■■ TGB Event Booking System

Overview

A secure full-stack web application for managing events, tickets, and reservations. Implements JWT authentication, role-based access control (RBAC), relational data modeling, and is packaged for deployment with Docker.

Features

- Authentication & Authorization (JWT, bcrypt, RBAC)
- Event Management (CRUD for Admins/Organizers)
- Reservations & Payments
- Frontend: React (Vite) with dashboards
- Backend: Node.js + Express + Prisma ORM
- Deployment: Docker + docker-compose
- Testing: Jest for backend
- CI/CD: GitHub Actions (sample workflow)

Project Structure

tgb-event-booking/
■■■ backend/ # Express + Prisma backend
■■■ frontend/ # React (Vite) frontend
■■■ docker-compose.yml
■■■ README.md

Quick Start (Local)

1. Clone repo: git clone
2. Backend: cd backend → npm install → npx prisma generate → npx prisma db push → npm run dev
3. Frontend: cd frontend → npm install → npm run dev

Backend → http://localhost:4000 | Frontend → http://localhost:5173

API Endpoints (Sample)

Auth:
- POST /api/auth/register
- POST /api/auth/login
- POST /api/auth/refresh
- POST /api/auth/logout

Events:
- GET /api/events
- POST /api/events (Organizer/Admin only)

Reservations:
- POST /api/reservations (Customer only)
- DELETE /api/reservations/:id

Database Schema

Entities:
- User (role: Admin/Organizer/Customer)
- Event

- Ticket
- Reservation
- Payment
- RefreshToken

SQLite used for demo. For production, switch to PostgreSQL and run migrations.

Docker Deployment

Run locally with:

```
docker-compose up --build
```

Backend → http://localhost:4000 | Frontend → http://localhost:5173