TITLE :- LEARNING MANAGEMENT SYSTEM

DIGITAL ASSIGNMENT – 1

SOFTWARE ENGINEERING LAB

COURSE CODE :- BCSE301P

NAME :- SUNKARA GOWTHAM SAI

REGISTRATION NUMBER :- 22BCT0062

INDEX :-

1. INTRODUCTION

A **Learning Management System (LMS)** is a software application designed to facilitate the creation, delivery, and management of educational content and learning experiences. It is widely used in educational institutions, corporate training environments, and other organizations to streamline teaching and learning processes.

The core idea behind an LMS is to provide a centralized platform where learners and educators can interact effectively. It typically includes features such as:

1. **Course Management**: Create, organize, and manage courses, modules, and assignments.
2. **User Roles**: Assign roles such as administrators, instructors, and learners with different access levels.
3. **Content Delivery**: Upload and share multimedia content like videos, documents, and quizzes.

1.1 PURPOSE

The purpose of this project is to develop a Learning Management System (LMS) that facilitates the creation and management of online courses by educators and provides students with a platform to enrol, learn, and interact. The system aims to streamline course management, enhance learning experiences, and ensure security and scalability for all users.

1.2 SCOPE OF THE PROJECT

The LMS is designed to:

- Allow educators to set up and manage various online courses.
- Enable students to enrol in and take courses.
- Provide tools for creating quizzes/exams and tracking course performance.
- Ensure secure access through robust authentication and data protection mechanisms.
- Support scalability for future feature additions and increased user base.

1.3 DEFINITONS

- LMS: Learning Management System
- Educator: A user responsible for creating and managing courses.
- Student: A user who enrols in and takes courses.
- CSRF: Cross-Site Request Forgery
- ORM: Object-Relational Mapping
- Sequelize: A promise-based ORM for Node.js
- EJS: Embedded JavaScript Templates

2. OVERALL DESCRIPTION

A **Learning Management System (LMS)** is a software platform that provides a structured environment for delivering educational content, tracking learning progress, and managing administrative tasks related to training and education. LMS solutions are designed to make teaching and learning more accessible, efficient, and engaging.

2.1 PRODUCT PERSPECTIVE

This LMS will integrate a web-based interface for both educators and students. It will employ a three-tier architecture comprising the frontend (HTML, CSS, JavaScript, EJS), backend (Node.js, Express.js), and database (PostgreSQL). The system ensures user data security and streamlined interactions using modern web technologies and secure coding practices.

2.2 PRODUCT FUNCTIONS

- Educator Features:

  - Sign in to the system.

  - Create, update, and delete courses.

  - Browse course reports, including enrolments and performance.

  - Set up quizzes/exams for courses.

- Student Features:

  - Sign in to the system.

  - Browse and enrol in available courses.

- Take quizzes/exams for enrolled courses.

- View course progress and results.

2.3 USER CHARACTERISTICS

• Educators: Familiar with basic computer operations, responsible for course content creation and monitoring student progress.

• Students: Diverse demographic; possess basic knowledge of web navigation and are eager to learn from the available courses.

2.4 CONSTRAINTS

• The system must be hosted on Render, limiting hosting configurations to their available offerings.

• Courses and content should adhere to a predefined format for compatibility.

• Performance and response times must remain optimal for up to 10,000 concurrent users.

2.5 ASSUMPTIONS AND DEPENDENCIES

- Users will access the system via modern web browsers.

- All external libraries and tools, such as Passport.js and Tiny-CSRF, are regularly maintained and updated.

- Internet connectivity is required for all system functionalities.

3. REQUIREMENTS

The requirements for developing an LMS can be divided into **functional requirements** and **non-functional requirements**. These requirements help define the features, behavior, and performance of the system.

3.1 FUNCTIONAL REQUIREMENTS

- Educators can create, edit, and delete courses.

- Students can enrol in courses and access content.

- The system supports user authentication and role-based access.

- Course performance reports are accessible to educators.

- Quizzes and exams can be created, taken, and scored within the system.

3.2 NON–FUNCTIONAL REQUIREMENTS

- The system should load within 3 seconds under normal conditions.
- Scalability to handle a growing number of courses and users.
- The user interface should be responsive and accessible on various devices.

3.3 SECURITY

- User passwords must be hashed using Bcrypt.
- CSRF protection implemented using Tiny-CSRF.
- Secure cookies for session management.
- Regular security audits and vulnerability scans.

3.4 SAFETY

- Backups of the database must be taken daily to prevent data loss.
- A fail-safe mechanism should be in place to handle unexpected crashes or server downtime.
- Data integrity checks must be performed regularly to ensure accuracy and consistency.