



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

**ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
Π.Μ.Σ Πληροφοριακά Συστήματα & Υπηρεσίες
Κατεύθυνση: Προηγμένα Πληροφοριακά Συστήματα**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**«Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής
Νοημοσύνης (AI)»**

Δημήτριος Μιχαηλίδης

Επιβλέπουσα : Ανδριάνα Πρέντζα, Καθηγήτρια

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής
Νοημοσύνης (AI)

Δημήτριος Μιχαηλίδης
Α.Μ.: ΜΕ2145

ΠΕΡΙΛΗΨΗ

Στην παρούσα μεταπτυχιακή διπλωματική εργασία θα εξετάσουμε το ρόλο που μπορούν να έχουν οι εφαρμογές τεχνητής νοημοσύνης στον έλεγχο λογισμικού. Ξεκινώντας με την κατανόηση του όρου “Έλεγχος Λογισμικού” μέσα από την θεωρητική ανάπτυξη και παράθεση των βασικών βημάτων του κύκλου ζωής του ελέγχου λογισμικού, των επιπέδων ελέγχου, μεθόδων και κατηγοριών που υπάρχουν και εφαρμόζονται.

Έπειτα θα κάνουμε μια επεξήγηση του όρου τεχνητής νοημοσύνης και θα προβούμε στην ανάλυση διάφορων μεθόδων, εφαρμογών, τύπων της τεχνητής νοημοσύνης και τους κλάδους της.

Εν συνεχεία θα εξετάσουμε πώς ενσωματώνεται και πώς μπορεί να χρησιμοποιηθεί η τεχνητή νοημοσύνη στον έλεγχο λογισμικού μέσω της ανάλυσης των διαφόρων εφαρμογών της σε διαφορετικά στάδια του κύκλου ζωής του ελέγχου λογισμικού. Επίσης θα γίνει εκτενής αναφορά των πλεονεκτημάτων και μειονεκτημάτων της ενσωμάτωσης της στον έλεγχο λογισμικού.

Τέλος, θα εστιάσουμε και θα εξετάσουμε εφαρμογές Generative AI που θα χρησιμοποιηθούν για την υλοποίηση ενός από των πιο σημαντικών και χρονοβόρων σταδίων του κύκλου ζωής ελέγχου λογισμικού που είναι η δημιουργία Σεναρίων Ελέγχου (Test Case Development) για τον ιστότοπο SauceDemo ορίζοντας ως τεχνική ελέγχου την τεχνική Μαύρου Κουτιού. Στη συνέχεια θα γίνει παραγωγή κώδικα σε γλώσσα TypeScript με την χρήση των εφαρμογών αυτών και θα πραγματοποιηθεί η εκτέλεση του με χρήση του framework αυτοματοποίησης ελέγχων Playwright της Microsoft.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Έλεγχος Λογισμικού

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Έλεγχος Λογισμικού, Τεχνητή Νοημοσύνη, Σενάκια Ελέγχου, Generative AI, Framework Playwright

ABSTRACT

In this Master's thesis, we will examine the role that artificial intelligence applications can have in software control. Starting with the understanding of the term "Software Audit" through the theoretical development and presentation of the basic steps of the software audit life cycle, the levels of control, methods and categories that exist and are applied.

Then we will explain the term artificial intelligence and analyze various methods, applications, types of artificial intelligence and into which disciplines it is divided.

Next, we will examine how AI is integrated and how it can be used in software control by analyzing its various applications at different stages of the software audit lifecycle. The advantages and disadvantages of its integration into software testing will also be extensively discussed.

Finally, we will focus on and examine Generative AI applications that will be used to implement one of the most important and time-consuming stages of the software audit lifecycle, which is the creation of Test Case Development for the SauceDemo website by defining the Black Box technique as a control method. It will then be generated in TypeScript language using these applications and will be executed using Microsoft's Playwright Control Automation Framework.

SUBJECT AREA: Software Testing

KEYWORDS: Software Auditing, Artificial Intelligence, Test cases, Generative AI, Playwright Framework

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω θερμά την κα. Ανδριάννα Πρέντζα για τη συνεργασία και την πολύτιμη συμβολή της. Από την πρώτη στιγμή με εμπιστεύτηκε και με τις παρατηρήσεις και συμβουλές της με βοήθησε στην βελτίωση της εργασίας και τελικά στην ολοκλήρωση της. Εν συνέχεια, θα ήθελα να ευχαριστήσω ιδιαίτερα την υποψήφια διδάκτωρ του τμήματος Ψηφιακών Συστημάτων κα. Σιαπέρα Μαρία για την καθοδήγηση της , τις πολύτιμες συμβουλές της και την άμεση ανταπόκριση της κατά τη διάρκεια υλοποίησης της εργασίας σπού με βοήθησε να καταλάβω καλύτερα τον τρόπο που πρέπει να δουλέψω για να πετύχω το καλύτερο δυνατό αποτέλεσμα.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	3
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ	8
ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ	9
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	10
1.ΕΙΣΑΓΩΓΗ	11
1.1 Στόχοι της Μεταπτυχιακής Διπλωματικής Εργασίας	11
1.2 Δομή της εργασίας.....	11
2.ΕΛΕΓΧΟΣ ΛΟΓΙΣΜΙΚΟΥ	12
2.1 Σημαντικότητα του Ελέγχου Λογισμικού	12
2.2 Κύκλος ζωής Ελέγχου Λογισμικού(Software Testing Life Cycle - STLC)	12
2.2.1 Στάδια του STLC.....	13
2.3 Διαφορετικοί τύποι Ελέγχου Λογισμικού	14
2.3.1 Ο Έλεγχος λογισμικού μπορεί γενικά να ταξινομηθεί σε 3 τύπους.....	14
2.3.2 Επίπεδα δοκιμών.....	16
2.3.3 Διαφορετικές τεχνικές ελέγχου λογισμικού	16
2.4 Βέλτιστες πρακτικές ελέγχου λογισμικού.....	19
3.ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ	21
3.1 Τι είναι η Τεχνητή Νοημοσύνη.....	21
3.2 Ιστορία της Τεχνητής Νοημοσύνης	21
3.3 Εφαρμογές της Τεχνητής Νοημοσύνης.....	22
3.4 Κατηγορίες Τεχνητής Νοημοσύνης.....	24
3.4.1 Artificial Narrow Intelligence(Τεχνητή στενή νοημοσύνη (ANI)).....	24
3.4.2 Τεχνητή γενική νοημοσύνη (Artificial General Intelligence (AGI))	24
3.4.3 Τεχνητή υπερνοημοσύνη (Artificial Super Intelligence (ASI)).....	25
3.5 Τύποι Τεχνητής Νοημοσύνης	25
3.6 Κλάδοι της Τεχνητής Νοημοσύνης.....	26
3.6.1 Μηχανική Μάθηση (Machine Learning).....	26
3.6.2 Deep Learning	27
3.6.3 Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing)	28
3.6.4 Robotics.....	30
3.6.5 Expert Systems.....	30
3.6.6 Fuzzy Logic.....	31
3.6.7 Neural Network	31
3.6.8 Computer Vision.....	31
4.ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΣΤΟΝ ΕΛΕΓΧΟ ΛΟΓΙΣΜΙΚΟΥ	33

4.1 Ο ρόλος και ο αντίκτυπος της τεχνητής νοημοσύνης στον έλεγχο λογισμικού	33
4.2 Τα σημαντικά πλεονεκτήματα της τεχνητής νοημοσύνης στον έλεγχο λογισμικού.	34
4.3 Τα μειονεκτήματα της τεχνητής νοημοσύνης στον έλεγχο λογισμικού	36
4.4 Τεχνικές και προσεγγίσεις της τεχνητής νοημοσύνης στον έλεγχο λογισμικού.	37
5.ΕΦΑΡΜΟΓΕΣ GENERATIVE AI , ΙΣΤΟΤΟΠΟΣ SAUDEMO ΚΑΙ ΠΛΑΙΣΙΟ ΑΝΟΙΧΤΟΥ ΚΩΔΙΚΑ PLAYWRIGHT	39
5.1 Η χειροκίνητη δημιουργία σεναρίων ελέγχου	39
5.2 Ο ρόλος της Τεχνητής Νοημοσύνης στην παραγωγή σεναρίων ελέγχου	39
5.3 Πλεονεκτήματα της παραγωγή σεναρίων ελέγχου με χρήση της Τεχνητής Νοημοσύνης	40
5.4 Εργαλεία ελέγχου λογισμικού με χρήση Τεχνητής Νοημοσύνης	41
5.5 Εργαλεία Τεχνητής Νοημοσύνης για την παραγωγή σεναρίων ελέγχου που θα χρησιμοποιηθούν	41
5.5.1 Large Language Models	41
5.5.2 Εφαρμογές Generative AI	43
5.6 Ιστότοπος SauceDemo E-Commerce	44
5.6.1 Βασικές Λειτουργίες του SauceDemo:	44
5.7 Framework Playwright	48
6. ΠΑΡΑΓΩΓΗ ΣΕΝΑΡΙΩΝ ΕΛΕΓΧΟΥ ΜΕ ΧΡΗΣΗ GENERATIVE AI ΕΦΑΡΜΟΓΩΝ.....	49
6.1 ChatGPT4.....	49
6.2 Copilot by Microsoft	53
6.3 BardAI by Google	58
6.4 TestCraft.app	60
7. ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΣΕΝΑΡΙΩΝ ΕΛΕΓΧΟΥ.....	67
7.1 Εύρος κάλυψης σεναρίων	67
7.2 Ορθότητα Κώδικα.....	70
8. ΣΥΜΠΕΡΑΣΜΑΤΑ.....	72
ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ	73
ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ	74
ΠΑΡΑΡΤΗΜΑ Ι.....	75
ΠΑΡΑΡΤΗΜΑ ΙΙ.....	76
ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ.....	80

ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

Σχήμα 1: Ποσοστά Επιτυχίας Σεναρίων	67
---	----

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: STLC	13
Εικόνα 2: Τύποι Ελέγχου Λογισμικού	15
Εικόνα 3: Αναπαράσταση Ελέγχου Μαύρου Κουτιού	17
Εικόνα 4: Αναπαράσταση Ελέγχου Λευκού Κουτιού	18
Εικόνα 5: Τεχνικές Ελέγχου Λογισμικού[4]	19
Εικόνα 6: Κατηγορίες Τεχνητής Νοημοσύνης[50]	25
Εικόνα 7: Νευρωνικό δίκτυο	28
Εικόνα 8: Φόρμα Σύνδεσης Χρήστη ιστότοπου SauceDemo	45
Εικόνα 9: Κατάλογος Προϊόντων ιστότοπου SauceDemo	45
Εικόνα 10: Λεπτομέριες Προϊόντος ιστότοπου SauceDemo	46
Εικόνα 11: Καλάθι Αγορών ιστότοπου SauceDemo.....	46
Εικόνα 12: Φόρμα Ολοκλήρωσης Αγοράς 1 ιστότοπου SauceDemo	47
Εικόνα 13: Φόρμα Ολοκλήρωσης Αγοράς 2 ιστότοπου SauceDemo	47
Εικόνα 14: Ολοκλήρωση Αγοράς ιστότοπου SauceDemo	48
Εικόνα 15: Διεπαφή Εφαρμογής TestCraft.app.....	61
Εικόνα 16: Εκτέλεση Σεναρίων ChatGPT4 1/2	76
Εικόνα 17: Εκτέλεση Σεναρίων ChatGPT4 2/2	77
Εικόνα 18: Εκτέλεση Σεναρίων Copilot	77
Εικόνα 19: Εκτέλεση Σεναρίων BardAI	78
Εικόνα 20: Εκτέλεση Σεναρίων TestCraft.app 1/3.....	78
Εικόνα 21: Εκτέλεση Σεναρίων TestCraft.app 2/3.....	79
Εικόνα 22: Εκτέλεση Σεναρίων TestCraft.app 3/3.....	79

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Τεχνικές τεχνητής νοημοσύνης που εφαρμόζονται σε δραστηριότητες ελέγχου λογισμικού[32].....	34
Πίνακας 2: Σενάρια Ελέγχου ChatGPT4.....	50
Πίνακας 3: Σενάρια Ελέγχου Copilot.....	54
Πίνακας 4: Σενάρια Ελέγχου BardAI	58
Πίνακας 5: Σενάρια Ελέγχου TestCraft.app.....	62
Πίνακας 6: Στατιστικά κώδικα ChatGPT4	70
Πίνακας 7:: Στατιστικά κώδικα Copilot	70
Πίνακας 8: Στατιστικά κώδικα BardAI	71
Πίνακας 9: Στατιστικά κώδικα TestCraft.app.....	71
Πίνακας 10: Συγκεντρωτικός πίνακας αποτελεσμάτων	71

1.ΕΙΣΑΓΩΓΗ

1.1 Στόχοι της Μεταπτυχιακής Διπλωματικής Εργασίας

Ο έλεγχος λογισμικού είναι μια κρίσιμη πτυχή στη διαδικασία ανάπτυξης λογισμικού, μέσω του οποίου διασφαλίζεται η ποιότητα και η αξιοπιστία ενός λογισμικού. Η ενσωμάτωση της τεχνητής νοημοσύνης στον έλεγχο λογισμικού αλλάζει και ενισχύει διάφορες διαδικασίες του, όπως η χρήση αλγορίθμου τεχνητής νοημοσύνης, αξιοποιώντας τη μηχανική μάθηση και την ανάλυση δεδομένων οι οποίες μπορούν να φέρουν επανάσταση στον έλεγχο λογισμικού, προσφέροντας βελτιωμένη απόδοση, αποτελεσματικότητα και δυνατότητες αυτοματισμού πολλών διαδικασιών.

Με τη χρήση της τεχνητής νοημοσύνης μπορούν να αυτοματοποιηθούν επαναλαμβανόμενες και χρονοβόρες διαδικασίες, όπως για παράδειγμα η δημιουργία, η εκτέλεση και η ανάλυση σεναρίων ελέγχου, με αποτέλεσμα οι μηχανικοί ελέγχου λογισμικού να έχουν περισσότερο χρόνο για να επικεντρωθούν σε πιο περίπλοκα σενάρια ελέγχου. Αυτό επιφέρει βελτίωση της απόδοσης και διασφάλιση καλύτερης ποιότητας.

Τέλος, θα εστιάσουμε και θα εξετάσουμε εφαρμογές Generative AI που θα χρησιμοποιηθούν για την υλοποίηση ενός από των πιο σημαντικών και χρονοβόρων σταδίων του κύκλου ζωής του ελέγχου λογισμικού που είναι η Δημιουργία Σεναρίων Ελέγχου (Test Case Development) και αξιολόγηση του παραχθέντα κώδικα αυτών.

1.2 Δομή της εργασίας

Κεφάλαιο 1^ο: Αναφέρεται το αντικείμενο έρευνας και οι στόχοι της εργασίας.

Κεφάλαιο 2^ο: Γίνεται εισαγωγή και θεωρητική επεξήγηση του όρου “Έλεγχος Λογισμικού”.

Κεφάλαιο 3^ο: Γίνεται εισαγωγή και θεωρητική επεξήγηση του όρου “Τεχνητής Νοημοσύνης”.

Κεφάλαιο 4^ο: Γίνεται θεωρητική ανάπτυξη για τον ρόλο που μπορεί να έχει τεχνητή νοημοσύνη και πώς μπορεί να εφαρμοστεί στον έλεγχο του λογισμικού.

Κεφάλαιο 5^ο: Γίνεται μια παρουσίαση των εργαλείων και εφαρμογών που θα χρησιμοποιηθούν για το πρακτικό κομμάτι της μεταπτυχιακής διπλωματικής εργασίας.

Κεφάλαιο 6^ο: Παρουσιάζονται τα αποτελέσματα του πρακτικού μέρους της εργασίας που θα εξετάστηκε.

Κεφάλαιο 7^ο: Παρατίθεται η αξιολόγηση των αποτελεσμάτων του πρακτικού μέρους της εργασίας που θα εξετάστηκε.

2.ΕΛΕΓΧΟΣ ΛΟΓΙΣΜΙΚΟΥ

Ο Έλεγχος Λογισμικού είναι μια διαδικασία που ελέγχει αν ένα λογισμικό ή εφαρμογή λειτουργεί χωρίς προβλήματα και διασφαλίζει ότι το λογισμικό είναι χωρίς σφάλματα, συμμορφώνεται με τις τεχνικές οδηγίες σχεδιασμού και ανάπτυξής του, και ικανοποιεί τις ανάγκες του χρήστη διαχειριζόμενο όλες τις ειδικές και ακραίες καταστάσεις. Πέρα από την ανακάλυψη σφαλμάτων, ο έλεγχος λογισμικού αναζητά επίσης τρόπους βελτίωσης της αποτελεσματικότητας, της ακρίβειας και της χρηστικότητας του λογισμικού.

Ο Έλεγχος λογισμικού μπορεί να χωριστεί σε δύο βήματα:

1. **Επαλήθευση:** Αναφέρεται στο σύνολο εργασιών που διασφαλίζουν ότι το λογισμικό υλοποιεί σωστά μια συγκεκριμένη λειτουργία. «Κατασκευάζουμε σωστά το προϊόν;».
2. **Επικύρωση:** Αναφέρεται σε ένα διαφορετικό σύνολο εργασιών που διασφαλίζουν ότι το λογισμικό που έχει κατασκευαστεί ανταποκρίνεται ή όχι στις απαιτήσεις των πελατών. «Κατασκευάζουμε το σωστό προϊόν;».

2.1 Σημαντικότητα του Ελέγχου Λογισμικού

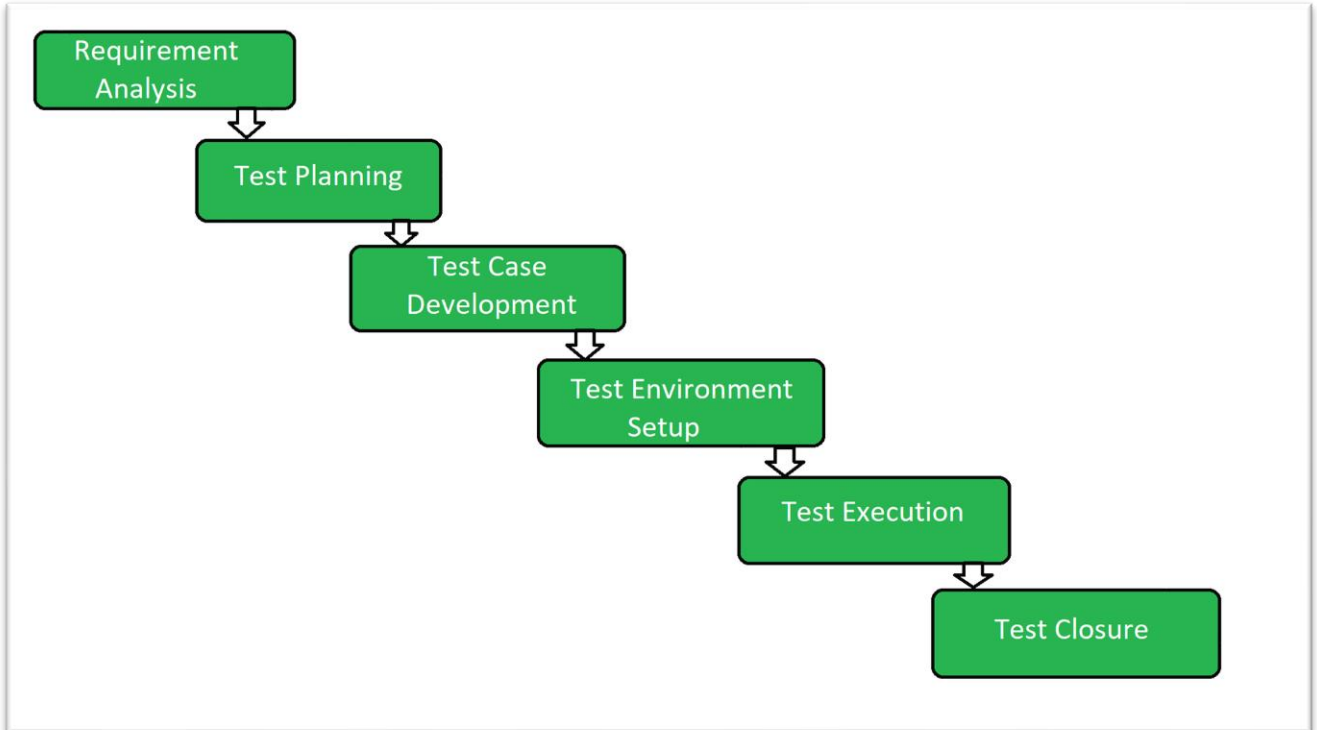
- **Έγκαιρη ανίχνευση ελαττωμάτων:** Ο έλεγχος λογισμικού είναι ζωτικής σημασίας γιατί επιτρέπει την έγκαιρη ανακάλυψη και διόρθωση των σφαλμάτων πριν από την παράδοση του λογισμικού.
- **Ποιότητα του λογισμικού:** Ο εντοπισμός και διόρθωση των προβλημάτων βελτιώνει την συνολική ποιότητα του λογισμικού.
- **Ικανοποίηση του πελάτη:** Ένα αξιόπιστο, ασφαλές και υψηλής απόδοσης λογισμικό μέσω των δοκιμών οδηγεί σε εξοικονόμηση χρόνου, κόστους και σε αυξημένη ικανοποίηση των πελατών.
- **Επεκτασιμότητα:** Ο έλεγχος λογισμικού αποκαλύπτει θέματα επεκτασιμότητας και τα όρια της λειτουργικότητας της εφαρμογής.
- **Εξοικονόμηση πόρων:** Η ταυτοποίηση και η αντιμετώπιση των προβλημάτων μετά την έναρξη είναι πιο δαπανηρή και χρονοβόρα. Γι' αυτό, συνιστάται ο τακτικός έλεγχος λογισμικού κατά τη διάρκεια της ανάπτυξης.
- **Ασφάλεια :** Στον ψηφιακό κόσμο, η Ασφάλεια είναι κρίσιμη. Οι πελάτες δίνουν προτεραιότητα σε συστήματα που εγγυώνται την ασφάλειά τους. Η μέθοδος ελέγχου ασφαλείας αξιολογεί τα πρότυπα ασφαλείας μιας εφαρμογής, με τους μηχανικούς ελέγχου λογισμικού να αναζητούν ευπάθειες.

2.2 Κύκλος ζωής Ελέγχου Λογισμικού(Software Testing Life Cycle - STLC)

Κατά τα αρχικά στάδια του STLC, ενώ αναπτύσσεται το λογισμικό ή η εφαρμογή, η ομάδα δοκιμών καθορίζει το πεδίο των ελέγχων, τα κριτήρια έναρξης / τερματισμού και σχεδιάζει τις δοκιμαστικές περιπτώσεις. Αυτή η διαδικασία συντομεύει το χρόνο ελέγχου και αναβαθμίζει την ποιότητα του προϊόντος. Μόλις ολοκληρωθεί η φάση ανάπτυξης, οι μηχανικοί ελέγχου λογισμικού είναι έτοιμοι με τις δοκιμαστικές περιπτώσεις και ξεκινούν την εκτέλεση, διευκολύνοντας την πρώιμη ανίχνευση των σφαλμάτων.

2.2.1 Στάδια του STLC

Το STLC είναι μια συστηματική διαδικασία για τον προγραμματισμό, το σχεδιασμό, την εκτέλεση και τη διαχείριση δραστηριοτήτων ελέγχου λογισμικού. Συνήθως αποτελείται από πολλές καλά καθορισμένες φάσεις, καθεμία με τους δικούς της στόχους και παραδοτέα. Ακολουθούν οι κοινώς αναγνωρισμένες φάσεις του STLC στην Εικόνα 1:



Εικόνα 1: STLC

- **Ανάλυση Απαιτήσεων (Requirement Analysis):** Η Ανάλυση Απαιτήσεων είναι το πρώτο βήμα του STLC. Σε αυτό το στάδιο, η ομάδα εξασφάλισης ποιότητας (QA) κατανοεί τις απαιτήσεις, όπως το τι πρέπει να ελεγχθεί. Εάν υπάρχει κάτι που λείπει ή δεν είναι κατανοητό, τότε η ομάδα εξασφάλισης ποιότητας συναντά τους ενδιαφερόμενους για να κατανοήσει καλύτερα τις λεπτομέρειες των απαιτήσεων.
- **Σχεδιασμός Ελέγχου (Test Planning):** Ο Σχεδιασμός Ελέγχου είναι το πιο αποτελεσματικό στάδιο, όπου καθορίζονται όλα τα σχέδια ελέγχου. Σε αυτό το στάδιο, ο υπεύθυνος της ομάδας ελέγχου υπολογίζει την εκτιμώμενη προσπάθεια και το κόστος για το έργο ελέγχων. Αυτό το στάδιο ξεκινά όταν ολοκληρώνεται το στάδιο Ανάλυση Απαιτήσεων.
- **Ανάπτυξη Περιπτώσεων Ελέγχου (Test Case Development):** Η φάση αυτή ξεκινά μετά την ολοκλήρωση της φάσης Σχεδιασμού Ελέγχου. Σε αυτό το στάδιο, η ομάδα ελέγχου λογισμικού καταγράφει λεπτομερείς ελέγχους. Η ομάδα ελέγχου λογισμικού ετοιμάζει επίσης τα απαιτούμενα δεδομένα δοκιμών. Όταν οι δοκιμές είναι έτοιμες, υπόκεινται σε αναθεώρηση από την ομάδα εξασφάλισης ποιότητας.
- **Εγκατάσταση Περιβάλλοντος Ελέγχου (Test Environment Setup):** Αποτελεί κρίσιμο στάδιο στο STLC, που καθορίζει τις συνθήκες κάτω από τις οποίες το λογισμικό υπόκειται σε έλεγχο. Αυτό το στάδιο μπορεί να συμβαίνει ταυτόχρονα με την Ανάπτυξη Περιπτώσεων Ελέγχου. Συνήθως δεν συμμετέχει η ομάδα ελέγχου λογισμικού ούτε οι προγραμματιστές ούτε οι πελάτες.

- **Εκτέλεση Ελέγχου(Test Execution):** Μετά την Ανάπτυξη Περιπτώσεων Ελέγχου και τη δημιουργία Περιβάλλοντος Δοκιμής, ξεκινά η φάση εκτέλεσης των ελέγχων. Κατά τη διάρκεια αυτής της φάσης, η ομάδα ελέγχου λογισμικού αρχίζει την εκτέλεση των δοκιμαστικών περιπτώσεων που προετοιμάστηκαν στο προηγούμενο βήμα.
- **Ολοκλήρωση Ελέγχου(Test Closure):** Η λήξη των ελέγχων αντιπροσωπεύει το τελικό στάδιο του STLC. Σε αυτό το σημείο, όλες οι δραστηριότητες δοκιμών έχουν ολοκληρωθεί και τεκμηριωθεί. Ο κύριος στόχος αυτού του σταδίου είναι να επιβεβαιώσει ότι κάθε δραστηριότητα σχετικά με τις δοκιμές έχει τελειώσει και το λογισμικό είναι έτοιμο για κυκλοφορία. Κατά την ολοκλήρωση της φάσης αυτής, η ομάδα ελέγχου πρέπει να έχει πλήρη επίγνωση της ποιότητας και της αξιοπιστίας του λογισμικού. Όλα τα προβλήματα ή τα σφάλματα που αναγνωρίστηκαν κατά τους ελέγχους πρέπει να έχουν αντιμετωπιστεί. Αυτό το στάδιο περιλαμβάνει επίσης την τεκμηρίωση της διαδικασίας ελέγχου και των εμπειριών που αποκτήθηκαν, συμβάλλοντας στη βελτίωση των μελλοντικών μεθόδων δοκιμών.

2.3 Διαφορετικοί τύποι Ελέγχου Λογισμικού

2.3.1 Ο Έλεγχος λογισμικού μπορεί γενικά να ταξινομηθεί σε 3 τύπους.

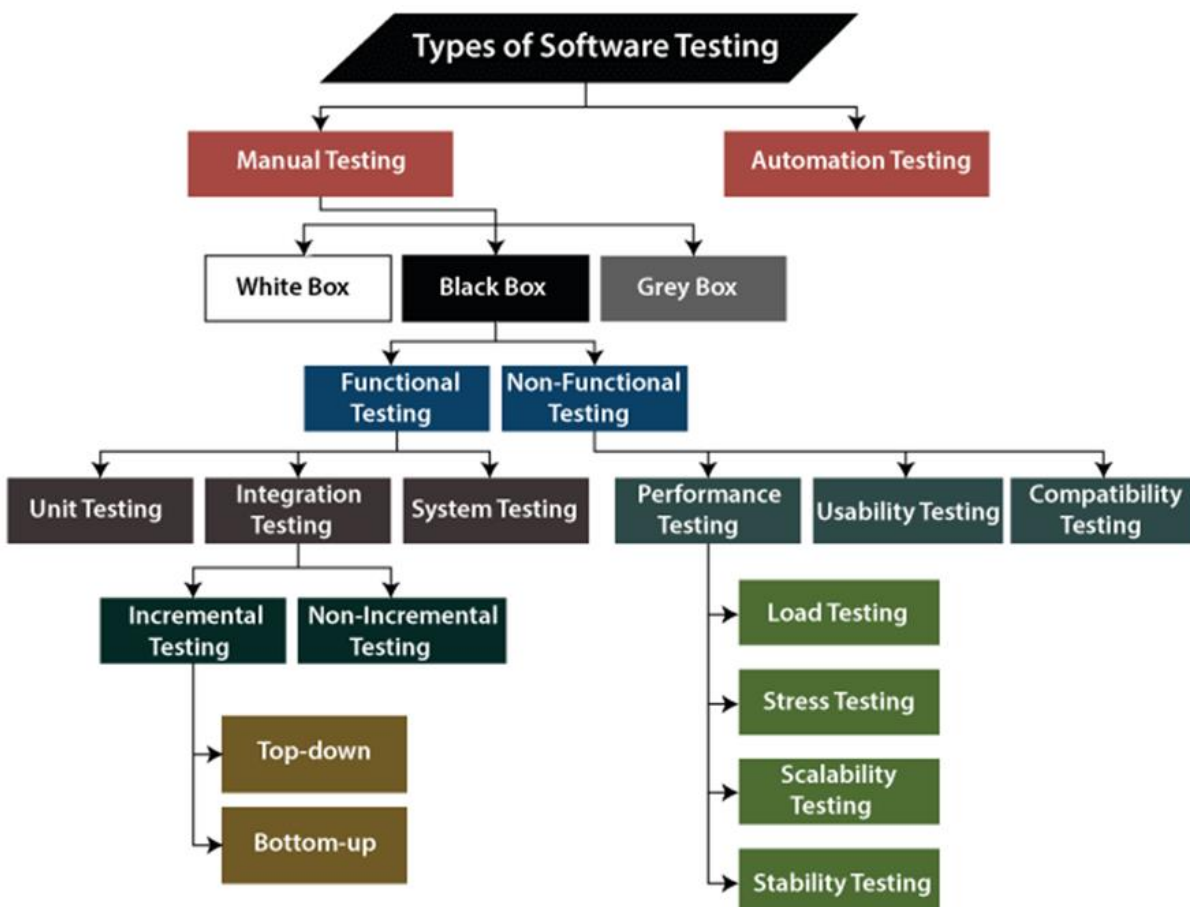
- **Λειτουργικός Έλεγχος(Functional Testing):** Οι λειτουργικοί έλεγχοι αποτελούν έναν τύπο ελέγχου λογισμικού που επιβεβαιώνουν την αντιστοιχία των συστημάτων λογισμικού με τις λειτουργικές απαιτήσεις τους. Διεξάγεται για να ελεγχθεί αν η εφαρμογή λειτουργεί σύμφωνα με τις λειτουργικές απαιτήσεις του λογισμικού ή όχι. Διάφοροι τύποι λειτουργικών ελέγχων περιλαμβάνουν τους Ελέγχους Μονάδας(Unit Testing), τους Ελέγχους Ενσωμάτωσης(Integration Testing), τους Ελέγχους Συστήματος(System Testing), τους ελέγχους ανίχνευσης σφαλμάτων(Smoke testing) και πολλούς άλλους.
- **Μη-Λειτουργικός Έλεγχος(Non-functional Testing):** Οι μη-λειτουργικοί έλεγχοι είναι μια μέθοδος ελέγχου λογισμικού που ελέγχει τη συμμόρφωση της εφαρμογής με μη-λειτουργικές απαιτήσεις, όπως απόδοση, επεκτασιμότητα, φορητότητα και αντοχή. Διάφοροι τύποι μη-λειτουργικών ελέγχων περιλαμβάνουν τους Ελέγχους Απόδοσης (Performance testing), τους Ελέγχους Αντοχής(Stress testing), τους Ελέγχους Χρησιμότητας(Usability Testing) και πολλούς άλλους.
- **Έλεγχος Συντήρησης(Maintenance Testing):** Οι έλεγχοι συντήρησης αποτελούν τη διαδικασία τροποποίησης, προσαρμογής και αναβάθμισης του λογισμικού για να προσαρμοστεί στις αναπτυσσόμενες ανάγκες του πελάτη. Αυτή η διαδικασία περιλαμβάνει τους Ελέγχους Αναγωγής(Regression Testing)που επαληθεύουν ότι πρόσφατες αλλαγές στον κώδικα δεν έχουν επηρεάσει αρνητικά προηγουμένως λειτουργικά τμήματα του λογισμικού.

Εκτός από την παραπάνω ταξινόμηση, ο Έλεγχος Λογισμικού μπορεί να χωριστεί περαιτέρω σε 2 ακόμη τύπους ελέγχου όπως παρουσιάζεται και στην Εικόνα 2:

- **Χειροκίνητος Έλεγχος(Manual Testing)** είναι μια μέθοδος στον Έλεγχο Λογισμικού όπου αξιολογούνται απευθείας οι λειτουργίες και οι δυνατότητες μιας εφαρμογής. Με αυτή την προσέγγιση, ένας μηχανικός ελέγχου λογισμικού αξιολογεί το λογισμικό χρησιμοποιώντας προκαθορισμένες περιπτώσεις δοκιμών. Κατά τη διάρκεια αυτής

της διαδικασίας, οι μηχανικοί ελέγχου σχεδιάζουν περιπτώσεις δοκιμών για τον κώδικα, αξιολογούν το λογισμικό και στη συνέχεια παράγουν μια σχετική τελική έκθεση γι' αυτό. Ο χειροκίνητος έλεγχος, επειδή πραγματοποιείται από άτομα, μπορεί να είναι χρονοβόρος και ενδέχεται να υπάρξουν ανθρώπινα λάθη. Πριν από την αυτοματοποίηση των δοκιμών κάθε νέας εφαρμογής, αυτή πρέπει πρώτα να υποβληθεί σε χειροκίνητο έλεγχο. Αν και ο χειροκίνητος έλεγχος απαιτεί περισσότερη προσπάθεια συγκριτικά με τον αυτοματοποιημένο, παρόλα αυτά κρίνεται ως ζωτικής σημασίας για τον καθορισμό της βιωσιμότητας της αυτοματοποίησης. Στον χειροκίνητο έλεγχο δεν απαιτείται εξοικείωση με κανένα συγκεκριμένο εργαλείο ελέγχου.

- **Αυτοματοποιημένος Έλεγχος(Automation Testing):** Ο αυτοματοποιημένος έλεγχος, που είναι γνωστός και ως Έλεγχος Αυτοματισμού, είναι η διαδικασία όπου η ομάδα ελέγχου δημιουργεί σενάρια και χρησιμοποιεί ένα άλλο λογισμικό για να αξιολογήσει το προϊόν. Αυτό περιλαμβάνει την αυτοματοποίηση μιας διαδικασίας που συνήθως γίνεται χειροκίνητα. Οι δοκιμές αυτοματοποίησης χρησιμοποιούνται συχνά για να εκτελέσουν γρήγορα και με συνέπεια σενάρια ελέγχου που προηγουμένως διεξάγονταν χειροκίνητα. Ο αυτοματοποιημένος έλεγχος χρησιμοποιείται επίσης για την αξιολόγηση μιας εφαρμογής από την άποψη του φόρτου, της απόδοσης και της αντοχής, ενώ παράλληλα αυξάνει την κάλυψη ελέγχου, βελτιώνει την ακρίβεια και προσφέρει εξοικονόμηση χρόνου και χρημάτων σε σύγκριση με τις χειροκίνητες δοκιμές.



Εικόνα 2: Τύποι Ελέγχου Λογισμικού

2.3.2 Επίπεδα δοκιμών

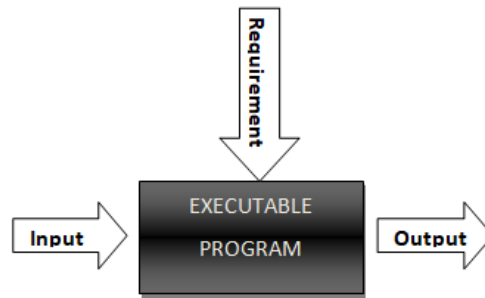
Μερικοί ερευνητές [1][2] έχουν προτείνει τα ακόλουθα επίπεδα δοκιμών:

1. **Έλεγχος μονάδων (Unit Testing):** Εμπίπτει στην κατηγορία του λευκού κουτιού, συχνά γνωστό ως συστατικό / μονάδα. Το μικρότερο δοκιμαστικό κομμάτι ενός μεγαλύτερου προγράμματος επιβεβαιώνει τη λειτουργικότητα ενός συγκεκριμένου τμήματος κώδικα.
2. **Έλεγχος ενσωμάτωσης(Integration Testing):** Ακολουθεί μια μεθοδολογία από πάνω προς τα κάτω, με κάθε τμήμα του κώδικα να αξιολογείται ξεχωριστά. Η ανάλυση χαρακτηριστικών όπως οι λειτουργικές απαιτήσεις, οι απαιτήσεις απόδοσης και αξιοπιστίας που επιβάλλονται σε σημαντικά στοιχεία σχεδιασμού είναι το πρωταρχικό καθήκον.
3. **Λειτουργικοί Έλεγχοι(Functional testing):** Είναι επίσης γνωστό ως "Black-Box." γίνεται με την παροχή αξιόπιστων πληροφοριών και στη συνέχεια αξιολογούνται τα ευρήματα. Η λειτουργία διασφαλίζει ότι το σύστημα είναι πλήρως λειτουργικό συγκρίνοντας την αναμενόμενη έξοδο με την πραγματική έξοδο.
4. **Έλεγχος σχηματισμού(Formation testing):** Επικεντρώνεται κυρίως στην εσωτερική αρχιτεκτονική του προγράμματος, όπως η κάλυψη διαδρομής, η κάλυψη δηλώσεων, η δομή ελέγχου και η πολυπλοκότητα του προγράμματος κ.λπ., ενισχύοντας τόσο την εμφάνιση όσο και τη χρησιμότητά του.[7]
5. **Έλεγχος συστήματος(System testing):** Θεωρείται ως ένα πιο εξειδικευμένο είδος ελέγχου που αναζητά ελαττώματα στα στοιχεία του λογισμικού που συνδέονται μεταξύ τους. Επιπλέον, είναι μια δοκιμή μαύρου κουτιού που αξιολογεί τη συμμόρφωση του συστήματος με ένα προκαθορισμένο κριτήριο.
6. **Έλεγχος αποδοχής(Acceptance Testing):** Είναι μια μορφή ελέγχου μαύρου κουτιού που εκτελεί ο χρήστης για να επαληθεύσει ότι ένα προϊόν, μια υπηρεσία ή ένα σύστημα είναι έτοιμο και συμμορφώνεται με τις επιχειρησιακές απαιτήσεις ως μέρος των διαδικασιών αξιολόγησης ποιότητας[3]. Επειδή ο χρήστης δεν γνωρίζει τις εσωτερικές λειτουργίες του συστήματος, αυτό γίνεται μετά την ολοκλήρωση των έργων από τους προγραμματιστές πριν το μεταβιβάσουν στους πελάτες ή τους χρήστες. Στόχος είναι η παροχή του υψηλότερου δυνατού επιπέδου βεβαιότητας για τη λειτουργία ενός συστήματος χωρίς σφάλματα με μέγιστη αξιοπιστία και αποτελεσματικότητα [4]

2.3.3 Διαφορετικές τεχνικές ελέγχου λογισμικού

Έλεγχος Μαύρου Κουτιού(Black Box Testing)

Ο έλεγχος μαύρου κουτιού είναι μια τεχνική ελέγχου λογισμικού, που χρησιμοποιείται για τον προσδιορισμό της λειτουργικότητας της εφαρμογής. Το κύριο επίκεντρο του ελέγχου μαύρου κουτιού είναι η διαθέσιμη είσοδος για μια εφαρμογή και οι αναμενόμενες έξοδοι για κάθε τιμή εισόδου. Αυτή η μέθοδος ελέγχου βασίζεται στις απαιτήσεις και τις προδιαγραφές λογισμικού. Αποτελεί μια τεχνική ελέγχου λογισμικού κατά την οποία οι εσωτερικές λειτουργίες του στοιχείου που δοκιμάζεται δεν είναι γνωστές από τον ελεγκτή. Μια απεικόνιση της τεχνικής παρουσιάζεται στην Εικόνα 3. Ονομάζεται επίσης έλεγχος βάση προδιαγραφών και δοκιμή συμπεριφοράς. Αυτή η τεχνική ονομάζεται έτσι επειδή σε αυτόν τον έλεγχο, ο μηχανικός ελέγχου δεν χρειάζεται να γνωρίζει την υλοποίηση του εσωτερικού κώδικα της εφαρμογής. Ο έλεγχος αυτός χειρίζεται τόσο έγκυρες όσο και μη έγκυρες εισόδους σύμφωνα με τις απαιτήσεις του πελάτη.[5]



Εικόνα 3: Αναπαράσταση Ελέγχου Μαύρου Κουτιού

Τεχνικές Ελέγχου Μαύρου Κουτιού:

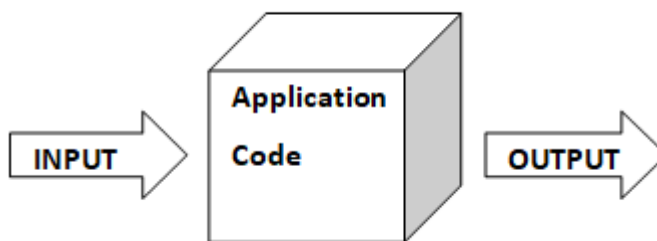
- **Διαίρεση ισοδυναμίας(Equivalence partitioning):** Είναι μια τεχνική για τον σχεδιασμό των δοκιμαστικών περιπτώσεων. Σε αυτή την τεχνική, λαμβάνονται όλες τις τιμές εισόδου και τις διαιρούνται σε κατηγορίες. Περιλαμβάνει τόσο τις έγκυρες όσο και τις άκυρες. Οι δοκιμαστικές περιπτώσεις σχεδιάζονται από κάθε κατηγορία για τον εντοπισμό των σφαλμάτων.
- **Ανάλυση οριακών τιμών :** Αυτή η τεχνική χρησιμοποιείται για τον σχεδιασμό των δοκιμαστικών περιπτώσεων με σκοπό τον εντοπισμό σφαλμάτων. Σε αυτή την τεχνική, λαμβάνονται οι τιμές στα όρια ή κοντινές στα όρια του πεδίου εισόδου για τα δεδομένα των δοκιμών. Σχεδιάζονται δοκιμαστικές περιπτώσεις τόσο για έγκυρες όσο και για άκυρες τιμές στα όρια. Επιλέγεται μια δοκιμαστική περίπτωση από κάθε οριακή τιμή.
- **Γραφική Ανάλυση Αιτίας-Αποτελέσματος(Cause Effect Graphing):** Είναι μια τεχνική σχεδιασμού δοκιμών λογισμικού που επικεντρώνεται στην εξωτερική συμπεριφορά του συστήματος. Καθορίζει τη λογική σχέση μεταξύ συνθηκών εισόδου και εξόδου με τη βοήθεια τελεστών Boolean. Οι τιμές εισόδου αντιπροσωπεύουν την «Αιτία» και οι τιμές εξόδου αντιπροσωπεύουν το «Αποτέλεσμα». Η σχέση αιτίας και αποτελέσματος βοηθά στη δημιουργία δοκιμαστικών περιπτώσεων.
- **Δοκιμές βασισμένες σε πίνακα αποφάσεων(Decision Table Based Testing):** Είναι μια καλή τεχνική για την αντιμετώπιση ενός μεγάλου αριθμού εισόδων και των αντίστοιχων αποτελεσμάτων τους. Ο πίνακας αποφάσεων έχει την ιδιότητα της πληρότητας και περιέχει όλες τις δυνατές τιμές των συνθηκών. Επιπλέον, είναι πολύ χρήσιμος για τη μετατροπή πολύπλοκων επιχειρηματικών διαδικασιών σε δοκιμαστικές περιπτώσεις.
- **Εκτίμηση Σφαλμάτων(Error Guessing):** Είναι μια τεχνική για υποθέσεις και προβλέψεις. Οι έμπειροι μηχανικοί ελέγχου λογισμικού ανακαλύπτουν τα σφάλματα. Η πλήρης επιτυχία αυτής της τεχνικής εξαρτάται από το επίπεδο δεξιοτήτων των μηχανικών ελέγχου. Ένας καλός μηχανικός ελέγχου λογισμικού γνωρίζει πού και ποιο είδος σφαλμάτων εντοπίζονται συνήθως.

Ο έλεγχος Μαύρου Κουτιού(Black Box) πραγματοποιείται στα ακόλουθα επίπεδα δοκιμών λογισμικού:

1. Έλεγχος ενσωμάτωσης
2. Έλεγχος συστήματος
3. Έλεγχος αποδοχής

Έλεγχος Λευκού Κουτιού(White-Box Testing)

Ο Έλεγχος Λευκού Κουτιού (White Box Testing) είναι μια τεχνική ελέγχου λογισμικού. Ο κύριος στόχος του ελέγχου λευκού κουτιού είναι η δομή μιας εφαρμογής. Εξετάζει την εσωτερική λογική, τη δομή του κώδικα και τη ροή ελέγχου της εφαρμογής. Αυτή η τεχνική είναι επίσης γνωστή ως Έλεγχος Καθαρού Κουτιού, Έλεγχος Ανοικτού Κουτιού, Έλεγχος Γυάλινου Κουτιού, Έλεγχος Διαφανούς Κουτιού, Έλεγχος Βασισμένος στον Κώδικα ή Έλεγχος Δομικής Κατάστασης. Στον έλεγχο λευκού κουτιού, ο ελεγκτής πρέπει να έχει πλήρη γνώση της γλώσσας προγραμματισμού (πηγαίος κώδικας). Αφορά την εσωτερική λειτουργία της εφαρμογής όπως απεικονίζεται και στην Εικόνα 4. Απαιτεί λεπτομερείς πληροφορίες σχετικά με την εφαρμογή. Ο έλεγχος λευκού κουτιού είναι μια μέθοδος ελέγχου που εφαρμόζεται στον πηγαίο κώδικα του λογισμικού. Αυτή η τεχνική εξετάζει όλα τα μονοπάτια του πηγαίου κώδικα. Για αυτή την τεχνική ελέγχου, ο ελεγκτής πρέπει να διαθέτει ισχυρές δεξιότητες προγραμματισμού. Η δοκιμή λευκού κουτιού απαιτεί εξαιρετικά εξειδικευμένους πόρους και γνώση υλοποίησης.



Εικόνα 4: Αναπαράσταση Ελέγχου Λευκού Κουτιού

Τεχνικές Ελέγχου Λευκού Κουτιού

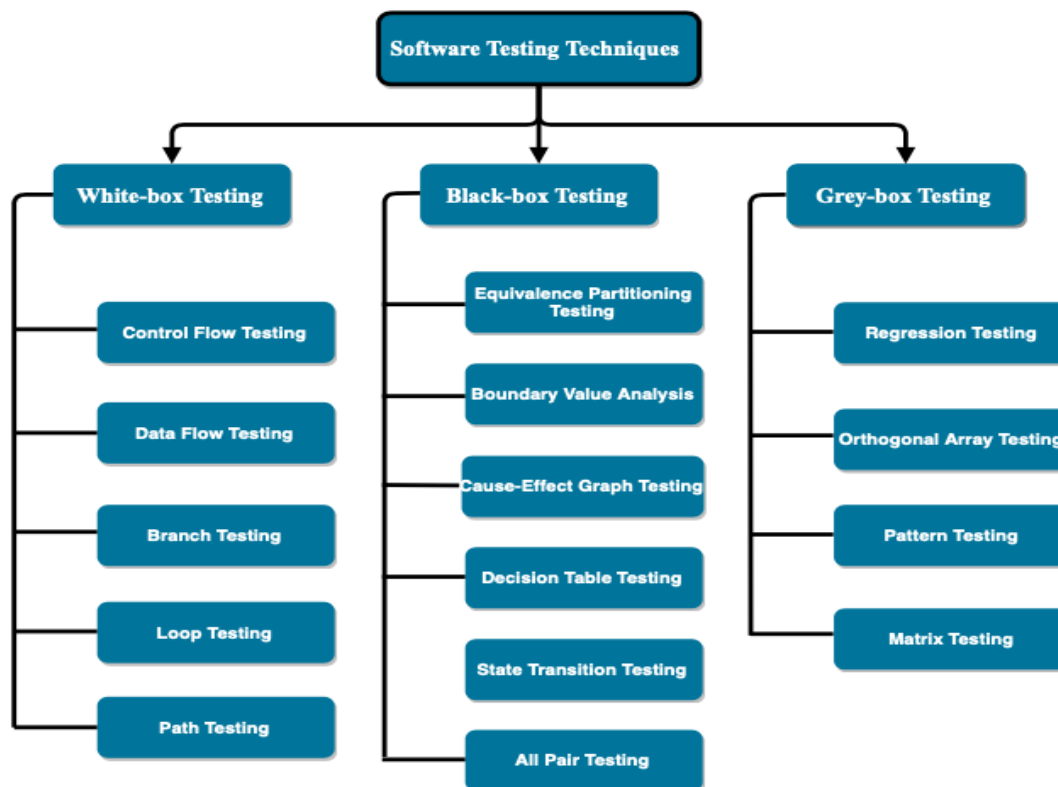
- **Δοκιμή ροής ελέγχου(Control Flow testing):** Αποτελεί μια τεχνική ελέγχου δοκιμών. Εξετάζει τη ροή ελέγχου του προγράμματος. Περιέχει απλούστερες και περίπλοκες διαδρομές. Αυτή η δοκιμή εφαρμόζεται σε όλα τα λογισμικά για τη δοκιμή της ροής ελέγχου, καθώς είναι μια θεμελιώδης τεχνική. Αυτή εφαρμόζεται σε μικρά προγράμματα και τμήματα μεγάλων προγραμμάτων.
- **Έλεγχος διαδρομής(Path Testing):** Ο έλεγχος διαδρομής είναι μία ολοκληρωμένη τεχνική, που ελέγχει όλες τις πιθανές διαδρομές του προγράμματος. Εξασφαλίζει ότι όλα τα μονοπάτια διασχίζονται τουλάχιστον μία φορά. Η τεχνική κάλυψης διαδρομής είναι καλύτερη από την κάλυψη τμήματος και είναι ιδανική για τον έλεγχο σύνθετων προγραμμάτων.
- **Έλεγχος ροής δεδομένων (Data Flow Testing):** Ο έλεγχος ροής δεδομένων επικεντρώνεται στα σημεία όπου οι μεταβλητές λαμβάνουν τιμές και όπου οι μεταβλητές χρησιμοποιούνται.

Η τεχνική ελέγχου White Box εφαρμόζεται στα ακόλουθα επίπεδα δοκιμών λογισμικού:

1. Έλεγχος μονάδας
2. Έλεγχος συστήματος
3. Έλεγχος ολοκλήρωσης

Έλεγχος Γκρίζου Κουτιού(Grey Box Testing)

Ο Έλεγχος Γκρίζου Κουτιού είναι ο συνδυασμός της τεχνικής δοκιμής Λευκού Κουτιού και Μαύρου Κουτιού που παρέχει τα πλεονεκτήματα και των δύο. Η ανάγκη για τέτοιου είδους ελέγχων προέκυψε επειδή σε αυτόν τον τύπο ελέγχων ο μηχανικός ελέγχου λογισμικού γνωρίζει την εσωτερική δομή της εφαρμογής και, ως εκ τούτου, δοκιμάζει τη λειτουργικότητα με τον βέλτιστο τρόπο, λαμβάνοντας υπόψη την εσωτερική δομή της εφαρμογής. [6]



Εικόνα 5: Τεχνικές Ελέγχου Λογισμικού[4]

2.4 Βέλτιστες πρακτικές ελέγχου λογισμικού

Ο έλεγχος λογισμικού ακολουθεί μια κοινή διαδικασία. Οι εργασίες ή τα βήματα περιλαμβάνουν τον καθορισμό του περιβάλλοντος δοκιμής, την ανάπτυξη περιπτώσεων δοκιμής, τη σύνταξη σεναρίων, την ανάλυση αποτελεσμάτων δοκιμών και την υποβολή αναφορών ελαττωμάτων. Ο έλεγχος μπορεί να είναι χρονοβόρος. Ο χειροκίνητος έλεγχος ή ο ad-hoc έλεγχος μπορεί να είναι αρκετός για μικρές κατασκευές. Ωστόσο, για μεγαλύτερα συστήματα, τα εργαλεία χρησιμοποιούνται συχνά για την αυτοματοποίηση εργασιών. Οι αυτοματοποιημένες δοκιμές βοηθούν τις ομάδες να υλοποιήσουν διαφορετικά σενάρια, να δοκιμάσουν διαφοροποιήσεις (όπως η μετακίνηση στοιχείων σε περιβάλλον cloud) και να λάβουν γρήγορα σχόλια σχετικά με το τι λειτουργεί και τι όχι. Μια καλή προσέγγιση δοκιμών περιλαμβάνει τη διεπαφή προγραμματισμού εφαρμογών (API), τη διεπαφή χρήστη και τα επίπεδα συστήματος. Επίσης, όσο περισσότερες δοκιμές είναι αυτοματοποιημένες και εκτελούνται νωρίς, τόσο το καλύτερο. Ορισμένες ομάδες δημιουργούν εσωτερικά εργαλεία αυτοματισμού δοκιμών. Ωστόσο, οι λύσεις προμηθευτών προσφέρουν δυνατότητες που μπορούν να βελτιστοποιήσουν βασικές εργασίες διαχείρισης δοκιμών, όπως:

- **Συνεχείς δοκιμές:** Οι ομάδες έργου δοκιμάζουν κάθε έκδοση μόλις γίνει διαθέσιμη. Αυτός ο τύπος ελέγχου λογισμικού βασίζεται στον αυτοματισμό ελέγχων που είναι ενσωματωμένος στη διαδικασία ανάπτυξης. Επιτρέπει την επικύρωση του λογισμικού σε ρεαλιστικά περιβάλλοντα δοκιμών - νωρίτερα στη διαδικασία - βελτιώνοντας το σχεδιασμό και μειώνοντας τους κινδύνους.
- **Διαχείριση διαμόρφωσης:** Οι οργανισμοί διατηρούν κεντρικά τους πόρους δοκιμής και παρακολουθούν ποιες εκδόσεις λογισμικού θα δοκιμάσουν. Οι ομάδες αποκτούν πρόσβαση σε στοιχεία όπως κώδικας, απαιτήσεις, έγγραφα σχεδίασης, μοντέλα, σενάρια δοκιμών και αποτελέσματα δοκιμών. Τα καλά συστήματα περιλαμβάνουν έλεγχο ταυτότητας χρήστη και διαδρομές ελέγχου για να βοηθήσουν τις ομάδες να πληρούν τις απαιτήσεις συμμόρφωσης με ελάχιστη προσπάθεια.
- **Εικονικοποίηση υπηρεσιών:** Τα περιβάλλοντα δοκιμών ενδέχεται να μην είναι διαθέσιμα, ειδικά στην αρχή της ανάπτυξης κώδικα. Η εικονική διαμόρφωση υπηρεσιών προσομοιώνει τις υπηρεσίες και τα συστήματα που λείπουν ή δεν έχουν ακόμη ολοκληρωθεί, επιτρέποντας στις ομάδες να μειώσουν τις εξαρτήσεις και να ελέγχουν νωρίτερα. Μπορούν να επαναχρησιμοποιήσουν, να αναπτύξουν και να αλλάξουν μια ρύθμιση παραμέτρων για να δοκιμάσουν διαφορετικά σενάρια χωρίς να χρειάζεται να τροποποιήσουν το αρχικό περιβάλλον.
- **Παρακολούθηση ελαττωμάτων ή σφαλμάτων:** Η παρακολούθηση ελαττωμάτων είναι σημαντική τόσο για τις ομάδες ελέγχου όσο και για τις ομάδες ανάπτυξης αναφορικά με τη μέτρηση και τη βελτίωση της ποιότητας. Τα αυτοματοποιημένα εργαλεία επιτρέπουν στις ομάδες να παρακολουθούν ελαττώματα, να μετρούν το εύρος και τον αντίκτυπό τους και να αποκαλύπτουν σχετικά ζητήματα.
- **Μετρήσεις και αναφορές:** Οι αναφορές και τα αναλυτικά στοιχεία επιτρέπουν στα μέλη της ομάδας να μοιράζονται την κατάσταση, τους στόχους και τα αποτελέσματα των ελέγχων. Τα προηγμένα εργαλεία ενσωματώνουν μετρήσεις έργου και παρουσιάζουν αποτελέσματα σε έναν πίνακα εργαλείων. Οι ομάδες βλέπουν γρήγορα τη συνολική εύρυθμη λειτουργία ενός έργου και μπορούν να παρακολουθούν τις σχέσεις μεταξύ ελέγχου, ανάπτυξης και άλλων στοιχείων του έργου.

3. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

3.1 Τι είναι η Τεχνητή Νοημοσύνη

Ο όρος τεχνητή νοημοσύνη επινοήθηκε από τον John McCarthy το 1955 σε συνέδριο που διοργανώθηκε από το Dartmouth Conference. Ο όρος χρησιμοποιήθηκε για να αναφερθεί σε όλα τα «συστήματα προγραμματισμού στα οποία η μηχανή προσομοιώνει κάποια ευφυή ανθρώπινη συμπεριφορά». Σύμφωνα με τον John McCarthy, είναι «Η επιστήμη και η μηχανική της κατασκευής ευφύων μηχανών, ειδικά ευφύων προγραμμάτων υπολογιστών»[8].

Η τεχνητή νοημοσύνη βασίζεται στην αρχή ότι η ανθρώπινη νοημοσύνη μπορεί να οριστεί με τέτοιο τρόπο ώστε μια μηχανή να μπορεί εύκολα να τη μιμηθεί και να εκτελέσει εργασίες, από τις πιο απλές έως εκείνες που είναι ακόμη πιο περίπλοκες. Οι στόχοι της τεχνητής νοημοσύνης περιλαμβάνουν τη μίμηση της ανθρώπινης γνωστικής δραστηριότητας. Οι ερευνητές και οι προγραμματιστές στον τομέα κάνουν εκπληκτικά γρήγορα βήματα στη μίμηση δραστηριοτήτων όπως η μάθηση, η συλλογιστική και η αντίληψη, στο βαθμό που αυτές μπορούν να οριστούν συγκεκριμένα. Μερικοί πιστεύουν ότι οι καινοτόμοι μπορεί σύντομα να είναι σε θέση να αναπτύξουν συστήματα που υπερβαίνουν την ικανότητα των ανθρώπων να μάθουν ή να αιτιολογήσουν οποιοδήποτε θέμα. Αλλά άλλοι παραμένουν σκεπτικοί επειδή όλη η γνωστική δραστηριότητα είναι δεμένη με αξιολογικές κρίσεις που υπόκεινται στην ανθρώπινη εμπειρία.[9]

Το ιδανικό χαρακτηριστικό της τεχνητής νοημοσύνης είναι η ικανότητά της να εξορθολογίζει και να λαμβάνει μέτρα που έχουν τις καλύτερες πιθανότητες επίτευξης ενός συγκεκριμένου στόχου. Ένα υποσύνολο της τεχνητής νοημοσύνης είναι η μηχανική μάθηση (ML), η οποία αναφέρεται στην έννοια ότι τα προγράμματα υπολογιστών μπορούν αυτόματα να μάθουν και να προσαρμοστούν σε νέα δεδομένα χωρίς να βοηθούνται από ανθρώπους. Οι τεχνικές βαθιάς μάθησης επιτρέπουν αυτήν την αυτόματη μάθηση μέσω της απορρόφησης τεράστιων ποσοτήτων μη δομημένων δεδομένων όπως κείμενο, εικόνες ή βίντεο.[9]

3.2 Ιστορία της Τεχνητής Νοημοσύνης

Χρονολόγιο της Τεχνητής Νοημοσύνης

[10]Οι αιώνες που οδήγησαν στη δεκαετία του 1950 είδαν την εμφάνιση αρκετών φιλοσοφικών και λογικών εννοιών που χρησίμευσαν ως θεμέλιο για τις θεωρίες της τεχνητής νοημοσύνης. Οι αρχαίοι Έλληνες φιλόσοφοι είχαν σημαντική επιρροή στον δυτικό πολιτισμό, με ιδέες για την ουσία της συνείδησης, της ανθρώπινης σκέψης και της μάθησης. Για εκατοντάδες χρόνια, αυτές οι έννοιες εξελίχθηκαν για να επικεντρωθούν τελικά περισσότερο στη δυνατότητα των μηχανών να αποκτήσουν την ικανότητα να μαθαίνουν και στην τεχνητή νοημοσύνη, καθώς η τεχνολογία ενσωματώθηκε περαιτέρω στην ανθρώπινη ζωή.

Το χρονοδιάγραμμα της τεχνητής νοημοσύνης χρονολογείται συγκεκριμένα από το 1763, όταν ο Thomas Bayes ανέπτυξε ένα πλαίσιο για την πιθανότητα γεγονότων, που ονομάζεται Bayesian αναφορά, το οποίο χρησίμευσε ως κορυφαία προσέγγιση για τη μηχανική μάθηση. Στις αρχές του 1900 είδαμε τις πρώτες απεικονίσεις ρομπότ σε δημοφιλή μέσα ενημέρωσης από όλο τον κόσμο σε ταινίες επιστημονικής φαντασίας.

1950: Ο Alan Turing δημοσιεύει το σημαντικό έργο του, "Computing Machinery and Intelligence" και ο όρος "τεχνητή νοημοσύνη" επινοήθηκε από τον John McCarthy. Ο McCarthy αναπτύσσει επίσης τη δημοφιλή γλώσσα προγραμματισμού Lisp, η οποία χρησιμοποιείται στην έρευνα AI.

1960: Το πρώτο βιομηχανικό ρομπότ αρχίζει να εργάζεται σε εργοστάσιο της General Motors. Αναπτύσσεται το πρόγραμμα ELIZA, το οποίο είναι σε θέση να συνομιλήσει με ένα άτομο στα αγγλικά.

1970: Κατασκευάζεται στην Ιαπωνία το πρώτο ανθρωπόμορφο ρομπότ με τη βασική ικανότητα να βλέπει, να κινείται και να συνομιλεί. Ένα πρώιμο σύστημα αναγνώρισης βακτηρίων αναπτύσσεται στο Πανεπιστήμιο του Στάνφορντ.

1980: Η Mercedes-Benz δοκιμάζει το πρώτο αυτοκίνητο χωρίς οδηγό που ενσωματώνει τις θεμελιώδεις αρχές τέτοιων αυτοκινήτων που κατασκευάζονται σήμερα. Το Jabberwacky κυκλοφορεί ως πρώιμο παράδειγμα ενός σύγχρονου συστήματος chatbot.

1990: Ο Deep Blue, ένας υπολογιστής που παίζει σκάκι, κερδίζει τον παγκόσμιο πρωταθλητή. Το πρώτο ευρετήριο ιστού της Google έχει 26 εκατομμύρια σελίδες.

2000: Αναπτύσσονται αρκετά νέα ρομπότ, όπως το ASIMO της Honda και το Kismet του MIT. Η ποσότητα των ψηφιακών πληροφοριών που παράγονται ανέρχεται σε εκατοντάδες exabytes και αυξάνεται με ταχείς ρυθμούς. Το ευρετήριο ιστού της Google φτάνει το 1 δισεκατομμύριο σελίδες σε διάστημα δύο ετών.

2010: Ο υπολογιστής επεξεργασίας φυσικής γλώσσας Watson της IBM νικά δύο πρώην πρωταθλητές στην τηλεοπτική εκπομπή "Jeopardy!" Ο αριθμός των χρηστών του διαδικτύου παγκοσμίως ξεπερνά τα 4 δισεκατομμύρια.

3.3 Εφαρμογές της Τεχνητής Νοημοσύνης

Η τεχνητή νοημοσύνη έχει τη δυνατότητα να αλλάξει τον τρόπο που ζούμε σήμερα και μπορεί να έχει καλό αντίκτυπο στη ζωή μας. Ας δούμε μερικά από τα ζωντανά παραδείγματα που έχουν ήδη υιοθετηθεί από τους ανθρώπους σήμερα:[11]

- **Τεχνητή Νοημοσύνη στο Ηλεκτρονικό Εμπόριο:** Η τεχνητή νοημοσύνη χρησιμοποιείται ευρέως στον τομέα του ηλεκτρονικού εμπορίου, καθώς βοηθά τον οργανισμό να δημιουργήσει μια καλή δέσμευση μεταξύ του χρήστη και της εταιρείας. Η τεχνητή νοημοσύνη βοηθά στην υποβολή κατάλληλων προτάσεων και συστάσεων σύμφωνα με το ιστορικό αναζήτησης χρήστη και τις προτιμήσεις προβολής. Υπάρχουν επίσης chatbots AI που χρησιμοποιούνται για την άμεση παροχή υποστήριξης πελατών και συμβάλλουν στη μείωση των παραπόνων και των ερωτημάτων σε μεγάλο βαθμό.
- **Τεχνητή νοημοσύνη στην εκπαίδευση:** Οι εκπαιδευτικοί τομείς είναι πλήρως οργανωμένοι και διοικούνται από την ανθρώπινη συμμετοχή μέχρι μερικά χρόνια πίσω. Αλλά αυτές τις μέρες, ο εκπαιδευτικός τομέας έρχεται επίσης υπό την επιρροή της Τεχνητής Νοημοσύνης. Βοηθά τη σχολή καθώς και τους μαθητές κάνοντας συστάσεις μαθημάτων, αναλύοντας ορισμένα δεδομένα και ορισμένες αποφάσεις σχετικά με τον μαθητή κ.λπ. Η δημιουργία αυτοματοποιημένων μηνυμάτων προς τους μαθητές και τους γονείς σχετικά με τυχόν διακοπές και τα αποτελέσματα των εξετάσεων γίνονται από την Τεχνητή Νοημοσύνη.
- **Τεχνητή Νοημοσύνη στη Ρομποτική:** Η τεχνητή νοημοσύνη είναι μία από τις σημαντικότερες τεχνολογίες που παρέχουν ώθηση, στον τομέα της ρομποτικής, για την αύξηση της αποτελεσματικότητάς τους. Η τεχνητή νοημοσύνη παρέχει στα ρομπότ τη δυνατότητα λήψης αποφάσεων σε πραγματικό χρόνο και αύξησης της παραγωγικότητας. Για παράδειγμα, ας υποθέσουμε ότι υπάρχει μια αποθήκη στην οποία χρησιμοποιούνται ρομπότ για τη διαχείριση καλών πακέτων. Τα

ρομπότ έχουν σχεδιαστεί μόνο για να παραδώσουν το έργο, αλλά η τεχνητή νοημοσύνη τα καθιστά ικανά να αναλύσουν τον κενό χώρο και να λάβουν την καλύτερη απόφαση σε πραγματικό χρόνο.

- **GPS και πλοήγηση:** Η τεχνολογία GPS χρησιμοποιεί τεχνητή νοημοσύνη για να κάνει την καλύτερη διαδρομή και να παρέχει την καλύτερη διαθέσιμη διαδρομή στους χρήστες για ταξίδια. Αυτό προκύπτει επίσης από έρευνα που παρέχεται από το Ινστιτούτο MIT ότι η τεχνητή νοημοσύνη είναι σε θέση να παρέχει ακριβείς, έγκαιρες και σε πραγματικό χρόνο πληροφορίες σχετικά με οποιαδήποτε συγκεκριμένη τοποθεσία. Βοηθά τον χρήστη να επιλέξει τον τύπο λωρίδας και δρόμων που αυξάνει τα χαρακτηριστικά ασφαλείας. Το GPS και η πλοήγηση χρησιμοποιούν το συνελκτικό και γραφικό νευρωνικό δίκτυο της Τεχνητής Νοημοσύνης για να παρέχουν αυτές τις προτάσεις.
- **Υγειονομική περίθαλψη:** Η τεχνητή νοημοσύνη χρησιμοποιείται ευρέως στον τομέα της υγειονομικής περίθαλψης και της ιατρικής. Οι διάφοροι αλγόριθμοι της Τεχνητής Νοημοσύνης χρησιμοποιούνται για την κατασκευή μηχανών ακριβείας που είναι σε θέση να ανιχνεύσουν μικρές ασθένειες μέσα στο ανθρώπινο σώμα. Επίσης, η Τεχνητή Νοημοσύνη χρησιμοποιεί το ιατρικό ιστορικό και την τρέχουσα κατάσταση ενός συγκεκριμένου ανθρώπου για να προβλέψει μελλοντικές ασθένειες. Η τεχνητή νοημοσύνη χρησιμοποιείται επίσης για την εύρεση των σημερινών κενών κλινών στα νοσοκομεία μιας πόλης που εξοικονομεί χρόνο για ασθενείς που βρίσκονται σε συνθήκες έκτακτης ανάγκης
- **Αυτοκίνητα:** Η τεχνητή νοημοσύνη φέρνει επαναστατικές αλλαγές στον τομέα των αυτοκινήτων. Από τα ταχύμετρα έως τα αυτο-οδηγούμενα αυτοκίνητα, η Τεχνητή Νοημοσύνη κάνει πραγματικά σημαντική διαφορά σε αυτούς τους τομείς. Η AI μηνύεται για να ανιχνεύσει την κυκλοφορία στο δρόμο και να παρέχει την καλύτερη διαδρομή από όλες τις παρούσες διαδρομές στον οδηγό. Χρησιμοποιεί αισθητήρες, τεχνολογία GPS και σήματα ελέγχου για να φέρει το όχημα την καλύτερη διαδρομή.
- **Γεωργία:** Η τεχνητή νοημοσύνη γίνεται επίσης μέρος της γεωργίας και της ζωής των αγροτών. Χρησιμοποιείται για την ανίχνευση διαφόρων παραμέτρων όπως η ποσότητα του νερού και της υγρασίας, η ποσότητα των ελλειπών θρεπτικών στοιχείων κ.λπ. στο έδαφος. Υπάρχει επίσης ένα μηχάνημα που χρησιμοποιεί τεχνητή νοημοσύνη για να ανιχνεύσει πού αναπτύσσονται τα ζιζάνια, πού το έδαφος είναι άγονο κ.λπ.
- **Ανθρώπινο Δυναμικό:** Όπως γνωρίζουμε, πολλές από τις διαδικασίες πρόσληψης γίνονται online αυτές τις μέρες. Οι ηλεκτρονικές διαδικασίες επιλογής γίνονται με τη χρήση της φωνητικής άδειας και της κάμερας της συσκευής του υποψηφίου. Εδώ η Τεχνητή Νοημοσύνη χρησιμοποιείται για να ανιχνεύσει κάθε είδους συμπεριφορά αθέμιτης πρακτικής και πολλά άλλα πράγματα. Χρησιμοποιείται επίσης για την ανίχνευση της προσωπικότητας οποιουδήποτε υποψηφίου σε ορισμένες περιπτώσεις. Αυτό μειώνει την προσπάθεια της ομάδας προσλήψεων και ενισχύει επίσης την αποτελεσματικότητα της διαδικασίας επιλογής.
- **Τρόπος ζωής:** Η τεχνητή νοημοσύνη έχει μεγάλο αντίκτυπο στον τρόπο ζωής μας. Υπάρχουν διάφορες καθημερινές εφαρμογές που χρησιμοποιούμε. Μερικά παραδείγματα είναι τα φίλτρα ανεπιθύμητης αλληλογραφίας στο ταχυδρομείο, η ανίχνευση κλήσεων απάτης, το ξεκλείδωμα προσώπου του κινητού, οι αισθητήρες δακτυλικών αποτυπωμάτων στο κινητό και τους φορητούς υπολογιστές μας κ.λπ.
- **Μέσα κοινωνικής δικτύωσης:** Υπάρχουν διάφορες χρήσεις της Τεχνητής Νοημοσύνης στον τομέα των κοινωνικών μέσων. Ορισμένες πλατφόρμες μέσων κοινωνικής δικτύωσης όπως το Facebook, το Instagram κ.λπ. χρησιμοποιούν

τεχνητή νοημοσύνη για να δείξουν σχετικό περιεχόμενο στον χρήστη. Χρησιμοποιεί το ιστορικό αναζήτησης και το ιστορικό προβολής ενός χρήστη για την εμφάνιση σχετικού περιεχομένου.

3.4 Κατηγορίες Τεχνητής Νοημοσύνης

Υπάρχουν διαφορετικοί τύποι Τεχνητής Νοημοσύνης και ο καθένας έχει το δικό του επίπεδο πολυπλοκότητας, δυνατοτήτων και χρήσεων. Οι τρεις κύριοι τύποι τεχνητής νοημοσύνης είναι: [12]

3.4.1 Artificial Narrow Intelligence(Τεχνητή στενή νοημοσύνη (ANI))

Η τεχνητή στενή νοημοσύνη (ANI) έχει σχεδιαστεί για την εκτέλεση συγκεκριμένων εργασιών. Γνωστό και ως αδύναμο AI, το ANI δεν είναι σε θέση να μάθει πέρα από τις προγραμματισμένες δυνατότητές του. Παραδείγματα στενής τεχνητής νοημοσύνης περιλαμβάνουν:

- Η Alexa της Amazon και η Siri της Apple, οι οποίες χρησιμοποιούν αναγνώριση ομιλίας.
- Αυτο-οδηγούμενα αυτοκίνητα, όπως αυτά που κατασκευάστηκαν από την Tesla του Elon Musk, τα οποία χρησιμοποιούν αναγνώριση όρασης και επεξεργασία εικόνας AI.
- Netflix και άλλες πλατφόρμες streaming, οι οποίες χρησιμοποιούν δεδομένα χρηστών για την παροχή προσαρμοσμένων προτάσεων.
- Υπολογιστική όραση, την οποία η Microsoft ορίζει ως έναν τύπο AI που επιτρέπει στις μηχανές να αναγνωρίζουν και να κατανοούν αντικείμενα και ανθρώπους σε εικόνες και βίντεο. Επιδιώκει να αναπαράγει τον τρόπο με τον οποίο βλέπει ο ανθρώπινος εγκέφαλος - και τον τρόπο με τον οποίο το ανθρώπινο μυαλό κατανοεί αυτό που βλέπει.

Το ANI χρησιμοποιείται επίσης σε διάφορους επαγγελματικούς τομείς, όπως:

- Υγειονομική περίθαλψη, όπου μπορεί να αναλύσει ιατρικά αρχεία και να βοηθήσει στη διάγνωση ασθενειών.
- Οικονομικών, όπου μπορεί να εντοπίσει απάτη.
- Μεταποίηση, όπου μπορεί να βελτιστοποιήσει την παραγωγή μέσω αυτοματισμού.
- Εξυπηρέτηση πελατών, όπου μπορεί να απαντήσει σε συχνές ερωτήσεις.
- Ασφάλεια, όπου μπορεί να χρησιμοποιήσει τεχνολογία αναγνώρισης προσώπου.
- Επιστήμη και ανάλυση δεδομένων, με συστήματα τεχνητής νοημοσύνης που βασίζονται στη μηχανική μάθηση και διασχίζουν μεγάλα σύνολα δεδομένων για να βοηθήσουν στην κατηγοριοποίηση και την ανάλυση τεράστιων ποσοτήτων δεδομένων.

3.4.2 Τεχνητή γενική νοημοσύνη (Artificial General Intelligence (AGI))

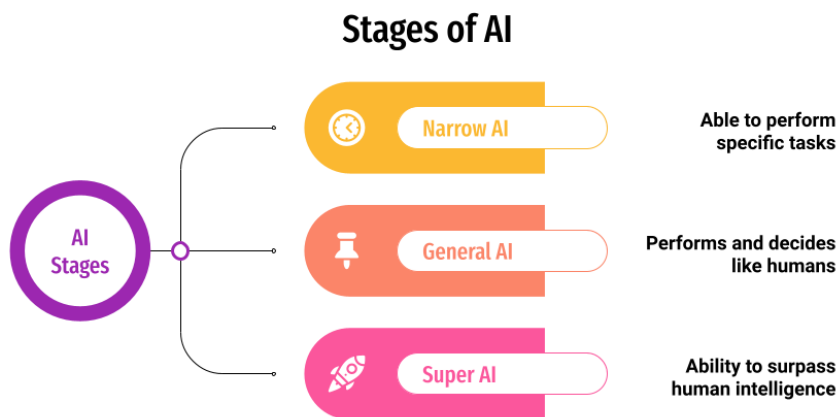
Η τεχνητή γενική νοημοσύνη (AGI) στοχεύει στην εκτέλεση πνευματικών εργασιών με τον τρόπο που μπορεί ένας άνθρωπος. Γνωστό και ως ισχυρό AI, το AGI στοχεύει να μάθει και να προσαρμοστεί σε νέες καταστάσεις, όπως θα έκανε ένα άτομο, και να μην περιορίζεται σε ένα συγκεκριμένο έργο ή τομέα. Αντ' αυτού, θα πρέπει να εφαρμοστεί σε διάφορους τομείς. Η γενική τεχνητή νοημοσύνη έχει πιθανές εφαρμογές στη ρομποτική, όπου οι μηχανές μπορούν να σκέφτονται και να λαμβάνουν αποφάσεις από μόνες τους

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

,καθιστώντας τες πιο αποτελεσματικές και παραγωγικές, αλλά θα μπορούσε επίσης να φέρει επανάσταση στις βιομηχανίες από την υγειονομική περίθαλψη έως τις μεταφορές.

3.4.3 Τεχνητή υπερνοημοσύνη (Artificial Super Intelligence (ASI))

Η τεχνητή υπερνοημοσύνη (ASI) λειτουργεί πέρα από τη νοημοσύνη σε ανθρώπινο επίπεδο, ικανή να ξεπεράσει τα ανθρώπινα όντα δυνητικά σε κάθε τομέα γνώσης και δραστηριότητας. Ωστόσο, επί του παρόντος, είναι μια υποθετική έννοια, διότι κανένα σύστημα δεν έχει ακόμη επιτύχει την ASI. Παρ'όλα αυτά, είναι ένα θέμα πολλών συζητήσεων στον τομέα της τεχνητής νοημοσύνης. Οι υποστηρικτές της ASI πιστεύουν ότι έχει τη δυνατότητα να αλλάξει τον κόσμο όπως τον ξέρουμε, λύνοντας πολλά από τα πιο περίπλοκα προβλήματα του κόσμου, από την κλιματική αλλαγή έως την εξάλειψη ασθενειών. Άλλοι, ωστόσο, ανησυχούν για τους κινδύνους που συνδέονται με την ASI, όπως η απώλεια του ανθρώπινου ελέγχου επί των μηχανών, ακόμη και η πιθανότητα οι μηχανές να αποκτήσουν αυτογνωσία ή απειλή για την ανθρώπινη ύπαρξη.



Εικόνα 6: Κατηγορίες Τεχνητής Νοημοσύνης[50]

3.5 Τύποι Τεχνητής Νοημοσύνης

1. **Reactive AI** χρησιμοποιεί αλγόριθμους για τη βελτιστοποίηση των εξόδων με βάση ένα σύνολο εισόδων. Τα AI που παίζουν σκάκι, για παράδειγμα, είναι αντιδραστικά συστήματα που βελτιστοποιούν την καλύτερη στρατηγική για να κερδίσετε το παιχνίδι. Η αντιδραστική τεχνητή νοημοσύνη τείνει να είναι αρκετά στατική, ανίκανη να μάθει ή να προσαρμοστεί σε νέες καταστάσεις. Έτσι, θα παράγει την ίδια έξοδο δεδομένων των ίδιων εισροών.
2. **Περιορισμένη μνήμη(Limited memory AI)** .Η τεχνητή νοημοσύνη μπορεί να προσαρμοστεί στην εμπειρία του παρελθόντος ή να ενημερωθεί με βάση νέες παρατηρήσεις ή δεδομένα. Συχνά, το ποσό της ενημέρωσης είναι περιορισμένο (εξ ου και το όνομα) και το μήκος της μνήμης είναι σχετικά μικρό. Τα αυτόνομα οχήματα, για παράδειγμα, μπορούν να «διαβάσουν το δρόμο» και να προσαρμοστούν σε νέες καταστάσεις, ακόμη και να «μάθουν» από την εμπειρία του παρελθόντος.
3. **Η θεωρία του νου AI(Theory-of-mind AI)** είναι πλήρως προσαρμοστική και έχει εκτεταμένη ικανότητα να μαθαίνει και να διατηρεί προηγούμενες εμπειρίες. Αυτοί οι τύποι Τεχνητής Νοημοσύνης περιλαμβάνουν προηγμένα chat-bots που θα μπορούσαν να περάσουν το τεστ Turing, ξεγελώντας ένα άτομο να πιστέψει ότι το AI

ήταν άνθρωπος. Αν και προηγμένες και εντυπωσιακές, αυτές οι AI δεν έχουν αυτογνωσία.

4. **Αυτογνωσιακή AI (Self-aware AI)**, όπως υποδηλώνει το όνομα, γίνεται αισθανόμενη και συνειδητοποιεί την ύπαρξή της. Ακόμα στη σφαίρα της επιστημονικής φαντασίας, ορισμένοι ειδικοί πιστεύουν ότι μια τεχνητή νοημοσύνη δεν θα γίνει ποτέ συνειδητή ή «ζωντανή». [13]

3.6 Κλάδοι της Τεχνητής Νοημοσύνης

3.6.1 Μηχανική Μάθηση (Machine Learning)

Η μηχανική μάθηση (ML) είναι ένας κλάδος της τεχνητής νοημοσύνης (AI) που παρέχει στις μηχανές τη δυνατότητα να μαθαίνουν αυτόματα από δεδομένα και προηγούμενες εμπειρίες, ενώ εντοπίζουν μοτίβα για να κάνουν προβλέψεις με ελάχιστη ανθρώπινη παρέμβαση. Οι μέθοδοι μηχανικής μάθησης επιτρέπουν στους υπολογιστές να λειτουργούν αυτόνομα χωρίς ρητό προγραμματισμό. Οι εφαρμογές Μηχανικής Μάθησης τροφοδοτούνται με νέα δεδομένα και μπορούν να μάθουν, να αναπτυχθούν και να προσαρμοστούν ανεξάρτητα. Η μηχανική μάθηση αντλεί διορατικές πληροφορίες από μεγάλους όγκους δεδομένων αξιοποιώντας αλγόριθμους για τον εντοπισμό προτύπων και τη μάθηση σε μια επαναληπτική διαδικασία. Οι αλγόριθμοι Μηχανικής Μάθησης χρησιμοποιούν υπολογιστικές μεθόδους για να μαθαίνουν απευθείας από δεδομένα αντί να βασίζονται σε οποιαδήποτε προκαθορισμένη εξίσωση που μπορεί να χρησιμεύσει ως μοντέλο. [14]

Κατηγορίες Αλγορίθμων Μηχανικής Μάθησης

- **Εποπτευόμενη μάθηση**

Στην εποπτευόμενη μάθηση, οι μεταβλητές εισόδου που αντιπροσωπεύονται ως X αντιστοιχίζονται σε μεταβλητές εξόδου που αντιπροσωπεύονται ως Y χρησιμοποιώντας έναν αλγόριθμο για να μάθουν τη συνάρτηση χαρτογράφησης f .

$$Y = f(X)$$

Ο στόχος του αλγορίθμου μάθησης είναι να προσεγγίσει τη συνάρτηση χαρτογράφησης για να προβλέψει την έξοδο (Y) για μια νέα είσοδο (X). Το σφάλμα από τις προβλέψεις που έγιναν κατά τη διάρκεια της εκπαίδευσης μπορεί να χρησιμοποιηθεί για τη διόρθωση της εξόδου. Η μάθηση μπορεί να σταματήσει όταν όλες οι είσοδοι εκπαιδευτούν για να πάρουν το στοχοθετημένο αποτέλεσμα. Η παλινδρόμηση χρησιμοποιείται για την επίλυση προβλημάτων παλινδρόμησης [15], υποστηρικτικών διανυσματικών μηχανών για ταξινόμηση [16] και τυχαίου δάσους για ταξινόμηση καθώς και προβλημάτων παλινδρόμησης.

- **Μάθηση χωρίς επίβλεψη**

Στη μάθηση χωρίς επίβλεψη, έχουμε μόνο τα δεδομένα εισόδου και καμία αντίστοιχη έξοδο για χάρτη. Αυτή η μάθηση στοχεύει να μάθει για τα δεδομένα μοντελοποιώντας την κατανομή των δεδομένων. Οι αλγόριθμοι μπορούν να είναι σε θέση να ανακαλύψουν τη συναρπαστική δομή που υπάρχει στα δεδομένα. Τα προβλήματα ομαδοποίησης και τα προβλήματα συσχέτισης χρησιμοποιούν μάθηση χωρίς επίβλεψη. Οι αλγόριθμοι μάθησης

χωρίς επίβλεψη, όπως ο αλγόριθμος K-means χρησιμοποιούνται σε προβλήματα ομαδοποίησης [17], και ο αλγόριθμος Apriori χρησιμοποιείται σε προβλήματα συσχέτισης.

- **Ενισχυτική μάθηση**

Η ενισχυτική μάθηση χρησιμοποιεί ένα σύστημα ανταμοιβής και τιμωρίας για την εκπαίδευση του αλγορίθμου. Σε αυτό, ο αλγόριθμος ή ένας μέσο μαθαίνει από το περιβάλλον του. Το μέσο λαμβάνει ανταμοιβές για σωστή απόδοση και ποινή για λανθασμένη απόδοση. Για παράδειγμα, σκεφτείτε την περίπτωση ενός αυτο-οδηγούμενου αυτοκινήτου: Το μέσο λαμβάνει ανταμοιβή για την ασφαλή οδήγηση στον προορισμό και ποινή για την έξοδο εκτός δρόμου. Ομοίως, στην περίπτωση ενός προγράμματος για να παίξετε σκάκι, η κατάσταση ανταμοιβής μπορεί να κερδίζει το παιχνίδι και η ποινή για να κάνετε ματ. Το μέσο προσπαθεί να μεγιστοποιήσει την ανταμοιβή και να ελαχιστοποιήσει την ποινή. Στην ενισχυτική μάθηση, ο αλγόριθμος δεν λέει πώς να εκτελέσει τη μάθηση. Ωστόσο, επιλύει το πρόβλημα από μόνο του [18].

- **Υβριδική μάθηση**

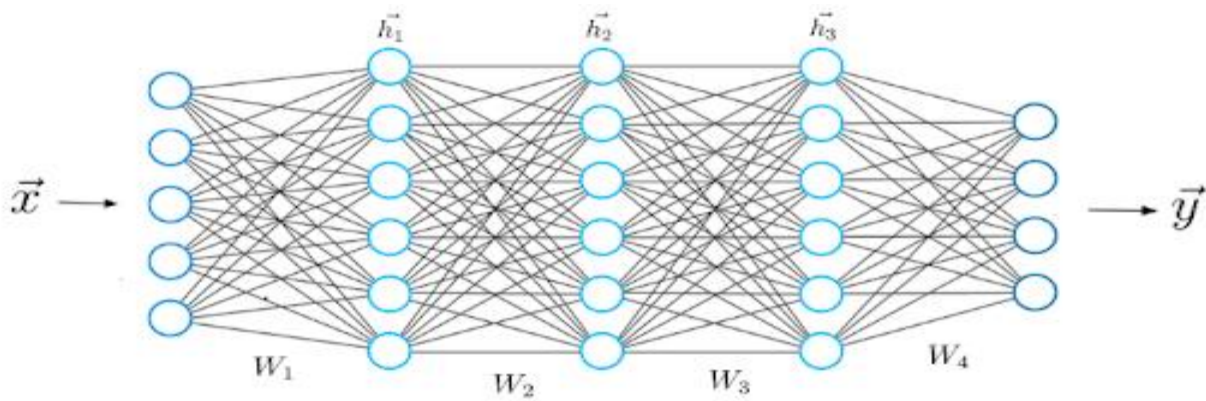
Η υβριδική μάθηση αναφέρεται σε αρχιτεκτονικές που χρησιμοποιούν παραγωγικές (χωρίς επίβλεψη) καθώς και διακριτικές (επιπτευόμενες) συνιστώσες. Ο συνδυασμός διαφορετικών αρχιτεκτονικών μπορεί να χρησιμοποιηθεί για το σχεδιασμό ενός υβριδικού με βάθος νευρωνικού δικτύου. Χρησιμοποιούνται για την αναγνώριση δράσης ανθρώπων ,χρησιμοποιώντας χαρακτηριστικά τράπεζας δράσης και αναμένοντας να παράγουν πολύ καλύτερα αποτελέσματα [19].

Δημοφιλής αλγόριθμοι Μηχανικής Μάθησης[20]

- Linear Regression
- Logistic Regression
- Support Vector Machine
- Decision Tree
- Random Forest
- Gradient Boosted Trees
- Artificial Neural Network
- Convolutional Neural Network

3.6.2 Deep Learning

Η βαθιά μάθηση είναι ένα υποσύνολο της μηχανικής μάθησης, το οποίο είναι ένα υποσύνολο της τεχνητής νοημοσύνης. Η τεχνητή νοημοσύνη είναι ένας γενικός όρος που αναφέρεται σε τεχνικές που επιτρέπουν στους υπολογιστές να μιμούνται την ανθρώπινη συμπεριφορά. Η μηχανική μάθηση αντιπροσωπεύει ένα σύνολο αλγορίθμων εκπαιδευμένων σε δεδομένα που καθιστούν όλα αυτά δυνατά. Η βαθιά μάθηση είναι απλώς ένας τύπος μηχανικής μάθησης, εμπνευσμένη από τη δομή του ανθρώπινου εγκεφάλου. [21] Οι αλγόριθμοι βαθιάς μάθησης προσπαθούν να βγάλουν παρόμοια συμπεράσματα με τους ανθρώπους αναλύοντας συνεχώς δεδομένα με δεδομένη λογική δομή. Για να επιτευχθεί αυτό, η βαθιά μάθηση χρησιμοποιεί μια πολυεπίπεδη δομή αλγορίθμων που ονομάζονται νευρωνικά δίκτυα. Στην Εικόνα 7 παρουσιάζεται ένα νευρωνικό δίκτυο με είσοδο x και έξοδο y .



Εικόνα 7: Νευρωνικό δίκτυο

Ο σχεδιασμός του νευρωνικού δικτύου βασίζεται στη δομή του ανθρώπινου εγκεφάλου. Ακριβώς όπως χρησιμοποιούμε τον εγκέφαλό μας για να αναγνωρίσουμε μοτίβα και να ταξινομήσουμε διαφορετικούς τύπους πληροφοριών, μπορούμε να διδάξουμε στα νευρωνικά δίκτυα να εκτελούν τις ίδιες διεργασίες στα δεδομένα.

Τα επιμέρους στρώματα των νευρωνικών δικτύων μπορούν επίσης να θεωρηθούν ως ένα είδος φίλτρου που λειτουργεί από γενικό προς το λεπτομερές, γεγονός που αυξάνει την πιθανότητα ανίχνευσης και εξαγωγής σωστού αποτελέσματος. Ο ανθρώπινος εγκέφαλος λειτουργεί παρόμοια. Κάθε φορά που λαμβάνουμε νέες πληροφορίες, ο εγκέφαλος προσπαθεί να τις συγκρίνει με γνωστά αντικείμενα. Η ίδια έννοια χρησιμοποιείται επίσης από βαθιά νευρωνικά δίκτυα.[22]

Τα νευρωνικά δίκτυα μας δίνουν τη δυνατότητα να εκτελούμε πολλές εργασίες, όπως ομαδοποίηση, ταξινόμηση ή παλινδρόμηση.

3.6.3 Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing)

Κύριες μέθοδοι Επεξεργασία Φυσικής Γλώσσας(NLP)

Σε γενικές γραμμές, μπορούμε να διακρίνουμε δύο πτυχές που είναι απαραίτητες για οποιοδήποτε πρόβλημα NLP[23][24]:

1. Η φάση της προεπεξεργασίας από το κείμενο στα δεδομένα

Ας υποθέσουμε ότι θέλετε να είστε σε θέση να προσδιορίσετε εάν ένα μήνυμα ηλεκτρονικού ταχυδρομείου είναι ανεπιθύμητο ή όχι, με βάση μόνο το περιεχόμενό του. Για να γίνει αυτό, είναι απαραίτητο να μετατρέψετε τα ακατέργαστα δεδομένα (το κείμενο του μηνύματος ηλεκτρονικού ταχυδρομείου) σε χρησιμοποιήσιμα δεδομένα.

Τα κύρια βήματα περιλαμβάνουν:

- **Καθαρισμός:** Ποικίλλει ανάλογα με την πηγή των δεδομένων, αυτή η φάση αποτελείται από την εκτέλεση εργασιών όπως η αφαίρεση διευθύνσεων URL, emoji κ.λπ.
- **Κανονικοποίηση δεδομένων:** Tokenization ή διάσπαση του κειμένου σε πολλά κομμάτια που ονομάζονται διακριτικά. Παράδειγμα: "Θα βρείτε συνημμένο το εν λόγω έγγραφο". "Εσείς", "θα βρείτε", "συνημμένο", "το έγγραφο", "εν λόγω".

- **Προέλευση:** η ίδια λέξη μπορεί να βρεθεί σε διαφορετικές μορφές ανάλογα με το φύλο (αρσενικό, θηλυκό), τον αριθμό (ενικός, πληθυντικός), το άτομο (εγώ, εσύ, αυτοί...) κ.λπ. Το Stemming αναφέρεται γενικά στην ακατέργαστη ευρετική διαδικασία αποκοπής του τέλους των λέξεων προκειμένου να διατηρηθεί μόνο η ρίζα της λέξης.
- **Λημματοποίηση:** αυτό συνίσταται στην εκτέλεση της ίδιας εργασίας αλλά χρησιμοποιώντας ένα λεπτόκοκκο λεξιλόγιο και ανάλυση κατασκευής λέξεων. Η λημματοποίηση επιτρέπει την αφαίρεση μόνο των άκαμπτων απολήξεων και έτσι την απομόνωση της κανονικής μορφής της λέξης, γνωστής ως λήμμα.
- **Άλλες λειτουργίες:** διαγραφή αριθμών, σημείων στίξης, συμβόλων και λέξεων διακοπής, αλλαγή σε πεζά.

Το "γλωσσικό" μέρος, το οποίο συνίσταται στην προεπεξεργασία και τη μετατροπή των πληροφοριών εισόδου σε ένα χρησιμοποιήσιμο σύνολο δεδομένων.

Το τμήμα "μηχανική μάθηση" ή "επιστήμη δεδομένων", το οποίο συνίσταται στην εφαρμογή μοντέλων μηχανικής μάθησης ή βαθιάς μάθησης σε αυτό το σύνολο δεδομένων.

Term Frequency-Inverse Document Frequency (TF-IDF): αυτή η μέθοδος συνίσταται στην καταμέτρηση του αριθμού των εμφανίσεων διακριτικών που υπάρχουν στο σώμα κειμένων για κάθε κείμενο, το οποίο στη συνέχεια διαιρείται με τον συνολικό αριθμό εμφανίσεων αυτών των ίδιων διακριτικών σε ολόκληρο το σώμα.

Για τον όρο x που υπάρχει στο έγγραφο y , μπορούμε να ορίσουμε το βάρος του με την ακόλουθη σχέση:

$$w_{x,y} = tf_{x,y} \cdot \log\left(\frac{N}{df_x}\right)$$

Πού:

$tf_{x,y}$ είναι η συχνότητα του όρου x στο y ·

df_x είναι ο αριθμός των εγγράφων που περιέχουν x ·

N είναι ο συνολικός αριθμός εγγράφων.

Έτσι, αυτή η προσέγγιση παρέχει μια διανυσματική αναπαράσταση για κάθε κείμενο που περιλαμβάνει διανύσματα βαρών και όχι εμφανίσεων.

Η αποτελεσματικότητα αυτών των μεθόδων διαφέρει ανάλογα με την περίπτωση εφαρμογής. Ωστόσο, έχουν δύο κύριους περιορισμούς:

- Όσο πιο πλούσιο είναι το λεξιλόγιο του σώματος κειμένων, τόσο μεγαλύτερο είναι το μέγεθος των διανυσμάτων, πράγμα που μπορεί να αποτελέσει πρόβλημα για τα μοντέλα μάθησης που χρησιμοποιούνται στο επόμενο βήμα.
- Η καταμέτρηση της εμφάνισης λέξεων δεν επιτρέπει τη λογιστική για την διάταξή τους και για το νόημα των προτάσεων.

Υπάρχει μια άλλη προσέγγιση που επιτρέπει την αντιμετώπιση αυτών των προβλημάτων: η Ενσωμάτωση Λέξεων (Word Embedding), η οποία αποτελείται από τη δημιουργία διανυσμάτων σταθερού μεγέθους που λαμβάνουν υπόψη το πλαίσιο στο οποίο βρίσκονται οι λέξεις. Έτσι, δύο λέξεις που βρίσκονται σε παρόμοια πλαίσια θα έχουν κοντινότερα διανύσματα (όσον αφορά τη διανυσματική απόσταση). Αυτό μας επιτρέπει να αποτυπώσουμε τόσο τις σημασιολογικές, συντακτικές ή θεματικές ομοιότητες των λέξεων.

2. Η φάση μάθησης: από τα δεδομένα στο μοντέλο

Συνολικά, μπορούμε να διακρίνουμε 3 κύριες προσεγγίσεις NLP: μεθόδους βασισμένες σε κανόνες, κλασικά μοντέλα μηχανικής μάθησης και μοντέλα βαθιάς μάθησης.

Μέθοδοι που βασίζονται σε κανόνες: Οι μέθοδοι που βασίζονται σε κανόνες βασίζονται σε μεγάλο βαθμό στην ανάπτυξη ειδικών κανόνων τομέα (π.χ. κανονικές εκφράσεις). Μπορούν να χρησιμοποιηθούν για την επίλυση απλών προβλημάτων, όπως η εξαγωγή δομημένων δεδομένων από μη δομημένα δεδομένα (π.χ. ιστοσελίδες).

Στην περίπτωση ανίχνευσης ανεπιθύμητων μηνυμάτων, αυτό θα μπορούσε να συνιστάται στην εξέταση ως ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου, εκείνων που περιέχουν τσιτάτα όπως "προώθηση", "περιορισμένη προσφορά" κ.λπ. Ωστόσο, αυτές οι απλές μέθοδοι μπορούν γρήγορα να κατακλυστούν από την πολυπλοκότητα της φυσικής γλώσσας και να αποδειχθούν αναποτελεσματικές.

Μοντέλα βαθιάς μάθησης: Η χρήση μοντέλων βαθιάς μάθησης για προβλήματα NLP αποτελεί επί του παρόντος αντικείμενο πολλών ερευνών. Αυτά τα μοντέλα γενικεύονται ακόμη καλύτερα από τις κλασικές προσεγγίσεις μάθησης, επειδή απαιτούν μια λιγότερο εξελιγμένη φάση προεπεξεργασίας κειμένου: τα νευρωνικά στρώματα μπορούν να θεωρηθούν ως αυτόματοι εξωλκείς χαρακτηριστικών. Αυτό καθιστά δυνατή τη δημιουργία μοντέλων από άκρο σε άκρο με μικρή προεπεξεργασία δεδομένων. Εκτός από το τμήμα μηχανικής χαρακτηριστικών, οι δυνατότητες μάθησης των αλγορίθμων Deep Learning είναι γενικά πιο ισχυρές από αυτές της κλασικής Μηχανικής Μάθησης, γεγονός που επιτρέπει την απόκτηση καλύτερων βαθμολογιών σε διάφορες πολύπλοκες σκληρές εργασίες NLP, όπως η μετάφραση.

3.6.4 Robotics

Η ρομποτική είναι ένας κλάδος της τεχνητής νοημοσύνης, που περιλαμβάνει Ηλεκτρολόγους Μηχανικούς, Μηχανολόγους Μηχανικούς και Επιστήμονες Υπολογιστών για το σχεδιασμό, την κατασκευή και την εφαρμογή ρομπότ. Ένα έξυπνο ρομπότ είναι ένας φυσικά τοποθετημένος Ευφυής πράκτορας που περιέχει 5 κύρια συστατικά: τελεστές, αντίληψη, έλεγχος, επικοινωνίες και ισχύς [9]. Οι τελεστές είναι τα περιφερειακά του ρομπότ που το βοηθάνε να κινηθεί και να αλληλοεπιδράσει με το περιβάλλον. Αντίληψη είναι ένα σύνολο αισθητήρων που παρέχουν στο ρομπότ την ικανότητα αίσθησης του περιβάλλοντος. Ο έλεγχος είναι ανάλογος στο κεντρικό νευρικό σύστημα και είναι ικανό για υπολογισμούς που επιτρέπουν στο ρομπότ να μεγιστοποιήσει τις πιθανότητες επιτυχίας του. Η επικοινωνία είναι ο τρόπος με τον οποίο ένα ρομπότ αλληλοεπιδρά με άλλους Πράκτορες όπως οι άνθρωποι χρησιμοποιούν γλώσσα, χειρονομίες και εγγύτητα να αλληλοεπιδρούν μεταξύ τους.[25]

3.6.5 Expert Systems

Έμπειρο σύστημα, ένα πρόγραμμα υπολογιστή που χρησιμοποιεί μεθόδους τεχνητής νοημοσύνης για την επίλυση προβλημάτων σε έναν εξειδικευμένο τομέα που συνήθως απαιτεί ανθρώπινη εμπειρογνωμοσύνη. Το πρώτο έμπειρο σύστημα αναπτύχθηκε το 1965 από τους Edward Feigenbaum και Joshua Lederberg του Πανεπιστημίου του Στάνφορντ στην Καλιφόρνια των Η.Π.Α. Το Dendral, όπως έγινε αργότερα γνωστό το έμπειρο σύστημά τους, σχεδιάστηκε για την ανάλυση χημικών ενώσεων. Έμπειρα συστήματα έχουν τώρα

εμπορικές εφαρμογές σε τομείς τόσο διαφορετικούς όσο η ιατρική διάγνωση, μηχανική πετρελαίου, και χρηματοοικονομικές επενδύσεις. Προκειμένου να επιτευχθούν κατορθώματα φαινομενικής νοημοσύνης, ένα έμπειρο σύστημα βασίζεται σε δύο συνιστώσες: μια βάση γνώσεων και μια μηχανή συμπερασμάτων. Μια βάση γνώσεων είναι μια οργανωμένη συλλογή γεγονότων σχετικά με τον τομέα του συστήματος. Μια μηχανή συμπερασμάτων ερμηνεύει και αξιολογεί τα γεγονότα στη βάση γνώσεων προκειμένου να δώσει μια απάντηση. Τυπικές εργασίες για έμπειρα συστήματα περιλαμβάνουν ταξινόμηση, διάγνωση, παρακολούθηση, σχεδιασμό, προγραμματισμό και προγραμματισμό για εξειδικευμένες προσπάθειες.[26]

3.6.6 Fuzzy Logic

Η ασαφής λογική (FL) είναι μια μέθοδος συλλογισμού που μοιάζει με την ανθρώπινη λογική. Η προσέγγιση του FL μιμείται τον τρόπο λήψης αποφάσεων των ανθρώπων, περιλαμβάνει όλες τις ενδιάμεσες δυνατότητες μεταξύ ψηφιακών ΝΑΙ και ΟΧΙ. Η FL βασίζεται στην ιδέα ότι δεν υπάρχει απότομη διάκριση μεταξύ των δύο άκρων. Η FL είναι μια μέθοδος συλλογιστική που εφαρμόζεται για τη λήψη αποφάσεων μέσω: ορισμένων κανόνων που συνδυάζονται μεταξύ τους για να παράγουν ένα αποτέλεσμα. Οι κανόνες είναι ασαφή σύνολα, τα οποία χρησιμοποιούνται ως βάση για τη λήψη αποφάσεων.[27]

3.6.7 Neural Network

Τα νευρωνικά δίκτυα αντιπροσωπεύουν μια κατηγορία αλγορίθμων μηχανικής μάθησης που έχουν δημιουργηθεί για να μιμούνται τον μηχανισμό επεξεργασίας πληροφοριών του ανθρώπινου εγκεφάλου. Περιλαμβάνουν πολυάριθμους διασυνδεδεμένους κόμβους, παρόμοιους με τεχνητούς νευρώνες, αντλώντας έμπνευση από βιολογικούς ομολόγους. Οι δια-νευρωνικές συνδέσεις φέρουν συγκεκριμένα βάρη, επιτρέποντας στο δίκτυο να μάθει από τα παρεχόμενα δεδομένα. Αυτά τα δίκτυα κατασκευάζονται με στρώματα διασυνδεδεμένων κόμβων ή τεχνητών νευρώνων, όπου κάθε κόμβος λαμβάνει εισόδους, επεξεργάζεται τα δεδομένα και τα μεταφέρει στο επόμενο στρώμα. Αυτή η επεξεργασία γίνεται μέσω λειτουργιών ενεργοποίησης, οι οποίες καθορίζουν εάν ένας νευρώνας πρέπει να «πυροδοτήσει» και να περάσει το σήμα του[28]. Μέσα σε ένα νευρωνικό δίκτυο, οι πληροφορίες διασχίζουν τα στρώματα, υποβάλλονται σε μετασχηματισμούς σε κάθε στάδιο, καταλήγοντας σε μια έξοδο που στη συνέχεια ταιριάζει με το επιδιωκόμενο αποτέλεσμα. Εάν υπάρχει διαφορά μεταξύ των δύο, το δίκτυο προσαρμόζει τα βάρη του που υπάρχουν στα κρυμμένα στρώματα, τα οποία μπορούν να θεωρηθούν ως δύναμη συνδέσεων για τη βελτίωση της απόδοσής του. Αυτή η διαδικασία προσαρμογής των βαρών και των κατωφλίων είναι γνωστή ως οπισθοπολλαπλασιασμός και αποτελεί σημαντικό μέρος της διαδικασίας κατάρτισης. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου το δίκτυο μπορέσει να χειριστεί με ακρίβεια τα δεδομένα [29]

3.6.8 Computer Vision

Ένας από τους πιο δημοφιλείς κλάδους της τεχνητής νοημοσύνης αυτή τη στιγμή, η υπολογιστική όραση, στοχεύει στην ανάπτυξη τεχνικών που βοηθούν τους υπολογιστές να βλέπουν και να κατανοούν ψηφιακές εικόνες και βίντεο. Η εφαρμογή μοντέλων μηχανικής μάθησης σε εικόνες επιτρέπει στους υπολογιστές να αναγνωρίζουν αντικείμενα, πρόσωπα, ανθρώπους, ζώα και πολλά άλλα. Τα αλγοριθμικά μοντέλα βοηθούν τους υπολογιστές να διδάξουν τον εαυτό τους σχετικά με τα περιβάλλοντα των οπτικών δεδομένων και με αρκετά δεδομένα που τροφοδοτούνται μέσω ενός μοντέλου, οι υπολογιστές μπορούν να διδάξουν

τον εαυτό τους να διακρίνουν μια εικόνα από μια άλλη. Ένα συνελικτικό νευρωνικό δίκτυο λειτουργεί παράλληλα με ένα μοντέλο για να σπάσει τις εικόνες σε εικονοστοιχεία, δίνοντάς τους ετικέτες. Στη συνέχεια, το νευρωνικό δίκτυο χρησιμοποιεί τις ετικέτες για να διεξάγει συνελίξεις, η οποία είναι μια μαθηματική πράξη σε δύο συναρτήσεις για να παράγει μια τρίτη συνάρτηση και να κάνει προβλέψεις για το τι βλέπει. Η υπολογιστική όραση έχει εφαρμογές σε διάφορους κλάδους, όπως:[30]

- Παρακολούθηση ή παρακολούθηση εντοπισμένων αντικειμένων.
- Ταξινόμηση εικόνας. Μια εικόνα ταξινομείται και προβλέπεται με ακρίβεια ότι ανήκει σε ορισμένες κατηγορίες.
- Αναγνώριση προσώπου. Το ξεκλείδωμα προσώπου σε smartphone ξεκλειδώνει συσκευές χαρτογραφώντας και αντιστοιχίζοντας τα χαρακτηριστικά του προσώπου.

4. ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΣΤΟΝ ΕΛΕΓΧΟ ΛΟΓΙΣΜΙΚΟΥ

4.1 Ο ρόλος και ο αντίκτυπος της τεχνητής νοημοσύνης στον έλεγχο λογισμικού

Οι τομείς στους οποίους οι τεχνικές τεχνητής νοημοσύνης έχουν αποδειχθεί χρήσιμες στην έρευνα και την πρακτική έλεγχου λογισμικού, μπορούν να χαρακτηριστούν από τις εφαρμογές τους στον κύκλο ζωής δοκιμών λογισμικού (STLC). Από τον σχεδιασμό έως την αναφορά, οι τεχνικές AI έχουν κάνει κυρίαρχο αποτύπωμα σε όλα τα στάδια του STLC. Για να μελετήσουμε τον αντίκτυπο της τεχνητής νοημοσύνης στις δοκιμές λογισμικού, εντοπίσαμε δραστηριότητες δοκιμών ή πτυχές δοκιμών για τις οποίες έχει διεξαχθεί σημαντική και σημαντική έρευνα εφαρμόζοντας AI. Αυτές οι δραστηριότητες δοκιμών καλύπτουν το μεγαλύτερο μέρος της STLC.

- **Δημιουργία δοκιμαστικών περιπτώσεων (Test Case Generation):** Έπειτα του ορισμού επαρκών κριτηρίων ελέγχου και δοκιμών είναι αρμοδιότητα των μηχανικών έλεγχου λογισμικού να διαμορφώσουν ένα σύνολο ελέγχων που θα πληροί όλα τα κριτήρια. Πλέον λόγω της πολυπλοκότητας των εφαρμογών, η χειροκίνητη δημιουργία σεναρίων είναι δύσκολο να διαχειριστεί και οι περισσότεροι μηχανικοί έλεγχου στρέφονται σε τεχνικές αυτοματοποιημένης παραγωγής σεναρίων. Τις τελευταίες δύο δεκαετίες, έχει παρουσιάζετε σημαντικά αυξανόμενο ενδιαφέρον για την εφαρμογή της τεχνητής νοημοσύνης στην αυτοματοποίηση των ελέγχων.
- **Παραγωγή δεδομένων δοκιμής (Test Data Generation):** Η παραγωγή δεδομένων δοκιμής είναι μια δραστηριότητα ή διαδικασία έλεγχου λογισμικού για τη δημιουργία δοκιμαστικών εισόδων και δεδομένα που βασίζονται σε λογικές περιπτώσεις και σεναρία δοκιμών. Αυτό είναι η ποιότητα των δεδομένων δοκιμών που καθορίζει την κάλυψη των δοκιμών ενός SUT.
- **Δοκιμή Oracle Construction (Test Oracle Construction):** Είναι η δραστηριότητα του ελέγχου κατά την οποία γίνεται επαλήθευση για την ορθή συμπεριφορά του STU με γνώμονα τις απαιτήσεις που έχουν οριστεί. Όταν ένα πρόγραμμα εκτελείτε με συγκεκριμένη είσοδο υπάρχει η ανάγκη ύπαρξης ενός μηχανισμού που θα μπορεί να διαχωρίζει τη σωστή από τη λανθασμένη συμπεριφορά. [57].
- **Προτεραιότητα δοκιμαστικής περίπτωσης (Test case prioritization):** Η προτεραιότητα των δοκιμαστικών περιπτώσεων περιλαμβάνει τη διάταξη της εκτέλεσης των δοκιμαστικών περιπτώσεων σε μια συγκεκριμένη σειρά έτσι ώστε να διασφαλιστεί ότι πολλαπλές εκτελέσεις δοκιμών πραγματοποιούνται με ποικίλους τρόπους με στόχο οι δοκιμαστικές περιπτώσεις που είναι πιο πιθανό να αποκαλύψουν ελαττώματα να εκτελούνται νωρίτερα στη διαδικασία δοκιμών. Οι δοκιμαστικές περιπτώσεις μπορούν επίσης να προτεραιοποιηθούν με βάση τον κίνδυνο, δηλαδή, τη σοβαρότητα του στοιχείου που ελέγχετε ή τον αντίκτυπο του κινδύνου. Τα διαθέσιμα εργαλεία δεν παρείχαν τη δυνατότητα να προτεραιοποιούνται αυτόματα οι δοκιμαστικές περιπτώσεις, ανοίγοντας τον δρόμο για τη μελέτη αυτού του τομέα.
- **Προδιαγραφή δοκιμής (Test Case Specification):** Στην αρχή του κύκλου ζωής του ελέγχου λογισμικού, οι δοκιμαστικές περιπτώσεις γράφονται με βάση τις λειτουργίες και τις απαιτήσεις του λογισμικού. Γράφονται σε ένα έγγραφο προδιαγραφής ελέγχου σε μορφή checklist για να διασφαλίζεται ότι κάθε απαίτηση του λογισμικού ελέγχετε. Περιλαμβάνει τον στόχο κάθε ελέγχου, προσδιορίζει τις απαιτούμενες εισόδους και

τα αναμενόμενα αποτελέσματα, παρέχει βήμα προς βήμα τις διαδικασίες για την εκτέλεση του ελέγχου και περιγράφει τα κριτήρια επιτυχίας/αποτυχίας για τον καθορισμό της αποδοχής. [32]

- **Βελτίωση περίπτωσης δοκιμής (Test Case Refinement):** Η βελτίωση των δοκιμαστικών περιπτώσεων είναι μια προγραμματισμένη δραστηριότητα που χρησιμοποιείται από τους μηχανικούς ελέγχου λογισμικού για την επιλογή των πιο αποτελεσματικών δοκιμαστικών περιπτώσεων για εκτέλεση, μειώνοντας έτσι το κόστος δοκιμής.
- **Εκτίμηση κόστους δοκιμής (Test Cost Estimation):** Η εκτίμηση κόστους λογισμικού είναι η διαδικασία πρόβλεψης της προσπάθειας που απαιτείται για τον έλεγχο ενός συστήματος λογισμικού. Η γενική πρακτική της ανάπτυξης λογισμικού είναι ότι δεν πρέπει να υπάρχουν ελλείψεις στην εκτίμηση του κόστους λογισμικού, όσο νωρίτερα γίνεται η εκτίμηση κόστους τόσο το καλύτερο για την ομάδα.

Πίνακας 1: Τεχνικές τεχνητής νοημοσύνης που εφαρμόζονται σε δραστηριότητες ελέγχου λογισμικού[32]

Software Testing Activity	AI Technique Applied
Test Case Generation	Inductive Learning - Active Learning - Ant colony Optimization - Markov Model - AI Planner - GA - Tabu Search - NLP - Re-enforcement Learning - C4.5 - Goal Based - Decision Tree - K-Nearest Neighbour - Logistic Regression - Random Forest - Multi-Layer Perceptron - K star - LSTM – Heuristic Search
Test Data Generation	GA - Simulated Annealing - Hill Climbing - Generative Model - LSTM – Deep Re-enforcement Learning - Ant Colony Optimization - Heuristic Methods
Test Oracle Construction	ANN - SVM - Decision Trees - AdaBoostM1 - Incremental Reduced Error Pruning (IREP) - Info Fuzzy Network
Test Case Prioritization	K-Means - Expectation-Maximization - c4.5 - Cob Web - Reinforcement Learning - CBR - ANN - Markov Model - K-NN - Logistic Regression – SVM Rank
Test Case Specification	IFN - C4.5
Test Case Refinement	IFN - Classification Tree Method
Test Cost Estimation	SVM - linear regression - k-NN - Naive Bayes - C4.5 - Random Forest – Multilayer Perceptron

4.2 Τα σημαντικά πλεονεκτήματα της τεχνητής νοημοσύνης στον έλεγχο λογισμικού.

- **Βελτιωμένη ακρίβεια:** Οι άνθρωποι μπορούν να κάνουν λάθη κάνοντας την ίδια μονότονη δουλειά καθημερινά, αλλά μια μηχανή δεν θα αποτύχει ποτέ να συλλάβει, να καταγράψει και να αναλύσει ακριβή δεδομένα με βελτιωμένη απόδοση. Οι μηχανικοί ελέγχου θα είναι απαλλαγμένοι από χειροκίνητες δοκιμές και μπορούν να χρησιμοποιήσουν αυτή τη φορά για την ανάπτυξη πιο προηγμένων και εξελιγμένων λειτουργιών δοκιμών AI. [33]

- **Εξοικονόμηση χρόνου και χρήματος:** Ο χειροκίνητος έλεγχος περιλαμβάνει επαναλαμβανόμενη εργασία κάθε φορά που τροποποιείται ο πηγαίος κώδικας. Είναι χρονοβόρα καθώς και δαπανηρή. Αντίθετα, μια λύση ελέγχων που βασίζεται σε AI μπορεί να εκτελεί αυτά τα βήματα συχνά και χωρίς επιπλέον κόστος. Το χρονικό διάστημα των δοκιμών λογισμικού μειώνεται και τροφοδοτεί επίσης τη βελτιστοποίηση του κόστους.
- **Μεγαλύτερη κάλυψη δοκιμών:** Με αυτοματοποιημένες δοκιμές που βασίζονται σε AI, μπορεί κανείς να αυξήσει το συνολικό βάθος και το εύρος των δοκιμών με αποτέλεσμα τη συνολική βελτίωση της ποιότητας του λογισμικού. Οι δοκιμές AI μπορούν να εμβαθύνουν στη μνήμη, τα περιεχόμενα αρχείων, τα εσωτερικά στατιστικά στοιχεία προγράμματος και τους πίνακες δεδομένων για να καταλάβουν τη βέλτιστη απόδοση του λογισμικού. Οι δοκιμές AI μπορούν να εκτελέσουν πολλές δοκιμές ταυτόχρονα, παρέχοντας πιο εκτεταμένη κάλυψη, κάτι που μπορεί να μην είναι δυνατό με χειροκίνητες δοκιμές.
- **Ενισχυμένη ανίχνευση ελαττωμάτων:** Στις συμβατικές και χειροκίνητες μεθόδους ελέγχου, τα σφάλματα παραμένουν απαρατήρητα για μεγάλο χρονικό διάστημα και δημιουργούν εμπόδια στο μέλλον. Η τεχνητή νοημοσύνη στις δοκιμές λογισμικού μπορεί να εντοπίσει ελαττώματα αυθόρμητα. Καθώς το λογισμικό μεγαλώνει, τα δεδομένα αυξάνονται και έτσι αυξάνεται ο αριθμός των σφαλμάτων. Η τεχνητή νοημοσύνη περιγράφει αυτά τα σφάλματα γρήγορα και αυτόματα, έτσι ώστε η ομάδα ανάπτυξης λογισμικού να μπορεί να λειτουργεί ομαλά. Η παρακολούθηση σφαλμάτων που βασίζεται σε AI εντοπίζει δακτυλικά αποτυπώματα αποτυχιών και αντιλαμβάνεται διπλά σφάλματα.
- **Βελτιωμένες δοκιμές παλινδρόμησης:** Με την προοδευτική και ταχεία ανάπτυξη, υπάρχει ταχύτερη απαίτηση δοκιμών παλινδρόμησης πέρα από την ανθρώπινη ικανότητα. Η τεχνητή νοημοσύνη μπορεί να εκτελέσει κουραστικά τεστ παλινδρόμησης. Οι οργανισμοί μπορούν να υιοθετήσουν τη μηχανική μάθηση για να δημιουργήσουν δοκιμαστικά σενάρια. Για παράδειγμα, σε μια αλλαγή διεπαφής χρήστη, μια λειτουργία που βασίζεται σε AI μπορεί να σαρώσει για τυχόν επικαλύψεις. Η τεχνητή νοημοσύνη μπορεί επίσης να χρησιμοποιηθεί για την επικύρωση των αλλαγών που διαφορετικά μπορεί να είναι δύσκολες στις χειροκίνητες δοκιμές.
- **Διεξαγωγή οπτικών δοκιμών:** Η λύση που βασίζεται σε AI βοηθά στην οπτική επικύρωση ιστοσελίδων και μπορεί να δοκιμάσει διαφορετικά περιεχόμενα στη διεπαφή χρήστη. Αυτές οι δοκιμές είναι δύσκολο να επικυρωθούν, καθώς απαιτεί ανθρώπινη συμμετοχή στην κρίση του σχεδιασμού. Οι δοκιμές αυτοματισμού μπορούν να τραβήξουν στιγμιότυπα οθόνης, να μετρήσουν το χρόνο φόρτωσης και πολλές άλλες λειτουργίες που διαφορετικά μπορεί να είναι δύσκολο για έναν μεμονωμένο άνθρωπο να επισημάνει. Οι δοκιμές AI εξαλείφουν τις μη αυτόματες δυσκολίες ενημέρωσης του μοντέλου αντικειμένου εγγράφου, δημιουργίας πλαισίου και σύνοψης κινδύνων.
- **Αυτοματοποιημένη παραγωγή δοκιμής API:** Ο αυτοματισμός δοκιμών API επιτρέπει στους χρήστες να αναπτύσσουν πολλαπλές περιπτώσεις δοκιμών για τη διασφάλιση ποιότητας API και να εκτιμούν τη λειτουργία πολλών εργαλείων τρίτων. Λίγες υπηρεσίες χρησιμοποιούν εκατοντάδες API, για τα οποία η αυτοματοποίηση είναι υποχρεωτική. Τα εργαλεία που βασίζονται σε AI έχουν σχεδιαστεί με τρόπο που αναλύουν τον μεγάλο όγκο δεδομένων και αξιολογούν γρήγορα εάν το API λειτουργεί σωστά ή όχι. Οι δοκιμές API στην ανάπτυξη προϊόντων διασφαλίζουν την ποιότητα επικοινωνίας μεταξύ προγραμμάτων που επικοινωνούν μέσω βάσεων δεδομένων και διακομιστών και χρησιμοποιούν διαφορετικά πρωτόκολλα.
- **Προγνωστική ανάλυση:** Οι δοκιμές τεχνητής νοημοσύνης μπορούν να χρησιμοποιήσουν τα υπάρχοντα δεδομένα πελατών και αναλυτικά δεδομένα για να καθορίσουν πώς θα εξελιχθούν οι απαιτήσεις και οι συμπεριφορές περιήγησης των

χρηστών στο μέλλον. Διασφαλίζει ότι οι μηχανικοί ελέγχου λογισμικού και οι προγραμματιστές είναι ένα βήμα μπροστά από τον χρήστη και τις απαιτήσεις του. Με λύσεις που βασίζονται στην τεχνητή νοημοσύνη, θα υπάρξει καλύτερη ποιότητα υπηρεσιών και βελτιωμένη πρόβλεψη των αυξανόμενων αναγκών.

- **Ενισχυμένη γραφή δοκιμαστικών περιπτώσεων:** Το AI θα βελτιώσει την ποιότητα των δοκιμαστικών σας περιπτώσεων για δοκιμές αυτοματισμού. Η τεχνητή νοημοσύνη θα προσφέρει πραγματικές δοκιμαστικές περιπτώσεις που είναι γρήγορες στη λειτουργία και εύκολες στη ρύθμιση. Η παραδοσιακή μέθοδος δεν επιτρέπει στους προγραμματιστές να αναλύσουν πρόσθετες δυνατότητες για δοκιμαστικές περιπτώσεις. Με τη βοήθεια της τεχνητής νοημοσύνης, η ανάλυση δεδομένων έργου πραγματοποιείται σε λίγα δευτερόλεπτα και επομένως θα επιτρέψει στους προγραμματιστές να καταλάβουν νέες προσεγγίσεις για δοκιμαστικές περιπτώσεις.
- **Αναδυόμενα και εξελισσόμενα bots:** Καθώς εκτελείται η δοκιμή, το AI ανιχνεύει την αλλαγή στον κώδικα. Με αλλαγές στον κώδικα, τα bots AI αναδύονται και εξελίσσονται μαθαίνοντας νέες λειτουργίες εφαρμογών. Ως αποτέλεσμα, τα ρομπότ AI μαθαίνουν ταυτόχρονα με αυτό και επομένως βελτιώνεται και η ποιότητα του προϊόντος.[33]

4.3 Τα μειονεκτήματα της τεχνητής νοημοσύνης στον έλεγχο λογισμικού

Εδώ είναι οι περιορισμοί που έχει και θα έχει η τεχνητή νοημοσύνη για σημαντικό χρονικό διάστημα.[34]

- **Στερείται δημιουργικότητας.** Οι αλγόριθμοι AI αντιμετωπίζουν μεγάλα προβλήματα στη δημιουργία δοκιμαστικών περιπτώσεων που εξετάζουν περιπτώσεις αιχμής ή απροσδόκητα σενάρια. Χρειάζονται βοήθεια με ασυνέπειες και γωνιακές καταστάσεις.
- **Εξαρτάται από τα δεδομένα εκπαίδευσης.** Η τεχνητή νοημοσύνη δεν είναι τίποτα άλλο παρά ένας αλγόριθμος, ένα μαθηματικό μοντέλο που τροφοδοτείται με δεδομένα για να λειτουργήσει. Δεν είναι μια δύναμη της φύσης ή ένα θέμα για φυσική ανάπτυξη. Έτσι, η ποιότητα των δοκιμαστικών περιπτώσεων που παράγονται από την τεχνητή νοημοσύνη εξαρτάται από την ποιότητα των δεδομένων που χρησιμοποιούνται για την εκπαίδευση των αλγορίθμων, η οποία μπορεί να είναι περιορισμένη ή μεροληπτική.
- **Χρειάζεται «τέλειες συνθήκες».** Η τεκμηρίωση του έργου δεν είναι σχεδόν καμία, οι περιπτώσεις χρήσης είναι ασαφείς και μη ρεαλιστικές και απλώς αποσπάζετε πληροφορίες από τον πελάτη σας. Η τεχνητή νοημοσύνη δεν μπορεί να το κάνει αυτό. Η ποιότητα της εργασίας της θα είναι ακριβώς τόσο καλή ή κακή όσο η ποιότητα των εισροών και του πλαισίου μετατρέπονται σε ποσοτικοποιήσιμα δεδομένα.
- **Έχει περιορισμένη κατανόηση του λογισμικού.** Τείνουμε να δίνουμε υπερδυνάμεις στην τεχνητή νοημοσύνη και στην κατανόησή της για τον κόσμο. Στην πραγματικότητα, είναι πραγματικά πολύ περιορισμένη προς το παρόν. Ενδέχεται να μην έχει βαθιά κατανόηση του λογισμικού που δοκιμάζεται, γεγονός που θα μπορούσε να οδηγήσει σε απώλεια σημαντικών σεναρίων ή ελαττωμάτων.
- **Απαιτεί εξειδικευμένους επαγγελματίες για να λειτουργήσει.** Για παράδειγμα, η ενσωμάτωση μιας στρατηγικής δοκιμών με αγωγούς CI/CD που λειτουργούν με AI μπορεί να είναι περίπλοκη για τη ρύθμιση, τη συντήρηση και την αντιμετώπιση προβλημάτων, καθώς απαιτεί προηγμένες τεχνικές δεξιότητες και γνώσεις αλγορίθμων και τεχνολογιών AI. Οι δοκιμασμένες και αληθινές μέθοδοι που

χρησιμοποιούμε τώρα μπορούν, για χρόνια, να παραμείνουν πολύ φθηνότερες και ευκολότερες στη συντήρηση.

4.4 Τεχνικές και προσεγγίσεις της τεχνητής νοημοσύνης στον έλεγχο λογισμικού.

Η τεχνητή νοημοσύνη (AI) μεταμορφώνει το τοπίο του ελέγχου λογισμικού, εισάγοντας νέες τεχνικές και προσεγγίσεις που ενισχύουν την αποδοτικότητα, την αποτελεσματικότητα και την κάλυψη. Τα εργαλεία και οι μέθοδοι δοκιμών που υποστηρίζονται από AI αποτελούν πλέον αναπόσπαστο μέρος των σύγχρονων πρακτικών ανάπτυξης λογισμικού, συμβάλλοντας στην παροχή προϊόντων λογισμικού υψηλής ποιότητας ταχύτερα από ποτέ. Ακολουθούν μερικές από τις βασικές τεχνικές AI που χρησιμοποιούνται στις δοκιμές λογισμικού:[35]

- **Αυτοματοποίηση δοκιμών με βάση τη μηχανική μάθηση:** Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται για την αυτοματοποίηση της δημιουργίας και εκτέλεσης δοκιμαστικών περιπτώσεων. Αυτοί οι αλγόριθμοι μπορούν να αναλύσουν ιστορικά δεδομένα δοκιμών, συμπεριφορά εφαρμογών και αλληλεπιδράσεις χρηστών για τον εντοπισμό πιθανών σεναρίων δοκιμής και τη δημιουργία αντίστοιχων δεσμών ενεργειών δοκιμής. Αυτό μειώνει σημαντικά τη χειρωνακτική προσπάθεια που απαιτείται για τη δημιουργία και τη συντήρηση δοκιμαστικών περιπτώσεων, ειδικά για πολύπλοκες και δυναμικές εφαρμογές.
- **Επεξεργασία φυσικής γλώσσας (NLP) για τη δημιουργία δοκιμαστικών περιπτώσεων:** Οι τεχνικές NLP [36] επιτρέπουν τη μετατροπή των περιγραφών φυσικής γλώσσας των δοκιμαστικών περιπτώσεων σε εκτελέσιμα δοκιμαστικά σενάρια. Αυτό επιτρέπει στους δοκιμαστές να εκφράσουν τις προθέσεις δοκιμών τους με πιο διαισθητικό και συνοπτικό τρόπο, μειώνοντας την ανάγκη για τεχνογνωσία προγραμματισμού. Τα εργαλεία NLP μπορούν επίσης να αναλύσουν την τεκμηρίωση και τις απαιτήσεις των χρηστών για την αυτόματη εξαγωγή σχετικών δοκιμαστικών περιπτώσεων, εξορθολογίζοντας περαιτέρω τη διαδικασία δοκιμής.
- **Οπτικός έλεγχος για UI Regression[37]:** Τα εργαλεία οπτικής δοκιμής χρησιμοποιούν αλγόριθμους αναγνώρισης εικόνας και μηχανικής μάθησης για τη σύγκριση στιγμιότυπων οθόνης διεπαφών εφαρμογών σε διαφορετικές εκδόσεις ή περιβάλλοντα. Αυτό βοηθά στον εντοπισμό οπτικών ανωμαλιών και διασφαλίζει ότι η διεπαφή χρήστη της εφαρμογής παραμένει συνεπής και λειτουργική καθ' όλη τη διάρκεια του κύκλου ζωής ανάπτυξης.
- **Ανίχνευση ανωμαλιών για δοκιμές απόδοσης και ασφάλειας[38]:** Τα εργαλεία ανίχνευσης ανωμαλιών που υποστηρίζονται από AI μπορούν να αναλύσουν αρχεία καταγραφής εφαρμογών, μετρήσεις απόδοσης και δεδομένα ασφαλείας για τον εντοπισμό ασυνήθιστων μοτίβων ή ακραίων τιμών που μπορεί να υποδεικνύουν πιθανά προβλήματα. Αυτό βοηθά τους μηχανικούς ελέγχου λογισμικού να εντοπίζουν προληπτικά και να αντιμετωπίζουν σημεία συμφόρησης επιδόσεων, ευπάθειες ασφαλείας και άλλα προβλήματα προτού επηρεάσουν τους χρήστες.
- **Αυτοθεραπευόμενο(Self-Healing) AI για συντήρηση δοκιμών[39]:** Τα εργαλεία AI αυτοθεραπείας μπορούν να προσαρμόσουν και να ενημερώσουν αυτόματα τα δοκιμαστικά σενάρια καθώς εξελίσσεται η υπό δοκιμή εφαρμογή. Αυτό αντιμετωπίζει την πρόκληση της συντήρησης δοκιμαστικών δεσμών ενεργειών, καθώς οι αλλαγές στον κώδικα ή τη δομή της εφαρμογής μπορούν να καταστήσουν τις υπάρχουσες δοκιμαστικές δέσμες ενεργειών παρωχημένες. Το AI αυτοθεραπείας διασφαλίζει ότι οι δοκιμαστικές περιπτώσεις παραμένουν σχετικές και αποτελεσματικές καθ' όλη τη διάρκεια του κύκλου ανάπτυξης.

- **Προτεραιότητα δοκιμών βάσει κινδύνου[40]:** Οι αλγόριθμοι AI μπορούν να αναλύσουν ιστορικά δεδομένα ελαττωμάτων και πολυπλοκότητα εφαρμογών για να δώσουν προτεραιότητα σε περιπτώσεις δοκιμών με βάση τον πιθανό κίνδυνο αποκάλυψης ελαττωμάτων. Αυτή η ιεράρχηση βοηθά τους μηχανικούς ελέγχου να εστιάσουν τις προσπάθειές τους στις πιο κρίσιμες περιοχές της εφαρμογής, μεγιστοποιώντας την αποτελεσματικότητα των προσπαθειών ελέγχου τους.
- **Αυτοματοποιημένη ανίχνευση και αναφορά ελαττωμάτων[41]:** Τα εργαλεία που υποστηρίζονται από AI μπορούν να αναλύσουν τα αποτελέσματα των δοκιμών, τα αρχεία καταγραφής εφαρμογών και τα σχόλια των χρηστών για τον αυτόματο εντοπισμό και την αναφορά πιθανών ελαττωμάτων. Αυτό μειώνει τη μη αυτόματη προσπάθεια που απαιτείται για τον εντοπισμό ελαττωμάτων και επιτρέπει στους δοκιμαστές να επικεντρωθούν στη διερεύνηση και την επίλυση αναφερόμενων ζητημάτων πιο αποτελεσματικά.
- **Διαχείριση περιβάλλοντος δοκιμών με τεχνολογία AI:** Η AI μπορεί να βελτιστοποιήσει την παροχή και τη διαμόρφωση του περιβάλλοντος δοκιμών, διασφαλίζοντας ότι τα σωστά περιβάλλοντα δοκιμών είναι διαθέσιμα όταν χρειάζεται. Αυτό απλοποιεί τη διαδικασία δοκιμής και μειώνει τις καθυστερήσεις που προκαλούνται από ζητήματα εγκατάστασης περιβάλλοντος και διαμόρφωσης.
- **Σχεδιασμός και εκτέλεση δοκιμών βάσει AI[36]:** Η τεχνητή νοημοσύνη μπορεί να βοηθήσει τους μηχανικούς ελέγχου στον προγραμματισμό δραστηριοτήτων δοκιμών, στη βελτιστοποίηση της κατανομής πόρων και στη δυναμική προσαρμογή της εκτέλεσης δοκιμών με βάση σχόλια και αποτελέσματα σε πραγματικό χρόνο. Αυτό βοηθά τους δοκιμαστές να λαμβάνουν τεκμηριωμένες αποφάσεις και να μεγιστοποιούν την αποτελεσματικότητα των προσπαθειών δοκιμών τους.

5. ΕΦΑΡΜΟΓΕΣ GENERATIVE AI , ΙΣΤΟΤΟΠΟΣ SAUDEMO ΚΑΙ ΠΛΑΙΣΙΟ ΑΝΟΙΧΤΟΥ ΚΩΔΙΚΑ PLAYWRIGHT

Στο συνεχώς εξελισσόμενο τοπίο της ανάπτυξης λογισμικού, η διασφάλιση της αξιοπιστίας και της λειτουργικότητας των εφαρμογών είναι πρωταρχικής σημασίας. Καθώς η πολυπλοκότητα των συστημάτων λογισμικού συνεχίζει να αυξάνεται, τόσο αυξάνεται η ανάγκη για αποτελεσματικές μεθοδολογίες ελέγχου. Οι παραδοσιακές μέθοδοι δημιουργίας σεναρίων ελέγχου συχνά δεν μπορούν να συμβαδίσουν με τους γρήγορους κύκλους ανάπτυξης. Στην εποχή της Τεχνητής Νοημοσύνης (AI), καινοτόμες λύσεις αναδιαμορφώνουν τον τρόπο με τον οποίο προσεγγίζουμε τις δοκιμές λογισμικού. Σε αυτό το κεφάλαιο, εξερευνούμε την πρωτοποριακή ιδέα της χρήσης τεχνητής νοημοσύνης για τη δημιουργία σεναρίων ελέγχου λογισμικού με βάση απαιτήσεις.

5.1 Η χειροκίνητη δημιουργία σεναρίων ελέγχου

Η δημιουργία παραδοσιακών σεναρίων ελέγχου συνήθως περιλαμβάνει μια χειροκίνητη διαδικασία, βασισμένη στην τεχνογνωσία των ανθρώπινων μηχανικών ελέγχου για τον εντοπισμό σεναρίων ελέγχου με βάση τις απαιτήσεις και τον κώδικα. Ωστόσο, αυτή η προσέγγιση έχει τους περιορισμούς της, όπως η πιθανότητα επίβλεψης, το ανθρώπινο λάθος και η αδυναμία αποτελεσματικού χειρισμού μεγάλων και πολύπλοκων εφαρμογών. Καθώς το λογισμικό γίνεται όλο και πιο περίπλοκο και δυναμικό, υπάρχει επιτακτική ανάγκη για μια πιο έξυπνη και αυτοματοποιημένη προσέγγιση για τη δημιουργία δοκιμαστικών περιπτώσεων.

Η χειροκίνητη δημιουργία σεναρίων ελέγχου είναι μια χρονοβόρα και επιρρεπής σε σφάλματα διαδικασία. Οι υπεύθυνοι ελέγχου πρέπει να εξετάζουν σχολαστικά τις απαιτήσεις λογισμικού, να σχεδιάζουν σενάρια ελέγχου και να δημιουργούν αντίστοιχες σεναρίων ελέγχου. Η πολυπλοκότητα του σύγχρονου λογισμικού, σε συνδυασμό με τις διαρκώς μεταβαλλόμενες απαιτήσεις, καθιστά αυτή τη διαδικασία ακόμη πιο δύσκολη. Επιπλέον, καθώς το λογισμικό εξελίσσεται μέσω διαφόρων φάσεων, η συντήρηση και η ενημέρωση των δοκιμαστικών περιπτώσεων καθίσταται δύσκολη εργασία. Η δημιουργία χειροκίνητων σεναρίων ελέγχου δεν απαιτεί μόνο πόρους, αλλά μπορεί επίσης να παραβλέψει πιθανές περιπτώσεις αιχμής και σενάρια που θα μπορούσαν να οδηγήσουν σε κρίσιμα ζητήματα στην παραγωγή. Η τεχνητή νοημοσύνη έρχεται να συνεισφέρει αυτοματοποιώντας τη διαδικασία δημιουργίας σεναρίων ελέγχου, αντιμετωπίζοντας αυτές τις προκλήσεις και ανοίγοντας το δρόμο για ένα πιο διεξοδικό και ολοκληρωμένο καθεστώς σεναρίων ελέγχου.

5.2 Ο ρόλος της Τεχνητής Νοημοσύνης στην παραγωγή σεναρίων ελέγχου

Η τεχνητή νοημοσύνη φέρνει ένα μετασχηματιστικό πλεονέκτημα στη δημιουργία δοκιμαστικών περιπτώσεων αξιοποιώντας προηγμένους αλγόριθμους και μοντέλα μηχανικής μάθησης. Εισάγει μια αλλαγή παραδείγματος στη διαδικασία δημιουργίας δοκιμαστικών περιπτώσεων, προσφέροντας τη δυνατότητα ενίσχυσης της κάλυψης των δοκιμών και επιτάχυνσης του κύκλου ζωής των δοκιμών. Ας εμβαθύνουμε στους τρόπους με τους οποίους η τεχνητή νοημοσύνη έχει ουσιαστικό αντίκτυπο στη δημιουργία περιπτώσεων δοκιμής προσβασιμότητας.[43]

1. **Αυτοματοποιημένη αναγνώριση βασικών στοιχείων προσβασιμότητας:**

Οι αλγόριθμοι τεχνητής νοημοσύνης μπορούν να αναλύσουν τη δομή ενός ψηφιακού στοιχείου και να προσδιορίσουν αυτόματα κρίσιμα στοιχεία προσβασιμότητας, όπως

πεδία φόρμας, εικόνες και διαδραστικά στοιχεία. Αυτή η αυτοματοποιημένη αναγνώριση βελτιστοποιεί τη διαδικασία δημιουργίας δοκιμαστικών περιπτώσεων, διασφαλίζοντας ότι οι υπεύθυνοι δοκιμών επικεντρώνονται στις πιο σημαντικές πτυχές της προσβασιμότητας.

2. Μάθηση από τις αλληλεπιδράσεις χρηστών:

Τα συστήματα τεχνητής νοημοσύνης μπορούν να εκπαιδευτούν σε τεράστια σύνολα δεδομένων αλληλεπιδράσεων χρηστών, επιτρέποντάς τους να κατανοήσουν πώς οι χρήστες με διαφορετικές ικανότητες προηγούνται και αλληλοεπιδρούν με ψηφιακές διεπαφές. Στη συνέχεια, αυτή η γνώση εφαρμόζεται για τη δημιουργία δοκιμαστικών περιπτώσεων που μιμούνται σενάρια χρήσης του πραγματικού κόσμου, ενισχύοντας τον ρεαλισμό και την αποτελεσματικότητα των δοκιμών προσβασιμότητας.

3. Προσαρμογή δυναμικής περίπτωσης δοκιμής:

Καθώς εξελίσσονται τα ψηφιακά στοιχεία, τα συστήματα που βασίζονται σε τεχνητή νοημοσύνη μπορούν να προσαρμόσουν δυναμικά τις δοκιμαστικές περιπτώσεις ώστε να αντικατοπτρίζουν τις αλλαγές στη διεπαφή χρήστη. Αυτή η προσαρμοστικότητα διασφαλίζει ότι η δοκιμή προσβασιμότητας παραμένει ευθυγραμμισμένη με τις πιο πρόσφατες επαναλήψεις σχεδιασμού και ανάπτυξης, μειώνοντας τον κίνδυνο παράβλεψης κρίσιμων ζητημάτων που εισάγονται κατά τις ενημερώσεις.

4. Αποτελεσματική κάλυψη διαφορετικών περιπτώσεων χρήσης:

Η τεχνητή νοημοσύνη υπερέχει στον χειρισμό της πολυπλοκότητας, καθιστώντας την ικανή στη δημιουργία δοκιμαστικών υποθέσεων που καλύπτουν ένα ευρύ φάσμα σεναρίων. Από διαφορετικές υποστηρικτικές τεχνολογίες έως παραλλαγές στις ικανότητες των χρηστών, η τεχνητή νοημοσύνη μπορεί να δημιουργήσει περιπτώσεις δοκιμών που αντιμετωπίζουν πλήρως τις διαφορετικές ανάγκες των χρηστών με αναπηρίες, οδηγώντας τελικά σε πιο ισχυρές δοκιμές προσβασιμότητας.

5. Επιτάχυνση του κύκλου ζωής των δοκιμών:

Με την τεχνητή νοημοσύνη να αυτοματοποιεί σημαντικά τμήματα της δημιουργίας δοκιμαστικών περιπτώσεων, ο κύκλος ζωής των δοκιμών μπορεί να επιταχυνθεί σημαντικά. Αυτή η επιτάχυνση επιτρέπει στις ομάδες ανάπτυξης να κυκλοφορούν προσβάσιμα προϊόντα γρηγορότερα χωρίς να διακυβεύεται η ποιότητα των δοκιμών προσβασιμότητας.

5.3 Πλεονεκτήματα της παραγωγή σεναρίων ελέγχου με χρήση της Τεχνητής Νοημοσύνης

Η τεχνητή νοημοσύνη φέρνει ένα σημαντικά πλεονεκτήματα στη δημιουργία σεναρίων ελέγχου αξιοποιώντας προηγμένους αλγόριθμους και μοντέλα μηχανικής μάθησης. Τα βασικά πλεονέκτημα είναι[44]:

- **Αυξημένη αποτελεσματικότητα:** Οι αλγόριθμοι τεχνητής νοημοσύνης μπορούν να επεξεργάζονται τεράστιες ποσότητες πληροφοριών γρήγορα, μειώνοντας τον χρόνο και την προσπάθεια που απαιτούνται για τη δημιουργία δοκιμαστικών περιπτώσεων. Η αυτοματοποιημένη δημιουργία δοκιμαστικών περιπτώσεων επιταχύνει τη διαδικασία δοκιμών, επιτρέποντας στις ομάδες ανάπτυξης να τηρούν αυστηρά χρονοδιαγράμματα έκδοσης.
- **Ενισχυμένη κάλυψη:** Η τεχνητή νοημοσύνη έχει τη δυνατότητα να εντοπίζει σενάρια δοκιμών που μπορεί να είναι δύσκολο να οραματιστούν οι ανθρώπινοι δοκιμαστές. Η ολοκληρωμένη κάλυψη οδηγεί σε πιο στιβαρές δοκιμές, μειώνοντας την πιθανότητα μη ανιχνεύσιμων ελαττωμάτων στην παραγωγή.
- **Προσαρμοστικότητα στην Αλλαγή:** Οι αλγόριθμοι τεχνητής νοημοσύνης μπορούν να προσαρμοστούν στις αλλαγές στις απαιτήσεις και στον κώδικα δυναμικά,

διασφαλίζοντας ότι οι δοκιμαστικές περιπτώσεις παραμένουν σχετικές και αποτελεσματικές καθώς εξελίσσεται το λογισμικό.

- Εξοικονόμηση κόστους: Η αυτοματοποιημένη δημιουργία δοκιμαστικών περιπτώσεων μειώνει την εξάρτηση από πόρους χειροκίνητων δοκιμών, με αποτέλεσμα σημαντική εξοικονόμηση κόστους για τις ομάδες ανάπτυξης.
- Μειωμένες χειρωνακτικές προσπάθειες: Η αυτοματοποίηση της δημιουργίας δοκιμαστικών περιπτώσεων μειώνει την επιβάρυνση των μηχανικών QA, επιτρέποντάς τους να επικεντρωθούν σε πιο σύνθετα σενάρια δοκιμών και στρατηγικές εργασίες διασφάλισης ποιότητας.
- Βελτιωμένη ποιότητα λογισμικού: Ο συνδυασμός ανάλυσης βάσει απαιτήσεων και ανάλυσης βάσει κώδικα οδηγεί σε υψηλότερη ποιότητα λογισμικού, με μικρότερη πιθανότητα ελαττωμάτων μετά την κυκλοφορία.
- Πρώιμος εντοπισμός ελαττωμάτων: Δημιουργώντας ένα ποικίλο σύνολο δοκιμαστικών περιπτώσεων νωρίς στον κύκλο ανάπτυξης, τα πιθανά ελαττώματα και τα τρωτά σημεία μπορούν να εντοπιστούν και να διορθωθούν στα αρχικά στάδια, μειώνοντας το κόστος και την προσπάθεια επιδιόρθωσης προβλημάτων αργότερα.

5.4 Εργαλεία ελέγχου λογισμικού με χρήση Τεχνητής Νοημοσύνης

Για τον σκοπό της εργασίας αναζητήθηκαν διαφορά εργαλεία για τον έλεγχο λογισμικού με εφαρμογή μεθόδων τεχνητής νοημοσύνης. Τα εργαλεία αυτά όπως το Functionize, Applitools , Testim.io , TestRigor χρησιμοποιούν διάφορες εφαρμογές τεχνητής νοημοσύνης. Τα προαναφερθέντα εργαλεία ελέγχου λογισμικού εφαρμόζουν διάφορες μεθόδους τεχνητής νοημοσύνης όπως Μηχανική Μάθηση(Machine Learning), Επεξεργασία Φυσικής Γλώσσας(Natural Language Processing) και Computer Visions στοχεύοντας στον αυτοματοποιημένο έλεγχο λογισμικού και την δημιουργία ελέγχων χωρίς κώδικα. Επίσης τα εργαλεία αυτά είναι συνδρομητικά και απευθύνονται σε εταιρίες ζητώντας για την εγγραφή δοκιμαστική χρήση εταιρικό λογαριασμό ηλεκτρονικού ταχυδρομείου. Η εφαρμογή testRigor έχει ως λειτουργικότητα την παραγωγή σεναρίων ελέγχου με επεξεργασία φυσικής γλώσσας αλλά η δοκιμαστική έκδοση 14 ημερών μπορούσε να παράξει έως 3 σενάρια ελέγχου.

Έτσι λόγω των προαναφερθέντων λόγων δεν επιλέχθηκαν οι εφαρμογές αυτές γιατί δεν μπορούσαν να χρησιμοποιηθούν για τον θέμα που ερευνά η συγκεκριμένη εργασία.

5.5 Εργαλεία Τεχνητής Νοημοσύνης για την παραγωγή σεναρίων ελέγχου που θα χρησιμοποιηθούν

Οι εφαρμογές που θα χρησιμοποιηθούν για την υλοποίηση αυτής της εργασίας χρησιμοποιούν δυο διαφορετικά μεγάλα γλωσσικά μοντέλα (Large Language Models).

5.5.1 Large Language Models

Palm2(Pathways Language Model) by Google

[45]Το PaLM 2 είναι ένα προηγμένο μοντέλο γλώσσας που αναπτύχθηκε από την Google, βασισμένο στην επιτυχία του αρχικού μοντέλου PaLM. Με τις 540 δισεκατομμύρια παραμέτρους του και την πυκνή αρχιτεκτονική Transformer μόνο με αποκωδικοποιητή, το PaLM επέδειξε εξαιρετική απόδοση σε διάφορες σύνθετες γλωσσικές εργασίες, όπως η κατανόηση, η δημιουργία περιεχομένου, η συλλογιστική και η κωδικοποίηση. Ξεπέρασε τα προηγούμενα μοντέλα σε διάφορα σημεία

αναφοράς Επεξεργασίας Φυσικής Γλώσσας (NLP) και πολύγλωσσες εργασίες. Το PaLM 2 βελτιώθηκε περαιτέρω σε σχέση με τον προκάτοχό του ενσωματώνοντας προτροπές και λεπτομέρεια αλυσίδων σκέψης, ενισχύοντας τις αριθμητικές και κοινής λογικής ικανότητές του σε πολλαπλά βήματα. Το μοντέλο κατασκευάστηκε χρησιμοποιώντας τη βιβλιοθήκη JAX της Google και την υποδομή TPU v4, επιτρέποντας υπολογισμούς υψηλής απόδοσης. Μια αξιοσημείωτη πτυχή του PaLM 2 είναι η εστίασή του στην προώθηση θετικών αλληλεπιδράσεων εκπαιδευόντας το στην αποκλιμάκωση επιθετικών ή τοξικών συνομιλιών. Αυτή η μοναδική προσέγγιση ανακατευθύνει τέτοιες συνομιλίες προς πιο εποικοδομητικές κατευθύνσεις, συμβάλλοντας σε ένα πιο θετικό διαδικτυακό περιβάλλον.

Το PaLM 2 αντιπροσωπεύει μια σημαντική πρόοδο στη μοντελοποίηση γλώσσας, επιδεικνύοντας βελτιωμένη απόδοση, συλλογιστικές ικανότητες και μια προληπτική προσέγγιση για την προώθηση θετικών αλληλεπιδράσεων.

- **Μετάφραση γλώσσας:** Επιτρέπει την ακριβή και λεπτή μετάφραση κειμένου σε πολλές γλώσσες, βοηθώντας επιχειρήσεις, ιδιώτες και οργανισμούς στην αποτελεσματική πολύγλωσση επικοινωνία.
- **Δημιουργία κώδικα και βοήθεια:** Η επάρκεια του PaLM 2 σε διάφορες γλώσσες προγραμματισμού του επιτρέπει να δημιουργεί κώδικα, να παρέχει προτάσεις, να μεταφράζει κώδικα και να βοηθά τους προγραμματιστές σε εργασίες ανάπτυξης λογισμικού.
- **Ολοκληρωμένες Ερωτήσεις & Απαντήσεις:** Η βαθιά κατανόηση του πλαισίου και του τεράστιου γνωστικού πλαισίου του PaLM 2 του δίνει τη δυνατότητα να παρέχει ολοκληρωμένες και διορατικές απαντήσεις σε διάφορα ερωτήματα, προς όφελος των σπουδαστών, των ερευνητών και των ατόμων που αναζητούν πληροφορίες.
- **Υγειονομική περίθαλψη (med-PaLM 2):** Προσαρμοσμένο για ιατρικές εφαρμογές, το Med-PaLM 2 βοηθά τους ιατρικούς ερευνητές, την ιατρική εκπαίδευση, την κλινική περίθαλψη και τις προσπάθειες δημόσιας υγείας αναλύοντας ιατρική βιβλιογραφία, διάγνωση, υποστήριξη λήψης αποφάσεων και ενισχύοντας την επιτήρηση ασθενειών και απάντηση.
- **Ασφάλεια (sec-PaLM):** Η Sec-PaLM ειδικεύεται σε εφαρμογές κυβερνοασφάλειας, προσφέροντας δυνατότητες όπως ανάλυση κακόβουλου λογισμικού, ευφυΐα απειλών, έρευνα ασφαλείας και αυτοματοποίηση των λειτουργιών ασφαλείας για τον αποτελεσματικό εντοπισμό και τον μετριασμό των απειλών στον κυβερνοχώρο.

GPT4 (Generative pre-trained transformer model-4)

Το GPT-4 είναι ένα μεγάλο γλωσσικά μοντέλο που αναπτύχθηκε από την OpenAI. Ανήκει στην οικογένεια των μοντέλων που βασίζονται σε μετασχηματιστές, τα οποία είναι σχεδιασμένα να επεξεργάζονται και να δημιουργούν κείμενο που μοιάζει με άνθρωπο. Το GPT-4 βασίζεται στην επιτυχία των προκατόχων του, GPT-1, GPT-2 και GPT-3, ενσωματώνοντας προηγμένες τεχνικές επεξεργασίας φυσικής γλώσσας (NLP) και Deep Learning. Το GPT-4 αποτελείται από ένα τεράστιο νευρωνικό δίκτυο με δισεκατομμύρια παραμέτρους, που του επιτρέπουν να κατανοεί και να δημιουργεί κείμενο που είναι συνεκτικό, σχετικό με τα συμφραζόμενα και εξαιρετικά ακριβές. Εκπαιδεύεται σε ένα τεράστιο σύνολο δεδομένων κειμένου, επιτρέποντάς του να μάθει τα μοτίβα και τις αποχρώσεις της ανθρώπινης γλώσσας. Το GPT-4 χρησιμοποιεί μια αρχιτεκτονική μετασχηματιστή, η οποία του επιτρέπει να χειρίζεται εξαρτήσεις μεγάλης εμβέλειας και να αποτυπώνει αποτελεσματικά πολύπλοκες γλωσσικές δομές.

Το GPT-4 έχει ένα ευρύ φάσμα εφαρμογών σε AI, ML και επιστήμη δεδομένων. Μερικές από τις βασικές χρήσεις του περιλαμβάνουν:

- Κατανόηση φυσικής γλώσσας (NLU): Μπορεί να αναλύσει και να κατανοήσει τη φυσική γλώσσα, επιτρέποντάς του να εκτελεί εργασίες όπως η ανάλυση συναισθήματος, η ταξινόμηση κειμένου και η αναγνώριση οντοτήτων. Η ικανότητά του να κατανοεί το πλαίσιο το καθιστά ιδιαίτερα πολύτιμο στην επεξεργασία μη δομημένων δεδομένων.
- Δημιουργία και σύνοψη κειμένου: Μπορεί να δημιουργήσει υψηλής ποιότητας, συνεκτικό κείμενο με βάση μια δεδομένη προτροπή ή είσοδο. Μπορεί να χρησιμοποιηθεί για εργασίες όπως η δημιουργία περιεχομένου, τα Chatbots και η αυτόματη σύνοψη εγγράφων, καθιστώντας το ένα πολύτιμο εργαλείο για δημιουργούς περιεχομένου και ερευνητές.
- Μετάφραση γλώσσας: Με τη βαθιά κατανόηση της γλώσσας, το GPT-4 μπορεί να χρησιμοποιηθεί για εργασίες μηχανικής μετάφρασης. Μπορεί να μεταφράσει κείμενο από τη μια γλώσσα στην άλλη, διατηρώντας παράλληλα το πλαίσιο και το νόημα του αρχικού περιεχομένου.
- Απάντηση σε ερωτήσεις: Το GPT-4 μπορεί να παρέχει ακριβείς και συναφείς απαντήσεις σε ερωτήσεις που τίθενται σε φυσική γλώσσα. Μπορεί να χρησιμοποιηθεί σε Chatbots, εικονικούς βοηθούς και μηχανές αναζήτησης για τη βελτίωση της εμπειρίας χρήστη και την ανάκτηση πληροφοριών.
- Ανάλυση δεδομένων και Ιδέες: Το GPT-4 μπορεί να βοηθήσει σε εργασίες ανάλυσης δεδομένων, κατανοώντας και εξάγοντας πληροφορίες από μεγάλους όγκους δεδομένων κειμένου. Μπορεί να εντοπίσει μοτίβα, τάσεις και συσχετισμούς, βοηθώντας στις διαδικασίες λήψης αποφάσεων.

5.5.2 Εφαρμογές Generative AI

- **TestCraft.app** 

Το TestCraft, μια εφαρμογή ανοιχτού κώδικα ως επέκταση προγράμματος περιήγησης. Η καινοτόμος λύση που συλλέγει HTML απλώς επιλέγοντας τα στοιχεία της ιστοτόπου και χρησιμοποιεί τις προηγμένες δυνατότητες του GPT-4 για τον εξορθολογισμό της διαδικασίας ελέγχου, τη βελτίωση της ποιότητας του έργου και τη σημαντική μείωση του χρόνου και της προσπάθειας που σχετίζονται με τον έλεγχο. Μπορεί να δημιουργεί ισχυρά, αυτοματοποιημένα σενάρια ελέγχου και τα μετατρέπει σε εκτελέσιμο κώδικα JavaScript ή TypeScript με συμβατότητα σε Cypress και Playwright, χωρίς να χρειάζεται να γράψετε τον κώδικα χειροκίνητα. [46]

- **ChatGPT4** 

Το ChatGPT-4 είναι η τέταρτη επανάληψη της σειράς προηγμένων γλωσσικών μοντέλων της OpenAI, που είναι γνωστό ότι δημιουργεί κείμενο που μοιάζει με άνθρωπο. Είναι βασισμένο στις επιτυχίες των προκατόχων του, προσφέροντας μια ευρύτερη βάση γνώσεων, βελτιωμένη κατανόηση σύνθετου κειμένου, βελτιωμένη επίγνωση των συμφραζομένων και πιο ακριβείς απαντήσεις. Εκπαιδευμένο σε μια τεράστια ποικιλία κειμένων στο Διαδίκτυο μέχρι τον Απρίλιο του 2023, το ChatGPT-4 είναι έμπειρο σε διάφορες γλώσσες και έχει σχεδιαστεί για να παρέχει πιο αξιόπιστες, συνεκτικές και ηθικά ασφαλέστερες αλληλεπιδράσεις. Ωστόσο, όπως όλα τα μοντέλα τεχνητής νοημοσύνης, έχει περιορισμούς, συμπεριλαμβανομένων πιθανών μεροληψιών και περιστασιακών ανακριβειών.

- **Copilot Microsoft**  **Copilot**

Το Microsoft Copilot είναι ένα εργαλείο παραγωγικότητας με τεχνητή νοημοσύνη που αξιοποιεί τη δύναμη της τεχνητής νοημοσύνης για την ενίσχυση της παραγωγικότητας, το ξεκλείδωμα της δημιουργικότητας και σας βοηθά να κατανοήσετε καλύτερα τις πληροφορίες με μια απλή εμπειρία συνομιλίας 1. Είναι ένα chatbot που αναπτύχθηκε από τη Microsoft και είναι ο κύριος αντικαταστάτης της Microsoft για τις καταργημένες Cortana 2. Το Microsoft Copilot για το Microsoft 365 είναι ένα εργαλείο παραγωγικότητας με τεχνητή νοημοσύνη που συντονίζει μεγάλα γλωσσικά μοντέλα (LLM) GPT4, περιεχόμενο στο Microsoft Graph και τις εφαρμογές Microsoft 365 που χρησιμοποιείτε καθημερινά, όπως Word, Excel, PowerPoint, Outlook, Ομάδες και άλλα 3. Μπορείτε να συνδεθείτε για να δοκιμάσετε το Microsoft Copilot σε προγράμματα περιήγησης Microsoft Edge ή Chrome, σε Windows και Mac OS.[47]

- **Bard AI**  **Bard**

Το Bard είναι ένα πειραματικό chatbot μεγάλου μοντέλου γλώσσας (LLM) που αναπτύχθηκε από την Google AI. Χρησιμοποιεί το μοντέλο Palm2. Αρχικά εκπαιδεύτηκε σε ένα τεράστιο σύνολο δεδομένων κειμένου και κώδικα και είναι ακόμα υπό ανάπτυξη. Το Bard έχει ένα ευρύ φάσμα δυνατοτήτων, όπως[48]:

- Δημιουργία κειμένου: Το Bard μπορεί να δημιουργήσει διαφορετικές μορφές δημιουργικού κειμένου περιεχομένου κειμένου, όπως ποιήματα, κώδικα, σενάρια, μουσικά κομμάτια, email, γράμματα κ.λπ. Μπορεί επίσης να μεταφράσει γλώσσες, να γράψει διαφορετικά είδη δημιουργικού περιεχομένου και να απαντήσει στις ερωτήσεις σας με ενημερωτικό τρόπο τρόπο.
- Σύνοψη κειμένου: Ο Bard μπορεί να συνοψίσει μεγάλα κομμάτια κειμένου, εξάγοντας τα βασικά σημεία και παρέχοντας μια συνοπτική επισκόπηση.
- Απάντηση σε ερωτήσεις: Ο Bard μπορεί να απαντήσει στις ερωτήσεις σας με κατατοπιστικό τρόπο, ακόμα κι αν είναι ανοιχτές, προκλητικές ή περίεργες.
- Ακολουθούν οδηγίες: Ο Bard μπορεί να κάνει το καλύτερο δυνατό για να ακολουθήσει τις οδηγίες σας και να ολοκληρώσει τα αιτήματά σας προσεκτικά.

Το Bard είναι ακόμα υπό ανάπτυξη, αλλά έχει ήδη μάθει να εκτελεί πολλά είδη εργασιών. Μαθαίνει και βελτιώνεται συνεχώς και έχει τη δυνατότητα να είναι ένα ισχυρό εργαλείο επικοινωνίας, εκπαίδευσης και ψυχαγωγίας.

5.6 Ιστότοπος SauceDemo E-Commerce

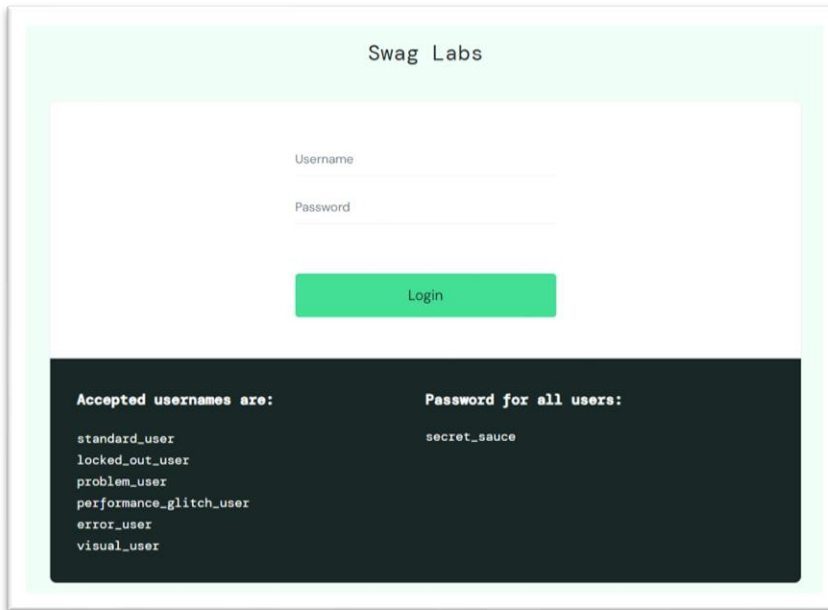
Η εφαρμογή ιστού που επιλέχθηκε για την οποία θα παραχθούν σενάρια και θα εκτελεστούν σενάρια ελέγχου είναι η sausedemo.com. Το SauceDemo είναι ένα δείγμα ιστότοπου ηλεκτρονικού εμπορίου που δημιουργήθηκε από την Sauce Labs με σκοπό την εξάσκηση και εφαρμογή αυτοματοποιημένων ελέγχων. Ο ιστότοπος προσομοιώνει μια εμπειρία διαδικτυακών αγορών. Ο ιστότοπος είναι ιδιαίτερα χρήσιμος για όσους μαθαίνουν για τον αυτοματοποιημένο έλεγχο διεπαφής χρήστη, καθώς περιλαμβάνει λειτουργίες κοινές σε ιστότοπους ηλεκτρονικού εμπορίου πραγματικού κόσμου, αλλά είναι απλοποιημένος και ελεγχόμενος για εκπαιδευτικούς σκοπούς.

5.6.1 Βασικές Λειτουργίες του SauceDemo:

Ακολουθεί μια ανάλυση των λειτουργιών του ιστότοπου.

- Έλεγχος ταυτότητας χρήστη:

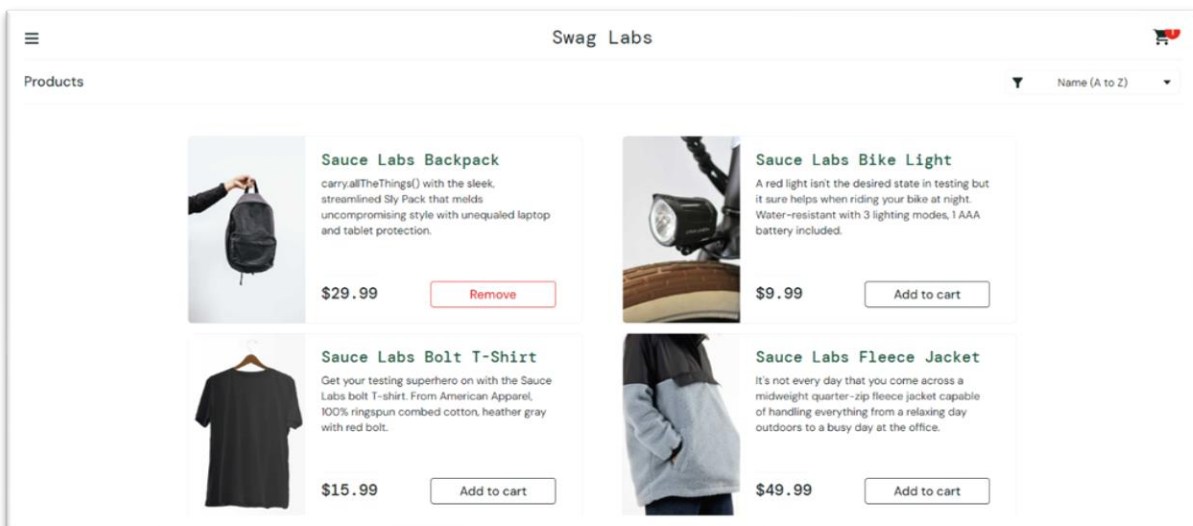
Όπως βλέπουμε και στην κάτωθι εικόνα περιλαμβάνει μια σελίδα σύνδεσης όπου ο χρήστης μπορεί να εισάγει τα διαπιστευτήρια τους. Η πλατφόρμα παρέχει διαφορετικά διαπιστευτήρια για την προσομοίωση διαφόρων ρόλων χρήσης και προσβάσεων.



Εικόνα 8: Φόρμα Σύνδεσης Χρήστη ιστότοπου SauceDemo

- Κατάλογος προϊόντων

Αυτή η σελίδα μιμείται ένα κατάλογο προϊόντων ενός τυπικού ηλεκτρονικού καταστήματος. Ο χρήστης μπορεί να δει μια λίστα προϊόντων, να κάνει προσθήκη των προϊόντων στο καλάθι αγορών, να κάνει αλφαβητική ταξινόμηση η ταξινόμηση βάση τιμής και να κάνει είσοδο στο καλάθι αγορών. Επίσης από το μενού που συμβολίζετε με τρεις οριζόντιες γραμμές όπου βρίσκετε στην επάνω αριστερή γωνία μπορεί να προβεί στις ακόλουθες ενέργειες: Εμφάνιση όλων των προϊόντων, μεταφορά στο σύνδεσμο “σχετικά” , αποσύνδεσης και επαναφοράς της εφαρμογής.

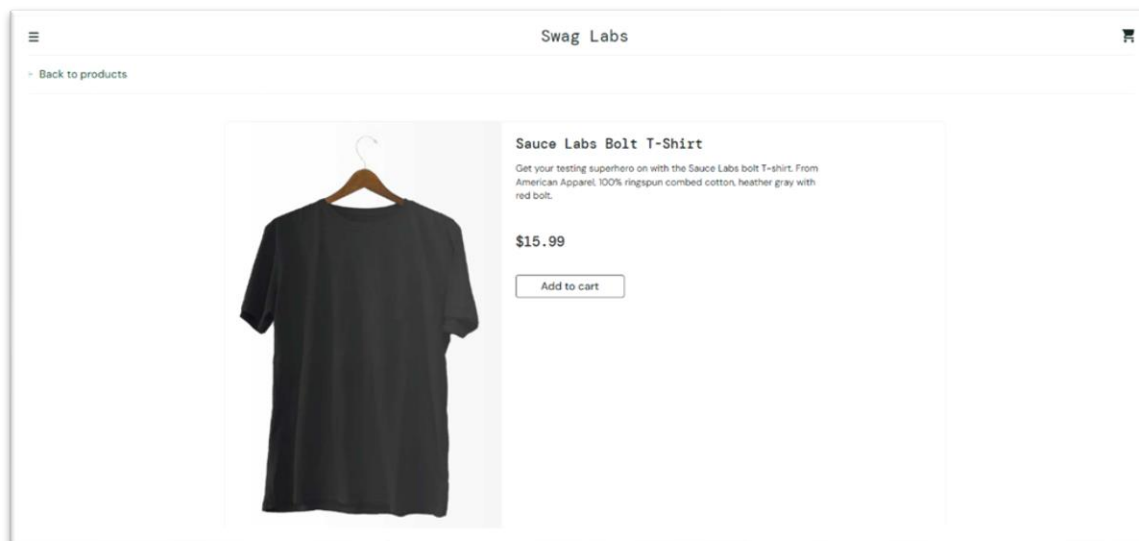


Εικόνα 9: Κατάλογος Προϊόντων ιστότοπου SauceDemo

- Λεπτομέρειες προϊόντος

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

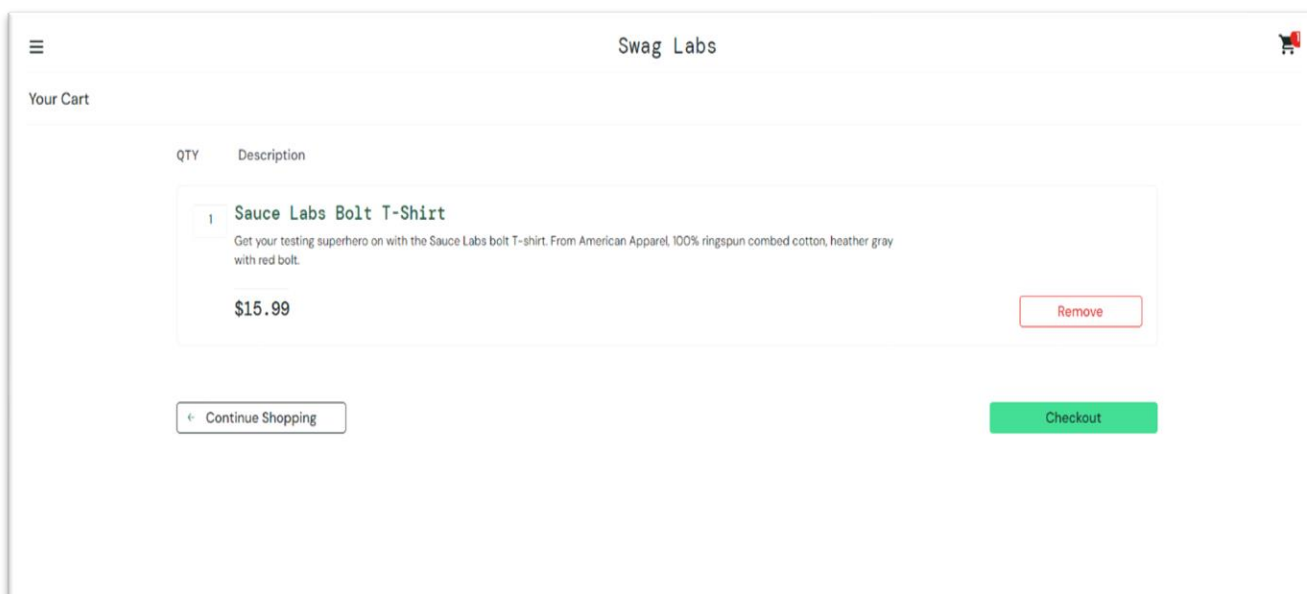
Ο χρήστης μπορεί να κάνει κλικ σε προϊόντα για να προβάλει περισσότερες λεπτομέρειες, παρόμοιες με αυτές που θα βρείτε σε πραγματικούς ιστότοπους ηλεκτρονικού εμπορίου, συμπεριλαμβανομένων περιγραφών και μεγαλύτερων εικόνων.



Εικόνα 10: Λεπτομέρειες Προϊόντος ιστότοπου SauceDemo

- Ηλεκτρονικό καλάθι αγορών

Ο χρήστης μπορεί να προβάλλει και να διαχειρίζεται το καλάθι αγορών του, συμπεριλαμβανομένου του ελέγχου των επιλεγμένων προϊόντων, της ποσότητας και της αφαίρεσης αντικειμένων από το καλάθι.

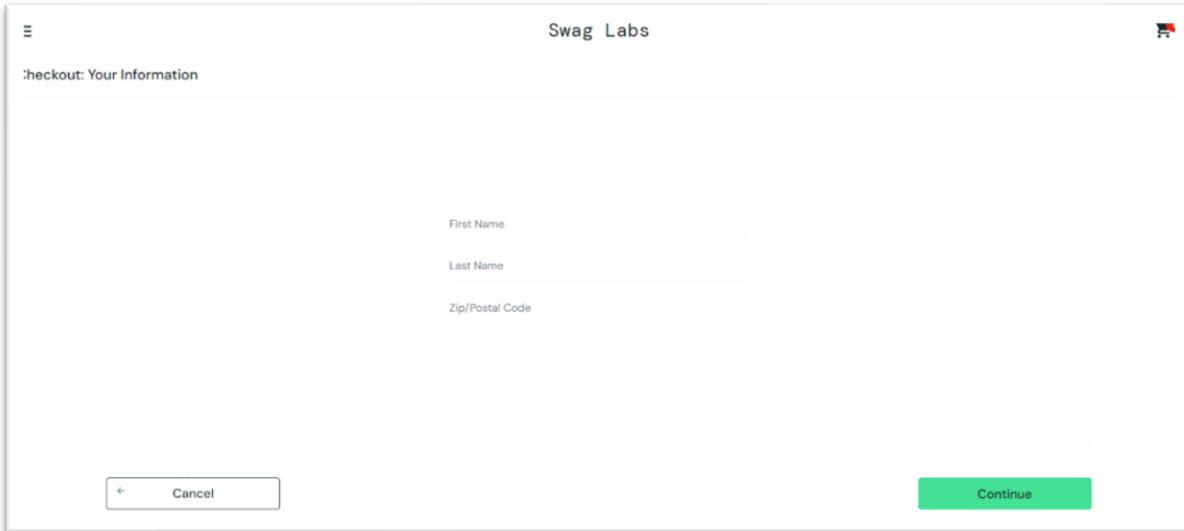


Εικόνα 11: Καλάθι Αγορών ιστότοπου SauceDemo

- Ολοκλήρωση της αγοράς

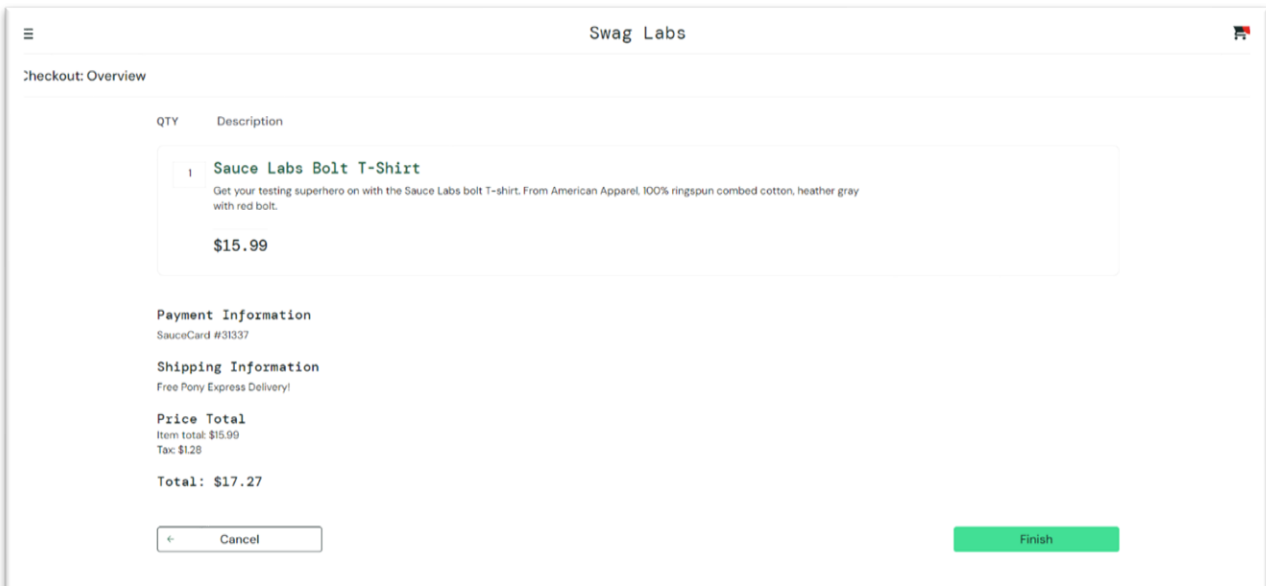
Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

Η διαδικασία της ολοκλήρωσης της αγοράς γίνεται σε τρία βήματα. Στην πρώτη σελίδα όπως φαίνεται στην παρακάτω εικόνα ο χρήστης πρέπει να εισάγει τις πληροφορίες ονόματος, επωνύμου και ταχυδρομικού κώδικα. Έχει τις επιλογές ακύρωσης της παραγγελίας ή συνέχεια για την ολοκλήρωσή της.



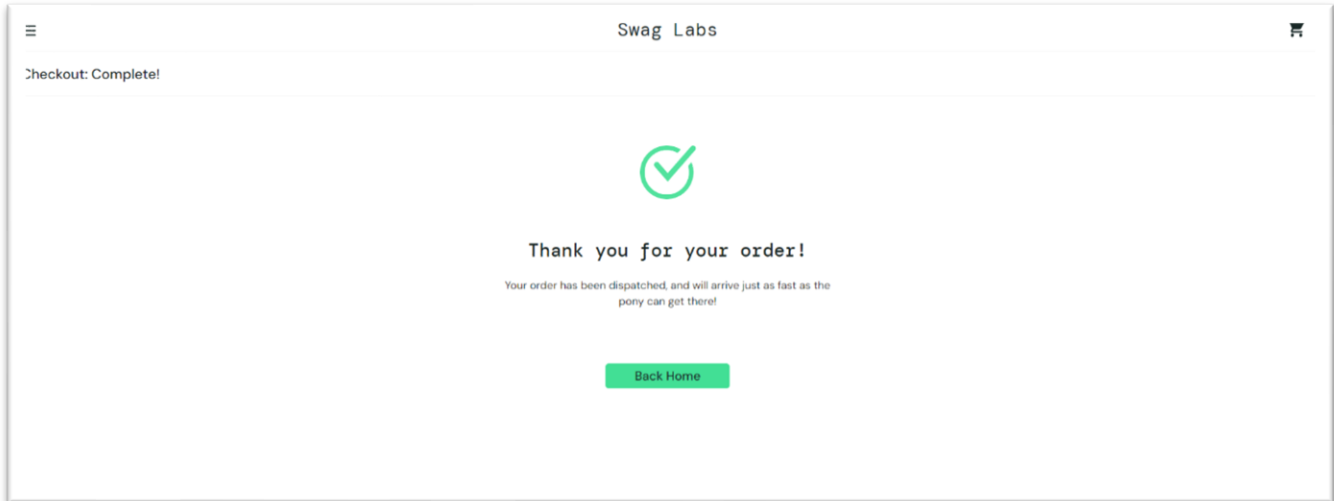
Εικόνα 12: Φόρμα Ολοκλήρωσης Αγοράς 1 ιστότοπου SauceDemo

Στην δεύτερη σελίδα φαίνονται οι λεπτομέρειες της παραγγελίας. Ο χρήστης έχει τις επιλογές ακύρωσης της παραγγελίας ή συνέχεια για την οριστικοποίησή της



Εικόνα 13: Φόρμα Ολοκλήρωσης Αγοράς 2 ιστότοπου SauceDemo

Στην Τρίτη και τελευταία σελίδα εμφανίζονται ένα ευχαριστήριο μήνυμα και αναφορά της κατάστασης της παραγγελίας. Προσομοιώνοντας έτσι μια πλήρη διαδικασία αγοράς.



Εικόνα 14: Ολοκλήρωση Αγοράς ιστότοπου SauceDemo

5.7 Framework Playwright

Το Playwright είναι ένα πλαίσιο ανοιχτού κώδικα που αναπτύχθηκε από τη Microsoft και κυκλοφόρησε το 2020. Το Playwright, ένα ταχέως αναπτυσσόμενο πλαίσιο ανοιχτού κώδικα, επιτρέπει αξιόπιστες δοκιμές από άκρο σε άκρο και αυτοματισμό για σύγχρονες εφαρμογές ιστού. Το Microsoft Playwright Testing είναι μια πλήρως διαχειριζόμενη υπηρεσία που χρησιμοποιεί το cloud για να σας δώσει τη δυνατότητα να εκτελείτε δοκιμές Playwright με πολύ υψηλότερο παραλληλισμό μεταξύ διαφορετικών συνδυασμών λειτουργικού συστήματος-προγράμματος περιήγησης ταυτόχρονα. Αυτό σημαίνει ταχύτερες δοκιμαστικές εκτελέσεις με ευρύτερη κάλυψη σεναρίων, κάτι που βοηθά στην επιτάχυνση της παράδοσης των χαρακτηριστικών χωρίς να θυσιάζεται η ποιότητα. Υποστηρίζει όλα τα σύγχρονα προγράμματα περιήγησης, συμπεριλαμβανομένων των Chromium, WebKit και Firefox, και μπορεί να χρησιμοποιηθεί σε TypeScript, JavaScript, Python, .NET, Java και άλλες γλώσσες.[49]

Η εγκατάσταση είναι αρκετά εύκολη για τον χρήστη με τα παρακάτω βήματα:

1. Ανοίξτε το τερματικό σας και μεταβείτε στον κατάλογο του έργου σας.
2. Εκτελέστε την ακόλουθη εντολή για να εγκαταστήσετε το Playwright:

```
npm init playwright@latest
```

3. Μόλις ολοκληρωθεί η εγκατάσταση, μπορείτε να αρχίσετε να γράφετε δοκιμές χρησιμοποιώντας το Playwright.

Εάν χρησιμοποιείτε τον κώδικα του Visual Studio, μπορείτε επίσης να εγκαταστήσετε την επέκταση Playwright Test Explorer για εκτέλεση και εντοπισμό σφαλμάτων των δοκιμών σας απευθείας από το πρόγραμμα επεξεργασίας

6. ΠΑΡΑΓΩΓΗ ΣΕΝΑΡΙΩΝ ΕΛΕΓΧΟΥ ΜΕ ΧΡΗΣΗ GENERATIVE AI ΕΦΑΡΜΟΓΩΝ

Για την παραγωγή σεναρίων ελέγχου επιλέχθηκε η μέθοδος του Μαύρου Κουτιού(Black Box) που αναφέρετε σε προηγούμενο κεφάλαιο. Θα γίνει η χρήση των εφαρμογών τεχνητής νοημοσύνης που προαναφέρθηκαν: ChatGPT4, Copilot by Microsoft, BardAI by Google και της open source εφαρμογής TestCraft.app . Στις τρεις εφαρμογές ChatGPT4, Copilot by Microsoft και BardAI by Google θα χρησιμοποιήσουμε φυσική γλώσσα μέσω του πεδίου διαλόγου των εφαρμογών για την παραγωγή σεναρίων ελέγχου που αφορούν τον ιστοτόπο SauDemo όπου η περιγραφή του έγινε στο Κεφάλαιο 5. Για τη τελευταία εφαρμογή TestCraft.app ως είσοδο χρησιμοποιούνται στοιχεία της διεπαφής που επιλέγονται μέσω της εφαρμογής αυτής. Εν συνεχεία θα ζητηθεί από τις εφαρμογές να παράγουν κώδικα TypeScript για να εκτελεστούν τα σενάρια ελέγχου με τη χρήση της βιβλιοθήκης Playwright της Microsoft. Ακολουθεί ανάλυση της παραγωγής σεναρίων ελέγχου κάθε εφαρμογής.

6.1 ChatGPT4

Το ChatGPT 4 όπως προαναφέρθηκε είναι μια δημιουργία της OpenAI, η οποία είναι πρωτοπόρος στην έρευνα τεχνητής νοημοσύνης. Ανέπτυξε το ChatGPT 4 για να ξεπεράσει τα όρια των μοντέλων τεχνητής νοημοσύνης. Έχει σχεδιαστεί για να φέρει επανάσταση στον τρόπο που αλληλοεπιδρούμε με της μηχανές. Είναι σαν να έχεις έναν εικονικό συνομιλητή που καταλαβαίνει και ανταποκρίνεται με έναν εντυπωσιακά ανθρώπινο τρόπο.

Είναι ένα εξαιρετικά έξυπνο chatbot που μπορεί να συμμετέχει σε συζητήσεις, να απαντά σε ερωτήματα και να παρέχει πληροφορίες για ένα ευρύ φάσμα θεμάτων. Σύμφωνα με τις εσωτερικές αξιολογήσεις, το GPT-4 είναι 82% λιγότερο πιθανό να ανταποκριθεί σε ευαίσθητο και επιβλαβές περιεχόμενο και 40% πιο πιθανό να δώσει ακριβείς απαντήσεις από το GPT-3.5.

Ως είσοδος στην εφαρμογή δόθηκε το παρακάτω κείμενο σε φυσική γλώσσα:

You
As a front-end tester engineer, I have to create test cases about <https://www.saucedemo.com/> . These scenarios must cover the entire scope of the application. Can you create these test cases?

Στο σύνολο δημιούργησε 45 σενάρια ελέγχου από τα οποία τα 35 έχουν την δυνατότητα να εκτελεστούν βάση των δυνατοτήτων του Playwright. Κάποια από αυτά τα σενάρια είναι :

- Δοκιμή σε διαφορετικά λειτουργικά συστήματα
- Δοκιμή συμβατότητας με προγράμματα ανάγνωσης οθόνης
- Ελέγξτε για επαρκή χρωματική αντίθεση
- Ελέγξτε για προστασία από επιθέσεις ωμής βίας
- Δοκιμή για ευπάθειες παραβίασης περιόδου λειτουργίας

Τα σενάρια που ανταποκρίνονται στο αίτημα που θέσαμε στην εφαρμογή παρατίθενται στον παρακάτω πίνακα.

Πίνακας 2: Σενάρια Ελέγχου ChatGPT4

Κατηγορία ελέγχου	Test Case ID	Περιγραφή περίπτωσης ελέγχου	Αναμενόμενο αποτέλεσμα
Λειτουργικός έλεγχος			
Σύνδεση χρήστη	TC01	Δοκιμή έγκυρων διαπιστευτηρίων σύνδεσης	Επιτυχής σύνδεση
	TC02	Δοκιμή μη έγκυρου ονόματος χρήστη ή/και κωδικού πρόσβασης	Εμφάνιση μηνύματος σφάλματος. Αποτροπή σύνδεσης
	TC03	Δοκιμή με κενά πεδία ονόματος χρήστη ή/και κωδικού πρόσβασης	Εμφάνιση μηνύματος σφάλματος. Αποτροπή σύνδεσης
	TC04	Λειτουργία δοκιμαστικής αποσύνδεσης	Επιτυχής αποσύνδεση
Κατάλογος Προϊόντων	TC05	Βεβαιωθείτε ότι παρατίθενται όλα τα προϊόντα	Όλα τα προϊόντα που εμφανίζονται
	TC06	Test sorting functionality (Name, Price)	Τα προϊόντα ταξινομήθηκαν σωστά
	TC07	Δοκιμή λειτουργίας φίλτρου, εάν υπάρχει	Τα προϊόντα φιλτράρονται σωστά
Λεπτομέρειες προϊόντος	TC08	Επαληθεύστε ότι κάνοντας κλικ σε ένα προϊόν εμφανίζονται τα στοιχεία του	Εμφανίζονται τα σωστά στοιχεία
	TC09	Ελέγξτε για σωστή περιγραφή, τιμή και εικόνα προϊόντος	Ακριβείς πληροφορίες που εμφανίζονται
Λειτουργικότητα καλαθιού	TC10	Δοκιμάστε να προσθέσετε προϊόντα στο καλάθι	Προϊόντα που προστέθηκαν σωστά στο καλάθι
	TC11	Δοκιμάστε να αφαιρέσετε αντικείμενα από το καλάθι	Τα προϊόντα αφαιρέθηκαν σωστά από το καλάθι
	TC12	Επαληθεύστε σωστά τις ενημερώσεις καλαθιού με πολλά στοιχεία	Το καλάθι εμφανίζει σωστό σύνολο και προϊόντα
	TC13	Δοκιμή συνέπειας καλαθιού μετά την αποσύνδεση και την επιστροφή	Το καλάθι διατηρεί τα προϊόντα μετά την εκ νέου σύνδεση
Διαδικασία ολοκλήρωσης αγοράς	TC14	Δοκιμή εισαγωγής πληροφοριών αποστολής	Οι πληροφορίες αποστολής έγιναν δεκτές και αποθηκεύτηκαν
	TC15	Δοκιμάστε τη λειτουργικότητα διαφορετικών μεθόδων πληρωμής	Η πληρωμή διεκπεραιώθηκε σωστά

	TC16	Επαληθεύστε την επιβεβαίωση παραγγελίας και τη δημιουργία αποδείξεων	Εμφανίζεται επιβεβαίωση και απόδειξη
	TC17	Δοκιμαστική ολοκλήρωση αγοράς με άδειο καλάθι	Αποτροπή ολοκλήρωσης αγοράς ή εμφάνιση κατάλληλου μηνύματος
Έλεγχος χρηστικότητα			
Πλοήγηση	TC18	Εξασφαλίστε συνεπή γραμμή πλοήγησης	Συνεπής πλοήγηση σε σελίδες
	TC19	Δοκιμάστε για εύκολη πρόσβαση στο καλάθι και αποσυνδεθείτε	Καλάθι και αποσύνδεση εύκολα προσβάσιμο
	TC20	Ελέγξτε για σαφείς, κατανοητές ετικέτες και οδηγίες	Οι ετικέτες και οι οδηγίες είναι σαφείς και κατανοητές
Αναγνωσιμότητα περιεχομένου	TC21	Δοκιμή απόκρισης και αναγνωσιμότητας για κινητές συσκευές	Ο ιστότοπος είναι αναγνώσιμος και λειτουργικός σε κινητές συσκευές
Μηνύματα σφάλματος	TC22	Βεβαιωθείτε ότι τα μηνύματα σφάλματος είναι σαφή και ενημερωτικά	Τα μηνύματα σφάλματος παρέχουν σαφείς πληροφορίες
	TC23	Ελέγξτε εάν εμφανίζονται μηνύματα σφάλματος για εσφαλμένη εισαγωγή στοιχείων από το χρήστη	Εμφανίζονται κατάλληλα μηνύματα σφάλματος για σφάλματα εισαγωγής
Έλεγχος διεπαφής			
Διάταξη και σχεδιασμός	TC24	Ελέγξτε τη συνέπεια της διάταξης και του σχεδιασμού σε διαφορετικές σελίδες	Συνεπής διάταξη και σχεδιασμός
	TC25	Δοκιμάστε τον ανταποκρινόμενο σχεδιασμό σε διάφορες συσκευές και μεγέθη οθόνης	Η διάταξη τοποθεσίας προσαρμόζεται κατάλληλα σε διαφορετικές οθόνες
Εικόνες και εικονίδια	TC26	Επαληθεύστε ότι όλες οι εικόνες και τα εικονίδια φορτώνονται σωστά	Όλες οι εικόνες και τα εικονίδια εμφανίζονται σωστά
	TC27	Δοκιμή κατάλληλου εναλλακτικού κειμένου σε εικόνες	Το εναλλακτικό κείμενο είναι παρόν και περιγραφικό
Συνδέσεις	TC28	Δοκιμή όλων των εσωτερικών και εξωτερικών συνδέσεων	Όλοι οι σύνδεσμοι οδηγούν στις σωστές σελίδες
	TC29	Έλεγχος για κατεστραμμένους	Χωρίς κατεστραμμένους συνδέσμους ή λανθασμένες ανακατευθύνσεις

		συνδέσμους ή εσφαλμένη ανακατεύθυνση	
Έλεγχος συμβατότητας			
Συμβατότητα προγράμματος περιήγησης	TC30	Δοκιμάστε τον ιστότοπο σε διαφορετικά προγράμματα περιήγησης	Λειτουργική συνέπεια και συνέπεια εμφάνισης μεταξύ των προγραμμάτων περιήγησης
Συμβατότητα συσκευών	TC31	Δοκιμή σε διαφορετικές συσκευές	Ο ιστότοπος λειτουργεί σωστά σε διάφορες συσκευές
Έλεγχος απόδοσης			
Χρόνος φόρτωσης	TC32	Ελέγξτε το χρόνο φόρτωσης του ιστότοπου και των σελίδων του	Αποδεκτοί χρόνοι φόρτωσης
Ταχύτητα και ανταπόκριση	TC33	Ελέγξτε την ανταπόκριση της εφαρμογής με διάφορες ενέργειες χρήστη	Η εφαρμογή ανταποκρίνεται γρήγορα στις αλληλεπιδράσεις των χρηστών
Έλεγχος ασφαλείας			
Έλεγχος ταυτότητας	TC34	Ελέγξτε για ευπάθειες στο σύστημα σύνδεσης	Δεν βρέθηκαν ευπάθειες
	TC35	Ελέγξτε για ευπάθειες έγχυσης SQL	Ο ιστότοπος είναι ασφαλής έναντι έγχυσης SQL
	TC36	Βεβαιωθείτε ότι οι περίοδοι σύνδεσης λήγουν με ασφάλεια κατά την αποσύνδεση	Η περίοδος λειτουργίας λήγει με ασφάλεια χωρίς διαρροή δεδομένων
Έλεγχος προσβασιμότητας			
Πλοήγηση μέσω πληκτρολογίου	TC37	Βεβαιωθείτε ότι η πλοήγηση στον ιστότοπο γίνεται χρησιμοποιώντας μόνο πληκτρολόγιο	Δυνατότητα πλήρους πλοήγησης με πληκτρολόγιο

Έπειτα ζητήθηκε να παράξει TypeScript κώδικα για τα συγκεκριμένα σενάρια έτσι ώστε να εκτελεστούν με χρήση της βιβλιοθήκης ανοιχτού κώδικα Playwright.



Όπως παρατηρείτε ο αριθμός των σεναρίων ελέγχου που παρήχθησαν και η κάλυψη των βασικών λειτουργιών του ιστότοπου είναι αρκετά ικανοποιητικά. Αξιοσημείωτη είναι η παράθεση διαφορετικών κατηγοριών σεναρίων ελέγχου. Τα σενάρια που δημιουργήθηκαν για τον ιστότοπο SauceDemo ,καλύπτουν ένα ευρύ φάσμα πτυχών που είναι απαραίτητες για μια ισχυρή στρατηγική ελέγχου. Περιλαμβάνουν λειτουργικούς ελέγχους για να διασφαλιστεί ότι όλες οι λειτουργίες ανταποκρίνονται όπως προβλέπεται, σενάρια ελέγχου χρησιμότητας για να διασφαλιστεί μια φιλική προς το χρήστη εμπειρία και σενάρια ελέγχου διεπαφής για έλεγχο των οπτικών στοιχείων και της συνέπειας της διάταξης. Περιλαμβάνεται δοκιμή συμβατότητας για να εξακριβωθεί η απόδοση του ιστότοπου σε διαφορετικά προγράμματα περιήγησης, συσκευές και λειτουργικά συστήματα.

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

Για τα σενάρια αυτά δημιουργήθηκαν δύο αρχεία TypeScript :

1. Ένα αρχείο με το όνομα ChatGPT4_tests_before.ts με περιεχόμενο τον παραχθέντα κώδικα χωρίς διορθώσεις.
2. Ένα αρχείο με το όνομα ChatGPT4_tests_after.ts με περιεχόμενο τον παραχθέντα κώδικα που έχει υποστεί διορθωθεί και να είναι εκτελέσιμος.

Παραδείγματα κώδικα αρχείων:

- ChatGPT4_tests_before.ts

```
//Test checkout with an empty cart
test('TC17:Checkout with an empty cart should be prevented', async ({ page }) => {
  await loginAndAddItem(page);
  await page.click('.cart_item:has-text("Sauce Labs Backpack") .cart_button');
  await page.click('.checkout_button');
  // Verify if checkout is prevented or a message is displayed
  // This depends on the actual implementation of the site
});
```

- ChatGPT4_tests_after.ts

```
// TC17: Test checkout with an empty cart
test('Checkout with an empty cart should be prevented', async ({ page }) => {
  await loginAndAddItem(page);
  await page.click('.cart_item:has-text("Sauce Labs Backpack") .cart_button');
  await page.click('.checkout_button');
  await page.click('.submit-button');
  // Verify if checkout is prevented or a message is displayed
  // This depends on the actual implementation of the site
  await expect(page.locator('[data-test="error"]')).toBeVisible();
});
```

6.2 Copilot by Microsoft

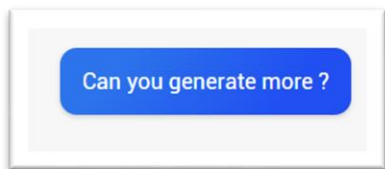
Το Microsoft Copilot είναι ένα chatbot που αναπτύχθηκε από τη Microsoft. Το Copilot χρησιμοποιεί το μοντέλο Microsoft Prometheus, που βασίζεται στο θεμελιώδες μοντέλο μεγάλης γλώσσας GPT-4 της OpenAI, το οποίο με τη σειρά του έχει βελτιστοποιηθεί χρησιμοποιώντας τόσο εποπτευόμενες όσο και ενισχυτικές τεχνικές εκμάθησης. Η διεπαφή συνομιλίας του Copilot μοιάζει με αυτή του ChatGPT.

Ως είσοδο στην εφαρμογή δόθηκε το παρακάτω κείμενο σε φυσική γλώσσα:



Can you generate test cases about <https://www.saucedemo.com/> ?

Η εφαρμογή στην πρώτη της απάντηση μας δημιούργησε μικρό αριθμό σεναρίων και έτσι ζητήθηκε να μας δημιουργήσει περισσότερα:



Το αίτημα για παραγωγή περισσότερων σεναρίων έγινε δύο φορές και έπειτα μας απάντησε ότι δεν μπορεί να παράξει περισσότερα σενάρια από αυτά που έχει ήδη δημιουργήσει.

Στο σύνολο δημιούργησε 12 σενάρια ελέγχου που παρατίθενται στον παρακάτω πίνακα.

Πίνακας 3: Σενάρια Ελέγχου Copilot

Test case ID	Περιγραφή Σεναρίου Ελέγχου	Αναμενόμενο αποτέλεσμα
TC01	Βεβαιωθείτε ότι ο χρήστης μπορεί να συνδεθεί με έγκυρα διαπιστευτήρια. Βήμα 1: Μεταβείτε στο https://www.saucedemo.com/ Βήμα 2: Εισαγάγετε ένα έγκυρο όνομα χρήστη και κωδικό πρόσβασης στα πεδία σύνδεσης Βήμα 3: Κάντε κλικ στο κουμπί σύνδεσης	Ο χρήστης θα πρέπει να ανακατευθυνθεί στη σελίδα αποθέματος και να δει μια λίστα προϊόντων
TC02	Βεβαιωθείτε ότι ο χρήστης μπορεί να προσθέσει ένα προϊόν στο καλάθι αγορών. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Επιλέξτε ένα προϊόν από τη σελίδα αποθέματος Βήμα 3: Κάντε κλικ στο κουμπί προσθήκης στο καλάθι	Το προϊόν θα πρέπει να προστεθεί στο καλάθι και το εικονίδιο του καλαθιού θα πρέπει να δείχνει τον αριθμό των προϊόντων στο καλάθι
TC03	Βεβαιωθείτε ότι ο χρήστης μπορεί να αφαιρέσει ένα προϊόν από το καλάθι αγορών. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Προσθέστε ένα προϊόν στο καλάθι Βήμα 3: Κάντε κλικ στο εικονίδιο του καλαθιού Βήμα 4: Κάντε κλικ στο κουμπί αφαίρεσης για το προϊόν	Το προϊόν θα πρέπει να αφαιρεθεί από το καλάθι και το εικονίδιο του καλαθιού θα πρέπει να εμφανίζει τον ενημερωμένο αριθμό προϊόντων στο καλάθι

TC04	<p>Βεβαιωθείτε ότι ο χρήστης μπορεί να ταξινομήσει τα προϊόντα κατά τιμή, όνομα ή δημοτικότητα. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Κάντε κλικ στο αναπτυσσόμενο μενού στη σελίδα αποθέματος Βήμα 3: Επιλέξτε μία από τις επιλογές ταξινόμησης (χαμηλή προς υψηλή, υψηλή προς χαμηλή, Α έως Ω, Ω έως Α)</p>	<p>Τα προϊόντα πρέπει να ταξινομούνται σύμφωνα με την επιλεγμένη επιλογή</p>
TC05	<p>Βεβαιωθείτε ότι ο χρήστης μπορεί να ολοκληρώσει την αγορά με τα προϊόντα στο καλάθι. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Προσθέστε ένα ή περισσότερα προϊόντα στο καλάθι Βήμα 3: Κάντε κλικ στο εικονίδιο του καλαθιού Βήμα 4: Κάντε κλικ στο κουμπί ολοκλήρωσης αγοράς Βήμα 5: Εισαγάγετε το όνομα, το επώνυμο και τον ταχυδρομικό κώδικα στη σελίδα πληροφοριών ολοκλήρωσης αγοράς Βήμα 6: Κάντε κλικ στο κουμπί συνέχειας Βήμα 7: Κάντε κλικ στο κουμπί τερματισμού στη σελίδα επισκόπησης ολοκλήρωσης αγοράς</p>	<p>Αναμενόμενο αποτέλεσμα: Ο χρήστης θα πρέπει να δει ένα μήνυμα επιβεβαίωσης ότι η παραγγελία έχει υποβληθεί</p>
TC06	<p>Βεβαιωθείτε ότι ο χρήστης μπορεί να αποσυνδεθεί από το μενού. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Κάντε κλικ στο εικονίδιο μενού στην επάνω αριστερή γωνία Βήμα 3: Κάντε κλικ στην επιλογή αποσύνδεσης</p>	<p>Ο χρήστης θα πρέπει να αποσυνδεθεί και να δει τη σελίδα σύνδεσης</p>
TC07	<p>Βεβαιωθείτε ότι ο χρήστης μπορεί να φιλτράρει τα προϊόντα ανά κατηγορία Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Κάντε κλικ στο εικονίδιο μενού στην επάνω αριστερή γωνία Βήμα 3: Επιλέξτε μία από τις επιλογές κατηγορίας (Όλα τα είδη, σακίδιο Sauce Labs, Sauce Labs Bike Light κ.λπ.)</p>	<p>Τα προϊόντα πρέπει να φιλτράρονται σύμφωνα με την επιλεγμένη κατηγορία και να εμφανίζουν μόνο τα σχετικά είδη</p>
TC08	<p>Βεβαιωθείτε ότι ο χρήστης μπορεί να επαναφέρει την κατάσταση της εφαρμογής από το μενού. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Προσθέστε ένα ή περισσότερα προϊόντα στο καλάθι Βήμα 3: Κάντε κλικ στο εικονίδιο μενού στην επάνω αριστερή γωνία Βήμα 4: Κάντε κλικ στην επιλογή επαναφοράς κατάστασης εφαρμογής</p>	<p>Τα προϊόντα θα πρέπει να αφαιρεθούν από το καλάθι και το εικονίδιο του καλαθιού θα πρέπει να εμφανίζει μηδενικά προϊόντα</p>

TC09	Βεβαιωθείτε ότι ο χρήστης μπορεί να προβάλει τη σελίδα πληροφοριών από το μενού. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Κάντε κλικ στο εικονίδιο μενού στην επάνω αριστερή γωνία Βήμα 3: Κάντε κλικ στην επιλογή σχετικά με	Ο χρήστης θα πρέπει να δει μια νέα καρτέλα ανοιχτή με τον ιστότοπο της Sauce Labs
TC10	Βεβαιωθείτε ότι ο χρήστης μπορεί να προβάλει τις λεπτομέρειες του προϊόντος. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Επιλέξτε ένα προϊόν από τη σελίδα αποθέματος Βήμα 3: Κάντε κλικ στην εικόνα ή το όνομα του προϊόντος	Ο χρήστης θα πρέπει να δει μια σελίδα λεπτομερειών προϊόντος με το όνομα, την περιγραφή, την τιμή και την εικόνα του προϊόντος
TC11	Βεβαιωθείτε ότι ο χρήστης μπορεί να αλλάξει την ποσότητα των προϊόντων στο καλάθι. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Προσθέστε ένα ή περισσότερα προϊόντα στο καλάθι Βήμα 3: Κάντε κλικ στο εικονίδιο του καλάθι Βήμα 4: Κάντε κλικ στο κουμπί συν ή πλην δίπλα στην ποσότητα του προϊόντος	Η ποσότητα του προϊόντος θα πρέπει να αυξηθεί ή να μειωθεί αναλόγως και το συνολικό ποσό θα πρέπει να επικαιροποιηθεί
TC12	Βεβαιωθείτε ότι ο χρήστης μπορεί να ακυρώσει τη διαδικασία ολοκλήρωσης αγοράς. Βήμα 1: Συνδεθείτε με έγκυρα διαπιστευτήρια Βήμα 2: Προσθέστε ένα ή περισσότερα προϊόντα στο καλάθι Βήμα 3: Κάντε κλικ στο εικονίδιο του καλάθι Βήμα 4: Κάντε κλικ στο κουμπί ολοκλήρωσης αγοράς Βήμα 5: Εισαγάγετε το όνομα, το επώνυμο και τον ταχυδρομικό κώδικα στη σελίδα πληροφοριών ολοκλήρωσης αγοράς Βήμα 6: Κάντε κλικ στο κουμπί συνέχειας Βήμα 7: Κάντε κλικ στο κουμπί ακύρωσης στη σελίδα επισκόπησης ολοκλήρωσης αγοράς	Ο χρήστης θα πρέπει να ανακατευθυνθεί στη σελίδα αποθέματος και τα προϊόντα θα πρέπει να παραμείνουν στο καλάθι

Τα σενάρια που μας δημιουργήθηκαν καλύπτουν ορισμένες από τις κύριες λειτουργίες του ιστότοπου, όπως σύνδεση χρήστη, περιήγηση προϊόντων, διαχείριση καλάθιού αγορών, ολοκλήρωση αγοράς και πλοήγηση στο μενού. Τα σενάρια ελέγχου έχουν σχεδιαστεί για να επαληθεύουν τη λειτουργικότητα και την ποιότητα της εφαρμογής και να διασφαλίσουν ότι ανταποκρίνεται στις απαιτήσεις και τις προσδοκίες των τελικών χρηστών. Για το κάθε σενάριο μας παραθέτει μια περιγραφή, τα βήματα που πρέπει να ακολουθήσουμε και το αναμενόμενο αποτέλεσμα. Ωστόσο, οι περιπτώσεις ελέγχου δεν είναι εξαντλητικές και δεν καλύπτουν όλα τα πιθανά σενάρια, τις ακραίες περιπτώσεις ή τα αρνητικά σενάρια. Στη συνέχεια ζητήθηκε να παραχθεί TypeScript κώδικας για τα συγκεκριμένα σενάρια έτσι ώστε να εκτελεστούν με χρήση της βιβλιοθήκης ανοιχτού κώδικα Playwright.

Generate typescript code compatible with playwright for the test cases.

Για τα σενάρια αυτά δημιουργήθηκαν δύο αρχεία TypeScript :

1. Ένα αρχείο με το όνομα Copilot_tests_before.ts με περιεχόμενο τον παραχθέντα κώδικα χωρίς διορθώσεις.
2. Ένα αρχείο με το όνομα Copilot_tests_after.ts με περιεχόμενο τον παραχθέντα κώδικα που έχει διορθωθεί και να είναι εκτελέσιμος.

Παραδείγματα κώδικα αρχείων:

- Copilot_tests_before.ts

```
//Verify that the user can reset the app state from the menu
test('TC08: Reset app state', async ({ page }) => {
  // Log in with valid credentials
  await page.goto(BASE_URL);
  await page.fill('#user-name', 'standard_user');
  await page.fill('#password', 'secret_sauce');
  await page.click('#login-button');
  // Add one or more products to the cart
  await page.click('.inventory_item:nth-child(1) .btn_inventory');
  await page.click('.inventory_item:nth-child(2) .btn_inventory');
  // Click on the menu icon on the top left corner
  await page.click('.bm-burger-button');
  // Click on the reset app state option
  await page.click('#reset_sidebar_link');
  // Expect that the products should be removed from the cart and the cart icon should show zero items
  await expect(page).not.toHaveText('.inventory_item:nth-child(1) .btn_inventory', 'REMOVE');
  await expect(page).not.toHaveText('.inventory_item:nth-child(2) .btn_inventory', 'REMOVE');
  await expect(page).not.toHaveText('.shopping_cart_badge', '2');
});
```

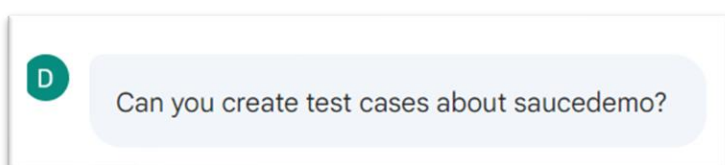
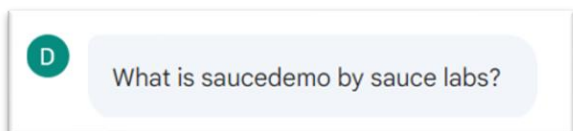
- Copilot_tests_after.ts

```
//Verify that the user can reset the app state from the menu
test('TC08: Reset app state', async ({ page }) => {
  // Log in with valid credentials
  await page.goto(BASE_URL);
  await page.fill('#user-name', 'standard_user');
  await page.fill('#password', 'secret_sauce');
  await page.click('#login-button');
  // Add one or more products to the cart
  await page.click('.inventory_item:nth-child(1) .btn_inventory');
  await page.click('.inventory_item:nth-child(2) .btn_inventory');
  // Click on the menu icon on the top left corner
  await page.click('.bm-burger-button');
  // Click on the reset app state option
  await page.click('#reset_sidebar_link');
  // Expect that the products should be removed from the cart and the cart icon should show zero items
  const products = await page.locator('.shopping_cart_badge').count();
  expect(products).toBe(0);
});
```

6.3 BardAI by Google

Το Bard είναι το εργαλείο συνομιλίας AI της Google που βρίσκεται ακόμα στην πειραματική του φάση. Με την υποστήριξη του μεγάλου γλωσσικού μοντέλου της Google, PaLM2, αυτή η παραγωγική υπηρεσία συνομιλίας AI αντλεί πληροφορίες από το Διαδίκτυο για να απαντήσει σε ερωτήσεις, να αντιμετωπίσει σφάλματα κώδικα και να απαντήσει σε άλλα αιτήματα.

Σαν είσοδο στην εφαρμογή δόθηκε το παρακάτω κείμενο σε φυσική γλώσσα:



Στην συγκεκριμένη εφαρμογή για να μπορέσει να μας δημιουργήσει σενάρια για τον ιστότοπο SauceDemo έπρεπε πρώτα να θέσουμε το ερώτημα να μας πει τι είναι το SauceDemo όπως βλέπουμε στην εικόνα. Για την παραγωγή ενός ικανοποιητικού αριθμού έπρεπε επίσης όπως την περίπτωση του Copilot να ζητήσουμε άλλες δυο φορές να μας παράξει περισσότερα.

Στο σύνολο παρήγαγε 14 σενάρια ελέγχου που παρατίθενται στον παρακάτω πίνακα.

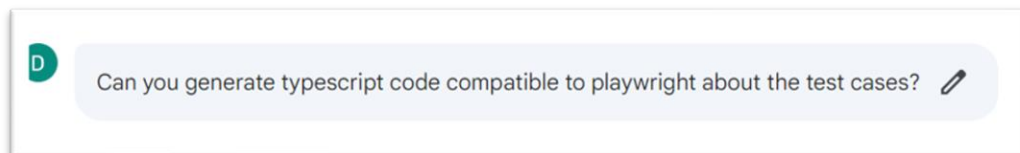
Πίνακας 4: Σενάρια Ελέγχου BardAI

Test Case ID	Περιγραφή	Αναμενόμενο αποτέλεσμα
TC01	Επαληθεύστε τη σύνδεση με έναν έγκυρο χρήστη	Ο χρήστης συνδέεται και ανακατευθύνεται στη σελίδα προϊόντων.
TC02	Προσθήκη προϊόντος στο καλάθι αγορών	Το προϊόν προστίθεται στο καλάθι αγορών.
TC03	Ολοκλήρωση αγοράς και ολοκλήρωση παραγγελίας	Η παραγγελία υποβάλλεται επιτυχώς και εμφανίζεται η σελίδα επιβεβαίωσης παραγγελίας.
TC04	Αποσύνδεση από συνδεδεμένο λογαριασμό	Ο χρήστης αποσυνδέεται και ανακατευθύνεται στη σελίδα σύνδεσης.
TC05	Επαλήθευση τιμών προϊόντων	Οι τιμές που εμφανίζονται στη σελίδα του προϊόντος και η κάρτα κατάδειξης ταιριάζουν.
TC06	Επικύρωση περιγραφών προϊόντων	Οι περιγραφές των προϊόντων είναι ακριβείς, ενημερωτικές και κατανοητές.
TC07	Ελέγξτε τους ποσοτικούς περιορισμούς του προϊόντος	Η ποσότητα ενός προϊόντος στο καλάθι αγορών περιορίζεται στη διαθέσιμη ποσότητα.

TC08	Επαληθεύστε τη λειτουργικότητα του καλαθιού αγορών	Η συνολική αξία καλαθιού ενημερώνεται σωστά καθώς τα προϊόντα προστίθενται, αφαιρούνται ή αδειάζουν από το καλάθι.
TC09	Επικύρωση διαδικασίας ολοκλήρωσης αγοράς	Ο χρήστης μπορεί να ολοκληρώσει με επιτυχία την ολοκλήρωση της παραγγελίας του.
TC10	Επαληθεύστε τη λειτουργία εγγραφής χρήστη	Ο χρήστης μπορεί να καταχωρήσει έναν νέο λογαριασμό με έγκυρα διαπιστευτήρια.
TC11	Επαληθεύστε τη λειτουργία ξεχασμένου κωδικού πρόσβασης	Ένα μήνυμα ηλεκτρονικού ταχυδρομείου με οδηγίες σχετικά με τον τρόπο επαναφοράς του κωδικού πρόσβασης αποστέλλεται στην καταχωρημένη διεύθυνση ηλεκτρονικού ταχυδρομείου.
TC12	Επαληθεύστε τη λειτουργία παρακολούθησης παραγγελιών	Ο χρήστης μπορεί να παρακολουθεί την κατάσταση της παραγγελίας του χρησιμοποιώντας τον αριθμό επιβεβαίωσης παραγγελίας.
TC13	Επαλήθευση λειτουργικότητας φόρμας επικοινωνίας	Εμφανίζεται ένα μήνυμα επιβεβαίωσης που υποδεικνύει ότι το ερώτημα έχει υποβληθεί.
TC14	Επαληθεύστε τη λειτουργία φιλτραρίσματος προϊόντων	Εμφανίζονται μόνο τα προϊόντα που ταιριάζουν με τα επιλεγμένα κριτήρια φιλτραρίσματος.

Τα σενάρια που δημιουργήθηκαν βλέπουμε ότι καλύπτουν τη βασική λειτουργικότητα του ιστότοπου. Δεν ελέγχουν όλα τα πιθανά σενάρια. Περιορίζονται στον λειτουργικό έλεγχο και δεν προχωρούν σε άλλες κατηγορίες ελέγχου.

Όπως και στις υπόλοιπες εφαρμογές ζητήθηκε να παραχθεί TypeScript κώδικα για τα συγκεκριμένα σενάρια έτσι ώστε να εκτελεστούν με χρήση της βιβλιοθήκης ανοιχτού κώδικα Playwright.



Για τα σενάρια αυτά δημιουργήθηκαν δύο αρχεία TypeScript :

1. Ένα αρχείο με το όνομα BardAI_tests_before.ts με περιεχόμενο τον παραχθέντα κώδικα χωρίς διορθώσεις.
2. Ένα αρχείο με το όνομα BardAI_tests_after.ts με περιεχόμενο τον παραχθέντα κώδικα που έχει διορθωθεί και να είναι εκτελέσιμος.

Παραδείγματα κώδικα αρχείων:

- BardAI_tests_before.ts

```
test('TC03: Checkout and complete an order', async () => {
  const browser = await Browser.launch();
  const page = await browser.newPage();
  // Go to the Saucedemo website
  await page.goto('https://www.saucedemo.com/');
  // Login to the website with a valid user
  await page.type('#username', 'standard_user');
  await page.type('#password', 'secret_sauce');
  await page.click('#login-button');
  // Add at least two items to the shopping cart
  await page.click('.product-list .product-item:first-child');
  await page.click('.add-to-cart-button');
  // Click on the Cart icon
  await page.click('#shopping_cart_button');
  // Click on the 'Proceed to Checkout' button
  await page.click('#checkout .action-button');
  // Enter the shipping address and billing information
  await page.type('#firstName', 'John');
  await page.type('#lastName', 'Doe');
  await page.type('#email', 'johndoe@example.com');
  await page.type('#zipCode', '12345');
  // Select the payment method
  await page.select('#paymentMethod', '#visa');
  // Click on the 'Place Order' button
  await page.click('#placeOrder');
  // Verify that the order is placed successfully
  expect(await page.textContent('#order_confirmation .subheading')).toBe('Thank you for your order!');
  // Close the browser
  await browser.close();
});
```

- BardAI_tests_after.ts

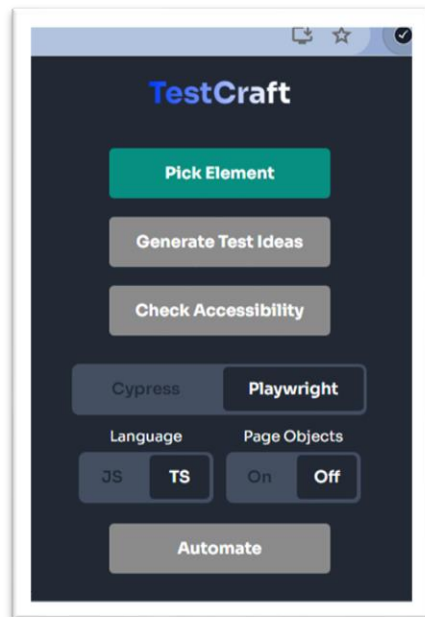
```
test('TC03: Checkout and complete an order', async ((page) => {
  // Go to the Saucedemo website
  await page.goto('https://www.saucedemo.com/');
  // Login to the website with a valid user
  await page.fill('#user-name', 'standard_user');
  await page.fill('#password', 'secret_sauce');
  await page.click('#login-button');
  // Add at least two items to the shopping cart
  await page.click('.inventory_item:nth-child(1) .btn_inventory');
  await page.click('.inventory_item:nth-child(2) .btn_inventory');
  // Click on the Cart icon
  await page.click('#shopping_cart_container > a');
  // Click on the 'Proceed to Checkout' button
  await page.click('#checkout');
  // Enter the shipping address and billing information
  await page.fill('#first-name', 'John');
  await page.fill('#last-name', 'Doe');
  await page.fill('#postal-code', '12345');
  // Click on the 'Place Order' button
  await page.click('#continue');
  await page.click('#finish');
  // Verify that the order is placed successfully
  expect(await page.textContent('#checkout_complete_container > h2')).toBe('Thank you for your order!');
  // Close the browser
  await page.close();
});
```

6.4 TestCraft.app

Το TestCraft είναι μια επέκταση προγράμματος περιήγησης που κάνει χρήση του μοντέλου GPT4. Χρησιμοποιεί την τεχνητή νοημοσύνη για να παράγει ιδέες σεναρίων ελέγχου

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

με βάση των elements της σελίδας. Επιλέγοντας ένα element από την σελίδα συλλέγει τον html κώδικα του element και χτίζει με δυναμικό τρόπο ένα ερώτημα προς το API της OpenAI.



Εικόνα 15: Διεπαφή Εφαρμογής TestCraft.app

Όπως φαίνεται και στην εικόνα έχουμε ρυθμίσει την εφαρμογή TestCraft για το element που θα επιλεγεί από την σελίδα ο κώδικας των σεναρίων ελέγχου να είναι σε γλώσσα TypeScript και συμβατός με την βιβλιοθήκη Playwright.

Για την παραγωγή σεναρίων επιλέχθηκαν μέσω του εργαλείου τα elements εκείνα που αφορούν την βασική λειτουργία του ιστότοπου SauceDemo:

- Φόρμα σύνδεσης χρήστη
- Κατάλογος Προϊόντων
- Λεπτομέρειες προϊόντος
- Μενού πλοήγησης
- Φόρμες των τριών βημάτων ολοκλήρωσης αγοράς

Στο σύνολο δημιούργησε 124 σεσάρια ελέγχου για όλα τα elements που επιλέχθηκαν από τα οποία τα 65 έχουν την δυνατότητα να εκτελεστούν βάση των δυνατοτήτων της Playwright. Τα σεσάρια αυτά τα σεσάρια που δεν μπορούν να πραγματοποιηθούν ήταν χειροκίνητα σεσάρια ελέγχου, σεσάρια που απαιτούσαν την χρήση περισσότερων εργαλείων για την προσομοίωση τους ή σεσάρια που απαιτούσαν πρόσβαση στον κώδικα, το οποίο δεν είναι αντικείμενο μελέτης της παρούσας εργασίας. Κάποια παραδείγματα από τα σεσάρια αυτά είναι :

- Προσομοιώστε μια επίθεση δέσμης ενεργειών μεταξύ ιστότοπων (XSS) για να διασφαλίσετε ότι η φόρμα σύνδεσης είναι ασφαλής έναντι τέτοιων απειλών.
- Δοκιμάστε τη λειτουργία σύνδεσης σε ώρες αιχμής για να αξιολογήσετε την απόδοση υπό φόρτωση.
- Δοκιμάστε τη λειτουργικότητα του κουμπιού "Προσθήκη στο καλάθι" για ένα προϊόν που μόλις εξαντλήθηκε ή αφαιρέθηκε από το απόθεμα.
- Δοκιμάστε τη συμπεριφορά του καλαθιού όταν αλλάζει η τιμή ενός προϊόντος ενώ βρίσκεται στο καλάθι (π.χ. κατά τη διάρκεια μιας ενημέρωσης τιμής ή προσφοράς).

Τα σενάρια που ανταποκρίνονται στο αίτημα που παραθέσαμε στην εφαρμογή παρατίθενται στον παρακάτω πίνακα.

Πίνακας 5: Σενάρια Ελέγχου TestCraft.app

Στοιχεία Σελίδας	Test Case ID	Περιγραφή Σεναρίου Ελέγχου
Σύνδεση χρήστη	TC01	Βεβαιωθείτε ότι ένας χρήστης μπορεί να συνδεθεί με ένα έγκυρο όνομα χρήστη (π.χ. 'standard_user') και τον σωστό κωδικό πρόσβασης ('secret_sauce').
	TC02	Βεβαιωθείτε ότι το κείμενο κράτησης θέσης 'Όνομα χρήστη' είναι ορατό στο πεδίο εισαγωγής ονόματος χρήστη όταν το πεδίο είναι κενό.
	TC03	Βεβαιωθείτε ότι το κείμενο κράτησης θέσης "Κωδικός πρόσβασης" είναι ορατό στο πεδίο εισαγωγής κωδικού πρόσβασης όταν το πεδίο είναι κενό.
	TC04	Επιβεβαιώστε ότι η πληκτρολόγηση στο πεδίο ονόματος χρήστη λειτουργεί σωστά και αντικατοπτρίζει το εισαγόμενο κείμενο.
	TC05	Επιβεβαιώστε ότι η πληκτρολόγηση στο πεδίο κωδικού πρόσβασης λειτουργεί σωστά και αντικατοπτρίζει το εισαγόμενο κείμενο, με χαρακτήρες που καλύπτονται καθώς πληκτρολογεί ο χρήστης.
	TC06	Επαληθεύστε ότι μπορείτε να κάνετε κλικ στο κουμπί σύνδεσης και υποβάλετε τα δεδομένα φόρμας.
	TC07	Βεβαιωθείτε ότι μετά την επιτυχή σύνδεση, ο χρήστης κατευθύνεται στην κατάλληλη σελίδα σύνδεσης ή στον πίνακα ελέγχου.
	TC08	Ελέγξτε ότι τα χαρακτηριστικά αυτόματης διόρθωσης και αυτόματης κεφαλαιοποίησης είναι απενεργοποιημένα τόσο για τα πεδία ονόματος χρήστη όσο και για τα πεδία κωδικού πρόσβασης.
	TC09	Προσπαθήστε να συνδεθείτε με έγκυρο όνομα χρήστη και λανθασμένο κωδικό πρόσβασης για να βεβαιωθείτε ότι η σύνδεση θα αποτύχει.
	TC10	Προσπαθήστε να συνδεθείτε με μη έγκυρο όνομα χρήστη και τον σωστό κωδικό πρόσβασης για να βεβαιωθείτε ότι η σύνδεση θα αποτύχει.
	TC11	Προσπαθήστε να συνδεθείτε τόσο με μη έγκυρο όνομα χρήστη όσο και με λανθασμένο κωδικό πρόσβασης για να βεβαιωθείτε ότι η σύνδεση θα αποτύχει.
	TC12	Προσπαθήστε να συνδεθείτε με το πεδίο ονόματος χρήστη κενό για να επιβεβαιώσετε ότι εμφανίζεται ένα μήνυμα σφάλματος.
	TC13	Προσπαθήστε να συνδεθείτε με το πεδίο κωδικού πρόσβασης κενό για να επιβεβαιώσετε ότι εμφανίζεται ένα μήνυμα σφάλματος

	TC14	Ελέγξτε για τη λειτουργία εμφάνισης του μηνύματος σφάλματος πληκτρολογώντας εσφαλμένα διαπιστευτήρια σύνδεσης.
	TC15	Βεβαιωθείτε ότι η φόρμα σύνδεσης δεν επιτρέπει επιθέσεις SQL injection ή script injection εισάγοντας τέτοια ωφέλιμα φορτία στα πεδία ονόματος χρήστη ή κωδικού πρόσβασης.
Κατάλογος Προϊόντων	TC16	Βεβαιωθείτε ότι κάνοντας κλικ στην εικόνα ενός προϊόντος ο χρήστης κατευθύνεται σε μια λεπτομερή προβολή αυτού του προϊόντος.
	TC17	Επιβεβαιώστε ότι κάνοντας κλικ στον σύνδεσμο ονόματος προϊόντος μεταβαίνετε επίσης στη λεπτομερή προβολή του προϊόντος.
	TC18	Βεβαιωθείτε ότι η περιγραφή του προϊόντος είναι ορατή και ταιριάζει με το προϊόν που αναφέρεται.
	TC19	Check if the "Add to cart" button adds the respective product to the shopping cart.
	TC20	Επαληθεύστε ότι η τιμή του προϊόντος εμφανίζεται σωστά για κάθε είδος.
	TC21	Ελέγξτε ότι το εναλλακτικό κείμενο για κάθε εικόνα προϊόντος περιγράφει με ακρίβεια το προϊόν.
	TC22	Επιβεβαιώστε ότι το κουμπί "Προσθήκη στο καλάθι" για κάθε προϊόν έχει ένα μοναδικό αναγνωριστικό που αντιστοιχεί στο προϊόν.
	TC23	Προσπαθήστε να πλοηγηθείτε σε έναν κατεστραμμένο σύνδεσμο στην εικόνα προϊόντος και στο όνομα προϊόντος για να επαληθεύσετε τον σωστό χειρισμό σφαλμάτων.
Μενού	TC24	Βεβαιωθείτε ότι το μενού είναι ορατό και προσβάσιμο στη σελίδα.
	TC25	Βεβαιωθείτε ότι μπορείτε να κάνετε κλικ στον σύνδεσμο "Όλα τα στοιχεία" και ανακατευθύνει τον χρήστη στην κατάλληλη ενότητα ή σελίδα.
	TC26	Βεβαιωθείτε ότι ο σύνδεσμος "Πληροφορίες" μεταβαίνει στη διεύθυνση URL "https://saucelabs.com/" όταν κάνετε κλικ.
	TC27	Επιβεβαιώστε ότι ο σύνδεσμος "Αποσύνδεση" ενεργοποιεί τη λειτουργία αποσύνδεσης και αποσυνδέει τον χρήστη.
	TC28	Επαληθεύστε ότι ο σύνδεσμος "Επαναφορά κατάστασης εφαρμογής" επαναφέρει την εφαρμογή στην προεπιλεγμένη της κατάσταση χωρίς σφάλματα.
	TC29	Προσπαθήστε να μεταβείτε στον σύνδεσμο "Πληροφορίες" με αργή σύνδεση στο διαδίκτυο για να ελέγξετε εάν υπάρχει ένδειξη φόρτωσης ή μήνυμα χρονικού ορίου.
	TC30	Κάντε κλικ στο σύνδεσμο "Αποσύνδεση" αφού η συνεδρία έχει ήδη λήξει για να δείτε εάν χειρίζεται το σενάριο ομαλά.

	TC31	Ελέγξτε τις συνδέσεις μενού με εσφαλμένη ή τροποποιημένη τιμή href για να δείτε πώς η εφαρμογή χειρίζεται κατεστραμμένες συνδέσεις.
Καλάθι αγορών	TC32	Βεβαιωθείτε ότι το κουμπί "Συνέχεια αγορών" μεταφέρει τον χρήστη πίσω στη σελίδα αγορών όταν κάνει κλικ.
	TC33	Επιβεβαιώστε ότι το κουμπί "Checkout" μεταφέρει τον χρήστη στη σελίδα ολοκλήρωσης αγοράς όταν κάνετε κλικ.
	TC34	Ελέγξτε ότι στο καλάθι εμφανίζεται η σωστή ποσότητα (QTY) και περιγραφή των προϊόντων που προστέθηκαν από τον χρήστη.
	TC35	Βεβαιωθείτε ότι το καλάθι ενημερώνεται σωστά όταν προστίθενται ή αφαιρούνται προϊόντα.
	TC36	Επαληθεύστε ότι οι ετικέτες ποσότητας και περιγραφής του καλαθιού είναι ορατές και ευανάγνωστες.
	TC37	Επιβεβαιώστε ότι οι εικόνες για το κουμπί "Συνέχεια αγορών" φορτώνονται σωστά και αντιπροσωπεύουν μια οπτική ένδειξη για επιστροφή.
	TC38	Ελέγξτε ότι εμφανίζεται το εναλλακτικό κείμενο για την εικόνα "Επιστροφή" εάν αποτύχει η φόρτωση της εικόνας.
	TC39	Βεβαιωθείτε ότι κάνοντας κλικ στο κουμπί "Συνέχεια αγορών" με άδειο καλάθι δεν προκαλεί σφάλματα.
	TC40	Ελέγξτε ότι το κουμπί "Checkout" είναι απενεργοποιημένο ή παρέχει ένα κατάλληλο μήνυμα όταν το καλάθι είναι άδειο.
	TC41	Προσπαθήστε να ολοκληρώσετε τη διαδικασία αγοράς χωρίς προϊόντα στο καλάθι και βεβαιωθείτε ότι εμφανίζεται ένα κατάλληλο μήνυμα ή αποτρέπεται η ολοκλήρωση.
Ολοκλήρωση παραγγελίας Βήμα 1 ^ο	TC42	Βεβαιωθείτε ότι η φόρμα δέχεται έγκυρο όνομα, επώνυμο και ταχυδρομικό κώδικα και επιτρέπει στον χρήστη να προχωρήσει κάνοντας κλικ στο κουμπί "Συνέχεια".
	TC43	Βεβαιωθείτε ότι οι δυνατότητες αυτόματης διόρθωσης και αυτόματης κεφαλοποίησης είναι απενεργοποιημένες για τα πεδία εισαγωγής όταν ο χρήστης αρχίζει να πληκτρολογεί.
	TC44	Επιβεβαιώστε ότι κάνοντας κλικ στο κουμπί "Ακύρωση" ανακατευθύνεται ο χρήστης στην προηγούμενη σελίδα ή ακυρώνεται η διαδικασία ολοκλήρωσης αγοράς.
	TC45	Επιβεβαιώστε ότι κάνοντας κλικ στο κουμπί "Ακύρωση" ανακατευθύνεται ο χρήστης στην προηγούμενη σελίδα ή ακυρώνεται η διαδικασία ολοκλήρωσης αγοράς.

	TC46	Επαληθεύστε ότι η φόρμα διατηρεί την εισαγωγή στοιχείων από το χρήστη εάν υπάρχει σφάλμα μετά την υποβολή και την επαναφόρτωση της σελίδας.
	TC47	Προσπαθήστε να υποβάλετε τη φόρμα με ένα ή περισσότερα κενά πεδία για να βεβαιωθείτε ότι ο χρήστης δεν μπορεί να προχωρήσει και εμφανίζεται ένα κατάλληλο μήνυμα λάθους.
	TC48	Εισαγάγετε αριθμούς ή ειδικούς χαρακτήρες στα πεδία ονόματος και επωνύμου για να επαληθεύσετε ότι εμφανίζεται ένα μήνυμα σφάλματος και ότι η φόρμα δεν υποβάλλεται.
	TC49	Εισαγάγετε μια μη έγκυρη μορφή για τον ταχυδρομικό κώδικα, όπως χρησιμοποιώντας γράμματα σε ένα πεδίο μόνο αριθμητικά, για να ελέγξετε για χειρισμό σφαλμάτων.
	TC50	Προσπαθήστε να υποβάλετε τη φόρμα με υπερβολικά μεγάλες τιμές στα πεδία εισαγωγής για να ελέγξετε για τυχόν ευπάθειες υπερχείλισης buffer.
	TC51	Δοκιμάστε τη φόρμα εισάγοντας κώδικα HTML ή JavaScript στα πεδία εισαγωγής για να βεβαιωθείτε ότι η φόρμα δεν είναι ευάλωτη σε επιθέσεις εισαγωγής κώδικα.
Ολοκλήρωση παραγγελίας Βήμα 2 ^ο	TC52	Βεβαιωθείτε ότι οι ετικέτες ποσότητας και περιγραφής είναι ορατές και σωστά επισημασμένες ως "Ποσότητα" και "Περιγραφή".
	TC53	Επιβεβαιώστε ότι τα στοιχεία πληρωμής εμφανίζονται σωστά και περιέχουν το κείμενο κράτησης θέσης "SauceCard #31337".
	TC54	Βεβαιωθείτε ότι οι πληροφορίες αποστολής είναι ορατές με το κείμενο "Free Pony Express Delivery!".
	TC55	Βεβαιωθείτε ότι η ενότητα συνόλου τιμής εμφανίζει σωστά το σύνολο στοιχείου, τον φόρο και το συνολικό σύνολο ως 0,00 \$ αρχικά.
	TC56	Ελέγξτε ότι το κουμπί "Ακύρωση" είναι ορατό και με δυνατότητα κλικ και ότι κατευθύνει τον χρήστη πίσω στην προηγούμενη σελίδα ή ενέργεια.
	TC57	Βεβαιωθείτε ότι το κουμπί "Τέλος" είναι ορατό, με δυνατότητα κλικ και οδηγεί στην ολοκλήρωση της διαδικασίας ολοκλήρωσης αγοράς.
	TC58	Βεβαιωθείτε ότι οι ετικέτες συνοπτικών πληροφοριών για την πληρωμή, την αποστολή και τη συνολική τιμή είναι σαφώς ορατές και σωστά επισημασμένες.
	TC59	Προσπαθήστε να προχωρήσετε με την ολοκλήρωση αγοράς κάνοντας κλικ στο "Τέλος" χωρίς προϊόντα στο καλάθι για να δείτε εάν χειρίζεται σενάρια κενού καλάθιού.
	TC60	Προσπαθήστε να αλληλεπιδράσετε με τη διαδικασία ολοκλήρωσης αγοράς κατά τη διάρκεια μιας

		διακοπής δικτύου για να δείτε εάν εμφανίζονται τα κατάλληλα μηνύματα σφάλματος.
Ολοκλήρωση της παραγγελίας	TC61	Βεβαιωθείτε ότι εμφανίζεται το μήνυμα "Ευχαριστούμε για την παραγγελία σας!" μετά από μια επιτυχημένη διαδικασία ολοκλήρωσης αγοράς.
	TC62	Ελέγξτε εάν η εικόνα του Pony Express εμφανίζεται σωστά και ταιριάζει με το αναμενόμενο σχέδιο.
	TC63	Βεβαιωθείτε ότι το κείμενο "Η παραγγελία σας έχει αποσταλεί και θα φτάσει τόσο γρήγορα όσο το πόνι μπορεί να φτάσει εκεί!" είναι ορατό και σωστά μορφοποιημένο.
	TC64	Επιβεβαιώστε ότι το κουμπί "Επιστροφή στην αρχική σελίδα" είναι ορατό και με δυνατότητα κλικ.
	TC65	Επαληθεύστε ότι κάνοντας κλικ στο κουμπί "Πίσω σπίτι" ανακατευθύνεται ο χρήστης στην αρχική σελίδα ή στη λίστα προϊόντων.

Τα σενάρια που δημιουργήθηκαν ανταποκρίνονται σε μεγάλο βαθμό στη βασική λειτουργικότητα της εφαρμογής. Βέβαια, τα περισσότερα από αυτά ανήκουν στην κατηγορία του ελέγχου της διεπαφής.

Για τα σενάρια αυτά δημιουργήθηκαν δύο αρχεία TypeScript :

1. Ένα αρχείο με το όνομα TestCraft_tests_before.ts με περιεχόμενο τον παραχθέντα κώδικα χωρίς διορθώσεις.
2. Ένα αρχείο με το όνομα TestCraft_tests_after.ts με περιεχόμενο τον παραχθέντα κώδικα που έχει διορθωθεί και να είναι εκτελέσιμος.

Παραδείγματα κώδικα αρχείων:

- TestCraft_tests_before.ts

```
test('TC35: Ensure that the cart updates correctly when items are added or removed', async ({ page }) => {
  // Add logic to add an item to the cart
  // Add logic to remove an item from the cart
  // Add assertions to confirm the cart updates correctly
});
```

- TestCraft_tests_after.ts

```
test('TC35: Ensure that the cart updates correctly when items are added or removed', async ({ page }) => {
  // Add logic to add an item to the cart
  // Add logic to remove an item from the cart
  // Add assertions to confirm the cart updates correctly
  // Click on the first product
  await page.click('.inventory_item:nth-child(1) .btn_inventory');
  expect(await page.textContent('#shopping_cart_container')).toBe('1');
});
```

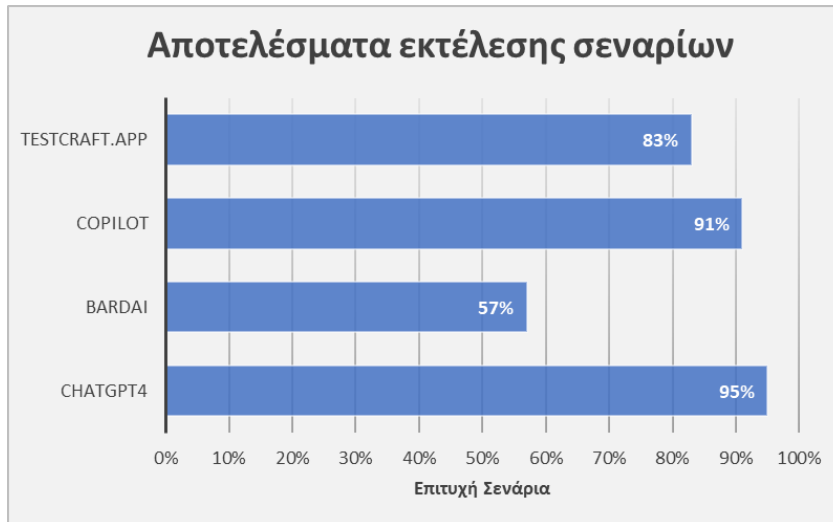
7. ΑΞΙΟΛΟΓΗΣΗ ΤΩΝ ΣΕΝΑΡΙΩΝ ΕΛΕΓΧΟΥ

Για την αξιολόγηση των αποτελεσμάτων που παρήγαγε η κάθε εφαρμογή θα χρησιμοποιηθούν δυο παράμετροι :

1. Το εύρος των περιπτώσεων και κατηγοριών ελέγχου των σεναρίων.
2. Την ορθότητα και τον βαθμό παρέμβασης μας στον κώδικα των σεναρίων.

7.1 Εύρος κάλυψης σεναρίων

Από την εκτέλεση των σεναρίων προκύπτει ο παρακάτω πίνακας.



Σχήμα 1: Ποσοστά Επιτυχίας Σεναρίων

- **ChatGPT4**

Τα σεναρία που παρήχθησαν από τη συγκεκριμένη εφαρμογή βλέπουμε ότι καλύπτουν σε αρκετά ικανοποιητικό βαθμό τη λειτουργικότητα της εφαρμογής, γνωρίζοντας τι πρέπει να ελεγχθεί και το πώς λειτουργεί η εφαρμογή. Επεκτάθηκε επίσης στην παραγωγή σεναρίων που καλύπτουν μεγάλο εύρος κατηγοριών ελέγχου, όπως:

- Λειτουργικού
- Διεπαφής
- Συμβατότητας
- Απόδοσης
- Ασφάλειας
- Προσβασιμότητας

Από τα 45 σεναρία που δημιούργησε τα 8 εκτός του πεδίου εφαρμογής διότι δεν μπορούσαν να εκτελεστούν με το τη χρήση του framework Playwright που είχαμε ορίσει εξ αρχής στο αίτημά μας προς την εφαρμογή. Από τα 37 σεναρία που ήταν συμβατά με το framework μόνο τα δύο δεν ανταποκρινόταν στις προδιαγραφές λειτουργίας της εφαρμογής και είναι αυτά που δεν πέρασαν επιτυχώς. Τα σεναρία αυτά είναι:

✘ SauceDemo Frontend Tests - Additional > TC07: Products should be filterable	30.0s
ChatGPT4_tests_after.spec.ts:75	
✘ SauceDemo Frontend Tests - Checkout and Usability > TC15: Different payment methods should be functional	30.0s
ChatGPT4_tests_after.spec.ts:182	

- **Copilot**

Τα σενάρια ελέγχου που μας πρότεινε αυτή η εφαρμογή καλύπτουν σε ένα ικανοποιητικό βαθμό την βασική λειτουργικότητα της εφαρμογής. Είχε γνώση της λειτουργίας της εφαρμογής. Περιορίστηκε όμως μόνο στον έλεγχο λειτουργικότητας και δεν μας πρότεινε σενάρια άλλων κατηγοριών ελέγχου. Τα σενάρια ελέγχου λειτουργικότητας απευθυνόταν σε πολύ βασικές λειτουργίες της εφαρμογής. Και τα 12 σενάρια που μας πρότεινε μπορούσαν να υλοποιηθούν με το framework Playwright. Από τα 12 σενάρια μόνο το 1 δεν ανταποκρινόταν στην λειτουργικότητα της εφαρμογής και αυτό ήταν που απέτυχε στον έλεγχο:

✘ TC11: Change product quantity Copilot_test_after.spec.ts:220	30.0s
---	-------

- **BardAI**

Το πειραματικό BardAI μας πρότεινε συνολικά 14 σενάρια. Από αυτά τα 6 δεν ανταποκρινόταν στις προδιαγραφές της εφαρμογής και έλεγχε περιπτώσεις που δεν υπήρχαν σε αυτή. Προκύπτει το συμπέρασμα ότι δεν γνώριζε ακριβώς την λειτουργικότητά της και τα σενάρια ήταν προτάσεις που σχετίζονται με την λειτουργικότητα γενικά ιστότοπων ηλεκτρονικού εμπορίου και όχι για τον συγκεκριμένο. Τα υπόλοιπα 8 που ανταποκρινόταν στην λειτουργικότητα της εφαρμογής καλύπτουν ένα πολύ μικρό εύρος ελέγχων και αφορούν με μικρό αριθμό σεναρίων τις κατηγορίες ελέγχου λειτουργικότητας και διεπαφής. Τα 6 σενάρια που δεν ανταποκρινόταν στις προδιαγραφές της εφαρμογής και απέτυχαν είναι τα παρακάτω:

✘ Login form tests > TC07: Check product quantity limitations BardAI_test_after.spec.ts:148	30.0s
✘ Login form tests > TC10: Verify user registration functionality BardAI_test_after.spec.ts:241	867ms
✘ Login form tests > TC11: Verify forgotten password functionality BardAI_test_after.spec.ts:269	849ms
✘ Login form tests > TC12: Verify order tracking functionality BardAI_test_after.spec.ts:295	1.1s
✘ Login form tests > TC13: Verify contact form functionality BardAI_test_after.spec.ts:341	992ms
✘ Login form tests > TC14: Verify product filtering functionality BardAI_test_after.spec.ts:366	955ms

- **TestCraft.app**

Το TestCraft.app πρότεινε μεγάλο πλήθος σεναρίων ελέγχου και αυτό ήταν αναμενόμενο διότι τα σενάρια που ζητούταν να παράξει αφορούσαν element της σελίδας και έτσι παρήγαγε σενάρια για την κάθε επιλογή ξεχωριστά. Συνολικά μας πρότεινε 124 σενάρια,

από αυτά βέβαια τα 59 δεν μπορούσαν να υλοποιηθούν στα με χρήση του framework που είχαμε ορίσει στην εφαρμογή. Για τα 65 που μπορούσαν να υλοποιηθούν καλύπτουν σε μεγάλο βαθμό την λειτουργικότητα της εφαρμογής και κάνει επίσης προτάσεις σεναρίων σε εύρος κατηγοριών ελέγχου όπως:

- Λειτουργικού
- Διεπαφής
- Απόδοσης
- Ασφάλειας
- Προσβασιμότητας

Παρατηρούμε επίσης ότι τα σενάρια τα οποία απέτυχαν στην εκτέλεση είναι σωστά και εντοπίζουν προβλήματα και τρωτά σημεία του ισότοπου ενώ τα σενάρια των υπολοίπων εφαρμογών που απέτυχαν απλά δεν ανταποκρινόταν στην λειτουργικότητα του ιστότοπου. Τα σενάρια της εφαρμογής TestCraft.app που απέτυχαν είναι:

✘ Menu Tests > TC29: Try to navigate to the "About" link with a slow internet connection to check if there's a loading indicator or timeout message. TestCraft_test_after.spec.ts:228	9.9s
✘ Menu Tests > TC30: Click the "Logout" link after the session has already expired to see if it handles the scenario gracefully. TestCraft_test_after.spec.ts:237	7.6s
✘ Menu Tests > TC31: Test the menu links with an incorrect or modified href value to see how the application handles broken links. TestCraft_test_after.spec.ts:248	8.1s
✘ Shopping Cart Page > TC40: Check that the "Checkout" button is disabled or provides an appropriate message when the cart is empty TestCraft_test_after.spec.ts:329	8.5s
✘ Shopping Cart Page > TC41: Attempt to navigate to the checkout page without any items in the cart and ensure an appropriate message is displayed or navigation is prevented TestCraft_test_after.spec.ts:338	9.4s
✘ Checkout form tests > TC48: Input invalid characters name fields TestCraft_test_after.spec.ts:409	9.8s
✘ Checkout form tests > TC49: Enter invalid zip/postal code format TestCraft_test_after.spec.ts:418	8.2s
✘ Checkout form tests > TC50: Submit form with excessively long values TestCraft_test_after.spec.ts:427	8.3s
✘ Checkout form tests > TC51: Test form for code injection vulnerabilities TestCraft_test_after.spec.ts:437	8.7s
✘ Checkout Summary Tests > TC59: Attempt to proceed with checkout with an empty cart TestCraft_test_after.spec.ts:510	8.1s
✘ Checkout Summary Tests > TC60: Interact with checkout during network interruption TestCraft_test_after.spec.ts:519	7.8s

7.2 Ορθότητα Κώδικα

Για την αξιολόγηση του κώδικα των σεναρίων που παρήγαγαν οι εφαρμογές εφαρμόσαμε την ακόλουθη διαδικασία.

- Δημιουργία 2 αρχείων κώδικα, όπως προαναφέρθηκε στο κεφάλαιο 6. Ένα αρχείο κώδικα χωρίς διορθώσεις και το διορθωμένο και προσαρμοσμένο αρχείο που εκτελέστηκε.
- Σύγκριση των 2 αρχείων και εξαγωγή αναφοράς με την χρήση της εφαρμογής Compare It 4 της Grig Software.

Το Compare It είναι ένα ευέλικτο εργαλείο σύγκρισης αρχείων που έχει σχεδιαστεί για λειτουργικά συστήματα Windows. Χρησιμοποιείται κυρίως για αρχεία κειμένου με αντίθεση, υπογραμμίζει τις διαφορές μεταξύ εγγράφων τόσο σε επίπεδο γραμμής όσο και σε επίπεδο χαρακτήρων, καθιστώντας το ιδιαίτερα χρήσιμο για κωδικοποιητές, συγγραφείς και επαγγελματίες που διαχειρίζονται αναθεωρήσεις εγγράφων. Παράλληλα με το κείμενο, μπορεί να χειριστεί διάφορες μορφές αρχείων, συμπεριλαμβανομένων των δυαδικών αρχείων. Τα βασικά χαρακτηριστικά περιλαμβάνουν επισήμανση σύνταξης για εύκολη σύγκριση κώδικα, προσαρμόσιμες ρυθμίσεις σύγκρισης και ενσωμάτωση.

Παρακάτω παρουσιάζονται τα στατιστικά για κάθε εφαρμογή ξεχωριστά και το τέλος ένα συγκεντρωτικό διάγραμμα με τα ποσοστά των διορθώσεων. Σχετικά με τα στατιστικά όπου αναφέρετε "Target" υποδηλώνει το αρχείο χωρίς τις διορθώσεις και τις αλλαγές και όπου αναφέρετε "Source" είναι το διορθωμένο αρχείο που εκτελέστηκε.

ChatGPT4

Πίνακας 6: Στατιστικά κώδικα ChatGPT4

Statistics
Total changed lines : 69 (13%); Same : 429
Target Only : 1
Source Only : 37
Changed : 31

Το συνολικό ποσοστό διορθώσεων/αλλαγών για τον κώδικα του ChatGPT βλέπουμε ότι είναι 13%. Προσθέσαμε 37 γραμμές κώδικα, αφαιρέσαμε 1 γραμμή και έγιναν αλλαγές σε 31 γραμμές κώδικα που προϋπήρχαν. Η παρατήρηση για το συγκεκριμένο κώδικα είναι ότι οι περισσότερες αλλαγές έπρεπε να γίνουν στους selectors που αφορούν τον ιστότοπο SauceDemo και την έλλειψη προ απαιτούμενων βημάτων σε κάποια σενάρια.

Copilot

Πίνακας 7: Στατιστικά κώδικα Copilot

Statistics
Total changed lines : 32 (11%); Same : 256
Target Only : 9
Source Only : 3
Changed : 20

Το συνολικό ποσοστό διορθώσεων/αλλαγών για τον κώδικα του Copilot βλέπουμε ότι είναι 11%. Προσθέσαμε 3 γραμμές κώδικα, αφαιρέσαμε 9 γραμμές και έγιναν αλλαγές σε 20

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

γραμμές κώδικα που προϋπήρχαν. Οι περισσότερες αλλαγές που χρειάστηκε να γίνουν ήταν στα assertions του κώδικα και σε εντολές που δεν υπήρχαν στην βιβλιοθήκη του Playwright.

BardAI

Πίνακας 8: Στατιστικά κώδικα BardAI

Statistics
Total changed lines : 118 (27%); Same : 308
Target Only : 25
Source Only : 11
Changed : 82

Το συνολικό ποσοστό διορθώσεων/αλλαγών για τον κώδικα του BardAI βλέπουμε ότι είναι 27%. Προσθέσαμε 11 γραμμές κώδικα, αφαιρέσαμε 25 γραμμές και έγιναν αλλαγές σε 82 γραμμές κώδικα που προϋπήρχαν. Στον συγκεκριμένο κώδικα οι περισσότερες αλλαγές αφορούσαν κατά κύριο λόγο του selectors και στη χρήση εντολών που δεν υπάρχουν στην βιβλιοθήκη του Playwright.

TestCraft.app

Πίνακας 9: Στατιστικά κώδικα TestCraft.app

Statistics
Total changed lines : 114 (19%); Same : 464
Target Only : 7
Source Only : 86
Changed : 21

Το συνολικό ποσοστό διορθώσεων/αλλαγών για τον κώδικα του TestCraft.app βλέπουμε ότι είναι 19%. Προσθέσαμε 86 γραμμές κώδικα, αφαιρέσαμε 7 γραμμές και έγιναν αλλαγές σε 21 γραμμές κώδικα που προϋπήρχαν. Στο συγκεκριμένο κώδικα έλλειπαν από πολλά σενάρια το assertion δηλαδή το αναμενόμενο αποτέλεσμα του ελέγχου, στις περισσότερες περιπτώσεις μας ανέφερε με σχόλια τι assertion να προσθέσουμε αλλά έλλειπε ο κώδικας. Επίσης κάτι άλλο που παρατηρήθηκε είναι ότι δεν υπήρχε συνοχή μεταξύ των σεναρίων την έλλειψη προ απαιτούμενων βημάτων με κώδικα για την υλοποίηση κάποιων σεναρίων.

Παρατίθεται συγκεντρωτικός πίνακας με τα αποτελέσματα των εφαρμογών:

Πίνακας 10: Συγκεντρωτικός πίνακας αποτελεσμάτων

Εφαρμογή	Πλήθος Σεναρίων	Κατηγορίες Ελέγχου	Ποσοστό διόρθωσης κώδικα	Large Language Models
ChatGPT4	37	6	13%	GPT-4
Copilot	12	1	11%	GPT-4
BardAI	14	1	27%	PaLM2
TestCraft.app	65	5	19%	GPT-4

8. ΣΥΜΠΕΡΑΣΜΑΤΑ

Το πρώτο συμπέρασμα που προκύπτει με την ενσωμάτωση της τεχνητής νοημοσύνης στο κύκλο ζωής του ελέγχου λογισμικού είναι η συνεισφορά της στην επιτάχυνση του μέσω της χρήσης διαφόρων τεχνικών και εφαρμογών της τεχνητής νοημοσύνης σε διάφορες εργασίες των σταδίων του. Καθιστώντας λόγω των αυξημένων απαιτήσεων και ανταγωνισμού της εποχής μας αυτήν την εξέλιξη αυτή πολύ σημαντική.

Μέσω της χρήσης εφαρμογών Generative AI που προβήκαμε για την δημιουργία σεναρίων ελέγχου για τον ιστότοπο SauceDemo, κατανοήσαμε και είδαμε στην πράξη πως εφαρμογές που χρησιμοποιούν Large Language Models μπορούν να συνεισφέρουν στην επιτάχυνση και ολοκλήρωση του σημαντικού και συνήθως χρονοβόρου σταδίου της Δημιουργίας Σεναρίων Ελέγχου.

Ποιο συγκεκριμένα όπως είδαμε και στον συγκεντρωτικό πίνακα, τα προτεινόμενα σενάρια ελέγχου του Copilot και του BardAI απευθύνονται αποκλειστικά στην κατηγορία του ελέγχου λειτουργικότητας του ιστότοπου. Αντίθετα παρατηρούμε ότι οι εφαρμογές ChatGPT4 και TestCraft.app πρότειναν σενάρια ελέγχου που κάλυπταν ένα μεγάλο φάσμα κατηγοριών ελέγχου λογισμικού και με αρκετά ικανοποιητικό πλήθος σεναρίων για κάθε μία από αυτές.

Παρατηρούμε σχετικά με τον κώδικα που παρήγαγε η κάθε εφαρμογή επίσης ότι, το μοντέλο GPT4 υπερέχει του PaLM2. Οι εφαρμογές που χρησιμοποιούν το GPT4 έχτισαν κώδικα που απαιτούσε μικρότερη παρέμβαση και λιγότερες διορθώσεις. Επίσης άξια αναφοράς είναι η παρατήρηση για τις τρεις εφαρμογές που χρησιμοποιούν το ίδιο Large Language Model GPT4 ότι εξήγαγαν αρκετά διαφορετικά αποτελέσματα το οποίο οφείλετε στον τρόπο που κάθε εταιρία διαχειρίζεται το μοντέλο αυτό.

Συνοψίζοντας όλα όσα αναπτύχθηκαν στην εργασία αυτή, καταλήγουμε στο συμπέρασμα ότι οι εφαρμογές Gerative AI μπορούν να βοηθήσουν και να επιταχύνουν την υλοποίηση και ολοκλήρωση του σταδίου της “Δημιουργίας Σεναρίων Ελέγχου” σημαντικά. Χρησιμοποιώντας ως περίπτωση τα αποτελέσματα του ChatGPT4, παρατηρούμε ότι μπορούν να καλυφθούν βασικά σενάρια ελέγχου διαφόρων κατηγοριών και με μικρό ποσοστό παρέμβασης στον κώδικα που παράχθηκε για αυτά. Καθιστώντας τα χρησιμοποιήσιμα για την αυτοματοποίηση τους εν συνέχεια, όπως και είδαμε κάνοντας χρήση του framework Playwright. Δεν μπορούμε να βασιστούμε ακόμα πλήρως σε αυτά τα εργαλεία αλλά μπορούν να παρέχουν μια σημαντική βοήθεια για ένα μηχανικό ελέγχου λογισμικού εξασφαλίζοντάς του χρόνο για να επικεντρωθεί στην ανάλυση και υλοποίηση πιο εξειδικευμένων και απαιτητικών σεναρίων ελέγχου.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	Ελληνικός Όρος
Generative	Παραγωγική
Framework	Πλαίσιο Εργασίας
Chatbot	Ρομπότ Συνομιλίας
Elements	Στοιχεία διεπαφής
Cloud	Υπολογιστικό Νέφος
UI Regression	Παλινδρόμηση Διεπαφής Χρήστη

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

GPS	Global Positioning System
SUT	System Under Testing
API	Application Programming Interface
CI/CD	Continuous Integration and Continuous Delivery/Continuous Deployment
STLC	Software Testing Life Cycle
UI	User Interface

ΠΑΡΑΡΤΗΜΑ Ι

Τα αρχεία με τον κώδικα που χρησιμοποιήθηκαν στην παρούσα εργασία μπορούν να προσπελαστούν μέσω του URL που ακολουθεί:

<https://github.com/dimmichlds/Test-Cases-.git>

ΠΑΡΑΡΤΗΜΑ II

Στο παράρτημα II παρουσιάζονται οι στιγμιότυπα της κονσόλας του Playwright από την εκτέλεση των σεναρίων κάθε εφαρμογής.

Test Case	Duration
✓ SauceDemo Frontend Tests	
✓ TC01: Valid login should succeed	5.5s
✓ TC02: Invalid login should fail	5.2s
✓ TC03: Login with empty credentials should fail	4.8s
✓ TC04: Logout should redirect to login page	3.0s
✓ TC05: All products should be listed on inventory page	2.0s
✗ SauceDemo Frontend Tests - Additional	
✓ TC06: Products should be sortable	2.4s
✗ TC07: Products should be filterable	30.0s
✓ TC08: Clicking on a product should show its details	2.1s
✓ TC09: Product details should be correct	2.2s
✓ TC10: Adding item to cart should increase cart count	2.2s
✓ SauceDemo Frontend Tests - Cart and Checkout	
✓ TC11: Removing items from the cart should update cart count	2.3s
✓ TC12: Cart should update correctly with multiple items	2.6s
✓ TC13: Cart should retain items after logout and login	4.0s
✓ TC14: Should be able to enter shipping information during checkout	2.9s
✗ SauceDemo Frontend Tests - Checkout and Usability	
✗ TC15: Different payment methods should be functional	30.0s
✓ TC16: Order confirmation and receipt generation	2.6s
✓ TC17: Checkout with an empty cart should be prevented	2.3s
✓ TC18: Navigation bar should be consistent across pages	2.6s
✓ TC19: Cart and logout should be easily accessible	3.6s
✓ TC20: Labels and instructions should be clear and understandable	879ms
✓ SauceDemo Frontend Tests - Usability and Interface	
✓ TC21: Mobile responsiveness and readability	926ms
✓ TC22: Error messages should be clear and informative	1.1s
✓ TC23: Error messages for incorrect user input	1.5s
✓ TC24: Consistency of layout and design	1.7s
✓ SauceDemo Frontend Tests - Interface and Links	
✓ TC25: Responsive design on different devices	2.8s

Εικόνα 16: Εκτέλεση Σεναρίων ChatGPT4 1/2

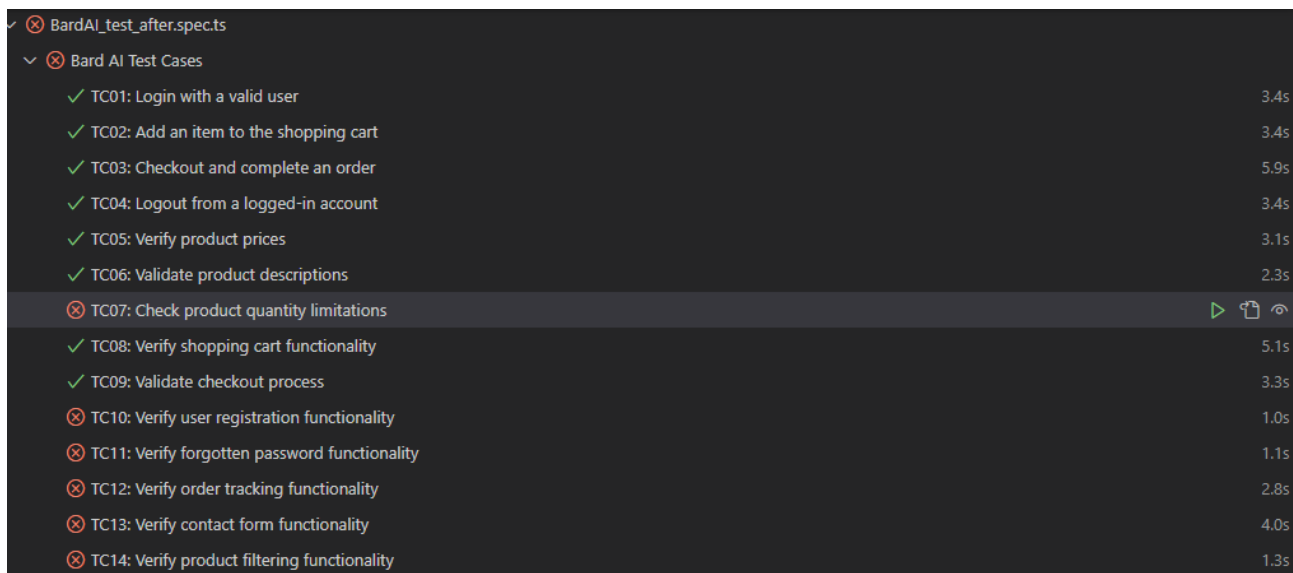
✓ TC26: All images and icons should load	3.3s
✓ TC27: Images should have appropriate alt text	1.4s
✓ TC28: All internal and external links should be valid	1.2s
✓ TC29: No broken links or incorrect redirections	1.4s
✓ SauceDemo Frontend Tests - Compatibility and Performance	
✓ TC30: Website should work on chromium	2.1s
✓ TC30: Website should work on firefox	6.8s
✓ TC30: Website should work on webkit	3.7s
✓ TC31: Website should be responsive on iPhone 11	1.6s
✓ TC31: Website should be responsive on Pixel 5	1.6s
✓ TC31: Website should be responsive on iPad Pro 11	1.6s
✓ TC32: Website load time should be within acceptable limits	5.2s
✓ SauceDemo Frontend Tests - Speed, Responsiveness, and Security	
✓ TC33: Application should be responsive to user actions	2.6s
✓ TC34: Login system should be secure	2.0s
✓ TC35: Application should be secure against SQL injection	2.1s
✓ SauceDemo Frontend Tests - Session Management and Accessibility	

Εικόνα 17: Εκτέλεση Σεναρίων ChatGPT4 2/2

✓ TC01: Login with valid credentials	6.5s
✓ TC02: Add a product to the cart	6.5s
✓ TC03: Remove a product from the cart	7.4s
✓ TC04: Sort products	3.4s
✓ TC05: Checkout	5.7s
✓ TC06: Logout	3.5s
✓ TC07: Filter products by category	3.5s
✓ TC08: Reset app state	3.5s
✓ TC09: View about page	4.7s
✓ TC10: View product details	3.7s
✗ TC11: Change product quantity	
✓ TC12: Cancel checkout	3.3s

Εικόνα 18: Εκτέλεση Σεναρίων Copilot

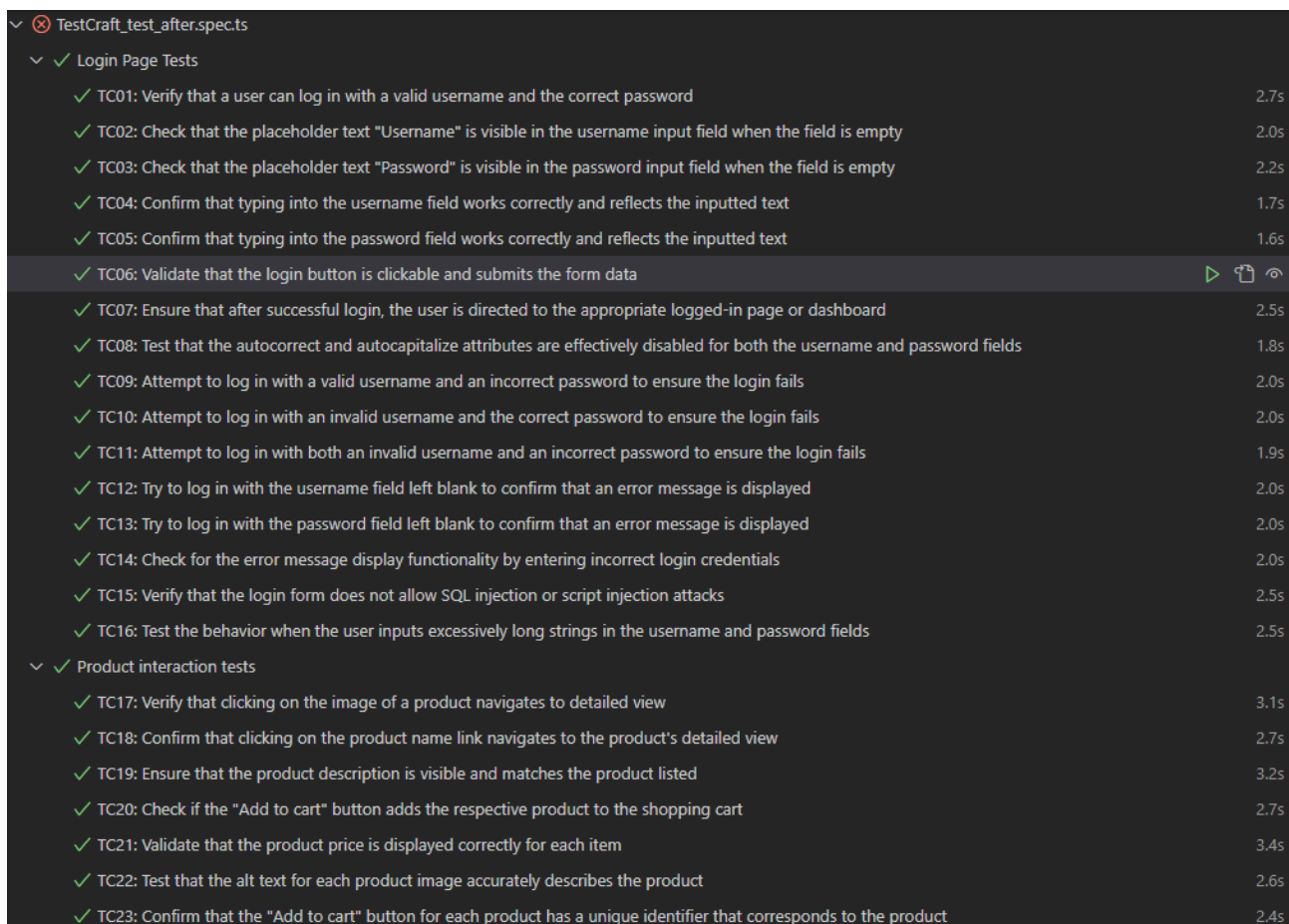
Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)



A screenshot of a test execution interface for Bard AI. The interface shows a list of test cases under the heading 'Bard AI Test Cases'. Each test case is preceded by a green checkmark, indicating successful execution. The test cases are listed with their descriptions and execution times in seconds. The test case 'TC07: Check product quantity limitations' is highlighted in a darker grey background, and it is preceded by a red 'X' icon, indicating a failed execution. To the right of this failed test case are three small icons: a play button, a refresh button, and a close button.

Test Case	Execution Time
TC01: Login with a valid user	3.4s
TC02: Add an item to the shopping cart	3.4s
TC03: Checkout and complete an order	5.9s
TC04: Logout from a logged-in account	3.4s
TC05: Verify product prices	3.1s
TC06: Validate product descriptions	2.3s
TC07: Check product quantity limitations	
TC08: Verify shopping cart functionality	5.1s
TC09: Validate checkout process	3.3s
TC10: Verify user registration functionality	1.0s
TC11: Verify forgotten password functionality	1.1s
TC12: Verify order tracking functionality	2.8s
TC13: Verify contact form functionality	4.0s
TC14: Verify product filtering functionality	1.3s

Εικόνα 19: Εκτέλεση Σεναρίων BardAI



A screenshot of a test execution interface for TestCraft app 1/3. The interface shows a list of test cases under the heading 'Login Page Tests' and 'Product interaction tests'. All test cases are preceded by a green checkmark, indicating successful execution. The test cases are listed with their descriptions and execution times in seconds. The test case 'TC06: Validate that the login button is clickable and submits the form data' is highlighted in a darker grey background, and it is preceded by a green checkmark. To the right of this test case are three small icons: a play button, a refresh button, and a close button.

Test Case	Execution Time
TC01: Verify that a user can log in with a valid username and the correct password	2.7s
TC02: Check that the placeholder text "Username" is visible in the username input field when the field is empty	2.0s
TC03: Check that the placeholder text "Password" is visible in the password input field when the field is empty	2.2s
TC04: Confirm that typing into the username field works correctly and reflects the inputted text	1.7s
TC05: Confirm that typing into the password field works correctly and reflects the inputted text	1.6s
TC06: Validate that the login button is clickable and submits the form data	
TC07: Ensure that after successful login, the user is directed to the appropriate logged-in page or dashboard	2.5s
TC08: Test that the autocorrect and autocapitalize attributes are effectively disabled for both the username and password fields	1.8s
TC09: Attempt to log in with a valid username and an incorrect password to ensure the login fails	2.0s
TC10: Attempt to log in with an invalid username and the correct password to ensure the login fails	2.0s
TC11: Attempt to log in with both an invalid username and an incorrect password to ensure the login fails	1.9s
TC12: Try to log in with the username field left blank to confirm that an error message is displayed	2.0s
TC13: Try to log in with the password field left blank to confirm that an error message is displayed	2.0s
TC14: Check for the error message display functionality by entering incorrect login credentials	2.0s
TC15: Verify that the login form does not allow SQL injection or script injection attacks	2.5s
TC16: Test the behavior when the user inputs excessively long strings in the username and password fields	2.5s
TC17: Verify that clicking on the image of a product navigates to detailed view	3.1s
TC18: Confirm that clicking on the product name link navigates to the product's detailed view	2.7s
TC19: Ensure that the product description is visible and matches the product listed	3.2s
TC20: Check if the "Add to cart" button adds the respective product to the shopping cart	2.7s
TC21: Validate that the product price is displayed correctly for each item	3.4s
TC22: Test that the alt text for each product image accurately describes the product	2.6s
TC23: Confirm that the "Add to cart" button for each product has a unique identifier that corresponds to the product	2.4s

Εικόνα 20: Εκτέλεση Σεναρίων TestCraft.app 1/3

Έλεγχος λογισμικού με χρήση εφαρμογών Τεχνητής Νοημοσύνης (AI)

✓	TC24: Verify that the menu is visible and accessible on the page.	3.1s
✓	TC25: Check that the "All Items" link is clickable and redirects the user to the appropriate section or page.	4.0s
✓	TC26: Ensure that the "About" link navigates to the "https://saucelabs.com/" URL when clicked.	4.5s
✓	TC27: Confirm that the "Logout" link triggers the logout functionality and signs the user out.	2.8s
✓	TC28: Validate that the "Reset App State" link resets the application to its default state without errors.	5.2s
⊗	TC29: Try to navigate to the "About" link with a slow internet connection to check if there's a loading indicator or timeout message.	
⊗	TC30: Click the "Logout" link after the session has already expired to see if it handles the scenario gracefully.	8.2s
⊗	TC31: Test the menu links with an incorrect or modified href value to see how the application handles broken links.	8.7s
✓	TC32: Verify that the "Continue Shopping" button takes the user back to the shopping page when clicked	4.6s
✓	TC33: Confirm that the "Checkout" button takes the user to the checkout page when clicked	3.4s
✓	TC34: Check that the cart displays the correct quantity (QTY) and description of items added by the user	3.2s
✓	TC35: Ensure that the cart updates correctly when items are added or removed	2.2s
✓	TC36: Validate that the cart's quantity and description labels are visible and readable	2.6s
✓	TC37: Confirm that the images for the "Continue Shopping" button load correctly and represent a visual cue for going back	2.5s
✓	TC38: Test that the alt text for the "Go back" image is displayed if the image fails to load	2.6s
✓	TC39: Verify that clicking the "Continue Shopping" button with an empty cart does not cause any errors	2.6s
⊗	TC40: Check that the "Checkout" button is disabled or provides an appropriate message when the cart is empty	6.9s
⊗	TC41: Attempt to navigate to the checkout page without any items in the cart and ensure an appropriate message is displayed or navigation is prevented	6.8s
✓	TC42: Verify form accepts valid input and allows proceeding	4.9s
✓	TC43: Ensure autocorrect and autocapitalize are	6.0s
✓	TC44: Check placeholder texts are visible	4.4s
✓	TC45: Confirm "Cancel" button redirects or cancels	4.4s
✓	TC46: Validate form retains input after error	5.4s
✓	TC47: Attempt to submit form with empty fields	3.6s
⊗	TC48: Input invalid characters name fields	7.5s
⊗	TC49: Enter invalid zip/postal code format	7.9s
⊗	TC50: Submit form with excessively long values	7.8s
⊗	TC51: Test form for code injection vulnerabilities	8.8s

Εικόνα 21: Εκτέλεση Σεναρίων TestCraft.app 2/3

✓	TC52: Verify quantity and description labels	3.4s
✓	TC53: Confirm payment information displays correctly	3.0s
✓	TC54: Ensure shipping information is visible	3.1s
✓	TC55: Check price total section displays \$0.00 initially	4.1s
✓	TC56: Test "Cancel" button functionality	4.1s
✓	TC57: Verify "Finish" button functionality	3.6s
✓	TC58: Ensure summary information labels are visible	4.3s
⊗	TC59: Attempt to proceed with checkout with an empty cart	7.5s
⊗	TC60: Interact with checkout during network interruption	
✓	TC61: Verify that the "Thank you for your order!" message is displayed after a successful checkout process	2.8s
✓	TC62: Check if the image of the Pony Express is displayed correctly and matches the expected design	2.5s
✓	TC63: Ensure the text "Your order has been dispatched, and will arrive just as fast as the pony can get there!" is visible and correctly formatted	3.9s
✓	TC64: Confirm that the "Back Home" button is visible and clickable	4.0s
✓	TC65: Validate that clicking the "Back Home" button redirects the user to the homepage or product list	2.7s

Εικόνα 22: Εκτέλεση Σεναρίων TestCraft.app 3/3

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

- [1] K. Latha, "An Evaluation and Comparative Analysis of Software Testing Techniques and Tools," *Int. J. Innov. Res. Sci. Eng. Technol.*, vol. 9, no. 2, pp. 185–190, 2020.
- [2] I. Shuaibu, M. Musa, and M. Ibrahim, "Investigation onto the Software Testing Techniques and Tools: An Evaluation and Comparative Analysis," *Int. J. Comput. Appl.*, vol. 177, no. 23, pp. 24–30, 2019, doi: 10.5120/ijca2019919685.
- [3] M. M. Syaikhuddin, C. Anam, A. R. Rinaldi, and M. E. B. Conoras, "Conventional Software Testing Using White Box Method," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 3, no. 1, pp. 65–72, 2018, doi: 10.22219/kinetik.v3i1.231.
- [4] M. Albarka Umar, "Comprehensive study of software testing: Categories, levels, techniques, and types Software Testing View project System Analysis View project Comprehensive Study of Software Testing: Categories, Levels, Techniques, and Types," *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 5, no. 6, pp. 32–40, 2020, [Online]. Available: <https://www.researchgate.net/publication/342538504>
- [5] Verma, Akanksha & Khatana, Amita & Chaudhary, Sarika. (2017). A Comparative Study of Black Box Testing and White Box Testing. *International Journal of Computer Sciences and Engineering*. 5. 301-304. 10.26438/ijcse/v5i12.301304.
- [6] Jamil, Abid & Arif, Muhammad & Abubakar, Normi & Ahmad, Akhlaq. (2016). Software Testing Techniques: A Literature Review. 177-182. 10.1109/ICT4M.2016.045.
- [7] Izzat, Sukaina & Saleem, Nada. (2023). Software Testing Techniques and Tools: A Review. *Journal of Education and Science*. 32. 31-40. 10.33899/edusj.2023.137480.1305.
- [8] J. McCarthy, "Programs with common sense," in *Proceedings of the Symposium on Mechanisation of Thought Processes*, vol. 1. London: Her Majesty's Stationery Office, 1958, pp. 77–84.
- [9] Russell, S. J., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach*. Pearson.
- [10] P. Salazar, "History of AI: Timeline and the Future," *Maryville University Online*, May 19, 2023. <https://online.maryville.edu/blog/history-of-ai/> (accessed Oct. 14, 2023).
- [11] N. Laskowski and L. Tucci, "artificial intelligence (AI)," *Enterprise AI*, 2023. <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence> (accessed Dec. 05, 2023).
- [12] B. Nancholas, "What are the different types of artificial intelligence? - University of Wolverhampton," *University of Wolverhampton*, Jun. 07, 2023. <https://online.wlv.ac.uk/what-are-the-different-types-of-artificial-intelligence/> (accessed Oct. 09, 2023).
- [13] "Artificial Intelligence (AI): What It Is and How It Is Used," *Investopedia*, 2024. <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp> (accessed Jan. 15, 2024).
- [14] P. Louridas and C. Ebert, "Machine learning," *IEEE Softw.*, vol.33, no. 5, pp. 110–115, Sep./Oct. 2016.
- [15] Panigrahi, A., Chen, Y., Kuo, C.C.J.: Analysis on gradient propagation in batch normalized residual networks (2018). arXiv:1812.00342
- [16] Seber, G.A.F., Lee, A.J.: *Linear Regression Analysis*, vol. 329. Wiley (2012)
- [17] Jain, A.K.: Data clustering: 50 years beyond k-means. *Pattern Recognit. Lett.* 31(8), 651–666 (2010). <https://doi.org/10.1016/j.patrec.2009.09.011>
- [18] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: *International Conference on Machine Learning*, pp. 1928–1937 (2016)

- [19] Deng, L., Dong, Y., et al.: Deep learning: methods and applications. *Found. Trends® Signal Process.* 7(3–4), 197–387 (2014). <https://doi.org/10.1007/978-981-13-3459-7>
- [20] Raina, V., Krishnamurthy, S. (2022). Machine Learning. In: Building an Effective Data Science Practice. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-7419-4_16
- [21] Mathew, A., Amudha, P., Sivakumari, S. (2021). Deep Learning Techniques: An Overview. In: Hassanien, A., Bhatnagar, R., Darwish, A. (eds) *Advanced Machine Learning Technologies and Applications. AMLTA 2020. Advances in Intelligent Systems and Computing*, vol 1141. Springer, Singapore. https://doi.org/10.1007/978-981-15-3383-9_54
- [22] Juergen Schmidhuber (2015) Deep Learning. *Scholarpedia*, 10(11):32832.
- [23] D. Team, "Natural Language Processing (NLP): Definition and Principles," *Data Science Courses | DataScientest*, Jun. 09, 2023. <https://datascientest.com/en/natural-language-processing-definition-and-principles> (accessed Oct. 29, 2023).
- [24] "What is artificial intelligence (AI)? Its applications, architecture and future," *SDxCentral*, Sep. 21, 2023. <https://www.sdxcntral.com/artificial-intelligence/definitions/what-is-artificial-intelligence/> (accessed Nov. 12, 2023).
- [25] Robin R Murphy, "Introduction to AI Robotics Second Edition," The MIT Press, Cambridge, Massachusetts, London England, 2019.
- [26] Zwass, Vladimir. "expert system". *Encyclopedia Britannica*, 10 Feb. 2016, <https://www.britannica.com/technology/expert-system>. Accessed 31 October 2023.
- [27] Khaliq, Zubair & Farooq, Sheikh Umar & Khan, Dawood. (2022). Artificial Intelligence in Software Testing : Impact, Problems, Challenges and Prospect.
- [28] Yamashita R, Nishio M, Do RK, Togashi K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*. 2018 Aug;9:611-29.
- [29] Priyanka Mahajan, Chhaya Nawale, Siddheshwar Kini, Prof. Krishnanjali Shinde (2017). Weatherforecasting using neural networks.
- [30] M. Garanhel, "What are the top 7 branches of artificial intelligence?," *AI Accelerator Institute*, Mar. 22, 2023. <https://www.aiacceleratorinstitute.com/what-are-the-top-7-branches-of-artificial-intelligence/> (accessed Oct. 21, 2023).
- [31] K. Sugali, "Software Testing: Issues and Challenges of Artificial Intelligence & Machine Learning," *Ssrn.com*, Oct. 24, 2021. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3948930 (accessed Nov. 22, 2023).
- [32] Khaliq, Zubair & Farooq, Sheikh Umar & Khan, Dawood. (2022). Artificial Intelligence in Software Testing : Impact, Problems, Challenges and Prospect.
- [33] "What are the Advantages of Artificial Intelligence in Software Testing?," *Maruti Techlabs*, 2019. <https://marutitech.com/artificial-intelligence-in-testing/> (accessed Dec. 11, 2023).
- [34] A. Khrupa, "Testfort," *TestFort Testing & QA Company*, Apr. 14, 2023. <https://testfort.com/blog/ai-in-software-testing-a-silver-bullet-or-a-threat-to-the-profession> (accessed Nov. 15, 2023).
- [35] V. Akila, A. Vasuki, J. A. Christaline, R. Sathiya, P. Rishi and A. S. Edward, "Enhancing Software Testing with Machine Learning Techniques," *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*, Erode, India, 2023, pp. 329-333, doi: 10.1109/ICSCDS56580.2023.10105028. keywords: {Software testing; Measurement; Machine learning algorithms; Statistical analysis; Software algorithms; Manuals; Software systems; Software testing; Machine Learning; Regression; decision tree; Random forest},
- [36] A. Salman, "Test Case Generation from Specifications Using Natural Language Processing," *DIVA*, 2020. <https://kth.diva-portal.org/smash/record.jsf?pid=diva2%3A1468729&dswid=1927> (accessed Nov. 12, 2023).

- [37] “Visual Regression Testing: Everything You Need to Know,” Testsigma Blog, Jan. 26, 2023. <https://testsigma.com/blog/visual-regression-tests/> (accessed Nov. 21, 2023).
- [38] Madhav Kashyap. October 2023. AI in Anomaly Detection: Uncovering Hidden Threats in Data in Real Time. <https://www.linkedin.com/pulse/ai-anomaly-detection-uncovering-hidden-threats-data-real-kashyap/>(accessed Nov. 21, 2023).
- [39] Self Healing Test Automation,” ResearchGate, 2020. https://www.researchgate.net/publication/343005103_Self_Healing_Test_Automation (accessed Nov. 21, 2023).
- [40] R. Weingartz, “Risk-based testing: benefits, real examples, and 4 mistakes to avoid,” aqua cloud - best software for testing, Aug. 29, 2023. <https://aqua-cloud.io/risk-based-testing/> (accessed Nov. 22, 2023).
- [41] R. Gupta, “Test Failure Analysis With AI-Critical To DevOps – Webomates,” Webomates, Aug. 10, 2021. <https://www.webomates.com/blog/artificial-intelligence/test-failure-analysis-with-ai-critical-to-devops/> (accessed Nov. 25, 2023).
- [42] <https://www.linkedin.com/in/bharath-hemachandran-b042ba2>, “Generative AI for Efficient Test Data Generation and Management | LambdaTest,” LambdaTest, Jul. 13, 2023. <https://www.lambdatest.com/blog/generative-ai-for-efficient-test-data-generation/> (accessed Nov. 25, 2023).
- [43] sdettech, “The Role of AI in Test Case Generation: Enhancing Test Coverage - SDET Tech,” SDET Tech, Dec. 11, 2023. <https://sdettech.com/the-role-of-ai-in-test-case-generation-enhancing-test-coverage/> (accessed Jan. 02, 2024).
- [44] B. Varshney, “Effortless Test Case Coverage: Unleash the Power of AI for Faster, More Complete Testing,” Medium, Jan. 05, 2024. <https://medium.com/@bharatvarshney567/effortless-test-case-coverage-unleash-the-power-of-ai-for-faster-more-complete-testing-software>. (accessed Jan. 02, 2024).
- [45] LeewayHertz, “An overview of Google’s Palm2 - Predict - Medium,” Medium, Aug. 06, 2023. <https://medium.com/predict/an-overview-of-googles-palm2-111bd75ca12a> (accessed Jan. 12, 2024).
- [46] “LinkedIn,” LinkedIn.com, 2024. <https://www.linkedin.com/company/testcraftapp/about/> (accessed Jan. 10, 2024).
- [47] Windows Experience Blog and Yusuf, “Introducing a new Copilot key to kick off the year of AI-powered Windows PCs,” Windows Experience Blog, Jan. 04, 2024. <https://blogs.windows.com/windowsexperience/2024/01/04/introducing-a-new-copilot-key-to-kick-off-the-year-of-ai-powered-windows-pcs/> (accessed Jan. 10, 2024).
- [48] Sundar Pichai, “An important next step on our AI journey,” Google, Feb. 06, 2023. <https://blog.google/technology/ai/bard-google-ai-search-updates/> (accessed Jan. 10, 2024).
- [49] M. Whaley, “Announcing Microsoft Playwright Testing: Scalable end-to-end testing for modern web apps | Microsoft Azure Blog,” Microsoft Azure Blog, Oct. 04, 2023. <https://azure.microsoft.com/en-us/blog/announcing-microsoft-playwright-testing-scalable-end-to-end-testing-for-modern-web-apps/> (accessed Jan. 5, 2024).
- [50] “Artificial Intelligence and Machine learning for Software Testers,” Testsigma Blog, Aug. 21, 2023. <https://testsigma.com/blog/ai-and-ml-in-software-testing/> (accessed Oct 25, 2023).