



DEPARTMENT OF AI, ML & DATA SCIENCE

Ann Mary Johnson, 2462039

V EBinesh, 2462167

Jins Thomas, 2462087

GUIDE

Nagaveena

FRONTEND UI/UX

BTOE361T5B

**B. Tech – Computer Science and Engineering
(Artificial Intelligence and Machine Learning)**

School of Engineering and Technology,

CHRIST (Deemed to be University),

Kumbalagodu, Bengaluru-560 074

September 2025



SCHOOL OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF AI, ML & DATA SCIENCE

Certificate

*This is to certify that **Ann Mary Johnson, Jins Thomas, V.Ebinesh** have successfully completed the project work titled, “**PERSONAL PORTFOLIO WEBSITE,TO-DO-LIST** for **Frontend UI/UX** in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence & Machine Learning) during the year 2025-2026.*

FACULTY- IN CHARGE
Nagaveena

CONTENTS

CHAPTER NO.	CHAPTER NAME	PAGE NO.
1.	ABSTRACT	
2.	OBJECTIVES	
3.	SCOPE OF PROJECT	
4.	TOOLS AND TEHNOLOGY USED	
5.	HTML STRUCTURE OVERVIEW	
6.	CSS STYLING STRATEGY	
7.	KEY FEATURES	
8.	CHALLENGES &SOLUTIONS	
9.	OUTCOME	
10.	FUTURE ENHANCEMENTS	
11.	THE PROGRAM	
12.	OUTPUT	
13.	CONCLUSION	

CHAPTER 1: ABSTRACT

In today's fast-paced digital world, efficient management of tasks has become a necessity for individuals as well as organizations. People often struggle with keeping track of their daily activities, which may result in decreased productivity, missed deadlines, or disorganized work routines. To address this issue, digital task management systems have gained significant importance. This project presents the design and implementation of a **Web-Based To-Do List Application** that provides users with an intuitive platform for managing their tasks effectively. The application is developed using **HTML, CSS, Bootstrap, JavaScript, and jQuery**, which are widely used front-end technologies for building interactive and responsive web applications.

The **primary objective** of the project is to provide a simple yet efficient interface where users can **add, edit, delete, and mark tasks as completed**. The application is divided into two categories—**Active Tasks** and **Completed Tasks**—which allows users to clearly distinguish between pending and finished work. Users also have the option to add tasks directly to either of these categories depending on their preference. The inclusion of task filtering and categorization features enhances the usability of the application, making it more practical for real-life scenarios such as academic planning, professional work management, and personal goal tracking.

Another key feature of the application is **data persistence**. By leveraging the **Local Storage API**, tasks are stored in the user's browser, ensuring that the data remains available even after the browser is closed or the page is refreshed. Unlike traditional paper-based methods or non-persistent applications, this feature guarantees that users do not lose their data unless they explicitly delete it. This makes the system reliable and efficient for daily use.

The design of the application follows **responsive design principles** through the use of **Bootstrap**, ensuring that it functions seamlessly on devices of varying screen sizes, including desktops, tablets, and smartphones. The use of **JavaScript and jQuery** enhances interactivity by enabling dynamic task updates without requiring a page reload. For instance, when a user adds a task, the task list updates immediately, providing instant feedback and improving user experience.

This project is particularly beneficial as a **learning model** for beginner-level web developers. It demonstrates how fundamental web technologies can be combined to build a fully functional application. At the same time, it provides users with a practical productivity tool that can be extended further by incorporating advanced features such as due dates, notifications, cloud synchronization, or user authentication.

In conclusion, the Web-Based To-Do List Application successfully meets its goal of providing an easy-to-use, responsive, and persistent task management system. By integrating modern web technologies, the project not only enhances personal productivity but also serves as a strong example of how simple applications can address everyday challenges. The project highlights the

importance of **usability, interactivity, and responsiveness** in software development and lays the foundation for more complex task management systems in the future

CHAPTER 2:OBJECTIVES

The primary objective of this project is to design and develop a **web-based To-Do List application** that allows users to manage their daily tasks effectively in a simple, user-friendly manner. In today's world, time management is one of the most crucial skills, and digital solutions have become an essential aid for individuals in both academic and professional environments. This project focuses on creating an application that helps users organize, track, and complete tasks with minimal effort. Unlike traditional paper-based lists, this application provides interactivity, responsiveness, and persistence, thereby addressing the limitations of manual methods. The ultimate aim is to combine **simplicity of use with the efficiency of modern web technologies**.

One of the key objectives of this project is to provide users with a reliable way to **add, edit, delete, and mark tasks as completed**. These features form the foundation of a task management system, enabling users to track their progress in real time. For example, when a user adds a new task, it should immediately appear on the task list without requiring a page reload. Similarly, editing a task ensures that mistakes can be corrected, while deleting allows users to remove unnecessary or outdated items. Marking tasks as completed is another essential functionality, as it gives users a sense of achievement and keeps the task list organized. This level of interactivity is achieved through **JavaScript and jQuery**, which allow instant updates to the interface.

Another significant objective is the **categorization of tasks**. The application separates tasks into two distinct sections: **Active Tasks** and **Completed Tasks**. This organization is important because it provides clarity to the user. Instead of having a long, mixed list where completed tasks and pending tasks are merged, users can easily distinguish between what is yet to be done and what has already been achieved. Furthermore, the system allows users to add tasks directly into either category, which makes the application more flexible. For instance, if a task is already completed at the time of entry, it can be directly placed into the Completed section. This saves time and provides users with more control over how they want to manage their workflow.

The project also aims to ensure **data persistence** through the use of the **Local Storage API**. A major drawback of many simple web applications is the loss of data once the page is refreshed or the browser is closed. By using local storage, this project ensures that all user tasks are preserved until the user decides to clear them. This persistence feature makes the application reliable and convenient, as users can continue their work at any time without worrying about data loss. This aligns the project with the needs of real-world users, where continuity and reliability are critical aspects of productivity tools.

Equally important is the objective of creating a **responsive design** that works seamlessly across multiple devices. In the present age, users access applications on a wide variety of devices, such as laptops, tablets, and smartphones. By utilizing **Bootstrap**, this project ensures that the interface automatically adapts to different screen sizes. Whether a user is creating a task on a desktop at work or checking off completed items on a mobile phone during travel, the application maintains its functionality and visual appeal. This responsiveness enhances accessibility and usability, making the application practical for all kinds of users.

Lastly, an important educational objective of this project is to serve as a **learning model for web development**. It demonstrates the practical implementation of **HTML for structure, CSS for design, Bootstrap for responsiveness, JavaScript for logic, and jQuery for interactivity**. Students and beginners in web development can learn how these technologies work together to build a complete application. By understanding this project, learners gain insight into how basic web technologies can be integrated into meaningful software solutions.

In conclusion, the objectives of this project extend beyond simply building a task management tool. They encompass the creation of a **functional, interactive, responsive, and persistent application** that addresses real-life needs. The project also highlights the importance of usability, clarity, and design in web applications. By meeting these objectives, the To-Do List application not only helps users manage tasks effectively but also demonstrates how simple web-based systems can be powerful tools for improving productivity and time management.

CHAPTER 3:SCOPE OF PROJECT

The scope of this project revolves around the design and implementation of a **web-based To-Do List application** that enables users to effectively manage their daily activities. The application is intended to be a simple yet powerful productivity tool that can be used by individuals in various contexts, such as students managing their study schedules, professionals keeping track of work assignments, or individuals organizing personal tasks. The project demonstrates how fundamental web technologies like **HTML, CSS, Bootstrap, JavaScript, and jQuery** can be integrated to develop a fully functional application with real-world relevance. By clearly defining its boundaries, the scope ensures that the project remains focused on its core purpose while providing opportunities for future enhancements.

At its core, the application allows users to **create, modify, delete, and organize tasks**. These features represent the minimum requirements for any task management system and form the foundation of this project. The application is not overly complex; instead, it emphasizes simplicity and efficiency. Its user interface is designed to be intuitive so that users can manage their tasks without requiring technical knowledge. By categorizing tasks into **Active** and **Completed** sections, the project ensures clarity and ease of use, helping users focus on pending tasks while keeping a record of completed ones. This scope covers the development of these essential features and guarantees that the application remains lightweight and easy to operate.

Another important element within the scope of this project is the use of **local storage for data persistence**. Unlike many basic applications where data is lost after closing the browser, this application ensures that tasks remain available until the user decides to remove them. This makes the application reliable for everyday use. However, since the project is limited to client-side technologies, the scope does not extend to building server-side storage or cloud synchronization. While such features could enhance the system, they fall outside the boundaries of this project and are identified as potential future improvements.

The scope also emphasizes **responsiveness and accessibility**. In today's digital environment, users expect applications to work seamlessly across a range of devices. By incorporating **Bootstrap**, this project ensures that the To-Do List application adapts to desktops, tablets, and mobile phones. This broadens its usability and ensures that it caters to a wider audience. However, the application is primarily designed for single-user interaction. It does not include features such as multi-user login, data sharing, or collaborative task management. These advanced features are beyond the current scope but remain areas for future extension.

In terms of educational scope, this project also serves as a **learning model for beginners in web development**. It demonstrates how different technologies can be combined to build a working system. For students, it provides a practical example of concepts like DOM manipulation, event handling, and responsive design. For educators, it offers a compact project idea that can be assigned to help learners understand real-world applications of theoretical concepts. The

simplicity of the project ensures that it can be easily understood, while its functionality highlights the potential of front-end development.

The project scope, therefore, is defined by its focus on **task management, usability, responsiveness, and persistence** within the limits of front-end technologies. It does not aim to replicate full-fledged commercial applications like Trello or Microsoft To-Do, but instead provides a simplified version suitable for personal use and academic learning. At the same time, the project lays the groundwork for future development, where features such as deadlines, notifications, or cloud integration could be added.

In conclusion, the scope of this project is both **practical and educational**. On the practical side, it offers users an effective way to manage tasks through a responsive and persistent web application. On the educational side, it provides learners with exposure to fundamental technologies and their integration into a functional project. By keeping the scope well-defined, the project remains achievable within a limited timeframe while still delivering meaningful outcomes.

CHAPTER 4: TOOLS AND TECHNOLOGIES USED

The To-Do List web application is developed using a combination of fundamental front-end web technologies that ensure functionality, interactivity, and responsiveness. The structure of the application is created using HTML5, which provides the basic framework and organizes the different sections such as input fields, task lists, and buttons. CSS3 is used to style the application, giving it an appealing look with proper colors, fonts, and layouts. To enhance the responsiveness of the design, Bootstrap 5 is integrated, ensuring that the application adapts seamlessly to different devices such as desktops, tablets, and smartphones. The interactivity and core logic of the application are handled by JavaScript (ES6), which is responsible for dynamically adding, editing, deleting, and marking tasks as completed. Additionally, jQuery is used to simplify DOM manipulation and event handling, making the implementation more efficient and concise. For data persistence, the Local Storage API of the browser is utilized, ensuring that the user's tasks remain saved even after the browser is closed or refreshed. Collectively, these tools and technologies provide a balance between simplicity and functionality, making the application both user-friendly and reliable.

Tool / Technology	Purpose / Role in the Project
HTML5	Provides the structure and layout of the web application (task input, list, buttons).
CSS3	Adds styling, colors, fonts, and custom design to improve the user interface.
Bootstrap 5	Ensures responsiveness and mobile-friendly design with pre-built UI components.
JavaScript (ES6)	Implements the core logic for task management (add, edit, delete, mark as completed).
jQuery	Simplifies DOM manipulation and event handling for better interactivity.
Local Storage API	Provides persistence by saving tasks in the browser even after refresh or restart.

CHAPTER 5:HTML STRUCTURE OVERVIEW

The HTML structure of the To-Do List application forms the foundation upon which styling and functionality are added. It defines the layout, organizes the content, and provides placeholders for interactivity. The structure is kept clean and semantic, ensuring both readability and compatibility across different devices. The file, named `index.html`, is divided into logical sections including the header, input area for adding tasks, categorized task lists, and footer scripts.

At the top of the document, the **HTML5 doctype declaration** (`<!DOCTYPE html>`) is used, which specifies that the file follows HTML5 standards. The `<html>` tag serves as the root element, enclosing the entire structure. Inside the `<head>` section, metadata such as the title of the project, character encoding, and viewport settings for responsiveness are defined. The head also links external resources including the **Bootstrap CSS framework** for responsiveness and the **custom stylesheet (styles.css)** for additional design. This ensures that the application not only has a proper structure but also a visually appealing layout.

The **body section** begins with a **header area** containing a navigation bar created with Bootstrap components. This gives the application a professional appearance and ensures that the user can easily identify the title of the system—“To-Do List Application.” The use of semantic tags like `<header>` improves readability and aligns with modern web development practices.

Following the header, the main content area is organized inside a `<div>` container with Bootstrap classes to ensure responsiveness. This section contains the **task input form**, where users can type a new task. The form includes a text input field (`<input type="text">`) for entering the task name and buttons for adding the task. Bootstrap button classes are applied to give the buttons a professional look, while placeholders and labels guide the user during task entry.

Below the input form, the application displays two distinct **task categories: Active Tasks and Completed Tasks**. These categories are separated into two different sections using `<div>` and `` elements. The unordered list (``) is used to display tasks dynamically, with each task represented as a list item (``). Each list item contains a checkbox for marking completion, the task text, and action buttons (edit and delete). The use of lists ensures that tasks are displayed in a clear and organized format, while Bootstrap grid classes manage the spacing and alignment across devices.

The **Active Tasks section** is designed to display tasks that are still pending. Whenever a user adds a new task, it automatically appears here unless marked as completed. The **Completed Tasks section**, on the other hand, displays tasks that have been checked off. This separation is implemented structurally within the HTML using different containers with proper headings. By maintaining two distinct lists, the user experience becomes more intuitive, as it is easy to distinguish between pending work and completed activities.

At the bottom of the body, before the closing `</body>` tag, external scripts are linked to provide interactivity. This includes the **Bootstrap JavaScript bundle**, the **jQuery library**, and the **custom app.js script** which contains the logic for adding, editing, deleting, and saving tasks. Placing scripts at the bottom of the HTML structure ensures that the page loads content first before executing functionality, which enhances performance.

In summary, the HTML structure of the To-Do List application is designed with a balance of **semantic clarity, responsiveness, and functionality**. It provides a solid foundation where CSS and JavaScript can enhance both the appearance and interactivity of the system. The use of Bootstrap classes ensures mobile-friendliness, while the division of sections into input, active tasks, and completed tasks maintains simplicity and usability. By keeping the structure clean and modular, the project is easy to extend with future features such as deadlines, categories, or priority levels.

CHAPTER 6:CSS STYLING STRATEGY

The styling of the To-Do List application is implemented using a combination of **CSS3** and **Bootstrap 5**, with a focus on maintaining simplicity, readability, and responsiveness. The primary goal of the styling strategy is to create a user-friendly interface that is both visually appealing and functional across different devices.

The **overall layout** is managed by Bootstrap's grid system, ensuring that the application adjusts smoothly to various screen sizes including desktops, tablets, and smartphones. This eliminates the need for writing excessive custom media queries, as Bootstrap provides built-in responsiveness. Custom CSS is then applied to refine the design, making the application unique and more engaging.

Typography and colors are chosen to maintain a clean and professional look. The font family is kept simple, typically sans-serif, to enhance readability. Task lists use slightly larger text to ensure clarity, while headings such as "Active Tasks" and "Completed Tasks" are styled with bold weight and spacing to distinguish them clearly. A consistent **color scheme** is followed, with darker tones for active tasks and lighter or muted tones for completed ones, creating a natural visual hierarchy.

For **task items**, CSS is used to add spacing, borders, and hover effects. Each task appears in a card-like format with padding and rounded corners to improve readability and aesthetics. Completed tasks are styled with strikethrough text and faded colors, giving users immediate visual feedback on their progress. Action buttons such as *edit* and *delete* are styled with icons and Bootstrap's button classes, but additional CSS ensures that hover states and active states provide clear feedback.

Interactivity-related elements, such as the input field and add button, are styled for better usability. The input box uses subtle borders and focus effects so that users can quickly identify where to type. The add button is emphasized with a contrasting color, making it the primary call-to-action on the interface.

Finally, spacing and alignment are managed through a combination of Bootstrap utility classes and custom CSS rules. Margins, padding, and shadows are applied carefully to avoid clutter and provide a modern, minimalistic design. The strategy prioritizes **clarity, consistency, and responsiveness**, ensuring that the application looks professional and is easy to navigate for all users.

CHAPTER 7 KEY FEATURES

The To-Do List Web Application is designed to provide an intuitive and efficient platform for managing daily tasks. Its features are carefully selected to balance simplicity with functionality, ensuring that users can organize their work with minimal effort. One of the most important features is the ability to **add new tasks** quickly. The application provides a clean input interface where users can type a task and immediately add it to their list. This allows for instant task entry without unnecessary steps, improving productivity.

Another significant feature is **task categorization**. Tasks are divided into two distinct sections: **Active Tasks** and **Completed Tasks**. This separation provides clarity, enabling users to distinguish between pending and finished tasks at a glance. Users can also directly add tasks to either category through a dropdown option, offering flexibility based on task status at the time of entry.

The application supports **editing tasks** so that users can correct mistakes or update details. This ensures accuracy and adaptability in task management. The **delete feature** allows users to remove tasks that are no longer relevant, keeping the list organized and clutter-free.

A crucial functionality is the ability to **mark tasks as completed** using a checkbox. When a task is marked completed, it moves to the Completed section with a visual change in styling, such as strikethrough text and faded color. This provides immediate feedback and reinforces a sense of progress.

The project also emphasizes **data persistence** through the use of the **Local Storage API**. This ensures that tasks remain saved even after the browser is closed or the page is refreshed, providing continuity and reliability for daily task management.

The interface is designed to be **responsive and mobile-friendly**, thanks to **Bootstrap's grid system**. Whether accessed from a desktop, tablet, or smartphone, the application maintains its functionality and visual appeal. This adaptability makes the application useful in a variety of contexts, whether at work, at school, or on the go.

Finally, the application provides **interactive feedback** for all actions. Whether adding, editing, deleting, or marking tasks completed, changes happen instantly without page reloads, thanks to **JavaScript and jQuery**. This enhances the user experience and makes the system feel modern and efficient.

These key features collectively make the To-Do List Web Application a practical tool for personal and professional task management, combining functionality, interactivity, and ease of use into a compact and reliable system.

CHAPTER 8 CHALLENGES AND SOLUTION

While developing the To-Do List Web Application, several challenges were encountered during the design and implementation stages. Addressing these challenges was essential to ensure that the final product was functional, efficient, and user-friendly. This section discusses the main challenges faced and the solutions implemented to overcome them.

One of the primary challenges was **ensuring data persistence** so that tasks would remain available even after refreshing the browser or closing it. Without persistence, the application would lose all tasks upon page reload, making it impractical for daily use. The solution to this problem was the integration of the **Local Storage API**. Local storage allowed task data to be stored in the user's browser, enabling retrieval even after the session ended. By converting tasks into JSON format before storage and parsing them back when retrieved, the application was able to maintain data continuity effectively.

Another challenge was **implementing dynamic interactivity** without reloading the page. Initially, adding, editing, or deleting tasks required manual refreshes, which negatively impacted the user experience. To solve this, **JavaScript and jQuery** were utilized to handle DOM manipulation efficiently. Event listeners were implemented for all interactive actions so that the interface updated instantly without reloading, creating a smooth and modern user experience.

Ensuring **responsive design** was another challenge. Users access web applications from devices of various screen sizes, and without proper adaptability, the user interface could break or become hard to use. The solution was to use **Bootstrap's grid system and utility classes**, which provided pre-built responsive design components. This ensured that the application worked seamlessly on desktops, tablets, and mobile phones without the need for extensive custom CSS media queries.

A further challenge was **organizing tasks into separate Active and Completed sections** in a way that was both functional and visually intuitive. The difficulty was ensuring that tasks moved between these sections dynamically without losing their data or formatting. The solution was to implement a robust categorization system in JavaScript, where each task carried a "completed" status flag. When the checkbox was toggled, the task was reclassified and moved to the appropriate section, with corresponding style changes applied to reflect its status.

Lastly, maintaining a **simple yet visually appealing interface** was a design challenge. Overcomplicating the UI could make the application harder to use. The solution was a focused

design strategy that combined Bootstrap for structure and responsiveness with custom CSS for visual appeal. Minimalistic design principles were applied, keeping the interface clean and intuitive while ensuring that all features were easily accessible.

In conclusion, while challenges arose during the development of the To-Do List Web Application, each was addressed with targeted solutions. These solutions not only resolved the specific issues but also strengthened the overall quality and usability of the application. Overcoming these challenges contributed significantly to creating a robust, interactive, and responsive task management tool suitable for everyday use.

CHAPTER 9 OUTCOME

The development of the To-Do List Web Application has resulted in a practical, interactive, and responsive task management tool that meets the initial objectives of the project. By integrating **HTML, CSS, Bootstrap, JavaScript, and jQuery**, the project successfully demonstrates the application of front-end technologies to build a real-world productivity solution. The final outcome is a system that allows users to add, edit, delete, and categorize tasks effectively, providing a structured approach to managing daily activities.

One of the primary outcomes of this project is the **successful implementation of a clean and intuitive user interface**. The interface is designed to be user-friendly and visually appealing, with clearly defined sections for Active and Completed tasks. The categorization improves usability by enabling users to focus on pending tasks while still keeping track of completed ones. The ability to add tasks directly to either section adds flexibility, enhancing the overall user experience.

Another significant outcome is the **interactivity and responsiveness** of the application. Actions such as adding, editing, deleting, and marking tasks as completed occur instantly without the need for page refreshes. This seamless interactivity is achieved through efficient DOM manipulation with JavaScript and jQuery. The responsiveness of the interface, ensured by Bootstrap's grid system, allows the application to adapt effortlessly to various devices, making it accessible on desktops, tablets, and smartphones.

A notable achievement is the **implementation of data persistence**. By using the Local Storage API, tasks remain saved even after the browser is closed or refreshed. This ensures continuity for users, enabling them to maintain their task lists without the fear of losing important data. This feature significantly enhances the usability and practicality of the application for everyday task management.

The project also delivers **educational outcomes**. It serves as a practical example of how front-end technologies can be integrated into a functional web application. Beginners in web development can gain valuable experience from understanding how HTML, CSS, Bootstrap, JavaScript, jQuery, and local storage work together to produce a working system. This project can be used as a stepping stone for more complex applications in the future, such as collaborative task management systems with cloud storage and user authentication.

In conclusion, the To-Do List Web Application successfully delivers a robust, interactive, and responsive solution for task management. The project not only fulfills its intended purpose but also provides a strong foundation for further enhancements. The outcome is a simple yet effective tool that can help users improve productivity, manage time efficiently, and organize their daily activities in a structured manner.

CHAPTER -10 Future Enhancement

While the current To-Do List Web Application offers a practical and efficient solution for basic task management, there is significant scope for further development and improvement. The future enhancements aim to make the application more powerful, versatile, and user-centric by incorporating advanced features that can increase productivity and improve user experience.

One of the most valuable enhancements would be the addition of user authentication and cloud storage. Currently, the application uses Local Storage for saving tasks, which works only on the local browser and device. By integrating a backend with user authentication, tasks could be stored securely on the cloud, allowing users to access their task lists from any device. This would also enable multi-user support and synchronization across different platforms.

Another potential improvement is the incorporation of task categorization and prioritization features. While the current system distinguishes tasks as Active or Completed, adding custom categories such as Work, Personal, or Urgent would allow users to organize tasks more effectively. Priority labels or color codes could help users focus on high-priority tasks, enhancing productivity.

Adding deadline and reminder functionality is another promising enhancement. Users could set due dates for tasks, and the system could send notifications or alerts before the deadlines. This feature would make the application not just a static task list but a dynamic productivity assistant. Integrating reminders through email or push notifications would further increase the application's usability.

The addition of drag-and-drop functionality could enhance the user experience by allowing tasks to be reordered easily. Users could prioritize tasks by simply dragging them up or down in the list, providing a more interactive and engaging interface.

Another possible enhancement is integration with calendars and productivity tools. Synchronizing tasks with calendar applications such as Google Calendar would help users visualize their schedules and deadlines in one place. Integration with other tools like note-taking apps or project management systems could make the application part of a larger productivity ecosystem.

Lastly, improving the UI/UX design with themes, dark mode, and customizable layouts could make the application more appealing to different users. Accessibility enhancements, such as keyboard navigation and screen reader support, would make the application usable for a wider audience.

In conclusion, the future enhancements outlined above would transform the To-Do List Web Application from a simple task management tool into a comprehensive productivity platform. These improvements would not only increase its functionality and usability but also make it adaptable to the evolving needs of users in various personal and professional contexts. By implementing these enhancements, the application can continue to grow and remain relevant as a practical productivity tool in the future.

Chapter 11 The Program

```
<!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 App</title>

  <!-- Bootstrap CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

  <!-- Your custom CSS -->
  <link rel="stylesheet" href="css/styles.css">
</head>
<body>

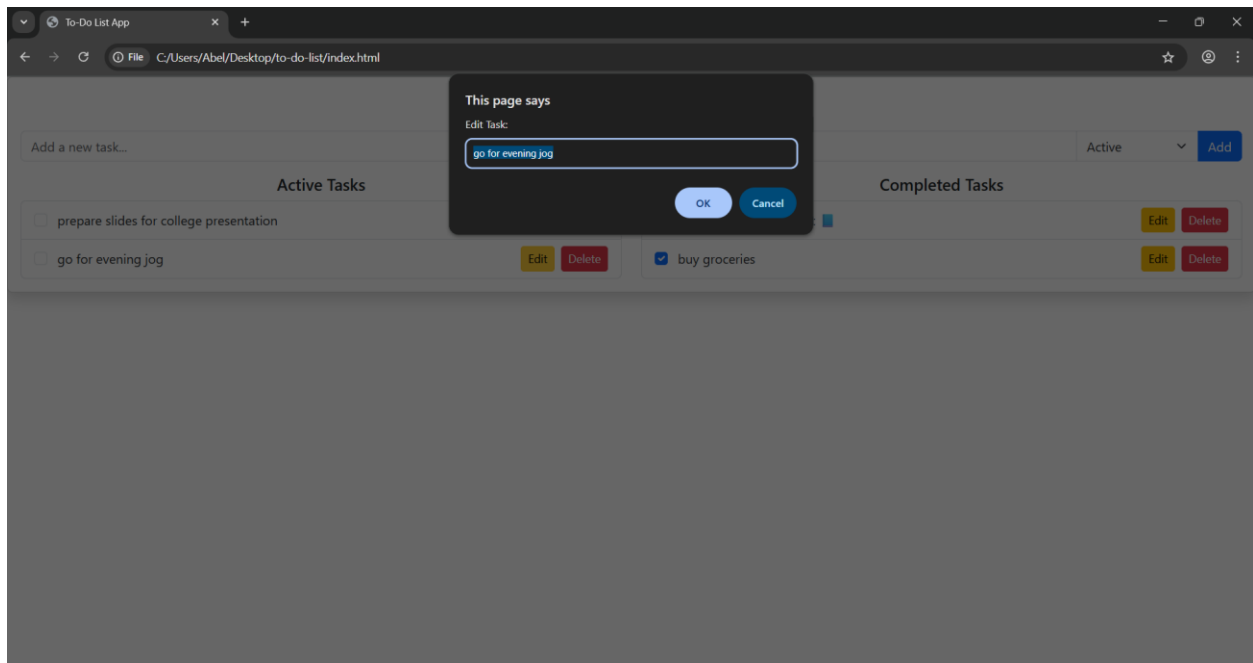
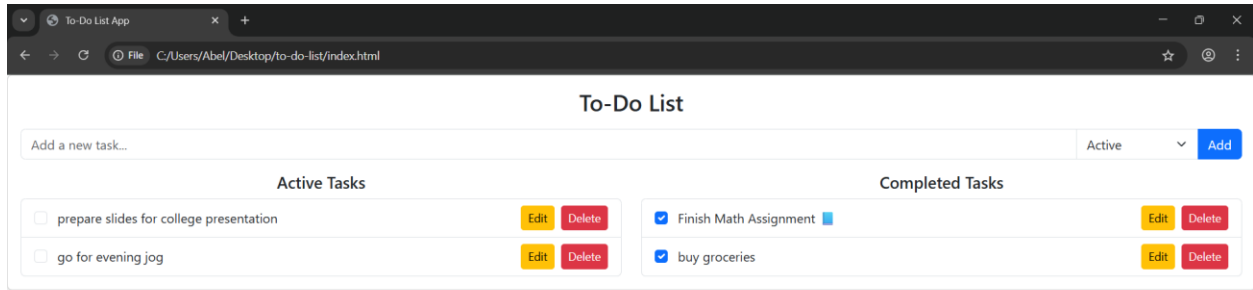
  <div class="card shadow todo-card">
    <div class="card-body">
      <h3 class="card-title text-center mb-3">To-Do List</h3>

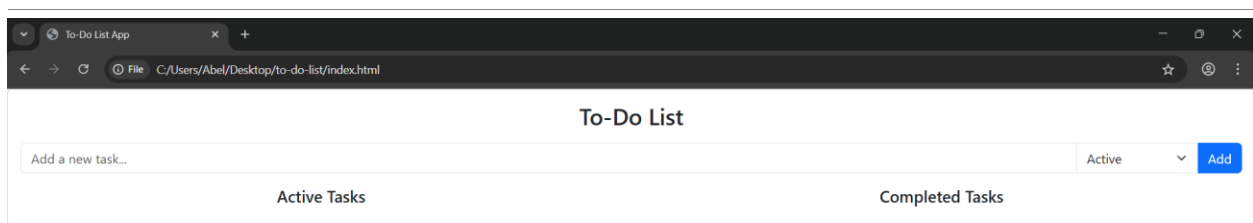
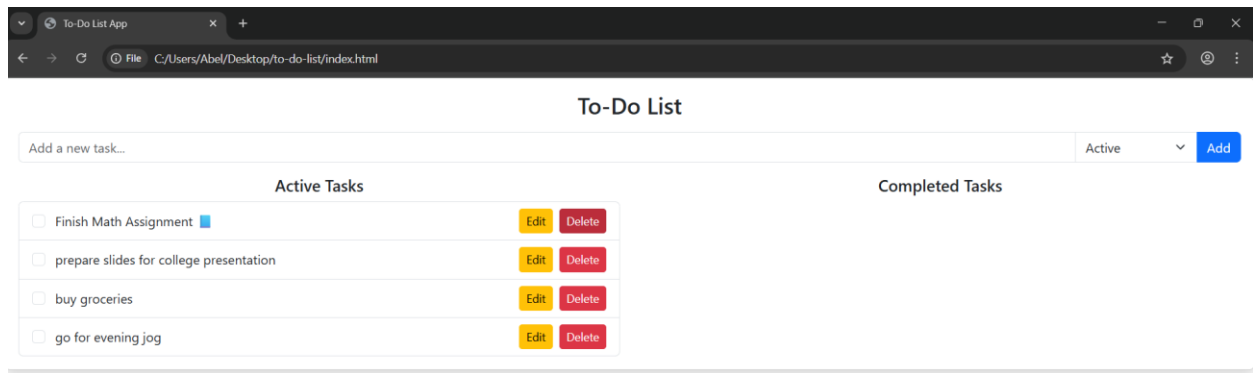
      <!-- Add Task Input with dropdown -->
      <div class="input-group mb-3">
        <input type="text" id="taskInput" class="form-control" placeholder="Add a new task...">
        <select id="statusSelect" class="form-select" style="max-width: 150px;">
          <option value="active">Active</option>
          <option value="completed">Completed</option>
        </select>
        <button class="btn btn-primary" id="addTaskBtn">Add</button>
      </div>

      <!-- Sections -->
      <div class="row">
        <div class="col-md-6">
          <h5 class="text-center">Active Tasks</h5>
          <ul id="activeList" class="list-group"></ul>
        </div>
        <div class="col-md-6">
          <h5 class="text-center">Completed Tasks</h5>
          <ul id="completedList" class="list-group"></ul>
        </div>
      </div>
    </div>
  </div>

  <!-- jQuery -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <!-- Bootstrap JS -->
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
  <!-- Custom JS -->
  <script src="js/app.js"></script>
</body>
</html>
```

Chapter 12 Output





Chapter 13 Conclusion

The To-Do List Web Application demonstrates how modern front-end technologies can be combined to create a simple yet powerful task management system. The project successfully meets its objectives by providing users with the ability to add, edit, delete, and organize tasks into Active and Completed sections. With a focus on responsiveness, interactivity, and usability, the application offers a clean and efficient platform for daily task management.

One of the key achievements of this project is its ability to deliver a seamless and interactive user experience. Through the use of JavaScript and jQuery, all operations—such as adding, editing, and marking tasks—are performed dynamically without requiring page reloads. This creates a smooth workflow that mirrors the expectations of modern web applications. In addition, the integration of Local Storage ensures that user data is preserved even after closing or refreshing the browser, making the system reliable for everyday use.

The project also highlights the importance of responsive design. By utilizing Bootstrap, the application adapts to various screen sizes, ensuring accessibility across desktops, tablets, and mobile devices. This responsiveness broadens the application's usability and makes it relevant in a variety of contexts, from personal use to professional task organization.

Beyond its technical outcomes, the project provided an opportunity to develop practical skills in web development. It reinforced the understanding of how HTML structures content, CSS enhances presentation, Bootstrap ensures responsiveness, and JavaScript/jQuery enables interactivity. Together, these technologies were applied effectively to create a real-world application that can be expanded in the future.

In conclusion, the To-Do List Web Application is a successful demonstration of designing and implementing a functional productivity tool using core web technologies. It not only serves as a practical solution for managing tasks but also provides a solid foundation for further enhancements, such as user authentication, reminders, and cloud integration. Ultimately, the project achieves its intended purpose of improving task management while showcasing the effective use of front-end development practices.