# JS Back-End Final Exam – Job Ads

## 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 file **"index.js"** on port **3000**
11. It is **forbidden** to **use** **React**, **Vue**, **Angular,** etc.
12. **Only the last submission will be evaluated!** **Job Ads**

## Application Overview

Get acquainted with the provided HTML and CSS and create an application for **job ads**.

The visitors can view the **Home page** and **All ads page**, they can also **register** withan **email**, **password**, and **description of skills**, which will allow them to create their **job ad**, as well as **apply** if interested (but only if the **current user** is **not** the **author**). Authors can **edit** or **delete** ads at any time.

## Functional Requirements

The Functional Requirements describe the functionality that the **application** must support.

### Guest (not logged in)

**Guest** navigation example:



The **application** should provide Guest (not logged in) users with the functionality to register, **login**, **view** the **Home Page**, **All Ads page**, and the **Details** **page**.

### Users (logged in)

**User** navigation example:



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

* **View Home Page and all other pages with logged-in navigation**
* **View All ads page**
* **Create new ad [Create Ad]**
* **Access ad details page [Details]**
* **Applying for an ad** **(if the current user is not the author of the ad)**
* **Delete or Edit ad depending on user's authentication (only for the author of the current ad)**

## Database Models

The **Database** of the **Job Ads** application needs to support **2 entities.**

**Note**: **You can add additional properties to the models if this will help you solve the task**.

### User

* **Email** - string (required),
* Password - string (required),
* **Description of skills -** string (required),
* **My ads - a collection of Ads (a reference to the Ad Model)**

Note: When a user creates a new ad, a reference to that ad is added to that collection (My ads).

### Ad

* **Headline** - string (required),
* **Location - string** (required),
* **Company name** **- string** (required),
* **Company description - string** (required),
* **Author - object Id (a reference to the User model),**
* **Users applied** **- a collection of Users (a reference to the User model)**

Note: When a user applies for an ad, their id is added to this collection (Users applied).

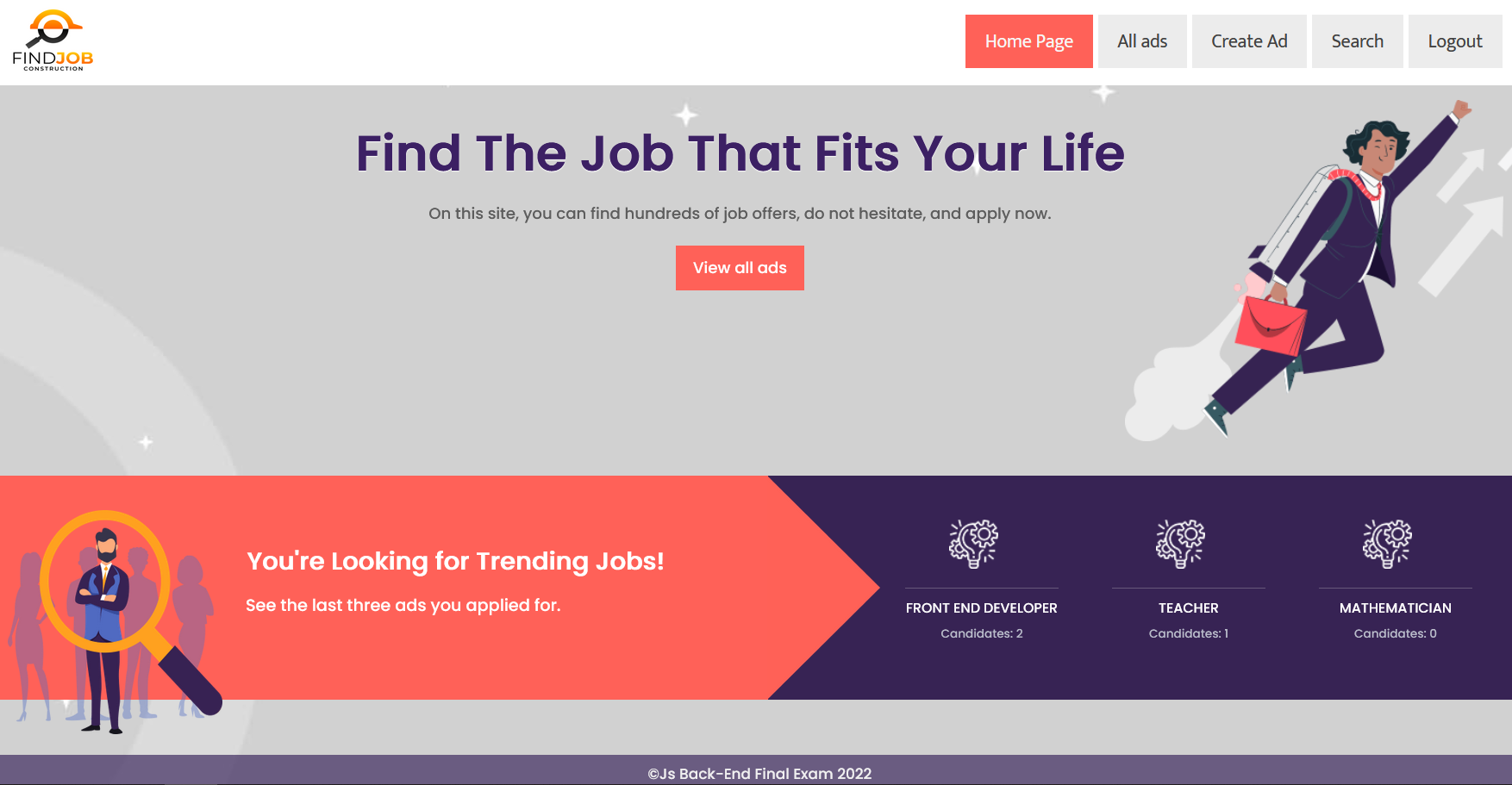
Implement objects with the correct data types.

## Application Pages (80 pts)

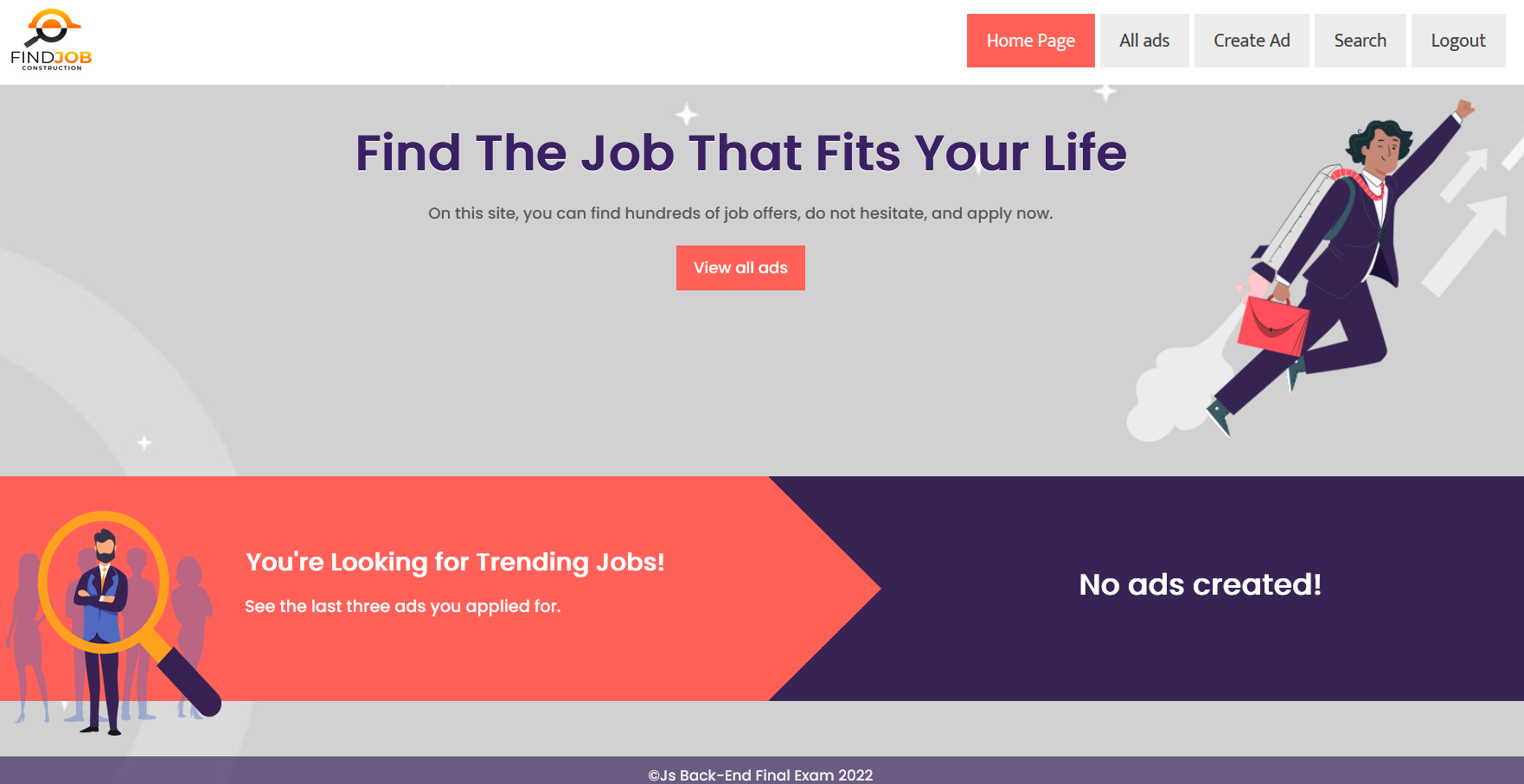
**Note:** Don't forget to change the paths to the CSS files, and the images.

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

Show the **first three added** job ads. Each **ad** must show information about the **headline**, and the **total number of candidates**.

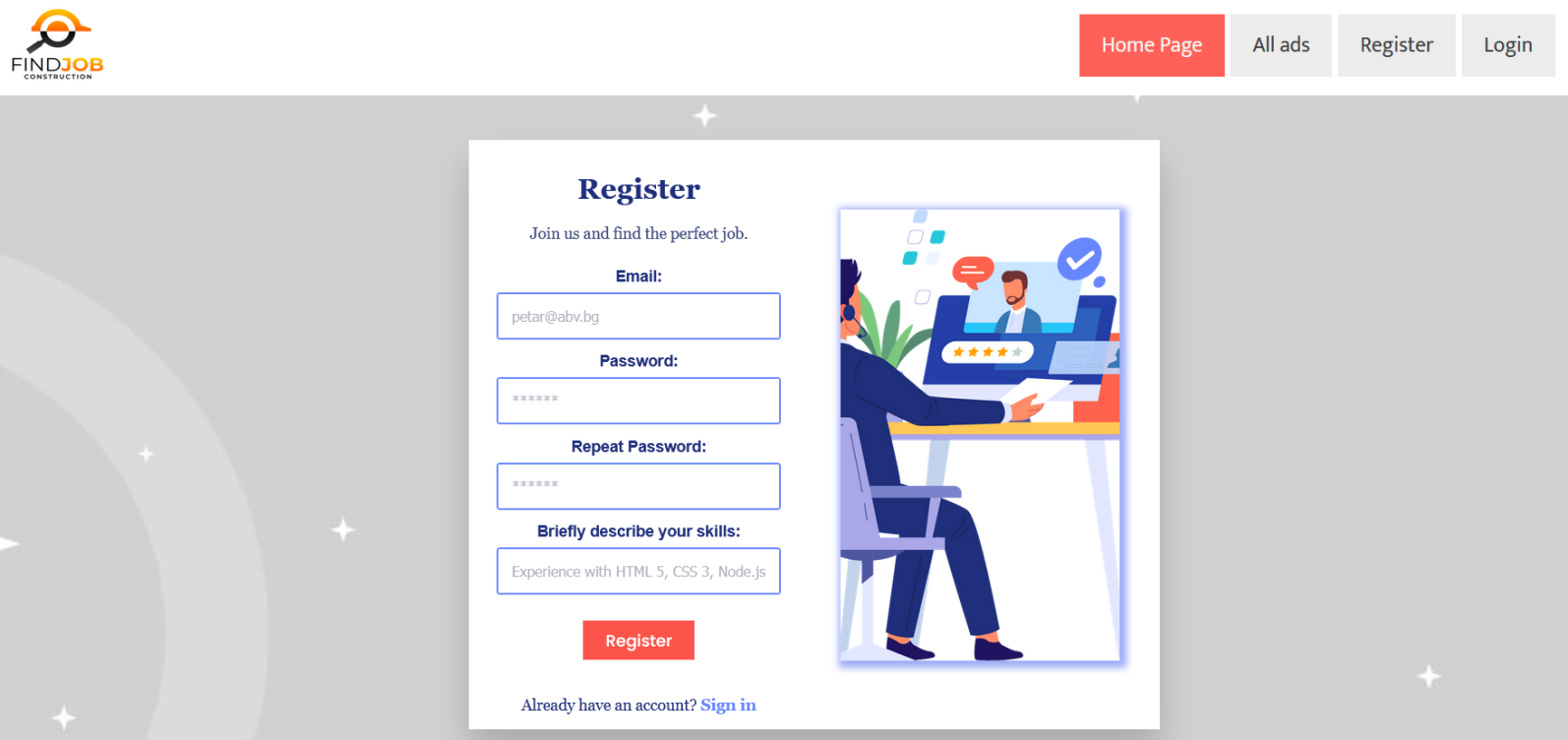


If there are still **no job ads**, display "**No ads created!**"



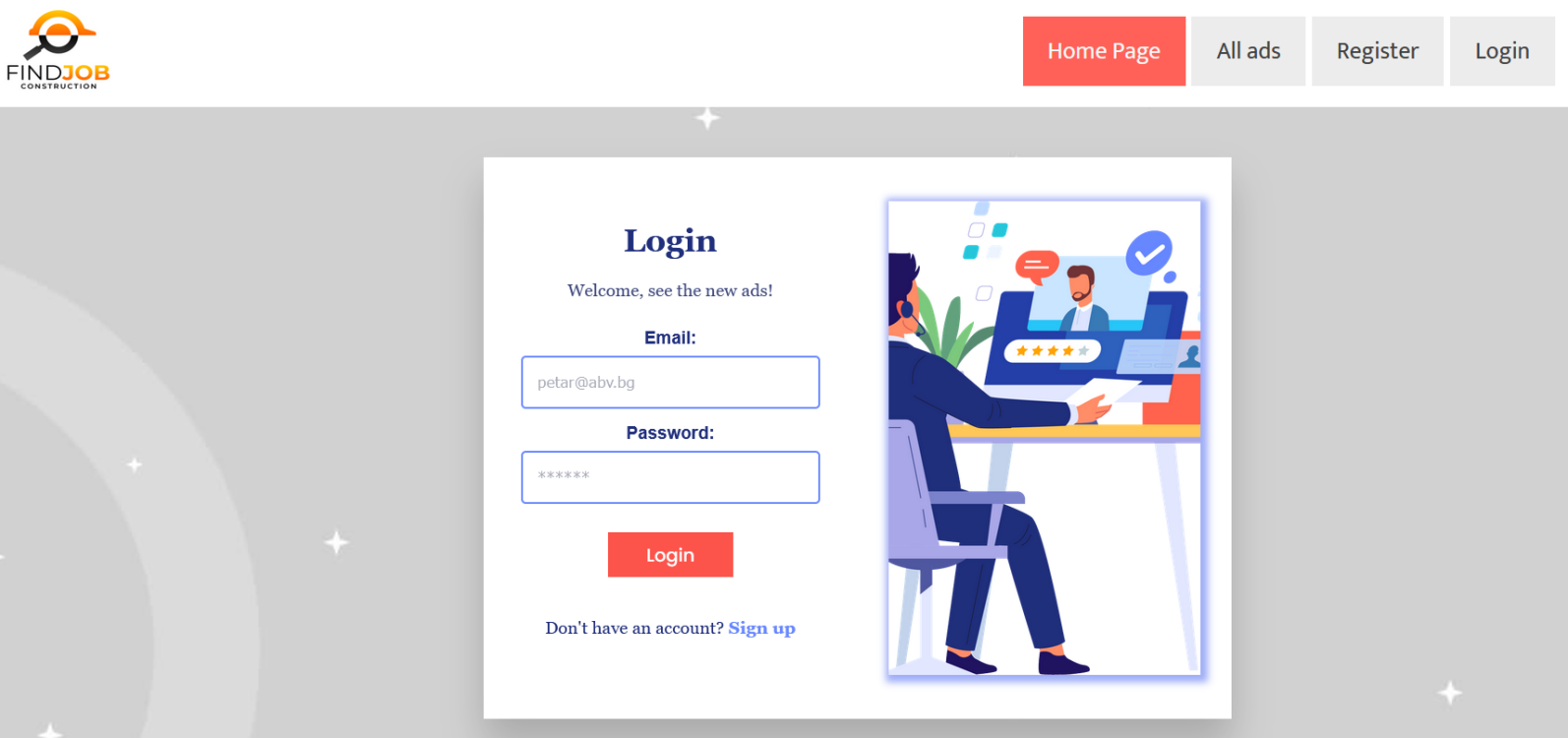
#### Register Page (logged out user)

Register a user inside the database with **email**, **password**, and **description of skills**. **Password** inside the database must be **hashed** (use bcrypt) and both **passwords** must **match**! After successful registration, **redirect to the Home page as an already logged-in user.**



#### Login Page (logged out user)

Logging an already registered user with the correct **email** and **password**. After successful login, **redirect to the Home page as an already logged-in user.**

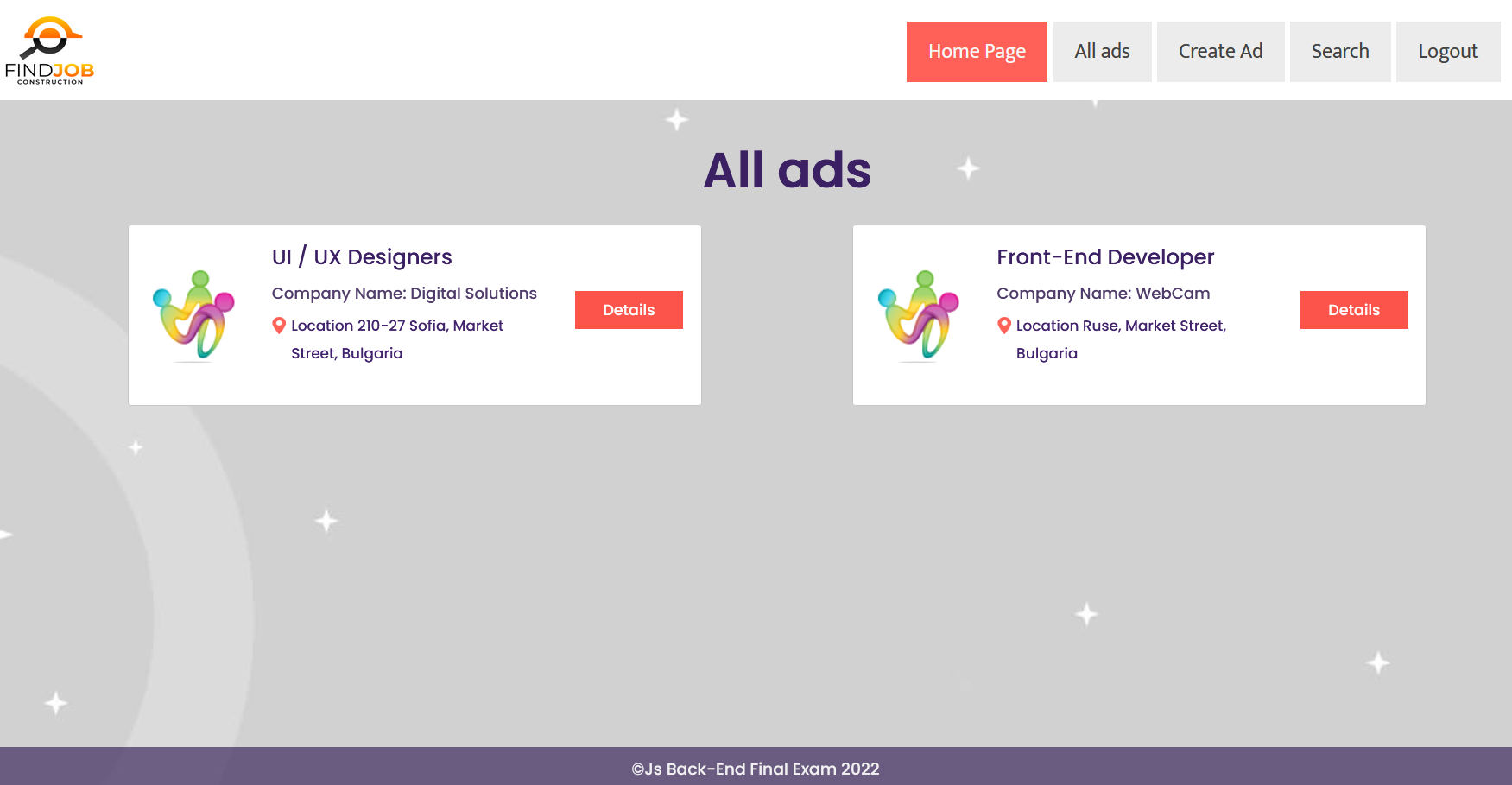


#### Logout (logged in user)

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

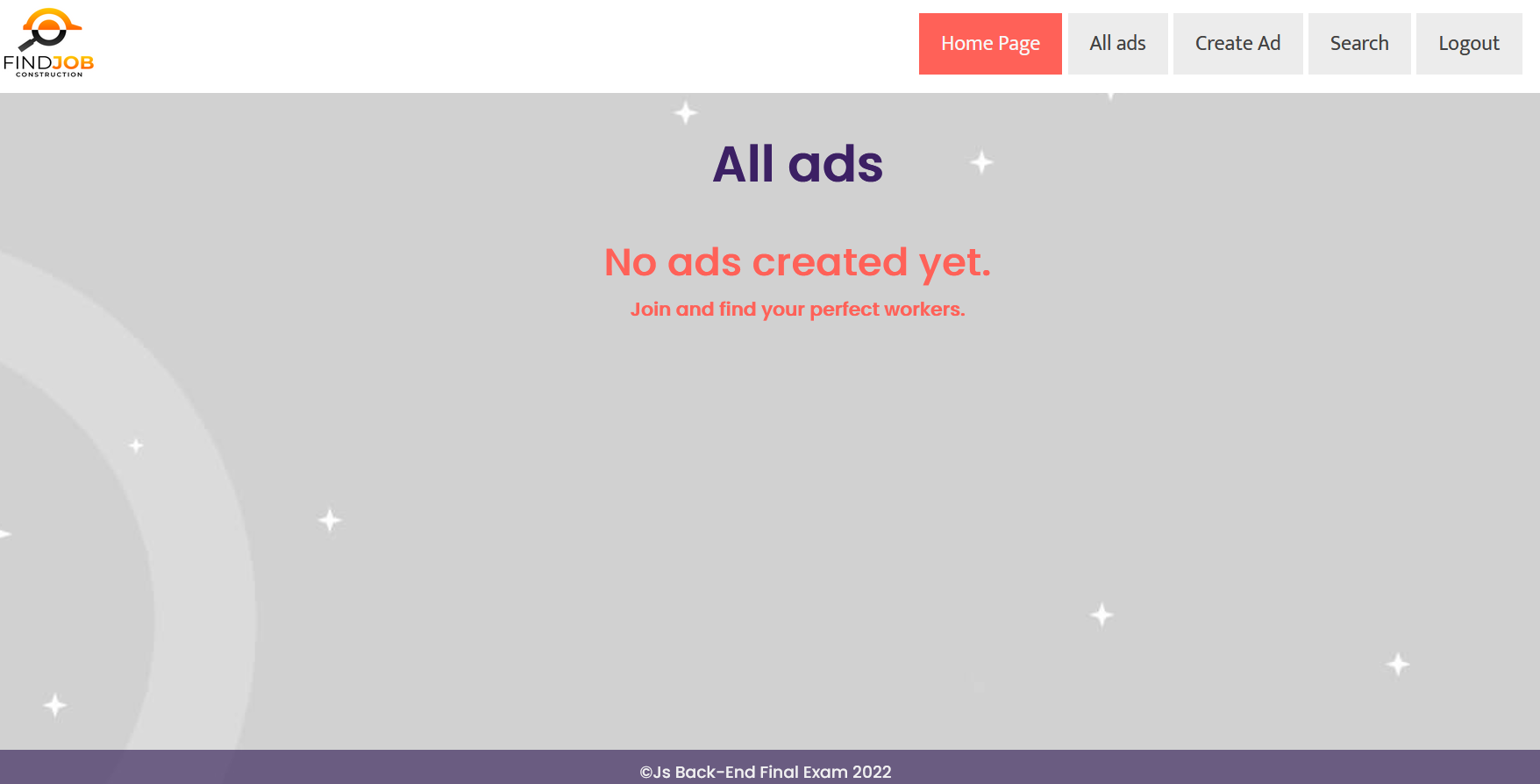
#### All ads (for logged in users and logged out users)

List of all **job ads**. **Each** **ad** should display information about the **headline**, the **company name**, the **location**, as well as the [**Details**] button that leads to the details page for the specific ad. Like in the picture below:



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

If **there are** **no job ads** in the **database**, display the following view:



**Details Page - (for logged in users)**

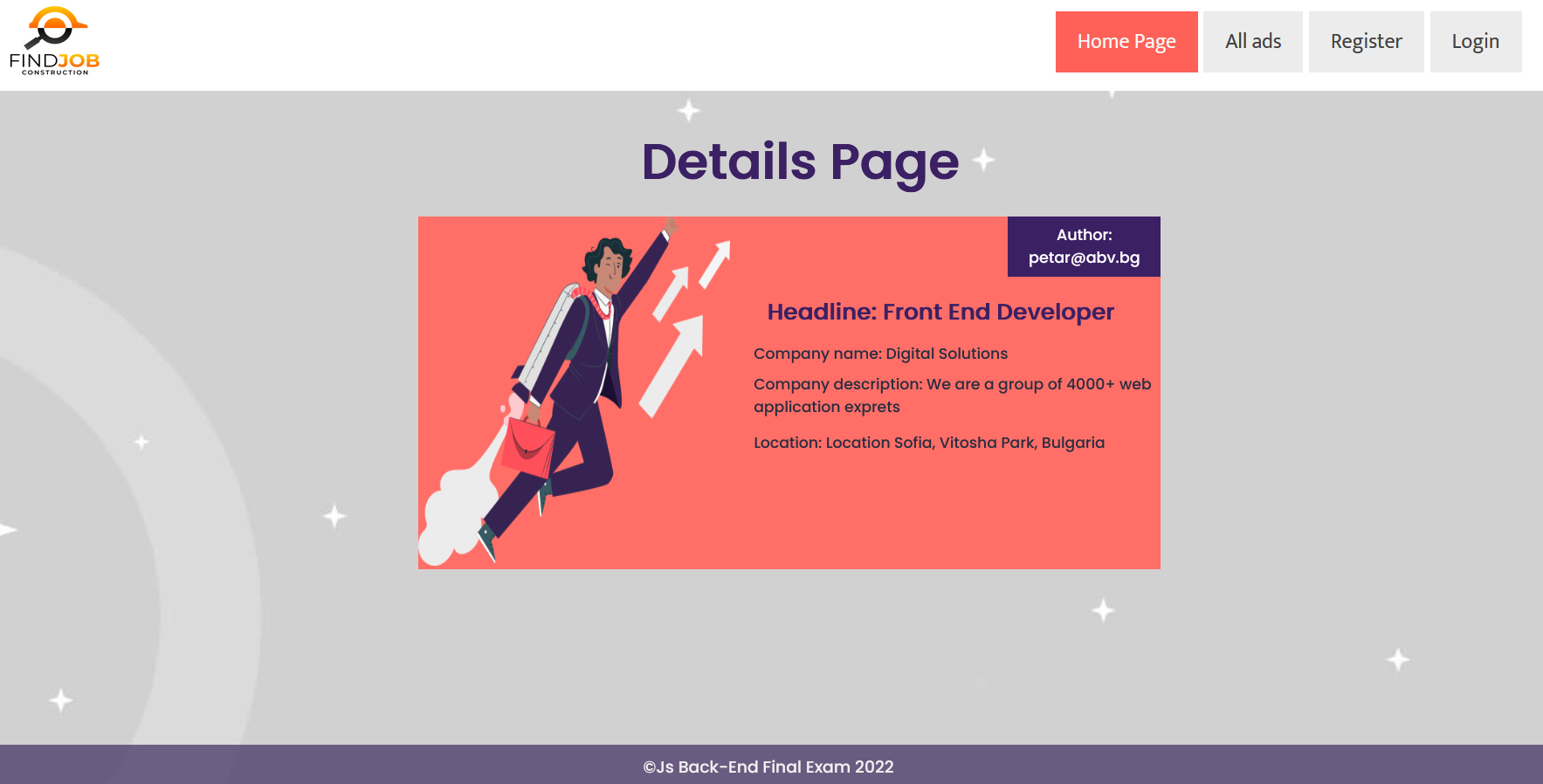
All users should be able to **view details** about thead. Clicking the **Details** button on the **ad card** should **display** the **Details** page. If the currently **logged-in user is the author** of the ad, the **Edit** and **Delete** buttonsshould be displayed, otherwise they should **not** be **available**.

Information about the ad:

* **Author**
* **Headline**
* **Company name**
* **Company description**
* **Location**
* **Buttons** (Depending on the status of the currently logged in user)

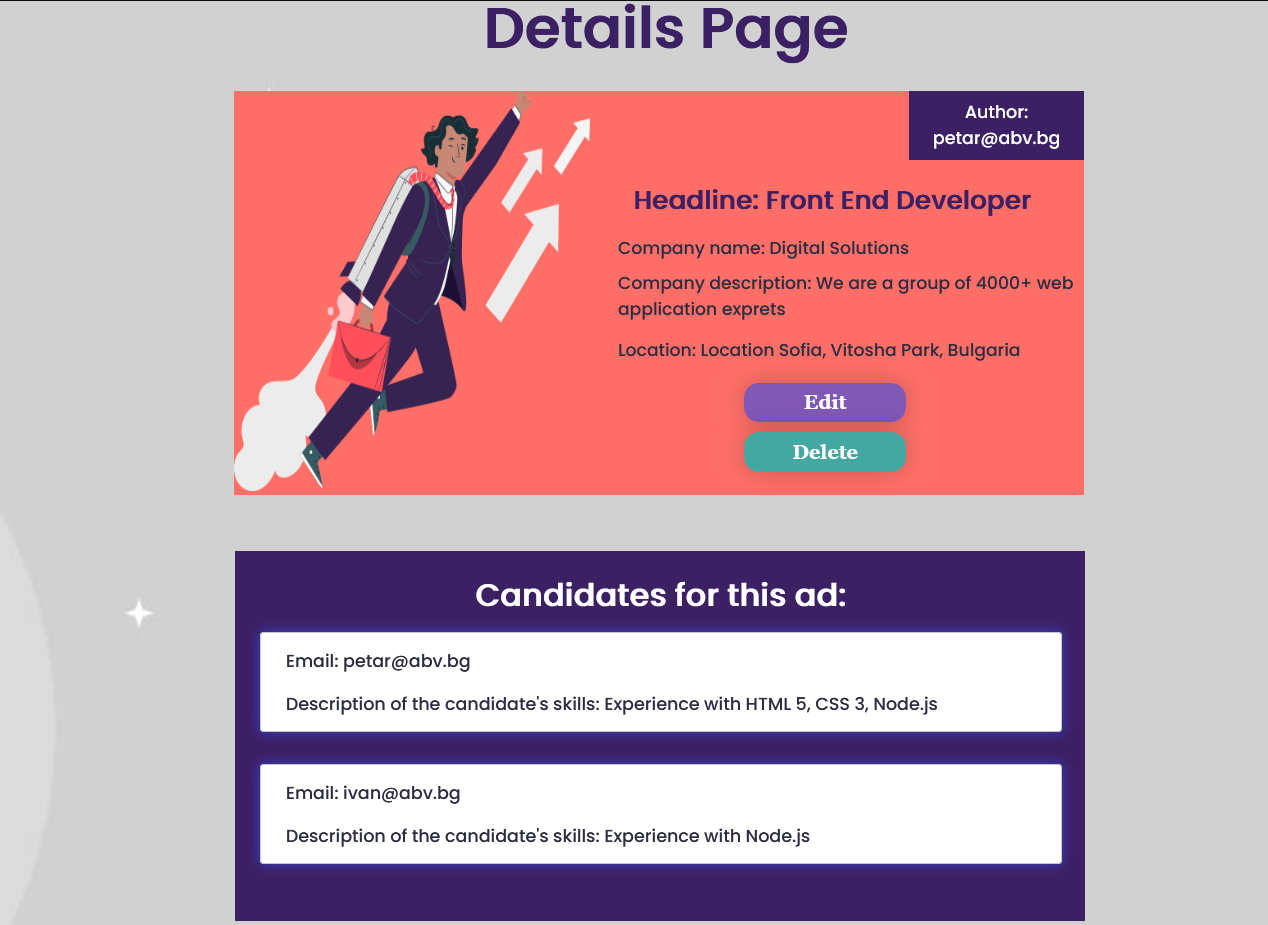
#### Details Page (logged out users)

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



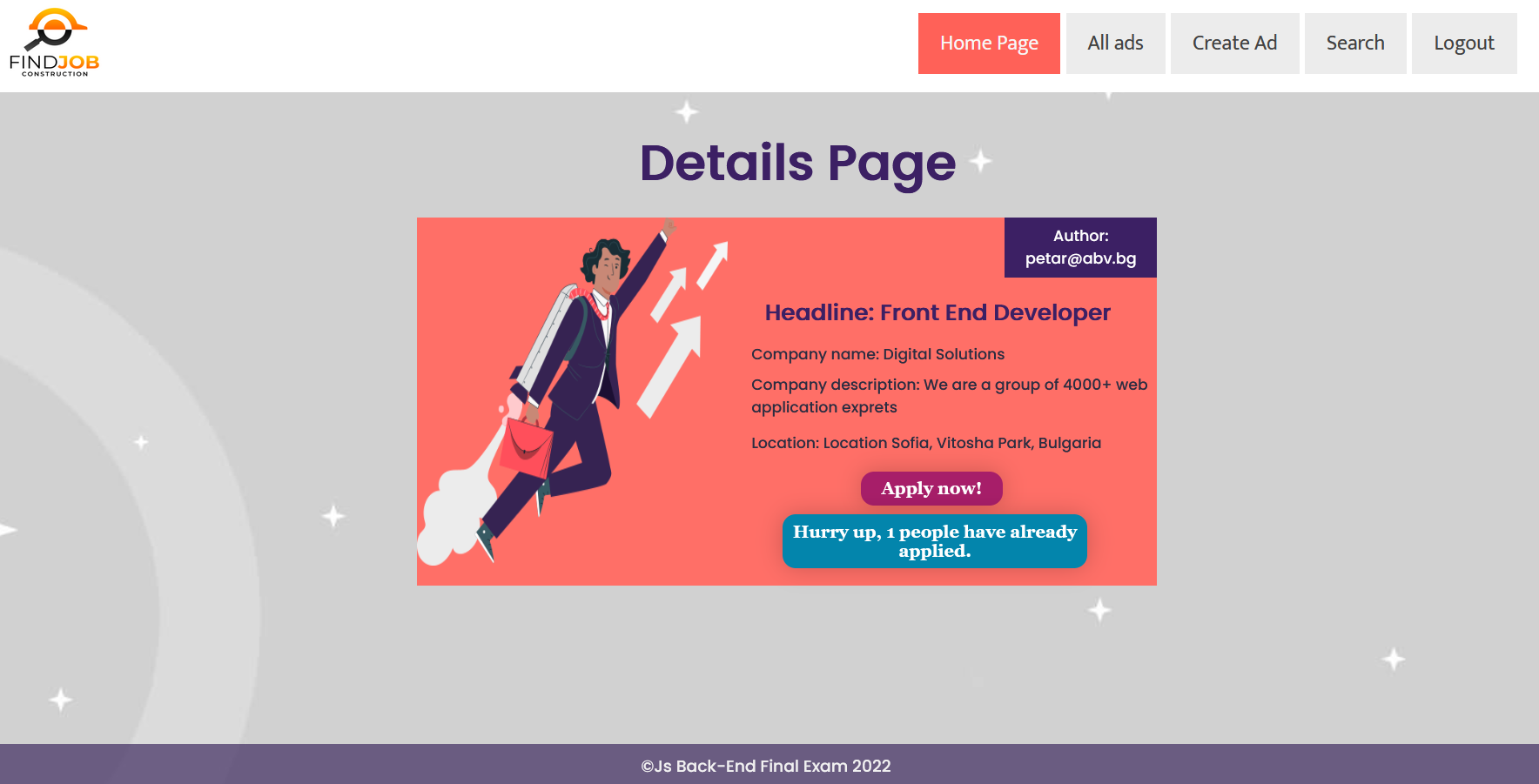
#### Details Page (logged in user and author of the current ad)

If the **currently logged-in** user is the **author** (the user who created the **job ad**), he should see the **[Edit]** and **[Delete]** buttons. Also, **only the author** has the right to **see the information about all candidates (**those who have applied for a current ad**)**. For **each candidate**, show their **email** and **description of their skills**. If there is **no candidats**, you **don't** neet to visualize anything further.



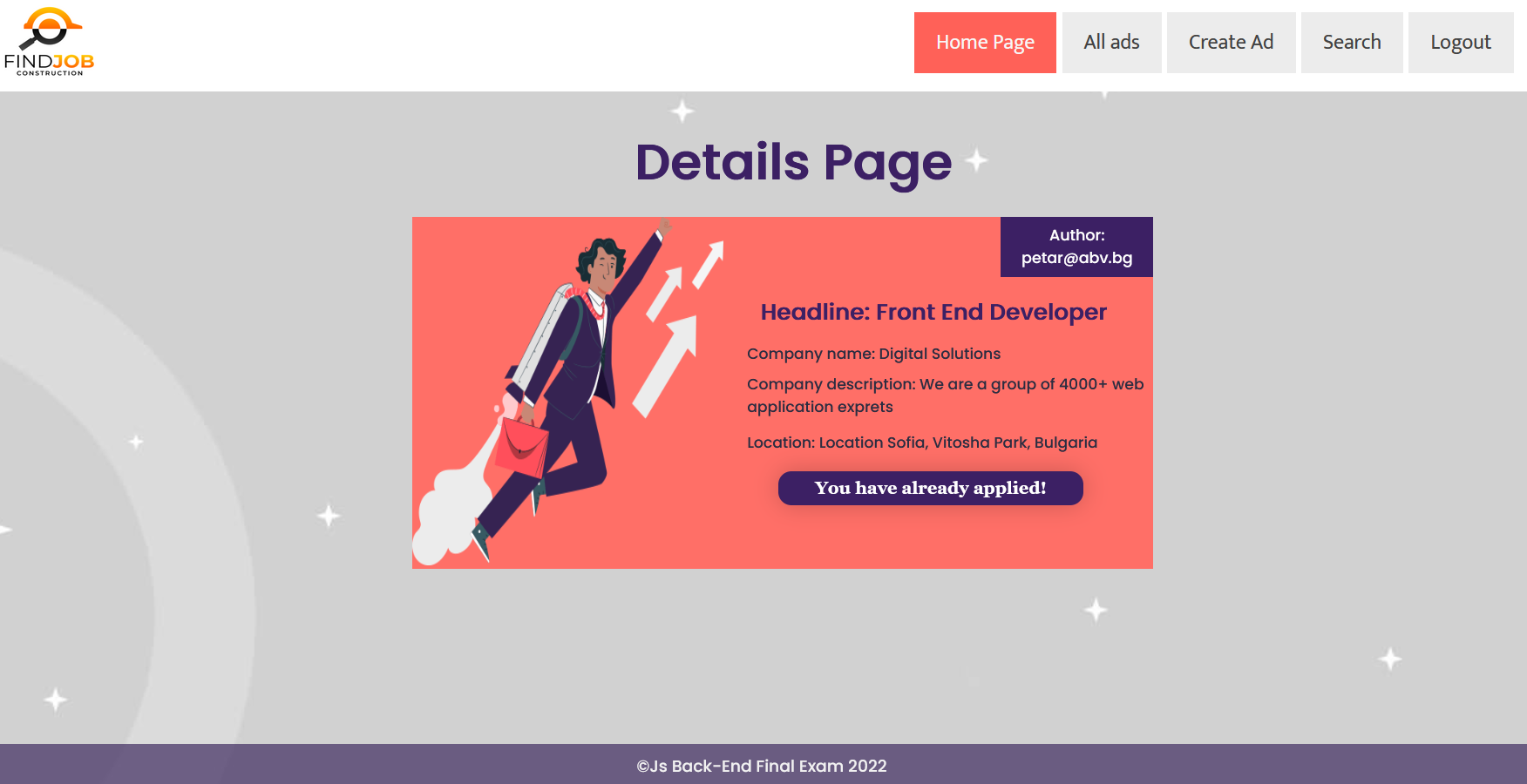
#### Details Page (logged in user who did not apply for a current ad)

If the currently logged-in user is **not the author** (the user that is not the creator) and has not applied for this ad, he should see the **[Apply now!]** button and paragraph **[Hurry up, {total number of candidates} people have already applied.]**.

****

#### Details Page (logged in and has already applied for a current ad)

If the currently logged-in user is not **the author** and has **already applied for** a current ad, he should see the paragraph [**You have already applied!**].



#### Apply for a job ad (logged in user who is not the author of the ad)

Any **registered user** who **is not the author** of the job ad must be **able to apply** (if they have **not already applied**).

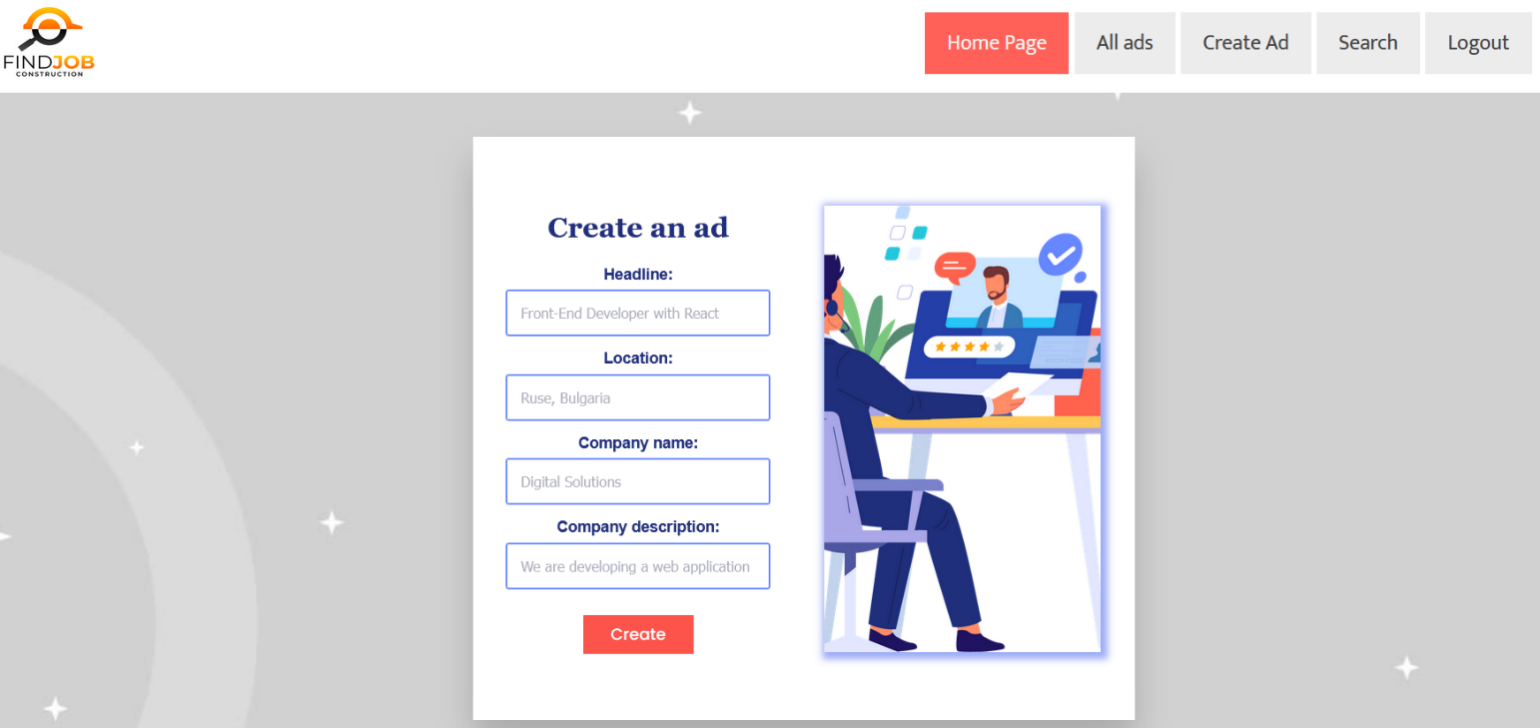
If the user applies successfully, the **userId** of the user must be added to the collection of **Users applied**. Then **redirect** the user to the **Details** page for the current **job ad.**

If a user has applied for a current job ad, he should see the paragraph "**You have already applied!**".

If the user has successfully applied, do not forget to **update** the value **total number of candidates** (**inside the span tag**) on the **Home page (**as you will need to visualize the three first added ads**)**.

#### Create Ad Page (logged in user)

The Create page is available to **logged-in users**. It contains a form for adding a new **job ad**. Upon success, **redirect** the user to the **All ads** **page**.

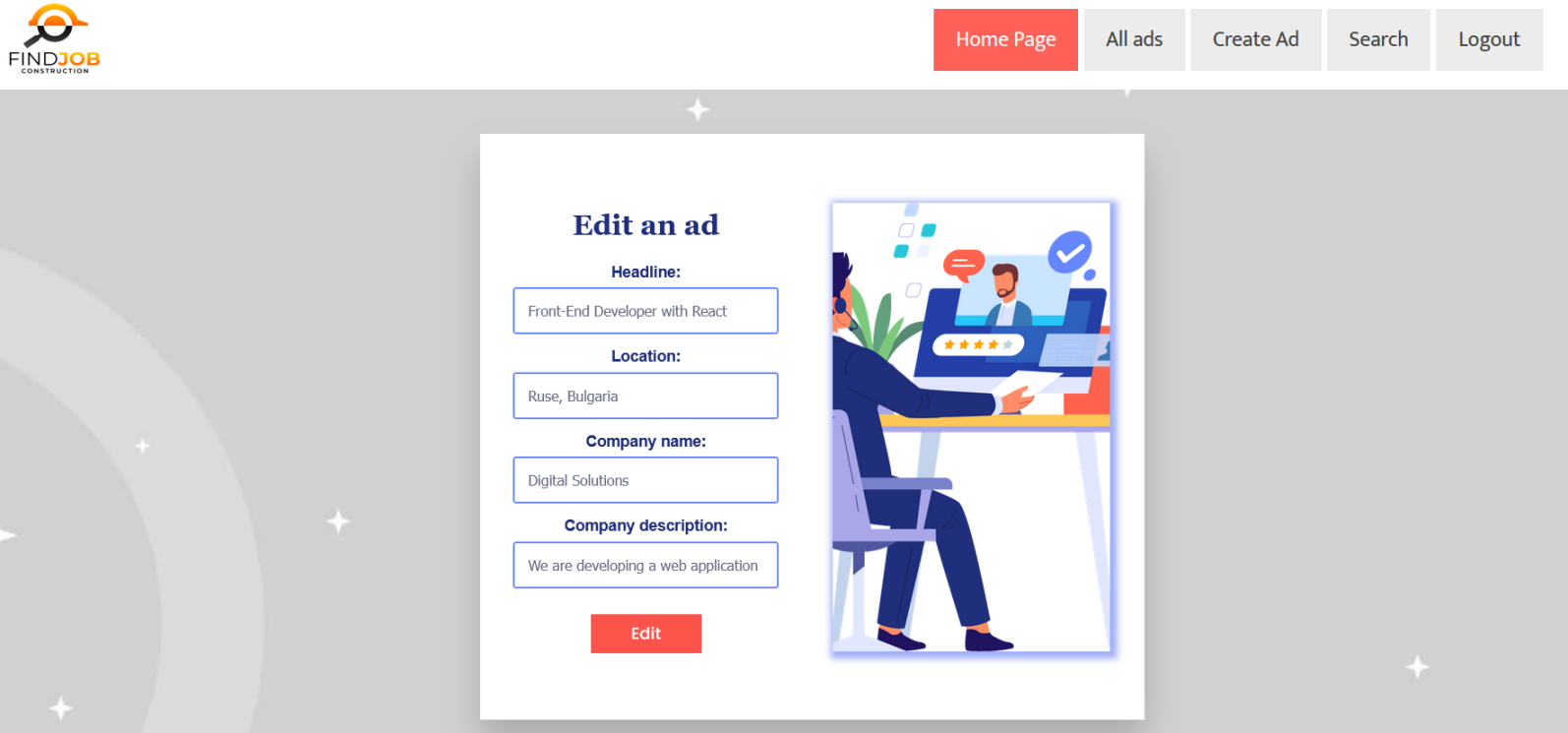


#### Delete Ad (logged in user and author of the current ad)

**Every author** should be able to click over the [**Delete**] **button** - deleting the current ad from the **database** and the user should be **redirected** to the **All ads page.**

#### Edit Ad (logged in user and author of the current ad)

The **Edit** **page** is available to logged-in users and it allows authors to **edit** their ads. Clicking the [**Edit**] **button** of a particular ad on the **Details** **page** should display the **Edit** **page**, with **all fields filled** with the data for the ad. It contains a form with input fields for all relevant properties. Upon success, **redirect** the user to the **Details** **page** for the current ad.



## Security Requirements (Routes Guards) - (10 pts)

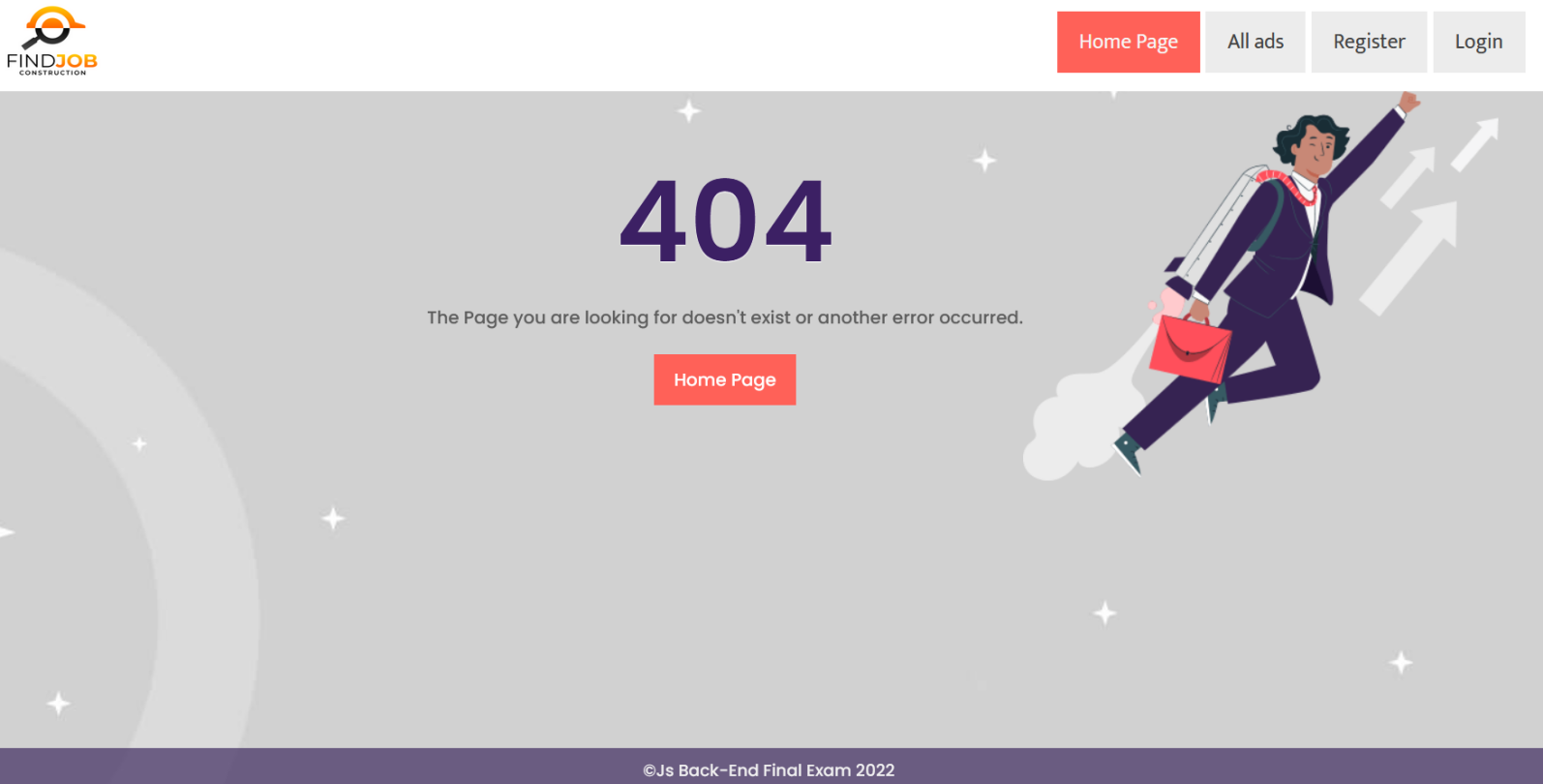
The Security Requirements are mainly **access** requirements. Configurations about which users **can access** specific functionalities and pages.

* 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 ads page.
* Guest (not logged in) can access the **Details page** without functionality.
* Users (logged in) can access **Home Page**.
* Users (logged in) can access the **Details page** and functionality.
* Users (logged in) can access **Create Ad** **page** and functionality.
* Users (logged in and not authors of the current ad) can access Apply for ad functionality.
* Users (logged in and author of the current ad can access the Delete and Edit ad functionaity
* Users (logged in) can access Logout functionality.

If Guests (not logged in) try to **access** а page they should **not be able to**, you must **redirect** them to the **Login** page.

If **Users** (logged in) try to **access** а page they should **not be able to**, you must **redirect** them to the **Home Page**.

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



## Validation and Error Handling (10 pts)

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

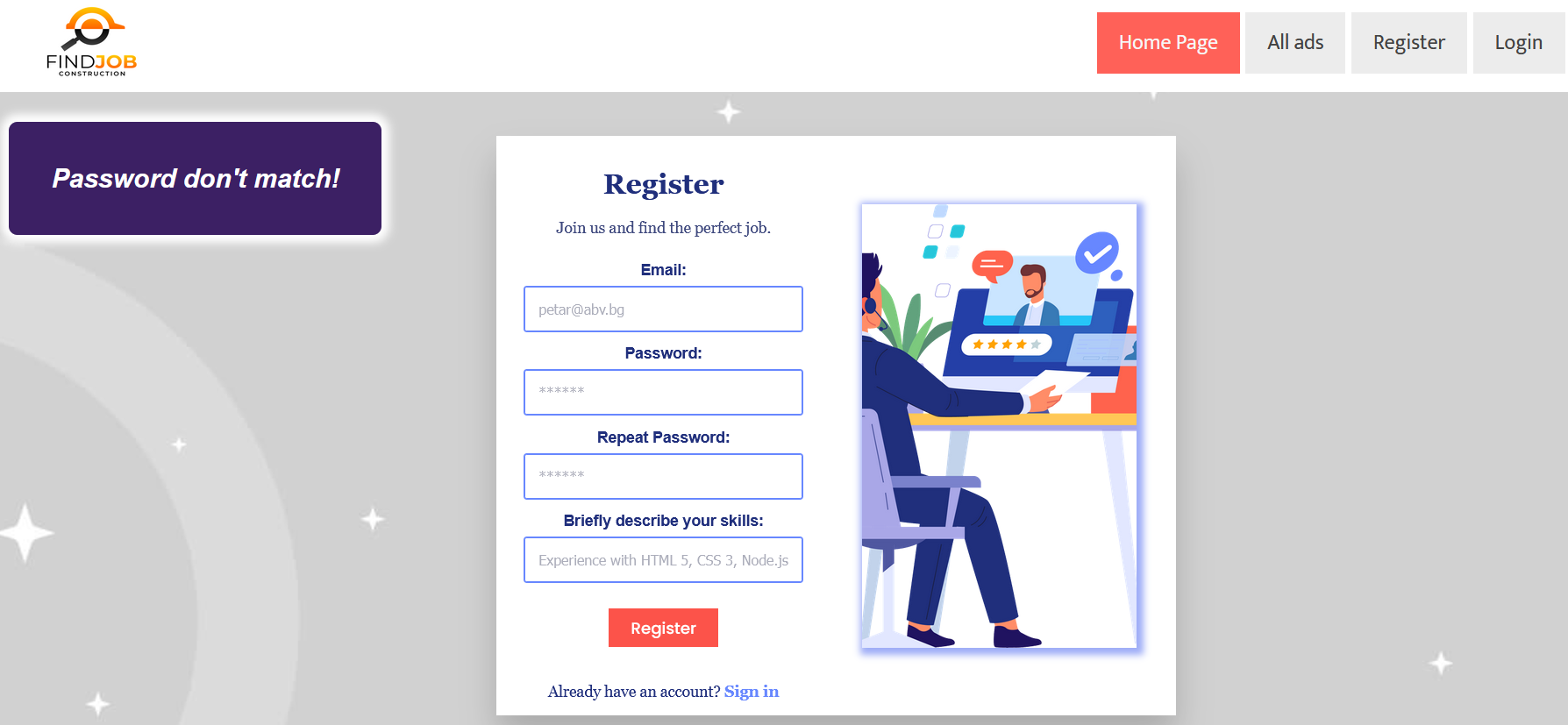
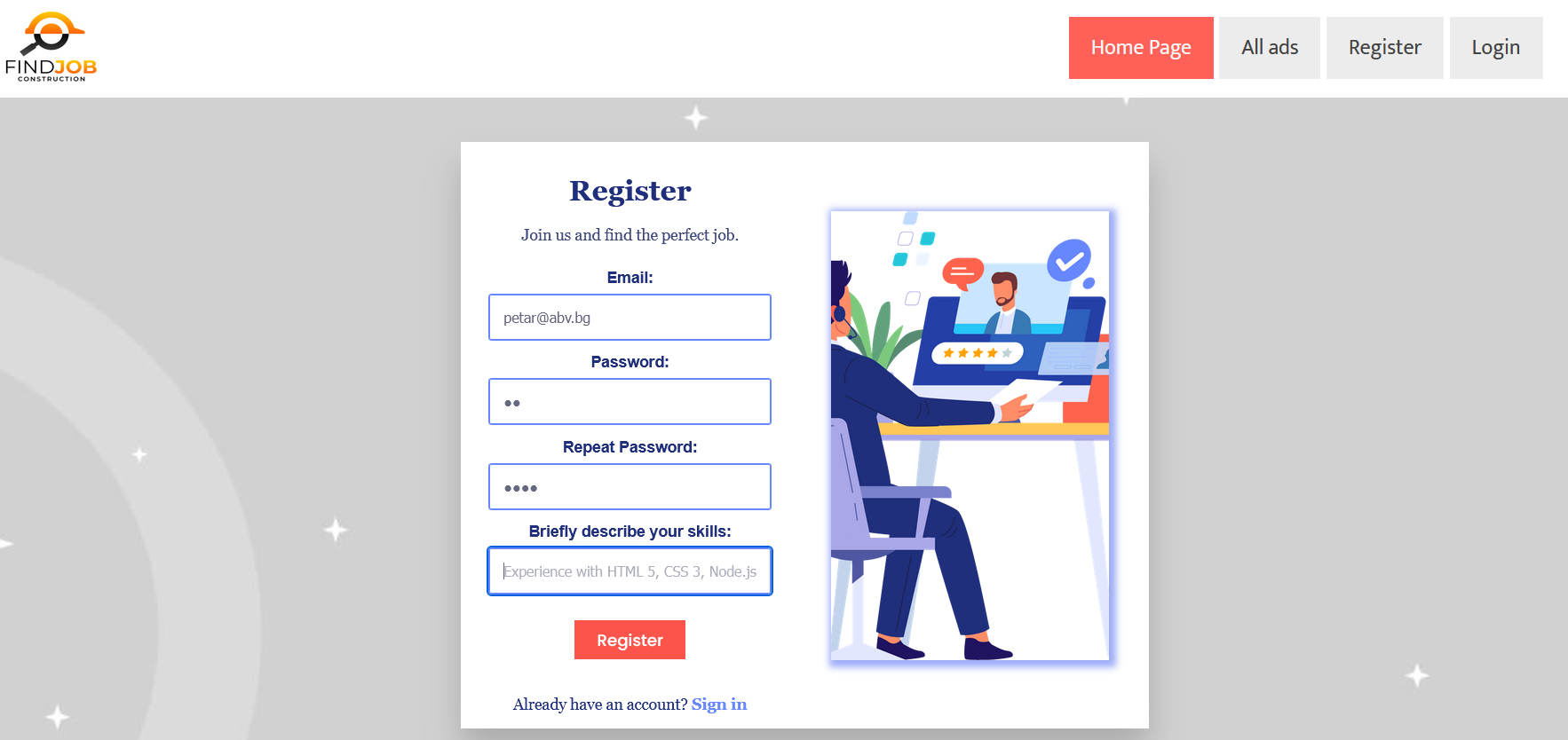
In case of error, you **should display div** with class **"error-box"**.

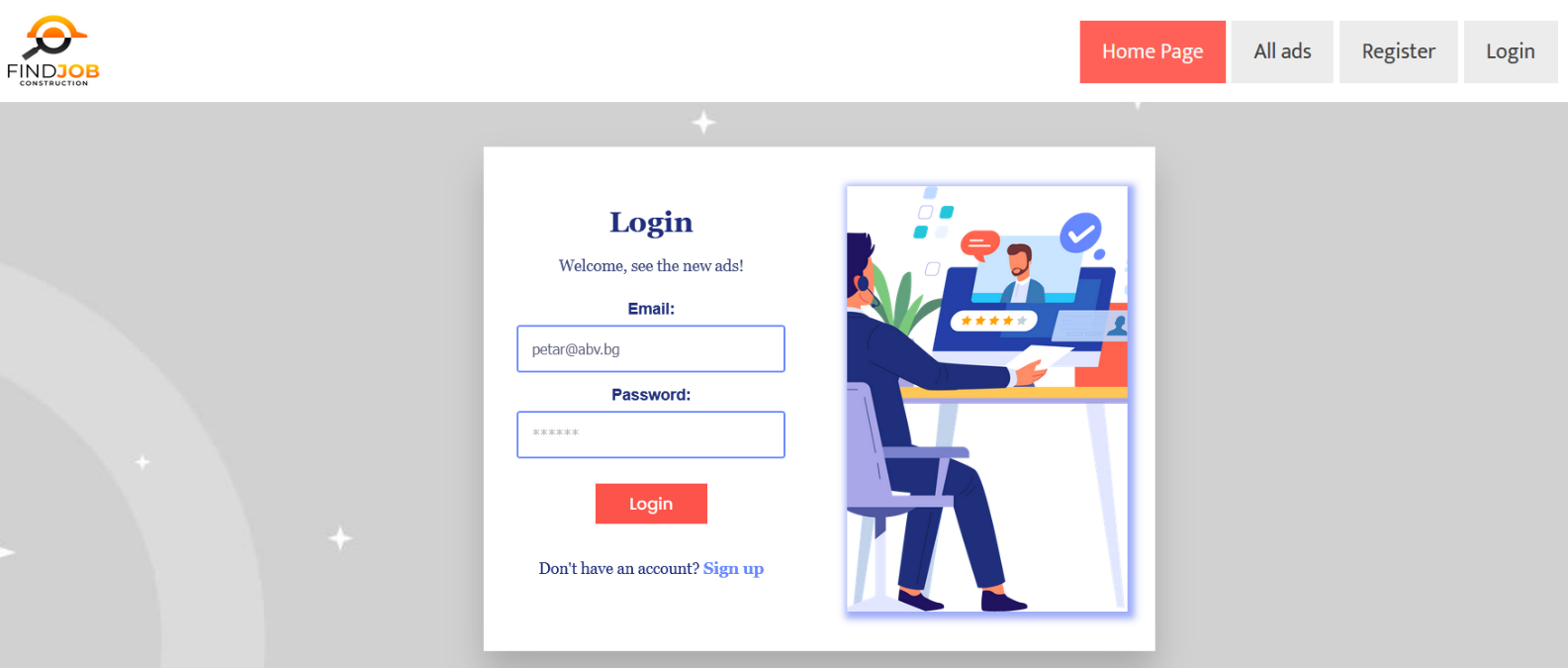
**You can choose to display the first error or all of them. You have complete freedom to choose the content of the error message you will display.**

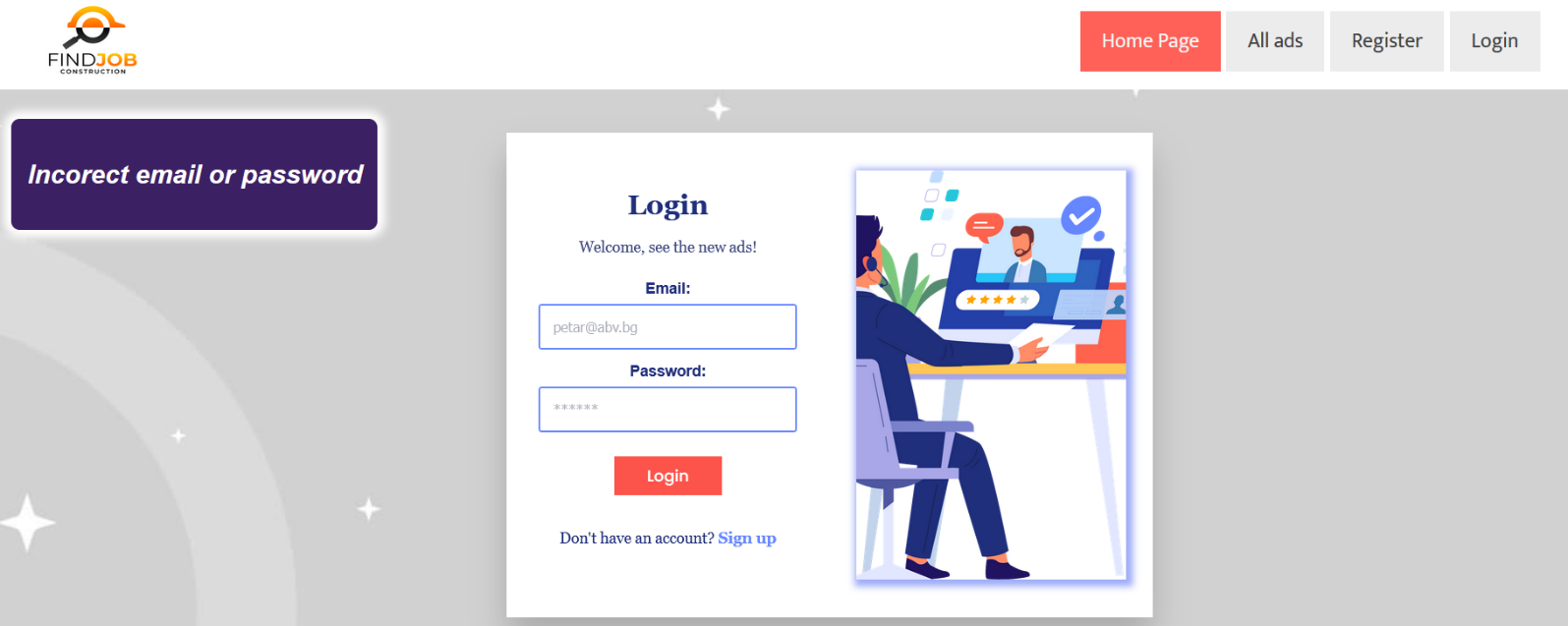
### Login / Register

You should make the following validations:

* The **email** should be **in the following format: <name>@<domain>.<extension>** 
  + Only English letters are allowed for any of the parts of the email
  + There must be a **@** after the **<name>**
  + There must be a **point(.)** after the **<domain>**
  + Example of a valid email - "**petar@softuni.bg**"
* The **password** should be **at least 5 characters** long
* The **repeat password** should be **equal to the password**
* The **description of skills** should be a maximum of **40 characters** long



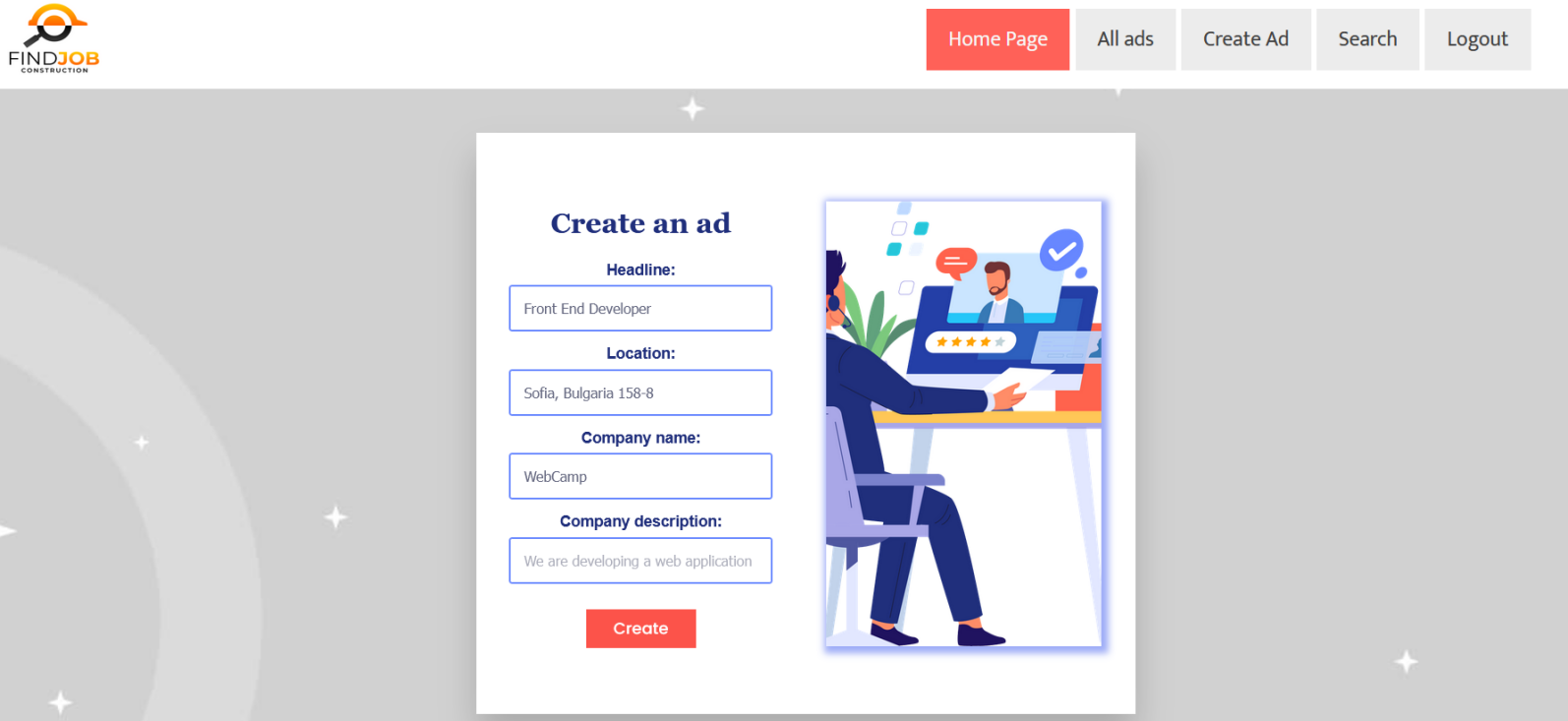


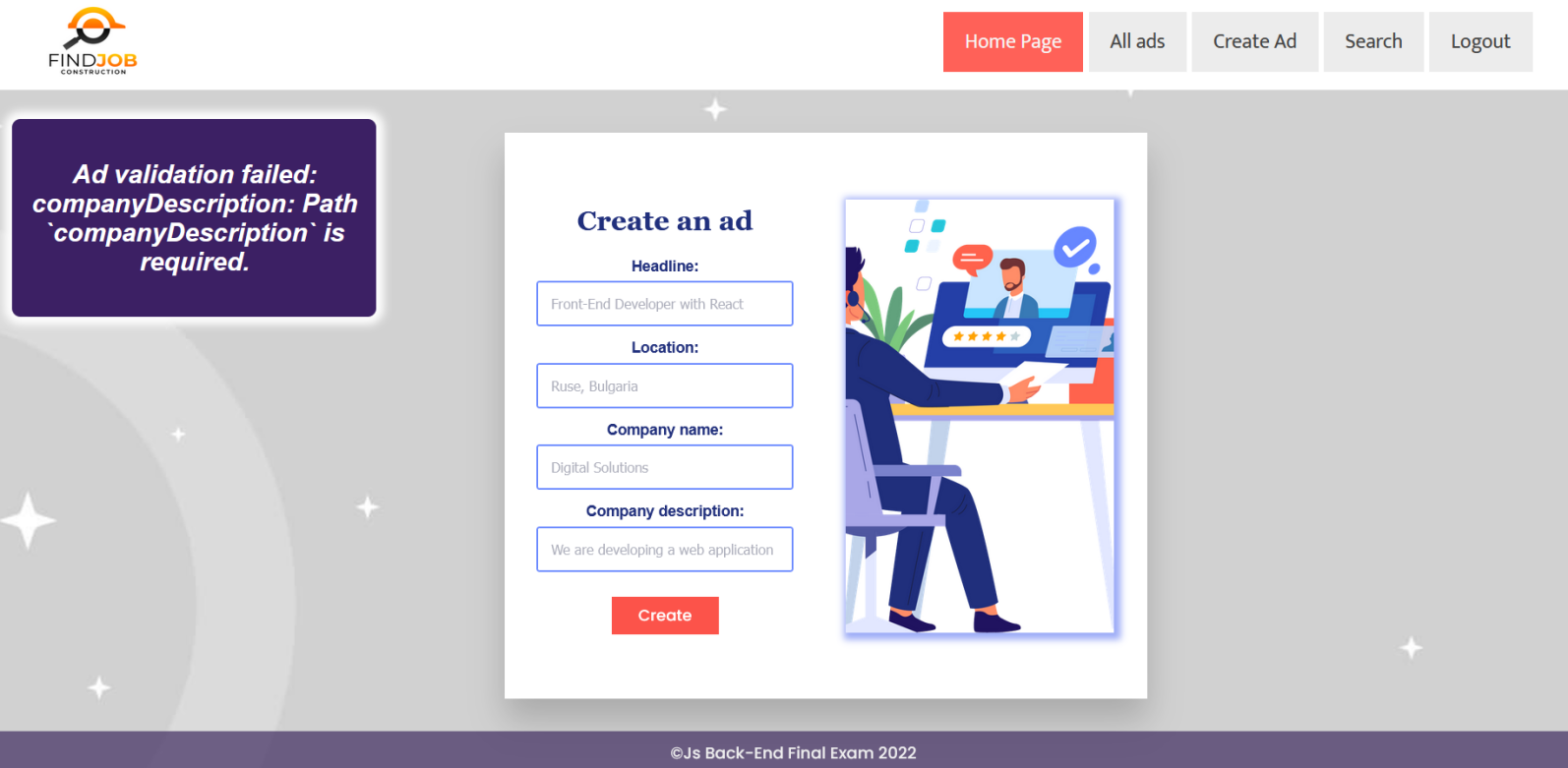


### Ad

You should make the following validations while **creating and editing ad**:

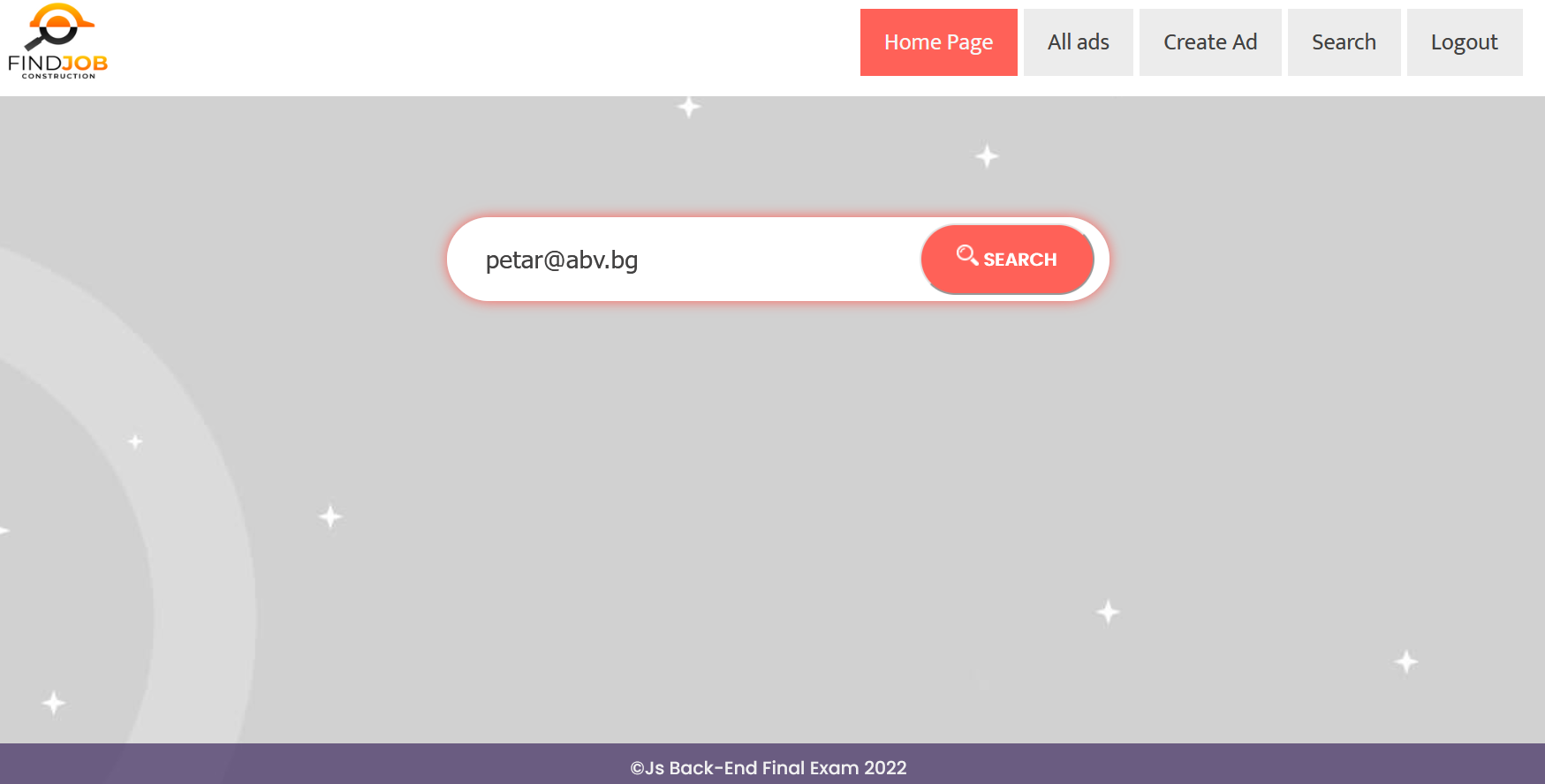
* The **Headline** should be a minimum of **4 characters** long
* The **Location** should be a minimum of **8 characters** long
* The **Company name** should be at least **3 characters**
* The **Company description** should be a maximum of **40 characters** long





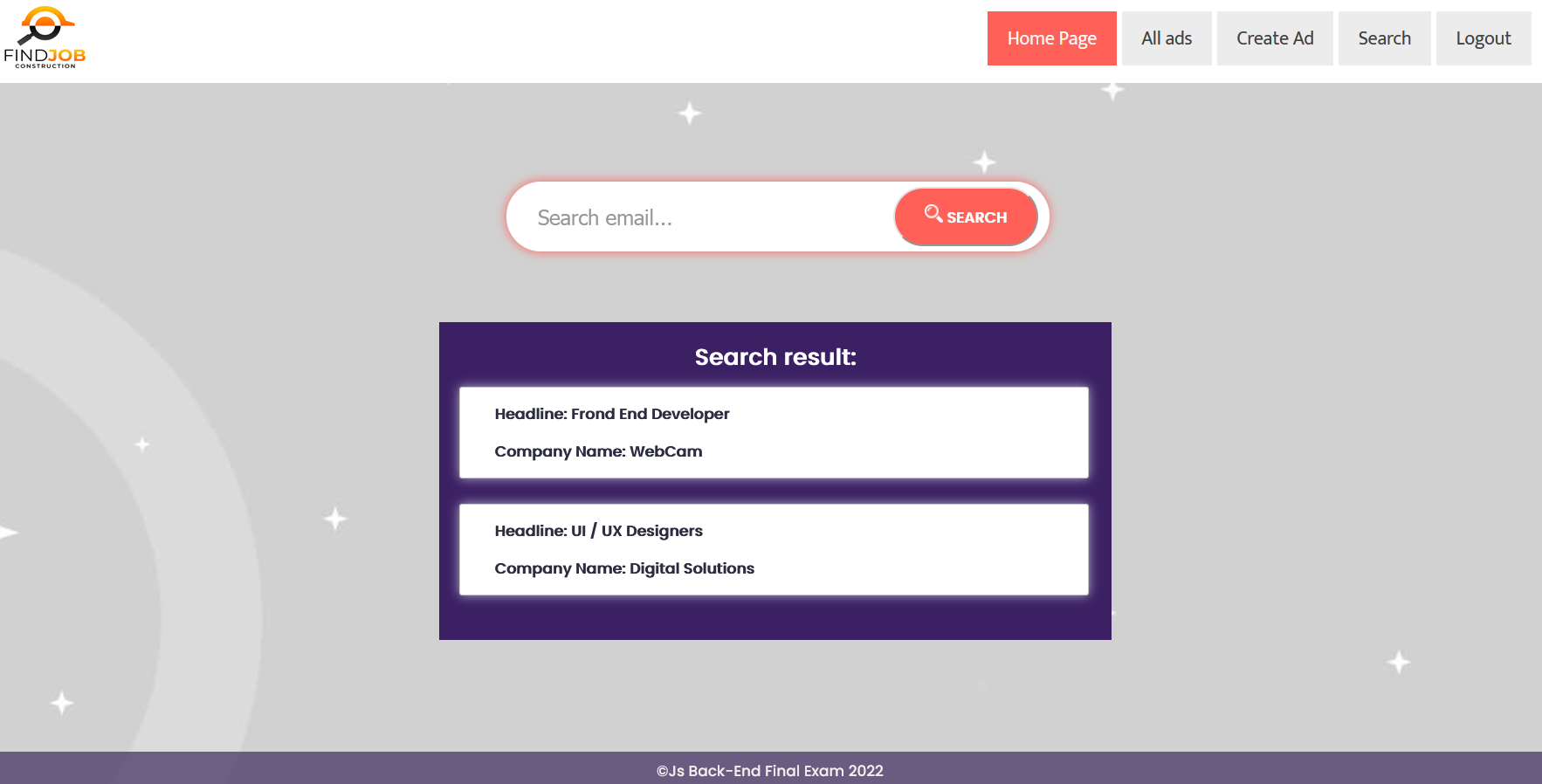
## Bonus - Search (10 pts)

Search by **email**. Filter **all ads** where the **author's email** matches a **search string** (regardless of uppercase).

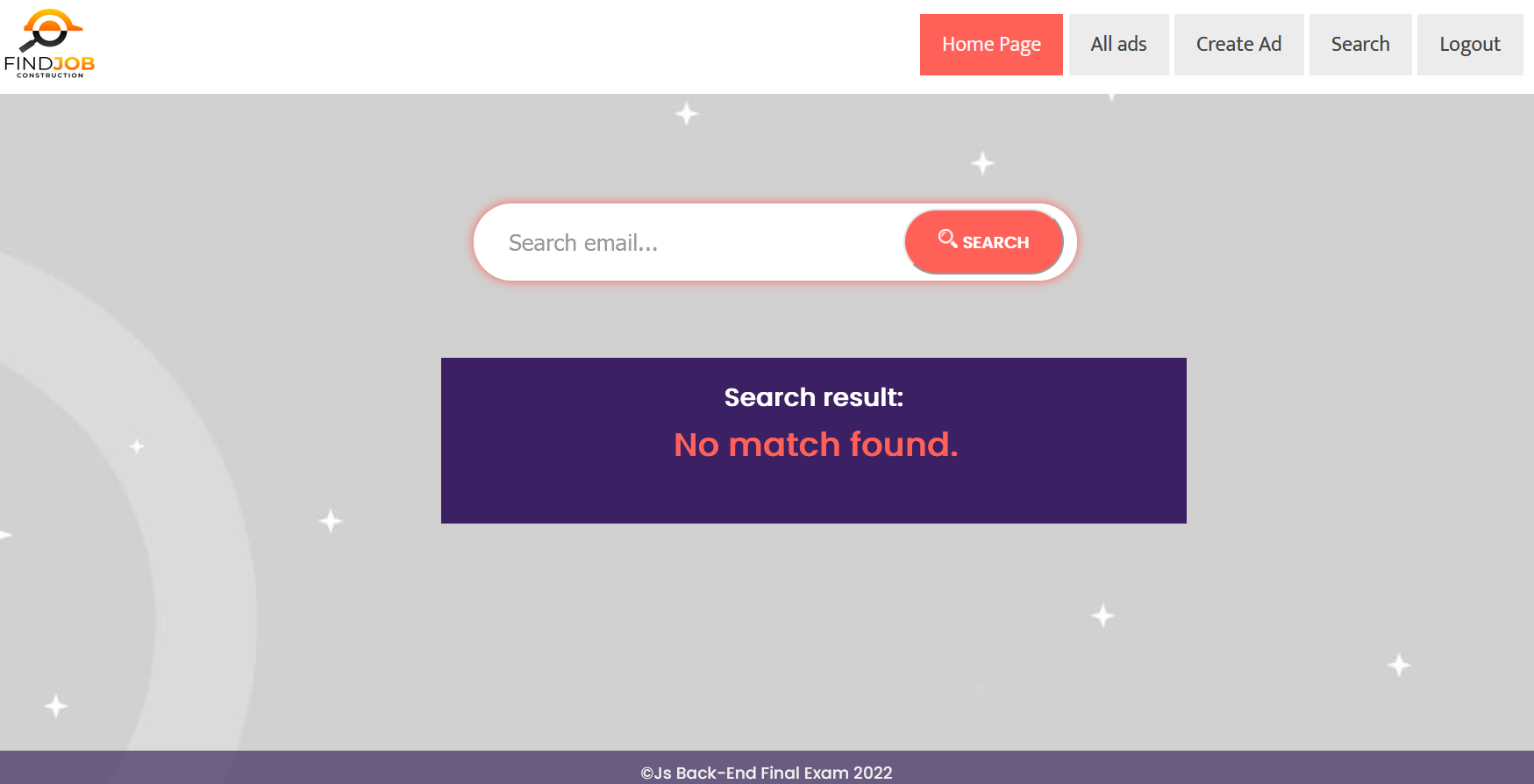


After clicking on the **[Search]** button, if there are matches from the search, display **each matching ad**.

**Each** **ad** should display information about the **headline**, the **company name**.

****

If there are **no** search matches, display:



## Submitting Your Solution

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









*GOOD LUCK! 😊*