MUST SOFTWARE DEVELOPMENT REPORT

STUDENT NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

REGISTRATION NUMBER: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DEPARTMENT: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LEVEL: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ACADEMIC YEAR: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

UNIVERSITY SUPERVISOR: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# ACKNOWLEDGMENT

I would like to express my deepest gratitude to my family, lecturers, supervisors, and classmates for their unwavering support throughout the learning process of software development. Special thanks go to my peers and fellow students who collaborated with me in group projects, shared coding ideas, and provided constant motivation. Finally, I thank God for the strength, wisdom, and good health that enabled me to accomplish this learning milestone.

# SUMMARY

This report presents a comprehensive explanation of the software development learning process, covering both theoretical and practical knowledge gained. It explores IDEs such as Visual Studio, IntelliJ IDEA, PgAdmin4, and Postman, and also introduces the curriculum project that integrates frontend (React.js) and backend (PostgreSQL) development.

# LIST OF ABBREVIATIONS

IDE: Integrated Development Environment – A platform providing coding and debugging tools. Example: Visual Studio.

API: Application Programming Interface – Communication rules between systems. Example: Postman tested endpoints.

DB: Database – Organized collection of data. Example: PostgreSQL.

SQL: Structured Query Language – Commands to manage relational databases.

UI: User Interface – Visual elements users interact with.

CRUD: Create, Read, Update, Delete – Core database operations.

MVC: Model-View-Controller – Architecture separating data, UI, and control logic.

HTTP: Hypertext Transfer Protocol – Foundation of web communication.

# CHAPTER ONE

## 1.0 INTRODUCTION

Software development is the systematic process of designing, creating, testing, and maintaining applications. It enables automation, innovation, and efficiency across industries.

## 1.1 INTEGRATED DEVELOPMENT ENVIRONMENTS (IDEs)

### 1.1.1 Visual Studio

Visual Studio is a Microsoft IDE supporting .NET, C#, and many languages. Features include Solution Explorer, IntelliSense, NuGet Manager, and advanced debugging.

[Insert screenshot of Visual Studio interface here]

### 1.1.2 IntelliJ IDEA

IntelliJ IDEA by JetBrains is strong for Java and Spring Boot. Features include Maven/Gradle, refactoring tools, Git integration, and REST API development.

[Insert screenshot of IntelliJ IDEA interface here]

### 1.1.3 PgAdmin4

PgAdmin4 manages PostgreSQL with a GUI for queries, ERDs, backups, and CRUD operations on databases.

[Insert screenshot of PgAdmin4 interface here]

### 1.1.4 Postman

Postman is used to test APIs with GET, POST, PUT, DELETE requests. It supports collections, automated tests, and debugging headers/responses.

[Insert screenshot of Postman interface here]

# CHAPTER TWO

## 2.0 CURRICULUM PROJECT STRUCTURE

### 2.1 Meaning of Curriculum

A curriculum is a structured plan of courses delivered by an institution. At MUST, Computer Engineering includes courses such as Programming, Data Structures, Operating Systems, and Networks. A digital curriculum system improves storage, retrieval, and updates.

[Insert screenshot of MUST curriculum structure here]

### 2.2 Tools Required

- Node.js and React.js  
- PostgreSQL and PgAdmin4  
- IntelliJ IDEA / Visual Studio  
- Postman  
- Git and GitHub

### 2.3 Step by Step Process

#### Step 1: Install Node.js

Install Node.js using: sudo apt update && sudo apt install nodejs npm

#### Step 2: Create React Project

Create React project: npx create-react-app curriculum-frontend

[Insert screenshot of React app in browser here]

#### Step 3: Frontend Structure

Components: CollegeList, DepartmentList, CourseList, SemesterList, CurriculumView.

[Insert screenshot of React components here]

#### Step 4: Install PostgreSQL

Install: sudo apt install postgresql postgresql-contrib. Create DB: CREATE DATABASE must\_curriculum;

[Insert screenshot of PostgreSQL DB creation here]

#### Step 5: Database Schema

Tables: colleges, departments, courses, semesters, curriculums with relationships (one-to-many, many-to-many).

[Insert screenshot of ERD here]

#### Step 6: CRUD Operations

Examples: INSERT INTO colleges; SELECT \* FROM courses; UPDATE courses; DELETE FROM students.

[Insert screenshot of CRUD queries in PgAdmin4 here]

#### Step 7: Backend API Development

APIs built with Node.js/Express or Spring Boot. Endpoints include GET /colleges, POST /departments, PUT /courses/:id, DELETE /semesters/:id.

[Insert screenshot of Postman tests here]

#### Step 8: Integration

React fetches backend data using Axios. Example: axios.get('http://localhost:5000/api/courses').

[Insert screenshot of React UI showing curriculum data here]

### 2.4 Problems and Solutions

Challenges: installation errors, DB connection issues, API errors, state management. Solutions: update packages, configure ports, debug routes, use React hooks.

# CHAPTER THREE

## 3.0 RECOMMENDATIONS

Use IDEs as per needs, expand knowledge with advanced frameworks, practice Git, improve deployment (cloud), emphasize security.

## 3.1 CONCLUSION

The project provided practical skills in frontend (React) and backend (PostgreSQL), along with IDE experience. It showed how digital systems improve curriculum management.

## 3.2 REFERENCES

1. PostgreSQL Documentation. https://www.postgresql.org/docs/  
2. React Documentation. https://react.dev/  
3. Node.js Documentation. https://nodejs.org/en/docs  
4. JetBrains IntelliJ IDEA Guide. https://www.jetbrains.com/idea/  
5. Microsoft Visual Studio Documentation. https://learn.microsoft.com/en-us/visualstudio/  
6. Postman Learning Center. https://learning.postman.com/  
7. Mbeya University of Science and Technology Curriculum Handbook.  
8. W3Schools Tutorials. https://www.w3schools.com/