**Software Requirements Specification (SRS) for To-Do List Web Application**

**1. Introduction**

1.1 Purpose  
The purpose of this document is to provide a comprehensive overview of the requirements for developing a To-Do List Web Application. This application will enable users to manage their tasks effectively by adding, viewing, marking as complete or incomplete, and deleting tasks.

1.2 Scope  
The To-Do List Web Application will be a web-based platform accessible through modern web browsers. It will allow users to perform various task management operations, providing a user-friendly interface for task entry and manipulation.

**2. Overall Description**

2.1 Product Perspective  
The To-Do List Web Application will be developed as a standalone system, independent of any other software. However, it may interact with browsers and utilize HTML, CSS, and JavaScript for frontend development. Additionally, it may utilize Node.js and MongoDB for backend functionalities.

2.2 User Classes and Characteristics  
The primary users of the application will be individuals seeking a digital platform to organize their tasks effectively. Users may vary in technical proficiency, ranging from novice to experienced individuals.

2.3 Operating Environment  
The application will be accessible via modern web browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge. It will support multiple operating systems, including Windows, macOS, and Linux.

**3. Functional Requirements**

3.1 Task Management  
3.1.1 Add Task

* Users should be able to input task descriptions into a text field.
* Users should be able to add tasks to the list by clicking an "Add Task" button.

3.1.2 View Task

* The application should display all tasks in a list format.
* Each task should include its description and options for marking as complete, incomplete, or deleting.

3.1.3 Mark as Complete/Incomplete

* Users should be able to toggle the completion status of tasks by clicking on them.
* Completed tasks should be visually distinguished from incomplete tasks (e.g., strike-through text).

3.1.4 Delete Task

* Users should be able to remove tasks from the list by clicking on a delete button associated with each task.

3.2 Persistence  
3.2.1 Local Storage

* Task data, including descriptions and completion status, should be persisted in the browser's local storage.
* Data should persist across sessions, allowing users to access their tasks even after refreshing the page or closing the browser.

**4. Non-Functional Requirements**

4.1 Usability

* The application should feature an intuitive user interface with clear navigation and concise instructions.
* Users of all skill levels should be able to use the application effectively without extensive training.

4.2 Performance

* The application should load quickly and respond promptly to user interactions.
* Backend operations, such as adding, viewing, marking, and deleting tasks, should execute efficiently to minimize latency.

4.3 Compatibility

* The application should be compatible with modern web browsers, including Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.
* It should support multiple operating systems, including Windows, macOS, and Linux.

**5. Appendix**

5.1 Glossary

* To-Do List Web Application: A web-based platform for managing tasks effectively.
* Backend: The server-side component of the application responsible for processing requests and interacting with the database.
* Frontend: The client-side component of the application responsible for presenting the user interface and handling user interactions.
* Local Storage: A browser feature that allows web applications to store data locally on the user's device.