Kanban App Project Report

Student: Muhammad Hussain Ahmed Farooqui

Platform: Full-Stack Web Application

# Abstract

This project is a Kanban task management application that allows users to create, read, update, and delete tasks efficiently. The frontend is developed using React and deployed on Netlify. The backend is built using NestJS, Express, Prisma, and PostgreSQL but is currently running locally and not deployed.

# 1. Introduction

Kanban boards are visual tools for managing tasks and workflows. This project implements a digital Kanban board to manage tasks effectively with real-time interaction.

# 2. Objectives

- Implement a task management system with CRUD operations  
- Provide user authentication and security via JWT  
- Deploy the frontend on Netlify  
- Ensure responsive and user-friendly UI

# 3. System Design

- Architecture: Frontend ↔ Backend (local) ↔ Database  
- Folder structure:  
 - Frontend: React components, pages, assets  
 - Backend: Modules (Users, Tasks, Auth), Controllers, Services  
- API Endpoints (Backend, local): /users, /tasks, /auth

# 4. Implementation

- Backend: NestJS + Express framework  
- REST API endpoints with CRUD functionality  
- JWT-based authentication  
- Prisma ORM for database interaction (local development)

- Frontend: React + Vite  
- Components: Kanban Board, Task Cards, User Authentication  
- Integration with local backend APIs

# 5. Deployment

- Frontend deployed on Netlify  
- Backend currently local (not deployed)  
- Environment variables for local development: JWT\_SECRET, DATABASE\_URL, CORS\_ORIGIN

# 6. Testing

- Manual testing of all API endpoints locally  
- Verified CRUD operations on tasks  
- Tested frontend-backend integration locally

# 7. Challenges & Solutions

- Backend not deployed → future deployment can use Vercel or another cloud provider  
- Local database needed → PostgreSQL used locally  
- Ensured frontend works seamlessly with local backend

# 8. Conclusion

The Kanban App frontend is successfully deployed on Netlify. Backend is functional locally, providing full-stack capabilities for testing and development.