Project Report

1. Title Page

Project Title: To-do List

Submitted By:

Team Member: Tirth Patel, Amiya Prakash

Roll Number: 2460494, 2460325

College E-mail ID: Tirth.manishbhai@btech.christuniversity.in, amiya.prakash@btech.christunversity.in

Course: UI/UX Design Fundamentals

Instructor Name: Nagaveena Mam

Institution: Christ University

Date of Submission: 26/09/2025

2. Abstract

This project presents the design and implementation of a simple To-Do List web application using HTML, CSS, JavaScript, and Bootstrap. The application provides users with an intuitive interface to add, update, and delete tasks, enabling efficient organization of daily activities. The integration of Bootstrap ensures a responsive and visually appealing layout, while JavaScript facilitates dynamic functionality such as real-time task management and interactive user experience. The system was developed to be lightweight, user-friendly, and easily adaptable for future enhancements like task categorization, priority levels, and persistent storage. This project demonstrates how fundamental web technologies can be combined to create practical productivity tools.

3. Objectives

- **Develop a user-friendly interface** that allows users to easily add, edit, and delete tasks.

- **Implement responsive design** using Bootstrap so the application works seamlessly across devices.

- **Enable dynamic task management** through JavaScript to provide real-time interaction without reloading the page.

- **Ensure simplicity and usability** so users can manage tasks efficiently with minimal effort.

- **Lay the foundation for scalability**, allowing future enhancements such as task categorization, deadlines, and data persistence using databases or local storage.

4. Scope of the Project

Provides a lightweight productivity tool for organizing daily tasks.

Supports core features: **add, edit, and delete tasks**.

Uses **HTML, CSS, JavaScript, and Bootstrap** for responsive design and interactivity.

Focuses on **client-side functionality** without server-side data storage.

Ensures a **simple, user-friendly, and visually appealing interface**.

Lays the foundation for future features such as:

- Task prioritization

- Categorization
- Deadline reminders
- Integration with local storage or cloud services
-

5. Tools & Technologies Used

Tool/Technology	Purpose
HTML5	Structure and markup
CSS3	Styling, responsive layout
Bootstrap	Responsive grid and components
JavaScript	Quiz logic, timer, feedback
jQuery	DOM manipulation and event handling
VS Code	Code editor
Chrome DevTools	Testing and debugging

6. HTML Structure Overview

- Used semantic tags: `<header>`, `<main>`, `<section>`, `<footer>` for proper structure.
- Structured into sections: Input container (task entry), Task list container (dynamic tasks), Action buttons (add/delete/edit).
- Task management handled dynamically using JavaScript/jQuery.

- Buttons created with `<button>` tags and styled using Bootstrap classes for consistency and responsiveness.

- Task list implemented with `<ul>` and `<li>` elements, updated dynamically based on user input.

7. CSS Styling Strategy

- External CSS file for main styling.
- Bootstrap used for grid, buttons, and responsive layout.
- Techniques used:
    - Flex box for button alignment and layout.
    - Progress bar animation using CSS transitions.
    - Hover effects for options.
    - Mobile-first design approach.
    - Color coding correct/incorrect answers dynamically.

8.Key Features

Feature Description

Add Tasks – Users can easily add new tasks using the input field and button.

Edit Tasks – Tasks can be updated inline with an edit option.

Delete Tasks – Unwanted tasks can be removed instantly.

Mark as Completed – Tasks can be checked off with a strike-through effect.

Filter Options – Tasks can be filtered by **All**, **Active**, or **Completed**.

Local Storage Support – Tasks are saved in the browser's local storage for persistence.

Responsive Design – Built with Bootstrap to ensure compatibility across devices.

Interactive UI – Smooth animations, hover effects, and icons enhance user experience.

9. Challenges Faced & Solutions

Challenge

Challenge: Managing task persistence across browser sessions.
Solution: Implemented **Local Storage** to save and retrieve tasks automatically.

Challenge: Updating tasks dynamically after edit or delete operations.
Solution: Used **jQuery DOM manipulation** and `renderTasks()` function to refresh the task list.

Challenge: Ensuring smooth user interaction and responsive design.
Solution: Integrated **Bootstrap** for styling and responsiveness across devices.

Challenge: Preventing empty tasks from being added.
Solution: Added **input validation** to check task text before saving.

Challenge: Maintaining a clean and interactive UI while handling multiple actions (edit, delete, complete).
Solution: Designed separate **action buttons** with Font Awesome icons and hover effects for clarity.

10. Outcome

- Successfully developed a **fully functional To-Do List application** using HTML, CSS, JavaScript, jQuery, and Bootstrap.

- Achieved a **user-friendly, responsive, and interactive interface**.

- Ensured **task persistence** with local storage, allowing users to retain their tasks even after refreshing or closing the browser.

- Improved productivity by providing essential task management features: add, edit, delete, complete, and filter tasks.

11. Future Enhancements

- Add **task prioritization** (High, Medium, Low).

- Enable **due dates and reminders** for time-sensitive tasks.

- Support **task categories or labels** for better organization.

- Integrate with **cloud storage or databases** for cross-device access.

- Implement **drag-and-drop reordering** of tasks.

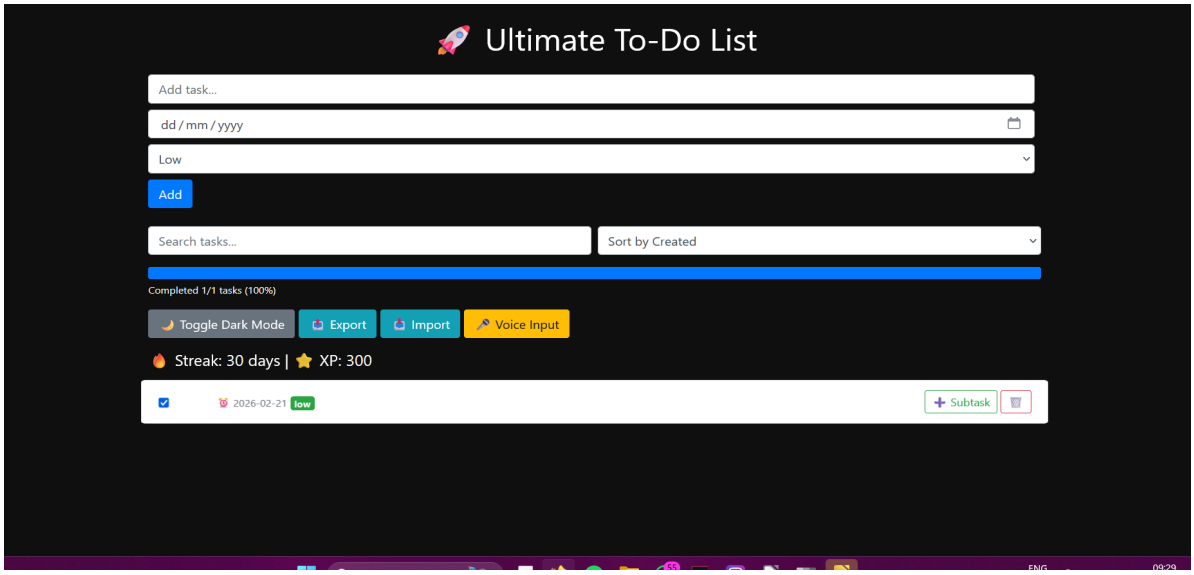
- Add **dark mode** for enhanced user experience.

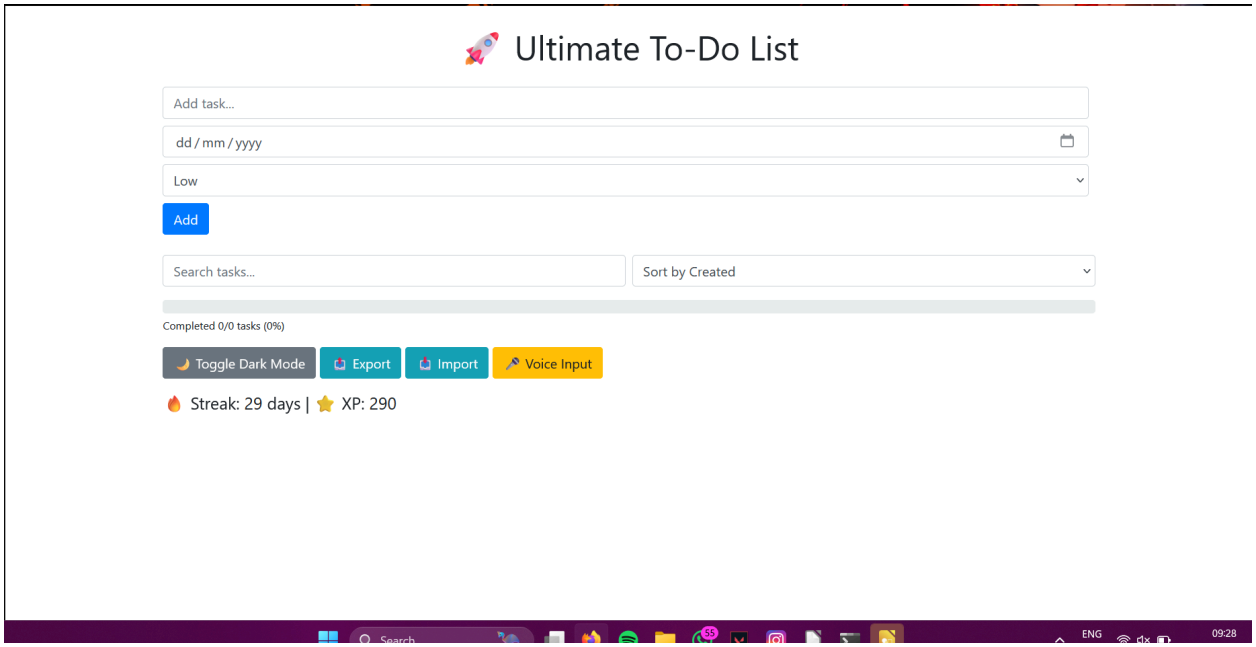
12. Sample Code

```
// Add new task

function addTask() {
```

```
    const text = $('#taskInput').val().trim();
    if (!text) {
        alert("Please enter a task!");
        return;
    }

    const tasks = loadTasks();
    tasks.push({ text, completed: false });
    saveTasks(tasks);

    $('#taskInput').val('');  // Clear input
    renderTasks();            // Refresh UI
}
```

13. Screenshots of Final Output





14. Conclusion

The To-Do List application successfully demonstrates how fundamental web technologies—HTML, CSS, JavaScript, jQuery, and Bootstrap—can be combined to create a practical and interactive productivity tool. The project provides essential task management features such as adding, editing, deleting, completing, and filtering tasks, while ensuring data persistence through local storage. It offers a responsive and user-friendly interface suitable for a wide range of devices. This project also lays the foundation for future enhancements, including task prioritization, reminders, categories, and cloud integration, making it a scalable and versatile solution for personal task management.

15. References

L&T LMS: https://learn.lntedutech.com/Landing/MyCourse