

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

### ΕΦΑΡΜΟΓΗ ΥΠΟΣΤΗΡΙΞΗΣ ΑΡΧΑΙΟΛΟΓΙΚΟΥ ΜΟΥΣΕΙΟΥ

Νικόλ, NN, Νικολάου

Φοιτήτρια, Πανεπιστήμιο Πατρών, 1078521, [up1078521@upnet.gr](mailto:up1078521@upnet.gr)

Σταμάτιος, ΣΝ, Νικολουδάκης

Φοιτητής, Πανεπιστήμιο Πατρών, 1080573, [up1080573@upnet.gr](mailto:up1080573@upnet.gr)

**Περίληψη:** Στα πλαίσια του μαθήματος του διαδικτυακού προγραμματισμού μας ανατέθηκε το project με θέμα την δημιουργία μιας εφαρμογής υποστήριξης αρχαιολογικού μουσείου. Με την βοήθεια της θεωρίας του μαθήματος του Προγραμματισμού Διαδικτύου και τις επιπλέον γνώσεις που είχαμε από το μάθημα των βάσεων δεδομένων, καταφέραμε να δημιουργήσουμε μια ιστοσελίδα. Η ιστοσελίδα αυτή είναι λειτουργική, καλαίσθητη, φιλική προς τον χρήστη αλλά και πρωτότυπη.

**CCS CONCEPTS** • Information systems • Data management systems

**Εμπρόσθετες Λέξεις και Φράσεις κλειδιά:** html, CSS, Bootstrap, javascript, node.js, express.js, handlebars, MYSQL.

## 1. ΕΙΣΑΓΩΓΗ

Το θέμα που ανατέθηκε στην ομάδα μας είναι η δημιουργία εφαρμογής υποστήριξης αρχαιολογικού μουσείου.

Αποφασίσαμε να προσεγγίσουμε το θέμα από τη σκοπιά του χρήστη, δίνοντας έμφαση στις ανάγκες και τις προτιμήσεις του, αντί να επικεντρωθούμε στους διαχειριστές του μουσείου.

Αναπτύξαμε μια σελίδα που προσφέρει στον χρήστη μια πλούσια εμπειρία, παρέχοντας λεπτομέρειες για τα εκθέματα του μουσείου και διάφορα άλλα ενδιαφέροντα γεγονότα που σχετίζονται με αυτό. Αφού ο χρήστης εγγραφεί στη σελίδα και έχει συνδεθεί με τον λογαριασμό του, ο χρήστης έχει τη δυνατότητα να αποκτήσει πρόσβαση σε επιπλέον λειτουργίες.

Συγκεκριμένα, μετά την εγγραφή του, ο χρήστης, μπορεί να αγοράσει εισιτήρια για την επίσκεψη του στο μουσείο και να διαχειριστεί το προφίλ του με ευκολία.

Ο στόχος μας ήταν να δημιουργήσουμε μια λειτουργική και ευχάριστη εμπειρία για τον χρήστη, προσέχοντας την αισθητική και την ευκολία χρήσης της σελίδας.

Με αυτόν τον τρόπο, επιδιώξαμε να δημιουργήσουμε μια εφαρμογή που θα αναδείξει την ομορφιά και την πολιτιστική αξία του αρχαιολογικού μουσείου, προσφέροντας ταυτόχρονα μια ευχάριστη και ενδιαφέρουσα εμπειρία στον χρήστη.

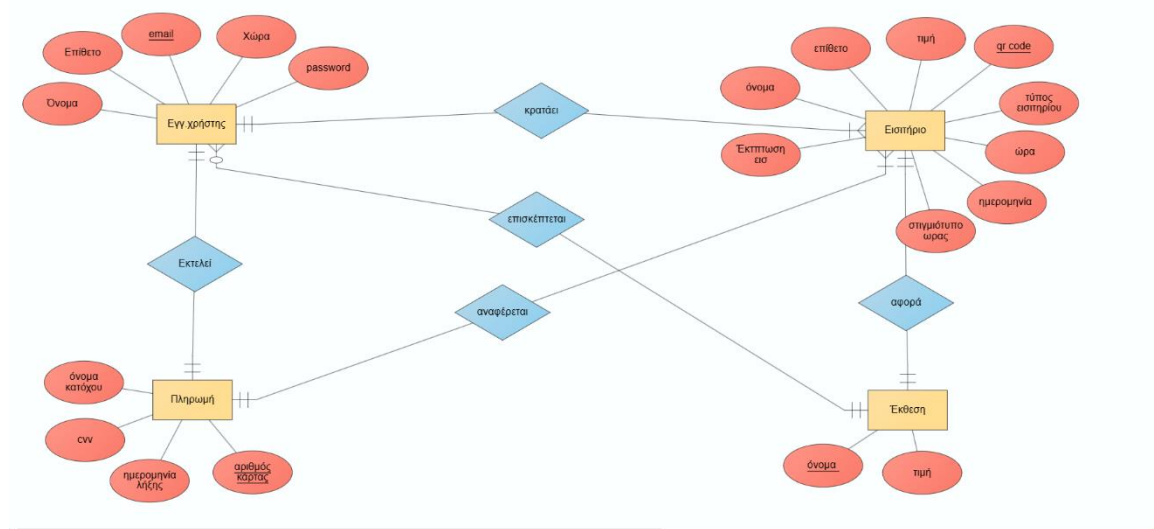
## 2. ΜΕΘΟΔΟΛΟΓΙΑ:

Αρχικά, ξεκινήσαμε μελετώντας ιστοσελίδες διάφορων μουσείων για να συλλέξουμε ιδέες. Σημειώσαμε ορισμένες αισθητικές που βρήκαμε ενδιαφέρουσες. Μαζέψαμε επίσης, φωτογραφικό υλικό. Προσπαθήσαμε οι φωτογραφίες μας να μην είναι copyrighted γι' αυτό ως επί το πλείστον χρησιμοποιήσαμε φωτογραφίες της οποίες είχαμε τραβήξει οι ίδιοι όταν είχαμε επισκεφθεί το μουσείο για το οποίο επiléξαμε να φτιάξουμε αυτή την ιστοσελίδα.

### 2.1 Εννοιολογικό Διάγραμμα:

Έπειτα, προχωρήσαμε στον σχεδιασμό του διαγράμματος οντοτήτων συσχετίσεων (erd) πάνω στο οποίο θα βασιστεί η βάση δεδομένων του μουσείου.

Το διάγραμμα οντοτήτων συσχετίσεων παρουσιάζεται στην πιο κάτω εικόνα:



## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

### 2.2 Σελίδες:

Αφού τελειώσαμε τον σχεδιασμό του πιο πάνω διαγράμματος ξεκινήσαμε να αναγράφουμε τις βασικές σελίδες που θα περιέχει ιστοσελίδα μας και καταλήξαμε στις εξής:

1. Home Page
2. What's on
3. Explore
4. Sign up
5. Log in
6. Tickets and Prices
7. Ticket Form
8. Card-info
9. Payment
10. Profile
11. Change Password

Πιο κάτω στην αναφορά θα εξηγήσουμε τον σκοπό της κάθε μιας ξεχωριστά.

### 2.3 Front-end:

Επόμενο βήμα για την υλοποίηση αυτής της εργασίας ήταν να ξεκινήσουμε να γράφουμε τον κώδικα στο επίπεδο του front-end.

Με την βοήθεια των εργαλείων HTML, CSS και Bootstrap ξεκινήσαμε την εγγραφή του κώδικα για τις πιο πάνω σελίδας.

Φυσικά, από το Front-end κομμάτι δεν μπορεί να λείπει η Javascript. Επομένως, αφού ολοκληρώσαμε την μορφή των σελίδων μας προχωρήσαμε, στο επόμενο βήμα το οποίο είναι να εντάξουμε την Javascript για να κάνουμε τις σελίδες πιο διαδραστικές.

Με την ένταξη της Javascript, τελειώνει το κομμάτι του Front-end της εργασίας.

### 2.4 Back-end και συνδεση με βαση:

Για το Back-end κομμάτι της εργασίας μας, αρχικά, χρησιμοποιήσαμε την node.js και βιβλιοθήκες της όπως την express.js.

Για αρχή, μαζέψαμε τους κώδικες που είχαμε γράψει στο Front-end μέσα στο φάκελο Public (με static αρχεία) όπου μέσα στους υποφάκελους images/css/js καταχωρήσαμε όλες της εικόνες τα styles και τα scripts που είχαμε χρησιμοποιήσει για την δημιουργία του front-end.

Προχωρώντας στο Back-end κομμάτι, ξεκινήσαμε την υλοποίηση εγκαθιστώντας τα απαραίτητα node modules. Και συνεχίσαμε, ορίζοντας το port στο οποίο θα τρέχει η σελίδα μας.

```
app.listen(process.env.PORT || 3000, () => {  
  console.log('Server is running on port',process.env.PORT || 3000);  
});
```

Έπειτα, φτιάξαμε routes για κάθε σελίδα. Όπως για παράδειγμα:

```
router.get("/login", (req, res) => { ... });
```

αυτό το path θα σε πηγαίνει στη φόρμα του login.

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Τα routes, τα φτιάξαμε σε διαφορετικό φάκελο (routes.mjs) και τα κάναμε export στο index.mjs.

```
( export {router}; )
```

Κατόπιν, φτιάξαμε τα handlebars.

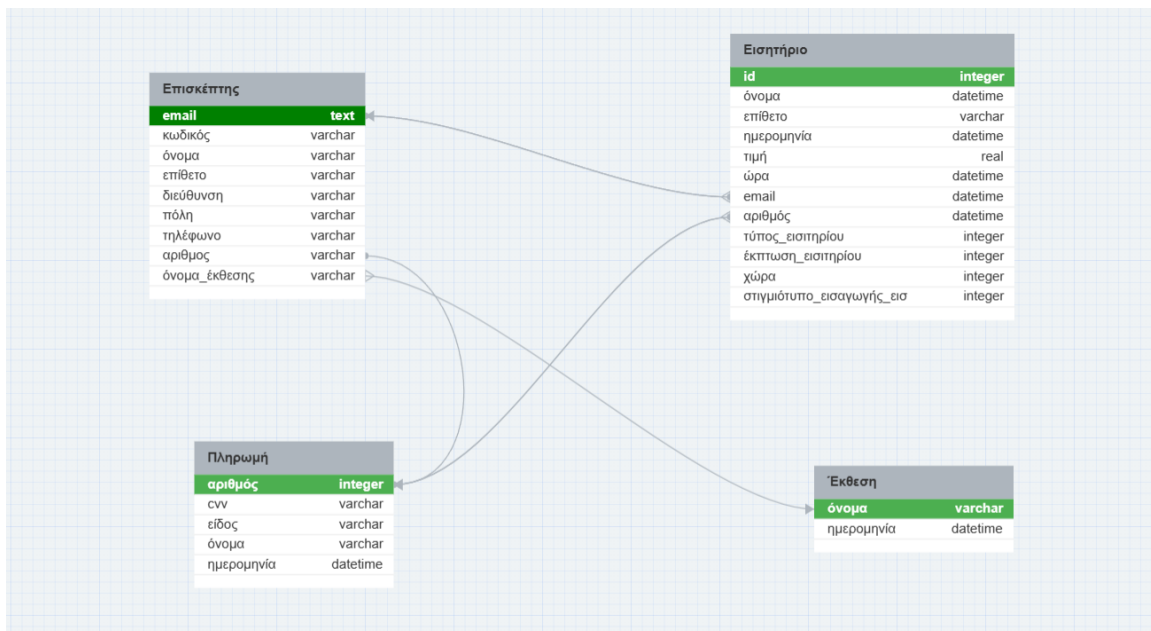
```
import { engine } from "express-handlebars";
```

Μαζέψαμε τους κώδικες της HTML σε ένα φάκελο views τους χωρίσαμε σε layouts και partials. Φτιάξαμε 5 layouts (όπου ορισμένα ταίριαζαν σε πάνω από μια σελίδες). Διορθώσαμε και τις εικόνες με το κατάλληλο path για την κάθε μία.

Έπειτα, διαμορφώσαμε τα routes με την βοήθεια των handlebars και καλώντας μέσω των routes διαφορετικά headers, footers και body για κάθε layout. Μαζί με τα κατάλληλα links (ccs, js) για κάθε σελίδα. Επομένως, τα routes σε αυτή τη φάση διαμορφώθηκαν ως εξής:

```
1. router.get("/whatson" , (req, res) =>{
2.   res.render("whatson",{
3.     title:"What's on",
4.     header:"partials/header",
5.     style:"public/css/stylewhatson.css",
6.     link1:"/",
7.     link2:"/tickets",
8.     link3:"/explore",
9.     name1:"Home",
10.    name2:"Book a Ticket",
11.    name3:"Explore",
12.    footer: "partials/footer",
13.    script:"public/js/scriptfooter.js"
14.  });
15. });
```

Τελειώνοντας με τα routes, προς το παρόν, προχωρήσαμε στη βάση δεδομένων. Η βάση αυτή στηρίζεται στο πιο πάνω erd και έπειτα στο πιο κάτω διάγραμμα οντοτήτων σχέσεων.



## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Για την υλοποίηση της βάσης δεδομένων της ιστοσελίδας μας αποφασίσαμε να χρησιμοποιήσουμε την MYSQL.

Αρχικά, φτιάξαμε την εξής βάση δεδομένων:

```
1. CREATE TABLE IF NOT EXISTS Visitor (
2.     email varchar(255) NOT NULL PRIMARY KEY UNIQUE,
3.     firstname varchar(255),
4.     lastname varchar(255),
5.     visitor_password varchar(255),
6.     date_of_birth date,
7.     exhibit_name varchar(255),
8.     card_no varchar(255),
9.     FOREIGN KEY (exhibit_name) REFERENCES exhibition(exhibit_name),
10.    FOREIGN KEY (card_no) REFERENCES payment(card_no)
11. );
12.
13.
14.
15. CREATE TABLE IF NOT EXISTS Payment (
16.     card_no varchar(255) NOT NULL PRIMARY KEY UNIQUE,
17.     cardholder_name varchar(255),
18.     cvv varchar(255),
19.     end_date date
20. );
21.
22. CREATE TABLE IF NOT EXISTS Ticket (
23.     ticket_id integer UNIQUE NOT NULL PRIMARY KEY auto_increment,
24.     book_firstname varchar(255),
25.     book_lastname varchar(255),
26.     ticket_type varchar(255),
27.     book_time varchar (255),
28.     book_date date (255),
29.     country varchar (255),
30.     ticket_discount varchar (255),
31.     price float,
32.     book_email varchar (255),
33.     card_no_book varchar (255),
34.     ticket_insertion_time TIMESTAMP NOT NULL,
35.     FOREIGN KEY (email_user) REFERENCES visitor(email),
36.     FOREIGN KEY (card_no_book) REFERENCES payment(card_no));
37.
38. CREATE TABLE IF NOT EXISTS Exhibition (
39.     name varchar(255) NOT NULL PRIMARY KEY,
40.     price float NOT NULL );
```

Αφού φτιάξαμε την βάση μας, την ενώσαμε με τον υπόλοιπο κώδικα για να μπορούμε να αποθηκεύουμε και επεξεργαζόμαστε δεδομένα. Αφού κάναμε import την mysql2, λοιπόν, ενώσαμε τη βάση ως εξής:

```
1. import mysql from "mysql2";
2. const connection = mysql.createConnection({
3.   host: 'localhost',
4.   user: 'root',
5.   password: 'password',
6.   database: 'museum_database'});
7. connection.connect(function(err) {
8.   if (err) {
9.     console.error('Error connecting to MySQL server: ' + err.message);
10.    return;
11.  }
12.  console.log('Connected to MySQL server.');
```

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Έπειτα, γεμίσαμε την βάση μας δεδομένα μέσω script που έχουμε φτιάξει. Το εξηγούμε πιο αναλυτικά πιο κάτω στην αναφορά.

Μετά, προχωρήσαμε στα sessions.

Ο πιο κάτω κώδικας χρησιμοποιείται για την δημιουργία και διαχείριση συνεδριών σε έναν διακομιστή εφαρμογών. Οι συνεδρίες επιτρέπουν στον διακομιστή να αποθηκεύει πληροφορίες για κάθε χρήστη ξεχωριστά κατά τη διάρκεια της αλληλεπίδρασής του με την εφαρμογή. Οι ρυθμίσεις που δίνονται στον κώδικα καθορίζουν τις λεπτομέρειες των συνεδριών, όπως το κλειδί και το μυστικό για την ασφάλεια, η αποθήκευση των συνεδριών και η ρύθμιση των cookies που χρησιμοποιούνται για την αναγνώριση των συνεδριών.

```
1. app.use(session({
2.   key:process.env.KEY,
3.   secret:process.env.SECRET,
4.   resave: false,
5.   saveUninitialized: false,
6.   store: sessionStore,
7.   cookie: {
8.     originalMaxAge: 86400000,
9.     expires: new Date(Date.now() + 86400000),
10.    httpOnly: true }
11. }));
```

Δηλαδή, με το login ενός χρήστη δημιουργείται ένα νέο session και η μεταβλητή isAuth παίρνει την τιμή true. Έτσι, με το κατάλληλο middleware ο χρήστης έχει πρόσβαση στις προστατευμένες περιοχές της σελίδας.

```
1.const isAuth=(req,res,next)=>
2.{ if(req.session.isAuth){
3.next() }
4.else{
5.   res.redirect("/login");
6.} }
```

Χρησιμοποιήσαμε το session storage για να αποθηκεύσουμε εκεί πέρα πληροφορίες για το χρήστη που είναι αποθηκευμένες και στη βάση. Τις πληροφορίες αυτές τις χρησιμοποιούμε για να τις εμφανίζουμε σε διάφορες φόρμες την ώρα που ο χρήστης είναι συνδεδεμένος στην σελίδα. Για παράδειγμα στο profile, όπου ο χρήστης θα μπαίνει για να επεξεργάζεται τα δεδομένα του, είτε, στο nav bar που εμφανίζεται το όνομα του χρήστη του οποίου είναι συνδεδεμένος.

Τα routes, λοιπόν, έχουν διαμορφωθεί όπως πιο κάτω:

```
1.router.get("/profile",isAuth,(req,res) =>{
2.const error = req.query.error;
3.res.render("profile",{
4.title:"profile",
5.layout: lay2,
6.header:"partials/profileheader",
7.footer:"partials/footerempty",
8.style:"public/css/profile.css",
9.script:"public/js/scriptprofile.js",
10.error:error,
11.errorMessage: "Email already exists,please try a new one!",
12.success:success,
13.successMessage: "Data updated successfully!",
14.email: req.session.email,
15.username: req.session.firstname,
16.userlastname: req.session.lastname,
17.userLoggedIn:req.session.isAuth
18. });
```

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

```
19.});
```

Έπειτα ενώσαμε τις συνεδρίες με τη βάση μας.

```
1.const MySQLStore = MySQLStoreFactory(session);
2.const sessionStore = new MySQLStore({
3.  host: 'localhost',
4.  port: 3306,
5.  user: 'root',
6.  password: 'password',
7.  database: 'museum_database',
8.  createDatabaseTable: true,
9.schema: {
10.  tableName: 'sessions',
11.  columnNames: {
12.    session_id: 'session_id',
13.    expires: 'expires',
14.    data: 'data'}}
15.});
```

Ο παραπάνω κώδικας, δημιουργεί μια αποθήκη συνεδριών στη βάση δεδομένων MySQL, ρυθμίζοντας τα στοιχεία σύνδεσης με τη βάση δεδομένων και ορίζοντας τον πίνακα και τα ονόματα των στηλών που θα χρησιμοποιηθούν για την αποθήκευση των συνεδριών.

Μετά, προχωρήσαμε στα queries. Τα queries γράφτηκαν ως συναρτήσεις στο φάκελο model εκεί που ενώσαμε και χειριζόμαστε τη βάση και έγιναν μετά exported και imported στα routes.mjs.

Αρχικά, ασχοληθήκαμε στο πως να βάζουμε μέσω της κάθε φόρμας δεδομένα στη βάση. Για παράδειγμα, η φόρμα register/sign in βάζει δεδομένα στον πίνακα του Visitor και η ticket form βάζει δεδομένα στον πίνακα Ticket. Για να επιτύχουμε αυτό το σκοπό χρησιμοποιήσαμε την εντολή INSERT της sql. Για παράδειγμα.

```
"INSERT INTO visitor(email,firstname,lastname,visitor_password,date_of_birth)
VALUES (?, ?, ?, ?, ?)";
```

Για να μπορεί ένας χρήστης να επεξεργάζεται τα δεδομένα του χρησιμοποιήσαμε την εντολή UPDATE. Για παράδειγμα για να μπορεί ένας χρήστης να αλλάζει τον κωδικό του έχουμε αυτή τη συνάρτηση.

Στο model.mjs:

```
1.async function UpdatePassword(currentPassword, password, emailToChange, req,res)
2.  {
3.    console.log(currentPassword);
4.    const userPassword = await retrievePassword(emailToChange);
5.    const passwordsMatch = await
6.    bcrypt.compare(currentPassword,userPassword[0][0].visitor_password);
7.    if (!passwordsMatch) {
8.      return res.redirect("/passwordsettings?error=true");}
9.
10.    if (currentPassword === password) {
11.      return res.redirect("/passwordsettings?same_password=true");}
```

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

```
12.      try {
13.          const newHashedPassword = await bcrypt.hash(password, 10);
14.          const updatePass = 'UPDATE visitor SET visitor_password=? WHERE email = ?';
15.          const result = await connection.promise().query(updatePass,
16.                  [newHashedPassword, emailToChange])
17.          return res.redirect("/passwordsettings?success=true");
18.      } catch (error) {
19.          throw error;} }
```

Στο routes.mjs:

```
1. router.post("/passwordsettings", async (req, res) =>{
2.     const passwordData=req.body;
3.     const currentPassword=passwordData.currentpassword;
4.     const password=passwordData.password;
5.     const emailToChange=req.session.email;
6.     awaitdb_model.UpdatePassword(currentPassword,password,emailToChange, req, res);
7. });
```

Χρησιμοποιήσαμε ακόμα την εντολή DELETE για να διαγράψουμε δεδομένα από τη βάση μας, ακόμα, φτιάξαμε αρκετά ακόμα query για επεξεργασία δεδομένων στη βάση μας.

Παράδειγμα, φέρει, η πιο κάτω συνάρτηση που αφορά τον πίνακα των εισιτηρίων:

```
1.async function
2.insertUserDataInTickets (date,time,country,ticketType,email,price, req, res) {

3.      try {
4.          const check ="SELECT COUNT(*) AS total FROM ticket WHERE book_time IS NULL";
5.          const getcount=await connection.promise().query(check);
6.          console.log(getcount[0][0].total);
7.          if(getcount[0][0].total===1){
8.              if(ticketType==="Adult Ticket"){
9.                  const updateTicket="UPDATE ticket SET
10. book_date=?,book_time=?,country=?,ticket_discount=?,price=? WHERE book_email = AND
11. book_time IS NULL";
12. const upd = await connection.promise().query(
13.     updateTicket, [date,time,country,ticketType,price,email]); }

14. connection.promise().query(updateTicket, [date,time,country,ticketType,price,email]
15. ); }

16.     else if(ticketType==="Student Ticket"){
```



## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

```
17.         const updateTicket="UPDATE ticket SET
18.     book_date=?,book_time=?,country=?,ticket_discount=?,price=?  WHERE book_email = ?
19.     AND book_time IS NULL";

20.     const upd=await
22.connection.promise().query(updateTicket,[date,time,country,ticketType,(price-
23. 0.5*price),email]); }

24.         else if(ticketType==="Under 15"){
25.             const updateTicket="UPDATE ticket SET
26. book_date=?,book_time=?,country=?,ticket_discount=?,price=?  WHERE book_email = ?
27. AND
27.book_time IS NULL";

28.     const upd=await
29.connection.promise().query(updateTicket,[date,time,country,ticketType,(price-
30.0.4*price),email]); } }

31. else{
32.         if(ticketType==="Adult Ticket"){
33.             const getBookInfo = "SELECT firstname, lastname, ticket_type, price,
34.book_email FROM ticket WHERE book_email=?";

35.             const bookData = await connection.promise().query(getBookInfo, [email]);

36.             const insertNewTicketSql="INSERT INTO ticket(ticket_id, firstname,
37.lastname, ticket_type,book_time,book_date,country,ticket_discount,price, book_email)
38.VALUES(?,?,?, ?, ?, ?, ?, ?, ?) "

39.             const insertNewTicket = await
40.connection.promise().query(insertNewTicketSql, [bookData[0][0].firstname,
41. bookData[0][0].lastname, req.session.ticket,time,date,country,ticketType,pr
42.ice,email]);}

43.         else if(ticketType==="Student Ticket"){
44.             const getBookInfo = "SELECT firstname, lastname, ticket_type, price,
45.book_email FROM ticket WHERE book_email=?";

46.             const bookData = await connection.promise().query(getBookInfo, [email]);

47.             const insertNewTicketSql="INSERT INTO ticket(ticket_id, firstname,
48.lastname, ticket_type,book_time,book_date,country,ticket_discount,price, book_email)
49.VALUES(?,?, ?, ?, ?, ?, ?, ?, ?) "

50.const insertNewTicket = await connection.promise().query(insertNewTicketSql,
51.[bookData[0][0].firstname,
52. bookData[0][0].lastname, req.session.ticket,time,date,country,ticketType,
53.(price-0.5*price),email]); }

54.         else if(ticketType==="Under 15"){
55.             const getBookInfo = "SELECT firstname, lastname, ticket_type, price,
56.book_email FROM ticket WHERE book_email=?";

57.             const bookData = await connection.promise().query(getBookInfo, [email]);

58.             const insertNewTicketSql="INSERT INTO ticket(ticket_id, firstname,
59.lastname, ticket_type,book_time,book_date,country,ticket_discount,price, book_email)
60.VALUES(?,?, ?, ?, ?, ?, ?, ?, ?) "
```

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

```
61.          const insertNewTicket = await
62.connection.promise().query(insertNewTicketSql, [bookData[0][0].firstname,
63.          bookData[0][0].lastname, req.session.ticket, time, date, country, ticketType,
64. (price-0.4*price), email]);
65.}          }          }
66. catch (error) {
67.          throw error; } }
```

Αναλυτικότερα, για την συνάρτηση insertUserDataInTickets:

Η συνάρτηση αυτή, αρχικά, επαληθεύει τον αριθμό των διαθέσιμων εισιτηρίων που δεν έχουν κρατηθεί ακόμα. Στη συνέχεια, ανάλογα με τον τύπο του εισιτηρίου, γίνεται είτε ανανέωση (UPDATE) ενός υπάρχοντος καταχωρημένου εισιτηρίου εάν υπάρχει ήδη μια κράτηση για τον χρήστη, είτε δημιουργία (INSERT) ενός νέου εισιτηρίου.

### 2.5 Debugging:

Αφού ολοκληρώσαμε τις βασικές λειτουργίες του προγράμματος, προχωρήσαμε στον αντιμετώπισμό σφαλμάτων (debugging). Αναθεωρήσαμε τον κώδικα και διορθώσαμε τυχόν σφάλματα, εξασφαλίζοντας την ομαλή λειτουργία του προγράμματος.

Στη συνέχεια, ασχοληθήκαμε με λεπτομέρειες στο front-end, προσθέτοντας τα απαραίτητα στοιχεία για να βελτιώσουμε την εμφάνιση και την λειτουργικότητα της ιστοσελίδας.

Εκτελέσαμε πολλαπλές δοκιμές της ιστοσελίδας για να εντοπίσουμε πιθανά σφάλματα ή παραλείψεις που μπορεί να είχαμε κατά τη μετάβαση από HTML σε handlebars και άλλα.

Αφού διορθώθηκαν αυτά τα σφάλματα και είχαμε πλέον μια λειτουργική ιστοσελίδα, προσθέσαμε επιπλέον σχόλια στον κώδικά μας, προκειμένου να γίνει πιο κατανοητός και αναγνώσιμος από τρίτους.

### 2.6 Flyio:

Τέλος, αφού η ιστοσελίδα ήταν έτοιμη την ανεβάσαμε στο Flyio..

## 3.ΑΞΙΟΛΟΓΗΣΗ

Για την αξιολόγηση της εργασίας μας χρησιμοποιήσαμε κάποια βασικά κριτήρια και βάσει αυτών κρίναμε το κατά πόσο η εργασία μας τα πληροί. Τα κριτήρια, ιεραρχημένα, από το πιο σημαντικό στο λιγότερο σημαντικό είναι :

### 3.1.Υλοποίηση του τι μάθαμε στο μάθημα. Εάν ακολουθήσαμε όλα τα απαραίτητα βήματα για την υλοποίηση της εργασίας.

Έχουμε ακολουθήσει όλα τα απαραίτητα βήματα για την υλοποίηση αυτής της ιστοσελίδας τόσο στο front-end όσο και στο back-end. Επίσης, θεωρούμε ότι η εργασία μας αντικατοπτρίζει τις γνώσεις που λάβαμε από το μάθημα του προγραμματισμού διαδικτύου αλλά και των βάσεων δεδομένων.

### 3.2.Λειτουργικότητα, Εάν ο κώδικας τρέχει, και εάν όλα λειτουργούν όπως πρέπει.

Θεωρούμε, πως το πρόγραμμα μας είναι λειτουργικό μιας και ο κώδικας τρέχει. Επιπλέον, δεν διαπιστώθηκε κάποιο σφάλμα κατά την εκτέλεση του προγράμματος.

## PROJECT Προγραμματισμός Διαδικτύου ECE\_GK802

### 3.3.Ασφάλεια: Αξιολόγηση των μέτρων ασφαλείας που έχουν ληφθεί για την προστασία της ιστοσελίδας και των δεδομένων των χρηστών.

Μέσω των συνεδριών θεωρούμε πως επιτεύχθηκε αυτός ο στόχος.

### 3.4.Φιλικότητα προς το χρήστη, το πόσο εύχρηστη είναι η εφαρμογή.

Τέλος, αφού ζητήσαμε από συγγενείς και φίλους να τρέξουν το πρόγραμμα μας διαπιστώθηκε ότι είναι, όντως, φιλικό προς τον χρήστη αφού δεν αντιμετώπισαν κάποια δυσκολία στην εμπειρία τους. Επιπλέον, λάβαμε θετικά σχόλια για το design και την αισθητική της ιστοσελίδας.

### 3.5.Συμβατότητα. Αξιολόγηση της συμβατότητας της ιστοσελίδας με διάφορους περιηγητές καθώς και με διάφορες συσκευές.

Η συμβατότητα της ιστοσελίδας μας είναι συμβατή με διάφορους περιηγητές καθώς και συσκευές.

## 4. ΣΥΛΛΟΓΗ ΔΕΔΟΜΕΝΩΝ:

Για την συλλογή δεδομένων φτιάξαμε script το οποίο δημιουργεί ψεύτικα δεδομένα.

Τα fake data τα δημιουργούμε με διάφορους τρόπους ένας εκ των οποίων είναι η βιβλιοθήκη Faker. Η βιβλιοθήκη αυτή είναι μια δημοφιλής βιβλιοθήκη η οποία δημιουργεί πλαστά (αλλά εύλογα) δεδομένα.

Αρχικά πρέπει να την εγκαταστήσουμε με την εντολή:

```
>>npm install faker
```

Έπειτα, να την κάνουμε import

```
import faker from "faker";
```

Τα δεδομένα που παράγουμε με την βοήθεια της faker είναι:

Το Όνομα του χρήστη.

Το επώνυμό του χρήστη.

Τον κωδικό του χρήστη.

Η ημερομηνία γεννήσεως του χρήστη.

Η χώρα του χρήστη.

Φτιάξαμε ακόμα τις εξής συναρτήσεις:

Για το email, φτιάξαμε μια συνάρτηση από όπου παίρνουμε το ονοματεπώνυμο του χρήστη και προσθέτουμε ένα αριθμό και την κατάληξη '@email.com'.

Έπειτα, βάσει των regex που είχαμε δημιουργήσει φτιάξαμε συνάρτηση η οποία φτιάχνει fake αριθμούς κάρτας.

Για το cvv παίρνουμε ένα random τριψήφιο αριθμό.

Για το id του εισιτηρίου κάναμε παρόμοια συνάρτηση η οποία παίρνει random αριθμούς.

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Για το είδος του εισιτηρίου συνάρτηση που επιλέγει ένα είδος εισιτηρίου τυχαία από λίστα δεδομένων, το ίδιο κάνουμε και για τα time zones και για την έκπτωση.

Κάτι παρόμοιο κάνουμε και για τις Εκθέσεις και τιμές.

Ένας ακόμα τρόπος που εισάγουμε δεδομένα είναι με τις φόρμες sign up και tickets που έχουμε φτιάξει. Όπου, με την sign up ο χρήστης κάνει την εγγραφή του στην ιστοσελίδα και την φόρμα των tickets όπου κλείνει το εισιτήριό του.

### 5. ΚΥΡΙΕΣ ΕΝΕΡΓΕΙΕΣ ΓΙΑ ΤΗΝ ΟΛΟΚΛΗΡΩΣΗ ΤΟΥ ΠΡΟΤΖΕΚΤ:

**Νικόλ Νικολάου:** top nav bar, bottom nav bar, Home Page, What's on, Explore, Profile, Change Password, erd, λογικό σχεσιακό μοντέλο, βάση δεδομένων, ένωση με βάση δεδομένων, script για fake data, handlebars, index/app, routes, queries.

**Σταμάτιος Νικολουδάκης:** Sign up, Log in, Tickets and Prices, Ticket Form, Card-info, Payment, erd, λογικό σχεσιακό μοντέλο, βάση δεδομένων, ένωση με βάση δεδομένων, index/app, routes, sessions, queries.

### 6.ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ:

Πίνακας 1: Χρονοδιάγραμμα

Εβδομάδα 1:	Αξιολόγηση και μελέτη του θέματος. Κατασκευή erd και διαγράμματος οντοτήτων συσχετίσεων.
Εβδομάδα 2:	Front end -html, css , bootstrap
Εβδομάδα 3:	Front end -html, css, bootstrap, js
Εβδομάδα 4:	Front end -html, css, bootstrap, js
Εβδομάδα 5:	Front end js.
Εβδομάδα 6:	Back-end hbs/routes και κατασκευή Βάσης δεδομένων.
Εβδομάδα 7:	Διακοπές Πάσχα.
Εβδομάδα 8:	Ένωση της βάσης δεδομένων με την ιστοσελίδα και εισαγωγή δεδομένων στη βάση. Sessions.
Εβδομάδα 9:	Sessions, Queries και διορθώσεις στο front end.
Εβδομάδα 10:	Γενικές διορθώσεις, upload στο Heroku, αναφορά και παρουσίαση.

## 7. ΕΓΧΕΙΡΙΔΙΟ ΧΡΗΣΗΣ:

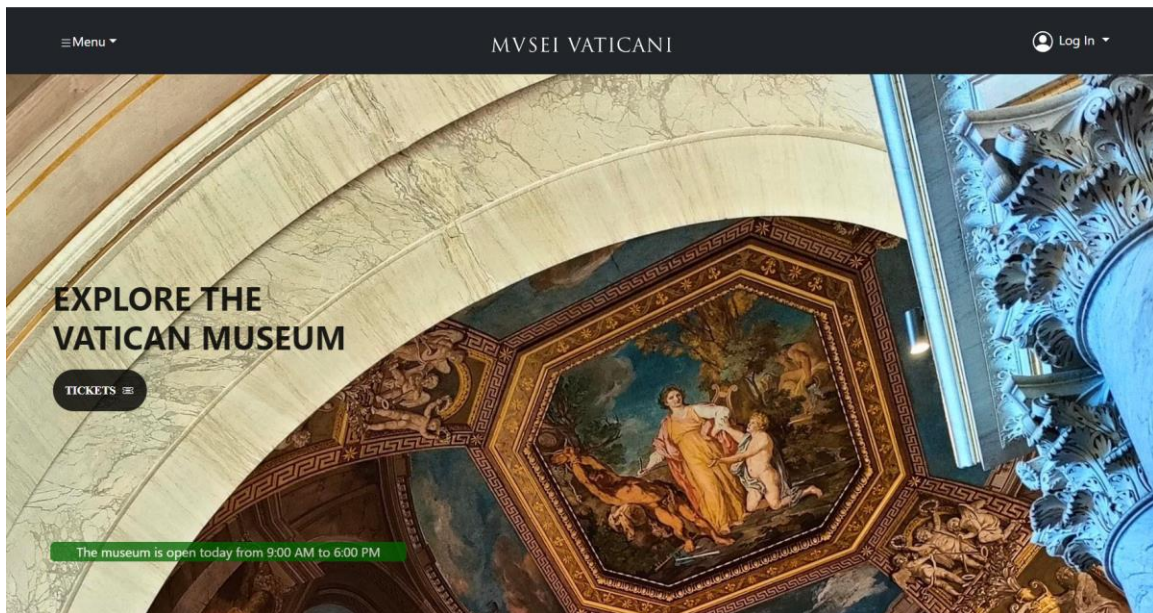
### 7.1 Έναρξη εφαρμογής:

Όταν το μουσείο είναι κλειστό:



Εικόνα 1.1: Αρχική σελίδα – Μουσείο Κλειστό.

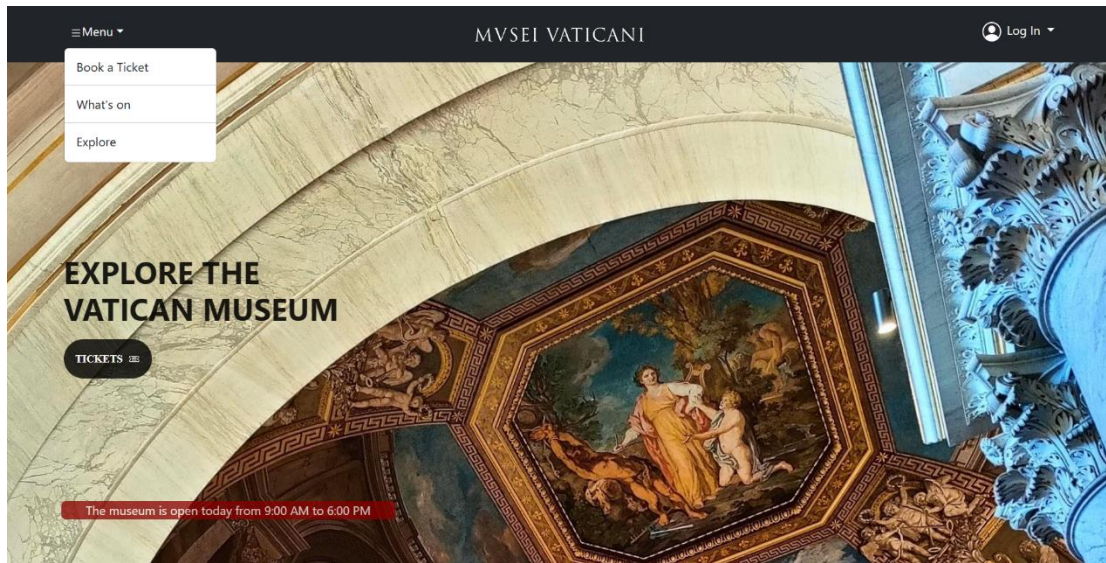
Όταν το μουσείο είναι ανοιχτό:



Εικόνα 2.2: Αρχική σελίδα – Μουσείο Ανοιχτό.



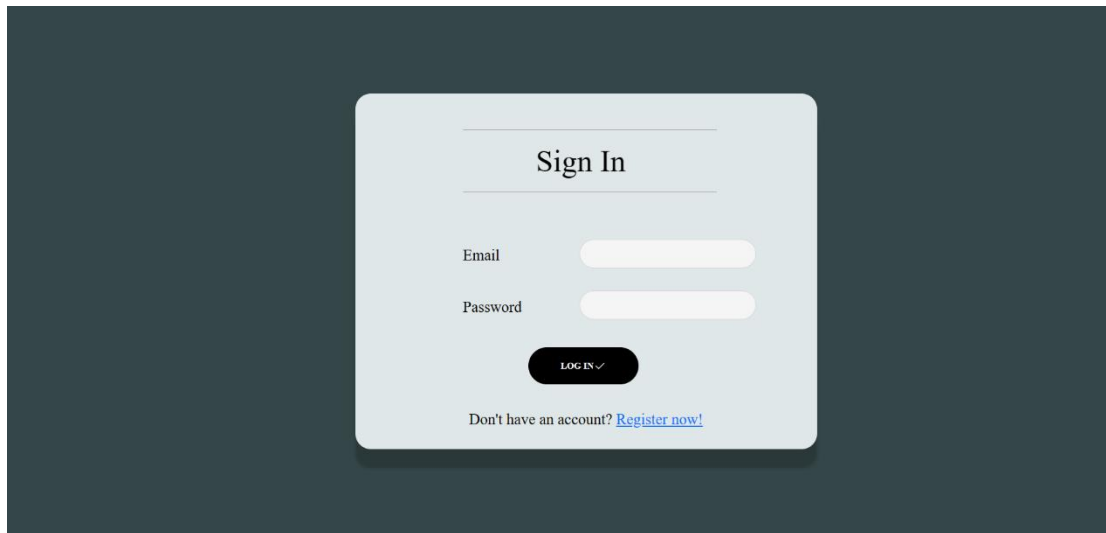
### 7.2 Επιλογή μενού:



Εικόνα 3:Μενού.

Επιλέγοντας το μενού ο χρήστης έχει την δυνατότητα να επισκεφθεί τις ιστοσελίδες what's on και explore στις οποίες μπορεί να εξερευνήσει τα περιεχόμενα του μουσείου καθώς τις εκθέσεις και τα εκθέματα τα οποία περιλαμβάνει. Ταυτόχρονα μπορεί να επιλέξει να μεταφερθεί στην ιστοσελίδα κράτησης εισιτηρίων.

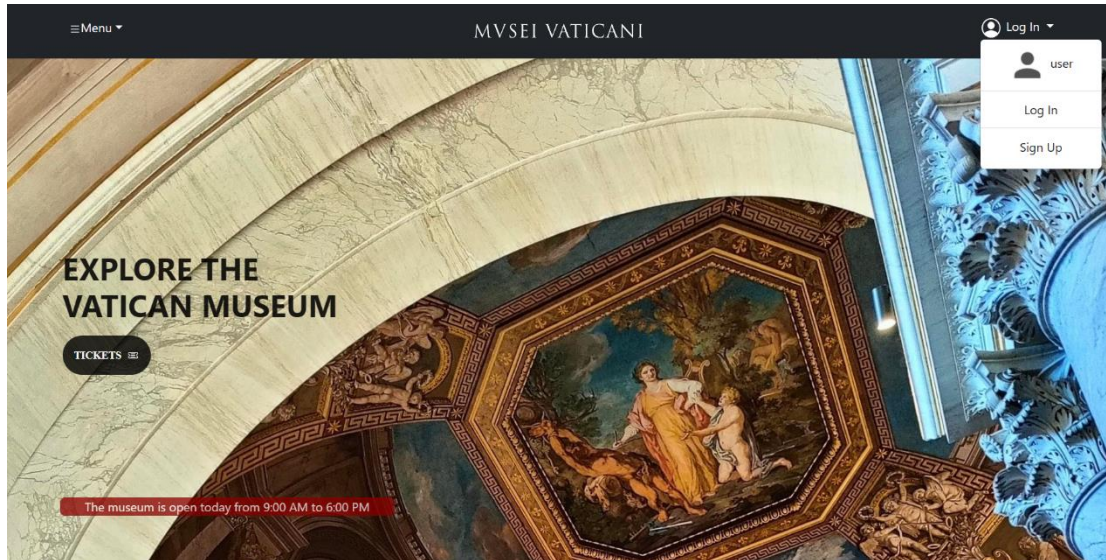
### 7.3 Επιλογή κράτησης εισιτηρίων:



Εικόνα 4:Φόρμα σύνδεσης χρήστη.

Με την επιλογή κράτησης εισιτηρίων, όταν ο χρήστης δεν είναι συνδεδεμένος μεταφέρεται στην σελίδα της σύνδεσης χρήστη, αν δεν έχει λογαριασμό θα πρέπει να δημιουργήσει έναν εκ νέου, κάνοντας χρήση της υπερσύνδεσης που βρίσκεται στο κάτω μέρος της φόρμας σύνδεσης.

### 7.4 Menu Log in / Sign up:



Εικόνα 5:Log in μενού.

Επίσης ο χρήστης έχει την δυνατότητα να μεταφερθεί στις ιστοσελίδες σύνδεσης και εγγραφής χρήστη μέσω της κεντρικής σελίδας με την επιλογή του log in.

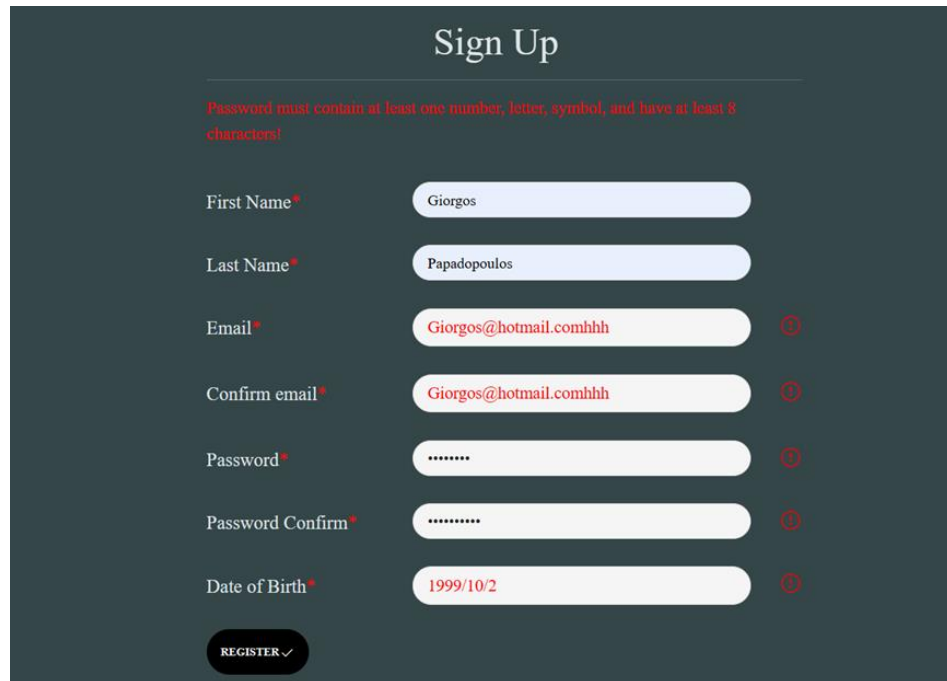
### 7.4 Sign up:

The image shows a 'Sign Up' form on a dark background. The form has the following fields: 'First Name' (with 'George' entered), 'Last Name' (with 'Papadopoulos' entered), 'Email' (with 'yourname@email.com' entered), 'Confirm email' (with 'yourname@email.com' entered), 'Password', 'Password Confirm', and 'Date of Birth' (with 'YYYY/MM/DD' entered). There is a red asterisk next to each field label. At the bottom, there is a 'REGISTER ✓' button. Below the button, there is a small text block: 'Register your account in order to access your orders later. We are committed to keeping your information secure and confidential.'

Εικόνα 6:Φόρμα εγγραφής χρήστη.

Ο χρήστης μεταφέρεται στην φόρμα εγγραφής στην οποία πρέπει να βάλει τα κατάλληλα στοιχεία.

Σε περίπτωση που τα στοιχεία είναι λάθος υπάρχει ανάδραση η οποία ενημερώνει τον χρήστη.



Sign Up

Password must contain at least one number, letter, symbol, and have at least 8 characters!

First Name\* Giorgos

Last Name\* Papadopoulos

Email\* Giorgos@hotmail.comhhh ❌

Confirm email\* Giorgos@hotmail.comhhh ❌

Password\* ..... ❌

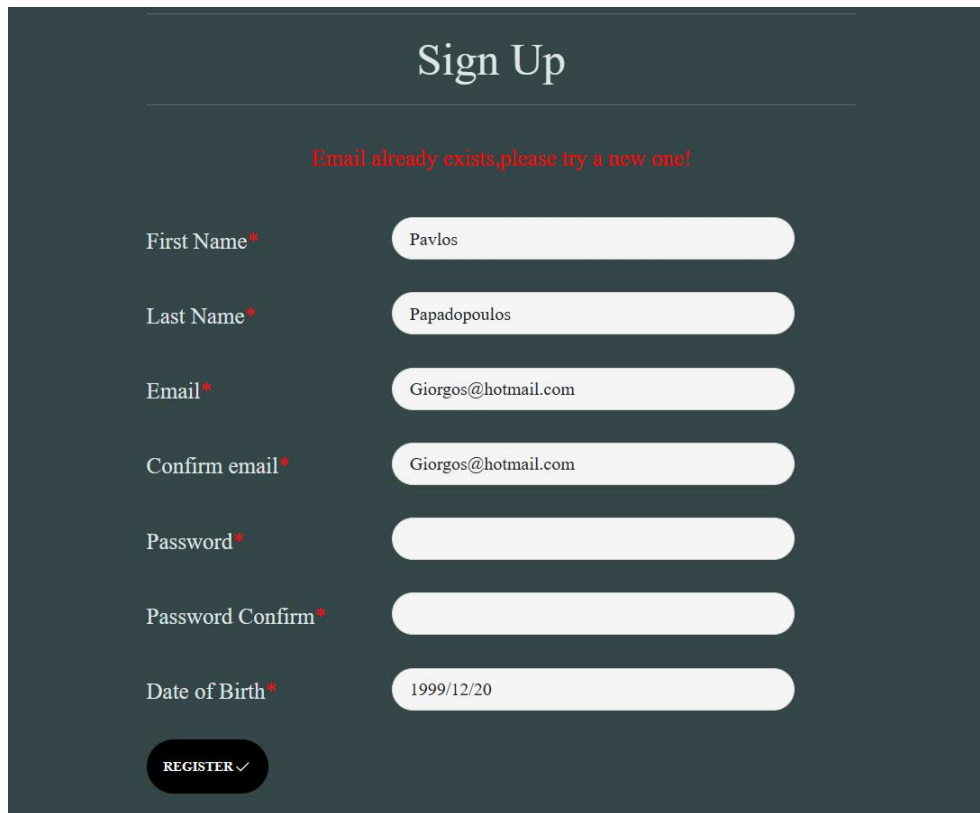
Password Confirm\* ..... ❌

Date of Birth\* 1999/10/2 ❌

REGISTER ✓

Εικόνα 5.1: Φόρμα εγγραφής με εσφαλμένα στοιχεία.

Ακόμη εάν ο χρήστης προσπαθήσει να εισαγάγει κάποιο email το οποίο υπάρχει ήδη στην βάση δεδομένων, τότε προκύπτει σφάλμα που ενημερώνει τον χρήστη.



Sign Up

Email already exists, please try a new one!

First Name\* Pavlos

Last Name\* Papadopoulos

Email\* Giorgos@hotmail.com

Confirm email\* Giorgos@hotmail.com

Password\*

Password Confirm\*

Date of Birth\* 1999/12/20

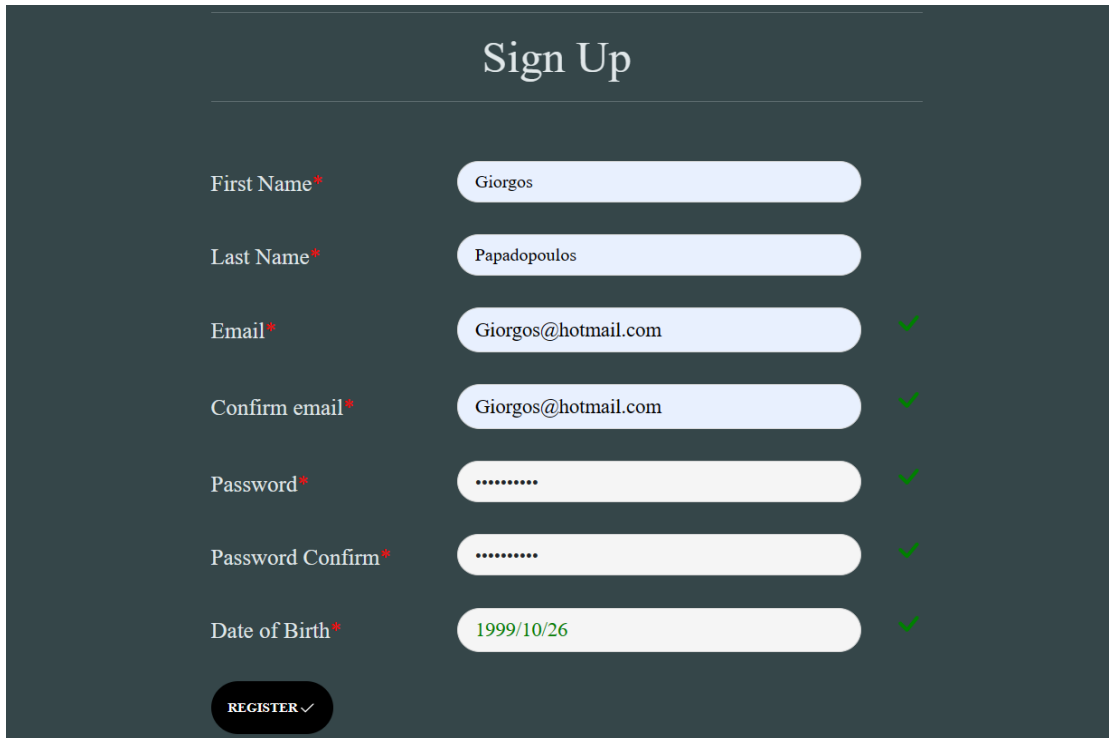
REGISTER ✓

Εικόνα 5.2: Φόρμα εγγραφής με υπάρχων email.



## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Αν τα δεδομένα που εισαγάγει ο χρήστης είναι σωστά, ενημερώνεται από ανάδραση και μπορεί να προχωρήσει στην εγγραφή.



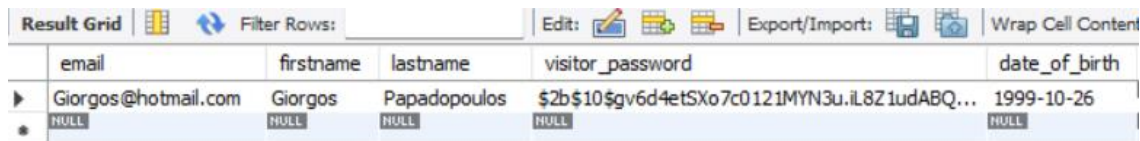
The image shows a 'Sign Up' form with the following fields and values:

Field	Value	Status
First Name*	Giorgos	
Last Name*	Papadopoulos	
Email*	Giorgos@hotmail.com	✓
Confirm email*	Giorgos@hotmail.com	✓
Password*	.....	✓
Password Confirm*	.....	✓
Date of Birth*	1999/10/26	✓

At the bottom, there is a black button labeled 'REGISTER ✓'.

Εικόνα 5.3: Επιτυχής εγγραφή.

Μετά την επιτυχή εγγραφή ο νέος χρήστης καταχωρείται στην βάση δεδομένων.



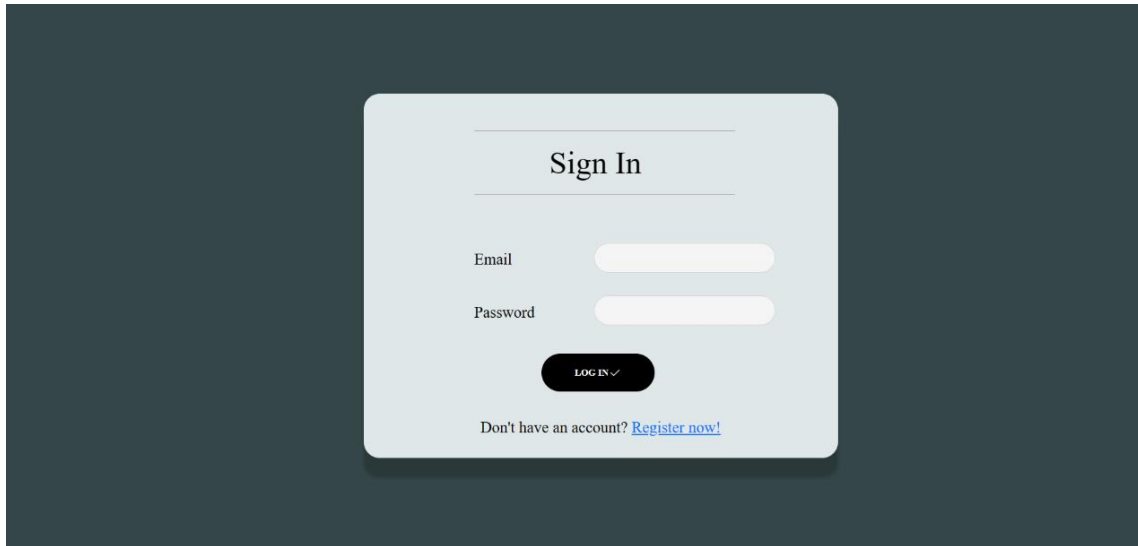
	email	firstname	lastname	visitor_password	date_of_birth
▶	Giorgos@hotmail.com	Giorgos	Papadopoulos	\$2b\$10\$gv6d4etSXo7c0121MYN3u..iL8Z1udABQ...	1999-10-26
*	NULL	NULL	NULL	NULL	NULL

Εικόνα 6: Εισαγωγή νέου χρήστη στη βάση δεδομένων.

Παρατηρούμε ότι ο κωδικός του χρήστη έχει κρυπτογραφηθεί για λόγους ασφαλείας.

Αφού γίνει η εγγραφή, ο χρήστης μεταφέρεται στην φόρμα σύνδεσης χρήστη.

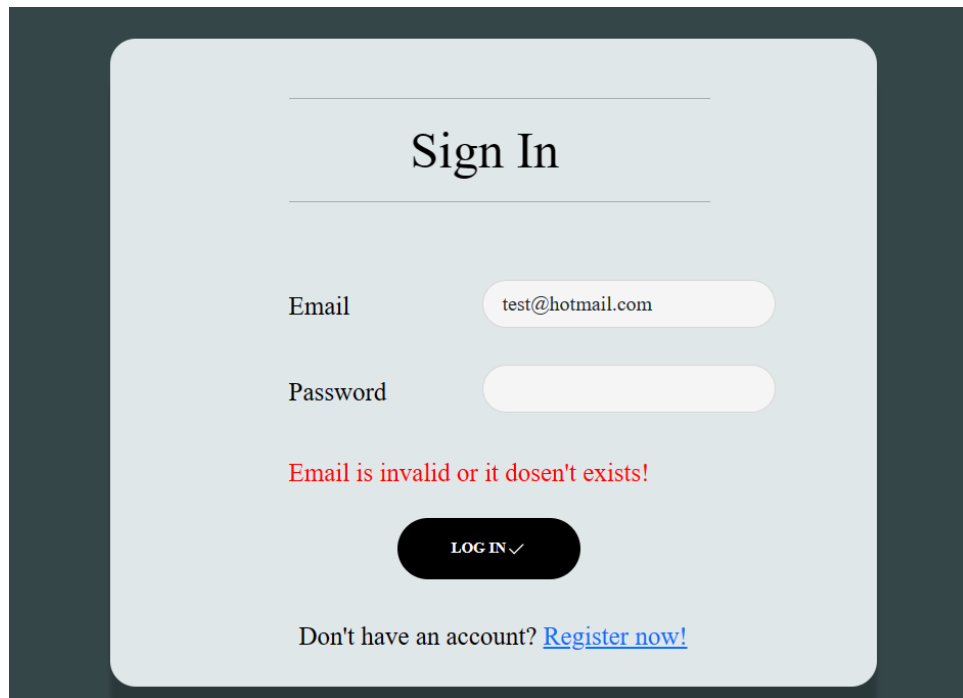
### 7.5 Σύνδεση χρήστη:



A screenshot of a 'Sign In' form. The form is light blue with rounded corners and is centered on a dark blue background. It has a title 'Sign In' at the top. Below the title are two input fields: 'Email' and 'Password'. Below the 'Password' field is a black button with the text 'LOG IN ✓'. At the bottom of the form, there is a link that says 'Don't have an account? [Register now!](#)'.

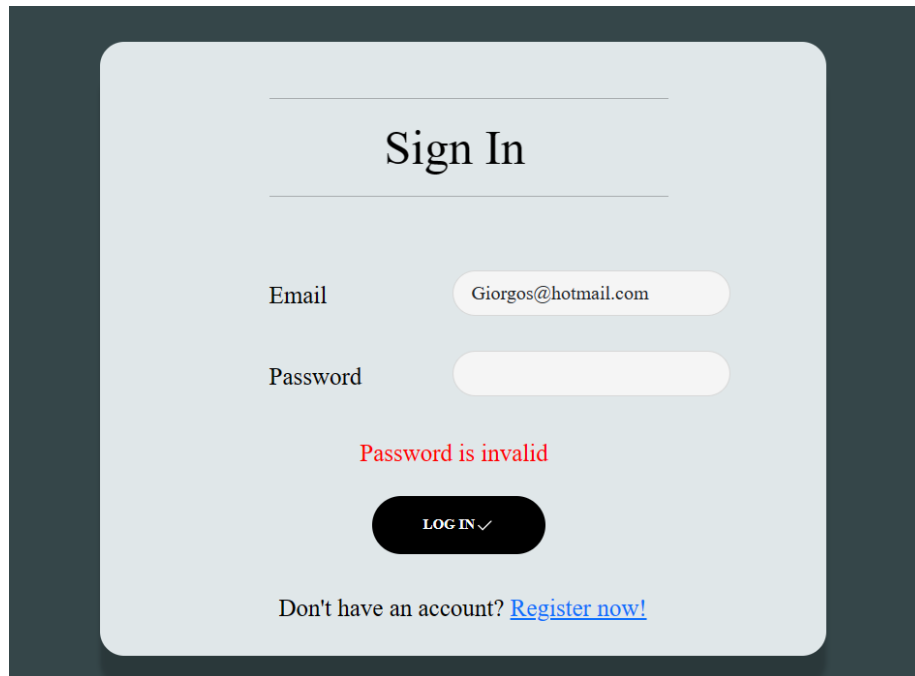
Εικόνα7: Σύνδεση χρήστη.

Ο χρήστης θα πρέπει να καταχωρήσει το email και τον κωδικό που έβαλε στην φόρμα εγγραφής, εάν το email ή ο κωδικός δεν είναι σωστά τότε ενημερώνεται με ανάδραση.



A screenshot of the 'Sign In' form, similar to the one above, but with an error message. The 'Email' field now contains the text 'test@hotmail.com'. Below the 'Password' field, there is a red error message that says 'Email is invalid or it dosen't exists!'. The 'LOG IN ✓' button and the 'Register now!' link are still present at the bottom.

Εικόνα7.1: Λανθασμένο email στην φόρμα εγγραφής.



Sign In

Email: Giorgos@hotmail.com

Password:

Password is invalid

LOG IN ✓

Don't have an account? [Register now!](#)

Εικόνα 7.2 :Λανθασμένος κωδικός σύνδεσης.

Αν ο χρήστης καταχωρήσει τα σωστά στοιχεία ,τότε ξεκινάει η συνεδρία η οποία αποθηκεύεται στην βάση και ο χρήστης μεταφέρεται στην σελίδα κράτησης εισιτηρίων.

session_id	expires	data
gg-R8wN46G1FZ_luGtFjOtizZNIksCBR	1685315889	{ "cookie": { "originalMaxAge": 85135727, "expires": "2023-05-28T23:18:09.017Z", "httpOnly": true, "path": "/", "email": "...
NULL	NULL	NULL

Εικόνα 8: Η συνεδρία στην βάση.

## 7.6 Σελίδα κράτησης εισιτηρίων:

Menu

MUSEI VATICANI

Giorgos

# Tickets and Prices

Plan your visit to the Vatican Museum and make the most of your time exploring the rich history and art on display. All entrance tickets include admission to the permanent collection as well as any current temporary exhibitions. We highly recommend booking your tickets in advance online. This will not only save you time but also guarantee your spot and allow you to skip the lines on the day of your visit. Don't miss out on this incredible opportunity to discover one of the world's most fascinating collections of art and artifacts.

## Ticket Prices

Ticket prices differ depending on the exhibition, tap here for more information.

1. Tickets

2. Your Information

3. Payment

Εικόνα 9: Σελίδα κράτησης εισιτηρίων.

Καθώς ο χρήστης έχει μεταφερθεί στην σελίδα έχει την επιλογή να επεξεργαστεί το προφίλ του ή να κλείσει εισιτήρια, η επεξεργασία του προφίλ είναι διαθέσιμη στις περισσότερες σελίδες.

### 7.7 Επεξεργασία προφίλ χρήστη:

The screenshot shows a dark header bar with a '← HOME' button on the left and 'MVSEI VATICANI' text on the right. Below the header is a light gray form with three input fields: 'Email:' containing 'Giorgos@hotmail.com', 'Name:' containing 'George', and 'Last Name:' containing 'Papadopoulos'. At the bottom of the form are two buttons: 'Change Password' and 'Submit'.

Εικόνα 10: Επεξεργασία προφίλ χρήστη.

Ο χρήστης μεταφέρεται στο προφίλ του, στο οποίο έχει την δυνατότητα να αλλάξει το email, το όνομα, και το επίθετο του.

Καταχωρώντας λανθασμένα στοιχεία ο χρήστης ενημερώνεται με ανάδραση και δεν μπορεί να υποβάλει τα νέα δεδομένα :

The screenshot shows the same form as before, but with red error messages. The 'Email:' field has the error 'Giorgos@hotmail.com', the 'Name:' field has 'Georgseel', and the 'Last Name:' field has 'Papadopoulosl'. The 'Change Password' and 'Submit' buttons are still present at the bottom.

Εικόνα 10.1: Λανθασμένα δεδομένα στο προφίλ.

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Όπως είδαμε και παραπάνω στην φόρμα εγγραφής αν το νέο email του χρήστη υπάρχει στην βάση τότε δεν έχει την δυνατότητα να το ενημερώσει:

The screenshot shows a web interface for MVSEI VATICANI. At the top is a dark header with a '←HOME' button and the text 'MVSEI VATICANI'. Below the header is a light blue registration form. The form contains the following fields and values:

- Email: Giorgos@hotmail.com
- Name: George
- Last Name: Papadopoulos

Below the form, there is a red error message: "Email already exists, please try a new one!". At the bottom of the form are two buttons: "Change Password" and "Submit".

Εικόνα 10.2: Αποτυχία ενημέρωσης στοιχείων.

Εφόσον τα στοιχεία είναι σωστά τότε μπορεί να τα υποβάλει και να ενημερωθούν στην βάση.

The screenshot shows the same web interface as before, but with a green success message: "Data updated successfully!". The form fields and buttons remain the same.

Εικόνα 10.3: Επιτυχία επεξεργασίας προφίλ.

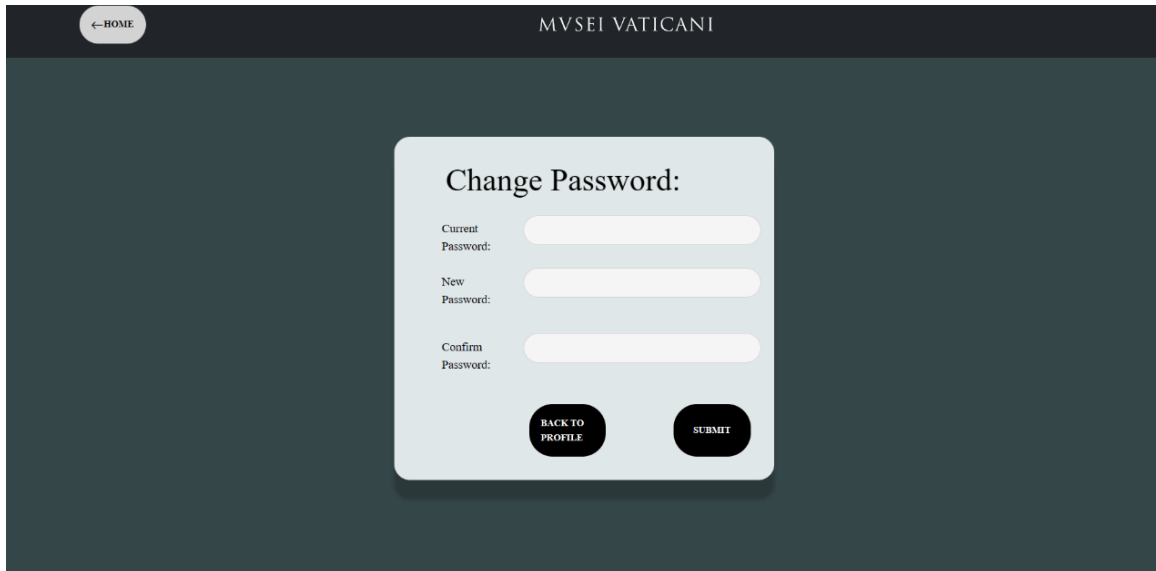
Ενημερωμένα στοιχεία στην βάση:

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap Cell Content:					
	email	firstname	lastname	visitor_password	date_of_birth
▶	Giorgos1@hotmail.com	George	Papadopoulos	\$2b\$10\$gv6d4etSXo7c0121MYN3u.iL8Z1udABQ...	1999-10-26
*	NULL	NULL	NULL	NULL	NULL

Εικόνα 11: Δεδομένα μετά την ενημέρωση

### 7.7 Αλλαγή κωδικού Χρήστη:

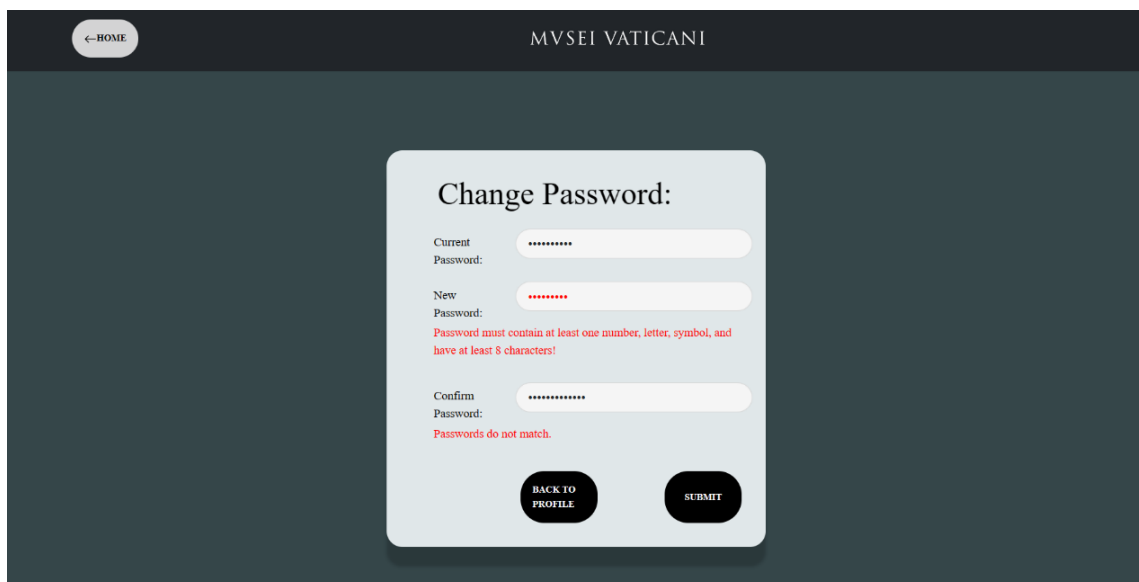
Στη συνέχεια ο χρήστης αν επιθυμεί μπορεί να επιλέξει το κουμπί αλλαγής κωδικού:



The screenshot shows a web application interface for 'MVSEI VATICANI'. At the top, there is a dark header with a '← HOME' button on the left and the text 'MVSEI VATICANI' on the right. The main content area has a dark blue background. In the center, there is a light blue rounded rectangle containing the 'Change Password' form. The form has the title 'Change Password:' and three input fields labeled 'Current Password:', 'New Password:', and 'Confirm Password:'. Below the input fields are two buttons: 'BACK TO PROFILE' and 'SUBMIT'.

Εικόνα 12.1: Σελίδα αλλαγής κωδικού.

Αν ο κωδικός δεν ικανοποιεί τα επιθυμητά πρότυπα τότε ο χρήστης ενημερώνεται με ανάδραση και δεν μπορεί να τον αλλάξει:

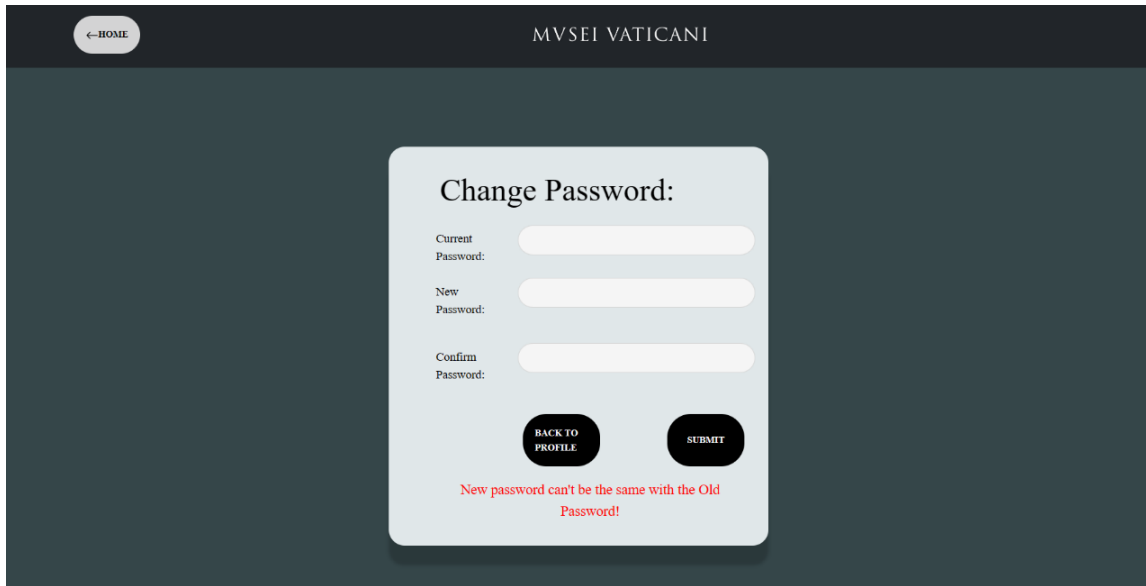


This screenshot shows the same 'Change Password' form as in the previous image, but with validation errors. The 'Current Password' field is filled with dots. The 'New Password' field is filled with red dots, and below it, a red error message reads: 'Password must contain at least one number, letter, symbol, and have at least 8 characters!'. The 'Confirm Password' field is also filled with dots, and below it, a red error message reads: 'Passwords do not match.'. The 'BACK TO PROFILE' and 'SUBMIT' buttons are still present at the bottom of the form.

Εικόνα 12.2: Εσφαλμένη εισαγωγή κωδικών.

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

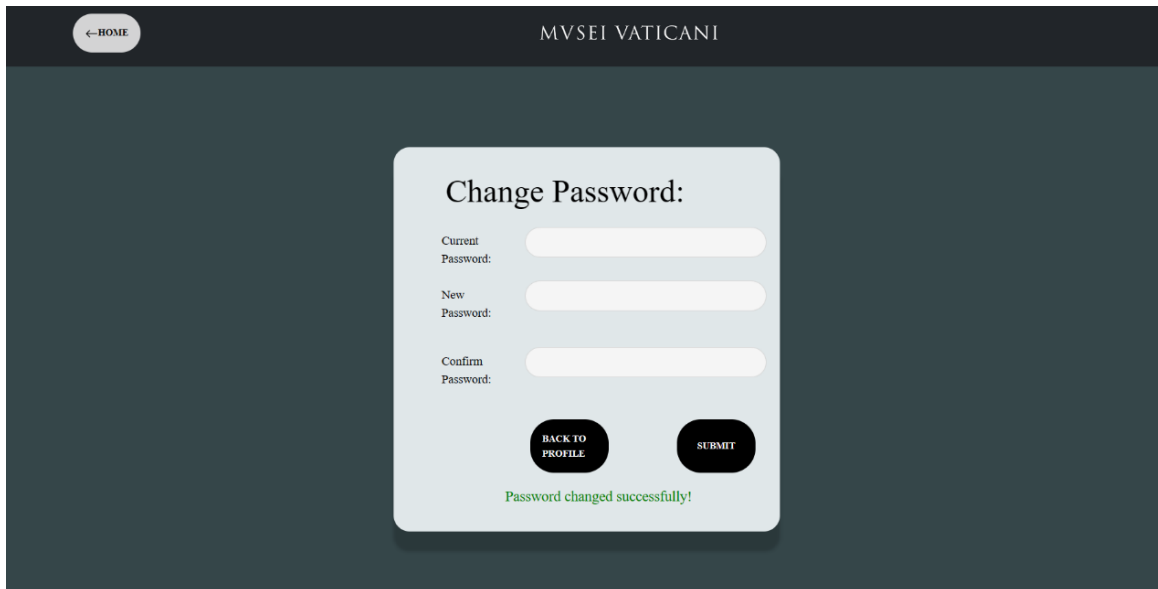
Επιπρόσθετα ο χρήστης δεν μπορεί να εισαγάγει τον ίδιο κωδικό με τον προηγούμενο που διέθετε:



The screenshot shows a web interface for 'MVSEI VATICANI'. At the top left is a '← HOME' button. The main content area is a light blue box titled 'Change Password:'. It contains three input fields labeled 'Current Password:', 'New Password:', and 'Confirm Password:'. Below the fields are two buttons: 'BACK TO PROFILE' and 'SUBMIT'. A red error message is displayed at the bottom: 'New password can't be the same with the Old Password!'.

Εικόνα 12.3: Εσφαλμένη αλλαγή κωδικού.

Τελικά αν δεν υπάρχουν σφάλματα ο κωδικός αλλάζει με επιτυχία:




The screenshot shows the same 'Change Password' form as in the previous image. However, the error message is gone, and a green success message is displayed at the bottom: 'Password changed successfully!'.

Εικόνα 12.4: Επιτυχής αλλαγή κωδικού.

### 7.8 Κράτηση εισιτηρίων:

## Choose Your Ticket




**Standard Ticket**

Vatican Museums & Sistine Chapel Entrance ticket.

Price:20€

[Read more...](#)




**Early Access Ticket**

Early Vatican,Sistine Chapel & Basilica Guided Tour.

Price:35€

[Read more...](#)




**Turn The lights On**

Turning the lights on at the Vatican Museums.

Price:45€

[Read more...](#)



**Skip-The-Ticket-Line-Tour**

Vatican Museums & Sistine Chapel Skip-The-Ticket-Line-Tour.

Price:60€

[Read more...](#)

[TICKETS](#)

1. Tickets

2. Your Information

3. Payment

Εικόνα 13: Επιλογή εισιτηρίου.

Ο χρήστης μπορεί να επιλέξει ανάμεσα σε τέσσερα διαφορετικά εισιτήρια για την κράτηση του.

## Book your tickets

How many people are in your group?

5

Pick a date

28/05/2023

Time slot

9:00-12:00

Country

Greece

Ticket type

Student

☐ I accept the [Terms and Conditions](#)

[SUBMIT](#)

1. Tickets

2. Your Information

3. Payment

Εικόνα 14: Κράτηση εισιτηρίου.



## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Ύστερα από την επιλογή εισιτηρίου επιλέγει για πόσα άτομα επιθυμεί να κρατήσει εισιτήρια, την ημερομηνία κράτησης, την ώρα κράτησης, την χώρα από την οποία κατάγεται αλλά και τον τύπο του εισιτηρίου. Υπάρχουν 3 τύποι εισιτηρίων, ενήλικοι οι οποίοι πληρώνουν ολόκληρη την τιμή του εισιτηρίου, μαθητές οι οποίοι πληρώνουν την μισή τιμή και παιδιά κάτω των 15 που έχουν 40% έκπτωση.

Ο χρήστης στέλνει τα δεδομένα ασύγχρονα στην βάση δεδομένων και προβάλλονται τα εισιτήρια τα οποία έχει κρατήσει:

Time slot

12:00-15:00

Country

Greece

Ticket type

Under 15

☐ I accept the [Terms and Conditions](#)

SUBMIT ✓

Your order will be processed and your tickets emailed to you as soon as possible.

Information:

*Vatican Museum Map*  
*Piazza della Citta Leonina, Rome*  
*00120, Vatican City*  
*Phone: +39 06 6988 4676*  
*Email: [info@thevaticanmuseums.com](mailto:info@thevaticanmuseums.com)*

Tickets:

*135€*  
*217.5€*  
*317.5€*  
*421€*

1. Tickets

2. Your Information

3. Payment

Εικόνα14.1: Κρατημένα εισιτήρια και τιμή.

Τα δεδομένα κράτησης αποθηκεύονται στην βάση δεδομένων:

ticket_id	book_firstname	book_lastname	ticket_type	book_time	book_date	country	ticket_discount	price	book_email	card_no_book
475	George	Papadopoulos	Early Access Ticket	12:00	2023-05-28	Greece	Adult Ticket	35	Giorgos@hotmail.com	S1Y5UzAxc1IybHlaNHd4dGdFcWJTnk1
476	George	Papadopoulos	Early Access Ticket	12:00	2023-05-28	Greece	Student Ticket	17.5	Giorgos@hotmail.com	S1Y5UzAxc1IybHlaNHd4dGdFcWJTnk1
477	George	Papadopoulos	Early Access Ticket	12:00	2023-05-28	Greece	Student Ticket	17.5	Giorgos@hotmail.com	S1Y5UzAxc1IybHlaNHd4dGdFcWJTnk1
478	George	Papadopoulos	Early Access Ticket	12:00	2023-05-28	Greece	Under 15	21	Giorgos@hotmail.com	S1Y5UzAxc1IybHlaNHd4dGdFcWJTnk1
479	George	Papadopoulos	Early Access Ticket	12:00	2023-05-28	Greece	Under 15	21	Giorgos@hotmail.com	S1Y5UzAxc1IybHlaNHd4dGdFcWJTnk1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Εικόνα15: Δεδομένα κράτησης στην βάση.

### 7.9 Πληρωμή των εισιτηρίων:

Μετά την κράτηση ο χρήστης προχωράει στην πληρωμή των εισιτηρίων.

Payment

Enter your card information

Card Number

Expiration Date

CVV

Cardholder Name

PAYMENT ✓

We accept the following methods of payment:

VISA mastercard Diners Club AMERICAN EXPRESS

1. Tickets 2. Your Information 3. Payment

Εικόνα 16: Φόρμα εισαγωγής στοιχείων κάρτας.

Αν τα στοιχεία της κάρτας είναι εσφαλμένα τότε ενημερώνεται με ανάδραση και δεν μπορεί να τα καταχωρήσει:

Payment

Enter your card information

Card Number   
Card Number is not valid!

Expiration Date   
Expiration date is not valid!

CVV

Cardholder Name

PAYMENT ✓

We accept the following methods of payment:

VISA mastercard Diners Club AMERICAN EXPRESS

1. Tickets 2. Your Information 3. Payment

Εικόνα 16.1: Αποτυχία εισαγωγής στοιχείων κάρτας.

## PROJECT Προγραμματισμός Διαδικτύου ECE\_ΓΚ802

Εφόσον τα στοιχεία είναι σωστά προχωράμε στην εισαγωγή τους στην βάση:

card_no	cardholder_name	cvv	end_date
aWgvcFAyNzVqaGhUdE1abFJ2d01yYmFoQIiwR...	ajdVTG5xVHJJOGprMWl3aDNqK2g0dz09	Mk45Tmcyb3hYN3J0MFNidStIR2NnQT09	2028-10-01
S1Y5UzAxc1IybHlaNHd4dGdFcWJTNkk1SXNyYk...	SEZsNGxRbTY0bi91dUs5VStqTjNWZz09	eVQ5N0xOMVIEUFF6TE53T25RUjJQZz09	2028-10-01
ZXI4YnN5d0ZoeTJHVFRZZVBd0ROOEJ4MjVmaF...	N2p4aFZRbEFLUzFyVCtSMIFjR0V4UT09	eVQ5N0xOMVIEUFF6TE53T25RUjJQZz09	2025-10-01
NULL	NULL	NULL	NULL

Εικόνα 17: Κρυπτογραφημένα δεδομένα κάρτας.

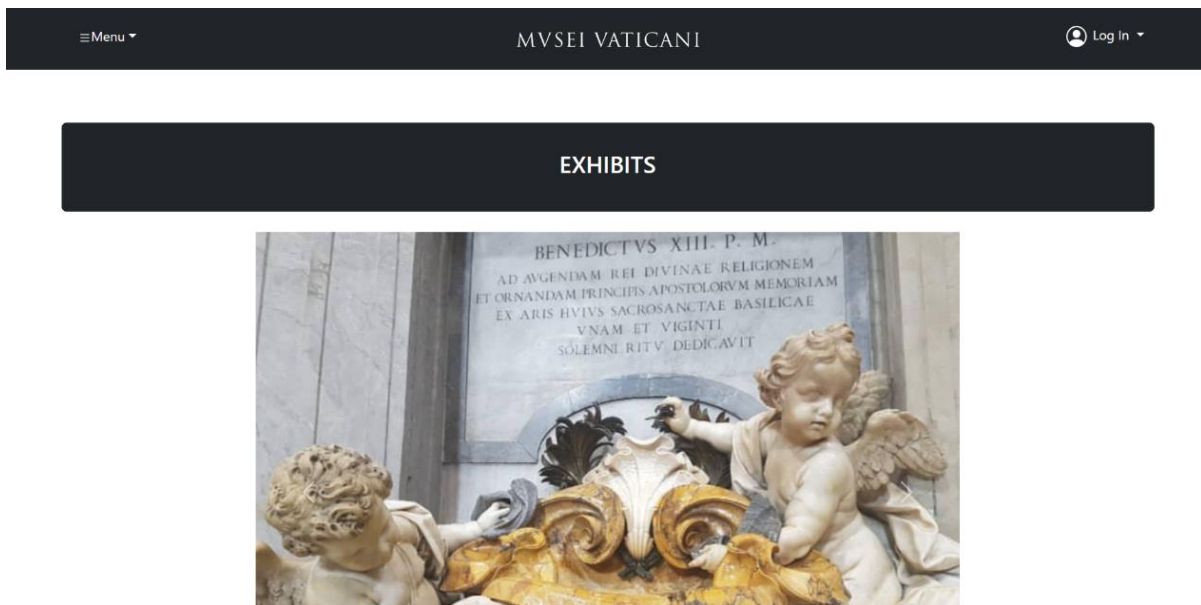
Όπως παρατηρούμε τα ευαίσθητα δεδομένα της κάρτας έχουν κρυπτογραφηθεί και εισαχθεί στην βάση.

### 7.10 Ενημέρωση και Πληροφόρηση του χρήστη:

Ο χρήστης μέσω των επιλογών explore και what's on μπορεί να δει εκθέματα του μουσείου, να διαβάσει πληροφορίες και ενδιαφέροντα γεγονότα για το μουσείο αλλά και να διαβάσει αναλυτικά για το κάθε Tour/Exhibition που προσφέρει το μουσείο.

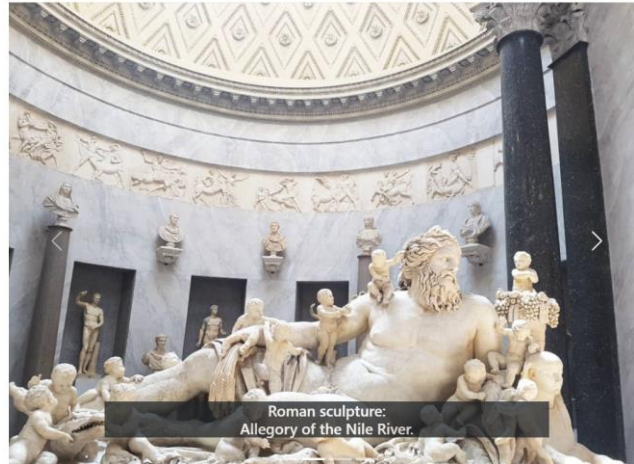
Να σημειωθεί ότι ο χρήστης έχει πρόσβαση σε αυτές τις σελίδες είτε είναι συνδεδεμένος είτε όχι.

#### 7.10.1 What's on:



Εικόνα 18.1: Ορισμένα εκθέματα του μουσείου.

## EXHIBITS



Εικόνα 18.2: Ορισμένα εκθέματα του μουσείου.

## GUIDED TOURS & EXHIBITIONS



### Vatican: Museums & Sistine Chapel Entrance Ticket

Take a stirring journey through art and religious history with skip-the-ticket-line entry to the Vatican Museums and Sistine Chapel. Avoid the long ticket lines and then enjoy your visit at your own pace. Wander through the Greek Cross Gallery to see elaborately-carved sarcophagi that might just contain the earthly remains of some of antiquity's famous kings and queens. Peek at the Cabinet of the Masks, or tour the Sala degli Animali to see a virtual menagerie of fantastic beasts, real and imagined. Walk the Upper Galleries, including the Gallery of Maps, to see how cartographers depicted the world through the ages. See masterpieces by Renaissance masters in the Raphael Rooms, and rest for a while in the Borgia Apartments. Then, it's time to marvel at Michelangelo's exquisite Sistine Chapel, the crown jewel of the Vatican, and perhaps even of all the world's art treasures.

Price: €20

[Book Now](#)

### Early Vatican, Sistine Chapel & Basilica Guided Tour

Εικόνα 18.2: πληροφορίες για τα Tours και τα Exhibitions.

### 7.10.2 Explore:

Menu

MUSEI VATICANI

Log In

## VATICAN MUSEUM

The **Vatican Historical Museum** was founded in **1973** at the behest of Paul VI, and was initially hosted in environments under the Square Garden. In 1987, it moved to the main floor of the Lateran Palace, where it opened in March 1991.

The Vatican Museums house the exquisite art collections, archeology and ethnology created by the various Pontiffs over the centuries, but they also feature some of the most unique and historically and artistically significant places of the Apostolic Palaces.

The most famous art pieces in the Vatican City are:

- Sistine Chapel's Ceiling
- Stefaneschi Triptych
- Laocoön and His Sons
- Raphael Rooms
- Saint Longinus
- The Entombment of Christ
- St Jerome in the Wilderness
- Belvedere Torso



Εικόνα 19: Η σελίδα explore.

### 8. ΒΙΒΛΙΟΓΡΑΦΙΑ:

[1] Τεχνολογίες και Προγραμματισμός στον Παγκόσμιο Ιστό 2η έκδοση Δουληγέρης Χρήστος, Μαυροπόδη Ρόζα, Κοπανάκη Εύη, Καράλης Απόστολος, New Tech Pub.

[2] <https://mathesis.cup.gr/dashboard> Εισαγωγή στην ανάπτυξη ιστοσελίδων με HTML5, CSS3, Javascript (Ε)

[3] <https://mathesis.cup.gr/dashboard> - Προχωρημένα θέματα ανάπτυξης ιστοσελίδων (Ε)

[4] <https://mathesis.cup.gr/dashboard> Ανάπτυξη διαδικτυακών εφαρμογών με node.js (Νέο)

[5] <https://www.w3schools.com/>

[6] <https://www.w3schools.com/css/>

[7] <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

[8] <https://stackoverflow.com/>

[9] <https://www.npmjs.com/package/node>

[10] <https://www.w3schools.com/js/>

[11] <https://eclass.upatras.gr/courses/EE767/>