Church Management System - Semester Project Roadmap

📋 **Phase 1: Project Setup & Foundation (Week 1-2)**

Week 1: Environment & Planning

1. **Team Setup & Tools**

   o   Create GitHub organization/repository

   o   Set up project board (GitHub Projects)

   o   Establish communication channels

   o   Create documentation folder

2. **Requirements & Design**

   o   Define user stories and acceptance criteria

   o   Create wireframes (Figma/Excalidraw)

   o   Database schema design (Firestore structure)

   o   API integration planning (Paystack, Firebase)

3. **Development Environment**

   o   Initialize React project (Vite/Next.js recommended)

   o   Set up ESLint, Prettier, Husky

   o   Configure Firebase project

   o   Create basic CI/CD pipeline for Vercel

Week 2: Core Architecture

1. **Project Structure Setup**

   o   Implement folder structure

   o   Set up routing (React Router)

   o   Create theme system (CSS-in-JS or Tailwind)

   o   Build reusable component library

2. **Firebase Integration**

   o   Authentication setup (Google, Email/Password)

   o   Firestore configuration with security rules

   o   Firebase Storage setup for media

3. **Authentication Flow**

   o   Login/Registration pages

- o Protected route wrapper

- o User role management (Admin, Pastor, Member)

🎌 Phase 2: Core Features Development (Week 3-5)

Week 3: Member Management Module

1. **Member CRUD Operations**

   - o Member registration form

   - o Member directory with search/filter

   - o Profile view/edit pages

   - o Family grouping functionality

2. **Database Integration**

   - o Firestore collections: members, families, groups

   - o Real-time updates for member data

Week 4: Events & Attendance

1. **Events Management**

   - o Create/edit events (Services, Meetings)

   - o Event calendar view

   - o Event registration

2. **Attendance Tracking**

   - o Check-in system (QR code or manual)

   - o Attendance reports

   - o Visitor tracking

Week 5: Financial Module

1. **Payment Integration Setup**

   - o Paystack API configuration

   - o Secure payment processing

   - o Payment history tracking

2. **Donations & Tithes**

   - o Online giving forms

   - o Recurring donation setup

   - o Receipt generation

⚙️ Phase 3: Advanced Features (Week 6-7)

Week 6: Communication & Reporting

1. **Communication Tools**
   - o Announcements board
   - o Email/SMS integration (Twilio/Resend)
   - o Notification system

2. **Reporting Dashboard**
   - o Analytics dashboard
   - o Financial reports
   - o Attendance trends
   - o Member growth charts

Week 7: Administration & Security

1. **Admin Features**
   - o User role management
   - o System configuration
   - o Audit logs

2. **Security & Performance**
   - o Implement Firestore security rules
   - o Add data validation
   - o Optimize performance
   - o Backup procedures

🚀 **Phase 4: Testing & Deployment (Week 8-10)**

Week 8: Testing

1. **Comprehensive Testing**
   - o Unit tests (Jest/Vitest)
   - o Integration tests
   - o User acceptance testing
   - o Security testing

2. **Bug Fixing**
   - o Critical bug resolution
   - o Performance optimization
   - o Accessibility improvements

Week 9: Deployment & Documentation

1. **Production Deployment**

   o   Final deployment to Vercel

   o   Domain configuration

   o   SSL setup

   o   Environment variables

2. **Documentation**

   o   User manuals

   o   Technical documentation

   o   API documentation

   o   Deployment guide

Week 10: Final Polish & Presentation

1. **Final Improvements**

   o   UI/UX polish

   o   Load testing

   o   Mobile responsiveness

   o   Browser compatibility

2. **Project Delivery**

   o   Final presentation preparation

   o   Demo preparation

   o   Code review and cleanup

   o   Handover documentation

🔧 **Technical Stack Checklist**

Frontend (React)

- React 18+ with TypeScript

- State management (Zustand/Context API)

- Form handling (React Hook Form)

- UI Library (Material-UI, Chakra, or custom)

- Chart library for reports (Recharts/Chart.js)

Backend & Services

- Firebase Authentication

- Firestore Database

- Firebase Storage

- Paystack API integration

- Optional: Cloud Functions for backend logic

DevOps & Tools

- GitHub for version control

- Vercel for hosting

- GitHub Actions for CI/CD

- Postman/Insomnia for API testing

- Figma for design collaboration

👥 Team Roles & Responsibilities

Suggested Division:

1. **Frontend Lead** - UI components, routing, state management

2. **Backend/DB Lead** - Firebase setup, security rules, data modeling

3. **Integration Lead** - Paystack, external APIs, authentication

4. **QA/Testing Lead** - Testing strategy, bug tracking, user testing

5. **DevOps Lead** - Deployment, CI/CD, performance monitoring

📈 Weekly Sprint Structure

Each Week:

- Monday: Sprint planning & task assignment

- Tuesday-Thursday: Development

- Friday: Internal review & testing

- Weekend: Catch-up & preparation for next sprint

🎯 Key Milestones

1. **Week 2 End**: Basic authentication working

2. **Week 5 End**: Member management & payment integration complete

3. **Week 7 End**: All core features functional

4. **Week 9 End**: System deployed and tested

5. **Week 10 End**: Project presentation ready

⚠️ Risk Mitigation

1. **Technical Risks**

- o Have fallback authentication method

- o Implement offline capabilities for critical features

- o Regular Firebase backups

2. **Team Risks**

- o Pair programming for complex features

- o Regular code reviews

- o Knowledge sharing sessions

3. **Timeline Risks**

- o Prioritize MVP features first

- o Weekly progress reviews

- o Buffer time for integration issues