

SIT725 - Project Proposal (3.3P)

Student Information

- **Student ID:** s217551712
 - **Name:** Antheis Maxenn
 - **Task:** 3.3P - Project Proposal
-

Project Title

EventSpot - Local Event Discovery Platform

Problem Statement

As a student, I've noticed how hard it is to find out what's happening around campus and the local area. Event info is all over the place - Facebook, Instagram, Discord, random websites - and it's easy to miss out on cool stuff. Plus, when student clubs or friends want to organize events, they don't have a good way to reach people who'd actually be interested. I want to build a simple platform where everyone can see what's on and easily create their own events.

Proposed Solution

EventSpot will be a web app where people can find and share local events. Whether it's a concert, study group, sports game, or just a hangout, users can post events and see what others are doing. I'll add a map feature so you can find things near you, an RSVP system to track who's coming, and real-time notifications when event details change.

Key Features

1. **User Authentication** - Sign up, log in, and manage your profile
 2. **Event Creation** - Post events with photos, location, date/time, and description
 3. **Browse & Search** - Find events by keyword, location, category, or date
 4. **Map View** - See events on an interactive Google Map and find what's nearby
 5. **RSVP System** - Click "Going" or "Interested" to track your events
 6. **Reviews** - Leave ratings and comments after attending events
 7. **Live Updates** - Get notified in real-time when event details change (using Socket.io)
-

Technology Stack

Frontend

- HTML5, CSS3, and JavaScript for the web pages
- Bootstrap 5 to make it look good and work on mobile

- Google Maps API to show event locations

Backend

- Node.js and Express.js for the server
- Socket.io for real-time notifications
- JWT for secure login tokens
- Multer to handle image uploads

Database

- MongoDB to store users, events, and reviews
- Mongoose to make database queries easier
- Geospatial indexing so I can search "events near me"

External Services

- Cloudinary to store and serve event images
- Google Maps & Places APIs for location features

Database Schema

Collections

users

```
{  
  username, email, password (hashed),  
  firstName, lastName, avatar,  
  location: { coordinates: [lng, lat] },  
  createdEvents: [eventId],  
  attendingEvents: [eventId]  
}
```

events

```
{  
  title, description, category,  
  startDate, endDate,  
  location: { venue, address, coordinates: [lng, lat] },  
  organizer: userId,  
  attendees: [{ user, status }],  
  images: [{ url, publicId }],  
  capacity, price, isFree,  
  averageRating, totalRatings  
}
```

reviews

```
{  
  event: eventId,  
  user: userId,  
  rating (1-5),  
  comment,  
  createdAt  
}
```

Core API Endpoints

Authentication

- `POST /api/auth/register` - Register new user
- `POST /api/auth/login` - User login
- `GET /api/auth/verify` - Verify JWT token

Events

- `GET /api/events` - Get all events (with filters)
- `GET /api/events/nearby` - Find events by coordinates & radius
- `GET /api/events/:id` - Get event details
- `POST /api/events` - Create new event
- `PUT /api/events/:id` - Update event
- `DELETE /api/events/:id` - Delete event

RSVP

- `POST /api/events/:id/rsvp` - RSVP to event
- `GET /api/events/:id/attendees` - Get attendee list

Reviews

- `GET /api/events/:id/reviews` - Get event reviews
- `POST /api/events/:id/reviews` - Add review

Users

- `GET /api/users/:id` - Get user profile
- `PUT /api/users/:id` - Update profile

Development Plan

Phase 1 (Weeks 1-4) - Core Features

- Project setup, database design, and authentication

- Event CRUD operations
- Basic frontend with event listing and details

Phase 2 (Weeks 5-8) - Advanced Features

- Google Maps integration with location search
- RSVP system and attendee management
- Reviews and ratings
- Image upload with Cloudinary

Phase 3 (Weeks 9-12) - Finishing Up

- Add Socket.io for live notifications
 - Make sure everything works on mobile
 - Fix bugs and test everything
 - Deploy the app so people can actually use it
-

Expected Outcomes

- A working web app where people can find and create local events
- Search for events near you using your location
- Get real-time updates when events change
- Works well on phones and computers
- Secure login system to protect user accounts