**Project Start: Setup & Planning**

1. ✅ **Created local project folder**: C:\banking-devopsproject
2. ✅ **Initialized Git and linked GitHub repo**
3. ✅ **Planned phase-wise execution structure**
   * Frontend → Backend → Database → Docker → EKS → CI/CD

**🔹 Phase 1: Project Setup ✅**

*Status: Completed*

1. ✅ Created folders: frontend/, backend/, database/, setup/, backup/
2. ✅ Installed tools: Node.js, Git Bash, VS Code
3. ✅ Created initial files: login.html, server.js, and MySQL script
4. ✅ Tested GitHub push/pull and backups

**🔹 Phase 2: Backend Development (Local) ✅**

*Status: Completed*

1. ✅ Installed dependencies (express, mysql2, body-parser, cors)
2. ✅ Created server.js for backend API
3. ✅ Connected to MySQL (bankdb)
4. ✅ Created users table in MySQL
5. ✅ Built and tested these APIs:

* /login → Checks login credentials
* /signup → Registers new users
* /balance → Returns dummy balance
* /transactions → Returns dummy transaction data

1. ✅ Tested backend using Postman & browser
2. ✅ Successfully ran Node server and got MySQL connected

**🔹 Phase 3: Frontend Development ✅**

*Status: Completed*

1. ✅ Created the following frontend pages:

* login.html → User login
* signup.html → New user registration
* balance.html → Displays balance
* transactions.html → Displays dummy transaction history

1. ✅ Connected each HTML to backend APIs using fetch()
2. ✅ Created login.js and other inline scripts
3. ✅ Used Live Server in VS Code to test the app
4. ✅ Implemented transitions:

* After login → navigate to balance page
* From balance → button to view transactions

1. ✅ Fixed CORS issues, script bugs, and response messages
2. ✅ Documented the entire Phase 3 in a downloadable PDF

**Current Status:**

* ✔️ Frontend + Backend + Database (local) are working
* ✔️ Successfully connected all parts
* ✔️ You’ve learned:
  + How APIs work
  + How front-end talks to back-end
  + How Node.js connects to MySQL
  + How login/signup flows are built

**🔜 Next Up: Phase 4 – Docker & Containerization**

**What’s Pending Now (Phase 4 and beyond):**

**🔹 BEFORE Phase 4 (Recommended Now):**

* 🔄 Push complete project to GitHub (I'll help)
* 📦 Zip backup to C:\InfraBankingSolutions\backup\phase3-backup.zip

**🔜 Phase 4: Docker Setup**

*(We will containerize the app)*

* 🐳 Create Dockerfile for backend
* 🐳 Create Dockerfile for frontend
* 🧩 Use Docker Compose to run both
* 🔁 Test app in containers locally

**Phase 4 – Docker Setup Checklist**

🔧 **Goal**: Run both backend and frontend in Docker containers and test them working together.

**🔹 Step 1: Docker Setup (Backend)**

* Create Dockerfile inside backend/
* Define base image (Node.js)
* Copy backend files into the container
* Install dependencies (npm install)
* Expose backend port (3000)
* Set CMD to run server.js
* Create .dockerignore to avoid node\_modules

**🔹 Step 2: Docker Setup (Frontend)**

* Create Dockerfile inside frontend/
* Use nginx as base image
* Copy HTML files to /usr/share/nginx/html/
* Expose frontend port (default 80)
* Optional: Use nginx.conf to configure routing

**🔹 Step 3: Docker Compose Setup**

* Create a docker-compose.yml file in root (InfraBankingSolutions/)
* Define **two services**: backend and frontend
* Connect them using a custom bridge **network**
* Map backend port 3000:3000, frontend 8080:80
* Confirm both containers run with one docker-compose up

**🔹 Step 4: Test Locally**

* Open browser at http://localhost:8080 to load frontend
* Test login, signup, and balance features
* Use browser developer tools (F12) to confirm API calls hit localhost:3000