

MBEYA UNIVERSITY OF SCIENCE AND TECHNOLOGY



COLLEGE OF INFORMATION SYSTEM AND TECHNOLOGY

DEPARTMENT: INFORMATION SYSTEM AND TECHNOLOGY
COURSE: BACHELOR OF SCIENCE IN INFORMATION AND
COMMUNICATION TECHNOLOGY
MODULE NAME: MOBILE APPLICATION DEVELOPMENT
MODULE CODE: IT 8203
FACILITATOR NAME: MR RUKOJO A
TASK: GROUP PROJECT
LEVEL: UQF 8 SECOND YEAR
YEAR OF STUDY: 2025/2026
TITLE OF PROJECT: PROJECT MANAGEMENT SYSTEM

S/NO	NAME OF STUDENT	REGISTRATION NUMBER	CA NUMBER
01	JOSHUA S. RAMADHAN	24101133370017	CA/BScICT/25/13470
02	JUMA S. JUMA	24101133370063	CA/BScICT/25/13457
03	FLORIAN B MISANGA	24101334410085	CA/BScICT/25/14381
04	NOEL N. TARIMO	24101133370079	CA/BScICT/25/13472

TABLE OF CONTENTS

Declaration	i
Acknowledgment	ii
Abstract	iii
Introduction	1
1.1 Problem Description	1
1.2 Motivation	2
1.3 Project Objectives	2
1.3.1 General Objective	3
1.3.2 Specific Objectives	3
System Requirements	4
2.1 Functional Requirements	4
2.2 Non-Functional Requirements	5
System Architecture	6
3.1 Overall System Design	7
3.2 Client-Server Architecture	7
3.3 Database Schema (ERD)	8
3.4 Authentication Mechanism	9
3.5 Security Measures Used	10
Implementation Details	11
4.1 Technologies Used	

4.2 Frontend Implementation	12
4.3 Backend Implementation	13
4.4 Database Design	14
4.5 Challenges Faced	15
4.6 Assumptions Made	16
Security Considerations	17
5.1 Authentication Handling	17
5.2 Password Storage	18
5.3 Data Protection	18
5.4 Access Control	19
Testing & Results	20
6.1 Testing Performed	20
6.1.1 Registration Testing	21
6.1.2 Login Testing	21
6.1.3 Project Creation Testing	22
6.1.4 Task Assignment Testing	22
6.1.5 Role-Based Access Testing	23
Future Work	24
7.1 Possible Improvements	24
7.2 Scalability Considerations	25
Conclusion	26

References	27
Appendices	28

Declaration

We are here by declare that this project titled “Project Management System” is our original work and has not been submitted to any other institution for academic purposes. All sources and references used in this project have been fully acknowledged.

Acknowledgment

We would like to express our sincere gratitude to our supervisor RUKOIJO AMANI for their guidance, support, and constructive feedback throughout the development of this project.

We are also grateful to the faculty and lecturers of college of information system and technology for providing the knowledge and skills that enabled us to complete this project successfully.

Special thanks to our families and friends for their encouragement and moral support.

Finally, we are appreciate all individuals who contributed indirectly to this project, including peers who provided valuable insights and suggestions.

Abstract

This project presents the design and implementation of a web-based Project Management System aimed at improving project planning, task tracking, and team collaboration. Traditional project management methods often suffer from lack of transparency, inefficient communication, and difficulty in monitoring progress.

The system was developed using HTML, CSS, JavaScript for the frontend, Node.js and Express.js for the backend, and PostgreSQL as the database. Key features of the system include user registration and authentication, role-based access control, project creation and project progress tracking.

Testing was conducted using unit, integration, system, and user acceptance tests to ensure all functionalities meet the specified requirements. Results show that the system efficiently manages projects and tasks, enforces secure authentication, and provides an intuitive interface for different user roles.

The Project Management System offers a reliable and scalable solution for organizations seeking to improve project efficiency and collaboration. Future enhancements may include email notifications, real-time updates, and advanced reporting dashboards.

PROJECT TITLE: PROJECT MANAGEMENT SYSTEM

1. Introduction

1.1 Problem Description

Many organizations and teams struggle to manage projects efficiently. Tasks are often assigned manually, progress is not tracked properly, deadlines are missed, and communication between team members becomes difficult. Without a centralized system, project monitoring becomes confusing and unorganized.

The Project Management System was developed to solve these problems by providing a centralized web-based platform where projects, tasks, deadlines, and users can be managed efficiently.

1.2 Motivation

The motivation behind developing the Project Management System is to improve project planning, monitoring, and collaboration among team members. Modern organizations require digital tools that allow real-time tracking of tasks and performance. This system ensures transparency, accountability, and productivity.

1.3 Project Objectives

General Objective

To design and implement a secure web-based Project Management System for managing projects and tasks efficiently.

Specific Objectives

- To allow administrators to create and manage projects.
- To allow team members to view assigned tasks.
- To track project progress.

- To implement secure user authentication.
- To ensure role-based access control.
- To store project data securely in a database.

2. System Requirements

2.1 Functional Requirements

The Project Management System shall:

- *Allow users to register and log in.
- *Allow administrators to create, edit, and delete projects.
- *Allow project managers to assign tasks to team members.
- *Allow users to update task status (Pending, In Progress, Completed).
- *Allow users to view project deadlines.
- *Allow administrators to manage users.
- *Generate project progress reports.

2.2 Non-Functional Requirements

The Project Management System shall:

- *Be secure and protect sensitive project data.
- *Be responsive and accessible on different devices.
- *Load within acceptable time limits.

*Support multiple users simultaneously.

*Be scalable for future expansion.

3. System Architecture

3.1 Overall System Design

The Project Management System follows a three-tier architecture:

>Presentation Layer (Frontend)

>Application Layer (Backend)

>Database Layer

This separation ensures maintainability, scalability, and security.

3.2 Client-Server Architecture

The Project Management System uses a client-server architecture:

Client (Web Browser)

↓

Frontend (HTML, CSS, JavaScript)

↓

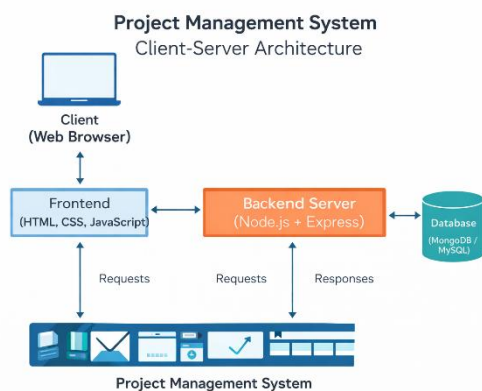
Backend Server (Node.js + Express)

↓

Database (postgresSQL)

Users interact with the frontend, which sends requests to the backend. The backend processes requests and communicates with the database.

DIAGRAM OF THE CLIENT_SERVER ARCHITECTURE



3.3 Database Schema (ERD)

1. Users Table

Stores all users of the system (Admin, Project Manager, Team Member).

Field Name

Data Type

Constraints

Description

id

INT / ObjectId

Primary Key, Auto Increment

Unique identifier

name

VARCHAR(100)

NOT NULL

Full name of user

email

VARCHAR(100)

UNIQUE, NOT NULL

User email for login

password

VARCHAR(255)

NOT NULL

Hashed password

role

ENUM

NOT NULL (Admin, Manager, Team Member)

User role for access control

created_at

Account creation date

2. Projects Table

Stores all projects created in the system.

Field Name

Data Type

Constraints

Description

id

INT / ObjectId

Primary Key, Auto Increment

Unique project identifier

project_name

VARCHAR(100)

NOT NULL

Project title

description

TEXT

NULL

Project description

start_date

DATE

NOT NULL

Project start date

end_date

DATE

NOT NULL

Project deadline

status

ENUM

NOT NULL (Pending, Ongoing, Completed)

Project current status

created_by

INT

Foreign Key → Users.id

User who created the project

created_at

TIMESTAMP

DEFAULT CURRENT_TIMESTAMP

Project creation date

3. Tasks Table

Stores tasks assigned to team members under projects.

Field Name

Data Type

Constraints

Description

id

INT / ObjectId

Primary Key, Auto Increment

Unique task identifier

task_name

VARCHAR(100)

NOT NULL

Task title

description

TEXT

NULL

Task description

project_id

INT

Foreign Key → Projects.id

Project to which task belongs

assigned_to

INT

Foreign Key → Users.id

User assigned to this task

deadline

DATE

NOT NULL

Task deadline

status

ENUM

NOT NULL (Pending, In Progress, Completed)

Relationships:

>One user can create many projects.

>One project can have many tasks.

>One user can be assigned many tasks.

3.4 Authentication Mechanism

The Project Management System uses token-based authentication (JWT). When a user logs in successfully, a secure token is generated. This token is required to access protected routes such as creating projects or assigning tasks.

3.5 Security Measures Used

- >Password hashing using bcrypt.
- >JWT authentication.
- >Input validation.
- >Role-based access control.
- >Protected API routes.
- >Secure database queries.

4. Implementation Details

4.1 Technologies Used

- >Frontend:HTML,CSS and JavaScript
- >Backend:Node.js,Express.js
- >Database:PostgreSQL
- >Tools:Visual Studio Code,Postman and Git

4.2 Frontend Implementation

The frontend of the Project Management System includes:

- >Login and registration pages.
- >Dashboard page.
- >Project creation form.
- >Task assignment interface.

>Project status view.

>API requests are handled using fetch or Axios.

4.3 Backend Implementation

The backend handles:

>User authentication routes (/register, /login).

>Project routes (/create-project, /update-project).

>Task routes (/assign-task, /update-status).

>Middleware for authentication and authorization.

>Database connection and queries.

4.4 Database Design

The database ensures:

>Proper relationships using foreign keys.

>Secure password storage (hashed).

>Data consistency using constraints.

4.5 Challenges Faced

>Connecting frontend to backend.

>Handling authentication tokens.

>Managing user roles.

>Database relationship design.

>Fixing network errors.

4.6 Assumptions Made

>Users have internet access.

>Only authorized users can manage projects.

>Each task belongs to only one project.

5. Security Considerations

Authentication Handling

JWT-based authentication ensures that only logged-in users access the system.

Password Storage

Passwords are hashed using bcrypt before being stored.

Data Protection

Server-side validation.

Secure API endpoints.

Access restriction based on roles.

Access Control

Admin: Full access.

Manager: Manage projects and tasks.

Team Member: View and update assigned tasks only.

6. Testing & Results

Testing is a critical phase in the development of the Project Management System. It ensures that the system functions according to the specified requirements and that all components work correctly together. The main goal of testing was to verify system reliability, security, and performance.

*Testing Performed

*Registration testing

*Login testing

*Project creation testing

*Task assignment testing

*Role-based access testing

*All major functionalities of the Project Management System were tested successfully.

7. Future Work

Possible Improvements

>Email notifications.

>Real-time updates using WebSockets.

>File attachments for tasks.

>Advanced reporting dashboard.

>Scalability Considerations

>Cloud deployment.

>Database indexing.

>Load balancing.

>Microservices architecture.

REFERENCES

*Use academic format (APA example):

Author, A. (Year). Title of book. Publisher.

OR

Website citation.

APPENDICES

Include:

*Source code snippets

*Full database schema

*Extra screenshots