MAKERERE UNIVERSITY BUSINESS SCHOOL

FACULTY OF COMPUTING AND INFORMATICS

BACHELOR OF BUSINESS COMPUTING YEAR 3

ADVANCED WEB APPLICATION DEVELOPMENT

COURSE WORK TWO

PROJECT REPORT & REQUIREMENT DOCUMENTATION

BY

NAME	REG NUMBER	SIGNATURE
ORISHABA GILBERT	22/U/6828	

Requirement Document for Chaapa Students Management System

Introduction

This document outlines the development of the Chaapa Students Management Information System using the Laravel framework. The system aims to assist educational institutions in managing student data, course information, enrollments, and report generation efficiently. By providing a structured approach to data management, Chaapa enhances the administrative processes within educational settings.

Functional Requirements

a) Student Information Management

CRUD Operations: Administrators can add, update, and delete student profiles.

Profile Access: View student profiles based on names, emails, or student IDs.

b) **Course Management**

Course Lifecycle Management: Admins can create, modify, and delete courses.

Enrollment Management: Link students to specific courses and manage their enrollment s**tatuses.**

c) Reports and Analytics

Reports Generation: Generate reports on student enrollments and course participation to monitor engagement and performance.

d) **User Authentication and Access Control**

Secure Login: Implemented authentication using Laravel's Jetstream.

Role-Based Access: Restrict functionalities to authorized administrators.

e) Admin Dashboard

Overview Metrics: Display key statistics such as student counts, available courses, and recent enrollments for quick insights.

3. Non-Functional Requirements

a) Performance

The system handles multiple concurrent users without significant delays.

b) Scalability

The architecture should support growth in the number of users and data volume.

c) Security

Implemented strong authentication, data encryption, and protections against vulnerabilities like SQL injection.

d) User Interface

Provide a user-friendly interface using Blade templates for intuitive navigation.

e) Maintainability

Ensured the system adheres to Laravel's MVC design principles for easy updates and maintenance.

4. System Architecture

The system utilizes Laravel's MVC architecture, which separates data handling, logic, and the user interface:

Students Entity: Stores student data, including names and enrolled courses.

Courses Entity: Contains details such as course names and credit hours.

Enrollments Entity: Manages relationships between students and courses. Provided as pivot table

Admins Entity: Contains information about administrators managing the system.

5. Database Design

Used Microsoft Workbench to be able to achieve a database structure design for the Chaapa Management Students Management.

The system will consist of four core tables:

Students Table: student_id (PK), name, email, phone_number, date_of_birth, enrollment_status, profile_image_url

Courses Table: course_id (PK) course_name, description, credit_hours

Admins Table: admin_id, student_id, name, email, role

Enrollments Tables: enrollment_id, student_name (FK), email (FK), course_id (FK), enrollment_date, status, grade, DOB, phone, student_id

Report table: report_type, generated_at, generated_by, course_id,

Version Control

In my Chaapa Student Management System used GitHub to be able to track my changes regularly

Report on the Development of the Chaapa Student Information and Course Management System

Introduction

The Chaapa Student Management System assists educational institutions in managing student and course data efficiently. This report summarizes the development process, architecture, key features, and challenges encountered during implementation.

System Overview

The Student Management System enables administrators to:

Manage student profiles.

Handle course lifecycle operations.

Enroll students and manage their statuses.

Generate insightful reports.

Access a secure, role-based admin panel with dashboard features.

Development Phases

a) Environment Setup

Configured the Laravel environment with Composer dependencies, routing, and Blade templates. GitHub version control ensured that changes were tracked effectively.

b) Database Design

An Entity-Relationship (ER) diagram was created to visualize relationships among key entities. The database was established using Laravel migrations with defined foreign keys to maintain integrity.

c) Module Development

Implemented CRUD operations for student profiles and lifecycle management for courses, linking students through the enrollment module.

d) Report Generation

Reports on student participation and engagement were generated to aid administrators in monitoring progress.

e) Authentication

Secured the admin panel using Laravel Jetstream, ensuring that only authorized users could access sensitive data.

Challenges and Solutions

a) Database Relationships

Managing relationships via the enrollments (that I wanted to create as a pivot table) table presented challenges, which were resolved by correctly defining foreign keys.

b) User Authentication

Implementing role-based access control was complex; however, using Laravel Breeze allowed for secure access to sensitive areas.

5. Conclusion

The Chaapa Students Management System offers a comprehensive solution for managing student and course data. By leveraging Laravel's MVC framework, the system provides a user-friendly interface, role-based access control, and essential reporting features, demonstrating effective database design, web development, and collaboration.

GitHub Repository Link: https://github.com/gilbertorishaba/AWADCW2