Week 2: JavaScript Fundamentals

Interactive Task Manager

Description
A responsive and interactive task management application built using **vanilla JavaScript**. It allows users to add, edit, delete, and manage tasks efficiently while storing data in the browser using **localStorage**. The project is part of **Week 2 – JavaScript Fundamentals** and showcases core concepts like DOM manipulation, event handling, arrays, objects, and form validation.

Features

- Add new tasks with due dates and priority
- Edit and delete tasks anytime
- Mark tasks as complete or incomplete
- Filter tasks (All, Pending, Completed)
- Clear all completed tasks
- Data persistence using localStorage
- Form validation with error handling
- Fully responsive layout for all devices

How to Use

1. Enter a task in the input field and click **Add**
2. Use checkboxes to mark tasks as complete/incomplete
3. Click the delete icon to remove a task
4. Click the edit icon to update a task
5. Use filter buttons to switch between All / Pending / Completed
6. Refreshing the page will still keep your tasks (localStorage)

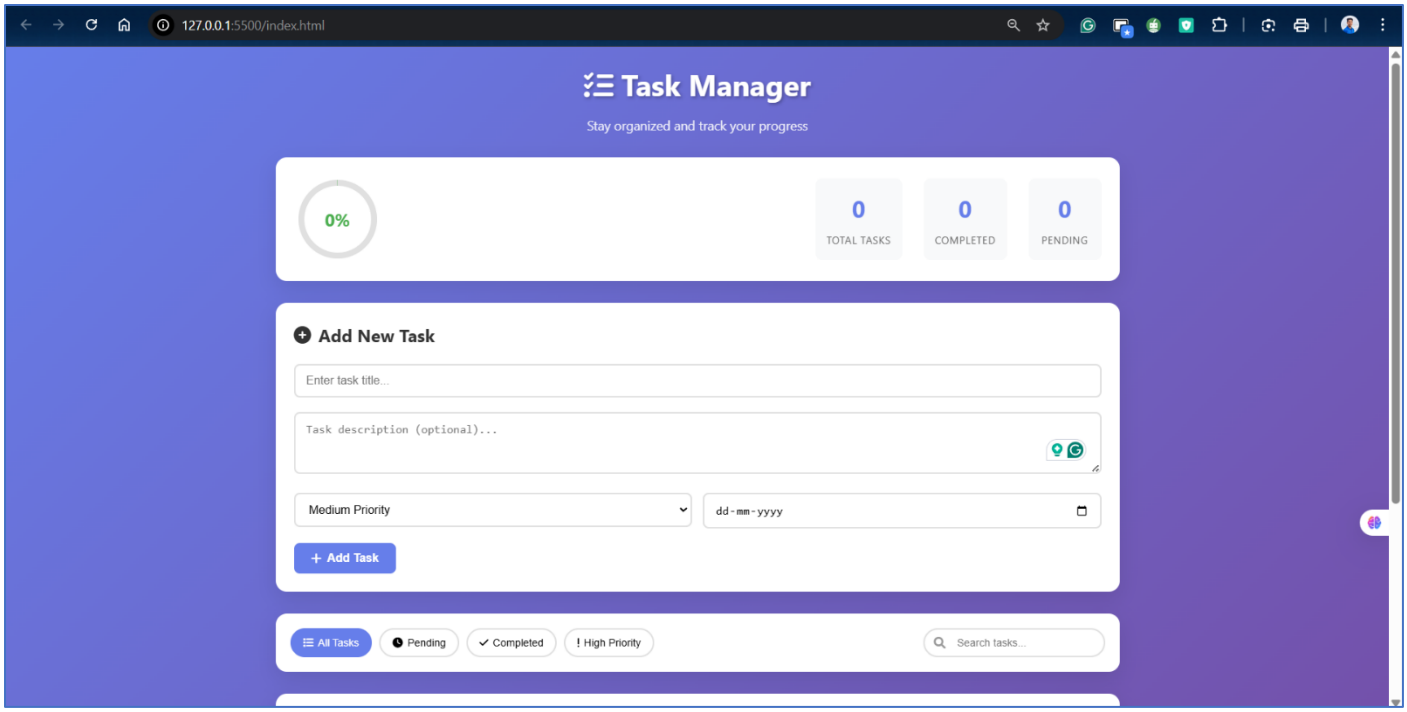
Code Structure

```
task-manager/
├── index.html    # Structural layout of UI
├── styles.css    # Styling, layout, responsive design
├── script.js     # JavaScript logic (DOM, events, storage)
└── README.md     # Documentation & usage guide
```

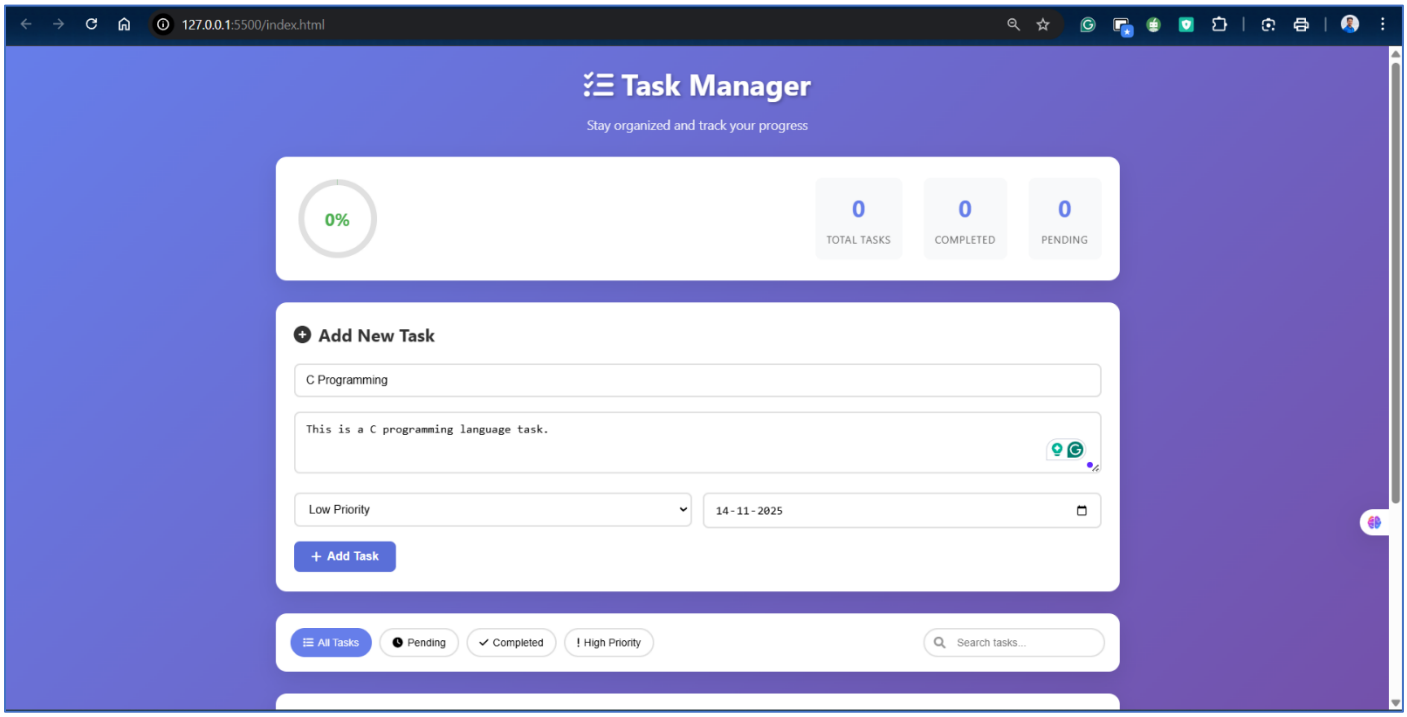
Visual Documentation

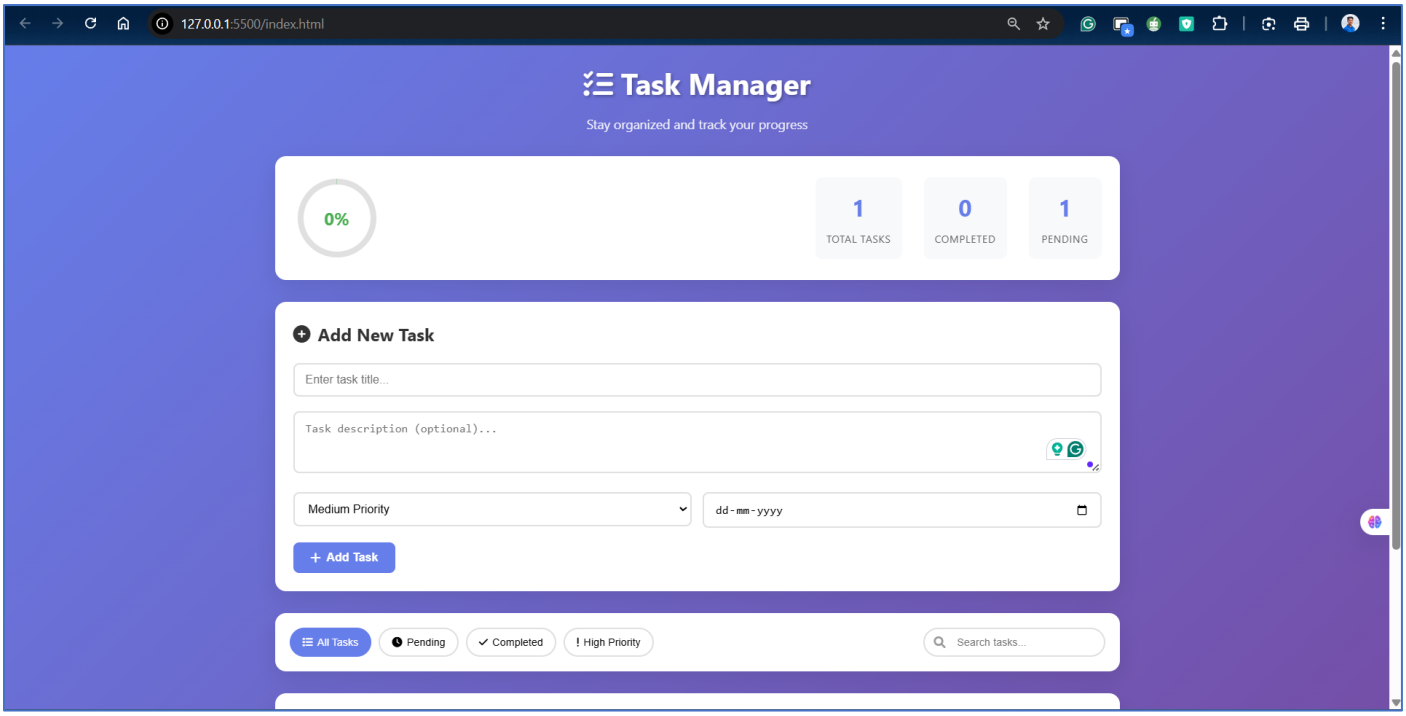
Include screenshots of:

- Home screen with task list

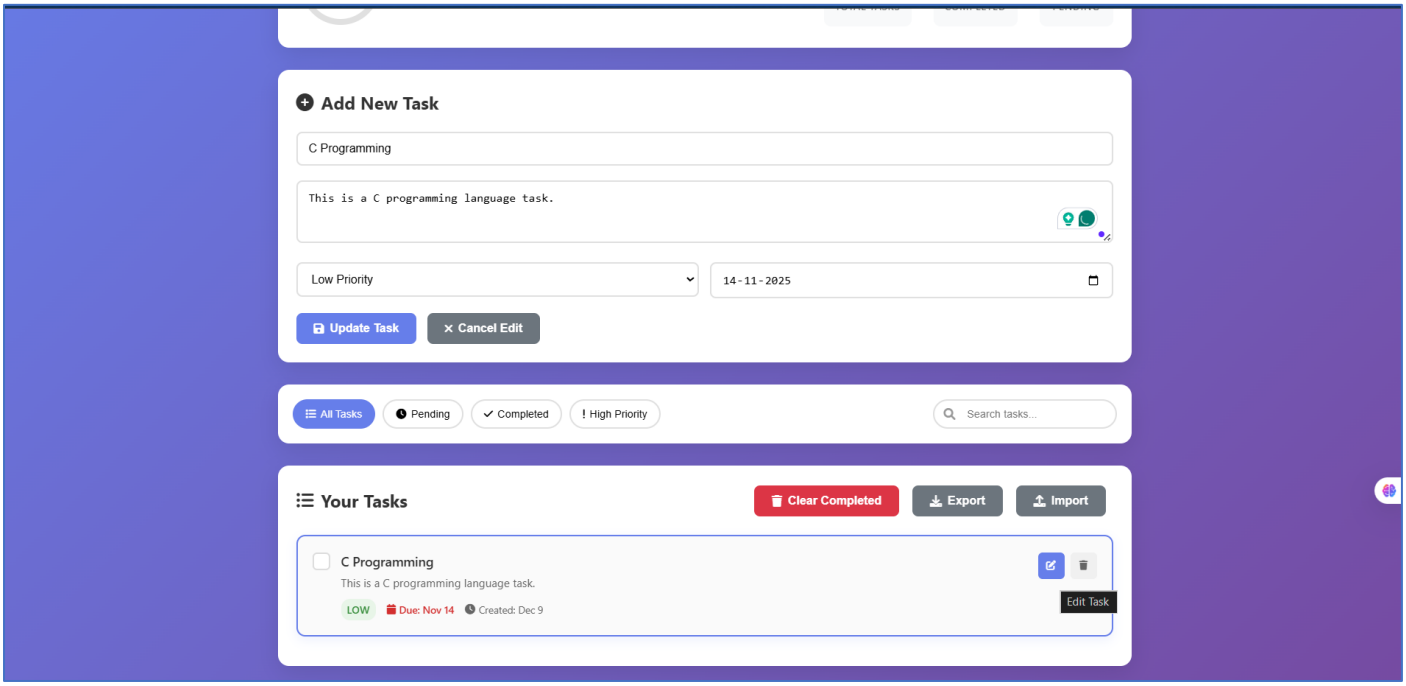


- Adding a task





- Editing a task



Technical Details

- **DOM Manipulation:**
  Creating, editing, deleting, and updating task elements dynamically
- **Event Handling:**
  Button clicks, checkbox changes, form submit, keypress (Enter)
- **Data Structures:**
  Array of task objects → each containing *id, text, date, priority, status*
  Data stored in localStorage for persistence
- **Validation:**
  Empty input checks
  Error message display
  Preventing invalid submissions

Testing Evidence

- Verified task CRUD operations
- Tested filters for all states
- Tested localStorage data persistence
- Checked validation (empty input, incorrect values)
- Tested responsiveness using browser DevTools