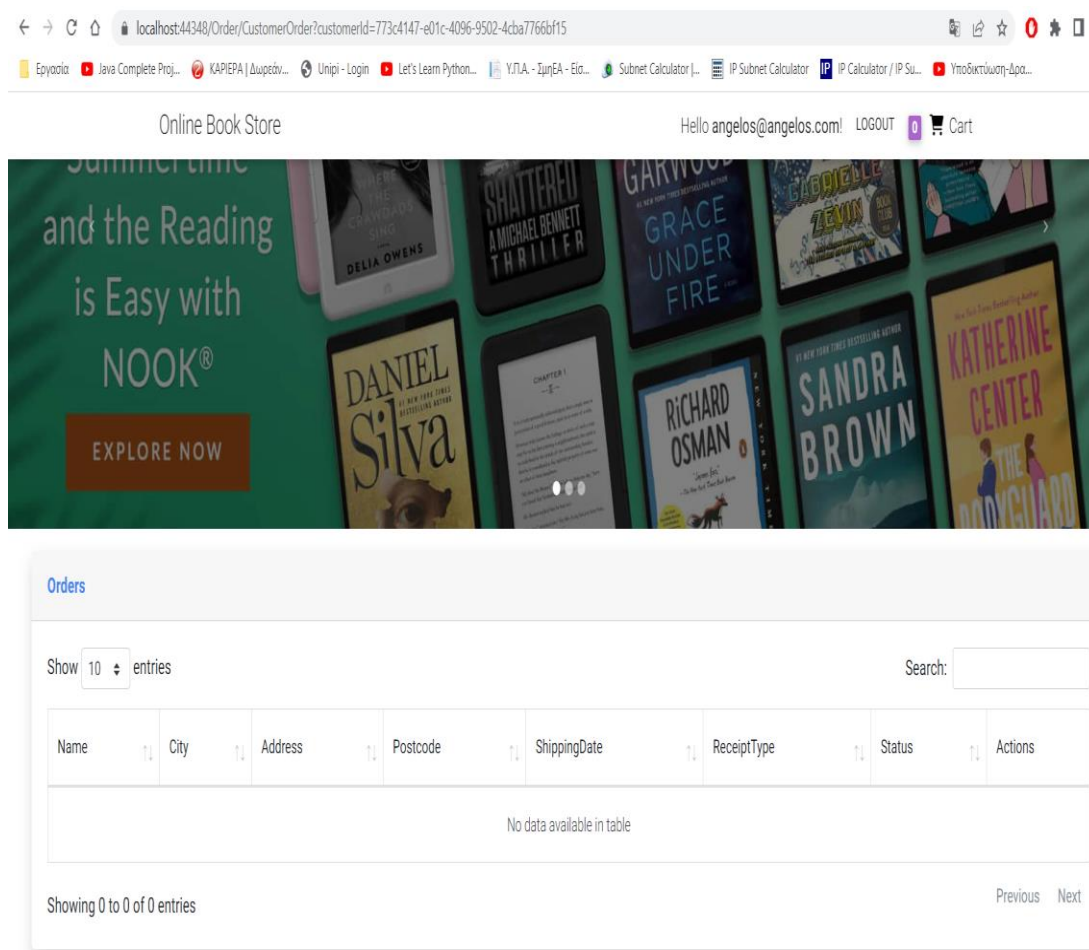
City

[SAVE](#)

Εικόνα 51



Εικόνα 52

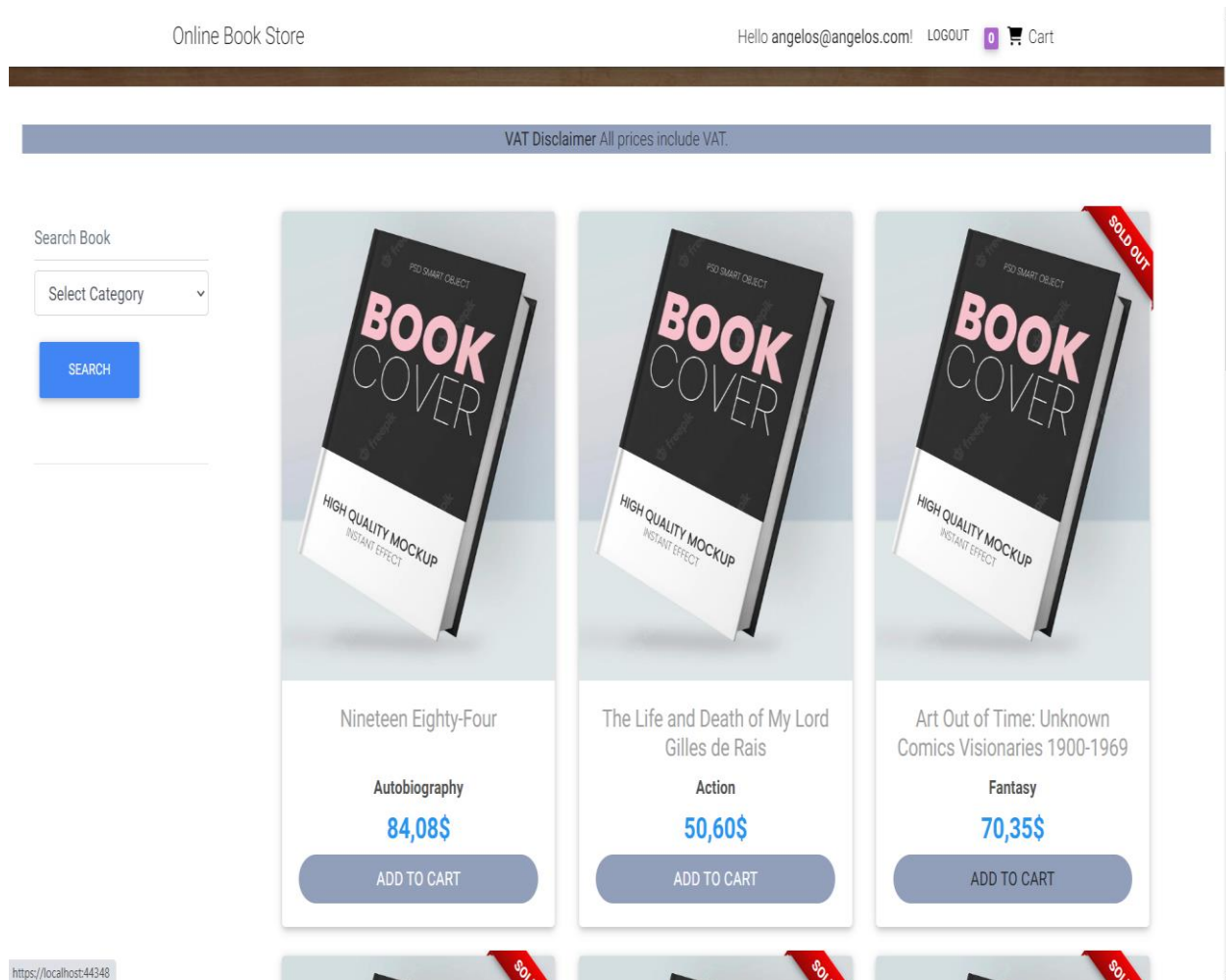


Εικόνα 53

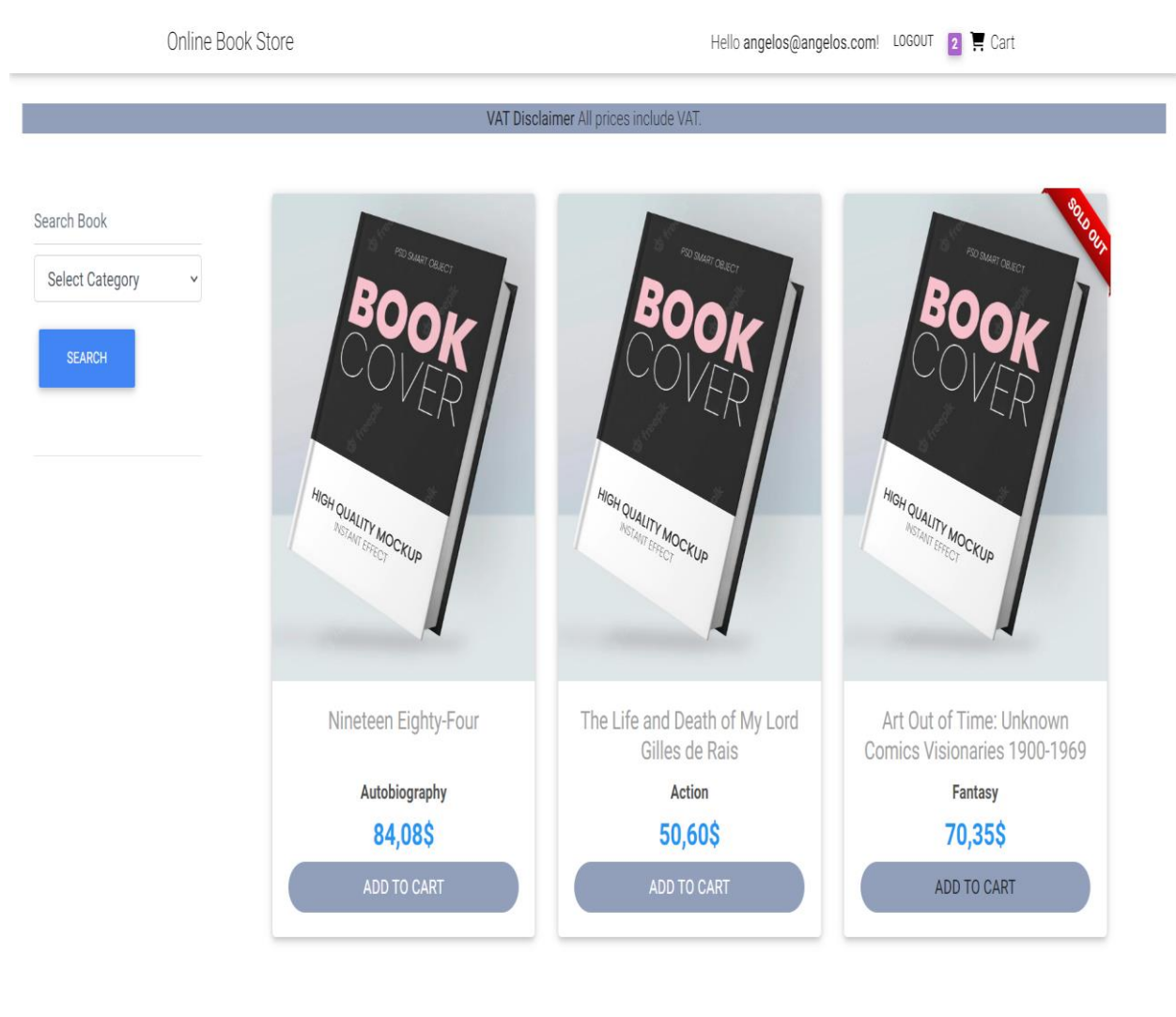
Στην παραπάνω εικόνα δεν εμφανίζονται παραγγελίες γιατί στο συγκεκριμένο λογαριασμό δεν έχουν πραγματοποιηθεί. Στην επόμενη ενότητα θα δούμε τη δημιουργία παραγγελίας.

5.4 Δημιουργία Παραγγελίας

Πατώντας κλικ στην επιλογή ADD TO CART σε ένα βιβλίο που είναι διαθέσιμο παρατηρούμε ότι το πλήθος των βιβλίων μεταβάλλεται. Στο παρακάτω σενάριο θα εισάγουμε στο καλάθι τα 2 πρώτα βιβλία που εμφανίζονται δηλαδή *Nineteen Eighty-Four* και *The Life and Death of My Lord Guiled de Rais*.

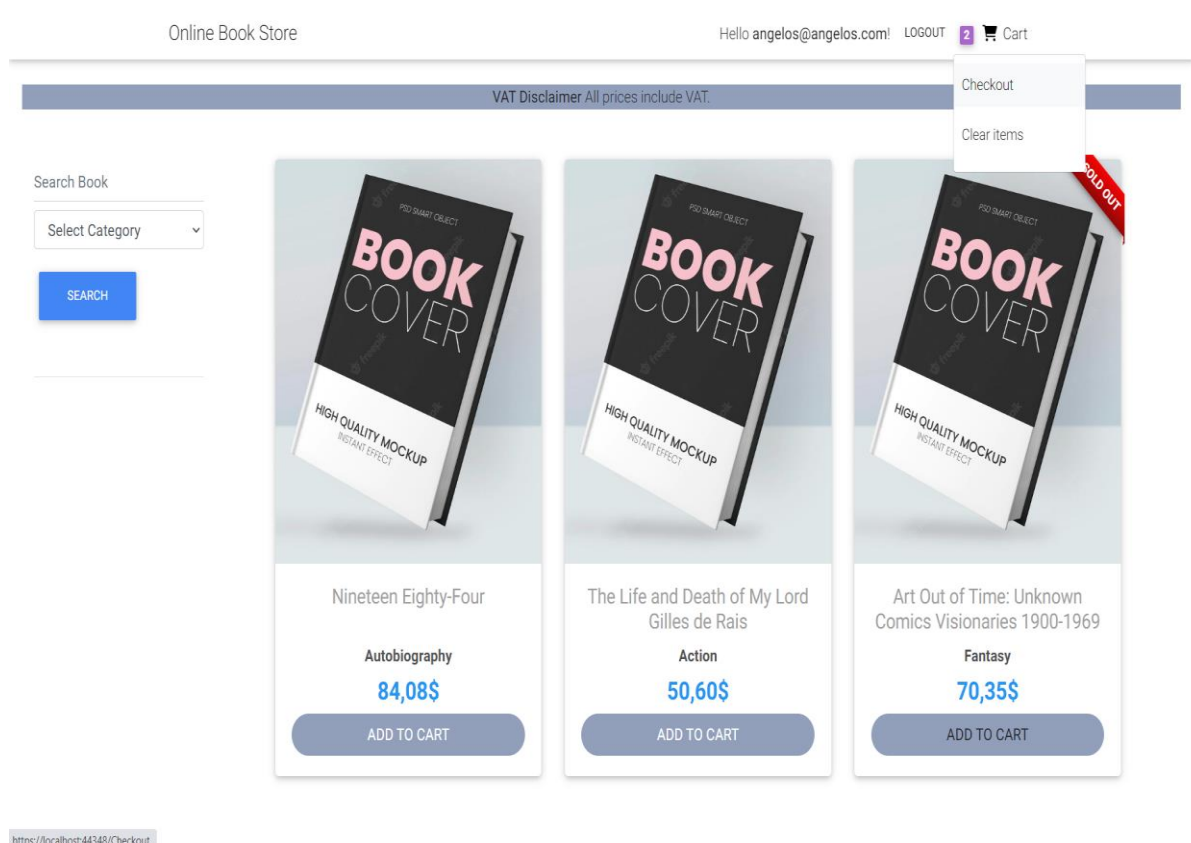


Εικόνα 54



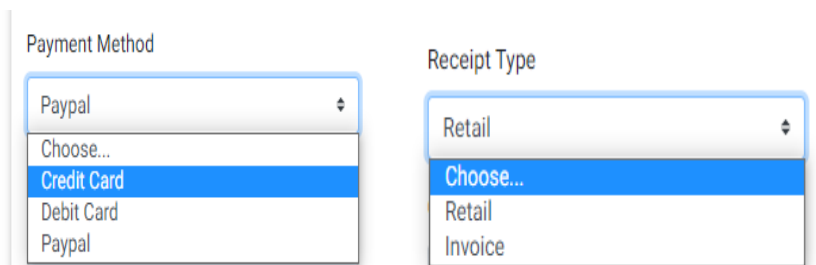
Εικόνα 55

Παρατηρούμε ότι το πλήθος των βιβλίων του καλαθιού είναι 2 . Πατώντας κλικ πάνω στο καλάθι εμφανίζονται οι επιλογές **CheckOut** και **ClearItems**. Η επιλογή **CheckOut** μας οδηγεί σε μία φόρμα συμπλήρωση στοιχείων παραγγελίας . Η επιλογή **ClearItems** αδειάζει το καλάθι. Στην εικόνα 67 εμφανίζονται οι 2 επιλογές.



Εικόνα 56

Πατώντας κλικ στο κουμπί **CheckOut** εμφανίζεται η παρακάτω φόρμα. Παρατηρούμε στη δεξιά πλευρά της φόρμας, τα βιβλία που έχουν τοποθετηθεί στο καλάθι από τον χρήστη σε ποια ποσότητα και ποια τιμή. Εμφανίζεται επίσης το τελικό κόστος. Απαιτείται να συμπληρωθούν όλα τα πεδία που εμφανίζονται για να μπορεί να ολοκληρωθεί επιτυχώς η παραγγελία. Ανάμεσα σε αυτά τα πεδία είναι και η επιλογή του τρόπου πληρωμής και η επιλογή φορολογικού παραστατικού. Παρακάτω εμφανίζονται οι επιλογές πληρωμής και τα είδη φορολογικών παραστατικών που μπορεί να επιλέξει ο χρήστης .



Εικόνα 68

Checkout form

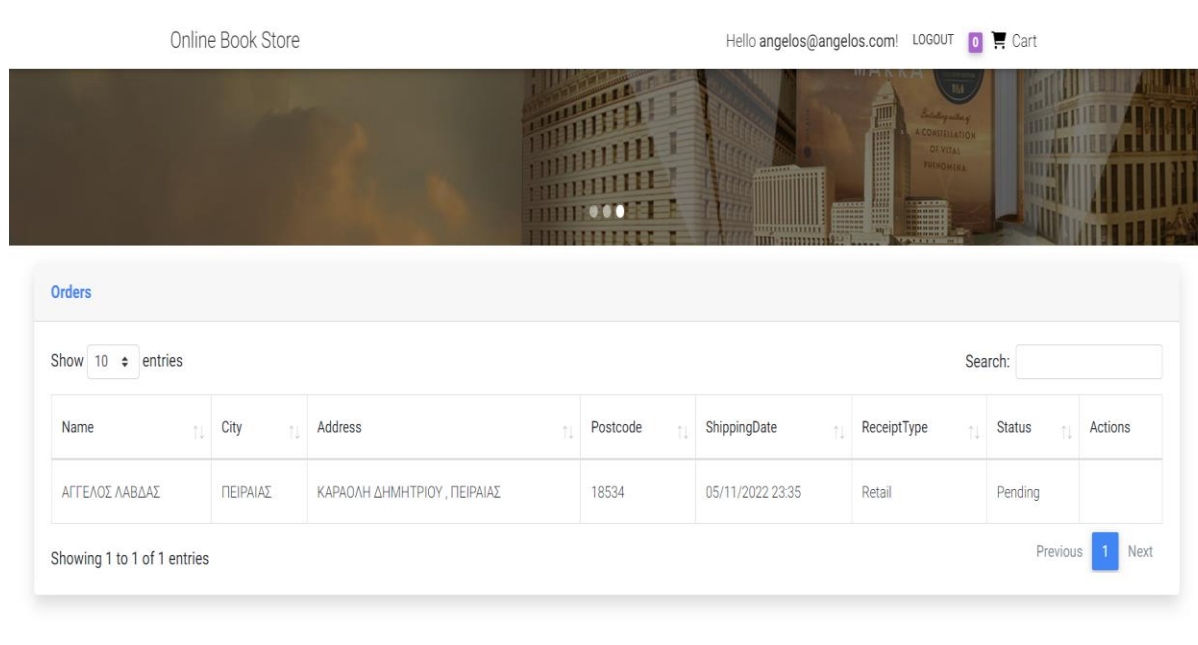
First Name	Last Name
ΑΓΓΕΛΟΣ	ΛΑΒΔΑΣ
Email angelos@angelos.com	
Phone 0000000000	
Address ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ, ΠΕΙΡΑΙΑΣ	
Address 2 (optional) Apartment or suite	
City ΠΕΙΡΑΙΑΣ	Post Code 18534
Payment Method Paypal	Receipt Type Retail
Name on card ΑΓΓΕΛΟΣ	Credit card number 0000 0000 0000 0000
Full name as displayed on card	
Expiration 12 2025	CVV 000
CONTINUE TO CHECKOUT	

Your cart 2

Nineteen Eighty-Four	1 x
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam cursus tincidunt nibh et dapibus. Nunc bibendum ex ut ornare pulvinar.	84,08
The Life and Death of My Lord Gilles de Rais	1 x
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam cursus tincidunt nibh et dapibus. Nunc bibendum ex ut ornare pulvinar.	50,60
Total (USD)	\$134,68

Εικόνα 69

Έχοντας συμπληρώσει όλα τα πεδία της φόρμας ο χρήστης- πελάτης, κλικάρει την επιλογή **CONTINUE TO CHECKOUT** . Μετά τον απαραίτητο έλεγχο από την εφαρμογή ότι όλα τα πεδία έχουν συμπληρωθεί, ο χρήστης μεταφέρεται αυτόματα στην ιστοσελίδα με το ιστορικό των παραγγελιών του. Αυτό σημαίνει ότι η παραγγελία ολοκληρώθηκε επιτυχώς και πλέον το **καλάθι** του είναι **άδειο**. Σε αυτή την ιστοσελίδα εμφανίζεται ένας πίνακας με το ιστορικό των παραγγελιών του πελάτη. Συγκεκριμένα εμφανίζονται βασικές πληροφορίες της παραγγελίας όπως ονοματεπώνυμο, διεύθυνση, Ταχυδρομικός Κώδικας, Ημερομηνία Παραγγελίας, Είδος Παραστατικού και Status παραγγελίας. Το Status εμφανίζεται Pending που σημαίνει ότι για να γίνει Confirmed θα πρέπει να ο διαχειριστής του ηλεκτρονικού καταστήματος να εγκρίνει την πώληση . Μετά την έγκριση από τον διαχειριστή, στο πεδίο Actions θα εμφανιστεί κουμπί με το οποίο ο χρήστης-πελάτης θα μπορεί να κάνει λήψη σε αρχείο.pdf το φορολογικό παραστατικό του. Στο συγκεκριμένο σενάριο, το επιλεγμένο παραστατικό είναι retail δηλαδή απόδειξη λιανικής πώλησης.



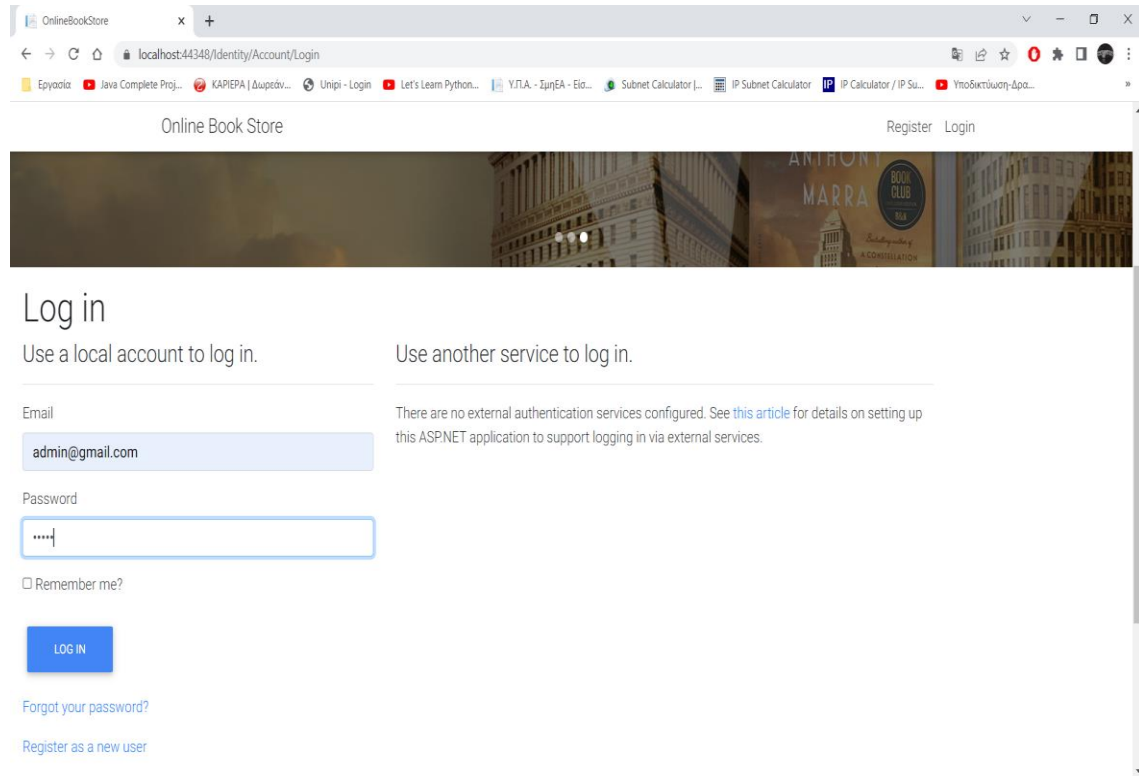
The screenshot shows the 'Orders' section of an online book store. At the top, there is a navigation bar with 'Online Book Store', the user email 'Hello angelos@angelos.com!', a 'LOGOUT' button, and a shopping cart icon with '0' items. Below the navigation bar is a banner image of a cityscape. The main content area is titled 'Orders' and contains a table with one order entry. The table has columns for Name, City, Address, Postcode, ShippingDate, ReceiptType, Status, and Actions. The order entry shows the name 'ΑΓΓΕΛΟΣ ΛΑΒΔΑΣ', city 'ΠΕΙΡΑΙΑΣ', address 'ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ, ΠΕΙΡΑΙΑΣ', postcode '18534', shipping date '05/11/2022 23:35', receipt type 'Retail', and status 'Pending'. Below the table, there is a search bar and pagination controls showing 'Showing 1 to 1 of 1 entries' and 'Previous 1 Next'.

Name	City	Address	Postcode	ShippingDate	ReceiptType	Status	Actions
ΑΓΓΕΛΟΣ ΛΑΒΔΑΣ	ΠΕΙΡΑΙΑΣ	ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ, ΠΕΙΡΑΙΑΣ	18534	05/11/2022 23:35	Retail	Pending	

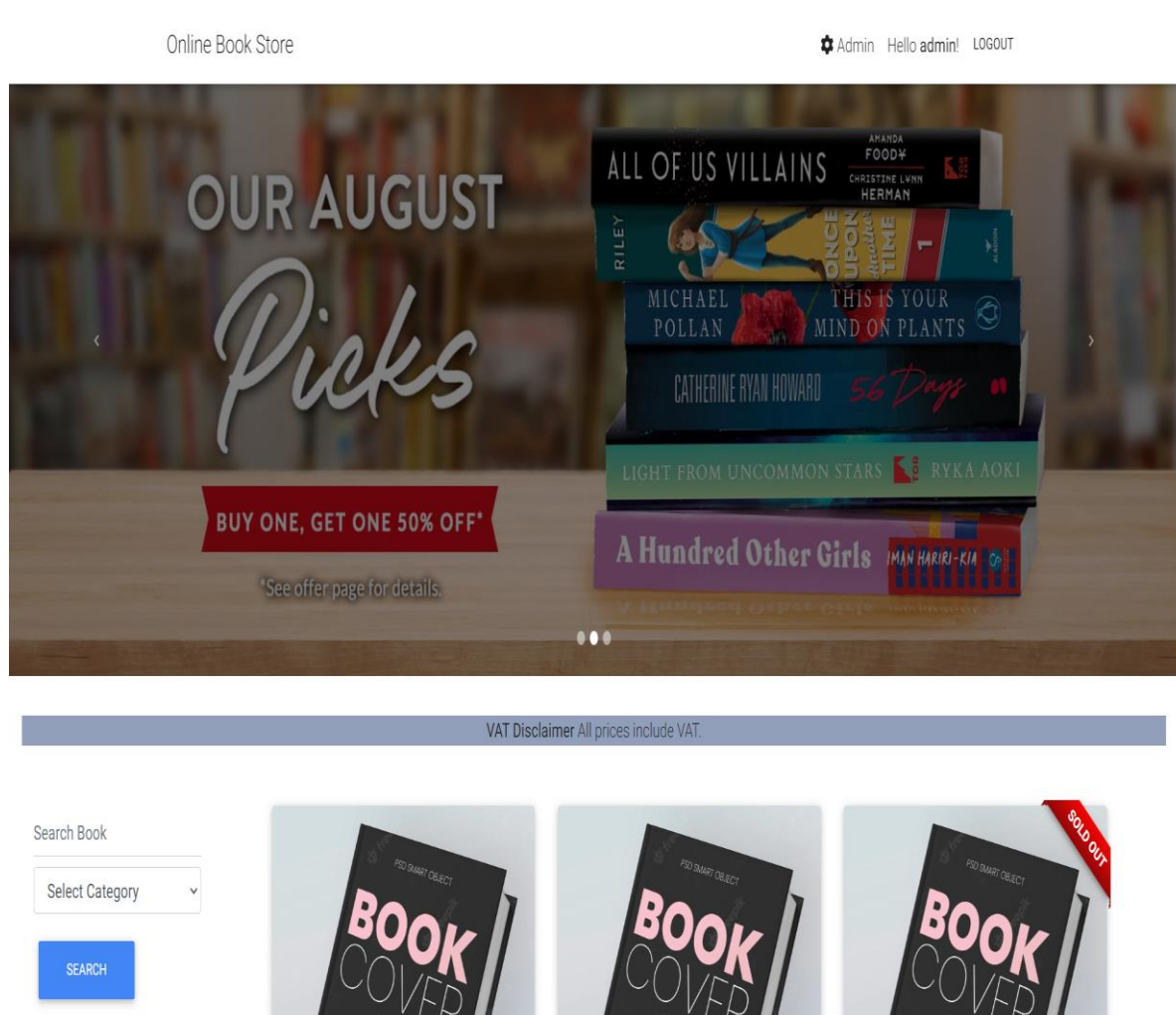
Εικόνα 70

5.5 Είσοδος Διαχειριστή (Admin Login) & Λειτουργίες Διαχειριστή

Με συγκεκριμένους κωδικούς (ρόλου Admin) εισέρχεται ο χρήστης διαχειριστής στην ιστοσελίδα.



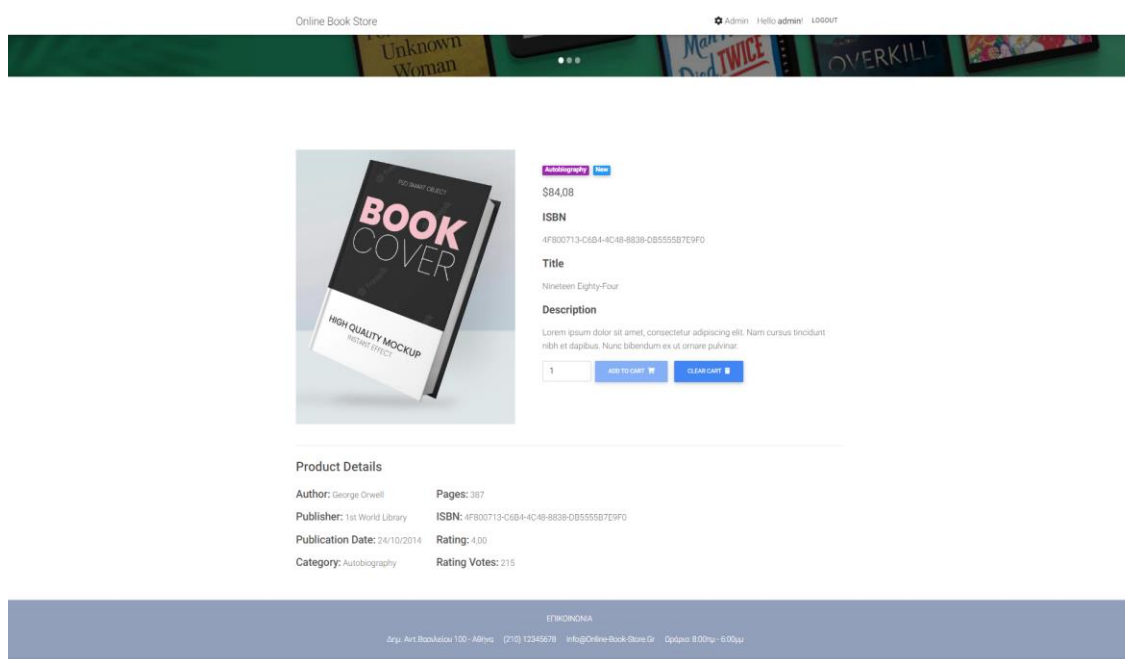
Εικόνα 71



Εικόνα 72

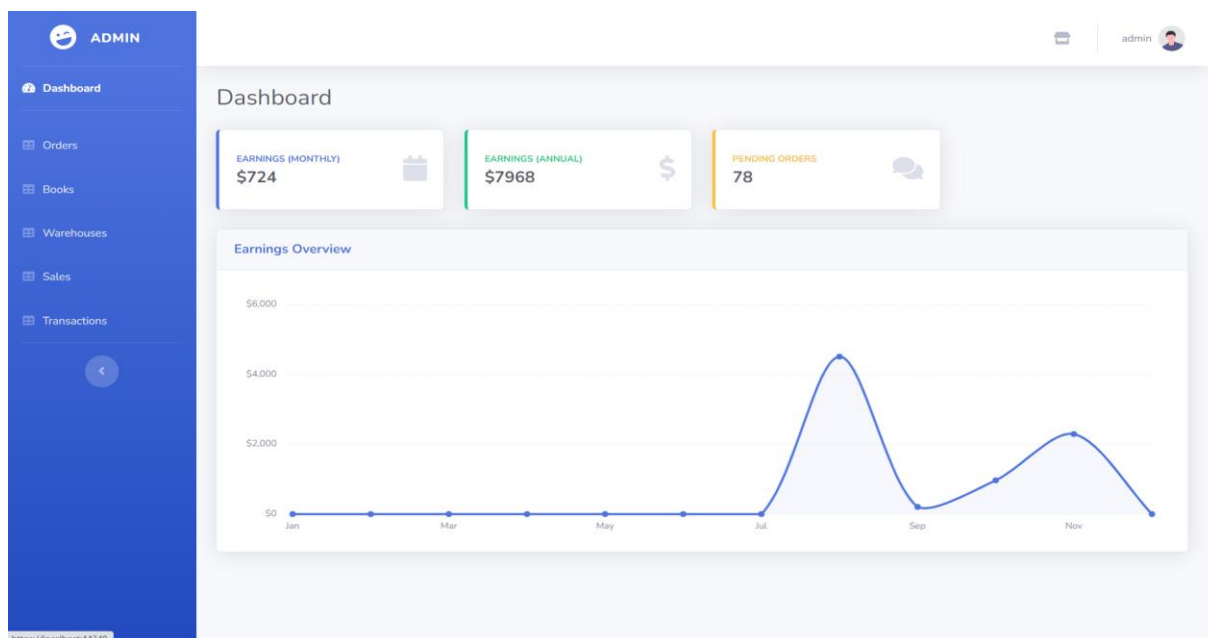
Εφόσον ο χρήστης **Admin** εισέλθει επιτυχώς στην ιστοσελίδα, παρατηρεί στο πάνω δεξιό μέρος της μπάρας ένα κουμπί με εικονίδιο ένα γρανάζι δίπλα στη λέξη Admin, την εμφανιζόμενη ετικέτα καλωσορίσματος *Hello admin!* και την επιλογή αποσύνδεσης LOGOUT.

Ο χρήστης **Admin** έχει κάποιες λειτουργίες ίδιες με αυτές του χρήστη-πελάτη όπως να μπορεί να πλοηγηθεί στην ιστοσελίδα για να δει τα βιβλία που είναι διαθέσιμα, να μπορεί να κάνει αναζήτηση βιβλίου με κριτήριο την **κατηγορία** και να μπορεί επιλέξει ένα βιβλίο κάνοντας κλικ πάνω στην φωτογραφία του βιβλίου για να δει αναλυτικές πληροφορίες γι' αυτό. Ο χρήστης-Admin δεν έχει τη δυνατότητα χρήσης του καλαθιού.



Εικόνα 73

Επιλέγοντας το κουμπί με το γρανάζι στον χρήστη-Admin εμφανίζεται η ιστοσελίδα που βρίσκεται στην παρακάτω εικόνα. Συγκεκριμένα εμφανίζεται ένα **Dashboard** που στο κέντρο του εμφανίζονται οι μηνιαίες πωλήσεις(έσοδα) , οι ετήσιες πωλήσεις και οι παραγγελίες που βρίσκονται σε κατάσταση «**Pending**» δηλαδή αναμονή έγκρισης . Επιπρόσθετα , υπάρχουν κάποιες επιλογές που βρίσκονται με μορφή πλευρικής μπάρας στην αριστερή άκρη της ιστοσελίδας .



Εικόνα 74

Ανάπτυξη Διαδικτυακής Εφαρμογής Ηλεκτρονικού Καταστήματος Βιβλιοπωλείου

5.5.1 Αναζήτηση Παραγγελιών Πελατών

The screenshot shows the Admin interface for managing orders. The sidebar on the left contains navigation options: Dashboard, Orders, Books, Warehouses, Sales, and Transactions. The main content area displays a table of orders under the heading 'Orders'. The table has columns for Name, City, Address, Postcode, ShippingDate, ReceiptType, Status, and Actions. The status of the last order is 'Pending'.

Name	City	Address	Postcode	ShippingDate	ReceiptType	Status	Actions
customer5@gmail.com	pireaus	labraki 77	18539	02/10/2022 22:02	Invoice	Confirmed	✓📄
angelos@gmail.com	Pireaus	Akth Moutsopoulou 180	18531	11/10/2022 18:58	Retail	Confirmed	✓📄
angelos@gmail.com	Pireaus	Akth Moutsopoulou 180	18531	11/10/2022 19:09	Retail	Confirmed	✓📄
angelos@gmail.com	Pireaus	Akth Moutsopoulou 180	18531	15/10/2022 13:30	Invoice	Confirmed	✓📄
angelos@gmail.com	Pireaus	Akth Moutsopoulou 180	18539	15/10/2022 13:34	Retail	Confirmed	✓📄
angelos@gmail.com	Pireaus	Akth Moutsopoulou 180	18531	15/10/2022 18:27	Retail	Confirmed	✓📄
Angelos	Pireaus	Plateia Korah Peiraias	18533	25/10/2022 11:58	Retail	Confirmed	✓📄
Angelos	ΑΘΗΝΑ	ΠΑΝΕΠΙΣΤΗΜΙΟΥ 59 , ΑΘΗΝΑ	105 64	31/10/2022 19:30	Retail	Confirmed	✓📄
ΑΓΓΕΛΟΣ	ΠΕΙΡΑΙΑΣ	Εθνάρχου Μακαρίου 3 , Πειραιάς	18547	31/10/2022 22:53	Retail	Confirmed	✓📄
ΑΓΓΕΛΟΣ	ΠΕΙΡΑΙΑΣ	Εθνάρχου Μακαρίου 3 , Πειραιάς	18547	31/10/2022 22:58	Retail	Confirmed	✓📄
ΑΓΓΕΛΟΣ ΛΑΒΔΑΣ	ΠΕΙΡΑΙΑΣ	ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ , ΠΕΙΡΑΙΑΣ	18534	05/11/2022 23:35	Retail	Pending	✓📄

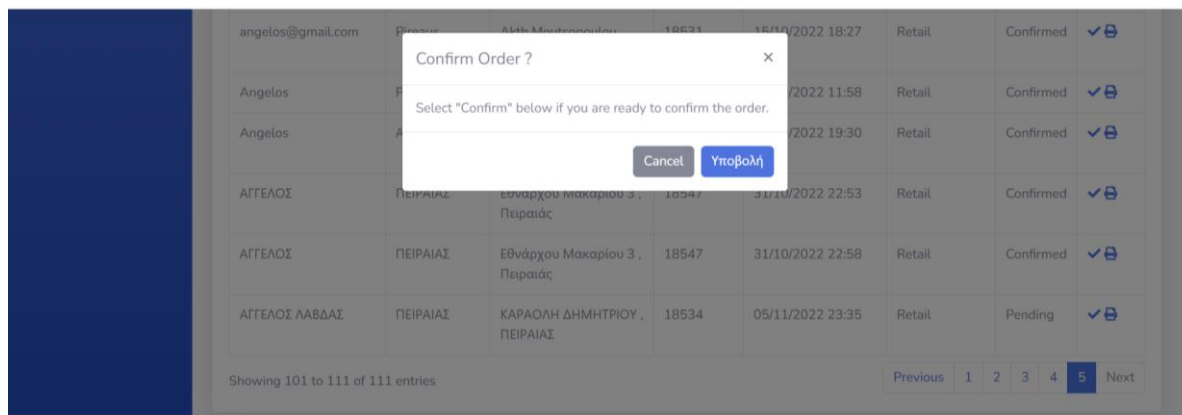
Showing 101 to 111 of 111 entries

Εικόνα 75

Η επιλογή **Orders** εμφανίζει στον διαχειριστή, όλες τις καταχωρημένες παραγγελίες των πελατών με αναλυτικά στοιχεία όπως ονοματεπώνυμο πελάτη ,Διεύθυνση, Πόλη, Ημερομηνία, είδος παραστατικού και status παραγγελίας. Το Status της παραγγελίας μπορεί να είναι «**Pending**» όταν ο Admin δεν έχει εγκρίνει την παραγγελία ή «**Confirmed**» όταν την έχει εγκρίνει. Οι επιλογές που εμφανίζονται ως προς το είδος παραστατικού είναι Retail και Invoice . Στην τελευταία στήλη του παραπάνω πίνακα εμφανίζονται 2 ακόμα κουμπιά. Το πρώτο κουμπί με εικονίδιο (✓) αλλάζει το status της παραγγελίας από «**Pending**» σε «**Confirmed**». Το δεύτερο κουμπί με το εικονίδιο έγγραφο (📄) δίνει την δυνατότητα στο χρήστη-Admin να κάνει λήψη του φορολογικού παραστατικού(Τιμολόγιο-Δελτίο Αποστολής ή Απόδειξη Λιανικής Πώλησης) της συγκεκριμένης παραγγελίας.

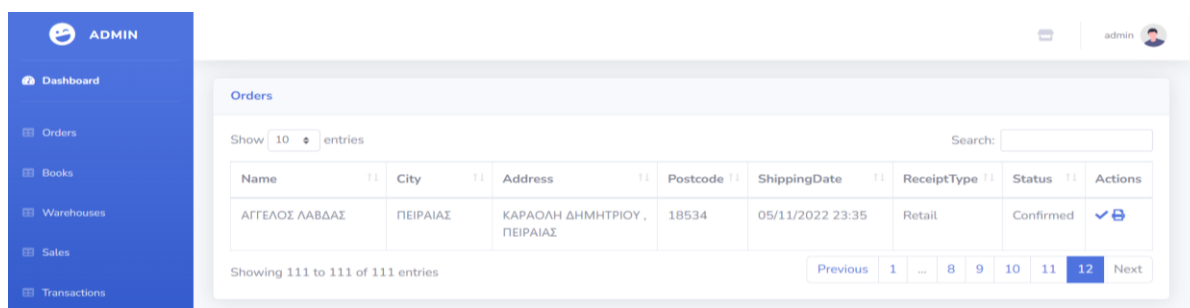
Στο τέλος της παραπάνω εικόνας παρατηρούμε και την παραγγελία που δημιούργησα ως χρήστης-πελάτης στην ενότητα 5.4 και συγκεκριμένα στην εικόνα 70.

Πατώντας κλικ στο κουμπί με το εικονίδιο (✓) μας εμφανίζεται το παρακάτω μήνυμα.



Εικόνα 76

Επιλέγουμε **Υποβολή** και αλλάζει η κατάσταση της παραγγελίας από «**Pending**» σε «**Confirmed**».



Εικόνα 77

Σε αυτό το σημείο, όπου η παραγγελία έχει γίνει «**Confirmed**», ο συγκεκριμένος πελάτης εφόσον συνδεθεί στο λογαριασμό του, έχει τη δυνατότητα να πάει στην επιλογή **Orders** και να κάνει λήψη του φορολογικού παραστατικού που σχετίζεται με τη παραγγελία του. Στη συγκεκριμένη περίπτωση είναι Απόδειξη Λιανικής Πώλησης.

Πατώντας κλικ στο κουμπί με το εικονίδιο (📄) γίνεται λήψη του φορολογικού παραστατικού που είχε επιλεγεί από τον πελάτη κατά την παραγγελία.

ΑΠΟΔΕΙΞΗ ΛΙΑΝΙΚΗΣ ΠΩΛΗΣΗΣ

Ημερομηνία: 03/11/2022 23:35
Κωδικός Πελάτη: 773c4147-e01c-4096-9502-4cba7766b15
Όνοματεπώνυμο: ΑΓΓΕΛΟΣ ΛΑΒΔΑΣ
Διεύθυνση: ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ, ΠΕΙΡΑΙΑΣ
Τηλέφωνο: 0000000000
Email: angelos@angelos.com
Μέθοδος Πληρωμής: Paypal
Τύπος Παραστατικού: Απόδειξη Λιανικής

Επωνυμία: OnlineBookStore
Διεύθυνση: Δημ. Αντ.Βασιλείου 100
Ταχυδρομικός Κώδικας: 19344
Σειρά Παραστατικού: 20143
Αριθμός Παραστατικού: ΑΠΑΠ20143
Τηλέφωνο: (210) 12345678
Email: Info@Online-Book-Store.gr
ΑΦΜ: 000123456

ΚΩΔ	ΠΕΡΙΓΡΑΦΗ ΕΙΛΟΥΣ	ΜΟΝ. ΜΕΤΡ.	ΠΟΣΟΤΗΤΑ	ΣΥΝΤ. ΦΠΑ	ΤΙΜΗ ΜΟΝ. ΧΩΡΙΣ ΦΠΑ	ΤΙΜΗ ΜΟΝ. ΜΕ ΦΠΑ	ΠΟΣΟΣΤΟ ΕΚΠΤΩΣΗΣ	ΚΑΘΑΡΗ ΑΞΙΑ	ΣΥΝΟΛΙΚΗ ΑΞΙΑ
4F800713-C6B4-4C48-8838-DB5555B7E9F0	Nineteen Eighty-Four	TEM	1,00	6%	79,04 €	84,08 €	0%	79,04 €	84,08 €
DC6CA43C-D229-4DC4-9BA0-D49B58EA9FB5	The Life and Death of My Lord Gilles de Rais	TEM	1,00	6%	47,56 €	50,60 €	0%	47,56 €	50,60 €

ΟΛΙΚΗ ΑΞΙΑ (Χωρίς ΦΠΑ)	ΣΥΝΟΛΙΚΟΣ ΦΠΑ	ΠΛΗΡΩΤΕΟ ΠΟΣΟ
126,60 €	8,08 €	134,68 €

Εικόνα 78

Στη περίπτωση που είχε επιλεγεί Τιμολόγιο θα εμφανιζόταν το παρακάτω παραστατικό. Στην περίπτωση που ο πελάτης επιλέξει Τιμολόγιο ,στο στάδιο του CheckOut ,θα χρειαστεί να συμπληρώσει στη φόρμα παραγγελίας, έξτρα πεδία που σχετίζονται , όπως Επωνυμία Επιχείρησης, ΑΦΜ, αρμόδια ΔΟΥ, Αντικείμενο Δραστηριότητας και Διεύθυνση.

ΤΙΜΟΛΟΓΙΟ - ΔΕΛΤΙΟ ΑΠΟΣΤΟΛΗΣ

Ημερομηνία: 11/11/2022 01:17
Κωδικός Πελάτη: 773e4147-e01c-4096-9502-4cba7766bf15
Επωνυμία: AE
Διεύθυνση: ΠΑΝΕΠΙΣΤΗΜΙΟΥ 59 ΑΘΗΝΑ
Τηλέφωνο: 000000000
Email: angelos@angelos.com
Μέθοδος Πληρωμής: Credit Card
Τύπος Παραστατικού: Τιμολόγιο
ΑΦΜ: 000000000
Δ.Ο.Υ: ΔΟΥ ΦΑΕΕ ΑΘΗΝΩΝ
Αντικείμενο Δραστηριότητας: ΔΙΑΝΙΚΗ ΠΩΛΗΣΗ ΒΙΒΛΙΩΝ

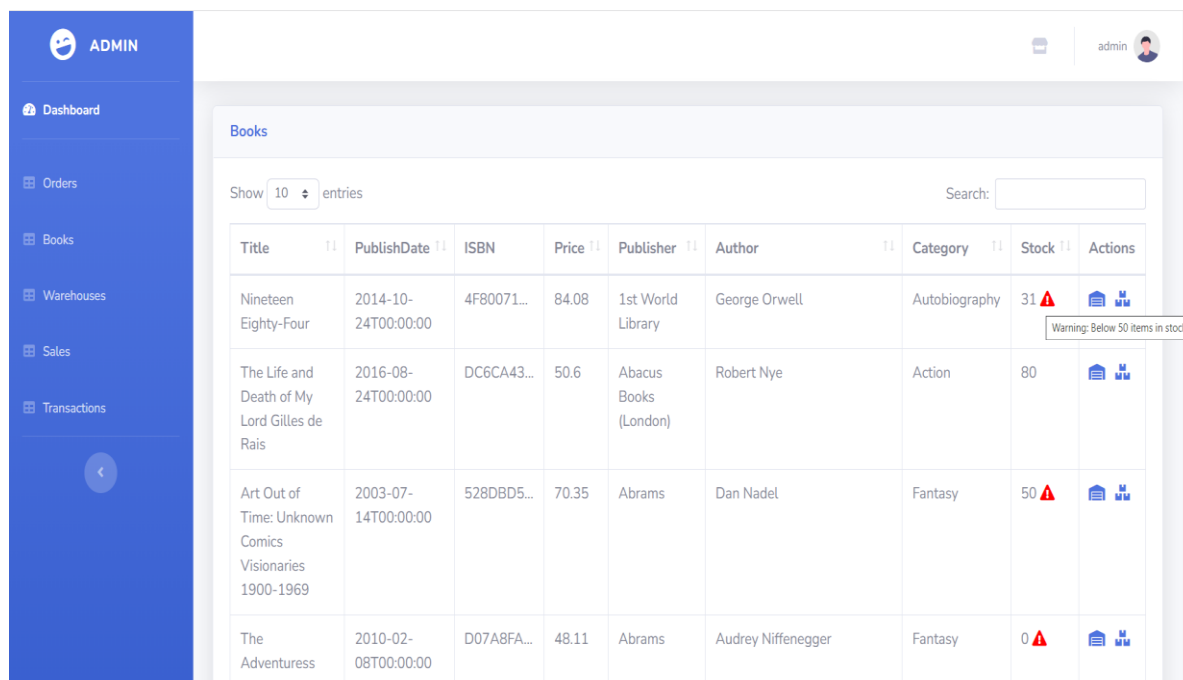
Επωνυμία: OnlineBookStore
Διεύθυνση: Δημ. Αντ.Βασιλείου 100
Ταχυδρομικός Κώδικας: 19344
Σειρά Παραστατικού: 20144
Αριθμός Παραστατικού: ΤΔΑΑ20144
Τηλέφωνο: (210) 12345678
Email: Info@Online-Book-Store.gr
ΑΦΜ: 123456

ΚΩΔ	ΠΕΡΙΓΡΑΦΗ ΕΙΔΟΥΣ	ΜΟΝ. ΜΕΤΡ.	ΠΟΣΟΠΗΤΑ	ΣΥΝΤ. ΦΠΑ	ΤΙΜΗ ΜΟΝ. ΧΩΡΙΣ ΦΠΑ	ΤΙΜΗ ΜΟΝ. ΜΕ ΦΠΑ	ΠΟΣΟΣΤΟ ΕΚΠΤΩΣΗΣ	ΚΑΘΑΡΗ ΑΞΙΑ	ΣΥΝΟΛΙΚΗ ΑΞΙΑ
528DBD5B-81A0-4345-B67D-45772F610D4E	Art Out of Time: Unknown Comics Visionaries 1900-1969	TEM	2,00	6%	66,13 €	70,35 €	0%	132,26 €	140,70 €

ΟΛΙΚΗ ΑΞΙΑ (Χωρίς ΦΠΑ)	ΣΥΝΟΛΙΚΟΣ ΦΠΑ	ΠΛΗΡΩΤΕΟ ΠΟΣΟ
132,26 €	8,44 €	140,70 €

Εικόνα 78Α

5.5.2 Απόθεμα Βιβλίων



Εικόνα 79

Η επιλογή **Books** στην αριστερή πλευρική μπάρα, εμφανίζει στον διαχειριστή, έναν αναλυτικό πίνακα με τις βασικές πληροφορίες του κάθε βιβλίου και το σχετικό απόθεμα αυτού. Κάθε γραμμή του πίνακα αποτελεί μία μοναδική εγγραφή βιβλίου και κάθε στήλη περιγράφει μία σημαντική πληροφορία γι’ αυτό. Οι βασικές πληροφορίες που εμφανίζονται στον πίνακα αυτό είναι ο τίτλος του βιβλίου, η ημερομηνία έκδοσης, ο Διεθνής Μοναδικός Αριθμός Βιβλίου ή αλλιώς ISBN, η τιμή, ο εκδοτικός οίκος, ο συγγραφέας, η κατηγορία και το απόθεμα.

Ο διαχειριστής μπορεί να αναζητήσει όποιο βιβλίο επιθυμεί, συμπληρώνοντας ως κριτήριο αναζήτησης την λέξη που τον ενδιαφέρει εντός του πεδίου **Search**. Συγκεκριμένα το πεδίο **Search** λειτουργεί ως φίλτρο. Όπως εμφανίζεται στην παρακάτω εικόνα, εισάγοντας τη λέξης **Action** στο πεδίο αναζήτηση, εμφανίζονται μόνο τα βιβλία της κατηγορίας Action.

Title	PublishDate	ISBN	Price	Publisher	Author	Category	Stock	Actions
The Life and Death of My Lord Gilles de Rais	2016-08-24T00:00:00	DC6CA43...	50.6	Abacus Books (London)	Robert Nye	Action	80	[Icons]
Madouc (Lyonesse #3)	2007-04-21T00:00:00	8199C75...	5.04	Ace Books	Jack Vance	Action	61	[Icons]
Providence of a Sparrow: Lessons from a Life Gone to the Birds	2018-01-05T00:00:00	EBEC5D2...	99.37	Anchor	Chris Chester	Action	0 ▲	[Icons]
Posing for Portrait	1997-10-31T00:00:00	DA31DB6...	50.43	Amherst Media	Jeff Smith	Action	0 ▲	[Icons]

Εικόνα 80

Αντίστοιχα αν εισάγουμε ως κριτήριο έναν εκδοτικό οίκο, όπως για παράδειγμα **Ace Books** στο πεδίο αναζήτησης, εμφανίζονται μόνο τα βιβλία του συγκεκριμένου εκδοτικού οίκου.

Title	PublishDate	ISBN	Price	Publisher	Author	Category	Stock	Actions
Accelerando	1997-04-11T00:00:00	14FDCF4...	77.44	Ace Books	Charles Stross	Picture book	0 ▲	[Icons]
Glasshouse	2010-01-26T00:00:00	E401802...	63.62	Ace Books	Charles Stross	Graphic novel	0 ▲	[Icons]
God Emperor of Dune (Dune Chronicles #4)	1997-10-22T00:00:00	F098C61...	74.68	Ace Books	Frank Herbert	True crime	0 ▲	[Icons]
Chapterhouse: Dune (Dune Chronicles #6)	1993-05-21T00:00:00	DCE2000...	19.84	Ace Books	Frank Herbert	Alternate history	0 ▲	[Icons]
Dune Messiah (Dune Chronicles #2)	2021-11-05T00:00:00	968257C...	74.62	Ace Books	Frank Herbert	Guide	0 ▲	[Icons]
Heretics of Dune (Dune Chronicles #5)	2010-11-07T00:00:00	E426291...	51.69	Ace Books	Frank Herbert	Paranormal romance	0 ▲	[Icons]

Εικόνα 81

Επιστρέφοντας στην επιλογή Books του χρήστη Admin, παρατηρούμε ότι κάποια βιβλία εμφανίζουν στην στήλη Stock ένα θαυμαστικό μέσα σε ένα κόκκινο τρίγωνο. Όταν εμφανίζεται δεξιά από το απόθεμα του βιβλίου, το κόκκινο τρίγωνο με το θαυμαστικό, σημαίνει ότι το συγκεκριμένο βιβλίο έχει χαμηλό απόθεμα. Η συγκεκριμένη ένδειξη εμφανίζεται όταν το πλήθος των βιβλίων είναι μικρότερο ή ίσο του 50. Στον ίδιο πίνακα εμφανίζεται άλλη μία στήλη με το όνομα **Actions**. Στη συγκεκριμένη στήλη για κάθε εγγραφή βιβλίου εμφανίζονται 2 κουμπιά. Όταν ο διαχειριστής τοποθετήσει το δείκτη του ποντικιού πάνω στο πρώτο εικονίδιο εμφανίζεται η ετικέτα **Stock** ενώ αν το τοποθετήσει στο δεύτερο εμφανίζεται η ετικέτα **Order**.



Εικόνα 82

Title	PublishDate	ISBN	Price	Publisher	Author	Category	Stock	Actions
Nineteen Eighty-Four	2014-10-24T00:00:00	4F80071...	84.08	1st World Library	George Orwell	Autobiography	31	Stock

Εικόνα 83

Παρατηρούμε στην παραπάνω εικόνα ότι το απόθεμα του βιβλίου με τίτλο « Nineteen Eighty-Four» είναι 31. Κάνοντας κλικ στο πρώτο κουμπί εμφανίζεται ιστοσελίδα η οποία μας εμφανίζει αναλυτικά σε ποιες αποθήκες υπάρχει το απόθεμα. Η αποθήκη της Αθήνας στην περιοχή της Ν. Ιωνίας έχει απόθεμα βιβλίου 11 και η αποθήκη της Θεσσαλονίκης στη περιοχή της Καλαμαριάς έχει απόθεμα 20 , όπως φαίνεται στην εικόνα 84.

Title	Stock	Warehouse	Warehouse Address	Warehouse City
Nineteen Eighty-Four	11	Warehouse 1	Aleku Panagouli 69, Nea Ionia 142 31	Athens
Nineteen Eighty-Four	20	Warehouse 3	Ellispontou 20, Kalamaria	Thessaloniki

Εικόνα 84

Το κουμπί **Add Stock** οδηγεί τον χρήστη σε μία φόρμα στην οποία μπορεί να αυξήσει το απόθεμα του συγκεκριμένου βιβλίου σε μία από τις 2 αποθήκες που διατηρεί το ηλεκτρονικό κατάστημα.

Εικόνα 85

Αναφορικά με το 2^ο κουμπί της στήλης Actions , ο διαχειριστής έχει τη δυνατότητα να συμπληρώσει μία φόρμα παραγγελίας προς τον προμηθευτή του συγκεκριμένου βιβλίου. Συγκεκριμένα μέσω αυτής της φόρμας μπορεί να παραγγείλει την ποσότητα του βιβλίου, την τιμή και σε ποια αποθήκη θα καταλήξει αυτή η παραγγελία.

Εικόνα 86

5.5.3 Διαθέσιμες Αποθήκες Βιβλιοπωλείου

Η τρίτη κατά σειρά επιλογή της αριστερής πλευρικής στήλης του Διαχειριστή είναι η επιλογή **Warehouses**. Το συγκεκριμένο κουμπί οδηγεί σε μία ιστοσελίδα , στην οποία εμφανίζεται ένας πίνακας με τις βασικές πληροφορίες των αποθηκών του ηλεκτρονικού καταστήματος.

Description	Address	Postcode	City	Phone	Actions
Warehouse 1	Aleku Panagouli 69, Nea Ionia 142 31	14231	Athens	2144168901	
Warehouse 3	Ellispontou 20, Kalamaria	55132	Thessaloniki	2144168901	

Εικόνα 87

Στην τελευταία στήλη του πίνακα εμφανίζεται στη στήλη **Actions** ,ένα κουμπί με εικονίδιο βιβλίου. Τοποθετώντας τον δείκτη του ποντικιού πάνω στο συγκεκριμένο εικονίδιο εμφανίζεται η ετικέτα *Books* .Κάνοντας κλικ στο κουμπί με το εικονίδιο βιβλίου, ο χρήστης οδηγείται σε ένα πίνακα παρόμοιο σε πληροφορία με εκείνον της επιλογής **Books** , με την μόνη διαφορά όμως ότι σε αυτόν τον πίνακα υπάρχουν βιβλία που βρίσκονται μόνο στην αποθήκη 1 . Αν ο διαχειριστής επιθυμεί να δει τα βιβλία που υπάρχουν στην αποθήκη 3 , θα επιλέξει το αντίστοιχο κουμπί της αποθήκης 3.

Title	PublishDate	ISBN	Price	Publisher	Author	Category	Stock	Actions
Nineteen Eighty-Four	2014-10-24T00:00:00	4F80071...	84.08	1st World Library	George Orwell	Autobiography	12	
The Life and Death of My Lord Gilles de Rais	2016-08-24T00:00:00	DC6CA43...	50.6	Abacus Books (London)	Robert Nye	Action	15	
Madouc (Lyonesse #3)	2007-04-21T00:00:00	8199C75...	5.04	Ace Books	Jack Vance	Action	61	
Dreamcatcher	2012-08-31T00:00:00	6490BFC...	1	Albin Michel	Stephen King/William Olivier Desmond/Maria Teresa Marengo	Textbook	31	
Les Fils des ténébres	2020-03-07T00:00:00	74085D0...	39.43	Albin Michel	Dan Simmons	Poetry	87	
The Subtle Knife (His Dark Materials #2)	2006-01-11T00:00:00	838F22D...	8.92	Alfred A. Knopf Books for Young Readers	Philip Pullman/Ian Beck	Biography	8	

Εικόνα 88

5.5.4 Πωλήσεις Βιβλίων

Η επιλογή **Sales** , όπως παρατηρούμε στην εικόνα 89 ,παρουσιάζει στο χρήστη-διαχειριστή σημαντική πληροφορία σχετικά με τις πωλήσεις. Ο παρακάτω πίνακας εμφανίζει αναλυτικά το πλήθος των πωλήσεων και το σύνολο των εσόδων που έχει φέρει το κάθε βιβλίο

ISBN	Title	Sales	Quantity
4F800713-C6B4-4C48-8B38-DB555B7E9F0	Nineteen Eighty-Four	2017.92 \$	24
DC6CA43C-D229-4DC4-9BA0-D49B58EA9FB5	The Life and Death of My Lord Gilles de Rais	860.2 \$	17
528DBD5B-81A0-4345-B67D-45772F610D4E	Art Out of Time: Unknown Comics Visionaries 1900-1969	914.55 \$	13
F098C610-5C17-43C3-AB04-49FD97601B01	God Emperor of Dune (Dune Chronicles #4)	170.01 \$	9
8199C75A-8B39-4EA2-97CB-62C473210825	Madouc (Lyonesse #3)	156.24 \$	31
1482B232-3096-4E6E-8E91-1CB1C22DC1CF	Unicorn Point (Apprentice Adept #6)	22.13 \$	1
6490BFCS-DBB2-48BC-B1D7-AC1A5F207C1E	Dreamcatcher	1 \$	1
740B5D03-D97E-4558-A606-45448E9D1552	Les Fils des ténèbres	78.86 \$	2
838F22D4-3109-496B-A449-13D30C0D936A	The Subtle Knife (His Dark Materials #2)	44.6 \$	5
A78A86AB-DEAF-4EA2-A877-00D808BAD66F	Homer's Daughter	156.69 \$	9

Εικόνα 89

Επιπρόσθετα, δίνεται η δυνατότητα στον χρήστη να αναζητήσει το πλήθος των πωλήσεων και το σύνολο των εσόδων σε μία συγκεκριμένη χρονική περίοδο ,συμπληρώνοντας τις αντίστοιχες ημερομηνίες που τον ενδιαφέρει.

ISBN	Title	Sales	Quantity
4F800713-C6B4-4C48-8B38-DB555B7E9F0	Nineteen Eighty-Four	420.4 \$	5
DC6CA43C-D229-4DC4-9BA0-D49B58EA9FB5	The Life and Death of My Lord Gilles de Rais	455.4 \$	9
8199C75A-8B39-4EA2-97CB-62C473210825	Madouc (Lyonesse #3)	105.84 \$	21
740B5D03-D97E-4558-A606-45448E9D1552	Les Fils des ténèbres	78.86 \$	2
838F22D4-3109-496B-A449-13D30C0D936A	The Subtle Knife (His Dark Materials #2)	44.6 \$	5
8A3DB5EF-3A83-4323-A034-91880AF3F003	House of Meetings	991.5 \$	10
05018C29-205C-477C-AA84-CA31E2C7AC4E	The Road	193.7 \$	2

Showing 1 to 7 of 7 entries (filtered from 171 total entries)

Εικόνα 90

5.5.5 Παραγγελίες βιβλίων σε Προμηθευτή

Η τελευταία λειτουργία που παρέχεται στον λίστα επιλογών του διαχειριστή ονομάζεται **Transactions**. Επιλέγοντας αυτή τη λειτουργία, ο διαχειριστής ανακατευθύνεται σε μία ιστοσελίδα, στην οποία εμφανίζεται ένας πίνακας με τις παραγγελίες του ηλεκτρονικού καταστήματος προς τους προμηθευτές των βιβλίων. Συγκεκριμένα η κάθε παραγγελία προμηθευτή, περιλαμβάνει τον όνομα του προμηθευτή, τον τίτλο του βιβλίου, την ποσότητα βιβλίων, την τιμή, την αποθήκη, το status και την ημερομηνία παραγγελίας.

The screenshot displays the 'Transactions' page in an admin dashboard. The page features a sidebar on the left with navigation options: Dashboard, Orders, Books, Warehouses, Sales, and Transactions. The main content area shows a table of transactions with the following columns: Supplier, Title, Quantity, Price \$, Warehouse, Status, and Date. The table contains 11 entries. At the top of the table, there is a search bar and a 'Show 25 entries' dropdown. At the bottom, there is a pagination control showing 'Showing 1 to 11 of 11 entries' and 'Previous 1 Next'.

Supplier	Title	Quantity	Price \$	Warehouse	Status	Date
Print Factory	Nineteen Eighty-Four	5	55	Warehouse 1	Confirmed	18/08/2022 20:38
Print Factory	Nineteen Eighty-Four	12	12.5	Warehouse 1	Confirmed	19/08/2022 20:32
Print Factory	Glasshouse	11	100	Warehouse 3	Pending	19/08/2022 20:34
Print Factory	Nineteen Eighty-Four	23	1000	Warehouse 3	Pending	20/08/2022 12:40
Print Factory	Nineteen Eighty-Four	12	1000	Warehouse 1	Pending	10/09/2022 16:35
Print Factory	The Life and Death of My Lord Gilles de Rais	3	34	Warehouse 1	Pending	30/09/2022 21:47
Print Factory	The Life and Death of My Lord Gilles de Rais	10	25	Warehouse 3	Pending	29/10/2022 18:06
Print Factory	Nineteen Eighty-Four	19	75	Warehouse 1	Pending	29/10/2022 18:23
Print Factory	Nineteen Eighty-Four	500	15	Warehouse 1	Pending	29/10/2022 18:31
Print Factory	Art Out of Time: Unknown Comics Visionaries 1900-1969	10	348	Warehouse 1	Confirmed	11/11/2022 00:20
Print Factory	Art Out of Time: Unknown Comics Visionaries 1900-1969	50	650	Warehouse 3	Pending	11/11/2022 00:21

Εικόνα 91

Κεφάλαιο 6 – Συμπεράσματα & Μελλοντικές Βελτιώσεις-Επεκτάσεις

Ο σκοπός της παρούσας εργασίας ήταν ο σχεδιασμός και η υλοποίηση μιας διαδικτυακής εφαρμογής δημιουργίας καταστήματος (e-bookstore) με χρήση πλατφόρμας διαχείρισης παραγγελιών και αποθήκης. Στην εφαρμογή αυτή μετέχουν δύο ρόλοι χρηστών . Ο ένας αφορά τον χρήστη-πελάτη και ο άλλος τον χρήστη-διαχειριστή του καταστήματος.

Η ενασχόληση για τη δημιουργία αυτής της εφαρμογής μου έδωσε την ευκαιρία να αποκομίσω σημαντικές γνώσεις και εμπειρίες. Οι ανάγκες και οι απαιτήσεις της εργασίας με οδήγησαν, στην αναζήτηση, έρευνα και γνωριμία με σύγχρονες τεχνολογίες του διαδικτύου αλλά και στο συνδυασμό τους για την επίτευξη του καλύτερου δυνατού αποτελέσματος.

Η ενασχόληση με τα διάφορα μέρη που αποτελούν τη συγκεκριμένη εφαρμογή συνέβαλε στην απόκτηση πολλών νέων γνώσεων και ικανοτήτων και στη δημιουργία της επιθυμίας για απόκτηση ακόμα περισσότερων.

Στην επιστήμη των υπολογιστών, το Separation of Concerns (SoC) είναι μια αρχή σχεδιασμού που διαχειρίζεται την πολυπλοκότητα χωρίζοντας το σύστημα λογισμικού έτσι ώστε κάθε τμήμα να είναι υπεύθυνο για ένα ξεχωριστό πρόβλημα, ελαχιστοποιώντας την επικάλυψη των προβλημάτων όσο το δυνατόν περισσότερο. Βασισμένο στη σχεδιαστική αρχή διαχωρισμού προβλημάτων (Separation of Concerns), επιλέχθηκε η αρχιτεκτονική Μοντέλο-Παρουσίαση-Ελεγκτής (Model-View-Controller, MVC), η οποία αποτελεί τη δημοφιλέστερη υλοποίηση αυτής της αρχής. Για αυτό το λόγο , χρησιμοποιήθηκε το ASP.NET Core MVC , το οποίο παρέχει έναν τρόπο που βασίζεται σε μοτίβα για τη δημιουργία δυναμικών ιστοτόπων και επιτρέπει τον καθαρό διαχωρισμό των προβλημάτων.

Για τις ανάγκες της συγκεκριμένης διπλωματικής εργασίας, χρησιμοποιήθηκε το σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) Microsoft SQL Server .Ο Microsoft SQL Server είναι ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS) που υποστηρίζει μια μεγάλη ποικιλία εφαρμογών επεξεργασίας συναλλαγών, επιχειρηματικής ευφυΐας και ανάλυσης σε εταιρικά περιβάλλοντα πληροφορικής.

Το Bootstrap είναι ένα από τα πιο δημοφιλή εργαλεία για εύκολη, γρήγορη και φιλική ανάπτυξη ιστοσελίδων. Ο σχεδιασμός ιστοσελίδων στο ηλεκτρονικό εμπόριο πρέπει να είναι τέτοιος που να προσελκύει νέους πελάτες και να παρέχει εξαιρετικές εμπειρίες σε κάθε επισκέπτη. Πολλές εταιρείες επιλέγουν ένα ευέλικτο πλαίσιο σχεδίασης που τους επιτρέπει να τροποποιούν τμήματα του σχεδιασμού για να διατηρούν τους ιστότοπους φρέσκους και ενημερωμένους. Η ευελιξία του Bootstrap είναι γνωστή καθώς είναι μια από τις πιο δημοφιλείς πλατφόρμες και περιλαμβάνει πλαίσια CSS, HTML και JS για σχεδιασμό ιστοσελίδων. Αυτό το εργαλείο ανταποκρίνεται σε όλες τις συσκευές και παρέχει πολλά οφέλη για ευέλικτες ανάγκες σχεδίασης.

Το JQuery είναι η πιο δημοφιλής βιβλιοθήκη στον ιστό σήμερα. Χρησιμοποιείται από μεγάλες εταιρείες όπως Microsoft, IBM, Netflix. Είναι μια βιβλιοθήκη λειτουργιών JavaScript που διευκολύνουν τους προγραμματιστές ιστοσελίδων να κάνουν κοινές εργασίες, όπως ο χειρισμός της ιστοσελίδας, η απόκριση σε συμβάντα χρήστη, η λήψη δεδομένων από τους διακομιστές, η δημιουργία εφέ και κινούμενων εικόνων και πολλά άλλα. Το jQuery είναι μια γρήγορη, πλούσια σε χαρακτηριστικά και ελαφριά βιβλιοθήκη. Ο κύριος σκοπός της χρήσης του jQuery είναι να διευκολύνει πολύ τη χρήση της JavaScript σε μια σύγχρονη και έξυπνη ιστοσελίδα. Αναπτύχθηκε για να εξοικονομήσει χρόνο από τους προγραμματιστές μειώνοντας τον κώδικα. Η χρήση της ήταν κάτι παραπάνω από απαραίτητη για να γίνει χρηστική, διαδραστική και ευχάριστη η διαδικτυακή εφαρμογή. Η συνεισφορά της στο project είναι σημαντική καθώς μέσω αυτής δημιουργήθηκαν animation-εφέ-κίνηση σε html elements, χρησιμοποιήθηκαν διάφορα AJAX requests και τέλος υλοποιήθηκαν αλλαγές στις ιδιότητες της HTML και στο CSS του κώδικα βάσει των ενεργειών του χρήστη. Η JQuery, όπως και οι τεχνολογίες HTML, CSS και Bootstrap, συνθέτουν το User Interface (UI) της εφαρμογής και διαδραματίζουν καθοριστικό ρόλο τόσο στον τρόπο παρουσίασης και διαχείρισης των δεδομένων της εφαρμογής όσο και στην εμφάνιση των διαφόρων σελίδων της.

Μελλοντικές βελτιώσεις και επεκτάσεις

Μελετήθηκαν κάποιες βελτιώσεις–επεκτάσεις στην διαδικτυακή εφαρμογή του καταστήματος βιβλιοπωλείου οι οποίες παρατίθενται παρακάτω.

Θα μπορούσε να βελτιωθεί η λειτουργία «αναζήτηση» του χρήστη πελάτη, δηλαδή να γίνει πιο εξειδικευμένη με χρήση κριτηρίων όπως, τον τίτλο του βιβλίου, την τιμή βιβλίου, τον εκδοτικό οίκο και άλλων. Αυτό του επιτρέπει να ψάχνει με πολλαπλά φίλτρα και θα έχει ως συνέπεια το ηλεκτρονικό κατάστημα να γίνει πιο λειτουργικό, πιο γρήγορο και πιο φιλικό στο χρήστη.

Επιπρόσθετα, θα μπορούσε να συμπεριληφθεί μηχανισμός αξιολόγησης των βιβλίων και εισαγωγή κριτικών από τους πελάτες .

Μια πρόταση βελτίωσης της εφαρμογής, η οποία χρησιμοποιείται ευρέως από πολλά ηλεκτρονικά καταστήματα , είναι η προσθήκη live chat support , στο οποίο θα μπορούσαν οι πελάτες να επικοινωνούν με το κατάστημα σε πραγματικό χρόνο. Σύμφωνα με έρευνες, αυτός ο τρόπος επικοινωνίας ωφελεί την επιχείρηση σε σύγκριση με τους παραδοσιακούς τρόπους επικοινωνίας όπως το ηλεκτρονικό ταχυδρομείο και η τηλεφωνική υποστήριξη. Μια τέτοια προσθήκη θα μπορούσε να αυξήσει τις πωλήσεις, να κάνει ποιοτικότερη την εξυπηρέτηση και να βελτιώσει την αφοσίωση των πελατών.

Τέλος, ο χρήστης θα μπορούσε να εισέρχεται και να εγγράφεται στην εφαρμογή με τα στοιχεία λογαριασμού πλατφορμών κοινωνικής δικτύωσης (Facebook, Google+, Twitter κ.α.).

Παράρτημα

Πίνακες Βάσης εφαρμογής

1. Πίνακας *AspNetRoles*

Id	Name	NormalizedName	ConcurrencyStamp
54bb2d97-66a7-4389-a99b-39642eee26ad	customer	CUSTOMER	4e7ea195-7b2c-454e-ac86...
be86d92b-6948-4095-b2ce-d869d42a4f05	admin	ADMIN	7753d8d0-5e0e-41a1-92e3...
NULL	NULL	NULL	NULL

2. Πίνακας *AspNetUsers*

Id	UserName	NormalizedUserName	Email	NormalizedEmail	EmailCo...	PasswordHash	SecurityStamp	ConcurrencyStamp	PhoneNumber	PhoneN
139-627c8766637f	angelos@gmail.com	ANGELOS@GMAIL.COM	angelos@gmail.com	ANGELOS@GMAIL.COM	True	AQAAAAEAACcQAAAA...	NV6377BQ77ZTTJ...	768fa0ad-595f-4d7...	NULL	False
5734e07d-0e08-4feb-9186-8444b...	customer	CUSTOMER	customer@gmail.com	CUSTOMER@GMAIL.COM	True	AQAAAAEAACcQAAAA...	2YIMDTMWCJDW...	ae379f24-88f3-40e...	NULL	False
594e7935-3528-4a1f-928b-22d1f...	customer2@gmail.com	CUSTOMER2@GMAIL...	customer2@gmail.com	CUSTOMER2@GMAIL.COM	True	AQAAAAEAACcQAAAA...	2IOQDY3JNRO3V...	ddf74002-ff88-44e...	NULL	False
773c4147-e01c-4096-9502-4cba7...	angelos@angelos.com	ANGELOS@ANGELOS...	angelos@angelos.com	ANGELOS@ANGELOS.COM	True	AQAAAAEAACcQAAAA...	4O5PFKAWRAYOJ...	8ffa4876-d123-4b...	000000000	False
9f19d0fb-2b3e-4d6a-8f62-117b5...	customer4@gmail.com	CUSTOMER4@GMAIL...	customer4@gmail.com	CUSTOMER4@GMAIL.COM	True	AQAAAAEAACcQAAAA...	C3P5OHO7LK2YD...	1def8d29-47b5-4b...	NULL	False
b91a4572-67d9-4be9-bb63-7939...	customer5@gmail.com	CUSTOMER5@GMAIL...	customer5@gmail.com	CUSTOMER5@GMAIL.COM	True	AQAAAAEAACcQAAAA...	TEX3DCLHOH7LU...	5f232510-bf49-499...	NULL	False
bad9ae22-eb0a-4237-b29b-1360...	angelos@angelos.gr	ANGELOS@ANGELOS.GR	angelos@angelos.gr	ANGELOS@ANGELOS.GR	True	AQAAAAEAACcQAAAA...	TWYyQLHOYZ2F...	2b53916c-78fe-48c...	2100000000	False
c76d31c1-67b5-453a-b7af-bf584...	admin	ADMIN	admin@gmail.com	ADMIN@GMAIL.COM	True	AQAAAAEAACcQAAAA...	T6PATHVE7MGDL...	636eb6e0-5147-43...	NULL	False
ebac2b3f-830c-498d-9189-799d7...	mppl18041@unipi.gr	MPPL18041@UNIPI.GR	mppl18041@unipi.gr	MPPL18041@UNIPI.GR	True	AQAAAAEAACcQAAAA...	H3HQI2G3MZ5IM...	dc9f3c40-3b7f-47a...	+300000000000	False
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Πίνακας Authors

Id	Name	Address	Phone	URL	CreatedOn	ModifiedOn
1	J.K. Rowling/Mary GrandPré	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:41:...	19/7/2022 7:41:...
2	A.B. Yehoshua/Hillel Halkin	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
3	Abdul Rahman Munif/Peter ...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
4	Abolqasem Ferdowsi/Dick ...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
5	Adam Rex	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
6	Adam Smith	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
7	Aeschylus/Richmond Lattim...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
8	Aesop/Laura Harris/Laura Gt...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
9	Al Gore	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
10	Alan Lee/Ian McKellen	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
11	Alan Aldridge/The Beatles	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
12	Alan Brenner	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
13	Alan Hausman/Paul Tidman	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
14	Albert Camus	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
15	Albert Camus/Carol Cosman...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
16	Albert Camus/Catherine Ca...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
17	Albert Camus/Justin O'Brien	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
18	Albert Jack/Ann Page	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
19	Alcoholics Anonymous	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
20	Aldous Huxley/Huston Smit...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
21	Alec MacLellan	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
22	Alex C. Jones/Diane M. Hans...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
23	Alexander Hunter	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
24	Alexander McCall Smith	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
25	Alfred Hitchcock/Don Tothe	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
26	Alice Hoffman	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
27	Alice Medrich/Patricia Braba...	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
28	Alice Munro	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
29	Alistair Horne	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
30	Allen Ginsberg/Barry Miles	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...
31	Alma Flor Ada/Leslie Tryon	Aleku Panagouli 91, Nea Ionia 142 34	+302102752057	www.authorurl.com	19/7/2022 7:49:...	19/7/2022 7:49:...

4. Πίνακας BookOrders

BookId	OrderId	Quantity	UnitPrice
486	122	1	84,08
486	123	1	84,08
486	124	1	84,08
486	127	1	84,08
486	128	3	84,08
486	129	2	84,08
486	130	2	84,08
486	131	1	84,08
486	10131	2	84,08
486	10132	1	84,08
486	10133	3	84,08
486	10143	1	84,08
486	10144	1	84,08
486	10146	3	84,08
486	20143	1	84,08
487	122	1	50,60
487	123	1	50,60
487	124	1	50,60
487	125	2	50,60
487	126	1	50,60
487	127	1	50,60
487	10144	1	50,60
487	10146	8	50,60
487	20143	1	50,60
488	122	2	70,35

5. Πίνακας Books

Id	Image	Title	Descri...	Price	PublishD...	AuthorId	PublisherId	CategoryId	CreatedOn	ModifiedOn	PageCount	Rating	RatingCount	ISBN	Suppl
486	0x5249464612700...	Nineteen Eighty-Four	Lorem ips...	84,08	24/10/20...	6100	1129	4	20/7/2022 1...	20/7/2022 1...	387	4,00	215	4F800713-C...	1
487	0x5249464612700...	The Life and Death of My Lor...	Lorem ips...	50,60	24/8/201...	6507	192	1	20/7/2022 1...	20/7/2022 1...	323	4,00	65	DC6CA43C-...	1
488	0x5249464612700...	Art Out of Time: Unknown Co...	Lorem ips...	70,35	14/7/200...	1661	2012	21	20/7/2022 1...	20/7/2022 1...	320	4,00	226	528DBD5B-...	1
489	0x5249464612700...	The Adventuress	Lorem ips...	48,11	8/2/2010 ...	188	2012	21	20/7/2022 1...	20/7/2022 1...	144	3,00	1058	D07A8FA1-...	1
490	0x5249464612700...	Accelerando	Lorem ips...	77,44	11/4/199...	1320	987	33	20/7/2022 1...	20/7/2022 1...	415	4,00	16527	14FDCF40-...	1
491	0x5249464612700...	Glasshouse	Lorem ips...	63,62	26/1/201...	1320	987	23	20/7/2022 1...	20/7/2022 1...	335	4,00	8975	E4018026-...	0
492	0x5249464612700...	God Emperor of Dune (Dune ...	Lorem ips...	74,68	22/10/19...	3556	987	42	20/7/2022 1...	20/7/2022 1...	423	4,00	2785	F098C610-...	1
493	0x5249464612700...	Chapterhouse: Dune (Dune Ch...	Lorem ips...	19,84	21/5/199...	3556	987	3	20/7/2022 1...	20/7/2022 1...	436	4,00	38778	DCE2000D-...	1
494	0x5249464612700...	Dune Messiah (Dune Chronic...	Lorem ips...	74,62	5/11/202...	3556	987	18	20/7/2022 1...	20/7/2022 1...	331	4,00	97494	968257C2-...	1
495	0x5249464612700...	Heretics of Dune (Dune Chron...	Lorem ips...	51,69	7/11/201...	3556	987	31	20/7/2022 1...	20/7/2022 1...	471	4,00	45388	E4262914-...	1
496	0x5249464612700...	Deerskin	Lorem ips...	35,31	12/6/200...	3639	987	22	20/7/2022 1...	20/7/2022 1...	384	4,00	17485	7C27B488-...	1
497	0x5249464612700...	Elic of Melniboné (The Elic S...	Lorem ips...	17,92	14/7/201...	6390	987	10	20/7/2022 1...	20/7/2022 1...	181	4,00	20559	7067F732-...	1
498	0x5249464612700...	The Cat Who Walks Through ...	Lorem ips...	51,18	15/7/199...	4550	987	9	20/7/2022 1...	20/7/2022 1...	388	4,00	19427	4A0C8EE1-...	1
499	0x5249464612700...	To Sail Beyond the Sunset	Lorem ips...	78,58	5/12/201...	4550	987	20	20/7/2022 1...	20/7/2022 1...	434	4,00	10293	18EBBC5A-...	1
500	0x5249464612700...	A Separate War and Other Sto...	Lorem ips...	66,23	1/3/2007 ...	3760	987	38	20/7/2022 1...	20/7/2022 1...	270	4,00	342	819E2521-...	1
501	0x5249464612700...	Madouc (Lyonesse #3)	Lorem ips...	5,04	21/4/200...	1811	987	1	20/7/2022 1...	20/7/2022 1...	426	4,00	1825	8199C75A-...	1
502	0x5249464612700...	Forever Peace (The Forever W...	Lorem ips...	4,83	20/7/200...	6222	987	2	20/7/2022 1...	20/7/2022 1...	351	4,00	16338	72D7CFC0-...	1
503	0x5249464612700...	Harrowing the Dragon	Lorem ips...	0,28	7/7/2009 ...	6433	987	34	20/7/2022 1...	20/7/2022 1...	308	4,00	1602	F5C1B675-...	1
504	0x5249464612700...	Cygnets (Cygnets #1-2)	Lorem ips...	27,33	13/3/199...	6433	987	27	20/7/2022 1...	20/7/2022 1...	406	4,00	543	27EB1865-...	1
505	0x5249464612700...	Solstice Wood (Winter Rose ...	Lorem ips...	24,49	15/9/201...	6433	987	6	20/7/2022 1...	20/7/2022 1...	278	4,00	1715	DEA8C6B3-...	1
506	0x5249464612700...	Od Magic	Lorem ips...	6,63	23/10/20...	6433	987	27	20/7/2022 1...	20/7/2022 1...	315	4,00	3140	8403C8ED-...	1
507	0x5249464612700...	Masterpieces: The Best Scienc...	Lorem ips...	67,26	29/9/200...	3000	987	21	20/7/2022 1...	20/7/2022 1...	422	4,00	1202	A2428208-...	1
508	0x5249464612700...	Vision of Tarot (Tarot #2)	Lorem ips...	42,59	8/3/2000 ...	6466	987	14	20/7/2022 1...	20/7/2022 1...	260	3,00	1485	80D9075A-...	1
509	0x5249464612700...	Unicom Point (Apprentice Ad...	Lorem ips...	22,13	26/5/199...	6466	987	26	20/7/2022 1...	20/7/2022 1...	337	4,00	4820	1482B232-...	1
510	0x5249464612700...	The God in the Moon (Age of ...	Lorem ips...	27,71	10/12/20...	4935	987	39	20/7/2022 1...	20/7/2022 1...	297	4,00	59	88F5844E-...	1
511	0x5249464612700...	Count Zero (Sprawl #2)	Lorem ips...	83,54	18/1/201...	4224	987	33	20/7/2022 1...	20/7/2022 1...	308	4,00	37360	B6F3C707-...	1
512	0x5249464612700...	Dayworld (Dayworld #1)	Lorem ips...	51,04	25/10/20...	4919	987	36	20/7/2022 1...	20/7/2022 1...	258	4,00	1315	A3BFC695-...	1
513	0x5249464612700...	The Other Wind (Earthsea Cyc...	Lorem ips...	54,05	25/12/20...	5849	987	40	20/7/2022 1...	20/7/2022 1...	211	4,00	14393	EE9B54C3-...	1
514	0x5249464612700...	From the Earth to the Moon (...	Lorem ips...	70,43	26/4/199...	1887	940	20	20/7/2022 1...	20/7/2022 1...	136	4,00	17325	6B1AE64E-...	1
515	0x5249464612700...	L'Aube du désert	Lorem ips...	98,43	21/8/201...	5381	1937	27	20/7/2022 1...	20/7/2022 1...	288	4,00	1	D7C99AB8-...	1
516	0x5249464612700...	Deemster	Lorem ips...	1,00	21/8/201...	4014	1027	40	20/7/2022 1...	20/7/2022 1...	688	4,00	126212	6A096C65-...	1

6. Πίνακας BookTransactions

BookId	TransactionId	Quantity
486	1	5
486	2	12
486	4	23
486	5	12
486	8	19
486	9	500
487	6	3
487	7	10
488	10007	10
488	10008	50
491	3	11
491	10009	19
NULL	NULL	NULL

7. BookWarehouses

BookId	WarehouseId	Stock
486	1	11
486	2	20
501	1	73
509	2	63
516	1	31
518	2	70
519	1	89
524	1	13
536	2	76
572	2	78
580	1	27
599	2	0
624	2	0
635	2	69
638	1	23
666	2	36
667	2	17
668	1	0
669	2	33
671	2	25
696	2	0
708	1	0
717	1	0
725	1	86
732	1	53
745	2	23
765	1	77
784	2	8
801	2	14
809	2	0
840	2	44

8. Πίνακας Suppliers

Id	Name	Address	Phone	URL	CreatedOn	ModifiedOn
1	Print Factory	Pindou 12	69712345567	printfactory@gmail.com	18/8/2022 7:58:...	18/8/2022 7:58:...
NULL	NULL	NULL	NULL	NULL	NULL	NULL

9. Πίνακας Categories

Id	Name	Description	CreatedOn	ModifiedOn
1	Action	Action	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
2	Art	Art	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
3	Alternate history	Alternate history	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
4	Autobiography	Autobiography	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
5	Anthology	Anthology	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
6	Biography	Biography	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
7	Business	Business	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
8	Childrens	Childrens	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
9	Crafts	Crafts	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
10	Classic	Classic	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
11	Cookbook	Cookbook	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
12	Comics	Comics	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
13	Diary	Diary	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
14	Dictionary	Dictionary	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
15	Crime	Crime	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
16	Encyclopedia	Encyclopedia	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
17	Drama	Drama	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
18	Guide	Guide	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
19	Fairytale	Fairytale	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
20	Health	Health	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
21	Fantasy	Fantasy	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
22	History	History	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
23	Graphic novel	Graphic novel	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
24	Home and gard...	Home and gard...	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
25	Historical fiction	Historical fiction	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
26	Humor	Humor	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
27	Horror	Horror	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
28	Journal	Journal	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
29	Mystery	Mystery	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
30	Math	Math	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ
31	Paranormal rom...	Paranormal rom...	19/7/2022 11:01:45 μμ	19/7/2022 11:01:45 μμ

10. Πίνακας Warehouses

Id	Phone	Address	Postcode	City	CreatedOn	ModifiedOn	Description
1	2144168901	Alekou Panagouli 69, Nea Ionia 142 31	14231	Athens	19/7/2022 11:4...	1/1/0001 12:00:...	Warehouse 1
2	2144168901	Ellispontou 20, Kalamaria	55132	Thessaloniki	19/7/2022 11:4...	1/1/0001 12:00:...	Warehouse 3
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

11. Πίνακας Publishers

Id	Name	Address	Phone	URL	CreatedOn	ModifiedOn
1	Dutton Books	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
2	Boosey & Hawkes Inc	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
3	Firebird Books	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
4	Mira (Mills and Boon)	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
5	Dodd Mead; 1st edition (September 1976)	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
6	Legend	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
7	Viking UK	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
8	Yosemite Conservancy	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
9	Tor Classics	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
10	Franklin Watts	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
11	NAL Books	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
12	Plexus Publishing Ltd	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
13	Sterling Publishing Co. Inc.	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
14	New York : Penguin Books	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
15	Cemetery Dance Publications	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
16	Orion Publishing Co	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
17	Allyn & Bacon	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
18	Vintage International	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
19	Fleming H Revell Co	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
20	Suhrkamp	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
21	A & C Black (Childrens books)	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
22	Bloomsbury Paperbacks	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
23	No Exit Press	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
24	Midnight Ink	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
25	University of Arkansas Press	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
26	Carl Hanser	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
27	Bureau of Public Secrets	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
28	Simon & Schuster Simon Pulse	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
29	Vintage Books a division of Random House	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
30	Black Dog & Leventhal Publishers	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...
31	University of Oklahoma Press	3is Septemvriou 1, Athina 104 31	2144168901	https://www.publisher.gr/	19/7/2022 11:4...	19/7/2022 11:4...

12. Πίνακας Transactions

Id	Price	WarehouseId	CreatedOn	ModifiedOn	Status
1	55,00	1	18/8/2022 8:38:33 ...	18/8/2022 8:38:...	1
2	12,50	1	19/8/2022 8:32:35 ...	19/8/2022 8:32:...	1
3	100,00	2	19/8/2022 8:34:17 ...	19/8/2022 8:34:...	0
4	1000,00	2	20/8/2022 12:40:4...	20/8/2022 12:4...	0
5	1000,00	1	10/9/2022 4:35:25 ...	10/9/2022 4:35:...	0
6	34,00	1	30/9/2022 9:47:31 ...	30/9/2022 9:47:...	0
7	25,00	2	29/10/2022 6:06:4...	29/10/2022 6:0...	0
8	75,00	1	29/10/2022 6:23:3...	29/10/2022 6:2...	0
9	15,00	1	29/10/2022 6:31:1...	29/10/2022 6:3...	0
10007	348,00	1	11/11/2022 12:20:...	11/11/2022 12:...	1
10008	650,00	2	11/11/2022 12:21:...	11/11/2022 12:...	0
10009	148,00	1	13/11/2022 7:27:5...	13/11/2022 7:2...	0
*	NULL	NULL	NULL	NULL	NULL

13. Πίνακας Orders

Id	ToCity	ToAddress	ToPo...	ShippingD...	Custom...	CreatedOn	ModifiedOn	Re...	Status	ToFirst...	ToLast...	Pa...	ToCompanyAd...	ToComp...	ToCompan...	ToEmail	ToInd...	ToPhone	ToTaxAut...	ToTaxNumber
124	Moschat...	pindou 134	183 45	8/19/2022 ...	1	17/8/2022 ...	17/8/2022 6...	1	0	Christos	Alexakis	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
125	Moschat...	pindou 134	183 45	8/19/2022 ...	1	17/8/2022 ...	17/8/2022 6...	0	0	sss	Alexakis	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
126	Moschat...	pindou 134	183 45	8/19/2022 ...	1	17/8/2022 ...	17/8/2022 6...	1	0	Christos	Alexakis	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
127	Moschat...	pindou 134	183 45	8/19/2022 ...	1	17/8/2022 ...	17/8/2022 6...	1	0	Christos	Alexakis	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
128	Athens	12312 main Str	123456	8/22/2022 ...	2	20/8/2022 ...	20/8/2022 1...	0	0	Joe	Doe	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
129	Athens	123 Pi Str	12345	8/22/2022 ...	2	20/8/2022 ...	20/8/2022 1...	0	1	Joe	Doe	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
130	athens	1234 main str	1234	12/09/2022...	3	10/9/2022 ...	10/9/2022 4...	0	1	test	test	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
131	athens	1234 main str	1234	12/09/2022...	3	10/9/2022 ...	10/9/2022 4...	1	1	test	test	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10130	pireaus	labraki 77	18539	02/10/2022...	10003	30/9/2022 ...	30/9/2022 9...	1	1	a	aaa	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10131	pireaus	labraki 77	18539	02/10/2022...	10003	30/9/2022 ...	30/9/2022 9...	1	1	ang	ang	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10132	pireaus	labraki 77	18539	02/10/2022...	10003	30/9/2022 ...	30/9/2022 9...	1	1	aaaa	aaa	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10133	pireaus	labraki 77	18539	02/10/2022...	10003	30/9/2022 ...	30/9/2022 9...	1	1	aaa	aaa	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10134	pireaus	labraki 77	18539	02/10/2022...	10003	30/9/2022 ...	30/9/2022 9...	1	1	aaaa	aaaa	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10135	pireaus	labraki 77	18539	02/10/2022...	10003	30/9/2022 ...	30/9/2022 1...	1	1	aaaa	aaa	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10136	Pireaus	Akth Moutsopoulou 180	18531	11/10/2022...	10004	9/10/2022 ...	9/10/2022 6...	0	1	Angelos	Angelou	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10137	Pireaus	Akth Moutsopoulou 180	18531	11/10/2022...	10004	9/10/2022 ...	9/10/2022 7...	0	1	Angelos	Angelou	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10138	Pireaus	Akth Moutsopoulou 180	18531	15/10/2022...	10004	13/10/2022...	13/10/2022 ...	1	1	Angelos	Angelou	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10139	Pireaus	Akth Moutsopoulou 180	18539	15/10/2022...	10004	13/10/2022...	13/10/2022 ...	0	1	Angelos	Angelou	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10140	Pireaus	Akth Moutsopoulou 180	18531	15/10/2022...	10004	13/10/2022...	13/10/2022 ...	0	1	Angelos	Angelou	0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
10141	Athens	Panepisthmiou 22	18531	18/10/2022...	10005	16/10/2022...	16/10/2022 ...	1	1	Angelos	Angelos	0	PANEPITHMIOY ...	ANGELOS...	210-XXXX...	angelos@angel...	PARAD...	2103333...	DOU A. AT...	999999999
10142	Athens	Panepisthmiou 22	18531	18/10/2022...	10005	16/10/2022...	16/10/2022 ...	0	1	Angelos	Angelo...	2	NULL	NULL	NULL	angelos@angel...	NULL	2103333...	NULL	NULL
10143	Pireaus	Plateia Korah Peiraias	18533	25/10/2022...	10005	23/10/2022...	23/10/2022 ...	0	1	Angelos	Angelou	2	NULL	NULL	NULL	angelos@angel...	NULL	2310000...	NULL	NULL
10144	ΑΘΗΝΑ	ΠΑΝΕΠΙΣΤΗΜΙΟΥ 59, ...	105 64	31/10/2022...	10005	29/10/2022...	29/10/2022 ...	0	1	ΑΓΓΕΛΟΣ	ΑΓΓΕΛ...	0	NULL	NULL	NULL	angelos@angel...	NULL	2100000...	NULL	NULL
10145	ΠΕΙΡΑΙΑΣ	Εθνάρχου Μακαρίου 3...	18547	31/10/2022...	10006	29/10/2022...	29/10/2022 ...	0	1	ΑΓΓΕΛΟΣ	ΛΑΒΔΑΣ	0	NULL	NULL	NULL	mpp118041@uni...	NULL	+300000...	NULL	NULL
10146	ΠΕΙΡΑΙΑΣ	Εθνάρχου Μακαρίου 3...	18547	31/10/2022...	10006	29/10/2022...	29/10/2022 ...	0	1	ΑΓΓΕΛΟΣ	ΛΑΒΔΑΣ	1	NULL	NULL	NULL	mpp118041@uni...	NULL	+300000...	NULL	NULL
20143	ΠΕΙΡΑΙΑΣ	ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ...	18534	05/11/2022...	20006	3/11/2022 ...	3/11/2022 1...	0	1	ΑΓΓΕΛΟΣ	ΛΑΒΔΑΣ	2	NULL	NULL	NULL	angelos@angel...	NULL	0000000...	NULL	NULL
20144	ΠΕΙΡΑΙΑΣ	ΚΑΡΑΟΛΗ ΔΗΜΗΤΡΙΟΥ...	18533	13/11/2022...	20006	11/11/2022...	11/11/2022 ...	1	1	ΑΓΓΕΛΟΣ	ΛΑΒΔΑΣ	0	ΠΑΝΕΠΙΣΤΗΜΙΟ... AE	000000000	000000000	angelos@angel...	ΛΙΑΝΙΚ...	0000000...	ΔΟΥ ΦΑΕ...	000000000
✪	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	ΠΑΝΕΠΙΣΤΗΜΙΟΥ 59 ΑΘΗΝΑ	NULL	NULL	NULL	NULL	NULL

14. Πίνακας Customers

Id	AspnetId	Name	Phone	PostCode	Email	City	CreatedOn	ModifiedOn	ApplicationU...	Address
1	5734e07d-0e08-4feb-9186-8444b7ca9881	John Doe	+302102752057	121 31	electroholic@gmail.com	Athens	19/7/2022 11:1...	19/7/2022 11:11:54 μμ	NULL	Μακρυγιάννη 22
2	594e7935-3528-4a1f-928b-22d1fd5032b3	customer2@gmail.com	12345678	12345	customer2@gmail.com	Athens	19/8/2022 9:29...	19/8/2022 9:29:52 μμ	NULL	12345
3	9f19d0fb-2b3e-4d6a-8f62-117b5d2272c8	customer4@gmail.com	1234	1234	customer4@gmail.com	1234	10/9/2022 4:03...	10/9/2022 4:03:55 μμ	NULL	123 pap
10003	b91a4572-67d9-4be9-bb63-79395819b46a	customer5@gmail.com	123456	18539	customer5@gmail.com	pireaus	30/9/2022 8:56...	30/9/2022 8:56:52 μμ	NULL	Lambraki 77
10004	20fb6fb3-5d8e-42fb-8139-627c8766637f	angelos@gmail.com	210210210	18531	angelos@gmail.com	Peiraias	9/10/2022 6:55...	9/10/2022 6:55:55 μμ	NULL	Akth Moutsopoulou 180
10005	bad9ae22-eb0a-4237-b29b-1360a80f6522	Angelos	2100000000	10564	angelos@angelos.gr	ΑΘΗΝΑ	16/10/2022 7:5...	16/10/2022 7:57:35 μμ	NULL	ΠΑΝΕΠΙΣΤΗΜΙΟΥ 59, ΑΘΗΝΑ
10006	eba2b3f-830c-498d-9189-799d71192728	ΑΓΓΕΛΟΣ	+300000000000	18547	mpp118041@unipi.gr	ΠΕΙΡΑΙΑΣ	29/10/2022 10:...	29/10/2022 10:49:46 μμ	NULL	Εθνάρχου Μακαρίου 3 Πειραιά
20006	773c4147-e01c-4096-9502-4cba7766bf15	ΑΓΓΕΛΟΣ ΛΑΒΔΑΣ	000000000	18534	angelos@angelos.com	ΠΕΙΡΑΙΑΣ	2/11/2022 10:0...	2/11/2022 10:03:18 μμ	NULL	Καραολή και Δημητρίου
✪	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

ΒΙΒΛΙΟΓΡΑΦΙΑ

1. **Asp.Net , Asp.Net Core and Entity Framework** ,

<https://learn.microsoft.com/en-us/ef/core/>

<https://learn.microsoft.com/el-gr/ef/ef6/>

<https://learn.microsoft.com/en-us/aspnet/overview>

<https://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>

<https://www.simplilearn.com/tutorials/asp-dot-net-tutorial/entity-framework-in-c-sharp>

<https://www.partech.nl/nl/publicaties/2020/11/introduction-to-entity-framework>

<https://www.c-sharpcorner.com/UploadFile/ff2f08/identifying-entity-framework-development-approaches/>

<https://www.infoworld.com/article/2979726/explore-the-different-approaches-to-model-entities-in-entity-framework.html>

<https://www.c-sharpcorner.com/article/a-brief-introduction-to-asp-net-core/>

<https://www.dotnettricks.com/learn/aspnetcore/entity-framework-core-crud-operations>

<https://www.interviewbit.com/blog/net-core-vs-net-framework/>

2. **Difference Between .NET Framework and ASP.NET Framework** ,

<https://www.geeksforgeeks.org/difference-between-net-and-asp-net-framework/>

<https://www.geeksforgeeks.org/differences-between-net-core-and-net-framework/>

<https://pediaa.com/what-is-the-difference-between-net-and-asp-net/>

3. **Razor** ,

https://www.w3schools.com/asp/razor_syntax.asp

4. **MVC (Model , View, Controller) ,**

<https://www.interviewbit.com/blog/mvc-architecture/>

<https://www.geeksforgeeks.org/benefit-of-using-mvc/>

<https://thebhwwgroup.com/blog/2013/10/asp-net-mvc-principles-part-3>

<https://thebhwwgroup.com/blog/2013/10/asp-net-mvc-principles-part-2>

<https://developer.mozilla.org/en-US/docs/Glossary/MVC>

<https://www.c-sharpcorner.com/article/managing-data-with-viewmodel-in-asp-net-mvc/>

<https://www.c-sharpcorner.com/UploadFile/8a67c0/prefer-to-use-viewmodel-in-Asp-Net-mvc/>

<https://developer.mozilla.org/en-US/docs/Glossary/MVC>

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-controller?view=aspnetcore-3.1&tabs=visual-studio>

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-view?view=aspnetcore-3.1&tabs=visual-studio>

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/adding-model?view=aspnetcore-3.1&tabs=visual-studio>

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/working-with-sql?view=aspnetcore-3.1&tabs=visual-studio>

5. **jQuery JavaScript Library,**

<https://jquery.com/>

<https://www.educative.io/blog/what-is-jquery-javascript-library>

<https://medium.com/sololearn/write-less-and-do-more-with-jquery-82fcae9c1955>

<http://net-informations.com/jq/default.htm>

6. **Bootstrap and CSS,**

<https://el.strephonsays.com/difference-between-bootstrap-and-css>

<https://www.clarity-ventures.com/blog/benefits-of-using-bootstrap-for-web-design>

<https://blog.hubspot.com/website/css-tutorial>

7. LINQ ,

<https://www.javatpoint.com/linq>

<https://www.geeksforgeeks.org/what-is-query-in-linq/>

8. Javascript

<https://www.lighthouse labs.ca/en/blog/what-is-javascript-used-for>

<https://www.hackreactor.com/blog/what-is-javascript-used-for>

9. C#

<https://learn.microsoft.com/en-us/dotnet/csharp/>

<https://www.javatpoint.com/csharp-history>

<https://www.quora.com/What-type-of-applications-can-be-created-in-C>

10. Microsoft Sql Server Management Studio

<https://learn.microsoft.com/en-us/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-ver16>

11. Online Learning Platforms

<https://www.udemy.com/>

12. UML

<http://archive.eclass.uth.gr/eclass/modules/document/file.php/MHX206/Slides/UMLReview.pdf>

http://edu.eap.gr/pli/pli24_old/B-tomos/Parousiaseis/P03.pdf

13. Visual Studio

https://el.wikipedia.org/wiki/Microsoft_Visual_Studio