

# Event Management System

The Event Management System allows users to browse, book, and manage tickets for various events, such as musical shows, concerts, and theater performances. Users can interact with the platform to search for events, purchase tickets, and manage their bookings. Organizers can also create and manage events, while the system tracks ticket sales and user interactions.

## Microservices

---

- **User Service:** Handles user registration, authentication, and profile management.
- **Event Service:** Manages the creation, editing, and retrieval of events.
- **Ticket Service:** Manages the booking, cancellation, and retrieval of tickets.
- **Payment Service:** Handles payment processing and payment history.
- **Notification Service:** Sends notifications to users about ticket bookings, cancellations, and event updates.
- **Analytics Service:** Tracks user interactions, ticket sales, and event metrics.

## User Service

---

### Features:

- Registration/Login/Logout: User account management.
- Profile Management: View and update user profiles.

### Business Logic:

- Registration: Create a new user account.
- Authentication: Validate user credentials.
- Profile Updates: Modify user information.

## Endpoints:

- `/register (POST)` : Create a new user.
- `/login (POST)` : Authenticate a user.
- `/logout (POST)` : Log out a user.
- `/profile (GET, PUT)` : Retrieve and update user profile.

## Database Tables:

- `users` : `id` , `name` , `email` , `password` , `role`

## Interactions:

Provide user authentication for accessing other services; share user data where needed.

## UI:

- Login/Registration Page: Forms for user login and registration.
- Profile Page: View and edit user profile information.

# Event Service

---

## Features:

- Create/Edit/Delete Events: Manage event details.
- View Events: Display individual and list of events.

## Business Logic:

- CRUD Operations: Create, read, update, delete event details.
- Event Management: Handle event content and metadata.

## Endpoints:

- `/events (POST)` : Create a new event.
- `/events/{id} (GET, PUT, DELETE)` : Retrieve, update, or delete a specific event.
- `/events (GET)` : List all events.

## Database Tables:

- `events` : `id` , `title` , `description` , `date` , `location` , `organizerId` , `categoryId`

## Interactions:

Provide event data to the Ticket Service; receive user data from the User Service.

## UI:

- Event Creation/Edit Page: Form to create or edit an event.
- Event List Page: Display a list of events.
- Event Detail Page: Show detailed view of an event.

## Ticket Service

---

### Features:

- Book/Cancel/View Tickets: Manage ticket bookings.
- View Ticket Details: Display details of individual tickets.

### Business Logic:

- Ticket Booking: Process booking requests and allocate tickets.
- Ticket Management: Handle ticket status and user requests.

### Endpoints:

- `/tickets` (POST) : Book a new ticket.
- `/tickets/{id}` (GET, DELETE) : Retrieve or cancel a specific ticket.
- `/tickets/user/{userId}` (GET) : List all tickets for a user.

## Database Tables:

- `tickets` : `id` , `eventId` , `userId` , `seatNumber` , `price` , `status` , `bookingDate`

## Interactions:

Provide ticket data to the Payment and Analytics Services; link tickets to users and events.

## UI:

- Ticket Booking Page: Interface for booking tickets.
- Ticket Management Page: View and manage ticket bookings.

## Payment Service

---

### Features:

- Process Payments: Handle payment transactions.
- View Payment History: Display payment records.

### Business Logic:

- Payment Processing: Validate and execute payment transactions.
- Payment Management: Record and manage payment details.

### Endpoints:

- `/payments (POST)` : Process a payment.
- `/payments/user/{userId} (GET)` : Retrieve payment history for a user.
- `/payments/ticket/{ticketId} (GET)` : Retrieve payment details for a specific ticket.

### Database Tables:

- `payments` : `id` , `userId` , `ticketId` , `amount` , `date` , `status` , `transactionId`

### Interactions:

Interact with the Ticket Service to confirm ticket bookings upon successful payment; provide payment data to the Analytics Service.

## UI:

- Payment Page: Form for processing payments.

- Payment History Page: View payment history and details.

## Notification Service

---

### Features:

- Send Notifications: Notify users about ticket bookings, cancellations, and event updates.
- Manage Notifications: Handle notification preferences and history.

### Business Logic:

- Notification Management: Create and manage notifications for users.
- Notification Delivery: Send notifications based on triggers such as booking confirmations or event cancellations.

### Endpoints:

- `/notifications` (POST) : Send a new notification.
- `/notifications/user/{userId}` (GET) : List all notifications for a user.
- `/notifications/{id}` (GET) : Retrieve a specific notification.

### Database Tables:

- `notifications` : `id` , `userId` , `content` , `date` , `readStatus`

### Interactions:

Send notifications based on activities from other services (e.g., ticket booking, payment success).

### UI:

- Notification Panel: Display recent notifications.
- Notification Management Page: View and manage all notifications.

# Analytics Service

---

## Features:

- Track User Interactions: Monitor user activities like event searches, ticket bookings, etc.
- Sales Metrics: Analyze metrics related to ticket sales and event popularity.

## Business Logic:

- Interaction Tracking: Record user activities and interactions.
- Metrics Analysis: Analyze and generate reports based on user data and sales figures.

## Endpoints:

- `/analytics/user/{userId}` (GET) : Retrieve interaction data for a user.
- `/analytics/event/{eventId}` (GET) : Retrieve metrics for a specific event.
- `/analytics/ticket/{ticketId}` (GET) : Retrieve metrics for a specific ticket.

## Database Tables:

- `user_interactions` : `id` , `userId` , `interactionType` , `date`
- `event_metrics` : `id` , `eventId` , `metricType` , `value` , `date`
- `ticket_sales` : `id` , `eventId` , `ticketId` , `quantity` , `date`

## Interactions:

Collect and analyze data from other services; provide reports and analytics to event organizers and administrators.

## UI:

- Analytics Dashboard: Visualize data and metrics.
- Reports Page: Generate and view detailed reports on user interactions and sales.