Task Manager Web Application – AIP Project Report

Submitted by-
Name: Pragati Gupta
UID: 24MCI10255
Section: 24MAM-4A
Semester: 2nd (MCA - AIML)
University: Chandigarh University
Session: 2024–2025

Submitted to-
Ms. Shruti Sharma
Assistant Professor
Emp.Id: E13567

Acknowledgement

I would like to express my sincere gratitude to Chandigarh University and the Department of Computer Applications for providing me with the opportunity to work on this project as a part of my Advanced Internet Programming subject.

I am especially thankful to my mentor and subject teacher, Assistant Professor Ms. Shruti Sharma, for their valuable guidance, continuous support, and encouragement throughout the development of this project. Their insightful feedback and expert knowledge were instrumental in the successful completion of my work.

A heartfelt thank you to all my faculty members for their inspiring teachings and to my peers for their motivation and support during this project journey.

Lastly, I am grateful to my family for their unwavering encouragement and belief in me.

Preface

The development of this project report titled "Task Manager Web Application" has been an insightful journey that helped me to apply and integrate various concepts learned during the Advanced Internet Programming (AIP) course. This project combines real-world problemsolving with hands-on implementation using technologies like Node.js, MySQL, ReactJS, and Bootstrap.

The purpose of this project is to design and develop an interactive, responsive task management application where users can efficiently create, view, edit, and manage their daily tasks. It gave me an opportunity to learn full-stack development, RESTful API integration, state management in React, and backend connectivity with MySQL.

This report covers the aim of the project, the task workflow, technologies used, flowchart, code structure, outcomes, screenshots of the UI, and future enhancements. I hope this project reflects the knowledge and efforts I've put into understanding AIP concepts and applying them to a functional application.

Index

1. Introduction
2. Objectives
3. Technologies used
4. Project flow
5. Code files
6. Functionalities
7. Future enhancements
8. Conclusion

1. Introduction

This project is to design and develop a full-stack web-based Task Manager that allows users to add, update, delete, and filter tasks using modern web technologies. The application focuses on user-friendly interface, responsiveness, and real-time task updates using RESTful APIs.

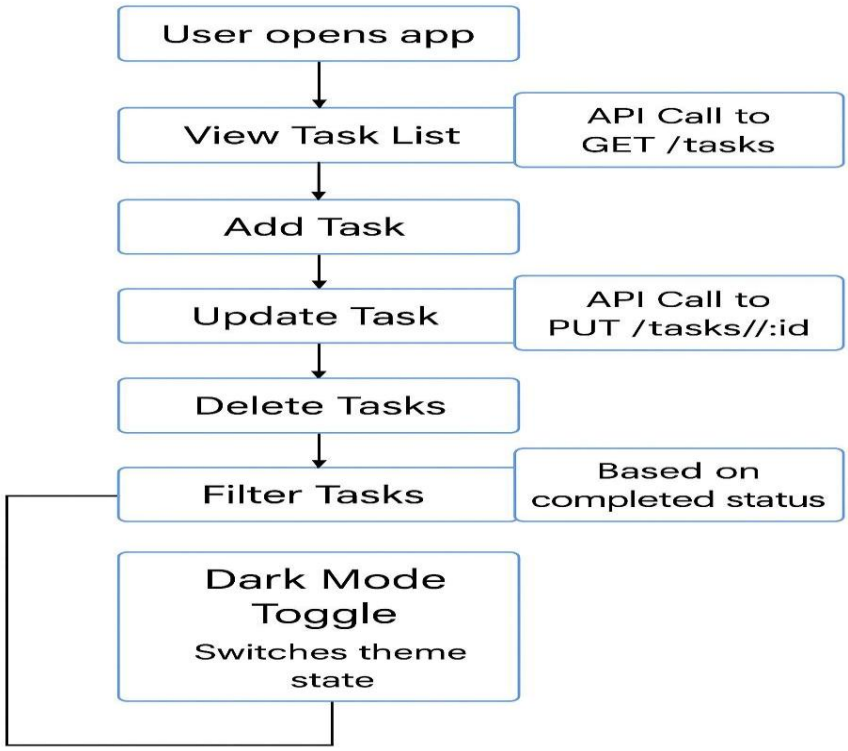
2. Objectives

- Create a responsive frontend using React.js, Bootstrap, and CSS.
- Build a backend API using Node.js and Express.js.
- Set up a MySQL database to store task records.
- Perform CRUD operations (Create, Read, Update, Delete).
- Implement filtering (All, Completed, Pending).
- Enable dark/light mode toggle.
- Handle API integration with frontend using Axios.
- Add edit functionality with form pre-fill.
- Connect backend with MySQL using proper queries.

3. Technologies Used

Technology	Purpose
React.js	Frontend Development (UI)
Node.js	Backend Server Logic
Express.js	Routing and REST API creation
MySQL	Database to store task data
Axios	API calls from frontend
Bootstrap	Responsive UI Components
CSS	Custom Styling
JavaScript (ES6)	Logic and interaction handling

4. Project Flow



5. Code Files

- Backend Files:

- index.js – Main server file with API routes const express = require('express'); const cors = require('cors'); const mysql = require('mysql'); const app = express();

app.use(cors()); app.use(express.json());

// MySQL connection setup const db = mysql.createConnection({
host:
'localhost', user: 'root',
password: 'Pragati5@#$', // Use your own MySQL password here database:
'taskdb'
});

db.connect((err) => {
if (err) throw err;
console.log(' MySQL
Connected!');
});

```
// Default route app.get('/', (req, res)
=>  {       res.send('      Backend  is
working!');
});

// Get all tasks app.get('/tasks', (req, res) => {  db.query('SELECT
* FROM tasks', (err, result) => {    if
(err) return res.status(500).send(err);    res.send(result);
 });
});

// Add new task
app.post('/tasks', (req, res) => {   console.log("Incoming
Task:", req.body);
 const { name, due_date, priority, completed } = req.body;  const sql = 'INSERT INTO  tasks
(name, due_date, priority, completed) VALUES (?, ?, ?,
?)';
 db.query(sql, [name, due_date, priority, completed || false], (err, result) => {    if
(err) {
    console.error("Insert error:", err);
return res.status(500).send(err);
  }
  res.send({ message: 'Task added!', id: result.insertId });
 });
});

// Update task (mark as completed or update details)
app.put('/tasks/:id', (req, res) => {  const { name, due_date, priority,
completed } = req.body;
 const sql = 'UPDATE tasks SET name=?, due_date=?, priority=?, completed=? WHERE
id=?';
 db.query(sql, [name, due_date, priority, completed, req.params.id], (err) => {    if
(err) return res.status(500).send(err);    res.send({ message: 'Task updated!'
});
 });
});

// Delete task app.delete('/tasks/:id',
(req, res) => {   db.query('DELETE FROM tasks WHERE id=?',
[req.params.id], (err) => {
  if (err) return res.status(500).send(err);
res.send({ message: 'Task deleted!' });
 });
});

// Start server app.listen(5000, () => {  console.log('  Server running on
http://localhost:5000');
```

});

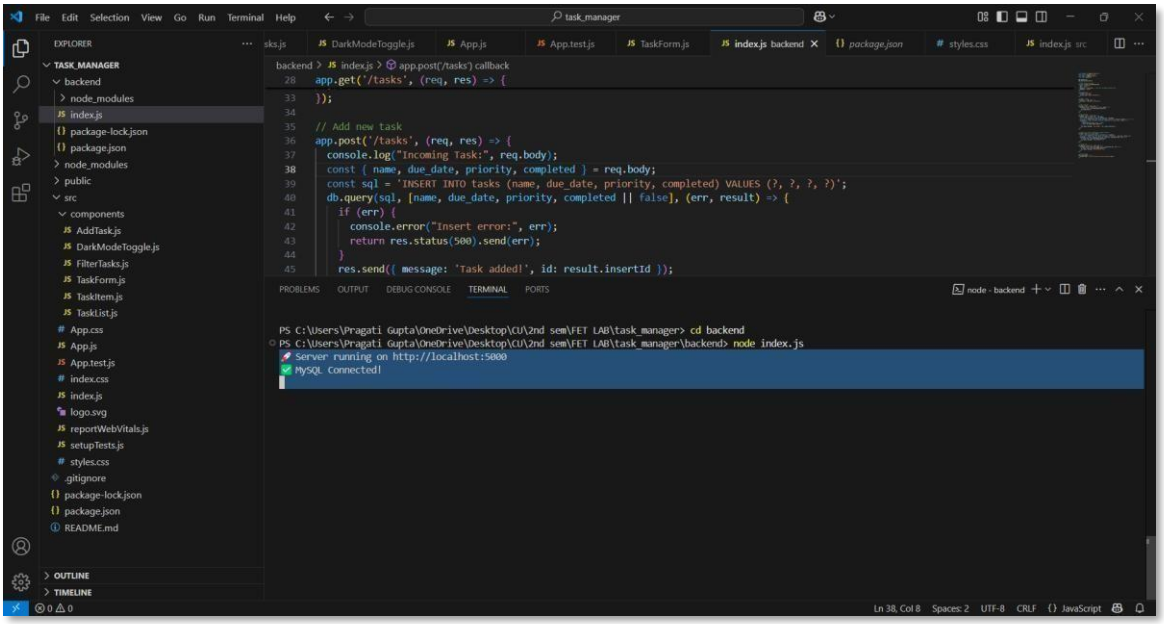
package.json – Backend dependencies

```
{
 "name": "backend",
 "version": "1.0.0",
 "main": "index.js",
 "scripts": {
  "test": "echo \"Error: no test specified\" && exit 1"
 },
 "keywords": [],
 "author": "",
 "license": "ISC",
 "description": "",
 "dependencies": {
  "cors": "^2.8.5",
  "express": "^5.1.0",
  "mysql": "^2.18.1"
 } }
```

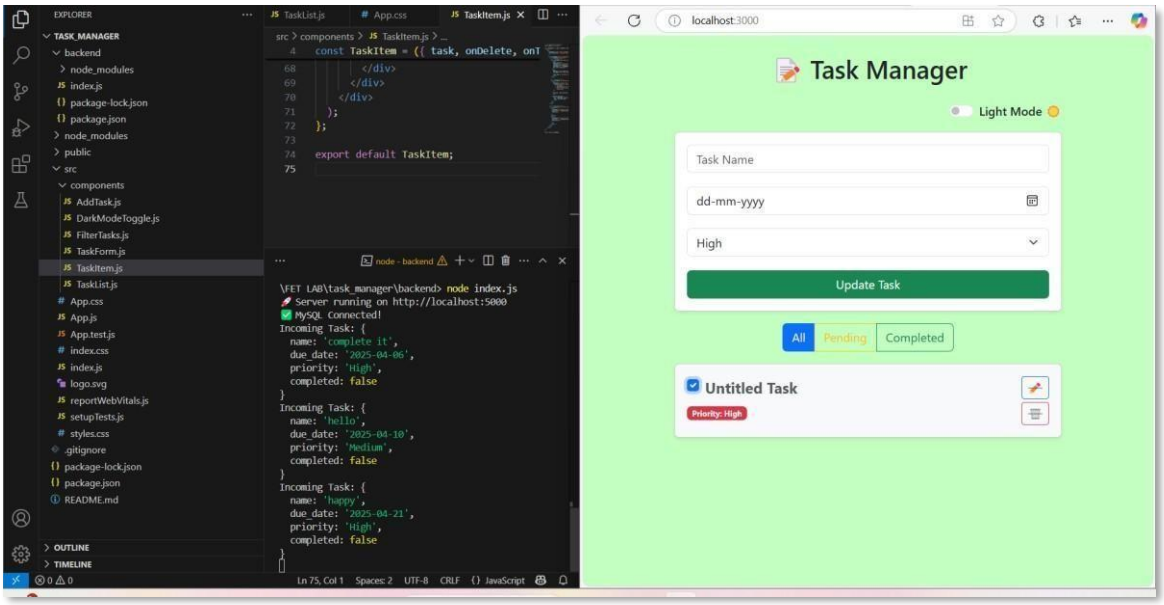
- Frontend Files:
  - App.js – Main component handling layout & routing
  - AddTask.js – Form component to add/update tasks o TaskList.js – Displays list of tasks o TaskItem.js – Displays individual task cards o DarkModeToggle.js – Toggle between dark and light mode o App.css – Custom styles

6. Functionalities

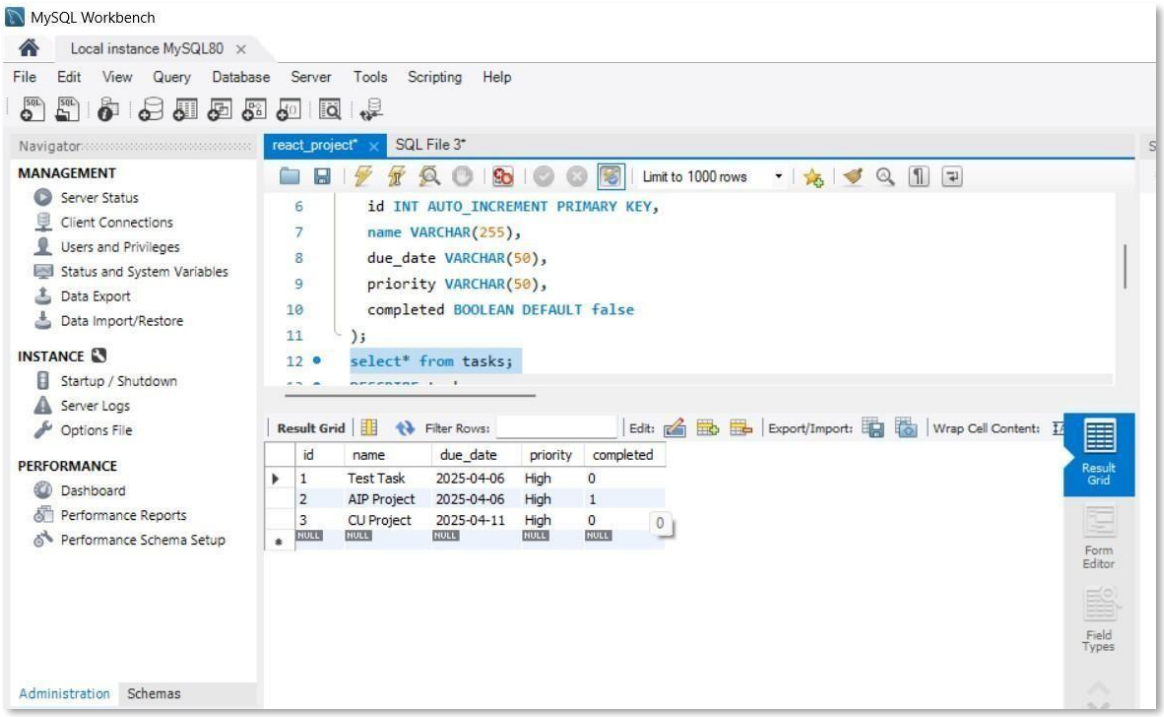
- Successfully connected frontend and backend.



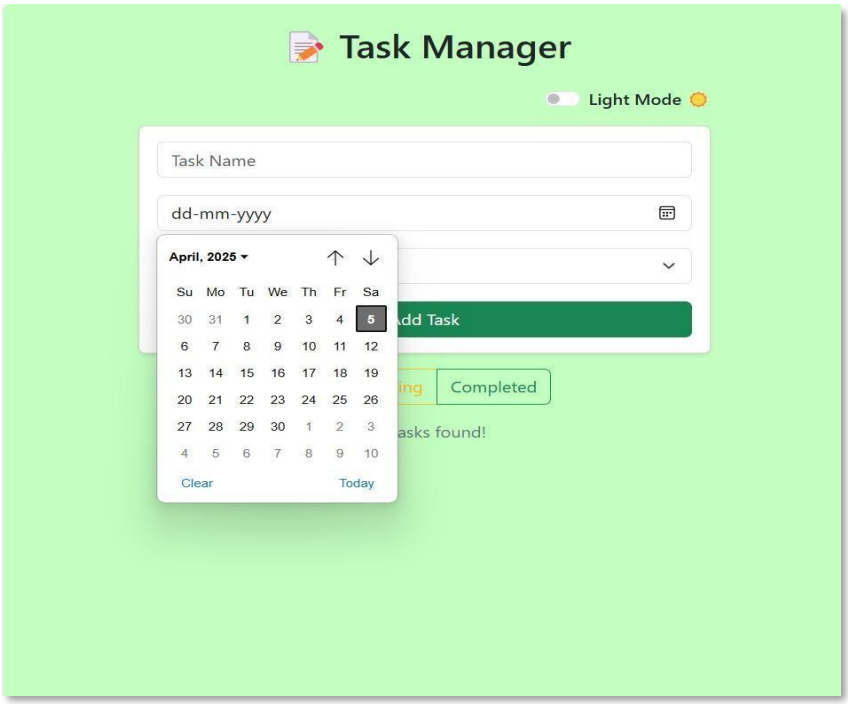
- User can add, edit, delete, and mark tasks.

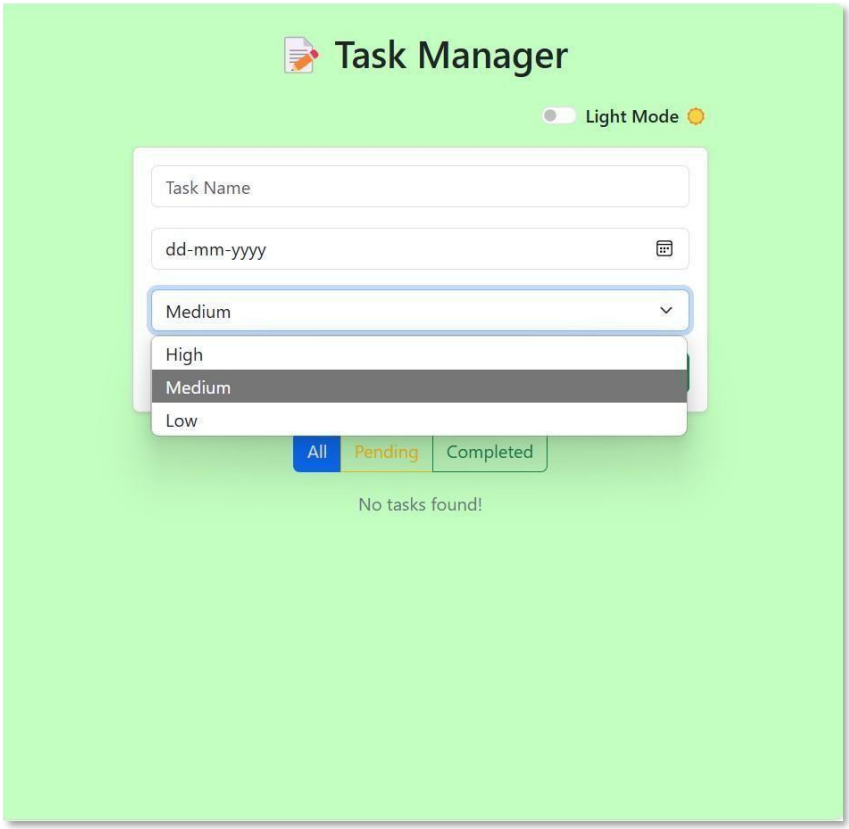


- Tasks are stored in MySQL with proper status.

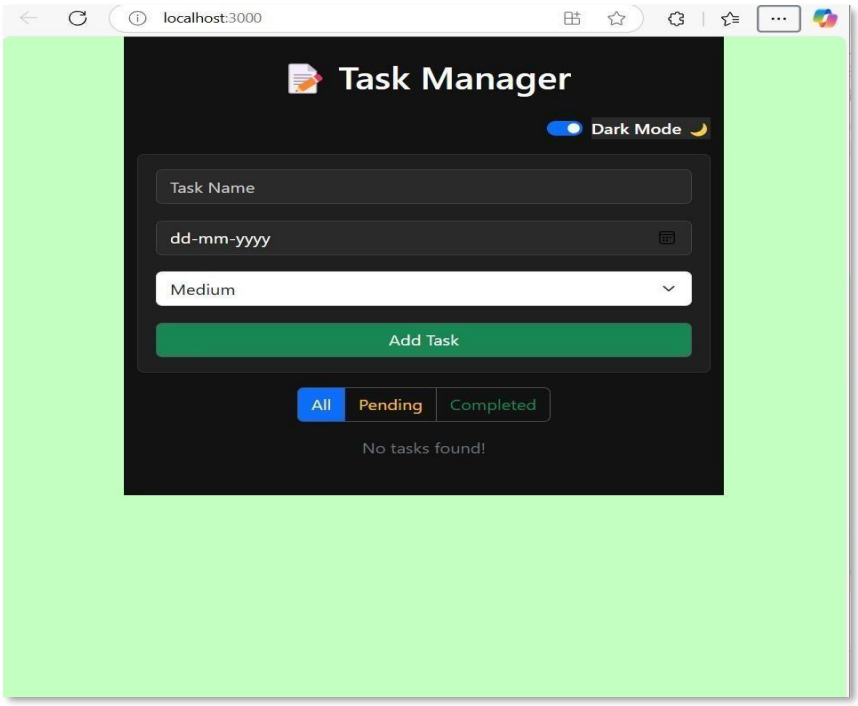


- Filter works in real-time.
- Smooth UI/UX with responsive design.





- Dark/light mode for better accessibility.



7. Future Enhancements

- Add user authentication (Login/Register).
- Implement drag-and-drop task sorting.
- Push notifications for due tasks.
- Add calendar view for due dates.
- Add mobile app version using React Native.

8. Conclusion

The Task Manager project effectively demonstrates the integration of frontend and backend using the MERN-like stack (React + Node.js + MySQL). It covers core topics from Advanced Internet Programming such as HTTP request handling, API development, frontend integration, and database connectivity. The project enhances understanding of full-stack web development and promotes clean UI practices.

GITHUB LINK: https://github.com/prag211/AIP-project