**Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές**

**Καζλάρης Ιωάννης, Παναγιωτίδης Ιωάννης, Εμμανουηλίδης Παναγιώτης, Τασιός Χρυσόστομος**

Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

Διεθνές Πανεπιστήμιο της Ελλάδος

**Σίνδος Δεκέμβριος 2022**

ΠΕΡΙΕΧΟΜΕΝΑ

1. **Εισαγωγή 3**
2. Μεθοδολογία Scrum 4
3. **Οργάνωση ομάδας και αναθέσεις αρμοδιοτήτων5**
4. Οι λειτουργικές απαιτήσεις της εφαρμογής 6
5. Διάγραμμα Περιπτώσεων Χρήσης της Εφαρμογής 7
6. Mockups 8
7. Βιβλιογραφικές αναφορές11
8. **Εισαγωγή**

Στα πλαίσια του μαθήματος “Μηχανική Λογισμικού για Διαδικτυακές Εφαρμογές” του μεταπτυχιακού προγράμματος σπουδών «Ευφυείς τεχνολογίες διαδικτύου» του Διεθνούς Πανεπιστημίου Θεσσαλονίκης συστάθηκε η ομάδα «**WebEng22\_Group2**» η οποία αποτελείται από τους ακόλουθους φοιτητές, σε αλφαβητική σειρά:

* ΕΜΜΑΝΟΥΗΛΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ
* ΚΑΖΛΑΡΗΣ ΙΩΑΝΝΗΣ
* ΠΑΝΑΓΙΩΤΙΔΗΣ ΙΩΑΝΝΗΣ
* ΤΑΣΙΟΣ ΧΡΥΣΟΣΤΟΜΟΣ

Σκοπός της παρούσας εργασίας είναι η ανάπτυξη ενός web application που να διαχειρίζεται δεδομένα που αφορούν στις βάσεις εισαγωγής των Πανεπιστημίων και τα αντίστοιχα στατιστικά στοιχεία τους.

Στόχος είναι η ανάπτυξη ενός **data/user driven web application** το οποίο ανακτά δεδομένα από το WEB API και τα χρησιμοποιεί λαμβάνοντας υπ’ όψιν τις ελάχιστες μη λειτουργικές απαιτήσεις οι οποίες είναι:

1. χρήση μιας βάσης δεδομένων όπου η εφαρμογή σας θα αποθηκεύει και θα ανακτά δεδομένα,
2. η εφαρμογή να είναι user/data driven, δηλαδή ο χρήστης(ες) ταυτοποιείται και θα κάνει κάποιες ενέργειες με τα δεδομένα του συστήματος και
3. θα πρέπει να υπάρχει ένα βασικό UX Design/Testing.

Η εφαρμογή αναπτύχθηκε σε συνεργασία με όλα τα μέλη της ομάδας μέσω του συστήματος source/versioning control Git και του Github στο ακόλουθο url, όπου και υπάρχουν όλα τα απαιτούμενα στοιχεία της υλοποιηθείσας εφαρμογής: <https://github.com/iee-ihu-gr-coursem102/WebEng22_Group2>

1. **Μεθοδολογία Scrum**

Για την εκπόνηση του project η ομάδα μας ακολούθησε την μεθοδολογία Scrum, η οποία είναι μια σύγχρονη ευέλικτη μέθοδος διαχείρισης έργων.

Όπως γνωρίζουμε η μεθοδολογία της Scrum χρησιμοποιείται σε πλήθος έργων ως επαναληπτική και αυξητική μέθοδος ανάπτυξης έργων, χρησιμοποιείται σαν μεθοδολογία ανάπτυξης λογισμικού, όσον αφορά την οργάνωση και λειτουργία των ομάδων ανάπτυξης αλλά και στον σχεδιασμό και διαχείριση των ευέλικτων (agility) έργων.

Με τη χρήση της Scum επιτυγχάνεται η διαδικασία ανάπτυξης ενός προϊόντος, σε όλες τις επί μέρους φάσεις, που συνήθως επικαλύπτονται και εκτελείται από μία λειτουργική «ομάδα». Προσδίδοντας ευελιξία στην οργάνωση και διαχείριση της διαδικασίας ανάπτυξης λογισμικού, αυξάνεται η παραγωγικότητα της ομάδας αλλά και η ικανότητα προσαρμογής της σύμφωνα με τις εκάστοτε ανάγκες που προκύπτουν. Ακόμα η μέθοδος αυτή επιδιώκει όσο το δυνατόν μικρότερο χρονικό διάστημα κύκλου ανάπτυξης και παράδοση τμημάτων κώδικα του συστήματος που είχαν συμφωνηθεί ανά κύκλο. Με αυτό τον τρόπο παραδίδεται στον πελάτη κάποιο μέρος ανά κύκλο το οποίο να είναι μερικώς εκτελέσιμο και να μπορεί να δουλέψει αυτό το μέρος του έργου που παραδόθηκε, χωρίς να περιμένει την πλήρη υλοποίηση του που ίσως να καθυστερήσει.

Η μέθοδος του Scrum διαχωρίζεται σε 3 σημαντικές φάσεις (Schwaber and Beedle, 2008):

1. **Αρχική Διερεύνηση (pre game)**
2. **Σχεδιασμός (game)**
3. **Ολοκλήρωση (post game)**

Στη **Φάση Αρχικής Διεύρυνσης** διακρίνουμε δύο υποκατηγορίες:

* 1. **Φάση Ανάλυσης**, όπου γίνεται ο καθορισμός των προδιαγραφών, οι απαιτήσεις, η δημιουργία του καταλόγου χαρακτηριστικών (product backlog list), προτεραιότητα απαιτήσεων, και η πιστοποίηση του συστήματος που πρόκειται να αναπτυχθεί. σύστημα.
  2. **Φάση υψηλού επιπέδου σχεδιασμού**, όπου υλοποιούνται η αρχιτεκτονική και ο σχεδιασμός του συστήματος με βάση τις αρχικές προδιαγραφές του συστήματος οι οποίες έχουν ήδη καθοριστεί στη προηγούμενη φάση.

Στη **Φάση Ανάπτυξης** σε κανονικές συνθήκες το σύστημα αναπτύσσεται σε πολλές επαναλήψεις (Sprints). Οι επαναλήψεις είναι επαναληπτικοί κύκλοι ανάπτυξης όπου ο κάθε κύκλος συμπεριλαμβάνει όλες τις φάσεις ανάπτυξης. Σε κάθε επανάληψη υλοποιούνται συγκεκριμένες απαιτήσεις, που ονομάζονται απαιτήσεις της επανάληψης (sprint backlog), και έχουν επιλεχθεί από τον κατάλογο των συνολικών απαιτήσεων του προϊόντος, με την βοήθεια του πελάτη. Οι απαιτήσεις αυτές πρέπει θα υλοποιηθούν μέσα στο χρονικό διάστημα της τρέχουσας επανάληψης. Η κάθε επανάληψη διαρκεί ένα συγκεκριμένο χρονικό διάστημα που κυμαίνεται από μία εβδομάδα έως ένα μήνα. Στο τέλος κάθε επανάληψης η ομάδα πρέπει να έχει "ένα εκτελέσιμο παραδοτέο προϊόν". Ένα έργο ανάπτυξης λογισμικού συνήθως υλοποιείται και παραδίδεται στον πελάτη σε τρεις έως οκτώ επαναλήψεις. Στη δική μας περίπτωση, θα υλοποιηθεί το σύστημα σε δύο φάσεις λόγω των χρονικών περιορισμών.

Τέλος, στη **φάση της ολοκλήρωσης** εκτελούνται όλες εκείνες οι διαδικασίες που απαιτούνται για τη περάτωση και την τελική παράδοσή του έργου στον πελάτη. Η ομάδα ανάπτυξης αφού επιβεβαιώσει, πάντα σε συνεργασία με τον πελάτη, ότι όλες οι απαιτήσεις έχουν υλοποιηθεί, κινεί τις διαδικασίες παράδοσης του συστήματος.

1. **Οργάνωση ομάδας και αναθέσεις αρμοδιοτήτων**

Λόγω του περιορισμού των ατόμων, η οργάνωση της ομάδας και η ανάθεση ρόλων σύμφωνα με τη μεθοδολογία Scrum επιλέχθηκαν και συμπτύχθηκαν ορισμένοι ρόλοι προκειμένου να γίνει η εφαρμογή της μεθοδολογίας.

Πιο συγκεκριμένα οι ρόλοι που υπήρξαν ήταν οι εξής:

* **product/project manager**
* **front end developer**
* **back-end developer**
* **UX tester**

Αφού έγινε συνάντηση των μελών και υπήρξε ανάπτυξη των δεξιοτήτων του κάθε μέλους της ομάδας, ομόφωνα καταλήξαμε στην ακόλουθη ανάθεση των ρόλων:

|  |  |  |  |
| --- | --- | --- | --- |
| product/project manager | front end developer | back-end developer | UX tester |
| ΚΑΖΛΑΡΗΣ  ΙΩΑΝΝΗΣ | ΠΑΝΑΓΙΩΤΙΔΗΣ ΙΩΑΝΝΗΣ | ΤΑΣΙΟΣ ΧΡΥΣΟΣΤΟΜΟΣ | ΕΜΜΑΝΟΥΗΛΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ |

1. **Οι λειτουργικές απαιτήσεις της εφαρμογής**

Κατόπιν σύσκεψης και διεργασιών από την ομάδα τόσο αρχικά όσο και σε καθημερινή βάση, η ομάδα κατάληξε στις παρακάτω λειτουργικές απαιτήσεις της εφαρμογής που σκοπεύουμε να υλοποιήσουμε, με προτεραιότητα που καθορίζεται αμέσως μετά με τον χρωματισμό, ενώ στην διπλανή στήλη θα εμφανίζονται τα αρχικά του κάθε μέλους που προέβη στην υλοποίηση της κάθε απαίτησης:

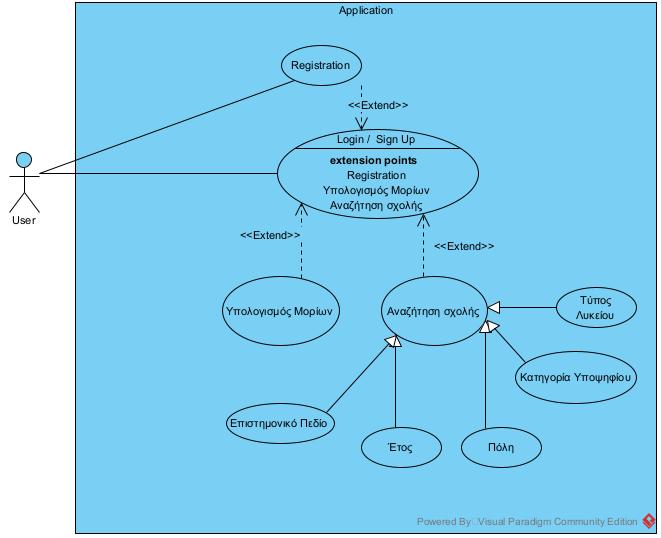
|  |
| --- |
| Μικρή |
| Μεσαία |
| Μεγάλη |

* ΚΑΖΛΑΡΗΣ ΙΩΑΝΝΗΣ → **ΚΙ**
* ΠΑΝΑΓΙΩΤΙΔΗΣ ΙΩΑΝΝΗΣ → **ΠΙ**
* ΤΑΣΙΟΣ ΧΡΥΣΟΣΤΟΜΟΣ → **ΤΧ**
* ΕΜΜΑΝΟΥΗΛΙΔΗΣ ΠΑΝΑΓΙΩΤΗΣ → **ΕΠ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Α/Α | Περιγραφή | Έχει υλοποιηθεί; (Ναι/Όχι) | Ποιος την υλοποιεί | Παρατηρήσεις |
| 1 | Δημιουργία αρχικής σελίδας index.html | Ναι | Καζλάρης Ιωάννης / Παναγιωτίδης Ιωάννης / Εμμανουηλίδης Παναγιώτης |  |
| 2 | Δημιουργία φόρμας για την εγγραφή χρήστη (sign up form) | Ναι | Καζλάρης Ιωάννης |  |
| 3 | Έλεγχος εισαγωγής μη κενού ονόματος χρήστη (username) στην φόρμα εγγραφής | Ναι | Καζλάρης Ιωάννης |  |
| 4 | Έλεγχος εισαγωγής ορθής μορφής email | Ναι | Καζλάρης Ιωάννης |  |
| 5 | Έλεγχος εισαγωγής μη κενού κωδικού στην φόρμα εγγραφής | Ναι | Καζλάρης Ιωάννης |  |
| 6 | Έλεγχος ότι ο πρώτος και ο δεύτερος κωδικός είναι ίδιοι | Ναι | Καζλάρης Ιωάννης |  |
| 7 | Λειτουργικότητα που επιτρέπει στον χρήστη να ελέγξει τι πληκτρολογεί στην φόρμα εγγραφής (eye icon) | Ναι | Καζλάρης Ιωάννης |  |
| 8 | Έλεγχος ότι ο χρήστης έχει συμφωνήσει με τους όρους πριν γίνει η εγγραφή | Όχι | Καζλάρης Ιωάννης |  |
| 9 | Προσθήκη μηνύματος ότι η εγγραφή του χρήστη έχει πραγματοποιηθεί επιτυχώς ή όχι | Όχι | Καζλάρης Ιωάννης / Τασιός Χρυσόστομος |  |
| 10 | Δημιουργία φόρμας για την είσοδο χρήστη (login form) | Ναι | Καζλάρης Ιωάννης |  |
| 11 | Λειτουργικότητα που επιτρέπει στον χρήστη να ελέγξει τι πληκτρολογεί στην φόρμα εισόδου (eye icon) | Ναι | Καζλάρης Ιωάννης |  |
| 12 | Έλεγχος ότι τα διαπιστευτήρια που έχει εισάγει ο χρήστης υπάρχουν στην βάση δεδομένων (ταυτοποίηση) | Όχι | Καζλάρης Ιωάννης / Τασιός Χρυσόστομος / Παναγιωτίδης Ιωάννης |  |
| 13 | Δημιουργία πίνακα (π.χ. Users) με τα στοιχεία των χρηστών | Όχι | Παναγιωτίδης Ιωάννης |  |
| 14 | Προσθήκη λειτουργικότητας απομνημόνευσης κωδικού | Όχι | Καζλάρης Ιωάννης |  |
| 15 | Προσθήκη λειτουργικότητας εξόδου του χρήστη | Όχι | Καζλάρης Ιωάννης |  |
| 16 | Έλεγχος επικοινωνίας front-end με back-end | Ναι | Τασιός Χρυσόστομος / Καζλάρης Ιωάννης / Παναγιωτίδης Ιωάννης / Εμμανουηλίδης Παναγιώτης |  |
| 17 | Δημιουργία υποδομής εισαγωγής - εξαγωγής πληροφορίαςΕμφάνιση πινάκων στην ιστοσελίδα με την χρήση εντολών HTTP GET | Όχι | Καζλάρης Ιωάννης / Παναγιωτίδης Ιωάννης / Εμμανουηλίδης Παναγιώτης |  |
| 18 | Αποστολή αιτημάτων HTTP requests (POST, GET) | Όχι | Παναγιωτίδης Ιωάννης / Καζλάρης Ιωάννης / Παναγιωτίδης Ιωάννης / Εμμανουηλίδης Παναγιώτης |  |
| 19 | Φιλτράρισμα αποτελεσμάτων των εμφανιζόμενων αποτελεσμάτων είτε σε front-end απ' ευθείας, είτε στο back-end από τους σχετικούς πίνακες με χρήση PHP/SQL | Όχι | Τασιός Χρυσόστομος / Καζλάρης Ιωάννης / Παναγιωτίδης Ιωάννης / Εμμανουηλίδης Παναγιώτης |  |
| 20 | UI / UX σχεδίαση εφαρμογής (δημιουργία αρχικών mock ups) | Ναι | Τασιός Χρυσόστομος / Εμμανουηλίδης Παναγιώτης |  |
| 21 | Έλεγχος ενδεικτική λειτουργικότητας (beta testing) | Όχι | Εμμανουηλίδης Παναγιώτης |  |
| 22 | Συγγραφή λειτουργικών απαιτήσεων | Ναι | Εμμανουηλίδης Παναγιώτης |  |
| 23 | Συγγραφή ενδιάμεσου παραδοτέου | Ναι | Εμμανουηλίδης Παναγιώτης |  |
| 24 | Συγγραφή τεκμηρίωσης - εγχειριδίου χρήσης | Όχι | Εμμανουηλίδης Παναγιώτης |  |

1. **Διάγραμμα Περιπτώσεων Χρήσης της Εφαρμογής**

Χρησιμοποιώντας το πρόγραμμα Visual Paradigm και έχοντας κατά νου όλες τις απαιτήσεις που θα υλοποιεί ο Χρήστης σχεδιάστηκε το ακόλουθο Use Case Diagramm για το τρέχον διάστημα υλοποίησης (Sprint 1) και το οποίο εν συνεχεία δύναται να τροποποιηθεί – εμπλουτιστεί σύμφωνα με τις επικείμενες βελτιώσεις που μπορεί να επιδεχθεί η εφαρμογή.

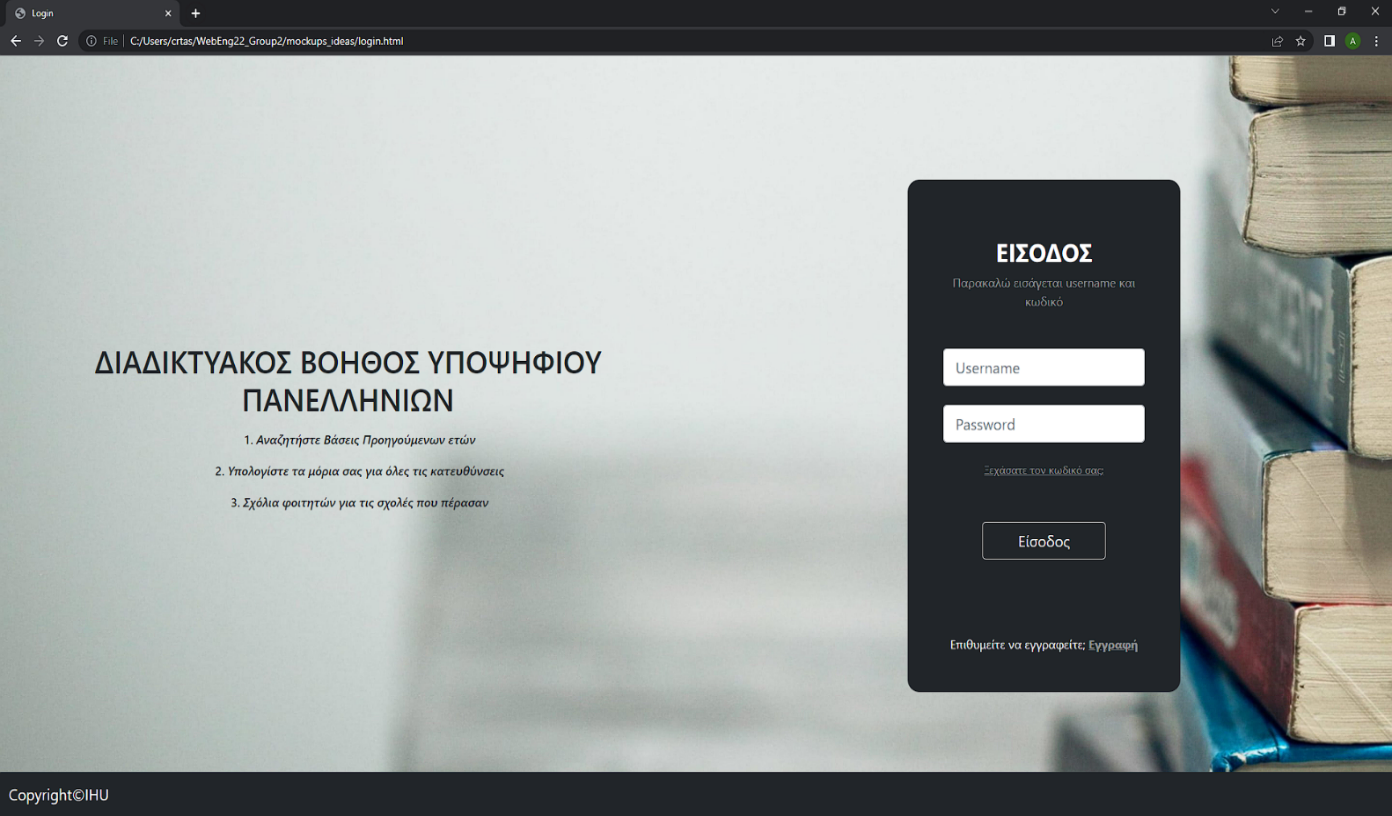


*Διάγραμμα Περίπτωσης Χρήσης*

1. **Mockups**

Από τις συνεχείς συζητήσεις που υπήρξαν από την ομάδα μας, καταλήξαμε στην ακόλουθη σχεδίαση και τις απεικονίσεις που θα έχει η εφαρμογή που υλοποιούμε στο αρχικό στάδιο. Η υλοποίηση ουσιαστικά θεωρούμε πως χωρίζεται σε τρία σκέλη που έχουμε να επιδείξουμε προς τον υποτιθέμενο πελάτη.

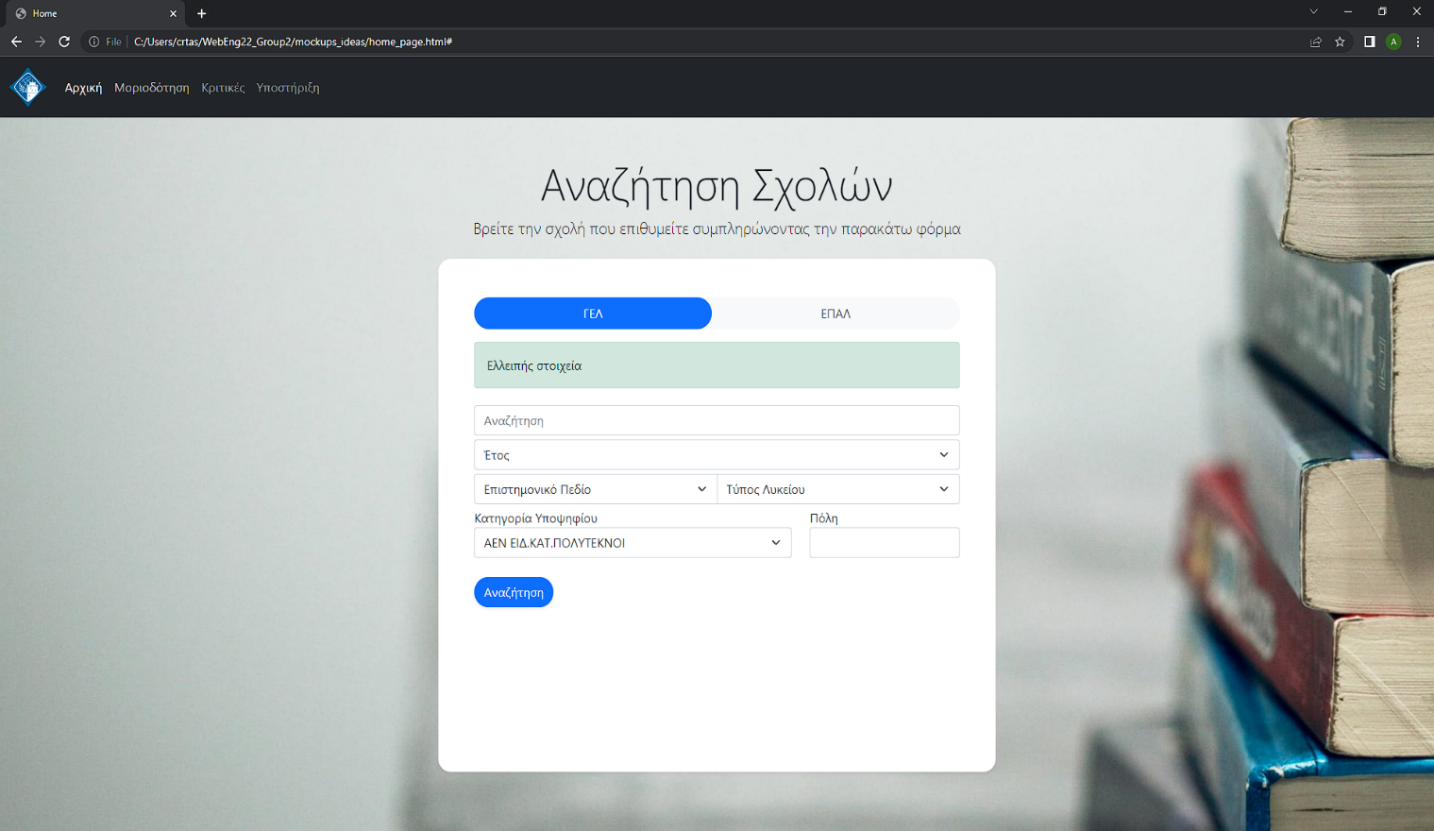
**Αρχικά:**



*Εικόνα Νο 1*

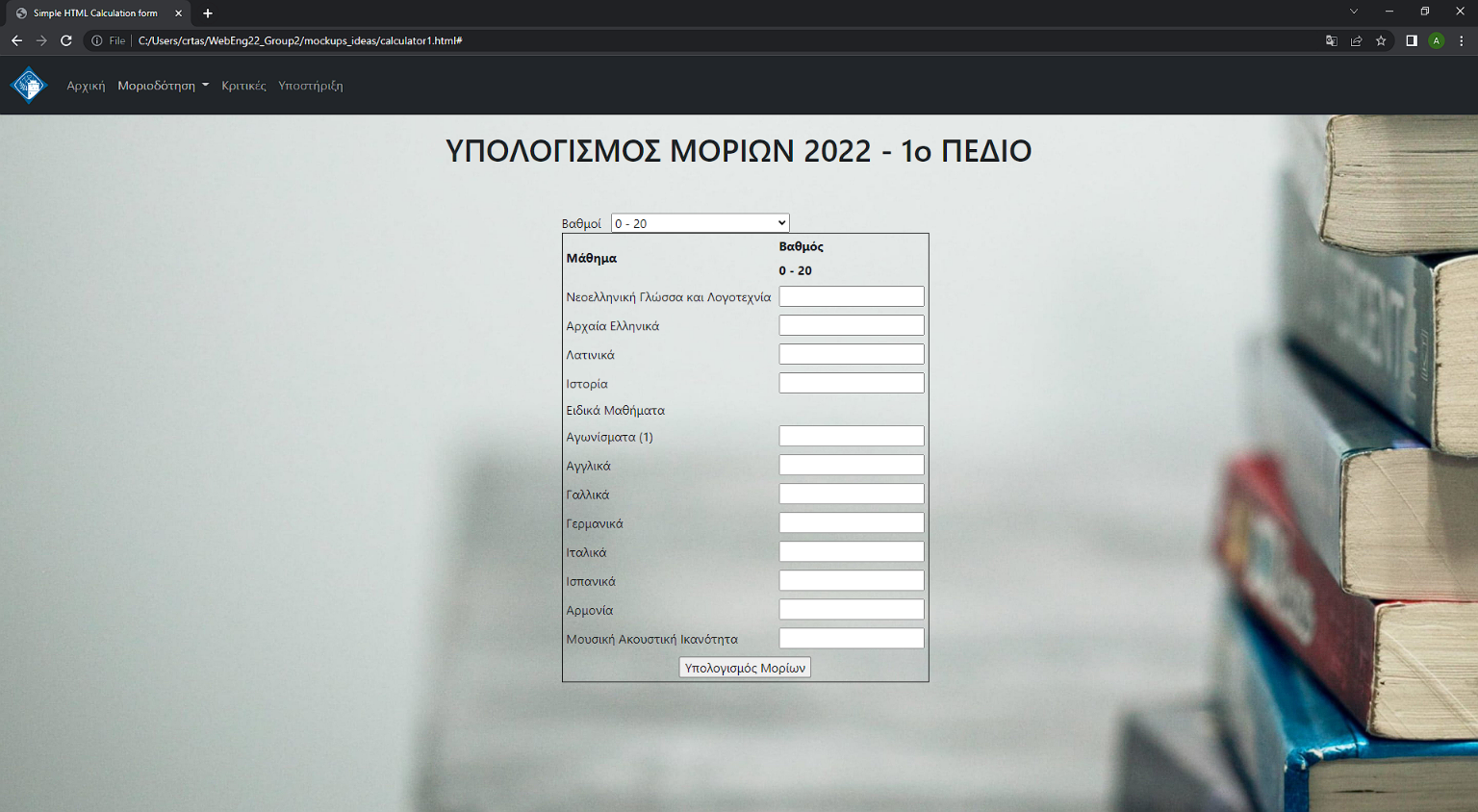
Θα υπάρχει μια εισαγωγική σελίδα (εικόνα Νο. 1) για να κάνει Login ή Register ο υποψήφιος μαθητής - χρήστης στην εφαρμογή, και να μπορεί να προβαίνει σε ορισμένες ενέργειες.

**Ενδιάμεσα**:



*Εικόνα Νο. 2*

Μετά την είσοδο του χρήστη, θα εμφανίζεται η παραπάνω σελίδα (εικόνα Νο. 2) στην οποία θα μπορεί να επιλέξει ανάμεσα σε ΓΕΛ ή ΕΠΑΛ καθώς και κάποιες ακόμα κατηγορίες (ανά έτος, πόλη, επιστημονικό πεδίο, κλπ.), ώστε να μπορεί εύκολα να εντοπίσει τις σχολές ενδιαφέροντος του με τα δοθέντα φίλτρα.



*Εικόνα Νο. 3*

**Τέλος,** θα υπάρχει η σελίδα που θα αφορά τον υπολογισμό των μορίων του μαθητή, με βάση τους βαθμούς που είχε επιτύχει στις Πανελλαδικές εξετάσεις, πράγμα το οποίο θα μπορεί κατ’ εκτίμηση να του δώσει τη δυνατότητα να εντοπίσει τις πιθανές σχολές που θα μπορούσε να εισαχθεί.

1. Βιβλιογραφικές αναφορές
2. Ο Οδηγός του Scrum, Ken Schwaber - Jeff Sutherland, 2020, [https://scrumguides.org/2020-Scrum-Guide-Greek.pdf](https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Greek.pdf)
3. Agile Μεθοδολογίες, Ελβίρα-Μαρία Αρβανίτου, ΔΙ.ΠΑ.Ε.
4. Μηχανική Απαιτήσεων, Ελβίρα-Μαρία Αρβανίτου, ΔΙ.ΠΑ.Ε.