**Project Summary for University Clubs and Societies App**

**Objective**: Create a Flutter application to facilitate collaboration among university clubs and societies, with features like individual club portals, recruitment tools, a general event calendar, and both public and private announcements. The backend will be built using Node.js and MongoDB. The project timeline is one month.

**Features**

1. **Club Initialization**:
   * Set leads, sub-leads, members, and club information.
2. **Recruitment**:
   * Tools and forms for recruiting new members.
3. **Calendar**:
   * Schedule and view events for all clubs.
4. **Announcements**:
   * Public announcements visible to all users.
   * Private announcements specific to individual clubs.
5. **Member and Leaders Profile**:
   * Profiles for members and leaders with relevant information.

**Roadmap**

**Week 1: Planning and Setup**

1. **Define Requirements**:
   * Finalize features and user roles.
2. **Design**:
   * Create wireframes and UI mockups.
   * Plan MongoDB database schema.
3. **Setup Development Environment**:
   * Initialize Flutter project.
   * Set up Node.js and MongoDB.
   * Configure necessary libraries.

**Week 2: Core Functionality**

1. **Authentication**:
   * Implement user registration, login, and role-based access control.
2. **Club Portals**:
   * Develop structure for individual club portals.
   * Implement club initialization feature.
3. **Backend APIs**:
   * Develop APIs for user, club, and event management.

**Week 3: Additional Features**

1. **Recruitment**:
   * Develop recruitment tools and forms.
2. **Event Calendar**:
   * Integrate calendar view and event management.
3. **Announcements**:
   * Implement public and private announcements.
   * Set up notifications.
4. **Profiles**:
   * Develop member and leader profile features.
5. **UI Enhancements**:
   * Refine UI based on feedback.
   * Ensure responsiveness and accessibility.

**Week 4: Testing and Deployment**

1. **Testing**:
   * Conduct unit and integration testing.
   * Perform user acceptance testing (UAT).
2. **Bug Fixes and Refinements**:
   * Address issues and optimize performance.
3. **Deployment**:
   * Deploy backend to a server or cloud platform.
   * Publish Flutter app to app stores.
4. **Documentation and Training**:
   * Create user manuals and technical documentation.
   * Conduct training sessions if needed.

**Tools and Libraries**

* **Flutter Packages**: provider or bloc, http or dio, flutter\_calendar\_carousel, firebase\_auth (optional)
* **Node.js Libraries**: express, mongoose, jsonwebtoken
* **Testing**: jest or mocha (backend), flutter\_test (frontend)

**Next Steps**

1. Gather detailed requirements and design the UI/UX.
2. Set up the development environment and initialize the project.