CHURCH MANAGEMENT SYSTEM

PROJECT PROPOSAL

COURSE NAME: APPLIED RESEARCH PROJECT
COURSE ID: CSIS 4495_003
GROUP MEMBERS: BASIL RUGOYI **STUDENT ID:** 300371550

Table of Contents

1. Introduction

1.1 Background and Domain Overview

Effective information management is essential for the efficient operation of organizations across all sectors, including religious institutions such as churches. Churches are responsible for managing large volumes of data related to member registration, attendance tracking, and financial contributions such as tithes, offerings, and donations. Traditionally, many churches rely on manual systems such as paper files, registers, and spreadsheet-based tools. While these methods may be inexpensive and familiar, they are increasingly inadequate in meeting modern administrative demands.

The rapid advancement of web-based technologies and database-driven systems provides opportunities for churches to modernize their administrative processes. Full-stack web development using JavaScript technologies—such as Node.js, Express.js, and MongoDB—offers a scalable, flexible, and cost-effective solution for building centralized management systems. These systems can enhance data accuracy, improve accessibility, and support data-driven decision-making.

1.2 Problem Framing and Research Questions

Despite technological advancements, many churches continue to face challenges related to inefficient record-keeping practices. These challenges include duplicated records, data inconsistency, slow retrieval of information, limited reporting capabilities, and vulnerability to data loss. Furthermore, manual systems make it difficult for church leadership to analysed trends in attendance and membership growth.

This research seeks to address the following questions:

- How can a centralized, web-based database system improve the efficiency of church administration?
- In what ways can attendance and membership data be accurately captured and analysed using modern web technologies?

- What practical benefits can a full-stack JavaScript solution provide compared to traditional manual systems?

1.3 Literature Review and Knowledge Gaps

Previous studies on information systems in non-profit and religious organizations indicate that digitization improves operational efficiency, data accuracy, and transparency. Research on church management systems has largely focused on basic membership databases or financial accounting tools. However, many existing solutions lack integration, real-time reporting, and flexibility for customization.

There is a notable gap in research and practical implementation of lightweight, customizable, full-stack JavaScript-based systems tailored specifically for church administration. This project aims to address this gap by developing an integrated system that combines membership management, attendance tracking, and reporting within a single platform.

1.4 Research Assumptions, Hypotheses, and Benefits

This project is based on the assumption that church administrators are willing to adopt digital tools if they are user-friendly and cost-effective. The primary hypothesis is that a centralized web-based system will significantly improve data accuracy, accessibility, and administrative efficiency.

The potential benefits of this research include improved operational workflows, better decision-making through data analysis, and a reusable framework that can be adapted by other churches or non-profit organizations.

2. Proposed Research Project

2.1 Research Design and Objectives

This research adopts a **design-and-development research methodology**, focusing on the practical implementation and evaluation of a software system. The primary objective is to design, develop, and evaluate a full-stack JavaScript-based church administration system.

Specific objectives include:

- Designing a centralized database for church records
- Developing backend APIs for data processing and validation
- Implementing a responsive web-based user interface
- Providing reporting features such as attendance summaries and CSV export
- Evaluating system performance and usability

2.2 Methodology and Justification

The chosen methodology is justified by the applied nature of the research. Similar studies in information systems development emphasize iterative design, prototyping, and user feedback. Knowledge gained from prior coursework in web development, databases, and software engineering informs the design decisions and implementation approach.

2.3 Data Collection and Analysis

Data Collection Methods:

- Interviews with church administrators to identify system requirements
- Observation of existing manual record-keeping processes
- Review of relevant academic and technical literature

Sample Size:

- A simulated dataset representing church members and attendance records
- Realistic test data for evaluation purposes

Data Analysis Techniques:

- Functional testing of system features
- Comparison of manual versus digital record retrieval times

- Evaluation of data accuracy and reporting capabilities

2.4 Technologies Used

- **Operating System / Platform:** Windows 10 / Web-based platform
- **Programming Languages:** JavaScript, HTML, CSS
- **Backend Framework:** Node.js with Express.js
- **Database:** MongoDB
- **Frontend:** HTML5, CSS3, Vanilla JavaScript
- **Development Tools:** Visual Studio Code, Postman, GitHub

2.5 Expected Results and Practical Applications

The expected outcome is a fully functional, secure, and user-friendly church administration system. The system is expected to reduce record duplication, improve data accuracy, and enable faster retrieval of information. The reporting features will support data-driven decision-making, while the modular design will allow future enhancements.

3. Riipen External Partner

This project is an academic independent project and does not currently involve a Riipen external partner.

4. Project Planning and Timeline

4.1 Project Phases and Milestones

Project Timeline (January 10 – April 7, 2026)

Phase	Task Description	Start Date	End Date	Key Deliverables
Phase 1	Project Initiation & Proposal Development	Jan 10	Jan 26	Approved project proposal
Phase 2	Literature Review & Requirements Analysis	Jan 27	Feb 9	Literature summary, requirements specification
Phase 3	System Architecture & Database Design	Feb 10	Feb 23	ER diagrams, system architecture design
Phase 4	Backend Development & Authentication	Feb 24	Mar 3	REST APIs, JWT authentication, role-based access
Phase 5	Core Module Development (Members, Attendance, Finance)	Mar 4	Mar 17	Functional backend and frontend modules
Phase 6	Chatbot Integration & UI Enhancements	Mar 18	Mar 25	Church information chatbot, UI improvements
Phase 7	Testing, Debugging & Validation	Mar 26	Apr 1	Test cases, bug fixes, validation results
Phase 8	Documentation & Final Report Preparation	Apr 2	Apr 6	Final report, technical documentation
Phase 9	Final Review & Submission	Apr 7	Apr 7	Final submission

4.2 Team Responsibilities

- **Team Lead (Basil Rugoyi):**
    - Overall project coordination
    - Backend and database development
    - Final integration and documentation

4.3 Project Management Approach

Project Management Tool: Gantt Chart

A **Gantt Chart–based planning approach** will be used to track milestones and deadlines. Tasks are divided into weekly deliverables to ensure steady progress and accountability. The Gantt Chart provides a structured visual representation of project phases, task dependencies, milestones, and deliverables. This approach enables effective progress tracking, accountability, and timely completion of the research project.

The Gantt Chart supports:

- Clear task sequencing and deadlines
- Identification of critical milestones
- Improved coordination among team members
- Alignment with Agile-inspired incremental delivery

Gantt Chart (Weeks, Tasks, Milestones)

Task / Phase	Week 1 (Jan 10-19)	Week 2 (Jan 20-26)	Week 3 (Jan 27-Feb 2)	Week 4 (Feb 3-9)	Week 5 (Feb 10-16)	Week 6 (Feb 17-23)	Week 7 (Feb 24-Mar 2)	Week 8 (Mar 3-9)	Week 9 (Mar 10-16)	Week 10 (Mar 17-23)	Week 11 (Mar 24-30)	Week 12 (Mar 31-Apr 7)
Project Planning & Proposal Drafting												
Literature Review & Background Research												
Requirements Analysis & System Design												
Database Design (MongoDB)												
Backend Development (Node.js / Express)												
Frontend Development (HTML, CSS, JS)												
Chatbot Integration & UI Enhancements												
System Testing & Bug Fixes												
Documentation & Final Report Writing												
Final Review & Submission												

5. Project Contract

I, the undersigned, agree to the scope, responsibilities, and timelines outlined in this project proposal. I commit to meeting regularly, completing assigned tasks on schedule, and collaborating effectively throughout the project duration.

Signatures:

Name: __Basil Rugoyi_____

Signature: ____B.Rugoyi_____

Date: ____20 January 2026_____

6. AI Use Section

AI Tool Name	Version / Account Type	Specific Use	Value Added	Human Contribution
ChatGPT	GPT-5.2 – Free	Code assistance, documentation	Improved productivity	Code validation, customization
DALL·E	Free	UI image generation	Visual enhancement	UI integration and styling

Appendix: Prompt History

(All AI prompts and responses used during development will be documented and stored in the appendix.)

7. Work Date / Hours Logs

Work Log Table (To Be Updated Regularly)

Date	Task Description	Hours
Jan 10	Requirement analysis and proposal drafting.	2
Jan 13	Clarify project goals and deliverables, Project proposal discussion.	1.5
Jan 14	Identify key stakeholders.	1.5
Jan 14	Draft communication plan.	1
Jan 15	Conduct research & collect relevant data.	3
Jan 16	Create Gantt chart or timeline.	3
Jan 19	Identify required resources.	2
Jan 20	Structure the project proposal document.	2
Jan 22	Define Scope and Objectives, Clarify system features and research goals.	1
Jan 23	Find relevant articles on church management systems.	2
Jan 26	Final checks, formatting, and submission of the Project Proposal report.	2

Note: Logs will be updated daily to reflect actual work completed.

8. Closing and References

Acknowledgements

I would like to sincerely acknowledge and express my gratitude to the course instructor, Padmapriya Arasanipalai Kandhadai (Douglas College), for her unwavering guidance, support, and valuable insights during the initial stages of this project. Her expert advice and encouragement have been instrumental in helping me navigate the early challenges and lay a strong foundation for the successful progression of this project.

References

- Sommerville, I. (2016). *Software Engineering*. Pearson.
- OWASP Foundation. (2023). *Web Application Security Guidelines*.
- MongoDB Documentation. (2024)
- Express.js Documentation. (2024).

9. Conclusion

This proposal presents a structured plan for the design and implementation of a full-stack JavaScript-based church administration system. By addressing the limitations of manual record-keeping and leveraging modern web technologies, the project aims to deliver a practical, scalable, and efficient solution that enhances church administration and supports data-driven leadership.