High Level Design Document

Introduction

This High Level Design (HLD) document outlines the architecture and core components for **Eventify - Event Management Backend**. Eventify is a backend API designed for creating and managing events, supporting user authentication and CRUD operations for events. The system is built using Node.js, Express, and PostgreSQL.

---

1. System Architecture Overview

Architecture Description:
Eventify follows a modular, layered architecture. The system exposes RESTful APIs via an Express server, handles business logic in service modules, manages data persistence with PostgreSQL, and secures endpoints with authentication middleware.

Main System Components:

Component	Description
API Layer	Handles HTTP requests/responses, routing, and input validation
Authentication	Manages user registration, login, and JWT-based authentication
Event Service	Implements business logic for event CRUD operations
User Service	Handles user-related operations
Data Access Layer	Interfaces with PostgreSQL for data persistence
Database (Postgres)	Stores users, events, and related data

---

2. Component Interactions

Sequence Step	Interaction Description
1. Client → API Layer	Sends HTTP requests (e.g., login, create event)
2. API Layer → Auth Middleware	Validates authentication (JWT) for protected endpoints
3. API Layer → Service Layer	Forwards validated requests to appropriate service modules
4. Service Layer → Data Layer	Executes database queries/transactions
5. Data Layer → Database	Reads/writes data in PostgreSQL
6. Service Layer → API Layer	Returns results or errors
7. API Layer → Client	Sends HTTP responses

---

3. Data Flow Overview

Data Flow	Description
User Registration/Login	User credentials sent to API, validated, JWT issued
Event CRUD Operations	Authenticated requests processed, events stored/retrieved
Data Persistence	All user/event data stored and managed in PostgreSQL

---

4. Technology Stack

Layer/Function	Technology/Framework
Language	Node.js (JavaScript)
Web Framework	Express.js
Database	PostgreSQL
Authentication	JWT (JSON Web Tokens)
ORM/Query Builder	(Optional) Sequelize/Knex
API Documentation	(Optional) Swagger/OpenAPI

---

5. Scalability & Reliability

- **Scalability:** Stateless API design enables horizontal scaling via load balancers. Database can be scaled vertically or via read replicas.
- **Reliability:** Input validation, error handling, and JWT-based authentication ensure secure and robust operations.
- **Security:** All sensitive endpoints require authentication; passwords are hashed; JWT tokens are used for session management.

End of Document