Figure: LET'S ORGANIZE !!

Event Organizer API

Introduction

- Welcome to the Event Organizer API presentation.
- The Event Organizer API simplifies the process of managing entertainment events through a secure and efficient backend system.

Motivation

- Current challenges in organizing events.
- The need for a centralized system to manage events.
- Advantages of using the Event Organizer API.

Key Features

- User signup and login functionality.
- CRUD operations for managing events (Create, Read, Update, Delete).
- Token-based authentication for secure user access.
- Use of bcrypt for password hashing.

Technology Stack

- Node.js: Backend JavaScript runtime.

- MongoDB: NoSQL database for storing user and event data.

- Postman: API development and testing tool.

- Visual Studio Code (VSCode): Integrated development environment.

Advantages

- Efficient event management.
- Secure user authentication.
- Scalability with MongoDB.
- Ease of API testing with Postman.

Disadvantages

- Learning curve for beginners with Node.js and Express.js.
- Dependency on external services (MongoDB, Postman).

User-related Code

- User model definition.
- User signup logic.
- User login logic.

Event-related Code

- Event model definition.
- Event creation logic.
- CRUD operations for events.

API Endpoints and Routes

- Detailed breakdown of API endpoints:
  - POST /auth/signup: User registration.
  - POST /auth/login: User login.
  - POST /event/create: Create a new event.
  - GET /event/display: Retrieve all events.
  - GET /event/display/:eventId: Retrieve a specific event.
  - PUT /event/update/:eventId: Update an event.
  - DELETE /event/delete/:eventId: Delete an event.
- Importance of route organization and structure.

Middleware for Authentication

- Overview of the `checkAuth` middleware.
- How JWT (JSON Web Token) is used for user authentication.
- Ensuring secure routes by implementing authentication middleware.

MongoDB Connection and Environment Variables

- How to connect to MongoDB using the `mongoose` library.
- The use of environment variables for sensitive data storage.
- Importance of securing database connection details.

Running and Testing the API

- Instructions for running the Express application.
- Screenshots of Postman requests to various endpoints.(see the video presntation please)
- Screenshots of MongoDB Compass showing event data.((see the video presntation please)

Conclusion

- In conclusion, the Event Organizer API provides a robust solution for efficiently managing entertainment events through a secure and user-friendly backend system. With features such as user authentication, event creation, and CRUD operations, the API addresses the challenges associated with event organization. The technology stack, including Node.js, Express.js, MongoDB, and Postman, contributes to the API's scalability, security, and ease of testing. The use of JSON Web Tokens (JWT) for authentication and bcrypt for password hashing ensures a secure environment for users. The code structure, explained in detail, provides a foundation for developers to understand and extend the functionality of the API. The inclusion of middleware for authentication and the use of environment variables for sensitive data showcase best practices in software development.

Future Steps

- Event Categories and Filters:
  Introduce a categorization system for events and allow users to filter events based on categories, date ranges, or locations.
- Notification System:
  Implement a notification system to keep users informed about upcoming events, changes in event details, or relevant announcements.
- Analytics and Reporting:
  Add analytics features to provide event organizers with insights into user engagement, attendance trends, and event popularity.