**Front-End UI/UX Mini Project**

**Project Submission Template**

**1. Title Page**

* **Project Title**: *e.g., "TO-DO LIST APPLICATION"*
* **Submitted By**: 
  + *Team Members- Krishna Kumar*

*Aman Varun Ekka*

*Samvrith Singh K*

* + *Register Number- 2460394*

*2460493*

*2460441*

* + *College-E-mailid-* [*krishna.kumar@btech.christuniversity.in*](mailto:krishna.kumar@btech.christuniversity.in)

[*samvrith.singh@btech.christuniversity.in*](mailto:samvrith.singh@btech.christuniversity.in)

*aman.varun@btech.christuniversity.in*

* **Course**: *e.g., UI/UX Design Fundamentals*
* **Instructor Name**: *Dhiraj Alate*
* **Institution**:  *Christ University*
* **Date of Submission**: *26/09/2025*

**2. Abstract**

This project involved the development of a dynamic To-Do List web application using HTML, CSS, Bootstrap, and jQuery. The application allows users to add, edit, delete, and filter tasks by their completion status. The primary objective was to create an intuitive and responsive UI that enhances user productivity by efficiently managing daily tasks. The use of Bootstrap ensures a consistent responsive design, while jQuery facilitates interactive features without reloading the page. The final product is a practical productivity tool demonstrating front-end web development skills.

**3.Objectives**

* Design a user-friendly and visually appealing task management interface.
* Implement task CRUD operations (Create, Read, Update, Delete) with live UI updates.
* Provide filtering options for tasks: All, Completed, and Active.
* Use Bootstrap for responsive and consistent styling.
* Employ jQuery for DOM manipulation and event handling**.**

**4.Scope of the Project**

The project focuses on front-end development of a task management tool. It includes task addition, editing, deletion, and filtering functionalities. The application supports responsive design for optimal usage across desktops, tablets, and mobiles. There is no backend or database integration; tasks are managed in the browser session.

| **Tool/Technology** | **Purpose** |
| --- | --- |
| HTML5 | Markup structure and content. |
| CSS3 | Styling and layouts. |
| Bootstrap 4.5.2 | Responsive design and components. |
| jQuery 3.5.1 | DOM manipulation and events. |
| Visual Studio Code | Code editing and development. |
| Chrome Developer Tools | Debugging and testing. |

5.**Tools and Technologies Used**

**6.HTML Structure Overview**

The HTML utilizes semantic elements for clarity and accessibility. The sidebar navigation and main content areas are structured for easy user interaction. Bootstrap classes are used extensively for layout grids, buttons, forms, and typography to ensure a neat UI.

**7.CSS Styling Strategy**

The project applies custom CSS alongside Bootstrap for personalized styling. A gradient background and rounded card elements enhance visual appeal. Task items visually indicate completion with strike-through text. Flexbox is used for aligning task list elements.

**8.JavaScript Functionality**

JavaScript plays a crucial role in this To-Do List application by enabling interactive and dynamic task management features. Utilizing the jQuery library, JavaScript handles user input events and manipulates the Document Object Model (DOM) to provide a seamless user experience without page reloads.

Key functions include:

* Adding Tasks: Captures task text input and appends new tasks to the list dynamically.
* Editing Tasks: Allows inline task editing by toggling between read-only text and editable input fields.
* Deleting Tasks: Removes selected tasks from the task array and updates the UI.
* Task Completion Toggle: Manages the completed state of each task by updating the UI and task data.
* Task Filtering: Filters visible tasks based on completion status (All, Completed, Active).

The JavaScript code ensures that the task list stays synchronized with the user's actions immediately through live rendering on each event. This provides a responsive and user-friendly productivity tool.

**Sample snippet:**

javascript

$('#task-form').submit(function(e) {

e.preventDefault();

let taskText = $('#task-input').val().trim();

if (taskText !== '') {

addTask(taskText);

$('#task-input').val('');

}

});

**9.Key Features**

| **Feature** | **Description** |
| --- | --- |
| Responsive Design | Adjusts seamlessly to various screen sizes and devices. |
| Task Management | Add, edit, delete, and toggle completion state of tasks. |
| Filtering | View all tasks or filter by completed or active tasks. |
| Bootstrap Styling | Consistent and modern look with ready-to-use UI components. |
| Interactive UI | Real-time task list updates without page reloads. |

| **Challenge** | **Solution** |
| --- | --- |
| Managing task state and UI sync | Used jQuery for efficient DOM manipulation |
| Responsive layout issues on mobile | Leveraged Bootstrap grid and media queries |
| Editing tasks inline in the list | Implemented toggling between text and input fields |

10.Challenges Faced and Solutions

**11.Outcome**

The To-Do List Application was successfully developed with all intended features working reliably. The interface is intuitive and responsive, making task management simple and effective. This project demonstrates practical front-end skills and the application of Bootstrap and jQuery for interactive web apps.

**12.Future Enhancements**

* Add persistent storage using localStorage or backend integration.
* Implement task priority levels and due dates.
* Add notification reminders.
* Enable drag-and-drop task reordering.
* Provide user authentication for personalized task lists.

**Sample Code Snippet**

javascript

*// Adding a new task*

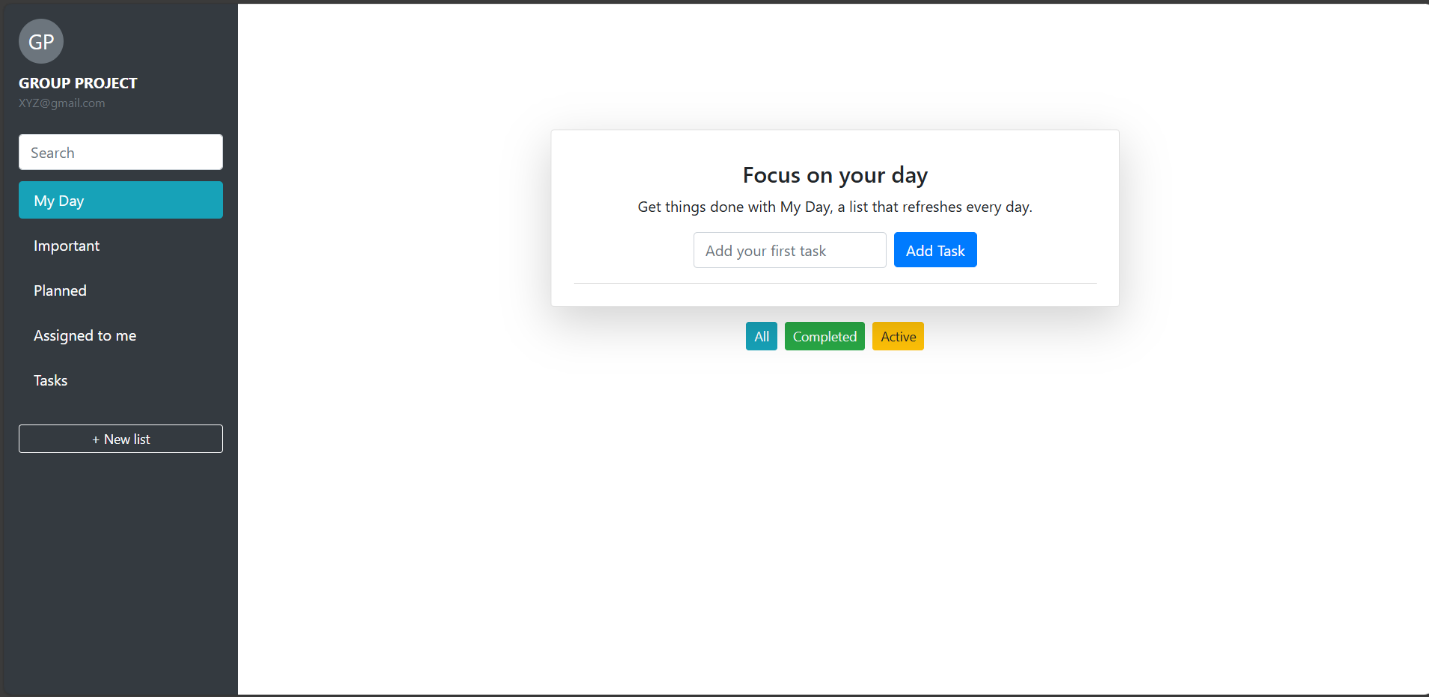
function addTask(text) {

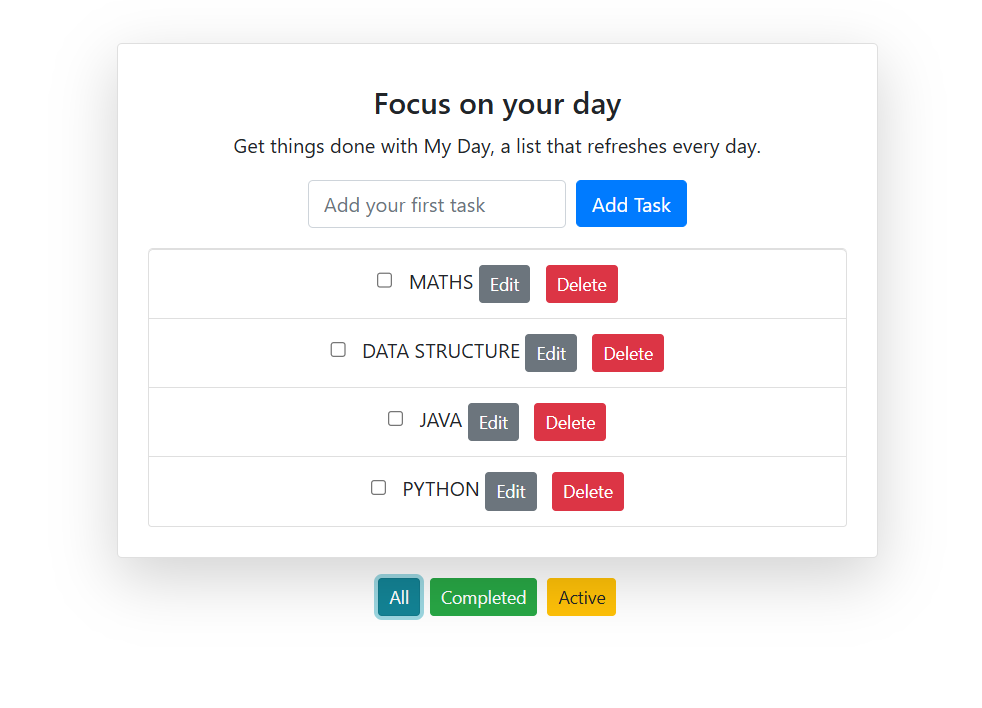
tasks.push({text: text, completed: false});

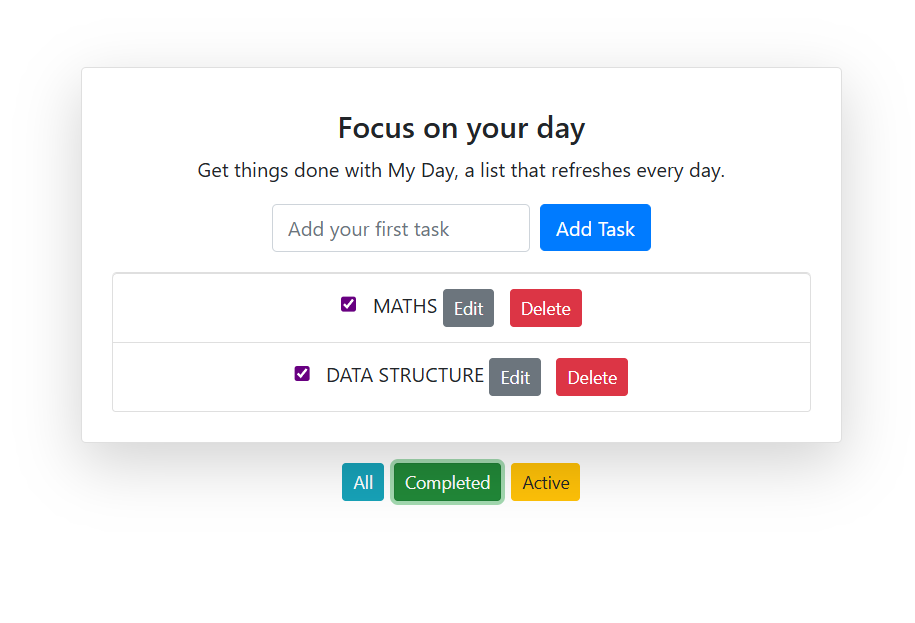
renderTasks();

}

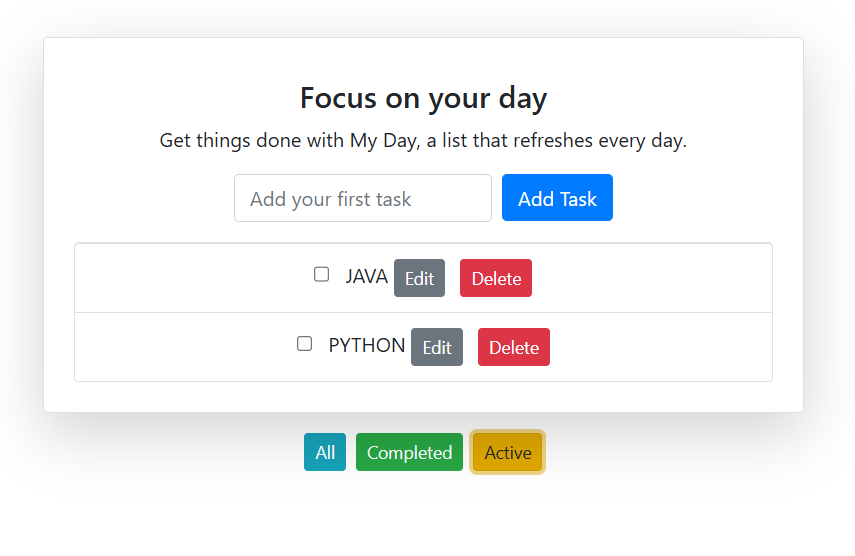
**Images/Screenshots**

* ** Image of the main UI here**
* **Screenshot of adding and editing tasks here**

****

* **Screenshot showing filtering tasks here**
  + **COMPLETED**
  + ****

**-ACTIVE**

****

**13.Conclusion**

This project enhanced our understanding of front-end development with real-time interactivity using jQuery and styling with Bootstrap. It provided valuable experience in implementing task management features with a clean user interface. The project lays a strong foundation for developing more complex and feature-rich web applications.

**14.References**

* Bootstrap Documentation: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>
* jQuery Documentation: <https://api.jquery.com/>
* MDN Web Docs: HTML, CSS, JavaScript guides