ISO 9001:2015
SOCOTEC Certificate No. SCP000722Q

REPUBLIC OF THE PHILIPPINES
BICOL UNIVERSITY
POLANGUI
Email: bu-cpro@bicol-u.edu.ph | bupc.bu@gmail.com
jULY 13, 2024

Professor: Dr. Guillermo V. Red, Jr.

Year/Sec: BSIS – 3A
Course: IS ELEC 3

Week 11 - Assignment

Backend Setup Summary

University Event Management System (UEMS)

Group: Sigma Coders

Date: 11/25/2025

NODE AND EXPRESS SETUP DESCRIPTION

Project Initialization:

 # Initialize Node.js project

npm init -y

 # Install production dependencies

npm install express mongoose cors dotenv bcryptjs jsonwebtoken multer helmet express-rate-limit validator uuid

 # Install development dependencies

npm install -D nodemon jest

 # Create project structure

mkdir models controllers routes middleware config utils uploads

Server Configuration:

The Express.js server is configured with:

- Security middleware (Helmet, CORS, Rate Limiting)
- JSON body parsing with 10MB limit
- Environment variables via dotenv
- MongoDB connection with Mongoose
- ODM Modular route handling with separate route files
- Global error handling and 404 routes
- Static file serving for uploaded event media

Development Scripts:

```
{
  "scripts": {
    "start": "node server.js",
    "dev": "nodemon server.js",
    "test": "jest"
  }}
```

Key Dependencies:

- **express:** Web server framework
- **mongoose:** MongoDB object modeling
- **bcryptjs:** Password hashing
- **jsonwebtoken:** Authentication tokens
- **multer:** File upload handling
- **cors:** Cross-origin resource sharing
- **helmet:** Security headers

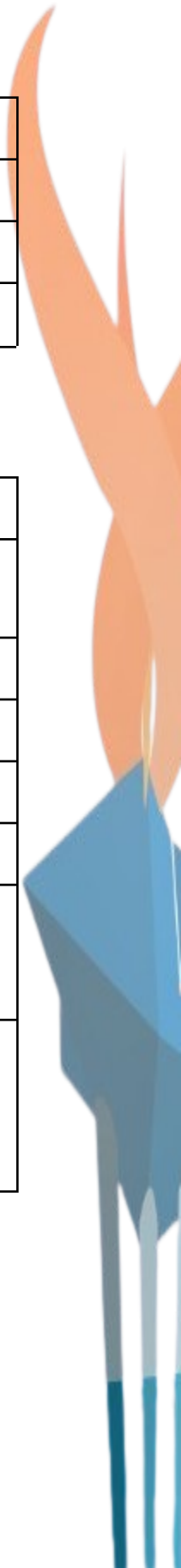
DEFINED API ENDPOINTS

Authentication Routes

Method	Route	Description	Access
POST	/api/auth/register	student registration	Public
POST	/api/auth/login	user login	Public
GET	/api/auth/me	get current user	Authenticated

Event Routes

Method	Route	Description	Access
GET	/api/events	get all events (with filters)	Public
GET	/api/events/:id	get event details	Public
POST	/api/events	create new event	Organizer+
PUT	/api/events/:id	update event	Owner/Admin
DELETE	/api/events/:id	delete event	Owner/Admin
GET	/api/events/:id/analytics	get detailed analytic for specific event	Owner
GET	/api/users/my-events/analytics	get analytics overview for all events owned by the current user	Owner



Event Proposal Routes

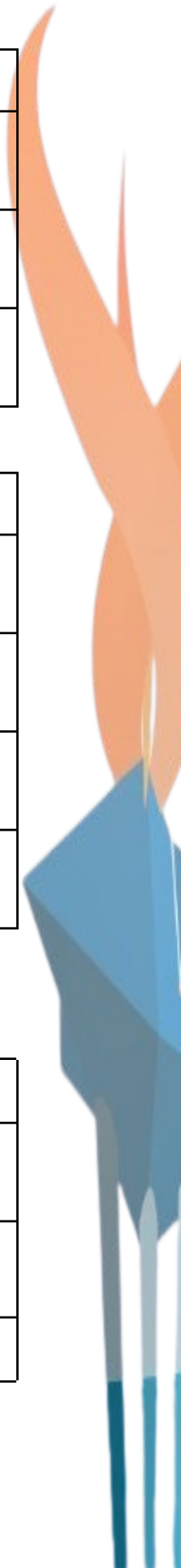
Method	Route	Description	Access
POST	/api/events/proposals	create event proposal	Student+
POST	/api/events/proposals/:id/submit	submit proposal	Owner
PUT	/api/events/proposals/:id	update draft proposal	Owner

Registration and Attendance Routes

Method	Route	Description	Access
POST	/api/events/:id/register	register for event	Student+
DELETE	/api/events/:id/register	unregister from events	Student+
GET	/api/users/registrations	get user's registration	Owner
POST	/api/events/:id/attend	mark attendance	Organizer+

Admin Routes

Method	Route	Description	Access
GET	/api/admin/pending-events	get pending approvals	Admin
PUT	/api/admin/events/:id/approve	approve event	Admin
PUT	/api/admin/events/	reject event with	Admin



	:id/reject	notes	
GET	/api/admin/users	get all users	Admin
GET	/api/admin/statistics	platform analytics	Admin

Notification and Sharing Routes

Method	**Route**	**Description**	**Access**
GET	/api/notifications	get user notifications	Owner
GET	/api/share/events/:id	generate sharable link	Public

BACKEND ARCHITECTURE

MVC Pattern Implementation:

uems-backend/

- **models/ (Mongoose Schemas)**
  - User.js
  - Event.js
  - Registration.js
  - Attendance.js
  - Notification.js
- **controllers/ (Business Logic)**
  - authController.js

```
    ├── eventController.js
    ├── eventProposalController.js
    ├── adminController.js
    └── notificationController.js
├── routes/ (API Endpoints)
    ├── authRoutes.js
    ├── eventRoutes.js
    ├── eventProposalRoutes.js
    ├── adminRoutes.js
    └── notificationRoutes.js
├── middleware/ (Custom Middleware)
    ├── auth.js
    ├── adminAuth.js
    ├── upload.js
    └── eventOwnership.js
└── config/ (Configuration)
    ├── database.js
```

Database Schema Design:

- **User Model:** Student accounts with role-based permissions
- **Event Model:** Event details with approval workflow
- **Registration Model:** Event registration tracking
- **Attendance Model:** Participation records
- **Notification Model:** User notifications system

Key Features Implemented:

- Event Proposal System with admin approval workflow
- Role-Based Access Control (Student, Organizer, Admin)
- File Upload System for event media
- JWT Authentication with secure tokens
- Social Sharing with unique event links
- Notification System for user updates
- Search & Filtering by categories and tags

SECURITY MEASURE

Authentication and Authorization:

- JWT-based authentication with 30-day expiry
- Password hashing using bcryptjs (12 salt rounds)
- Role-based middleware for route protection
- Rate limiting (100 requests per 15 minutes per IP)

Data Protection:

- Environment variables for sensitive configuration
- Input validation and sanitization
- File type and size restrictions for uploads
- CORS configuration for frontend integration

Database Security:

- MongoDB Atlas with secure connection string
- Mongoose schema validation
- Proper indexing for performance
- Error handling for database operations

DEVELOPMENT AND DEPLOYMENT

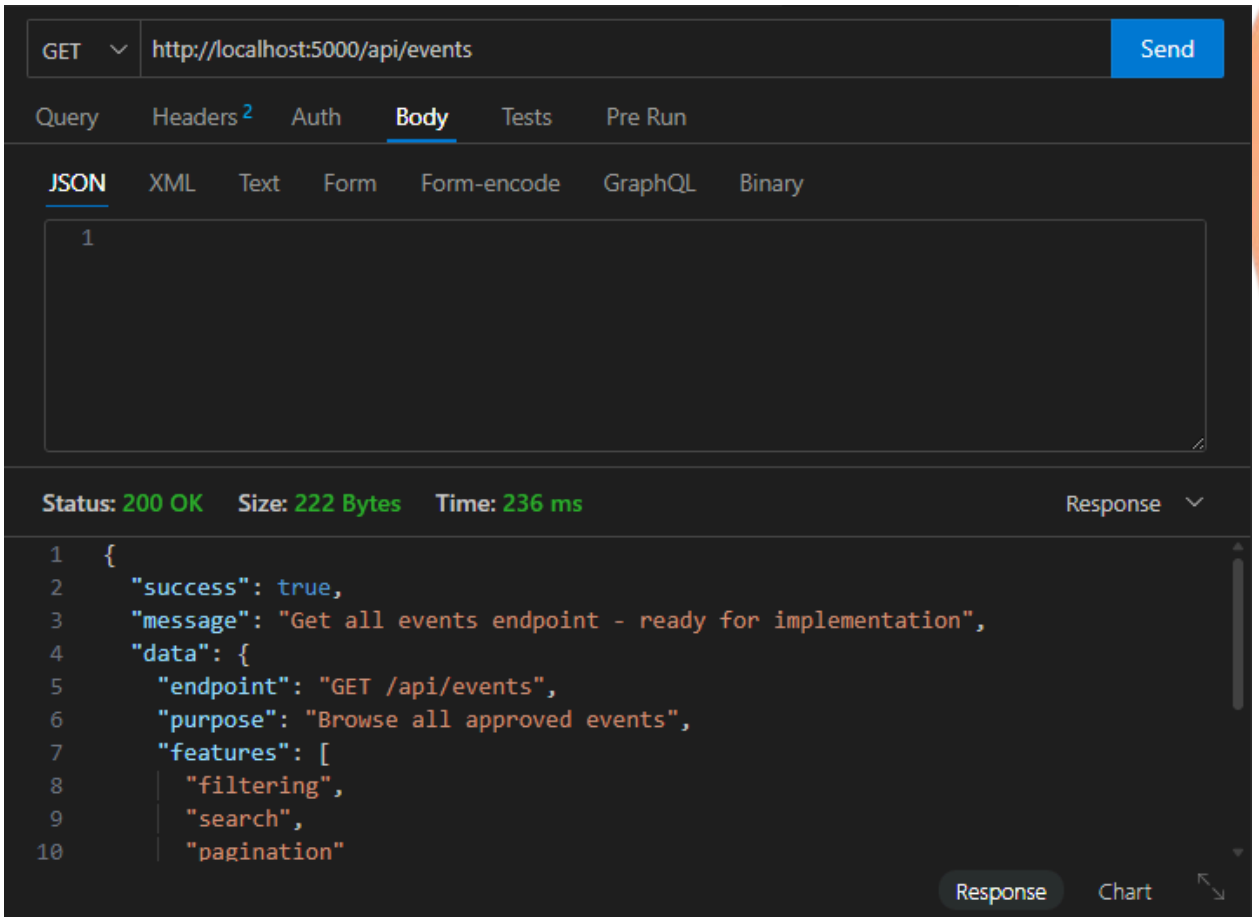
Environment Configuration:

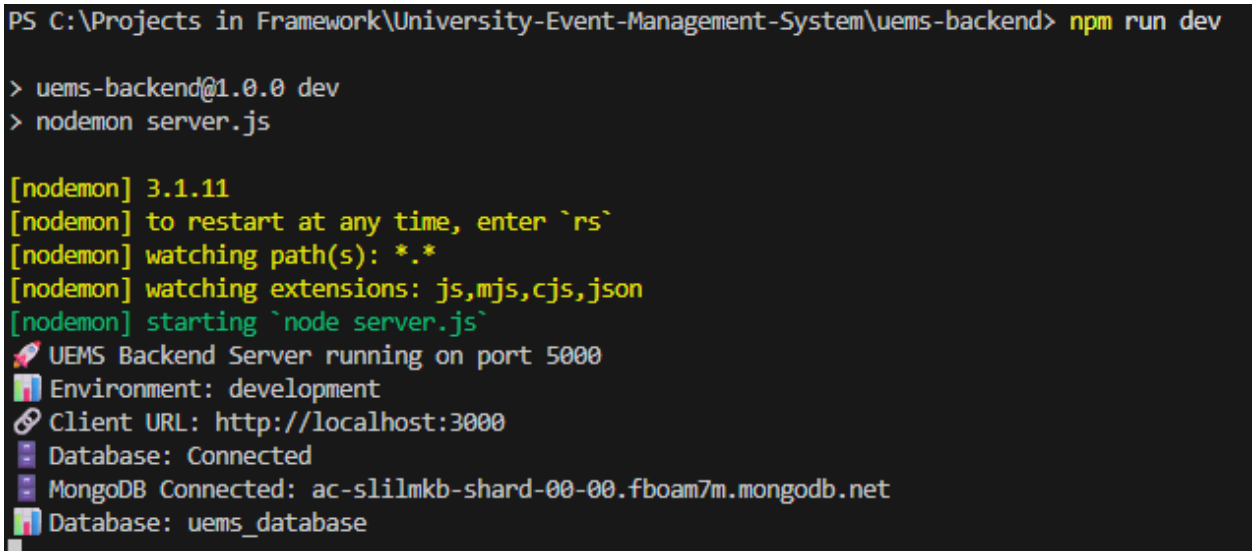
- **MONGODB_URI:** MongoDB connection string
- **JWT_SECRET:** JWT token secret key
- **CLIENT_URL:** Frontend application
- **URL UPLOAD_PATH:** File upload directory

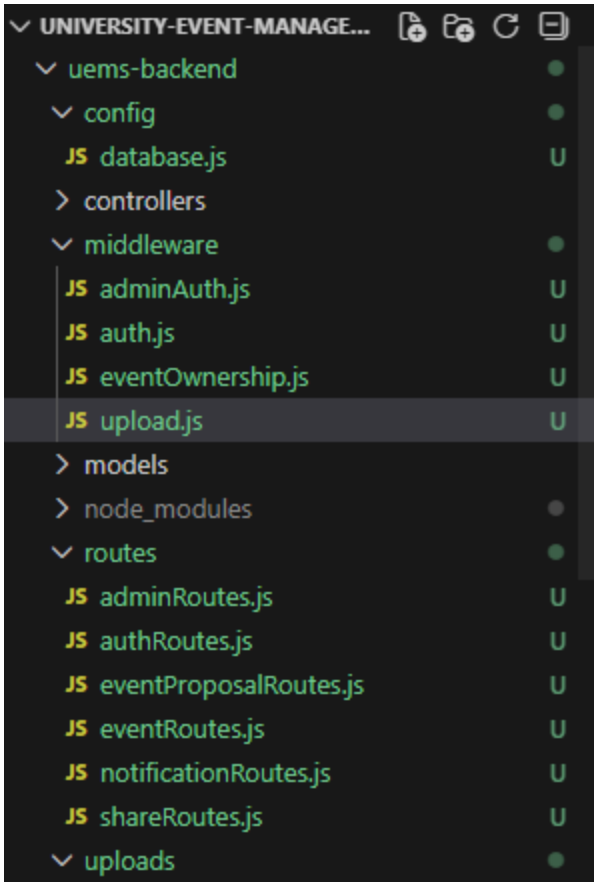
Production Readiness

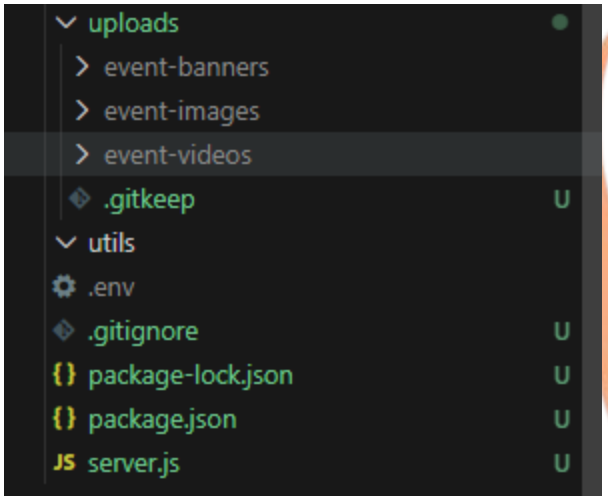
- Environment-based configuration
- Error handling and logging
- Performance optimization
- Security headers and
- CORS Rate limiting and input validation

TESTING EVIDENCE:






UNIVERSITY-EVENT-MANAGE...
uems-backend
config
database.js U
controllers
middleware
adminAuth.js U
auth.js U
eventOwnership.js U
upload.js U
models
node_modules
routes
adminRoutes.js U
authRoutes.js U
eventProposalRoutes.js U
eventRoutes.js U
notificationRoutes.js U
shareRoutes.js U
uploads


uploads
event-banners
event-images
event-videos
.gitkeep U
utils
.env
.gitignore U
package-lock.json U
package.json U
server.js U