



Dhirubhai Ambani University Technology

Formerly DA-IICT

**Project Created By:-
Group-2**

**Guided By:-
Prof. JayPrakash Lalchandani**

Team members:-
202412013 - Parth Chauhan
202412026 - Jain Tisha Navinkumar
202412029 - Joshi Preksha
202412032 - kaival Shah
202412066 - Patel Dhruv Alkeshbhai
202412079 - Rajvi Khatri
202412101 - Shah Yashvi Devangkumar
202412119 - Ritik kalal

1. Project Description:-

Project Title: 'EventHub - Event Management System'

Description:

- EventHub is a modern, all-in-one event management system designed to simplify event creation, promotion, ticketing, attendee engagement, and analytics. It provides both web (admin-focused) applications that help organizers efficiently manage all aspects of events while giving attendees an easy discovery and booking experience.

Target Users:

- 1. Individual Organizers** – hosting workshops, classes, parties, or small events.
- 2. Event Agencies** – managing multiple events, ticketing, and promotions.
- 3. Corporates** – arranging conferences, exhibitions, and webinars.
- 4. Communities & Non-Profits** – running fundraisers, awareness programs, and festivals.
- 5. Attendees** – individuals who want to discover, attend, and track events easily.

Use Cases:

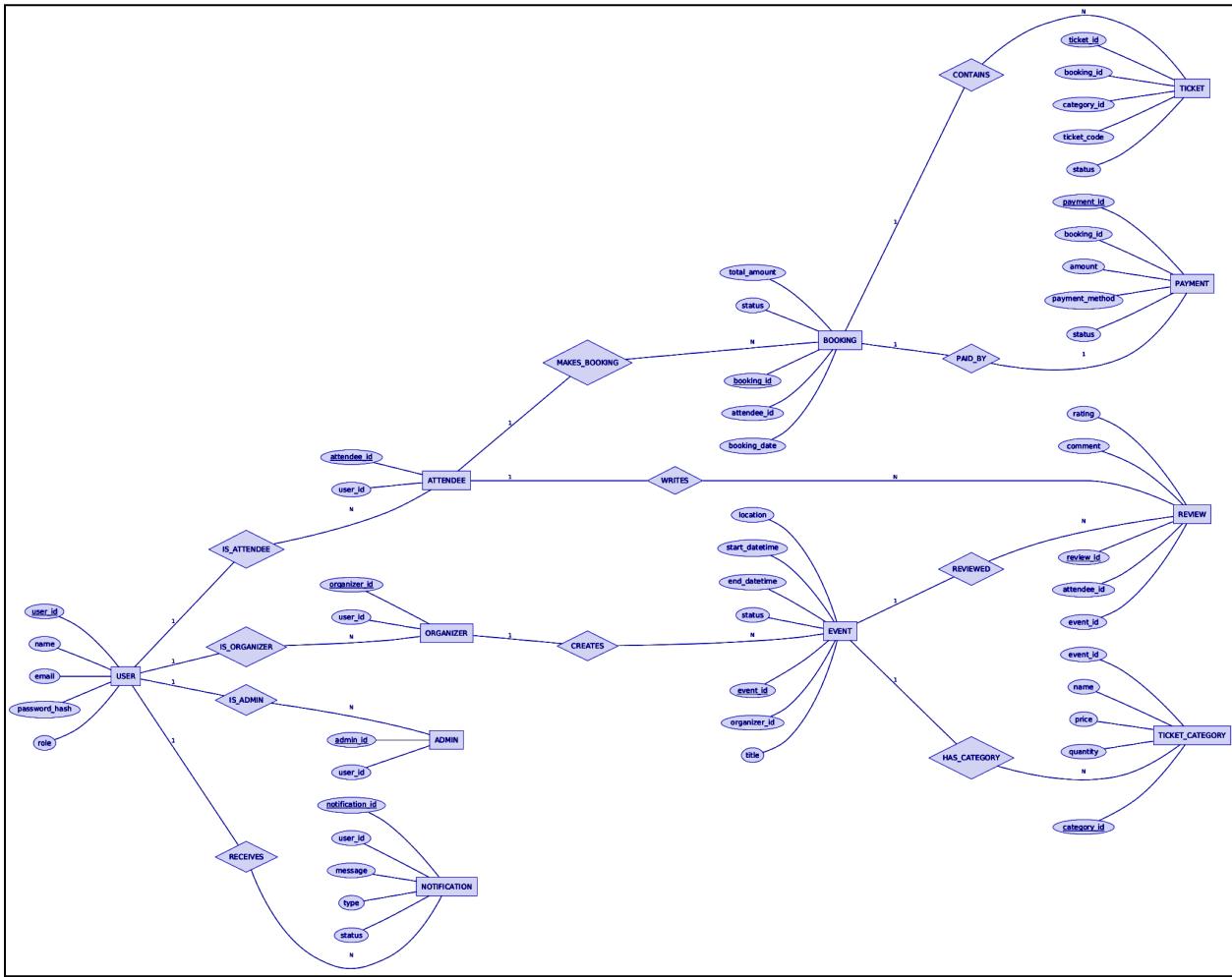
● For Organizers:

- Create customized events with images, categories, and descriptions.
- Define ticket types (free, paid, early bird, discounts).
- Track organizer's revenue, and ticket sales.

● For Attendees:

- Discover upcoming events nearby through search and categories.
- Register or buy tickets with secure online payment options.
- Get booking reminders and updates.
- Store and show tickets digitally at the venue.
- Share events with friends and community via social media.

ER Diagram:-



Frontend Architecture:-

Current Frontend Features Implemented:-

1. Authentication & User Access

- **Login & Signup pages** designed with clean Tailwind components.
- Basic validation and responsive form layouts.
- Modular structure for authentication logic (as seen in `AUTHENTICATION_SETUP.md`).

2. Homepage & Navigation

- Dynamic **Navbar** and **Footer** components with category links.
- Responsive design tested on mobile and desktop.
- Footer includes social media and category sections (updated per `FOOTER_CATEGORIES_UPDATE.md`).

3. Routing & Layout

- React Router integrated for smooth page navigation.
- Folder structure supports future expansion (e.g., `/pages`, `/components`, `/layouts`).



4. Styling Setup

- Fully configured **TailwindCSS** with `postcss.config.js` and `tailwind.config.js`.
- Global styles defined for buttons, forms, and headings.
- Theme supports light/dark adaptability.



5. Bug Fixes & Performance

- Issues fixed per `BUGFIXES_SUMMARY.md`:
 - Footer overlapping resolved.
 - Layout alignment bugs fixed.
 - Responsive margins and padding improved.
- Performance and lint fixes documented in `COMPLETION_SUMMARY.md`.

Frontend Weekly Breakdown:

Week	Timeline	Tasks Completed
Week 1	15 – 21 Sept	Project setup, initialized React app, added Tailwind CSS, set up basic folder structure (<code>src/components</code> , <code>src/pages</code>).
Week 2	22 – 28 Sept	Created authentication UI (Login, Signup), responsive Navbar & Footer.

Week 3	29 Sept – 5 Oct	Integrated routing between pages (Home, Login, Signup, Dashboard placeholder). Fixed alignment bugs.
Week 4	6 – 12 Oct	Enhanced responsiveness, added color palette & reusable components, performed minor bug fixes, documentation updates (COMPLETION_SUMMARY.md).



Remaining Tasks & Future Work:-



Pages Yet to Be Developed:

Page	Description
About Page	To display platform purpose, mission, and contact details.
Create Event Page	Form for event organizers to create and manage events.
Book Events Page	User-facing page for viewing and booking available events.
Payment Page	Secure checkout interface with payment gateway integration.
User Dashboard	To show user's created/booked events and account settings.

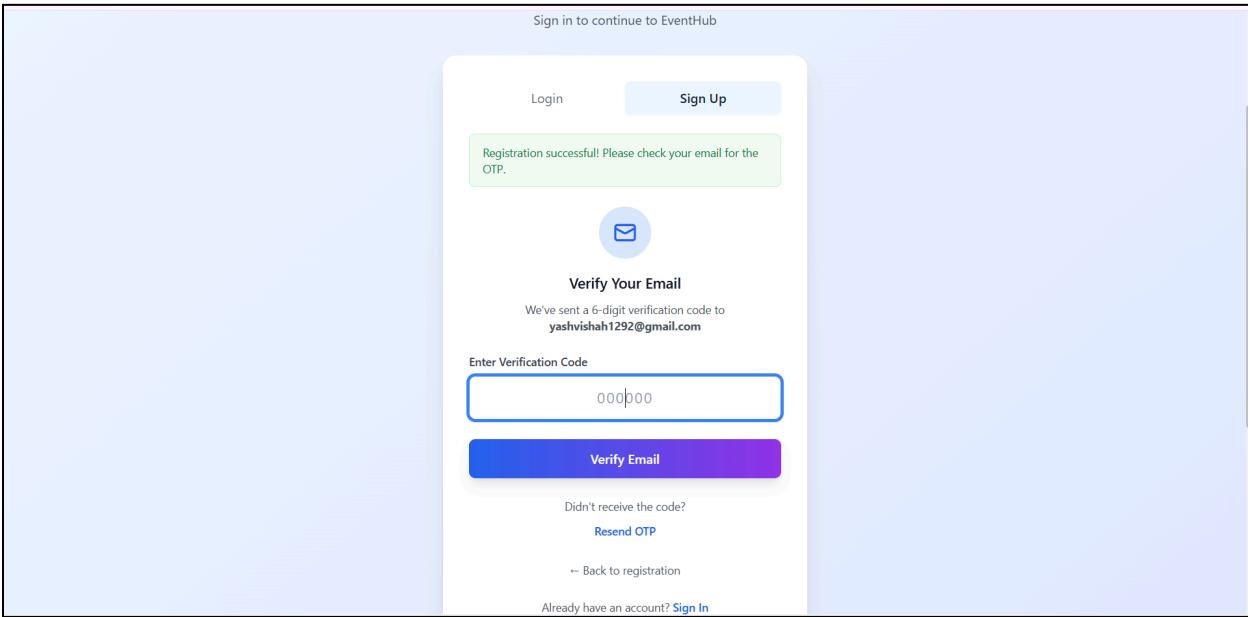


Planned Enhancements:

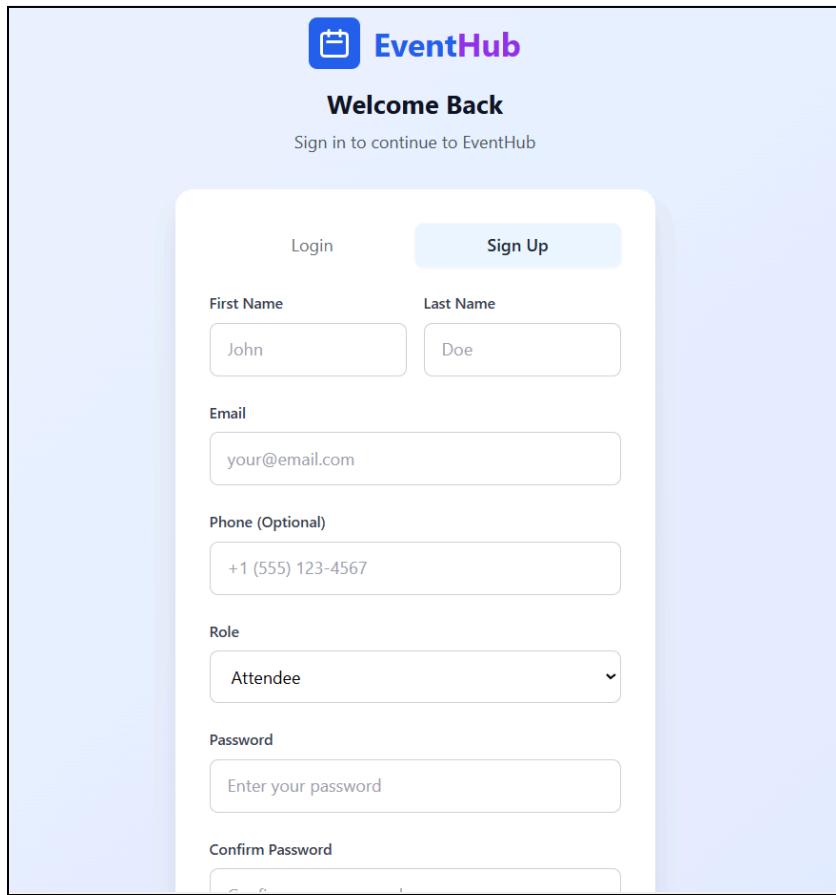
- API Integration with backend endpoints (once ready).
- Form Validation and input sanitization.
- Dynamic Data Rendering for events fetched from backend.
- Responsive Animations for smoother user experience.

Frontend Screenshots :-

Verify Email OTP page:



Sign Up Page:



The image shows a sign-up form for EventHub. At the top, there is a logo consisting of a blue square with a white calendar icon and the text "EventHub" in blue and purple. Below the logo, the text "Welcome Back" is displayed in bold black font, followed by the subtext "Sign in to continue to EventHub". There are two buttons at the top right: "Login" and "Sign Up", with "Sign Up" being highlighted with a light blue background. The form fields are arranged in a grid. The first row contains "First Name" and "Last Name" fields, both containing the placeholder text "John" and "Doe" respectively. The second row contains an "Email" field with the placeholder "your@email.com". The third row contains a "Phone (Optional)" field with the placeholder "+1 (555) 123-4567". The fourth row contains a "Role" dropdown menu set to "Attendee". The fifth row contains a "Password" field with the placeholder "Enter your password". The sixth row contains a "Confirm Password" field with the placeholder "Enter your password again".

First Name	Last Name
John	Doe

Email
your@email.com

Phone (Optional)
+1 (555) 123-4567

Role
Attendee

Password
Enter your password

Confirm Password
Enter your password again

Login Page:



EventHub

Welcome Back

Sign in to continue to EventHub

[Login](#) [Sign Up](#)

Email

Password

Remember me [Forgot password?](#)

[Sign In](#)

Don't have an account? [Create Account](#)

Organiser Dashboard:

 EventHub

Home Events Categories Create Event About

Yashvi Shah Logout

Discover Amazing Events

Find and book tickets to the best events happening around you

[Explore Events](#) [Create Event](#)

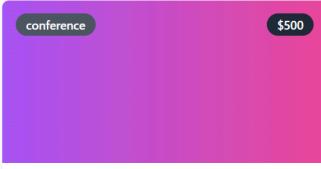
Browse by Category

Find events that match your interests

Featured Events

Don't miss out on these popular events

[View All Events](#)



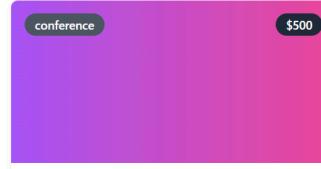
Tech Conference 2025
A one-day event exploring the latest trends in AI, Web3, and Cloud Computing.
Nov 20, 2025 3:30 PM - 11:30 PM Mumbai 2 attending

[View Details](#)



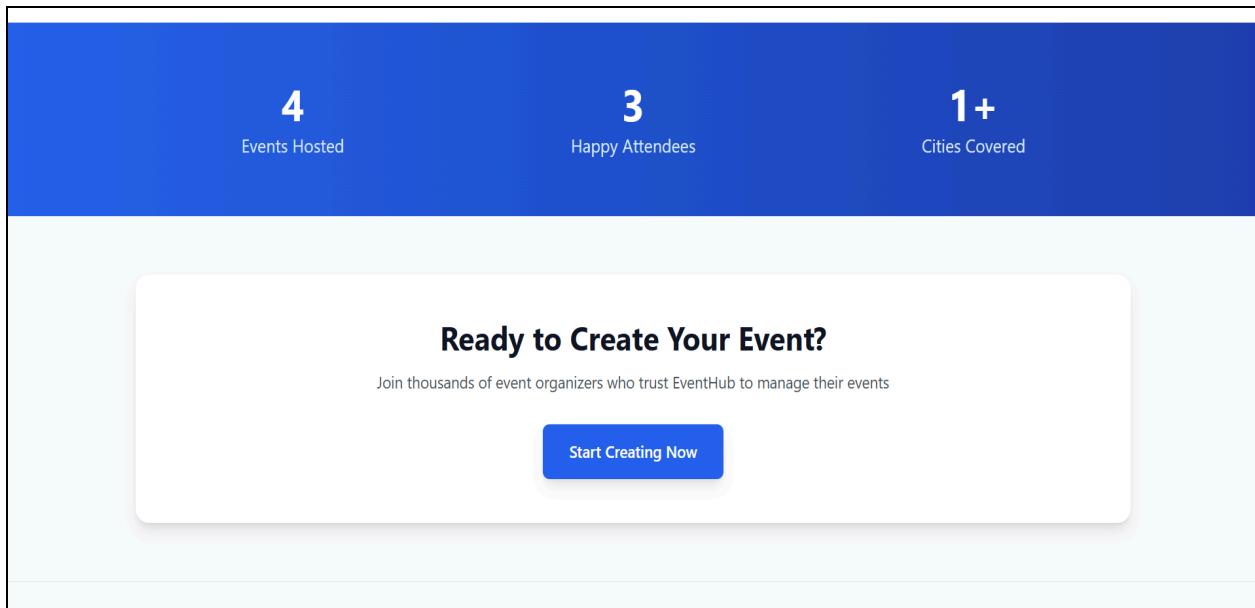
Tech Conference 2025
A one-day event exploring the latest trends in AI, Web3, and Cloud Computing.
Nov 20, 2025 3:30 PM - 11:30 PM Mumbai 1 attending

[View Details](#)



Tech Conference 2025
A one-day event exploring the latest trends in AI, Web3, and Cloud Computing.
Nov 20, 2025 3:30 PM - 11:30 PM Mumbai 0 attending

[View Details](#)



Event Categories page:

A screenshot of the EventHub 'Categories' page. The top navigation bar includes links for Home, Events, Categories (which is underlined), Create Event, and About. A user profile for 'Yashvi Shah' is also visible. The main section is titled 'Event Categories' and features six categories with corresponding icons and descriptions:

- Conference**: Business conferences, tech talks, and professional events. 4 events. [Explore →](#)
- Workshop**: Hands-on learning, skill development, and training. 0 events. [Explore →](#)
- Seminar**: Educational seminars, lectures, and presentations. 0 events. [Explore →](#)
- Networking**: Professional networking, meetups, and social events. 0 events. [Explore →](#)
- Concert**: Live performances, concerts, and music festivals. 0 events. [Explore →](#)
- Festival**: Cultural festivals, food events, and celebrations. 0 events. [Explore →](#)

Event Details page:

The screenshot shows the EventHub website's event details page for a 'Tech Conference 2025'. At the top, there is a navigation bar with links for Home, Events, Categories, Create Event, and About. On the right, there are user profile icons for 'Yashvi Shah' and a 'Logout' button. Below the navigation, a large purple gradient banner features a small 'conference' tag and two circular icons (share and heart). To the left of the banner, a blue link says 'Back to Events'. The main content area starts with the event title 'Tech Conference 2025'. Below it, the date is listed as 'Thursday, November 20, 2025' and the time as '3:30 PM - 11:30 PM'. The location is 'Mumbai, Maharashtra'. A section titled 'About This Event' contains the text: 'A one-day event exploring the latest trends in AI, Web3, and Cloud Computing.' To the right of the event details is a 'Get Tickets' sidebar. It lists two ticket types: 'General Admission' at \$500 (100 left) and 'VIP Pass' at \$2000 (20 left). Each listing has a radio button for selection. Below these are dropdown menus for 'Quantity' (set to 1) and 'Total' (\$500). A large blue 'Book Now' button is at the bottom of the sidebar.

Backend Architecture:-

Content:

- **Framework:** Node.js with Express
- **Database:** MongoDB using Mongoose
- **Architecture:** MVC (Model-View-Controller)
- **Authentication:** JWT (JSON Web Token) or Sessions
- **API Communication:** REST APIs between frontend and backend

Database Models (Mongoose Schemas)

Content:

user.js

- Stores user details (name, email, password, role)
- Passwords are hashed before saving

- Handles user authentication

event.js

- Stores event title, date, location, and organizer info
- Linked with booking and review collections

booking.js

- Tracks which users booked which events
- Contains booking time

Event Hub Backend Weekly Breakdown:

Week 1: 15 Sept – 21 Sept

Goal: Backend Setup & Project Foundation

Tasks:

- Initialize Node.js project and configure Express.js
- Set up MongoDB connection using Mongoose
- Create project folder structure following MVC architecture
- Implement basic server setup and test routes (/api/test)
- Configure environment variables (.env for DB URI, JWT secret)
- Set up GitHub repository and version control

Week 2: 22 Sept – 28 Sept

Goal: Implement Core Database Models (Schemas)

Tasks:

- Design and implement **user.js** model
 - Fields: name, email, password (hashed), role
 - Add password hashing middleware using bcrypt
- Create **event.js** model
 - Fields: title, description, date, location, organizer
 - Establish relationship with bookings and reviews
- Build **booking.js** model
 - Fields: user ID, event ID, booking date/time
 - Ensure references to user and event collections

- Create initial sample data and test CRUD operations

Week 3: 29 Sept – 5 Oct

Goal: Core Functionalities (Authentication + CRUD APIs)

Tasks:

- Implement **JWT-based authentication** (login/signup)
 - Signup: register new users
 - Login: generate JWT tokens
 - Middleware: verifyToken for protected routes
- Build **RESTful APIs** for:
 - User management (GET, POST, PUT, DELETE)
 - Event management (create, update, delete, view)
- Integrate error handling and response formatting
- Postman API testing for all routes

Remaining Tasks & Future Work

Content:

- **Payment Integration:**
Implement a secure payment gateway (Razorpay) for event ticket purchases and confirmations.
- **Profile Update Module:**
Allow users to update personal information, profile pictures, and preferences.
- **Queueing System:**
Use a message queue (Redis) to manage simultaneous bookings and prevent double registration.
- **Enhanced Review System:**
Add features like edit/delete reviews and average event rating calculation.
- **Notification Module:**
Send real-time or scheduled notifications (in-app) for upcoming events, booking confirmations, and updates.

Backend Screenshots:

Register

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the center, a 'New Collection / EventHub / Register[Admin]' collection is selected. A POST request is being made to `http://localhost:5000/api/auth/register`. The 'Body' tab contains the following JSON payload:

```
1 {
2   "firstName": "Preksha",
3   "lastName": "Joshi",
4   "email": "joshipreksha2@gmail.com",
5   "password": "1124007j10",
6   "role": "admin",
7   "phone": "8320984867"
8 }
```

The response status is `201 Created` with a response time of 4.59 s and a size of 1.17 KB. The response body is:

```
1 {
2   "success": true,
3   "message": "Registration successful. Please verify your email with the OTP sent.",
4   "userId": "68ebe164913d0a788c24665c"
5 }
```

User Login

The screenshot shows the Postman application interface. On the left, the 'My Workspace' sidebar lists various collections and environments. In the center, a 'New Collection / EventHub / Login[User]' collection is selected. A POST request is being made to `http://localhost:5000/api/auth/login`. The 'Body' tab contains the following JSON payload:

```
1 {
2   "email": "itspreksha54@gmail.com",
3   "password": "1124007j10"
4 }
```

The response status is `200 OK` with a response time of 681 ms and a size of 1.62 KB. The response body is:

```
1 {
2   "success": true,
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.\n        eyJpZC16Ijy4Y23jMWE3YThMDc4MGMBOTIwYm3YiisImhdCI6MTc2MDI6MjA0NCwiZXhwIjoxNzYwOOk20DQ8fQ.\n        f77-Ee9vTxPsnkFE0g78oLb3ypBW4tXI46E1e06h_eI",
4   "user": [
5     {
6       "id": "68ebe164913d0a788c24665c",
7       "firstName": "Preksha",
8       "lastName": "Joshi",
9       "email": "itspreksha54@gmail.com",
10      "role": "attendee",
11      "isVerified": true
12    }
13  ]
14 }
```

Verify-otp

The screenshot shows the Postman interface with a collection named "EventHub". A POST request titled "Verify-otp" is selected. The request URL is `http://localhost:5000/api/auth/verify-otp`. The "Body" tab is active, showing raw JSON input:

```
1 {
2   "email": "joshiprehsha2@gmail.com",
3   "otp": "524953"
4 }
```

The response status is 200 OK, with a response body indicating success:

```
1 {
2   "success": true,
3   "message": "Email verified successfully"
4 }
```

Admin login

The screenshot shows the Postman interface with a collection named "EventHub". A POST request titled "Login[admin]" is selected. The request URL is `http://localhost:5000/api/auth/login`. The "Body" tab is active, showing raw JSON input:

```
1 {
2   "email": "joshiprehsha2@gmail.com",
3   "password": "116400Tj10"
4 }
```

The response status is 200 OK, with a response body containing session information and a user object:

```
1 {
2   "success": true,
3   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
4 eyJpZC16IjY4Y23jNjBYThkMDc4MGM0OTiwyM4ZSiisImhdCI6MTc2MDIBMTU5NiwiZXhwIjoxNzYwODk2Mzk2fQ.
5 RCS-CqcJSoKG_lfn6audjdjtFeCzQLI4iWegj0FymA",
6   "user": [
7     {
8       "id": "638cc60ea8d078c4920bc8e",
9       "firstName": "Preksha",
10      "lastName": "Joshi",
11      "email": "joshiprehsha2@gmail.com",
12      "role": "admin",
13      "isVerified": true
14    }
15  ]
16 }
```

Create Event

The screenshot shows the Postman application interface. On the left, the sidebar lists collections, environments, flows, and history. The 'EventHub' collection is selected, showing various API endpoints like 'POST Register[Admin]', 'POST Verify-otp', 'POST Login[admin]', 'GET Auth/Me', and 'POST CreateEvent'. The 'POST CreateEvent' endpoint is currently selected. The main workspace displays a POST request to 'http://localhost:5000/api/events/'. The 'Headers' tab is active, containing an 'Authorization' header with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ...'. The 'Body' tab shows a JSON response with the following content:

```
1: {
2:   "success": true,
3:   "event": {
4:     "title": "Tech Conference 2025",
5:     "description": "A one-day event exploring the latest trends in AI, Web3, and Cloud Computing.",
6:     "category": "conference",
7:     "tags": [],
8:     "organizer": {
9:       "_id": "68cbc60ea8d0780c4920bc8e",
10:      "firstName": "Preksha",
11:      "lastName": "Joshi"
12:    }
13:  }
14: }
```

The status bar at the bottom indicates a '201 Created' response with a duration of 482 ms and a size of 2.26 KB.

Create Booking

The screenshot shows the Postman application interface, similar to the previous one but with a different collection. The sidebar lists collections, environments, flows, and history. The 'EventHub' collection is selected, showing various API endpoints. The 'POST CreateBooking[attendee]' endpoint is currently selected. The main workspace displays a POST request to 'http://localhost:5000/api/bookings'. The 'Headers' tab is active, containing an 'Authorization' header with the value 'Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ...'. The 'Body' tab shows a JSON response with the following content:

```
1: {
2:   "success": true,
3:   "booking": {
4:     "bookingNumber": "BKMG08IAZV4562",
5:     "user": {
6:       "_id": "68cbc1a7a8d0780c4920bc7b",
7:       "firstName": "Preksha",
8:       "lastName": "Joshi",
9:       "email": "itspreksha54@gmail.com"
10:    },
11:    "event": {
12:      "venue": {
13:        ...
14:      }
15:    }
16:  }
17: }
```

The status bar at the bottom indicates a '201 Created' response with a duration of 436 ms and a size of 2.43 KB.