

# Node.js Developer - Machine Task

## Event Booking API

### A.Task Objective

Develop an Event Booking API with custom rules:

- Roles: **ORGANIZER | CUSTOMER**
- Event workflow: **DRAFT → PUBLISHED → CANCELLED**
- Booking: atomic seat update with rollback if overbooked
- Organizer analytics: revenue + top-selling event
- Custom IDs: **EVT-XXXX, BK-XXXX**
- Real-time notifications using Socket.io: organizer gets notified immediately when a new booking is made

### B. Requirements

#### 1. Authentication

- **POST /register** → { name, email, password, role }
- **POST /login** → JWT
- Protect routes & enforce role

#### 2. Event Management

- **POST /events** → create DRAFT event
- **PATCH /events/:id** → update event (only owner)
- **POST /events/:id/publish** → mark PUBLISHED
- **POST /events/:id/cancel** → mark CANCELLED
- **GET /events/me** → organizer's events + tickets sold + revenue
- **GET /events** → list published events, support filter by tags, city, date, pagination & sorting
- Optional: cache event details (Redis)

#### 3. Booking System

- **POST /bookings** → { eventId, qty }
  - Validate seats available
  - Atomic update: decrement seats or rollback
  - Booking ID custom: **BK-XXXX**
  - **GET /bookings/me** → list own bookings

#### 4. Analytics

**GET /analytics/me** → total events, tickets sold, total revenue, top-selling event

### C. Deliverables

- Full working API with auth, event CRUD, booking logic
- Atomic seat update + rollback
- **Socket.io real-time notifications**
- Minimal tests & documentation
- Optional: caching

### D. Do's: -

- Read instructions carefully
- Write clean and readable code, **Test your code.**
- Follow the specified tech stack.
- Use browser in case of need.
- **Use MongoDB as Database.**
- Submit task in given time.

### E. Don't:-

- **Do not use any AI tool.**
- **Do not use any code from previous or other project or anywhere.**
- **Do not use AI-based solutions for code generation or task completion.**

### F. Result

- Gracefully handle error and loading cases. . .
- Use Proper architecture and good code structure.
- **After code completion, create a public git repository on your own GitHub account, and push all the code and share the GitHub URL.**

### G. Task Duration : 2 hour 30 Minute

“ All the best ”