

POTHOLE DETECTION PROJECT — FULL FOLDER STRUCTURE + EXECUTION GUIDE

=====

FOLDER STRUCTURE (MERN + PostgreSQL + Python YOLOv8 Microservice)

project-root/

■

■■■ client/ # React Frontend

■ ■■■ public/

■ ■■■ src/

■ ■ ■■■ components/

■ ■ ■■■ pages/

■ ■ ■■■ hooks/

■ ■ ■■■ context/

■ ■ ■■■ utils/

■ ■ ■■■ App.jsx

■ ■■■ package.json

■ ■■■ vite.config.js

■

■■■ server/ # Node.js + Express Backend

■ ■■■ src/

■ ■ ■■■ controllers/

■ ■ ■■■ routes/

■ ■ ■■■ middleware/

■ ■ ■■■ models/

■ ■ ■■■ services/

- ■ ■■■ config/
- ■ ■■■ app.js
- ■■■ uploads/ # Stored user images
- ■■■ package.json
- ■■■ .env
-
- python-service/ # FastAPI YOLOv8 Microservice
- ■■■ best.pt # Your trained pothole model
- ■■■ app.py
- ■■■ utils.py
- ■■■ requirements.txt
- ■■■ static/
-
- database/
- ■■■ init.sql # PostgreSQL + PostGIS setup
- ■■■ migration_scripts/
- ■■■ seed_data.sql
-
- docker/
- ■■■ docker-compose.yml
- ■■■ Dockerfile.server
- ■■■ Dockerfile.client
- ■■■ Dockerfile.python
-
- README.md

HOW THE SYSTEM RUNS

1. User uploads image from React UI.
2. Node receives image → forwards to Python YOLO microservice.
3. Python detects pothole using best.pt → returns JSON.
4. Node saves:
 - image_url
 - latitude & longitude
 - severity
 - confidence
 - detected: true/false
5. Traveller searches Source → Destination.
6. Backend uses PostGIS to find potholes along route.
7. React map (LeafletJS) displays:
 - Route polyline
 - Pothole markers
 - Popup with image + severity.

COMMANDS TO RUN

1. Start Python service:
`uvicorn app:app --reload --host 0.0.0.0 --port 8001`
2. Backend (Node):
`npm install`
`npm run dev`

3. Frontend (React):

npm install

npm run dev

4. PostgreSQL + PostGIS setup:

Run database/init.sql

YOU CAN LOAD THIS PDF IN ANTIGRAVITY TO GENERATE THE FULL PROJECT

Simply upload this PDF to Antigravity and use this prompt:

"Build this entire project exactly as described in the loaded PDF.

Implement all folder structure, APIs, YOLO integration, map features,

PostGIS queries, and full MERN stack."