

JS Back-End Retake Exam – Gaming Team

1. 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**.
8. You can use whatever **view engine** you like (**express-handlebars**, EJS, Pug, etc.).
9. Use **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!**

2. Application Overview

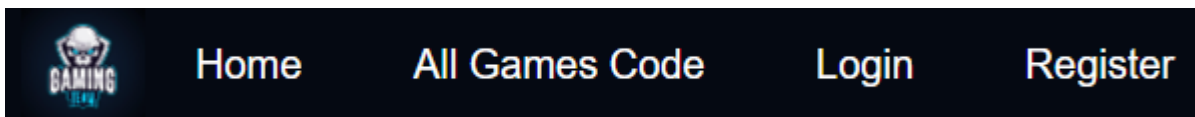
Get acquainted with the provided **HTML and CSS** and create an application for the **game code**.

The visitors can **view** the **home page** and **All Games Code** catalog with the available offers. They can also **register** with an **email**, **username**, and **password**, which will allow them to create their offers for game code and **buy games** (if the **current user** is **not the owner of the game**). Authors can edit or delete posts at any time.

3. Functional Requirements

Guests (Not Logged-in)

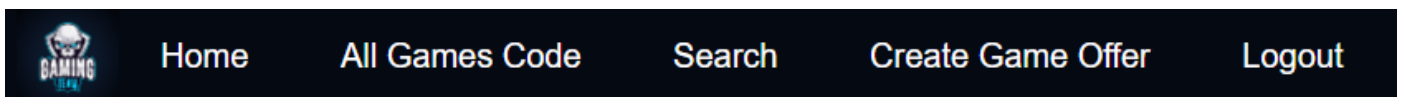
Guest navigation example:



The **application** should provide **guests** (not logged-in users) with the functionality to **login**, **register**, and **view** the **home page**, the **All Games Code** page, and the **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 the other pages with logged-in navigation;**
- **view the All Games Code page;**

- create a new game offer (Create Game Offer);
- access the game details pages (Details);
- buy games (if the current user is not the owner of the offer);
- delete or edit the offer depending on the user's authentication (only for the owner of the current game offer).

4. Database Models

The **database** of the **Gaming Team** application needs to support **two entities**:

User

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

Game

- name: string (required),
- image: string (required),
- price: number (required),
- description: string (required),
- genre: string (required),
- platform: string (required; one of the following: "PC", "Nintendo", "PS4", "PS5", "XBOX"),
- boughtBy: a collection (array) of users (references to the "User" model)
- owner: object ID (a reference to the "User" model)

Note: When a user **buys** a game, their **ID** is added to the collection **boughtBy** of the game they bought.

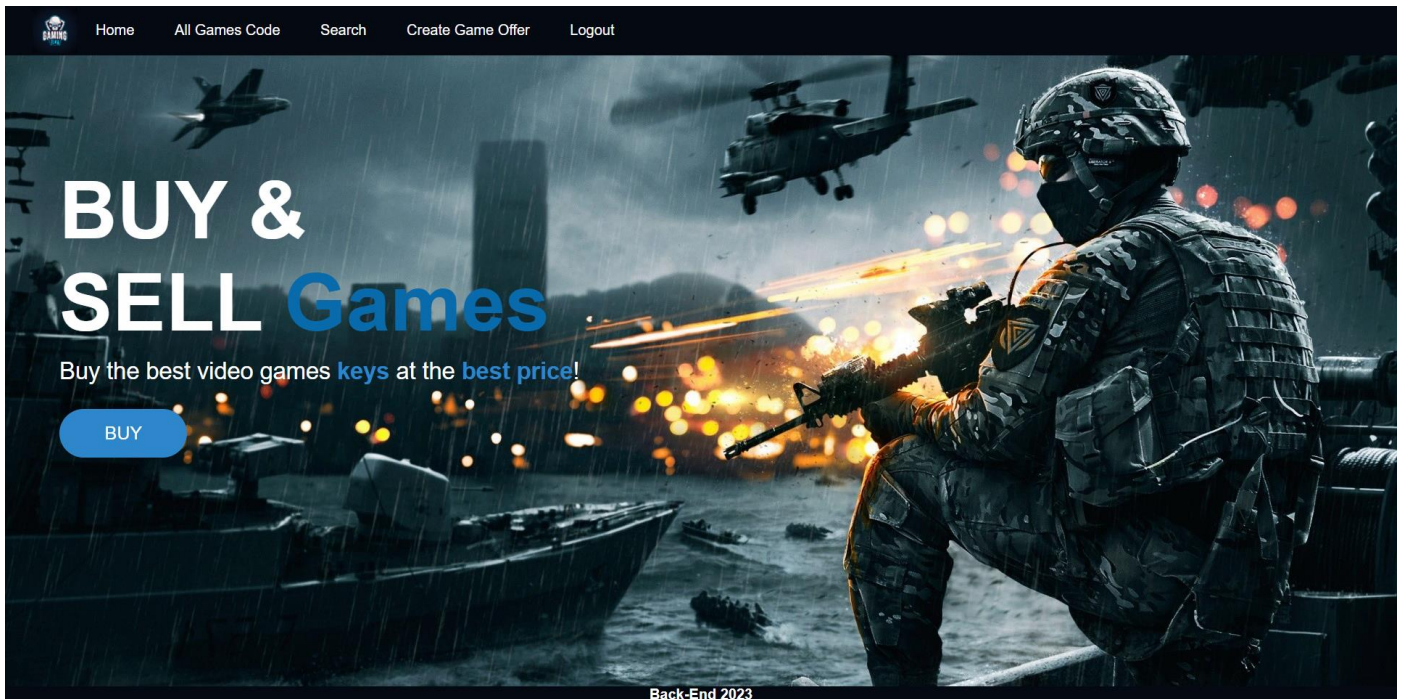
Implement the entities with the **correct data types**.

5. Application Pages (80 pts)

Home Page (Guests and Logged-in Users)

Visualize the following **static home page**:

Clicking the **[Buy]** button, the user should redirect the user to the **catalog** page.

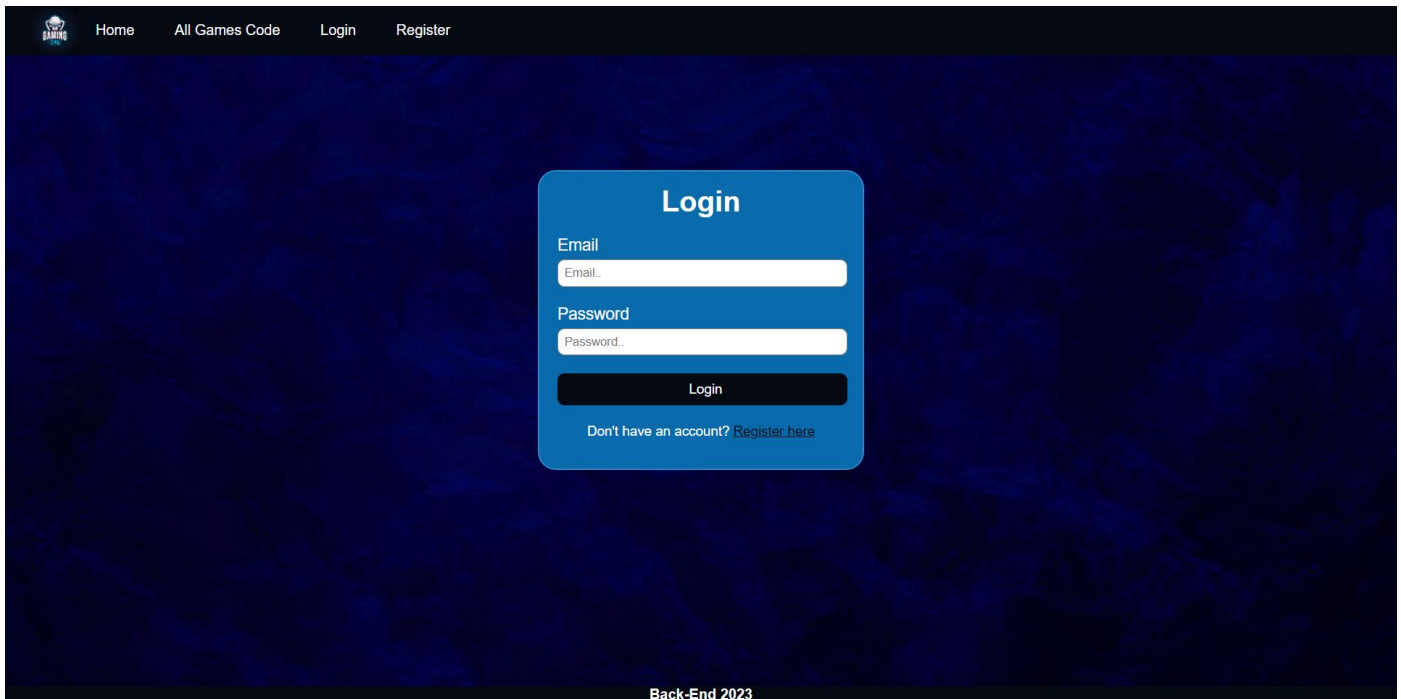


Register Page (Guests)

Register a user in 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**.

Login Page (Guests)

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

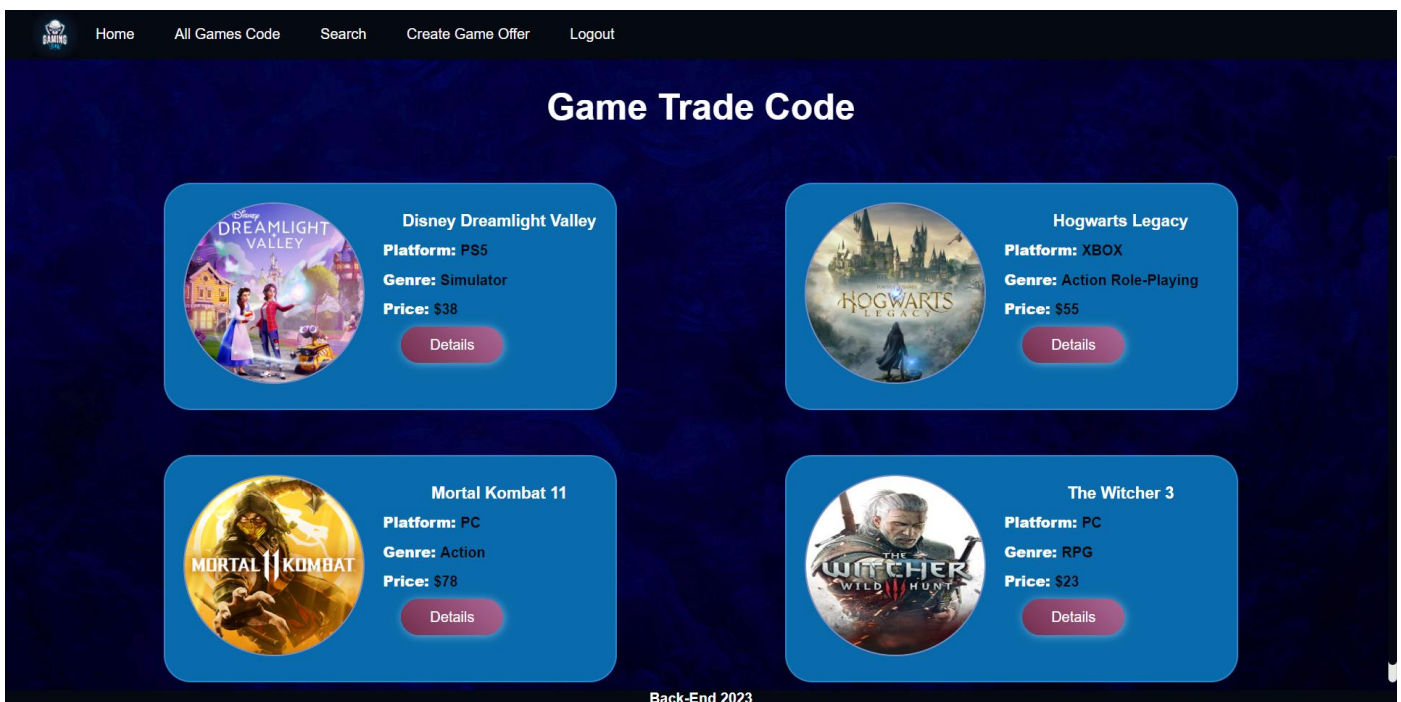


Logout (Logged-in User)

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

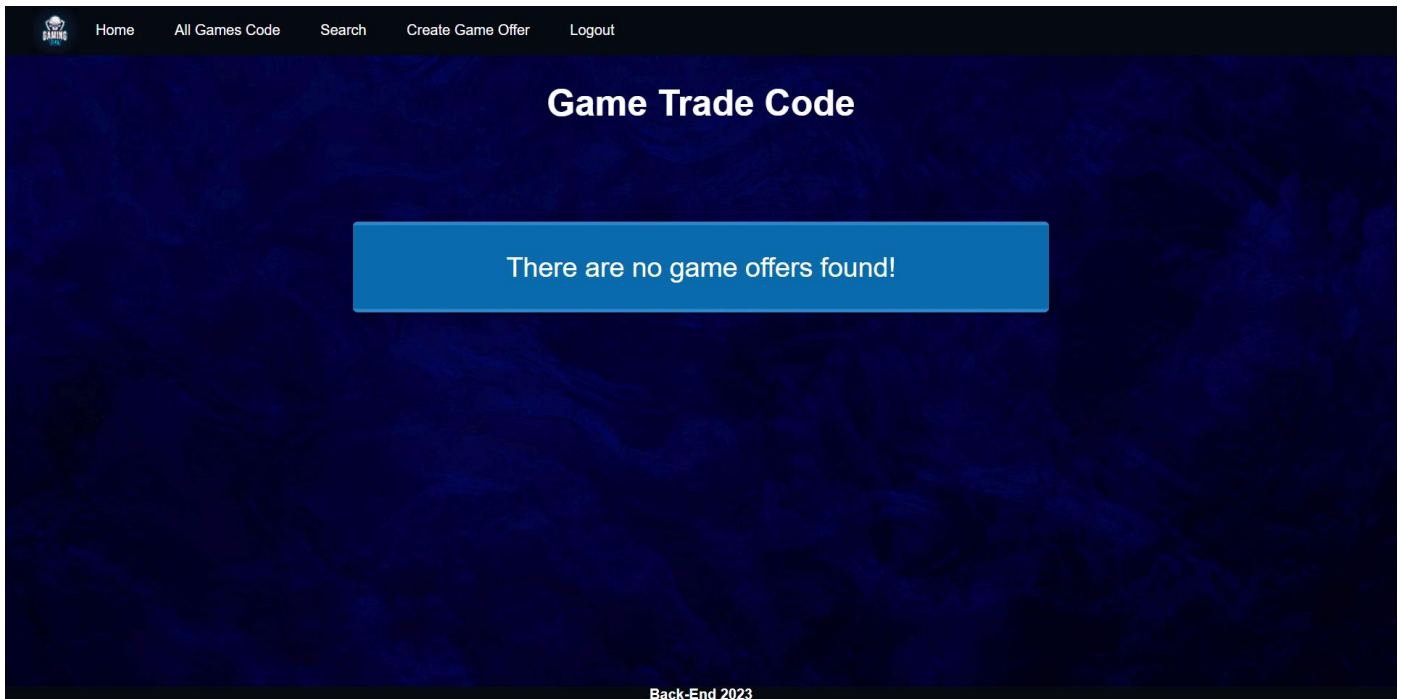
Game Catalog (Guests and Logged-in Users)

List all game code offers. Each offer must display the **game image**, the **name**, the **price**, the **genre**, the **platform**, and a button for **details** about the **specific game**. As in the picture below:



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

If there are **NO** game offers in the database yet, display **"There are no game offers found!"**.



Details Page (Guests and Logged-in Users)

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

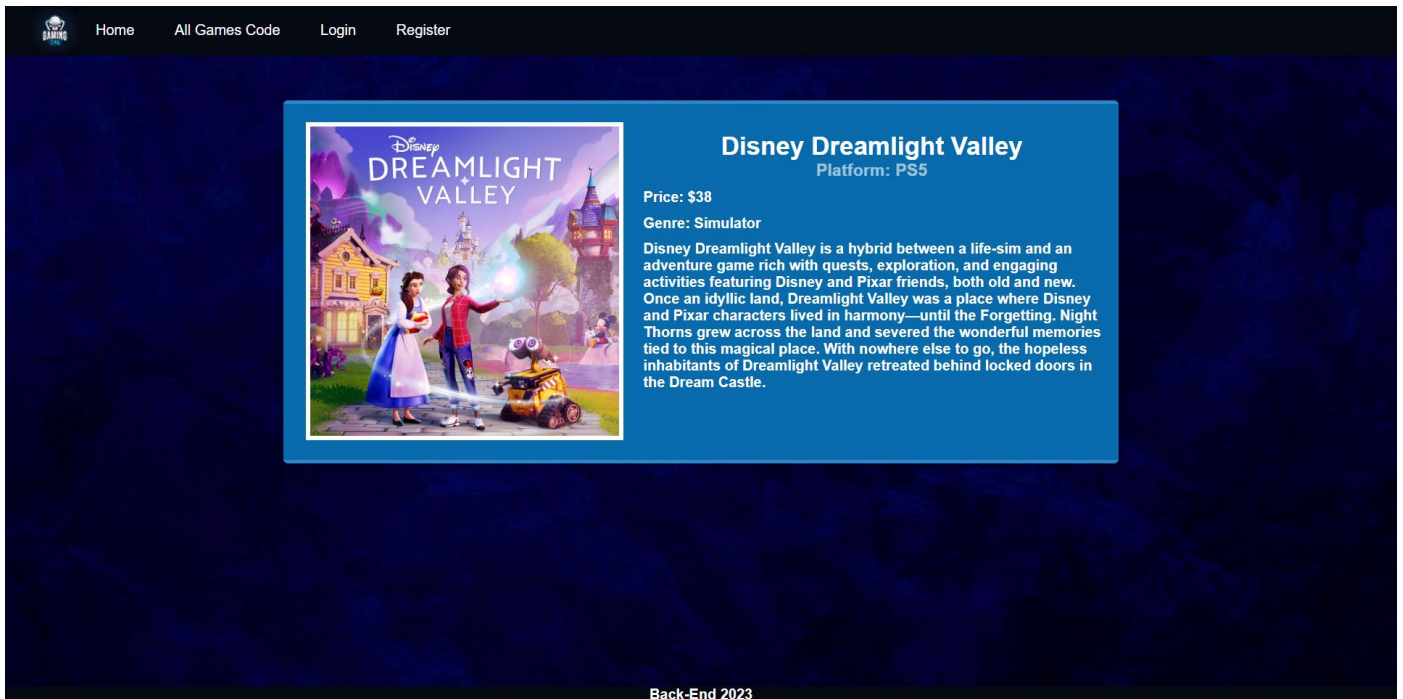
Information about the game:

- **name**
- **platform**
- **price**
- **genre**
- **description**

Buttons: depending on the status of the currently logged-in user.

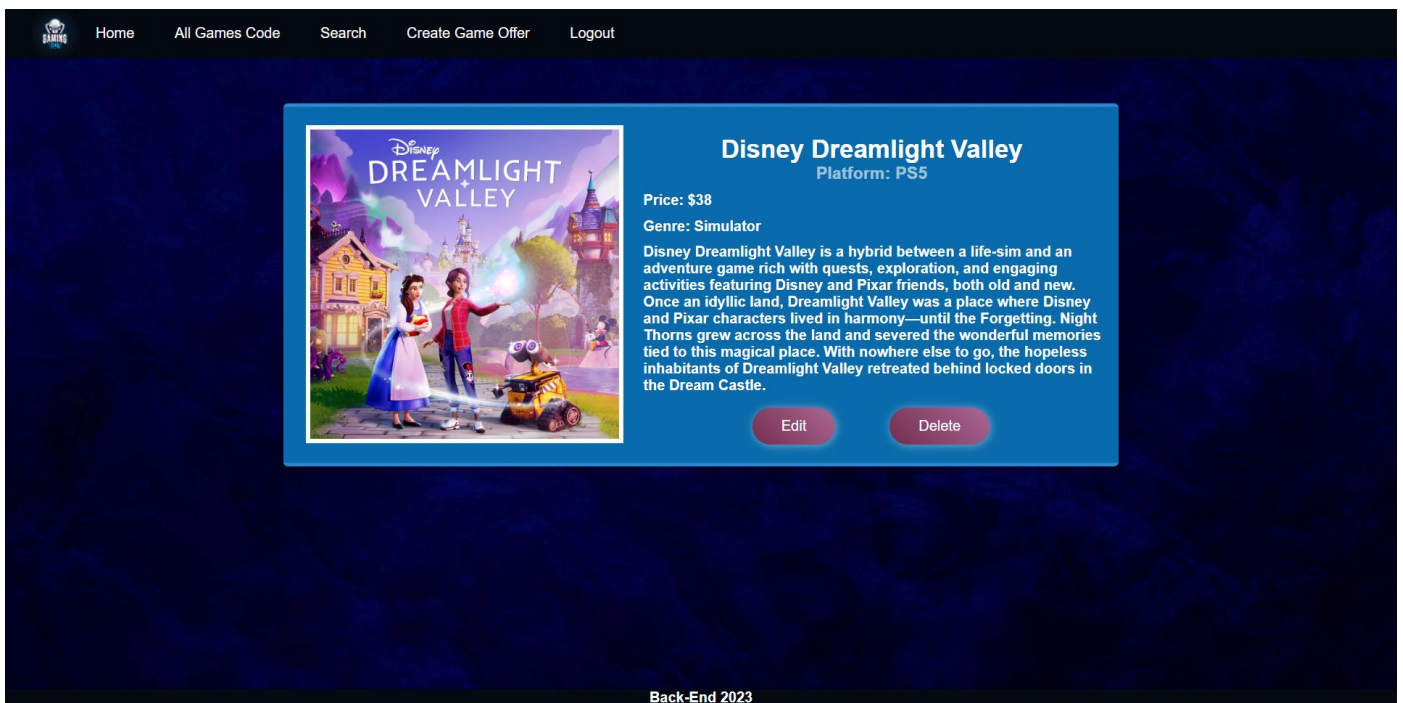
Details Page (Guests)

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



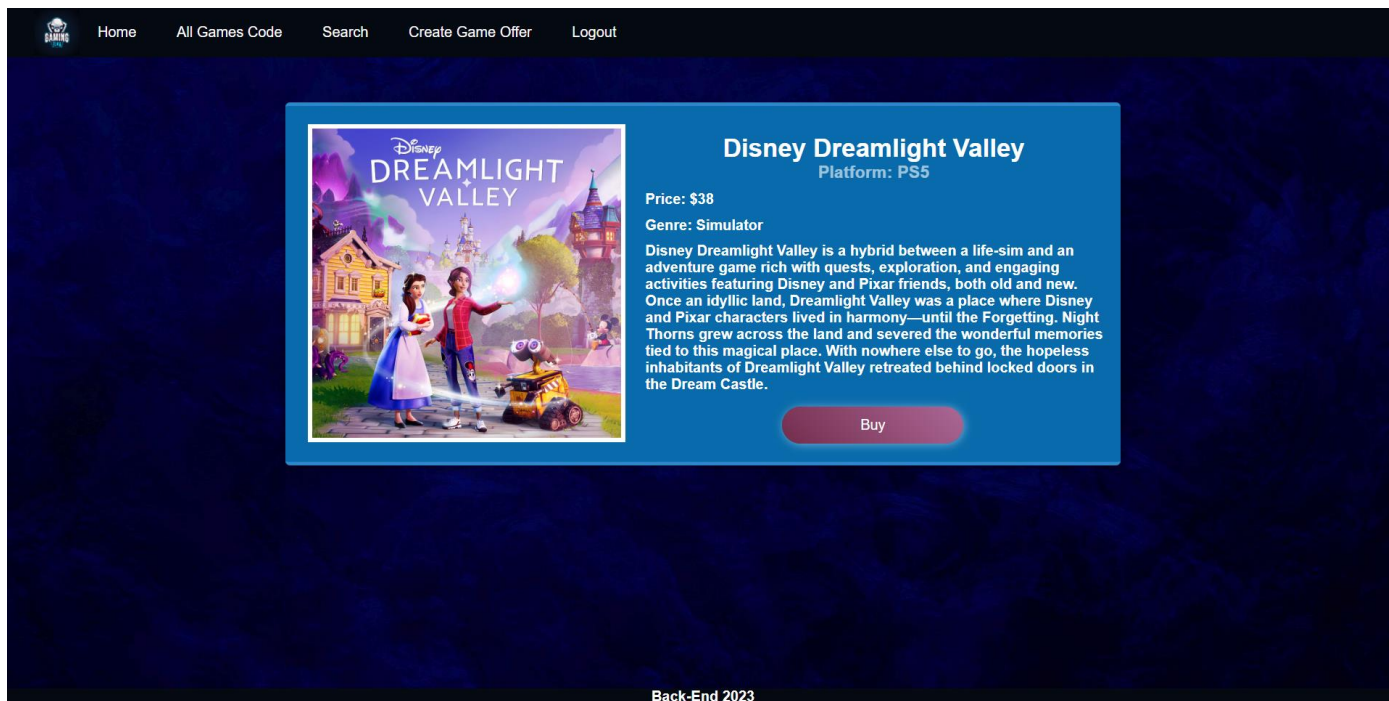
Details Page (Logged-in User, Owner of the Current Offer)

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



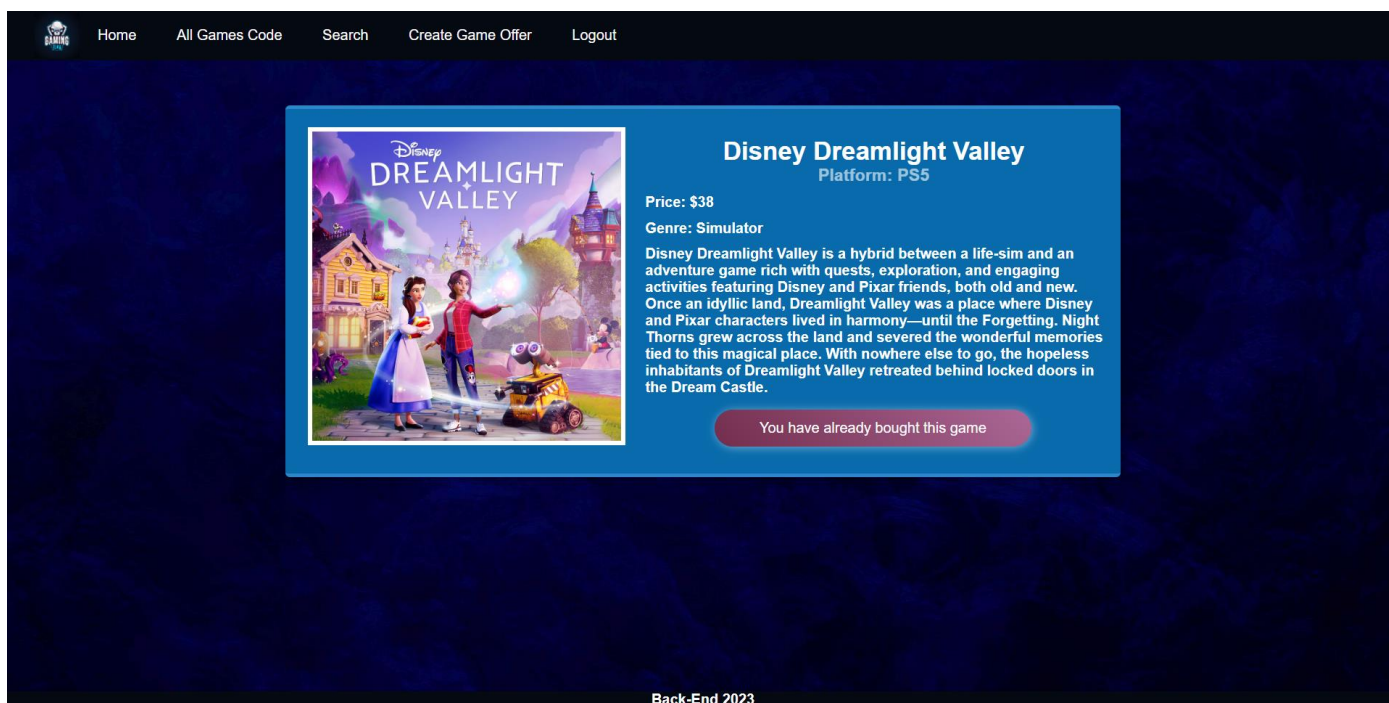
Details Page (Logged-in User, NOT Owner of the Current Offer)

If the currently logged-in user is **not the owner** (the creator) of the game offer and **has not bought** the current game, they should see a **[Buy]** button.



Details Page (Logged-in User, Already Bought the Game)

If the currently logged-in user is **not the owner** and has **already bought the game**, they should see [You have already bought this game].



Buy a Game (Logged-in User, NOT Owner of the Current Offer)

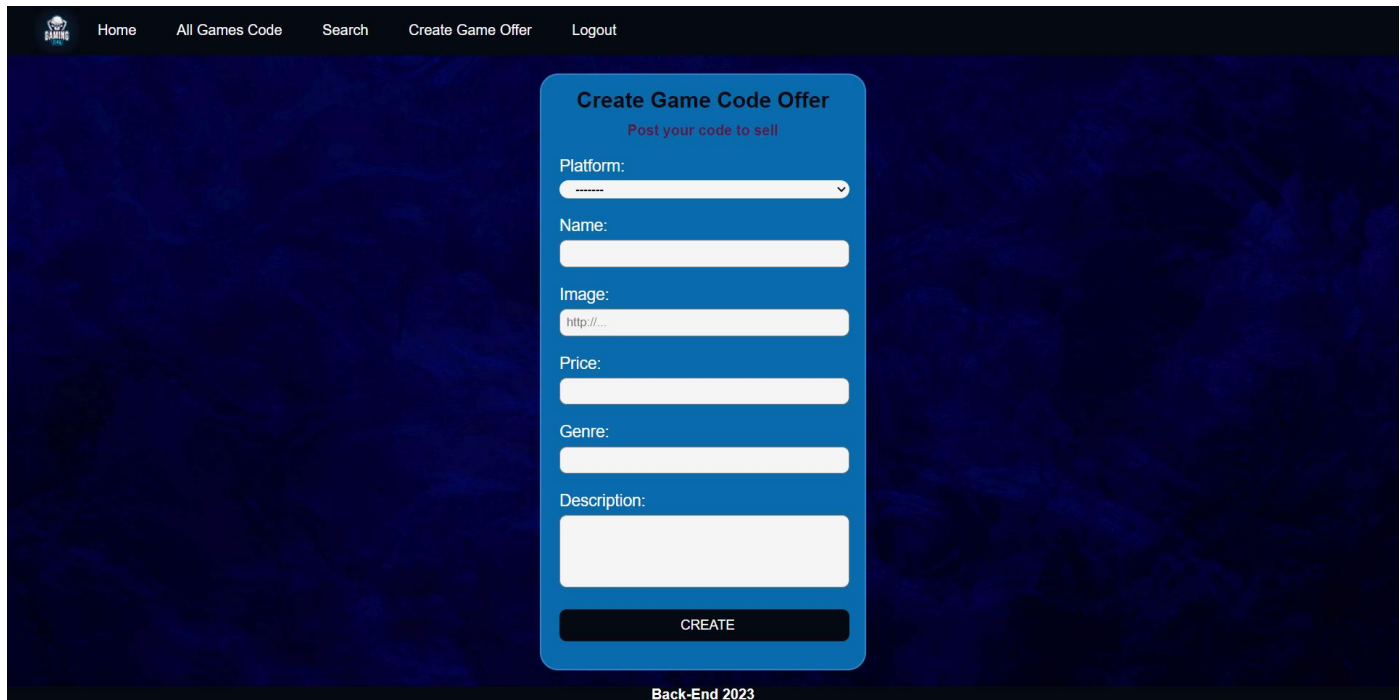
Any registered user who is **not the owner** of the current game offer must be able **to buy it**.

If they manage to **buy the game successfully**, their **ID** must be added to the **boughtBy** collection of the game and the application should **redirect** them to the **details** page for the current **game offer**.

If a user has already bought the current game, they should see "**You have already bought this game**".

Create an Offer (Logged-in User)

The **Create Game Offer** page is available to **logged-in users**. It contains a form for adding a new game offer. Upon success, **redirect** the user to the **All Games Code** page.



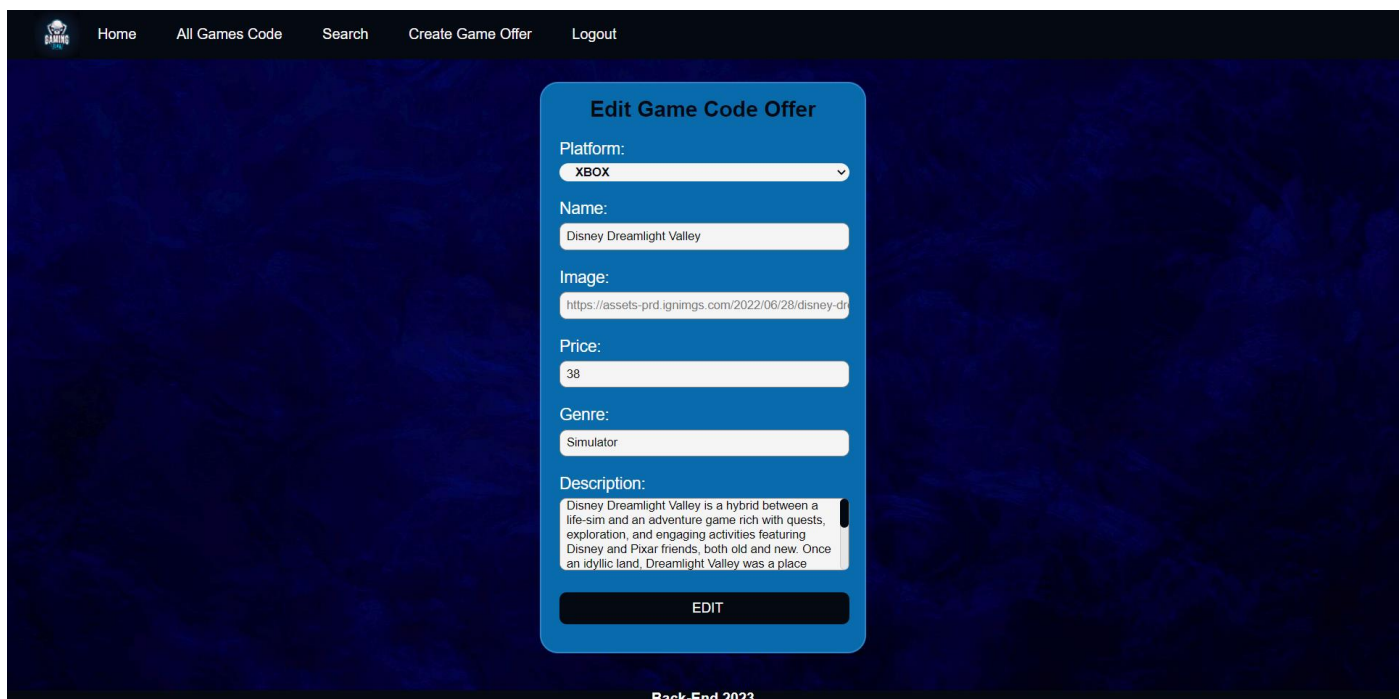
The screenshot shows a web application with a dark blue background. At the top, there is a navigation bar with links: Home, All Games Code, Search, Create Game Offer, and Logout. The main content area features a light blue rounded rectangle titled "Create Game Code Offer" with the subtitle "Post your code to sell". Inside this rectangle is a form with the following fields: Platform (a dropdown menu), Name (a text input), Image (a text input with a placeholder "http://..."), Price (a text input), Genre (a text input), and Description (a larger text input). At the bottom of the form is a black button labeled "CREATE". The footer of the page says "Back-End 2023".

Delete an Offer (Logged-in User, Owner of the Current Offer)

Each **owner** of a game offer must be able to click on the **[Delete]** button and **delete the current game** from the database. After successful deletion, they must be redirected to the **All Games Code** page.

Edit Offer (Logged-in User, Owner of the Current Offer)

Each **owner** can edit their **game offers**. Clicking the **[Edit]** button for a specific offer on the details page should display the **edit page**. It contains a form with input fields for all the relevant properties that must be **prepopulated** with the game data. If the edit is successful, redirect the user to the **details page of the current game**.



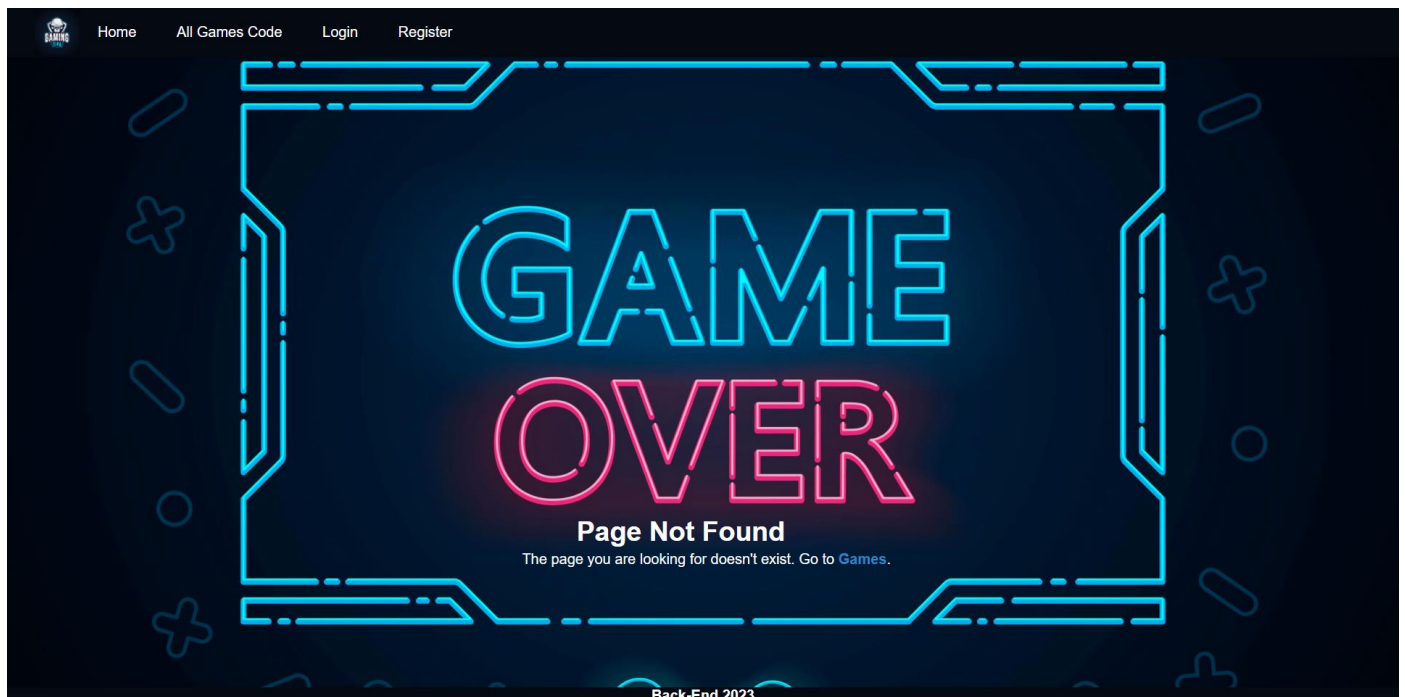
The screenshot shows the same web application as before, but with the "Edit Game Code Offer" form displayed. The form is prepopulated with the following data: Platform (XBOX), Name (Disney Dreamlight Valley), Image (https://assets-prd.ignimgs.com/2022/06/28/disney-dr...), Price (38), Genre (Simulator), and Description (Disney Dreamlight Valley is a hybrid between a life-sim and an adventure game rich with quests, exploration, and engaging activities featuring Disney and Pixar friends, both old and new. Once an idyllic land, Dreamlight Valley was a place...). At the bottom of the form is a black button labeled "EDIT". The footer of the page says "Back-End 2023".

6. Security Requirements / Routes Guards (10 Pts)

The **security requirements** are mainly **access** requirements, i.e., configurations about which users can access specific functionalities and pages.

- Guests (not logged-in) can access the home page.
- Guests (not logged-in) can access the login page and functionality.
- Guests (not logged-in) can access the register page and functionality.
- Guests (not logged-in) and users (logged-in) can access the **All Games Code** (where all game offers are listed).
- Guests (not logged-in) can access the **details** page without functionality.
- Logged-in users can access the **home** page.
- Logged-in users can access the **details** page and functionality.
 - **Logged-in users** (not the owner of the current offer) can **buy the game code**.
 - **The logged-in owner** of the current game offer can **edit** and **delete** it.
- Logged-in users can access **Create Game Offer** page and functionality.
- Logged-in users can access **logout** functionality.

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



7. Validation and Error Handling (10 Pts)

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

In case of an error, you should display the `<div>` with class **"errorContainer"**.

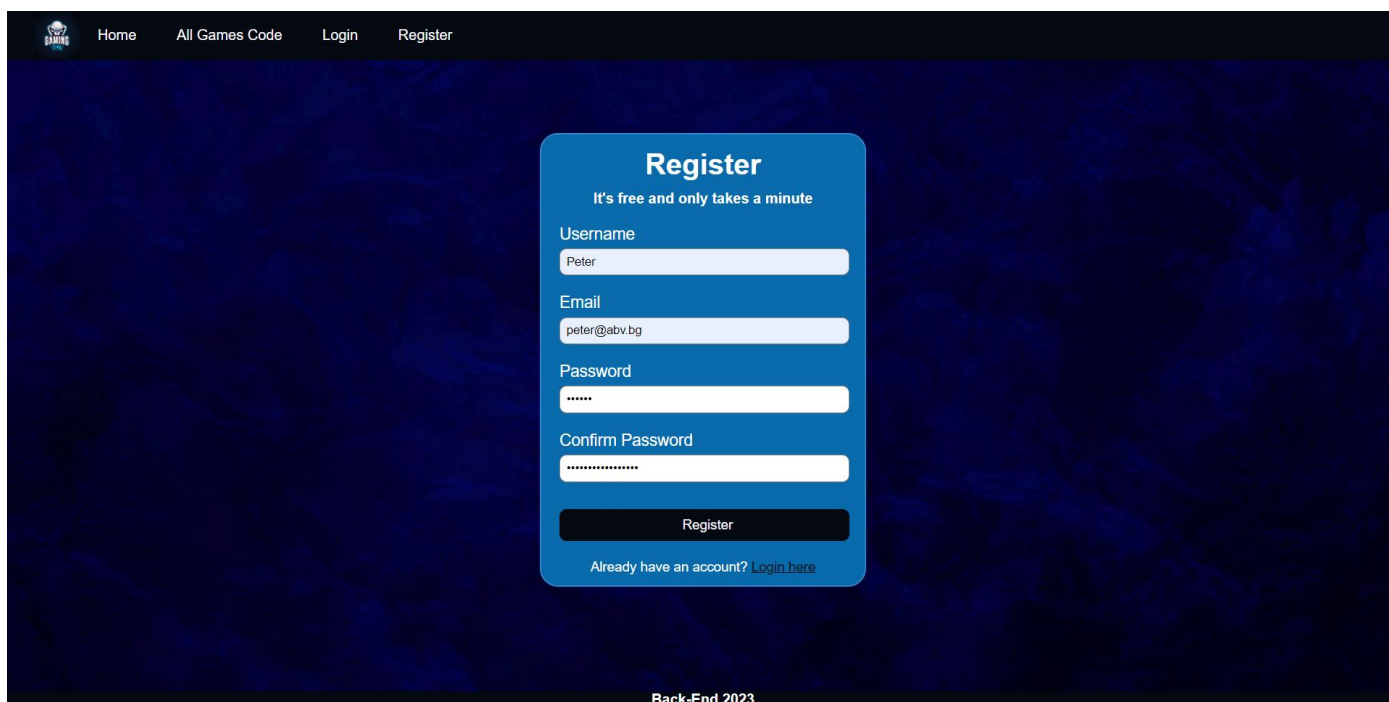
You can choose to display the first error or all of them. You have complete freedom to choose the content of the error message you will display.

Login & Register

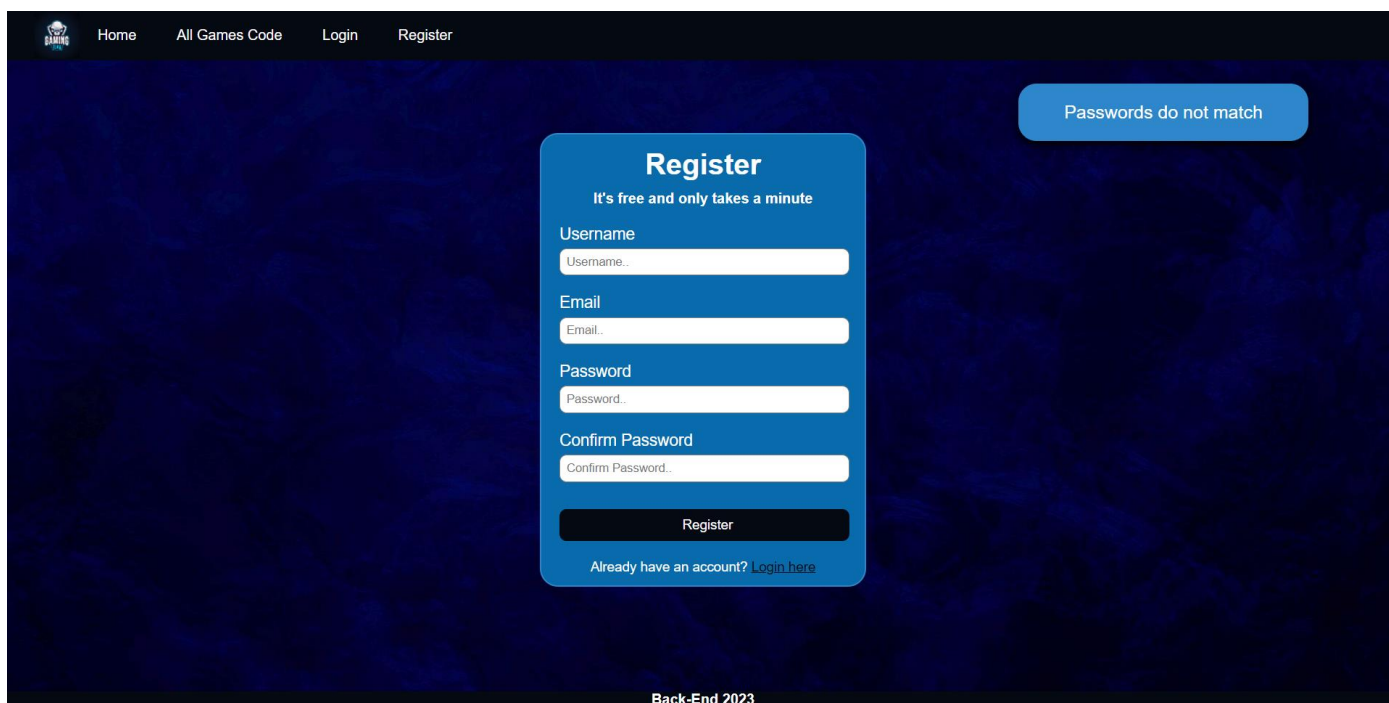
You should make the following validations:

- The **username** should be **at least five characters** long.

- The **email** should be **at least ten character** long.
- The **password** should be **at least four characters** long.
- The **password confirmation** should be **equal to the password**.



The screenshot shows a web application with a dark blue background and a black navigation bar at the top containing a logo and links for Home, All Games Code, Login, and Register. In the center is a light blue 'Register' form. The form includes the text 'It's free and only takes a minute' and four input fields: Username (filled with 'Peter'), Email (filled with 'peter@abv.bg'), Password (filled with six dots), and Confirm Password (filled with ten dots). A black 'Register' button is at the bottom of the form, with a link 'Already have an account? Login here' below it. The footer of the page reads 'Back-End 2023'.



This screenshot shows the same 'Register' form as above, but with an error state. A light blue rounded rectangle in the top right corner displays the message 'Passwords do not match'. The input fields are now empty, showing placeholder text: 'Username..', 'Email..', 'Password..', and 'Confirm Password..'. The 'Register' button and the 'Login here' link remain at the bottom of the form. The footer still reads 'Back-End 2023'.

Create Game Code Offer
Post your code to sell

Platform:

Name:

Image:

Price:

Genre:

Description:

CREATE

Path `description` is required.

Back-End 2023

8. * Bonus – Search (Logged-in Users) (10 Pts)

Search for a specific game. When you first access the search page, you should be able to see all the game offers. Filter all matches by name and platform (case-insensitive). If search fields are empty show all games.

Search

Search here... **Search**

Disney Dreamlight Valley
Platform: PS5
Genre: Simulator
Price: \$38
Details

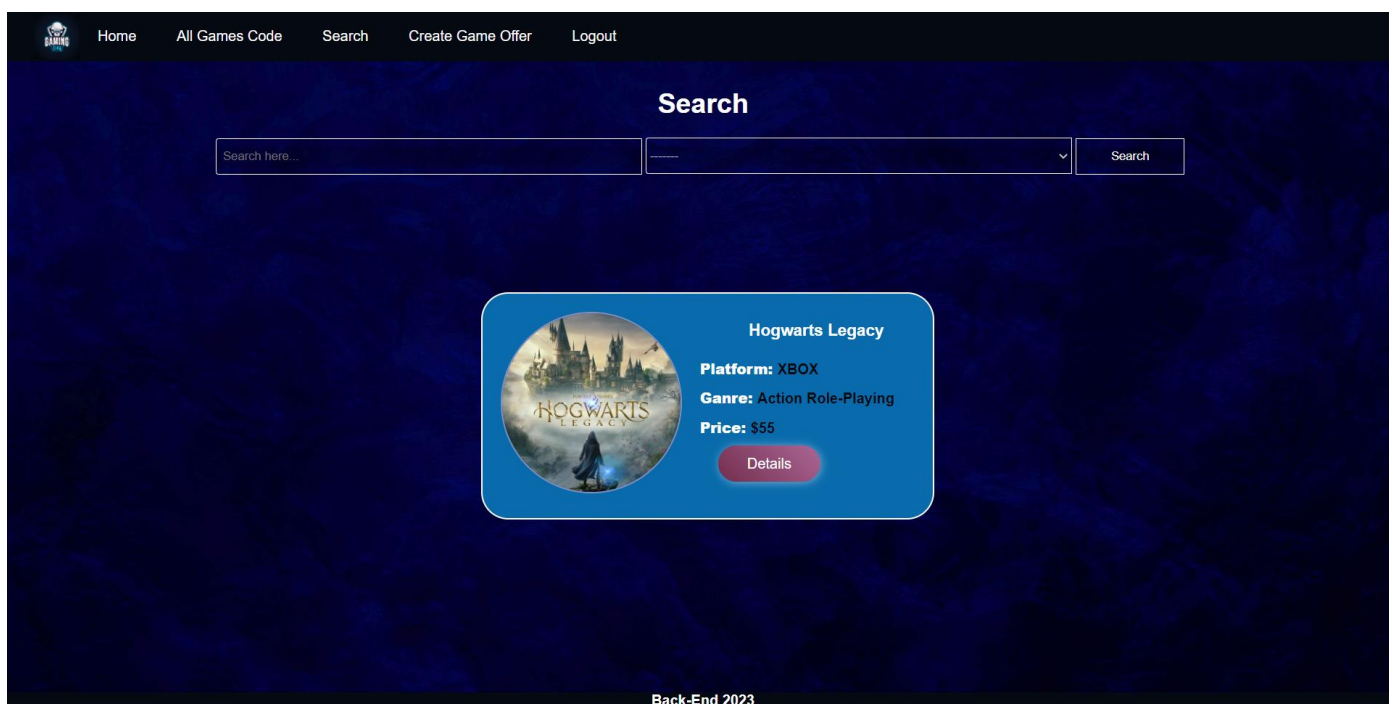
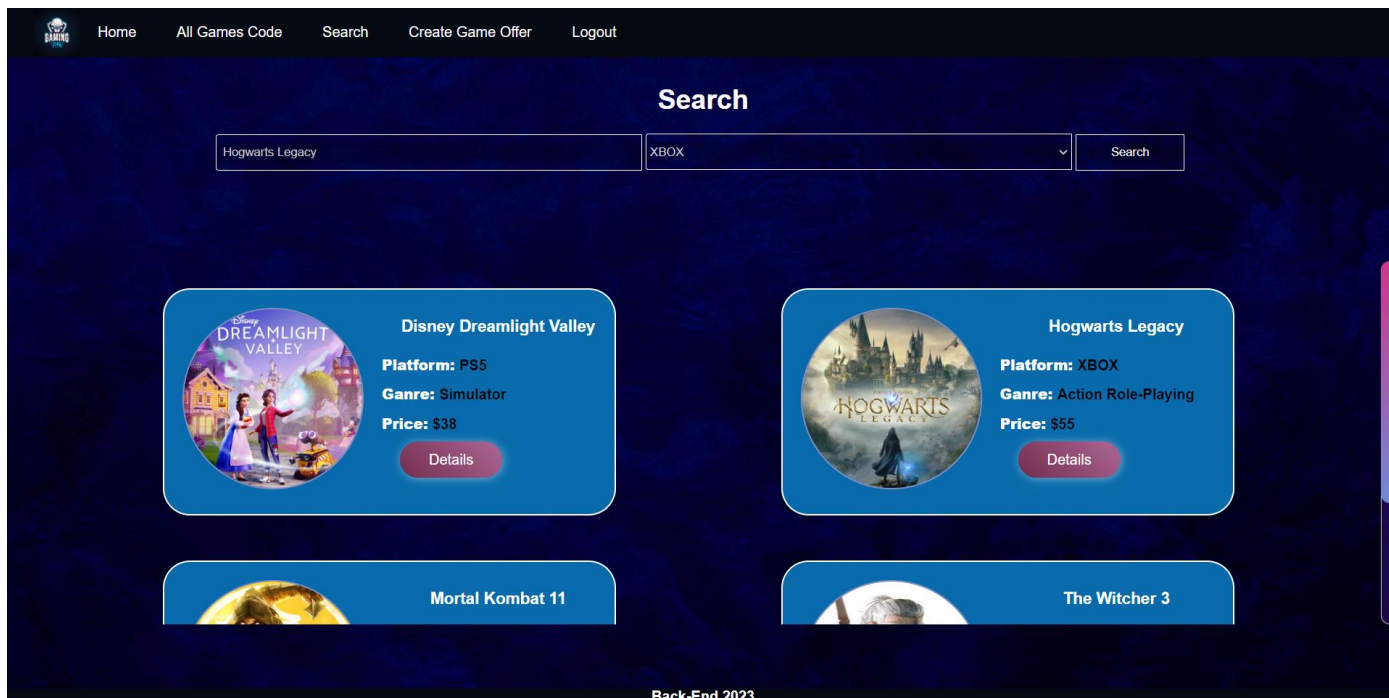
Hogwarts Legacy
Platform: XBOX
Genre: Action Role-Playing
Price: \$55
Details

Mortal Kombat 11

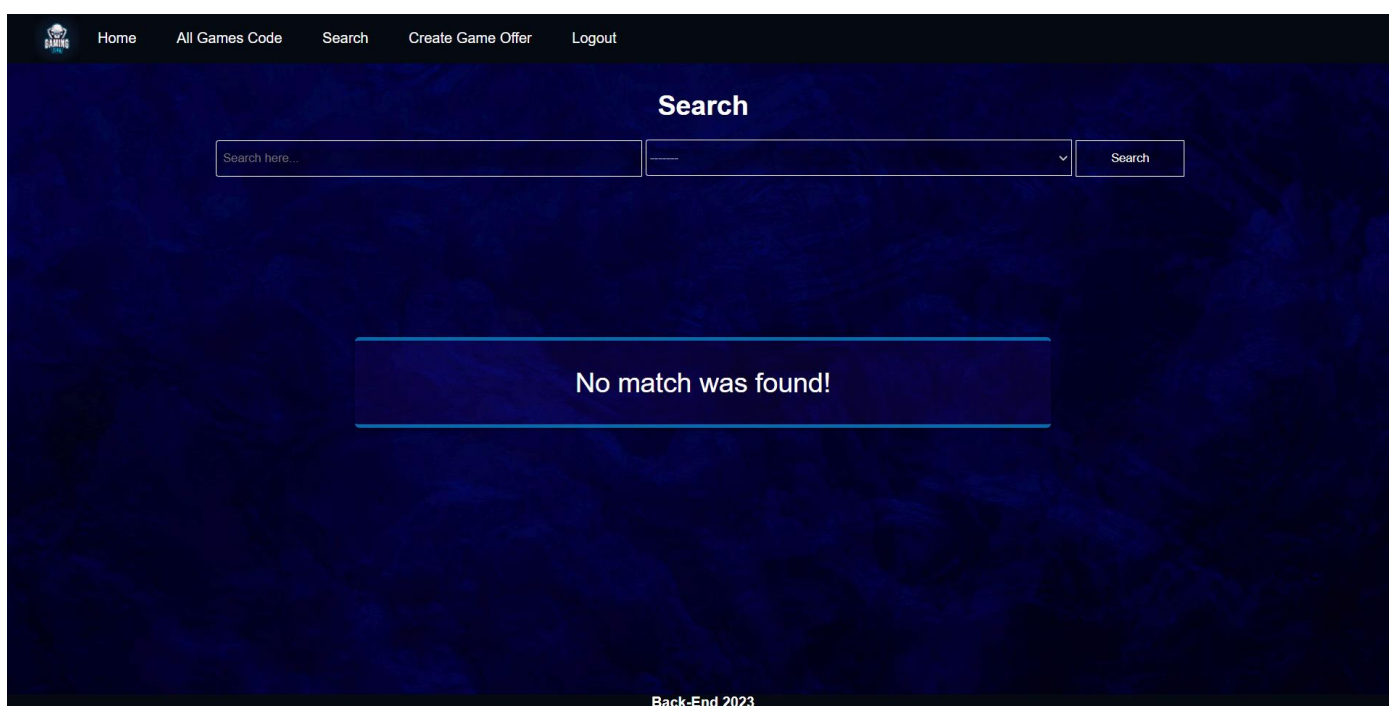
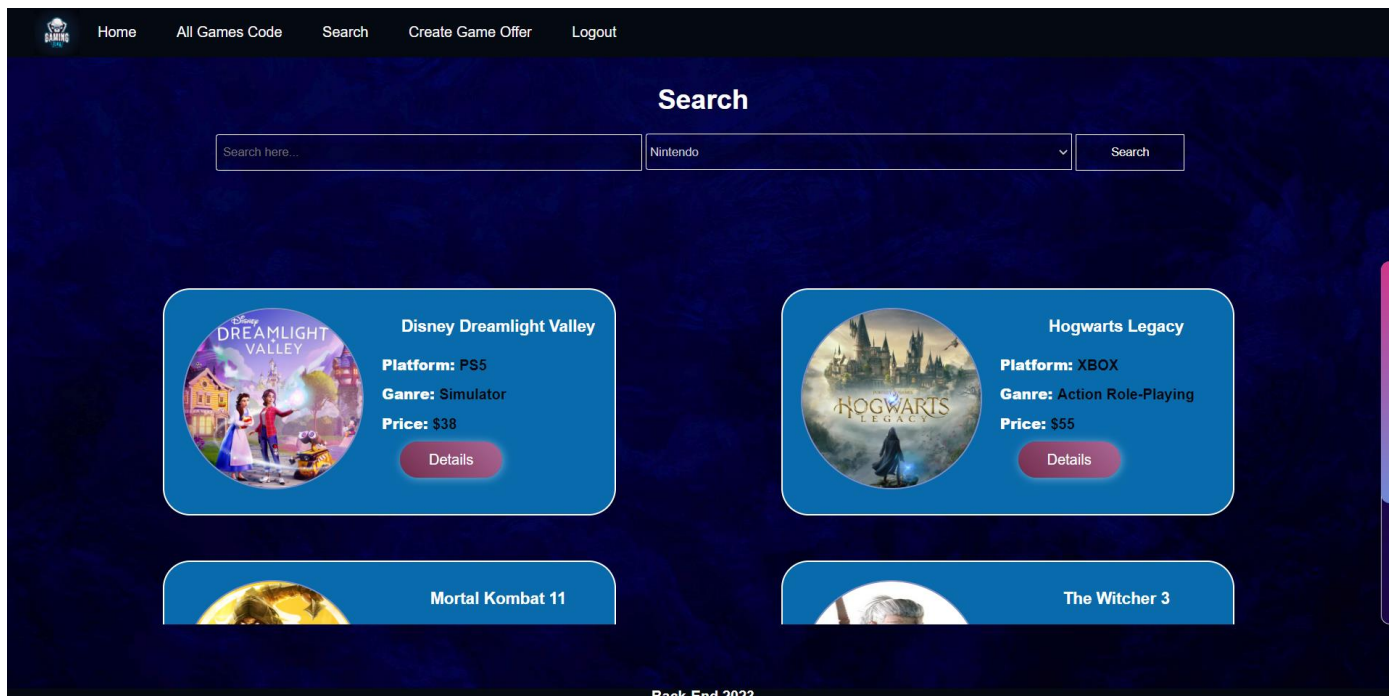
The Witcher 3

Back-End 2023

After clicking on the **[Search]** button, all the matching games (if there are any) should be displayed.

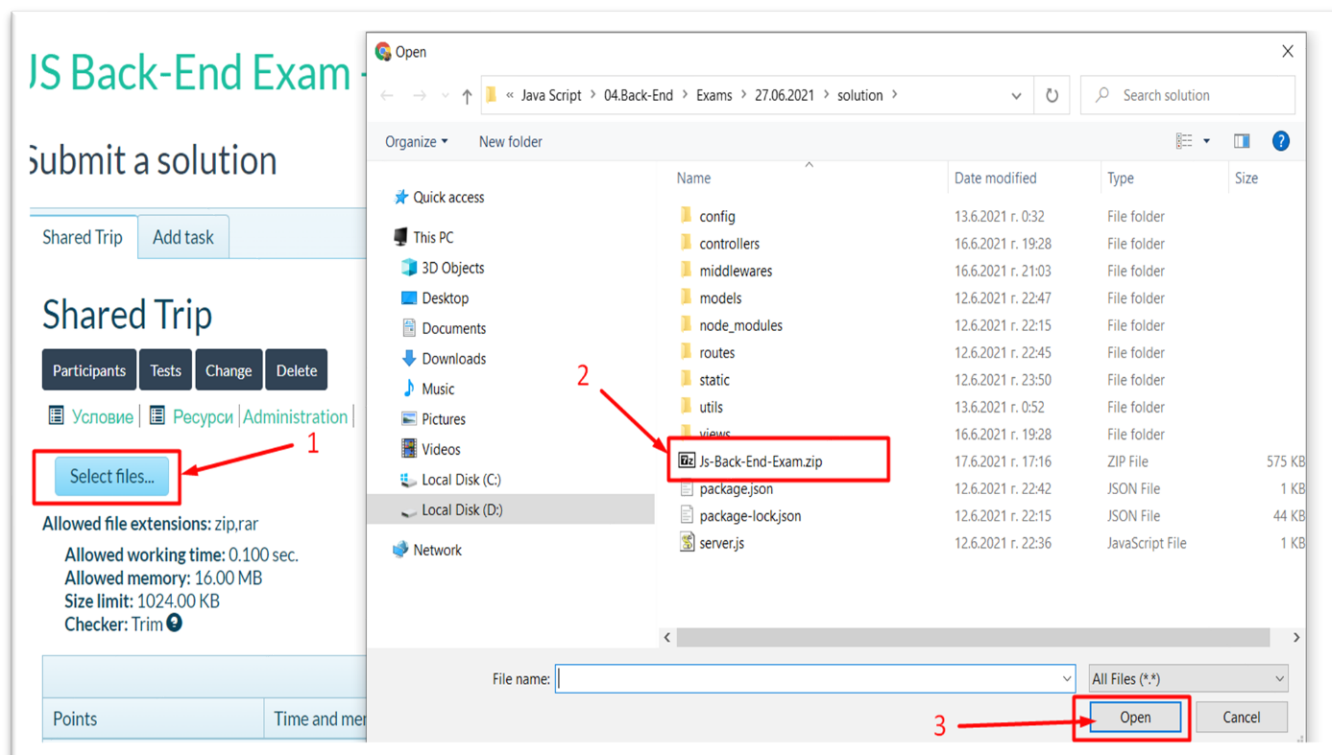
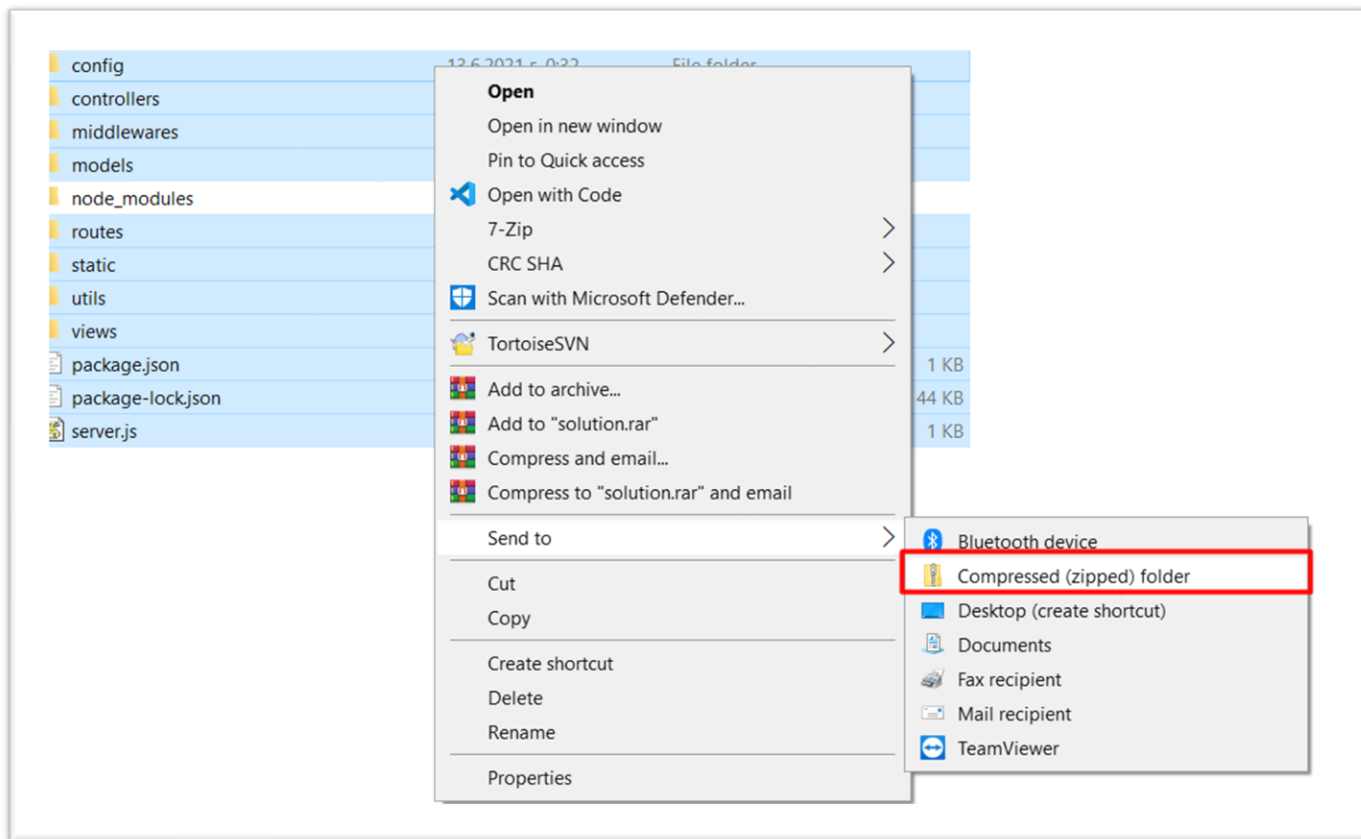


If there are **no** search matches, display:



9. Submitting Your Solution

Select the content of your project folder. Exclude the `node_modules` folder. Archive the rest in a **ZIP** file. Upload the archive to Judge.



Shared Trip

Add task

Shared Trip

Participants

Tests

Change

Delete

Условие

Ресурсы

Administration

Select files...

Js-Back-End-Exam.zip

Allowed file extensions: zip,rar
 Allowed working time: 0.100 sec.
 Allowed memory: 16.00 MB
 Size limit: 1024.00 KB
 Checker: Trim

File upload

Submit

Shared Trip

Participants

Tests

Change

Delete

Условие

Ресурсы

Administration

Select files...

Allowed file extensions: zip,rar

Allowed working time: 0.100 sec.

Allowed memory: 16.00 MB

Size limit: 1024.00 KB

Checker: Trim

File upload

Submit

Problem results

Participant	Result
shampion	0 / 100

Submissions

Points	Time and memory used	Submission date
Compile time error		17:24:04 17.06.2021

This is not a problem !

GOOD LUCK! 😊