WorkSphere Development Journal

Generated on: April 03, 2025

Project: WorkSphere – Remote Team Productivity & Well-being Platform

Developer: Dinesh Chalicheemala

# 📅 Day 1 — Project Kickoff & Full-Stack Setup

🎯 Objective:  
Set up the base full-stack project with Node.js backend and React frontend, and ensure initial API and component integration works.  
  
🔧 What Was Done:  
- Initialized Node.js + Express backend with a test GET endpoint.  
- Built a simple React frontend dashboard (Dashboard.js).  
- Connected frontend to backend via fetch call and displayed static productivity data.  
- Verified integration in the Chrome browser.  
- Pushed initial project structure to GitHub.  
  
🧠 Skills Gained:  
- Full-stack folder structure setup  
- Basic Express.js API creation  
- React state & component basics  
- Connecting frontend to backend using fetch()

# 📅 Day 2 — Database Integration & Dynamic Backend

🎯 Objective:  
Connect MySQL to the backend and serve live data to frontend from database.  
  
🔧 What Was Done:  
- Installed and configured mysql2 for backend.  
- Connected Node.js backend to MySQL database.  
- Created a 'productivity' table.  
- Updated GET endpoint to fetch live data from MySQL.  
- Displayed live productivity records in the dashboard frontend.  
- Ran and verified in browser.  
  
🧠 Skills Gained:  
- MySQL connection with Node.js  
- SQL table creation & data insertion  
- Backend DB queries using db.query()  
- Replacing hardcoded data with live database records

# 📅 Day 3 — Full-Stack Data Flow + Dashboard Revamp

🎯 Objective:  
Enable real users to submit productivity data, validate it, store it in MySQL, and display insights meaningfully in the dashboard.  
  
🔧 What Was Built:  
- React Form: MUI form collecting name, task count, focus hours, stress level  
- POST API: Backend route accepts JSON, validates data, inserts into DB  
- Form Validation: React + Express validations to prevent invalid input  
- Dashboard 2.0: Shows team overview, most productive person, least stressed  
- Deletion Support: Users can remove entries from DB via frontend  
- Real-time Updates: Dashboard auto-refreshes every 30s using setInterval()  
- Summary Cards: Team-wide stats: averages, totals, top performers  
- Cursor Integration: Used AI to scaffold, refactor, and document code faster  
  
🧠 Skills Gained:  
- Full-stack POST handling  
- Form validation (client + server)  
- Data summarization  
- React state syncing with useEffect & map()  
- Modular, responsive MUI layout  
- AI pair programming with Cursor  
  
🛠️ Git Commit Summary:  
git add .  
git commit -m "Day 3: React form + POST API + validations + dashboard summary cards"  
git push