



Block Intelligence

1st Floor, 103, Christine Heights, Kanakia Mira Rd, opp. M.G.L CNG Pump, Unique Gardens,
Beverly Park, Mira Road East, Thane, Mira Bhayandar, Maharashtra 401107

connect@blockintelligence.io

+91 73042 83042

Senior Python Developer Test

Problem Statement: Multi-User Event Management API

You are tasked with creating an Event Management System API that will allow multiple users to create, manage, and participate in events. The system should provide different user roles: **admin**, **event organizer**, and **attendee**. Each user role should have different access rights, and the API must be secure, scalable, and optimized for performance.

Requirements:

1. User Management:

- Allow users to register and log in using secure authentication (JWT or OAuth2).
- Admins can create, delete, and manage users.
- Event organizers can create and manage events.
- Attendees can view and join events.

2. Event Management:

- Event organizers should be able to create, update, and delete events.
- Events should have attributes such as title, description, location, date, time, and maximum attendees.
- Users should be able to view event details and sign up for events.
- Admins can oversee all events and modify or delete any event.

3. API Features:

- CRUD operations for users and events.
- Event listing and search functionality (e.g., filter events by date, location, or availability).
- Role-based access control for different endpoints.
- Proper error handling and validation of inputs.

4. Security:

- Implement user authentication with JWT or OAuth2.
- Ensure that the API is secure against common vulnerabilities (e.g., SQL injection, XSS, CSRF).



Block Intelligence

1st Floor, 103, Christine Heights, Kanakia Mira Rd, opp. M.G.L CNG Pump, Unique Gardens,
Beverly Park, Mira Road East, Thane, Mira Bhayandar, Maharashtra 401107

connect@blockintelligence.io

+91 73042 83042

- Include authorization to restrict access to endpoints based on user roles.
- Rate limiting for API calls to prevent abuse.

5. Testing & Documentation:

- Write unit tests for key parts of the system.
- Provide API documentation (e.g., using Swagger or Redoc).

Expectations:

- **Directory Structure:** Organize the project in a way that makes it easy to navigate and debug.
- **Code Modularity:** Make use of blueprints/modules in FastAPI or Django to separate functionality (user management, event management, etc.).
- **Performance Optimization:** Consider database indexing, caching, or pagination for large event listings.
- **Use Libraries:** Leverage libraries for authentication, validation, and database interaction.
- **Database Design:** Use any relational database (e.g., PostgreSQL, MySQL) and ensure proper relationships between users and events (e.g., many-to-many for attendees and events).

Bonus Points:

- Implementing a notification system that sends emails to users when they join or create an event.
- Integrating third-party APIs (e.g., Google Maps for event locations).

Duration & Submission:

- 1 Week from the day problem statement is released to candidate
- Submission must be done via Github in a public repository, and relevant links must be shared to HR via mail.

Best of luck, we can't wait to evaluate your result and take you onboard.

Thanks and Regards

Block Intelligence Team

connect@blockchain.ai.in

blockintelligence.io

LLPIN/CIN/Form INC-1 Ref No : [U80220MH2021PTC359578](#)

GSTIN : 27AAOCM5515B1ZK