

|  |  |  |  |
| --- | --- | --- | --- |
| Academic Year | 2023 | | |
| Semester | Fall | Winter | Summer |
| Course Code - Name | WEB 422 - Web Programming for Apps and Services | | |
| Instructor | Dr. Razi Iqbal | | |
| Assessment | Assignment 3 |  | |
| Deadline | Wednesday, December 06, 2023 | | |

**Student ID**

**Student Name**

**Assignment 3**

The main purpose of this assignment is to test your knowledge of Angular framework. This assignment assumes that you have already setup your environment.

**Instructions:**

* This assignment is 15% of your total marks in this course.
* For the submission of this assignment, you are required to upload your angular app on any cloud, e.g., Google Drive, Dropbox or even GitHub and share the link as a submission on Blackboard. While submitting, no need to submit node\_modules folder.
* Images required for this assignment are similar to assignment 1, so you should already have them.
* Submissions through emails will not be accepted.
* Students having exactly similar code will get a straight 0.
* The deadline for submission of this assignment is midnight Wednesday, December 06, 2023.

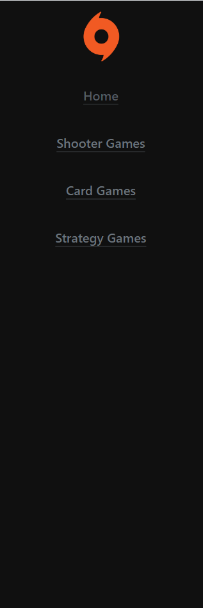
**Question**

In this assignment, you are expected to create a Game Portal in Angular. The portal is expected to show games available on various consoles, e.g., PC, PlayStation, Xbox and etc. You should be fetching the data for games from an API. Your website should allow users to search games by their titles and should be able to find games by category, e.g., Shooter, Card or Strategy games.

This assignment requires you to use Angular framework. You are free to use Bootstrap. Make sure you keep each component in their separate modules. You should have at least the following angular components:

**Design**

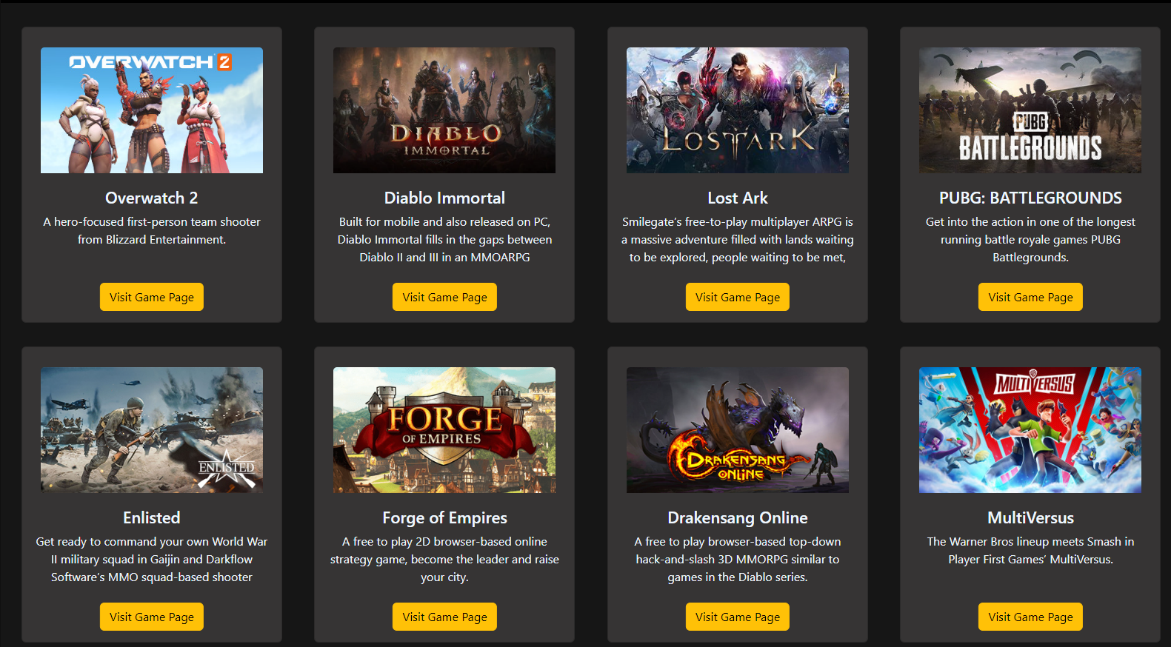
* SideBar: Provided users to navigate through your website. Should have at least Home, Shooter Games, Card Games and Strategy Games links. Clicking on these links to take user to their specific category games. Home should bring them back to the home page where all games are shown. Below is the expected output of this component. You are free to use any free images as required.



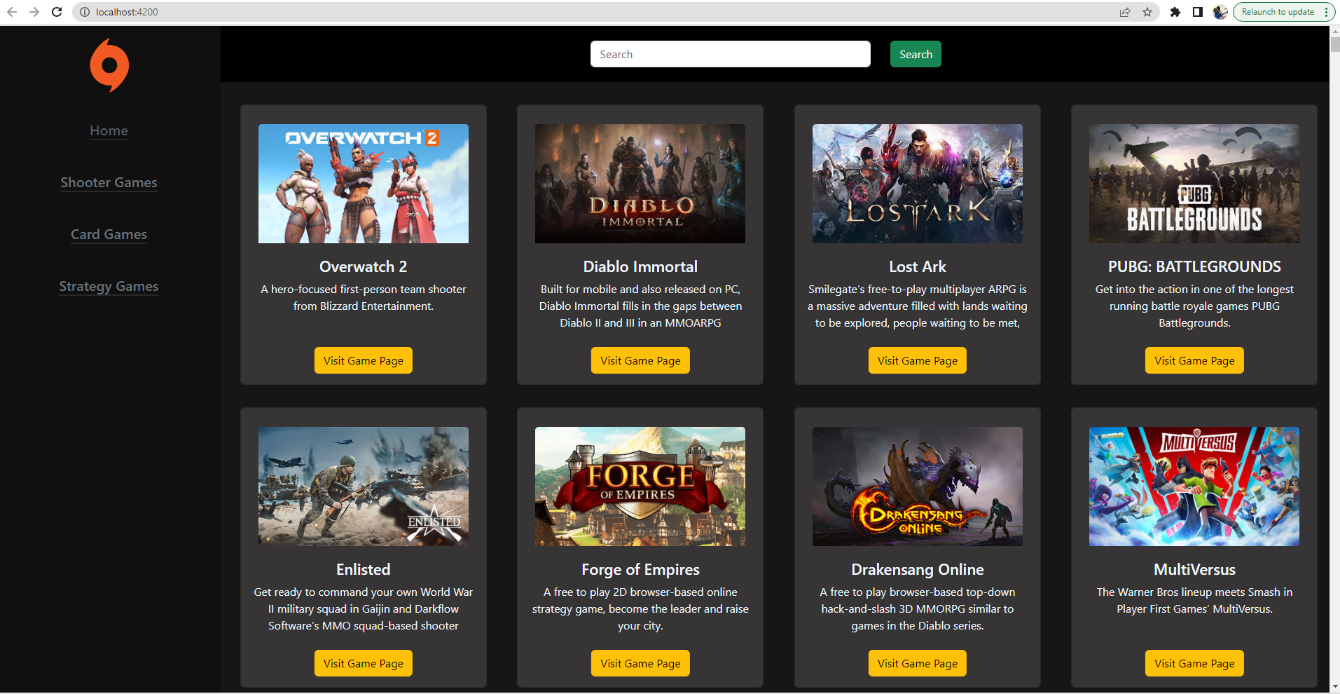
* Search: Should have an input and a button to it that allows user to end game titles and on click of the Search button, results should appear based on the matching titles. Below is the expected screenshot:



* Content: A content area that shows all the games, searched games, Shooter Games, Card Games or Strategy Games. This area should nest into it the below CardComponent:
  + Card Component: Should show all the details about the game
    - Image
    - Title
    - Short Description
    - A Button that takes you to the URL of this game
  + For CardComponent, make sure you reuse the components instead of copy pasting and duplicating the code. *Failing to reuse the components might result in losing the marks.*



Finally, call these components accordingly in the app component to show below the final output of this design. You are free to add more components as you like:



**Functionality:**

Once you have completed the design of the website, you now have to move to the overall working of this website. Make sure the website has the following functionality:

* API Call: You are required to create an angular service that fetches data from an API. For this purpose, you have to go to RapidAPI and signup for an account. You can use the following link (https://www.freetogame.com/api/games) to review the content of the API. Use the following code to fetch the data about games from the API by using the following code (make sure to replace API\_Key with your own API\_Key from RapidAPI).:

getGames(): Observable<Game[]> {

let headers = new HttpHeaders()

.set(

'x-rapidapi-key',

'YOUR\_API\_KEY\_GOES\_HERE'

)

.set('x-rapidapi-host', 'free-to-play-games-database.p.rapidapi.com');

return this.http.get<Game[]>(

'https://free-to-play-games-database.p.rapidapi.com/api/games',

{ headers }

);

}

* Interface: Do not forget to create an angular interface to get all the details of the games as below:

export interface Game {

id: string;

title: string;

thumbnail: string;

short\_description: string;

game\_url: string;

genre: string;

platform: string;

publisher: string;

developer: string;

release\_date: string;

freetogame\_profile\_url: string;

}

Once data is fetched, place them into their corresponding components inside the CardComponent.

* Components: Make sure to have appropriate components to hold data to be shown in the Content component, e.g., Home, Shooter Games, Card Games, Strategy Games. You might want to create some more functions in the service to get specific games based on their genre (check interface for this data member). You might want to look into pipe, filter and map functions in Typescript that can help you in filtering an array.
* Routing: An important component of this assignment is routing. Make sure you carefully create routes for links like Home, Shooter Games, Card Games, Strategy Games. For example, clicking on Home on the SideBar should take you to home page of this website etc. Try to reuse the Content component in each of these components to avoid rewriting the code.
* Searching: Your website should be able to allow searching of games by their title. You might want to create a SearchComponent which fills in the Content component when user inputs a title in the input box and clicks the button. The Content component should update to show only the titles that have the input string in them. For example, when user inputs “Game” in the search box and hits Search button, below should be the expected output:

