

ΔΙΕΘΝΕΣ ΠΑΝΕΠΙΣΤΗΜΙΟ ΤΗΣ ΕΛΛΑΔΟΣ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

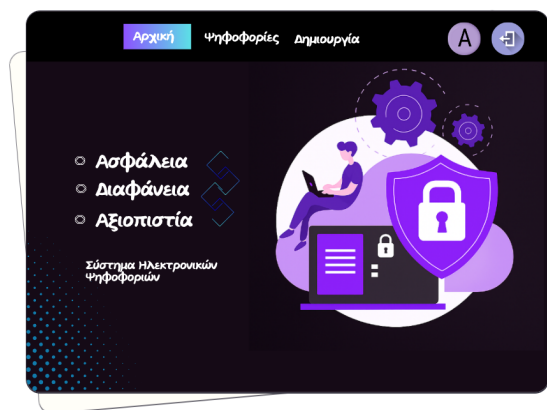
«Ανάπτυξη διαφανούς διαδικτυακού συστήματος ηλεκτρονικών ψηφοφοριών»

📄 Ηλεκτρονικές Ψηφοφορίες



Σύνδεση

Εγγραφή



Καλωσορίσατε στην πλατφόρμα ηλεκτρονικών ψηφοφοριών!

Ασφαλείς & Διαφανείς Ηλεκτρονικές Ψηφοφορίες

Η πλατφόρμα μας προσφέρει πλήρη ασφάλεια και διαφάνεια . Εύκολη χρήση από οποιαδήποτε συσκευή. Αξιοπιστία και εμπιστοσύνη σε κάθε ψηφοφορία.

Προηγούμενο

Επόμενο

Του φοιτητή
174871
Μαυρίδη Ευστάθιου

Επιβλέπων
Ουγιάρογλου Στέφανος
Επίκουρος Καθηγητής

Σεπτέμβριος 2025

Τίτλος Π.Ε.: Ανάπτυξη διαφανούς διαδικτυακού συστήματος ηλεκτρονικών ψηφοφοριών

Κωδικός Π.Ε.: 25212

Ονοματεπώνυμο φοιτητή: Ευστάθιος Μαυρίδης

Ονοματεπώνυμο εισηγητή: Ουγιάρογλου Στέφανος

Ημερομηνία ανάληψης Π.Ε.: 27-03-2025

Ημερομηνία περάτωσης Π.Ε.: 12-09-2025

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως πτυχιακή εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Δ.Ι.Π.Α.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μαυρίδη Ευστάθιου που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκφράζει στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, αποθήκευσης, παρουσιάσής της στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων πνευματικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), μετατροπή (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συγκατάθεση του συγγραφέα/δημιουργού.

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

«Στην Οικογένειά μου»

Πρόλογος

Η παρούσα πτυχιακή εργασία αναπτύχθηκε από τον φοιτητή Ευστάθιο Μαυρίδη υπό την επίβλεψη του καθηγητή κ. Στέφανου Ουγιάρογλου και εστιάζει στην ανάπτυξη ενός ολοκληρωμένου συστήματος ηλεκτρονικών ψηφοφοριών. Η ανάπτυξη της εφαρμογής ηλεκτρονικών ψηφοφοριών αποτέλεσε μια εξαιρετική ευκαιρία πρακτικής αξιοποίησης των θεωρητικών γνώσεων που αποκτήθηκαν κατά τη διάρκεια των σπουδών, ενώ παράλληλα συνέβαλε στην εμβάθυνση σε σύγχρονες τεχνολογίες web development. Μέσα από τη διαδικασία υλοποίησης, δόθηκε η δυνατότητα να εφαρμοστούν στην πράξη έννοιες όπως η ασφαλής διαχείριση ταυτότητας χρηστών με το Clerk, η διασύνδεση frontend και backend με React και Node.js, καθώς και η οργάνωση και διαχείριση δεδομένων με τη χρήση ORM εργαλείων όπως το Prisma. Επιπλέον, η σχεδίαση φιλικών και λειτουργικών διεπαφών ενίσχυσε τις δεξιότητες στη δημιουργία προσβάσιμων και διαδραστικών web εφαρμογών, ενώ η ενασχόληση με ζητήματα ασφάλειας, επαλήθευσης και διαφάνειας ανέδειξε τη σημασία της τεχνολογίας στη βελτίωση των δημοκρατικών διαδικασιών.

Περίληψη

Στις μέρες μας , η ραγδαία εξέλιξη της τεχνολογίας προσφέρει καινοτόμες λύσεις σε διάφορες διαδικασίες οι οποίες μέχρι στιγμής γίνονταν με την καταβολή μεγάλης προσπάθειας και χρόνου . Μια από τις λύσεις που προσφέρει αφορά και την διαδικασία της ψηφοφορίας .Η ηλεκτρονική ψηφοφορία μπορεί να αποτελέσει την σύγχρονη μορφή των εκλογικών διαδικασιών , προσφέροντας ασφάλεια , ταχύτητα και ακεραιότητα στην καταμέτρηση των ψήφων και στα αποτελέσματα . Στα πλαίσια της συγκεκριμένης πτυχιακής εργασίας αναπτύχθηκε ένα σύστημα ηλεκτρονικών ψηφοφοριών για οποιοδήποτε θέμα , το οποίο υποστηρίζει την εγγραφή των χρηστών την ταυτοποίηση των προσωπικών τους στοιχείων ώστε να γίνεται διαχωρισμός στο είδος της ψηφοφορίας και να εμφανίζονται οι κατάλληλες ψηφοφορίες μόνο στους χρήστες που τους αφορούν , την προσθήκη και διαγραφή μιας ψηφοφορίας από έναν χρήστη και την υποβολή ψήφων στις εκάστοτε ψηφοφορίες . Ιδιαίτερη έμφαση έχει δοθεί στην διασφάλιση της ανωνυμίας και στην ακεραιότητα της εκλογικής διαδικασίας . Η εργασία περιλαμβάνει την αξιολόγηση κάποιων συστημάτων ηλεκτρονικής ψηφοφορίας που έχουν υλοποιηθεί , την ανάλυση και τον σχεδιασμό ενός ιδανικού συστήματος ώστε να μην παραβιάζεται η ακεραιότητα των χρηστών καθώς και να είναι ακέραια τα αποτελέσματα . Επιπλέον παρουσιάζεται αναλυτικά ολόκληρη η υλοποίηση της διαδικτυακής εφαρμογής με όλες τις τεχνολογίες που έχουν χρησιμοποιηθεί. Στην συνέχεια γίνεται περιγραφή της εφαρμογής απο την οπτική πλευρά του χρήστη με την διεξαγωγή κάποιων δοκιμών ώστε να υπάρχει η αξιοπιστία του συστήματος . Τέλος το σύστημα ηλεκτρονικών ψηφοφοριών μπορεί να εφαρμοστεί σε διάφορες διαδικασίες ψηφοφορίας . Κάποια παραδείγματα είναι οι φοιτητικές εκλογές , σε διάφορα ζητήματα που αφορούν τις τοπικές κοινωνίες , ακόμα και σε απλά καθημερινά προβλήματα προσφέροντας μια γρήγορη και αξιόπιστη λύση.

«Open Source Web-based E-Voting System Ensuring Transparency and Integrity»

«Mavridis Efstathios»

Abstract

The drive behind the development of this web application was the need to create a real time electroning voting system to organizing elections . The application utilizes Next Js to implement both server side and client side components . This application can give speed in the procedure , better security and trust in the counting of the votes . The system allows the user to create a vote and to add candidates to that spesific vote . In addition the user has the right to vote in elections relevant to them based on their personal and demographic information . A strong focus was given to keep the anonymity of voters throw some security practises that took place during the development . This Thesis also includes some study of existing e voting systems and their evaluations . To develop the application , technologies such as React , HTML Typescript and Tailwind CSS were used for frontend while Next Js and PostgreSQL were utilized on the backend . Through this Thesis , the final system could be used in different cases . Some examples are student elections , internal organizations and some small elections . It offers a simple and modern way to do elections in a digital environment .

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω τον επιβλέποντα καθηγητή μου, κ. Στέφανο Ουγιάρογλου , για τη συνεχή υποστήριξή του κατά τη διάρκεια των σπουδών μου και της εκπόνησης της παρούσας πτυχιακής εργασίας. Η εμπιστοσύνη που έδειξε στην πρότασή μου για το θέμα της εργασίας μου, η συνεργασία στην επεξεργασία και διαμόρφωση της ιδέας, καθώς και οι εποικοδομητικές του παρατηρήσεις συνέβαλαν στην επιτυχή ολοκλήρωση αυτής της προσπάθειας.

Θα ήθελα, επίσης, να ευχαριστήσω θερμά όλους τους διδάσκοντες του Τμήματος για την στήριξη που μου παρείχαν σε όλη τη διάρκεια των σπουδών μου. Οι γνώσεις, οι συμβουλές και η καθοδήγησή τους αποτέλεσαν θεμέλιο για την ακαδημαϊκή μου εξέλιξη ενώ παράλληλα, με εφοδίασαν με δεξιότητες και εργαλεία που αποδείχθηκαν πολύτιμα και στην επαγγελματική μου δραστηριότητα.

Επιπλέον, θα ήθελα να εκφράσω την εκτίμησή μου στην εταιρεία Sigma Soft, στην οποία και πραγματοποίησα την πρακτική μου άσκηση. Η συνεργασία μου με τα στελέχη της εταιρείας και τους συναδέλφους μου με βοήθησε να εξοικειωθώ με σύγχρονες τεχνολογίες, αποκτώντας πολύτιμη εμπειρία σε σύγχρονες πρακτικές του κλάδου.

Τέλος, ένα μεγάλο και ειλικρινές ευχαριστώ στους φίλους στην οικογένειά , και στην κοπέλα μου για την αδιάκοπη στήριξη και ενθάρρυνση που μου προσέφεραν σε κάθε στάδιο αυτής της πορείας.

Περιεχόμενα

Πρόλογος	v
Περίληψη	vi
Abstract	vii
Ευχαριστίες	viii
Κατάλογος Εικόνων	xiii
Συντομογραφίες	xiv
1 Κεφάλαιο 1ο: Εισαγωγή	1
1.1 Ηλεκτρονική Ψηφοφορία και Αναγκαιότητα Ανάπτυξης Διαφανούς Συστήματος	1
1.2 Ανάγκη και Καινοτομία	1
1.3 Κίνητρο και Αναγκαιότητα Ανάπτυξης Ηλεκτρονικής Ψηφοφορίας	1
1.4 Ηλεκτρονική ψηφοφορία και συμμετοχή πολιτών	2
1.5 Σύγχρονες Τεχνολογικές Προσεγγίσεις στην Ηλεκτρονική Ψηφοφορία	3
1.5.1 Η τεχνολογία του Blockchain	3
1.5.2 Η χρήση του blockchain στο σύστημα ηλεκτρονικών ψηφοφοριών	4
1.5.3 Προκλήσεις στην χρήση του blockchain στο σύστημα ηλεκτρονικών ψηφοφοριών	4
1.5.4 Η ομοιομορφική κρυπτογράφηση	4
1.6 Επίλογος	5
2 Νομικό και Θεσμικό Πλαίσιο Ηλεκτρονικής Ψηφοφορίας	6
2.1 Εισαγωγή στο νομικό περιβάλλον	6
2.2 Διεθνείς προσεγγίσεις και καλές πρακτικές	6
2.3 Αναγκαίες θεσμικές προβλέψεις για ασφαλή υιοθέτηση	7
2.3.1 Κοινωνικές και Ηθικές Προεκτάσεις	8
2.3.2 Ψηφιακός Αναλφαβητισμός και Αποκλεισμός	8
2.3.3 Ζητήματα Εμπιστοσύνης και Νομιμοποίησης	8
2.3.4 Ηθικά Διλήμματα σε Τεχνολογίες Αναγνώρισης	9
2.3.5 Συμπεράσματα	9
3 Ασφάλεια και Προστασία Προσωπικών Δεδομένων σε Συστήματα Ηλεκτρονικής Ψηφοφορίας	10
3.0.1 Ανώνυμη και Ασφαλής Ψήφος	10
3.0.2 Νομικό Πλαίσιο και GDPR	10
4 Τεχνολογίες που χρησιμοποιήθηκαν στην εργασία	11
4.1 Εισαγωγή στις Τεχνολογίες	11
4.2 Next.js	12
4.3 TypeScript	13
4.4 Tailwind CSS	14
4.5 UI Components και Animation Libraries	14

4.6	State Management και Data Fetching	14
4.7	Διαχείριση Φορμών και Επικύρωση Δεδομένων	15
4.8	Αυθεντικοποίηση και Διαχείριση Χρηστών	15
4.9	Βάση Δεδομένων και ORM: Prisma και PostgreSQL	17
4.10	Εργαλεία Ανάπτυξης: Visual Studio Code , Supabase , Prisma και Vercel	17
4.11	Σύνοψη Κεφαλαίου	18
5	Τεχνική Υλοποίηση της Εφαρμογής	19
5.1	Εισαγωγή στην Υλοποίηση της Εφαρμογής	19
5.2	Λειτουργικές Απαιτήσεις	19
3.2	Λειτουργικές Απαιτήσεις	19
5.3	Σχεδίαση της Βάσης Δεδομένων	21
5.3.1	User	21
5.3.2	Election	21
5.3.3	PollCandidates	21
5.3.4	Takepart	22
5.3.5	Vote	22
5.3.6	Παραδείγματα SQL αιτημάτων σημαντικά για την εφαρμογή.	23
5.3.7	Συσχετίσεις Μεταξύ Αντικειμένων (Relationships)	28
3.4.5	Συσχετίσεις Μεταξύ Αντικειμένων (Relationships)	28
6	Περιγραφή των API του Συστήματος	30
6.1	API Endpoints	30
6.2	GET /api/elections	30
6.3	POST /api/elections	30
6.4	GET /api/elections/[id]	31
6.5	DELETE /api/elections/[id]	31
6.6	GET /api/elections/[id]/results	33
6.7	GET /api/elections/[id]/candidates	34
6.8	POST /api/elections/[id]/candidates	35
6.9	DELETE /api/elections/[id]/candidates/[candidateId]	36
6.10	POST /api/elections/[id]/vote	36
6.11	GET /api/elections/filtered	37
6.12	GET /api/elections/my	38
6.13	PollCandidates Endpoints	39
6.14	User και Profile Endpoints	40
7	Προβολή της Εφαρμογής	42
7.1	Διαδικασία Εγγραφής Νέου Χρήστη	44
7.2	Διαδικασία Σύνδεσης Χρήστη	44
7.3	Υποχρεωτική Συμπλήρωση Στοιχείων Κατά την Πρώτη Είσοδο	45
7.4	Επαλήθευση και Συμπλήρωση Προφίλ Χρήστη	46
7.5	Υποβολή Ψήφου και Προβολή Ενεργών Ψηφοφοριών	47
7.6	Προβολή Ολοκληρωμένων Ψηφοφοριών	51
7.7	Κεντρικός Πίνακας Διαχείρισης – Αναλυτική Περιγραφή	53
7.8	Σελίδα Διαχείρισης Ψηφοφοριών	54
7.9	Δημιουργία Νέας Ψηφοφορίας	55

7.10	Διαχείριση Υποψηφίων	61
7.11	Προβολή Όλων των Ψηφοφοριών	63
8	Συμπεράσματα και μελλοντικές επεκτάσεις	65
8.1	Συμπεράσματα	65
8.2	Μελλοντικές Επεκτάσεις και Προοπτικές	66

Κατάλογος Σχημάτων

1	Δομή blockchain με blocks και hash συνδέσεις (προσαρμοσμένο από NIST).	3
2	Overview of Homomorphic Encryption workflow. Source: [21]	4
3	Τεχνολογίες που έχουν χρησιμοποιηθεί στην υλοποίηση της εφαρμογής.	11
4	Σύγκριση τεχνικών χαρακτηριστικών: Next.js vs React	12
5	Έτοιμη φόρμα για εγγραφή [49]	16
6	Διάγραμμα Οντοτήτων και Συσχετίσεων (ER Diagram)	20
7	Διάγραμμα Οντοτήτων και Συσχετίσεων (ER Diagram)	28
8	Παράδειγμα επιστροφής του Api: GET /api/elections σε μορφή JSON	30
9	Παράδειγμα επιστροφής του Api: POST /api/elections σε μορφή JSON	31
10	Παράδειγμα επιστροφής του Api: GET /api/elections/[id] σε μορφή JSON	31
11	Παράδειγμα Database Schema σε μορφή schema.prisma	32
12	Επιτυχής επιστροφή του Api: DELETE /api/elections/[id] σε μορφή JSON	33
13	Παράδειγμα επιστροφής του Api: GET /api/elections/[id]/results σε μορφή JSON	34
14	Παράδειγμα επιστροφής του Api: GET /api/elections/filtered σε μορφή JSON	38
15	Παράδειγμα επιστροφής του Api: GET /api/poll-candidates σε μορφή JSON	39
16	Παράδειγμα επιστροφής του Api: POST /api/poll-candidates σε μορφή JSON	40
17	Παράδειγμα επιστροφής του Api: GET /api/users σε μορφή JSON	40
18	Παράδειγμα επιστροφής της αίτησης για το Api: POST /api/profile-sync σε μορφή JSON	40
19	Παράδειγμα επιστροφής της απόκρισης μετά την αίτηση για το Api: POST /api/profile-sync σε μορφή JSON	41
20	Παράδειγμα επιστροφής του Api: POST /api/verify σε μορφή JSON	41
21	Αρχική Σελίδα σε Φωτεινή Λειτουργία	42
22	Αρχική Σελίδα σε Σκοτεινή Λειτουργία (Dark Mode)	43
23	Φόρμα εγγραφής νέου χρήστη μέσω Clerk	44
24	Φόρμα σύνδεσης υφιστάμενου χρήστη μέσω Clerk	45
25	Μήνυμα ειδοποίησης για συμπλήρωση στοιχείων προφίλ χρήστη	45
26	Φόρμα Επαλήθευσης Προφίλ Χρήστη	46
27	Προβολή Ενεργών Ψηφοφοριών	47
28	Διαδικασία Υποβολής Ψήφου - Δημόσια ψηφοφορία	49
29	Διαδικασία Υποβολής Ψήφου - Περιορισμένη ψηφοφορία	49
30	Διαδικασία Υποβολής Ψήφου - Ιδιωτική ψηφοφορία	50
31	Διαδικασία Υποβολής Ψήφου - Προσκεκλημένοι σε ψηφοφορία	50
32	Προβολή Ολοκληρωμένων Ψηφοφοριών	52
33	Προβολή Αποτελέσματος Ψηφοφορίας	52
34	Κεντρικός πίνακας διαχείρισης της πλατφόρμας ηλεκτρονικής ψηφοφορίας.	53
35	Σελίδα Διαχείρισης Ψηφοφοριών	54
36	Φόρμα δημιουργίας νέας ψηφοφορίας – Δημόσια ψηφοφορία.	56
37	Επιλογή εναρξης και λήξης της ψηφοφορίας.	57
38	Φόρμα δημιουργίας νέας ψηφοφορίας – Ιδιωτική με κωδικό πρόσβασης	58
39	Φόρμα δημιουργίας νέας ψηφοφορίας – Μόνο ειδικά προσκεκλημένοι.	59
40	Προσθήκη προσκεκλημένων χρηστών.	59
41	Φόρμα δημιουργίας νέας ψηφοφορίας – Περιορισμένοι / targeting criteria.	60

42	Σελίδα διαχείρισης υποψηφίων – Κατάσταση χωρίς εγγεγραμμένους υποψηφίους.	61
43	Αναδυόμενο παράθυρο προσθήκης υποψηφίου – επιλογή από εγγεγραμμένους χρήστες.	61
44	Σελίδα διαχείρισης επιλογών κειμένου.	62
45	Σελίδα με επισκόπηση όλων των ψηφοφοριών σε μορφή καρτών.	63

Συντομογραφίες

FHE Fully Homomorphic Encryption

PHE Partially Homomorphic Encryption

DRM Digital Rights Management

SSR Server Side Rendering

SSG Static Site Generation

MVP Minimum Viable Product

UI User Interface

CSS Cascading Style Sheets

SVG Scalable Vector Graphics

CMD Command Menu for React

VSC Visual Studio Code

API Application Programming Interface

MFA Multi-Factor Authentication

SDK Software Development Kit

ORM Object-Relational Mapping

RDBMS Relational Database Management System

ΕΕ Ευρωπαϊκή Ένωση

GDPR General Data Protection Regulation (Γενικός Κανονισμός για την Προστασία Δεδομένων, ΕΕ 2016/679)

URL Uniform Resource Locator

ID Identifier

JSON JavaScript Object Notation

HTTP HyperText Transfer Protocol

Web World Wide Web

GIT Version Control System

1 Κεφάλαιο 1ο: Εισαγωγή

1.1 Ηλεκτρονική Ψηφοφορία και Αναγκαιότητα Ανάπτυξης Διαφανούς Συστήματος

Η συμμετοχή των πολιτών στις εκλογικές διαδικασίες αποτελεί θεμέλιο της δημοκρατίας [24, 27], ωστόσο ένα μεγάλο ποσοστό των πολιτών απέχει από τις ψηφοφορίες καθώς δεν εμπιστεύονται την διαδικασία[2]. Αυτή ακριβώς η αποχή μαζί με τα διάφορα ζητήματα ασφάλειας που υπάρχουν όπως για παράδειγμα στην καταμέτρηση των ψήφων μαζί με την αδυναμία πολλών πολιτών στην μετακίνηση τους προς τα εκλογικά κέντρα αναδεικνύουν ότι το πρόβλημα υπάρχει και βοηθούν στην δημιουργία μιας καινοτομίας στην συγκεκριμένη διαδικασία , δηλαδή την δημιουργία ενός ηλεκτρονικού συστήματος ψηφοφοριών . Μια πρώτη λύση έχει δημιουργηθεί από το σύστημα "e-voting" [1, 12]. Το συγκεκριμένο σύστημα έχει ως στόχο την πρόσβαση όλων των πολιτών στην διαδικασία των εκλογικών ψηφοφοριών από το σπίτι και την γρήγορη εκπόνηση της διαδικασίας. [5, 7]

Παρά τα πλεονεκτήματα, με την εφαρμογή συστημάτων e-voting έρχονται στην επιφάνεια ζητήματα τα οποία αφορούν την ασφάλεια και την εμπιστοσύνη των πολιτών ως προς το αποτέλεσμα της διαδικασίας. Τα παραδείγματα ηλεκτρονικών συστημάτων ψηφοφοριών, όπως το σύστημα iVote στην Αυστραλία [7] και το Estonian I-vote [5] , αποκάλυψαν τα ενδεχόμενα ρίσκα όταν δεν εφαρμόζονται κατάλληλες τεχνικές προστασίας δεδομένων και κρυπτογραφίας [6, 8] .

1.2 Ανάγκη και Καινοτομία

Η εκλογική διαδικασία και κάθε μορφή ψηφοφορίας μέχρι σήμερα γίνεται με μεγάλη καθυστέρηση επειδή προαπαιτεί την μετακίνηση όλων των πολιτών στα εκλογικά κέντρα , τον διαμοιρασμό τους σε αίθουσες με βάση τα ονοματεπώνυμα και την πολύωρη καταμέτρηση των ψήφων από τους ανθρώπους που έχουν αναλάβει την διαδικασία. Αντίθετα ένα καλά σχεδιασμένο σύστημα ψηφοφορίας μπορεί να κάνει ολόκληρη την διαδικασία σύντομη , ασφαλή και να προσφέρει με ταχύτητα τα αποτελέσματα [2, 12].

Επιπλέον απο την περιβαλλοντική πλευρά [28] η δημιουργία ενός ηλεκτρονικού συστήματος ψηφοφοριών , περιορίζει την άσκοπη σπατάλη χαρτιού και τις ανούσιες μετακινήσεις πολλών οχημάτων προς τα εκλογικά κέντρα [3] . Επιπρόσθετα στον κοινωνικό τομέα παρέχει ίσες ευκαιρίες συμμετοχής στις ψηφοφορίες για τις ευπαθείς και εύλωτες ομάδες ανθρώπων οι οποίες δεν μπορούν να ψηφίσουν με τον παραδοσιακό τρόπο [12].

Τέλος όταν ένα σύστημα παρέχει την δυνατότητα να πραγματοποιηθεί έλεγχος από ειδικούς φορείς καθώς προσφέρει και ανοιχτό κώδικα , τότε ενισχύει την εμπιστοσύνη και λειτουργεί κατά της χειραγώγησης[3, 8].

1.3 Κίνητρο και Αναγκαιότητα Ανάπτυξης Ηλεκτρονικής Ψηφοφορίας

Συνοπτικά , οι λόγοι για την δημιουργία ενός συστήματος ηλεκτρονικών ψηφοφοριών μπορούν να διατυπωθούν σε 4 κατηγορίες:

1. **Αύξηση του ποσοστού συμμετοχών:** Η απομακρυσμένη ψηφοφορία βοηθάει στην άνοδο των συμμετεχόντων αφού μπορούν να συμμετέχουν με ευκολία πολίτες του εξωτερικού και άτομα με διάφορες μορφές αναπηρίας [2, 12]

2. **Ταχύτητα της διαδικασίας :** Η καταμέτρηση των ψήφων μέσα από ένα ηλεκτρονικό σύστημα περιορίζει σημαντικά τα διάφορα ανθρώπινα λάθη που ενδέχεται να συμβούν και επιταχύνει την διαδικασία [6].
3. **Εμπιστοσύνη και διαφάνεια:** Ο ανοιχτός κώδικας μιας εφαρμογής μπορεί να ελεγχθεί από αρμόδιους φορείς υποστηρίζοντας έτσι την πλήρη διαφάνεια της διαδικασίας [3, 8, 11].
4. **Αποδοτικότητα πόρων:** Η μείωση των χαρτιών και η εξοικονόμηση των καυσίμων από τα οχήματα των πολιτών συμβάλλουν ενεργά στην εξοικονόμηση των πόρων και στην μείωση των ρύπων προς το περιβάλλον [9].

Η Εσθονία, για παράδειγμα, απέδειξε την αποτελεσματικότητα ενός τέτοιου συστήματος, με πάνω από 40% συμμετοχής στο i-vote [5].

1.4 Ηλεκτρονική ψηφοφορία και συμμετοχή πολιτών

Η ηλεκτρονική ψηφοφορία (e - voting) έχει δημιουργηθεί ως μία λύση στην μείωση της συμμετοχής των πολιτών στις εκλογικές διαδικασίες, καθώς διευκολύνει την ψηφοφορία και μειώνει τα εμπόδια συμμετοχής [1]. Ωστόσο, το αν είναι αποτελεσματικό παραμένει υπό συζήτηση. Διάφορες μελέτες από Καναδά, Ελβετία [1], Ηνωμένο Βασίλειο [12], Εσθονία [5] και άλλες χώρες αναφέρουν μικρή άνοδο στη συνολική συμμετοχή, αλλά θετική επίδραση σε ομάδες οι οποίες παρουσίαζαν χαμηλή ή μηδενική συμμετοχή [1, 2, 13].

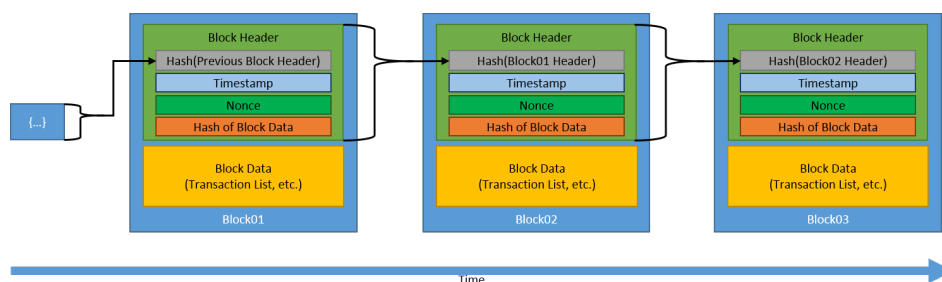
Σύμφωνα με τον Petitpas, Jaquet και Sciarini (2020) [14], το (e - voting) βοηθάει στην ενεργη συμμετοχή των ψηφοφόρων οι οποίοι δεν δείχνουν ιδιαίτερο ενδιαφέρον σε ολόκληρη την διαδικασία , χωρίς να επηρεάζει όσους ήδη ψηφίζουν τακτικά [2]. Αντίθετα, για τους ψηφοφόρους που έχουν συνηθίσει με τον κλασσικό τρόπο το (e - voting) αποτελεί συνήθως υποκατάστατο της παραδοσιακής ψηφοφορίας.

Από πλευράς ισότητας, το (e - voting) βοηθάει στην εύκολη συμμετοχή ευάλωτων ομάδων στην διαδικασία αλλά με κίνδυνο να απέχουν κοινωνικές ομάδες οι οποίες θέλουν να ακολουθήσουν την παραδοσιακή διαδικασία όπως τα άτομα τρίτης ηλικίας. [2, 28].

1.5 Σύγχρονες Τεχνολογικές Προσεγγίσεις στην Ηλεκτρονική Ψηφοφορία

1.5.1 Η τεχνολογία του Blockchain

Το blockchain ή διαφορετικά τεχνολογία κατακευματισμένης εγγραφής είναι ένας ψηφιακός κατάλογος στον οποίο μπορούμε να διακρίνουμε εγγραφές συναλλαγών που έχουν δημιουργηθεί με χρονολογική σειρά [17]. Ουσιαστικά είναι μια αλυσίδα από κόμβους . Κάθε κόμβος περιλαμβάνει τα δεδομένα τα οποία μπορεί να είναι μια συναλλαγή ή οποιαδήποτε άλλη πληροφορία , το hash ή "τιμή σύνοψης" του τρέχοντος κόμβου το οποίο είναι ένα μοναδικό αποτύπωμα για το περιεχόμενο που διαθέτει ο κόμβος και το hash του προηγούμενου κόμβου ώστε να υπάρχει η συνοχή της αλυσίδας [21]. Παρακάτω θα γίνει μια αναλυτική περιγραφή της τεχνολογίας του blockchain και το τι μπορεί να προσφέρει σε ένα σύστημα ηλεκτρονικών ψηφοφοριών.



Σχήμα 1: Δομή blockchain με blocks και hash συνδέσεις (προσαρμοσμένο από NIST).
[17]

Η τεχνολογία του blockchain είναι μια από τις πιο καινοτόμες τεχνολογίες που έχει δημιουργηθεί την τελευταία δεκαετία . Βλέπουμε ότι χρόνο με τον χρόνο χρησιμοποιείται η εφαρμογή του σε διάφορους τομείς όπως τα χρηματοοικονομικά , η υγεία καθώς και σε συστήματα ηλεκτρονικής ασφάλειας [17] . Το blockchain μπορούμε να το φανταστούμε σαν ένα λογιστικό βιβλίο το οποίο καταγράφει συναλλαγές σε μια αλυσίδα από κόμβους . Έτσι κάθε κόμβος συνδέεται με τον προηγούμενο δημιουργώντας μια αλυσίδα[15, 17]. Η βασική ιδέα του blockchain είναι ο διαμοιρασμός αυτής της αλυσίδας ώστε να μην υπάρχει σε κάποιον αποκλειστικό διακομιστή αλλά σε ένα σύνολο υπολογιστών , ώστε να μπορεί να αποφύγει επιθέσεις [15]. Έτσι κάθε νέα συναλλαγή που πραγματοποιείται στο blockchain πρέπει να επικυρωθεί σε ολόκληρο αυτό το δίκτυο των υπολογιστών με την βοήθεια διάφορων μηχανισμών συναίνεσης όπως το proof of work ή το proof of stake για να μπορεί να διασφαλιστεί η ακεραιότητα των δεδομένων . Ένα από τα βασικά χαρακτηριστικά του είναι η αμεταβλητότητα (immutability) των δεδομένων . Μόλις μια συναλλαγή κατοχυρωθεί στο blockchain δεν μπορεί να μετατραπεί ούτε να διαγραφεί [15]. Επιπλέον μια άλλη καινοτομία που προσφέρει είναι η πλήρη διαφάνεια , καθώς κάθε κόμβος μπορεί να έχει πρόσβαση σε ολόκληρο το ιστορικό των συναλλαγών [15] . Αυτά τα χαρακτηριστικά κάνουν το blockchain μια ιδανική λύση για τις εφαρμογές οι οποίες έχουν σαν προτεραιότητα την ασφάλεια , την διαφάνεια και την εμπιστοσύνη , απαιτήσεις που έχει και ένα σύστημα ηλεκτρονικών ψηφοφοριών [16].

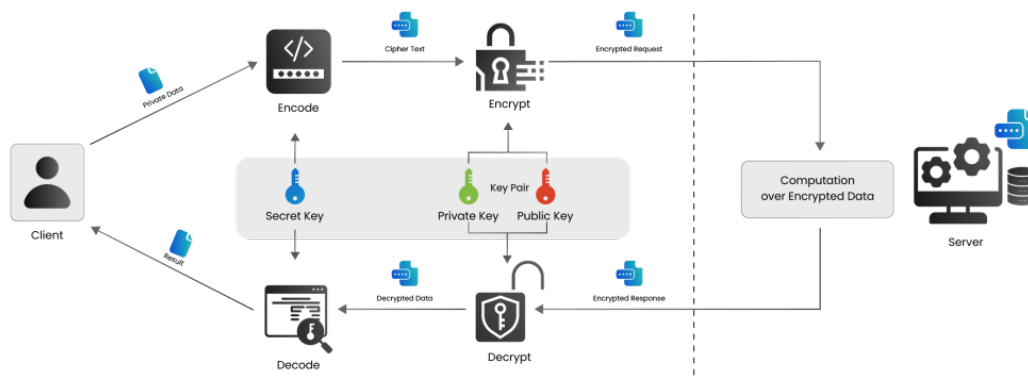
1.5.2 Η χρήση του blockchain στο σύστημα ηλεκτρονικών ψηφοφοριών

Το blockchain σε ένα σύστημα ηλεκτρονικών ψηφοφοριών μπορεί να προσφέρει πολλά πλεονεκτήματα . Αρχικά αποφεύγει την διαφθορά καθώς ενισχύεται η διαφάνεια της διαδικασίας . Αυτό συμβαίνει επειδή κάθε ψήφος είναι καταγεγραμμένη και μπορεί να ελεγχθεί οποιαδήποτε στιγμή χωρίς να μπορεί να γίνει κάποια αλλοίωση της αργότερα [15]. Δεύτερον αυξάνει την ασφάλεια αφού κάθε μια από τις συναλλαγές που πραγματοποιείται είναι κρυπτογραφημένη [15] . Τρίτον δίνει την επιλογή στον κάθε ψηφοφόρο να μπορεί να ελέγξει την ψήφο του . Τέλος συμβάλλει στην μείωση του κόστους ολόκληρης της διαδικασίας [15].

1.5.3 Προκλήσεις στην χρήση του blockchain στο σύστημα ηλεκτρονικών ψηφοφοριών

Κάποια από τα βασικά προβλήματα του blockchain σχετίζονται με την κλιμάκωση (scalability) των συναλλαγών στο δίκτυο , με την ταχύτητα της επεξεργασίας των συναλλαγών και με την συμμόρφωση στα νομικά πλαίσια [9, 15, 16]. Επιπλέον το blockchain έχει ιδιαίτερη σε δυσκολία τεχνολογία και αυτό μπορεί να δημιουργήσει πρόβλημα καθώς αυξάνεται η ανάγκη για εκμάθηση . Παρόλα αυτά πολλές χώρες έχουν ξεκινήσει να υλοποιούν ηλεκτρονικά συστήματα με την τεχνολογία του blockchain και διερευνούν την χρήση του σε συστήματα ηλεκτρονικών ψηφοφοριών [15].

1.5.4 Η ομομορφική κρυπτογράφηση



Σχήμα 2: Overview of Homomorphic Encryption workflow. Source: [21]

Η ομομορφική κρυπτογράφηση είναι πλέον ένα από τα πιο προηγμένα πεδία της κρυπτογραφίας. Δίνει τεράστια έμφαση στην προστασία της ιδιωτικότητας και κυρίως στην ασφάλεια των δεδομένων στα περισσότερα πληροφοριακά συστήματα όπως και στα συστήματα ηλεκτρονικών ψηφοφοριών [18, 21] . Στην κλασσική κρυπτογράφηση για να γίνει επεξεργασία κάποιων δεδομένων πρέπει να προηγηθεί η αποκρυπτογράφηση τους . Αντίθετα στην ομομορφική κρυπτογράφηση μπορεί να πραγματοποιηθεί εκτέλεση υπολογισμών επάνω σε κρυπτογραφημένα δεδομένα , χωρίς να χρειάζεται να γίνει πρώτα η διαδικασία της αποκρυπτογράφησης [6]. Το αποτέλεσμα εμφανίζεται σε κρυπτογραφημένη μορφή και μπορεί να αποκρυπτογραφηθεί σε μεταγενέστερο χρόνο , δίνοντας ακριβώς το ίδιο αποτέλεσμα που θα πέραμε αν κάναμε την

διαδικασία με τον κλασσικό τρόπο κρυπτογράφησης [19]. Η έννοια της ομομορφικής κρυπτογράφησης δημιουργήθηκε πρώτη φορά από τους Rivest , Adleman και Dertouzos το 1978 [20] αλλά η πρώτη ολοκληρωμένη ομομορφική κρυπτογράφηση υλοποιήθηκε από τον Craig Centry το 2009 [19, 20]. Υπάρχουν διαφορα είδη ομομορφικής κρυπτογράφησης , για παράδειγμα υπάρχει η μερικώς ομομορφική κρυπτογράφηση στην οποία μπορεί να γίνει μόνο ένας τύπος πράξης (πρόσθεση ή πολλαπλασιασμός) και η πλήρης ομομορφική κρυπτογράφηση στην οποία μπορούν να πραγματοποιηθούν όλες οι πράξεις πάνω στα κρυπτογραφημένα δεδομένα . Η δεύτερη χρησιμοποιείται κυρίως στα συστήματα τα οποία απαιτούν υψηλή ασφάλεια και εμπιστοσύνη όπως το cloud computing [19, 20, 21]. Τέλος η ομομορφική κρυπτογράφηση προσφέρει την πλήρη διαφάνεια και την προστασία της ιδιωτικότητας στο σύστημα ηλεκτρονικής ψηφοφορίας .

1.6 Επίλογος

Στο παρόν κεφάλαιο πραγματοποιήθηκε η ανάλυση του ζητήματος της παραδοσιακής ψηφοφορίας με αναφορά στις συνέπειες που απορρέουν από την ανυπαρξία ενός ηλεκτρονικού συστήματος ψηφοφοριών . Έξεταστηκαν αναλυτικά τα προβλήματα που προκύπτουν , όπως η αποχή πολλών πολιτών απο την διαδικασία , ο χρόνος που απαιτείται για να ολοκληρωθεί και διαφάνεια του αποτελέσματος . Επιπλέον παρουσιάστηκαν κάποιες απο τις νέες τεχνολογικές εξελίξεις οι οποίες μπορούν να φέρουν ικανή λύση σε πολλά ζητήματα που έχουν προκύψει στην διαδικασία δημιουργίας ενός ηλεκτρονικού συστήματος και αφορούν κυρίως την ασφάλεια . Τέτοιες τεχνολογίες είναι το blockchain και η ομομορφική κρυπτογράφηση . Στο κεφάλαιο 2 ακολουθεί η λεπτομερής ανάλυση όσον αφορά το νομικό και θεσμικό πλαίσιο για την δημιουργία ενός τέτοιου συστήματος .

2 Νομικό και Θεσμικό Πλαίσιο Ηλεκτρονικής Ψηφοφορίας

2.1 Εισαγωγή στο νομικό περιβάλλον

Για να μπορέσει να υπάρξει σύστημα ηλεκτρονικών ψηφοφοριών στην Ελλάδα θα πρέπει να γίνουν ριζικές αλλαγές σε νομικό και θεσμικό πλαίσιο. Ιδιαίτερα πρέπει να δίνεται έμφαση στα νομικά πλαίσια όταν οι ψηφοφορίες αφορούν δημόσιες ή κρατικές εκλογές. Μια τέτοια ψηφοφορία είναι αρκετά σημαντική καθώς αφορά το μέλλον της χώρας και κρίνεται αναγκαίο να διασφαλιστεί η μυστικότητα της κάθε ψήφου καθώς και η διαφάνεια ολόκληρης της διαδικασίας [12].

Στην Ελλάδα, δεν υπάρχει ακόμα σύστημα ηλεκτρονικών ψηφοφοριών για την διεξαγωγή των ψηφοφοριών που αφορούν την χώρα και ακριβώς για αυτό τον λόγο δεν υπάρχει συνταγματικό άρθρο για αυτήν την ψηφιακή διαδικασία. Οι μόνες αναφορές γίνονται στο άρθρο 51 του συντάγματος το οποίο αφορά τον παραδοσιακό τρόπο εκπόνησης των ψηφοφοριών το οποίο αναφέρει ότι κάθε πολίτης έχει το δικαίωμα να ψηφίζει ανώνυμα τους βουλευτές οι οποίοι πρόκειται να εκπροσωπήσουν το έθνος [24]. Αυτό σημαίνει ότι κανείς δεν πρέπει να γνωρίζει τι έχει ψηφίσει ένας συμπολίτης του. Σε ένα σύστημα ηλεκτρονικής ψηφοφορίας αυτό είναι ένα δύσκολο κομμάτι γιατί σε πολλές περιπτώσεις χρειάζεται να ελεγχθεί αν η ψήφος καταγράφηκε σωστά. Παρακάτω θα δούμε ότι για αυτόν ακριβώς τον λόγο πολλές χώρες χρησιμοποιούν τεχνολογίες προστασίας όπως τα mix nets, η ομομορφική κρυπτογράφηση και τα zero knowledge proofs [8, 19, 20, 21]. Η απαίτηση για την προσθήκη όσο το περισσότερο λιγότερο δεδομένων του χρήστη στην διαδικασία της ψηφοφορίας δημιουργεί σύγκρουση σε τεχνολογίες όπως το blockchain, που διατηρούν *αμετάβλητα* και *μόνιμα* δεδομένα. Έτσι, απαιτείται προσεκτικός σχεδιασμός ολόκληρου του συστήματος ηλεκτρονικών ψηφοφοριών ώστε να μην εμφανίζονται προβλήματα σχετικά με την ακεραιότητα και την ασφάλεια. Στην Ελλάδα, οι μόνες πλατφόρμες που έχουν δημιουργηθεί με εμπειρία e-voting είναι πλατφόρμες όπως η Ζευς για φοιτητικές ή και εσωτερικές εκλογές φορέων, όμως προσφέροντας μόνο βασικές λειτουργίες ταυτοποίησης, ψήφου και καταμέτρησης [25]. Παρότι η πλατφόρμα Ζευς έχει αναπτυχθεί σε πολύ καλό επίπεδο και διαθέτει αρκετές από τις τεχνικές προδιαγραφές για παράδειγμα το end-to-end verifiability, έχει περιορισμένη χρήση και δεν έχει χρησιμοποιηθεί σε εκλογές εθνικού ενδιαφέροντος [25].

2.2 Διεθνείς προσεγγίσεις και καλές πρακτικές

Ορισμένες χώρες έχουν ήδη βάλει σε λειτουργία την ηλεκτρονική ψηφοφορία στο εθνικό τους επίπεδο. Η **Εσθονία** [5] είναι το καταλληλότερο παράδειγμα, η οποία έχει εφαρμόσει το σύστημα i-Vote ήδη από το 2005. Η επιτυχία του συστήματος της αποδίδεται στο ισχυρό eID σύστημα, το νομικό της υπόβαθρο και την αποδοχή από τους πολίτες. Ο Εσθονικός εκλογικός νόμος προβλέπει δυνατότητα επαναψηφοφορίας, στοιχείο που μειώνει την πιθανότητα εξαναγκασμού. [5, 10, 12].

Η **Ελβετία** [2], από την άλλη, έχει θεσπίσει το e-voting [14] αλλά έχει επίσης δημιουργήσει και εφαρμογές οι οποίες θα χρησιμοποιηθούν σε περιπτώσεις τεχνικών αδυναμιών ή κενών διαφάνειας. Το Ομοσπονδιακό Γραφείο της Πληροφορικής της χώρας προτασσει την ανάγκη του ανοικτού κώδικα, πολλών τρίτων ελέγχων και επιστημονικών αξιολογήσεων [12, 13].

2.3 Αναγκαίες θεσμικές προβλέψεις για ασφαλή υιοθέτηση

Για την ασφαλή ενσωμάτωση του συστήματος ηλεκτρονικών ψηφοφοριών e-voting, προτείνονται οι εξής ρυθμίσεις:

- Να υπάρξει ένα **ενιαίο πλαίσιο ελέγχου και πιστοποίησης**, σε συνεργασία με διάφορους ανεξάρτητους οργανισμούς.
- Ο κωδικας να είναι **υποχρεωτικά ανοικτός** και να μπορεί να **επαλήθευθεί από τους πολίτες**.
- Να υπάρχει νομικά το δικαίωμα για **επαναψήφιση**, καθώς και να αποφυγονται οι καταναγκασμοί και να μην υπάρχουν προσβολές βούλησης.
- Προδιαγραφές **προσβασιμότητας** για ΑμεΑ και πολίτες χωρίς ψηφιακή ταυτότητα.
- Να υπάρχουν ποινικές διώξεις αν γίνει οποιαδήποτε παραβίαση της μυστικότητας ή αν υπάρξει αλλοίωση του αποτελέσματος [26].

Συμπερασματικά η υιοθέτηση ενός ηλεκτρονικού συστήματος απαιτεί την σωστή νομική θεμελίωση. Η τεχνολογία μπορεί να εγγυηθεί την διαφάνεια και την ασφάλεια ενός τέτοιου συστήματος μόνο όταν υπάρχει θεσμική λογοδοσία, δημοκρατικός έλεγχος και πλήρη συμμόρφωση όσον αφορά τα δικαιώματα των πολιτών.

2.3.1 Κοινωνικές και Ηθικές Προεκτάσεις

Η ανάπτυξη ενός ηλεκτρονικού συστήματος ψηφοφοριών δεν είναι απλά η επίλυση και η δημιουργία ενός λειτουργικού ζητήματος αλλά περιέχει και κοινωνικές και ηθικές διαστάσεις . Ο κάθε πολίτης έχει το δημοκρατικό δικαίωμα στο να εκλεχθεί καθώς και να ψηφίσει [24] και για αυτό τον λόγο η μετάβαση σε ηλεκτρονικό σύστημα μπορεί να δημιουργήσει κρίσιμα ερωτήματα και προβλήματα .Κάποια από τα ερωτήματα που προκύπτουν έχουν να κάνουν με το πως μπορεί να προστατευτεί η ελευθερία για την ψήφο που ο κάθε πολίτης επιθυμεί πως μπορεί να διασφαλιστεί η ισότητα συμμετοχής [28], υπάρχει τίμημα για κάποιες κοινωνικές ομάδες μέσα από την ψηφιακή διαδικασία [29]. Κάποιες έρευνες που έχουν γίνει δείχνουν ότι η ηλεκτρονική διαδικασία βοηθάει την διαφάνεια και την συμμετοχή των πολιτών[27] αλλά μπορούν να δημιουργήσουν προβλήματα ψηφιακής ανισότητας [28, 29].

2.3.2 Ψηφιακός Αναλφαβητισμός και Αποκλεισμός

Η εφαρμογή ενός ηλεκτρονικού συστήματος μπορεί να δημιουργήσει κοινωνικές ανισότητες κυρίως σε πληθυσμούς οι οποίοι δεν χρησιμοποιούν την τεχνολογία ή δεν έχουν πρόσβαση σε αυτήν [28]καθώς και αν έχουν ανεπαρκείς ψηφιακές γνώσεις [4]. Επιπλέον επηρεάζονται άμεσα οι ηλικιωμένοι άνθρωποι οι οποίοι δεν γνωρίζουν καθόλου από την τεχνολογία[12] όπως και άτομα με ειδικές ανάγκες citeVVS2021accessκαθώς και άτομα με χαμηλό μορφωτικό επίπεδο [28]. Η αλλαγή σε ψηφιακή διαδικασία δεν αφορά μόνο το μέσο με το οποίο θα ψηφίζουν οι άνθρωποι αλλά αφορά και την ικανότητα κατανόησης και χρήσης των σχετικών εφαρμογών [4, 12, 28].

2.3.3 Ζητήματα Εμπιστοσύνης και Νομιμοποίησης

Για να δημιουργηθεί η ψηφιακή ηλεκτρονική ψηφοφορία θα πρέπει αρχικά να μπορούν οι πολίτες να εμπιστευτούν ένα τέτοιο σύστημα ώστε να ακολουθήσουν την διαδικασία [2] . Σε χώρες οι οποίες έχουν χαμηλό μορφωτικό επίπεδο είναι σκέδον αδύνατο να γίνει κάτι τέτοιο .Αντιθέτως σε αυτές τις χώρες η δημιουργία ενός τέτοιο συστήματος θα δημιουργήσει έντονη αβεβαιότητα και καχυποψία ως προς τα αποτελέσματα [28] . Οι πολίτες πρέπει να μπορούν να κατανοήσουν τον τρόπο με τον οποίο διεξάγεται η διαδικασία και να μπορούν να είναι σίγουροι ότι το αποτέλεσμα της ψηφοφορίας δεν έχει δεχθεί κάποια παρέμβαση [8] . Σε αυτό το σημείο η διαφάνεια του λογισμικού δηλαδή να είναι με ανοιχτό κώδικα αποτελεί καθοριστική σημασία για την κοινωνική αποδοχή [3, 4, 12].

2.3.4 Ηθικά Διλήμματα σε Τεχνολογίες Αναγνώρισης

Η ενσωμάτωση των βιομετρικών δεδομένων του χρήστη δηλαδή δακτυλικά αποτυπώματα, αναγνώριση προσώπου ή ταυτοποίησης μέσω του κινητού τους θέτει σοβαρά νομικά ερωτήματα [29]. Η διαχείριση αυτών των προσωπικών δεδομένων του κάθε χρήστη είναι ένα πεδίο μεγάλης ευθύνης καθώς αν υπάρξει κάποια διαρροή μπορεί να προκαλέσει βλάβη στο δικαίωμα του χρήστη ως προς το απόρρητο της ψηφοφορίας [15]. Επιπλέον ένας ακόμα σημαντικός λόγος είναι ότι η αποθήκευση των βιομετρικών δεδομένων των πολιτών έρχεται σε σύγκρουση με την ελεύθερη βουλή τους . [29].

2.3.5 Συμπεράσματα

Η ηλεκτρονική ψηφοφορία δεν είναι απλά ένα τεχνολογικό πρόβλημα , αλλά μια επιλογή που μπορεί να επηρεάσει την δημοκρατία ενός κράτους . Για την υιοθέτηση της πρέπει να πραγματοποιηθούν ριζικές αλλαγές σε νομικό και κοινωνικό πλαίσιο .

Παράλληλα , πρέπει να λαμβάνεται υπόψη ότι δεν έχουν όλοι οι πολίτες το ίδιο μορφωτικό επίπεδο και επιπλέον υπάρχουν κοινωνικές ανισότητες οι οποίες επηρεάζουν κυρίως άτομα τρίτης ηλικίας , άτομα με ειδικές ανάγκες και άτομα τα οποία δεν μπορούν να ανταπεξέλθουν στο κόστος καθώς δεν διαθέτουν ηλεκτρονικές συσκευές . Επιπλέον κρίνεται αναγκαία η εμπιστοσύνη των πολιτών σε ένα τέτοιο σύστημα και για αυτόν τον λόγο ο πηγαίος κώδικας της εφαρμογής η οποία θα αναλάβει την διαδικασία θα πρέπει να είναι ανοιχτού τύπου .

Τέλος η προσθήκη των τεχνολογιών ταυτοποίησης και των βιομετρικών δεδομένων μπορεί να δημιουργήσει σοβαρά ηθικά και νομικά διλήμματα τα οποία απευθύνονται στην ιδιωτικότητα του χρήστη της εφαρμογής . Συνεπώς, η επιτυχία ενός συστήματος ηλεκτρονικής ψηφοφορίας δεν έχει να κάνει μόνο με το τεχνολογικό κομμάτι αλλά συνδιάζεται με την κοινωνική αποδοχή , με την σωστή νομική και θεσμική θέση από την πλευρά του κράτους και κυρίως με τον απόλυτο σεβασμό στο δημοκρατικό δικαίωμα του κάθε πολίτη . Στο επόμενο κεφάλαιο θα δούμε πως μπορεί να προαχθεί η ασφάλεια και η προστασία των προσωπικών δεδομένων σε συστήματα ηλεκτρονικής ψηφοφορίας .

3 Ασφάλεια και Προστασία Προσωπικών Δεδομένων σε Συστήματα Ηλεκτρονικής Ψηφοφορίας

Σε αυτήν την ενότητα θα δούμε ότι η ασφάλεια της ιδιωτικότητας των χρηστών είναι κρίσιμη για την σωστή λειτουργία οποιουδήποτε συστήματος . Θα γίνει αναλυτική περιγραφή με τους τρόπους και τις τεχνολογίες με τις οποίες μπορεί να διαφυλαχθεί η προστασία των δεδομένων , η εξασφάλιση της ανωνυμίας , η ακεραιότητα και η προστασία του συστήματος και της διαδικασίας .

3.0.1 Ανώνυμη και Ασφαλής Ψήφος

Το σύστημα ηλεκτρονικών ψηφοφοριών πρέπει να βεβαιώνει τον χρήστη ότι η ψήφος του είναι μυστική ώστε να μην φοβάται ότι αν ψηφίσει θα υπάρξουν κάποια αντίποινα και να μην του δημιουργείται κοινωνική πίεση . Παράλληλα όμως πρέπει να φαίνεται ότι η ψήφος του έχει λάβει μέρος στην διαδικασία . Η συγκεκριμένη αντίθεση μπορεί να λυθεί με χρήση των παρακάτω τεχνολογιών ασφάλειας :

- Homomorphic Encryption [19, 20, 21]: Με χρήση Homomorphic Encryption μπορεί να πραγματοποιηθεί καταμέτρηση των ψήφων χωρίς να χρειάζεται να γίνει πρώτα αποκρυπτογράφηση τους .
- Mixnets [8]: Τα συγκεκριμένα δίκτυα είναι δίκτυα ανάμειξης και αυτό που κάνουν είναι να μπερδεύουν τις ψηφους ώστε να μην φαίνεται η προέλευση τους .
- Zero Knowledge Proofs: Με τους συγκεκριμένους μηχανισμούς επιβεβαιώνεται η εγκυρότητα μιας ψήφου χωρίς όμως να φαίνεται ποιος είναι ο δημιουργός της .

Αυτες οι τεχνολογίες έχουν δοκιμαστεί σε διάφορα συστήματα όπως το Helios [11] και το Prêt à Voter σε χώρες που έχουν ήδη υλοποιήσει σύστημα ηλεκτρονικών ψηφοφοριών και έχει αποδειχθεί ότι λειτουργούν σωστά και δεν καταπατούν τα δημοκρατικά δικαιώματα των πολιτών. [5, 10].

3.0.2 Νομικό Πλαίσιο και GDPR

Η Ευρωπαϊκή Ένωση έχει δημιουργήσει ένα γενικό κανονισμό GDPR ο οποίος ορίζει ότι κάθε σύστημα το οποίο δέχεται προσωπικά δεδομένα πολιτών πρέπει να πληροί συγκεκριμένα κριτήρια [15]. Για τα συστήματα τα οποία απευθύνονται σε ηλεκτρονική ψηφοφορία πρέπει :

- Να υπάρχει όσο το δυνατόν λιγότερη προσθήκη προσωπικών δεδομένων .
- Να υπάρχει πλήρη διαφάνεια στην επεξεργασία όλων των δεδομένων.
- Να υπάρχει δικαίωμα στην ανωνυμία του κάθε χρήστη .

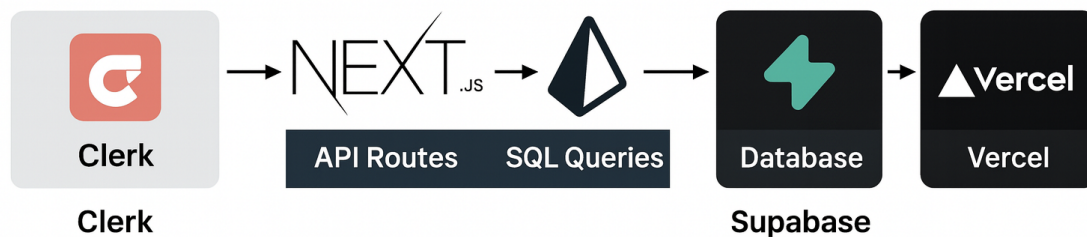
Το δικαίωμα στην ανωνυμία έρχεται σε σύγκρουση με τις τεχνολογίες blockchain καθώς αποτυπώνονται μέσω των smart contracts κάθε ψήφος μέσα σε μια ξεχωριστή συναλλαγή[15, 18]. Στην Ελλάδα, δεν υπάρχει ακόμα θεσμικό πλαίσιο για την διεξαγωγή των εκλογών με ψηφιακή μορφή αλλά υπάρχουν αυστηρές ρυθμίσεις για την ταυτοποίηση και τις διαδικασίες των εκλογών [12, 24].

4 Τεχνολογίες που χρησιμοποιήθηκαν στην εργασία

4.1 Εισαγωγή στις Τεχνολογίες

Για την δημιουργία της εφαρμογής ηλεκτρονικού συστήματος χρησιμοποιήθηκαν διάφορες σύγχρονες τεχνολογίες οι οποίες καλύπτουν το front end και το backend της . Η βασική τεχνολογία στην οποία βασίζεται ολόκληρη η εφαρμογή είναι το Next Js το οποίο επιτρέπει την δημιουργία εφαρμογών με server-side rendering (SSR) και static site generation (SSG). Επιπλέον στα πλαίσια της ασφαλούς ανάγνωσης τυπών χρησιμοποιήθηκε η γλώσσα προγραμματισμού Typescript ενώ για την διαμόρφωση της διεπαφής χρήστη της εφαρμογής χρησιμοποιήθηκε το Tailwind Css σε συνεργασία με κάποιες βιβλιοθήκες υποστήριξης κινούμενων εφέ .

Επιπλέον , χρησιμοποιήθηκαν διάφορες τεχνολογίες οι οποίες συνεισφέρουν στην απευθείας αυθεντικοποίηση των χρηστών , στην εγγραφή και σύνδεση και στην καλύτερη διαχείριση της βάσης δεδομένων . Για την εγγραφή των χρηστών χρησιμοποιήθηκε το Clerk ενώ η βάση δεδομένων υλοποιήθηκε με PostgreSQL σε συνδιασμό με το prisma ORM. Η επιλογή των παραπάνω τεχνολογιών έγινε επειδή καθιστούν σύγχρονες τεχνολογίες στις οποίες βασίζονται πολλαπλές επιχειρήσεις και προσφέρουν στον χρήστη ασφάλεια και απόδοση . Παρακάτω θα πραγματοποιηθεί αναλυτική περιγραφή όλων των τεχνολογιών που προαναφέρθηκαν μαζί με τα πλεονεκτήματα και τους λόγους επιλογής τους .



Σχήμα 3: Τεχνολογίες που έχουν χρησιμοποιηθεί στην υλοποίηση της εφαρμογής.

4.2 Next.js

Η τεχνολογία Next Js είναι ανοιχτού κώδικα και βασίζεται στη react [53] έχει σχεδιαστεί για την ανάπτυξη διαδικτυακών εφαρμογών για SSR και SSG[54]. Η λειτουργία του SSR σημαίνει ότι το περιεχόμενο της ιστοσελίδας δημιουργείται την ίδια στιγμή που ο χρήστης την επισκέπτεται. Αντίθετα το SSG σημαίνει ότι το περιεχόμενο της ιστοσελίδας έχει δημιουργηθεί από πριν[53, 54]. Έτσι η τεχνολογία του Next Js μπορεί να προσφέρει στατικές και δυναμικές σελίδες μαζί. Επιπλέον περιέχει ένα ενσωματωμένο σύστημα δρομολόγησης το οποίο βοηθάει ενεργά τον προγραμματιστή της ιστοσελίδας να οργανώσει καλύτερα τις σελίδες[53]. Μια ακόμα λειτουργία που προσφέρει είναι ότι δημιουργεί Api routes , δηλαδή μπορεί να αλληλεπιδρά με το backend χωρίς να χρειάζεται επιπλέον κάποιος ξεχωριστός διακομιστής . Με αυτήν την λειτουργία γίνεται ιδανικό για μικρές εφαρμογές ή εφαρμογές τύπου minimum viable products MVP επειδή ο προγραμματιστής τα δημιουργεί όλα στο ίδιο μέρος [52, 53]. Τέλος υποστηρίζει components τα οποία αλλάζουν με δυναμικό τρόπο και φορτώνει τις σελίδες από πριν για πιο γρήγορη εμπειρία. Επομένως το next js λόγω της ευελιξίας του επιλέγεται πολύ συχνά από τους προγραμματιστές[52].

NEXT.js	React
✓ Server-side rendering (SSR) ✓ Static export (SSG) ✓ Pre-rendering ✓ Automatic build size optimization ✓ Enhanced development compilation	✓ Declarative UI ✓ Component-Based Architecture ✓ Virtual DOM ✓ JSX (JavaScript XML) ✓ Unidirectional Data Flow ✓ Context API ✓ Server-Side Rendering (SSR) ✓ Large Ecosystem ✓ Strong Community Support ✓ Performance Optimization

Σχήμα 4: Σύγκριση τεχνικών χαρακτηριστικών: Next.js vs React .
[52, 53, 54]

4.3 TypeScript

Η γλώσσα typescript έχει αναπτυχθεί από τη Microsoft και αποτελεί μέρος της javascript με την διαφορά ότι χρησιμοποιεί στατική τυποποίηση [37, 39]. Με την στατική τυποποίηση οι προγραμματιστές καθορίζουν ρητά τους τύπους δεδομένων που χρησιμοποιούν στον κώδικα. Με αυτόν τον τρόπο μπορούν να ανιχνεύουν τα σφάλματα κατά το στάδιο της μεταγλώττισης [6]. Η συγκεκριμένη λειτουργία δημιουργεί μεγαλύτερη αξιοπιστία σε εφαρμογές με πολλές λειτουργίες και σύνθετες δομές δεδομένων όπως είναι το σύστημα ηλεκτρονικών ψηφοφοριών [6]. Η typescript χρησιμοποιεί τις δυνατότητες που προσφέρει η javascript. Μια από αυτές τις ιδιότητες η οποία χρησιμοποιήθηκε αρκετές φορές μέσα στην εφαρμογή ηλεκτρονικών ψηφοφοριών είναι το `async / await`. Με την συγκεκριμένη λειτουργία ο κώδικας γίνεται ασύγχρονος καθώς περιμένει να ολοκληρωθεί μια διαδικασία για να μπορέσει να τρέξει η επόμενη [37]. Για παράδειγμα αν απαιτείται να λάβεις δεδομένα από έναν server η συγκεκριμένη ιδιότητα απλοποιεί την διαδικασία συγκριτικά με τα `callbacks`. Το `async` μπροστά από την συνάρτηση σημαίνει ότι δουλεύει με ασύγχρονο τρόπο και μπορεί να περιμένει τα δεδομένα που θα έρθουν από το `await` [37]. Επιπλέον η typescript έχει πιο αυστηρή δομή από την javascript άρα αυτόματα βοηθάει τους προγραμματιστές εξ αρχής να αποφεύγουν πιθανά λάθη και κάνει την διαδικασία πιο απλή και πιο εύκολη για τον κώδικα στον οποίο μπορεί να υπάρχει και συνεργασία πολλών ταυτόχρονα προγραμματιστών [37]. Επιπρόσθετα διαθέτει το `type inference` που σημαίνει ότι η typescript από μόνη της μπορεί να καταλάβει τι τύπου είναι μια αναφορά χωρίς να χρειάζεται απαραίτητα η ρητή δήλωση της συγκεκριμένης μεταβλητής. Η typescript μπορεί και λειτουργεί χωρίς κανένα πρόβλημα στις βιβλιοθήκες της javascript [37]. Τέλος στην συγκεκριμένη εργασία του συστήματος ηλεκτρονικών ψηφοφοριών η γλώσσα typescript χρησιμοποιήθηκε τόσο στο front end όσο και στο back end της εφαρμογής. Σε συνδιασμό με το Next Js δημιουργήθηκαν ασφαλή διασυνδέσεις API με την χρήση τυποποιημένων παραμέτρων οι οποίες συμβάλουν στην αποφυγή σφαλμάτων κατά την σύνδεση με άλλες βιβλιοθήκες όπως το `prisma`, το `react` και το `clerk` [40].

4.4 Tailwind CSS

Το tailwind css είναι ένα σύγχρονο εργαλείο για την σχεδίαση των διεπαφών χρήστη UI το οποίο ανήκει στην κατηγορία των css με προτεραιότητα στην πρώτη κλάση [38]. Αυτό σημαίνει ότι σε αντίθεση με άλλα πλαίσια εργασίας όπως το Bootstrap που προσφέρουν στους προγραμματιστές έτοιμα κομμάτια κώδικα, το tailwind css προσφέρει μικρές κλάσεις css μέσα από τις οποίες μπορεί ο προγραμματιστής να ορίσει τα μεγέθη των παραθύρων, τα χρώματα τις γραμματοσειρές και άλλες κλάσεις css όπως είναι το "hover", "focus" ακόμα και την δημιουργία της σκοτεινής περιήγησης [38]. Ένα σημαντικό πλεονέκτημα που προσφέρει το tailwind css έναντι των υπολοίπων είναι το αρχείο tailwind.config.js στο οποίο οι προγραμματιστές μπορούν να δηλώσουν όλες τις κλάσεις για διάφορα θέματα που πρόκειται να χρησιμοποιήσουν στον κώδικα με τις αντίστοιχες ονομασίες που έχουν δοθεί [38]. Στην συγκεκριμένη εργασία το tailwind css χρησιμοποιήθηκε για την δημιουργία όλων των διεπαφών τόσο για τις στατικές δομές όσο και για τις δυναμικές ενσωματώσεις όπως το "tailwindcss -animate" με το οποίο μπορούν οι προγραμματιστές να προσθέσουν ζωντανές και διαδραστικές εικόνες. Συνολικά η συγκεκριμένη υλοποίηση με την τεχνολογία του tailwind css επέτρεψε την γρήγορη και καθαρή υλοποίηση της διεπαφής χρήστη του συστήματος ηλεκτρονικών ψηφοφοριών [38].

4.5 UI Components και Animation Libraries

Στο ηλεκτρονικό σύστημα ψηφοφοριών έχει χρησιμοποιηθεί ένα σύνολο βιβλιοθηκών και ζωντανών εικόνων, ώστε να γίνει η ιστοσελίδα περισσότερο διαδραστική ως προς τον χρήστη και να κάνει την διαδικασία του πιο ευχάριστη και εύκολη. Οι βασικές βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι Radix ui, lucide-react, framer-motion, και cmdk [41, 42, 43, 44]. Η Radix Ui προσφέρει σχεδιασμένα λειτουργικά στοιχεία τα οποία μπορεί ο προγραμματιστής να διαχειριστεί και να τα διαμορφώσει με την χρήση του tailwind css [38, 41]. Για τα εικονίδια της εφαρμογής χρησιμοποιήθηκε η βιβλιοθήκη Lucide-react η οποία περιέχει svg εικόνες [42]. Επιπλέον το framer motion δημιούργησε μια καλύτερη αισθητική στην εφαρμογή καθώς προστέθηκαν κινήσεις για να φαίνονται πολλά από τα κομμάτια της διαδραστικά. Έτσι έχει επιτευχθεί να γίνονται ομαλές κινήσεις με δυναμικό τρόπο [43]. Η cmdk χρησιμοποιήθηκε για την γρήγορη πλοήγηση μέσω της γραμμής εντολών καθώς βοηθάει ενεργά τον προγραμματιστή να αναζητάει και να εκτελεί γρήγορα εντολές μέσω του VSC [44]. Ο συνδιασμός όλων των παραπάνω δίνουν σαν αποτέλεσμα στον χρήστη του συστήματος ένα σύγχρονο, όμορφο και εύκολο ui για ένα σύστημα ηλεκτρονικών ψηφοφοριών.

4.6 State Management και Data Fetching

Για την δημιουργία διαδικτυακών εφαρμογών απαιτείται καλή διαχείριση της κατάστασης και λήψη δεδομένων ώστε να λειτουργούν αποτελεσματικά. Στην συγκεκριμένη πτυχική εργασία επιλέχθηκε η βιβλιοθήκη "@tanstack/react-query". Αυτή η βιβλιοθήκη αλλάζει τον τρόπο με τον οποίο οι προγραμματιστές διαχειρίζονται τα API. Ουσιαστικά η συγκεκριμένη βιβλιοθήκη αναλαμβάνει αυτόματα την διαχείριση των δεδομένων και την προσωρινή αποθήκευση πράγμα το οποίο συμβάλλει ενεργά στην καλύτερη απόδοση της εφαρμογής [45]. Στην εφαρμογή του συστήματος ηλεκτρονικών ψηφοφοριών χρησιμοποιήθηκε για την επικοινωνία με τα API του Next Js. Αφορά κυρίως τις σελίδες οι οποίες εμφανίζουν δυναμικά δεδομένα όπως οι λίστες ψηφοφοριών ή το ιστορικό των ψήφων. Με το useQuery τα δεδομένα φορτώνονται με ασφαλή τρόπο, παραμένουν ίδια ακόμα και στην ανανέωση της σελίδας και γίνεται αυτόματη ανανέωση

τους όταν χρειαστεί [45] . Η συγκεκριμένη βιβλιοθήκη λειτουργεί με την στρατηγική η κρυφή μνήμη να παρέχει τα δεδομένα στους χρήστες ακόμα και αν αυτά έχουν λήξει ενώ ταυτόχρονα στο παρασκήνιο γίνεται μια διαδικασία επανεπιβεβαίωσης για να ληφθούν και να ενημερωθούν τα πιο πρόσφατα στοιχεία [47] . Οι λειτουργίες με τις οποίες μεταβάλλονται τα δεδομένα για παράδειγμα στην υποβολή μαις ψήφου γίνεται με το useMutation το οποίο είναι ένα react hook το οποίο χρησιμοποιείται για την δημιουργία , ενημέρωση και διαγραφή δεδομένων . Με την βοήθεια του onSuccess και του onError το ui συγχρονίζεται αυτόματα με τον server . Επιπρόσθετα η βιβλιοθήκη προσφέρει προηγμένες δυνατότητες όπως πολλαπλά caching για αυτόματο καθαρισμό , ακύρωση αιτήσεων και διάφορα εργαλεία προγραμματιστή ώστε να μπορεί να πραγματοποιηθεί παρακολούθηση των δεδομένων [45, 46] .Συνολικά η βιβλιοθήκη react query βοήθησε στην απόδοση , την σταθερότητα και την βελτίωση της εφαρμογής συμβάλλοντας στην καλή διαχείριση των δεδομένων .

4.7 Διαχείριση Φορμών και Επικύρωση Δεδομένων

Κάθε εφαρμογή αντιμετωπίζει ένα σοβαρό πρόβλημα σχετικά με την διαχείριση των δεδομένων . Το ηλεκτρονικό σύστημα ψηφοφοριών έχει να αντιμετωπίσει το συγκεκριμένο πρόβλημα καθώς τα δεδομένα που εισάγονται από τους χρήστες αποτελούν ευαίσθητες πληροφορίες . Στην παρούσα εφαρμογή χρησιμοποιήθηκε η βιβλιοθήκη react hook form για την διαχείριση των διαφόρων φορμών καθώς μπορεί να προσφέρει μεγάλη απόδοση και απλότητα [48] . Η χρήση της συγκεκριμένης βιβλιοθήκης βοηθάει σημαντικά στην ταχύτητα της εφαρμογής αφού αποφεύγει την πλήρη επανεκτέλεση των διεπαφών . Επιπλέον έχει χρησιμοποιηθεί η zod ώστε να γίνεται σωστή επικύρωση των δεδομένων . Η συγκεκριμένη βιβλιοθήκη βασίζεται στην γλώσσα προγραμματισμού typescript και επιτρέπει την περιγραφή σχημάτων με απλό τρόπο [48]. Η σύνδεση με το react hook form γίνεται μέσω του @hookform/resolvers/zod πραγματοποιώντας ελέγχους με βάση το εκάστοτε σχήμα πριν τον έλεγχο της φόρμας [48]. έτσι και στην δημιουργία μιας νέας ψηφοφορίας , γίνεται έλεγχος αν έχουν συμπληρωθεί όλα τα απαραίτητα πεδία , αν οι ημερομηνίες είναι συμβατές και έγκυρες και αν οι τιμές είναι δηλωμένες με σωστούς τύπους δεδομένων .Σε περίπτωση κάποιου λάθους εμφανίζονται κατάλληλα μηνύματα στο ui . Ο συνδιασμός του react hook form και zod προσφέρουν ένα πλήρες ελεγχόμενο περιβάλλον για το σύστημα ηλεκτρονικών ψηφοφοριών .

4.8 Αυθεντικοποίηση και Διαχείριση Χρηστών

Για την αυθεντικοποίηση των χρηστών χρησιμοποιήθηκε το clerk το οποίο είναι μια πλατφόρμα αυθεντικοποίησης χρηστών για εφαρμογές [26]. Προσφέρει λειτουργίες όπως σύνδεσης και εγγραφής χρηστών με πολλαπλά επίπεδα ταυτοποίησης MFA. Για την σύνδεση με το Next Js χρησιμοποιήθηκε η βιβλιοθήκη @clerk/clerk-sdk-node ενώ για την προσθήκη της Ελληνικής γλώσσας χρησιμοποιήθηκε η βιβλιοθήκη @clerk/localization. Το clerk προσφέρει έτοιμα components για σύνδεση και εγγραφή [49] του χρήστη τα οποία οι προγραμματιστές μπορούν να ενσωματώσουν απευθείας στον κώδικα και μπορούν να δημιουργήσουν απευθείας τις φόρμες για σύνδεση και για εγγραφή τα οποία γλυτώνουν χρόνο από τον προγραμματιστή . Το clerk παρέχει την δυνατότητα προστασίας των χρηστών στο Next js με την ενσωμάτωση του αρχείου Middleware [53]το οποίο επιτρέπει την χρήση της εφαρμογής μόνο από πιστοποιημένους χρήστες . Η επιλογή του clerk έγινε καθώς μπορεί να υποστηριχθεί άμεσα η χρήση του μέσα απο το Next Js . Τέλος υποστηρίζει το clerk backend sdk . Το Clerk Backend SDK επιτρέπει

στον προγραμματιστή να καλεί και να διαχειρίζεται λειτουργίες αυθεντικοποίησης χρήστες συν-
νεδρίες και άλλο backend λογισμικό πρόγραμμα από περιβάλλοντα JavaScript στον διακομιστή
(server-side) [49, 50].

Create your account
Welcome! Please fill in the details to get started.

Google GitHub

or

First name Optional Last name Optional

Username

Email address

Phone number

US +1

Password

Continue ▶

Already have an account? [Sign in](#)

Secured by clerk

Σχήμα 5: Έτοιμη φόρμα για εγγραφή [49]

4.9 Βάση Δεδομένων και ORM: Prisma και PostgreSQL

Η Επιλογή της βάσης δεδομένων που θα χρησιμοποιηθεί σε ένα σύστημα είναι μια από τις πιο σοβαρές αποφάσεις καθώς απαιτεί να είναι αξιόπιστη επεκτάσιμη και αποδοτική. Για τον σκοπό αυτό επιλέχθηκε για βάση δεδομένων στο σύστημα ηλεκτρονικών ψηφοφοριών η PostgreSQL σε συνδιασμό με το ORM Prisma. Το PostgreSQL είναι ένα rdbms σύστημα ανοιχτού κώδικα το οποίο προσφέρει ασφάλεια και αξιοπιστία καθώς και όλες τις απαραίτητες λειτουργίες για εφαρμογές με υψηλές απαιτήσεις [51]. Παρακάτω θα γίνει αναλυτική περιγραφή του εργαλείου με το οποίο δημιουργήθηκε η βάση δεδομένων.

4.10 Εργαλεία Ανάπτυξης: Visual Studio Code, Supabase, Prisma και Vercel

Για να γίνει ανάπτυξη μιας σύγχρονης εφαρμογής web είναι κρίσιμο να επιλεγθούν εξ αρχής οι σωστές εφαρμογές οι οποίες θα βοηθήσουν στην υλοποίηση της. Για τον σκοπό αυτό και στα πλαίσια της πτυχιακής εργασίας δηλαδή το σύστημα ηλεκτρονικών ψηφοφοριών επιλέχθηκε να υλοποιηθεί στο Visual Studio Code το οποίο είναι ένας δωρεάν, εύχρηστος και με πολλές δυνατότητες επεξεργαστής κώδικα [30]. Το συγκεκριμένο εργαλείο υποστηρίζει αρκετές γλώσσες προγραμματισμού και μπορεί να προσαρμοστεί στις ανάγκες του κάθε προγραμματιστή καθώς υπάρχει μεγάλο πλήθος επεκτάσεων ώστε να καλύπτει τις ανάγκες όλων των προγραμματιστών που το χρησιμοποιούν οι οποίες προσθέτουν καινούργια χαρακτηριστικά και βελτιώνουν την διαδικασία της ανάπτυξης του κώδικα. Επιπλέον υπάρχει η επιλογή ζωντανής μετάδοσης ώστε να βλέπεις σε πραγματικό χρόνο όλες τις αλλαγές που κάνεις στον κώδικα σου καθώς και η αποσφαλμάτωση για να μπορεί ο προγραμματιστής να εντοπίζει γρήγορα κάποια λάθη. Παράλληλα προσφέρει άμεση σύνδεση με το git κάνοντας ευκολότερη την συνεργασία μεταξύ των προγραμματιστών και την δημιουργία εκδόσεων του προγράμματος. Συνοπτικά αποτελεί ένα από τα κορυφαία πολυεργαλεία τόσο για την ανάπτυξη μικρών εφαρμογών όσο και πιο σύνθετων και σε μεγάλο όγκο εφαρμογών [30].

Στην συνέχεια για το κομμάτι των βάσεων δεδομένων χρησιμοποιήθηκε το Supabase. Το Supabase παρέχει μια ολοκληρωμένη διεπαφή η οποία βασίζεται στην PostgreSQL. Παράλληλα προσφέρει μια ολοκληρωμένη βάση δεδομένων στο cloud με διάφορες επιπλέον ενέργειες όπως είναι η αυθεντικοποίηση και οι ενημερώσεις σε πραγματικό χρόνο [31].

Για την εγγραφή της βάσης δεδομένων του συστήματος ηλεκτρονικών ψηφοφοριών χρησιμοποιήθηκε το Prisma. Το συγκεκριμένο εργαλείο προσφέρει την δυνατότητα στους προγραμματιστές να δημιουργούν την βάση δεδομένων με χρήση Javascript ή Typescript χωρίς να χρειάζεται να γράψουν SQL κώδικα [33, 34, 40]. Χρησιμοποιήθηκε ως γέφυρα μεταξύ του προγράμματος και του Supabase και υλοποίησε τον κώδικα Sql που ήταν απαραίτητος στο σύστημα ηλεκτρονικών ψηφοφοριών. Τα πλεονεκτήματα στην συνδιαστική χρήση αυτών των δύο τεχνολογιών αφορούν τον χρόνο που κερδίζουν οι προγραμματιστές από την έτοιμη βάση σε PostgreSQL του Supabase μαζί με τα APIs και την ισχυρή μοντελοποίηση των δεδομένων που προσφέρει το Prisma απευθείας μέσα από τον κώδικα της εφαρμογής [31, 33].

Το τρίτο σημαντικό εργαλείο που χρησιμοποιήθηκε στην ανάπτυξη του συστήματος ηλεκτρονικών ψηφοφοριών είναι το Vercel. Πρόκειται για μια πλατφόρμα στην οποία μπορεί ο προγραμματιστής να κάνει deployment την εφαρμογή του και να δει απευθείας αν υπάρχουν προβλήματα στον κώδικα. Η χρήση του γίνεται με εύκολο τρόπο καθώς παίρνει τον κώδικα και όλα τα απαραίτητα αρχεία απευθείας μέσα από το Github [32]. Μετά την διαδικασία του

deployment προσφέρει ένα μοναδικό URL για την προεπισκόπηση της εφαρμογής . Επιπλέον μέσω της διεπαφής της διαχείρισης του προγραμματιστή γίνεται δυνατή η προβολή των προβλημάτων του συστήματος η παρακολούθηση της και η εναλλαγή σε κάποια προηγούμενη έκδοση [32].

Με απλά λόγια η επιλογή των συγκεκριμένων τεχνολογιών βοήθησαν στην ολοκληρωμένη ροή απο την αρχή έως το τέλος ολόκληρης της πτυχιακής εργασίας για την δημιουργία του συστήματος ηλεκτρονικών ψηφοφοριών .

4.11 Σύνοψη Κεφαλαίου

Σε αυτό το Κεφάλαιο πραγματοποιήθηκε η αναλυτική παρουσίαση των τεχνολογιών, βιβλιοθηκών και εργαλείων που χρησιμοποιήθηκαν για την υλοποίηση της εφαρμογής ψηφοφορίας. Στο frontend, η επιλογή του Next.js ως κύριου framework επέτρεψε την ανάπτυξη ενός σύγχρονου, δυναμικού και ταυτόχρονα στατικού ιστότοπου, με δυνατότητες server-side rendering, routing και API integration. Η χρήση της TypeScript ενίσχυσε την ασφάλεια και την επεκτασιμότητα του κώδικα, προσφέροντας αυστηρή τυποποίηση και αυτοσυμπλήρωση.

Για το οπτικό και διαδραστικό μέρος, αξιοποιήθηκε το Tailwind CSS μαζί με βιβλιοθήκες όπως οι Radix UI, Lucide-react, framer-motion και cmdk, ενισχύοντας την προσβασιμότητα, την αισθητική και την εμπειρία χρήστη. Η διαχείριση δεδομένων υλοποιήθηκε με τη χρήση του React Query, ενώ η διαχείριση φορμών και η επικύρωση δεδομένων πραγματοποιήθηκε με τις react-hook-form και zod.

Για τις λειτουργίες αυθεντικοποίησης και διαχείρισης χρηστών επιλέχθηκε το Clerk, το οποίο πρόσφερε μια ασφαλή και έτοιμη προς χρήση λύση ενσωματωμένη με το Next.js. Η αποθήκευση των δεδομένων υλοποιήθηκε με χρήση της βάσης PostgreSQL, ενώ το Prisma ORM αποτέλεσε τον κύριο μηχανισμό επικοινωνίας με τη βάση, προσφέροντας ευκολία ανάπτυξης, ασφάλεια και βελτιστοποιημένα queries.

Παράλληλα, χρησιμοποιήθηκαν βοηθητικές βιβλιοθήκες όπως η date-fns για τη διαχείριση ημερομηνιών, η uuid για δημιουργία μοναδικών ταυτοτήτων και η dotenv για τη φόρτωση μεταβλητών περιβάλλοντος. Συνολικά, η επιλογή των τεχνολογιών έγινε με κριτήριο τη σταθερότητα, την απόδοση, την ασφάλεια και τη δυνατότητα διασύνδεσης μεταξύ των επιμέρους συστατικών της εφαρμογής.

Στο Κεφάλαιο 5 που ακολουθεί, παρουσιάζεται αναλυτικά η διαδικασία ανάπτυξης της εφαρμογής, καθώς και τα επιμέρους στάδια υλοποίησης του έργου.

5 Τεχνική Υλοποίηση της Εφαρμογής

5.1 Εισαγωγή στην Υλοποίηση της Εφαρμογής

Σε αυτό το κεφάλαιο γίνεται αναλυτική περιγραφή του συστήματος ηλεκτρονικών ψηφοφόρων. Η εφαρμογή έχει σαν στόχο την δημιουργία κάθε είδους τύπου ψηφοφορίας με ασφάλεια και διαφάνεια σε ένα σύγχρονο περιβάλλον προς τον χρήστη. Το front end της εφαρμογής υλοποιήθηκε με το Next Js σε συνεργασία με την react και το backend με το node js σε συνεργασία με το prisma και τα rest Api . Αρχικά αναλύονται οι απαιτήσεις ενός τέτοιου συστήματος σχετικά με την δημιουργία και διαχείριση ψηφοφόρων την επαλήθευση των υποψηφίων και την καταγραφή των ψηφών . Στη συνέχεια γίνεται περιγραφή των διασυνδέσεων μεταξύ των components και των restful apis . Ακολουθεί αναλυτική περιγραφή της βάσης δεδομένων η οποία περιέχει τις οντότητες χρήστες , ψηφοφορίες και ψήφους . Τέλος γίνεται αναφορά στην διαδικασία της πλοήγησης από την πλευρά του χρήστη στο σύστημα ηλεκτρονικών ψηφοφοριών το οποίο επιτρέπει την εγγραφή του , την δημιουργία κάποιας ψηφοφορίας , την προσθήκη υποψηφίων , την υποβολή ψήφων και την προβολή των αποτελεσμάτων .

5.2 Λειτουργικές Απαιτήσεις

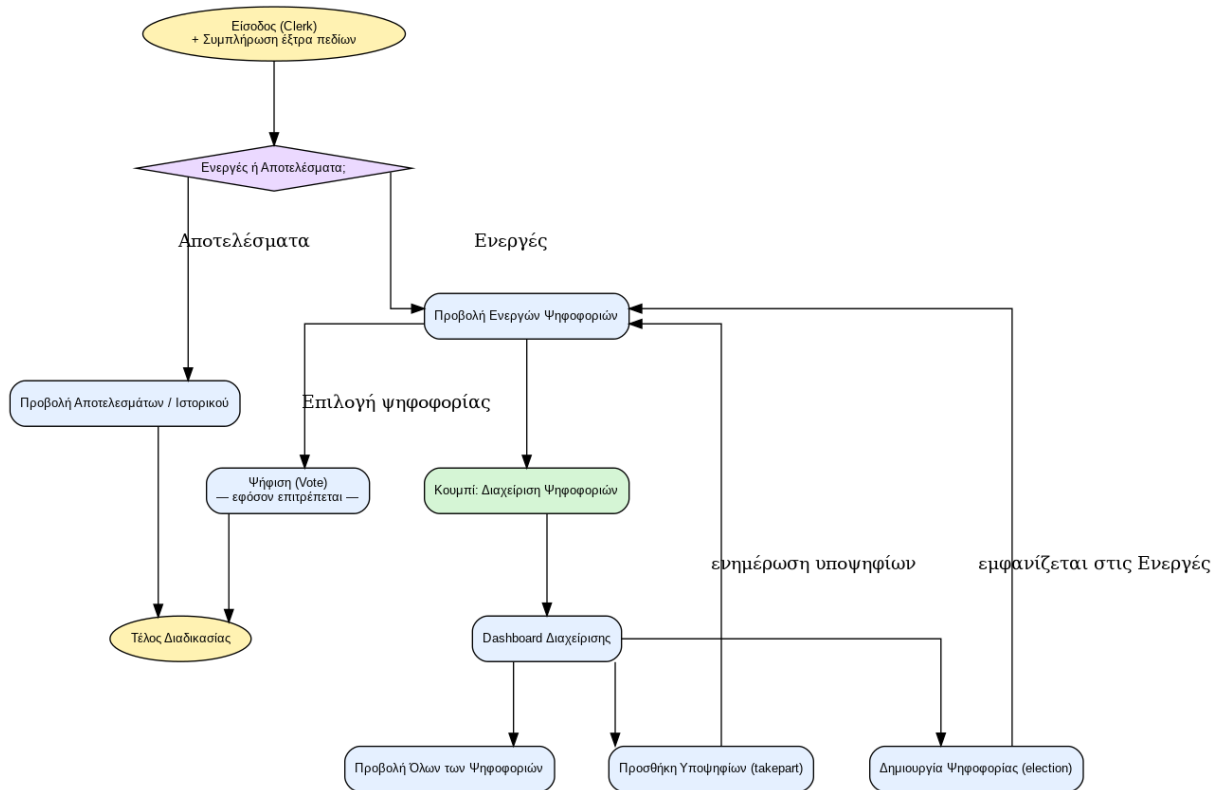
Σε αυτό το υποκεφάλαιο παρουσιάζονται αναλυτικά όλες οι λειτουργικές απαιτήσεις του συστήματος ηλεκτρονικών ψηφοφοριών με μια συνοπτική περιγραφή των λειτουργιών που προσφέρει το σύστημα. Στα πληροφοριακά συστήματα ο όρος απαιτήσεις αναφέρεται σαν μια πληροφορία η οποία περιγράφει τις δυνατότητες ενός λειτουργικού συστήματος με σκοπό να επιλύει διάφορα προβλήματα[22]. Ο διαχωρισμός αυτών των απαιτήσεων χωρίζεται στις λειτουργικές και στις ποιοτικές. Οι λειτουργικές απαιτήσεις παρουσιάζουν τις ενέργειες που πρέπει να εκτελεί ένα σύστημα ώστε να θεωρείται λειτουργικό. Αντίθετα, οι ποιοτικές αφορούν διάφορες τεχνολογίες οι οποίες κάνουν το σύστημα περισσότερο ασφαλές και πιο αποδοτικό[23].

Για την καλύτερη κατανόηση οι λειτουργικές απαιτήσεις μπορούν να κατηγοριοποιηθούν με τον εξής τρόπο :

- **Είσοδοι του συστήματος :** περιλαμβάνουν όλα τα δεδομένα της εφαρμογής τα οποία εισάγονται. Για την καλύτερη κατανόηση οι λειτουργικές απαιτήσεις μπορούν να κατηγοριοποιηθούν με τον εξής τρόπο :
- **Έξοδοι του συστήματος :** Αφορούν όλα τα αποτελέσματα που προβάλλονται στους χρήστες της εφαρμογής όπως είναι η λίστα των ψηφοφοριών και τα αποτελέσματα της κάθε ψηφοφορίας.
- **Αποθηκευμένα δεδομένα:** αφορούν όλα τα δεδομένα που μπορεί να περιέχει μια εφαρμογή . Για παράδειγμα οι χρήστες , οι ψηφοφορίες και οι υποψήφιοι.
- **Ενέργειες και υπολογισμοί:** Αφορούν τις απαραίτητες ενέργειες που πρέπει να γίνουν ώστε η εφαρμογή να είναι αξιόπιστη και ασφαλής . περιλαμβάνουν την ταυτοποίηση των χρηστών και την ορθή καταμέτρηση των ψήφων .
- **Συγχρονισμός δεδομένων :** Αφορά την άμεση ανταπόκριση όλων των ενεργειών στην διεπαφή του χρήστη . Για παράδειγμα η προσθήκη μιας νέας ψηφοφορίας να εμφανίζεται απευθείας στον χρήστη .

Στο σύστημα ηλεκτρονικών ψηφοφοριών οι είσοδοι αφορούν δεδομένα τα οποία εισέρχονται από τους χρήστες οι οποίοι θεωρούνται και διαχειριστές καθώς μπορούν να δημιουργήσουν και να διαγράψουν ψηφοφορίες .

Οι έξοδοι του συστήματος αφορούν την εμφάνιση των ψηφοφοριών μαζί με τους υποψηφίους στην αρχική σελίδα του χρήστη καθώς και τα αποτελέσματα κάθε ψηφοφορίας .



Σχήμα 6: Διάγραμμα Οντοτήτων και Συσχετίσεων (ER Diagram)

5.3 Σχεδίαση της Βάσης Δεδομένων

5.3.1 User

Ένα από τα σημαντικότερα αντικείμενα είναι το **User** και έχει δημιουργηθεί ώστε να καταγράφει όλες τις απαραίτητες πληροφορίες των χρηστών του συστήματος ηλεκτρονικών ψηφοφοριών. Η δομή του συγκεκριμένου αντικείμενου ακολουθεί τον διαχωρισμό του αντικείμενου σε δύο ενότητες ώστε ο προγραμματιστής να μπορεί να εστιάζει είτε στα βασικά του στοιχεία από την αρχική εγγραφή του χρήστη στο σύστημα είτε στα στοιχεία τα οποία συμπληρώνει αργότερα ο χρήστης ώστε να μπορεί να λάβει μέρος σε ψηφοφορίες . Η πρώτη ενότητα αφορά τα στοιχεία που λαμβάνονται από το clerk δηλαδή το ονοματεπώνυμο του , το αναγνωριστικό του clerk , το username και τον κώδικό του . Η δεύτερη ενότητα αφορά τα στοιχεία που συμπληρώνει ο χρήστης κατά την εγγραφή του στην εφαρμογή δηλαδή η ηλικία του , η τοποθεσία , το επάγγελμα και το φύλο .

Το αναγνωριστικό πεδίο id αποτελεί το πρωτεύον κλειδί για τον κάθε ένα χρήστη το οποίο δημιουργείται αυτόματα από το σύστημα ώστε να εξασφαλίζονται τα δεδομένα . Τα πεδία username , email και clerkId δεν μπορούν να επαναληφθούν , ενώ τα πεδία gender , location , birthday , occupation και location μπορούν να είναι κενά χωρίς να επηρεάζεται η λειτουργικότητα της εφαρμογής . Το αντικείμενο **User** δημιουργείται στην βάση δεδομένων μέσω του αρχείου **schema.prisma**. Οι λειτουργίες που προσφέρει στην εφαρμογή είναι να ελέγχει τα δικαιώματα των χρηστών , να ελέγχεται η στόχευση των χρηστών στις ανάλογες ψηφοφορίες δηλαδή στις ψηφοφορίες οι οποίες έχουν δημιουργηθεί και έχουν ως φίλτρο συγκεκριμένα στοιχεία . Επιπλέον χρησιμοποιείται για την αποφυγή της διπλής ψήφου και για τον υπολογισμό των ψήφων που έλαβε και έδωσε ο χρήστης .

5.3.2 Election

Δεύτερο σημαντικό αντικείμενο είναι το **Election** και έχει δημιουργηθεί με σκοπό να καταγράφει τις ψηφοφορίες που δημιουργούνται από τους χρήστες και όλες τις πληροφορίες για κάθε εκλογική διαδικασία. Τα πεδία που έχει σαν παράμετρους το συγκεκριμένο αντικείμενο είναι η καταγραφή του τίτλου της ψηφοφορίας , της περιγραφής της σχετικά με το τι είναι και σε ποιο κοινό απευθύνεται , την ημερομηνία έναρξης καθώς και την ημερομηνία λήξης της κάθε ψηφοφορίας , την κατάσταση στην οποία βρίσκεται η ψηφοφορία αν είναι ενεργή ή έχει ολοκληρωθεί . Επιπλέον περιέχει τις τιμές οι οποίες είναι υπεύθυνες για το κοινό το οποίο αφορά η κάθε ψηφοφορία και ο διαχωρισμός γίνεται με το φύλο , την τοποθεσία , το επάγγελμα και το εύρος των ηλικιών . Τέλος αποθηκεύεται το αναγνωριστικό του χρήστη και ένα ξεχωριστό αναγνωριστικό id το οποίο χρησιμοποιείται ως πρωτεύον κλειδί για κάθε μια ψηφοφορία η οποία δημιουργείται από τους χρήστες . Το αντικείμενο **Election** δημιουργείται εξίσου μέσω του αρχείου **schema.prisma** το οποίο μετράτρέπει τον κώδικα javascript σε κώδικα sql .

5.3.3 PollCandidates

Τρίτο σημαντικό αντικείμενο είναι το **Poll Candidates** και έχει δημιουργηθεί με σκοπό να καταγράφει την προσθήκη κάποιου υποψηφίου σε μια ψηφοφορία . Τα πεδία που έχει σαν παραμέτρους το συγκεκριμένο αντικείμενο είναι το αναγνωριστικό πεδίο id της ψηφοφορίας στην οποία ενδιαφέρεται ο χρήστης να προσθέσει έναν υποψήφιο , το αναγνωριστικό πεδίο του υποψηφίου τον οποίο θέλει να προσθέσει και την ακριβής χρονική στιγμή στην οποία πραγματοποιείται η προσθήκη. Επιπλέον περιέχει το δικό του μοναδικό αναγνωριστικό id το οποίο αποτελεί το

πρωτεύον κλειδί του αντικειμένου . Η εγγραφή στο συγκεκριμένο αντικείμενο μπορεί να πραγματοποιηθεί μια μόνο φορά για τον κάθε έναν υποψήφιο . Οι λειτουργίες του συγκεκριμένου αντικειμένου είναι να λειτουργεί σαν γέφυρα μεταξύ των ψηφοφοριών και των υποψηφίων , να υπάρχει η καταγραφή σχετικά με το πότε πραγματοποιήθηκε η συγκεκριμένη ενέργεια και μπορεί να συμβάλλει στον έλεγχο αν ένας χρήστης έχει προστεθεί σε κάποια συγκεκριμένη ψηφοφορία .

5.3.4 Takepart

Τέταρτο σημαντικό αντικείμενο είναι το **TakePart** και έχει δημιουργηθεί με σκοπό να καταγράφει τις ψηφοφορίες στις οποίες έχει πάρει μέρος ένας υποψήφιος και τους ψήφους που έχει λάβει σε κάθε μια ψηφοφορία . Στον συγκεκριμένο πίνακα το πρωτεύον κλειδί του είναι σύνθετο με τις τιμές **ElectionId** για μια συγκεκριμένη ψηφοφορία και **CandidateId** για έναν συγκεκριμένο υποψήφιο ώστε να διασφαλίζεται ότι ο κάθε χρήστης μπορεί να συμμετέχει στην ίδια ψηφοφορία το πολύ μια φορά . Συνοπτικά το αντικείμενο επιτυγχάνει μοναδική συμμετοχή υποψηφίου σε ψηφοφορία και αποθηκεύει τις ψήφους που έχει λάβει.

5.3.5 Vote

Το τελευταίο σημαντικό αντικείμενο του συστήματος ηλεκτρονικών ψηφοφοριών είναι το **Vote** και έχει δημιουργηθεί με σκοπό να καταγράφει τις ψήφους που έχουν δώσει οι χρήστες στους υποψήφιους που επιθυμούν σε συγκεκριμένες ψηφοφορίες .Με πιο απλό τρόπο δείχνει αναλυτικά το ποιος χρήστης ψήφισε ποιον υποψήφιο σε ποια ψηφοφορία . Τα πεδία που έχει το συγκεκριμένο αντικείμενο είναι το αναγνωριστικό του χρήστη **UserId** , το αναγνωριστικό της ψηφοφορίας **ElectionId** , το αναγνωριστικό του υποψηφίου **CandidateId** και τον ακριβή χρόνο στον οποίο πραγματοποιήθηκε η ψήφος **VoteAt** . Επιπλέον έχει το δικό του μοναδικό αναγνωριστικό **id** το οποίο αποτελεί και το πρωτεύον κλειδί του αντικειμένου .

Σημειώσεις σχεδίασης

Ολά τα αντικείμενα δημιουργούνται στο αρχείο **schema.prisma** μέσω του κώδικα το οποίο μετατρέπεται μέσω του **prisma Api** από κώδικα javascript σε κώδικα **Sql** και δημιουργεί τον πίνακα στο **Supabase** . Η χρήση **onDelete: Cascade** σε κάποιες κρίσιμες σχέσεις οι οποίες έχουν περιορισμούς μέσα στην εφαρμογή όπως έχουν οι πίνακες **Takepart** και **Vote** βοηθούν στην ακεραιότητα δεδομένων και παράλληλα ενσωματώνουν διάφορους κανόνες στην εφαρμογή.

5.3.6 Παραδείγματα SQL αιτημάτων σημαντικά για την εφαρμογή.

```
1 SELECT
2   tp."candidateId"      AS "candidateId",
3   u."fullName"          AS "candidateName",
4   tp."numberOfVotes"    AS "votes"
5 FROM "takepart" tp
6 JOIN "user" u
7   ON u."id" = tp."candidateId"
8 WHERE tp."electionId" = $1
9 ORDER BY tp."numberOfVotes" DESC;
```

Το query επιστρέφει όλους τους υποψηφίους μιας εκλογής μαζί με τις ψήφους τους, ταξινομημένους κατά φθίνουσα σειρά.

Listing 1: Αποτελέσματα ψηφοφορίας

```
1 UPDATE "takepart"
2 SET "numberOfVotes" = "numberOfVotes" + 1
3 WHERE "electionId" = $1
4   AND "candidateId" = $2;
```

Αυξάνει τον μετρητή ψήφων για συγκεκριμένο υποψήφιο σε συγκεκριμένη εκλογή.

Listing 2: Καταχώριση ψήφου

```

1 SELECT
2   e."id",
3   e."title",
4   e."description",
5   e."start_date",
6   e."end_date",
7   e."is_active",
8   e."target_occupation",
9   e."target_location",
10  e."birthdate_min",
11  e."birthdate_max",
12  e."target_gender",
13  e."userId"
14 FROM "election" e
15 WHERE
16  (e."target_gender" = $1 OR e."target_gender" IS NULL OR e."target_gender" =
17   ⇨ 'all')
18  AND (e."target_occupation" = $2 OR e."target_occupation" IS NULL OR
19   ⇨ e."target_occupation" = 'all')
20  AND (e."target_location" = $3 OR e."target_location" IS NULL OR
21   ⇨ e."target_location" = 'all')
22  AND ($4::timestamp IS NULL OR e."birthdate_min" <= $4)
23  AND ($4::timestamp IS NULL OR e."birthdate_max" >= $4)
24 ORDER BY e."start_date" DESC;

```

Φέρνει εκλογές που ταιριάζουν στα χαρακτηριστικά του χρήστη (φύλο, ρόλος, τοποθεσία, ηλικία).

Listing 3: Φιλτράρισμα εκλογών

```

1 SELECT
2   u."id"           AS "id",
3   u."fullName"     AS "name",
4   u."email"        AS "email",
5   u."occupation"   AS "occupation",
6   tp."numberOfVotes" AS "numberOfVotes"
7 FROM "takepart" tp
8 JOIN "user" u
9   ON u."id" = tp."candidateId"
10 WHERE tp."electionId" = $1;

```

Επιστρέφει όλους τους υποψηφίους μιας εκλογής με τα στοιχεία τους και τον αριθμό ψήφους.

Listing 4: Υποψήφιοι εκλογής με ψήφους

```

1 -- 5a) Π      (poll_candidates)
2 INSERT INTO "poll_candidates" ("poll_id", "user_id")
3 VALUES ($1, $2);
4
5 -- 5b) Upsert      (takepart)      0
6 INSERT INTO "takepart" ("electionId", "candidateId", "numberOfVotes")
7 VALUES ($1, $2, 0)
8 ON CONFLICT ("electionId", "candidateId") DO NOTHING;

```

Εισάγει υποψήφιο σε εκλογή (στην poll candidates και στην takepart με upsert).

Listing 5: Προσθήκη υποψηφίου σε εκλογή

```

1 DELETE FROM "takepart"
2 WHERE "electionId" = $1 AND "candidateId" = $2;
3
4 DELETE FROM "poll_candidates"
5 WHERE "poll_id" = $1 AND "user_id" = $2;

```

Διαγράφει έναν υποψήφιο από εκλογή (και από poll candidates και από takepart).

Listing 6: Αφαίρεση υποψηφίου

```

1 SELECT
2   e."id",
3   e."title",
4   e."description",
5   e."start_date",
6   e."end_date",
7   e."is_active",
8   e."target_occupation",
9   e."target_location",
10  e."target_gender",
11  e."birthdate_min",
12  e."birthdate_max",
13  e."userId"
14 FROM "election" e
15 ORDER BY e."start_date" DESC;

```

Επιστρέφει όλες τις εκλογές ταξινομημένες κατά ημερομηνία.

Listing 7: Λίστα όλων των εκλογών

```
1 SELECT
2   e."id",
3   e."title",
4   e."description",
5   e."start_date",
6   e."end_date",
7   e."is_active",
8   e."target_occupation",
9   e."target_location",
10  e."target_gender",
11  e."birthdate_min",
12  e."birthdate_max",
13  e."userId"
14 FROM "election" e
15 WHERE e."userId" = $1
16 ORDER BY e."start_date" DESC;
```

Φέρνει τις εκλογές που έχει δημιουργήσει ο χρήστης (owner).

Listing 8: Εκλογές που δημιούργησε ο χρήστης

```
1 SELECT
2   pc."id",
3   pc."poll_id",
4   pc."user_id",
5   pc."invited_at",
6   u."fullName",
7   u."email",
8   u."occupation",
9   u."id" AS "candidateId"
10 FROM "poll_candidates" pc
11 JOIN "user" u
12   ON u."id" = pc."user_id"
13 WHERE pc."poll_id" = $1;
```

Δίνει όλους τους υποψηφίους μιας εκλογής με join στα στοιχεία χρήστη.

Listing 9: Λίστα προσκεκλημένων υποψηφίων

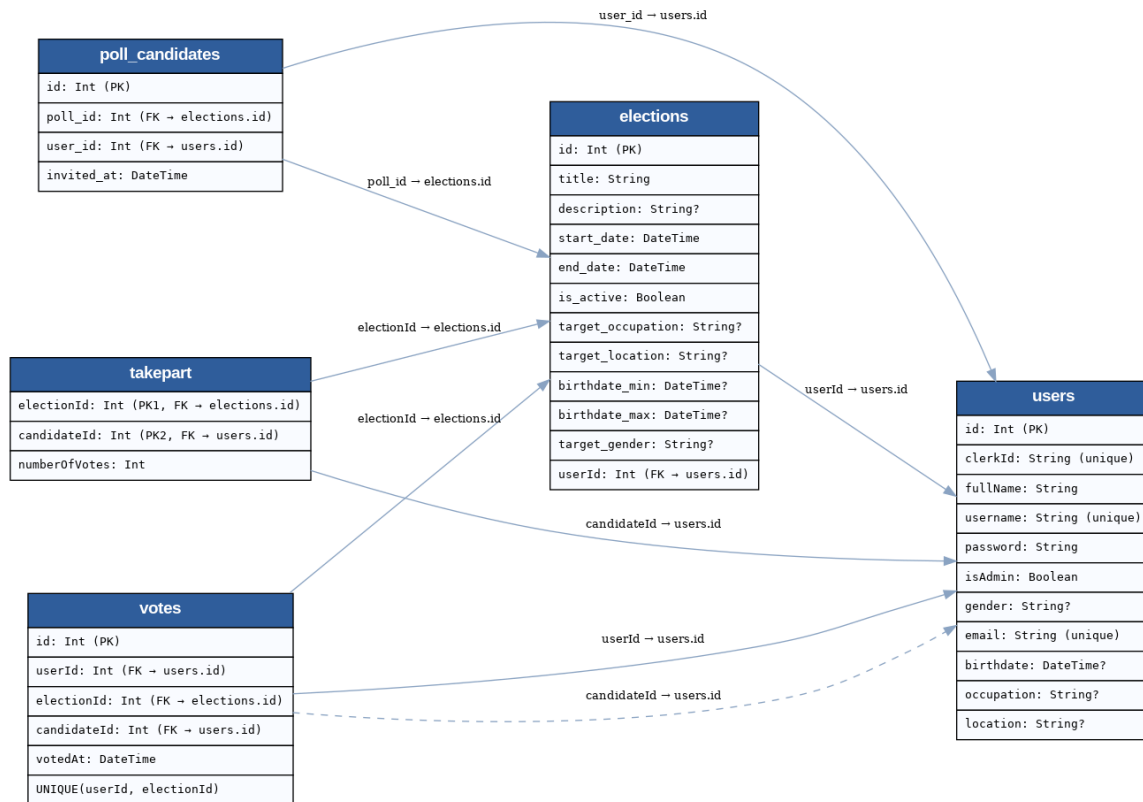
```
1 SELECT
2   u."id",
3   u."fullName",
4   u."email",
5   u."occupation",
6   u."location",
7   u."gender"
8 FROM "user" u
9 WHERE
10  ($1 IS NULL)
11 OR (u."fullName" ILIKE '%' || $1 || '%')
12 OR u."email" ILIKE '%' || $1 || '%')
13 OR u."occupation" ILIKE '%' || $1 || '%');
```

Αναζήτηση με βάση όνομα, email ή ειδικότητα (case-insensitive).

Listing 10: Αναζήτηση χρηστών

5.3.7 Συσχετίσεις Μεταξύ Αντικειμένων (Relationships)

Οι συσχετίσεις των αντικειμένων της εφαρμογής γίνονται μέσω των πεδίων foreign key, όπως γίνεται στις περισσότερες βάσεις δεδομένων. Ουσιαστικά οι συσχετίσεις αυτές λειτουργούν σαν μια γέφυρα με την οποία μπορούμε να ενώσουμε κάποιες εγγραφές .



Σχήμα 7: Διάγραμμα Οντοτήτων και Συσχετίσεων (ER Diagram)

Οι βασικοί τύποι συσχετίσεων που συναντώνται είναι:

- Ένα προς Ένα (1:1): Κάθε εγγραφή ενός αντικειμένου μπορεί να συσχετιστεί με μια μόνο εγγραφή ενός άλλου αντικειμένου [35].
- Ένα προς Πολλά (1:N): Μια εγγραφή για ένα αντικείμενο μπορεί να συσχετιστεί με πολλές εγγραφές από ένα άλλο αντικείμενο [36].
- Πολλά προς Ένα (N:1): Πολλές εγγραφές από ένα αντικείμενο μπορούν να συσχετιστούν με μια εγγραφή ενός άλλου αντικειμένου [35].
- Πολλά προς Πολλά (N:M): Πολλές εγγραφές από ένα αντικείμενο σχετίζονται με πολλές εγγραφές από ένα άλλο αντικείμενο [35].

Στην συγκεκριμένη εφαρμογή του συστήματος ηλεκτρονικών ψηφοφοριών οι αντιστοιχίες των αντικειμένων είναι οι εξής:

- Elections – Poll_Candidates (1:N) - Περιγραφή Τύπου: Κάθε μια από τις ψηφοφορίες μπορεί να περιλαμβάνει πολλούς υποψηφίους ενώ κάθε υποψήφιος ανήκει σε μια συγκεκριμένη ψηφοφορία.

- Users – Poll_Candidates (1:N) - Περιγραφή Τύπου: Κάθε χρήστης μπορεί να λάβει μέρος σαν υποψήφιος σε πολλές ψηφοφορίες ενώ κάθε υποψήφιος συνδέεται με έναν χρήστη.
- Elections – Votes (1:N) - Περιγραφή Τύπου: Κάθε μια από τις ψηφοφορίες μπορεί να πάρει πολλούς ψήφους ενώ κάθε ψήφος ανήκει σε μια μόνο ψηφοφορία.
- Users – Votes (1:N) - Περιγραφή Τύπου: Κάθε χρήστης συμμετέχει σε 1 η περισσότερες ψηφοφορίες και καταχωρεί πολλές ψήφους ενώ κάθε ψήφος συνδέεται με έναν χρήστη .
- Elections – Takepart (1:1) - Περιγραφή Τύπου: Κάθε ψηφοφορία έχει 1 αποτέλεσμα και κάθε αποτέλεσμα συνδέεται με μια συγκεκριμένη ψηφοφορία.
- Users – Elections (1:N) - Περιγραφή Τύπου: Κάθε χρήστης μπορεί να δημιουργήσει πολλές ψηφοφορίες ενώ η κάθε μια ψηφοφορία δημιουργείται από έναν χρήστη .

6 Περιγραφή των API του Συστήματος

6.1 API Endpoints

Η χρήση των Api είναι απαραίτητη για την δημιουργία του συστήματος ηλεκτρονικών ψηφοφοριών . Τα api endpoints αποτελούν τον ενδιάμεσο κρίκο για όλες τις πληροφορίες που χρειάζεται η υλοποίηση του συγκεκριμένου συστήματος . Υποστηρίζουν όλες τις λειτουργίες όπως είναι η είσοδος και έξοδος του χρήστη στο σύστημα , η επεξεργασία των στοιχείων του ώστε να μπορεί να λάβει μέρος στις ψηφοφορίες , η δημιουργία ψηφοφοριών και η προσθήκη υποψηφίων σε αυτές καθώς και η εμφάνιση των ψηφοφοριών και των αποτελεσμάτων τους . Παρακάτω θα δούμε αναλυτική περιγραφή όλων των api endpoints που χρησιμοποιούνται στο ηλεκτρονικό σύστημα ψηφοφοριών .

6.2 GET /api/elections

Στο συγκεκριμένο api endpoint γίνεται η επιστροφή όλων των ψηφοφοριών του συστήματος . Το json αρχείο που επιστρέφεται περιλαμβάνει τις ψηφοφορίες μαζί με τα στοιχεία των υποψηφίων και επιστρέφονται σε ταξινομημένη σειρά με παράγοντα την ημερομηνία έναρξης της ψηφοφορίας .

```
{
  {
    "id": 1,
    "title": "Εκλογές Διαδραστικού Συστήματος",
    "description": "Εκλογές για το μάθημα Διαδραστικού Συστήματος",
    "dateRange": {
      "startDate": "2024-01-15T00:00:00.000Z",
      "endDate": "2024-01-20T23:59:59.000Z"
    },
    "targeting": {
      "roles": ["student"],
      "locations": ["Athens"]
    },
    "candidates": [...],
    "createdAt": "2024-01-15T00:00:00.000Z",
    "isActive": true
  }
}
```

Σχήμα 8: Παράδειγμα επιστροφής του Api: GET /api/elections σε μορφή JSON

6.3 POST /api/elections

Σε αυτό το api endpoint γίνεται η δημιουργία μιας καινούργιας ψηφοφορίας από έναν χρήστη . Για να μπορέσει ο χρήστης να δημιουργήσει μια ψηφοφορία πρέπει να έχει συμπληρώσει αρχικά τα στοιχεία του ώστε να του επιτρέπεται η είσοδος στην σελίδα του διαχειριστή των ψηφοφοριών . Έτσι ο χρήστης πρέπει να ορίσει τον τίτλο της ψηφοφορίας , την περιγραφή της , τις ημερομηνίες στις οποίες θα είναι ενεργή η ψηφοφορία δηλαδή ημερομηνία έναρξης και λήξης και τα κριτήρια στόχευσης της ψηφοφορίας τα οποία είναι το φύλο , το επάγγελμα , η τοποθεσία και η ηλικία . Εφόσον έχει ολοκληρώσει την εγγραφή όλων των παραπάνω γίνεται η καταχώριση της νέας ψηφοφορίας στην βάση με ένα post αίτημα και στην συνέχεια μπορεί να γίνει και η επιστροφή της μέσα από το get αίτημα απο την βάση δεδομένων της εφαρμογής .

```
{
  "title": "Εκλογές Διαδραστικού Συστήματος",
  "description": "Εκλογές για το μάθημα Διαδραστικού Συστήματος",
  "startDate": "2024-01-15T00:00:00.000Z",
  "endDate": "2024-01-20T23:59:59.000Z",
  "targetOccupation": "student",
  "targetLocation": "Athens",
  "targetGender": "all",
  "birthdateMin": "1990-01-01T00:00:00.000Z",
  "birthdateMax": "2005-12-31T23:59:59.000Z"
}
```

Σχήμα 9: Παράδειγμα επιστροφής του Api: **POST /api/elections** σε μορφή **JSON**

6.4 GET /api/elections/[id]

Αυτό το endpoint επιστρέφει όλες τις τιμές για μια συγκεκριμένη ψηφοφορία με κριτήριο αναζήτησης το id που στέλνεται μέσω του url .Στην επιστροφή του συγκεκριμένου json βλέπουμε τις τιμές που αφορούν τον υποψήφιο , τους ψήφους που έχει λάβει καθώς και όλες τις τιμές της κάθε μιας ψηφοφορίας .

```
{
  "id": 1,
  "title": "Εκλογές Διαδραστικού Συστήματος",
  "description": "...",
  "dateRange": {...},
  "targeting": {...},
  "candidates": [
    {
      "id": 1,
      "name": "Γιώργος Παπαδόπουλος",
      "email": "giorgos@example.com",
      "occupation": "student",
      "numberOfVotes": 15
    }
  ],
  "createdAt": "...",
  "isActive": true
}
```

Σχήμα 10: Παράδειγμα επιστροφής του Api: **GET /api/elections/[id]** σε μορφή **JSON**

6.5 DELETE /api/elections/[id]

Το Delete api χρησιμοποιείται όταν ο χρήστης επιθυμεί να διαγράψει μια συγκεκριμένη ψηφοφορία την οποία έχει δημιουργήσει ο ίδιος .Στο αρχείο schema.prisma υπάρχει το πεδίο userId.Αυτό το πεδίο έχει δημιουργηθεί ώστε να μπορεί το σύστημα να ξέρει ποιος είναι ο δημιουργός της κάθε ψηφοφορίας για να μην μπορεί να διαγράψει κάποιος χρήστης οποιαδήποτε άλλη ψηφοφορία πέρα απο την δικιά του .

```

model election {
  id          Int          @id @default(autoincrement())
  title       String
  description  String?
  start_date  DateTime
  end_date    DateTime
  is_active   Boolean
  target_occupation String?
  target_location String?
  birthdate_min DateTime?
  birthdate_max DateTime?
  target_gender String?
  userId      Int          // ← Αυτό το πεδίο δείχνει τον δημιουργό

  user        user          @relation(fields: [userId], references: [id])
  poll_candidates poll_candidates[]
  takepart    takepart[]
  votes       Vote[]

  @@index([userId])
}

```

Σχήμα 11: Παράδειγμα **Database Schema** σε μορφή **schema.prisma**

Στο **βήμα 1** παίρνουμε το αναγνωριστικό του χρήστη από το clerk id . Αν ο χρήστης δεν είναι επαληθευμένος τότε επιστρέφεται **401 Unauthorized** .

Στο **βήμα 2** γίνεται η μετατροπή του clerk id σε database user id ώστε να μπορεί να γίνει η σύγκριση .

Στο **βήμα 3** βρίσκουμε την εκλογή που μας ενδιαφέρει και παίρνουμε μόνο το userId δηλαδή τον δημιουργό της ψηφοφορίας . Το **election.userId** είναι το **Database User Id** του δημιουργού .

Στο **βήμα 4** συγκρίνουμε αν το **election.userId** είναι ίδιο με το **Database User Id** . Αν δεν ταιριάζουν τότε εμφανίζεται μήνυμα λάθους HTTP **403 forbidden** . Αν ταιριάζουν τότε συνεχίζουμε στην διαγραφή της ψηφοφορίας .

Στο **βήμα 5** πραγματοποιείται η διαγραφή της ψηφοφορίας .

```

1 export async function DELETE(request: Request) {
2   const electionId = extractIdFromUrl(request);
3   if (electionId === null) {
4     return NextResponse.json({ error: "Invalid election id" }, { status: 400
      ↪   });
5   }
6
7   try {
8     // 1 AUTHENTICATION:
9     const { userId: clerkId } = await auth();
10    if (!clerkId) {
11      return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
12    }
13
14    // 2 USER LOOKUP:

```

```

15     const dbUser = await prisma.user.findUnique({
16       where: { clerkId },
17       select: { id: true },
18     });
19     if (!dbUser) {
20       return NextResponse.json({ error: "User not registered" }, { status: 404
21         ↪ });
22     }
23     // 3 ELECTION LOOKUP:
24     const election = await prisma.election.findUnique({
25       where: { id: electionId },
26       select: { userId: true }, // + Π          userId (      )
27     });
28     if (!election) {
29       return NextResponse.json({ error: "Election not found" }, { status: 404
30         ↪ });
31     }
32     // 4 AUTHORIZATION CHECK:
33     if (election.userId !== dbUser.id) {
34       return NextResponse.json({ error: "Forbidden" }, { status: 403 });
35     }
36
37     // 5 DELETE:
38     await prisma.election.delete({ where: { id: electionId } });
39     return NextResponse.json({ success: true });
40   } catch (error) {
41     console.error("DELETE /api/elections/[id] error:", error);
42     return NextResponse.json({ error: "Failed to delete election" }, { status:
43       ↪ 500 });
44   }

```

```
200 OK με { "success": true }
```

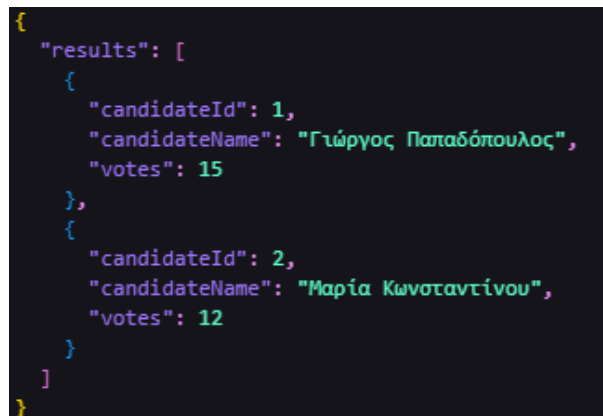
Σχήμα 12: Επιτυχής επιστροφή του **Api: DELETE /api/elections/[id]** σε μορφή **JSON**

6.6 GET /api/elections/[id]/results

Το συγκεκριμένο endpoint επιστρέφει τα αποτελέσματα μιας ψηφοφορίας . Εμφανίζονται οι ψήφοι σε φθίνουσα σειρά για κάθε υποψήφιο σε κάθε ψηφοφορία . Στην έξοδο στο json αρχείο εμφανίζονται το id , το ονοματεπώνυμο και οι ψήφοι που έχει λάβει ο κάθε υποψήφιος . Δεν απαιτεί να είναι αληθές το authentication . Για να εμφανιστούν οι ψήφοι για τον κάθε έναν υποψήφιο πρέπει να βρούμε τα δεδομένα από την βάση για τα συγκεκριμένα id . Αυτό επιτυγχάνεται με το να παίρνουμε το id του κάθε χρήστη απο το url . Η διαδικασία γίνεται με τον

διαχωρισμό του url και αποθήκευση όλων των τιμών σε έναν πίνακα . Στην συνέχεια παίρνοντας το προτελευταίο κομμάτι αυτού του πίνακα βρίσκουμε το εκάστοτε id .Αν το id δεν είναι έγκυρος αριθμός τότε εμφανίζεται μήνυμα λάθους με κωδικό **400 Bad request** . Παρακάτω βλέπουμε αναλυτικά τον κώδικα που υλοποιεί την συγκεκριμένη ενέργεια .

```
1 //      id      path
2 const url = new URL(request.url);
3 const parts = url.pathname.split("/");
4 const rawId = parts[parts.length - 2]; //      [id]
5 const electionId = parseInt(rawId, 10);
6
7 if (isNaN(electionId)) {
8   return NextResponse.json({ error: "Invalid election id" }, { status: 400 });
9 }
```



```
{
  "results": [
    {
      "candidateId": 1,
      "candidateName": "Γιώργος Παπαδόπουλος",
      "votes": 15
    },
    {
      "candidateId": 2,
      "candidateName": "Μαρία Κωνσταντίνου",
      "votes": 12
    }
  ]
}
```

Σχήμα 13: Παράδειγμα επιστροφής του Api: **GET /api/elections/[id]/results** σε μορφή **JSON**

6.7 GET /api/elections/[id]/candidates

Το endpoint επιστρέφει όλους τους υποψήφιους της ψηφοφορίας με όλα τα στοιχεία τους καθώς και με τον αριθμό των ψήφων που έχουν λάβει . Πρέπει να λάβει ως είσοδο το αναγνωριστικό της ψηφοφορίας το οποίο παίρνει από το url.

```
1 export async function GET(
2   _req: Request,
3   { params }: { params: Promise<{ id: string }> }
4 ) {
5   const { id } = await params;
6   const electionId = parseInt(id, 10);
7   if (isNaN(electionId)) {
8     return NextResponse.json({ error: "Invalid election id" }, { status: 400
9       ↪   });
10  }
```

6.8 POST /api/elections/[id]/candidates

Το συγκεκριμένο api χρησιμοποιείται για να γίνει προσθήκη ενός νέου υποψηφίου σε μια ψηφοφορία . Αποτελεί διπλή εγγραφή η οποία ενημερώνει παράλληλα τον πίνακα pollcandidates και τον πίνακα takepart .

Αρχικά δημιουργείται η νέα πρόσκληση ενός υποψηφίου στον πίνακα pollcandidates και ορίζει σαν δεδομένα το id της εκλογής , το id του υποψηφίου , καθώς και ακριβώς την χρονική στιγμή στην οποία δημιουργήθηκε η πρόσκληση .

```
1
2 await prisma.poll_candidates.create({
3   data: { poll_id: electionId, user_id: userId },
4 });
```

Στον πίνακα takepart γίνεται η αναλυτική καταμέτρηση των ψήφων στο σύστημα των ηλεκτρονικών ψηφοφοριών . Ο σκοπός της δημιουργίας του είναι να περιέχει τις πληροφορίες σχετικά με τους υποψήφιους και τον αριθμό των ψήφων που έχουν λάβει .

Αρχικά παίρνει το **election id** από το url και ελέγχει αν είναι έγκυρος αριθμός . Αν δεν είναι έγκυρος τότε θα επιστρέψει μήνυμα λάθους με κωδικό **400**.

```
1 const { id } = await params;
2 const electionId = parseInt(id, 10);
3 if (isNaN(electionId)) {
4   return NextResponse.json({ error: "Invalid election id" }, { status: 400 });
5 }
```

Στην συνέχεια δημιουργεί ένα αίτημα στο οποίο ζητάει σαν τιμή το **user id** και πραγματοποιεί έλεγχο ώστε να είναι αριθμός διαφορετικά πετάει εξίσου μήνυμα λάθους με κωδικό **400**.

```
1 const { userId } = await request.json();
2 if (typeof userId !== "number") {
3   return NextResponse.json({ error: "Missing or invalid userId" }, { status:
4     ↳ 400 });
4 }
```

Έπειτα χρησιμοποιεί την συνάρτηση **upsert()** ώστε να γίνει έλεγχος αν υπάρχει ήδη εγγραφή με τον συνδυασμό του **election id** και του **user id**. Αν βρει ότι υπάρχει ήδη η εγγραφή τότε κάνει ανανέωση χωρίς να αλλάξει κάτι . Αν δεν βρεθεί η συγκεκριμένη εγγραφή τότε την ενσωματώνει με 0 ψήφους .

```
1 await prisma.takepart.upsert({
2   where: {
3     electionId_candidateId: {
4       electionId,
```

```
5     candidateId: userId,
6   },
7 },
8 update: {},
9 create: {
10   electionId,
11   candidateId: userId,
12   numberOfVotes: 0,
13 },
14 });
```

6.9 DELETE /api/elections/[id]/candidates/[candidateId]

Το συγκεκριμένο api endpoint αφαιρεί έναν υποψήφιο από μια εκλογή . Δέχεται ως είσοδο το αναγνωριστικό της εκλογής και του υποψηφίου και αν επιβεβαιωθούν πραγματοποιείται η διαγραφή . Η διαγραφή γίνεται και από τους δύο πίνακες pollcandidates και takepart.

```
1 // What happens in the database
2 const electionId = 123;
3 const candId = 456;
4
5 // Delete from takepart
6 await prisma.takepart.deleteMany({
7   where: { electionId: 123, candidateId: 456 },
8 });
9
10 // Delete from poll_candidates
11 await prisma.poll_candidates.deleteMany({
12   where: { poll_id: 123, user_id: 456 },
13 });
```

6.10 POST /api/elections/[id]/vote

Σε αυτό το endpoint γίνεται η καταγραφή του ψήφου σε μια ψηφοφορία . Όταν ψηφίζει ένας χρήστης τον υποψήφιο που επιθυμεί αυξάνει τις ψήφους του . Αρχικά γίνεται έλεγχος αν ο χρήστης έχει ήδη ψηφίσει . Μετά την ψηφοφορία γίνεται ενημέρωση του πίνακα takepart με **takepart.numberOfVotes** και επιστρέφει μήνυμα επιτυχίας **201 Created** ή μήνυμα αποτυχίας **409 Conflict**.

```
1 export async function POST(request: Request) {
2   const { userId: clerkId } = await auth();
3   const { electionId, candidateId } = await request.json();
4
5   // Check if already voted
6   const existingVote = await prisma.vote.findUnique({
7     where: { userId_electionId: { userId: clerkId, electionId } }
```

```

8   });
9   if (existingVote) return NextResponse.json({ error: "Already voted" }, {
    ↪   status: 409 });
10
11  // Update vote count in takepart
12  const existingTakePart = await prisma.takepart.findUnique({
13    where: { electionId_candidateId: { electionId, candidateId } }
14  });
15
16  if (existingTakePart) {
17    await prisma.takepart.update({
18      where: { electionId_candidateId: { electionId, candidateId } },
19      data: { numberOfVotes: existingTakePart.numberOfVotes + 1 }
20    });
21  } else {
22    await prisma.takepart.create({
23      data: { electionId, candidateId, numberOfVotes: 1 }
24    });
25  }
26
27  // Create vote record
28  await prisma.vote.create({
29    data: { userId: realUserId, electionId, candidateId }
30  });
31
32  return NextResponse.json({ success: true });
33 }

```

6.11 GET /api/elections/filtered

Αυτό Το endpoint επιστρέφει μόνο τις εκλογές που ταιριάζουν στα στοιχεία που έχει δηλώσει ο χρήστης στο προφίλ του. Η επιλογή γίνεται με τα κριτήρια φύλου, επαγγέλματος τοποθεσίας και ηλικίας. Η έξοδος είναι μία λίστα με τις κατάλληλες εκλογές.

```

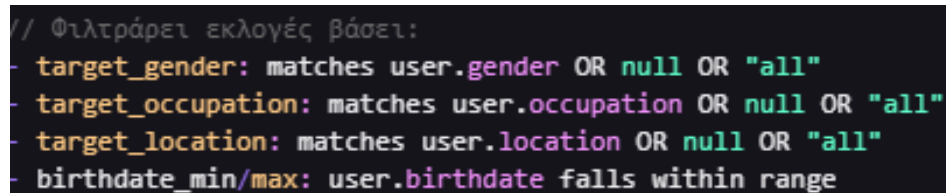
1  export async function GET() {
2    const { userId: clerkId } = await auth();
3    const dbUser = await prisma.user.findUnique({
4      where: { clerkId },
5      select: { id: true, gender: true, occupation: true, location: true,
    ↪    birthdate: true }
6    });
7
8    const elections = await prisma.election.findMany({
9      where: {
10        AND: [
11          { OR: [{ target_gender: dbUser.gender }, { target_gender: null }, {
    ↪    target_gender: "all" }] }],

```

```

12     { OR: [{ target_occupation: dbUser.occupation }, { target_occupation:
    ↪ null }, { target_occupation: "all" }] },
13     { OR: [{ target_location: dbUser.location }, { target_location: null },
    ↪ { target_location: "all" }] },
14     ...(dbUser.birthdate ? [
15         { birthdate_min: { lte: dbUser.birthdate } },
16         { birthdate_max: { gte: dbUser.birthdate } }
17     ] : [])
18   ],
19 },
20 include: { takepart: { include: { user: true } } },
21 orderBy: { start_date: "desc" }
22 });
23
24 return NextResponse.json(formatted);
25 }

```



```

// Φιλτράρει εκλογές βάσει:
- target_gender: matches user.gender OR null OR "all"
- target_occupation: matches user.occupation OR null OR "all"
- target_location: matches user.location OR null OR "all"
- birthdate_min/max: user.birthdate falls within range

```

Σχήμα 14: Παράδειγμα επιστροφής του Api: **GET /api/elections/filtered** σε μορφή JSON

6.12 GET /api/elections/my

Με το συγκεκριμένο endpoint api εμφανίζονται μόνο οι ψηφοφορίες που έχει δημιουργήσει αυτός ο χρήστης . Έχει δημιουργηθεί για την διαχείριση των εκλογών και δείχνει και τους υποψήφιους της κάθε ψηφοφορίας.

```

1 export async function GET() {
2   const { userId: clerkId } = await auth();
3   const dbUser = await prisma.user.findUnique({
4     where: { clerkId },
5     select: { id: true }
6   });
7
8   const elections = await prisma.election.findMany({
9     where: { userId: dbUser.id },
10    include: { takepart: { include: { user: true } } },
11    orderBy: { start_date: "desc" }
12  });
13
14  const formatted = elections.map((el) => ({
15    id: el.id,

```

```

16     title: el.title,
17     description: el.description,
18     dateRange: {
19         startDate: el.start_date.toISOString(),
20         endDate: el.end_date.toISOString()
21     },
22     targeting: {
23         roles: el.target_occupation ? [el.target_occupation] : [],
24         locations: el.target_location ? [el.target_location] : []
25     },
26     candidates: el.takepart.map((tp) => tp.user),
27     createdAt: el.start_date.toISOString(),
28     isActive: el.is_active,
29     createdByCurrentUser: true
30 }));
31
32 return NextResponse.json(formatted);
33 }

```

6.13 PollCandidates Endpoints

Για την προσθήκη την επιστροφή και την διαγραφή των υποψηφίων χρησιμοποιούνται τα εξής api endpoints :

- **GET /api/poll-candidates:** Επιστρέφει τις εγγραφές των υποψηφίων για συγκεκριμένες εκλογές με όλα τα στοιχεία τους. Τα στοιχεία που επιστρέφονται είναι το αναγνωριστικό της εγγραφής , του υποψηφίου και της ψηφοφορίας . Την ημερομηνία στην οποία προστέθηκε ο υποψήφιος σε μια ψηφοφορία . Τέλος επιστρέφει το ονοματεπώνυμο , το email και το επάγγελμα του.

```

[
  {
    "id": 1,
    "poll_id": 1,
    "user_id": 2,
    "invited_at": "2024-01-15T10:00:00.000Z",
    "fullName": "Γιώργος Παπαδόπουλος",
    "email": "giorgos@example.com",
    "occupation": "student",
    "candidateId": 2
  }
]

```

Σχήμα 15: Παράδειγμα επιστροφής του Api: **GET /api/poll-candidates** σε μορφή JSON

- **POST /api/poll-candidates:** Προσθέτει έναν νέο υποψήφιο σε μια νέα ψηφοφορία .

```
{
  "pollId": "1",
  "userId": 2
}
```

Σχήμα 16: Παράδειγμα επιστροφής του **Api: POST /api/poll-candidates** σε μορφή **JSON**

- **DELETE /api/poll-candidates/[id]:** Διαγράφει έναν υποψήφιο από μια ψηφοφορία .

6.14 User και Profile Endpoints

Για τη διαχείριση χρηστών και προφίλ υποστηρίζονται:

- **GET /api/users:** Επιστρέφει όλα τα στοιχεία των χρηστών και επιτρέπει την αναζήτηση των χρηστών με φίλτρα το ονοματεπώνυμο το email και το επάγγελμα.

```
[
  {
    "id": 1,
    "clerkId": "user_123",
    "fullName": "Γιώργος Παπαδόπουλος",
    "username": "giorgos",
    "email": "giorgos@example.com",
    "occupation": "student",
    "location": "Athens",
    "gender": "male"
  }
]
```

Σχήμα 17: Παράδειγμα επιστροφής του **Api: GET /api/users** σε μορφή **JSON**

- **POST /api/profile-sync:** Συγχρονίζει τα στοιχεία για έναν χρήστη τα οποία λαμβάνει απο το σύστημα Clerk.

```
{
  "clerkId": "user_123",
  "email": "giorgos@example.com",
  "fullName": "Γιώργος Παπαδόπουλος",
  "username": "giorgos"
}
```

Σχήμα 18: Παράδειγμα επιστροφής της αίτησης για το **Api: POST /api/profile-sync** σε μορφή **JSON**

```
{
  "success": true,
  "user": {
    "id": 1,
    "clerkId": "user_123",
    "email": "giorgos@example.com",
    "fullName": "Γιώργος Παπαδόπουλος",
    "username": "giorgos",
    "isAdmin": true,
    "hasProfile": true,
    "profile": {
      "gender": "male",
      "birthdate": "1995-05-15T00:00:00.000Z",
      "occupation": "student",
      "location": "Athens"
    }
  }
}
```

Σχήμα 19: Παράδειγμα επιστροφής της απόκρισης μετά την αίτηση για το **Api: POST /api/profile-sync** σε μορφή JSON

- **POST /api/verify (και παραλλαγές):** Ενημερώνει τα στοιχεία του προφίλ ενός χρήστη (φύλο, ημερομηνία γέννησης, επάγγελμα, τοποθεσία).

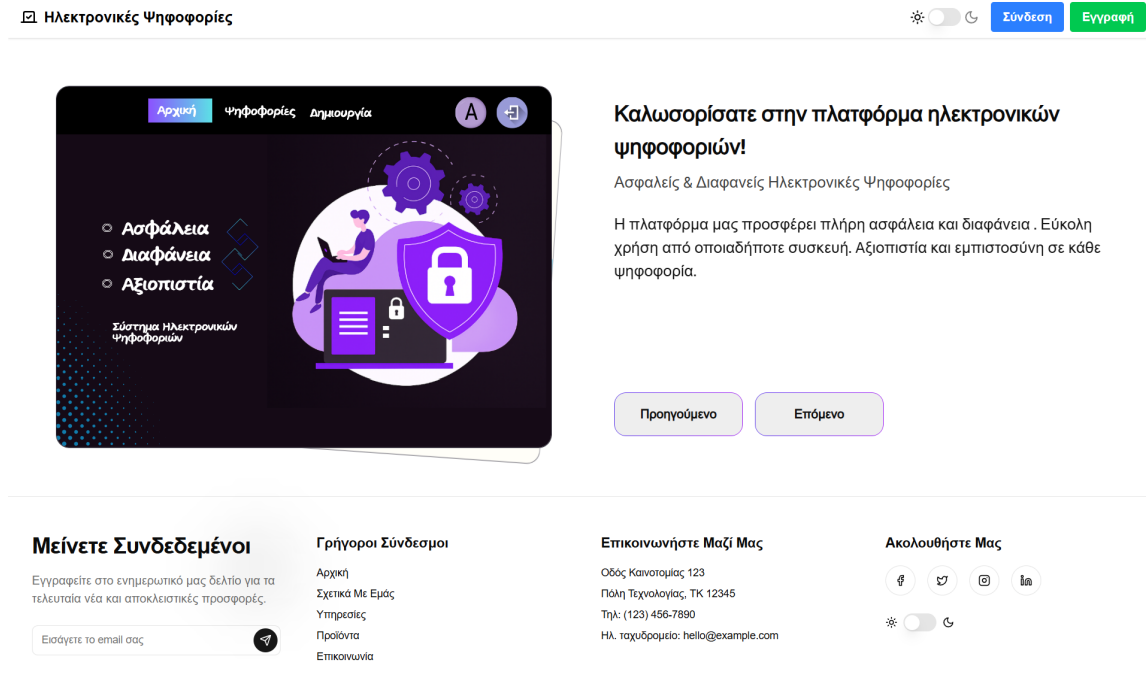
```
{
  "idNumber": "user_123",
  "gender": "male",
  "birthdate": "1995-05-15",
  "occupation": "student",
  "location": "Athens"
}
```

Σχήμα 20: Παράδειγμα επιστροφής του **Api: POST /api/verify** σε μορφή JSON

7 Προβολή της Εφαρμογής

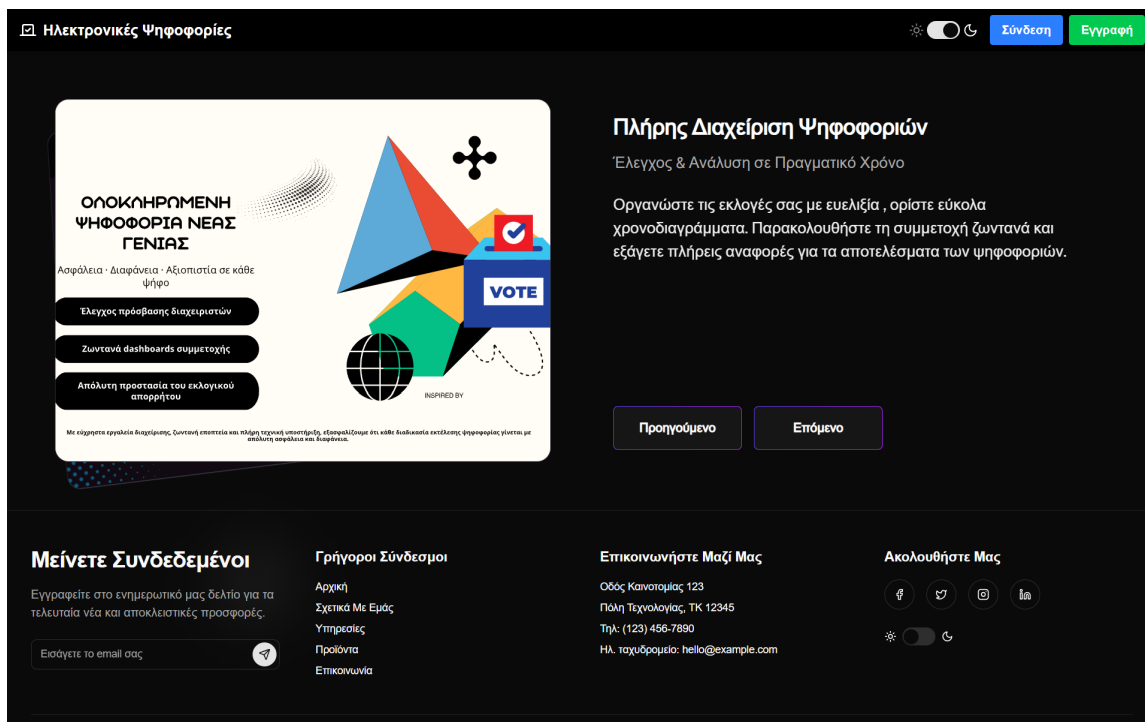
Όταν συνδέεται ο χρήστης στην πλατφόρμα ηλεκτρονικών ψηφοφοριών, βλέπει την αρχική οθόνη της εφαρμογής, η οποία έχει σχεδιαστεί να είναι φιλική προς τον χρήστη και έχει ως αρχικό στόχο την πληροφόρηση του σχετικά με τον λόγο ύπαρξης της συγκεκριμένης πλατφόρμας.

Στο επάνω μέρος της σελίδας υπάρχει η μπάρα για την σύνδεση ή εγγραφή του χρήστη με το λογότυπο και τον τίτλο, ενώ υπάρχει και διακόπτης για την εναλλαγή της διεπαφής του χρήστη σε σκοτεινή ή φωτεινή εμφάνιση (*dark mode*).



Σχήμα 21: Αρχική Σελίδα σε Φωτεινή Λειτουργία

Στο κεντρικό τμήμα της σελίδας, παρουσιάζεται ένα εντυπωσιακό γραφικό που πλαισιώνεται από εικονίδια ασφάλειας και τεχνολογίας, συνοδευόμενο από τις βασικές αξίες του συστήματος: *Ασφάλεια*, *Διαφάνεια* και *Αξιοπιστία*. Δίπλα από το γραφικό, εμφανίζεται ένα σύντομο καλωσόρισμα που ενημερώνει τον χρήστη για τα πλεονεκτήματα της πλατφόρμας, όπως η πλήρης ασφάλεια, η διαφάνεια στη διαδικασία και η εύκολη πρόσβαση από οποιαδήποτε συσκευή.



Σχήμα 22: Αρχική Σελίδα σε Σκοτεινή Λειτουργία (Dark Mode)

Στο κάτω μέρος της σελίδας, ο χρήστης μπορεί να κάνει εγγραφή με το email ώστε να ενημερώνεται σχετικά με καινούργιες ψηφοφορίες και με τυχόν αλλαγές την πλάτφόρμα . Επιπλέον παρέχονται γρήγοροι σύνδεσμοι για να μπορεί να πλοηγηθεί μέσα στην πλατφόρμα με πιο σύντομο τρόπο . Αυτοί οι σύνδεσμοι αφορούν τις διαδρομές (*Αρχική, Σχετικά με Εμάς, Υπηρεσίες, Επικοινωνία*). Στη δεξιά πλευρά, βλέπουμε τα στοιχεία επικοινωνίας με την ανάλογη δημόσια υπηρεσία που θα έχει τον έλεγχο της πλατφόρμας καθώς και διάφορα εικονίδια για τα μέσα κοινωνικής δικτύωσης, προσφέροντας όλους τους τρόπους της επικοινωνίας και ενημέρωσης.

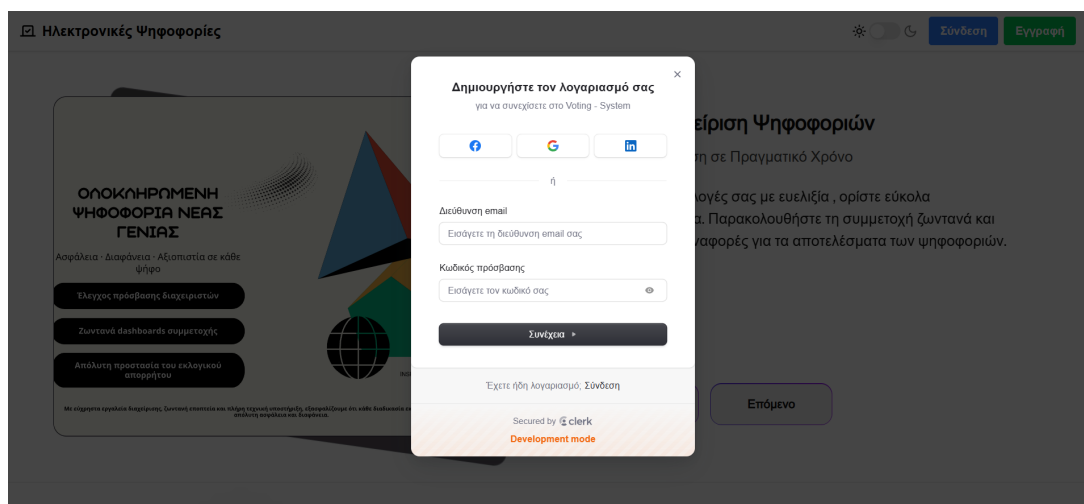
Η αρχική σελίδα έχει σχεδιαστεί με τρόπο ώστε να προσφέρει μια ολοκληρωμένη γνώμη για το σύστημα.

7.1 Διαδικασία Εγγραφής Νέου Χρήστη

Όταν ο χρήστης πατήσει το κουμπί για την εγγραφή του στο σύστημα θα πρέπει να δημιουργήσει τον λογαριασμό του μέσω του clerk . Η φόρμα εμφανίζεται σε παράθυρο και τονίζεται έντονα , επειδή γίνεται σκοτεινή η υπόλοιπη οθόνη και φαίνεται έντονα το αναδυόμενο παράθυρο ώστε να εστιάσει ο χρήστης όλη την προσοχή του στην διαδικασία της εγγραφής στο σύστημα .

Επιπλέον παρέχεται η δυνατότητα στον χρήστη να πραγματοποιήσει την εγγραφή του μέσα από διάφορα κοινωνικά δίκτυα όπως το Facebook , Google και LinkedIn ή και ακόμα με το προσωπικό του email . Τα πεδία της φόρμας είναι απλά και κατανοητά, μόνο με τη διεύθυνση email και με τον κωδικό πρόσβασης. Μετά τη συμπλήρωση όλων των στοιχείων, ο χρήστης πατάει το κουμπί «Συνέχεια» για να ολοκληρώσει τη διαδικασία.

Η διαδικασία έχει σχεδιαστεί με τέτοιο τρόπο ώστε να είναι γρήγορη, ασφαλής και φιλική προς τον χρήστη. Όσον αφορά την ασφάλεια της εφαρμογής , διασφαλίζεται από το Clerk, το οποίο εγγυάται για την προστασία των προσωπικών δεδομένων και την ταυτοποίηση του κάθε χρήστη.

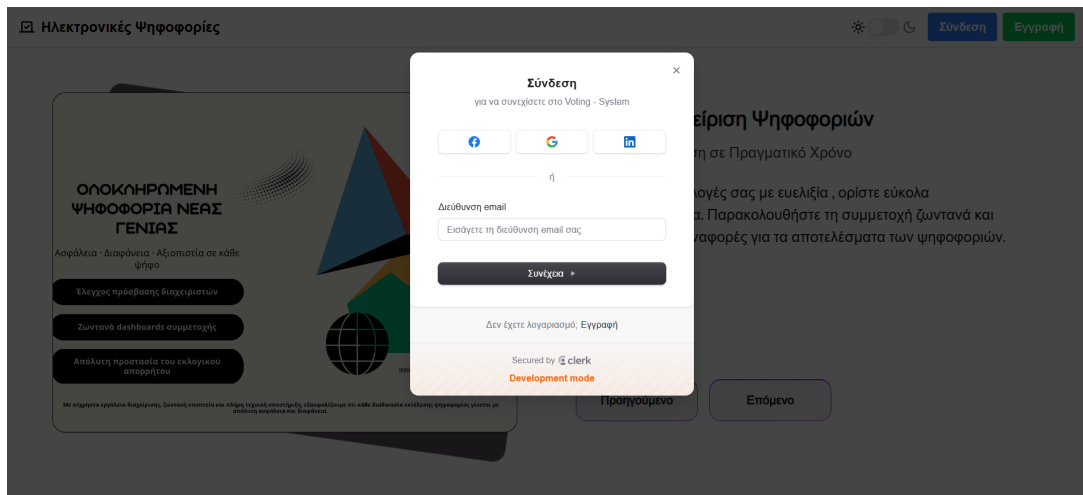


Σχήμα 23: Φόρμα εγγραφής νέου χρήστη μέσω Clerk

7.2 Διαδικασία Σύνδεσης Χρήστη

Για τους χρήστες οι οποίοι διαθέτουν ήδη λογαριασμό στο σύστημα ηλεκτρονικών ψηφοφοριών , η διαδικασία σύνδεσης πραγματοποιείται εξίσου μέσω του Clerk, με τη χρήση ακριβώς του ίδιου αναδυόμενου παραθύρου αλλά αυτήν την φορά για να πραγματοποιηθεί η σύνδεση του. Ο χρήστης μπορεί να επιλέξει να συνδεθεί είτε μέσω κοινωνικών δικτύων (Facebook, Google, LinkedIn) όπως ακριβώς και στην διαδικασία της εγγραφής είτε εισάγοντας τη διεύθυνση email και τον κωδικό πρόσβασης που έχει δηλώσει κατά την εγγραφή.

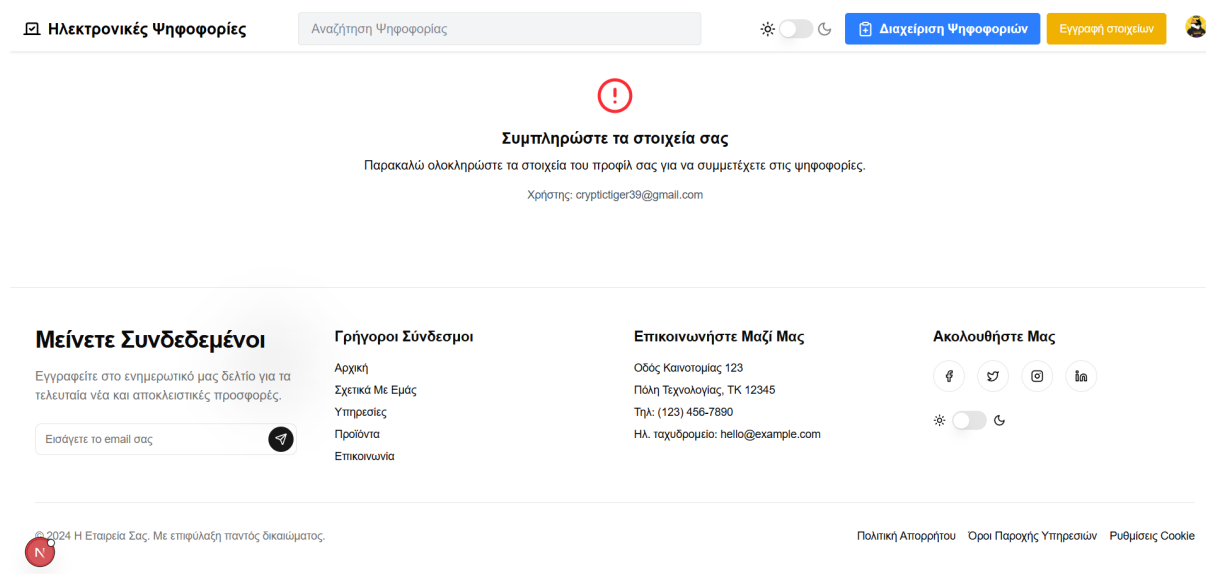
Η φόρμα σύνδεσης είναι απλή ώστε να μπορεί ο μέσος πολίτης να συνδεθεί με ευκολία στο σύστημα .Μετά τη συμπλήρωση όλων των στοιχείων ο χρήστης πατάει το κουμπί «Συνέχεια» για να συνδεθεί στην πλατφόρμα.



Σχήμα 24: Φόρμα σύνδεσης υφιστάμενου χρήστη μέσω Clerk

7.3 Υποχρεωτική Συμπλήρωση Στοιχείων Κατά την Πρώτη Είσοδο

Όταν ο χρήστης συνδέεται για πρώτη φορά στην πλατφόρμα ηλεκτρονικών ψηφοφοριών, η εφαρμογή τον καθοδηγεί στη διαδικασία να συμπληρώσει τα επιπλέον στοιχεία του που χρειάζονται ώστε να μπορέσει να συμμετέχει σε ψηφοφορίες που θα τον αφορούν .



Σχήμα 25: Μήνυμα ειδοποίησης για συμπλήρωση στοιχείων προφίλ χρήστη

Στο επάνω μέρος της σελίδας συνεχίζει να διατηρείται η μπάρα πλοήγησης της εφαρμογής με το λογότυπο, καθώς προσφέρεται και η δυνατότητα αναζήτησης μιας ψηφοφορίας μέσω απο την μπάρα αναζήτησης , το διακόπτη της φωτεινής και της σκοτεινής λειτουργίας και τα κουμπιά για *Διαχείριση Ψηφοφοριών* και *Εγγραφή Στοιχείων*.

7.4 Επαλήθευση και Συμπλήρωση Προφίλ Χρήστη

Μετά την επιτυχή εγγραφή ή σύνδεση στην πλατφόρμα, ο χρήστης καλείται να ολοκληρώσει τη διαδικασία επαλήθευσης του προφίλ του μέσω μιας ειδικής φόρμας που εμφανίζεται σε αναδυόμενο παράθυρο (*modal*). Η φόρμα αυτή αποτελεί βασικό βήμα για τη διασφάλιση της ταυτοποίησης κάθε συμμετέχοντα και την αποτροπή πολλαπλών ψήφων στις ψηφοφορίες.

Η φόρμα της επαλήθευσης του προφίλ περιέχει τα εξής υποχρεωτικά πεδία:

- **Φύλο:** Επιλογή από κάποιες προκαθορισμένες τιμές , δηλαδή άνδρας γυναίκα ή άλλο.
- **Ημερομηνία Γέννησης:** Η εισαγωγή της ημερομηνίας γίνεται με την χρήση ημερολογίου για να αποφευχθούν λάθη.
- **Επάγγελμα:** Επιλογή από μια λίστα διάφορων επαγγελμάτων.
- **Πόλη:** Επιλογή τοποθεσίας από μια προκαθορισμένη λίστα με πόλεις της Ελλάδας.

Στο κάτω μέρος της φόρμας, εμφανίζεται ένα πλαίσιο με μια ενημερωτική σημαντική σημείωση, που διασφαλίζει τον χρήστη ότι τα στοιχεία που παρέχει θα χρησιμοποιηθούν αποκλειστικά για την προσθήκη του μόνο στις ψηφοφορίες που τον αφορούν καθώς και για την επαλήθευση της ταυτότητας του , διασφαλίζοντας τη διαφάνεια και την αξιοπιστία της διαδικασίας.

Η διαδικασία ολοκληρώνεται μόνο όταν ο χρήστης συμπληρώσει όλα τα πεδία και πατήσει το κουμπί «Υποβολή». Μόνο εφόσον έχει γίνει επιτυχή συμπλήρωση και υποβολή όλων των απαιτούμενων στοιχείων ο χρήστης αποκτά την πλήρη πρόσβαση στην διαδικασία και μπορεί να συμμετέχει κανονικά στις ηλεκτρονικές ψηφοφορίες.

Σχήμα 26: Φόρμα Επαλήθευσης Προφίλ Χρήστη

7.5 Υποβολή Ψήφου και Προβολή Ενεργών Ψηφοφοριών

Αφού ο χρήστης έχει ολοκληρώσει με επιτυχία την εγγραφή και την ταυτοποίηση του προφίλ του μπορεί να δει τις ενεργές ψηφοφορίες που είναι διαθέσιμες για τον ίδιο. Παρακάτω στα σχήματα 27 και 31 παρουσιάζεται η διαδικασία της υποβολής ψήφου για μια ψηφοφορία καθώς και η γενική εμφάνιση των ενεργών ψηφοφοριών.

Στο κύριο μέρος της εφαρμογής ο χρήστης βλέπει την λίστα με τις ενεργές ψηφοφορίες. Κάθε ψηφοφορία αφορά διαφορετικούς χρήστες, προβάλλεται σε ξεχωριστή κάρτα και περιλαμβάνει:

- Τον τίτλο της ψηφοφορίας.
- Μία σύντομη περιγραφή.
- Την ημερομηνία λήξης.
- Ένα κουμπί «Ψήφισε τώρα» για εκκίνηση της διαδικασίας.

Με την επιλογή συμμετοχής, εμφανίζεται αναδυόμενο παράθυρο με τις διαθέσιμες επιλογές υποψηφίων. Για τον κάθε υποψήφιο της ψηφοφορίας εμφανίζονται ονοματεπώνυμο, email και επάγγελμα. Ο χρήστης επιλέγει τον υποψήφιο που επιθυμεί και πατά «Υποβολή ψήφου». Μετά την καταχώριση, η κάρτα της ψηφοφορίας ενημερώνεται και το κουμπί αλλάζει σε *έχει ήδη ψηφίσει*, ώστε να αποτραπεί μια καινούργια συμμετοχή από τον ίδιο χρήστη.

Εγγεγραμμένος χρήστης - Chain Scriptor

Αυτόματη ενημέρωση κάθε 3 δευτερόλεπτα - Ψηφοφορίες που σας αφορούν (Τελευταία ενημέρωση: 4:50:56 PM)

Ενεργές

Ολοκληρωμένες

5 ενεργές ψηφοφορίες

Ιδιωτική ψηφοφορία Ατει

Με κωδικό πρόσβασης

ιδιωτική ψηφοφορία Ατει

Λήξη: 01/10/2025

Έχεις ήδη ψηφίσει

Ιδιωτική

Νέα από καινούργιο χρήστη

Όλοι μπορούν να ψηφίσουν

Νέα από καινούργιο χρήστη

Λήξη: 02/10/2025

Ψήφισε τώρα

Δημόσια

Παρουσίαση ΠΕ - Ιδιωτική

Με κωδικό πρόσβασης

Παρουσίαση ΠΕ - Ιδιωτική

Λήξη: 01/10/2025

Ψήφισε τώρα

Ιδιωτική

Περιορισμένη Ψηφοφορία

Με targeting criteria

Περιορισμένη Ψηφοφορία

Λήξη: 01/10/2025

Ψήφισε τώρα

Περιορισμένη

Σχήμα 27: Προβολή Ενεργών Ψηφοφοριών

Η δημιουργία νέας ψηφοφορίας ξεκινά με το πάτημα του κουμπιού «+ Δημιουργία Ψηφοφορίας». Αυτό εμφανίζει το αντίστοιχο παράθυρο στο οποίο ο χρήστης καλείται να συμπληρώσει τα στοιχεία της ψηφοφορίας που θέλει να δημιουργήσει.

Το παράθυρο έχει τίτλο «Δημιουργία Νέας Ψηφοφορίας» και περιλαμβάνει τα εξής βασικά πεδία:

- Τίτλος: Το όνομα της ψηφοφορίας.
- Περιγραφή: Προαιρετικό επεξηγηματικό κείμενο με λεπτομέρειες για τον σκοπό της ψηφοφορίας.
- Τύπος Ψηφοφορίας: Ορίζεται αν η ψηφοφορία θα είναι δημόσια, ιδιωτική με κωδικό πρόσβασης, μόνο για προσκεκλημένους ή περιορισμένη με κριτήρια στόχευσης.
- Κωδικός Πρόσβασης: Συμπληρώνεται όταν η ψηφοφορία είναι ιδιωτική.
- Προσκεκλημένοι Χρήστες: Στην περίπτωση που η ψηφοφορία είναι μόνο για προσκεκλημένους, ο δημιουργός επιλέγει συγκεκριμένους χρήστες από το σύστημα.
- Ημερομηνίες Έναρξης και Λήξης: Καθορίζονται οι ημερομηνίες διεξαγωγής της ψηφοφορίας.
- Κριτήρια Στόχευσης: Στην περίπτωση περιορισμένης ψηφοφορίας, επιλέγονται φίλτρα όπως επάγγελμα, τοποθεσία, ελάχιστη και μέγιστη ηλικία ή φύλο.
- Επιλογές Ψηφοφορίας: Ο δημιουργός μπορεί να προσθέσει υποψήφιους είτε επιλέγοντας χρήστες του συστήματος είτε εισάγοντας επιλογές με κείμενο.

Στο κάτω μέρος της φόρμας υπάρχει το κουμπί «Δημιουργία Ψηφοφορίας» το οποίο ενεργοποιεί τον έλεγχο εγκυρότητας ώστε να επιβεβαιωθεί ότι όλα τα πεδία έχουν συμπληρωθεί σωστά και στη συνέχεια καταχωρεί την ψηφοφορία στο σύστημα.

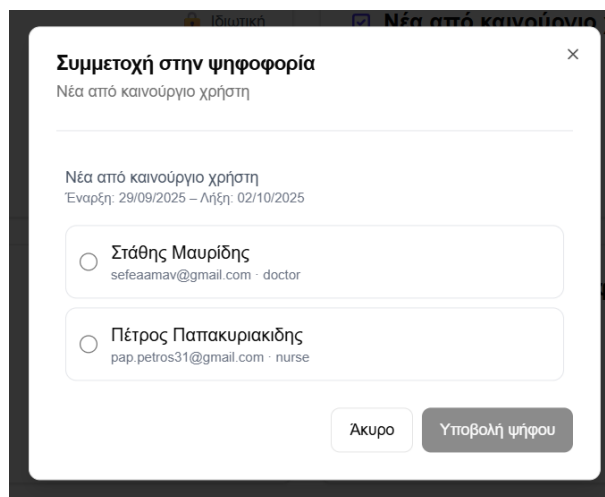
Η φόρμα έχει σχεδιαστεί ώστε να προσφέρει σαφήνεια και ευκολία στη χρήση. Με τη χρήση dropdowns, date pickers και δυναμικών φίλτρων μειώνεται η πιθανότητα λαθών και αυξάνεται η αποτελεσματικότητα. Η δυνατότητα επιλογής διαφορετικών τύπων ψηφοφορίας και η προσθήκη κριτηρίων στόχευσης παρέχουν ευελιξία στον δημιουργό, διασφαλίζοντας ότι η διαδικασία μπορεί να προσαρμοστεί ανάλογα με τις ανάγκες.

Κατά τη συμμετοχή σε μία ψηφοφορία, ο χρήστης βλέπει διαφορετική οθόνη ανάλογα με τον τύπο της ψηφοφορίας:

- Στη δημόσια ψηφοφορία εμφανίζεται η λίστα των υποψηφίων και ο χρήστης μπορεί να επιλέξει έναν και να υποβάλει την ψήφο του.
- Στην περιορισμένη ψηφοφορία εμφανίζεται σχετική ειδοποίηση ότι ισχύουν περιορισμοί βάσει προφίλ χρήστη, ενώ στη συνέχεια παρουσιάζονται οι διαθέσιμοι υποψήφιοι.
- Στην ιδιωτική ψηφοφορία απαιτείται εισαγωγή κωδικού πρόσβασης για να μπορέσει ο χρήστης να δει τους υποψήφιους και να συμμετάσχει.
- Στην ψηφοφορία για προσκεκλημένους εμφανίζεται ειδοποίηση ότι η ψηφοφορία είναι διαθέσιμη μόνο σε χρήστες που έχουν προσκληθεί, ενώ ακολουθεί η λίστα με τις επιλογές που έχουν καθοριστεί.

Σε όλες τις περιπτώσεις, στο κάτω μέρος υπάρχει το κουμπί «Υποβολή ψήφου» για την οριστικοποίηση της συμμετοχής και το κουμπί «Άκυρο» για την ακύρωση της διαδικασίας.

Συνολικά, η φόρμα δημιουργίας και η διεπαφή συμμετοχής στις ψηφοφορίες αποτελούν βασικά εργαλεία της πλατφόρμας, εξασφαλίζοντας την ορθή και έγκυρη διεξαγωγή των εκλογικών διαδικασιών με ευκολία και ασφάλεια για τους χρήστες.



Συμμετοχή στην ψηφοφορία

Νέα από καινούργιο χρήστη

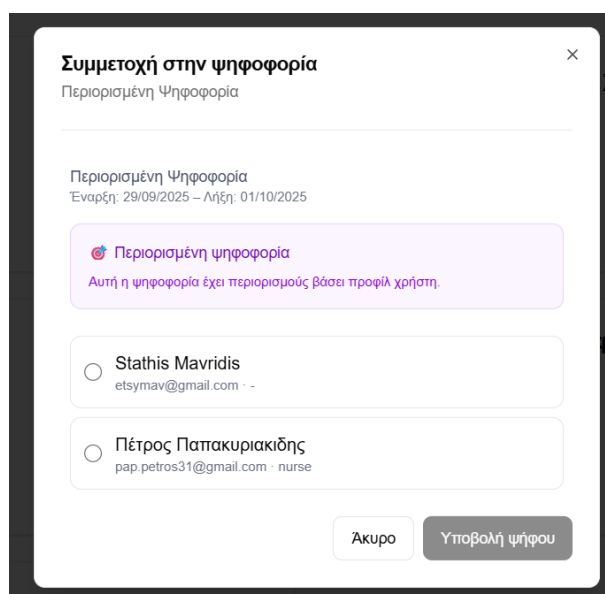
Νέα από καινούργιο χρήστη
Έναρξη: 29/09/2025 – Λήξη: 02/10/2025

☐ Στάθης Μαυρίδης
sefeamav@gmail.com · doctor

☐ Πέτρος Παπακυριακίδης
pap.petros31@gmail.com · nurse

Άκυρο Υποβολή ψήφου

Σχήμα 28: Διαδικασία Υποβολής Ψήφου - Δημόσια ψηφοφορία



Συμμετοχή στην ψηφοφορία

Περιορισμένη Ψηφοφορία

Περιορισμένη Ψηφοφορία
Έναρξη: 29/09/2025 – Λήξη: 01/10/2025

 Περιορισμένη ψηφοφορία
Αυτή η ψηφοφορία έχει περιορισμούς βάσει προφίλ χρήστη.

☐ Stathis Mavridis
etsymav@gmail.com · -

☐ Πέτρος Παπακυριακίδης
pap.petros31@gmail.com · nurse

Άκυρο Υποβολή ψήφου

Σχήμα 29: Διαδικασία Υποβολής Ψήφου - Περιορισμένη ψηφοφορία

Συμμετοχή στην ψηφοφορία
Παρουσίαση ΠΕ - Ιδιωτική

Παρουσίαση ΠΕ - Ιδιωτική
Έναρξη: 29/09/2025 – Λήξη: 01/10/2025

🔒 Αυτή η ψηφοφορία είναι ιδιωτική
Εισάγετε τον κωδικό πρόσβασης για να συμμετάσχετε.

Εισάγετε τον κωδικό πρόσβασης **Επιβεβαίωση**

Εισάγετε τον κωδικό πρόσβασης για να δείτε τους υποψηφίους.

Ακυρο **Υποβολή ψήφου**

Σχήμα 30: Διαδικασία Υποβολής Ψήφου - Ιδιωτική ψηφοφορία

Συμμετοχή στην ψηφοφορία
Φοιτητικές Εκλογές Ατει - Προέδρος Τμήματος

Φοιτητικές Εκλογές Ατει
Έναρξη: 29/09/2025 – Λήξη: 01/10/2025

🗳️ Ψηφοφορία για προσκεκλημένους
Αυτή η ψηφοφορία είναι διαθέσιμη μόνο για προσκεκλημένους χρήστες.

☐ **Στάθης Μαυρίδης**
sefeamav@gmail.com · doctor

☐ **Πέτρος Παπακυριακίδης**
pap.petros31@gmail.com · nurse

☐ **Nikos Liolis**
nikosliolis21@gmail.com · doctor

Ακυρο **Υποβολή ψήφου**

Σχήμα 31: Διαδικασία Υποβολής Ψήφου - Προσκεκλημένοι σε ψηφοφορία

7.6 Προβολή Ολοκληρωμένων Ψηφοφοριών

Μετά την ολοκλήρωση της εγγραφής όλων των στοιχείων του χρήστη και την επαλήθευση του προφίλ του , αποκτάει πλήρη πρόσβαση στο περιβάλλον της εφαρμογής . Παρακάτω βλέπουμε την προβολή των ψηφοφοριών που αφορούν τον συγκεκριμένο χρήστη σε μια φιλική προς αυτον διεπαφή προσφέροντας του την δυνατότητα να συμμετέχει σε οποιαδήποτε επιθυμεί .

Στο επάνω μέρος της σελίδας συνεχίζει να διατηρείται η μπάρα πλοήγησης. Στη δεξιά πλευρά της εμφανίζεται το εικονίδιο του προφίλ απο την εικόνα που έχει χρησιμοποιήσει ο χρήστης στον λογαριασμό του email του , προσφέροντας πρόσβαση σε επιπλέον ρυθμίσεις που αφορούν κυρίως την αποσύνδεση του από την εφαρμογή ή πληροφορίες του λογαριασμού του .

Ακριβώς κάτω από τη μπάρα της πλοήγησης εμφανίζεται μια πράσινη μπάρα η οποία αντικαθιστά το παράθυρο που ζητάει τα επιπλέον στοιχεία του χρήστη και πλέον ενημέρωνει ότι έχει γίνει επιτυχής εγγραφή και ταυτοποίηση των στοιχείων του χρήστη.

Στο κύριο μέρος της εφαρμογής βλέπουμε 2 επιλογές σχετικά με τις ενεργές ψηφοφορίες και τις ολοκληρωμένες . Ο χρήστης μπορεί να πλοηγηθεί και στις δύο . Στην πρώτη βλέπει τις ψηφοφορίες οι οποίες είναι ακόμα ενεργές και μπορεί να ψηφίσει σε αυτές . Στην δεύτερη επιλογή με τις ολοκληρωμένες , ο χρήστης μπορεί να δει τις ψηφοφορίες οι οποίες έχουν ολοκληρωθεί καθώς και τα αποτελέσματα τους .

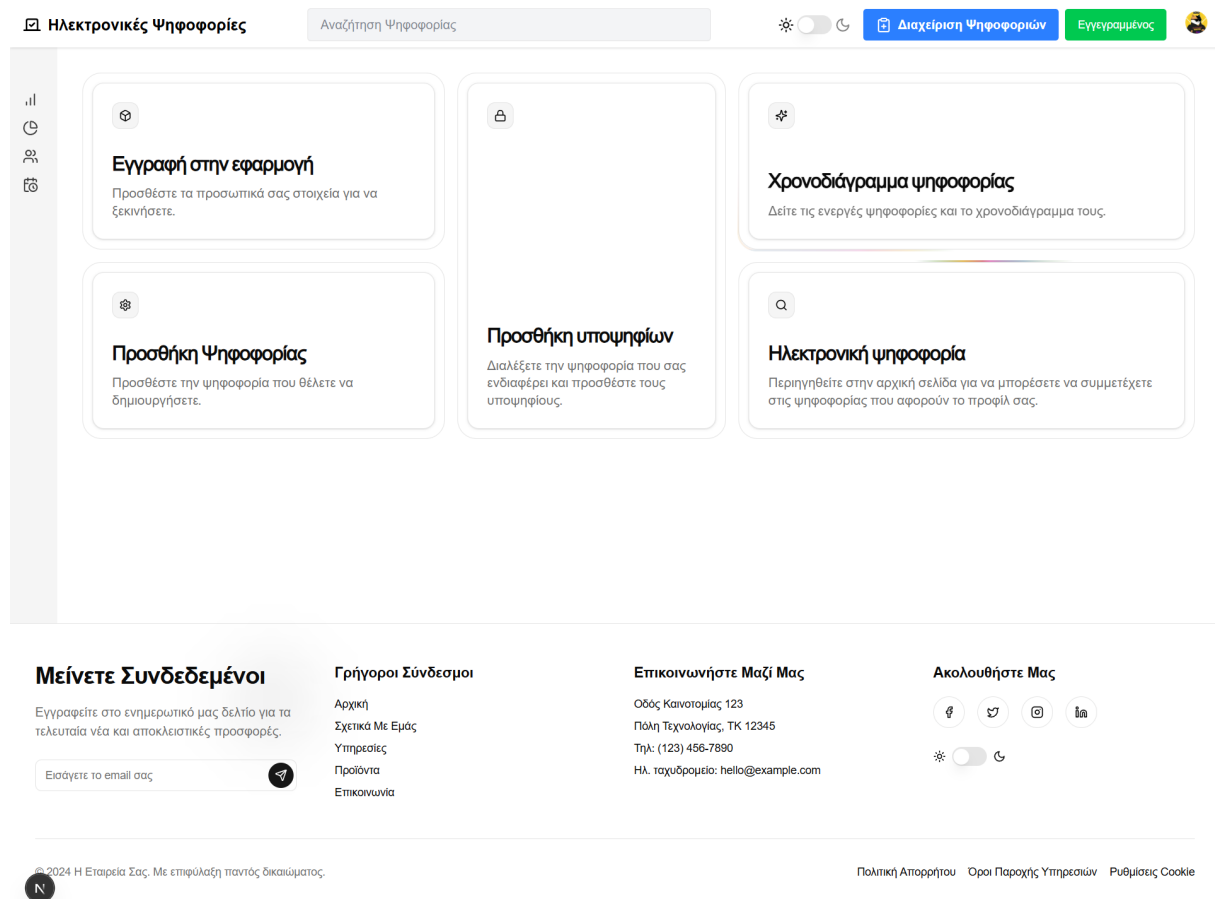
Κάθε ολοκληρωμένη ψηφοφορία εμφανίζεται με ξεχωριστή κάρτα . Η κάρτα περιλαμβάνει:

- Τον τίτλο της ψηφοφορίας.
- Μια σύντομη περιγραφή της ψηφοφορίας.
- Την ημερομηνία λήξης.
- Ένα μπλε κουμπί «Δες Αποτελέσματα» για προβολή των αναλυτικών αποτελεσμάτων.

Η διάταξη των καρτών έχει σχεδιαστεί με τέτοιο τρόπο ώστε να εμφανίζεται με ευανάγνωστο τρόπο για διάφορα μεγέθη οθόνων, διασφαλίζοντας την ευχρηστία της πλατφόρμας. Η χρήση των συγκεκριμένων χρωμάτων βοηθά στην εύκολη αναγνώριση των βασικών λειτουργιών της .

7.7 Κεντρικός Πίνακας Διαχείρισης – Αναλυτική Περιγραφή

Στο Σχήμα 34 βλέπουμε τον **κεντρικό πίνακα διαχείρισης (dashboard)** της εφαρμογής από την πλευρά της διαχείρισης των ψηφοφοριών η οποία ενημερώνει τον χρήστη σχετικά με τις λειτουργίες της εφαρμογής και την διαδικασία που πρέπει ο χρήστης να ακολουθήσει για να προσθέσει μια νέα ψηφοφορία και υποψηφίους. Η σχεδίαση της σελίδας εστιάζει στην *ευχρηστία*, και την *άμεση πρόσβαση* σε όλες τις ενέργειες που έχει την δυνατότητα να εκτελέσει ο χρήστης σαν διαχειριστής ή σαν απλός συμμετέχων.



Σχήμα 34: Κεντρικός πίνακας διαχείρισης της πλατφόρμας ηλεκτρονικής ψηφοφορίας.

Στην κύρια ενότητα της σελίδας εντοπίζουμε **έξι διακριτές κάρτες**, καθεμία από τις οποίες δείχνει την διαδρομή που πρέπει να ακολουθηθεί και αντιπροσωπεύει μια βασική λειτουργία της πλατφόρμας:

- **Εγγραφή στην εφαρμογή:** Ο χρήστης κάνει εγγραφή στην εφαρμογή μέσω του clerk και στην συνέχεια προσθέτει τα επιπλέον πεδία των στοιχείων του που χρειάζονται.
- **Προσθήκη Ψηφοφορίας:** Ο χρήστης δημιουργεί μια νέα ψηφοφορία στην οποία δηλώνει όλα τα απαραίτητα στοιχεία για να διαχωρίσει το κοινό που θα συμμετέχει.
- **Χρονοδιάγραμμα Ψηφοφορίας:** Παρουσιάζονται όλες οι ψηφοφορίες του συστήματος, ενεργές και αυτές που έχουν ήδη ολοκληρωθεί

- **Προσθήκη υποψηφίων:** Ο χρήστης επιλέγει την ψηφοφορία που έχει δημιουργήσει και μπορεί να προσθέσει και να αφαιρέσει υποψηφίους
- **Ηλεκτρονική ψηφοφορία:** Ο χρήστης μπορεί να ψηφίσει έναν υποψήφιο που θέλει στις ψηφοφορίες που τον ενδιαφέρουν.

Κάθε κάρτα περιλαμβάνει ένα *εικονίδιο* και μια *σύντομη περιγραφή* για να καταλαβαίνει ο χρήστης τι ακριβώς πρέπει να κάνει και ποια είναι τα επόμενα βήματα που θα πρέπει να ακολουθήσει. Η διάταξη είναι **προσαρμόσιμη** για να αλλάζει ανάλογα με τα μεγέθη των οθονών καθώς και για τις διαφορετικές συσκευές, ενώ η γενικότερη σχεδίαση ενισχύει την αισθητική πληρότητα της εφαρμογής.

7.8 Σελίδα Διαχείρισης Ψηφοφοριών

Η σελίδα με την προσθήκη ψηφοφοριών αποτελεί την κύρια λειτουργία του συστήματος ηλεκτρονικών ψηφοφοριών. Στην συγκεκριμένη σελίδα ο χρήστης μπορεί να δει ποιες ψηφοφορίες έχει δημιουργήσει, την ημερομηνία που έχουν δημιουργηθεί καθώς και σε ποιους χρήστες απευθύνεται η κάθε μια ψηφοφορία.

Οι Ψηφοφορίες Μου

Διαχείριση των ψηφοφοριών που έχεις δημιουργήσει

Δημιουργία Νέας Ψηφοφορίας

Παρουσίαση ΠΕ - Ιδιωτική

Παρουσίαση ΠΕ - Ιδιωτική

Έναρξη: 29/09/2025 00:00

Λήξη: 01/10/2025 00:00

Υποψήφιοι: 2

Κωδικός: vote2

Αποτελέσματα (1 συνολικές ψήφοι):

Στάθης Μαυρίδης	0 ψήφοι	Nikos Liolis	1 ψήφοι
-----------------	---------	--------------	---------

Υποψήφιοι:

Στάθης Μαυρίδης

Nikos Liolis

Διαγραφή

Παρουσίαση ΠΕ - Προσκεκλημένοι

Παρουσίαση ΠΕ - Προσκεκλημένοι

Έναρξη: 29/09/2025 00:00

Λήξη: 01/10/2025 00:00

Υποψήφιοι: 2

Αποτελέσματα (1 συνολικές ψήφοι):

1	0 ψήφοι	2	1 ψήφοι
---	---------	---	---------

Υποψήφιοι:

1

2

Διαγραφή

Σχήμα 35: Σελίδα Διαχείρισης Ψηφοφοριών.

Στο επάνω μέρος αριστερά φαίνεται ο τίτλος Οι Ψηφοφορίες μου και στα δεξιά υπάρχει το κουμπί Δημιουργία Ψηφοφορίας με το οποίο ο χρήστης μπορεί να δημιουργήσει μια νέα ψηφοφορία στο σύστημα.

Στον πίνακα παρατηρούμε τις εξής στήλες:

- **Τίτλος:** Το όνομα της ψηφοφορίας με μια σύντομη περιγραφή για το τι είναι από κάτω.

- **Κατάσταση:** Φαίνεται αν η ψηφοφορία είναι ενεργή ή έχει ολοκληρωθεί.
- **Διάστημα:** Ημερομηνία έναρξης και λήξης της ψηφοφορίας .
- **Στόχευση:** Σε ποιους χρήστες απευθύνεται η εκάστοτε ψηφοφορία.
- **Υποψήφιοι:** Αριθμός των καταχωρημένων υποψηφίων στην ψηφοφορία.
- **Δημιουργήθηκε:** Ημερομηνία δημιουργίας της ψηφοφορίας.

Η διάταξη του συγκεκριμένου πίνακα βοηθάει τον χρήστη να βλέπει με άμεσο τρόπο τις ψηφοφορίες και τα χρώματα καθώς και τα εικονίδια που ενισχύουν την εμπειρία χρήσης.

Επιπλέον η σελίδα δίνει την δυνατότητα για φιλτραρίσμα ή αναζήτηση ψηφοφορίας και η διαδικασία για την δημιουργία μιας νέας ψηφοφορίας γίνεται με το αντίστοιχο κουμπί το οποίο οδηγεί στην φόρμα εισαγωγής των βασικών πληροφοριών της ψηφοφορίας .

Επιπλέον η σελίδα αυτή παρέχει πλήρη εικόνα για την κατάσταση κάθε ψηφοφορίας (ολοκληρωμένη, σε εξέλιξη, προγραμματισμένη).

7.9 Δημιουργία Νέας Ψηφοφορίας

Η διαδικασία δημιουργίας νέας ψηφοφορίας είναι σχεδιασμένη ώστε να προσφέρει στον χρήστη την δυνατότητα να ορίσει με ακρίβεια όλες τις απαραίτητες παραμέτρους για την δημιουργία της .

Ο χρήστης βλέπει μια φόρμα με διαδοχικά πεδία και επιλογές. Αρχικά καλείται να εισάγει τον Τίτλο της ψηφοφορίας και προαιρετικά μια περιγραφή με επιπλέον λεπτομέρειες. Στη συνέχεια επιλέγει τον Τύπο Ψηφοφορίας με δυνατότητα να την ορίσει ως δημόσια όπου όλοι οι εγγεγραμμένοι χρήστες μπορούν να συμμετάσχουν ή ιδιωτική με χρήση κωδικού πρόσβασης. Υπάρχει επίσης η δυνατότητα καθορισμού Ημερομηνιών Διεξαγωγής μέσα από ημερολόγιο. Για τις Επιλογές Ψηφοφορίας, ο χρήστης μπορεί είτε να προσθέσει Χρήστες Συστήματος ως υποψήφιους είτε να δημιουργήσει απλές Επιλογές με Κείμενο. Τέλος, μέσω του κουμπιού Δημιουργία Ψηφοφορίας, ολοκληρώνεται η διαδικασία και καταχωρείται η νέα ψηφοφορία στο σύστημα.

Δημιουργία Νέας Ψηφοφορίας

Τίτλος

π.χ. Εκλογή Προέδρου Συλλόγου

Περιγραφή (Προαιρετικά)

Δώστε μερικές λεπτομέρειες για τη ψηφοφορία...

Τύπος Ψηφοφορίας

 Δημόσια - Όλοι μπορούν να ψηφίσουν

Όλοι οι εγγεγραμμένοι χρήστες μπορούν να ψηφίσουν.

Κωδικός Πρόσβασης

π.χ. VOTE2025

Διαθέσιμο μόνο για ιδιωτικές ψηφοφορίες.

Ημερομηνίες Διεξαγωγής

 Επιλέξτε εύρος

Επιλογές Ψηφοφορίας

Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με κείμενο.

Τύπος Επιλογών

☒  Χρήστες Συστήματος ☐  Επιλογές με Κείμενο

 Προσθήκη Υποψηφίου

Δημιουργία Ψηφοφορίας

Σχήμα 36: Φόρμα δημιουργίας νέας ψηφοφορίας – Δημόσια ψηφοφορία.

Περιγραφή (Προαιρετικά)

Δώστε μερικές λεπτομέρειες για τη ψηφοφορία...

<

September 2025

October 2025

>

Su	Mo	Tu	We	Th	Fr	Sa	Su	Mo	Tu	We	Th	Fr	Sa
31	1	2	3	4	5	6	28	29	30	1	2	3	4
7	8	9	10	11	12	13	5	6	7	8	9	10	11
14	15	16	17	18	19	20	12	13	14	15	16	17	18
21	22	23	24	25	26	27	19	20	21	22	23	24	25
28	29	30	1	2	3	4	26	27	28	29	30	31	1

 Επιλέξτε εύρος

Επιλογές Ψηφοφορίας

Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με κείμενο.

Τύπος Επιλογών

☒  Χρήστες Συστήματος ☐  Επιλογές με Κείμενο

 Προσθήκη Υποψηφίου

Σχήμα 37: Επιλογή έναρξης και λήξης της ψηφοφορίας.

Στην περίπτωση που η ψηφοφορία οριστεί ως ιδιωτική με κωδικό πρόσβασης, ο χρήστης βλέπει την αντίστοιχη επιλογή στον τύπο ψηφοφορίας. Εμφανίζεται μήνυμα που ενημερώνει ότι απαιτείται κωδικός πρόσβασης για συμμετοχή, ενώ το πεδίο κωδικού γίνεται υποχρεωτικό με την ένδειξη ότι είναι απαραίτητο για ιδιωτικές ψηφοφορίες. Οι υπόλοιπες επιλογές της φόρμας παραμένουν ίδιες.

Δημιουργία Νέας Ψηφοφορίας

Τίτλος

π.χ. Εκλογή Προέδρου Συλλόγου

Περιγραφή (Προαιρετικά)

Δώστε μερικές λεπτομέρειες για τη ψηφοφορία...

Τύπος Ψηφοφορίας

 Ιδιωτική - Με κωδικό πρόσβασης

Απαιτείται κωδικός πρόσβασης για συμμετοχή.

Κωδικός Πρόσβασης

π.χ. VOTE2025

Απαιτείται για ιδιωτικές ψηφοφορίες.

Ημερομηνίες Διεξαγωγής

 Επιλέξτε εύρος

Επιλογές Ψηφοφορίας

Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με κείμενο.

Τύπος Επιλογών

☒  Χρήστες Συστήματος ☐  Επιλογές με Κείμενο

 Προσθήκη Υποψηφίου

Δημιουργία Ψηφοφορίας

Σχήμα 38: Φόρμα δημιουργίας νέας ψηφοφορίας – Ιδιωτική με κωδικό πρόσβασης .

Όταν οριστεί ο τύπος ψηφοφορίας ως προσκεκλημένοι μόνο ειδικά προσκεκλημένοι χρήστες μπορούν να συμμετάσχουν. Στην οθόνη εμφανίζεται νέο πεδίο για την επιλογή των προσκεκλημένων χρηστών με δυνατότητα προσθήκης μέσω σχετικού κουμπιού. Επίσης, υπάρχει ενημέρωση ότι δεν έχουν επιλεγεί χρήστες μέχρι να γίνει η προσθήκη τους από τον δημιουργό της ψηφοφορίας. Οι υπόλοιπες επιλογές της φόρμας παραμένουν ίδιες.

Δημιουργία Νέας Ψηφοφορίας

Τίτλος
π.χ. Εκλογή Προέδρου Συλλόγου

Περιγραφή (Προαιρετικά)
Δώστε μερικές λεπτομέρειες για τη ψηφοφορία...

Τύπος Ψηφοφορίας
■ Προσκεκλημένοι - Μόνο ειδικά προσκεκλημένοι

Μόνο προσκεκλημένοι χρήστες μπορούν να ψηφίσουν.

Κωδικός Πρόσβασης
π.χ. VOTE2025

Διαθέσιμο μόνο για ιδιωτικές ψηφοφορίες.

Προσκεκλημένοι Χρήστες
Επιλέξτε τους χρήστες που θα μπορούν να συμμετάσχουν στην ψηφοφορία.

+ Προσθήκη Χρηστών

Δεν έχουν επιλεγεί χρήστες ακόμα
Πατήστε "Προσθήκη Χρηστών" για να επιλέξετε

Ημερομηνίες Διεξαγωγής
📅 Επιλέξτε εύρος

Επιλογές Ψηφοφορίας
Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με καίμενο.

Τύπος Επιλογών
👤 Χρήστες Συστήματος ○ 🗳️ Επιλογές με Καίμενο

👤 Προσθήκη Υποψηφίου

Δημιουργία Ψηφοφορίας

Σχήμα 39: Φόρμα δημιουργίας νέας ψηφοφορίας – Μόνο ειδικά προσκεκλημένοι.

Στην ενότητα των προσκεκλημένων χρηστών εμφανίζεται πλέον λίστα με τους χρήστες που έχουν επιλεγεί για να συμμετάσχουν στη ψηφοφορία. Κάθε χρήστης εμφανίζεται με το όνομά του και το αντίστοιχο email, ενώ υπάρχει η δυνατότητα αφαίρεσης μέσω του κουμπιού διαγραφής δίπλα σε κάθε καταχώρηση.

Προσκεκλημένοι Χρήστες
Επιλέξτε τους χρήστες που θα μπορούν να συμμετάσχουν στην ψηφοφορία.

Επιλεγμένοι χρήστες (3):

Στάθης Μαυρίδης (sefeamav@gmail.com) ✕ Πέτρος Παπακυριακίδης (pap.petros31@gmail.com) ✕ Nikos Liolis (nikosliolis21@gmail.com) ✕

Ημερομηνίες Διεξαγωγής
📅 Επιλέξτε εύρος

Σχήμα 40: Προσθήκη προσκεκλημένων χρηστών.

Τέλος, όταν οριστεί ο τύπος ψηφοφορίας ως περιορισμένη, εμφανίζεται η ενότητα των κριτηρίων στόχευσης. Σε αυτήν ο δημιουργός μπορεί να ορίσει συγκεκριμένα φίλτρα για τους συμμετέχοντες όπως επάγγελμα, τοποθεσία, ελάχιστη και μέγιστη ηλικία, καθώς και φύλο. Με τον τρόπο αυτό η συμμετοχή περιορίζεται μόνο στους χρήστες που πληρούν τα κριτήρια αυτά.

Δημιουργία Νέας Ψηφοφορίας

Τίτλος

π.χ. Εκλογή Προέδρου Συλλόγου

Περιγραφή (Προαιρετικά)

Δώστε μερικές λεπτομέρειες για τη ψηφοφορία...

Τύπος Ψηφοφορίας

Περιορισμένη - Me targeting criteria

Ψηφοφορία με περιορισμούς (ηλικία, επάγγελμα, κλπ).

Κωδικός Πρόσβασης

π.χ. VOTE2025

Διαθέσιμο μόνο για ιδιωτικές ψηφοφορίες.

Ημερομηνίες Διεξαγωγής

Επιλέξτε εύρος

Κριτήρια Στόχευσης

Επάγγελμα Στόχευσης

-- Επιλέξτε Επάγγελμα --

Τοποθεσία Στόχευσης

-- Επιλέξτε Τοποθεσία --

Ελάχιστη Ηλικία

Ελάχιστη ημερομηνία γέννησης

ή εισάγετε ηλικία: π.χ. 1 έτη

Μέγιστη Ηλικία

Μέγιστη ημερομηνία γέννησης

ή εισάγετε ηλικία: π.χ. 6 έτη

Φύλο

-- Επιλέξτε Φύλο --

Επιλογές Ψηφοφορίας

Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με κείμενο.

Τύπος Επιλογών

Χρήστες Συστήματος

Επιλογές με Κείμενο

Προσθήκη Υποψηφίου

Δημιουργία Ψηφοφορίας

Σχήμα 41: Φόρμα δημιουργίας νέας ψηφοφορίας – Περιορισμένοι / targeting criteria.

7.10 Διαχείριση Υποψηφίων

Στις συγκεκριμένες διεπαφές ο χρήστης μπορεί να οργανώσει και να διαχειριστεί όλους τους υποψήφιους που θέλει να προσθέσει ή να αφαιρέσει από τις ψηφοφορίες ή εναλλακτικά το κείμενο που θέλει να προσθέσει σαν επιλογές στις ψηφοφορίες .

Επιλογές Ψηφοφορίας
Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με κείμενο.

Τύπος Επιλογών
☒ Χρήστες Συστήματος ☐ Επιλογές με Κείμενο

Σχήμα 42: Σελίδα διαχείρισης υποψηφίων – Κατάσταση χωρίς εγγεγραμμένους υποψηφίους.

Ηλεκτρονικές Ψηφοφορίες Αναζήτηση Ψηφοφορίας

Διαχείριση Υποψηφίων

Επιλογή Ψηφοφορίας
-- Επιλέξτε --

Υποψήφιοι
Δεν υπάρχουν υποψήφιοι.

Προσθήκη Υποψηφίου
Επιλέξε χρήστη για πρόσκληση

Αναζήτηση Χρηστών

Όνομα	Email	Θέση	Ενέργεια
Στάθης Μαυρίδης	sefeamav@gmail.com	-	✚ Προσθήκη
texnl emmonis	texnlthsemonis@gmail.com	student	✚ Προσθήκη
Chain Scriptor	cryptictiger39@gmail.com	doctor	✚ Προσθήκη

Μείνετε Συνδεδεμένοι
Εγγραφείτε στο ενημερωτικό μας δελτίο για τα τελευταία νέα και αποκλειστικές προσφορές.

Γρήγοροι Σύνδεσμοι
Αρχική
Σχετικά Με Εμάς
Υπηρεσίες
Προϊόντα
Επικοινωνία

Επικοινωνήστε Μαζί Μας
Οδός Καινοτομίας 123
Πόλη Τεχνολογίας, ΤΚ 12345
Τηλ: (+123) 456-7890
Ηλ. ταχυδρομείο: hello@example.com

Ακολουθήστε Μας

© 2024 Η Εταιρεία Σας. Με επιφύλαξη παντός δικαιώματος.

Πολιτική Απορρήτου Όροι Παροχής Υπηρεσιών Ρυθμίσεις Cookie

Σχήμα 43: Αναδυόμενο παράθυρο προσθήκης υποψηφίου – επιλογή από εγγεγραμμένους χρήστες.

Επιλογές Ψηφοφορίας

Επιλέξτε αν θέλετε να προσθέσετε χρήστες του συστήματος ως υποψηφίους ή απλά επιλογές με κείμενο.

Τύπος Επιλογών

☐  Χρήστες Συστήματος ☒  Επιλογές με Κείμενο

Προσθέστε τουλάχιστον 2 επιλογές. Μπορείτε να διαγράψετε πεδία αν έχετε περισσότερα από 2.

Επιλογή #1 (π.χ. Ναι, Όχι, Αποχή)



Επιλογή #2 (π.χ. Ναι, Όχι, Αποχή)



Επιλογή #3 (π.χ. Ναι, Όχι, Αποχή)



 Προσθήκη Επιλογής

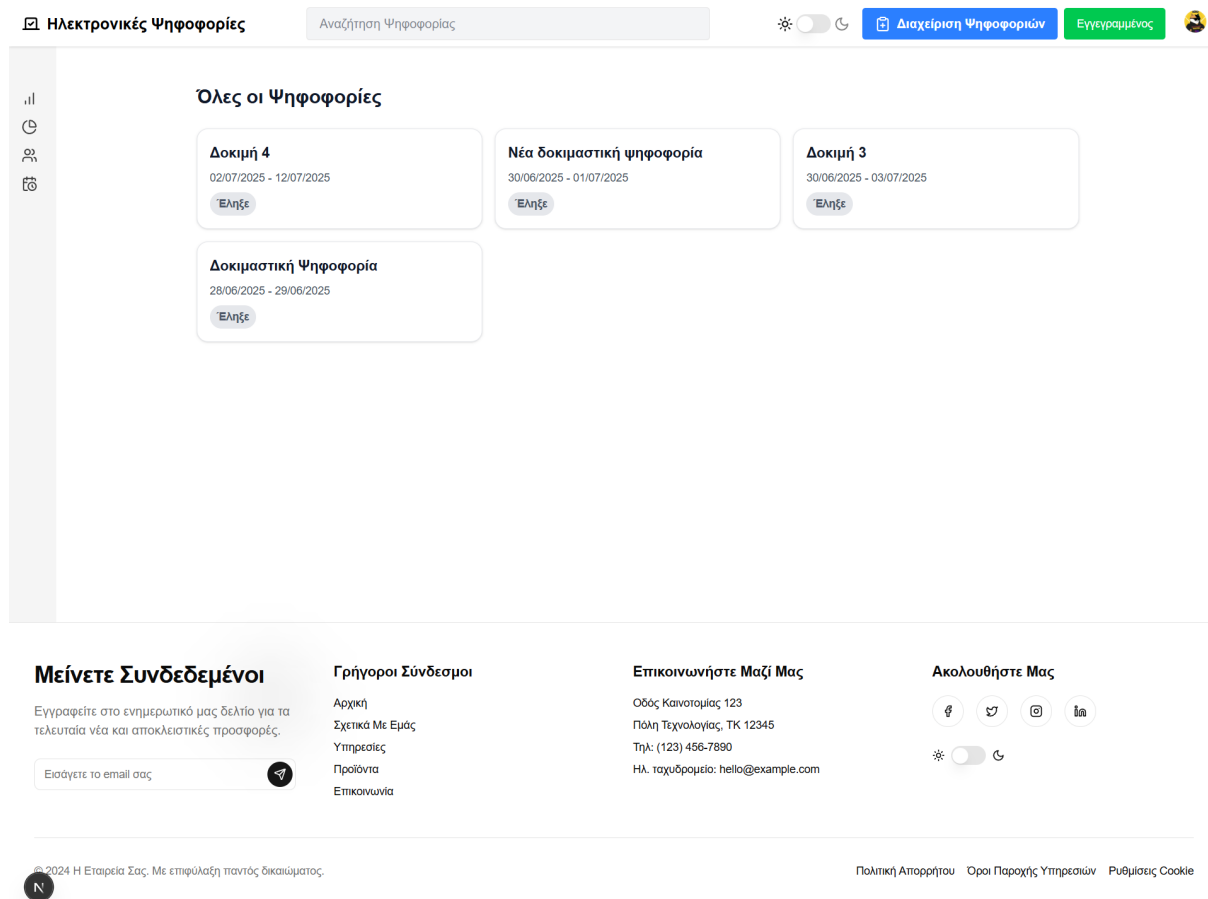
Δημιουργία Ψηφοφορίας

Σχήμα 44: Σελίδα διαχείρισης επιλογών κειμένου.

Η σχεδίαση της σελίδας έχει δημιουργηθεί με απλό τρόπο για να μπορεί να την διαχειριστεί ο μέσος χρήστης ώστε να διασφαλίζεται η ορθή καταχώρηση και διαχείριση των υποψηφίων ή των επιλογών κειμένου χωρίς όμως περιττές πληροφορίες ή πολύπλοκες διεπαφές.

7.11 Προβολή Όλων των Ψηφοφοριών

Η σελίδα προβολής όλων των ψηφοφοριών αποτελεί ένα από τα βασικά σημεία πλοήγησης για τον χρήστη στην πλατφόρμα ηλεκτρονικών ψηφοφοριών. Η σχεδιάσή της είναι προσαρμοσμένη στην απλότητα, την καθαρότητα και την άμεση ενημέρωση, προσφέροντας μια συνοπτική αλλά πλήρη εικόνα για όλες τις ψηφοφορίες που έχουν δημιουργηθεί στο σύστημα.



Σχήμα 45: Σελίδα με επισκόπηση όλων των ψηφοφοριών σε μορφή καρτών.

Στο επάνω μέρος της κύριας ενότητας, ο τίτλος «**Όλες οι Ψηφοφορίες**» οριοθετεί με σαφήνεια το περιεχόμενο της σελίδας και προετοιμάζει τον χρήστη για την επισκόπηση των διαθέσιμων εκλογικών διαδικασιών. Ακριβώς από κάτω, εμφανίζεται μια σειρά από κάρτες, καθεμία από τις οποίες αντιπροσωπεύει μια ξεχωριστή ψηφοφορία.

Η διάταξη των καρτών είναι οριζόντια και προσαρμόζεται δυναμικά στο πλάτος της οθόνης, διασφαλίζοντας την ευχρηστία και την αισθητική αρτιότητα της σελίδας σε οποιαδήποτε ανάλυση.

Κάθε κάρτα περιλαμβάνει:

- **Τίτλο ψηφοφορίας** με έντονη γραμματοσειρά στο επάνω μέρος.
- **Ημερομηνίες διεξαγωγής** ακριβώς από κάτω, σε μορφή *ημέρα/μήνας/έτος*.
- **Ετικέτα κατάστασης** στο κάτω μέρος, όπως «Έληξε», που υποδεικνύει την ολοκλήρωση της διαδικασίας.

Η χρήση ετικετών κατάστασης επιτρέπει την εύκολη αναγνώριση ενεργών, προγραμματισμένων ή ολοκληρωμένων διαδικασιών. Οι κάρτες έχουν καθαρή διάταξη και λειτουργούν ως σύνδεσμοι για την απόκτηση περισσότερων πληροφοριών.

Η εμπειρία χρήσης χαρακτηρίζεται από σαφήνεια, ταχύτητα και ευκολία πλοήγησης. Ο χρήστης μπορεί να ενημερωθεί άμεσα για όλες τις εκλογές στην πλατφόρμα και να αποκτήσει πρόσβαση σε αποτελέσματα ή λεπτομέρειες ψηφοφορίας, ενισχύοντας τη διαφάνεια και την εμπιστοσύνη στο σύστημα.

Η διακριτική χρήση έντονων στοιχείων (τίτλοι), καθαρής τυπογραφίας και λειτουργικών καρτών καθιστούν τη σελίδα προβολής ψηφοφοριών βασικό εργαλείο για την εποπτεία της εκλογικής δραστηριότητας.

8 Συμπεράσματα και μελλοντικές επεκτάσεις

Στο πλαίσιο της παρούσας πτυχιακής εργασίας σχεδιάστηκε και υλοποιήθηκε μια σύγχρονη διαδικτυακή εφαρμογή ηλεκτρονικών ψηφοφοριών που έχει ως στόχο την δημιουργία ενός ασφαλούς διαφανούς και αξιόπιστου συστήματος για τη διεξαγωγή ηλεκτρονικών ψηφοφοριών . Η εφαρμογή απευθύνεται τόσο σε απλούς χρήστες και τους προσφέρει ένα ολοκληρωμένο σύνολο εργαλείων για τη δημιουργία διαχείριση και συμμετοχή σε ψηφοφορίες, δίνοντας έμφαση στην εύκολη χρήση , και την προστασία των προσωπικών τους δεδομένων .

Η υλοποίηση βασίστηκε σε κάποιες σύγχρονες τεχνολογίες του web και χρησιμοποιήθηκαν βιβλιοθήκες αιχμής για το *frontend* και το *backend*. Η χρήση των μηχανισμών ταυτοποίησης συγκεκριμένα το *Clerk* με την επιπλέον προσθήκη στοιχείων από την πλευρά του χρήστη, διασφαλίζουν ότι η κάθε μια ψήφος είναι μοναδική και δεν μπορεί κάποιος χρήστης να ψηφίσει περισσότερες από μια φορές .

Η εφαρμογή προσφέρει στον χρήστη τη δυνατότητα να δημιουργεί νέες ψηφοφορίες στις οποίες θα μπορεί να ορίσει το χρονικό διάστημα που θα κρατήσει ενεργή μια εφαρμογή , το επάγγελμα που θα αφορά , τις ηλικίες που μπορούν να λάβουν μέρος και το φύλο . Επιπλέον, μπορεί να προσθέσει υποψηφίους σε μια ψηφοφορία οι οποίοι είναι εγγεγραμμένοι χρήστες του συστήματος . Τέλος μπορεί να λάβει μέρος σε ψηφοφορίες που τον αφορούν . Η διαδικασία όλων αυτών των βημάτων είναι εύκολη για τον απλό χρήστη καθώς είναι απλή και του παρέχονται μέσα απο το σύστημα αναλυτικές πληροφορίες σχετικά με το πως να κινηθεί μέσα στην πλατφόρμα .

Στο κοινωνικό επίπεδο η δημιουργία ενός τέτοιου συστήματος βοηθάει στην συμμετοχή περισσότερων ανθρώπων στην διαδικασία των ψηφοφοριών καθώς μειώνονται προβλήματα μετακίνησης και δίνεται η πρόσβαση σε όλους τους πολίτες να ψηφίσουν με απόλυτη διαφάνεια .

8.1 Συμπεράσματα

Η μελέτη και η δημιουργία ενός ηλεκτρονικού συστήματος ψηφοφοριών έδειξε την ανάγκη για την αξιοποίηση τεχνολογικών λύσεων ώστε να υπάρχει μεγαλύτερη συμμετοχή πολιτών καλύτερη ασφάλεια και βελτίωση της διαφάνειας στις ψηφοφορίες . Η εφαρμογή που υλοποιήθηκε , είναι ένα ολοκληρωμένο σύστημα ηλεκτρονικών ψηφοφοριών όπου ο χρήστης μπορεί να δημιουργήσει μια ψηφοφορία , να προσθέσει υποψηφίους καθώς και να συμμετέχει ενεργά σε ψηφοφορίες που τον αφορούν με βάση την ηλικία , το φύλο και το επάγγελμα του μέσα από μια φιλική διεπαφή .

Η ανάπτυξη της συγκεκριμένης πλατφόρμας έδειξε ότι η αλλαγή σε ψηφιακές διαδικασίες μπορεί να γίνει με ασφάλεια και διαφάνεια . Αυτό όμως προϋποθέτει να υλοποιηθεί με μεγάλη προσοχή και να δοθεί έμφαση στην ταυτοποίηση καθώς και να δημιουργήσει την εμπιστοσύνη των χρηστών ως προς το σύστημα , ώστε να υπάρξει η επιθυμητή συμμετοχή .

8.2 Μελλοντικές Επεκτάσεις και Προοπτικές

Υπάρχουν αρκετές μελλοντικές δυνατότητες βελτίωσης του συστήματος ηλεκτρονικού συστήματος ψηφοφοριών .Μία από αυτές είναι η προσθήκη πιο εξελιγμένων μεθόδων ταυτοποίησης όπως για παράδειγμα βιομετρικός έλεγχος ή χρήση επαλήθευσης μέσω εγγράφων όπως διαβατήριο , ταυτότητα και δίπλωμα . Επιπλέον η σύνδεση με κάποια επίσημη κρατική βάση δεδομένων θα μπορούσε να αυτοματοποιήσει την διαδικασία της ταυτοποίησης των χρηστών .

Παρότι η εφαρμογή που αναπτύχθηκε καλύπτει σε μεγάλο βαθμό τις βασικές ανάγκες μιας πλατφόρμας ηλεκτρονικών ψηφοφοριών υπάρχουν ακόμα αρκετά περιθώρια βελτίωσης ώστε να φτάσει στο κατάλληλο αποτέλεσμα . Από την λειτουργική πλευρά της εφαρμογής , μια σημαντική μελλοντική επέκταση είναι η δυνατότητα να προστεθούν ψηφοφορίες που δεν θα εμπεριέχουν υποψηφίους αλλά θεματικές ενότητες ώστε να καλύψει το ενδεχόμενο να δημιουργείται μια ψηφοφορία για θέματα που δεν αφορούν ανθρώπους αλλά διαφορετικές καταστάσεις .Επιπλέον μελλοντικά είναι σημαντικό να προστεθούν δυνατότητες όπως μαζική προσθήκη υποψηφίων σε πολλές ψηφοφορίες χωρίς να χρειάζεται ο χρήστης να προσθέτει ξεχωριστά τον κάθε υποψήφιο στην ψηφοφορία . Επιπρόσθετα ακόμα μια μελλοντική επέκταση είναι η αναλυτική περιγραφή των αποτελεσμάτων με διαγράμματα ώστε να μπορούν οι χρήστες να βλέπουν αναλυτικά τι έχει φέρει το συγκεκριμένο αποτέλεσμα σε κάθε ψηφοφορία . Τέλος θα μπορούσε να προστεθεί ένας πράκτορας τεχνητής νοημοσύνης ο οποίος θα απαντάει στις διάφορες ερωτήσεις των χρηστών σχετικά με το πώς λειτουργεί η εφαρμογή και να τους κατευθύνει σε όλα τα βήματα που χρειάζεται να κάνουν για να δημιουργήσουν μια ολοκληρωμένη ψηφοφορία .

Στην πλευρά της ασφάλειας και της διαφάνειας του συστήματος , μπορεί να γίνει αξιοποίηση τεχνολογιών όπως το blockchain και η ομομορφική κρυπτογράφηση. Επιπλέον η ανάπτυξη της εφαρμογής για χρήση σε κινήτο το οποίο θα λειτουργούσε με ειδοποιήσεις θα διευκόλυνε την συμμετοχή και την ενημέρωση των χρηστών άμεσα .

Επιπλέον, η πλατφόρμα μπορεί να προσαρμοστεί με τρόπο ώστε να καλύπτει κάποιες επιπλέον ανάγκες για να χρησιμοποιηθεί σε διάφορους οργανισμούς όπως πανεπιστήμια , συλλόγους , δήμους ή και ακόμα σε ιδιωτικές εταιρείες με δυνατότητες πολλών παράλληλων ψηφοφοριών .

Τέλος ίσως μια απο τις σημαντικότερες μελλοντικές προσθήκες για την εποχή μας είναι η ενσωμάτωση της τεχνητής νοημοσύνης ώστε να γίνεται ανάλυση αποτελεσμάτων , ανίχνευση ανωμαλιών και ίσως πρόβλεψη τιμών για την βελτίωση της διαδικασίας .

Συμπερασματικά,η παρούσα εργασία αποδεικνύει ότι η προσθήκη νέων σύγχρονων τεχνολογιών μπορεί να συμβάλει στην βελτίωση της συμμετοχής και στην διαφάνεια του αποτελέσματος των ψηφοφοριών .

Αναφορές

- [1] Ε. Αβραμίδου, “Συστήματα Ασφαλούς Ηλεκτρονικής Ψηφοφορίας,” *Μεταπτυχιακή Διατριβή*, Τεχνολογικό Εκπαιδευτικό Ίδρυμα Καβάλας, 2020. Available: <http://ikee.lib.auth.gr/record/122554/files/AVRAMIDOU.pdf>
- [2] M. Germann and U. Serdült, “Internet voting and turnout: Evidence from Switzerland,” *Electoral Studies*, 2017. Available: <https://michagermann.github.io/publication/germannserduelt2017/GermannSerduelt2017.pdf>
- [3] International IDEA, *The Use of Open Source Technology in Elections*, 2023. Available: <https://www.idea.int/sites/default/files/2023-02/the-use-of-open-source-technology-in-elections.pdf>
- [4] OSCE/ODIHR, *Handbook for the Observation of New Voting Technologies (NVT)*, 2013. Available: <https://www.osce.org/files/f/documents/0/6/104939.pdf>
- [5] D. Springall, T. Finkenauer, Z. Durumeric, et al., “Security Analysis of the Estonian Internet Voting System,” in *Proc. ACM CCS*, 2014. Available: <https://jhalderm.com/pub/papers/ivoting-ccs14.pdf>
- [6] H. Kim, M. Lee, and Y. Choi, “E-voting System Using Homomorphic Encryption and Blockchain Technology,” *arXiv preprint*, arXiv:2111.05096, 2021. Available: <https://arxiv.org/pdf/2111.05096>
- [7] J. A. Halderman and V. Teague, “The New South Wales iVote System: Security Failures and Verification Flaws,” *arXiv preprint*, arXiv:1504.05646, 2015. Available: <https://arxiv.org/abs/1504.05646>
[Ημερομηνία πρόσβασης 3 Ιουνίου 2025]
- [8] D. Chaum, P. Y. A. Ryan, and S. Schneider, “A Practical Voter-Verifiable Election Scheme,” in *Computer Security – ESORICS 2005*, Springer, 2005, pp. 118–139. Available: <https://scispace.com/pdf/a-practical-voter-verifiable-election-scheme-2jxmb19pbf.pdf>
- [9] U. Jafar, A. M. Qamar, A. A. Khan, and S. Ahmad, “Blockchain for Electronic Voting System—Review and Open Research Challenges,” *Sensors*, vol. 21, no. 16, 2021, Art. 5563. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC8434614/pdf/sensors-21-05563.pdf>
- [10] P. Pereira et al., “Breaking and Fixing Vote Privacy of the Estonian E-Voting Protocol,” *arXiv preprint*, 2023. Available: <https://arxiv.org/abs/2301.10839>
[Ημερομηνία πρόσβασης 4 Ιουνίου 2025]
- [11] B. Adida, “Helios: Web-based Open-Audit Voting,” in *Proc. 17th USENIX Security Symposium*, 2008. Available: https://www.usenix.org/legacy/event/sec08/tech/full_papers/adida/adida.pdf

- [12] European Commission, “Study on E-Voting practices in the EU,” DG JUST, May 2023. Available: https://commission.europa.eu/document/download/23076478-987e-4dea-b9ef-df5e0dfd6cd2_en?filename=DG_JUST_EVoting_Report_pdf.pdf
- [13] M. Germann, “Internet voting increases expatriate voter turnout,” *Electoral Studies*, 2021 (preprint). Available: <https://michagermann.github.io/publication/germann2021/Germann2021.pdf>
- [14] A. Petitpas, J. M. Jaquet, and P. Sciarini, “Does E-Voting matter for turnout, and to whom?” *Electoral Studies*, vol. 71, p. 102245, Jun. 2021. Available: <https://doi.org/10.1016/j.electstud.2020.102245>
[Ημερομηνία πρόσβασης 3 Ιουνίου 2025]
- [15] European Parliament STOA, “Blockchain and the General Data Protection Regulation,” Jun. 2019. Available: [https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU\(2019\)634445_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf)
- [16] W. Bokslag, M. de Vries, “Evaluating e-voting: theory and practice,” arXiv:1602.02509, 2016. Available: <https://arxiv.org/abs/1602.02509>
[Ημερομηνία πρόσβασης 3 Ιουνίου 2025]
- [17] Dylan Yaga, Peter Mell, Nik Roby, and Karen Scarfone. *Blockchain Technology Overview*. NIST Interagency/Internal Report (NIST IR) 8202, National Institute of Standards and Technology, 2018. <https://doi.org/10.6028/NIST.IR.8202>
- [18] R. Belenç̇ Saglam et al., “SLR on GDPR vs Public Blockchain Systems,” arXiv:2210.04541, 2022. Available: <https://arxiv.org/abs/2210.04541>
[Ημερομηνία πρόσβασης 4 Ιουνίου 2025]
- [19] C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, 2009. Available: <https://dl.acm.org/doi/10.1145/1536414.1536440>
[Ημερομηνία πρόσβασης 4 Ιουνίου 2025]
- [20] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On Data Banks and Privacy Homomorphisms,” *Foundations of Secure Computation*, pp. 169–180, Academic Press, 1978. <https://luca-giuzzi.unibs.it/corsi/Support/papers-cryptography/RAD78.pdf>
- [21] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A Survey on Homomorphic Encryption Schemes: Theory and Implementation,” *ACM Computing Surveys*, 2018. Preprint Available: <https://arxiv.org/pdf/1704.03578>
- [22] IEEE Computer Society. *Software requirements*. Wikipedia, the free encyclopedia (accessed 2025-08-17), based on IEEE Standard Glossary of Software Engineering Terminology. Definition: “a condition or capability needed by a stakeholder to solve a problem or achieve an objective.” :contentReference[oaicite:4]index=4 https://www.informatik.htw-dresden.de/~hauptman/SEI/IEEE_Standard_Glossary_of_Software_Engineering_Terminology%20.pdf

- [23] H. Washizaki (ed.), *Guide to the Software Engineering Body of Knowledge (SWEBOK Guide), Version 4.0a*. IEEE Computer Society, 2025. Available: <https://ieeecs-media.computer.org/media/education/swebok/swebok-v4.pdf>
- [24] Hellenic Parliament. *Σύνταγμα της Ελλάδας, Άρθρο 51*. Official Portal of the Hellenic Parliament. Available: <https://www.hellenicparliament.gr/Vouli-ton-Ellinon/To-Politevma/Syntagma/article-51/>
[Ημερομηνία πρόσβασης 5 Ιουνίου 2025]
- [25] G. Tsoukalas, K. Papadimitriou, P. Louridas, and P. Tsanakas, “From Helios to Zeus,” *USENIX Journal of Election Technology and Systems (JETS)*, vol. 1, no. 1, Aug. 2013. Available: <https://www.usenix.org/system/files/jets/issues/jets-0101-tsoukalas.pdf>
- [26] U.S. Election Assistance Commission. *Voluntary Voting System Guidelines (VVSG) 2.0*. Election Assistance Commission, February 10, 2021. Encompasses requirements for testing & certification, accessibility (WCAG / Section 508), transparency (open-source), voter privacy, re-voting provisions, and auditability. Available: https://www.eac.gov/sites/default/files/TestingCertification/Voluntary_Voting_System_Guidelines_Version_2_0.pdf
- [27] Asimakopoulos, G.; Antonopoulou, H.; Giotopoulos, K.; Halkiopoulos, C. *Impact of Information and Communication Technologies on Democratic Processes and Citizen Participation*. Societies, 2025, 15(2), 40. Available: <https://doi.org/10.3390/soc15020040>
[Ημερομηνία πρόσβασης 5 Ιουνίου 2025]
- [28] Council of Europe. *E-voting: Ensuring accessibility and addressing the digital divide*. Council of Europe, 2020. Available: <https://www.coe.int/en/web/electoral-assistance/e-voting>
[Ημερομηνία πρόσβασης 12 Ιουνίου 2025]
- [29] Wendehorst, C.; Duller, Y. *Biometric Recognition and Behavioural Detection: Assessing the ethical aspects of biometric techniques*. Study requested by the European Parliament (JURI & PETI Committees), August 2021. Available: [https://www.europarl.europa.eu/RegData/etudes/STUD/2021/696968/IPOL_STU\(2021\)696968_EN.pdf](https://www.europarl.europa.eu/RegData/etudes/STUD/2021/696968/IPOL_STU(2021)696968_EN.pdf)
- [30] J. Han, M. Kim, and D. Lee, “A systematic review of modern code editors and IDEs,” in *Proc. ACM Symposium on the Foundations of Software Engineering (FSE)*, 2020. Available: <https://dl.acm.org/doi/pdf/10.1145/3368089.3409711>
[Ημερομηνία πρόσβασης 12 Ιουνίου 2025]
- [31] Ayezabu Amanuel, “Supabase vs Firebase: Evaluation of performance and development of Progressive Web Apps,” Bachelor of Engineering Thesis, Metropolia University of Applied Sciences, Finland, May 20, 2022. Available: https://www.theseus.fi/bitstream/handle/10024/771009/Ayezabu_Amanuel.pdf?sequence=2
- [32] Josef Spillner, “Quantitative Analysis of Cloud Function Evolution in the AWS Serverless Application Repository,” arXiv preprint arXiv:1905.04800, 2019. Available: <https://arxiv.org/pdf/1905.04800.pdf>

- [33] Prisma. *Next-generation Node.js and TypeScript ORM*. Prisma Official Website, 2024. Διαθέσιμο: <https://www.prisma.io/orm> [Ημερομηνία πρόσβασης 28 Ιουνίου 2025]
- [34] Prisma. *Prisma on GitHub*. GitHub Repository, 2024. Διαθέσιμο: <https://github.com/prisma/prisma> [Ημερομηνία πρόσβασης 2 Ιουλίου 2025]
- [35] Peter P. Chen, “The Entity–Relationship Model—Toward a Unified View of Data,” *ACM Transactions on Database Systems*, vol. 1, no. 1, pp. 9–36, 1976. Open copy: <https://dspace.mit.edu/bitstream/handle/1721.1/47432/entityrelationshx00chen.pdf>
- [36] Ανοικτό Πανεπιστήμιο Ιονίου, Σημειώσεις Μαθήματος ”Συστήματα Διαχείρισης Βάσεων Δεδομένων και Εφαρμογές τους στο Διαδίκτυο”, σελ. 27–29. Διαθέσιμο: [https://opencourses.ionio.gr/modules/document/file.php/DDI169/1.%20Σ%202025%20\(skaragiannis@ionio.gr\)/Λ/B/24-0554-01_Systimata-Diacheirisis-Baseon-Dedomenon-kai-Efarmoges-tous-sto-Diadiktyo_B-EPAL_Vivlio-Mathiti.pdf](https://opencourses.ionio.gr/modules/document/file.php/DDI169/1.%20Σ%202025%20(skaragiannis@ionio.gr)/Λ/B/24-0554-01_Systimata-Diacheirisis-Baseon-Dedomenon-kai-Efarmoges-tous-sto-Diadiktyo_B-EPAL_Vivlio-Mathiti.pdf)
- [37] Microsoft. *TypeScript Documentation*. Microsoft Docs, 2025. Available: <https://www.typescriptlang.org/docs> [Ημερομηνία πρόσβασης 14 Ιουλίου 2025]
- [38] A. Mesbah, S. Mirshokraie, and B. Vasilescu, “Automated analysis of CSS code quality: Challenges and opportunities,” in *Proc. IEEE/ACM International Conference on Software Engineering (ICSE)*, 2019. Available: <https://arxiv.org/pdf/1902.04560>
- [39] Microsoft. *TypeScript Handbook*. Microsoft Docs, 2025. Available: <https://www.typescriptlang.org/docs/handbook/intro.html> [Ημερομηνία πρόσβασης 21 Ιουλίου 2025]
- [40] Prisma. *Prisma Documentation*. Prisma Docs, 2024. Available: <https://www.prisma.io/docs> [Ημερομηνία πρόσβασης 3 Αυγούστου 2025]
- [41] Radix UI. *Radix UI Documentation*. Radix UI, 2025. Available: <https://www.radix-ui.com> [Ημερομηνία πρόσβασης 3 Αυγούστου 2025]
- [42] Lucide. *Lucide React Icons*. GitHub Repository, 2025. Available: <https://github.com/lucide-icons/lucide> [Ημερομηνία πρόσβασης 3 Αυγούστου 2025]
- [43] Framer. *Framer Motion Documentation*. Framer, 2025. Available: <https://www.framer.com/motion> [Ημερομηνία πρόσβασης 3 Αυγούστου 2025]
- [44] Cmdk. *Cmdk Documentation*. Cmdk, 2025. Available: <https://cmdk.xyz> [Ημερομηνία πρόσβασης 7 Αυγούστου 2025]

- [45] TanStack. *React Query - Official Documentation*. TanStack, 2025. Available: <https://tanstack.com/query/v5/docs/react>
[Ημερομηνία πρόσβασης 7 Αυγούστου 2025]
- [46] TanStack. *useMutation React Hook Documentation*. TanStack, 2022. Available: <https://tanstack.com/query/v5/docs/react/reference/useMutation>
[Ημερομηνία πρόσβασης 7 Αυγούστου 2025]
- [47] J. Wang, “A Survey of Web Caching Schemes for the Internet,” Cornell University Technical Report, 1999. Available: <https://jia-wang.org/wp-content/uploads/2018/03/web-survey-techrep.pdf>
- [48] E. Murphy-Hill, C. Parnin, and A. Bird, “How We Program with React,” in *Proc. IEEE/ACM International Conference on Software Engineering (ICSE)*, 2020. Available: <https://arxiv.org/pdf/2001.08261>
- [49] Clerk. *SignUp Component - Next.js Documentation*. Clerk Documentation, 2025. Available: <https://clerk.com/docs/nextjs/components/authentication/sign-up>
[Ημερομηνία πρόσβασης 17 Αυγούστου 2025]
- [50] Clerk. *JavaScript Backend SDK - Clerk Documentation*. Clerk Documentation, 2025. Available: <https://clerk.com/docs/references/backend/overview>
[Ημερομηνία πρόσβασης 2 Σεπτεμβρίου 2025]
- [51] M. Stonebraker and L. A. Rowe, “The design of POSTGRES,” *ACM Transactions on Database Systems*, vol. 10, no. 4, pp. 286–316, 1986. Available: <https://db.cs.berkeley.edu/papers/ERL-M85-95.pdf>
- [52] S. Pati, Y. Zaki, “A Comparative Analysis with React.js on Performance, SEO, and Global Network Equity,” arXiv preprint arXiv:2502.15707, May 2025. Διαθέσιμο: <https://arxiv.org/abs/2502.15707> [Ημερομηνία πρόσβασης 9 Σεπτεμβρίου 2025]
- [53] D. Zerihun, “React.js vs. Next.js,” Bachelor’s Thesis, Håme University of Applied Sciences, 2022. Διαθέσιμο: https://www.theseus.fi/bitstream/handle/10024/750122/Dinku_Zerihun.pdf
- [54] S. Tilkov and S. Vinoski, “Server-side rendering vs. client-side rendering in modern web applications,” *IEEE Software*, vol. 37, no. 3, pp. 88–92, 2020. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9091234>