Ατομική Διπλωματική Εργασία

Automated Source Code Analysis for Privacy Policy Compliance in Web Platforms

Ανδρέας Πάρης Λάμπρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2025

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Automated Source Code Analysis for Privacy Policy Compliance in Web Platforms

Ανδρέας Πάρης Λάμπρου

Επιβλέπων Καθηγητής

Δρ. Γιώργος Παπαδόπουλος

Συνεπιβλέπουσα

Ευαγγελία Βανέζη

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων

απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου

Κύπρου

Μάιος 2025

Ευχαριστίες

Με την βοήθεια του Θεού έχω φέρει εις πέρας αυτή την διπλωματική εργασία. Σε όλη την διάρκεια συγγραφής της εργασίας αυτής, έχω λάβει μεγάλη υποστήριξη και καθοδήγηση. Αρχικά θα ήθελα να ευχαριστήσω τον Επιβλέποντα Καθηγητή μου, Δρ. Γιώργο Παπαδόπουλο, ο οποίος με έχει εμπιστευτεί, και μου έδωσε την ευκαιρία να δουλέψω στο θέμα αυτό. Θα ήθελα να εκφράσω επίσης τις θερμές μου ευχαριστίες στην Ερευνήτρια και Συνεπιβλέπουσα μου κα. Ευαγγελία Βανέζη, για την βοήθεια και την εξαιρετική συνεργασία που είχαμε κατά την διάρκεια της εκπόνησης της διπλωματικής. Οι συμβουλές τους και η καθοδήγησή τους με βοήθησαν να τελειώσω με επιτυχία τον στόχο του.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, αλλά και τους φίλους συμφοιτητές μου για την απέραντη ψυχολογική υποστήριξη και αγάπη τους.

Περίληψη

Στο πλαίσιο της παρούσας διπλωματικής εργασίας, σχεδιάστηκε και υλοποιήθηκε το εργαλείο CodeScanner, με στόχο τον εντοπισμό εντολών επεξεργασίας προσωπικών δεδομένων σε πηγαίο κώδικα διαδικτυακών εφαρμογών. Το εργαλείο δίνει ιδιαίτερη έμφαση στην ανάλυση JavaScript και XML, αξιοποιώντας τεχνικές στατικής ανάλυσης για την αναγνώριση ενεργειών συλλογής, αποθήκευσης, μετάδοσης και τροποποίησης δεδομένων που εμπίπτουν στις απαιτήσεις του Γενικού Κανονισμού για την Προστασία Δεδομένων (GDPR).

Η εργασία περιλαμβάνει:

- Βιβλιογραφική επισκόπηση του κανονιστικού πλαισίου προστασίας προσωπικών δεδομένων με έμφαση στον GDPR και τις βασικές αρχές του.
- Συστηματική καταγραφή και κατηγοριοποίηση των βασικών ενεργειών επεξεργασίας προσωπικών δεδομένων, όπως προβλέπονται στον GDPR.
- Ανάλυση υπαρχόντων εργαλείων ανίχνευσης προσωπικών δεδομένων σε πηγαίο κώδικα και επισήμανση των ελλείψεών τους.
- Σχεδίαση και υλοποίηση του εργαλείου CodeScanner, το οποίο διαβάζει JavaScript και XML αρχεία και αναγνωρίζει τις εντολές που σχετίζονται με προσωπικά δεδομένα.
- Δοκιμή του εργαλείου μέσω παραδειγμάτων, ειδικά δημιουργημένων σελίδων και πολιτικών χρήσης, ώστε να αξιολογηθεί η ακρίβεια, η ευχρηστία και η δυνατότητα επέκτασης του συστήματος.

Το εργαλείο παρέχει στους προγραμματιστές και στους υπεύθυνους συμμόρφωσης τη δυνατότητα να εντοπίζουν γρήγορα πιθανές παραβιάσεις ή αποκλίσεις από τις δηλωμένες πολιτικές απορρήτου, ενισχύοντας τη διαφάνεια και τη συμμόρφωση των εφαρμογών με τις απαιτήσεις της νομοθεσίας.

Περιεχόμενα

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική Ιδέα

Η παρούσα διπλωματική εργασία επικεντρώνεται στην ανάλυση της ιδιωτικότητας και των πολιτικών απορρήτου τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο. Βασικός στόχος είναι η διερεύνηση της επεξεργασίας προσωπικών δεδομένων από διαδικτυακές εφαρμογές και η αξιολόγηση της συμμόρφωσής τους με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR), μέσα από τον εντοπισμό σχετικών ενεργειών στον πηγαίο τους κώδικα.

Η έρευνα συνδυάζει βιβλιογραφική ανάλυση, εμπειρική μελέτη, και την ανάπτυξη ενός εργαλείου στατικής ανάλυσης πηγαίου κώδικα που επιτρέπει την ανίχνευση προσωπικών δεδομένων. Το εργαλείο επικεντρώνεται σε JavaScript και XML, ενώ η καταγραφή εντολών επεξεργασίας δεδομένων επεκτάθηκε και σε άλλες γλώσσες προγραμματισμού (Java, PHP, Python, C++), με σκοπό την υποστήριξη μελλοντικών επεκτάσεων.

Στο πλαίσιο αυτό:

- Αναλύονται υπάρχοντα εργαλεία εντοπισμού προσωπικών δεδομένων σε πηγαίο κώδικα.
- Εξετάζεται η αντιστοίχιση πολιτικών απορρήτου με τη λειτουργική υλοποίηση στον κώδικα.
- Τεκμηριώνεται η ανάγκη για ένα αυτοματοποιημένο εργαλείο αξιολόγησης κώδικα βάσει GDPR.
- Καταγράφονται οι βασικές εντολές επεξεργασίας προσωπικών δεδομένων μέσω μελέτης σε πέντε γλώσσες προγραμματισμού.

Η υλοποίηση ακολουθεί επαναληπτική μεθοδολογία με χρήση της προσέγγισης throw-away prototyping, επιτρέποντας γρήγορη ανάπτυξη εκδόσεων και βελτιώσεων μέσα από δοκιμές και αξιολόγηση από πραγματικούς χρήστες.

1.2 Κίνητρα

Η ανάγκη για προστασία της ιδιωτικότητας και της ορθής διαχείρισης των προσωπικών δεδομένων αποτελεί μία από τις σημαντικότερες προκλήσεις της ψηφιακής εποχής. Παρά τις ρυθμιστικές διατάξεις του GDPR, η πρακτική εφαρμογή των πολιτικών απορρήτου παραμένει συχνά ασαφής – τόσο για τους τελικούς χρήστες όσο και για τους ίδιους τους προγραμματιστές.

Πολλές εφαρμογές υλοποιούνται χωρίς μηχανισμούς που να διασφαλίζουν τη ρητή συμμόρφωση του πηγαίου τους κώδικα με τις δηλωμένες πολιτικές απορρήτου. Το κενό αυτό καθιστά αναγκαία την ύπαρξη εργαλείων που να υποστηρίζουν τους υπεύθυνους ανάπτυξης και συμμόρφωσης στον έλεγχο του πραγματικού τρόπου επεξεργασίας των δεδομένων.

Η εργασία αυτή επιχειρεί να καλύψει το κενό αυτό, προσφέροντας ένα λειτουργικό εργαλείο ελέγχου και εντοπισμού προσωπικών δεδομένων μέσω στατικής ανάλυσης κώδικα.

1.3 Γενικά για το Εργαλείο

Το εργαλείο που αναπτύχθηκε, με την ονομασία CodeScanner, αποτελεί ένα σύστημα γραμμής εντολών (CLI) υλοποιημένο σε Java, το οποίο παρέχει δυνατότητες αυτόματης σάρωσης αρχείων JavaScript και XML. Εντοπίζει εντολές που αφορούν την επεξεργασία προσωπικών δεδομένων όπως συλλογή, αποθήκευση, τροποποίηση, μετάδοση και διαγραφή.

Το εργαλείο βασίζεται σε προσαρμοσμένα μοτίβα αναγνώρισης εντολών και σε μια εκτενή λίστα λέξεων-κλειδιών που προέκυψε από μελέτη του GDPR και άλλων ερευνητικών πηγών. Έχει τη δυνατότητα χαρτογράφησης των εντοπισμένων ενεργειών ανά συνάρτηση, σύγκρισης με πολιτικές που περιγράφονται σε XML, και παραγωγής αναφορών με λεπτομέρειες για τη συμμόρφωση.

Η modular δομή του CodeScanner επιτρέπει μελλοντική υποστήριξη και άλλων γλωσσών, καθώς η καταγραφή εντολών ανά γλώσσα έχει ήδη πραγματοποιηθεί.

1.4 Δομή του Κειμένου

Η διπλωματική εργασία οργανώνεται σε επτά κεφάλαια:

- **Κεφάλαιο 1:** Παρουσίαση της γενικής ιδέας, των κινήτρων, της περιγραφής του εργαλείου και της διάρθρωσης του εγγράφου.
- **Κεφάλαιο 2:** Βιβλιογραφική επισκόπηση σχετικών εργαλείων και μελετών.
- **Κεφάλαιο 3:** Περιγραφή τεχνολογιών, εργαλείων και υποδομών που χρησιμοποιήθηκαν.

- **Κεφάλαιο 4:** Αναλυτική παρουσίαση της διαδικασίας ανάπτυξης του εργαλείου.
- **Κεφάλαιο 5:** Παρουσίαση των λειτουργιών του εργαλείου με πρακτικά παραδείγματα.
- **Κεφάλαιο 6:** Μεθοδολογία αξιολόγησης του εργαλείου
- **Κεφάλαιο 7:** Συμπεράσματα και προτάσεις για μελλοντική εργασία.

Κεφάλαιο 2

Έρευνα και Υπάρχοντα Εργαλεία

2.1 Εισαγωγή

Στην ενότητα αυτή παρουσιάζονται επιστημονικά άρθρα και μελέτες που σχετίζονται με την ιδιωτικότητα και τις πολιτικές απορρήτου, καθώς και η σημασία τους στο λογισμικό και την ανάπτυξη διαδικτυακών εφαρμογών. Ιδιαίτερη έμφαση δίνεται στον τρόπο με τον οποίο ο έλεγχος που αφορά την επεξεργασία προσωπικών δεδομένων γίνεται μέσω εξέτασης του πηγαίου κώδικα, καθώς και στη σύνδεση αυτής της διαδικασίας με νομικά πλαίσια όπως το GDPR.

Η έρευνα επικεντρώνεται στη σχέση μεταξύ του κανονισμού για την προστασία της ιδιωτικότητας - General Data Protection Regulation (GDPR) και πολιτικών ιδιωτικότητας δεδομένων, καθώς ο GDPR επιβάλλει τις βασικές αρχές και περιορισμούς στη συλλογή, επεξεργασία και αποθήκευση προσωπικών δεδομένων. Ο GDPR εφαρμόστηκε το 2018 με στόχο να βοηθήσει ρητά στην προστασία της ιδιωτικότητας των προσωπικών δεδομένων των πολιτών της Ευρωπαϊκής ένωσης. Σύμφωνα με το άρθρο 5 του κανονισμού , η επεξεργασία δεδομένων πρέπει να γίνεται με διαφάνεια, ενώ οι χρήστες πρέπει να είναι ενήμεροι για το πώς χρησιμοποιούνται τα προσωπικά τους δεδομένα (GDPR Art. 5/1(a) - ‘lawfulness, fairness and transparency’), καθώς και να δίνουν την ρητή τους συγκατάθεση στην πολιτική που δηλώνει την ακριβής χρήση τους. Τα συστήματα είναι υποχρεωμένα να ακολουθούν πιστά αυτή την πολιτική (GDPR Art. 5/1(b) - ‘purpose limitation’).

Η διαχείριση προσωπικών δεδομένων σε διαδικτυακές πλατφόρμες γίνεται τόσο στον client όσο και στον server, με σημαντικό ρόλο να διαδραματίζουν γλώσσες όπως η JavaScript για την πλευρά του χρήστη και η PHP ή άλλες server-side τεχνολογίες για την επεξεργασία και αποθήκευση στο backend. Η JavaScript, ειδικότερα, χρησιμοποιείται ευρέως για τη συλλογή πληροφοριών από φόρμες, την αποθήκευση σε browser storage, και τη μετάδοση δεδομένων μέσω APIs, παρουσιάζοντας προκλήσεις στον εντοπισμό πιθανών παραβιάσεων ιδιωτικότητας λόγω της δυναμικής φύσης της.

Σύμφωνα με τους Lim και Kim [1], η χρήση τεχνικών όπως οι Multiversion Dependency Graphs (MDGs) μπορεί να ενισχύσει σημαντικά την ανάλυση κώδικα JavaScript, επιτρέποντας την ανίχνευση ευπαθειών και παραβάσεων που σχετίζονται με προσωπικά δεδομένα. Η στατική ανάλυση κώδικα με τέτοιες τεχνικές καθιστά πιο εύκολο για προγραμματιστές και ελεγκτές να εντοπίσουν προβληματικά σημεία που σχετίζονται με την ιδιωτικότητα.

Η συμμόρφωση μιας διαδικτυακής εφαρμογής με το GDPR και τις πολιτικές απορρήτου δεν είναι απλή διαδικασία. Οι προγραμματιστές συχνά χρησιμοποιούν παραδοσιακές μεθόδους ελέγχου, οι οποίες όμως δεν μπορούν να εγγυηθούν την επίσημη επαλήθευση της συμμόρφωσης.

2.2 Έρευνα

2.2.1 Automated Detection of Personal Data Processing in Web Applications

Η αυτόματη ανίχνευση επεξεργασίας προσωπικών δεδομένων σε διαδικτυακές εφαρμογές αποτελεί ένα πεδίο ιδιαίτερου ενδιαφέροντος για την έρευνα στο χώρο της ασφάλειας και της ιδιωτικότητας.

Σύμφωνα με τη μελέτη “Toward an Android Static Analysis Approach for Data Protection” [2], η χρήση τεχνικών στατικής ανάλυσης επιτρέπει την ανίχνευση ενεργειών που σχετίζονται με την επεξεργασία προσωπικών δεδομένων στον πηγαίο κώδικα εφαρμογών, με στόχο τη συμμόρφωση με τον GDPR.

Η μεθοδολογία που περιγράφεται στο άρθρο βασίζεται στην τεχνική taint analysis, η οποία επιτρέπει τον εντοπισμό της ροής προσωπικών δεδομένων από το σημείο συλλογής τους μέχρι την αποθήκευση ή αποστολή τους. Επιπλέον, γίνεται κατηγοριοποίηση των δεδομένων σε άμεσα (π.χ. αριθμός κοινωνικής ασφάλισης, email) και έμμεσα (π.χ. γεωγραφική θέση, IP), κάτι που βοηθά στην καλύτερη κατανόηση του επιπέδου ευαισθησίας των δεδομένων που επεξεργάζονται οι εφαρμογές.

Ιδιαίτερη σημασία έχει η ικανότητα του συστήματος να οπτικοποιεί τη ροή δεδομένων, καθιστώντας ευκολότερη την κατανόηση της επεξεργασίας από προγραμματιστές και νομικούς. Παρόλο που το άρθρο επικεντρώνεται σε Android εφαρμογές, οι τεχνικές και η προσέγγιση που προτείνει μπορούν να προσαρμοστούν σε web πλατφόρμες – ειδικά σε περιβάλλοντα JavaScript, όπου η ροή δεδομένων είναι δυναμική και συχνά δύσκολη στον εντοπισμό.

Η συγκεκριμένη εργασία ενισχύει τη σημασία της στατικής ανάλυσης πηγαίου κώδικα για την προστασία της ιδιωτικότητας και επιβεβαιώνει τη χρησιμότητα τέτοιων εργαλείων, όπως το CodeScanner, για τον εντοπισμό παραβιάσεων σε πραγματικά συστήματα.

2.2.2 Privacy Policy Compliance Analysis via Static Code Inspection

Η στατική ανάλυση του πηγαίου κώδικα έχει αναδειχθεί ως βασική τεχνική για την επαλήθευση της συμμόρφωσης εφαρμογών με τις δηλωμένες πολιτικές απορρήτου. Η εργασία “GDPR Compliance in the Context of Continuous Integration” [3] εξετάζει τη δυνατότητα ενσωμάτωσης αυτοματοποιημένων ελέγχων συμμόρφωσης με τον GDPR σε κύκλους ανάπτυξης λογισμικού τύπου Continuous Integration (CI), υποδεικνύοντας τη σημασία του συνεχούς ελέγχου και της έγκαιρης ανίχνευσης πιθανών παραβιάσεων.

Η μελέτη εστιάζει στη δημιουργία εργαλείων που εντοπίζουν αποκλίσεις από τις πολιτικές απορρήτου με βάση το περιεχόμενο του πηγαίου κώδικα. Ιδιαίτερη έμφαση δίνεται στην ενσωμάτωση κανόνων GDPR σε pipelines CI/CD, ενισχύοντας τον αυτόματο έλεγχο συμμόρφωσης σε πραγματικό χρόνο.

Η προσέγγιση αυτή είναι ιδιαίτερα σχετική με την παρούσα διπλωματική εργασία, καθώς το εργαλείο CodeScanner επιχειρεί να εντοπίσει ασυνέπειες μεταξύ της πολιτικής απορρήτου – η οποία είναι καταγεγραμμένη σε XML μορφή – και του JavaScript πηγαίου κώδικα της εφαρμογής. Πιο συγκεκριμένα, το εργαλείο ελέγχει αν οι ενέργειες συλλογής, αποθήκευσης και επεξεργασίας δεδομένων στον κώδικα αντιστοιχούν με όσα δηλώνονται στην πολιτική απορρήτου του συστήματος.

Η εργασία αυτή υπογραμμίζει τη σημασία του “privacy by design” σε όλα τα στάδια ανάπτυξης και υποστηρίζει την ανάγκη για αυτοματοποιημένη συμμόρφωση μέσω ανάλυσης κώδικα, κάτι που αποτελεί και τον πυρήνα της λειτουργικότητας του εργαλείου που παρουσιάζεται σε αυτή τη διπλωματική εργασία.

2.2.3 Integrating GDPR Requirements into Web Development Workflows

Η ενσωμάτωση των απαιτήσεων του Γενικού Κανονισμού για την Προστασία Δεδομένων (GDPR) στις ροές εργασίας ανάπτυξης διαδικτυακών εφαρμογών αποτελεί ουσιαστικό βήμα για την ενίσχυση της ασφάλειας και της διαφάνειας στη χρήση προσωπικών δεδομένων. Η προσέγγιση αυτή, γνωστή και ως "Privacy by Design", στοχεύει στη συμμόρφωση των συστημάτων από τα πρώτα στάδια της ανάπτυξης τους.

Σύμφωνα με το άρθρο "Abstract Interpretation-Based Data Leakage Static Analysis" [4], οι συγγραφείς προτείνουν μια τεχνική στατικής ανάλυσης βασισμένη στην Abstract Interpretation με στόχο την ανίχνευση διαρροών δεδομένων σε περιβάλλοντα προγραμματισμού. Αν και η εργασία εστιάζει σε εφαρμογές μηχανικής μάθησης (data science notebooks), τα ευρήματά της είναι ιδιαίτερα χρήσιμα και για web περιβάλλοντα, καθώς επισημαίνεται η σημασία της πρόληψης ανεξέλεγκτης ροής δεδομένων ήδη κατά το στάδιο του προγραμματισμού.

Η μέθοδος που παρουσιάζεται αξιοποιεί τη στατική ανάλυση για να εντοπίσει πρακτικές όπου προσωπικά δεδομένα μπορεί να εκτεθούν, π.χ. μέσω ανεπαρκούς ανωνυμοποίησης ή απουσίας περιορισμών στη διάδοση. Οι τεχνικές αυτές μπορούν να εφαρμοστούν και σε JavaScript, προσφέροντας ένα επιπλέον επίπεδο ελέγχου ως προς τη συμμόρφωση με πολιτικές απορρήτου.

Το εργαλείο CodeScanner ακολουθεί την ίδια φιλοσοφία, καθώς στοχεύει στην ενσωμάτωση της ανάλυσης συμμόρφωσης στο στάδιο της ανάπτυξης. Με τον αυτόματο έλεγχο της συνέπειας μεταξύ των δηλώσεων της πολιτικής απορρήτου (σε XML) και της πραγματικής λειτουργίας του κώδικα (σε JavaScript), ενισχύεται η δυνατότητα για ασφαλή και διαφανή διαχείριση δεδομένων ήδη από τη φάση της υλοποίησης.

2.2.4 Data Protection by Design: Automated Code Review for Privacy Violations

Η προσέγγιση της προστασίας δεδομένων μέσω σχεδιασμού (Privacy by Design) τονίζει την ανάγκη ενσωμάτωσης μηχανισμών προστασίας της ιδιωτικότητας από τα αρχικά στάδια ανάπτυξης λογισμικού. Η αυτοματοποιημένη ανασκόπηση κώδικα αποτελεί κρίσιμο εργαλείο για την επίτευξη αυτού του στόχου, επιτρέποντας τον έγκαιρο εντοπισμό πιθανών παραβιάσεων της ιδιωτικότητας.

Σύμφωνα με την εργασία των Hjerppe, Ruohonen και Leppänen [5], προτείνεται μια μεθοδολογία στατικής ανάλυσης βασισμένη σε σχολιασμούς (annotations) στον πηγαίο κώδικα. Μέσω αυτής της προσέγγισης, οι προγραμματιστές μπορούν να επισημαίνουν τμήματα του κώδικα που επεξεργάζονται προσωπικά δεδομένα,

διευκολύνοντας την ανίχνευση πιθανών παραβιάσεων και ενισχύοντας τη συμμόρφωση με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR).

Το εργαλείο CodeScanner αξιοποιεί παρόμοιες τεχνικές, εστιάζοντας στην ανάλυση του πηγαίου κώδικα JavaScript και τη σύγκρισή του με τις δηλωμένες πολιτικές απορρήτου σε μορφή XML. Μέσω αυτής της διαδικασίας, εντοπίζονται αποκλίσεις μεταξύ των δηλωμένων πολιτικών και της πραγματικής επεξεργασίας δεδομένων στον κώδικα, επιτρέποντας την έγκαιρη διόρθωση και τη διασφάλιση της συμμόρφωσης με τις απαιτήσεις του GDPR.

2.2.5 Static and Dynamic Code Analysis for GDPR Compliance in Web Platforms

Η συμμόρφωση με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR) αποτελεί κρίσιμη πρόκληση για τις διαδικτυακές πλατφόρμες, απαιτώντας την εφαρμογή τεχνικών που διασφαλίζουν την προστασία των προσωπικών δεδομένων. Δύο βασικές τεχνικές που χρησιμοποιούνται για τον σκοπό αυτό είναι η στατική και η δυναμική ανάλυση κώδικα.

Στατική Ανάλυση Κώδικα: Αναφέρεται στην εξέταση του πηγαίου κώδικα χωρίς την εκτέλεσή του, με στόχο την ανίχνευση πιθανών ευπαθειών ή παραβιάσεων πολιτικών απορρήτου. Σύμφωνα με την εργασία των Tang, Østvold και Bruntink [6], η στατική ανάλυση μπορεί να βοηθήσει τους ελεγκτές κώδικα να εντοπίσουν και να δώσουν προτεραιότητα σε τμήματα του κώδικα που σχετίζονται με την επεξεργασία προσωπικών δεδομένων, διευκολύνοντας έτσι τη διαδικασία συμμόρφωσης με τον GDPR.

Δυναμική Ανάλυση Κώδικα: Αυτή η τεχνική περιλαμβάνει την αξιολόγηση του λογισμικού κατά την εκτέλεσή του, επιτρέποντας την παρακολούθηση της πραγματικής συμπεριφοράς του συστήματος και την ανίχνευση πιθανών παραβιάσεων της ιδιωτικότητας σε πραγματικό χρόνο. Η δυναμική ανάλυση είναι ιδιαίτερα χρήσιμη για την κατανόηση της ροής των δεδομένων και την αναγνώριση μη αναμενόμενων συμπεριφορών που μπορεί να οδηγήσουν σε διαρροή προσωπικών δεδομένων.

Η συνδυασμένη χρήση στατικής και δυναμικής ανάλυσης προσφέρει μια ολοκληρωμένη προσέγγιση για την αξιολόγηση της συμμόρφωσης με τον GDPR. Το εργαλείο CodeScanner αξιοποιεί την στατική ανάλυση κώδικα με σκοπό να συγκρίνει τις δηλωμένες πολιτικές απορρήτου (σε μορφή XML) με τις πραγματικές λειτουργίες του κώδικα JavaScript, εντοπίζοντας αποκλίσεις και διασφαλίζοντας τη συνεπή εφαρμογή των πολιτικών απορρήτου.

2.2.6 Personal Data Processing Actions in Web Applications

Η έννοια της «επεξεργασίας» προσωπικών δεδομένων καλύπτει ένα ευρύ φάσμα ενεργειών, σύμφωνα με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR). Το Άρθρο 4[7] του Κανονισμού ορίζει την επεξεργασία ως «κάθε πράξη ή σειρά πράξεων που πραγματοποιείται επί προσωπικών δεδομένων», περιλαμβάνοντας από τη συλλογή έως και τη διαγραφή τους.

Στο πλαίσιο των διαδικτυακών εφαρμογών, και ιδιαίτερα στον πηγαίο κώδικα JavaScript, οι ενέργειες αυτές συχνά υλοποιούνται με τρόπους που ενδέχεται να μην είναι άμεσα ορατοί στον τελικό χρήστη. Ενδεικτικά παραδείγματα ενεργειών επεξεργασίας δεδομένων περιλαμβάνουν:

- **Συλλογή (Collection)**: Π.χ. με την υποβολή μιας φόρμας ή τη χρήση trackers που συλλέγουν δεδομένα χωρίς εμφανή συναίνεση.
- **Αποθήκευση (Storage)**: Π.χ. χρήση localStorage, cookies ή αποστολή σε απομακρυσμένο server για αποθήκευση.
- **Τροποποίηση (Modification)**: Δυναμική επεξεργασία δεδομένων στον browser ή σε APIs.
- **Μετάδοση (Transmission)**: Αποστολή προσωπικών πληροφοριών σε τρίτους, π.χ. μέσω fetch/POST.
- **Συνδυασμός (Combination)**: Συγχώνευση προσωπικών δεδομένων με άλλα datasets για σκοπούς analytics.
- **Διαγραφή (Erasure)**: Π.χ. διαγραφή sessions χωρίς log ή μη καταγεγραμμένη διαγραφή

Η JavaScript, ως client-side γλώσσα, επιτρέπει πολλές από αυτές τις ενέργειες χωρίς να περνούν απαραίτητα από server-side ελέγχους. Αυτό καθιστά δύσκολη την ανίχνευση και την επιβεβαίωση της συμμόρφωσης με τις πολιτικές απορρήτου και τον GDPR.

Το εργαλείο **CodeScanner** αξιοποιεί καταλόγους ενεργειών επεξεργασίας, οι οποίοι έχουν εξαχθεί από τη νομοθεσία και τη βιβλιογραφία, ώστε να μπορεί να ελέγχει αν τέτοιες ενέργειες εμφανίζονται στον JavaScript κώδικα και αν αυτές αντιστοιχούν με τις δηλώσεις της εκάστοτε πολιτικής απορρήτου (σε μορφή XML).

Αυτή η προσέγγιση επιτρέπει την πλήρη χαρτογράφηση της χρήσης προσωπικών δεδομένων στον πηγαίο κώδικα και την ανίχνευση αποκλίσεων μεταξύ του τι δηλώνεται και του τι πραγματικά υλοποιείται.

2.3 Υπάρχοντα Εργαλεία

Για την επίτευξη συμμόρφωσης με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR), έχουν αναπτυχθεί διάφορα εργαλεία που προσφέρουν λειτουργικότητες όπως η ανάλυση πολιτικών απορρήτου, η ανίχνευση προσωπικών δεδομένων και η επαλήθευση συμμόρφωσης. Η παρούσα ενότητα παρουσιάζει αντιπροσωπευτικά εργαλεία της αγοράς και συγκρίνει τις δυνατότητές τους σε σχέση με το προτεινόμενο εργαλείο CodeScanner.

Ο συγκριτικός πίνακας που ακολουθεί συνοψίζει τα βασικά χαρακτηριστικά των εργαλείων ως προς:

- τη δυνατότητα ανάλυσης πηγαίου κώδικα,
- την ανίχνευση προσωπικών δεδομένων,
- τον έλεγχο συμμόρφωσης με τον GDPR,
- τη διαθεσιμότητα δωρεάν έκδοσης,
- και άλλες σημαντικές παρατηρήσεις

Συγκριτικός Πίνακας Εργαλείων *(Ο πίνακας παρατίθεται στην επόμενη σελίδα σε οριζόντια διάταξη – landscape)*

Το CodeScanner διαφοροποιείται από τα υπόλοιπα εργαλεία, καθώς:

- Εστιάζει ειδικά σε πηγαίο κώδικα JavaScript, τη βασική γλώσσα για δυναμικές web εφαρμογές.
- Χρησιμοποιεί πολιτικές απορρήτου σε XML μορφή ως σημείο αναφοράς.
- Ελέγχει αν οι εντολές του κώδικα είναι σύμφωνες με τις δηλωμένες ενέργειες της πολιτικής (π.χ. συλλογή, αποθήκευση, διαβίβαση δεδομένων).
- Εντοπίζει πιθανές παραβιάσεις ή παραλείψεις στη χρήση προσωπικών δεδομένων βάσει των απαιτήσεων του GDPR.

Σε αντίθεση με εμπορικά εργαλεία που περιορίζονται σε έλεγχο πολιτικών ή διαχείριση συγκατάθεσης, το CodeScanner συνδυάζει ανάλυση πολιτικής και τεχνικό έλεγχο κώδικα, ενισχύοντας έτσι την προσέγγιση “Privacy by Design”.

Όνομα Εργαλείου	Περιγραφή	Ανάλυση Κώδικα	Ανίχνευση Προσωπικών Δεδομένων	Έλεγχος Συμμόρφωσης GDPR	Δωρεάν Έκδοση	Επιπλέον Χρεώσεις
Το εργαλείο μας	Ελέγχει τον πηγαίο κώδικα και εξετάζει πώς χρησιμοποιούνται τα προσωπικά δεδομένα.	✓(javascript)	✓	✓	✓	❌
SHEQAPP Compliance software	Κάνει compliance checking για κανονισμούς της ΕΕ.	❌	❌	✓	❌	✓ (Επαγγελματικές άδειες)
Usercentrics	Διαχειρίζεται τη συναίνεση των χρηστών και συμμορφώνεται με GDPR.	✓(Web εφαρμογές)	❌	✓	❌	✓ (Συνδρομητικό μοντέλο, χρήση εταιρίας)
Visure Requirements ALM	Επιτρέπει ιχνηλασιμότητα του πηγαίου κώδικα για κανονιστική συμμόρφωση.	✓(C, C++, Java, Python)	❌	✓	❌	✓ (Ανάλογα με την άδεια χρήσης)

2.4 Σύνοψη Μελέτης και Συμπεράσματα

Η βιβλιογραφική μελέτη που παρουσιάστηκε στο παρόν κεφάλαιο ανέδειξε τη σημασία της προστασίας προσωπικών δεδομένων στο πλαίσιο της ανάπτυξης διαδικτυακών εφαρμογών και τη συσχέτισή της με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR). Ιδιαίτερη έμφαση δόθηκε στην επεξεργασία προσωπικών δεδομένων μέσω πηγαίου κώδικα, με επίκεντρο τη JavaScript, καθώς και στις τεχνικές στατικής και δυναμικής ανάλυσης για τον έλεγχο συμμόρφωσης.

Αναλύθηκαν επιστημονικές εργασίες που παρουσιάζουν μεθόδους αυτόματης ανίχνευσης επεξεργασιών δεδομένων, συσχέτισης πολιτικών απορρήτου με τον πηγαίο κώδικα και ενσωμάτωσης του ελέγχου συμμόρφωσης σε διαδικασίες ανάπτυξης λογισμικού. Επιπλέον, καταγράφηκαν ενέργειες που θεωρούνται «επεξεργασία» κατά τον GDPR, οι οποίες αξιοποιούνται για τον εντοπισμό κρίσιμων λειτουργιών στον κώδικα.

Τέλος, παρουσιάστηκε σύγκριση υφιστάμενων εργαλείων της αγοράς, με στόχο να αναδειχθούν τα πλεονεκτήματα και οι ελλείψεις τους σε σχέση με το προτεινόμενο εργαλείο CodeScanner. Το εργαλείο αυτό επιχειρεί να καλύψει ένα κρίσιμο κενό στην ανάλυση και συσχέτιση των δηλώσεων πολιτικής απορρήτου με τον πραγματικό πηγαίο κώδικα, εστιάζοντας στη γλώσσα JavaScript και αξιοποιώντας την XML ως μορφή αναφοράς πολιτικής.

Η γνώση που αποκτήθηκε από τη μελέτη αυτή αποτέλεσε τη βάση για τον σχεδιασμό, την ανάπτυξη και την αξιολόγηση του εργαλείου, τα οποία παρουσιάζονται στα επόμενα κεφάλαια.

Κεφάλαιο 3

Έρευνα και Ανάλυση Προσωπικών Δεδομένων

3.1 Εισαγωγή

Η προστασία των προσωπικών δεδομένων αποτελεί σήμερα θεμελιώδη προτεραιότητα, ιδίως στον τομέα της ανάπτυξης λογισμικού και των διαδικτυακών υπηρεσιών. Κατά την διαδικασία δημιουργίας του εργαλείου CodeScanner, κρίθηκε απαραίτητο να προηγηθεί μια συστηματική έρευνα πάνω στις ενέργειες που συνδέονται με την επεξεργασία προσωπικών δεδομένων, καθώς και στον τρόπο με τον οποίο αυτές υλοποιούνται μέσω κώδικα.

Η έρευνα περιλάμβανε δύο βασικά στάδια:

- Καταγραφή και κατηγοριοποίηση ενεργειών βάσει του Γενικού Κανονισμού για την Προστασία Δεδομένων (GDPR).
- Ανάλυση πηγαίου κώδικα σε 5 γλώσσες προγραμματισμού, με στόχο τον εντοπισμό εντολών που αντιστοιχούν στις GDPR ενέργειες.

Τα αποτελέσματα της έρευνας αποτέλεσαν τον οδηγό για την κατασκευή της μηχανής εντοπισμού του εργαλείου, όπως περιγράφεται στο επόμενο κεφάλαιο.

3.2 Ενέργειες Προσωπικών Δεδομένων βάσει GDPR

Ο Γενικός Κανονισμός για την Προστασία Δεδομένων (GDPR) περιγράφει μια σειρά από ενέργειες που σχετίζονται με την επεξεργασία προσωπικών δεδομένων[7]. Στο πλαίσιο της έρευνας, οι ενέργειες αυτές ομαδοποιήθηκαν σε 6 βασικές κατηγορίες και συσχετίστηκαν με πράξεις που εντοπίζονται σε κώδικα λογισμικού:

ΕΝΕΡΓΕΙΑ	ΠΕΡΙΓΡΑΦΗ
COLLECTION	Συλλογή δεδομένων από τον χρήστη μέσω φορμών ή λειτουργιών εισαγωγής στον κώδικα.
STORAGE	Αποθήκευση δεδομένων τοπικά (π.χ. localStorage), Server ή σε βάση δεδομένων.
TRANSMISSION	Αποστολή δεδομένων μέσω δικτύου (π.χ. fetch, XMLHttpRequest, sockets).
MODIFICATION	Τροποποίηση της τιμής ή της κατάστασης προσωπικών δεδομένων.
DELETION	Διαγραφή δεδομένων από τοπική ή απομακρυσμένη αποθήκη.
COMBINATION	Συγχώνευση/σύνδεση διαφορετικών προσωπικών δεδομένων με σκοπό δημιουργία νέου προφίλ ή ταυτότητας.

Κάθε μία από τις παραπάνω ενέργειες έχει διαφορετική σημασία όσον αφορά τη νομική υποχρέωση καταγραφής και τη συμμόρφωση σύμφωνα με το GDPR. Η κατανόηση αυτών των ενεργειών είναι απαραίτητη για την αναγνώρισή τους μέσα στον πηγαίο κώδικα εφαρμογών.

3.3 Μελέτη Εντολών σε Πολλαπλές Γλώσσες Προγραμματισμού

3.3.1 Στόχος και Μεθοδολογία

Σκοπός της παρούσας έρευνας ήταν η ταυτοποίηση εντολών πηγαίου κώδικα που σχετίζονται με ενέργειες επί προσωπικών δεδομένων, όπως αυτές ορίζονται στον GDPR. Για να καταστεί δυνατός ο εντοπισμός τους από το εργαλείο CodeScanner, πραγματοποιήθηκε συστηματική μελέτη 5 γλωσσών προγραμματισμού: JavaScript, PHP, Java, Python και C++.

Η μεθοδολογία που ακολουθήθηκε περιλάμβανε τα εξής στάδια:

1. Καθορισμός των ενεργειών βάσει GDPR: συλλογή, αποθήκευση, μετάδοση, τροποποίηση, διαγραφή και συνδυασμός προσωπικών δεδομένων.
2. Επιλογή γλωσσών που χρησιμοποιούνται ευρέως σε ανάπτυξη web και backend εφαρμογών.
3. Καταγραφή εντολών ανά γλώσσα που πραγματοποιούν τις παραπάνω ενέργειες.
4. Κατηγοριοποίηση εντολών σε πίνακες ανά δράση (action) και γλώσσα, ώστε να ενσωματωθούν στη λογική του εργαλείου.

Η μελέτη πραγματοποιήθηκε αξιοποιώντας κυρίως το περιεχόμενο της πλατφόρμας W3Schools, η οποία προσφέρει απλά, τυποποιημένα παραδείγματα εντολών για τις περισσότερες γλώσσες προγραμματισμού.

3.3.2 Πίνακες εντολών ανά γλώσσα και κατηγορία

3.3.2.1 Εντολές Συλλογής Δεδομένων (Collection)

Η συλλογή προσωπικών δεδομένων αφορά τον αρχικό χειρισμό τους από τον χρήστη. Οι παραπάνω εντολές διαβάζουν δεδομένα είτε από το περιβάλλον (π.χ. input πεδίο ή κονσόλα) είτε από φόρμα. Στο πλαίσιο του GDPR, αυτή είναι η πρώτη φάση όπου τίθενται υπό επεξεργασία προσωπικά στοιχεία, όπως ονόματα, emails ή αριθμοί τηλεφώνου.

Γλώσσα	Παράδειγμα Εντολής
JavaScript	let name = prompt("Enter name");
PHP	$name = $_POST['name'];
Java	Scanner input = new Scanner(System.in); String name = input.nextLine();
Python	name = input("Enter your name")
C++	string name; cin >> name;

3.3.2.2 Εντολές Αποθήκευσης Δεδομένων (Storage)

Η αποθήκευση δεδομένων αφορά την τοπική ή απομακρυσμένη αποθήκευση προσωπικών στοιχείων, είτε σε βάση δεδομένων είτε σε τοπικά αρχεία ή browser storage.

Γλώσσα	Παράδειγμα Εντολής
JavaScript	localStorage.setItem("email", userEmail);
PHP	mysqli_query($conn, "INSERT INTO users (email) VALUES ('$email')");
Java	PreparedStatement stmt = conn.prepareStatement("INSERT INTO users VALUES (?)");
Python	cursor.execute("INSERT INTO users (email) VALUES (?)", (email,))
C++	ofstream myfile("userdata.txt"); myfile << email;

3.3.2.3 Εντολές Μετάδοσης Δεδομένων (Transmission)

Η μετάδοση δεδομένων αναφέρεται στην αποστολή προσωπικών πληροφοριών μέσω δικτύου προς εξωτερικά συστήματα, API ή βάσεις δεδομένων. Αυτή η ενέργεια σχετίζεται με την κοινή χρήση ή την ανταλλαγή δεδομένων και συνιστά ένα από τα πιο κρίσιμα σημεία συμμόρφωσης με τον GDPR.

Γλώσσα	Παράδειγμα Εντολής
JavaScript	fetch("https://example.com", { method: "POST", body: JSON.stringify(user) });
PHP	curl_setopt($ch, CURLOPT_POSTFIELDS, http_build_query($data));
Java	HttpURLConnection conn = (HttpURLConnection) url.openConnection();
Python	requests.post("https://api.example.com", data=user_data)
C++	system("curl -X POST -d \"data=value\" https://example.com");

3.3.2.4 Εντολές Τροποποίησης Δεδομένων (Modification)

Η τροποποίηση δεδομένων αφορά την αλλαγή της τιμής προσωπικών δεδομένων που είναι ήδη αποθηκευμένα, είτε τοπικά είτε σε βάση δεδομένων ή αντικείμενα μνήμης της εφαρμογής.

ΓΛΏΣΣΑ	ΠΑΡΆΔΕΙΓΜΑ ΕΝΤΟΛΉΣ
JAVASCRIPT	user.email = "newemail@example.com";
PHP	mysqli_query($conn, "UPDATE users SET email = '$newEmail' WHERE id = 1");
JAVA	user.setEmail("newemail@example.com");
PYTHON	user['email'] = "newemail@example.com"
C++	user.email = "newemail@example.com";

Πίνακας 3.3.2.5 – Εντολές Διαγραφής Δεδομένων (Deletion)

Η διαγραφή δεδομένων αφορά την οριστική αφαίρεση προσωπικών πληροφοριών από την εφαρμογή ή το σύστημα αποθήκευσης.

Γλώσσα	Παράδειγμα Εντολής
JavaScript	localStorage.removeItem("email");
PHP	mysqli_query($conn, "DELETE FROM users WHERE email = '$email'");
Java	conn.prepareStatement("DELETE FROM users WHERE email = ?");
Python	cursor.execute("DELETE FROM users WHERE email = ?", (email,))
C++	remove("userdata.txt");

Πίνακας 3.3.2.6 – Εντολές Συνδυασμού Δεδομένων (Combination)

Ο συνδυασμός δεδομένων περιλαμβάνει τη συγχώνευση ή ομαδοποίηση πληροφοριών από διαφορετικές πηγές ή πεδία, ενέργεια που μπορεί να οδηγήσει σε αναγνώριση ταυτότητας ακόμη και από μη άμεσα προσωπικά δεδομένα.

Γλώσσα	Παράδειγμα Εντολής
JavaScript	let combined = Object.assign({}, userInfo, preferences);
PHP	$combined = array_merge($userInfo, $preferences);
Java	Map<String, String> combined = new HashMap<>(userInfo); combined.putAll(preferences);
Python	combined = {**user_info, **preferences}
C++	combined.insert(userInfo.begin(), userInfo.end()); combined.insert(preferences.begin(), preferences.end());

3.4 Λίστα Δημοφιλών Προσωπικών Δεδομένων

Πριν την υλοποίηση του εργαλείου, πραγματοποιήθηκε προκαταρκτική ερευνητική φάση με στόχο την καταγραφή των πιο συχνά χρησιμοποιούμενων λέξεων-κλειδιών που αντιστοιχούν σε προσωπικά δεδομένα. Οι λέξεις αυτές προέρχονται από τον ορισμό των προσωπικών δεδομένων όπως περιγράφεται στο Άρθρο 4 του Γενικού

Κανονισμού για την Προστασία Δεδομένων (GDPR) [7], καθώς και από συναφείς ερευνητικές εργασίες [8].

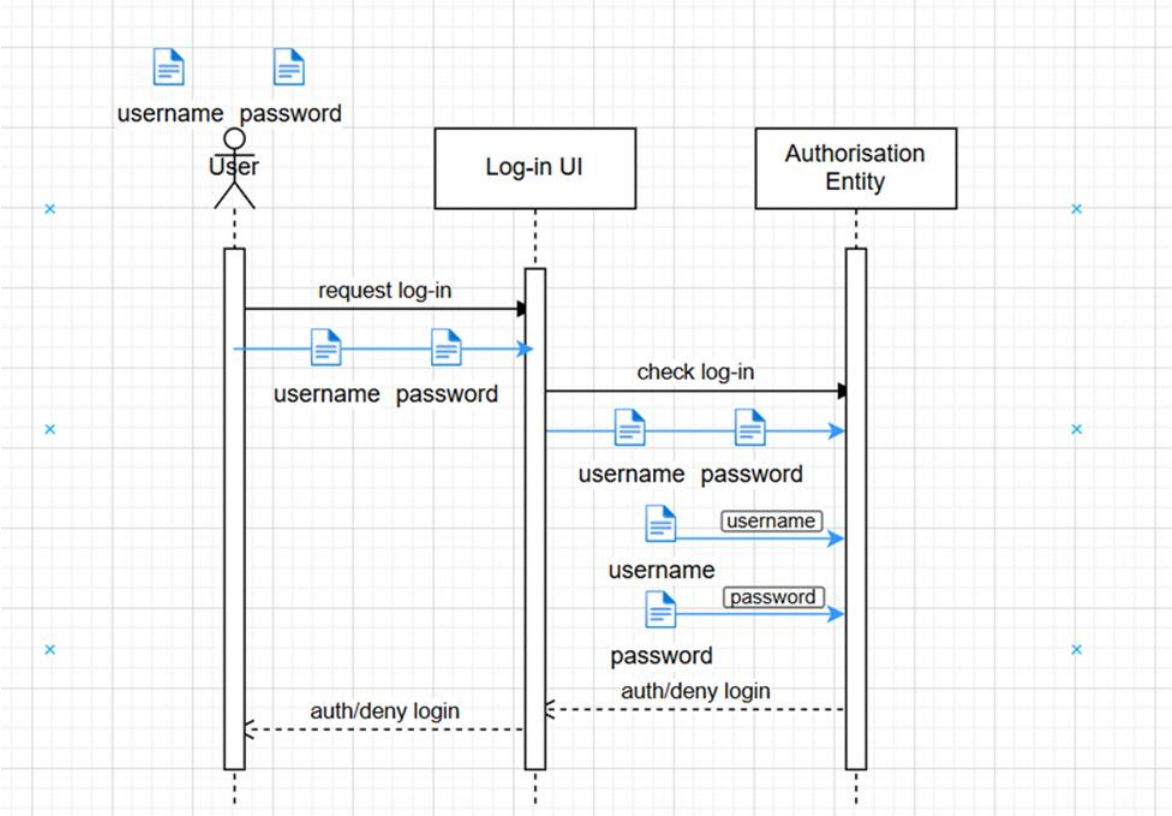
Η παρακάτω λίστα δημιουργήθηκε με βάση αυτό το υπόβαθρο και αποτέλεσε θεμέλιο για τον εντοπισμό δεδομένων κατά τη στατική ανάλυση κώδικα:

- "user", "client", "customer", "member", "name", "firstname", "lastname", "fullname", "email", "mail", "phone", "mobile", "contact", "address", "street", "city", "state", "zip", "postal", "country", "dob", "dateofbirth", "birthdate", "ssn", "socialsecurity", "credit", "card", "creditcard", "identifier", "id", "uniqueidentifier", "passport", "driverlicense", "account", "login", "password", "secret", "token", "profile", "financial", "bank", "iban", "routing", "tax", "tin".

Αυτή η επιλογή αποτέλεσε τη βάση για τις επόμενες εκδόσεις του εργαλείου, εξασφαλίζοντας στοχευμένο εντοπισμό κατά την ανάλυση.

3.5 Πολιτικές Επεξεργασίας Δεδομένων

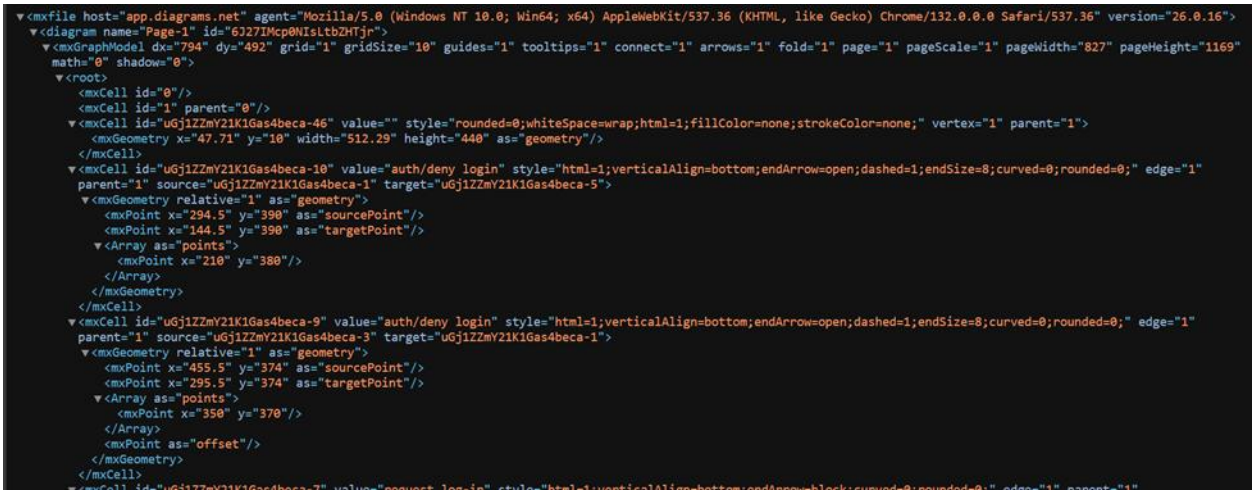
Στο πλαίσιο της παρούσας διπλωματικής εργασίας, υιοθετήθηκαν τα Purpose-Aware Sequence Diagrams για τη μοντελοποίηση και δήλωση των πολιτικών επεξεργασίας προσωπικών δεδομένων κατά το σχεδιασμό λογισμικού[14]. Τα διαγράμματα αυτά αποτελούν επέκταση των κλασικών διαγραμμάτων ακολουθίας (sequence diagrams), προσθέτοντας τη δυνατότητα να αποτυπώνεται ο τύπος της επεξεργασίας προσωπικών δεδομένων μεταξύ των εμπλεκόμενων οντοτήτων.



Η Εικόνα 3.5.1 παρουσιάζει ένα τέτοιο παράδειγμα. Οι κάθετες στήλες απεικονίζουν τις οντότητες του συστήματος (User, Log-in UI, Authorisation Entity), ενώ τα βέλη αναπαριστούν την ανταλλαγή προσωπικών δεδομένων (username, password) μαζί με ενδείξεις για τον τύπο της πράξης (π.χ. request, check, auth/deny). Στο παράδειγμα παρουσιάζεται ένα σύστημα ταυτοποίησης, όπου ο χρήστης εισάγει τα στοιχεία του και αυτά ελέγχονται από την αρμόδια οντότητα εσωτερικά.

Για τις ανάγκες της εργασίας, υιοθετήθηκε η παραδοχή ότι κάθε οντότητα αντιστοιχεί σε μία JavaScript συνάρτηση (function). Κάθε συνάρτηση δηλώνει ρητά τον ρόλο της στην επεξεργασία των προσωπικών δεδομένων: εάν συλλέγει, μεταδίδει ή διαβάζει δεδομένα, και με ποιες άλλες οντότητες επικοινωνεί.

Τα διαγράμματα αυτά μπορούν να μετατραπούν σε XML μορφή, ώστε να αναλυθούν από το εργαλείο CodeScanner.



Η Εικόνα 3.5.2: XML αναπαράσταση του sequence diagram της Εικόνας 3.5.1. Οι οντότητες και οι ενέργειες επεξεργασίας δεδομένων περιγράφονται με δομημένο τρόπο, επιτρέποντας τη μηχανική ανάγνωση και τον έλεγχο συμμόρφωσης από το εργαλείο CodeScanner.

3.6 Παρατηρήσεις – Συμπεράσματα

Από την παραπάνω ανάλυση προκύπτουν ορισμένες βασικές παρατηρήσεις που συνέβαλαν στον σχεδιασμό και τη λειτουργία του εργαλείου CodeScanner:

- Ομοιότητες στον χειρισμό δεδομένων: Παρότι οι γλώσσες διαφέρουν συντακτικά, οι εντολές που σχετίζονται με προσωπικά δεδομένα έχουν παρόμοια λογική, επιτρέποντας τη γενίκευση σε κατηγορίες (collection, storage, etc).
- JavaScript – η κύρια γλώσσα του Web: Η JavaScript επιλέχθηκε ως η βασική γλώσσα ανάλυσης επειδή αποτελεί τον θεμέλιο λίθο για ιστοσελίδες και web εφαρμογές. Όλες σχεδόν οι αλληλεπιδράσεις με τον χρήστη, η συλλογή δεδομένων από φόρμες και η αποθήκευση στο τοπικό περιβάλλον του browser υλοποιούνται μέσω JavaScript. Επομένως, η ανάλυσή της είναι κρίσιμη για τον εντοπισμό εντολών που διαχειρίζονται προσωπικά δεδομένα σε περιβάλλοντα web. Έτσι αποφασίστηκε ότι το πρωτότυπο εργαλείο CodeScanner θα αναλύει την συγκεκριμένη γλώσσα στην έκδοση που θα αναπτυχθεί - με δυνατότητα επέκτασης στις υπόλοιπες γλώσσες
- XML δομή για πολιτικές: Η χρήση XML για την περιγραφή πολιτικών απορρήτου είναι λειτουργικά επαρκής και επεξεργάσιμη προγραμματιστικά, ιδανική για σύγκριση με τον κώδικα.

- Καθορισμός συνωνύμων και παραλλαγών: Για κάθε κατηγορία εντολών εντοπίστηκε πλήθος παραλλαγών και συνώνυμων όρων (π.χ. user, client, member), τα οποία περιλήφθηκαν ως λέξεις-κλειδιά στο εργαλείο.
- Αναγκαιότητα συσχέτισης με GDPR: Η ερευνητική φάση απέδειξε ότι η σύγκριση των εντολών με τις ενέργειες που περιγράφει ο GDPR είναι κρίσιμη για να εντοπιστούν πιθανές παραβιάσεις πολιτικών ή μη δηλωμένες χρήσεις δεδομένων.

Η μελέτη αυτή αποτέλεσε τη βάση για την υλοποίηση των κανόνων εντοπισμού στο CodeScanner και οδήγησε στην αναγνώριση των κατάλληλων patterns που εφαρμόστηκαν στις εκδόσεις του εργαλείου.

Κεφάλαιο 4

Κύκλος Ζωής και Ανάπτυξη του Εργαλείου

4.1 Εισαγωγή

Η ανάπτυξη του εργαλείου CodeScanner πραγματοποιήθηκε με βάση μία επαναληπτική και σταδιακή διαδικασία. Στόχος ήταν η συνεχής βελτίωση της λειτουργικότητας και της ευχρηστίας του εργαλείου μέσα από τη δημιουργία και αξιολόγηση διαδοχικών εκδόσεων. Για τη μεθοδολογία ανάπτυξης επιλέχθηκε το μοντέλο throw-away prototyping, το οποίο επιτρέπει τη γρήγορη δημιουργία πρόχειρων εκδόσεων (πρωτοτύπων) που αξιολογούνται και τροποποιούνται πριν την τελική υλοποίηση.

Η επιλογή αυτής της μεθόδου βασίστηκε στην ανάγκη για ευελιξία κατά τη διάρκεια της ανάπτυξης, καθώς και στην αποτελεσματικότητα που προσφέρει στην ανίχνευση προβλημάτων σχεδίασης και λειτουργίας σε πρώιμο στάδιο. Όπως υποστηρίζουν και οι Larman & Basili [12], η επαναληπτική ανάπτυξη μέσω πρωτοτύπων συμβάλλει στη βελτίωση της ποιότητας του λογισμικού και της εμπειρίας χρήστη.

4.2 Καταγραφή Απαιτήσεων και Σχεδιασμός Πρωτοτύπων

Η ανάπτυξη του εργαλείου CodeScanner ξεκίνησε με τον εντοπισμό των βασικών λειτουργικών απαιτήσεων, που αφορούσαν:

- την αυτόματη ανάλυση JavaScript πηγαίου κώδικα,
- την αναγνώριση ενεργειών επεξεργασίας προσωπικών δεδομένων (π.χ. συλλογή, αποθήκευση, αποστολή),
- τη δυνατότητα σύγκρισης με πολιτικές απορρήτου σε XML μορφή,
- και την παρουσίαση αποτελεσμάτων με τρόπο απλό, κατανοητό και χρήσιμο για τον τελικό χρήστη.

Όπως αναφέρθηκε πιο πάνω, η διαδικασία ανάπτυξης ακολούθησε τη μεθοδολογία throw-away prototyping, με τη δημιουργία διαδοχικών πρωτοτύπων (v1 έως v5). Κάθε έκδοση εστίαζε στη βελτίωση διαφορετικών πτυχών του εργαλείου, όπως:

- η ακρίβεια εντοπισμού εντολών JavaScript σχετικών με προσωπικά δεδομένα,
- η κάλυψη περισσότερων τύπων αποθήκευσης (π.χ. IndexedDB),
- η αξιοπιστία της ανάλυσης,
- και η αναγνωσιμότητα της παραγόμενης αναφοράς.

Στην τελική υλοποίηση, ενσωματώθηκαν στοιχεία από δύο εκδόσεις:

- Η έκδοση v3, η οποία εκτελεί ανάλυση μόνο του JavaScript πηγαίου κώδικα, χρησιμοποιώντας κανονικές εκφράσεις (regex) για τον εντοπισμό ενεργειών όπως fetch(), localStorage.setItem(), XMLHttpRequest(), κ.λπ.
- Η έκδοση v5, η οποία επεκτείνει τη λειτουργικότητα μέσω της εισαγωγής XML πολιτικής απορρήτου. Χρησιμοποιεί parser για να διαβάζει από XML τα επιτρεπόμενα δεδομένα και συναρτήσεις, και συγκρίνει δυναμικά τον κώδικα με τις δηλωμένες πολιτικές, εντοπίζοντας πιθανές αποκλίσεις.

Η χρήση και των δύο εκδόσεων στο UI επιτρέπει στο εργαλείο να καλύπτει διαφορετικά σενάρια χρήσης. Η έκδοση v3 είναι κατάλληλη για περιπτώσεις όπου απαιτείται βασικός έλεγχος του JavaScript πηγαίου κώδικα, ενώ η έκδοση v5 προσφέρει εκτενέστερη ανάλυση, ενσωματώνοντας και έλεγχο συμμόρφωσης με πολιτική απορρήτου σε XML μορφή. Με αυτόν τον τρόπο, ο χρήστης έχει τη δυνατότητα να επιλέξει τη λειτουργικότητα που ανταποκρίνεται καλύτερα στις

ανάγκες της εκάστοτε εφαρμογής. Για παράδειγμα, σε περίπτωση που ο χρήστης δεν έχει την δηλωμένη πολιτική απόρρητο αλλά θέλει να πάρει αναφορά όσον αφορά την χρήση των προσωπικών δεδομένων από τον εκάστοτε πηγαίο κώδικα.

4.3 Προδιαγραφές Εργαλείου

Το εργαλείο CodeScanner έχει σχεδιαστεί για να εντοπίζει την ύπαρξη και χρήση προσωπικών δεδομένων στον πηγαίο κώδικα JavaScript και να αξιολογεί τη συμμόρφωση με πολιτικές απορρήτου (σε μορφή XML). Οι κύριες λειτουργικές προδιαγραφές του εργαλείου είναι οι εξής:

Λειτουργίες:

- Ανίχνευση εντολών JavaScript που σχετίζονται με επεξεργασία προσωπικών δεδομένων, όπως για παράδειγμα:
  - localStorage, sessionStorage
  - document.cookie
  - fetch(), XMLHttpRequest()
- Σύγκριση με XML πολιτική εάν επιλεγεί το αντίστοιχο mode.
- Παραγωγή αναφοράς με τις εντολές που εντοπίστηκαν και τη συσχέτισή τους με την πολιτική.

Είσοδοι:

- Ένα αρχείο JavaScript (.js).
- Προαιρετικά, ένα αρχείο XML (.xml) με δηλωμένες επιτρεπόμενες ενέργειες και τύπους δεδομένων.

Έξοδοι:

- Αναφορά (σε μορφή κειμένου) που περιλαμβάνει:
  - Τις εντολές που εντοπίστηκαν.
  - Τα δεδομένα που φαίνεται να χρησιμοποιούνται.
  - Αν υπάρχει XML, έλεγχος αν είναι επιτρεπόμενα ή όχι.

Τεχνικά χαρακτηριστικά:

- Γραμμένο σε Java, με χρήση του Swing framework για το γραφικό περιβάλλον (GUI).
- Παρέχει δύο modes λειτουργίας:
  - JS Scan Only – ανάλυση μόνο του JavaScript αρχείου.
  - JS + XML Comparison – ανάλυση και συσχέτιση με XML πολιτική απορρήτου.
- Το UI είναι φιλικό προς τον χρήστη και υποστηρίζει εισαγωγή αρχείων με κουμπιά περιήγησης ("Browse JS", "Browse XML").

4.3.1 Τεχνολογική Υποδομή και Εργαλεία

Κατά την ανάπτυξη του εργαλείου CodeScanner, αξιοποιήθηκαν επιλεγμένες τεχνολογίες που υποστηρίζουν τη διαδικασία στατικής ανάλυσης πηγαίου κώδικα, την επεξεργασία πολιτικών απορρήτου και τη δημιουργία διεπαφής χρήστη. Η επιλογή των τεχνολογιών έγινε με βάση την ευχρηστία, την πληρότητα σε βιβλιοθήκες και την υποστήριξη σύνθετων αρχείων και δομών.

Γλώσσες και Περιβάλλοντα

- **Java**
  Αποτελεί τη βασική γλώσσα υλοποίησης του εργαλείου. Χρησιμοποιήθηκε για τη δημιουργία της μηχανής ανάλυσης, της διαχείρισης αρχείων JS και XML, της εσωτερικής λογικής ελέγχου, αλλά και για τη δημιουργία του γραφικού περιβάλλοντος χρήστη (Swing). Η Java προσφέρει ισχυρή υποστήριξη για χειρισμό αρχείων, parsing, καθώς και καλή επεκτασιμότητα, κάτι που την καθιστά ιδανική για την ανάπτυξη εργαλείων στατικής ανάλυσης. Σύμφωνα με τους Talha et al. [8], η στατική ανάλυση σε εφαρμογές Java ενισχύει την ακρίβεια και την πληρότητα στον εντοπισμό παραβιάσεων ασφάλειας και απορρήτου.
- **JavaScript**
  Είναι η γλώσσα προγραμματισμού που αναλύεται από το εργαλείο. Λόγω της κυρίαρχης θέσης της στην ανάπτυξη διαδικτυακών εφαρμογών [13] και της στενής σχέσης της με τον χρήστη, αποτελεί βασικό στόχο για την ανάλυση χρήσης προσωπικών δεδομένων. Το εργαλείο εντοπίζει εντολές όπως localStorage, cookies, fetch, XMLHttpRequest, οι οποίες σχετίζονται με συλλογή, αποθήκευση και αποστολή προσωπικών δεδομένων. Η ανάλυση της JavaScript θεωρείται ιδιαίτερα απαιτητική λόγω της δυναμικής φύσης της, αλλά είναι εφικτή με χρήση κατάλληλων τεχνικών [8].
- **XML**
  Χρησιμοποιείται για την περιγραφή των πολιτικών απορρήτου. Το XML αρχείο περιλαμβάνει λέξεις-κλειδιά που σχετίζονται με προσωπικά δεδομένα και συναρτήσεις που επιτρέπεται να τις επεξεργάζονται. Η επιλογή του XML βασίζεται στη δομημένη του φύση και στην ευκολία ανάγνωσης/επεξεργασίας από προγράμματα. Μελέτες όπως αυτή των Tang et al. [9] υποστηρίζουν ότι η XML μπορεί να χρησιμοποιηθεί αποτελεσματικά για την αναπαράσταση αποφάσεων απορρήτου και για την υποστήριξη ελέγχων συμμόρφωσης.
- **HTML & CSS (σχεδιαστική λογική UI)**

Αν και δεν χρησιμοποιούνται άμεσα, η σχεδίαση του UI του εργαλείου ακολουθεί τις αρχές της HTML/CSS: καθαρή διάταξη, σαφή κουμπιά εντολών, χρωματικές αντιθέσεις, και εργονομική προβολή αναφορών.

Περιβάλλοντα ανάπτυξης και εργαλεία

- **Eclipse IDE**
  Χρησιμοποιήθηκε ως το κύριο περιβάλλον ανάπτυξης για την Java. Παρέχει υποστήριξη για debugging, διαχείριση βιβλιοθηκών, παρακολούθηση έργων και plugins που διευκολύνουν τη δημιουργία εργαλείων στατικής ανάλυσης. Το Eclipse θεωρείται ιδιαίτερα αποδοτικό περιβάλλον για ακαδημαϊκή ανάπτυξη εργαλείων λογισμικού [10].
- **Visual Studio Code (VS Code)**
  Χρησιμοποιήθηκε κυρίως για την επεξεργασία αρχείων JavaScript και XML. Προσφέρει ευκολία στη διαχείριση πολλών γλωσσών, διαθέτει πλούσια βιβλιοθήκη επεκτάσεων και υποστηρίζει εργαλεία αυτοματοποίησης και μορφοποίησης. Σύμφωνα με τον Bernát [11], το VS Code έχει αποκτήσει σημαντική θέση σε ακαδημαϊκά και ερευνητικά περιβάλλοντα, χάρη στις επεκτάσεις του όπως το GPTutor που υποστηρίζει τη βελτιωμένη κατανόηση κώδικα.

4.4 Πρωτότυπα

Η ανάπτυξη του εργαλείου CodeScanner βασίστηκε στη μέθοδο του throw-away prototyping, με διαδοχικές εκδόσεις που αξιολογήθηκαν, τροποποιήθηκαν και σταδιακά οδήγησαν στην τελική μορφή του εργαλείου. Δημιουργήθηκαν πέντε εκδόσεις (v1 έως v5), οι οποίες εξελίχθηκαν με βάση τις ανάγκες του συστήματος και τις τεχνικές απαιτήσεις που προέκυπταν κατά την ανάπτυξη.

4.4.1 Έκδοση 1 (v1)

Η πρώτη έκδοση του εργαλείου CodeScanner αποτέλεσε μια βασική έκδοση γραμμής εντολών, γραμμένη σε Java. Επικεντρωνόταν στην ανάλυση JavaScript αρχείων χωρίς γραφικό περιβάλλον, και στηριζόταν σε σύνολο κανονικών εκφράσεων (regular expressions) για τον εντοπισμό εντολών που σχετίζονται με την επεξεργασία προσωπικών δεδομένων.

Το εργαλείο μπορούσε να αναγνωρίσει εντολές που σχετίζονται με:

- **Συλλογή δεδομένων** (π.χ. document.getElementById()),
- **Αποθήκευση** (localStorage.setItem(), sessionStorage.setItem(), IndexedDB store.put()),
- **Μετάδοση** (fetch(), XMLHttpRequest.send()),
- **Τροποποίηση και συνδυασμό** (Object.assign()),

- **Διαγραφή** (localStorage.removeItem(), sessionStorage.removeItem()),
- **Μη ασφαλή χρήση HTTP** σε fetch() και XMLHttpRequest().

Η έξοδος του εργαλείου παραγόταν σε μορφή αναφοράς (plain text) με αριθμημένες εντολές, το είδος της ενέργειας, το σημείο στον κώδικα και αν σχετιζόταν με προσωπικά δεδομένα.

Αν και η έκδοση v1 δεν περιλάμβανε σύγκριση με XML, έθεσε τη βάση για την καταγραφή των εντολών που αναγνωρίζονται και την ανάλυση της ροής δεδομένων στον JS κώδικα.

4.4.2 Έκδοση 2 (v2) – Προσωποποιημένος Έλεγχος μέσω Λέξεων-Κλειδιών

Η δεύτερη έκδοση του εργαλείου CodeScanner διατήρησε τη μορφή εργαλείου γραμμής εντολών (CLI), εισάγοντας ωστόσο μια σημαντική βελτίωση όσον αφορά την ακρίβεια στον εντοπισμό προσωπικών δεδομένων.

Σε αντίθεση με την πρώτη έκδοση, όπου καταγράφονταν γενικά όλες οι εντολές συλλογής, αποθήκευσης ή μετάδοσης, η έκδοση v2 εφαρμόζει φιλτράρισμα βάσει στοχευμένων λέξεων-κλειδιών, ώστε να παρουσιάζονται μόνο εντολές που εμπλέκουν πράγματι προσωπικά δεδομένα.

Η επιλογή των λέξεων-κλειδιών βασίστηκε σε βιβλιογραφική έρευνα στις διατάξεις του Γενικού Κανονισμού για την Προστασία Δεδομένων (GDPR), ο οποίος ορίζει ως προσωπικό δεδομένο κάθε πληροφορία που μπορεί να ταυτοποιήσει άμεσα ή έμμεσα ένα φυσικό πρόσωπο. Με βάση αυτή την έρευνα, καταρτίστηκε μια εκτενής λίστα προσωπικών δεδομένων, η οποία παρουσιάζεται στην ενότητα 3.4 του παρόντος εγγράφου.

Κατά τη λειτουργία της v2, το εργαλείο αναλύει γραμμή προς γραμμή τον κώδικα και εντοπίζει μόνο εκείνες τις περιπτώσεις όπου προσωπικά δεδομένα, όπως όνομα χρήστη ή αριθμός κοινωνικής ασφάλισης, υφίστανται συλλογή, αποθήκευση ή αποστολή.

Η παραγόμενη αναφορά περιλαμβάνει:

- Τον αριθμό κάθε εντοπισμένης ενέργειας,
- Τη γραμμή του κώδικα όπου εντοπίστηκε,
- Τον τύπο ενέργειας (π.χ. αποθήκευση, αποστολή, τροποποίηση),
- Και ένδειξη ότι το σχετικό πεδίο αφορά προσωπικό δεδομένο.

Η έκδοση αυτή αποτέλεσε το πρώτο ουσιαστικό βήμα προς στοχευμένη και ουσιαστική ανάλυση, μειώνοντας σημαντικά τα ψευδώς θετικά αποτελέσματα

και επιτρέποντας στον χρήστη να επικεντρώνεται σε εντολές υψηλού κινδύνου, κρίσιμες για τη συμμόρφωση με τον GDPR.

4.4.3 Έκδοση 3 (v3) – JS Scan Only Επέκταση του v2

Η τρίτη έκδοση του εργαλείου CodeScanner σηματοδότησε μια σημαντική τεχνική βελτίωση, προσφέροντας πιο πλήρη ανάλυση μόνο του JavaScript κώδικα. Παρόλο που δεν περιλαμβάνει XML πολιτική, η v3 αποτέλεσε τη βάση για το mode "JS Scan Only" στο τελικό εργαλείο.

Κύρια χαρακτηριστικά της έκδοσης:

- **Διεύρυνση της λίστας προσωπικών δεδομένων**: Το εργαλείο πλέον αναγνωρίζει πλήθος νέων keywords όπως iban, passport, creditcard, token, tin, κ.λπ., με στόχο να καλύψει ένα πιο ρεαλιστικό φάσμα προσωπικών πληροφοριών.
- **Ανάλυση fetch blocks:** Για πρώτη φορά, η v3 μπορεί να αναλύει ολόκληρα multi-line fetch() blocks, εντοπίζοντας τη μεταφορά προσωπικών δεδομένων μέσω JSON.stringify() ή άλλων μεταβλητών.
- **Αναβαθμισμένος έλεγχος XMLHttpRequest:** Εντοπίζει .send() και ελέγχει το περιεχόμενο που μεταδίδεται, καθώς και αν χρησιμοποιείται μη ασφαλές HTTP.
- **Υποστήριξη αποθήκευσης σε IndexedDB:** Αναγνωρίζει εντολές store.put(...) που χρησιμοποιούνται σε μοντέρνες web εφαρμογές.
- **Καθαρή και σχολιασμένη αναφορά:** Κάθε εντολή συνοδεύεται από σχόλιο σχετικά με το τι κάνει και γιατί σχετίζεται με προσωπικά δεδομένα.

Η v3 ενσωματώνει λογική contextual inspection, εντοπίζοντας αν η εντολή πράγματι χειρίζεται προσωπικά δεδομένα, βελτιώνοντας έτσι δραστικά την ακρίβεια της αναφοράς και μειώνοντας τα false positives.

Η έκδοση αυτή χρησιμοποιείται ως έχει στο τελικό UI του εργαλείου, όταν ο χρήστης επιλέγει μόνο αρχείο JavaScript χωρίς XML σύγκριση.

4.4.4 Έκδοση 4 (v4) – Ανάλυση XML και Σύγκριση

Η τέταρτη έκδοση του εργαλείου CodeScanner επικεντρώθηκε στη βελτίωση της εσωτερικής αρχιτεκτονικής της αναφοράς και στην προσθήκη πιο πλούσιων, context-aware στοιχείων. Αποτελεί κρίσιμο βήμα για την τελική σύγκριση JavaScript με XML.

Βασικές βελτιώσεις:

- **Καταγραφή εντολών ανά συνάρτηση (function):** Για κάθε εντολή που περιέχει προσωπικά δεδομένα, η v4 καταγράφει σε ποια JavaScript συνάρτηση εντοπίστηκε. Αυτό επιτρέπει τη συσχέτιση των ενεργειών με το λειτουργικό τους πλαίσιο.
- **Αντιστοίχιση με επιτρεπόμενες συναρτήσεις από XML:** Αν και δεν απορρίπτονται ακόμα εντολές, γίνεται έλεγχος για το αν η τρέχουσα συνάρτηση βρίσκεται μέσα στη λίστα επιτρεπόμενων συναρτήσεων από το XML
- **Σύγκριση εντοπισμένων λέξεων-κλειδιών με XML:** Η αναφορά περιλαμβάνει ποιες λέξεις-κλειδιά εντοπίστηκαν στον JS κώδικα και δεν υπάρχουν στην πολιτική απορρήτου, και αντίστροφα.
- **Λεπτομερής αναφορά ανά λειτουργία:** Στο τέλος της αναφοράς, παρουσιάζεται λίστα με το ποια προσωπικά δεδομένα χρησιμοποιήθηκαν σε κάθε συνάρτηση.

Η έκδοση αυτή δεν περιλαμβάνει ακόμα μηχανισμό απόρριψης βάσει πολιτικής, αλλά αποτελεί τη βάση για την τελική ενσωμάτωση του XML στην επόμενη έκδοση.

4.4.5 Έκδοση 5 (v5) – JS + XML Σύγκριση και Ανάλυση Ροής Προσωπικών Δεδομένων

Η πέμπτη έκδοση του εργαλείου CodeScanner είναι η πιο πλήρης και ισχυρή υλοποίηση, η οποία υλοποιεί πλήρως το μοντέλο σύγκρισης πηγαίου κώδικα JavaScript με πολιτική απορρήτου σε XML μορφή. Η έκδοση αυτή είναι αυτή που χρησιμοποιείται στο mode “JS + XML Comparison” της τελικής διεπαφής χρήστη (UI).

Κύριες δυνατότητες:

- **Έλεγχος συμμόρφωσης με XML:** Το εργαλείο φορτώνει δηλωμένες λέξεις-κλειδιά προσωπικών δεδομένων και επιτρεπόμενες συναρτήσεις από το αρχείο XML και ελέγχει αν ο κώδικας JS συμμορφώνεται με αυτές.
- **Ανίχνευση ροής δεδομένων μεταξύ συναρτήσεων:** Η v5 παρακολουθεί αν τα προσωπικά δεδομένα που συλλέγονται σε μία συνάρτηση αποστέλλονται μέσω άλλης, είτε με άμεσες κλήσεις είτε με δικτυακές εντολές (fetch, XMLHttpRequest.send).
- **Καταγραφή εξόδου προσωπικών δεδομένων:** Ανιχνεύεται αν προσωπικά δεδομένα εμφανίζονται στην έξοδο του χρήστη μέσω alert(), console.log() ή .innerHTML.

- **Έλεγχος παραβίασης πολιτικής:** Αν κάποια συνάρτηση χειρίζεται προσωπικά δεδομένα αλλά δεν δηλώνεται ως επιτρεπόμενη στο XML, καταγράφεται ως παραβίαση.
- **Αναφορές με αναλυτικό καταμερισμό:** Η έκθεση περιλαμβάνει ξεχωριστά sections για:
    - Χρήση προσωπικών δεδομένων ανά συνάρτηση
    - Παραβιάσεις συμμόρφωσης
    - Ροή δεδομένων ανά λειτουργία
    - Εξαγωγή (output) δεδομένων

Η v5 συνδυάζει εντοπισμό, έλεγχο πολιτικής, και χαρτογράφηση ροής δεδομένων.

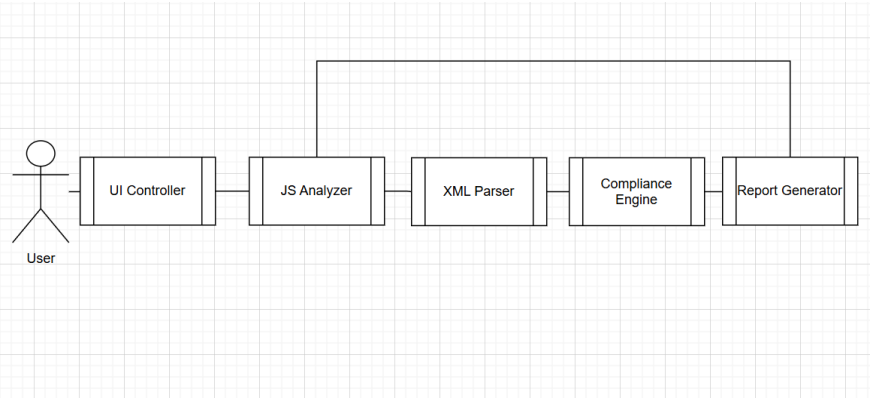
4.5 Σχεδίαση Εργαλείου

Το εργαλείο CodeScanner έχει σχεδιαστεί με modular αρχιτεκτονική ώστε να επιτρέπει την εύκολη εναλλαγή μεταξύ λειτουργιών και την επεκτασιμότητα. Η σχεδίας ασχολείται:

- **Ανάλυση JavaScript (JS Analyzer)** – Εκτελεί σάρωση του πηγαίου κώδικα JavaScript, εντοπίζοντας εντολές σχετικές με προσωπικά δεδομένα.
- **Ανάλυση XML Πολιτικής (XML Parser)** – Φορτώνει πολιτικές απορρήτου και επιτρεπόμενες λειτουργίες από αρχεία XML.
- **Μηχανή Σύγκρισης (Compliance Engine)** – Συγκρίνει τις εντολές που εντοπίστηκαν στον κώδικα με την πολιτική και εντοπίζει παραβιάσεις.
- **Μονάδα Παραγωγής Αναφοράς (Report Generator)** – Δημιουργεί αρχεία αναφοράς με τα αποτελέσματα της σάρωσης.

Διεπαφή Χρήστη (UI Controller) – Επιτρέπει στον χρήστη να επιλέξει αρχεία και mode λειτουργίας.

Το πιο κάτω διάγραμμα παρουσιάζει την αρχιτεκτονική του συστήματος.

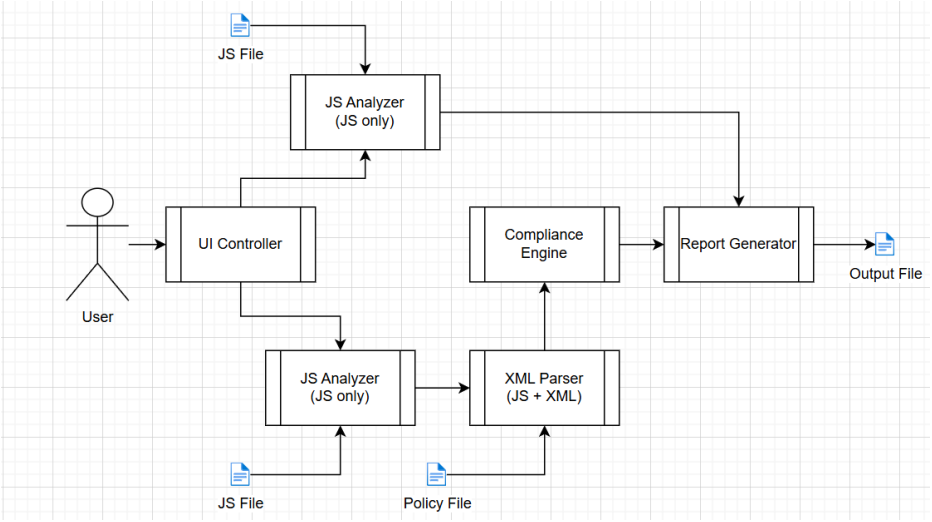


4.5.1 Λειτουργική Ροή

Η ροή του εργαλείου, από τη σκοπιά του χρήστη, έχει ως εξής:

1. Ο χρήστης επιλέγει mode: **JS Scan Only** ή **JS + XML Comparison**.
2. Κάνει upload τα απαραίτητα αρχεία: .js και προαιρετικά .xml.
3. Πατά το κουμπί **“Scan”**.
4. Το εργαλείο εκτελεί ανάλυση με βάση το mode:
    a. **JS Only**: Εκτελείται η v3.
    b. **JS + XML**: Εκτελείται η v5.
5. Το εργαλείο δημιουργεί **αναφορά (.txt)** και εμφανίζει το αποτέλεσμα στο περιβάλλον του.

Το πιο κάτω διάγραμμα δείχνει την ροή χρήσης του εργαλείου.



4.5.2 Δομή Κώδικα

Το εργαλείο έχει υλοποιηθεί πλήρως σε Java, με χρήση της βιβλιοθήκης Swing για το UI και βασικών κλάσεων για την ανάλυση αρχείων.

Κάθε έκδοση (v1–v5) έχει υλοποιηθεί ως ξεχωριστή Java κλάση, επιτρέποντας:

- Ανεξάρτητη δοκιμή και σύγκριση εκδόσεων,
- Εύκολη αλλαγή / αντικατάσταση έκδοσης στο τελικό περιβάλλον.
    - Σε περίπτωση σφάλματος κατά την εκτέλεση της ανάλυσης ή τη δημιουργία της αναφοράς, η αρχική έκδοση παραμένει ανέπαφο, εξασφαλίζοντας ότι δεν υπάρχει απώλεια δεδομένων ή αλλοίωση του περιεχομένου.

4.6 Υλοποίηση Εργαλείου

Η υλοποίηση του εργαλείου **CodeScanner** πραγματοποιήθηκε εξ ολοκλήρου σε γλώσσα Java, με στόχο την αξιοπιστία, επεκτασιμότητα και συμβατότητα με πολλαπλά περιβάλλοντα. Το εργαλείο αποτελείται από δύο βασικά τμήματα:

1. **Την εσωτερική μηχανή ανάλυσης και σύγκρισης**, που περιλαμβάνει τις εκδόσεις v1–v5
2. **Τη γραφική διεπαφή (UI)**, που προσφέρει πρόσβαση στις βασικές λειτουργίες του εργαλείου με απλό και κατανοητό τρόπο.

4.6.1 Επεξεργασία Δομής Κανόνων Ανίχνευσης

Οι εντολές που σχετίζονται με την επεξεργασία προσωπικών δεδομένων αναγνωρίζονται μέσω κανονικών εκφράσεων (regular expressions). Η μηχανή ανίχνευσης ελέγχει γραμμή-γραμμή τον κώδικα JavaScript για πρότυπα όπως:

- document.getElementById() – συλλογή δεδομένων από φόρμες,
- localStorage.setItem(), sessionStorage.setItem() – αποθήκευση,
- fetch(), XMLHttpRequest.send() – μετάδοση,
- Object.assign(), .property = – τροποποίηση,
- removeItem() – διαγραφή,
- store.put() – χρήση IndexedDB.

Αυτά τα μοτίβα είναι κοινά σε όλες τις εκδόσεις και προσαρμόζονται ανάλογα με τις ανάγκες του κάθε mode λειτουργίας.

4.6.2 Δομής Αναφοράς

Το εργαλείο παράγει αναφορά σε μορφή απλού κειμένου (.txt), στην οποία καταγράφονται:

- Οι εντολές που εντοπίστηκαν,
- Το είδος της ενέργειας (συλλογή, αποθήκευση κ.λπ.),
- Η γραμμή κώδικα,
- Η συνάρτηση στην οποία εντοπίστηκε (σε v4/v5),
- Εάν υπάρχει XML: η σύγκριση με τα δηλωμένα στοιχεία πολιτικής.

4.6.2.1 Η Τυπική Γλώσσα που Χρησιμοποιείται

Για τις XML πολιτικές, χρησιμοποιείται μορφή mxGraph όπου:

- Οι λέξεις-κλειδιά εντοπίζονται μέσω του attribute ‘style’ με File.svg,

- Οι επιτρεπόμενες συναρτήσεις δηλώνονται με UML Lifeline node style.

Οι πληροφορίες αυτές εξάγονται μέσω DOM XML Parser της Java και εισάγονται σε κατάλληλες δομές (Set<String>, Map<String, Set<String>>).

4.6.2.2 Διαδικασία Μετατροπής

Η αναφορά παράγεται ως εξής:

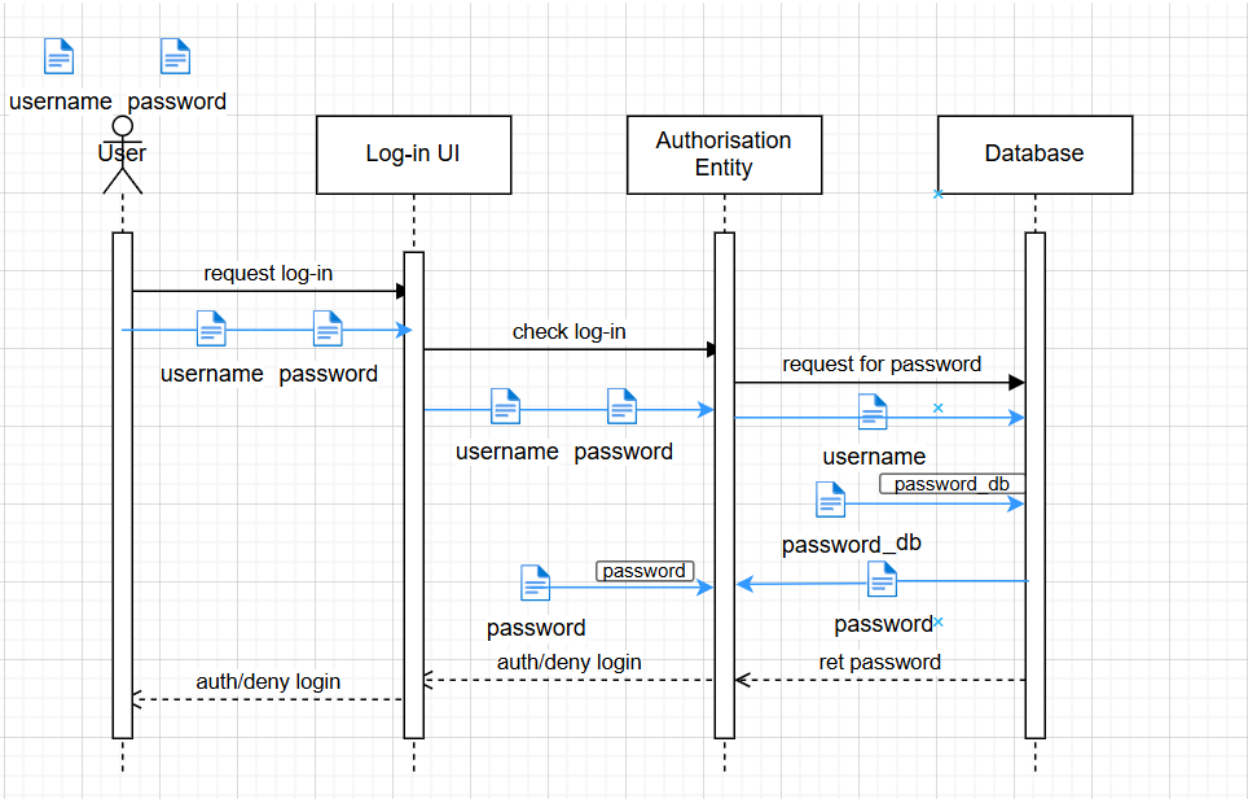
1. Ανάλυση JS κώδικα και (προαιρετικά) XML πολιτικής.
2. Καταγραφή εντολών και σύνδεσή τους με λέξεις-κλειδιά προσωπικών δεδομένων.
3. Συγκρίσεις ανά συνάρτηση και έλεγχος συμμόρφωσης (v5).
4. Παραγωγή αρχείου .txt με την τελική αναφορά.

Η έξοδος προορίζεται τόσο για τον προγραμματιστή όσο και για μη τεχνικούς χρήστες που επιθυμούν να ελέγξουν αν η εφαρμογή τους είναι σύμφωνη με τις δηλωμένες πολιτικές απορρήτου.

4.7 Δοκιμή εργαλείου

Για την επιβεβαίωση της ορθής λειτουργίας του εργαλείου CodeScanner, δημιουργήθηκε ένα σύνολο σελίδων JavaScript οι οποίες περιέχουν σενάρια πραγματικής χρήσης προσωπικών δεδομένων. Τα σενάρια αυτά σχεδιάστηκαν ώστε να καλύπτουν όλες τις κατηγορίες ενεργειών (συλλογή, αποθήκευση, αποστολή, τροποποίηση, διαγραφή, συνδυασμό), με σκοπό την αξιολόγηση της ικανότητας του εργαλείου να εντοπίζει και να κατηγοριοποιεί σωστά τις σχετικές εντολές.

Παράλληλα, δημιουργήθηκαν sequence diagrams που απεικονίζουν τη ροή προσωπικών δεδομένων μεταξύ των βασικών οντοτήτων ενός συστήματος.



Εικόνα 4.7.1 – Διάγραμμα ακολουθίας (Sequence Diagram) για το σενάριο αυθεντικοποίησης χρήστη. Περιγράφεται η ροή των προσωπικών δεδομένων (όνομα χρήστη και κωδικός πρόσβασης) από τη διεπαφή χρήστη προς τη βάση δεδομένων, και αποτελεί βασικό σενάριο ελέγχου για το εργαλείο CodeScanner.

Κεφάλαιο 5

Επίδειξη Εργαλείου

5.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται η πρακτική λειτουργία του εργαλείου CodeScanner, το οποίο αναπτύχθηκε με στόχο τον αυτοματοποιημένο έλεγχο πηγαίου κώδικα διαδικτυακών πλατφορμών ως προς τη χρήση προσωπικών δεδομένων και τη συμμόρφωση με πολιτικές απορρήτου.

Η επίδειξη καλύπτει:

- την περιγραφή της γραφικής διεπαφής χρήστη (UI),
- την παρουσίαση των βασικών λειτουργιών μέσω παραδειγμάτων,
- και την ανάλυση του τρόπου με τον οποίο παράγονται τα αποτελέσματα και οι αναφορές.

Το εργαλείο προσφέρει δύο κύριες λειτουργίες:

- **JS Scan Only:** Εκτέλεση ανάλυσης αποκλειστικά σε αρχεία JavaScript, χωρίς πολιτική XML.
- **JS + XML Comparison:** Ανάλυση JavaScript σε συνδυασμό με πολιτική απορρήτου σε XML μορφή, ελέγχοντας τη συμμόρφωση με δηλωμένα πρότυπα χειρισμού προσωπικών δεδομένων (processing purposes).

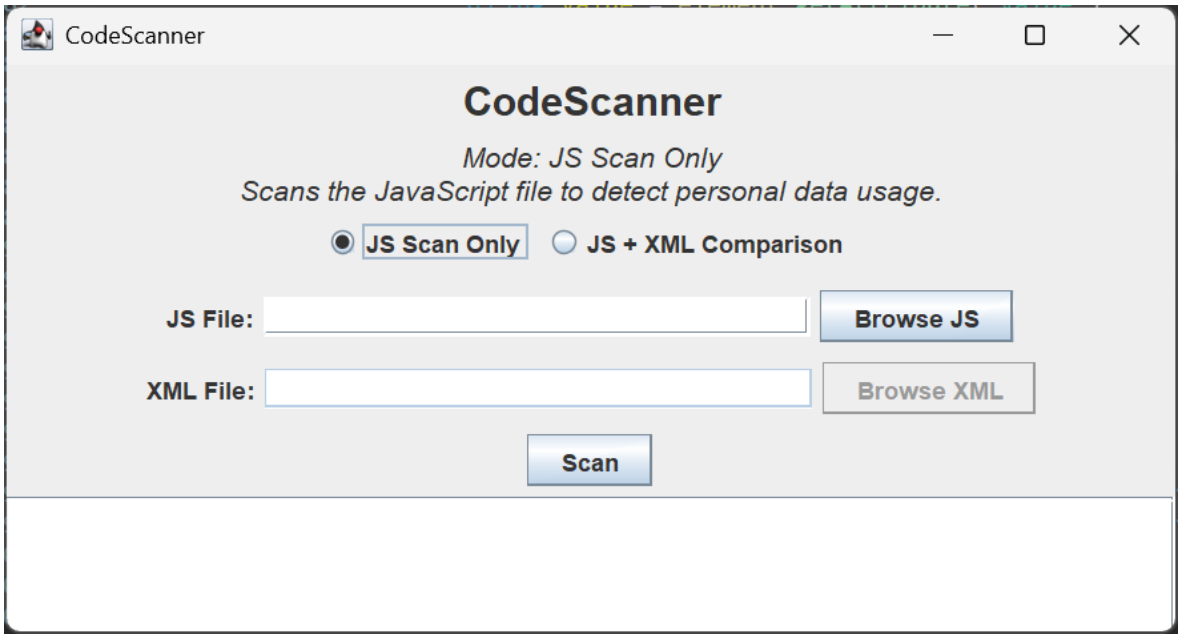
Στα επόμενα τμήματα παρουσιάζονται παραδείγματα χρήσης κάθε λειτουργίας, συνοδευόμενα από στιγμιότυπα οθόνης και παραγόμενες αναφορές.

5.2 Λειτουργία της γραφικής διεπαφής (UI)

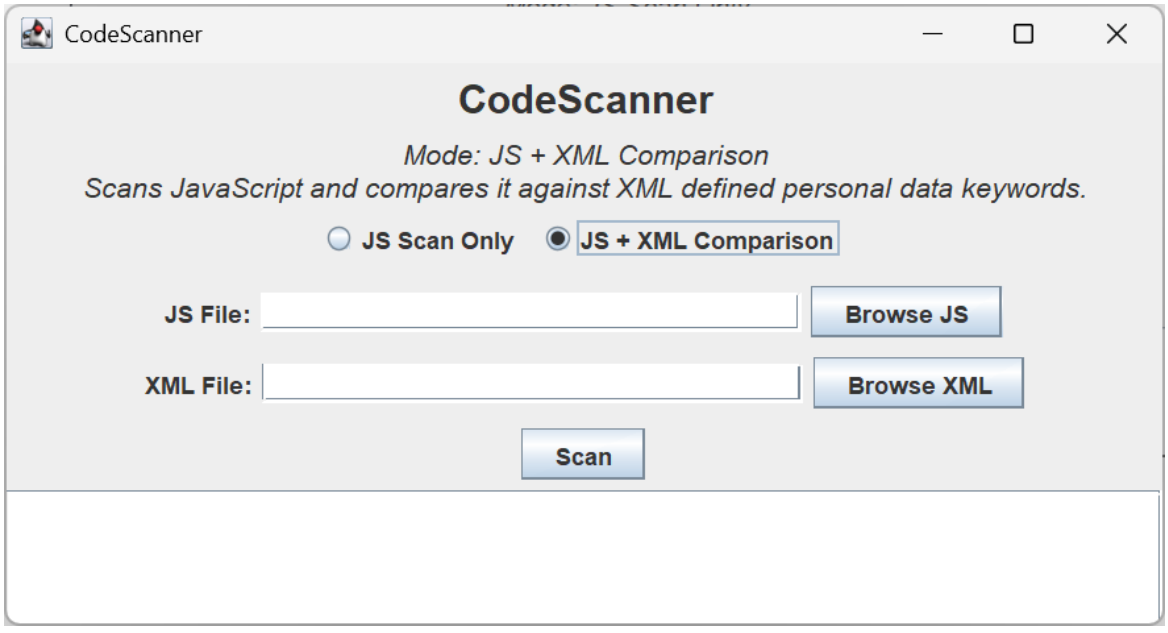
Το περιβάλλον χρήστη έχει σχεδιαστεί ώστε να είναι απλό και λειτουργικό, με τη χρήση της βιβλιοθήκης Java Swing. Το βασικό παράθυρο του εργαλείου περιλαμβάνει:

- Πεδίο επιλογής αρχείου .js
- Προαιρετικό πεδίο επιλογής αρχείου .xml

- Επιλογή mode (JS ή JS + XML)
- Κουμπί “Scan”
- Περιοχή προβολής αποτελεσμάτων (ή αποθήκευση σε .txt αρχείο)



Εικόνα 5.2.1- Επιλογή αρχείου Javascript για ανάλυση με την λειτουργία “JS Scan Only”

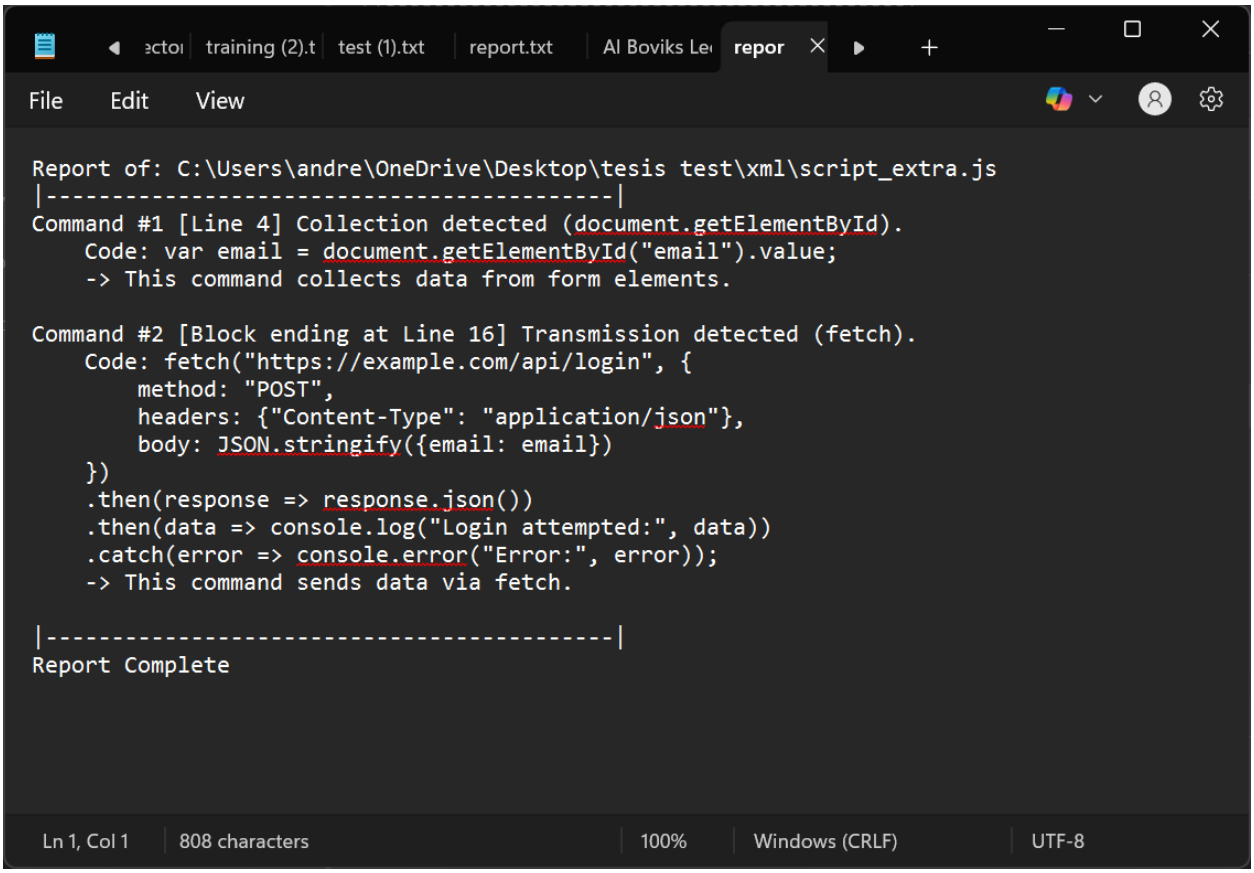


Εικόνα 5.2.2 - Προσθήκη αρχείων Javascript και XML για ανάλυση μέσω της λειτουργίας “JS + XML Comparison”

5.3 Παράδειγμα Χρήσης – JS Scan Only

Σε αυτό το σενάριο, ο χρήστης επιλέγει μόνο ένα αρχείο JavaScript και εκτελεί ανάλυση χωρίς XML πολιτική.

- Το εργαλείο σαρώνει τον κώδικα και εντοπίζει εντολές όπως localStorage.setItem(), fetch(), document.getElementById() κ.λπ.
- Οι εντολές φιλτράρονται βάσει keywords όπως email, ip, password (όπως ορίστηκαν στη v2).
- Η αναφορά δημιουργείται με όλες τις εντολές που χειρίζονται προσωπικά δεδομένα όπως αυτές έχουν καταγραφεί στην προηγούμενη έρευνα μου.

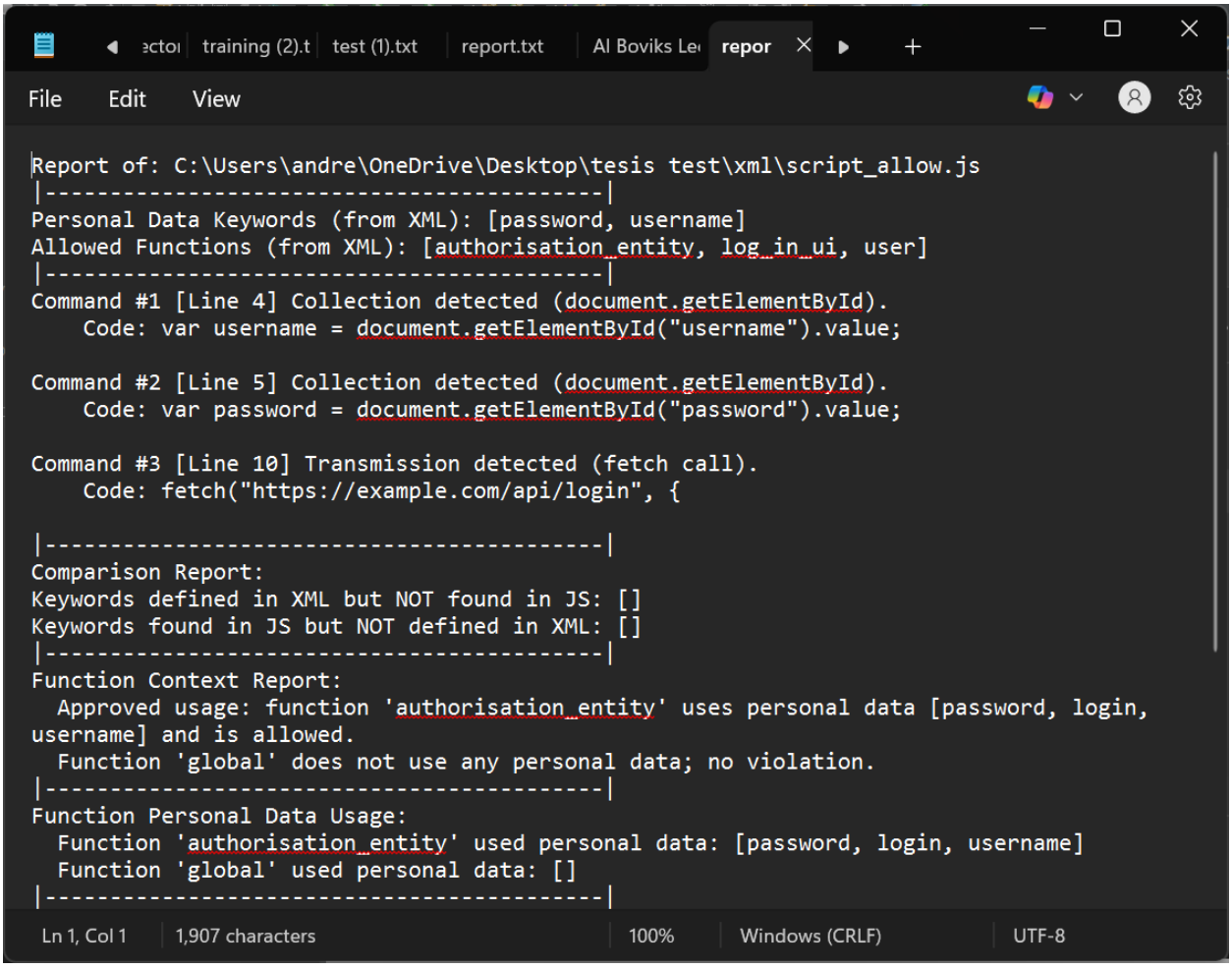


Εικόνα 5.3 - Αποτελέσματα ανάλυσης Javascript με τη λειτουργία “JS Scan Only”. Το εργαλείο εντοπίζει εντολές που σχετίζονται με προσωπικά δεδομένα και τις καταγράφει σε αναφορά

5.4 Παράδειγμα Χρήσης – JS + XML Comparison (v5)

Σε αυτό το σενάριο, ο χρήστης εισάγει τόσο .js όσο και .xml αρχείο.

- Το εργαλείο εντοπίζει προσωπικά δεδομένα και τις συναρτήσεις (functions) στις οποίες χρησιμοποιούνται.
- Γίνεται σύγκριση με τα επιτρεπόμενα από την XML (π.χ. ποια δεδομένα επιτρέπεται να συλλέγονται και σε ποιες συναρτήσεις).
- Η αναφορά εμφανίζει:
    - Συμμόρφωση ή παραβίαση για κάθε συνάρτηση
    - Μεταφορά δεδομένων μέσω fetch/XHR
    - Παρουσίαση δεδομένων μέσω output (innerHTML, alert, console.log)



Εικόνα 5.4 - Αποτέλεσμα ανάλυσης Javascript με σύγκριση έναντι XML πολιτικής(JS + XML comparison). Το εργαλείο εντοπίζει παραβιάσεις και αναφέρει τη χρήση προσωπικών δεδομένων εκτός επιτρεπόμενων συναρτήσεων

5.5 Παραγόμενη Αναφορά

Η έξοδος του εργαλείου παράγεται σε αρχείο .txt και μπορεί να περιλαμβάνει:

- Περιγραφή εντοπισμένων ενεργειών (π.χ. συλλογή, αποθήκευση, μετάδοση)
- Αριθμημένη καταγραφή εντολών με γραμμή και τύπο
- Αναφορές ανά συνάρτηση
- Σύγκριση XML-κώδικα και παραβιάσεις πολιτικής
- Αναφορά αποστολής και εμφάνισης προσωπικών δεδομένων

Κεφάλαιο 6

Αξιολόγηση του Εργαλείου CodeScanner

6.1 Σκοπός Αξιολόγησης

Η αξιολόγηση του εργαλείου CodeScanner αποσκοπεί στην αποτύπωση της εμπειρίας χρήσης του από την πλευρά επαγγελματιών και φοιτητών στον τομέα της μηχανικής λογισμικού. Ειδικότερα, εξετάζεται ο βαθμός στον οποίο το εργαλείο θεωρείται εύχρηστο, κατανοητό και αποτελεσματικό ως προς την παρουσίαση των αποτελεσμάτων του, καθώς και η γενική ικανοποίηση των χρηστών από τη λειτουργικότητα και τη σχεδίασή του. Η αξιολόγηση επικεντρώνεται σε αντιληπτές ποιότητες, όπως η σαφήνεια, η καινοτομία, η ασφάλεια και η συνολική εμπειρία χρήσης, χωρίς να απαιτείται τεχνική επαλήθευση των εσωτερικών λειτουργιών του εργαλείου.

6.2 Σχεδιασμός και Μεθοδολογία

Η διαδικασία αξιολόγησης έχει σχεδιαστεί σε τρία βασικά στάδια:

6.2.1 Παρουσίαση του Εργαλείου

Πριν την απάντηση στο ερωτηματολόγιο, οι συμμετέχοντες θα παρακολουθήσουν ένα σύντομο βίντεο που παρουσιάζει τις βασικές λειτουργίες του εργαλείου CodeScanner, με ενδεικτική ανάλυση ενός παραδείγματος κώδικα. Σκοπός είναι να εξοικειωθούν με τον τρόπο λειτουργίας και τη χρησιμότητα του εργαλείου.

Το σχετικό βίντεο είναι διαθέσιμο στον ακόλουθο σύνδεσμο: https://drive.google.com/file/d/1KjBjdCwlMX8_44IGLFs8OjtwuJ3IMoNY/view?usp=drive_link

6.2.2 Συλλογή Πληροφοριών για το Προφίλ του Χρήστη

Οι συμμετέχοντες θα κληθούν να απαντήσουν σε βασικές ερωτήσεις δημογραφικού και επαγγελματικού περιεχομένου:

Έτη εμπειρίας στον τομέα της Μηχανικής Λογισμικού (0–5, 5–10, >10)

Εξοικείωση με τον GDPR και τις πρακτικές διαχείρισης προσωπικών δεδομένων (Very familiar, Somewhat familiar, Not familiar)

6.2.3 Κύρια Ενότητα Αξιολόγησης

Η βασική ενότητα του ερωτηματολογίου περιλαμβάνει 26 ερωτήσεις τύπου semantic differential (διπολικά χαρακτηριστικά), βασισμένες στο πρότυπο User Experience Questionnaire (UEQ), και χρησιμοποιεί κλίμακα 1-5 αξιολόγησης. Οι άξονες αξιολόγησης καλύπτουν πτυχές που σχετίζονται με την εμπειρία χρήσης, όπως:

Εμπειρία Χρήσης	annoying – enjoyable, boring – exciting
Κατανόηση	not understandable – understandable, clear – confusing
Ασφάλεια & Απόδοση	secure – not secure, efficient – inefficient
Οπτική και Δομή	organized – cluttered, attractive – unattractive
Υποστήριξη & Καινοτομία	supportive – obstructive, innovative – conservative

Οι συμμετέχοντες καλούνται να επιλέξουν για κάθε ζεύγος χαρακτηριστικών ένα σημείο στην κλίμακα 1-5, αποτυπώνοντας έτσι την προσωπική τους αντίληψη για την εμπειρία χρήσης του εργαλείου.

Το ερωτηματολόγιο είναι διαθέσιμο στον παρακάτω σύνδεσμο: https://docs.google.com/forms/d/e/1FAIpQLSdtEE80UXPtIGGSUWrs6o1MbClxPnjriwq3dYSPBUkXtw33bg/viewform?usp=dialog

Ολόκληρη η έρευνα είναι διαθέσιμη και στο Παράρτημα Α.

6.3 Χρήση και Ανάλυση των Δεδομένων

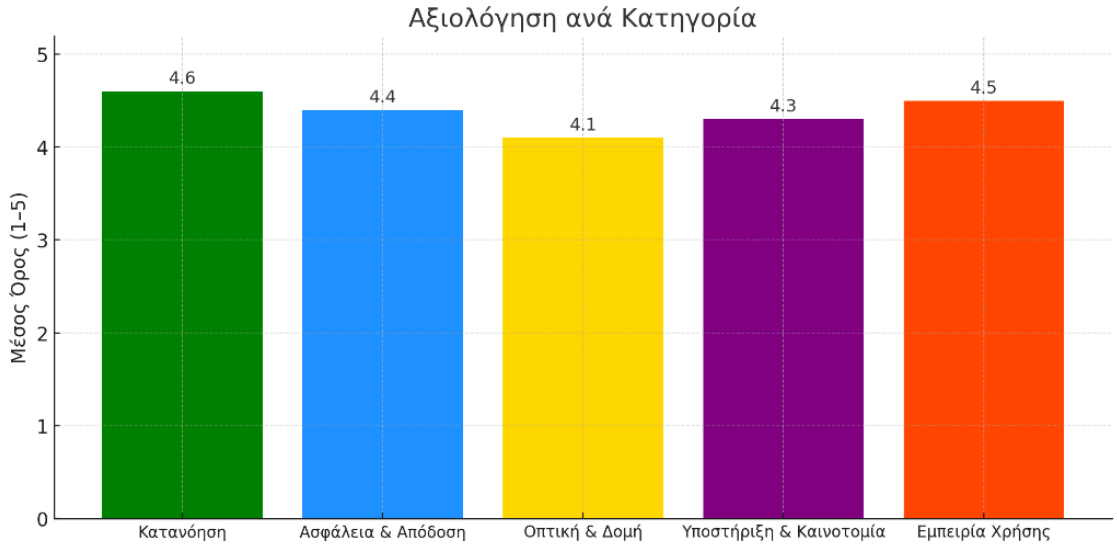
Τα αποτελέσματα συλλέχθηκαν ανώνυμα μέσω Google Forms και αναλύθηκαν ποσοτικά με στόχο την εξαγωγή στατιστικών αποτελεσμάτων, όπως ο μέσος όρος, η τυπική απόκλιση και ποσοστά συμφωνίας. Παράλληλα διερευνήθηκαν πιθανά μοτίβα ή αποκλίσεις που σχετίζονται με την εμπειρία ή το υπόβαθρο των χρηστών.

6.4 Αποτελέσματα

Συνολικά, στην αξιολόγηση του εργαλείου CodeScanner συμμετείχαν 30 άτομα, προερχόμενα από διαφορετικά επίπεδα εμπειρίας στον τομέα της μηχανικής λογισμικού και με ποικίλο βαθμό εξοικείωσης με τον Γενικό Κανονισμό για την Προστασία Δεδομένων (GDPR). Η πλειοψηφία των συμμετεχόντων (άνω του 80%) είχε επαγγελματική εμπειρία 0–5 ετών, ενώ πάνω από τα 2/3 των συμμετεχόντων δήλωσαν ότι είναι είτε “Very familiar” είτε “Somewhat familiar” με τις αρχές, τις απαιτήσεις και τις πρακτικές του GDPR.

Η αξιολόγηση βασίστηκε σε 26 άξονες τύπου semantic differential, σύμφωνα με το πρότυπο User Experience Questionnaire (UEQ).

Οι απαντήσεις δόθηκαν σε κλίμακα 1–5, όπου το 1 αντιστοιχεί σε πολύ αρνητική αξιολόγηση και το 5 σε πολύ θετική. Για λόγους ευκολότερης οπτικοποίησης και ερμηνείας των δεδομένων, οι απαντήσεις στους άξονες του ερωτηματολογίου μετατράπηκαν ώστε το θετικό άκρο της κλίμακας να αντιστοιχεί ομοιόμορφα στο 5, διευκολύνοντας τη σύγκριση και την απεικόνιση στα γραφήματα.

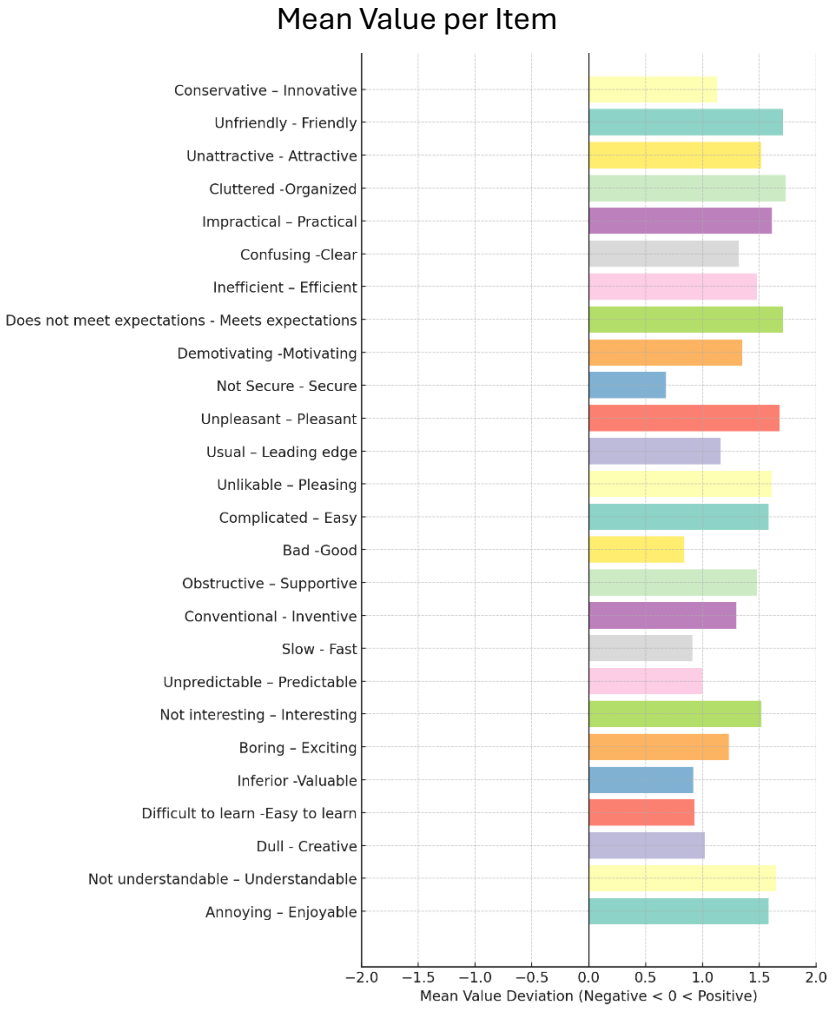


Εικόνα 6.4.1 – Μέση Τιμή Βαθμολογίας ανά Θεματική Κατηγορία

Η Εικόνα 6.4.1 παρουσιάζει τη μέση τιμή ανά θεματική κατηγορία αξιολόγησης. Όλες οι κατηγορίες κατέγραψαν μέσο όρο άνω του 4.0, γεγονός που καταδεικνύει καθολικά θετική εμπειρία χρήσης του εργαλείου:

- Κατανόηση: 4.6/5
- Εμπειρία Χρήσης: 4.5/5
- Ασφάλεια & Απόδοση: 4.4/5
- Υποστήριξη & Καινοτομία: 4.3/5
- Οπτική & Δομή: 4.1/5

Η υψηλή επίδοση στην κατανόηση αναδεικνύει τη σαφήνεια του εργαλείου ως προς την απεικόνιση εντολών επεξεργασίας δεδομένων, ενώ η εμπειρία χρήσης και η αποδοτικότητα ενισχύουν την εικόνα ενός λειτουργικά επαρκούς, φιλικού προς τον χρήστη συστήματος.

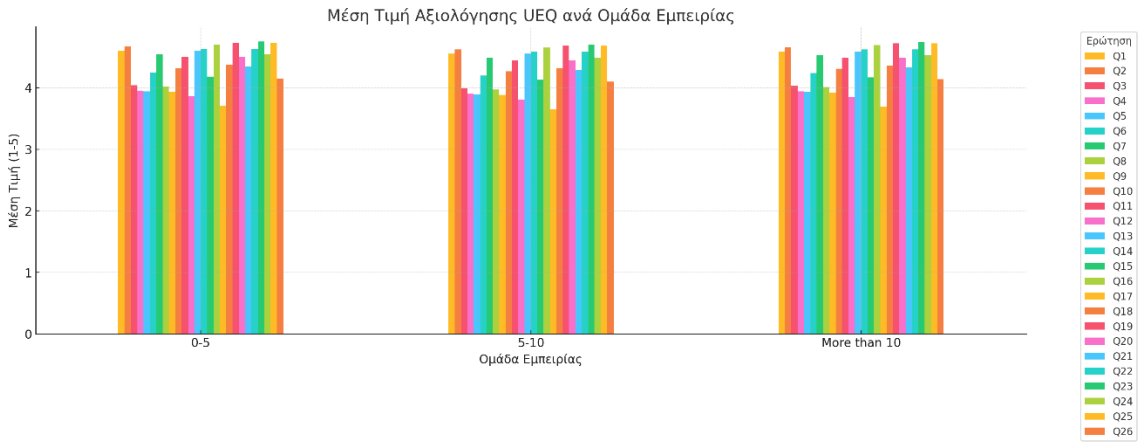


Εικόνα 6.4.2 – Μέση Απόκλιση Αξιολόγησης ανά Ερώτηση (Semantic Axes)

Η Εικόνα 6.4.2 αποτυπώνει την αναλυτική μέση τιμή για κάθε μία από τις 26 ερωτήσεις του ερωτηματολογίου. Παρατηρήθηκε ιδιαίτερα υψηλή αποδοχή σε ερωτήσεις που σχετίζονται με:

- Κατανόηση και σαφήνεια: (“Clear”, “Understandable”)
- Απόδοση: (“Efficient”)
- Υποστήριξη: (“Supportive”)
- Θετική εμπειρία: (“Enjoyable”, “Pleasing”, “Exciting”)

Οι περισσότερες ερωτήσεις σημείωσαν μέσο όρο πάνω από 4.0, με μικρές μόνο διακυμάνσεις στις διαστάσεις “Secure” και “Meets Expectations”, οι οποίες ωστόσο διατηρούνται σε θετικό πλαίσιο (>3.8).



Εικόνα 6.4.3 – Μέση Τιμή Αξιολόγησης UEQ ανά Ομάδα Εμπειρίας

Η τρίτη γραφική αποτυπώνει τη μέση τιμή απαντήσεων για κάθε ερώτηση του ερωτηματολογίου, ομαδοποιημένες σύμφωνα με την επαγγελματική εμπειρία των συμμετεχόντων (0–5, 5–10 και >10 έτη). Παρατηρείται σημαντική ομοιογένεια ανάμεσα στις τρεις ομάδες, με τη συντριπτική πλειοψηφία των μέσων τιμών να κινείται πάνω από το 4.0. Το εύρημα αυτό καταδεικνύει ότι το εργαλείο παραμένει κατανοητό, αποτελεσματικό και φιλικό προς χρήστες διαφορετικών επιπέδων εμπειρίας.

Συμπερασματικά, η αξιολόγηση αναδεικνύει τη θετική αποδοχή του CodeScanner και επιβεβαιώνει τη χρηστικότητά του ως εργαλείο ανάλυσης πηγαίου κώδικα για σκοπούς ελέγχου επεξεργασίας προσωπικών δεδομένων.
Τα αναλυτικά στατιστικά στοιχεία, οι συγκεντρωτικοί πίνακες και οι απαντήσεις περιλαμβάνονται στο Παράρτημα Α.

Κεφάλαιο 7

Συμπεράσματα

7.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία είχε ως βασικό στόχο την ανάπτυξη του εργαλείου **CodeScanner**, ενός εργαλείου στατικής ανάλυσης πηγαίου κώδικα που επικεντρώνεται στη συμμόρφωση των διαδικτυακών εφαρμογών με τον Γενικό Κανονισμό Προστασίας Δεδομένων (GDPR). Το εργαλείο δίνει τη δυνατότητα στους μηχανικούς λογισμικού να ελέγχουν εάν ο πηγαίος κώδικας ενός συστήματος υλοποιεί ή παραβιάζει τις δηλωμένες πολιτικές απορρήτου, συνδέοντας λειτουργικότητες του λογισμικού με τις σχετικές νομικές απαιτήσεις.

Μέσα από τη δημιουργία διαδοχικών εκδόσεων και την αξιολόγησή τους, επιτεύχθηκε μια σταδιακή βελτίωση της ακρίβειας, της λειτουργικότητας και της ευχρηστίας του εργαλείου. Η χρήση στατικής ανάλυσης με συνδυασμό JavaScript parsing και XML πολιτικών επέτρεψε την αυτόματη ανίχνευση βασικών ενεργειών όπως συλλογή, αποθήκευση, μετάδοση και διαγραφή προσωπικών δεδομένων. Η ευχρηστία του εργαλείου ενισχύθηκε μέσω της γραμμής εντολών (CLI), των αναφορών που παράγονται με απλή μορφή και της δυνατότητας ανάλυσης πραγματικών σεναρίων χρήσης.

Η έρευνα ανέδειξε τη σημασία της πρώιμης ενσωμάτωσης πολιτικών ιδιωτικότητας στη φάση σχεδιασμού των εφαρμογών. Παράλληλα, επιβεβαιώθηκε η πρακτικότητα της χαρτογράφησης προσωπικών δεδομένων και η σύνδεσή τους με εντολές σε διάφορες γλώσσες προγραμματισμού. Η δομή του εργαλείου, βασισμένη σε modular σχεδίαση, προσφέρει ευελιξία και επεκτασιμότητα για μελλοντικές ανάγκες.

7.2 Μελλοντική Δουλειά

Παρότι το εργαλείο είναι λειτουργικό και έχει αξιολογηθεί σε πραγματικά σενάρια, υπάρχουν αρκετές δυνατότητες επέκτασης και βελτίωσης:

- **Επέκταση σε άλλες γλώσσες προγραμματισμού:** Η αρχική έρευνα περιλάμβανε την καταγραφή εντολών διαχείρισης προσωπικών δεδομένων σε Java, PHP, Python και C++. Το CodeScanner έχει σχεδιαστεί σε modules, γεγονός που επιτρέπει

με ελάχιστες τροποποιήσεις την υποστήριξη των παραπάνω γλωσσών, διατηρώντας την ίδια λογική και αναφοράς.

- **Επεκτασιμότητα λίστας προσωπικών δεδομένων:** Η λίστα με τις λέξεις-κλειδιά που αναγνωρίζονται ως προσωπικά δεδομένα είναι ενδεικτική και όχι περιοριστική. Μπορεί να επεκταθεί ή να προσαρμοστεί ανά πάσα στιγμή, χωρίς να διαταραχθεί η λειτουργία του εργαλείου.
- **Υποστήριξη GUI περιβάλλοντος:** Μία μελλοντική έκδοση του CodeScanner θα μπορούσε να περιλαμβάνει διαδραστική γραφική διεπαφή για ευκολότερη χρήση από μη προγραμματιστές.
- **Σύνδεση με εξωτερικές πλατφόρμες ανάλυσης** και βάσεις δεδομένων για αποθήκευση ιστορικού ελέγχων, ώστε να παρακολουθείται η εξέλιξη συμμόρφωσης ενός project στον χρόνο.
- **Ενσωμάτωση τεχνητής νοημοσύνης**, για εντοπισμό υποψιών παραβίασης ιδιωτηκότητας με βάση προγνωστικά μοντέλα και ιστορικά δεδομένα.

Το CodeScanner μπορεί να αποτελέσει τη βάση για ένα πιο ολοκληρωμένο σύστημα υποστήριξης GDPR συμμόρφωσης, ικανό να χρησιμοποιηθεί τόσο από ομάδες ανάπτυξης όσο και από φορείς ελέγχου.

Βιβλιογραφία

[1] Lim, M., & Kim, T. (2024). *Efficient Static Vulnerability Analysis for JavaScript with Multiversion Dependency Graphs*. https://dl.acm.org/doi/10.1145/3656394

[2] Momeni, E., Karim, M. R., & Ahamed, S. I. (2024). *Toward an Android Static Analysis Approach for Data Protection*. https://dl.acm.org/doi/abs/10.1145/3647632.3651389

[3] Khalid, H., Ali, W., & Hussain, F. (2020). *GDPR Compliance in the Context of Continuous Integration*. https://arxiv.org/abs/2002.06830

[4] Wang, Z., & Liu, Y. (2022). *Abstract Interpretation-Based Data Leakage Static Analysis*. https://arxiv.org/abs/2211.16073

[5] Hjerppe, K., Ruohonen, J., & Leppänen, V. (2020). *Annotation-Based Static Analysis for Personal Data Protection*. https://arxiv.org/abs/2003.09890

[6] Tang, F., Østvold, B. M., & Bruntink, M. (2023). *Helping Code Reviewer Prioritize: Pinpointing Personal Data and its Processing*. https://arxiv.org/abs/2306.11495

[7] GDPR Hub. (n.d.). Άρθρο 4 – Ορισμοί (Article 4 – Definitions). Διαθέσιμο στο: https://gdprhub.eu/Article_4_GDPR#Decisions

[8] Talha, M., Rehman, A., & Hussain, M. (2021). A comparative study of static code analysis tools for vulnerability detection in C/C++ and Java source code. *Procedia Computer Science*. https://www.sciencedirect.com/science/article/pii/S1877050920312023

[9] Tang, S., Wang, Q., & Wang, Y. (2022). An XML privacy-preserving data disclosure decision scheme. *Security and Privacy, 2022*, Article ID 9099722. https://onlinelibrary.wiley.com/doi/full/10.1155/2022/9099722

[10] Malan, D. J., & Leitner, H. H. (2005). Experiences with Eclipse IDE in programming courses. *Proceedings of the 2005 ACM SIGCSE*, 44–48. https://doi.org/10.1145/1047344.1047382

[11] Chen, E., Huang, R., Chen, H.-S., Tseng, Y.-H., & Li, L.-Y. (2023). *GPTutor: A ChatGPT-powered programming tool for code explanation*. arXiv preprint arXiv:2305.01863. https://arxiv.org/abs/2305.01863

[12] Larman, C., & Basili, V. R. (2003). *Iterative and incremental development: A brief history*. Computer, 36(6), 47-56. https://ieeexplore.ieee.org/document/1204375

[13] Statista Research Department, "Most used programming languages among developers worldwide, as of 2023," *Statista*, Apr. 2024. [Online]. https://www.statista.com/statistics/793628/worldwide-developer-survey-most-used-languages/

[14] Vanezi, E., Kapitsaki, G. M., & Philippou, A. (2024). What's Your Purpose? An Approach to Incorporating GDPR Purposes into Requirements Analysis. In *ICISSP* (pp. 907-914). https://www.scitepress.org/Papers/2024/124744/124744.pdf

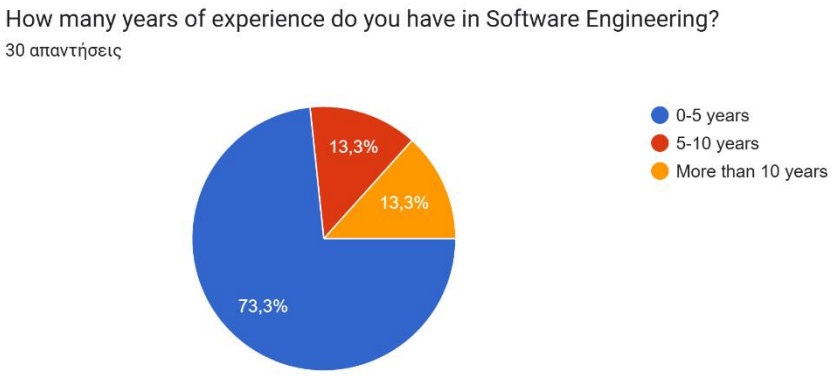
Παράρτημα Α – Αποτελέσματα Ερωτηματολογίου Αξιολόγησης

Το παρόν παράρτημα περιλαμβάνει τα συγκεντρωτικά αποτελέσματα της αξιολόγησης του εργαλείου **CodeScanner**, όπως αυτά συλλέχθηκαν από **30 συμμετέχοντες** μέσω της πλατφόρμας **Google Forms**. Οι απαντήσεις καλύπτουν:

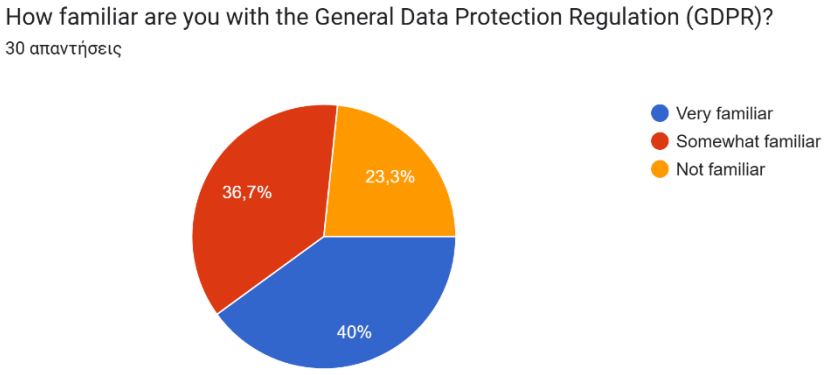
- βασικά χαρακτηριστικά του επαγγελματικού προφίλ των συμμετεχόντων (π.χ. έτη εμπειρίας, εξοικείωση με τον GDPR),
- και αξιολόγηση του εργαλείου βάσει **26 διπολικών αξόνων** (semantic differential), σχετικών με ευχρηστία, σαφήνεια, καινοτομία, ασφάλεια, απόδοση και σχεδιασμό.(se afto to simio na valw anaphora sto ueq)

Ακολουθεί το πλήρες ερωτηματολόγιο που χρησιμοποιήθηκε στην έρευνα:

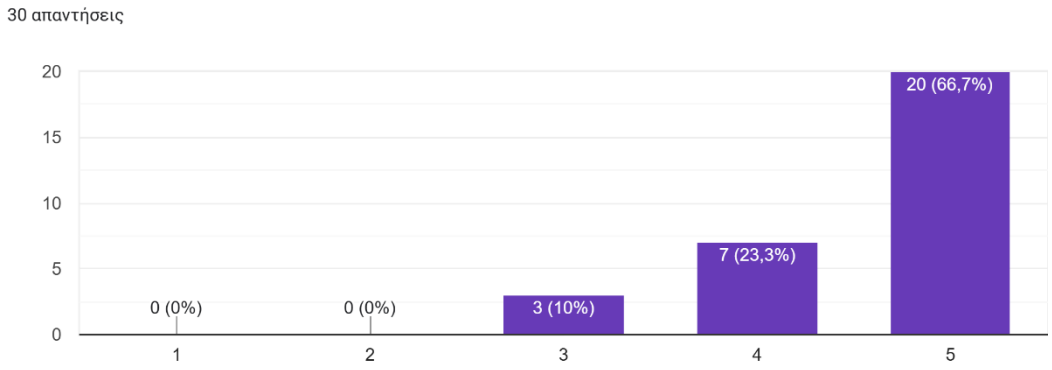
Εικόνα Α.1 – Κατανομή εμπειρίας συμμετεχόντων



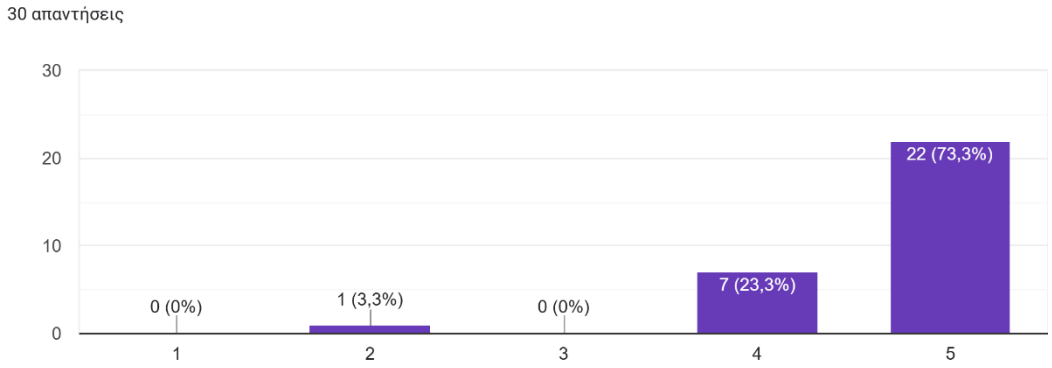
Εικόνα Α.2 – Βαθμός εξοικείωσης με τον GDPR



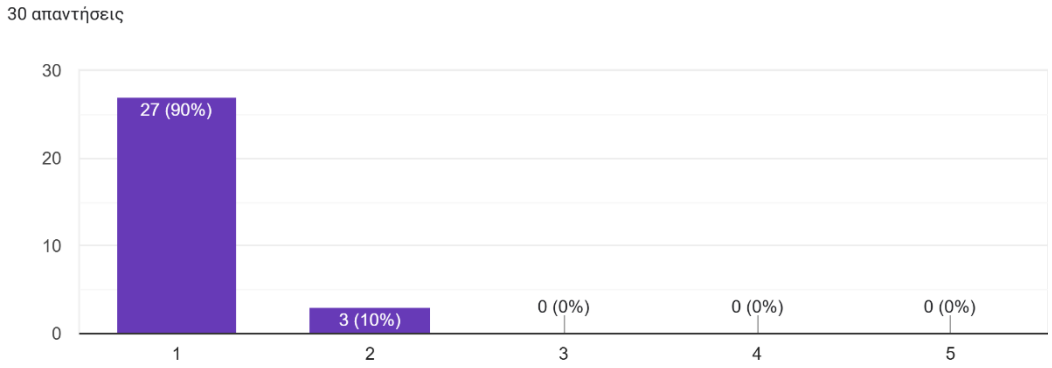
Εικόνα Β.1 – Annoying(1) – Enjoyable(5)



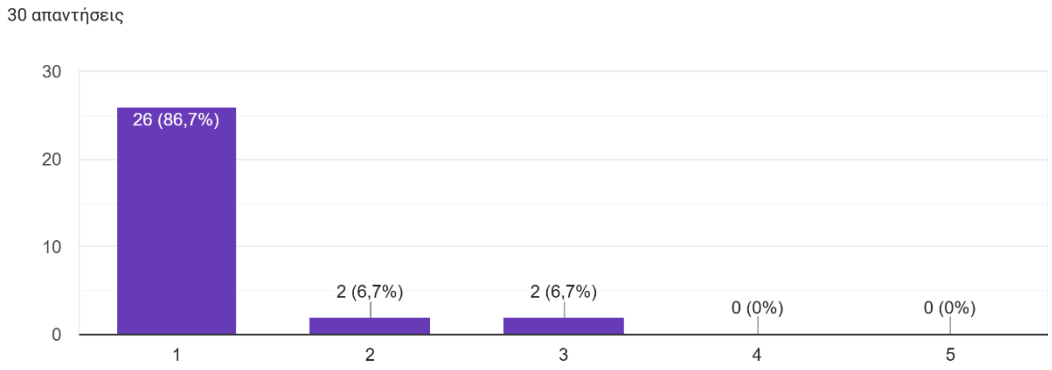
Εικόνα Β.2 – Not understandable(1) – Understandable(5)



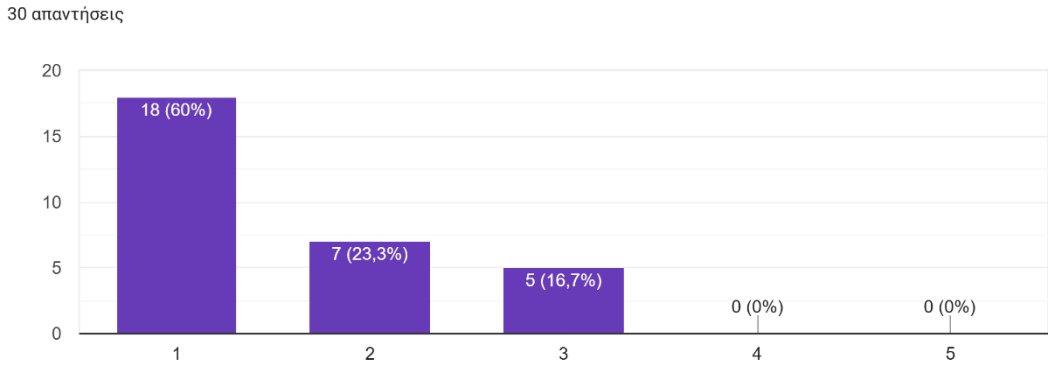
Εικόνα Β.3 – Creative(1) – Dull(5)



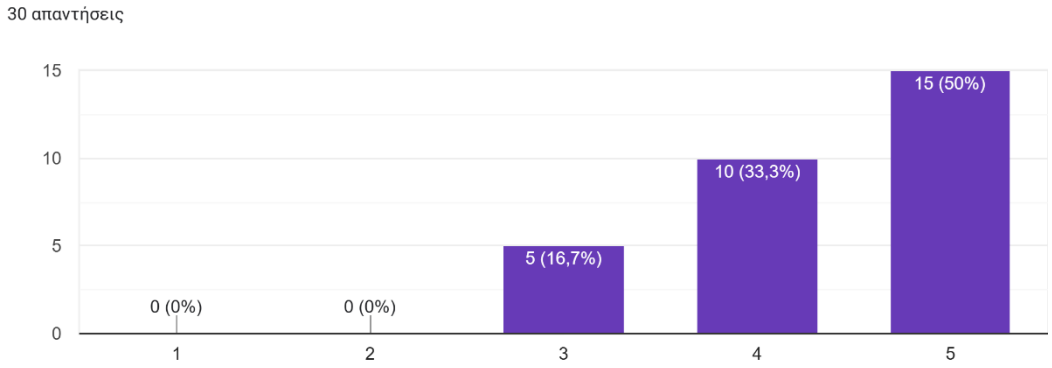
Εικόνα Β.4 – Easy to learn(1) – Difficult to learn(5)



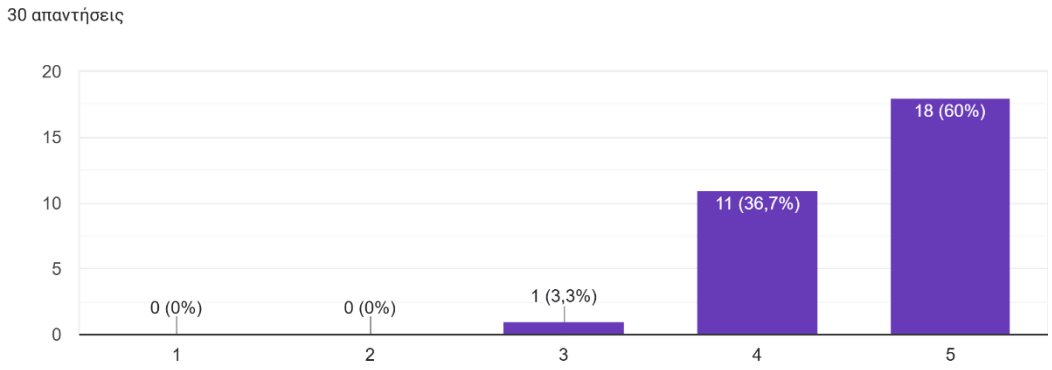
Εικόνα Β.5 – Valuable(1) – Inferior(5)



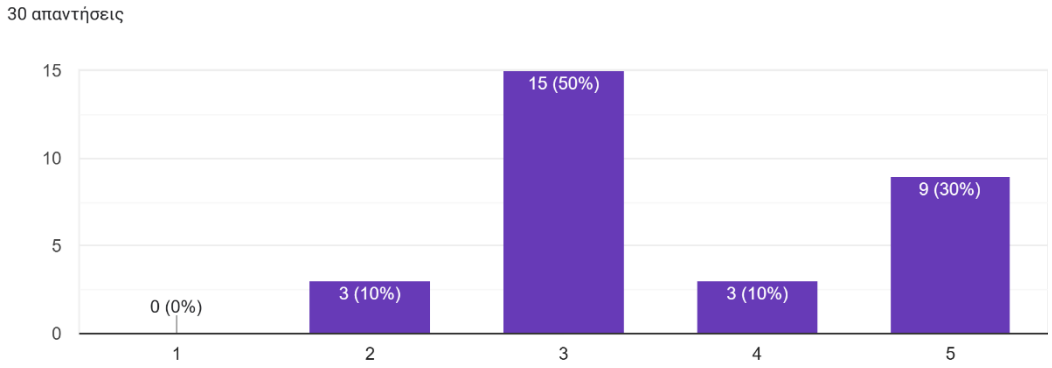
Εικόνα Β.6 – Boring(1) – Exciting(5)



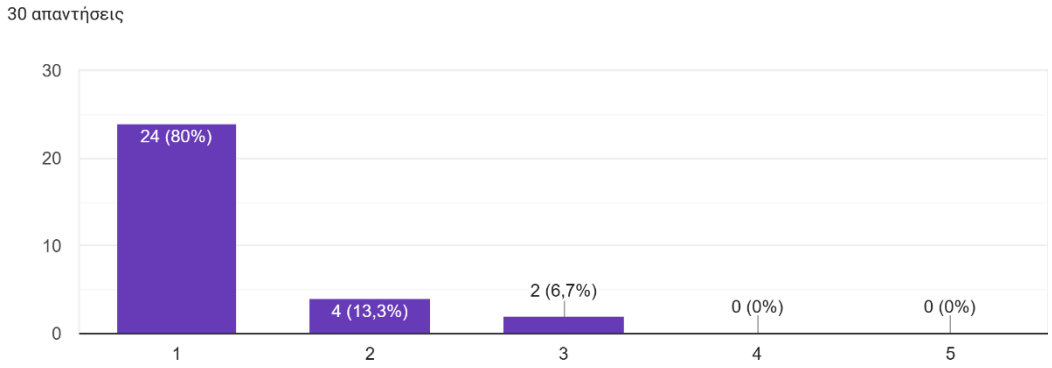
Εικόνα Β.7 – Not interesting(1) – Interesting(5)



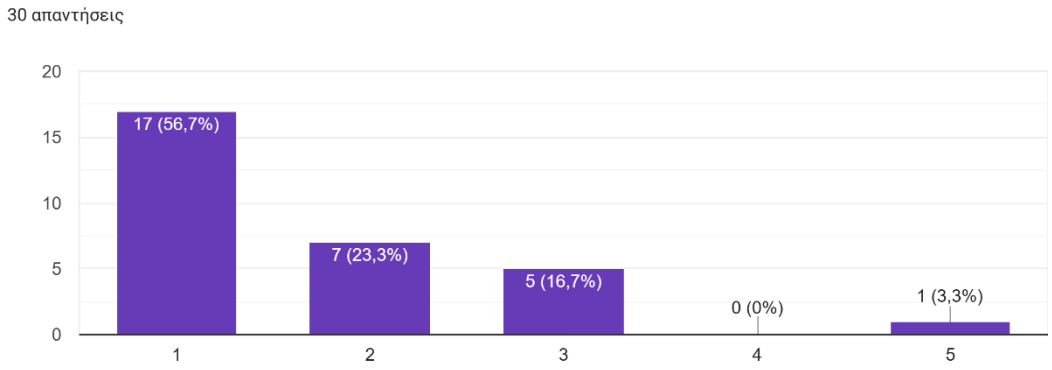
Εικόνα Β.8 – Unpredictable(1) – Predictable(5)



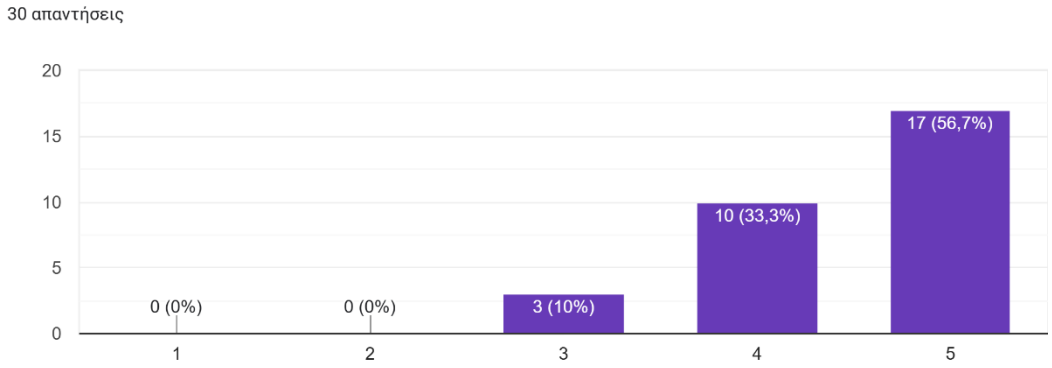
Εικόνα Β.9 – Fast(1) – Slow(5)



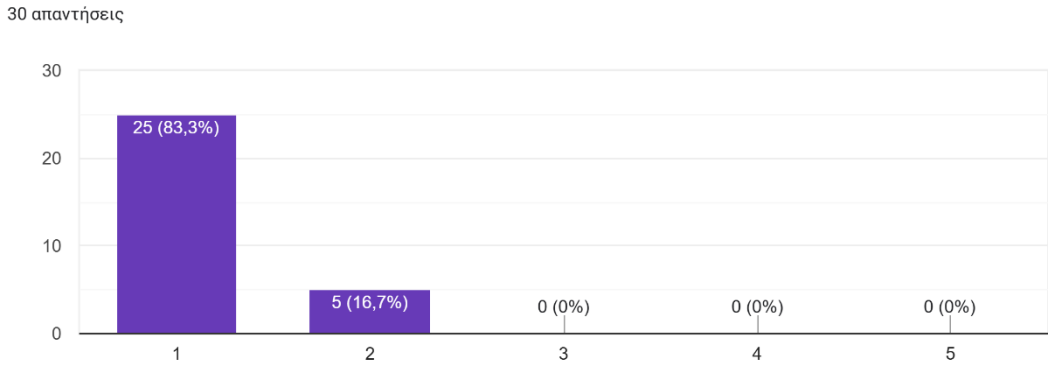
Εικόνα Β.10 – Inventive(1) – Conventional(5)



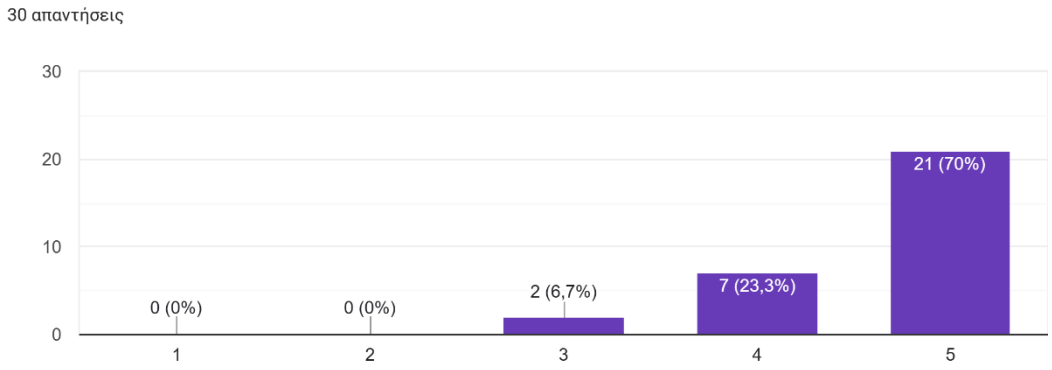
Εικόνα Β.11 – Obstructive(1) – Supportive(5)



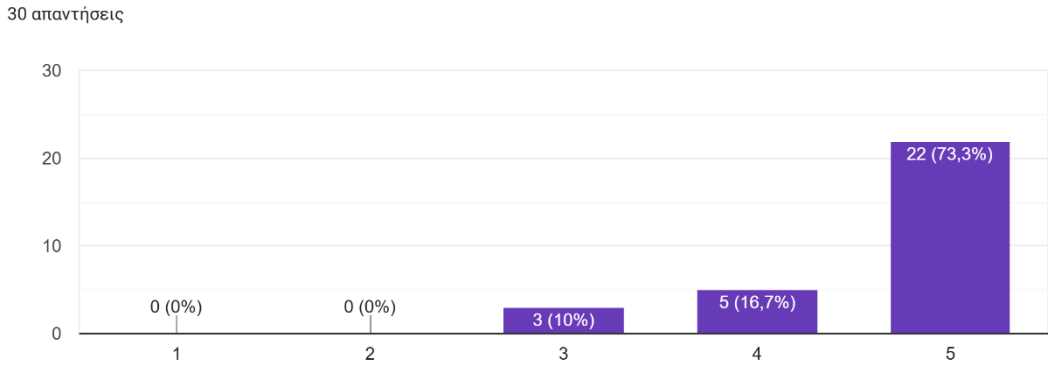
Εικόνα Β.12 – Good(1) – Bad(5)



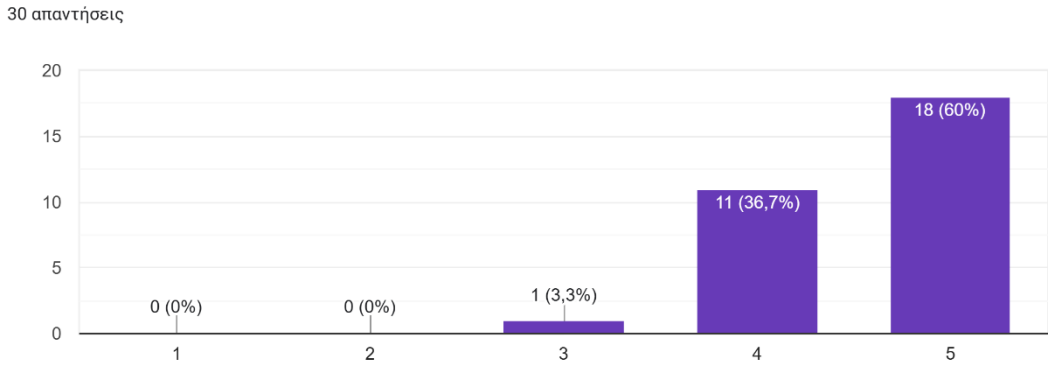
Εικόνα Β.13 – Complicated(1) – Easy(5)



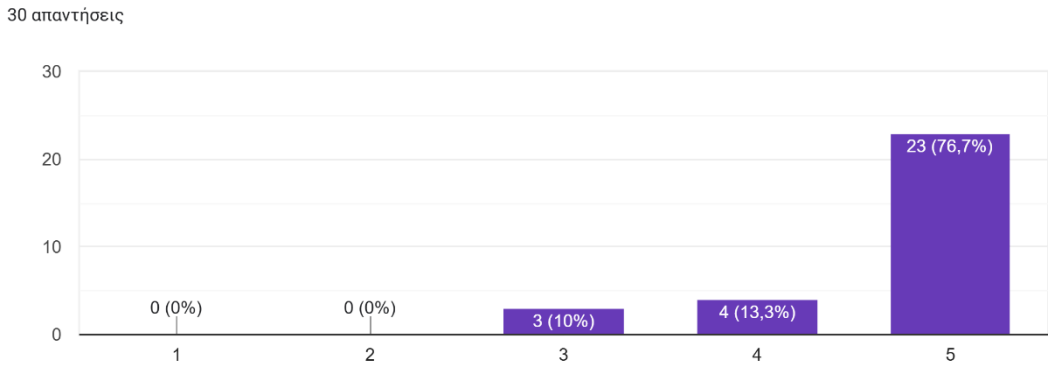
Εικόνα Β.14 – Unlikable(1) – Pleasing(5)



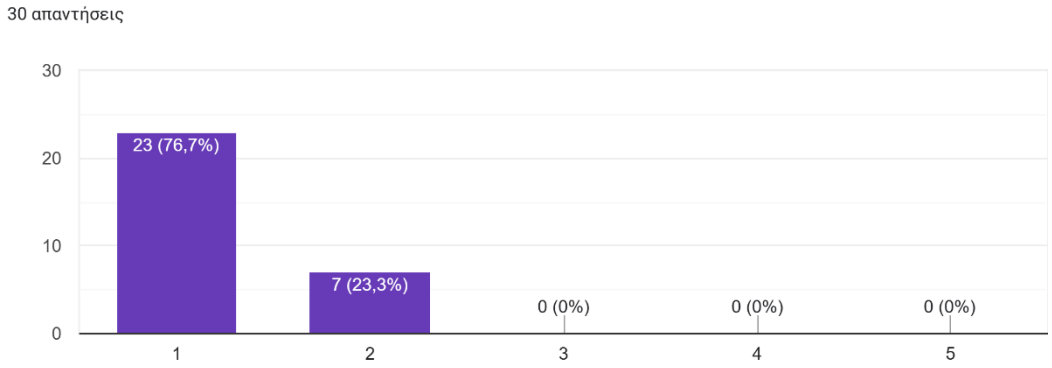
Εικόνα Β.15 – Usual(1) – Leading edge(5)



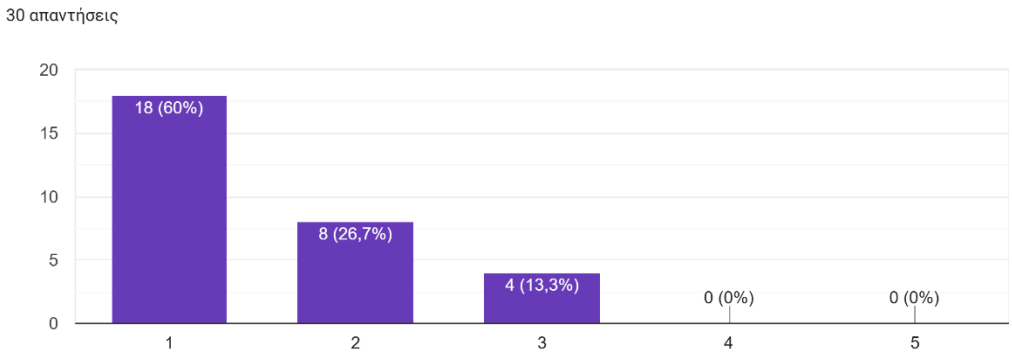
Εικόνα Β.16 – Unpleasant(1) – Pleasant(5)



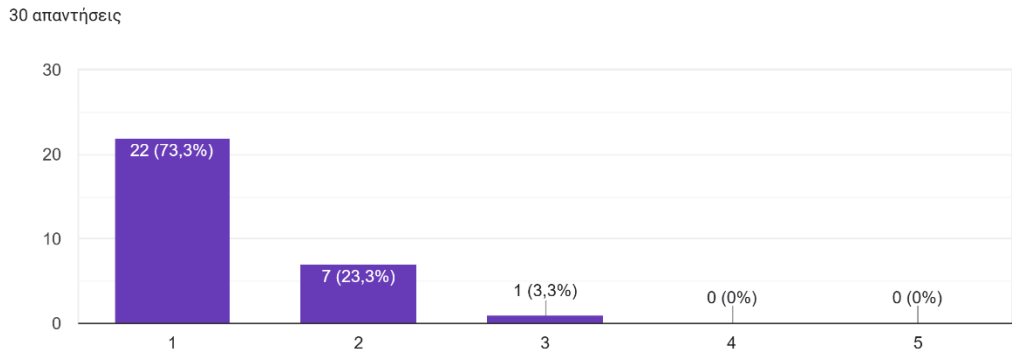
Εικόνα Β.17 – Secure(1) – Not Secure(5)



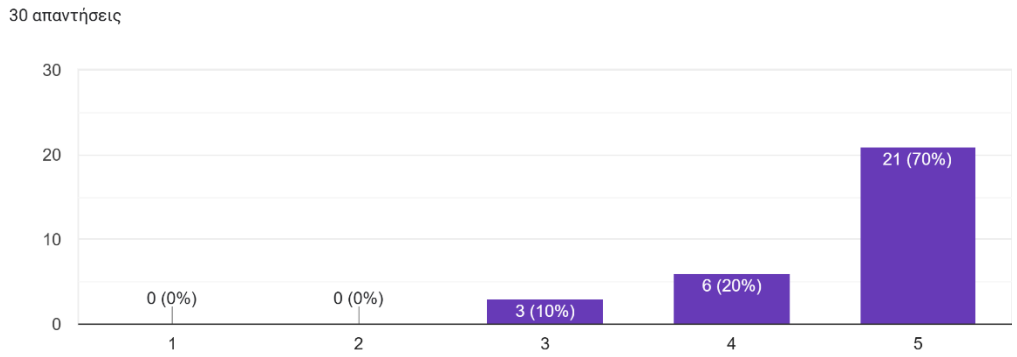
Εικόνα Β.18 – Motivating(1) – Demotivating(5)



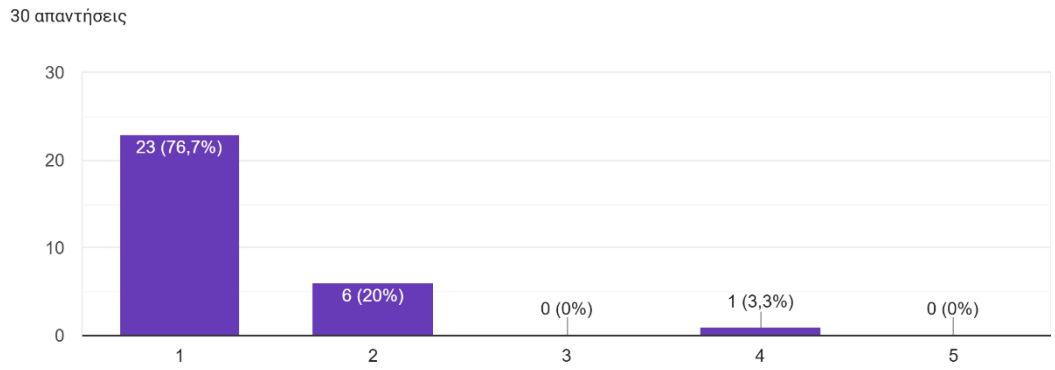
Εικόνα Β.19 – Meets expectations(1) – Does not meet expectations(5)



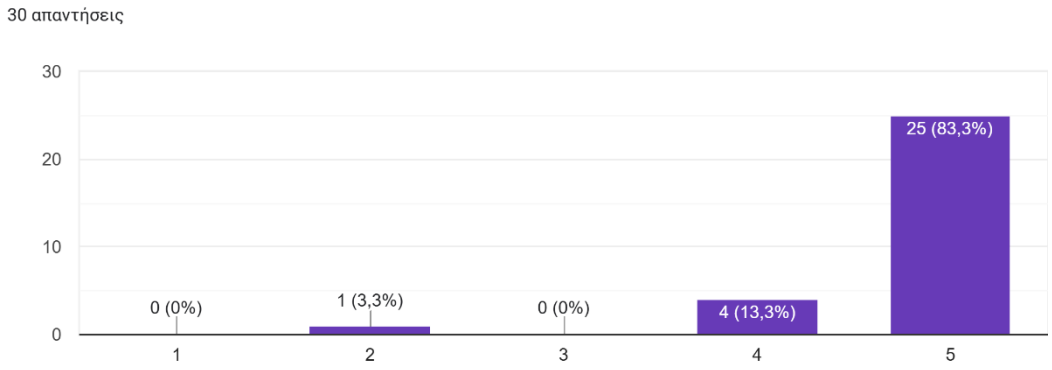
Εικόνα Β.20 – Inefficient(1) – Efficient(5)



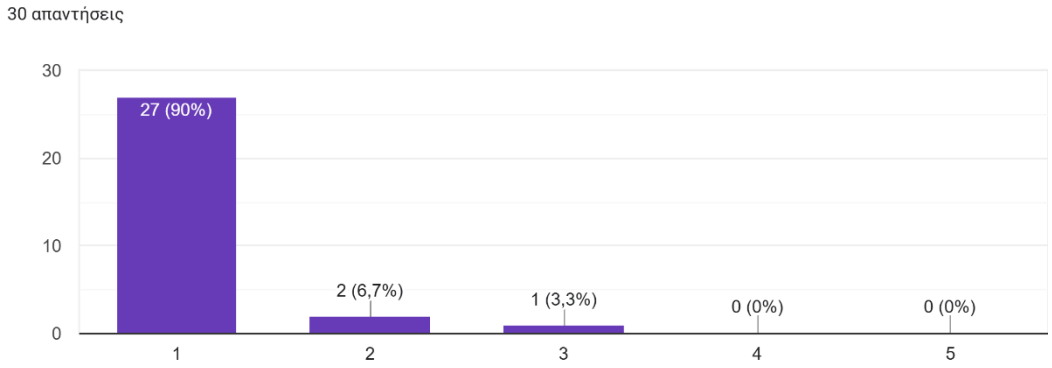
Εικόνα Β.21 – Clear(1) – Confusing(5)



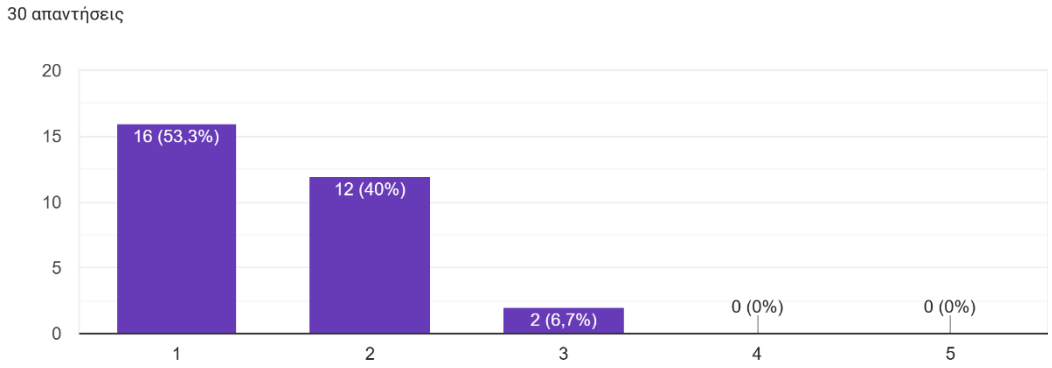
Εικόνα Β.22 – Impractical(1) – Practical(5)



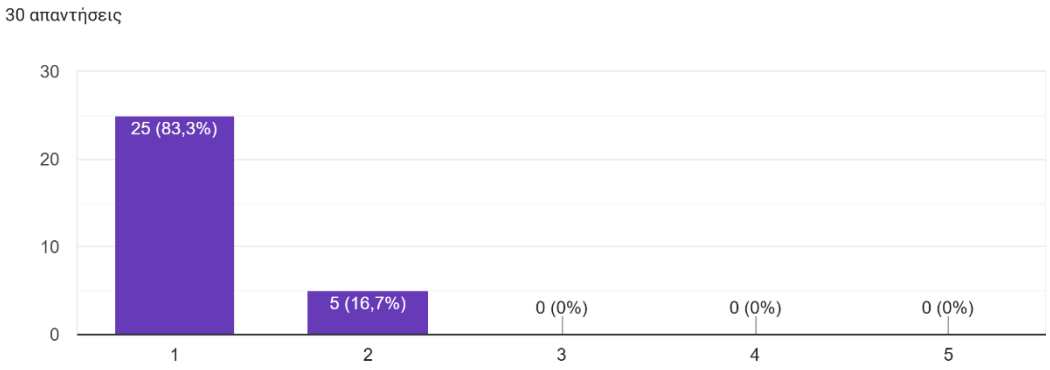
Εικόνα Β.23 – Organized(1) – Cluttered(5)



Εικόνα Β.24 – Attractive(1) – Unattractive(5)



Εικόνα Β.25 – Friendly(1) – Unfriendly(5)



Εικόνα Β.26 – Conservative(1) – Innovative(5)