

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Design Document | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Elastic Event Platform |  | | | |
|  | | | | A Scalable and Fault-Tolerant Cloud-Based Event Management Solution—Software and Data Engineering—Mini Project  |  |  | | --- | --- | | Name | Roll Number | | Prateek Singhal | M22AIE215 | | Aryan Kumar | M23CSA510 | | Harsh Parashar | M22AIE210 | |  | | | |
|  | | |  | | |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  | | |  |
|  | INTRODUCTION | | | | | | |  |
|  |  | | |  |  | | |  |
|  |  | |  | | |  | |  |
|  |  |  | The Cloud-Based Event Management System lets you create, run, book tickets, and pay securely for events. For multi-tenant environments, it guarantees scalability, fault tolerance, and effective performance by using cloud concepts including auto-scaling, containerizing, and load balancing. | | |  |  |  |
|  | | | | |
|  |  |  |  |
|  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decorative | |  |  | | |  |  | |
|  | | Relationships and Design | | | | |  | |
|  | |  |  | | |  |  | |
|  | Architectural Patterns Model-View-Controller (MVC) Pattern   * Models: Represent data and business logic (using ORM models with SQLAlchemy). * Views: In a web API context, represent API responses. * Controllers: FastAPI endpoints handling HTTP requests.   Repository Pattern (Implemented in DAL)   * Each DAL class serves as a repository for its respective model, encapsulating data access logic.   Design Patterns   * Singleton Pattern: Used for DAL (Data Access Layer) instance. * Factory Pattern:Used for payment gateway initialization. | | | | | | |  |
|  | Classes | | |  |  | | |  |
|  | **User**   * ***Attributes*:**   name, email, country\_code, phone\_number, role, company\_id, keycloak\_id, username  **Company**   * **Attributes**: name, address, email, country\_code, phone\_number, registration\_number | | |  | **Event**   * **Attributes**: name, event\_date, event\_time, venue, location\_lat, location\_long, available\_tickets, base\_price, surge\_price, surge\_threshold, version   **Booking**   * **Attributes**: event\_id, user\_id, booking\_time, quantity, total\_cost   **Payment**   * **Attributes**: booking\_id, amount, status, transaction\_id, payment\_time, idempotency\_key | | |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Document | PAGE 3 |  |

|  |
| --- |
| Class Diagram |
| Architecture Diagram |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Design Document | PAGE 4 |  |