Web Development Front-end Internship

Project Title: **To-Do List Using HTML**

Krishna Joshi
krishnajoshi177.in@gmail.com
Phone - 7976309141
Institution - Graphic Era Hill University, Dehradun

Date of Submission – 25 September, 2024

To-Do List Using HTML

ABSTRACT

This project revolves around the development of a web-based **To-Do List** application using basic web technologies such as **HTML**, **CSS**, and **JavaScript**. The **To-Do List** is a commonly used tool for personal task management, helping individuals organize, track, and prioritize their daily activities. The project's objective is to build a user-friendly application that allows users to input tasks, display them in a list format, and remove them when they are completed. This simple yet functional project demonstrates the power of **HTML** for structuring a webpage, **CSS** for improving the design and layout, and **JavaScript** for enabling interactive features.

The project begins with understanding the user's need for an efficient task management tool and then explores how web development technologies can be utilized to fulfill this need. Using **HTML**, a basic structure for the page is created, consisting of an input field for users to add tasks, and a task list to display these tasks. **CSS** adds aesthetic appeal, providing a clean and organized layout that enhances the user experience. **JavaScript** is responsible for making the list interactive, enabling the functionality of adding new tasks to the list and deleting them as necessary.

Through this project, we aim to provide a solution to a fundamental problem – organizing and managing tasks effectively – while showcasing the application of basic web development skills. The completion of this project will provide a deep understanding of how to combine these technologies to create a functional web application. The project report also covers the challenges encountered during development, the steps taken to overcome them, and the significance of the learning experience gained.

OBJECTIVE

The primary objective of this project is to develop a simple and intuitive **To-Do List** application using **HTML**, **CSS**, and **JavaScript** that provides users with the ability to manage their daily tasks effectively. More specifically, the project aims to achieve the following:

1. **User-Friendly Interface**: Develop a web page where users can easily interact with the To-Do List by adding new tasks, viewing their tasks, and deleting them once they are completed.
2. **Task Management**: Provide users with the ability to input tasks dynamically, add them to a list, and remove them with ease. Each task should appear in the list upon being added, and the list should update as tasks are deleted.
3. **Cross-Browser Compatibility**: Ensure that the application works across different web browsers, demonstrating a fundamental understanding of web development standards.
4. **Visual Appeal**: Use **CSS** to ensure that the layout of the **To-Do List** is visually appealing, clear, and easy to navigate. The appearance of the application should enhance usability by being simple yet effective.
5. **Interactive Functionality**: Implement **JavaScript** to introduce interactive behavior to the webpage. This includes allowing users to add and delete tasks without refreshing the page, improving overall user experience.

The project aims to combine **HTML** for structuring the webpage, **CSS** for styling and improving aesthetics, and **JavaScript** for enabling user interactivity, thereby creating a well-rounded web application. Through this project, the goal is to build a tool that not only enhances productivity but also provides valuable insight into the basics of front-end web development.

INTRODUCTION

In today's fast-paced world, managing daily tasks effectively is crucial for both personal and professional success. The need for organized task management tools has led to the development of numerous applications, with the **To-Do List** being one of the simplest yet most effective tools for improving productivity. A **To-Do List** allows individuals to list out tasks they need to accomplish, track progress, and remove tasks as they are completed. While many advanced task management apps are available, a basic web-based **To-Do List** provides a simple and accessible way to manage tasks directly from a browser without the need for complex installations or software.

This project focuses on developing a **To-Do List** using three core web technologies: **HTML** for structuring content, **CSS** for designing the layout and enhancing the visual appeal, and **JavaScript** for adding interactivity to the webpage. These technologies are foundational to web development and provide a seamless user experience when combined effectively.

HTML (Hypertext Markup Language) is the backbone of web pages. It is used to define the structure and content of a webpage, providing the necessary framework on which additional functionality can be built. In this project, **HTML** is used to create an input field where users can type their tasks, a button to submit those tasks, and an unordered list (<ul>) to display the tasks.

CSS (Cascading Style Sheets) is used to enhance the visual appearance of a webpage by applying styles to the **HTML** elements. It plays a crucial role in making web applications more user-friendly and visually appealing. In this project, **CSS** is used to style the input field, buttons, and task list, ensuring the user interface is clean, organized, and easy to navigate.

JavaScript is a scripting language that enables dynamic functionality in web applications. It allows for the manipulation of the webpage's content in real time without requiring a page reload. In this project, **JavaScript** is used to handle user inputs, append tasks to the list dynamically, and provide the option to remove tasks once they are completed. This interactivity makes the **To-Do List** functional and responsive, allowing users to efficiently manage their tasks.

By combining these technologies, this project aims to create a simple yet powerful tool that demonstrates the importance of basic web development skills. A **To-Do List** may seem like a simple application, but it encompasses important concepts such as event handling, DOM manipulation, and user input management. Additionally, the project serves as an excellent example of how even the most basic web technologies can be used to solve real-world problems, such as personal task management, and how these skills can be applied to build more complex applications in the future.

This report delves into the development process, detailing how each technology is utilized and explaining the methodology used to implement the functionality of the **To-Do List**. Through this project, we aim to provide a comprehensive understanding of how to create a simple, interactive web application from scratch, while also showcasing the potential of **HTML**, **CSS**, and **JavaScript** in building practical solutions.

METHODOLOGY

The **To-Do List** project was developed through a systematic process involving multiple phases. Each phase was designed to focus on a specific aspect of the web application's development, ensuring a structured and efficient approach. Below is a detailed explanation of the methodology followed during the development:

1. **Requirement Analysis**: In the first phase, the functional requirements of the To-Do List application were identified. The goal was to understand what features the application should offer and how it would interact with users. This included determining key features such as the ability to add new tasks, display them in a list format, and delete tasks when completed.
2. **Designing the Application Structure**: The next phase involved designing the structure of the application. A basic layout was created using **HTML** to define the elements on the webpage. This included an input box for users to type their tasks, an "Add Task" button, and an unordered list (<ul>) where tasks would appear. The **HTML** provided the basic skeleton of the webpage.
3. **Adding Style Using CSS**: Once the structure was in place, **CSS** was used to style the webpage. A clean, minimalistic design was adopted to ensure the page remained simple and easy to use. CSS rules were applied to ensure the input box, task list, and buttons were clearly defined and visually appealing. **CSS** also improved the readability and overall user experience by ensuring a balanced and organized layout.
4. **Implementing Interactivity Using JavaScript**: **JavaScript** was introduced to add interactivity to the webpage. JavaScript functions were written to handle user actions like adding and deleting tasks. When a user inputs a task and clicks the "Add Task" button, the task is appended to the list dynamically without refreshing the page. Each task in the list also includes a "Delete" button that allows users to remove tasks once they are completed. **JavaScript** functions also ensured that once a task was deleted, the list would automatically update to reflect the changes.
5. **Testing and Debugging**: After implementing the core functionality, the project was thoroughly tested across different web browsers to ensure compatibility. Bugs related to task addition, deletion, and UI responsiveness were identified and fixed. This phase ensured that the project worked smoothly and consistently across different environments.
6. **Final Deployment**: Once the application was tested and debugged, the final version was deployed. The project was hosted locally and could be used by users to manage their daily tasks. The application is simple, easy to use, and visually appealing.

CODE

HTML Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>To-Do List</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-
icons.min.css">
    <link rel="stylesheet" href="style.css">
    <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+hVqc2pM8ODewa9r"
crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
    <script src="script.js" defer></script>
</head>
<body>
```

HTML Code:

```html
<!-- Navigation Bar -->
    <nav class="navbar navbar-expand-lg navbar-dark ">
        <div class="container-fluid">
            <a class="navbar-brand text-white" href="#">To-Do List</a>
            <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="d-flex ms-auto"> <!-- Aligns logo to the right -->
                <img src="assets/1Stop_logo_New_Png.png" alt="Logo" style="height: 40px;"> <!-- Adjust height as needed -->
            </div>
        </div>
    </nav>
<div class="container mt-4">
        <div class="input-group mb-3">
            <input type="text" id="taskInput" class="form-control" placeholder="Add a new task" aria-label="Task input">
            <button class="btn btn-primary" id="addTaskBtn">Add Task</button>
        </div>
```

HTML Code:

```
<!-- Task Table -->
        <div class="card task-card">
            <div class="card-body">
                <h5 class="card-title">Task List</h5>
                <table class="table table-dark table-striped">
                    <thead>
                        <tr>
                            <th scope="col">#</th>
                            <th scope="col">Task</th>
                            <th scope="col">Responsible</th>
                            <th scope="col">ETA</th>
                            <th scope="col">Actions</th>
                        </tr>
                    </thead>
                    <tbody id="taskTableBody">
                        <!-- Tasks will be added here -->
                    </tbody>
                </table>
            </div>
        </div>
    </div>
```

```html
<!-- Task Modal -->
    <div class="modal fade" id="taskModal" tabindex="-1" aria-labelledby="taskModalLabel" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content" style="color: black;">
                <div class="modal-header">
                    <h5 class="modal-title" id="taskModalLabel" >Add Task</h5>
                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body">
                    <form id="taskForm">
                        <div class="mb-3">
                            <label for="taskDescription" class="form-label">Task Description</label>
                            <input type="text" class="form-control" id="taskDescription" required>
                        </div>
                        <div class="mb-3">
                            <label for="responsiblePerson" class="form-label">Responsible Person</label>
                            <input type="text" class="form-control" id="responsiblePerson" required>
                        </div>
                        <div class="mb-3">
                            <label for="taskETA" class="form-label">ETA (Date and Time)</label>
                            <input type="datetime-local" class="form-control" id="taskETA" required>
                        </div>
                        <input type="hidden" id="taskIndex">
                    </form>
                </div>
                <div class="modal-footer">
                    <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
                    <button type="button" class="btn btn-primary" id="saveTaskBtn">Save Task</button>
                </div>
            </div>
        </div>
    </div>

</body>
</html>
```

CSS CODE (STYLE.CSS)

```css
body {
    background-color: #2b95b5; /* Dark background */
    color: #e0e0e0; /* Light text for contrast */
}

.navbar {
    background-color: #082586; /* Red color for navbar */
}

.input-group input {
    background-color: #c3bbbb; /* Dark input field */
    color: #000000; /* Light text in input */
    border: 1px solid #444; /* Input border */

}

.input-group .btn-primary {
    background-color: #2b00ff; /* Bootstrap primary color */
    border: none; /* No border */
}

.task-card {
    background-color: #20cfff; /* Dark card background */
}
```

CSS CODE (STYLE.CSS)

```css
.table {
    color: #ffffff; /* White text in table */
}

.table-dark {
    background-color: #1e1e1e; /* Dark background for table */
}

.table-striped tbody tr:nth-of-type(odd) {
    background-color: #292929; /* Slightly lighter dark for odd rows */
}

.btn-success {
    background-color: #28a745; /* Green color for edit button */
}

.btn-danger {
    background-color: #ff939d; /* Red color for complete button */
}
```

JAVASCRIPT CODE

```
let tasks = [];
let editMode = false;
let editIndex = -1;

$(document).ready(function () {
    // Add task button click event
    $('#addTaskBtn').click(function () {
        $('#taskModalLabel').text('Add Task');
        $('#taskForm')[0].reset();
        editMode = false; // Reset edit mode
        $('#taskIndex').val(''); // Clear index
    });

    // Save task button click event
    $('#saveTaskBtn').click(function () {
        const taskDescription = $('#taskDescription').val();
        const responsiblePerson = $('#responsiblePerson').val();
        const taskETA = $('#taskETA').val();
        const index = $('#taskIndex').val();

        if (editMode) {
            // Update existing task
            tasks[index] = { description: taskDescription, responsible: responsiblePerson,
eta: taskETA };
        } else {
            // Add new task
            tasks.push({ description: taskDescription, responsible: responsiblePerson, eta:
taskETA });
        }
```

JAVASCRIPT CODE

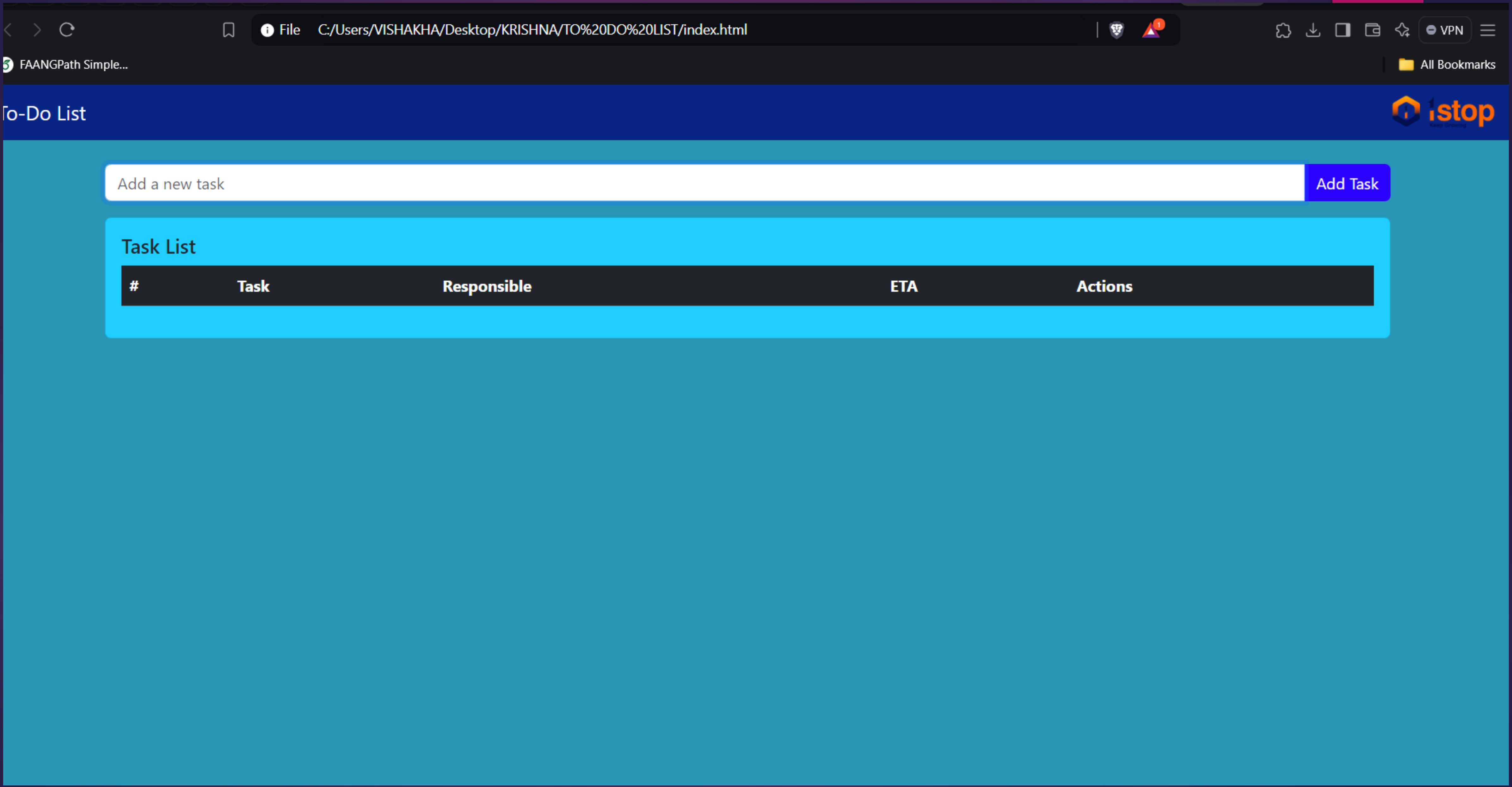
```
        $('#taskModal').modal('hide');
        renderTasks();
    });

    // Function to render tasks in the table
    function renderTasks() {
        const taskTableBody = $('#taskTableBody');
        taskTableBody.empty();
        tasks.forEach((task, index) => {
            taskTableBody.append(`
                <tr>
                    <td>${index + 1}</td>
                    <td>${task.description}</td>
                    <td>${task.responsible}</td>
                    <td>${new Date(task.eta).toLocaleString()}</td>
                    <td>
                        <button class="btn btn-success btn-sm" onclick="editTask(${index})"><i class="bi bi-pencil"></i></button>
                        <button class="btn btn-danger btn-sm" onclick="completeTask(${index})"><i class="bi bi-check-circle"></i></button>
                    </td>
                </tr>
            `);
        });
    }
```

JAVASCRIPT CODE

```
      // Edit task function
    window.editTask = function (index) {
        const task = tasks[index];
        $('#taskDescription').val(task.description);
        $('#responsiblePerson').val(task.responsible);
        $('#taskETA').val(task.eta);
        $('#taskModalLabel').text('Edit Task');
        $('#taskIndex').val(index);
        editMode = true; // Set edit mode
        $('#taskModal').modal('show');
    };

    // Complete task function
    window.completeTask = function (index) {
        tasks.splice(index, 1); // Remove completed task
        renderTasks();
    };
});
```



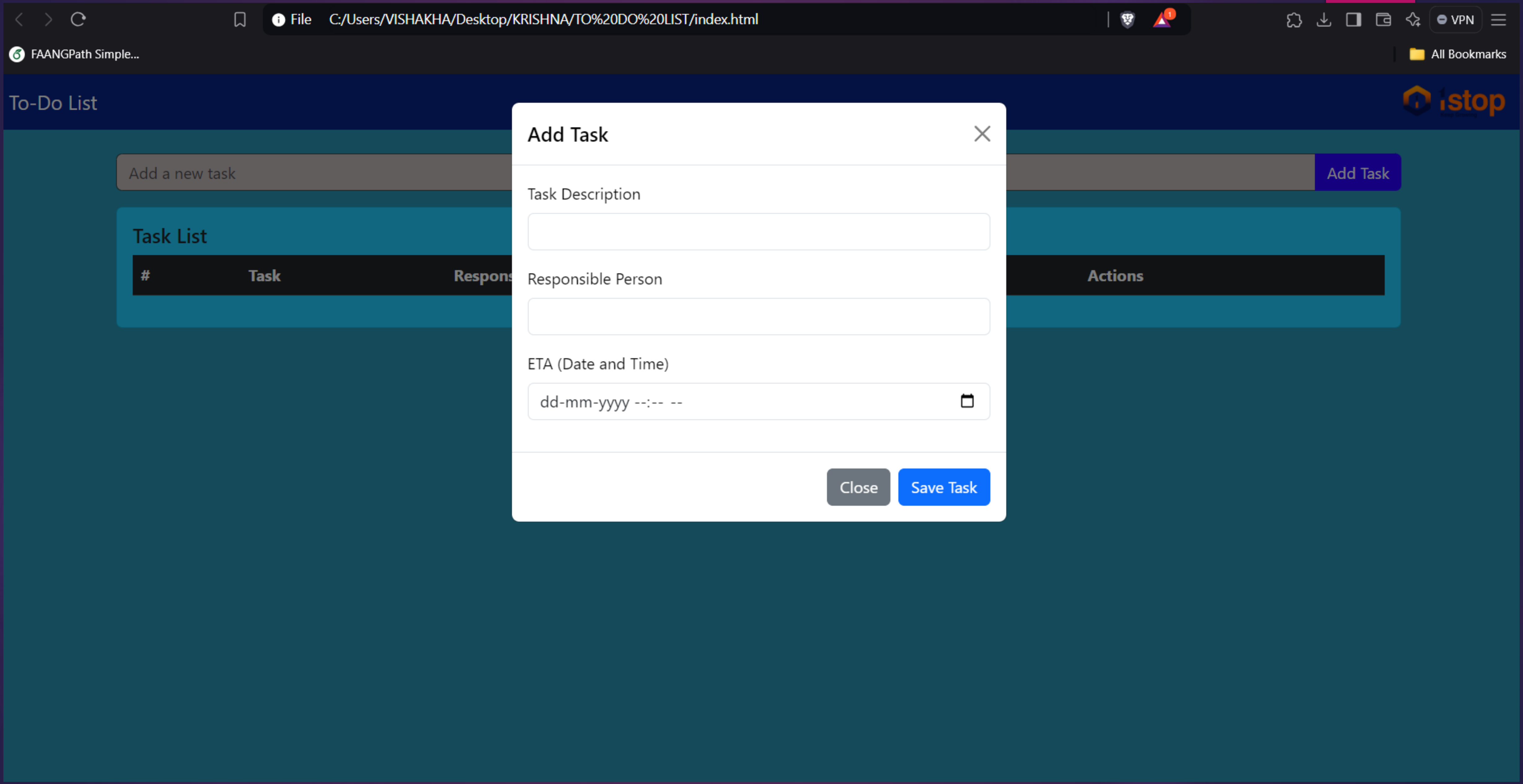
o-Do List

Add a new task

Add Task

Task List

#	Task	Responsible	ETA	Actions



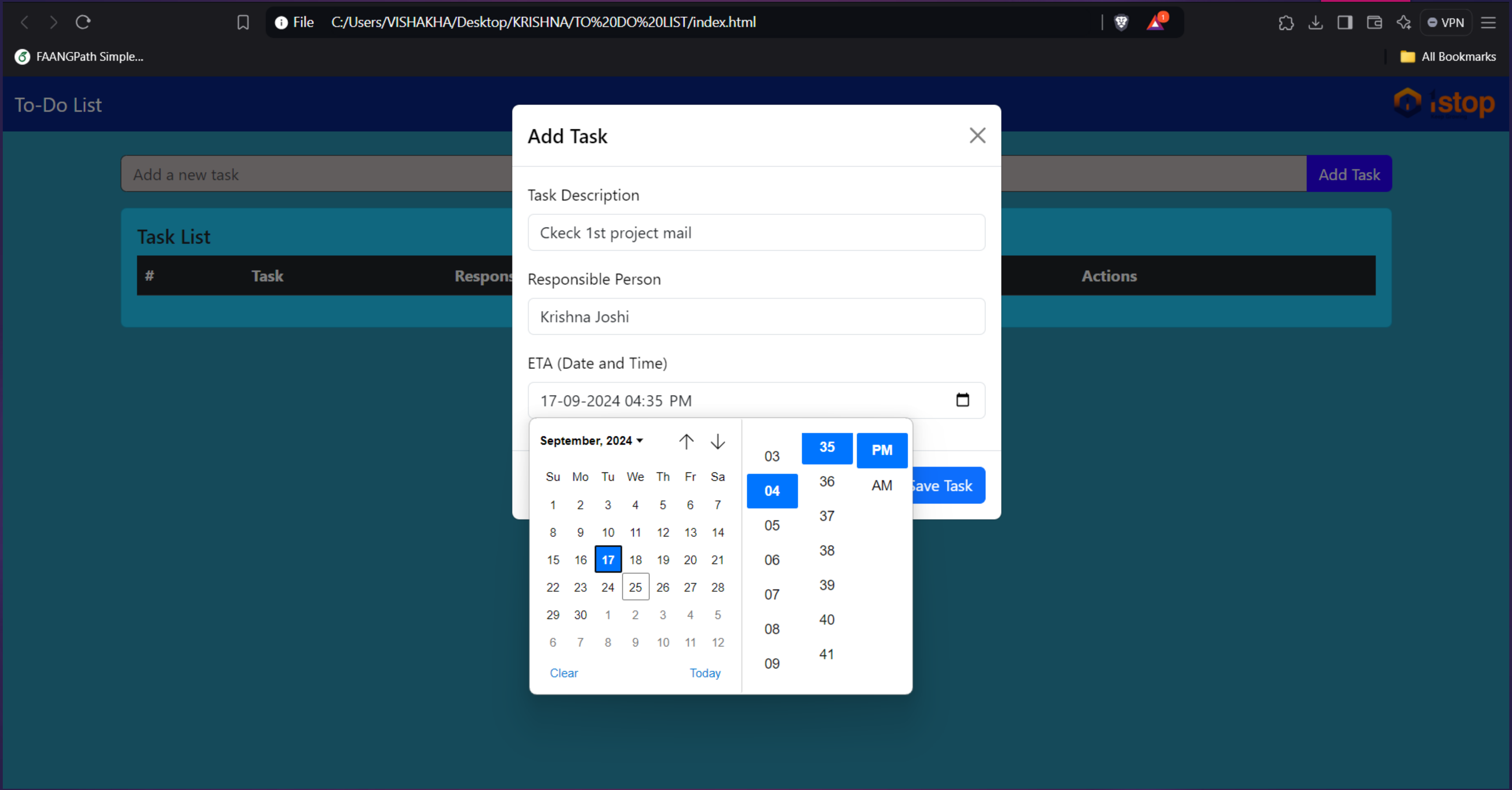
Add Task

Task Description

Responsible Person

ETA (Date and Time)

dd-mm-yyyy --:-- --



To-Do List

Add a new task

Task List

#	Task	Respons	Actions

Add Task

Task Description

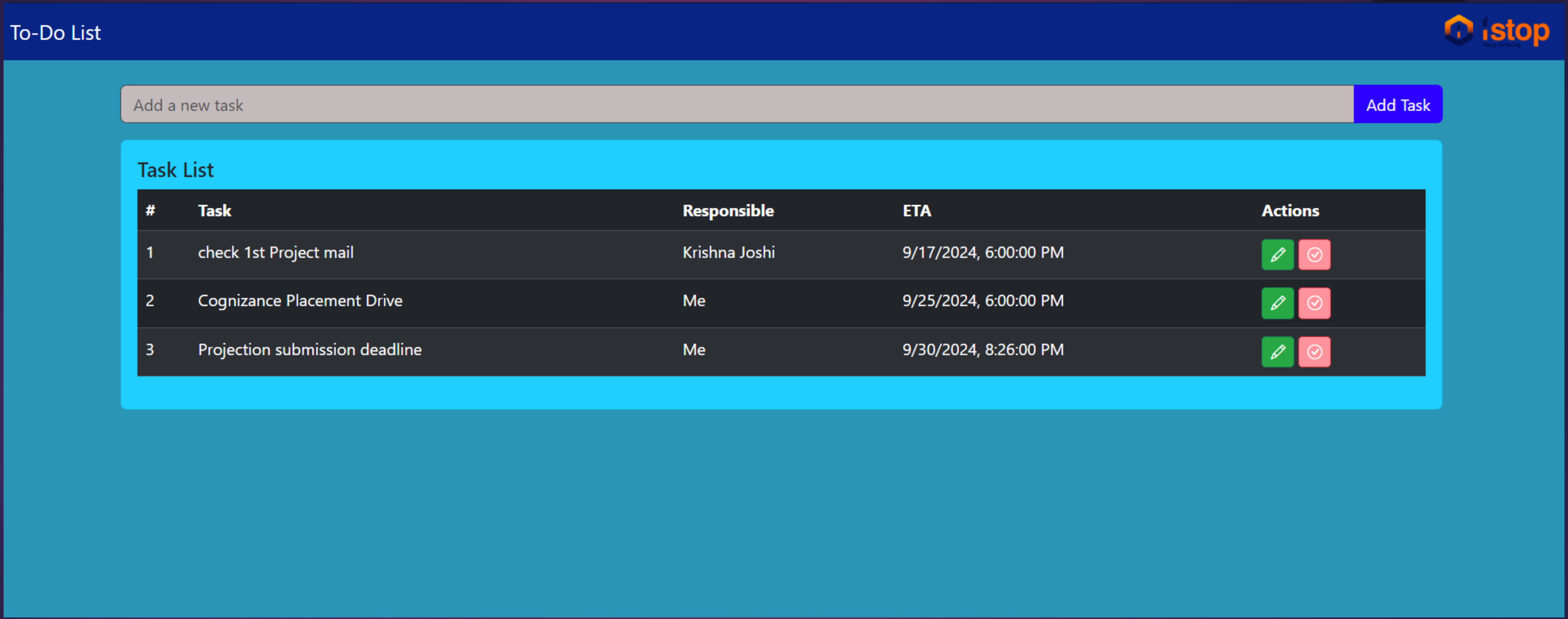
Ckeck 1st project mail

Responsible Person

Krishna Joshi

ETA (Date and Time)

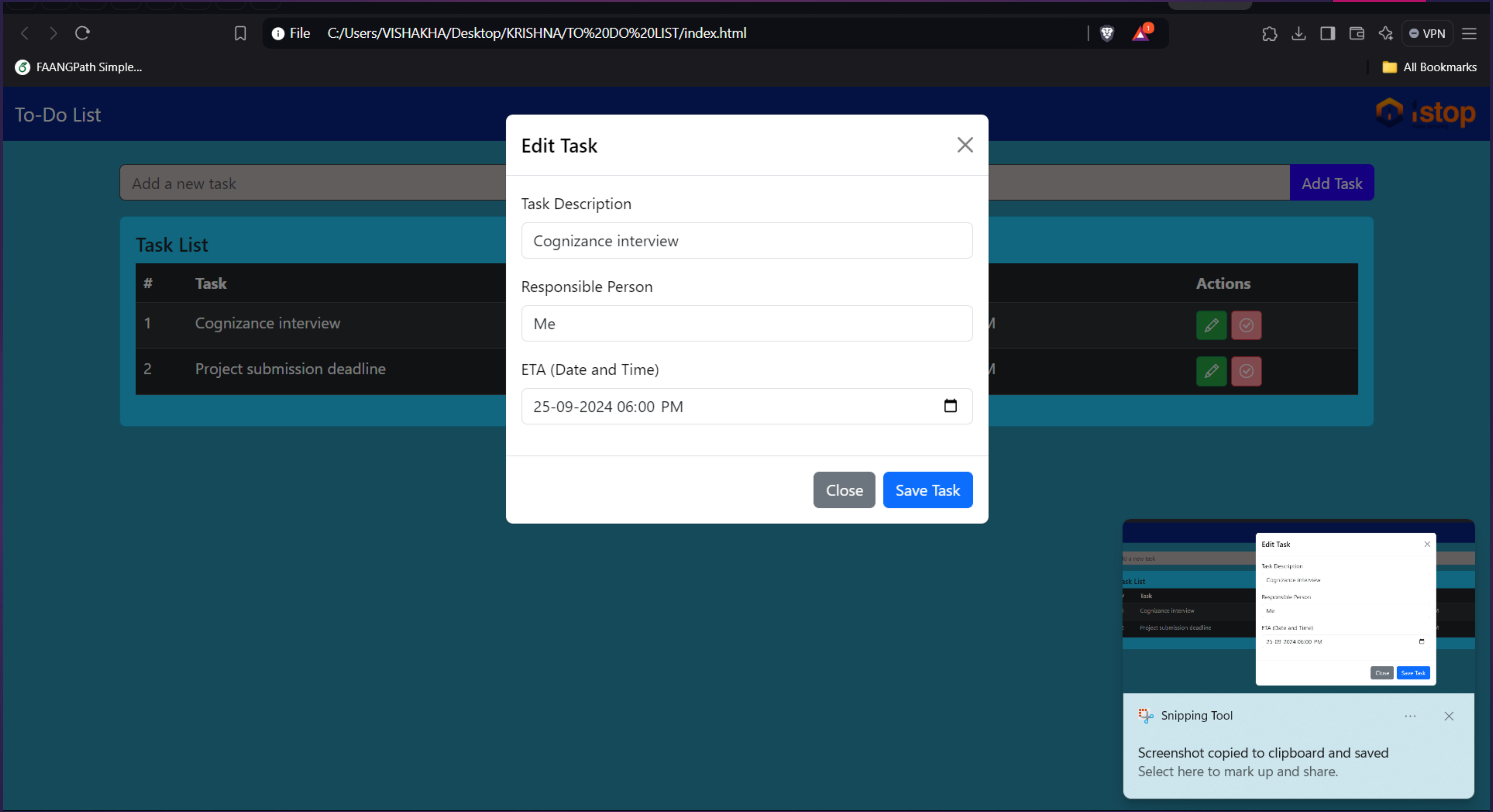
17-09-2024 04:35 PM



Task List

#	Task	Responsible	ETA	Actions
1	check 1st Project mail	Krishna Joshi	9/17/2024, 6:00:00 PM	✎ ✓
2	Cognizance Placement Drive	Me	9/25/2024, 6:00:00 PM	✎ ✓
3	Projection submission deadline	Me	9/30/2024, 8:26:00 PM	✎ ✓

EDIT TASK



To-Do List

Add a new task

Task List

#	Task	Actions
1	Cognizance interview	
2	Project submission deadline	

Edit Task

Task Description: Cognizance interview

Responsible Person: Me

ETA (Date and Time): 25-09-2024 06:00 PM

Close | Save Task

Screenshot copied to clipboard and saved
Select here to mark up and share.

CONCLUSION

The **To-Do List** project was a valuable learning experience that provided insight into the development of web applications using basic web technologies like **HTML**, **CSS**, and **JavaScript**. Through this project, a fully functional and interactive **To-Do List** was developed, allowing users to manage tasks by adding and removing them from the list.

One of the key takeaways from this project was the importance of combining various technologies to build a cohesive web application. **HTML** served as the foundational structure of the webpage, defining the elements required for task management. **CSS** played an important role in enhancing the visual aspects of the webpage, making the application both aesthetically pleasing and easy to use. **JavaScript** brought the much-needed interactivity to the application by allowing users to dynamically add and delete tasks without refreshing the page.

The development process not only reinforced the understanding of basic web development principles but also introduced new concepts such as dynamic DOM manipulation using **JavaScript**. It also highlighted the importance of thorough testing, especially when developing applications meant for different browsers and platforms.

In conclusion, the **To-Do List** project successfully met its objectives by providing a functional, interactive, and user-friendly tool for task management. The project exemplifies how even basic web development skills can be applied to create useful and practical applications. Through this project, we gained a deeper understanding of how to structure, style, and script web pages to deliver effective solutions to everyday problems.