HTML TODO List

ADD TASK

#	Task / Description	Responsible	ETA	Action
1	1	2	2024-04-05T11:33	

CONTENTS

ABSTRACT

This project focuses on developing a dynamic and interactive web-based To-Do List application using HTML, CSS (Bootstrap), and JavaScript. Task management is a fundamental aspect of personal and professional productivity, and effective tools can significantly enhance efficiency. The To-Do List application aims to streamline task organization by allowing users to add, edit, mark as complete, and delete tasks through an intuitive interface. Leveraging modern web technologies ensures a responsive and user-friendly experience across various devices. By providing clear functionalities for managing tasks, the application seeks to improve productivity and help users stay organized effectively.

INTRODUCTION

Task management is vital for navigating the complexities of daily responsibilities, whether personal or professional. As individuals juggle numerous tasks, having an organized system to track and prioritize these tasks becomes indispensable. This project addresses this need by developing a web-based To-Do List application that combines robust functionality with a user-friendly interface. Built upon HTML, CSS (Bootstrap), and JavaScript, the application aims to provide seamless task management capabilities accessible from any device with an internet connection.

The primary goal of this project is to create a tool that not only enhances productivity but also simplifies the process of task management. By allowing users to add new tasks, edit existing ones, mark tasks as complete, and remove tasks as needed, the application aims to foster a more organized approach to task tracking. The interface is designed to be intuitive, ensuring that users can quickly grasp how to navigate and utilize its features effectively.

OBJECTIVE

The objective of this project is to create a responsive and user-friendly To-Do List application designed to enhance task management efficiency for users across various platforms and devices. Task management is a fundamental aspect of personal and professional productivity, and the To-Do List application aims to streamline this process through intuitive functionalities.

The primary focus is on providing users with essential features such as adding new tasks, editing existing ones, marking tasks as complete, and deleting tasks. These functionalities are crucial for maintaining an organized task list and ensuring that users can easily update and prioritize their responsibilities. By allowing users to add details like task descriptions, responsible persons, and estimated completion times, the application enables comprehensive task tracking and management.

A key aspect of the application's design is its responsiveness, ensuring that users can access and manage their tasks seamlessly across desktops, tablets, and smartphones. The interface is designed to be intuitive, featuring clear navigation and straightforward controls that facilitate quick task updates and modifications. Visual cues, such as color-coded task states or status indicators, are employed to provide users with immediate feedback on task completion status, enhancing usability and clarity.

Furthermore, the To-Do List application aims to foster productivity by reducing the cognitive load associated with task management. By centralizing task information and providing a structured environment for task updates and deletions, the application helps users stay focused on their priorities and deadlines. This organizational efficiency is complemented by the application's ability to adapt to user preferences and evolving task requirements, ensuring flexibility and usability over time.

Overall, the objective of this project is to deliver a comprehensive To-Do List application that not only meets the functional needs of task management but also enhances user experience through responsive design, intuitive interface, and effective task tracking capabilities. By incorporating user feedback and best practices in web development, the application strives to become a valuable tool for individuals and teams seeking to optimize their productivity and manage tasks with clarity and efficiency.

Methodology

The development of the To-Do List application followed a structured approach to ensure functionality, usability, and scalability:

1. **Design and Layout:** The initial phase focused on designing a responsive and visually appealing layout using Bootstrap, a popular front-end framework. Bootstrap's grid system provided a flexible structure for arranging elements across various screen sizes, ensuring consistency and readability. Components such as navigation bars and cards were utilized to organize content effectively, enhancing user navigation and interaction. Emphasis was placed on creating a clean and modern design that prioritized user experience and accessibility.
2. **HTML Structure:** Semantic HTML tags were employed to define the application's structure, promoting accessibility and search engine optimization (SEO). The layout was organized into meaningful sections, including the navigation bar, task display table, and modal forms for task management operations. Each HTML element was carefully chosen to reflect the content's purpose and improve usability. Form elements were used to capture user inputs efficiently, facilitating seamless interaction and data submission.
3. **CSS Styling:** Bootstrap's pre-defined CSS classes were leveraged to style components consistently throughout the application. Custom CSS rules were

implemented to fine-tune specific visual elements and enhance user interface (UI) aesthetics. Styling choices focused on readability, contrast, and responsive design principles to ensure optimal viewing and interaction across devices. Attention was given to typography, color schemes, and layout spacing to create a visually cohesive and engaging user interface.

4. **JavaScript Functionality:** JavaScript played a crucial role in implementing interactive features and enhancing user interactivity. Event-driven programming was used to respond to user actions such as adding, editing, marking as complete, and deleting tasks dynamically. DOM manipulation techniques were employed to update task lists in real-time, providing users with immediate feedback on their actions. Functions were modularized to maintain code organization and facilitate future enhancements or modifications. Error handling mechanisms were integrated to manage edge cases and ensure robust application behavior under different scenarios.
5. Input validation mechanisms were implemented to validate user inputs before processing task operations. This included ensuring that mandatory fields were filled correctly and handling potential input errors gracefully. Feedback mechanisms such as alerts, notifications, and visual indicators were incorporated to inform users of successful task updates or error conditions promptly. This proactive approach to user feedback enhanced usability by reducing confusion and guiding users through the task management process effectively.
6. **Testing and Iteration:** Throughout the development process, iterative testing was conducted to validate functionality, identify usability issues, and gather user feedback. Both manual testing and automated testing tools were used to ensure application stability and performance across different browsers and devices. User-centered design

principles guided iterative improvements based on feedback, aiming to optimize the application's usability and user satisfaction.

7. The methodology adopted for developing the To-Do List application emphasized a structured and iterative approach to deliver a responsive, intuitive, and scalable task

management solution. By leveraging Bootstrap for design consistency, semantic HTML for accessibility, JavaScript for interactive functionality, and rigorous testing for quality assurance, the application aimed to meet the diverse needs of users seeking efficient task management tools. Continuous feedback and improvement cycles ensured that the application evolved to address user expectations and technological advancements, reinforcing its value as a practical productivity tool in both personal and professional contexts.

CODE

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>HTML ToDo List | Major</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.m
in.css" rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhj
Y6hW+ALEwIH" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.4.2/
css/fontawesome.min.css"
integrity="sha384-BY+fdrpOd3gfeRvTSMT+VUZmA728cfF9Z2G42xpaRkUGu2i3DyzpT
URDo5A6CaLK" crossorigin="anonymous">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstra
p-icons.min.css">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
    <div class="container-fluid">
        <a href="#" class="navbar-brand">
            <img src="logo.png" class="img-fluid" alt="logo"
width="200">
        </a>
        <button type="button" class="navbar-toggler"
data-bs-toggle="collapse" data-bs-target="#navbar">
            <i class="bi bi-list"></i>
        </button>
```

```
        <div class="collapse navbar-collapse" id="navbar">
            <div class="navbar-nav ms-auto"></div>
        </div>
    </div>
</nav>




<div class="container p-5">
    <div class="mb-3">
        <button type="button" class="btn btn-outline-primary"
onclick="showAddTaskModal()">ADD TASK</button>
    </div>
    <div class="d-flex justify-content-center">
        <div class="col-sm-12 col-md-12 col-lg-12">
            <div class="card">
                <div class="card-body">
                    <table class="table">
                        <thead class="text-center">
                            <tr>
                                <th>#</th>
                                <th>Task / Description</th>
                                <th>Responsible</th>
                                <th>ETA</th>
                                <th>Action</th>
                            </tr>
                        </thead>
                        <tbody class="text-center" id="taskTableBody">
                            <!-- Tasks will be added here dynamically
-->
                        </tbody>
                    </table>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
<div class="modal fade" id="addTaskModal" data-bs-backdrop="static"
data-bs-keyboard="false" tabindex="-1"
aria-labelledby="addTaskModalLabel" aria-hidden="true">
    <form id="addTaskForm">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="modal-header">
                    <h5 class="modal-title" id="addTaskModalLabel">Add
Task</h5>
                    <button type="button" class="btn-close"
data-bs-dismiss="modal" aria-label="close"></button>
                </div>
                <div class="modal-body">
                    <div class="mb-1">
                        <label for="addTaskTextArea"
class="form-label">Task / Description</label>
                        <textarea class="form-control"
id="addTaskTextArea" name="taskDescription" rows="3" placeholder="Add
your Task/Description"></textarea>
                    </div>
                    <div class="mb-1">
                        <label for="addTaskResponsible"
class="form-label">Responsible</label>
                        <input type="text" class="form-control"
id="addTaskResponsiblePerson" name="taskResponsiblePerson"
placeholder="Add the Responsible person's Name">
                    </div>
                    <div class="mb-1">
                        <label for="addTaskResponsible"
class="form-label">ETA</label>
                        <input type="datetime-local"
class="form-control" id="addETA" name="taskETA" placeholder="click to
Add time">
                    </div>
                </div>


                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary"
data-bs-dismiss="modal">Cancel</button>
                    <button type="button" class="btn btn-primary"
onclick="addTask()">Add Task</button>
                </div>
```

```
            </div>
        </div>
    </form>
</div>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js">
</script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.8/dist/umd/popper
.min.js"
integrity="sha384-I7E8VVD/ismYTF4hNIPjVp/Zjvgyol6VFvRkX/vR+Vc4jQkC+hVqc
2pM8ODewa9r" crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bun
dle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK
1eN7N6jIeHz" crossorigin="anonymous"></script>
<script>
    let taskCount = 1;

    function showAddTaskModal(){
        $("#addTaskModal").modal('show');
    }

    function addTask() {
        const taskDescription =
document.getElementById('addTaskTextArea').value;
        const taskResponsiblePerson =
document.getElementById('addTaskResponsiblePerson').value;
        const taskETA = document.getElementById('addETA').value;

        if(taskDescription && taskResponsiblePerson && taskETA) {
            const taskTableBody =
document.getElementById('taskTableBody');
            const newRow = document.createElement('tr');

            newRow.innerHTML = `
                <td>${taskCount}</td>
                <td>${taskDescription}</td>
                <td>${taskResponsiblePerson}</td>
                <td>${taskETA}</td>
                <td>
```

```
                    <i class="bi bi-check-circle-fill text-success"
onclick="markTaskDone(this)"></i>
                    <i class="bi bi-pencil-square text-primary"
onclick="editTask(this)"></i>
                    <i class="bi bi-trash text-danger"
onclick="deleteTask(this)"></i>
                </td>
            `;

            taskTableBody.appendChild(newRow);
            taskCount++;

            // Clear the form
            document.getElementById('addTaskForm').reset();
            $("#addTaskModal").modal('hide');
        } else {
            alert('Please fill in all fields');
        }
    }







    function markTaskDone(element) {
        const row = element.closest('tr');
        row.style.textDecoration = 'line-through';
    }

    function editTask(element) {
        const row = element.closest('tr');
        const taskDescription = row.children[1].innerText;
        const taskResponsiblePerson = row.children[2].innerText;
        const taskETA = row.children[3].innerText;

        document.getElementById('addTaskTextArea').value =
taskDescription;
        document.getElementById('addTaskResponsiblePerson').value =
taskResponsiblePerson;
        document.getElementById('addETA').value = taskETA;
```

```
// Remove the current row
        row.remove();

        // Show the modal
        $("#addTaskModal").modal('show');
    }

    function deleteTask(element) {
        const row = element.closest('tr');
        row.remove();
    }
</script>
</body>
</html>
```

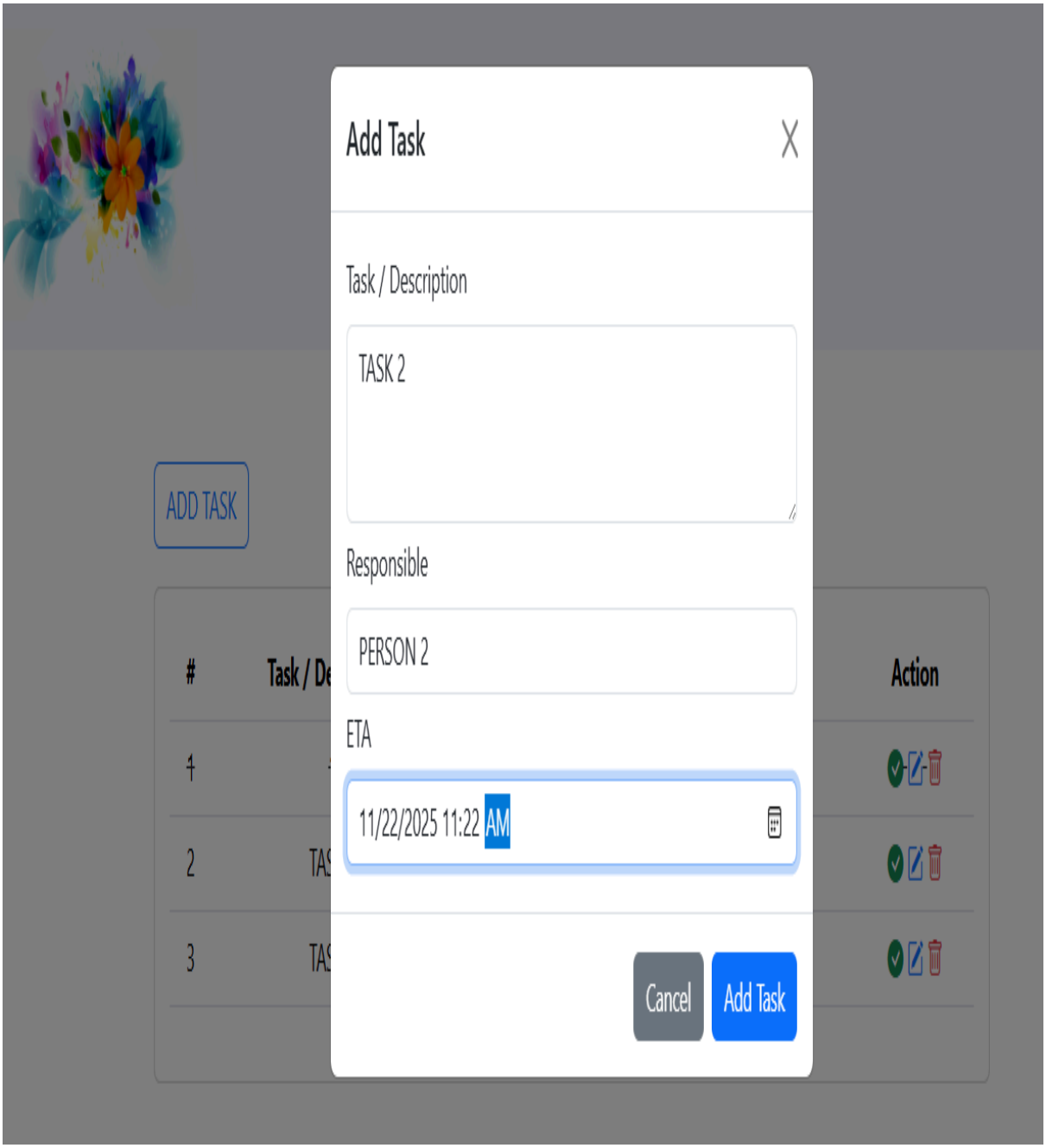
RESULT

1.Initial Interface: - The main interface shows the "ADD TASK" button prominently at the top, inviting users to add a new task. The task list is initially empty, ready to be populated with user tasks.

2. Adding a Task: - When the "ADD TASK" button is clicked, a modal form appears, allowing users to enter the task description, the responsible person, and the estimated time of arrival (ETA). This form ensures that all necessary information is collected before adding the task to the list.

3. Task List with Entries: - After adding tasks, the main interface updates to show a table with all the tasks entered by the user. Each task entry includes the task description, the responsible person, the ETA, and action icons for marking the task as complete, editing the task, or deleting the task.

4. Editing and Completing Tasks: - Users can click on the edit icon to open the modal form pre-filled with the task details, allowing them to make changes and update the task. The check icon marks the task as complete, The delete icon allows users to remove tasks from the list, ensuring that only relevant tasks remain visible.


Add Task
Task / Description
TASK 2
Responsible
PERSON 2
ETA
11/22/2025 11:22 AM
Cancel
Add Task
ADD TASK
#
Action

ADD TASK

#	Task / Description	Responsible	ETA	Action
1	1	2	2024-05-04T11:33	
2	TASK 2	PERSON 2	2025-11-22T11:22	
3	TASK 3	PERSON 3	2025-11-30T14:50	

CONCLUSION

The web-based To-Do List application successfully meets its objective of providing an easy-to-use task management tool. By utilizing HTML, CSS (Bootstrap), and JavaScript, the application offers a responsive and interactive user experience. The functionalities for adding, editing, marking as complete, and deleting tasks are implemented effectively, making task management straightforward and efficient. This project demonstrates the potential of web technologies in creating practical and user-friendly productivity tools

In conclusion, the web-based To-Do List application project successfully delivers on its objective of providing a robust task management tool. By leveraging HTML, CSS (Bootstrap), and JavaScript, the application offers a responsive and intuitive user interface that facilitates efficient task organization. The functionalities for adding, editing, marking as complete, and deleting tasks are implemented effectively, catering to both individual and collaborative task management needs.

The project underscores the capabilities of modern web technologies in creating practical productivity tools that enhance organizational efficiency. Future iterations could explore additional features such as task prioritization, recurring tasks, notifications, and integration with external calendars or project management tools. Overall, the To-Do List application exemplifies the potential of web-based solutions to simplify complex tasks and improve user productivity in diverse environments.