

Guide Approval (initials/date): \_\_\_\_\_

## CAP4001 - Capstone Project Proposal Report

(Individual Report)

<b>Student Name</b>	Mayakuntla Lokesh
<b>Student Register Number</b>	22BCE9911
<b>Programme</b>	Bachelor of Technology
<b>Semester/Year</b>	Fall sem (2025-26)
<b>Guide(s)</b>	Saroj Kumar Panigrahy
<b>Project Title</b>	A Real-Time Player Finding System

### Team Composition

Provide the information below for each member of the project team. Include all project team members, not just those in your discipline or those enrolled for Capstone project. Please also include yourself!

<b>Reg. No</b>	<b>Name</b>	<b>Major</b>	<b>Specialization</b>
22BCE9357	Adnan Hasshad Md	CSE	Core
22BCE20420	Tatikonda Srilekha	CSE	Core
22BCE9911	Mayakuntla Lokesh	CSE	Core
22BCE9745	Thokala Sravan	CSE	Core

## **Project and Task Description**

---

Provide a brief (one or two page) technical description of the design project and your specific tasks, as outlined below: (use a separate sheet)

### **Provide a summary of the project**

The project, "A Real-Time Player Finding System," is an MVP (Minimum Viable Product) social platform designed to help users find and connect with others for specific gaming activities. The app's purpose is to address the common problem of finding compatible teammates or opponents for online games. The MVP focuses on core functionality and a good UI, including user profiles, a public request board, a 1v1 and team finder. The approach is to build a modern, serverless application that prioritizes immediate, action-oriented requests.

### **Describe the specific role and tasks that you individually will be completing**

- (1) Configure Express.js server with TypeScript, establish backend structure, setup environments, configure middleware.
- (2) Design database schema with Drizzle ORM, create tables for users, profiles, requests, notifications, implement indexing.
- (3) Develop REST API endpoints for authentication, discovery, search, requests, implement Zod validation, build authorization logic.
- (4) Implement player finding and search with filtering, build user management, develop request board logic, create WebSocket handlers.
- (5) Integrate third-party services, implement WebSocket support, conduct API testing, optimize database queries.

### **Discuss in detail the specific approach that will be used to complete your portion of the design**

**Phase 1 (Week 1-2):** Backend setup, database schema design, architecture planning, environment configuration.

**Phase 2 (Week 3-4):** API endpoint development, database implementation, authentication setup, indexing strategy.

**Phase 3 (Week 5-6):** Business logic implementation (discovery, search), WebSocket integration, service integration, notifications.

**Phase 4 (Week 7-8):** Performance optimization, API testing, query optimization, documentation, refinements.

### **Describe the phases of the design process**

**Phase 1 (Week 1-2):** Planning, requirements analysis, architecture design, technology selection, database schema planning.

**Phase 2 (Week 3-4):** Core infrastructure development, API implementation, authentication setup.

**Phase 3 (Week 5-6):** Real-time features, request board logic, system integration, performance optimization.

**Phase 4 (Week 7-8):** Testing, documentation, deployment preparation, quality assurance.

## Outcome Matrix

Describe your plan to demonstrate each of the outcomes below.

<b>Outcomes</b>	<b>Plan for demonstrating outcome</b>
a) an ability to apply knowledge of mathematics, science, and engineering	Will apply software engineering principles, data structures for algorithms, relational database theory, and distributed systems patterns.
c) an ability to design a system, component, or process to meet desired needs within realistic constraints	Will design system architecture balancing feature completeness, performance, scalability, 8-week timeline, and free/low-cost cloud platform constraints.
d) an ability to function on multidisciplinary teams	Will collaborate with team members, facilitate communication, coordinate efforts across frontend, backend, and management roles.
e) an ability to identify, formulate, and solve engineering problems	Will identify bottlenecks, formulate solutions for challenges, solve scalability issues, and troubleshoot integration problems.
g) an ability to communicate effectively	Will create technical documentation, communicate decisions clearly, provide guidance, and maintain code documentation.
k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice	Will utilize React, Express.js, TypeScript, PostgreSQL, Drizzle ORM, WebSocket API, OAuth 2.0, Cloudflare R2, and Git.

## Realistic Constraints

---

**Development Timeline:** 8-week cycle with MVP prioritization. **Team Resources:** 4-member team requiring coordination. **Infrastructure:** Free/low-cost cloud platform (Replit, Neon PostgreSQL). **Technical Complexity:** Real-time request board, WebSocket connectivity, service integration. **Performance:** Responsive design across devices with acceptable load times. **Scalability:** MVP design with extensible architecture for future growth.

## Engineering Standards

---

**Code Standards:** TypeScript strict mode, ESLint configuration, consistent naming. **Database:** Normalized design (3NF), proper indexing, referential integrity. **API:** RESTful principles, HTTP status codes, Zod validation, documentation. **Security:** Input validation, SQL injection prevention, OAuth 2.0, CORS headers. **Testing:** Unit, integration, end-to-end tests. **Documentation:** API docs, architecture diagrams, code comments. **Version Control:** Meaningful commits, branch management, code reviews.