22 March 2025 11:33

🚀 **Project To-Do List**

## 🛠️ Hour 1-2: Exploratory Data Analysis (EDA)

✅ Load dataset in **Google Colab** or **Jupyter Notebook**
✅ Import dependencies: pandas, numpy, matplotlib, seaborn
✅ Inspect dataset: .info(), .describe(), .isnull().sum(), .nunique()
✅ Clean data: Remove duplicates, handle missing values
✅ Visualize:

- Distribution of transaction amounts
- Transactions over time ✅ Identify key insights

## 🛢️ Hour 3: Database Setup (PostgreSQL on Render)

✅ Convert CSV into PostgreSQL schema ✅ Use pandas.to_sql() to insert data ✅ Deploy PostgreSQL on **Render** (Free Plan) ✅ Copy **DATABASE_URL** for backend

## 🖥️ Hour 4-5: Backend (FastAPI on Render)

✅ Set up FastAPI project: mkdir backend && cd backend
✅ Install dependencies: fastapi, uvicorn, sqlalchemy, psycopg2
✅ Create FastAPI app with a basic / route
✅ Connect to PostgreSQL using SQLAlchemy
✅ Create API endpoints:

- /transactions → Fetch all transactions
- /query → Convert user query to SQL (later step)
  - ✅ Deploy **FastAPI backend** on Render (link with **PostgreSQL**)
  - ✅ Set **DATABASE_URL** as an environment variable in Render

## 🎨 Hour 6-7: Frontend (React + Vite on Vercel)

✅ Create React app: npx create-vite@latest frontend --template react
✅ Install dependencies: axios, recharts
✅ Fetch data from backend using axios
✅ Display charts using recharts
✅ Create:

- Line chart for transactions over time
- Bar chart for category-wise spending ✅ Ensure UI is responsive

## 🌍 Hour 8: Deploy Frontend (Vercel)

✅ Push frontend code to **GitHub**
✅ Deploy frontend on **Vercel**
✅ Set **VITE_BACKEND_URL** as an environment variable in Vercel
✅ Test API & frontend integration

## 💬 Hour 9: Add Natural Language Query (Optional)

✅ Install openai API
✅ Create FastAPI route /query/ to process user queries
✅ Convert user query → SQL using OpenAI
✅ Fetch & return meaningful data
✅ Integrate query function in React frontend

## 🎯 Final Checks & Deployment

✅ Ensure **database queries work correctly**
✅ **Test API responses** in Postman
✅ **Verify frontend & backend connection**
✅ **Deploy & share project** 🚀

## 🔥 Bonus (Post-Deployment Enhancements)

- Add **authentication** (JWT with FastAPI)
- Optimize **database queries**
- Implement **dynamic filtering**