6-Week Roadmap: AI-Powered Loan Eligibility Advisory System

Week 1 – Project Setup & Data Preparation

- • Tech Stack: Backend (FastAPI/Flask), Database (PostgreSQL on AWS RDS), Frontend (React with chatbot placeholder), ML (Scikit-learn).
- • Finalize requirements & architecture (block diagram).
- • Setup GitHub repo (backend, frontend, ML).
- • Collect dataset (Loan Prediction Dataset - Kaggle).
- • Perform EDA (age, income, credit score, loan term, etc.).
- • Data preprocessing: handle missing values, encoding, scaling.

Week 2 – ML Model Development

- • Tech Stack: Scikit-learn, XGBoost, SHAP (explainability).
- • Train models: Logistic Regression, Random Forest, XGBoost.
- • Compare performance with metrics (Accuracy, F1, ROC-AUC).
- • Integrate explainability (SHAP for feature impact).
- • Save best model (joblib/pickle).

Week 3 – Backend Development (API + Auth)

- • Tech Stack: FastAPI + PostgreSQL (AWS RDS) + JWT Auth.
- • Setup backend project (FastAPI).
- • Create DB schema: Users, Loan Applications, Roles.
- • Implement JWT authentication (/register, /login).
- • API endpoints: /loan/apply, /loan/predict, /admin/dashboard.

- • Integrate ML model into API.
- • Add Swagger docs.

Week 4 – Frontend Development (User + Admin UI + Chatbot)

- • Tech Stack: React + Tailwind + Chart.js (admin charts) + Chatbot (Dialogflow/Rasa).
- • Setup React frontend.
- • Build User UI: Loan application form with validation.
- • Show prediction results (eligibility, confidence score, advice).
- • Build Admin Dashboard: loan stats, approval/rejection charts.
- • Integrate chatbot for FAQs and advisory assistance.

Week 5 – Integration, Advisory & Testing

- • Tech Stack: React + FastAPI + SHAP + Jest/Pytest.
- • Connect frontend ↔ backend APIs.
- • Display advisory tips dynamically from SHAP results.
- • Enhance chatbot with advisory insights.
- • Write unit tests (backend, ML) and integration tests (frontend-backend).
- • Security testing (JWT validation, SQL injection prevention).

Week 6 – Deployment & Documentation (AWS)

- • Tech Stack: Docker + AWS (EC2/ECS + RDS + S3 + CloudWatch).
- • Dockerize backend & frontend.
- • Deploy backend on AWS EC2/ECS with Docker.
- • Database on AWS RDS (PostgreSQL).

- • Frontend on AWS S3 + CloudFront (or EC2).
- • Chatbot hosted with backend (or AWS Lambda).
- • Setup monitoring with AWS CloudWatch.
- • Documentation: README, API docs (Swagger), deployment guide.
- • Record demo walkthrough.