# Developer-Focused Project Plan for Academic Connect MVP (Extended Timeline)

## 1. Project Title:

Academic Connect MVP Development

## 2. Project Duration:

1.5 months (6 weeks)

## 3. Primary Goals:

- Develop a streamlined, functional MVP for Academic Connect, focusing on essential features.  
- Complete development, testing, and deployment in 1.5 months.  
- Ensure high-quality code, secure authentication, and efficient data handling.

## 4. Expected Features of the MVP (Technical Specifications)

### User Profiles

* Functionality: Researchers can create and manage their profiles.
* Technical Requirements: Secure authentication, user profile fields, profile privacy using Firebase Authentication and Firebase Firestore for data storage.
* Key Tasks:  
   - User sign-up and login forms.  
   - Profile creation and updates (with data validation).

### Search and Match System

* Functionality: Users can search for other researchers based on interests, institution, etc.
* Technical Requirements: Firestore indexing and Firebase functions for keyword-based searches, filtering, and fuzzy search.
* Key Tasks:  
   - Implement search and filter functionality on backend and frontend.

### Messaging System (Basic)

* Functionality: Allows direct messaging between users.
* Technical Requirements: WebSockets (e.g., Socket.IO) for real-time messaging and secure storage.
* Key Tasks:  
   - Create backend endpoints for sending and receiving messages.  
   - Build frontend messaging interface.

### Forum and Discussion Board

* Functionality: Enables users to post topics and comment on them.
* Technical Requirements: Firestore for posts, comments, and moderation functionality, with pagination.
* Key Tasks:  
   - Backend API for creating and managing posts.  
   - Implement front-end UI for forum browsing and interactions.

### Security and Privacy Features

* Functionality: Ensures secure user data and protects communication.
* Technical Requirements: JWT or Firebase Authentication for access control and data encryption.
* Key Tasks:  
   - Implement token-based authentication.  
   - Ensure encryption of messages and sensitive data.

## 5. Development Phases and Milestones

### Phase 1: Planning and Initial Setup (Week 1)

* - Set up project architecture (Express.js backend, React/Vue frontend).
* - Configure Firebase database and create data schema for profiles, messages, and posts.
* - Create repository with version control (Git), and integrate CI/CD.
* - Identify necessary libraries (e.g., Socket.IO for messaging, Firebase for storage).
* Milestone: Project setup complete, repository created, and initial configurations finished.

### Phase 2: User Profiles and Authentication (Weeks 2-3)

* - Develop user registration, login, and profile management using Firebase Authentication.
* - Set up secure data handling and validation in Firebase Firestore.
* - Create frontend forms for user registration, profile editing, and viewing.
* Milestone: User authentication and profile management fully functional.

### Phase 3: Search and Match System (Week 3)

* - Build backend logic for user search using Firestore indexing.
* - Implement search filters (interests, location, institution).
* - Integrate frontend with search results and filters.
* Milestone: Search system operational with filtering options.

### Phase 4: Messaging System (Week 4)

* - Develop backend logic for sending and receiving messages in real-time via Socket.IO.
* - Create a basic frontend chat interface.
* - Implement message encryption and validation checks.
* Milestone: Messaging system functional and real-time.

### Phase 5: Forum and Discussion Board (Week 5)

* - Backend API for forum post creation, commenting, and viewing using Firebase.
* - Frontend for displaying posts, allowing commenting, and moderating content.
* - Pagination and sanitization of user input.
* Milestone: Forum and discussion board operational.

### Phase 6: Security, Testing, and Optimization (Week 6)

* - Implement security features (JWT/Firebase Authentication, encryption).
* - Write unit and integration tests for core features (profile management, search, messaging).
* - Optimize database queries and refactor code.
* - Conduct load testing on search and messaging features.
* Milestone: Security protocols in place, performance optimized, and tests completed.

### Buffer Time (End of Week 6)

* - Address any final issues or bugs discovered during testing.
* - Prepare documentation (API docs, setup guides).
* - Deploy MVP to staging and conduct final checks.
* Milestone: Deployment ready for production.

## 6. Technical Stack

Backend: Express.js for REST API  
Frontend: React.js or Vue.js for UI  
Database: Firebase Firestore for data storage and Firebase Authentication for secure access  
Real-Time Messaging: Socket.IO for WebSocket connections  
Search and Indexing: Firestore indexing and Firebase functions  
Security: JWT or Firebase Authentication, bcrypt for password hashing, encryption for messages  
Version Control: GitHub/GitLab with CI/CD integration

## 7. Testing and Quality Assurance

- Unit Tests: Core functions (profile management, messaging, search).  
- Integration Tests: User flow between frontend and backend for seamless data communication.  
- Security Audits: Ensure authentication and sensitive data handling are secure.  
- Load Testing: Test search and messaging under simulated user load.  
- Manual Testing: Test user interactions for usability and functionality.

## 8. Communication Protocols

- Daily Standups: Short 15-minute meetings to check progress.  
- Weekly Sprint Review: Progress review and potential adjustments.  
- Git Repository: All code will be stored on GitHub/GitLab with proper documentation.  
- Documentation: Comprehensive API documentation and setup guides for the project.

## 9. Final Deliverable

Complete MVP for Academic Connect, ready for deployment and initial user testing. All core features will be functional, and the application will be secure, optimized, and tested.

## 10. Summary Timeline

|  |  |  |
| --- | --- | --- |
| Phase | Duration (Weeks) | Milestone |
| Phase 1: Planning & Setup | 1 | Project setup, Git repository, configurations |
| Phase 2: User Profiles | 2 | User authentication and profile management |
| Phase 3: Search System | 1 | Search and filtering functional |
| Phase 4: Messaging System | 1 | Messaging system with real-time capability |
| Phase 5: Forum & Discussion | 1 | Forum and discussion board operational |
| Phase 6: Security & Optimization | 1 | Security, testing, and performance optimization |
| Buffer Time | 0.5 | Final bug fixes, documentation, and deployment |