



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	«Διαδραστικό Εργαλείο Οπτικοποίησης Μουσικών Στατιστικών με το Spotify API» “Interactive Music Stats Visualization Tool Using the Spotify API”
Ονοματεπώνυμο Φοιτητή	Νικόλαος Γράβος
Πατρώνυμο	Γεώργιος
Αριθμός Μητρώου	Π21021
Επιβλέπων	Ευθύμιος Αλέπης, Καθηγητής

Ημερομηνία Παράδοσης:

Ιούνιος 2025

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Copying, storing and distributing this work, in whole or in part, for commercial purposes is prohibited. Reproduction, storage and distribution for non-profit, educational or research purposes is permitted, provided that the source is acknowledged and this message is retained.

The opinions and conclusions contained in this document are those of the author alone and do not represent the official position of the University of Piraeus.

As the author of this paper, I declare that this paper is not a product of plagiarism and does not contain material from unattributed sources.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ειλικρινείς μου ευχαριστίες στον επιβλέποντα καθηγητή κ. Ευθύμιο Αλέπη για την καθοδήγηση, την υποστήριξη και τις χρήσιμες παρατηρήσεις καθ' όλη τη διάρκεια της εκπόνησης της παρούσας εργασίας.

Επίσης, ευχαριστώ θερμά την οικογένεια και τους φίλους μου για την ηθική στήριξη, την ενθάρρυνση και την πολύτιμη παρουσία τους σε όλη αυτή τη διαδρομή. Δεν θα τα κατάφερνα χωρίς αυτούς.

Περίληψη

Στην εποχή της ραγδαίας εξέλιξης των τεχνολογιών εικονικής και επαυξημένης πραγματικότητας, καθώς και της εκθετικής ανόδου της βιομηχανίας των βιντεοπαιχνιδιών, οι άνθρωποι αναζητούν νέους, δημιουργικούς τρόπους για να εκφράσουν την προσωπικότητα και τα ενδιαφέροντά τους σε διαδικτυακά και εικονικά περιβάλλοντα. Η μουσική αποτελεί μία από τις πιο άμεσες και αυθεντικές μορφές έκφρασης των προσωπικών προτιμήσεων, των συναισθημάτων και της ταυτότητας του κάθε ατόμου.

Η παρούσα εργασία επιχειρεί να γεφυρώσει τον δημιουργικό σχεδιασμό με την τεχνική υλοποίηση, προσφέροντας μια καινοτόμα, διαδραστική και πλήρως προσωποποιημένη εμπειρία χρήστη. Μέσω της οπτικοποίησης των μουσικών του προτιμήσεων — όπως τα αγαπημένα τραγούδια, οι κορυφαίοι καλλιτέχνες και τα κυρίαρχα μουσικά είδη — ο χρήστης καλείται να ανακαλύψει ένα μοναδικό ψηφιακό «δωμάτιο», το οποίο αντικατοπτρίζει την αισθητική και τον μουσικό του χαρακτήρα. Η υλοποίηση βασίστηκε στην αξιοποίηση του **Spotify Web API** για την ανάκτηση προσωποποιημένων δεδομένων μουσικής, ενώ η ανάπτυξη της εφαρμογής πραγματοποιήθηκε με χρήση της γλώσσας **C#** και του **ASP.NET framework**.

Λέξεις Κλειδιά: Μουσική, Προσωποποίηση, Οπτικοποίηση, Spotify API, Εμπειρία Χρήστη, ASP.NET

Abstract

In an era of rapid advancements in virtual and augmented reality technologies, as well as the exponential growth of the video game industry, people are seeking new, creative ways to express their personality and interests within online and virtual environments. Music is one of the most direct and authentic forms of expressing personal preferences, emotions, and individual identity.

This thesis attempts to bridge creative design with technical implementation, offering an innovative, interactive, and fully personalized user experience. Through the visualization of their musical preferences — such as favorite songs, top artists, and dominant music genres — the user is invited to discover a unique digital "room" that reflects their aesthetic and musical identity. The implementation was based on utilizing the **Spotify Web API** to retrieve personalized music data, while the application was developed using the **C#** language and the **ASP.NET framework**.

Key Words: Music, Personalization, Visualization, Spotify API, User Experience, ASP.NET

ΠΕΡΙΕΧΟΜΕΝΑ

Copyright ©	i
Ευχαριστίες	ii
Περίληψη	iii
Abstract.....	iii
Εισαγωγή.....	1
1. Ανάλυση Δεδομένων.....	2
1.1 Αναζήτηση Στατιστικών	2
1.2 Πίνακας Κατηγοριών	3
2. Σχεδίαση 3D Δωματίων	4
2.1 Επεξήγηση Λογισμικού.....	4
2.2 Σχεδιασμός Βασικού Δωματίου	4
2.3 Σχεδιασμός Θεματικών Δωματίων.....	6
3. Ανάπτυξη Backend	7
3.1 Επεξήγηση Αρχιτεκτονικής	7
3.2 Δημιουργία Εφαρμογής Spotify.....	7
3.3 Ροή Εξουσιοδότησης Χρήστη με το Spotify API.....	8
3.3.1 Σελίδα Authorize	9
3.3.2 Σελίδα Callback	10
3.4 Αρχιτεκτονική και Λειτουργικότητα του Συστήματος	12
3.4.1 Σύνοψη Λειτουργιών της Σελίδας Room.....	12
3.4.2 Μέθοδος GetUsername.....	13
3.4.3 Μέθοδος GetTopTracks.....	14
3.4.4 Μέθοδος GetTopArtists	15
3.4.5 Μέθοδος GetTrackGenres	16
3.4.6 Μέθοδος GetSpotifyPreviewUrlAsync	17
3.4.7 Μέθοδος OnGetAsync.....	18
4. Ανάπτυξη Frontend	20
4.1 Επιλογή Αισθητικής.....	20
4.2 Σελίδα Connect	21
4.3 Σελίδα Room	22
5. Εγχειρίδιο Χρήσης	24

5.1. Σύνδεση στην εφαρμογή	24
5.2. Περιήγηση στην εφαρμογή	26
5.3 Χώρος Δωματίου	28
5.4 Χώρος κατοικίδιου	28
5.5 Χώρος Στατιστικών	29
5.6 Χώρος Πληροφοριών	29
6. Συμπεράσματα & Μελλοντικές Επεκτάσεις	30
7. Βιβλιογραφία	31

Εισαγωγή

Η έμπνευση για την παρούσα εργασία προήλθε από σύγχρονες ψηφιακές εμπειρίες που συνδυάζουν την τεχνολογία με την προσωπική έκφραση, όπως το **My Internet Bedroom** του δημιουργικού στούντιο **MadeOnVerse**, καθώς και από την ετήσια ανασκόπηση **Spotify Wrapped**, η οποία προσφέρει στους χρήστες μια προσωποποιημένη αναδρομή στις μουσικές τους προτιμήσεις.

Το *Spotify Wrapped* έχει πλέον καθιερωθεί ως ένα παγκόσμιο πολιτισμικό φαινόμενο. Μόνο το 2023, η πλατφόρμα μέτρησε πάνω από **600 εκατομμύρια ενεργούς χρήστες παγκοσμίως**, με εκατοντάδες εκατομμύρια από αυτούς να συμμετέχουν ενεργά κάθε Δεκέμβριο, κοινοποιώντας τα αποτελέσματά τους στα κοινωνικά δίκτυα. Το γεγονός αυτό αναδεικνύει τον ισχυρό δεσμό ανάμεσα στη μουσική και την προσωπική ταυτότητα, αλλά και την ανάγκη των χρηστών να εκφράσουν τη μοναδικότητά τους μέσα από τα ψηφιακά μέσα.

Η ανάπτυξη της εφαρμογής απαιτήσε τον συνδυασμό γνώσεων και την χρήση εργαλείων από διαφορετικά πεδία. Συγκεκριμένα, αξιοποιήθηκαν τεχνικές **3D σχεδίασης και απόδοσης γραφικών (rendering)** μέσω του λογισμικού **Blender**, ώστε να δημιουργηθεί μια ευρεία συλλογή από δωμάτια με αισθητική συνοχή αλλά και επαρκή παραμετροποίηση ώστε να προσαρμόζονται στις μουσικές προτιμήσεις κάθε χρήστη. Παράλληλα, η ανάπτυξη της εφαρμογής περιλάμβανε τόσο **frontend** όσο και **backend development**, βασισμένο σε σύγχρονες τεχνολογίες ιστού, επιτρέποντας την ομαλή ενσωμάτωση διεπαφών και την ασφαλή διαχείριση των δεδομένων του χρήστη.

Για την ανάλυση των μουσικών δεδομένων αξιοποιήθηκαν το **Spotify Web API** και η πλατφόρμα **EveryNoise**, μέσω των οποίων συγκεντρώθηκαν πληροφορίες για τα κορυφαία είδη μουσικής, τραγούδια και καλλιτέχνες κάθε χρήστη. Τα δεδομένα αυτά αποτέλεσαν τη βάση ώστε να σχεδιαστούν εξατομικευμένα δωμάτια για περίπου **100 διαφορετικά είδη μουσικής**. Κάθε δωμάτιο αποδόθηκε σε **ισομετρική προοπτική** ως δισδιάστατη εικόνα, η οποία λειτουργεί ως “καμβάς” πάνω στον οποίο προβάλλονται δυναμικά τα κορυφαία τραγούδια και καλλιτέχνες του εκάστοτε χρήστη.

Ο βασικός στόχος του έργου ήταν η δημιουργία μιας καινοτόμου **διαδικτυακής εφαρμογής**, η οποία μετατρέπει τα μουσικά δεδομένα του χρήστη σε μια **οπτικοποιημένη και διαδραστική εμπειρία**. Το αποτέλεσμα είναι ένα προσωποποιημένο “ψηφιακό δωμάτιο”, το οποίο αντικατοπτρίζει όχι μόνο τη μουσική ταυτότητα του χρήστη, αλλά και στοιχεία της προσωπικότητάς του. Μέσω της δυνατότητας κοινοποίησης στα κοινωνικά δίκτυα, το δωμάτιο αυτό λειτουργεί ως **μέσο έκφρασης, δημιουργικής προβολής και συμμετοχής σε μια ευρύτερη πολιτισμική συνομιλία γύρω από τη μουσική**.

1. Ανάλυση Δεδομένων

1.1 Αναζήτηση Στατιστικών

Για την ορθή υλοποίηση της ιδέας και τη βελτιστοποίηση των αποτελεσμάτων, πραγματοποιήθηκε σύντομη έρευνα σχετικά με τα δημοφιλέστερα μουσικά είδη παγκοσμίως. Η πλατφόρμα Spotify αναγνωρίζει περισσότερα από **6.000 μουσικά είδη**, αριθμός που καθιστά την εξατομικευμένη σχεδίαση δωματίου για το καθένα σχεδόν αδύνατη.

Για τον σκοπό αυτό, χρησιμοποιήθηκε το εργαλείο **EveryNoise**, μέσω του οποίου εντοπίστηκαν τα 100 δημοφιλέστερα είδη μουσικής με βάση τον αριθμό των ακροατών τους στο Spotify. Τα συγκεκριμένα είδη καλύπτουν περισσότερο από το **99% των χρηστών παγκοσμίως**, καθιστώντας τα αντιπροσωπευτικά για την περαιτέρω κατηγοριοποίηση.

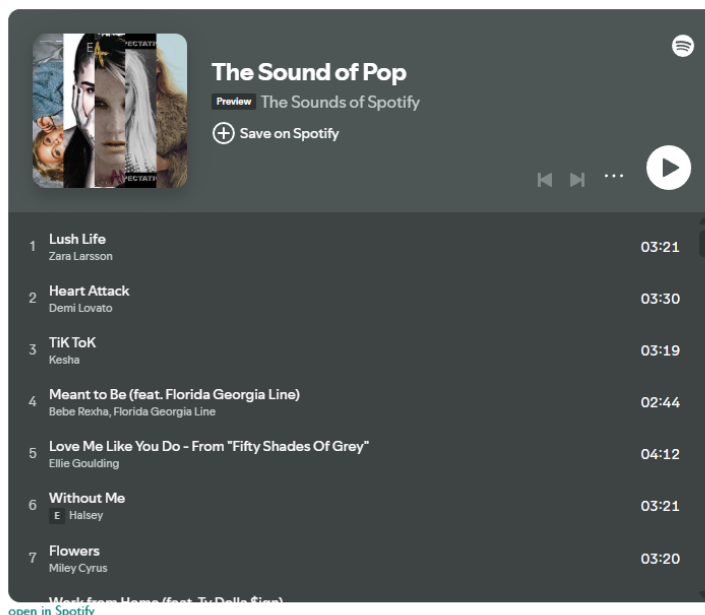
Ιδιαίτερη προσοχή δόθηκε στη μορφή των δεδομένων, καθώς το Spotify ακολουθεί συγκεκριμένο πρότυπο για την αναγνώριση και αποθήκευση μουσικών ειδών. Για παράδειγμα, το είδος **Pop** αναγνωρίζεται από την πλατφόρμα ως *pop*, και σχετίζεται με άλλα είδη βάσει ενός **συντελεστή ομοιότητας (similarity score)** που έχει ορίσει η πλατφόρμα. Το EveryNoise αξιοποιεί αυτό το μέτρο για να προσδιορίσει ποια είδη και υποείδη βρίσκονται πλησιέστερα στο *pop*, προσφέροντας σημαντική βοήθεια στην ομαδοποίηση παρόμοιων μουσικών στυλ.

Παράδειγμα ομαδοποίησης ειδών μουσικής με το εργαλείο EveryNoise με βάση την ομοιότητά τους στο είδος *pop*:

Every Noise at Once map

sort by popularity emergence modernity youth femininity engagement background tempo duration color name added similarity to pop

- 1 🎧 **pop**
- 2 🎧 dance pop
- 3 🎧 post-teen pop
- 4 🎧 talent show
- 5 🎧 boy band
- 6 🎧 uk pop
- 7 🎧 alt z
- 8 🎧 pop dance
- 9 🎧 edm
- 10 🎧 melodipop
- 11 🎧 r&b
- 12 🎧 mexican pop
- 13 🎧 swedish idol pop
- 14 🎧 deep dance pop
- 15 🎧 k-pop
- 16 🎧 electropop
- 17 🎧 ukrainian pop
- 18 🎧 japanese teen pop
- 19 🎧 chinese idol pop
- 20 🎧 latin pop
- 21 🎧 finnish dance pop
- 22 🎧 pop argentino
- 23 🎧 social media pop
- 24 🎧 classic k-pop
- 25 🎧 spanish pop
- 26 🎧 taiwan idol pop
- 27 🎧 danish pop
- 28 🎧 latin arena pop
- 29 🎧 argentine telepop
- 30 🎧 p-pop



1.2 Πίνακας Κατηγοριών

Με βάση την ίδια λογική, δημιουργήθηκαν **17 κύριες κατηγορίες μουσικής**, καθεμία από τις οποίες περιλαμβάνει **πλήθος σχετικών υποκατηγοριών**.

Η ταξινόμηση αυτή βασίστηκε τόσο στη δημοτικότητα όσο και στη μουσική συνάφεια των ειδών, όπως ορίζεται από τον συντελεστή ομοιότητας στο Spotify.

#	Κατηγορία	Υποκατηγορίες
1	Pop	pop: indie, uk pop, art pop, singer-songwriter, alt z, singer-songwriter pop, new wave pop, italian pop, indie pop
2	Rap	pop urbaine
3	Rock	classic rock, permanent wave, album rock, soft rock, hard rock, mellow gold, alternative rock, rock en espanol, latin alternative, heartland rock, pop rock, latin rock, indie rock, folk rock
4	Urbano Latino	musica mexicana, latin pop, corrido, norteno, sierreno, sad sierreno, banda, corridos tumbados, sertanejo universitario, sertanejo, arrocha, colombian pop, mexican pop, ranchera, latin arena pop, agronejo
5	Hip Hop	southern hip hop, french hip hop, german hip hop, gangster rap, conscious hip hop, east coast hip hop
6	Trap Latino	latin hip hop, trap argentino
7	Reggaeton	reggaeton colombiano
8	Dance Pop	pop dance
9	Pop Rap	canadian pop, canadian hip hop, canadian contemporary r&b
10	Modern Rock	new romantic, new wave, neo mellow, modern alternative rock
11	Trap	atl hip hop, chicago rap
12	Alternative Metal	post-grunge, nu metal, pop punk, rap metal, metal
13	K-Pop	j-pop, indonesian pop, k-pop boy group, k-pop girl group
14	R&B	urban contemporary, soul
15	EDM	indietronica, electro house, uk dance, slap house, house
16	Melodic Rap	(καμία υποκατηγορία)
17	Country	contemporary country, country road, modern country pop, modern country rock

2. Σχεδίαση 3D Δωματίων

2.1 Επεξήγηση Λογισμικού

Το λογισμικό που χρησιμοποιήθηκε για την δημιουργία των δωματίων είναι το **Blender**.

Το Blender είναι ένα **δωρεάν και open source λογισμικό για 3D γραφικά**, το οποίο χρησιμοποιείται ευρέως για τη δημιουργία μοντέλων, animation, rendering και παιχνιδιών. Προσφέρει ένα ισχυρό και ολοκληρωμένο σύνολο εργαλείων που το καθιστούν ιδανικό για καλλιτέχνες και προγραμματιστές.

Επιλέχθηκε το Blender για τη δημιουργία των δωματίων στην εφαρμογή επειδή:

- Είναι δωρεάν και εύκολα προσβάσιμο.
- Διαθέτει προηγμένες δυνατότητες μοντελοποίησης και ρεαλιστικού φωτισμού.
- Επιτρέπει εύκολη εξαγωγή 3D αντικειμένων σε πολλές μορφές για χρήση σε web εφαρμογές.

2.2 Σχεδιασμός Βασικού Δωματίου

Για την επίτευξη **ομοιομορφίας και συμβατότητας** των δωματίων με τα υπόλοιπα στοιχεία του γραφικού περιβάλλοντος (UI) της εφαρμογής, σχεδιάστηκε ένα βασικό δωμάτιο το οποίο λειτουργεί ως πρότυπο. Στόχος του βασικού αυτού δωματίου είναι να διευκολύνει τη διαδικασία δημιουργίας νέων θεματικών δωματίων, παρέχοντας ένα σταθερό αισθητικό και τεχνικό υπόβαθρο.

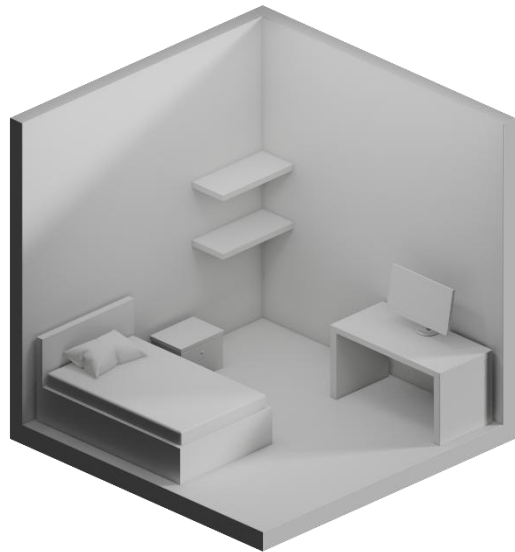
Η αισθητική προσέγγιση που ακολουθήθηκε βασίστηκε στην απεικόνιση ενός **υπνοδωματίου**, το οποίο θεωρήθηκε κατάλληλο λόγω της οικείας και ήρεμης ατμόσφαιρας που προσφέρει. Το βασικό δωμάτιο περιλαμβάνει **αποκλειστικά custom 3D μοντέλα**, τα οποία έχουν σχεδιαστεί ειδικά για τις ανάγκες της εφαρμογής. Συγκεκριμένα, τα μοντέλα που περιλαμβάνονται είναι τα εξής:

- Τοίχοι και πάτωμα
- Κρεβάτι
- Κομοδίνο και Ράφια
- Γραφείο
- Οθόνη υπολογιστή

Πέραν του σχεδιασμού των μοντέλων, στο δωμάτιο προστέθηκαν δύο πηγές φωτισμού προκειμένου να επιτευχθεί ρεαλιστική απεικόνιση του χώρου. Οι φωτιστικές πηγές που χρησιμοποιήθηκαν είναι:

- Overhead Area Light
- Sun Light

Με την ολοκλήρωση της σκηνής, το τελικό rendering πραγματοποιήθηκε με τη χρήση του **EEVEE Engine**, η κάμερα είναι ρυθμισμένη σε **Orthographic** απεικόνιση και σε ανάλυση εικόνας **4000x4000 pixels**, επιτρέποντας έτσι την απόδοση υψηλού επιπέδου λεπτομέρειας. Επιπλέον, αξιοποιήθηκε **Ray Tracing** για την ενίσχυση της ρεαλιστικότητας στον φωτισμό και τις αντανακλάσεις, συμβάλλοντας στη συνολική αληθοφάνεια του εικονικού χώρου.



2.3 Σχεδιασμός Θεματικών Δωματίων

Έχοντας ολοκληρωθεί ο σχεδιασμός του βασικού δωματίου, η διαδικασία ανάπτυξης των επιμέρους θεματικών δωματίων καθίσταται σημαντικά ευκολότερη και πιο αποδοτική. Κάθε θεματικό δωμάτιο βασίζεται σε αυτό το κοινό πρότυπο, ωστόσο διαμορφώνεται ξεχωριστά, με σκοπό να αποδώσει μια ιδιαίτερη αισθητική και να προσφέρει μοναδική οπτική ταυτότητα.

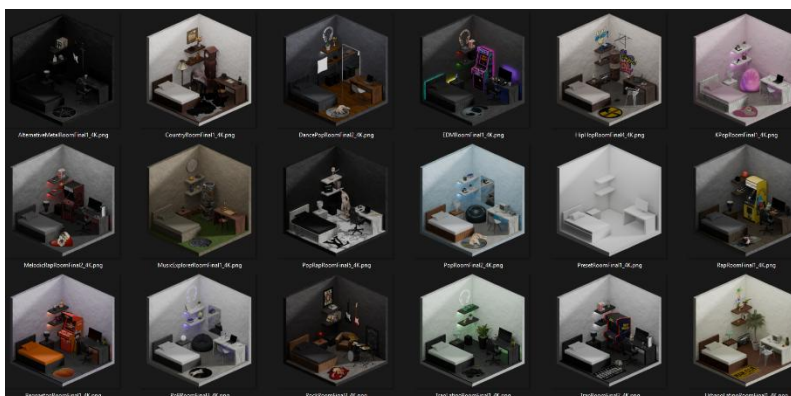
Για κάθε δωμάτιο χρησιμοποιήθηκαν εξατομικευμένα 3D μοντέλα και υλικά (textures), τα οποία επιλέχθηκαν και διαμορφώθηκαν με τρόπο που να ενισχύουν το εκάστοτε θέμα και χαρακτήρα. Η αισθητική διαφοροποίηση επιτυγχάνεται τόσο μέσω της επιλογής αντικειμένων όσο και μέσω της χρωματικής παλέτας και της συνολικής διαρρύθμισης.

Η πλειονότητα των 3D μοντέλων προήλθε από την online βιβλιοθήκη **Sketchfab**, η οποία παρέχει πληθώρα δωρεάν μοντέλων διαθέσιμων για μη εμπορική ή εμπορική χρήση, ανάλογα με την άδεια. Ανάμεσα στα μοντέλα που αξιοποιήθηκαν συγκαταλέγονται:

- Καρέκλες
- Έπιπλα
- Φωτιστικά
- Χαλιά
- Κατοικίδια
- Διακοσμητικά αντικείμενα

Όλα τα textures που χρησιμοποιήθηκαν στα θεματικά δωμάτια δημιουργήθηκαν εξολοκλήρου στο **Blender**, μέσω της τεχνικής **Procedural Texturing**, η οποία επιτρέπει τη δημιουργία υλικών χωρίς την ανάγκη εξωτερικών αρχείων εικόνας. Η μέθοδος αυτή προσφέρει ευελιξία, μικρότερο μέγεθος αρχείων και υψηλό επίπεδο παραμετροποίησης. Ενδεικτικά, μερικά από τα textures που δημιουργήθηκαν είναι τα εξής:

- Σοβάς
- Ξύλο
- Μάρμαρο
- Χαλί
- Ύφασμα



3. Ανάπτυξη Backend

3.1 Επεξήγηση Αρχιτεκτονικής

Η γλώσσα προγραμματισμού που επιλέχθηκε για την ανάπτυξη του backend είναι η **C#**, σε συνδυασμό με το **ASP.NET MVC framework**. Η επιλογή αυτή έγινε με βάση την ανάγκη για μια σταθερή, αξιόπιστη και επεκτάσιμη αρχιτεκτονική που να υποστηρίζει την κατασκευή μοντέρνων web εφαρμογών με υψηλό επίπεδο οργάνωσης.

Το **ASP.NET MVC (Model–View–Controller)** προσφέρει σαφή διαχωρισμό ανάμεσα στη λογική της εφαρμογής, την παρουσίαση και τα δεδομένα, κάτι που διευκολύνει σημαντικά τη διαδικασία ανάπτυξης και δοκιμής. Επιπλέον, το framework υποστηρίζεται πλήρως από τη Microsoft, με εκτενή τεκμηρίωση, μεγάλο οικοσύστημα εργαλείων, καθώς και πλήθος έτοιμων βιβλιοθηκών και πακέτων, στοιχεία που επιταχύνουν την ανάπτυξη και μειώνουν την πολυπλοκότητα.

Η **C#** αποτελεί μία ώριμη, αντικειμενοστραφή γλώσσα προγραμματισμού που ενδείκνυται για την ανάπτυξη ισχυρών backend λύσεων. Διακρίνεται για τη στατική της τυποποίηση, την καθαρότητα σύνταξης και τη στενή ενσωμάτωσή της με το .NET οικοσύστημα, παρέχοντας υψηλή απόδοση, ασφάλεια και δυνατότητα διασύνδεσης με εξωτερικά APIs και υπηρεσίες.

Η ενσωμάτωση του **Spotify API** πραγματοποιήθηκε ομαλά, χάρη στη συμβατότητα του ASP.NET με τις σύγχρονες τεχνολογίες REST και την εύκολη διαχείριση HTTP αιτημάτων μέσω της κλάσης `HttpClient`. Η αυθεντικοποίηση μέσω OAuth 2.0 εφαρμόστηκε αποτελεσματικά, επιτρέποντας την ασφαλή πρόσβαση στα δεδομένα των χρηστών του Spotify με τη χρήση access tokens. Η επιλογή αυτού του tech stack αποδείχθηκε ιδιαίτερα αποδοτική, καθώς προσέφερε την απαραίτητη ευελιξία για την υλοποίηση των λειτουργιών της εφαρμογής, ενώ ταυτόχρονα διατήρησε υψηλά επίπεδα απόδοσης και ασφάλειας.

3.2 Δημιουργία Εφαρμογής Spotify

Η πλατφόρμα **Spotify for Developers** παρέχει ένα ισχυρό σύνολο εργαλείων και APIs που επιτρέπουν τη δημιουργία εφαρμογών οι οποίες αλληλεπιδρούν με το οικοσύστημα του Spotify, προσφέροντας πρόσβαση σε δεδομένα χρηστών, playlists, κομμάτια, στατιστικά προτιμήσεων και άλλες λειτουργίες μέσω του **Spotify Web API**.

Για την ενσωμάτωση του Spotify στην παρούσα εφαρμογή, ήταν απαραίτητη η δημιουργία μίας custom εφαρμογής στο **Spotify Developer Dashboard**, διαδικασία που περιλαμβάνει τα εξής βήματα:

- **Σύνδεση στον Λογαριασμό Spotify Developer**
- **Δημιουργία Νέας Εφαρμογής (App Registration)**
- **Λήψη Credentials: Client ID και Client Secret**
- **Ορισμός Redirect URI**
- **Ρυθμίσεις Πρόσβασης (Scopes)**

Η παραπάνω διαδικασία διασφαλίζει ότι η εφαρμογή πληροί τα πρότυπα ασφαλείας και λειτουργικότητας του Spotify, επιτρέποντας την ασφαλή πρόσβαση σε δεδομένα χρήστη με τη συγκατάθεσή του.

3.3 Ροή Εξουσιοδότησης Χρήστη με το Spotify API

Το Spotify API χρησιμοποιεί το *OAuth 2.0* πρωτόκολλο για την εξουσιοδότηση χρηστών. Η συγκεκριμένη ροή (*Authorization Code Flow*) είναι κατάλληλη για server-side εφαρμογές, καθώς διασφαλίζει την ασφάλεια της μετάδοσης των credentials και της λήψης του *access token* μέσω ενός *authorization code* που ανταλλάσσεται με το token.

Η διαδικασία περιλαμβάνει δύο βασικά στάδια:

- **Εξακρίβωση της συγκατάθεσης του χρήστη**, μέσω ανακατεύθυνσης στη σελίδα εξουσιοδότησης του Spotify.
- **Ανταλλαγή του authorization code με access token**, ώστε να μπορούν να γίνουν αυθεντικοποιημένες κλήσεις στο Spotify API.

3.3.1 Σελίδα Authorize

Η κλάση **AuthorizeModel** είναι υπεύθυνη για το πρώτο στάδιο: την ανακατεύθυνση του χρήστη στο Spotify Authorization URL.

Η μέθοδος **OnGet()** καλείται αυτόματα όταν η σελίδα /Authorize ζητείται μέσω HTTP GET. Η μέθοδος:

- Ορίζει τις παραμέτρους εξουσιοδότησης:
 - i. **clientId**: Το μοναδικό αναγνωριστικό της εφαρμογής στο Spotify Developer Dashboard.
 - ii. **redirectUri**: Η διεύθυνση στην οποία θα επιστρέψει ο χρήστης μετά την έγκριση.
 - iii. **scope**: Οι άδειες που ζητά η εφαρμογή από τον χρήστη. Στην παρούσα περίπτωση:
 - a) **user-top-read**: Πρόσβαση στα πιο δημοφιλή τραγούδια/καλλιτέχνες του χρήστη.
 - b) **user-modify-playback-state**: Δυνατότητα ελέγχου της αναπαραγωγής.
 - c) **streaming**: Πρόσβαση σε δυνατότητες streaming.
- Κατασκευάζει το URL εξουσιοδότησης:
string authUrl = \$"https://accounts.spotify.com/authorize?...";
- Τελικά, ανακατευθύνει τον χρήστη σε αυτό το URL με: **return Redirect(authUrl);**

Αποτέλεσμα: ο χρήστης βλέπει τη σελίδα του Spotify όπου καλείται να επιτρέψει ή να αρνηθεί την πρόσβαση της εφαρμογής στα δεδομένα του.

3.3.2 Σελίδα Callback

Η σελίδα **Callback** αποτελεί το δεύτερο και πιο κρίσιμο στάδιο της ροής εξουσιοδότησης μέσω του Spotify. Αφού ο χρήστης αποδεχτεί την παροχή των απαιτούμενων αδειών στην πλατφόρμα του Spotify, ο ίδιος ανακατευθύνεται αυτόματα στη διεύθυνση **redirect_uri**, την οποία η εφαρμογή είχε ορίσει εξαρχής κατά το αίτημα εξουσιοδότησης.

Η σελίδα Callback είναι εκείνη που αντιστοιχεί στη συγκεκριμένη διεύθυνση και είναι υπεύθυνη για την παραλαβή του **authorization code**, καθώς και για την ανταλλαγή του με ένα **access token** που θα χρησιμοποιηθεί για τη μελλοντική επικοινωνία με το Spotify API.

Η σελίδα υλοποιείται ως Razor Page και η λειτουργικότητα της βρίσκεται στην ασύγχρονη μέθοδο **OnGetAsync**. Η μέθοδος αυτή ενεργοποιείται μόλις η σελίδα φορτωθεί με τη μέθοδο **GET**, φέρνοντας ως όρισμα τον authorization code που αποστέλλεται από το Spotify ως παράμετρος στο URL.

Το πρώτο βήμα της μεθόδου είναι να ελέγξει αν το code είναι κενό ή λείπει, γεγονός που μπορεί να σημαίνει αποτυχία εξουσιοδότησης ή μη συγκατάθεση από τον χρήστη. Σε μια τέτοια περίπτωση, επιστρέφεται κατάλληλη HTTP απάντηση λάθους (BadRequest).

Εφόσον ο κωδικός υπάρχει, η εφαρμογή προχωρά στην αποστολή ενός αιτήματος **POST** προς το **Spotify Token API (https://accounts.spotify.com/api/token)**. Το αίτημα αυτό περιλαμβάνει τα εξής στοιχεία:

- Τον τύπο παραχώρησης (**grant_type**), ο οποίος είναι "authorization_code".
- Τον ληφθέντα **code**.
- Την **redirect_uri**, η οποία οφείλει να είναι απόλυτα ταυτόσημη με εκείνη που δηλώθηκε στο πρώτο στάδιο εξουσιοδότησης.
- Τα **client_id** και **client_secret**, που ταυτοποιούν με ασφάλεια την εφαρμογή στο Spotify.

Τα δεδομένα αυτά κωδικοποιούνται σε μορφή **application/x-www-form-urlencoded** και αποστέλλονται προς τον διακομιστή του Spotify μέσω ενός αντικειμένου **HttpClient**.

Μετά την αποστολή, η απόκριση παραλαμβάνεται σε μορφή **JSON**. Το περιεχόμενο αυτής της απόκρισης αποδομείται σε ένα αντικείμενο τύπου **TokenResponse**, το οποίο περιλαμβάνει το **access_token**, τον **χρόνο λήξης** του token, τον **τύπο** token και το **εύρος των δικαιωμάτων (scope)**.

Εφόσον το token ληφθεί επιτυχώς, η εφαρμογή προχωρά σε ανακατεύθυνση του χρήστη προς τη σελίδα **/Room**, η οποία αποτελεί το βασικό περιβάλλον της εφαρμογής, μεταφέροντας το token ως **παράμετρο URL (π.χ. /Room?token=...)**.

Αυτό το access token είναι απαραίτητο για να γίνουν μελλοντικές αυθεντικοποιημένες κλήσεις προς το Spotify εκ μέρους του χρήστη, όπως η λήψη των κορυφαίων τραγουδιών του, η ενεργοποίηση της αναπαραγωγής ή άλλες λειτουργίες που προβλέπονται από το scope που ζητήθηκε αρχικά.

Τέλος, αν για οποιονδήποτε λόγο αποτύχει η διαδικασία ανταλλαγής του κωδικού με access token (π.χ. σφάλμα διακομιστή, λανθασμένα credentials), τότε επιστρέφεται επίσης απάντηση σφάλματος προς τον χρήστη, με μήνυμα που διευκρινίζει την αποτυχία.

Η σελίδα Callback λοιπόν ενσωματώνει τον πυρήνα της ασφάλειας και λειτουργικότητας της σύνδεσης της εφαρμογής με το Spotify, καθώς από αυτήν εξαρτάται η επιτυχής λήψη και σωστή χρήση του access token. Η υλοποίησή της με ασύγχρονο τρόπο (async/await) διασφαλίζει την αποδοτική εκτέλεση των HTTP αιτημάτων χωρίς να μπλοκάρει τον κύριο εκτελέσιμο νήμα της εφαρμογής.

3.4 Αρχιτεκτονική και Λειτουργικότητα του Συστήματος

3.4.1 Σύνοψη Λειτουργιών της Σελίδας Room

Όλες οι βασικές λειτουργίες της εφαρμογής είναι συγκεντρωμένες στη σελίδα **Room**, η οποία αποτελεί τον πυρήνα της προσωποποιημένης εμπειρίας χρήστη. Στο σημείο αυτό πραγματοποιείται η συλλογή και επεξεργασία των δεδομένων που σχετίζονται με τον εκάστοτε χρήστη, ώστε να παραχθεί ένα μοναδικό και εξατομικευμένο οπτικοακουστικό αποτέλεσμα, προσαρμοσμένο στις μουσικές του προτιμήσεις.

Πιο συγκεκριμένα, οι κύριες λειτουργίες που υλοποιούνται περιλαμβάνουν:

- Ανάκτηση του **ονόματος χρήστη (username)** από την πλατφόρμα του Spotify.
- Ανάκτηση των **κορυφαίων τραγουδιών (top tracks)** του χρήστη.
- Ανάκτηση των **κορυφαίων καλλιτεχνών (top artists)** του χρήστη.
- Ανάκτηση των **κορυφαίων μουσικών ειδών (top genres)** του χρήστη.
- Ανάκτηση του **preview URL** για τα κορυφαία τραγούδια, επιτρέποντας την αναπαραγωγή σύντομων ηχητικών αποσπασμάτων εντός της εφαρμογής.
- Αποθήκευση των πληροφοριών σε **dictionary** που σχετίζονται με κάθε δημιουργημένο δωμάτιο και την αντίστοιχη μουσική κατηγορία.
- **Κατηγοριοποίηση** του χρήστη σε μία μουσική θεματική κατηγορία.

Η παραπάνω διαδικασία ενσωματώνει λειτουργικότητες τόσο από την πλευρά του backend όσο και του frontend, με στόχο τη δημιουργία ενός δυναμικού, διαδραστικού περιβάλλοντος που αντικατοπτρίζει τις μουσικές προτιμήσεις του κάθε χρήστη με μοναδικό τρόπο.

3.4.2 Μέθοδος GetUsername

```
1 reference
private async Task<string> GetUsername(string token)
{
    using (var client = new HttpClient())
    {
        client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", token);
        var response = await client.GetAsync("https://api.spotify.com/v1/me");

        if (response.IsSuccessStatusCode)
        {
            var content = await response.Content.ReadAsStringAsync();
            var profile = JsonConvert.DeserializeObject<UserProfile>(content);
            return profile.DisplayName ?? "Your";
        }
    }

    return "Your";
}
```

Η μέθοδος **GetUsername** καλεί το Spotify API για να ανακτήσει το όνομα προφίλ του χρήστη, χρησιμοποιώντας ένα access token που έχει αποκτηθεί μέσω εξουσιοδότησης.

Στέλνει ένα GET αίτημα στο endpoint **https://api.spotify.com/v1/me** και, αν η απόκριση είναι επιτυχής, διαβάζει τα δεδομένα και επιστρέφει το **DisplayName** του χρήστη. Αν υπάρξει σφάλμα ή δεν υπάρχει όνομα, επιστρέφει την προκαθορισμένη τιμή **"Your"**.

Είσοδος (Input):

- **token (string):** Το access token του χρήστη από το Spotify.

Έξοδος (Output):

- **(string):** Το όνομα προφίλ του χρήστη (DisplayName), ή "Your" αν αποτύχει η διαδικασία.

3.4.3 Μέθοδος GetTopTracks

```
1 reference
private async Task<List<Track>> GetTopTracks(string token)
{
    var allTracks = new List<Track>();
    var selectedTracks = new List<Track>();
    var spotifyService = new SpotifyService();

    using (var client = new HttpClient())
    {
        client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", token);
        var response = await client.GetAsync("https://api.spotify.com/v1/me/top/tracks?limit=15&time_range=long_term");

        if (response.IsSuccessStatusCode)
        {
            var content = await response.Content.ReadAsStringAsync();
            var topTracksResponse = JsonConvert.DeserializeObject<TopTracksResponse>(content);

            if (topTracksResponse?.Items != null)
            {
                allTracks = topTracksResponse.Items;

                var random = new Random();
                selectedTracks = allTracks.OrderBy(x => random.Next()).Take(9).ToList();

                foreach (var track in selectedTracks)
                {
                    track.FetchedPreviewUrl = await spotifyService.GetSpotifyPreviewUrlAsync(track.Id);
                }
            }
            else
            {
                ErrorMessage = "No top tracks found.";
            }
        }
        else
        {
            ErrorMessage = $"Failed to fetch top tracks: {response.StatusCode}";
        }
    }

    return selectedTracks;
}
```

Η μέθοδος **GetTopTracks** είναι ασύγχρονη και χρησιμοποιείται για να ανακτήσει τα κορυφαία τραγούδια ενός χρήστη από το Spotify, χρησιμοποιώντας το access token του. Πραγματοποιεί ένα GET αίτημα στο endpoint **https://api.spotify.com/v1/me/top/tracks** με όριο 15 τραγουδιών για μεγάλο χρονικό διάστημα (**long_term**). Αν η κλήση είναι επιτυχής και ληφθούν δεδομένα, επιλέγει τυχαία 9 μοναδικά τραγούδια από αυτά και για το καθένα καλεί μια βοηθητική υπηρεσία (**SpotifyService**) για να πάρει το **preview URL**.

Είσοδος (Input):

- **token (string):** Το access token του χρήστη για εξουσιοδοτημένη πρόσβαση στο Spotify API.

Έξοδος (Output):

- **List<Track>:** Λίστα με 9 τυχαία επιλεγμένα τραγούδια (από τα 15 κορυφαία του χρήστη), με ενημερωμένο το preview URL για κάθε ένα.

3.4.4 Μέθοδος GetTopArtists

```
1 reference
private async Task<ArtistDetail> GetTopArtist(string token)
{
    using (var client = new HttpClient())
    {
        client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", token);
        var response = await client.GetAsync("https://api.spotify.com/v1/me/top/artists?limit=3&time_range=long_term");

        if (response.IsSuccessStatusCode)
        {
            var content = await response.Content.ReadAsStringAsync();
            var topArtistsResponse = JsonConvert.DeserializeObject<TopArtistsResponse>(content);

            if (topArtistsResponse.Items != null && topArtistsResponse.Items.Count > 0)
            {
                var random = new Random();
                int randomIndex = random.Next(topArtistsResponse.Items.Count);
                return topArtistsResponse.Items[randomIndex];
            }
        }

        return null;
    }
}
```

Η μέθοδος **GetTopArtist** είναι ασύγχρονη και χρησιμοποιείται για να ανακτήσει έναν τυχαίο καλλιτέχνη από τους κορυφαίους 3 αγαπημένους καλλιτέχνες του χρήστη στο Spotify, βασισμένη σε μεγάλο χρονικό διάστημα (**long_term**).

Η μέθοδος στέλνει ένα GET αίτημα στο endpoint **https://api.spotify.com/v1/me/top/artists**, χρησιμοποιώντας το access token για εξουσιοδότηση. Αν η κλήση πετύχει και ληφθούν δεδομένα, επιλέγει τυχαία έναν καλλιτέχνη από τη λίστα των κορυφαίων και τον επιστρέφει. Σε περίπτωση αποτυχίας ή έλλειψης δεδομένων, επιστρέφει null.

Είσοδος (Input):

- **token (string):** Το access token για πρόσβαση στο Spotify API.

Έξοδος (Output):

- **ArtistDetail:** Ένας τυχαία επιλεγμένος καλλιτέχνης από τους 3 κορυφαίους του χρήστη ή null αν δεν βρεθεί κάποιος.

3.4.5 Μέθοδος GetTrackGenres

```
1 reference
private async Task<List<GenreCount>> GetTrackGenres(string token)
{
    var genreCounts = new Dictionary<string, int>();
    using (var client = new HttpClient())
    {
        client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", token);
        var response = await client.GetAsync("https://api.spotify.com/v1/me/top/tracks?limit=15&time_range=long_term");

        if (response.IsSuccessStatusCode)
        {
            var content = await response.Content.ReadAsStringAsync();
            var topTracksResponse = JsonConvert.DeserializeObject<TopTracksResponse>(content);

            foreach (var track in topTracksResponse.Items)
            {
                foreach (var artist in track.Artists)
                {
                    var artistResponse = await client.GetAsync($"https://api.spotify.com/v1/artists/{artist.Id}");
                    if (artistResponse.IsSuccessStatusCode)
                    {
                        var artistContent = await artistResponse.Content.ReadAsStringAsync();
                        var artistData = JsonConvert.DeserializeObject<Artist>(artistContent);
                        foreach (var genre in artistData.Genres)
                        {
                            if (genreCounts.ContainsKey(genre))
                                genreCounts[genre]++;
                            else
                                genreCounts[genre] = 1;
                        }
                    }
                }
            }
        }
    }

    return genreCounts.Select(g => new GenreCount { Genre = g.Key, Count = g.Value })
        .OrderByDescending(g => g.Count)
        .ToList();
}
```

Η μέθοδος **GetTrackGenres** είναι ασύγχρονη και έχει ως στόχο να υπολογίσει και να επιστρέψει τη συχνότητα εμφάνισης των μουσικών ειδών (genres) που σχετίζονται με τα κορυφαία τραγούδια ενός χρήστη στο Spotify. Αρχικά, με χρήση του access token, η μέθοδος ζητά από το Spotify API τα 15 κορυφαία τραγούδια του χρήστη για μεγάλο χρονικό διάστημα (**long_term**). Στη συνέχεια, για κάθε τραγούδι, παίρνει τους καλλιτέχνες που συμμετέχουν και κάνει επιπλέον αιτήματα για κάθε καλλιτέχνη ξεχωριστά, ώστε να ανακτήσει τα genres που τον χαρακτηρίζουν. Μετά, συγκεντρώνει και μετρά πόσες φορές εμφανίζεται κάθε genre σε όλα τα τραγούδια. Τέλος, επιστρέφει μια λίστα με τα genres ταξινομημένα κατά φθίνουσα συχνότητα εμφάνισης.

Είσοδος (Input):

- **token (string):** Το access token του χρήστη για εξουσιοδοτημένη πρόσβαση στο Spotify API.

Έξοδος (Output):

- **List<GenreCount>:** Μια λίστα με αντικείμενα που περιέχουν το όνομα του genre και το πλήθος εμφάνισής του στα κορυφαία τραγούδια του χρήστη, ταξινομημένα από το πιο συχνό προς το λιγότερο.

3.4.6 Μέθοδος GetSpotifyPreviewUrlAsync

```
1 reference
public class SpotifyService
{
    private static readonly HttpClient _httpClient = new HttpClient();

    1 reference
    public async Task<string> GetSpotifyPreviewUrlAsync(string spotifyTrackId)
    {
        try
        {
            var embedUrl = $"https://open.spotify.com/embed/track/{spotifyTrackId}";
            var response = await _httpClient.GetStringAsync(embedUrl);

            var match = Regex.Match(response, @""audioPreview"":\s*\s*"url":\s*"([^\"]+)"");

            if (match.Success)
            {
                return match.Groups[1].Value;
            }
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Failed to fetch Spotify preview URL: {ex.Message}");
        }

        return null;
    }
}
```

Η μέθοδος **GetSpotifyPreviewUrlAsync** είναι ασύγχρονη και έχει ως σκοπό να ανακτήσει το URL του preview ενός συγκεκριμένου τραγουδιού στο Spotify, με βάση το μοναδικό αναγνωριστικό (**spotifyTrackId**) του τραγουδιού.

Χρησιμοποιεί ένα στατικό **HttpClient** για να στείλει αίτημα στην ενσωματωμένη σελίδα του Spotify για το συγκεκριμένο τραγούδι (**embed/track/{spotifyTrackId}**). Στη συνέχεια, αναλύει το περιεχόμενο της σελίδας με τη βοήθεια κανονικής έκφρασης (**regex**) για να εντοπίσει και να εξαγάγει τη διεύθυνση URL του preview audio.

Αν η διαδικασία ολοκληρωθεί επιτυχώς, επιστρέφει το URL. Σε περίπτωση σφάλματος ή αποτυχίας εύρεσης του URL, επιστρέφει null και καταγράφει το σφάλμα.

Είσοδος (Input):

- **spotifyTrackId (string):** Το μοναδικό αναγνωριστικό του τραγουδιού στο Spotify.

Έξοδος (Output):

- **string:** Το URL του audio preview του τραγουδιού ή null αν δεν βρεθεί ή παρουσιαστεί σφάλμα.

3.4.7 Μέθοδος OnGetAsync

```

0 references
public async Task<ActionResult> OnGetAsync(string token)
{
    if (string.IsNullOrEmpty(token))
    {
        ErrorMessage = "Token is missing or invalid.";
        return Page();
    }

    TopTracks = await GetTopTracks(token);
    TrackGenreList = await GetTrackGenres(token);

    bool matched = false;
    foreach (var genreCount in TrackGenreList)
    {
        var genreKey = genreCount.Genre.ToLower();
        if (genreData.TryGetValue(genreKey, out var data))
        {
            UserName = data.nickname;
            UserRoom = data.room;
            UserPet = data.pet;
            PetName = data.petName;
            PetDescription = data.petDescription;
            matched = true;
            break;
        }
    }

    if (!matched)
    {
        UserName = "The Music Explorer";
        UserRoom = "MusicExplorerRoomFinal_4M";
        UserPet = "MusicExplorerRoomPet";
        PetName = "Scarab";
        PetDescription = "A rare beetle that feeds on hidden sounds. It appears only to true music explorers.";
    }

    UserRoomInfos = TrackGenreList
        .Select(tc => tc.Genre.ToLower())
        .Where(g => genreData.ContainsKey(g))
        .Select(g => {
            var d = genreData[g];
            return new RoomInfo
            {
                Room = d.room,
                Nickname = d.nickname,
                Pet = d.pet,
                PetName = d.petName,
                PetDescription = d.petDescription
            };
        })
        .DistinctBy(r => r.Room)
        .ToList();

    if (!UserRoomInfos.Any())
    {
        var demo = new RoomInfo
        {
            Room = "MusicExplorerRoomFinal_4M",
            Nickname = "The Music Explorer",
            Pet = "MusicExplorerRoomPet",
            PetName = "Scarab",
            PetDescription = "A rare beetle that feeds on hidden sounds. It appears only to true music explorers."
        };
        UserRoomInfos.Add(demo);
    }

    TopArtist = await GetTopArtist(token);
    Username = await GetUsername(token);

    return Page();
}

```


Η μέθοδος **OnGetAsync** είναι ασύγχρονη και εκτελείται κατά το αίτημα φόρτωσης της σελίδας (**HTTP GET**) σε μια **Razor Page**. Αρχικά, ελέγχει αν το access token που λαμβάνει ως παράμετρος είναι έγκυρο. Αν το token είναι κενό ή μη έγκυρο, εμφανίζει μήνυμα λάθους και επιστρέφει την ίδια σελίδα χωρίς περαιτέρω ενέργειες.

Σε περίπτωση έγκυρου token, η μέθοδος καλεί ασύγχρονα τις μεθόδους **GetTopTracks** και **GetTrackGenres** για να ανακτήσει τα κορυφαία τραγούδια και τα είδη (genres) που σχετίζονται με αυτά. Έπειτα, προσπαθεί να βρει το πρώτο genre από τη λίστα που ταιριάζει με κάποια προκαθορισμένα δεδομένα **genreData**. Αν βρεθεί ταίριασμα, ορίζει τις αντίστοιχες ιδιότητες χρήστη (**ψευδώνυμο, δωμάτιο, κατοικίδιο και περιγραφή**).

Αν δεν βρεθεί κανένα ταίριασμα, υπάρχει μια fallback περίπτωση όπου εμφανίζεται το Music Explorer Room για όσους χρήστες δεν έχουν στα δεδομένα τους κανένα από τα κορυφαία 100 είδη παγκοσμίως.

Επιπλέον, δημιουργεί μια πλήρη λίστα με όλα τα genres που ταιριάζουν, για πιθανή παρουσίαση πολλαπλών δωματίων, φιλτράροντας και αφαιρώντας διπλότυπα. Αν δεν βρεθεί κανένα ταίριασμα, προσθέτει ένα προεπιλεγμένο δωμάτιο.

Τέλος, ανακτά τον κορυφαίο καλλιτέχνη και το όνομα χρήστη μέσω των **GetTopArtist** και **GetUsername** αντίστοιχα, και επιστρέφει την πλήρως προετοιμασμένη σελίδα.

Είσοδος (Input):

- **token (string):** Access token για το Spotify API.

Έξοδος (Output):

- **IActionResult:** Επιστρέφει τη σελίδα με όλα τα δεδομένα χρήστη, τα κορυφαία τραγούδια, τα είδη μουσικής, και τα αντίστοιχα στοιχεία για το περιβάλλον και το κατοικίδιο. Σε περίπτωση σφάλματος επιστρέφει την ίδια σελίδα με μήνυμα λάθους.

4. Ανάπτυξη Frontend

4.1 Επιλογή Αισθητικής

Η αισθητική προσέγγιση που επιλέχθηκε για τον σχεδιασμό και την ανάπτυξη του **frontend** της εφαρμογής στοχεύει πρωτίστως στη δημιουργία μιας ευχάριστης, οπτικά αρμονικής και ξεκούραστης εμπειρίας για τον τελικό χρήστη. Η χρωματική παλέτα κινείται σε **σκούρους και διάφανους τόνους**, με κυρίαρχο στοιχείο το **χαρακτηριστικό πράσινο χρώμα της πλατφόρμας του Spotify**, ενισχύοντας τη μοντέρνα και τεχνολογικά προσανατολισμένη ταυτότητα της εφαρμογής.

Ιδιαίτερη έμφαση δόθηκε στη **διαδραστικότητα** και την άμεση ανταπόκριση του περιβάλλοντος χρήστη (UI) στις ενέργειες που εκτελεί ο επισκέπτης. Όλα τα κουμπιά, καθώς και οι σύνδεσμοι, είναι πλήρως **responsive** και παρέχουν άμεσο **οπτικό feedback**, ενισχύοντας το αίσθημα ελέγχου και αλληλεπίδρασης με το σύστημα. Επιπλέον, η χρήση ομαλών και διακριτικών **animations** συμβάλλει στη συνοχή της εμπειρίας, αποφεύγοντας απότομες μεταβάσεις που ενδέχεται να αποσπάσουν ή να κουράσουν τον χρήστη.

Σημαντική επίσης ήταν η προσαρμογή της εφαρμογής σε διαφορετικές συσκευές και αναλύσεις οθόνης. Μέσω της υλοποίησης responsive σχεδιαστικών τεχνικών και της ανάπτυξης προσαρμοσμένων στίλ για τις **desktop και mobile** εκδόσεις, επιτεύχθηκε υψηλό επίπεδο συμβατότητας και βελτιστοποιημένη εμπειρία πλοήγησης, ανεξαρτήτως συσκευής. Η προσέγγιση αυτή ενισχύει τη λειτουργικότητα και καθιστά την εφαρμογή προσβάσιμη σε ευρύ φάσμα χρηστών, ενσωματώνοντας βασικές αρχές του σύγχρονου σχεδιασμού διεπαφής (**UI/UX**).

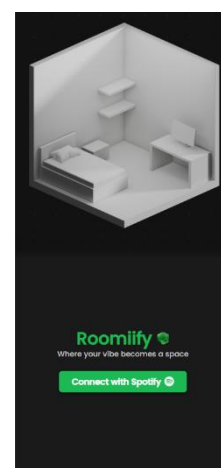
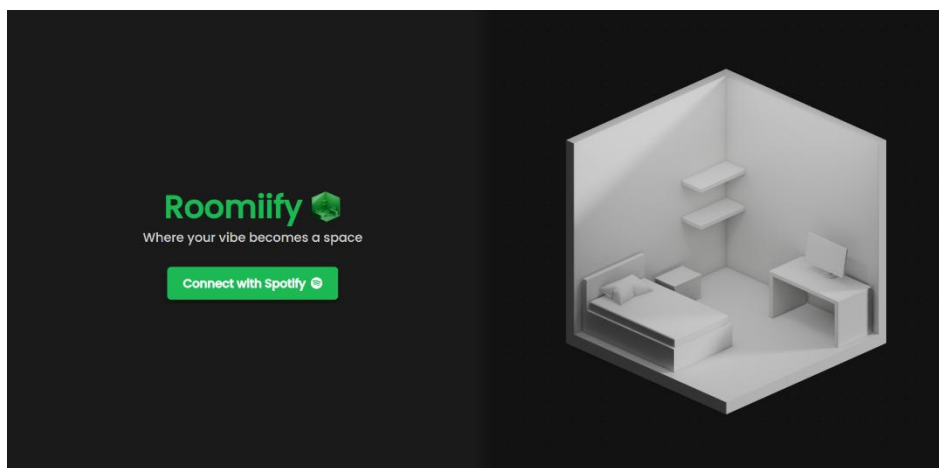
4.2 Σελίδα Connect

Η σελίδα **Connect** αποτελεί το πρώτο σημείο επαφής του χρήστη με την εφαρμογή και έχει σχεδιαστεί με στόχο να τον εισάγει άμεσα στο ύφος και το κλίμα της εμπειρίας που πρόκειται να ακολουθήσει. Από την πρώτη στιγμή γίνεται αντιληπτή η αισθητική ταυτότητα του περιβάλλοντος, με το σκούρο φόντο να κυριαρχεί και τις χαρακτηριστικές πράσινες λεπτομέρειες να παραπέμπουν άμεσα στην ταυτότητα του **Spotify**, ενισχύοντας τη συνέπεια και την αναγνωρισιμότητα της εφαρμογής.

Με ευδιάκριτο μέγεθος γραμματοσειράς φαίνεται ο τίτλος της εφαρμογής, συνοδευόμενος από μια σύντομη και περιεκτική λεζάντα, η οποία προσφέρει στον χρήστη μια πρώτη γεύση από τη λειτουργικότητα και τον χαρακτήρα της πλατφόρμας.

Επίσης εντοπίζεται εύκολα το κουμπί "**Connect with Spotify**", το οποίο λειτουργεί ως βασική πύλη εισόδου στην εμπειρία. Με την επιλογή του, ο χρήστης ανακατευθύνεται στη φόρμα σύνδεσης της επίσημης πλατφόρμας του Spotify, διασφαλίζοντας την ασφάλεια και την αξιοπιστία της διαδικασίας ταυτοποίησης.

Παράλληλα, σημαντικό στοιχείο της σελίδας αποτελεί η εμφάνιση ενός κυκλικού animation, στο οποίο προβάλλονται διαδοχικά όλα τα διαθέσιμα "δωμάτια" της εφαρμογής. Η παρουσία αυτή λειτουργεί ως ελκυστικό οπτικό ερέθισμα, κρατώντας το ενδιαφέρον του χρήστη ενεργό κατά τη διάρκεια της φόρτωσης ή της πρώτης του επαφής με την εφαρμογή, ενώ ταυτόχρονα τον προετοιμάζει για το περιεχόμενο που ακολουθεί. Η συνεχής, βρόγχος (looped) προβολή των δωματίων ενισχύει τη διαδραστικότητα και λειτουργεί ως ένα εισαγωγικό preview της συνολικής εμπειρίας.



4.3 Σελίδα Room

Η σελίδα **Room** αποτελεί τον βασικό κορμό της εφαρμογής και είναι η μοναδική σελίδα περιεχομένου, πέραν της εισαγωγικής σελίδας **Connect**. Ο σχεδιασμός της βασίζεται στην ίδια σκοτεινή αισθητική, διατηρώντας τη συνοχή με το υπόλοιπο γραφικό περιβάλλον της εφαρμογής. Ωστόσο, εμπλουτίζεται με νέες **διαφανείς διεπαφές (UI elements)**, οι οποίες αντιδρούν δυναμικά ανάλογα με το φόντο, ενισχύοντας την αίσθηση βάθους και αλληλεπίδρασης.

Το **header** και το **footer** έχουν σταθερή θέση και παραμένουν ορατά κατά τη διάρκεια της κύλισης, προσφέροντας στον χρήστη άμεση και συνεχή πρόσβαση σε όλα τα βασικά κουμπιά πλοήγησης και λειτουργιών. Τα στοιχεία αυτά έχουν σχεδιαστεί ώστε να είναι ευδιάκριτα και άμεσα αναγνωρίσιμα, είτε μέσω κειμένου είτε μέσω εικονιδίων, επιτρέποντας γρήγορη, εύκολη και αποκριτική **(responsive)** αλληλεπίδραση με την εφαρμογή.

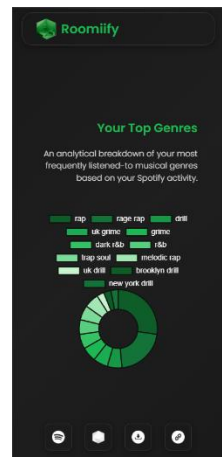
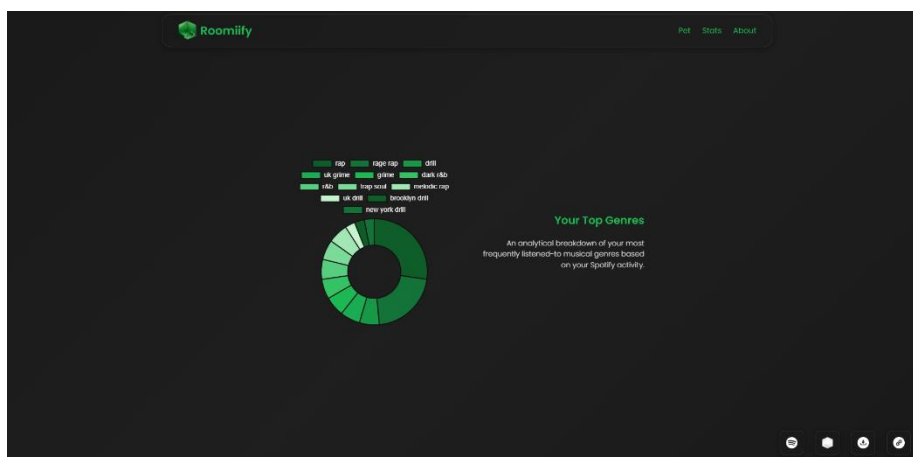
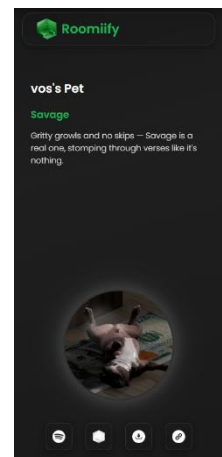
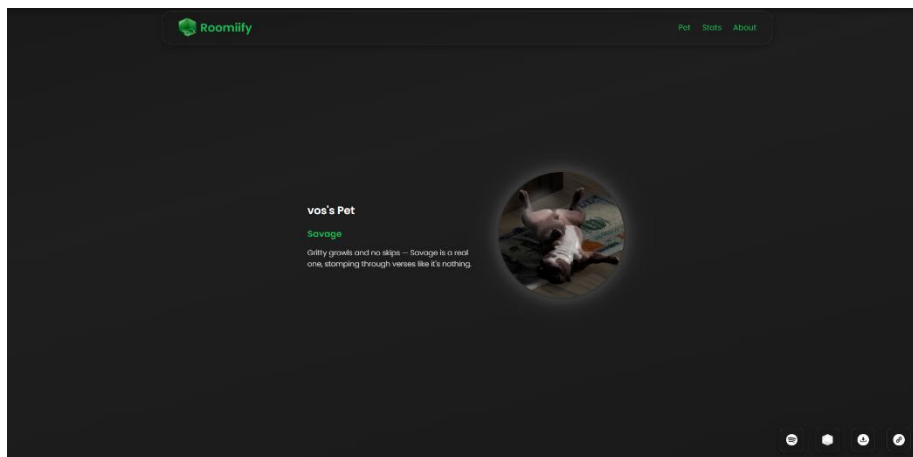
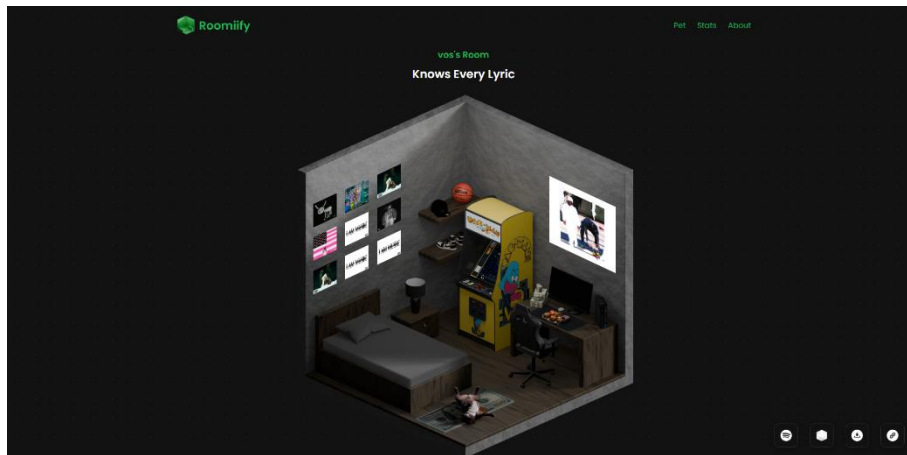
Στον βασικό χώρο του "δωματίου", έχουν τοποθετηθεί εννέα εικόνες που αντιστοιχούν σε εξώφυλλα τραγουδιών, καθώς και μία αφίσα καλλιτέχνη. Η διάταξή τους ακολουθεί τη λογική της ενσωμάτωσης στο περιβάλλον, δίνοντας την αίσθηση ότι αποτελούν φυσικό τμήμα του χώρου. Μέσω της ιδιότητας **skew** της CSS, προσαρμόζεται η γωνία θέασης των εικόνων, ώστε να ευθυγραμμίζονται οπτικά με την προοπτική των τοίχων του εικονικού δωματίου.

Κατά την αιώρηση του δείκτη του ποντικιού (**hover**) πάνω από κάθε εικόνα, εμφανίζεται ένα διαφανές και responsive **tooltip** με βασικές πληροφορίες, όπως τον τίτλο του τραγουδιού και το όνομα του καλλιτέχνη, προσφέροντας έτσι πρόσθετο περιεχόμενο χωρίς να επιβαρύνεται η οπτική απλότητα της σελίδας.

Σε χαμηλότερο σημείο της σελίδας, εντοπίζεται το πεδίο που παρουσιάζει τη μεγεθυμένη εικόνα του κατοικίδιου που αντιστοιχεί στο εκάστοτε δωμάτιο. Εκεί εφαρμόζεται ένα απλό αλλά αποτελεσματικό animation, το οποίο δημιουργεί την αίσθηση πως ο χρήστης παρατηρεί το κατοικίδιο μέσω ενός μεγεθυντικού φακού, ενισχύοντας τη διαδραστικότητα και τη φανταστική διάσταση της εμπειρίας.

Όλα τα επιμέρους στοιχεία της ιστοσελίδας εμφανίζονται σταδιακά, μέσω διακριτικών **animations** κατά την πρώτη τους εμφάνιση στην οθόνη, προσδίδοντας αίσθηση αναμονής και ενδιαφέροντος **(suspense and excitement)**. Τα animations αυτά, σε συνδυασμό με έναν ενδεικτικό δείκτη κύλισης **(scroll indicator)**, καθοδηγούν τον χρήστη και συμβάλλουν στην καλύτερη κατανόηση της λειτουργίας της εφαρμογής.

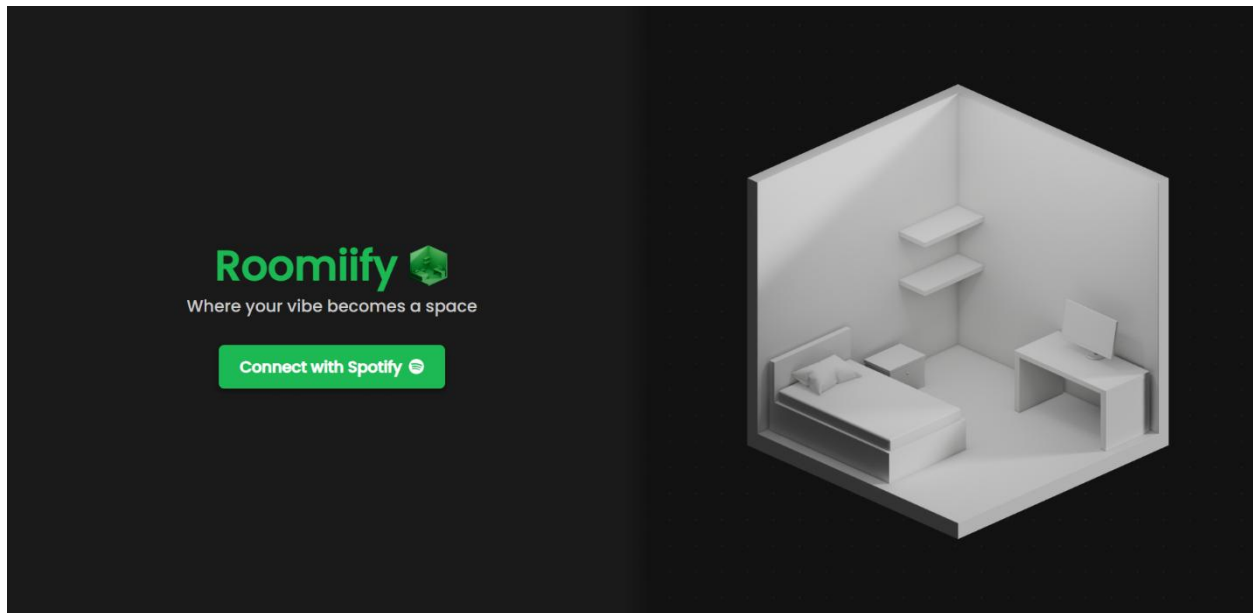
Τέλος, ιδιαίτερη μέριμνα έχει δοθεί στη **συμβατότητα** με ένα ευρύ φάσμα συσκευών. Όλα τα στοιχεία έχουν σχεδιαστεί με γνώμονα την πλήρη ανταπόκριση σε διαφορετικές αναλύσεις και μεγέθη οθονών, εξασφαλίζοντας ομοιόμορφη εμπειρία σε desktop, tablet και κινητά τηλέφωνα.



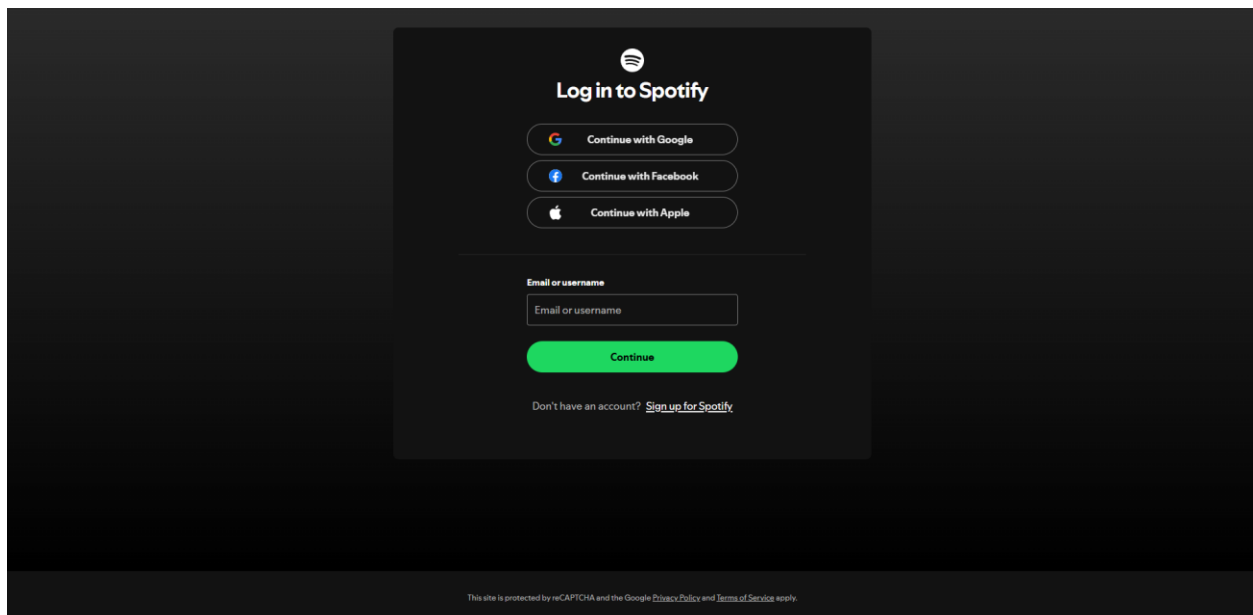
5. Εγχειρίδιο Χρήσης

5.1. Σύνδεση στην εφαρμογή

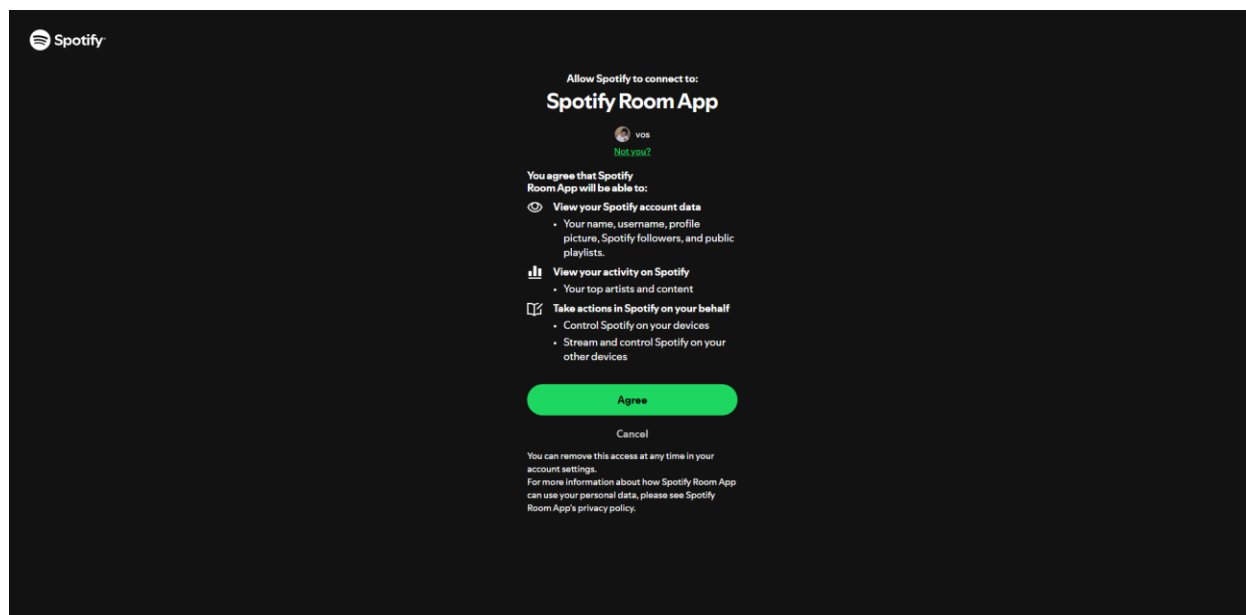
5.1.1 Κάντε κλικ στο κουμπί **Connect with Spotify**.



5.1.2 Ακολουθήστε την διαδικασία σύνδεσης στον προσωπικό σας λογαριασμό του **Spotify**.



5.1.3 Κάντε κλικ στο κουμπί **Agree** για να αποδεχθείτε τους ορούς χρήσης της εφαρμογής.



5.2. Περιήγηση στην εφαρμογή

5.2.1 Μπάρα Πλοήγησης

- Κάντε κλικ στο κουμπί **Roomiify** για να επιστρέψετε στην κορυφή της σελίδας.
- Κάντε κλικ στο κουμπί **Pet** για να πάτε στο Pet Section.
- Κάντε κλικ στο κουμπί **Stats** για να πάτε στο Stats Section.
- Κάντε κλικ στο κουμπί **About** για να πάτε στο About Section.

Σε **mobile** συσκευές είναι διαθέσιμο μόνο το κουμπί Roomiify.



5.2.2 Χρήσιμες Λειτουργίες

Εδώ βρίσκονται όλες οι χρήσιμες λειτουργίες της εφαρμογής. Τα κουμπιά αυτά είναι τοποθετημένα κάτω και αριστερά σε **desktop** ενώ σε **mobile** συσκευές είναι τοποθετημένα κάτω και στην μέση και είναι πάντα ορατά ανεξάρτητα αν ο χρήστης κάνει scroll.



5.2.3 Κουμπί αναπαραγωγής

Αρχικά το κουμπί αναπαραγωγής έχει την εξής μορφή και επισημαίνει πως δεν υπάρχει κάποιο επιλεγμένο τραγούδι από τον χρήστη.



Κάνοντας κλικ σε ένα τραγούδι μέσα από το δωμάτιο το κουμπί παίρνει την εικόνα του τραγουδιού και νότες ξεκινούν να φαίνονται από πάνω σηματοδοτώντας πως το τραγούδι βρίσκεται σε αναπαραγωγή .



Αφού δείτε την εικόνα να αλλάζει και αφού ξεκινήσει το animation με τις νότες κάντε κλικ πάνω στο κουμπί αναπαραγωγής για να κάνετε **mute**.

Το κουμπί αναπαραγωγής θα σκοτεινιάσει και το animation με τις νότες θα σταματήσει σηματοδοτώντας πως το τραγούδι έπαψε να βρίσκεται σε αναπαραγωγή.



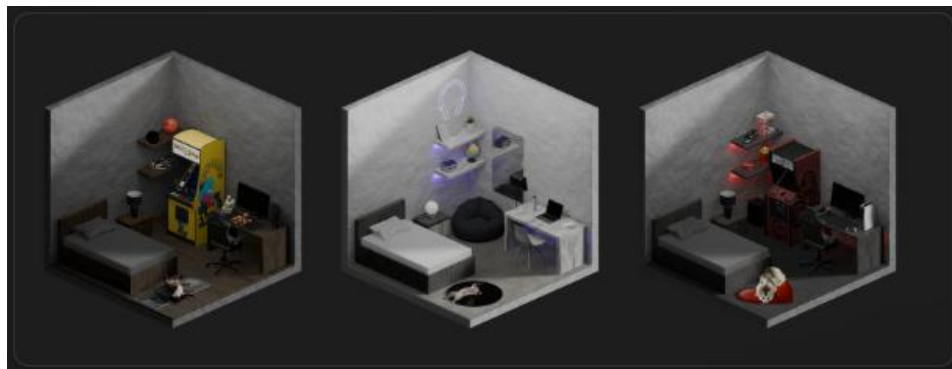
Για να κάνετε **unmute** κάντε κλικ στο κουμπί αναπαραγωγής ξανά.

5.2.4 Κουμπί Αλλαγής Δωμάτιου

Το κουμπί αλλαγής δωμάτιου είναι το δεύτερο σε σειρά κουμπί και βοηθάει τον χρήστη να αλλάξει το δωμάτιο του σε περίπτωση που το προεπιλεγμένο δεν τον αντιπροσωπεύει.



Κάντε κλικ στο κουμπί αλλαγής δωμάτιου για την εμφάνιση του μενού επιλογής.



Κάντε scroll μέσα στο μενού επιλογής δωμάτιου μέχρι να βρείτε το επιθυμητό δωμάτιο και έπειτα κάντε κλικ στο δωμάτιο της επιλογής σας.

5.2.5 Κουμπί Αποθήκευσης Δωμάτιου

Το κουμπί αποθήκευσης δωμάτιου βρίσκεται τρίτο σε σειρά και δίνει στον χρήστη την δυνατότητα να αποθηκεύσει το δωμάτιο του σε μορφή .png εικόνας με όνομα **my-music-room.png**.



Η λειτουργία αυτή προσφέρει στον χρήστη την ελευθέρια να μοιραστεί το δωμάτιο του μέσω οποιουδήποτε χώρου κοινωνικής δικτύωσης ή μέσω προσωπικών μηνυμάτων.

Για να γίνει η αποθήκευση του δωμάτιου κάντε κλικ στο κουμπί. Η αποθήκευση μπορεί να γίνει πάνω από μια φορές σε περίπτωση αλλαγής δωμάτιου.

5.2.6 Κουμπί Αντιγραφής Συνδέσμου

Το κουμπί αντιγραφής συνδέσμου βρίσκεται τελευταίο σε σειρά και δίνει στον χρήστη την δυνατότητα να μοιραστεί την ιστοσελίδα με άλλους μέσω μηνυμάτων ή με αλλά μέσα.

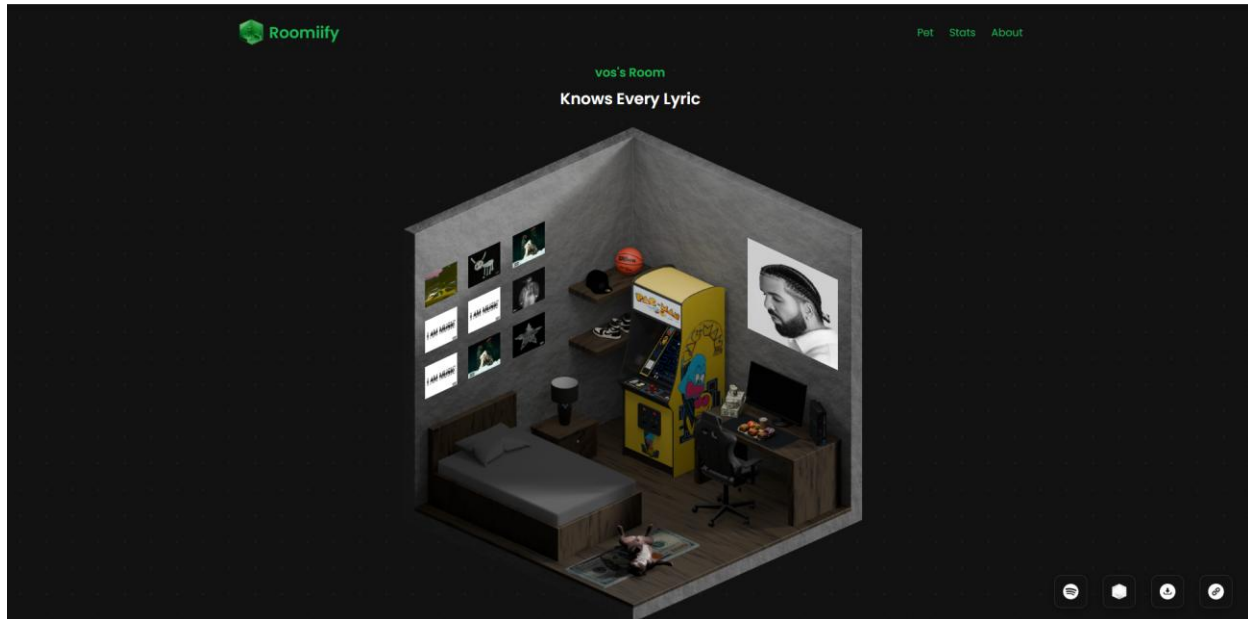


Για την αντιγραφή του συνδέσμου κάντε κλικ στο κουμπί. Ο σύνδεσμος θα αποθηκευτεί αυτόματα στο **clipboard**. Κάντε επικόλληση σε οποιοδήποτε μέσο επιθυμείτε.

5.3 Χώρος Δωμάτιου

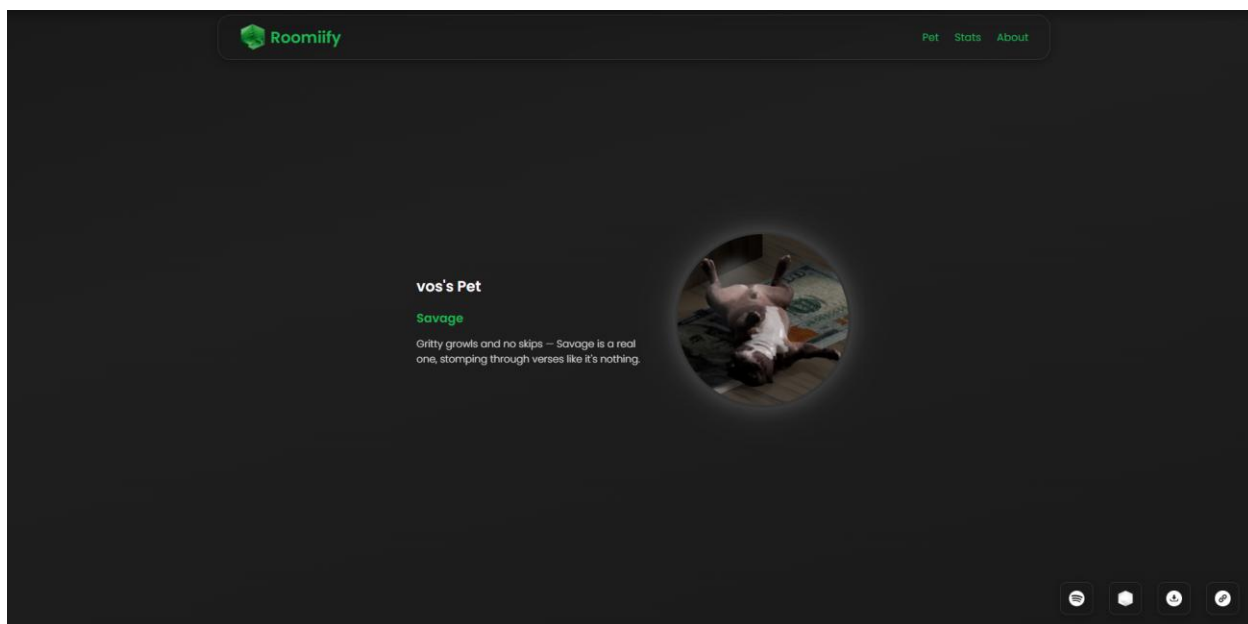
Κάντε κλικ σε οποιοδήποτε από τα 9 διαθέσιμα τραγούδια για να ξεκινήσει η αναπαραγωγή του.

Κάντε κλικ σε ένα νέο τραγούδι για να αλλάξετε την σειρά αναπαραγωγής.



5.4 Χώρος κατοικίδιου

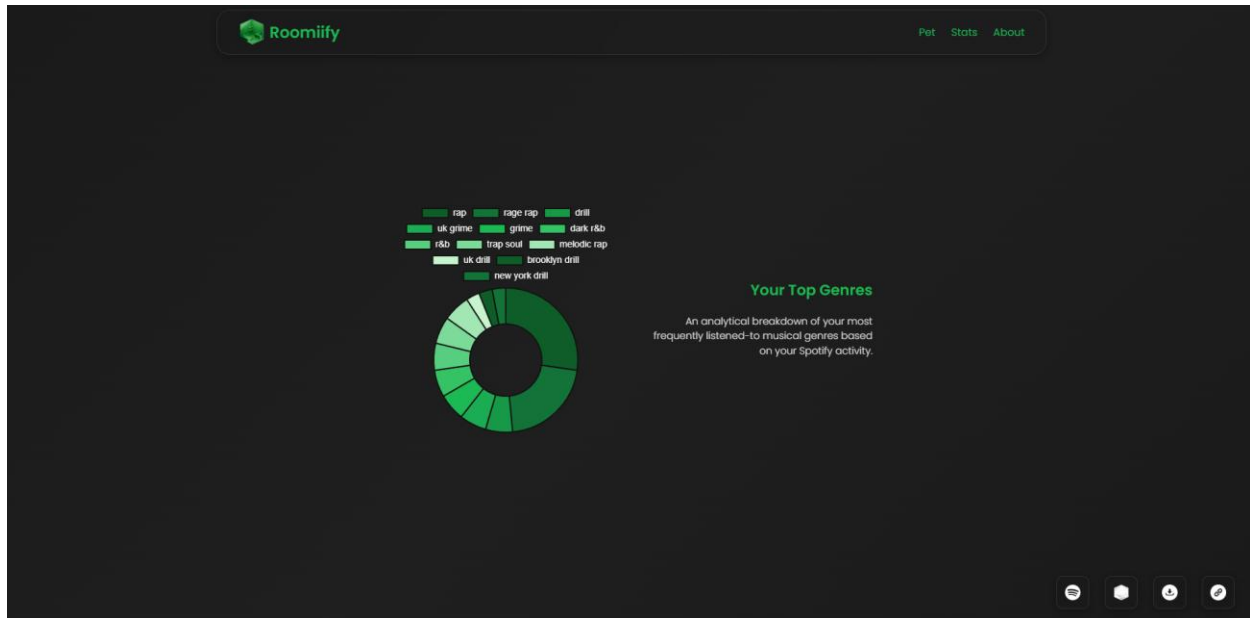
Στον χώρο κατοικίδιου βρίσκονται πληροφορίες για το κατοικίδιο που απεικονίζεται στο κάθε δωμάτιο όπως το **όνομα**, μια **σύντομη περιγραφή** και μια **μεγεθυμένη εικόνα** του.



5.5 Χώρος Στατιστικών

Στον χώρο στατιστικών βρίσκονται τα στατιστικά για τα κορυφαία είδη μουσικής του κάθε χρήστη σε ένα **Donut Chart**. Τα δεδομένα εμφανίζονται με φθίνουσα σειρά και είναι χρωματισμένα ανάλογα ώστε το πιο σκούρο χρώμα να είναι πρώτο.

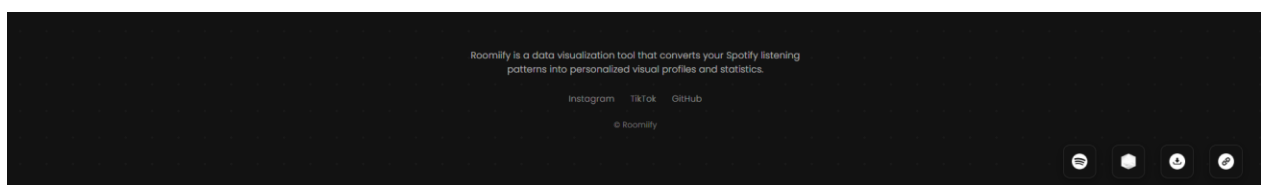
Κάντε **hover** πάνω από ένα κομμάτι του donut για να εμφανιστεί το είδος της μουσικής.



5.6 Χώρος Πληροφοριών

Στον χώρο πληροφοριών βρίσκεται μια σύντομη περιγραφή του σκοπού της εφαρμογής καθώς και κάποιοι χρήσιμοι σύνδεσμοι.

Κάντε κλικ σε έναν από τους συνδέσμους για να κατευθυνθείτε στην ανάλογη ιστοσελίδα.



6. Συμπεράσματα & Μελλοντικές Επεκτάσεις

Η εφαρμογή που αναπτύχθηκε αξιοποιεί με επιτυχία τα σύγχρονα δεδομένα από την πλατφόρμα του **Spotify**, προσφέροντας στον χρήστη μια προσωποποιημένη ψηφιακή εμπειρία βασισμένη στις μουσικές του προτιμήσεις. Μέσα από το **ψηφιακό δωμάτιο** που διαμορφώνεται, ο χρήστης μπορεί να δει και να αλληλεπιδρά με στοιχεία που αντικατοπτρίζουν την προσωπικότητα και τον μουσικό του χαρακτήρα.

Η σχεδίαση της εφαρμογής έχει ως προτεραιότητα την ευκολία στη χρήση και την ομαλή λειτουργία σε διάφορες συσκευές, εξασφαλίζοντας ταυτόχρονα μια ελκυστική και αρμονική αισθητική εμπειρία. Παρά το γεγονός ότι το περιβάλλον προσφέρει ήδη μια ικανοποιητική διαδραστικότητα, η εξέλιξή του προς έναν πιο δυναμικό και συμμετοχικό χώρο αποτελεί σημαντική προοπτική.

Στο **μέλλον**, σχεδιάζεται η ενσωμάτωση λειτουργιών που θα επιτρέπουν στον χρήστη να προσαρμόζει το δωμάτιό του με διακοσμητικά και άλλα αντικείμενα, τα οποία θα ξεκλειδώνονται μέσα από τη συνεχή χρήση και αλληλεπίδραση με την εφαρμογή. Αυτή η δυνατότητα θα μετατρέψει το δωμάτιο σε μια βάση όπου ο χρήστης θα μπορεί να εκφράσει δημιουργικά την ταυτότητά του, σε ένα περιβάλλον παρόμοιο με τα **casual παιχνίδια κατασκευής χώρων**.

Η ανάπτυξη ενός τέτοιου διαδραστικού και εξελισσόμενου ψηφιακού χώρου ανοίγει το δρόμο για κοινωνικές λειτουργίες, όπως η ανταλλαγή και επίσκεψη σε δωμάτια άλλων χρηστών, ενισχύοντας έτσι την κοινότητα και την αλληλεπίδραση μεταξύ τους. Επιπλέον, η συλλογή αναλυτικών δεδομένων για τις προτιμήσεις των χρηστών θα συμβάλει στην περαιτέρω βελτίωση της εμπειρίας μέσω προσωποποιημένων προτάσεων.

Συνολικά, η παρούσα εφαρμογή δημιουργεί τη βάση για μια νέα μορφή ψηφιακής έκφρασης, συνδυάζοντας την τεχνολογία και τη μουσική με στόχο την ενίσχυση της προσωπικής και κοινωνικής αλληλεπίδρασης σε ψηφιακά περιβάλλοντα.

7. Βιβλιογραφία

[Διαδικτυακές Πηγές]

Spotify Developer Resources

Spotify for Developers – Web API:

<https://developer.spotify.com/documentation/web-api>

Spotify Authorization Guide (OAuth 2.0):

<https://developer.spotify.com/documentation/web-api/tutorials/code-flow>

Spotify Embed Documentation:

<https://developer.spotify.com/documentation/embeds/>

Spotify Preview URL Retrieval Alternative:

<https://stackoverflow.com/questions/79237053/android-spotify-api-preview-url-for-tracks-is-suddenly-being-returned-as-null>

.NET / C# Development

C# Documentation (Microsoft):

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

ASP.NET Core MVC Docs:

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

ASP.NET Core Razor Pages Overview:

<https://learn.microsoft.com/en-us/aspnet/core/razor-pages/>

HttpClient Class (System.Net.Http):

<https://learn.microsoft.com/en-us/dotnet/api/system.net.http.httpclient>

Newtonsoft.Json (Json.NET) for .NET:

<https://www.newtonsoft.com/json>

3D Design and Blender Resources

Blender Manual:

<https://docs.blender.org/manual/en/latest/>

EEVEE Rendering Engine in Blender:

<https://docs.blender.org/manual/en/latest/render/eevee/index.html>

Procedural Texturing Guide:

https://docs.blender.org/manual/en/latest/render/shader_nodes/textures/index.html

3D Models and Design Inspiration

Sketchfab – 3D Model Library:

<https://sketchfab.com/>

My Internet Bedroom by MadeOnVerse:

<https://madeonverse.com/>

Music Genre and Statistics Tools

EveryNoise at once:

<https://everynoise.com/>

Spotify Wikipedia Page:

<https://en.wikipedia.org/wiki/Spotify>

Frontend and Web Design Resources

CSS Transform: Skew():

<https://developer.mozilla.org/en-US/docs/Web/CSS/transform-function/skew>

Chart.js Donut/Pie Charts:

<https://www.chartjs.org/docs/latest/charts/doughnut.html>