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.

Online Store App

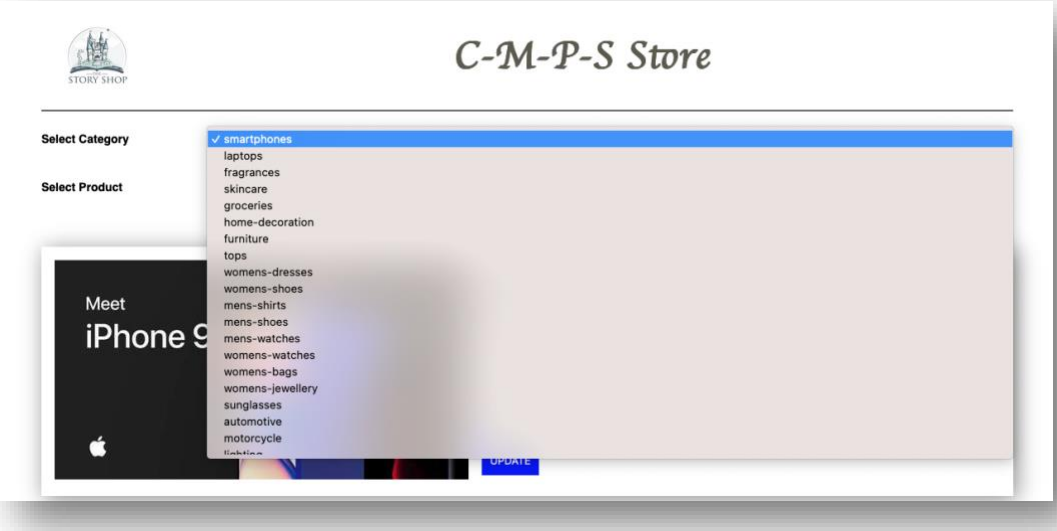
In this exam, you will apply your front-end web development skills to build an online store web app that allows users to browse available products. The UI design is provided in the figure below, and the app's demo will be shown at the start of the exam.

Useful Links

- To access the list of all categories, use this link: https://dummyjson.com/products/categories.
- To retrieve all the products listed under a specific category, visit this link: https://dummyjson.com/products/category/smartphones.

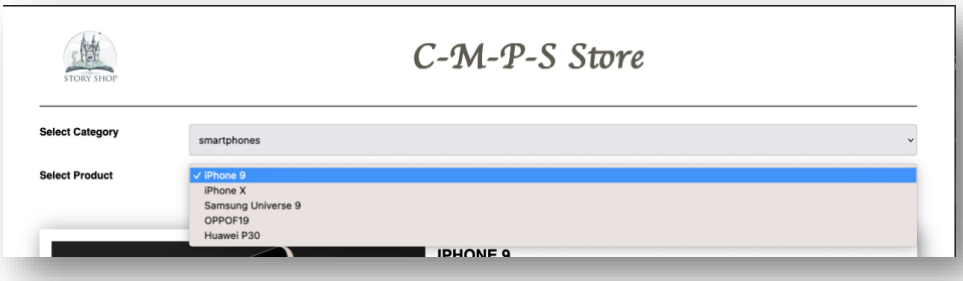
Implementation Tasks:

1. **[15 pts] Retrieve Categories Data**:
   When the page loads, your app should request the category data from the API provided. Once you receive the categories data, you should populate a dropdown list with them, as shown in the below figure.

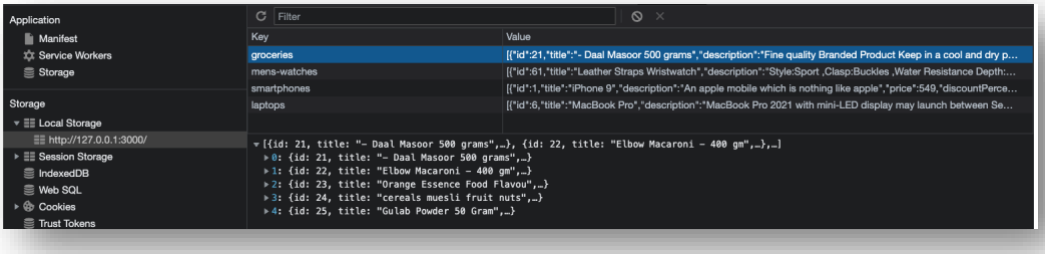


[15 pts] Retrieve Products Data:

Once the user selects a category from the first dropdown list, your app should request all the products for that specific category from the API provided. Once you receive the products data, you should populate a second dropdown list with the products that belong to that category, as shown in the figure below.



2. **[15 pts] Caching Product Data in Local Storage**
To improve your app's performance, you should implement a caching mechanism that stores the products data for the selected category in the local storage. This means that if the user visits the page again, your app should first check if the products data for the selected category is stored in the local storage. If it is, your app should retrieve the data from the local storage instead of requesting it from the API. **Only if the product's data is not available in the local storage**, should your app make a request to the API to retrieve it.



3. **[15 pts] Display Product Information**
When the user selects a product from the second dropdown list, your app should display the information of the selected product in a card format. This card should include the product's title, image, price, rating, stock as shown in the figure below.

4. **[25 pts] Update Product Details**:
   To allow users to update product information, your app should include an "Update" button within the product card. When the user clicks this button, a form should appear that allows them to modify the product's details, such as the **title**, **price**, **rating**, and all the other properties of the selected product object.

   Once the form loads, you should pre-populate the form fields with the current product data, so the user can easily see what they are editing. When the user clicks the "Save" button, your app should update the product details that is saved in the localStorage and navigate the user back to the home page.