## CMPS 350 – Web Development Fundamentals
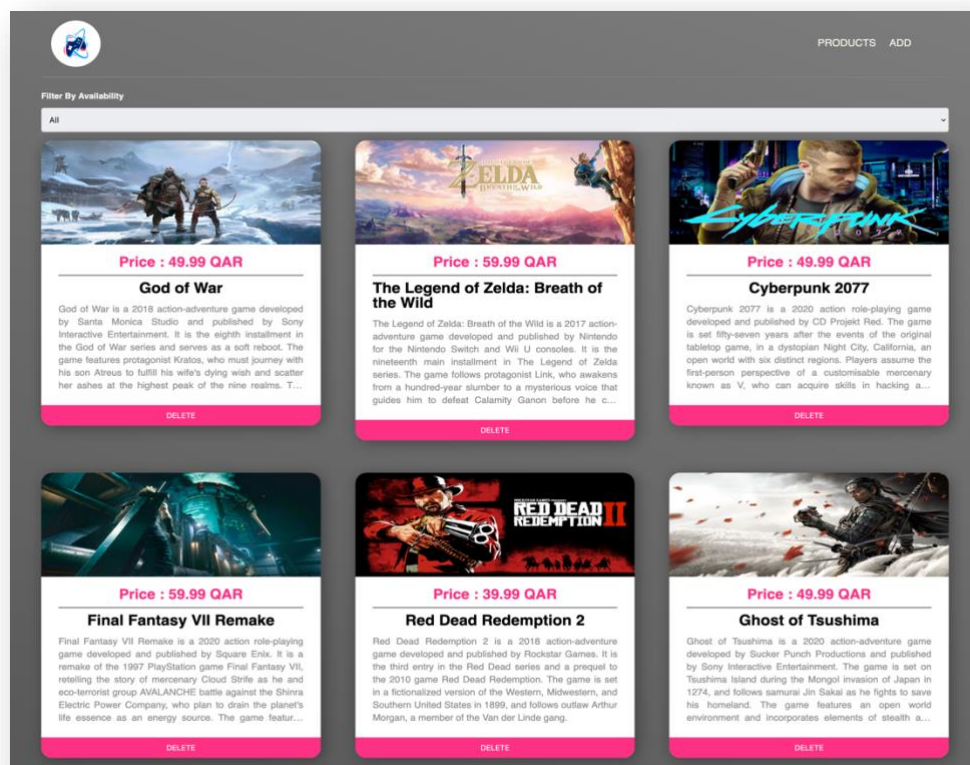
## Practical Midterm Exam

- The exam duration is 120 minutes. So, read the exam questions carefully and plan your time accordingly.
- Push your code to GitHub regularly (at least every 30 minutes) to avoid unpleasant surprises, as your computer might hang!
- The Exam is an open book. In case of plagiarism, both parties will receive 0 points. Hence do not share or receive any code from anyone.
- Once you complete the exam, you should:
    - Add a screenshot for each question to the provided testing sheet.
    - Push your code and testing sheet to your GitHub repo under **midterm** subfolder.
    - Demo your work before leaving the Exam.

### <ins>Gaming Store App</ins>

In this exam, you will apply your frontend web development skills to build a gaming web app that displays gaming products in a grid. The UI design is provided in the figure below and a demo of the app will be shown at the start of the exam.
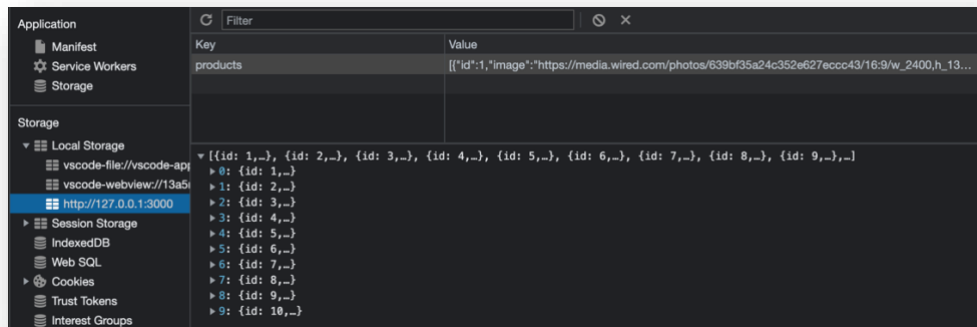
### <ins>Implementation Tasks:</ins>

1. **[30 pts]** When the page loads **Get gaming products** from <ins>the following API</ins> and display them in a grid of cards as shown in the image below. The game product card should display the `Product Image`, `Product Name`, `Description`, `Price`, and `Availability`.
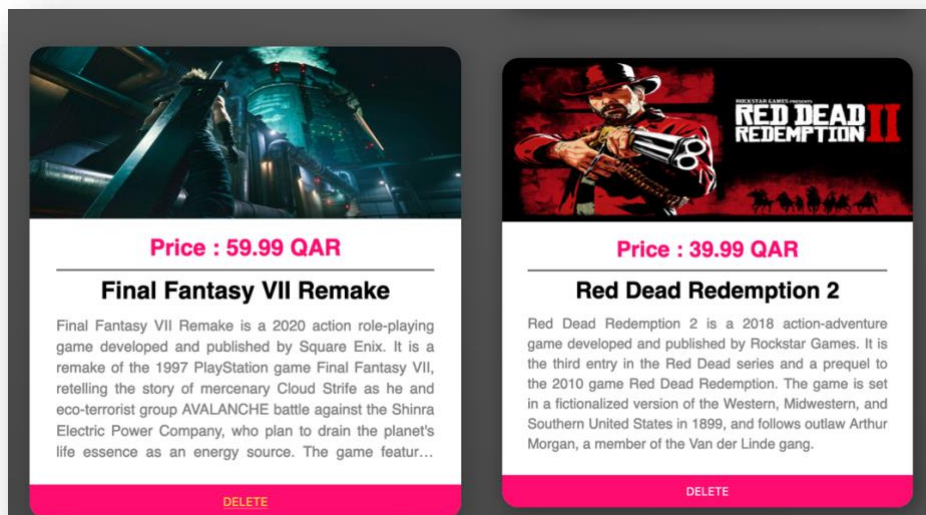
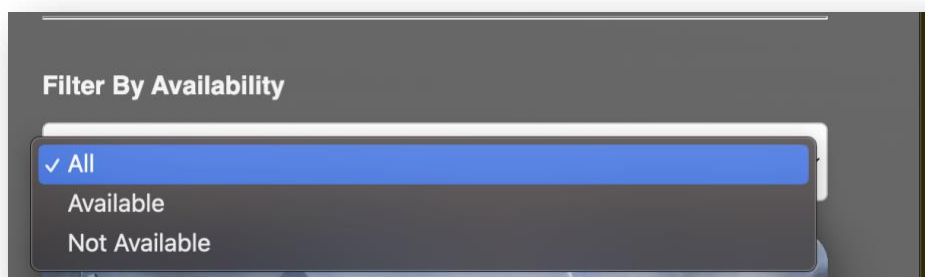2. **[15 pts] Caching Products in the Local Storage**
   Improve your app performance by caching the products data in the local storage. When you get the products data, save them to the local storage. Subsequent visits should first read the data from the local storage. Then only fetch from the remote Web API if the products data is not available in the local storage.
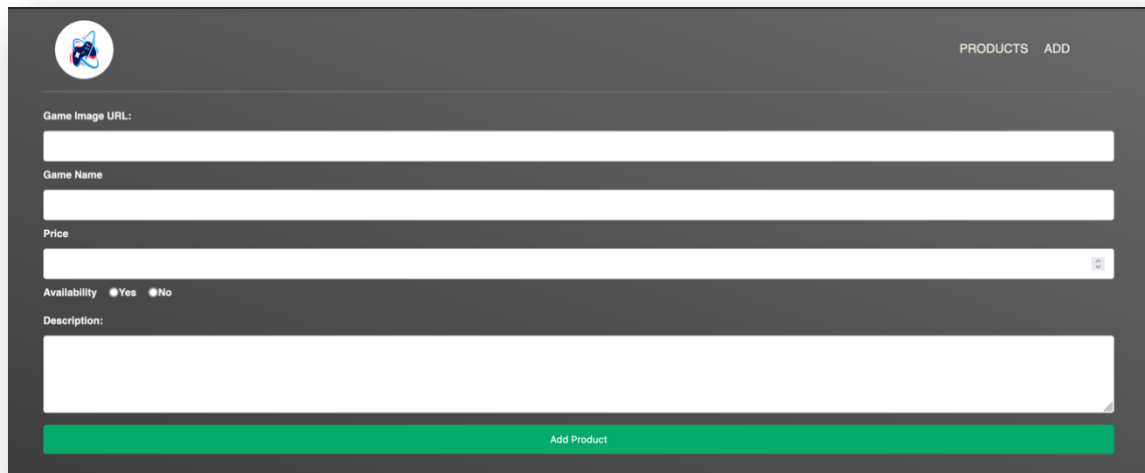


3. **[15 pts] Delete Product**: add a "Delete" button to each card. When the user clicks it for a specific product, the product should be removed from the grid.



4. **[15 pts]** Add **Filtering** ability to allow sorting products by availability. Use a drop-down to select the availability option.

5. **[25 pts] Add Product** form to the page that allows users to add a new gaming product. The form should have input fields for [Product Name, Description, Price, and Availability]. When the user clicks the "Add Product" button, the new product should be added to the grid.