To-Do List Application

A simple web-based To-Do List application built with Node.js, Express.js, and MySQL. This application allows users to create, view, update, and delete tasks, with persistent storage in a MySQL database.

Features

1. **Create Tasks**
   Users can create tasks by providing a title and description.
2. **View All Tasks**
   Displays a list of all tasks with options to mark them as complete, edit, or delete.
3. **Mark as Completed**
   Users can mark tasks as completed. Already completed tasks cannot be marked again.
4. **Edit Tasks**
   Users can edit the title and description of existing tasks.
5. **Delete Tasks**
   Users can delete tasks permanently.

Getting Started

Prerequisites

- **Node.js** (v14 or above)
- **npm** (Node Package Manager)
- **MySQL** Database
- **XAMPP** (or similar software for MySQL server)

Code Structure

1. Server (index.js)

- **Routes:**
  - /: Home page with a form to create a task and a button to view all tasks.
  - /tasks: Displays all tasks in a tabular format with options to edit, delete, or mark as complete.
  - /create-task: Handles task creation and saves it to the database.
  - /mark-complete/:id: Marks a task as completed.
  - /edit-task/:id: Renders the task edit form.
  - /update-task/:id: Updates task details in the database.
  - /delete-task/:id: Deletes a task from the database.

2. Static Files

- **CSS (public/style.css):**
  - Defines the overall look and feel of the application, including button styles, table formatting, and layout.

3. Database

- A MySQL database is used for persistent storage. Tasks are stored in the tasks table with fields for:
  - id: Unique identifier for each task.
  - title: Title of the task (required).
  - description: Optional description of the task.
  - is_completed: Boolean flag indicating completion status.

Key Decisions

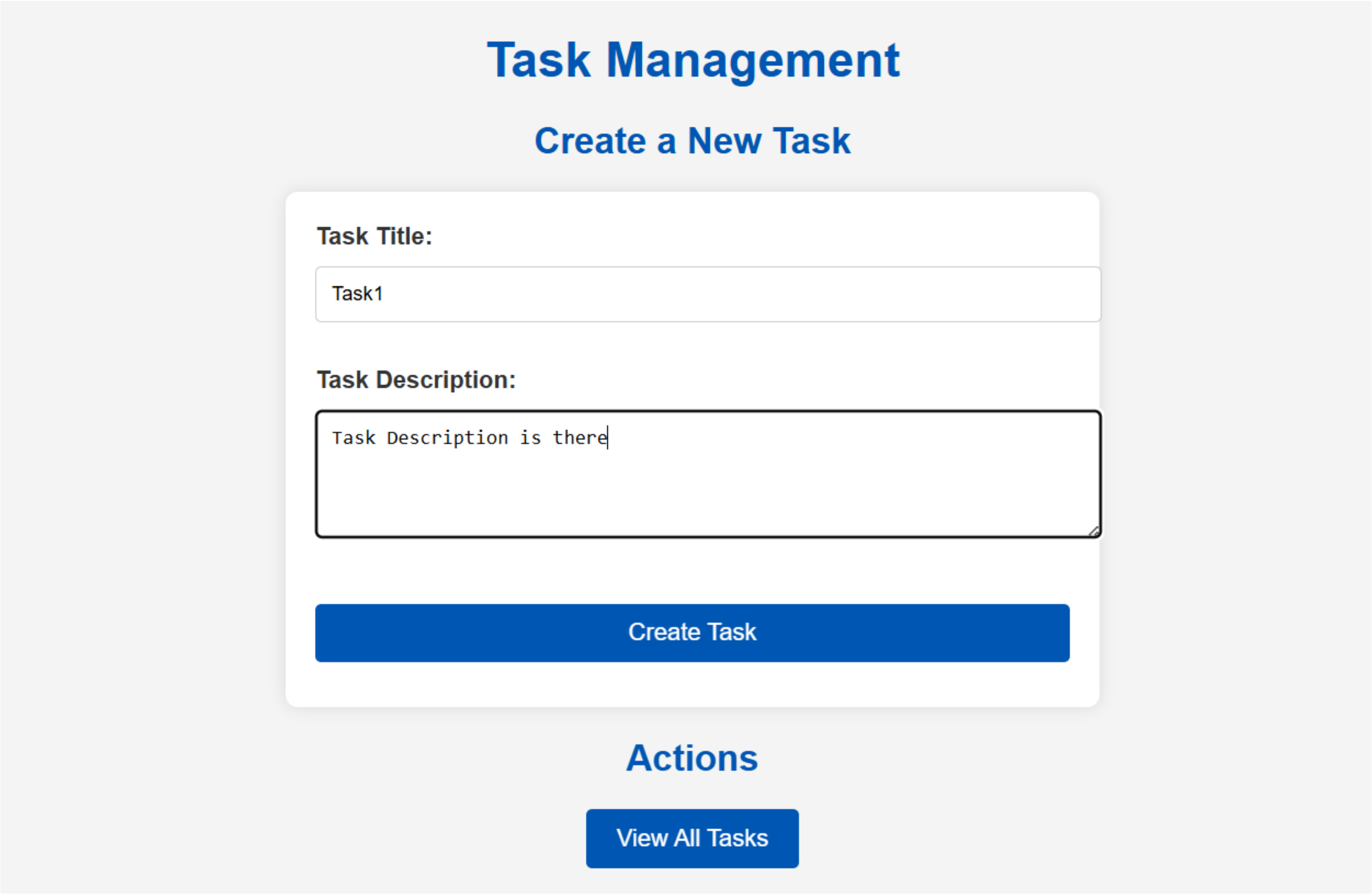
1. **Technology Stack:**
   - **Node.js and Express.js** were chosen for their simplicity and flexibility in building RESTful applications.
   - **MySQL** was selected for its robustness and familiarity.
2. **Validation:**
   - Task title is required when creating or updating a task.
   - Completed tasks cannot be marked again to avoid redundancy.
3. **UI Simplicity:**
   - The UI is designed to be simple and user-friendly, focusing on task management without unnecessary complexity.
4. **Separation of Concerns:**
   - The application separates server-side logic (index.js) from static assets (CSS) to ensure maintainability.

How to Use

1. **Create a Task**
   On the home page (/), fill out the "Task Title" and "Task Description" fields, then click **"Create Task"**.
2. **View Tasks**
   Click on **"View All Tasks"** to see all tasks. Each task has buttons for **editing, deleting, or marking as completed**.
3. **Edit a Task**
   - Click the **"Edit"** button next to a task.
   - Update the title or description, then click **"Update Task"**.
4. **Mark as Completed**
   - Click the **"Complete"** button next to a task.

  - If the task is already completed, the button will be disabled.

5. **Delete a Task**
   - Click the **"Delete"** button next to a task.
   - Confirm the deletion, and the task will be permanently removed.



Task Management

Create a New Task

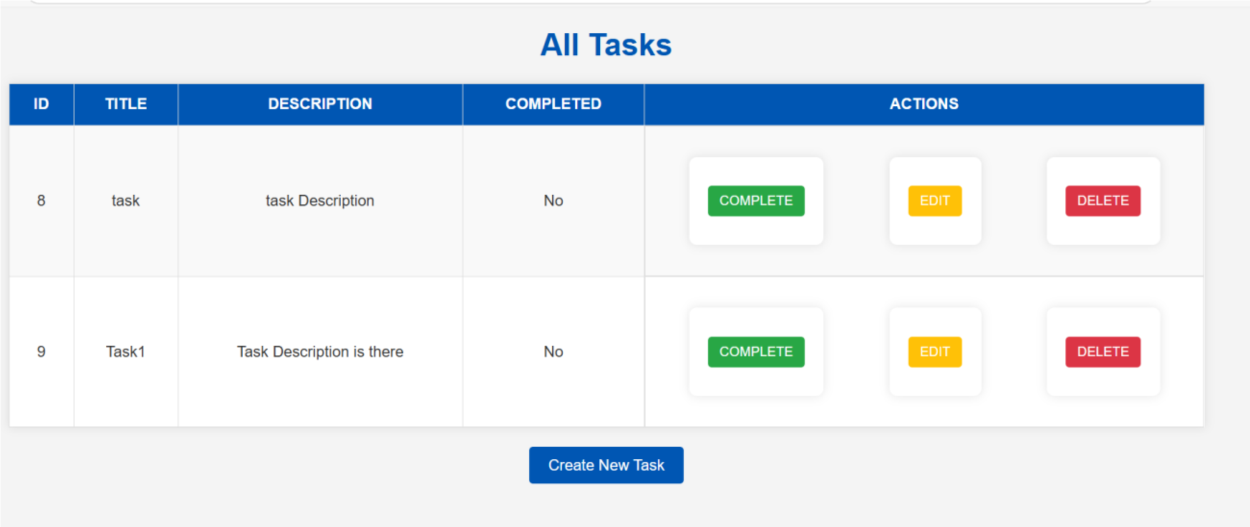
Task Title:

Task1

Task Description:

Task Description is there

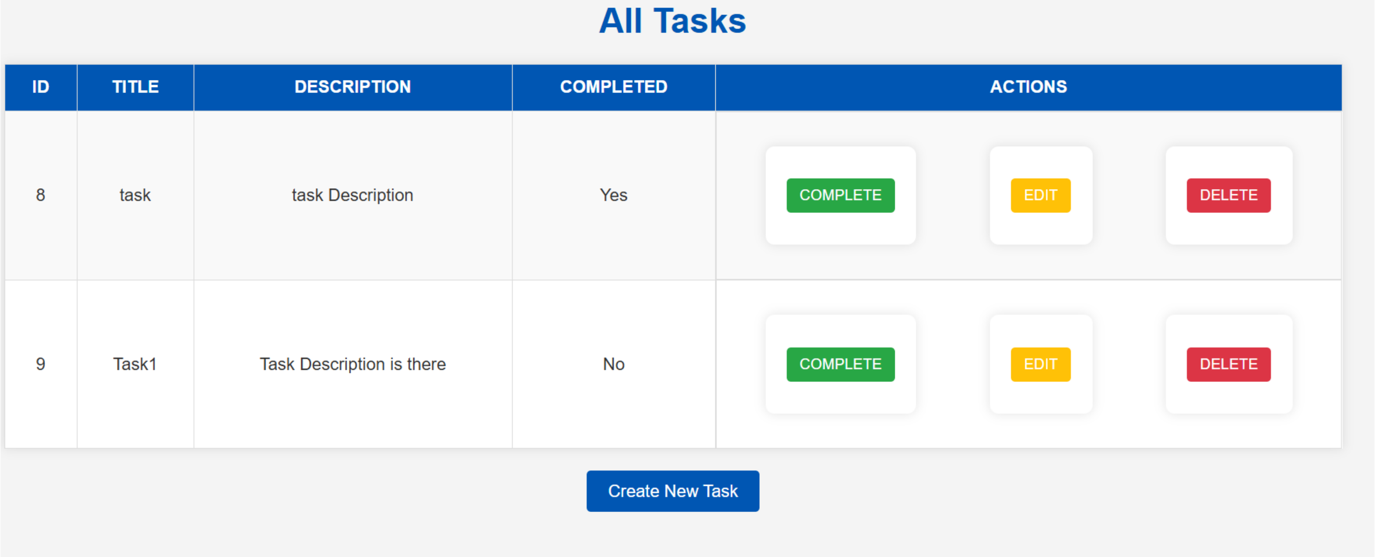
View All Tasks:



All Tasks

ID	TITLE	DESCRIPTION	COMPLETED	ACTIONS
8	task	task Description	No	COMPLETE EDIT DELETE
9	Task1	Task Description is there	No	COMPLETE EDIT DELETE

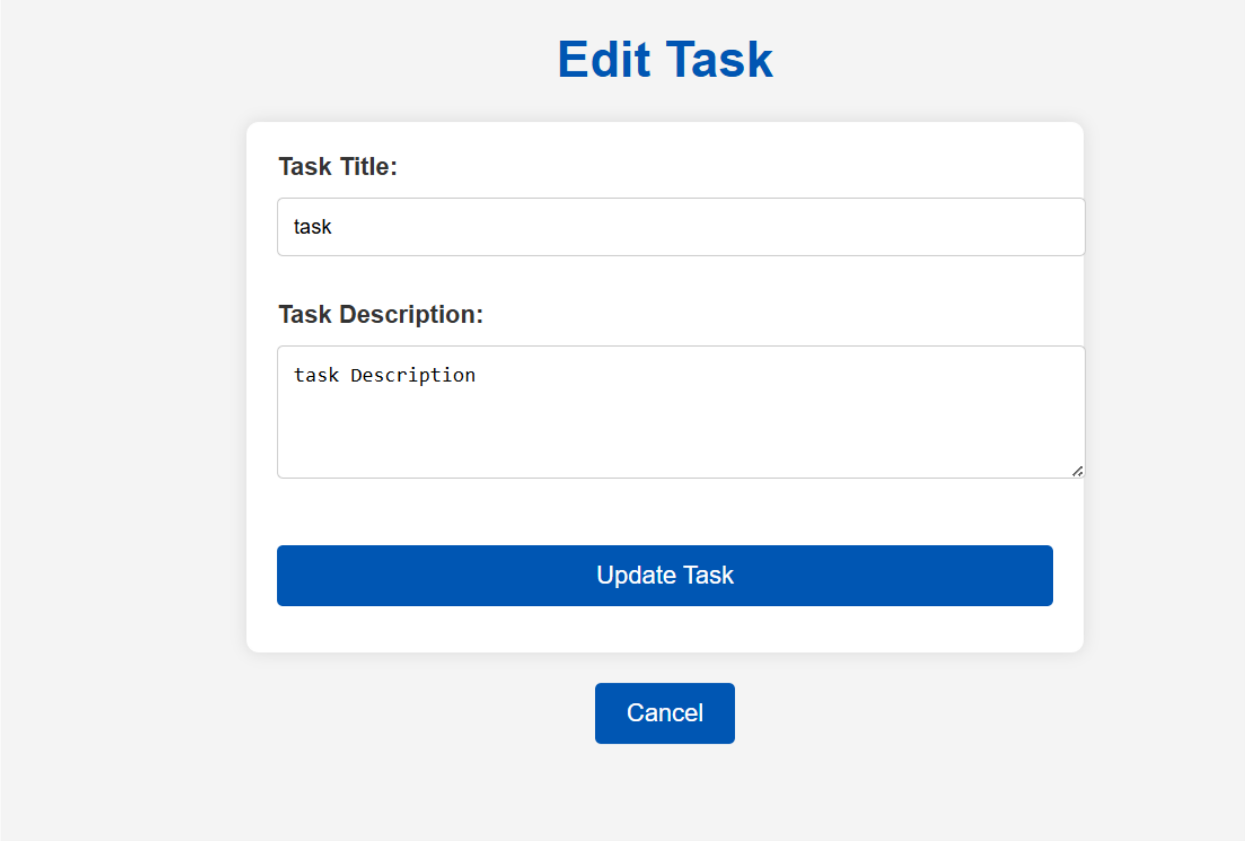
Complete Option : mark as Yes



All Tasks

ID	TITLE	DESCRIPTION	COMPLETED	ACTIONS
8	task	task Description	Yes	COMPLETE EDIT DELETE
9	Task1	Task Description is there	No	COMPLETE EDIT DELETE

Edit Task:



Edit Task

Task Title: task

Task Description: task Description