1. Maritime Vessel Tracking, Port Analytics, and Safety Visualization Platform

2. Objective & Outcomes Objective

Develop a full-stack web platform to provide interactive live vessel tracking, cargo classification, port congestion analytics, and safety overlays using open maritime and weather data sources. This serves shipping companies, port authorities, and maritime insurers with actionable insights and historical audits.

Outcomes

- Deployed REST app with real-time interactive map and vessel details.
- Integration of global maritime APIs for ship movements, port stats, and weather/safety overlays.
- Port congestion and risk visualization tools.
- Historical voyage playback and compliance tracking.
- Real-world experience in time-series analytics, map visualizations, and API integration.

3. Modules to be Implemented

1. Authentication & Role Management
2. Live Vessel Tracking & Metadata
3. Port Congestion Analytics
4. Safety Overlays (piracy, accidents, storms, ocean weather)
5. Historical Voyage Replay & Audit Module
6. Dashboards for companies, port authorities, and insurers
7. Admin Tools (API status, logs, external source management)
8. Deployment, Testing, Documentation

4. Week-wise Module Implementation and High-Level

Requirements Milestone 1: Week 1 & 2 — Project Setup &

Authentication

- Define user roles: Operator, Analyst, Admin.
- Design and implement database schema (users, vessels, ports, voyages, events, notifications).
- Initialize Django REST backend and React frontend projects.
- Implement JWT authentication endpoints, user profile registration, and login UI.

Milestone 2: Week 3 & 4 — Live Tracking & Metadata Integration

- Integrate MarineTraffic and AIS Hub APIs for real-time vessel position and metadata.
- Build vessel search and filter UI (by type, flag, route, cargo, etc.).
- Store and display ship details, positions, and routes on interactive map (frontend).
- Enable basic subscription to selected vessel alerts/status.

Milestone 3: Week 5 & 6 — Port Congestion & Safety Analytics

- Integrate UNCTAD port/trade stats for congestion analysis.
- Implement port dashboard: average wait times, arrivals/departures, congestion alerts.
- Overlay safety data from NOAA (storms, piracy zones, accident history) on the live map.
- Trigger notifications for adverse weather, safety risks, or port congestion events.

Milestone 4: Week 7 & 8 — Historical Replay, Dashboards & Deployment

- Enable voyage history replay and compliance audit features.
- Build company, port, and insurer dashboards with analytics/visualizations.
- Create admin tools for API source management, monitoring, and data exports.
- Finalize UI polish, run tests, deploy app (Render/Heroku/AWS), and prepare documentation/demo materials.

5. Evaluation Criteria

Milestone 1 (Week 2)

- Authentication, database schema, user profile CRUD, and basic UI setup complete.

Milestone 2 (Week 4)

- Vessel tracking and metadata integration working.
- Interactive map and vessel filter UI operational.

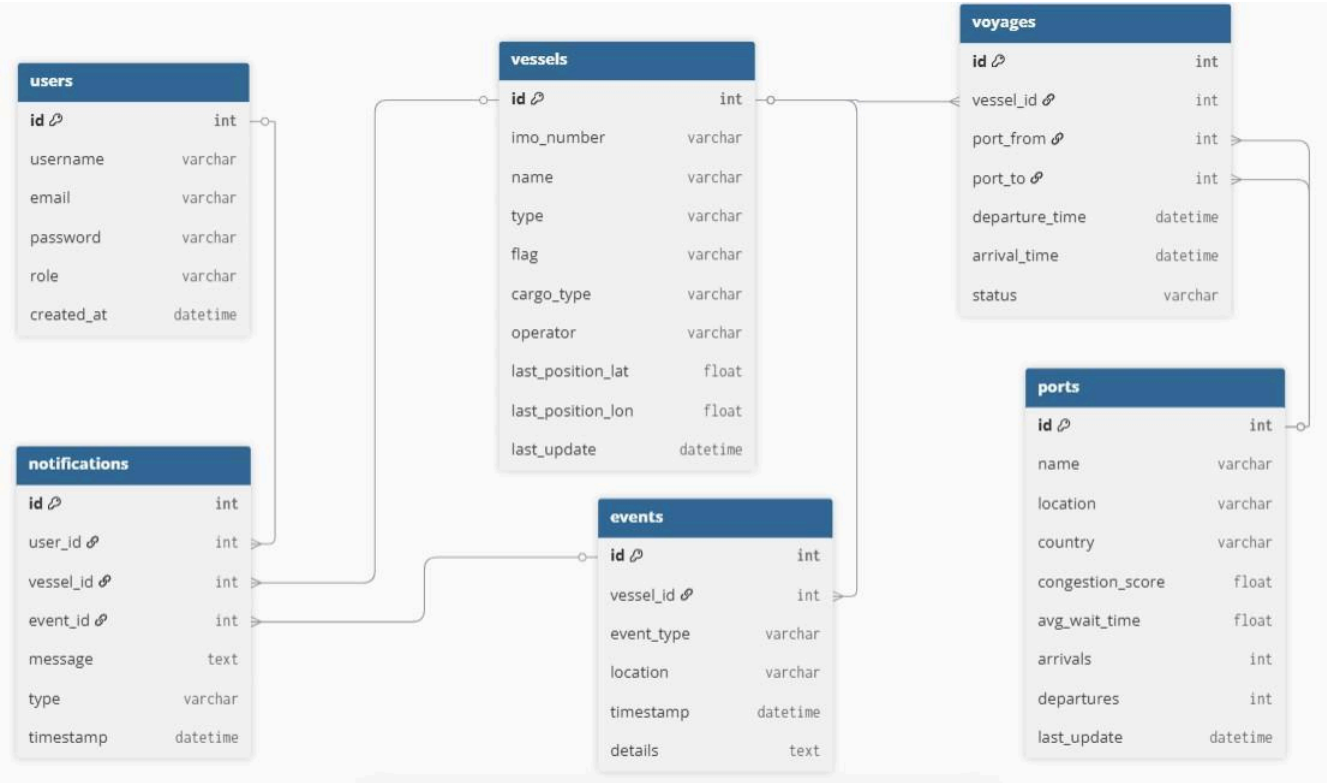
Milestone 3 (Week 6)

- Port congestion analytics and safety overlays live.
- Event notifications functional.

Milestone 4 (Week 8)

- Historical voyage playback, analytics dashboards, admin tools, and full deployment delivered.

6. Design Diagrams



7. Tools & Tech

Stack Programming

Languages

- Backend: Python (Django + Django REST Framework)
- Frontend: JavaScript (React.js)

Database

- SQLite (development), PostgreSQL (production)

Authentication

- JWT (djangorestframework-simplejwt), OAuth2 (optional)

Libraries & Frameworks

- Backend: Django, DRF, Celery (optional for background jobs)
- Frontend: React, React Router, Axios, Redux/Context, Tailwind/Material-UI, Leaflet/Mapbox
- Visualization: Recharts, D3.js
- Testing: pytest/Django test client, Jest/React Testing Library

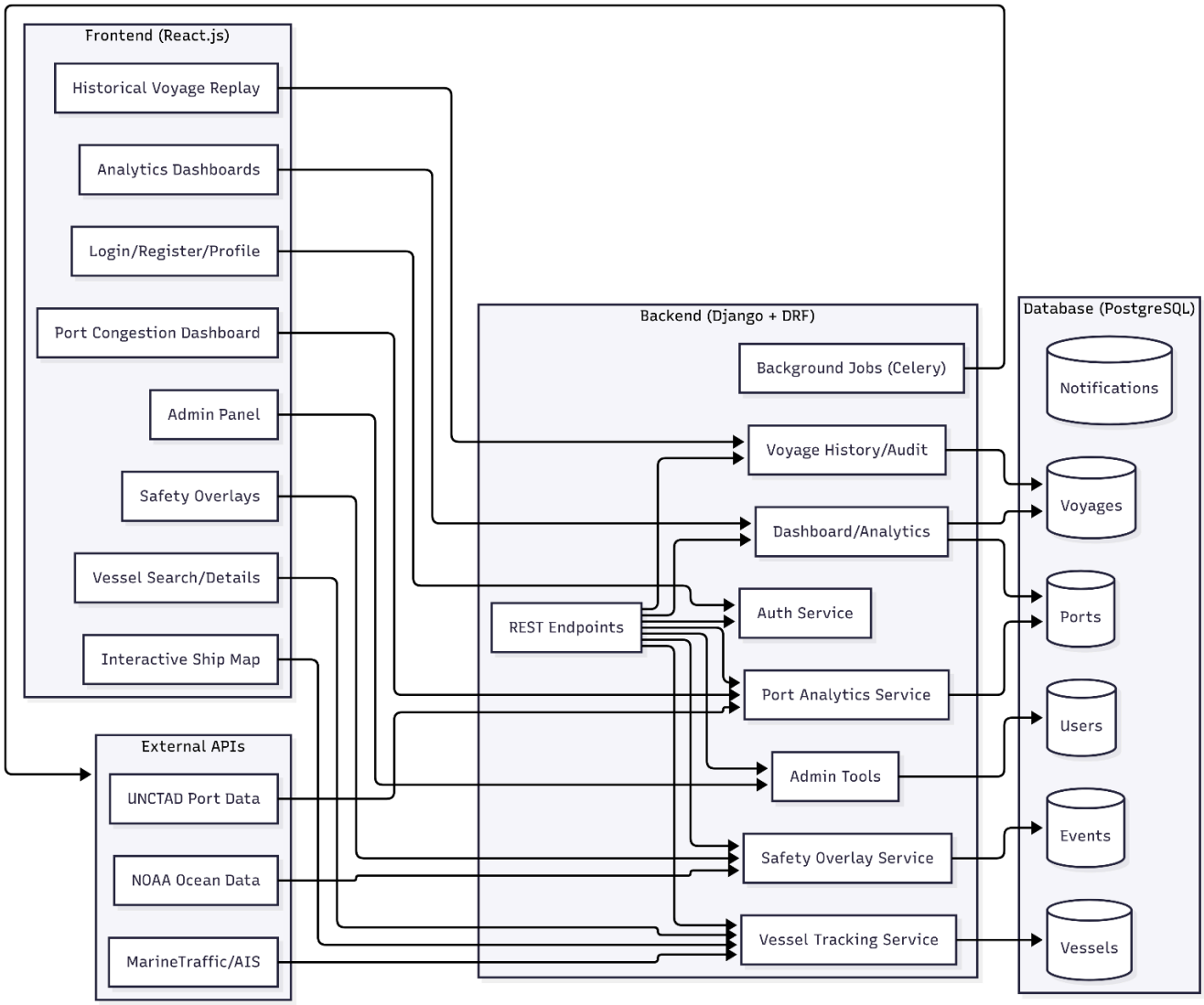
External Data/API Sources

- MarineTraffic API / AIS Hub
- UNCTAD Maritime Data
- NOAA Ocean Data

DevOps & Deployment

- Git + GitHub, Docker & Docker Compose
- CI/CD: GitHub Actions (optional)
- Deployment: Render / Heroku / AWS / DigitalOcean
- Monitoring: Sentry (optional)
- API Testing: Postman / Insomnia

8. Architectural Diagram


Frontend (React.js)
Historical Voyage Replay
Analytics Dashboards
Login/Register/Profile
Port Congestion Dashboard
Admin Panel
Safety Overlays
Vessel Search/Details
Interactive Ship Map
External APIs
UNCTAD Port Data
NOAA Ocean Data
MarineTraffic/AIS
Backend (Django + DRF)
Background Jobs (Celery)
Voyage History/Audit
Dashboard/Analytics
REST Endpoints
Auth Service
Port Analytics Service
Admin Tools
Safety Overlay Service
Vessel Tracking Service
Database (PostgreSQL)
Notifications
Voyages
Ports
Users
Events
Vessels