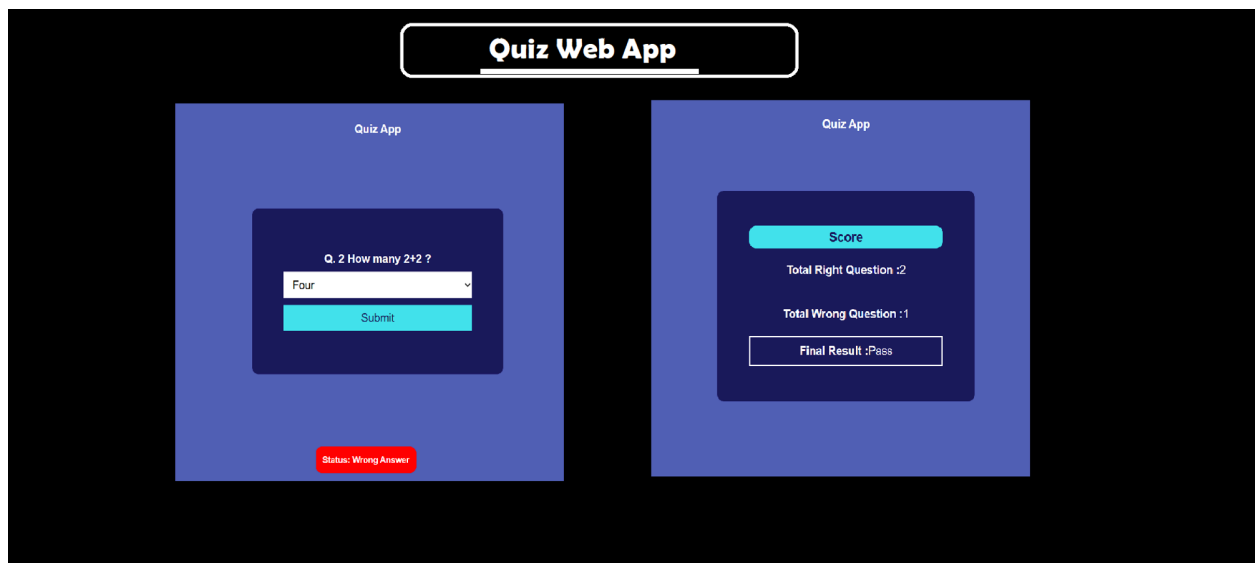


# Assignment Questions 7

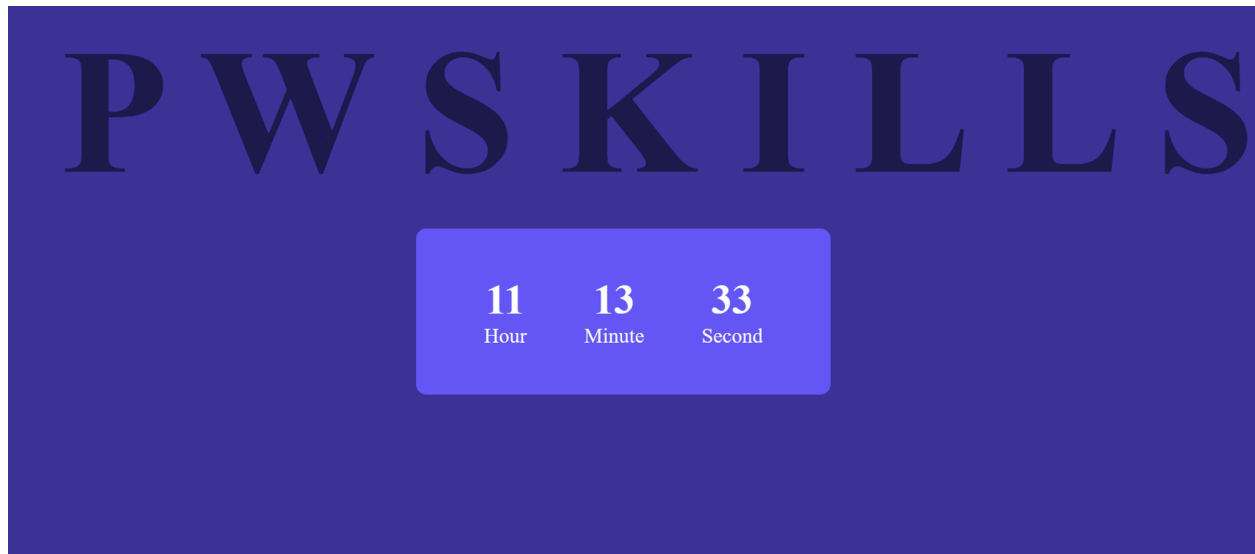
💡 **Q.1** Develop a captivating quiz web application with an engaging user interface using HTML, CSS, and JavaScript, incorporating a dynamic leaderboard feature to track and display the scores of participants in real-time with following features

- When the user submits an answer, display a popup to indicate whether the answer is right or wrong as shown in the image.
- Provide immediate feedback to the user, enhancing their engagement and learning experience.
- Once the user answers all the questions, present them with a leaderboard.
- The **leaderboard** should display the number of questions the user answered correctly and incorrectly.
- Display the user's overall result, showcasing their performance in the quiz.

**Note:** The overall result and leaderboard format should be designed similar to the image provided below.

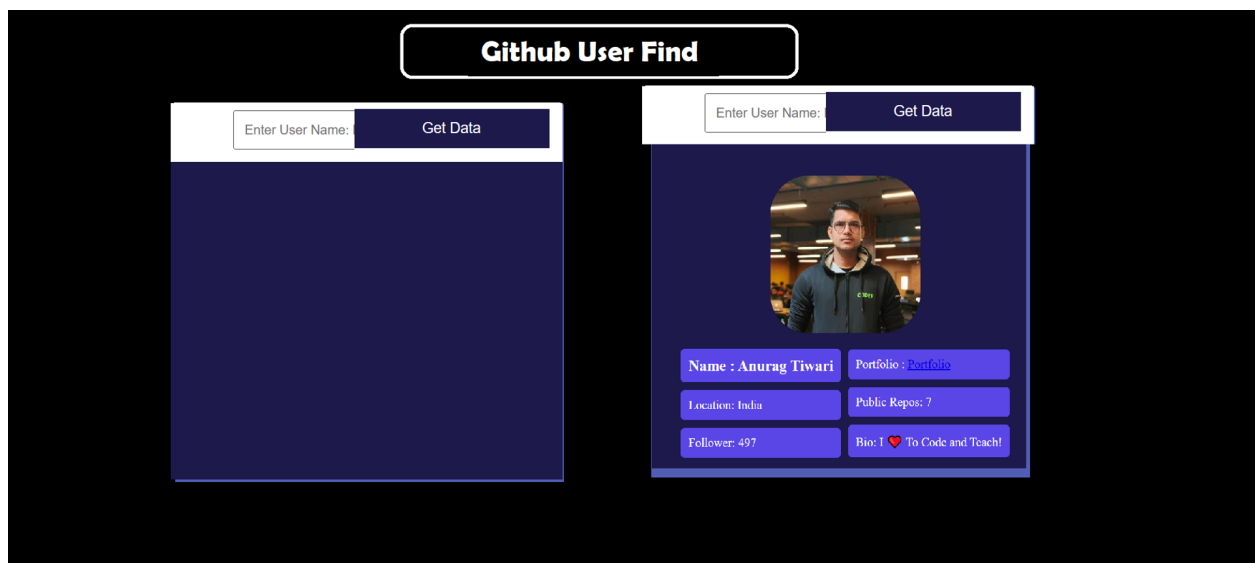


💡 **Q.2** Create an elegant and functional digital clock using HTML, CSS, and JavaScript, where the current time is displayed with precision and style as shown in the below image



💡 **Q.3** Develop a **GitHub User Finder** web application using **HTML**, **CSS**, and **JavaScript**. The application should allow users to enter a **GitHub username** and display relevant information about the user, including their avatar and name. The design of the application should follow the layout provided in the image below.

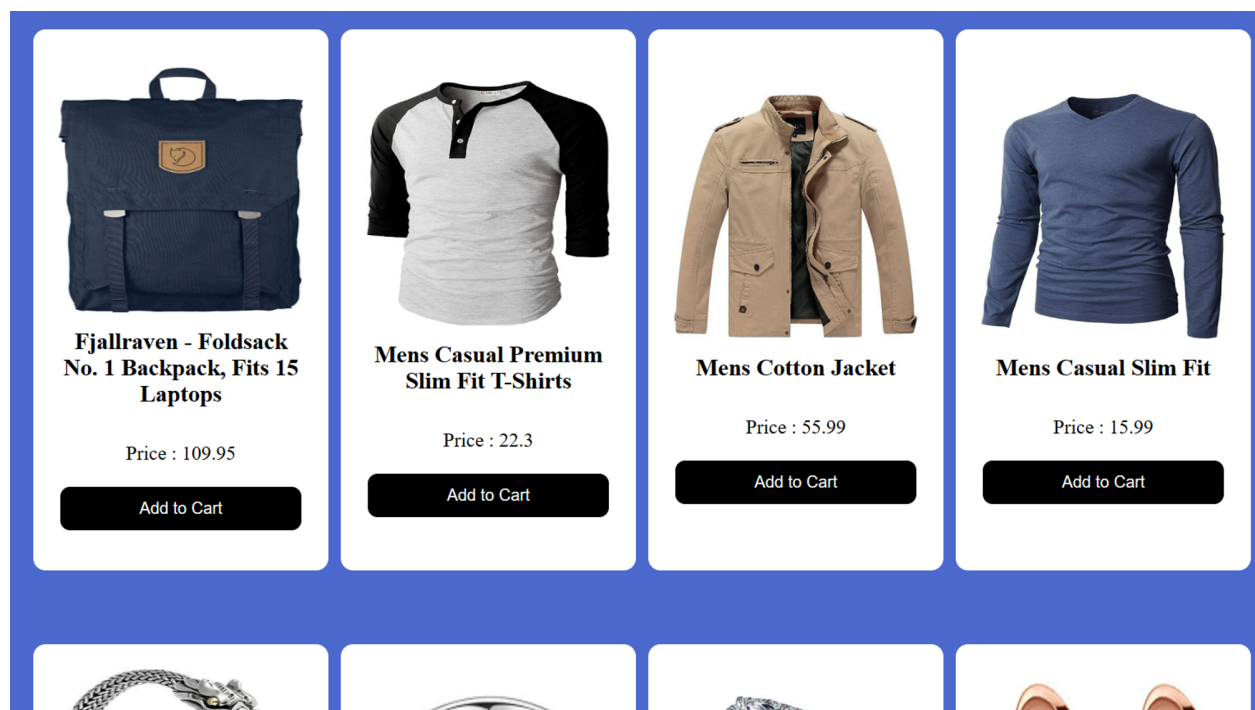
- Use Github Api to get User Data "<https://api.github.com/users>"



💡 **Q.4** Develop a simple website using **HTML**, **CSS**, and **JavaScript** to fetch and display products from the "<https://fakestoreapi.com/products>" API. The website should have the following features:

- Fetch product data from the "<https://fakestoreapi.com/products>" API.
- Display the products in a user-friendly UI.
- Show **four products** in a single row for optimal visibility and layout.
- Use **HTML, CSS, and JavaScript** to achieve the desired layout and functionality.
- Ensure that the UI is visually appealing and responsive.
- Implement error handling to handle any issues with API requests.
- Test the website thoroughly to ensure proper functionality and performance.
- Provide clear and concise documentation to guide users on how to interact with the website and understand its features.

**Note:** Refer to the provided image or design specification for the desired layout and styling of the product display in a single row.



💡 **Q.5** Extend the previous website that fetches products from the <https://fakestoreapi.com/products> API using HTML, CSS, and JavaScript. Implement the following additional functionalities and features:

1. Integrate an 'Add to Cart' functionality:
  - When a user clicks on the 'Add to Cart' button on a product card, the selected product should be added to the cart.
  - Use JavaScript to handle the 'Add to Cart' functionality.
2. Display the Cart UI:
  - Create a separate section or modal that displays the products added to the cart.

- The cart UI should show the added product cards, including their image, name, price, and quantity.
- Ensure that the cart UI is visually appealing and user-friendly.

**Note:** Refer to the provided image or design specification for the desired layout and styling of the cart UI.



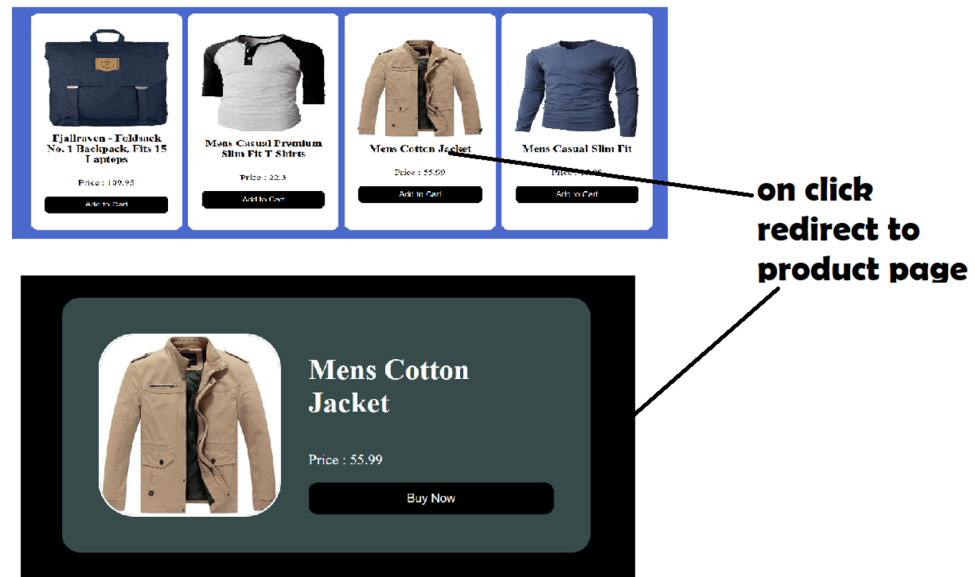
Cart Item



💡 **Q.6 Extension to Previous Problem\*\*,\*\*** The task is to create a single product page with the following functionalities:

- On clicking on any product, the user should be redirected to a single product page.
- The single product page should display the data specific to the clicked product.

Hint : Use LocalStorage to store data



💡 **Q.7** develop a live preview project using HTML, CSS, and JavaScript, consisting of two sections: a form section and a live preview section. The live preview section should dynamically display the information filled in the form section in real time. as shown in the below image

### Publish Your Story

TITLE OF THE STORY WILL BE HERE

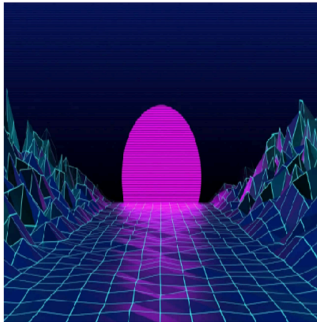
Enter Image URL

Enter Author Name

Category Eg Educational

Enter Your Story

Submit My Story



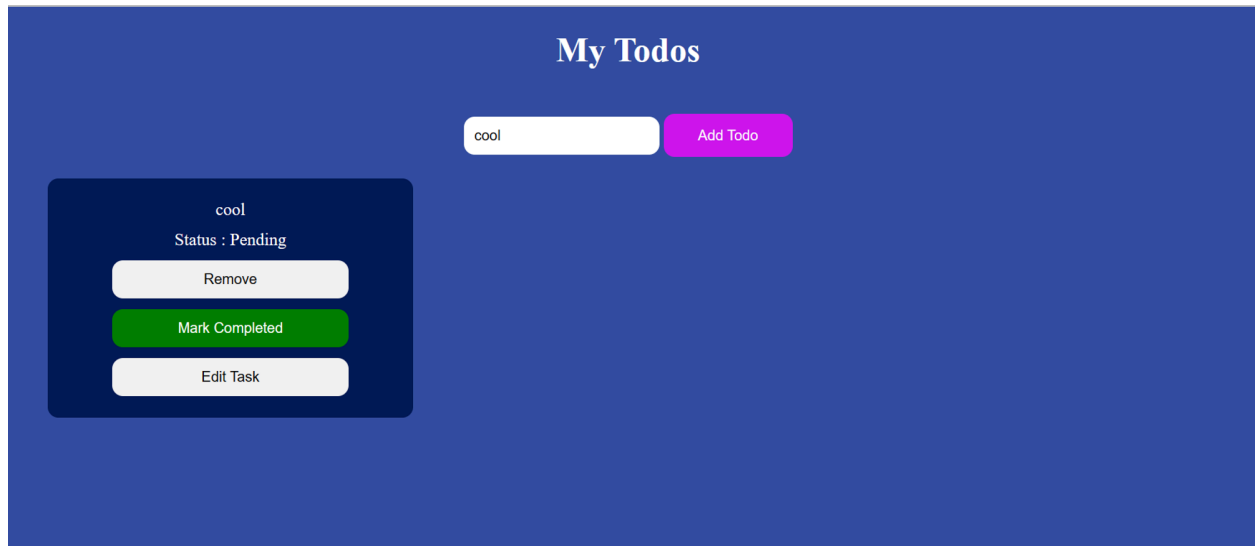
**TITLE OF THE STORY WILL BE HERE**

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Quod similique eaque accusantium aperiam optio necessitatibus, provident excepturi nostrum, et totam, numquam adipisci ex recusandae? Fugit beatae, autem temporibus veniam omnis voluptates aperiam nostrum, possimus repudiandae culpa ullam, necessitatibus ad totam. Doloremque sunt, accusamus ratione ipsa optio assumenda perspiciatis at! Incidunt rem laboriosam dignissimos! Debitis rem inventore nemo numquam a, facilis commodi in autem

Written By Farman

💡 **Q.8** Create a Simple Todo Web App with following features :

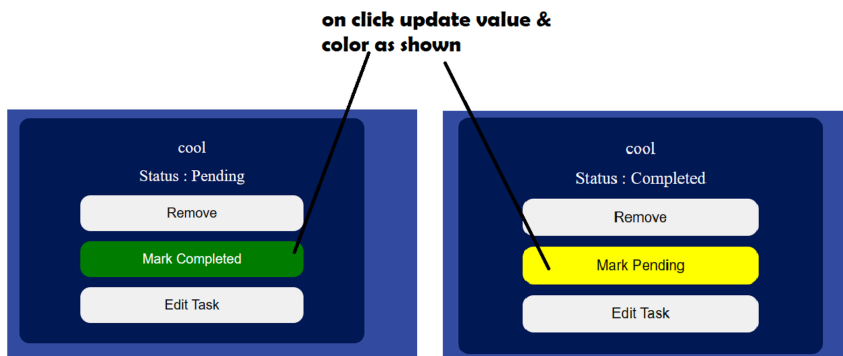
- An input button where users can type their tasks.
- On clicking the submit button, the entered task should be displayed in the UI.
- The UI layout should resemble the image provided.



💡 **Q.9** Expand the existing todo web app with two additional functionalities:

1. On clicking the "Update Status" button:
  - Update the status of the task.
  - Change the background color of the task to indicate the updated status.
2. On clicking the "Delete Todo" button:
  - Remove the selected todo from the UI.
  - Ensure that the deleted todo is no longer visible in the todo list.

See below **image** for **Reference**



💡 **Q.10** Assignment Extension: Add Edit Functionality to the Todo Web App

On clicking the "**Edit**" button for a specific todo:

- Display a Form where the user can update the data of the selected todo.
- Provide input fields or text areas in the modal to edit the todo's details.
- On clicking the "Update" button in the modal:
  - Update the data of the todo with the user's changes.
  - Ensure that the updated data is immediately visible in the UI, reflecting the changes made by the user.

See below image for reference

