Progress Report 1

1. Work Log Table

Work Date / Hours Logs

Date	Task Description	Hours
Jan 27	Reviewed literature on church management systems and administrative web applications.	2
Jan 28	Studied role-based access control (RBAC) and authentication approaches for web systems.	2
Jan 29	Data collection by interviewing the church administrator at Beria Baptist Church. Understanding the current system flow.	2
Jan 30	Identified system stakeholders and defined user roles (Admin, Pastor, Treasurer, Staff).	1.5
Jan 31	Defined functional requirements for authentication, user management, and navigation.	2
Feb 1	Defined non-functional requirements (security, usability, scalability).	1.5
Feb 2	Designed initial system architecture and selected technology stack (Node.js, MongoDB, HTML/CSS/JS).	2
Feb 3	Set up MongoDB database and created initial collections and schemas.	2
Feb 4	Implemented backend project structure and database connection. Code checked in the repo -> implementation folder -> backend	2
Feb 5	Developed backend authentication logic and JWT-based login flow. Code checked in the repo -> implementation folder -> backend	2
Feb 6	Created Welcome Page UI and navigation flow. Code checked in the repo -> implementation folder -> frontend	1.5

Date	Task Description	Hours
Feb 7	Developed Login Page UI and connected it to backend authentication API. Code checked in the repo -> implementation folder -> frontend	2
Feb 8	Tested login flow, token storage, and basic access control.	1.5
Feb 9	Reviewed progress, cleaned codebase, and committed work to repository.	1

2. Summary Description of Work Done During This Reporting Period

This week, I completed Phase 2 of the project, which included literature review, requirements analysis, and the start of system development. Research was conducted on church management system, role-based access control, authentication mechanisms, and RESTful web application architecture. These findings were used to define clear functional and non-functional requirements for the system.

In addition to documentation and planning, initial development was completed. This included setting up the MongoDB database, creating the backend project structure, and implementing authentication using JSON Web Tokens (JWT). The Welcome Page and Login Page were also developed to establish the main entry points into the system.

One challenge encountered was aligning security requirements with usability, particularly in managing authentication and access control. This was addressed by implementing token-based authentication while keeping the login flow simple and user-friendly. No major changes were made to the original project proposal; however, development was started earlier than initially planned to validate requirements through implementation. The next phase will focus on expanding backend APIs and developing core administrative features.

3. AI Use Section

AI Usage Description

AI tools were used as a development aid to:

- Clarify concepts related to JWT authentication and RBAC implementation

All AI-generated content was reviewed, modified, and integrated manually to ensure correctness and alignment with project requirements.

Sample Prompt Used (Appendix Reference)

"Explain how to implement role-based access control using JWT in a Node.js and Express application."

4. Repo Check-In of Implementation Completed

Since the last Project Proposal report, the foundational backend and frontend components of the system have been implemented and checked into the repository. This work establishes the technical base required for future development phases.

The files/folders checked into the repository include:

- MongoDB database setup and schema definitions
- Backend project structure (Node.js / Express)
- Authentication logic with JWT
- User login API endpoint
- Welcome Page (HTML/CSS)
- Login Page (HTML/CSS/JavaScript)

Appendix

- AI prompt(s) used (see AI Use Section)