To-Do List Application

Introduction:

- **Project Name**: To-Do List Application
- **Developed By**: Lov Gupta
- **Technology Stack**: JavaScript, HTML, CSS
- **Purpose**: The project aims to build a simple yet functional to-do list where users can add, remove, and view tasks. The application utilizes JavaScript arrays and loops to manage the list of tasks dynamically.

Project Objectives:

- Implement a to-do list application that allows users to:
  - Add new tasks.
  - Remove completed tasks.
  - Display the list of tasks dynamically.
- Use JavaScript arrays to store tasks.
- Use loops to dynamically display and update tasks in the list.

Project Requirements:

- **Frontend**:
  - HTML for the layout and structure of the to-do list.
  - CSS for styling the interface.
- **JavaScript**:
  - Arrays to store tasks.
  - Loops to iterate over the array and update the task list on the UI.

System Design:

- ## Task Management:
    - **Add Tasks**: Users can input tasks into a text field and add them to the list using an "Add" button. Tasks are stored in an array.
    - **Remove Tasks**: Users can click a "Remove" button to delete tasks from the list.
    - **Display Tasks**: A loop is used to display all tasks from the array on the web page.
- **User Interface**: A minimal design with an input field, "Add" button, and a list of tasks that updates dynamically.

Implementation:

- **JavaScript Array**: Use an array to store the list of tasks.
- **Loops**: Use loops (such as for or forEach) to iterate over the array and display tasks on the page.
- **Event Handling**: Add event listeners for buttons to add or remove tasks dynamically.
- **DOM Manipulation**: Update the DOM whenever a task is added or removed.

Features:

- **Add Tasks**: Allows users to add tasks to their list.
- **Remove Tasks**: Users can remove tasks from the list when they are done.
- **Dynamic Updates**: The task list updates in real-time as tasks are added or removed.

Challenges and Solutions:

- **Dynamic List Management**: Managing the display of the task list using loops and ensuring it updates correctly when tasks are added or removed.
- **Efficient Task Removal**: Using array methods (splice) to efficiently remove tasks from the list.

Future Scope:

- Add task prioritization or categories.
- Add an option to mark tasks as completed instead of removing them.
- Implement a local storage feature to save the list even after refreshing the page.

Conclusion:

The To-Do List application successfully demonstrates the use of JavaScript arrays and loops to build a dynamic and interactive task management system. The project is a good foundation for learning basic JavaScript concepts and can be further extended with more features.