## JS Back-End Exam – Course Book

## Exam Rules:

1. You have **4 hours**
2. When you are ready, delete the ***node\_modules***folder, make sure all dependencies are listed in the ***package.json*** file and submit your archiving project.
3. You are provided with **HTML & CSS** resources
4. You may **add attributes** (such as **class** and **dataset**), but it is forbidden to **change** **existing** attributes (such as **class** and **id**)
5. You may **change "href"** attributes on links and add/change the **method** and **action** attributes of HTML Forms.
6. Use **Express.js** as a back-end framework
7. Use **MongoDB** as a database with [**mongoose**](https://mongoosejs.com/docs/)
8. You can use whatever **view engine** you like ([**express-handlebars**](https://www.npmjs.com/package/express-handlebars), EJS, Pug, etc.)
9. Use [**bcrypt**](https://www.npmjs.com/package/bcrypt) **for hashing the password**
10. The application **must start** from the file **"index.js"** on port **3000**
11. It is **forbidden** to **use** **React**, **Vue**, **Angular,** etc.
12. All applications will be **examined** as follows:
    * The needed **dependencies** are installed with the command "**npm install**"
    * The application is **started** with the command "**npm run start**"
13. The examiners are **NOT** allowed to **change** anything in your code and will **NOT** use any **instruments** **outside** of the course scope.
14. **Only the last submission will be evaluated!**

## Application Overview

Get acquainted with the provided **HTML and CSS** and create an application for the **course posts**.

The visitors can **view** the **Home page** and **All Course** page with available courses. They can also **register** with an **email**, **username**, and **password**, allowing them to create their offers and to **sign up for a course they are interested in** (if the **current user** is **not** the author of the offer). Authors can edit or delete their courses at any time.

## Functional Requirements

### Guest (Not logged in)

**Guest** navigation example:



The **application** should provide Guest (not logged in) users with the functionality to Login, Register, and **view** the Home page, **All Course** page, and **Details** page.

### Users (Logged in)

**User** navigation example:



The **application** should provide **Users** (logged in) with the functionality to:

* **View the Home page and all other pages with logged-in navigation**
* **View Course page**
* **Create а new Course Offer [Create Course Offers]**
* **Access course details page [Details]**
* **Sign up for a course offer (**if the **current user** is **not** the **owner of the course)**
* **Delete or Edit the course depending on the user's authentication (only for the owner of the current course)**

## Database Models

The **Database** of the **Course Book** application needs to support **two entities**:

### User

* username - string (required),
* email - string (required),
* password - string (required)

### Course

* **title - string (required),**
* **type - string (required),**
* **certificate - string (required),**
* **image: string (required),**
* **description: string (required),**
* **price – number (required),**
* **signUpList - a collection of Users (a reference to the User model)**
* **owner - object ID (a reference to the User model)**

**Note:** When a user **signs up for a course**, their **id** is added to the **signUpList** collection.

Implement the entities with the correct data types.

## Application Pages (80 pts)

#### Home Page (For logged in users and logged-out users)

Visualize **the last 3 added course offers. Each course must** show information about the **course title**, **type, certificate** and **price**, as well as a page with **details** about the **specific course.**

Картина, която съдържа текст, екранна снимка, човек

Описанието е генерирано автоматично

If there are **NO** courses in the database yet, display **"No course created yet."**

**Картина, която съдържа текст, екранна снимка, човек

Описанието е генерирано автоматично**

#### Register Page (Logged Out User)

**Register** a user inside the database with a **username, email, and password**. The **password** inside the **database** must be hashed (**use bcrypt**), and both passwords **must match**! After successful registration, you should **redirect** the user to the **Home page**. If the user tries to submit **invalid data**, an **error message** should be displayed and the already entered **data should be kept** in the input fields (except for the passwords).

A user who is already logged in should **NOT** be able to reach this page by typing its URL (e.g. “http://localhost:300/register”) in the browser address bar.

Картина, която съдържа текст, екранна снимка, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

#### Login Page (Logged Out User)

Logging an already registered user with the correct **email** and **password**. After successful login, you should **redirect the user to the Home page.** If the user tries to submit **invalid data**, an **error message** should be displayed and the already entered **data should be kept** in the input fields (except for the passwords).

A user who is already logged in should **NOT** be able to reach this page by typing its URL (e.g. “http://localhost:300/login”) in the browser address bar.

Картина, която съдържа екранна снимка, текст, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

#### Logout (Logged in user)

The logout action is available to **logged-in** users. Upon success, clear any session information and **redirect** the user to the **Home** page.

A user who is already logged in should **NOT** be able to reach this functionality by typing its URL (e.g. “http://localhost:300/logout”) in the browser address bar.

#### Create Course Offer (Logged in User)

The **Create** page is available to **logged-in users**. It contains a form for adding new course offers. Upon success, **redirect** the user to the **All Course** page. If the user tries to submit **invalid data**, an **error message** should be displayed and the already entered **data should be kept** in the input fields.

A guest user (who is not logged in) should **NOT** be able to reach this page by typing its URL (e.g. “http://localhost:300/courses/create/1234”) in the browser address bar.

Картина, която съдържа екранна снимка, текст, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

#### Course Catalog (For logged in users and logged out users)

A list of all course offers. Each course must be displayed with the **course image**, the **title**, the **type,** the **price**, and a button for **details** about the **specific course**. As in the picture below:

**Картина, която съдържа текст, екранна снимка, Шрифт

Описанието е генерирано автоматично**

The [**Details**] button should be a link to the **details** **page** for the current course post.

If there are **NO** offers in the database yet, display **"No course created yet."**

**Картина, която съдържа текст, екранна снимка

Описанието е генерирано автоматично**

#### Details Page (For logged in users and logged out users)

All users should be able **to see the details about the courses**. Clicking the **Details button** on the course card should display the details page. If the currently registered user **is the creator** of the course offer, the **Edit** and **Delete** buttons should be displayed. Otherwise, they **should not be available**.

Information about the course:

* **Title**
* **Type**
* **Certificate**
* **Price**
* **Email of the creator**
* **People who have signed up for this course:**
  + If any, separate their usernames with comma and space **", "**
  + If **not**, display **"Nobody has signed in for this course yet"**
* **Description**
* **Buttons** (Depending on the status of the currently logged in user)

#### Details Page (Logged out users)

If the user has not **logged in**, **no buttons** should be displayed.

Картина, която съдържа текст, екранна снимка, софтуер, Уебсайт

Описанието е генерирано автоматично

#### Details Page (Logged in user and creator of the current course)

If the **currently logged-in** user is the **owner** (the user who **created the course offer**), he should see the **[Delete]** and **[Edit]** buttons.

**Картина, която съдържа текст, електроника, екранна снимка, софтуер

Описанието е генерирано автоматично**

#### Details Page (Logged in user who is not signed up)

If the currently logged-in user is **not the owner** (a user who is **not** the creator of this course offer) and **has not signed up** for that course, he should see a **Sign up** button.

**Картина, която съдържа електроника, текст, екранна снимка, софтуер

Описанието е генерирано автоматично**

#### Details Page (Logged in user already sign up for course)

If the currently logged in user **is not the owner** and has **already sign up for that course**, he should see **[You’ve already signed up for this course!].**

**Картина, която съдържа текст, екранна снимка, софтуер, Уебсайт

Описанието е генерирано автоматично**

#### Delete Course (Logged in user and owner of the current course)

Each **owner** of the course post must be able to click on the **[Delete]** button and **delete the current course** from the database. After this, the user must be redirected to the **All Course** page.

A user who is not the owner of the course should **NOT** be able to reach this functionality by typing its URL (e.g. “http://localhost:300/courses/delete/1234”) in the browser address bar.

#### Edit Course (Logged in user and owner of the current course offer)

Each **owner** can edit their **course post**. Clicking the **[Edit]** button for a specific course on the details page should display the **Edit page**, all fields being populated with course data. It contains a form with input fields for all relevant properties. If **successful**, redirect the user to the **current course post details page**. If the user tries to submit **invalid data**, an **error message** should be displayed and the already entered **data should be kept** in the input fields.

A user who is not the owner of the course should **NOT** be able to reach this page by typing its URL (e.g. “http://localhost:300/courses/edit/1234”) in the browser address bar.

Картина, която съдържа текст, екранна снимка, софтуер, Компютърна икона

Описанието е генерирано автоматично

## Security Requirements (Routes Guards)

The Security Requirements are mainly **access** requirements. Configurations about which users can access specific functionalities and pages.

* Guest (not logged in) users can access the Home page.
* Guest (not logged in) users can access the Login page and functionality.
* Guest (not logged in) users can access the Register page and functionality.
* Guest (not logged in) and Users (logged in) can access the **All Course P**age (Listed all course offers).
* Guest (not logged in) can access the **Details** page without functionality.
* Users (logged in) can access the Home page.
* Users (logged in) can access the Details page and functionality.
  + **Users** (not course post owners) can **sign up for other courses**.
  + **Users** (course post owners) can **Edit** and **Delete** the current course post
* Users (logged in) can access the **Profile** page and functionality.
* Users (logged in) can access the **Create Course Offers** page and functionality.
* Users (logged in) can access the Logout functionality.

Use the following view for **invalid** **paths**:

Картина, която съдържа текст, екранна снимка, Шрифт, дизайн

Описанието е генерирано автоматично

## Validation and Error Handling (10 Pts)

### The application should notify the users about the result of their actions.

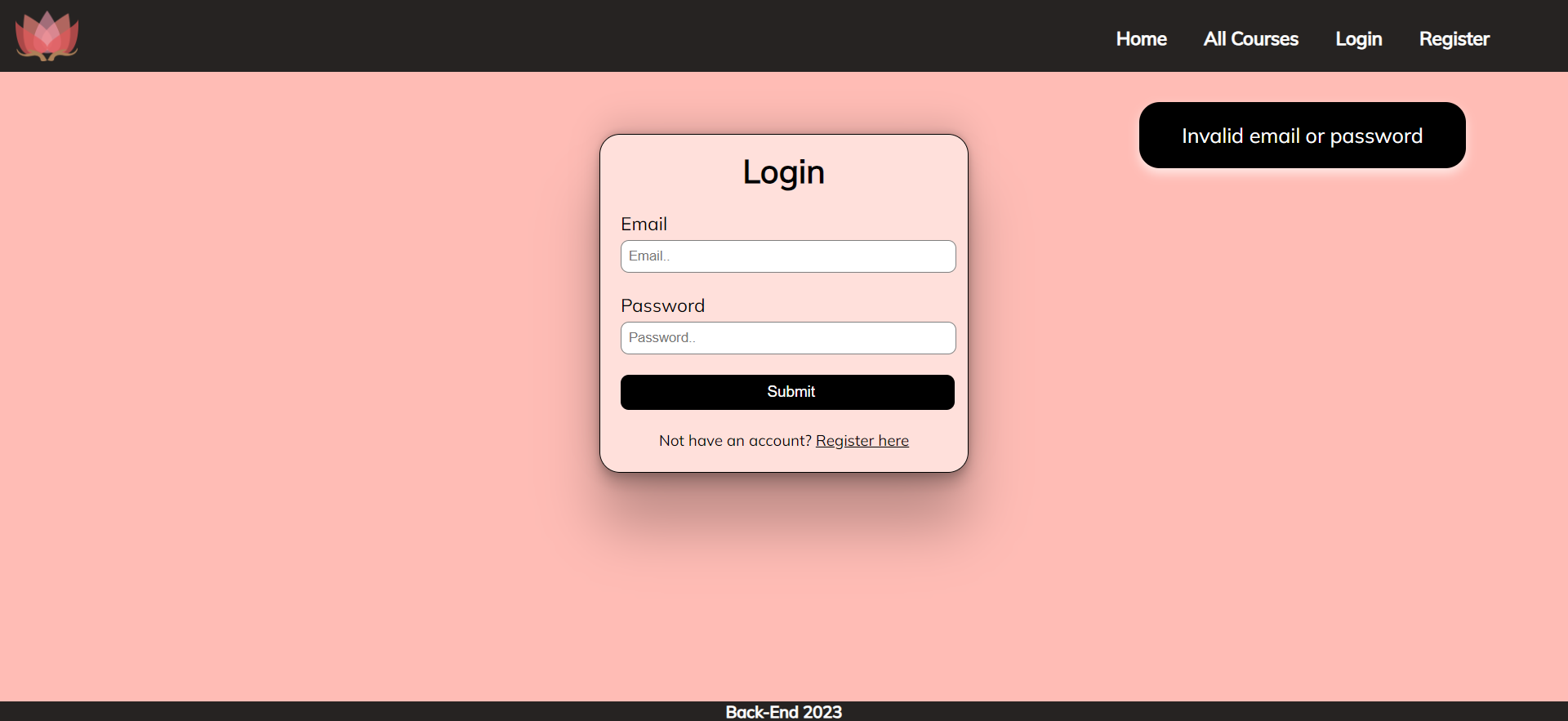
In case of error, you should display div with class "**errorContainer**"

### You can choose to display the first error or all of them. You have complete freedom to choose the content of the error message you will display.

### Login / Register

You should make the following validations:

* The **username** should be **at least** **2 characters** long
* The **email** should be **at least 10 characters** long
* The **password** should be **at least 4 characters** long
* The **repeat password** should be **equal to the password**



### Course

You should make the following validations while **creating** or **editing a** **course post**:

* The **Title** should be **at least 5 characters**
* The **Course Image** should start with **http:// or https://**
* The **Description** should be a minimum of **10 characters** **long**
* The **Type** should be a minimum of **3 characters long**
* The **Certificate** should be a minimum of **2 characters long**
* The **Price** should be a **positive number**

Картина, която съдържа екранна снимка, текст, софтуер, Мултимедиен софтуер

Описанието е генерирано автоматично

## \*Bonus – Profile

Each **logged-in** **user** should be able to view their profile information by clicking on [**Profile**]. On the profile page, the user should see their **email** and the **courses** they have **created**, as well as the courses they have **signed up for**.Under the **email**, the user has a **total count of the created courses** and the ones they have **sighed up** for**.**

Картина, която съдържа текст, екранна снимка, софтуер, Компютърна икона

Описанието е генерирано автоматично

If there are **no created courses** yet, the message **"You haven't created a course yet…"** should be visualized.

Картина, която съдържа текст, екранна снимка, софтуер, Шрифт

Описанието е генерирано автоматично

If the user **hasn’t signed up for any course** yet, the message **"You haven’t signed up for any course yet…"** should be visualized.

Картина, която съдържа текст, екранна снимка, софтуер, Компютърна икона

Описанието е генерирано автоматично

## Submitting Your Solution

Place a **ZIP** file in your project folder. Exclude the node\_modules folder. Upload the archive to Judge.









*GOOD LUCK! 😊*