

ΑΝΑΠΤΥΞΗ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

**ΠΜΣ «Προηγμένα Συστήματα Πληροφορικής – Ανάπτυξη Λογισμικού
και Τεχνητής Νοημοσύνης»**

Απαλλακτική Εργασία Χειμερινού Εξαμήνου 2020-21.

Τίτλος Εργασίας: «Seminar Booking Web Application»

04.04.2021

Μέλη ομάδας:

1. Κασώτης Δημήτρης – Α.Μ: ΜΠΣΠ20017.

Email: mpsp20017@unipi.gr

2. Πσενιτσίνι Ντανιήλ – Α.Μ: ΜΠΣΠ20044.

Email: mpsp20044@unipi.gr

3. Ασημακόπουλος Γεώργιος – Α.Μ: ΜΠΣΠ20004.

Email: mpsp20004@unipi.gr

1. Σύντομη περιγραφή εφαρμογής:

Η παρούσα web εφαρμογή, υλοποιεί ένα σύστημα διαχείρισης και κράτησης σεμιναρίων. Οι χρήστες της εφαρμογής θα μπορούν να περιηγηθούν στις σελίδες της εφαρμογής, να δουν σεμινάρια για διάφορους τομείς και αν το επιθυμούν να κάνουν κράτηση για αυτά. Για την αποθήκευση όλων των δεδομένων η εφαρμογή μας χρησιμοποιεί μία τοπική βάση. Πιο αναλυτικές πληροφορίες για την υλοποίησή της, για τους ρόλους και τις βασικές λειτουργίες της περιέχονται στις παρακάτω παραγράφους.

1.1. Στόχος της εργασίας:

Στόχος της εργασίας, είναι ο σχεδιασμός και η λειτουργία ολοκληρωμένης δυναμικής εφαρμογής που αφορά στην διαχείριση κρατήσεων σεμιναρίων.

1.2. Ανάλυση απαιτήσεων:

Σε ένα σύστημα διαχείρισης κρατήσεων πρέπει αρχικά να γίνει διαχωρισμός των διακριτών ρόλων ανάμεσα στους χρήστες του. Ανάλογα με τα δικαιώματα του κάθε χρήστη, μπορεί να πραγματοποιεί δημιουργία/επεξεργασία σεμιναρίων, να επιλέγει σεμινάριο ως επισκέπτης ή να πραγματοποιεί έλεγχο και πληρωμή εισιτηρίων.

Η βασική λειτουργία της εφαρμογής είναι να δίνει τη δυνατότητα προγραμματισμού ενός σεμιναρίου.

Έπειτα πρέπει να οριστούν οι ενέργειες που μπορεί να κάνει ο κάθε χρήστης της εφαρμογής.

Ο διαχειριστής του συστήματος θα μπορεί να κάνει τις παρακάτω ενέργειες:

- Δημιουργία σεμιναρίου.
- Επεξεργασία σεμιναρίου.
- Διαγραφή σεμιναρίου.
- Διαχείριση ρόλων χρηστών.

Οι επισκέπτες (απλοί χρήστες) θα μπορούν να κάνουν τις παρακάτω ενέργειες:

- Αναζήτηση σεμιναρίων.
- Επιλογή σεμιναρίων.
- Εκτύπωση εισιτηρίων.
- Πληρωμή εισιτηρίων.

Οι ελεγκτές (εισπράκτορας εισιτηρίων) θα μπορούν να κάνουν την παρακάτω ενέργεια:

- Αναζήτηση και επικύρωση εισιτηρίων.

Αφού έγινε καταγραφή που αφορά στις λειτουργίες της εφαρμογής και τις ενέργειες των χρηστών θα προχωρήσουμε σε περαιτέρω ανάλυση των παραπάνω με περιγραφές και διαγράμματα περιπτώσεων χρήσης.

2. Ρόλοι & βασικές λειτουργίες:

Η εφαρμογή μας αποτελείται από τους εξής 3 βασικούς ρόλους:

1. Διαχειριστής εφαρμογής (admin), ο οποίος εκτελεί τις παρακάτω λειτουργίες:

- Μπορεί να εισέρχεται στο σύστημα της εφαρμογής (Login) με δικαιώματα διαχειριστή, αφού γίνει έλεγχος στη βάση.

- Στην αρχική σελίδα (Home) μπορεί να βλέπει όλα τα διαθέσιμα σεμινάρια που υπάρχουν στη βάση δεδομένων. Τα σεμινάρια εμφανίζονται με τη μορφή καρτών.
- Στο τέλος κάθε κάρτας υπάρχουν δύο επιλογές για κάθε σεμινάριο. Ο διαχειριστής είτε μπορεί να ενημερώσει/αλλάξει (Update) το σεμινάριο ή να το διαγράψει (Delete) από τη βάση.
- Με την επιλογή ενημέρωσης (Update) μεταφέρεται σε μία νέα σελίδα, όπου εκεί τροποποιεί τα βασικά στοιχεία του σεμιναρίου που επέλεξε (τίτλο, περιγραφή, ημερομηνία διεξαγωγής του, χωρητικότητα και τιμή).
- Υπάρχει και η δυνατότητα δημιουργίας νέου σεμιναρίου (Create). Από το βασικό μενού, ο διαχειριστής μεταβαίνει σε μία νέα σελίδα όπου συμπληρώνει από την αρχή τα δεδομένα ενός σεμιναρίου και το δημιουργεί, αυτόματα εισάγεται στη βάση.
- Ο διαχειριστής μπορεί να κάνει έξοδο από το σύστημα, οποτεδήποτε κάνοντας κλικ στο κουμπί Logout.

2. Απλός χρήστης (user), ο οποίος εκτελεί τις παρακάτω λειτουργίες:

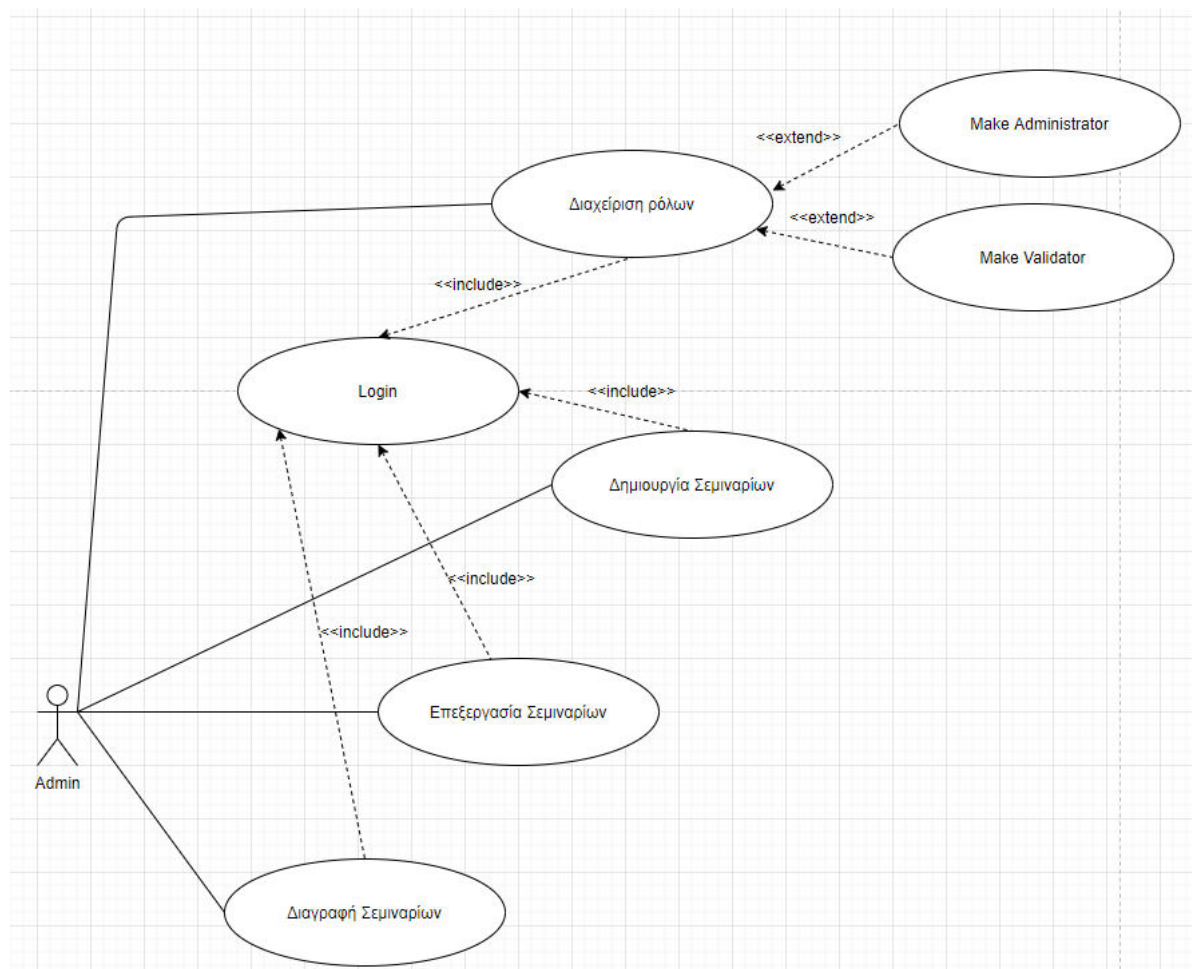
- Μπορεί να εισέλθει στο σύστημα της εφαρμογής (Login) με δικαιώματα απλού χρήστη, αφού γίνει έλεγχος στη βάση. Επίσης, αν δεν έχει λογαριασμό στη σελίδα μπορεί να εγγραφεί σε αυτή (Register). Εναλλακτικά υπάρχει και η δυνατότητα εισόδου ή εγγραφής με Google λογαριασμό.
- Στην αρχική σελίδα (Home) μπορεί να βλέπει όλα τα διαθέσιμα σεμινάρια. Τα σεμινάρια εμφανίζονται με τη μορφή καρτών.
- Ο χρήστης κάνοντας κλικ το κουμπί «Read More» στο τέλος κάθε κάρτας, μεταφέρεται σε νέα σελίδα όπου εκεί μπορεί να μάθει περισσότερες πληροφορίες για το σεμινάριο που τον ενδιαφέρει, όπως τη περιγραφή του, τότε θα διεξαχθεί και αν υπάρχει διαθέσιμος χώρος έτσι ώστε να συμμετάσχει σε αυτό.
- Συμπληρώνοντας τη φόρμα (όνομα, επίθετο, τηλέφωνο και φωτογραφία) που υπάρχει κάτω από τις πληροφορίες του σεμιναρίου και κάνοντας κλικ το κουμπί «Join», κάνει κράτηση για το τρέχον σεμινάριο.
- Μέσω του κεντρικού μενού, μπορεί να δει όλες τις τρέχουσες κρατήσεις του (My Bookings) και να εκτυπώσει το εισιτήριο του (με ή χωρίς QR code) για όποια κράτηση έχει στο όνομα του, αν το επιθυμεί. Υπάρχει και η δυνατότητα ακύρωσης κάποιας κράτησης.
- Ο χρήστης μπορεί να κάνει έξοδο από το σύστημα, οποτεδήποτε κάνοντας κλικ στο κουμπί Logout.

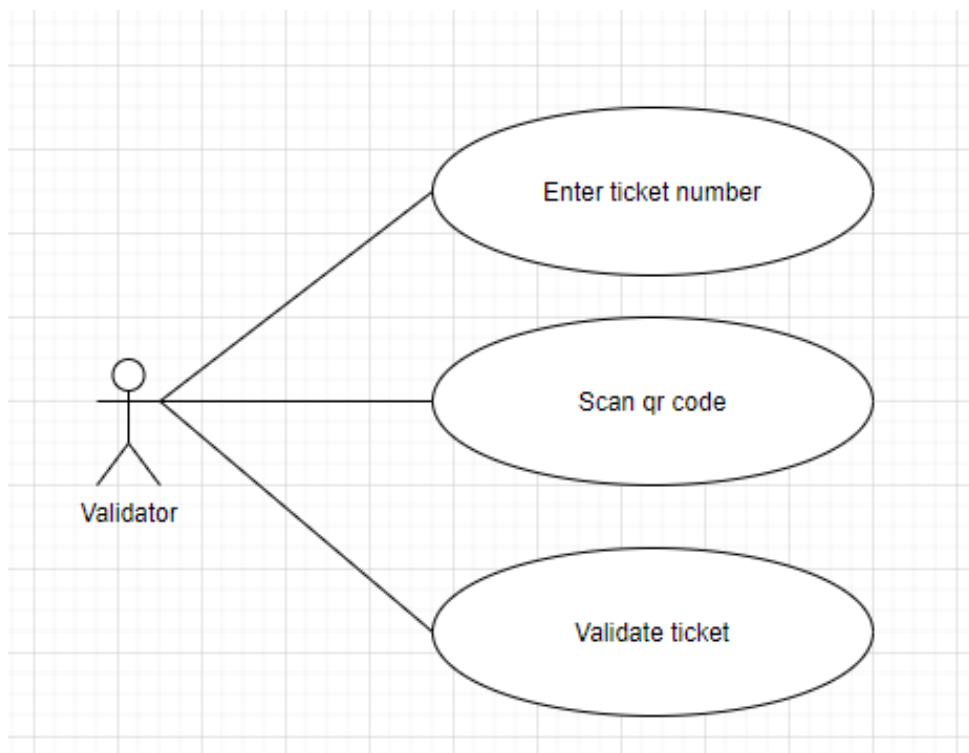
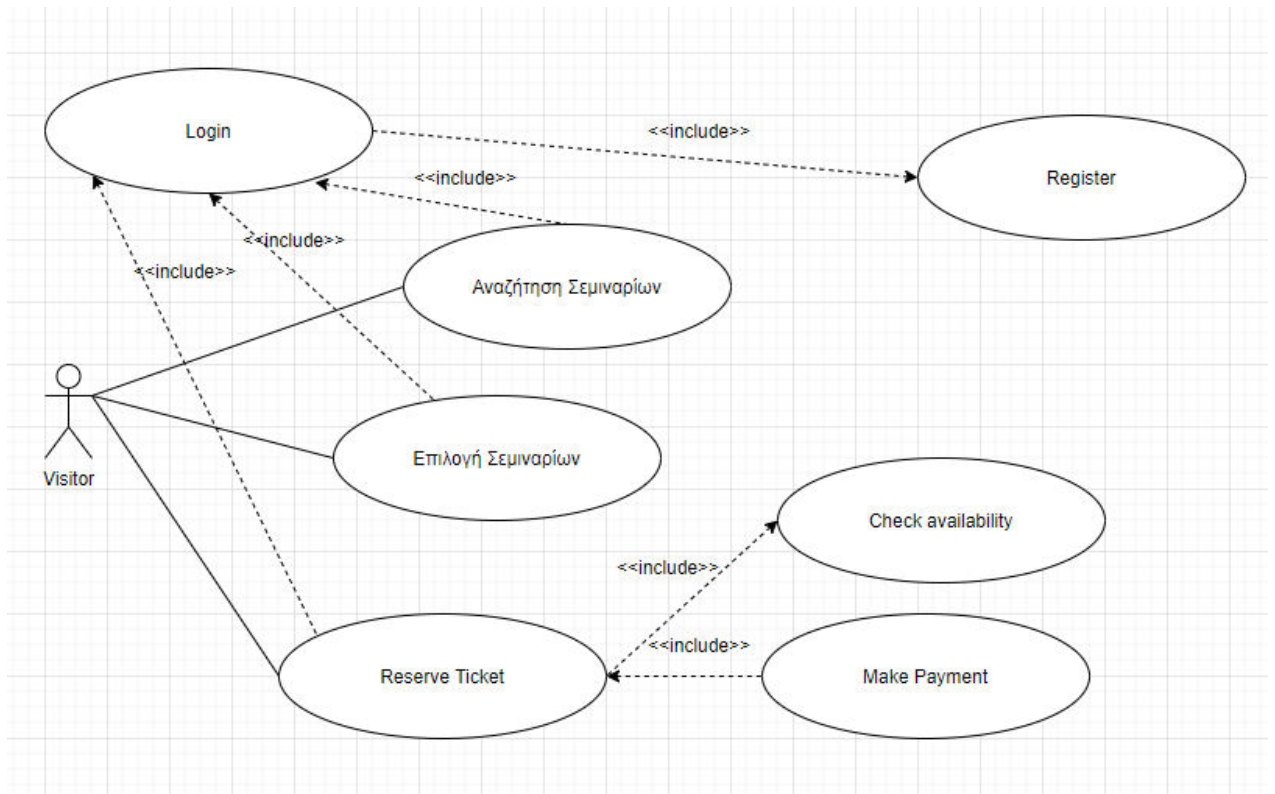
3. Εισπράκτορας εισιτηρίων (ticket collector/validator), ο οποίος εκτελεί τις παρακάτω λειτουργίες:

- Μπορεί να εισέλθει στο σύστημα της εφαρμογής (Login) με δικαιώματα εισπράκτορα εισιτηρίων, αφού γίνει έλεγχος στη βάση για το ρόλο του.
- Στην αρχική σελίδα του, μπορεί αφού σαρώσει το QR code που έχει το εισιτήριο του χρήστη να το εισάγει στο αντίστοιχο πεδίο και αν υπάρχει, να βρει τη κράτηση και να πληρωθεί από τον πελάτη (απλός χρήστης).
- Ο εισπράκτορας μπορεί να κάνει έξοδο από το σύστημα, οποτεδήποτε κάνοντας κλικ στο κουμπί Logout.

3. Διαγράμματα περιπτώσεων χρήσης:

Οι παρακάτω εικόνες παρουσιάζουν τα διαγράμματα περιπτώσεων χρήσης για τους 3 βασικούς ρόλους της εφαρμογής μας, όπως περιγράφηκαν παραπάνω:





4. Τεχνολογίες:

Οι τεχνολογίες που χρησιμοποιήθηκαν για τη δημιουργία της εργασίας είναι οι:

- **Front-End:** HTML, CSS, Bootstrap, EJS, JavaScript, Calendar: (npm evo-calendar).
- **Back-End:** Node.js, Express.js
- **Databases:** MongoDB, Mongoose.

5. Προγράμματα, προεργασία & εκτέλεση:

5.1. Προγράμματα:

1. Κάνουμε λήψη και εγκατάσταση από την επίσημη σελίδα του, το [Node.js](#)
2. Κάνουμε λήψη και εγκατάσταση από την επίσημη σελίδα της, τη [MongoDB](#).
3. Για την εισαγωγή και ανάκτηση δεδομένων προς και από τη τοπική βάση, έγινε χρήση του προγράμματος [Robo 3T](#).
4. Για την εκτέλεση εντολών, δημιουργία φακέλων & αρχείων αλλά και για την εισαγωγή πακέτων στο κώδικά μας, έγινε χρήση του [Hyper Terminal](#).

5.2. Προεργασία:

Τα δεδομένα που βλέπουμε στις σελίδες της εφαρμογής μας (στοιχεία σεμιναρίων, χρήστες, κρατήσεις) πρέπει να φορτωθούν αρχικά μέσα στη τοπική βάση που χρησιμοποιούμε. Για να γίνει αυτό:

1. Ανοίγουμε το φάκελο της εργασίας μας και μεταβαίνουμε στο φάκελο `databases`.
2. Εκεί υπάρχουν 3 αρχεία JSON (ένα για κάθε πίνακα: `bookings`, `events` & `users`). Τα οποία έχουν γίνει εξαγωγή από τη τοπική βάση μας στο Robo 3T.
3. Ανοίγουμε τα δύο αυτά αρχεία με έναν editor.
4. Ανοίγουμε το πρόγραμμα Robo 3T, στο πρώτο αναδυόμενο παράθυρο κάνουμε κλικ στο κουμπί **Connect** και δημιουργούμε μία νέα βάση με όνομα: **projectDB** (New Connection > Create Database), ανοίγουμε τη βάση αυτή και κάνοντας δεξί κλικ στο φάκελο **Collections** δημιουργούμε 3 νέους πίνακες με ονόματα: `bookings`, `events` & `users` αντίστοιχα. Κάνοντας δεξί κλικ στους πίνακες, μπορούμε να εισάγουμε δεδομένα σε αυτούς (**Insert Document**), στο παράθυρο που θα εμφανιστεί παίρνουμε όλα τα δεδομένα του κάθε πίνακα από τα αντίστοιχα αρχεία JSON με αντιγραφή και επικόλληση εδώ. Τέλος κάνουμε κλικ στο **Save**.

5.3. Εκτέλεση:

Για την εκτέλεση της εφαρμογής, κάνουμε τα παρακάτω βήματα:

1. Ανοίγουμε το τερματικό (Hyper) στο φάκελο της εργασίας και εκτελούμε τη παρακάτω εντολή για αρχικοποίηση του NPM:

```
$ npm init -y
```

2. Στο αρχείο package.json υπάρχουν όλα τα dependencies (ξεκινά γραμμή 11) της εφαρμογής μας, για την εγκατάσταση ενός πακέτου δίνουμε την εντολή:

```
$ npm install <package_name> (π.χ: $ npm install express)
```

3. Στο ίδιο τερματικό ανοίγουμε μία νέα δεύτερη καρτέλα (με CTRL+SHIFT+T) και δίνουμε την εντολή (παραμένει ανοικτό για επικοινωνία με τη τοπική βάση):

```
$ mongod
```

4. Για την εκτέλεση της εφαρμογής μας στην αρχική καρτέλα του τερματικού, δίνουμε την εντολή (όπου app.js είναι το αρχείο εισόδου σε γλώσσα JavaScript):

```
$ nodemon app.js
```

5. Τέλος, ανοίγουμε έναν browser και στη διεύθυνση URL δίνουμε (η εφαρμογή μας «ακούει» στη πόρτα 3000):

```
> http://localhost:3000
```

Επεξήγηση!: Τα βήματα 1 και 2 γίνονται μόνο μία φορά όταν δεν έχει εκτελεστεί η εφαρμογή μας στο συγκεκριμένο υπολογιστή καμία φορά. Τα υπόλοιπα βήματα 3,4 και 5 εκτελούνται κάθε φορά που θέλουμε να εκτελέσουμε την εφαρμογή μας, ανεξαρτήτως (υπάρχουν και στο παρακάτω βίντεο).

6. Βίντεο εκτέλεσης εφαρμογής:

Ο παρακάτω σύνδεσμος περιέχει το βίντεο εκτέλεσης της εφαρμογής μας:

<https://drive.google.com/file/d/11Jr18AAa-ZYgC3hwSRashqeMmZJpYEng/view?usp=sharing>

Η καταγραφή του βίντεο έγινε με το πρόγραμμα [OBS Studio](#).