## JS Back-End Exam – Mind Blog

## 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. **Only the last submission will be evaluated!**

## Application Overview

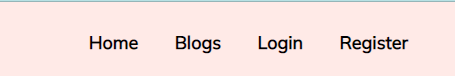
Get acquainted with the provided **HTML and CSS** and create an application for the **blog posts**.

The visitors can **view** the **Home page** and **Catalog** page with available blogs. They can also **register** with an **email**, **username**, and **password**, allowing them to create their blogs and **follow blogs they are interested in** (if the **current user** is **not** the author of the blog). Authors can edit or delete posts at any time.

## Functional Requirements

### Guest (not logged in)

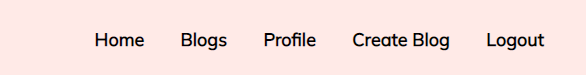
**Guest** navigation example:



The **application** should provide Guest (not logged in) users with the functionality to Login, Register, and **view** the Home page, **Blogs** 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 Blogs page**
* **Create а new Blog Post [Create Blog]**
* **Access blog details page [Details]**
* **Follow blog post (**if the **current user** is **not** the **owner of the blog)**
* **Delete or Edit the blog depending on the user's authentication (only for the owner of the current blog)**

## Database Models

The **Database** of the **Mind Blog** application needs to support **two entities**:

### User

* Username - string (required),
* Email - string (required),
* Password - string (required)

### Blog

* **Title - String (required),**
* **Image: String (required),**
* **Content: String (required),**
* **Blog Category: String (required),**
* **FollowList - a collection of Users (a reference to the User model)**
* **Owner - object Id (a reference to the User model)**

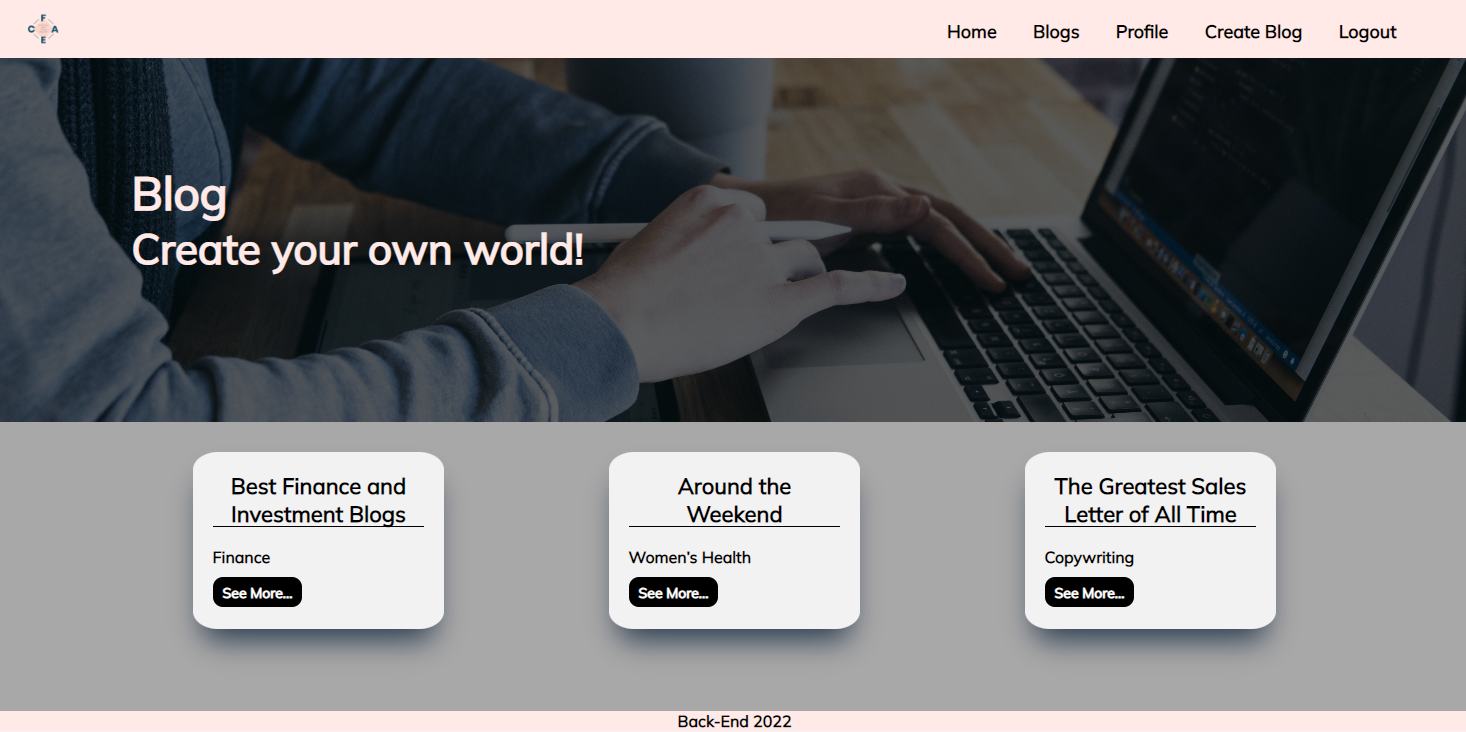
**Note:** When a user **follows blogs**, their **id** is added to that collection (**Follow blogs**)

Implement the entities with the correct data types.

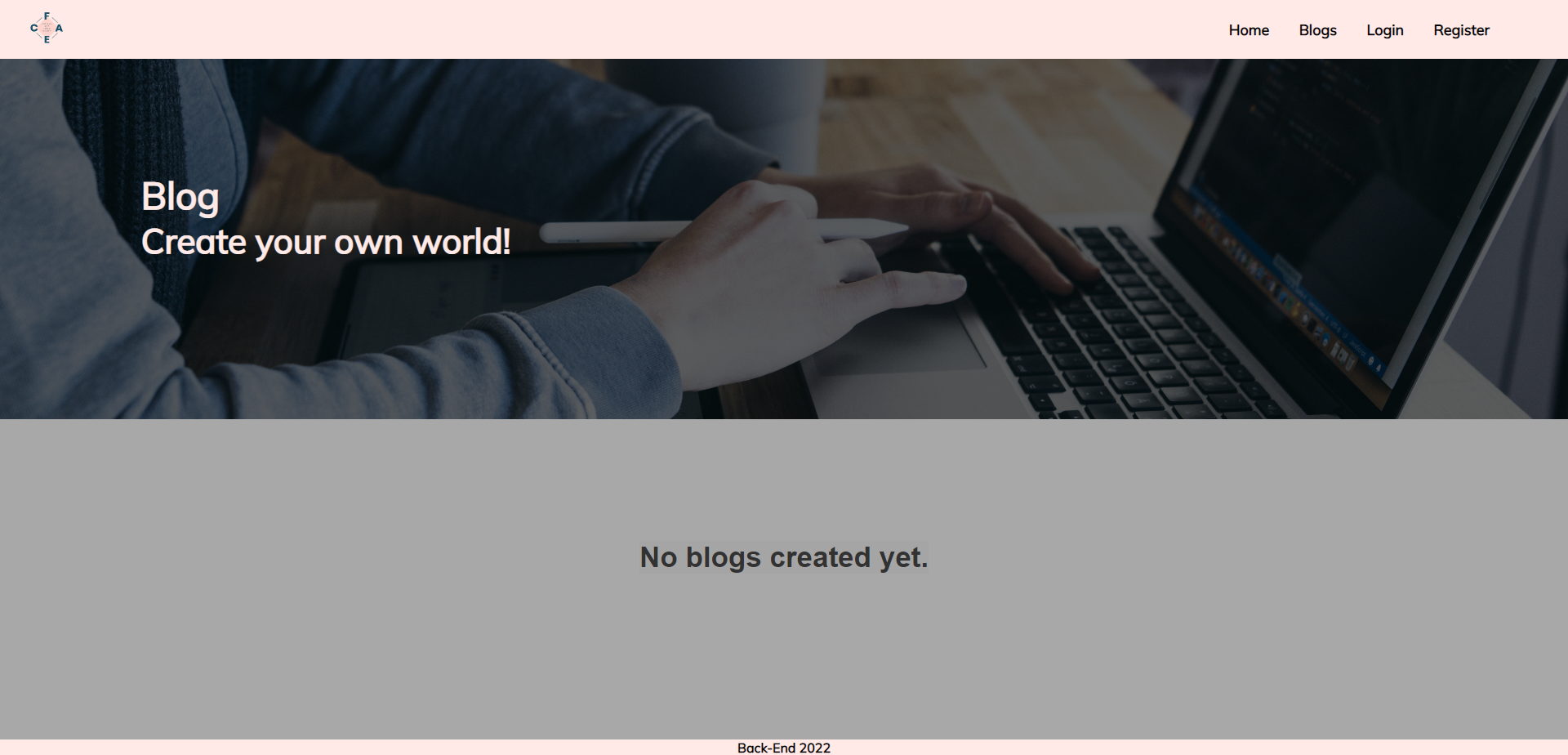
## Application Pages (80 pts)

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

Visualize **the last 3 added blog posts. Each blog must** show information about the **blog title**, and **blog category**, as well as a page with **details** about the **specific blog.**

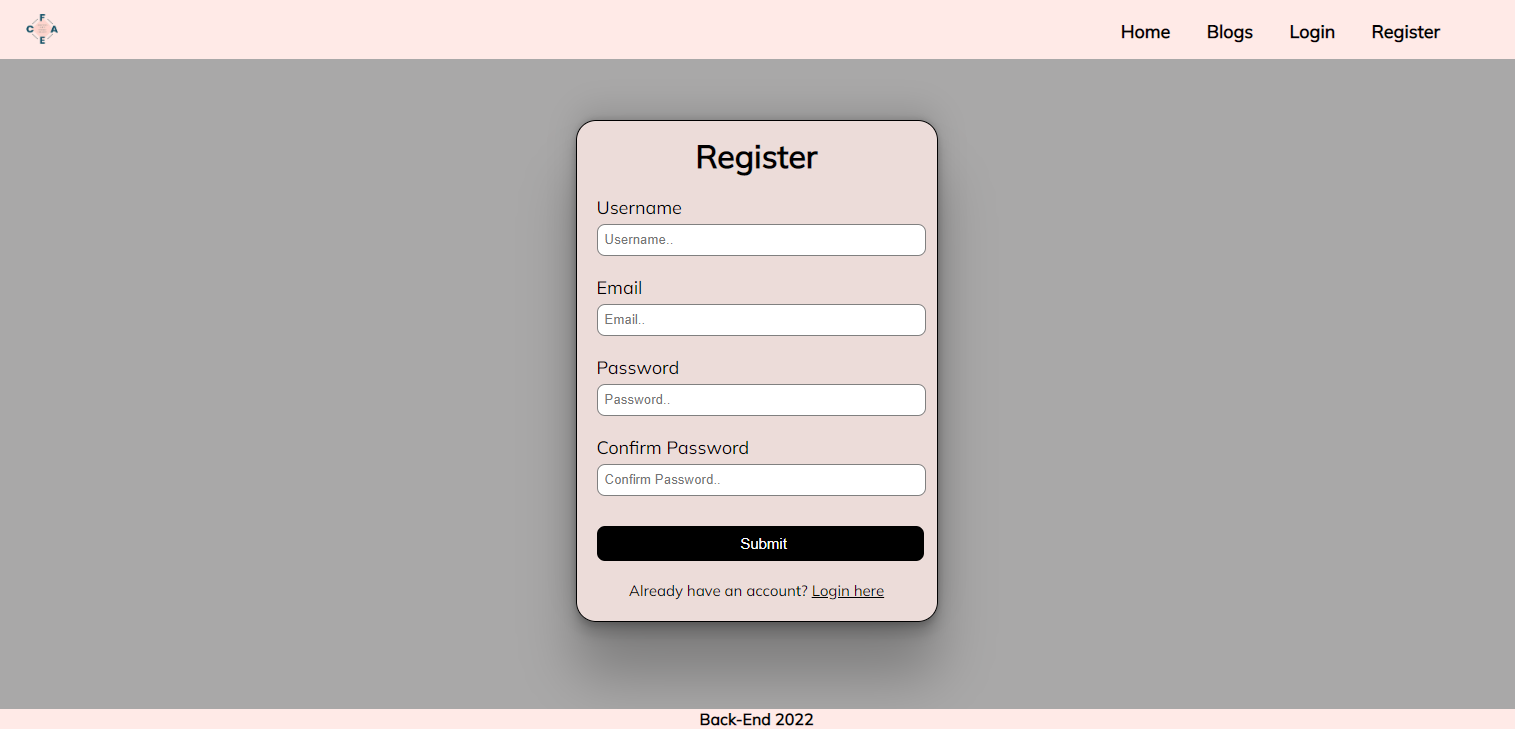


If there are **NO** blogs in the Database yet, display **"No blogs created yet."**



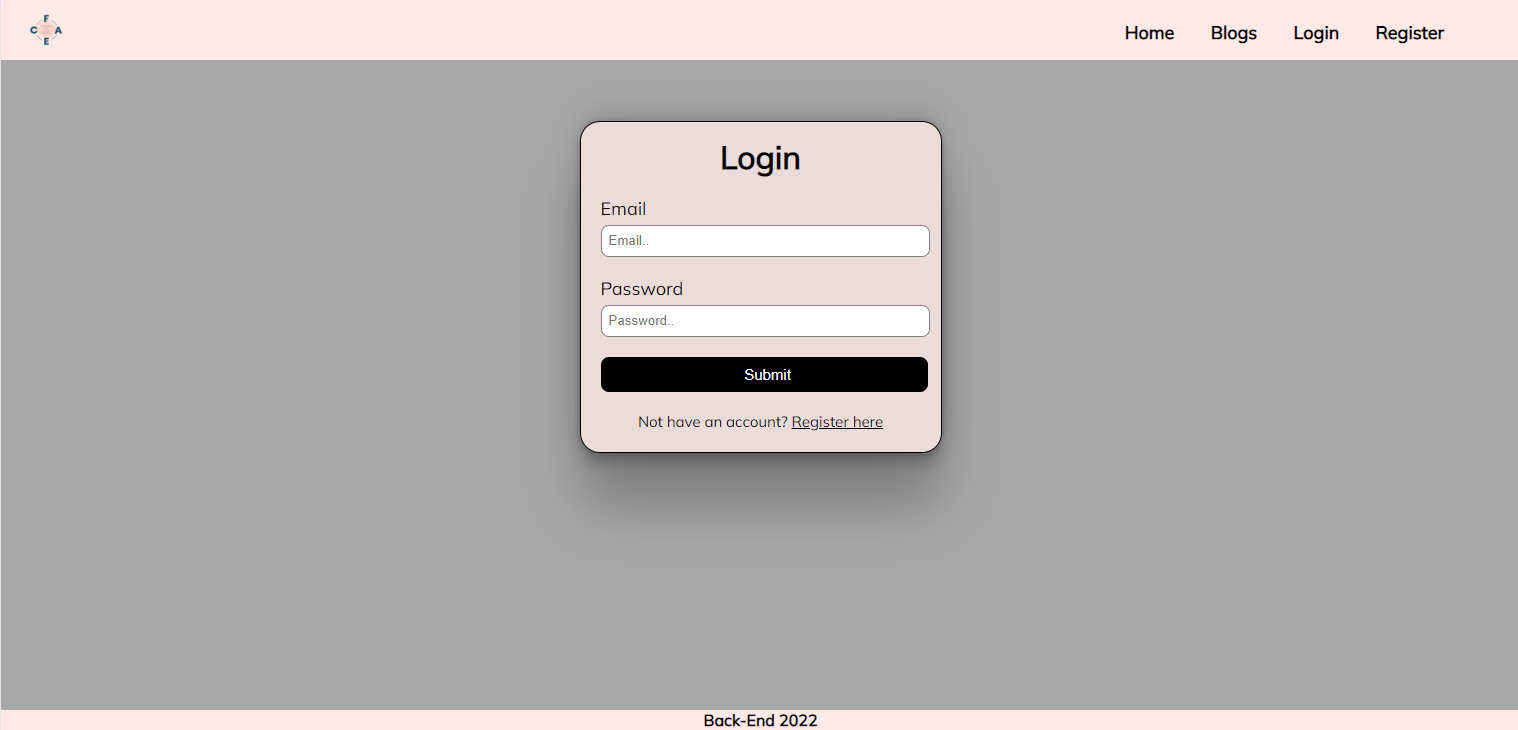
#### Register Page (Logged Out User)

**Register** a user inside the Database with a **username, email, and password**. **Password** inside the **Database** must be hashed (**use bcrypt**), and both passwords **must match**! After successful registration, you should **redirect** to the **Home page**.



#### Login Page (Logged Out User) - (5 Pts)

Logging an already registered user with the correct **email** and **password**. After successful login, you should **redirect to the Home page.**

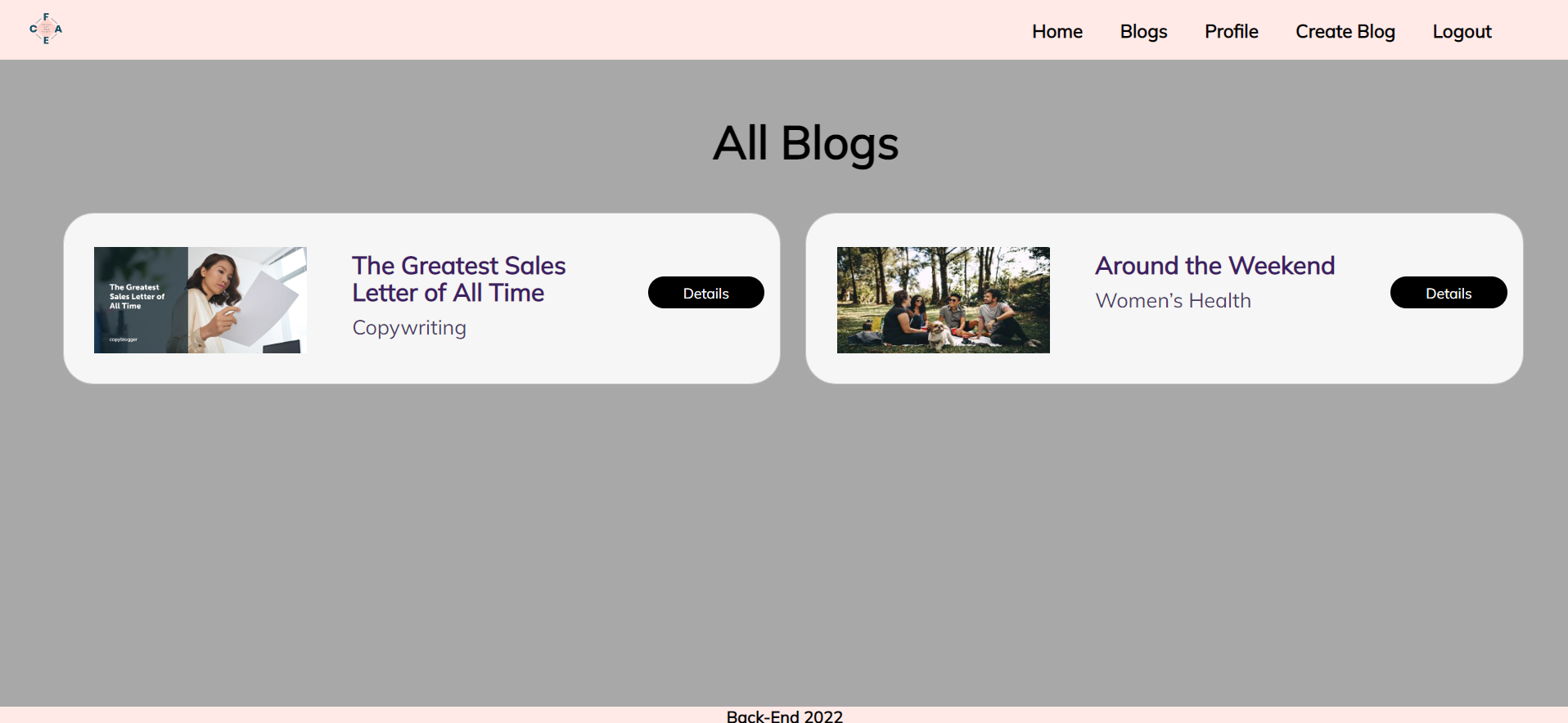


#### Logout (logged in user)

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

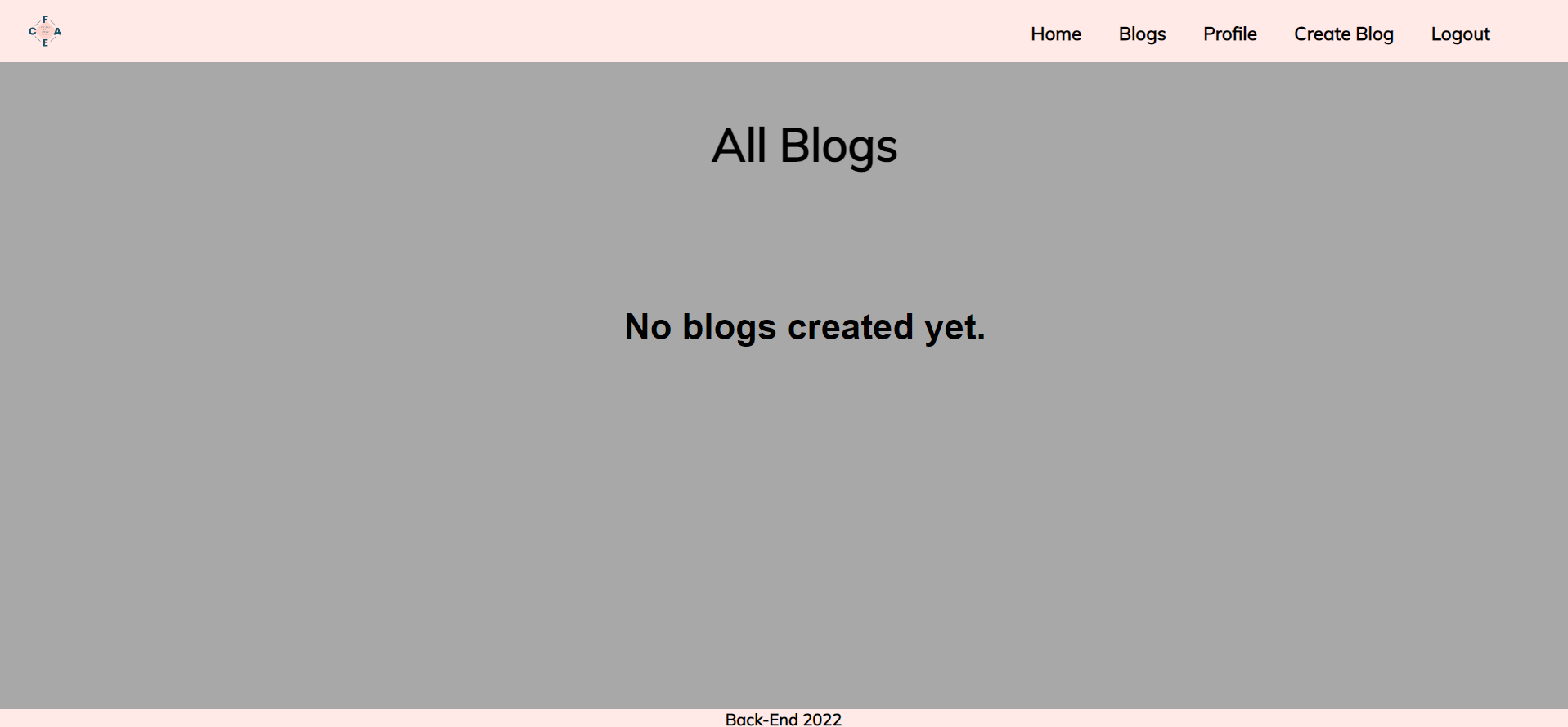
#### Blog Catalog (For logged in users and logged out users)

List all blog posts. Each post must display information about the **blog image**, the **title**, the **category,** and a button for **details** about the **specific blog**. As in the picture below:



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

If there are **NO** blogs in the Database yet, display **"No blogs created yet."**



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

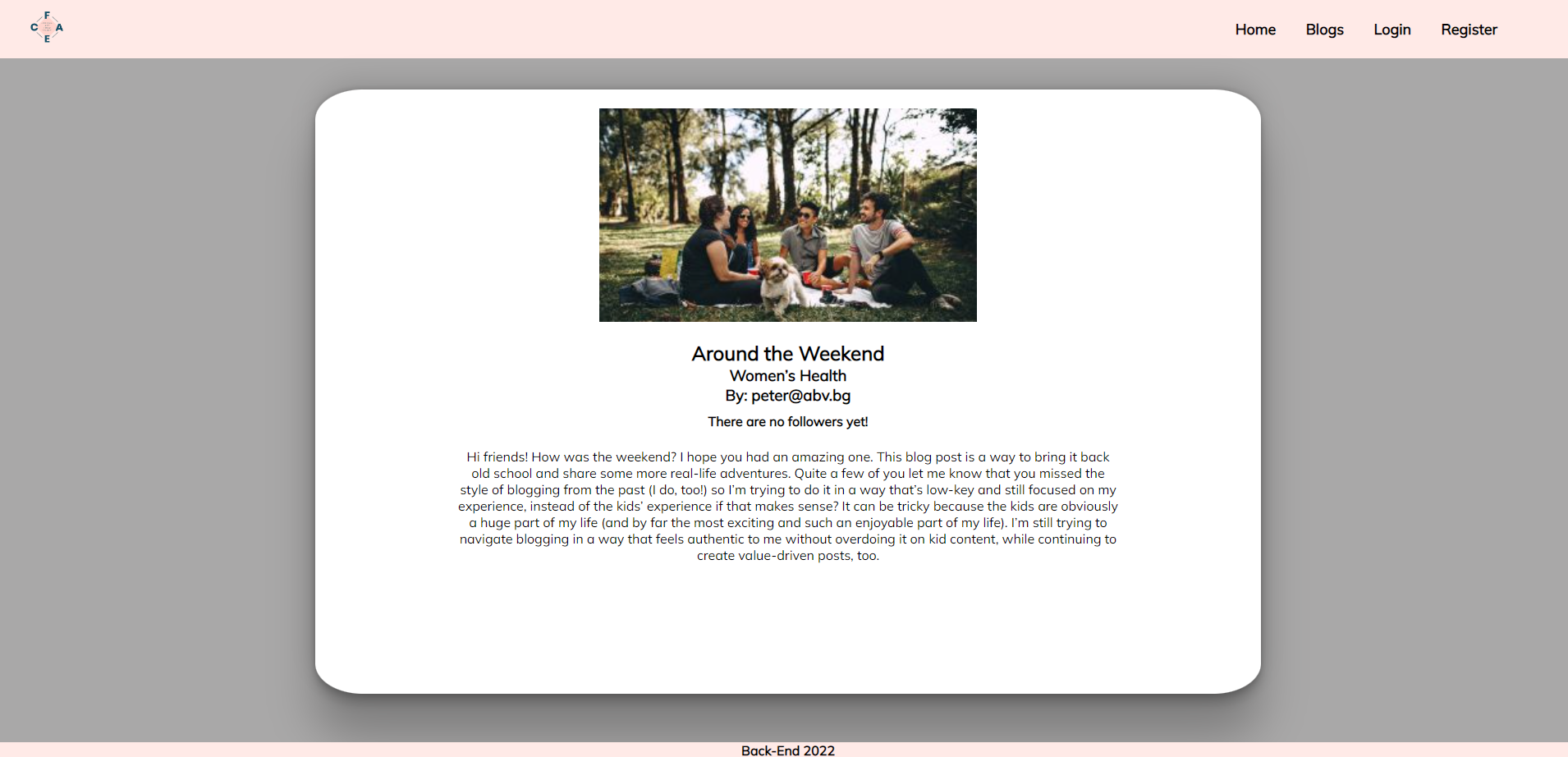
All users should be able **to see details**. Clicking the **Details button** on the blog card should display the details page. If the currently registered user **is the creator** of the blog post, the **Edit** and **Delete** buttons should be displayed. Otherwise, they **should not be available**.

Information about the blog:

* **Title**
* **Category**
* **Email of the creator**
* **People followed this blog:**
  + If any, separate their emails with comma and space **", "**
  + If **not**, display **"There are no followers yet!"**
* **Content**
* **Buttons** (Depending on the status of the currently logged in user)

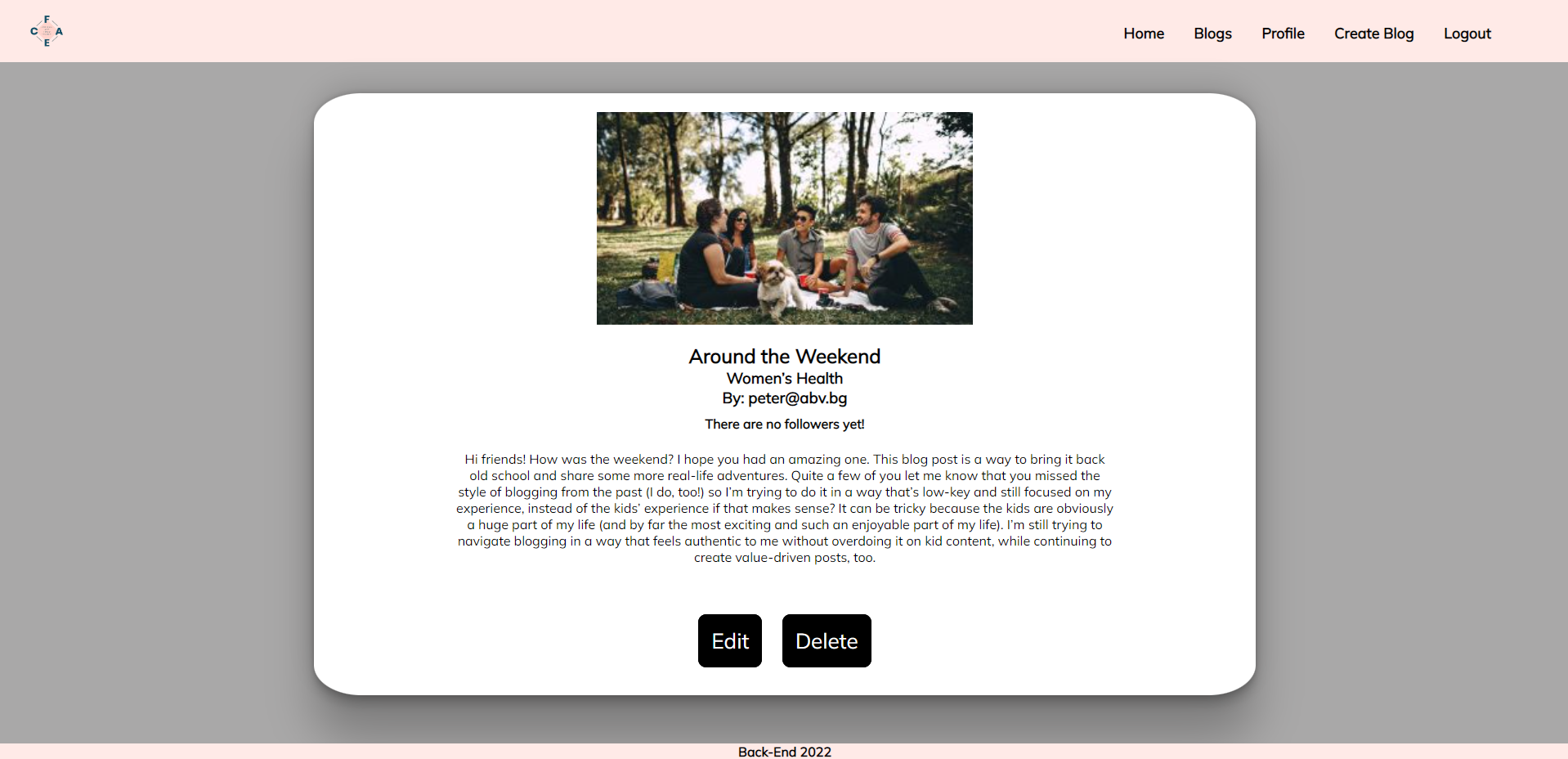
#### Details Page (logged out users)

If there are **no logged**-in users, **no buttons** should be displayed.



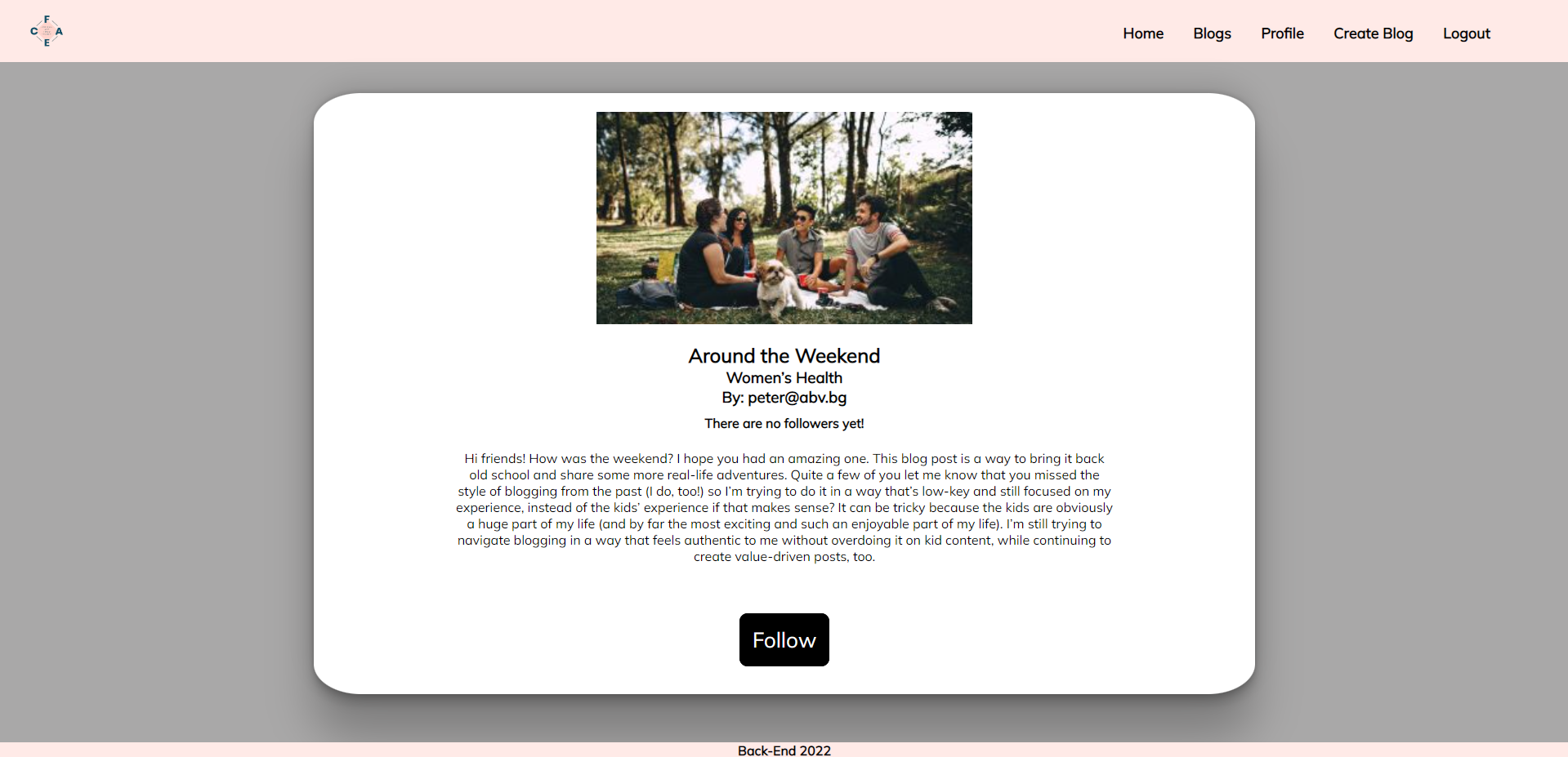
#### Details Page (logged in user and creator of the current blog)

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



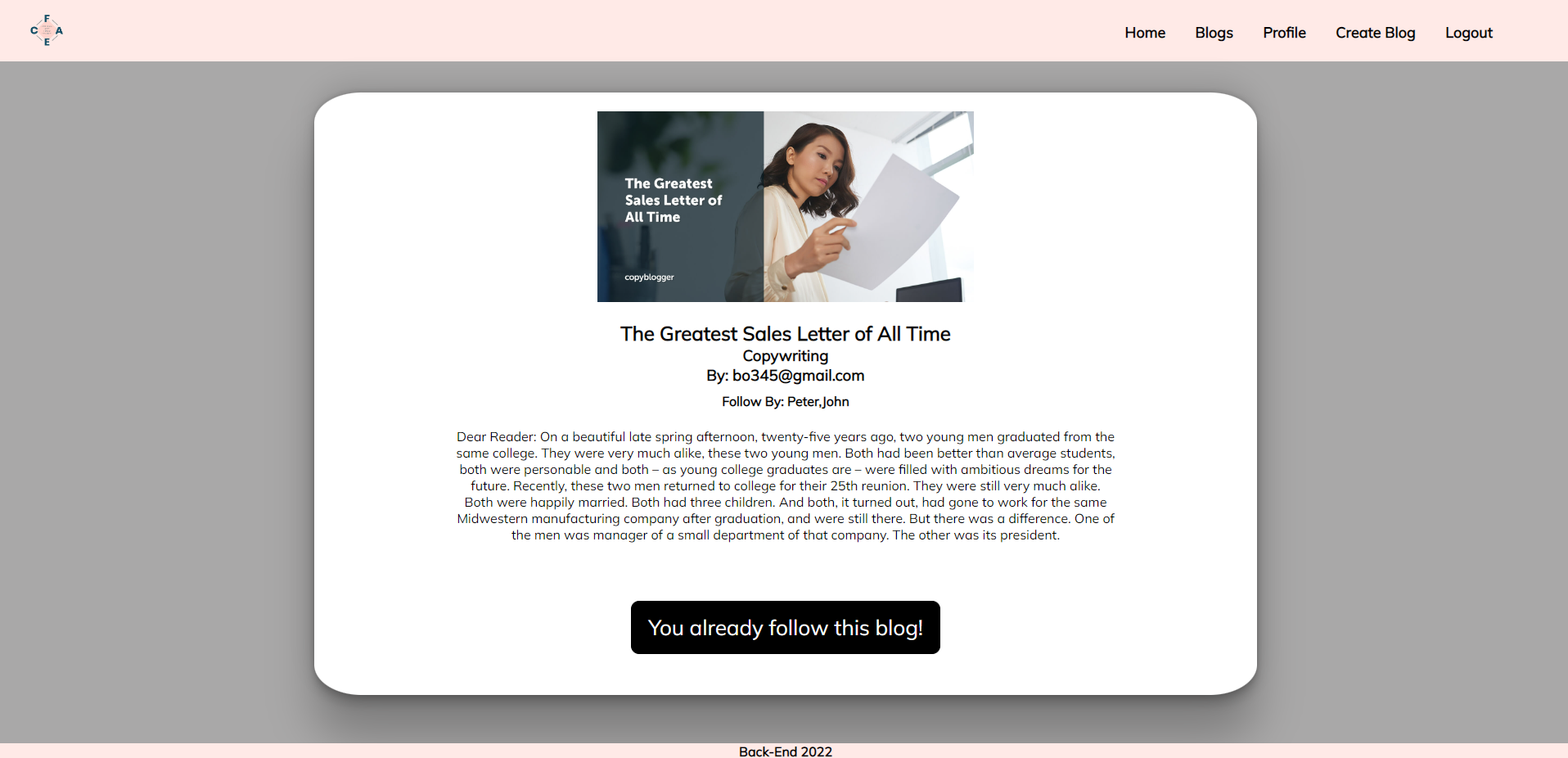
#### Details Page (logged in user who is not followed)

If the currently logged-in user is **not the owner** (a user who is **not** the creator of this blog post) and has not followed that blog, he should see a **follow** button.



#### Details Page (logged in user already followed blog)

If the currently logged in user **is not the owner** and has **already followed that blog**, he should see **[You already follow this blog!].**



#### Follow (logged in user who is not the current blog owner)

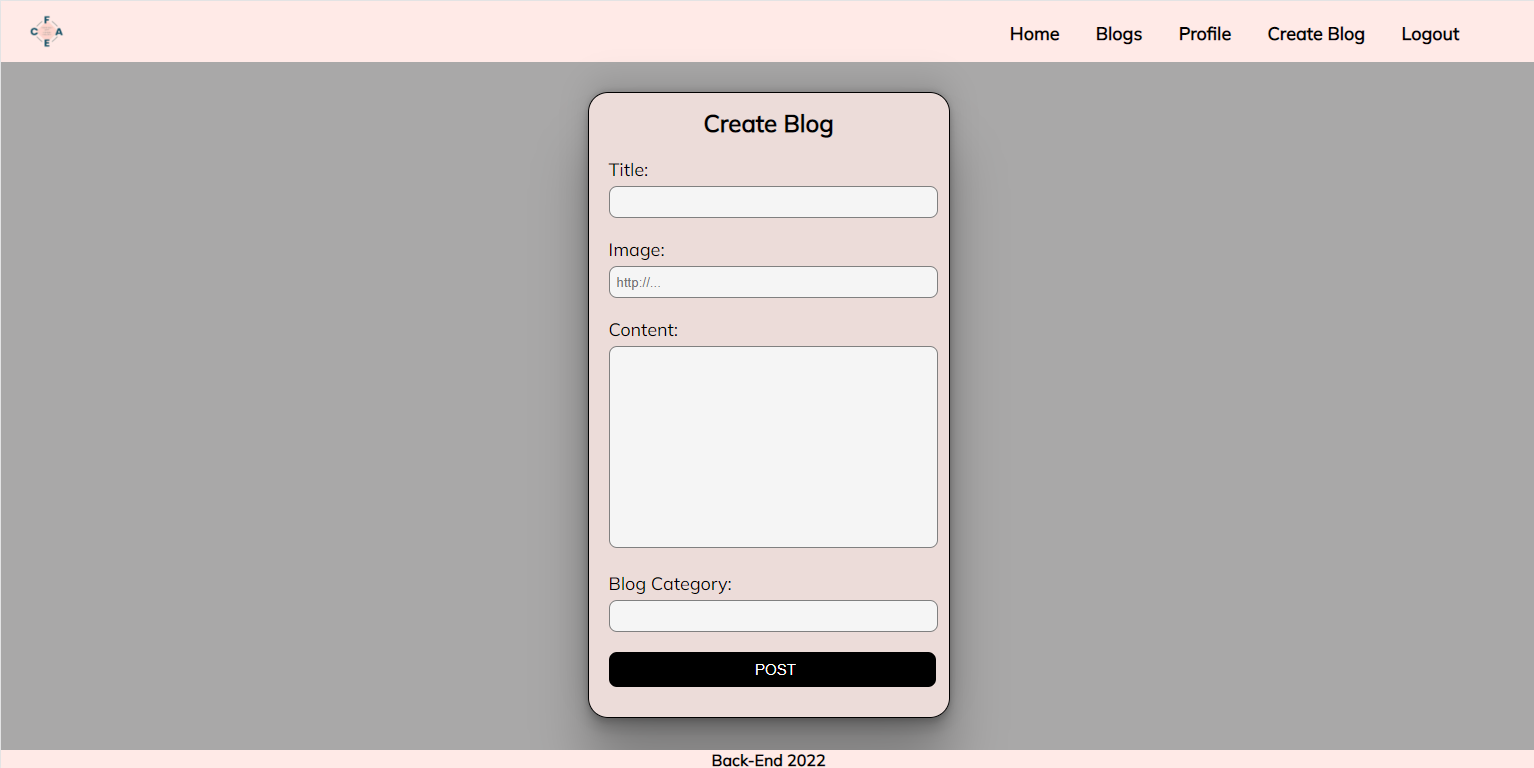
Any registered user who is **not** the current **owner** of the blog post must be able **to follow the blog** (if any).

If he manages to **follow the blog successfully**, his **userId** must be added to the collection of **FollowList** and **redirect** the user to the **Details** page for the current **blog post**.

If a user has once followed the current blog post, he should see " **You already follow this blog!**".

#### Create Blog (Logged in User)

The **Create** page is available to **logged-in users**. It contains a form for adding a new blog post. Upon success, **redirect** the user to the **Blogs** page.

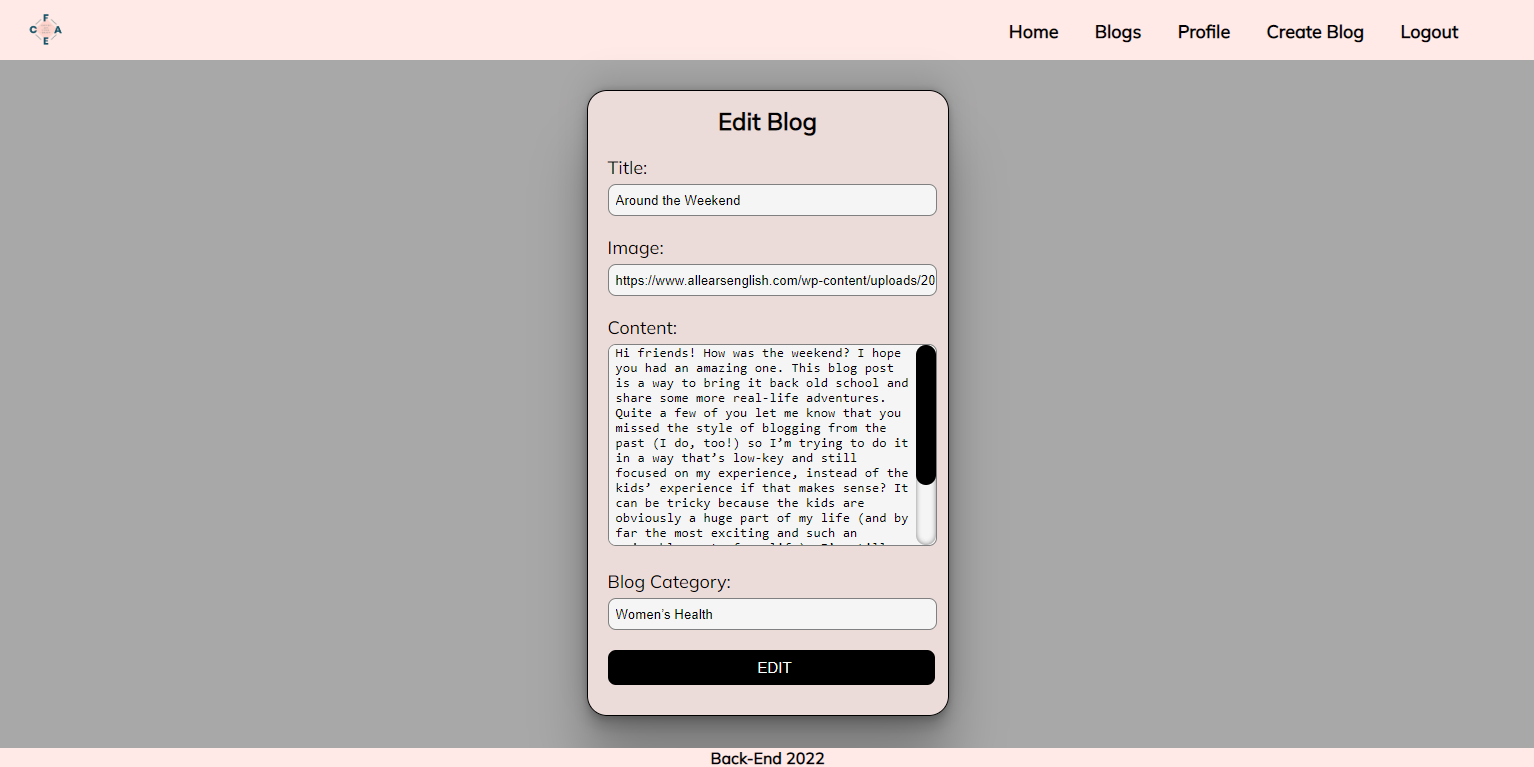


#### Delete Blog (logged in user and owner of the current blog)

Each **owner** of the blog post must be able to click on the **[Delete]** button and **delete the current blog** from the Database, and the user must be redirected to the **Blogs** page.

#### Edit Blog (logged in user and owner of the current blog)

Each **owner** can edit their **blog post**. Clicking the **[Edit]** button for a specific blog on the details page should display the **Edit page**, all fields being populated with blog data. It contains a form with input fields for all relevant properties. If **successful**, redirect the user to the **current blog post details page**.

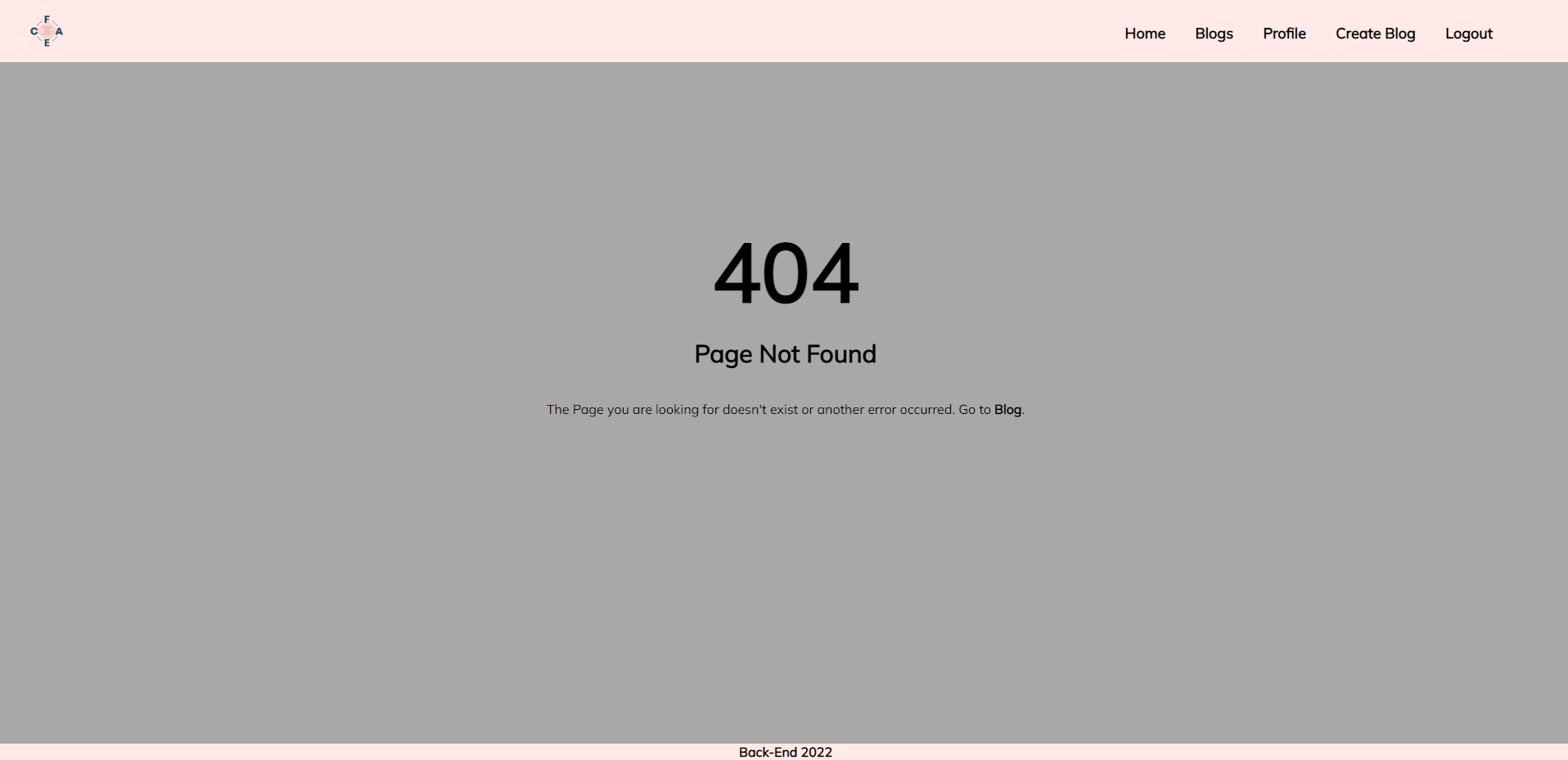


## Security Requirements (Routes Guards) - (10 Pts)

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 **Blogs** page (Listed all blog posts).
* 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 blog post owners) can **follow blogs**.
  + **Users** (blog post owners) can **Edit** and **Delete** the current blog post
* Users (logged in) can access the **Profile** page and functionality.
* Users (logged in) can access the **Create Blog** page and functionality.
* Users (logged in) can access the Logout functionality.

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.

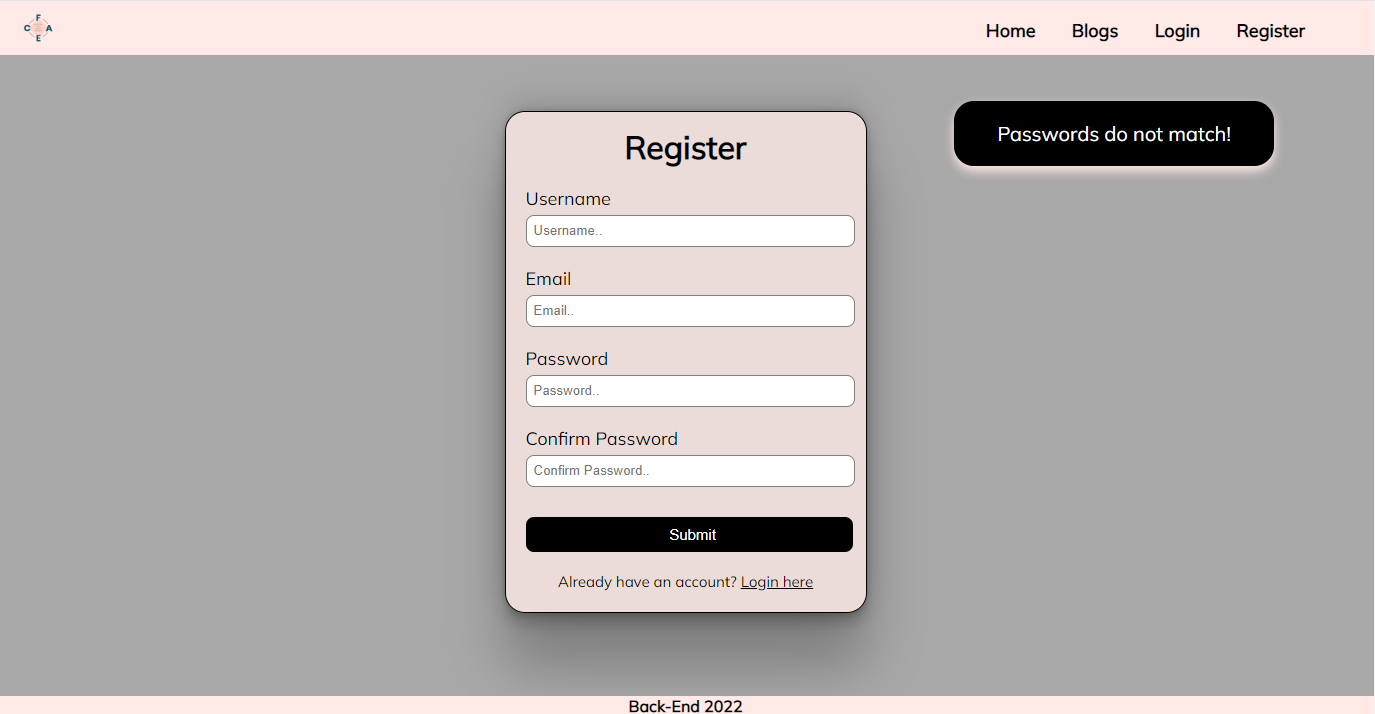
In case of error, you should display div with class "**errorContainer**"

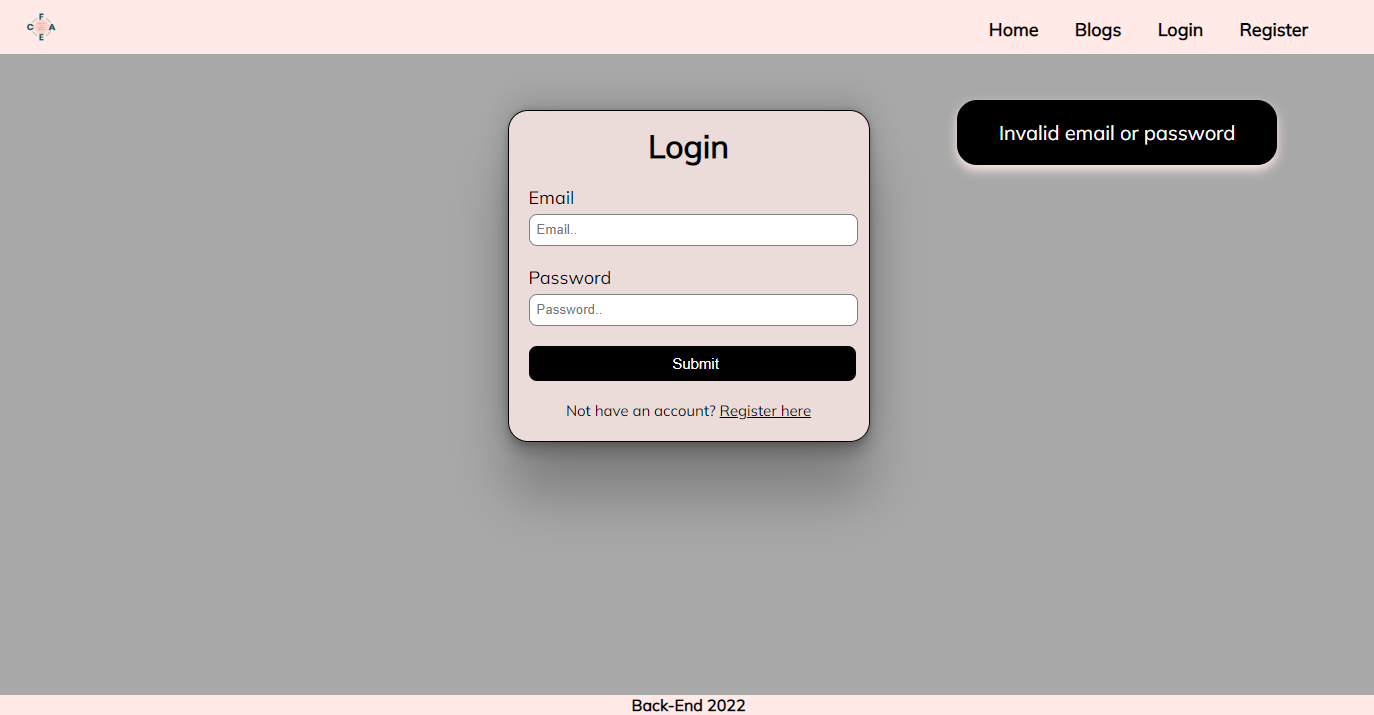
### You can choose to display the first error or all of them. You have complete freedom to choose the content of the error message you will display.

### 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**



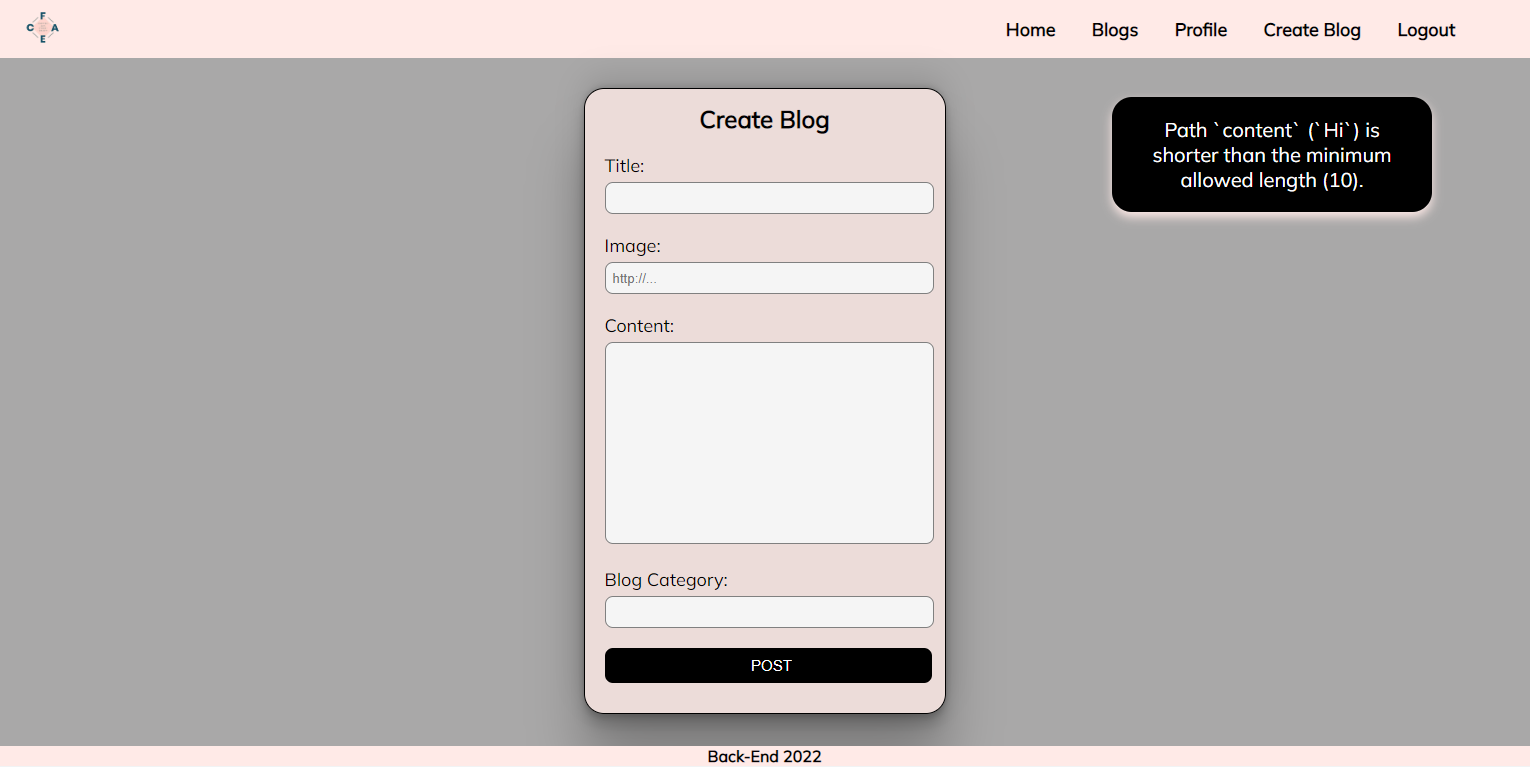


### Blog

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

* The **Title** should be **at least 5 characters and no longer than 50 characters**
* The **Blog Image** should start with **http:// or https://**
* The **Content** should be a minimum of **10 characters** **long**
* The **Category** should be a minimum of **3 characters long**



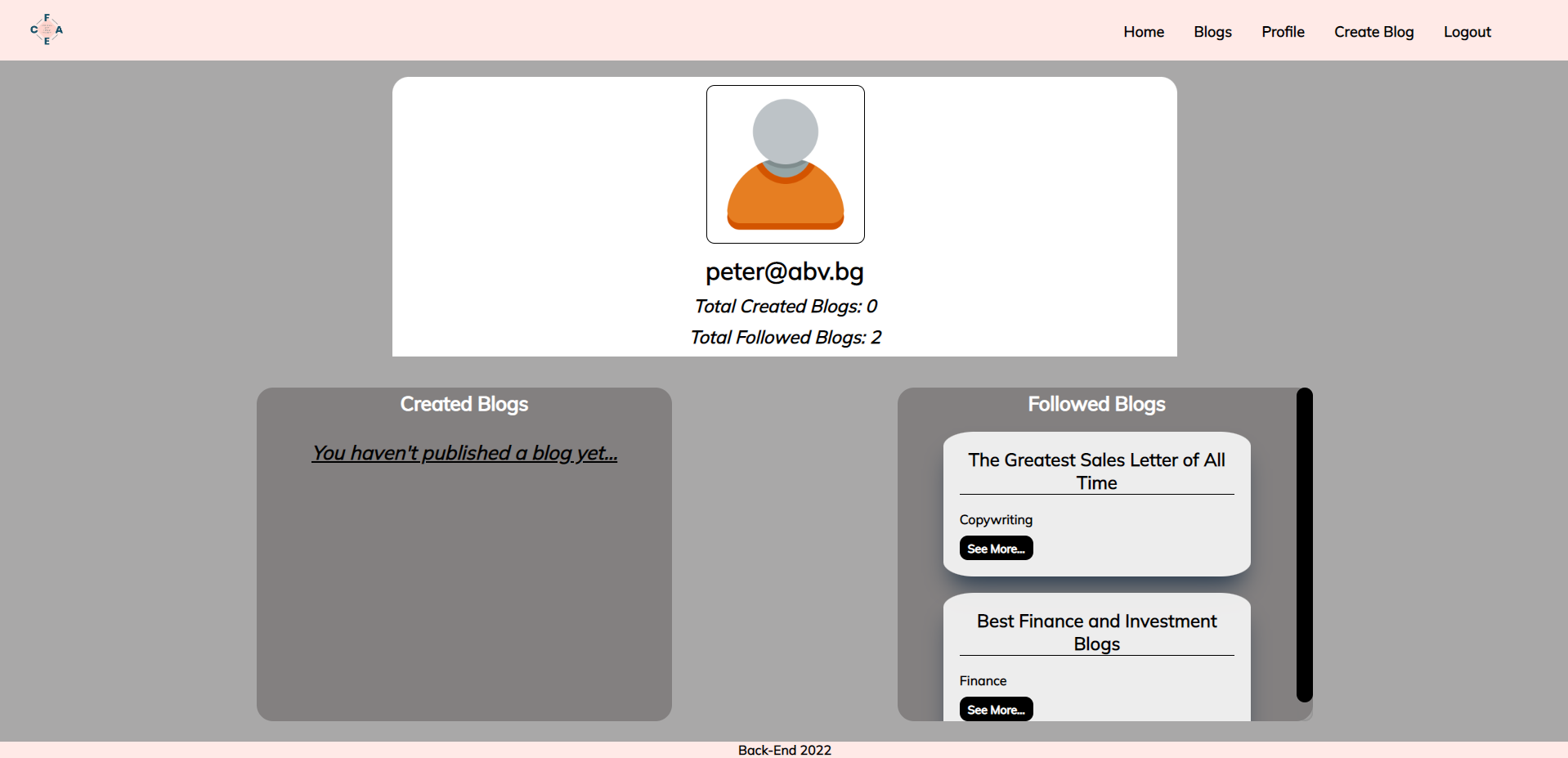


## \* Bonus – Profile (10 Pts)

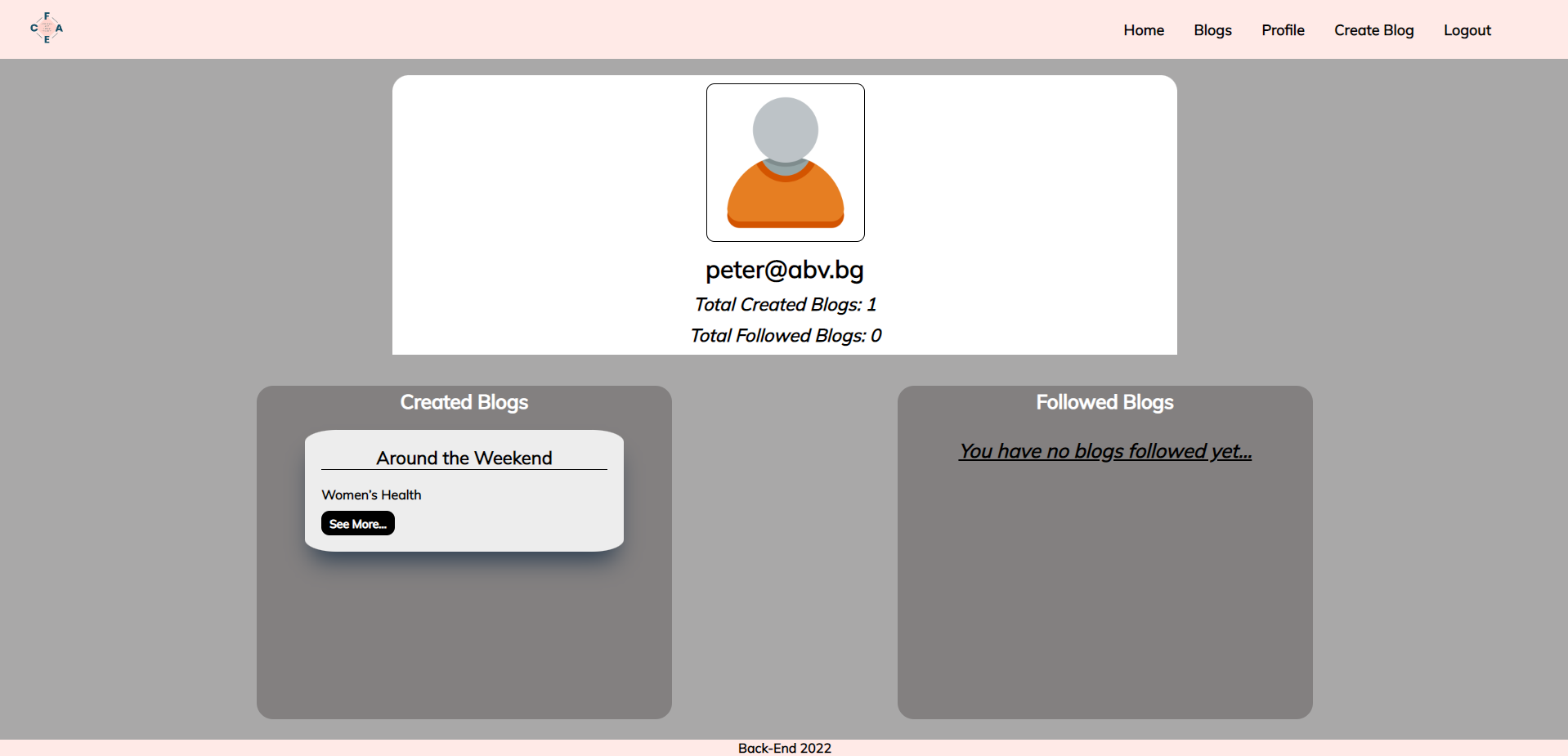
Each **logged-in** **user** should be able to view their profile information by clicking [**Profile**]. **Email, and write blog titles (**which the **current user** has **written) and followed blogs.** Under the **email**, the user has a **total count of created and followed blogs.**



If there are **no written blogs** yet, the message **"You haven't published a blog yet…"** should be visualized.



If there are **no followed blogs** yet, the message **"You have no blogs followed yet…"** should be visualized.



## Submitting Your Solution

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









*GOOD LUCK! 😊*