Benedict XVI Catholic International Institute of Higher Education

BCI

Student Lecturer Management System

Project Report

Group Software Project - 2024

Project ID: Group 01

Submitted by:

1. BSIT221015 – Prasansa Nawanjalee
2. BSIT221002 – Kavindu Pansilu
3. BSIT221016 – Nuwan Sri Lal

Submitted to:

Mr Kasun Jayananda

08.07.2024

Declaration

We declare that this project report or part of it was not a copy of a document done by any organization, university any other institute, or a previous student project group at BCI and was not copied from the Internet or other sources.

Project Details

Project Title	Student Lecturer Management System
Project ID	Group 01

Group Members

Reg. No	Name	Signature
BSIT221015	Prasansa Nawanjalee	
BSIT221002	Kavindu Pansilu	
BSIT221016	Nuwan Sri Lal	

Supervisor Name Date Signature

Mr Kasun Jayananda
…….. …….. ……..

Abstract

This report outlines the development and implementation of a Student-Lecture Management System designed to optimize operations in significant challenges in managing student and lecturer information a accurately recording attendance. These inefficiencies lead to errors and data management issues. To address these problems, the software was designed and developed the Student-Lecture Management System (SLMS), a web-based application using the Laravel framework and QR technology. The SLMS provides centralized information management, automated attendance tracking, comprehensive reporting. The system ensures data integrity, security, and a user-friendly interface by leveraging advanced technologies. In conclusion, the SLMS offers a comprehensive solution to the administrative challenges faced by BCI Campus, contributing to academic excellence and administrative efficiency.

Acknowledgement

The development team would like to express its deepest gratitude to all who have supported the creation of the Student-Lecture Management System (SLMS). First and foremost, sincere thanks are extended to the project supervisor, Mr. Kasun Jayananda, for invaluable guidance, encouragement, and insightful feedback, which greatly contributed to the successful completion of this project. The supervisor's expertise and continuous support were instrumental in navigating the challenges encountered. Acknowledgment is also given to team members, M.P.Nawanjalee, L.D.K.P.Randeepa, N.S.L.Senevirathna for their collaborative spirit, constructive suggestions, and support. Thanks are extended to all contributors and supporters, whose involvement was fundamental to the success of this project.

Table Of Content

List of Figures

List of Tables

1. Introduction

As the development of the technology there are many opportunities in it. So in this process client face some difficulties. As the solution for it, the project is to design and develop a web-based Student-Lecture Management System for BCI Campus. This system aims to solve issues previously faced by students and lecturers. The system discussed enable student attendance tracking and attendance report messaging. Developed using Laravel and QR technology, the system will feature a user-friendly interface and offer functionalities such as student enrollment, course registration, attendance tracking, and academic performance monitoring. This comprehensive tool is designed to meet the specific needs of educational institutions, aiding in strategic planning and resource management.

1.1 Problem Statement

The BCI Campus faces challenges in efficiently managing student and lecturer information and as well as in accurately recording attendance. Currently, there is a lack of a centralized system to streamline these processes, leading to inefficiencies and errors in data management.

Student and Lecturer Information Management:

The existing manual system for managing student and lecturer information is prone to errors and inefficiencies. There is a need for a centralized system to store and manage student and lecturer data securely and efficiently.

Attendance Tracking:

The current method of recording student attendance is time-consuming and prone to inaccuracies. There is a need for an automated system to track student attendance reliably and generate reports for analysis.

Reporting and Communication:

There is a lack of a system for generating and distributing attendance reports to students and lecturers. There is a need for a system that automates the generation and distribution of attendance reports to relevant stakeholders.

Security and Data Integrity:

The current system lacks adequate security measures to protect sensitive student and lecturer information. There is a need for a system that implements robust security measures to safeguard data integrity and confidentiality.

Maintenance and Updates:

The current system is difficult to maintain and update with new features and information. There is a need for a system that is easy to maintain and update, allowing for seamless integration of new functionalities and data updates.

1.2 Product Scope

The scope of this project is to design and develop a web based application to BCI Campus. This system is make to full fill the requirements and as a solution for the problems that faced by them. The system will enhance data accuracy, operational efficiency, and communication within the campus community through features like automated attendance tracking using QR technology and streamlined leave management. Finally this can achieved to take student attendance and send messages of the student attendance report.

1.3 Project Report Structure

This report is structured as follows: the methodology section details the requirements, design, implementation, and testing processes. The evaluation section assesses the project results, discusses lessons learned, and proposes future work. The report concludes with a summary of findings and benefits of the SLMS. Appendices provide additional technical details, including design diagrams, test results, and selected code listings.

2. Related Work

2.1 Introduction

The requirements for the Student-Lecture Management System (SLMS) were gathered through interviews with stakeholders and observation of current processes, and analysis of existing documentation

2.2 User Management

Ability to add, update, and delete student and lecturer information.

2.3 Attendance Management

Automated attendance tracking using QR technology.

2.4 Reporting

Generation of comprehensive reports on attendance, and other relevant data.

2.5 Communication

Enhanced communication channels between students, lecturers, and administrators.

3. System Analysis

3.1 Introduction

The Student-Lecture Management System (SLMS) is developed to streamline the administrative tasks of managing student and lecturer information and tracking attendance at BCI Campus. The current manual processes lead to inefficiencies, errors, and data management issues. The system aims to address these challenges by providing an automated, and user-friendly web-based application. This system leverages the Laravel framework and QR technology to ensure data integrity, security, and efficient management of academic operations.

3.2 System Requirements

3.2.1 User requirements

Students:
Receive notifications regarding student QR.
Mark attendance using QR technology.

Lecturers:
Receive students attendance.
Receive user details through email.

Administrators:
Manage student and lecturer profiles.
Generate and view reports on attendance.
Send notifications to lecturers.

3.2.2 Functional Requirements

User Management:

The system must allow administrators to add, update, and delete student and lecturer profiles.
The system must provide user authentication and authorization features to ensure secure access.

Attendance Management:

The system must generate and store attendance records for each student.
The system must allow lecturers to mark attendance manually if needed.

Reporting:

The system print reports on student attendance.

Communication:

The system must provide a messaging system for communication between students, lecturers, and administrators.

3.2.3 Non Functional Requirements

Performance:

The system must handle simultaneous access by multiple users without significant performance degradation.

Attendance marking should be completed within 5 seconds per user.

Security:

The system must ensure secure user authentication and authorization.

The system must provide data encryption for sensitive information.

Usability:

The system must provide a user-friendly interface.

The system must be provide responsive to ensure access on various devices.

Maintainability:

The system must have well-documented and modular code to facilitate maintenance and updates.

The system must use updated version control for tracking changes.

4. Design

4.1 Introduction

The design of the Student - Lecture Management System (SLMS) focuses on creating an user-friendly web-based application using the Laravel framework and QR technology. The design process includes ER diagram, use case diagram, activity diagram to ensure the user experience.

4.2 ER diagram

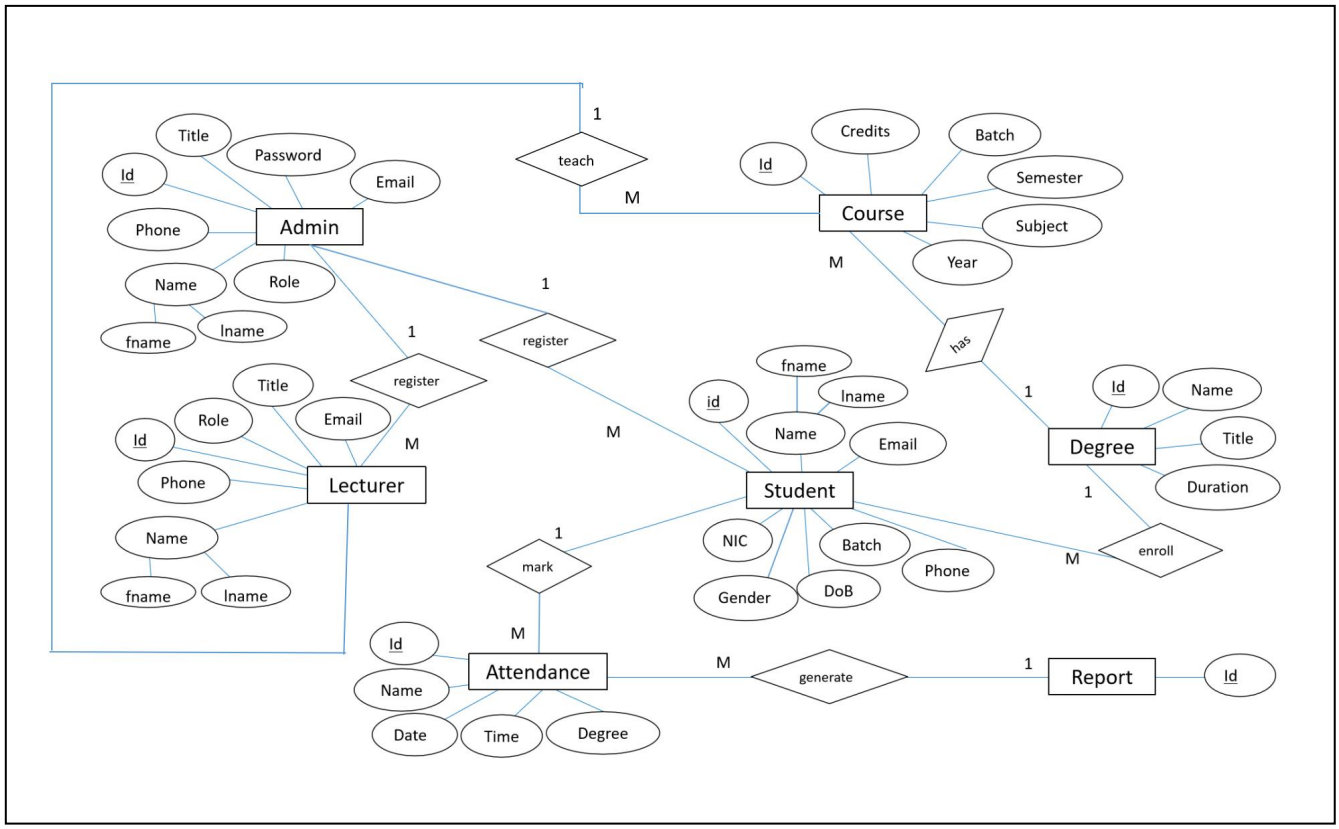


figure 4. 1 ER Diagram

4.3 Use Case diagram

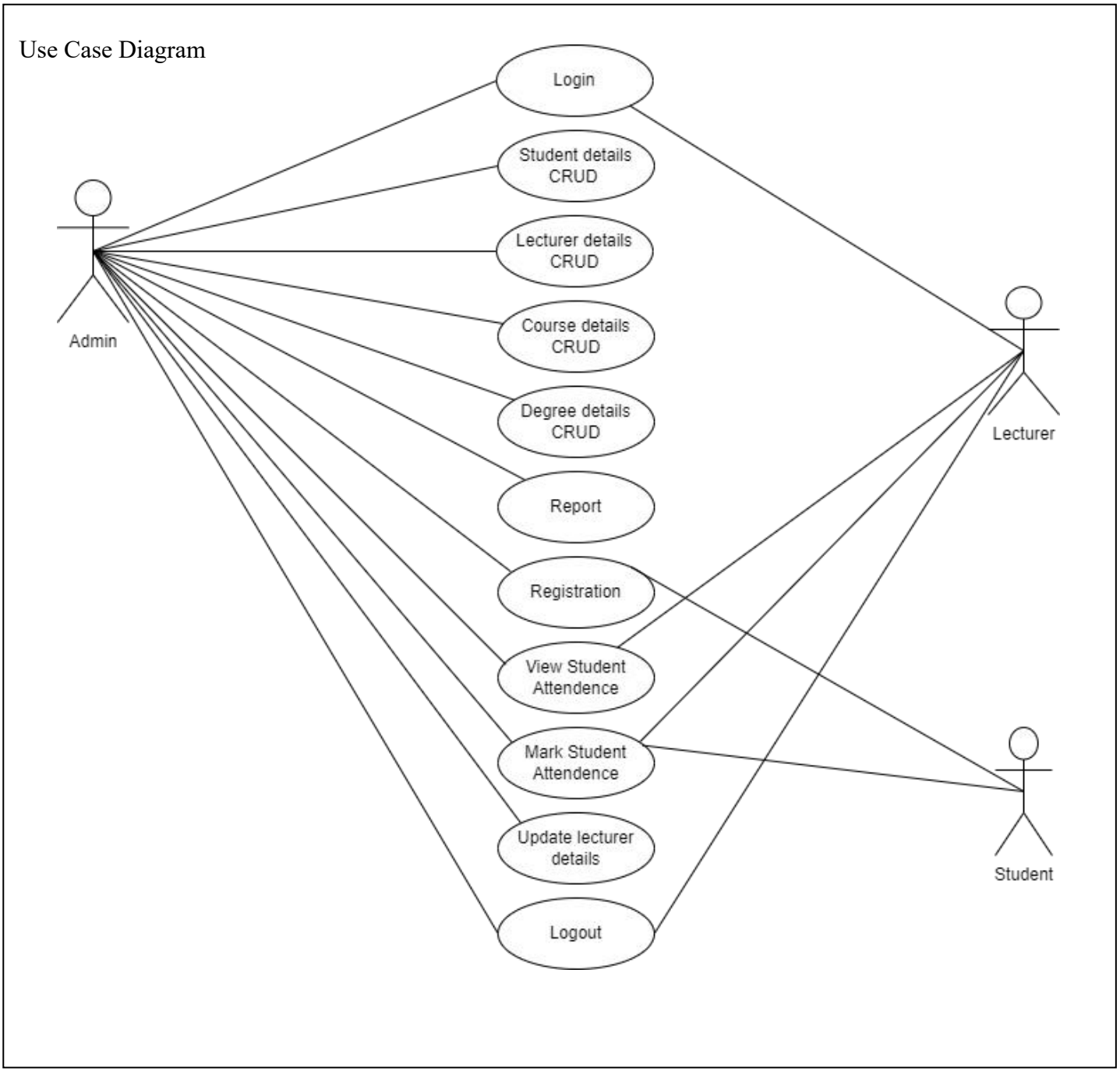


figure 4. 2 Use Case Diagram

4.4 Activity diagram

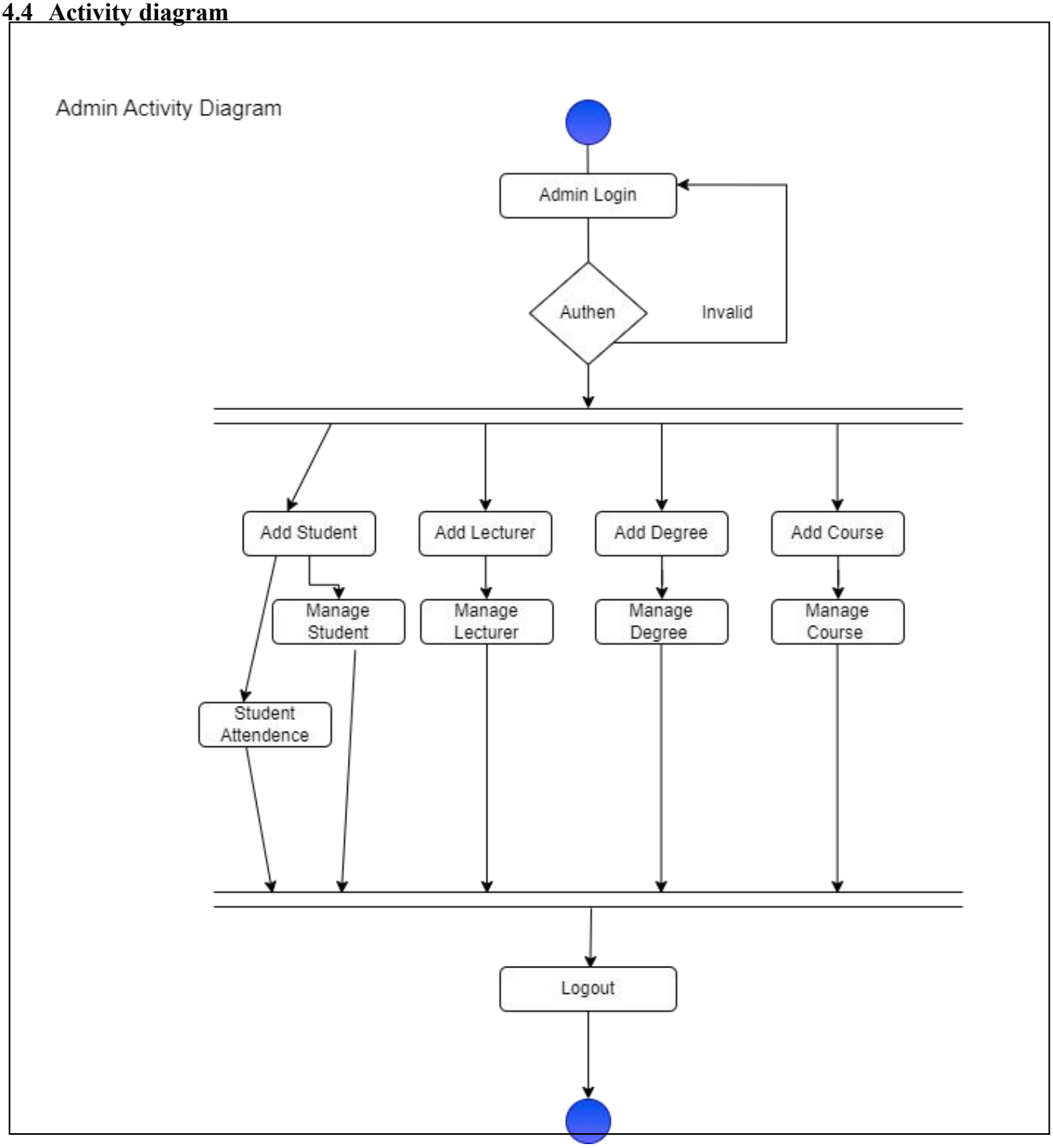


figure 4. 3 Activity Diagram

5. Implementation

5.1 Technology Adopted

Name of the component	Specification
Development	Web based system
Operating System	Windows
Framework	Laravel 11v
Language	Php 8v HTML CSS
Database	MySQL
Technology	QR

Table 5. 1 Technology Adopted

5.2 System Interface

Welcome Page

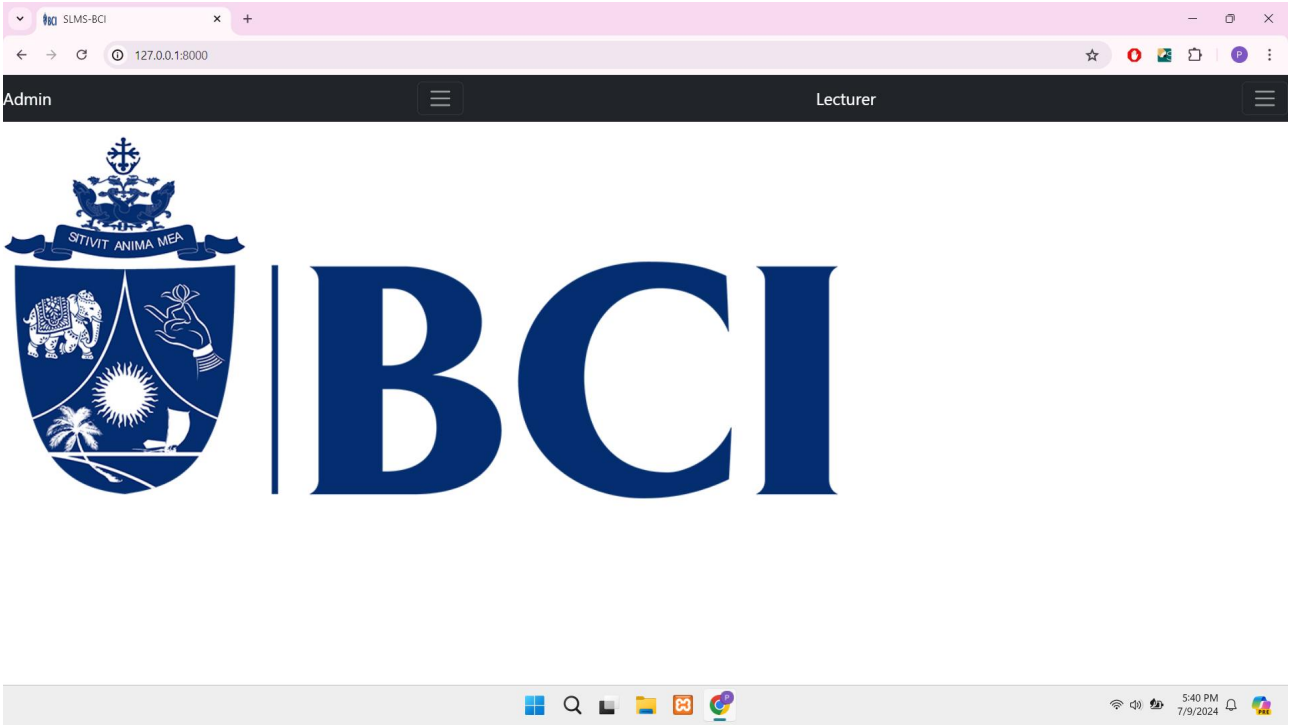


figure 5. 1 Welcome Page

Admin Page

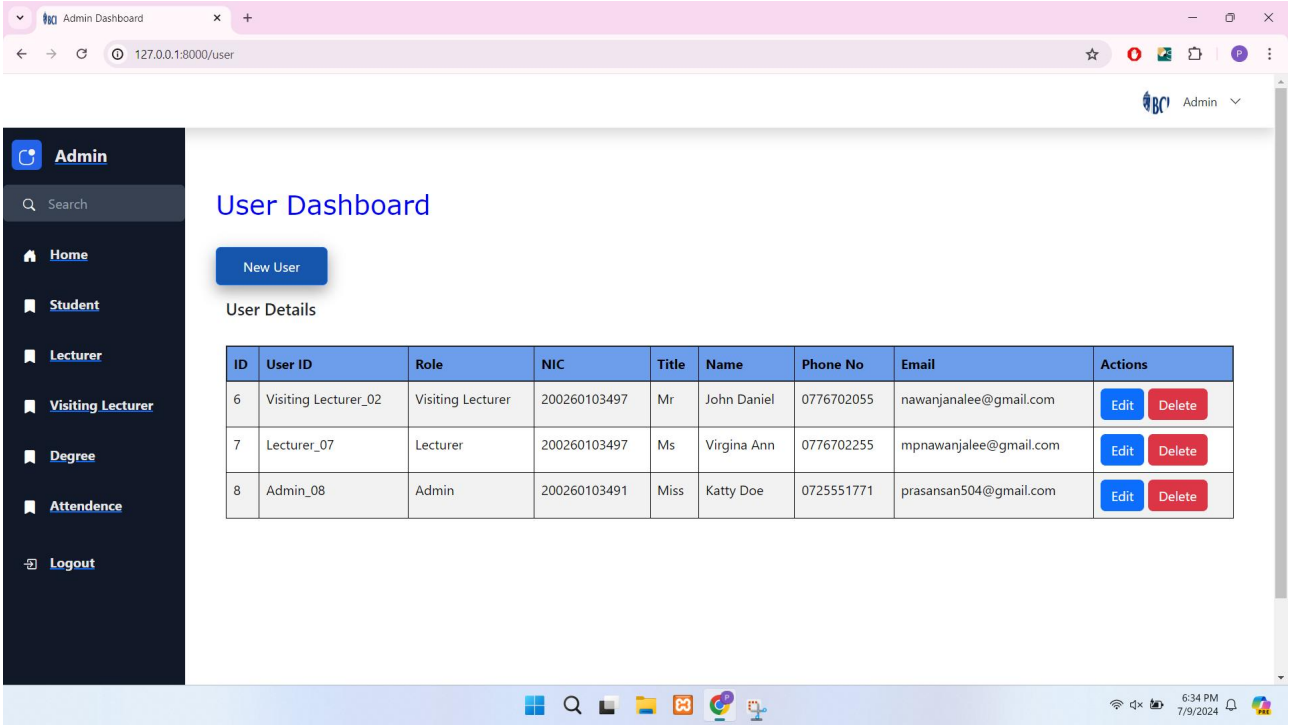


figure 5. 2 Admin Page

Lecturer Page

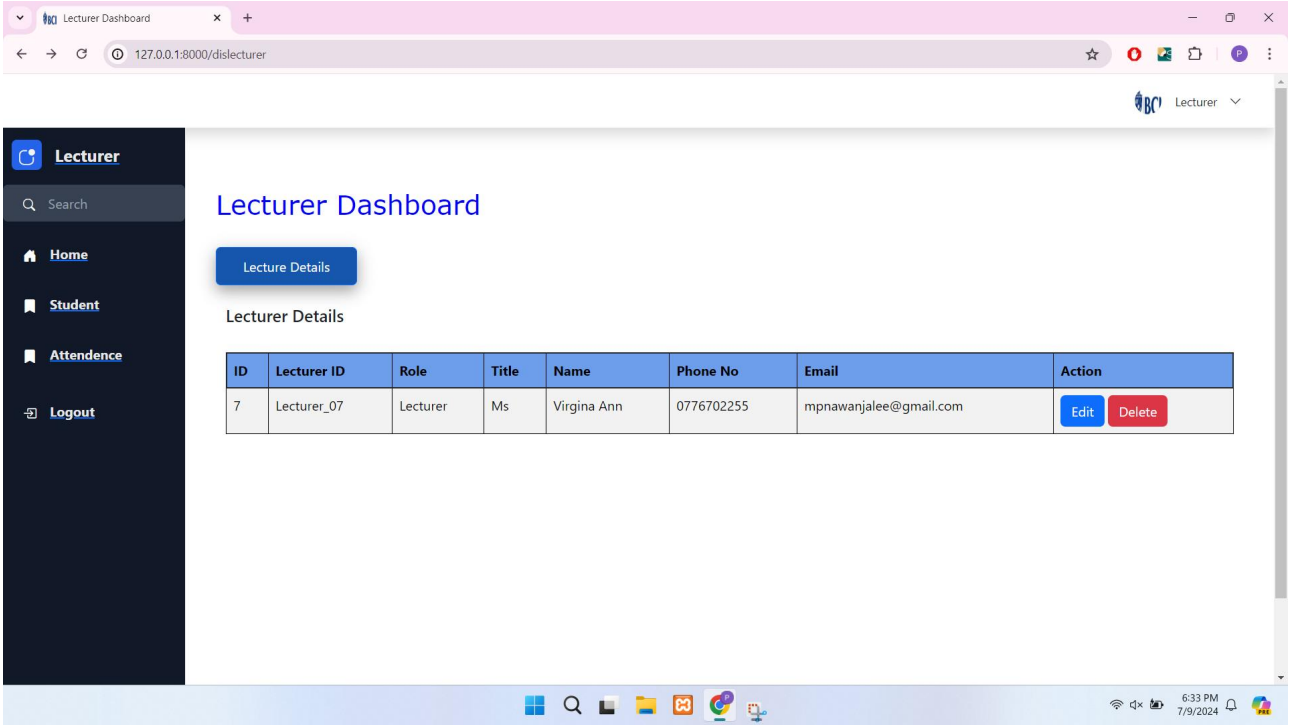


figure 5. 3 Lecturer Page

6. System Testing

6.1 Introduction

System testing for the Student-Lecture Management System (SLMS) ensures that all functions work as intended and meet the specified requirements. This phase involves executing test cases to validate the system's performance. The testing process includes unit testing, integration testing, system testing, and user acceptance testing. Each type of testing is designed to identify and deliver a high-quality final product.

6.2 Test cases

Test ID	Test	Test Inputs	Expected Output	Actual Output	Result (Pass/ Fail)	Description
Test 01	User Registration	Valid user details Valid email	Send email with User login credentials	Receive an email with User login credentials	Pass	Receiving an email after the successful registration
Test 02	User Login	User Id Password	User is red irected to the dash board	User successfully redirected to the dashboard after entering valid login credentials	Pass	Navigate to Login page and enter correct login details which send to the email. After click Login button it redirected to dashboard.

Test 03	Register Lecturer	Lecturer details	Register Lecturers	After the successful registration the userId and Password send to the lecturer email	Pass	Navigate to User registration page and select the role as Lecturer then after the registration it automatically send user details through email
Test 04	Register Student	Student details	Register the Student	After the successful registration a QR send to the student email	Pass	Navigate to Student registration page and after the registration it automatically create a student QR code and send it to through email
Test 05	Receive Email	Valid student details Valid email	Send email with Student QR code	Receive an email with Student QR code	Pass	Receiving an email after the successful registration
Test 06	Mark Attendance	Scan QR	Students can mark attendance using QR	Students can use the QR code which send to the email and they can mark the attendance	Pass	Students can scan the QR and mark the attendance
Test 07	Print Attendance Report	Mark the attendance	Ability to print the attendance after marking	Admin or Lecturer can print the attendance after marking	Pass	After student marking the attendance Admin or Lecturers can print those data
Test 08	Add New Degree	Correct degree name with duration and title	Add and view the degrees	Degree shows with their correct degree id	Pass	After adding a new degree it shows the all added degrees with their respective ids

Test 09	Add New Course	Correct course name with related data	Add new course	Added a new course and it shows under the respective degree	Pass	After giving a correct degree id and add a course then the course is displayed under the related degree
Test 10	Filter data in Attendance	Date Batch Degree name	Filtering the data	Shows only the filtered data	Pass	After giving the inputs it shows the filtered data in the view table
Test 11	Update Attendance	New values to be updated	Update the attendance table	It update the column cell that updated	Pass	After click the Edit button then can update a record in it in any special case
Test 12	Update Lecturer Record	New values to be updated	Update the lecturer record	Lecturer can update their own details	Pass	Lecturers have ability to update their own details in the lecturer page after login to the system
Test 13	Check Buttons	Click buttons	Respond to every action	Respond their related work / action and do the route	Pass	After click a button it response their work and route
Test 14	Get Student ID	Scan QR	Get Student_id automatically	Automatically get the Student id	Pass	After scan the QR code it shows the Student_id and related details
Test 15	Data Validation / Security	User credentials	Avoid unrelated data	Shows an error message	Pass	Prevent adding unrelated values and shows an error message
Test 16	Database	Table name Columns names	Store data	Store entered data and can retrieve when needed	Pass	Store in the correct table and columns after adding data in the system

Table 6. 1 Test Cases

7. Conclusion and Future Work

7.1 Introduction

The development of the Student-Lecture Management System (SLMS) successfully achieved the administrative challenges faced by BCI Campus, such as managing student and lecturer information and tracking attendance. The primary objective of the project was to create a centralized, automated, and user-friendly web-based application to replace the existing manual processes. This system fully realized through the implementation of the SLMS using the Laravel framework and QR technology.

7.2 Limitations

Initial Learning Curve:
Users may face some difficulties when transitioning from manual processes to the new system.
Suggested Solution:
Conduct comprehensive training sessions and provide user manuals to ensure a smooth transition.

Dependence on Internet Connectivity:
The web-based SLMS nature requires stable internet connectivity
Suggested Solution:
Implement offline functionality for critical features.

Scalability Concerns:
As the campus grows, the system may require enhancements to handle increased data volume and user load.
Suggested Solution:
Regularly review and update the system architecture and database design to ensure scalability and performance optimization.

Data Security Risks:

While the system incorporates security measures, there is always a risk of data breaches.

Suggested Solution:

Implement advanced security protocols, and stay updated with the latest security practices with technology.

7.3 Future Work

In future work,the system is discussed send reports to the coordinators of the relevant degree. The reports are generated once in a week.

Introduce a QR scanner to every class

QR code updated to include both student details and course details

8. References

[1] Snowball, J., & Mostert, M. (2010). Introducing a learning management system in a large first year class: Impact on lecturers and students. *South African Journal of Higher Education*, *24*(5), 818-831.

[2] Akintoye, K. A., O. T. Arogundade, and Oluwakemi Oke. *"Development of a Web-based Student–Lecturer Relationship Information System (E-Assessment)."* Development 25, no. 8 (2011).

[3] Masalha, Fadi, and Nael Hirzallah. "A students attendance system using QR code." *International Journal of Advanced Computer Science and Applications* 5, no. 3 (2014).

9. Appendix

9.1 Appendix A: Design Diagrams

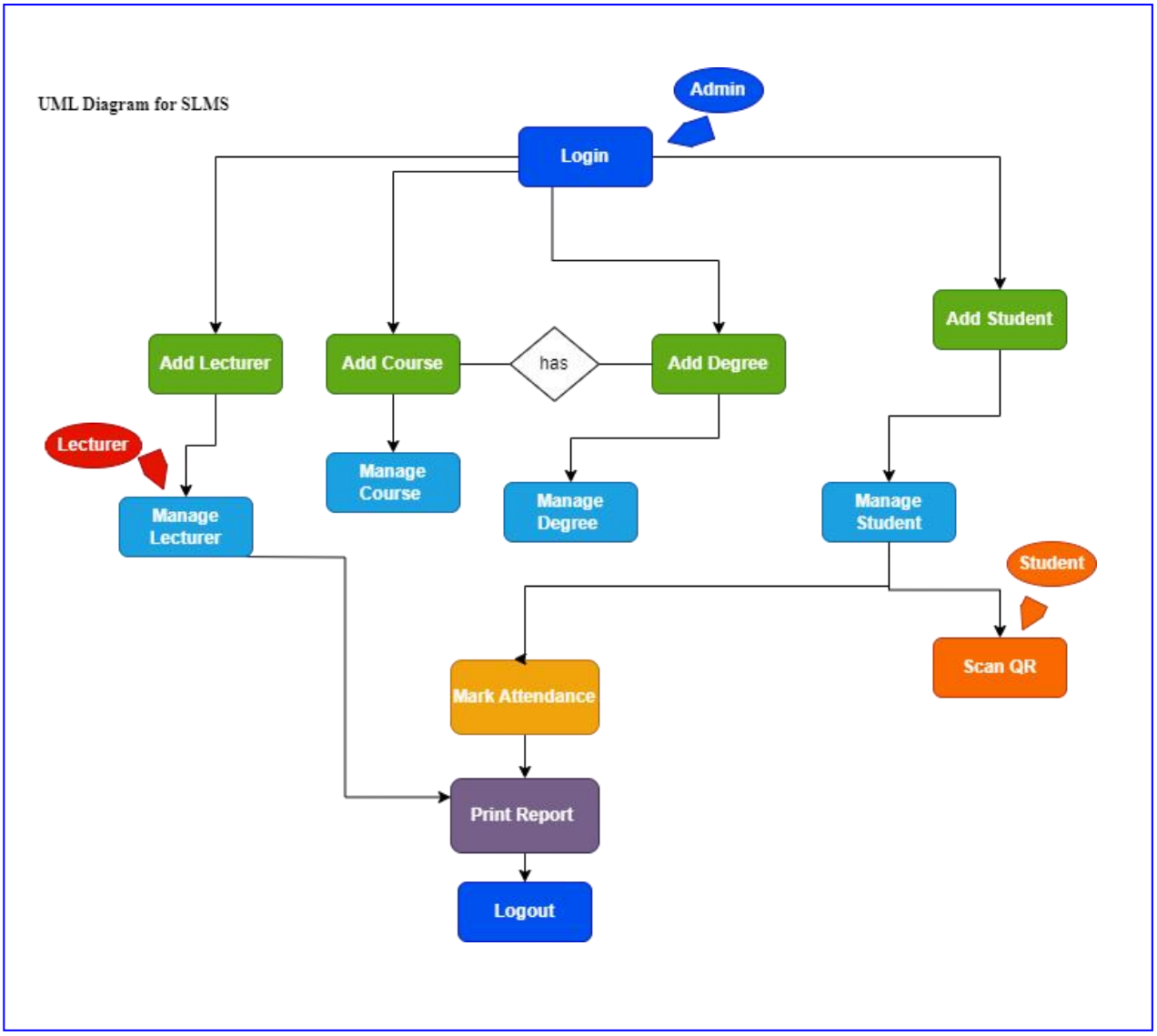


figure 9. 1 UML Diagram

9.2 Appendix B: Selected Code Listings

Registration Page

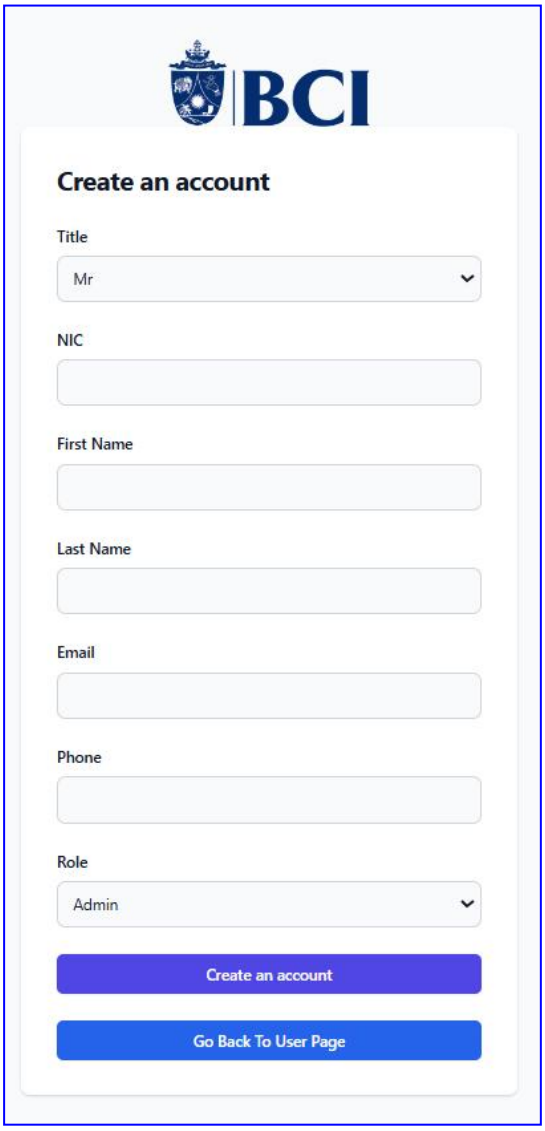


figure 9. 2 Register Page

register.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Register</title>
    <link rel="icon" href="logo.png">
    <link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.8.2/dist/alpine.min.js"></script>
    <script>
        function validateName() {
            var nameInput = document.getElementById('name');
            var isValid = /^[a-zA-Z\s]+$/.test(nameInput.value);
            if (!isValid) {
                alert('Name should only contain letters A-Z or a-z.');
                return false;
            }
            return true;
        }

        function validatePhone() {
            var phoneInput = document.getElementById('phone');
            var isValid = /^0?7(0|1|2|3|4|5|6|7|8|9)\d{7}$/.test(phoneInput.value);
            if (!isValid) {
                alert('Phone number should start with 070, 071, 072, 073, 074, 075, 076, 077, 078, or 079
followed by 7 digits.');
                return false;
            }
            return true;
        }

        function validateForm() {
            return validateName() && validatePhone();
        }
    </script>
</head>

<body>
    <section class="bg-gray-50 dark:bg-gray-900">
        <div class="flex flex-col items-center justify-center px-6 py-8 mx-auto md:h-screen lg:py-0">
            <div><br>
                <a href="/">
                    <img src="asset/images/logo.png" class="w-30 h-20 fill-current text-gray-500" />
                </a>
            </div>
            <div class="w-full bg-white rounded-lg shadow dark:border md:mt-0 sm:max-w-md xl:p-0 dark:bg-
gray-800 dark:border-gray-700">
                <div class="p-6 space-y-4 md:space-y-6 sm:p-8">
                    <h1 class="text-xl font-bold leading-tight tracking-tight text-gray-900 md:text-2xl
dark:text-white">
```

```
                        Create an account
                    </h1>
                    <form action="{{ route('register.save') }}" method="POST" onsubmit="return
validateForm()" class="space-y-4 md:space-y-6">
                        @csrf
                        <div>
                            <label for="title" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Title</label>
                            <select name="title" id="title" class="bg-gray-50 border border-gray-300 text-
gray-900 sm:text-sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5 dark:bg-
gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500
dark:focus:border-blue-500" required>
                                <option value="Mr">Mr</option>
                                <option value="Ms">Ms</option>
                                <option value="Miss">Miss</option>
                                <option value="Dr">Dr</option>
                                <option value="Prof">Prof</option>
                            </select>
                            @error('title')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <div>
                            <label for="nic" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">NIC</label>
                            <input type="text" name="nic" id="nic" class="bg-gray-50 border border-gray-300
text-gray-900 sm:text-sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5
dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500
dark:focus:border-blue-500" required>
                            @error('nic')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <div>
                            <label for="fname" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">First Name</label>
                            <input type="text" name="fname" id="fname" pattern="[A-Za-z]+" title="Name should
only contain letters A-Z or a-z." class="bg-gray-50 border border-gray-300 text-gray-900 sm:text-sm rounded-
lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600
dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500" required>
                            @error('fname')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <div>
                            <label for="lname" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Last Name</label>
                            <input type="text" name="lname" id="lname" pattern="[A-Za-z]+" title="Name should
only contain letters A-Z or a-z." class="bg-gray-50 border border-gray-300 text-gray-900 sm:text-sm rounded-
lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600
dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500" required>
                            @error('lname')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <div>
```

```
                            <label for="email" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Email</label>
                            <input type="email" name="email" id="email" class="bg-gray-50 border border-gray-
300 text-gray-900 sm:text-sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5
dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500
dark:focus:border-blue-500" required>
                            @error('email')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <div>
                            <label for="phone" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Phone</label>
                            <input type="tel" name="phone" id="phone"
pattern="^0?7(0|1|2|3|4|5|6|7|8|9)\d{7}$" title="Phone number should start with 070, 071, 072, 073, 074, 075,
076, 077, 078, or 079 followed by 7 digits." class="bg-gray-50 border border-gray-300 text-gray-900 sm:text-
sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5 dark:bg-gray-700
dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-
blue-500" required>
                            @error('phone')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <div>
                            <label for="role" class="block mb-2 text-sm font-medium text-gray-900 dark:text-
white">Role</label>
                            <select name="role" id="role" class="bg-gray-50 border border-gray-300 text-gray-
900 sm:text-sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-full p-2.5 dark:bg-gray-700
dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-
blue-500" required>
                                <option value="Admin">Admin</option>
                                <option value="Lecturer">Lecturer</option>
                                <option value="Coordinator">Coordinator</option>
                                <option value="Student">Student</option>
                                <option value="Visiting Lecturer">Visiting Lecturer</option>
                            </select>
                            @error('status')
                            <span class="text-red-600">{{ $message }}</span>
                            @enderror
                        </div>
                        <button type="submit" class="flex w-full justify-center rounded-md bg-indigo-600 px-3
py-1.5 text-sm font-semibold leading-6 text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-
visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600">Create an account</button>
                        <h2 class="text-sm font-light text-gray-500 dark:text-gray-400">
                        <a href="{{ route('user') }}" class="flex w-full justify-center rounded-md bg-blue-
600 px-3 py-1.5 text-sm font-semibold leading-6 text-white shadow-sm hover:bg-blue-500 focus-visible:outline
focus-visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-blue-600">Go Back To User
Page</a>
                        </h2>
                    </form>
                </div>
            </div>
        </div>
    </section>
</body>
```

```
</html>
```

Login Page

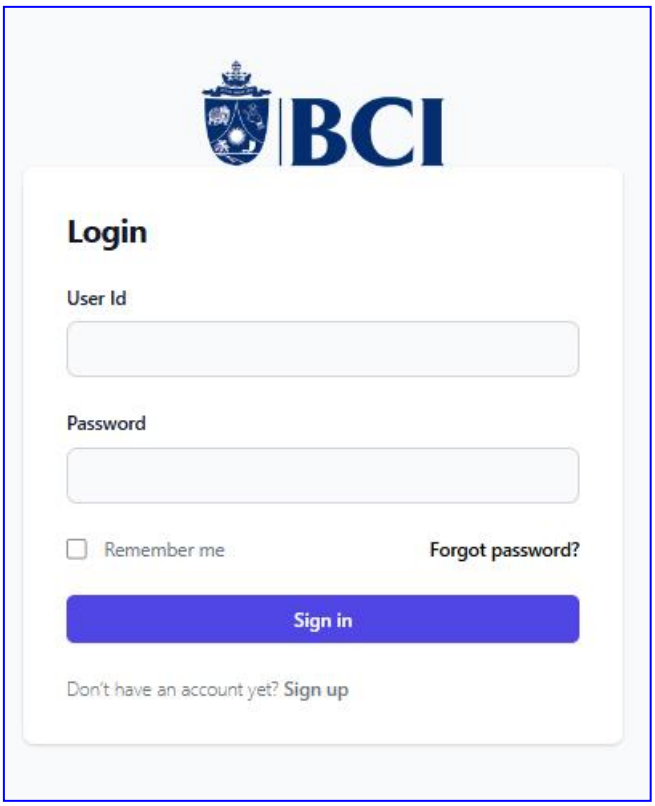


figure 9. 3 Login Page

login.blade.php

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <title>Login</title>
    <link rel="icon" href="{{ ('logo.png') }}">
    <link href="https://unpkg.com/tailwindcss@^2/dist/tailwind.min.css" rel="stylesheet" />
    <script src="https://cdn.jsdelivr.net/gh/alpinejs/alpine@v2.8.2/dist/alpine.min.js"></script>
</head>

<body>
```

```
    <section class="bg-gray-50 dark:bg-gray-900">
        <div class="flex flex-col items-center justify-center px-6 py-8 mx-auto md:h-screen lg:py-0">
        <div><br>
                <a href="/">
                    <img src="{{ ('asset/images/logo.png') }}"  class="w-30 h-20 fill-current text-gray-500"
/>
                </a>
            </div>
            <div class="w-full bg-white rounded-lg shadow dark:border md:mt-0 sm:max-w-md xl:p-0 dark:bg-
gray-800 dark:border-gray-700">
                <div class="p-6 space-y-4 md:space-y-6 sm:p-8">
                    <h1 class="text-xl font-bold leading-tight tracking-tight text-gray-900 md:text-2xl
dark:text-white">
                        Login
                    </h1>
                    <form class="space-y-4 md:space-y-6" method="post" action="{{ route('login.action') }}">
                        @csrf
                        @if ($errors->any())
                        <div class="bg-red-100 border border-red-400 text-red-700 px-4 py-3 rounded relative"
role="alert">
                            <strong class="font-bold">Error!</strong>
                            <ul>
                                @foreach ($errors->all() as $error)
                                <li><span class="block sm:inline">{{ $error }}</span></li>
                                @endforeach
                            </ul>
                            <span class="absolute top-0 bottom-0 right-0 px-4 py-3">
                                <svg class="fill-current h-6 w-6 text-red-500" role="button"
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 20 20">
                                    <title>Close</title>
                                    <path d="M14.348 14.849a1.2 1.2 0 0 1-1.697 0L10 11.819l-2.651 3.029a1.2
1.2 0 1 1-1.697-1.697l2.758-3.15-2.759-3.152a1.2 1.2 0 1 1 1.697-1.697L10 8.183l2.651-3.031a1.2 1.2 0 1 1
1.697 1.697l-2.758 3.152 2.758 3.15a1.2 1.2 0 0 1 0 1.698z" />
                                </svg>
                            </span>
                        </div>
                        @endif
                        <div>
                            <label for="user_id" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">User Id</label>
                            <input type="user_id" name="user_id" id="user_id" class="bg-gray-50 border
border-gray-300 text-gray-900 sm:text-sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-
full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-
blue-500 dark:focus:border-blue-500"required="">
                        </div>
                        <div>
                            <label for="password" class="block mb-2 text-sm font-medium text-gray-900
dark:text-white">Password</label>
                            <input type="password" name="password" id="password" class="bg-gray-50 border
border-gray-300 text-gray-900 sm:text-sm rounded-lg focus:ring-primary-600 focus:border-primary-600 block w-
full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-
blue-500 dark:focus:border-blue-500" required="">
                        </div>
                        <div class="flex items-center justify-between">
                            <div class="flex items-start">
                                <div class="flex items-center h-5">
```

```
                                    <input name="remember" id="remember" aria-describedby="remember"
type="checkbox" class="w-4 h-4 border border-gray-300 rounded bg-gray-50 focus:ring-3 focus:ring-primary-300
dark:bg-gray-700 dark:border-gray-600 dark:focus:ring-primary-600 dark:ring-offset-gray-800" required="">
                                </div>
                                <div class="ml-3 text-sm">
                                    <label for="remember" class="text-gray-500 dark:text-gray-300">Remember
me</label>
                                </div>
                            </div>
                            <a href="#" class="text-sm font-medium text-primary-600 hover:underline
dark:text-primary-500">Forgot password?</a>
                        </div>
                        <button type="submit" class="flex w-full justify-center rounded-md bg-indigo-600 px-3
py-1.5 text-sm font-semibold leading-6 text-white shadow-sm hover:bg-indigo-500 focus-visible:outline focus-
visible:outline-2 focus-visible:outline-offset-2 focus-visible:outline-indigo-600">Sign in</button>
                        <p class="text-sm font-light text-gray-500 dark:text-gray-400">
                            Don’t have an account yet? <a href="{{ route('register') }}" class="font-medium
text-primary-600 hover:underline dark:text-primary-500">Sign up</a>
                        </p>
                    </form>
                </div>
            </div>
        </div>
    </section>
</body>

</html>
```

Scan Student Attendance Page

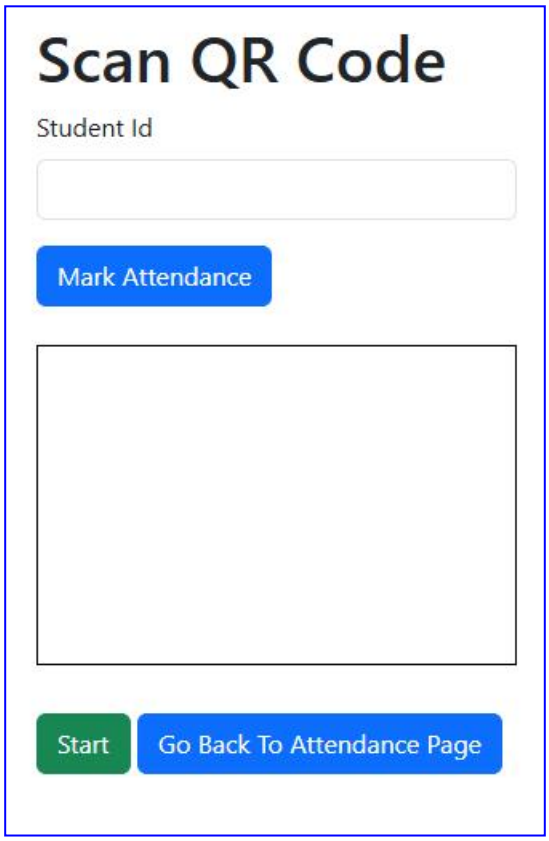


figure 9. 3 Scan QR Page

scan.blade.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Scan QR Code</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/@zxing/library@0.18.5"></script>
</head>
<body>
    <div class="container">
        <h1 class="mt-5">Scan QR Code</h1>
        <form method="POST" action="{{ route('attendances.mark') }}">
            @csrf
            <div class="mb-3" style = "max-width: 300px;">
                <label for="student_id" class="form-label">Student Id</label>
```

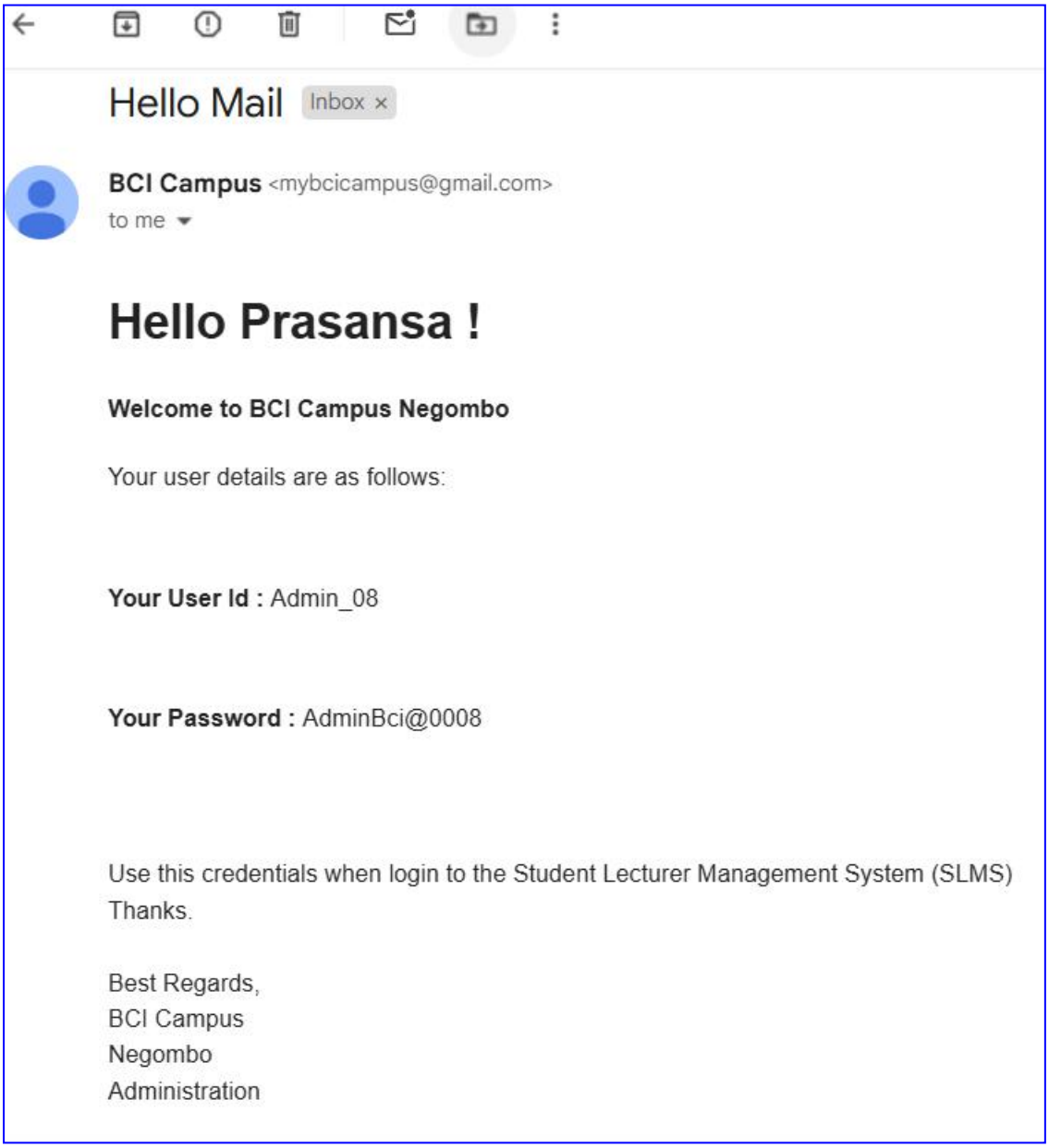
```
                <input type="text" class="form-control" id="student_id" name="student_id" required
pattern="[A-Za-z0-9_]+" title="Student Id should only contain letters, numbers, and underscores.">
            </div>
            <button type="submit" class="btn btn-primary">Mark Attendance</button>
        </form>
        <br>
        <div>
            <video id="video" width="300" height="200" style="border: 1px solid black"></video>
            <br><br>
            <button id="startButton" class="btn btn-success">Start</button>
            <a href="{{ route('attendance') }}" class="btn btn-primary">Go Back To Attendance Page</a>
        </div>
    </div>

    <script src="{{ asset('js/index.min.js') }}"></script>
    <script>
        window.addEventListener('load', function () {
            let selectedDeviceId;
            const codeReader = new ZXing.BrowserQRCodeReader();

            console.log('ZXing code reader initialized');

            document.getElementById('startButton').addEventListener('click', () => {
                codeReader.getVideoInputDevices()
                    .then((videoInputDevices) => {
                        selectedDeviceId = videoInputDevices[0].deviceId;
                        codeReader.decodeFromVideoDevice(selectedDeviceId, 'video', (result, err) => {
                            if (result) {
                                document.getElementById('student_id').value = result.text;
                                console.log(result);
                            }
                            if (err && !(err instanceof ZXing.NotFoundException)) {
                                console.error(err);
                            }
                        });
                        console.log(`Started continuous decode from camera with id ${selectedDeviceId}`);
                    })
                    .catch((err) => {
                        console.error(err);
                    });
            });
        });
    </script>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Send Email



Hello Mail Inbox ×

BCI Campus <mybcicampus@gmail.com>
to me

Hello Prasansa !

Welcome to BCI Campus Negombo

Your user details are as follows:

Your User Id : Admin_08

Your Password : AdminBci@0008

Use this credentials when login to the Student Lecturer Management System (SLMS)
Thanks.

Best Regards,
BCI Campus
Negombo
Administration

figure 9. 4 Email

hello.blade.php

```
<!DOCTYPE html>
<body>

<h1>  Hello {{ $fname }} ! </h1>
<h4> Welcome to BCI Campus Negombo </h4>

<p>Your user details are as follows:</p>
<br>
<p><b> Your User Id  : </b> {{ $user_id }}</p> <br>
<p><b> Your Password : </b> {{ $password }}</p>

<br><br>
<p> Use this credentials when login to the Student Lecturer Management System (SLMS)<br>
Thanks.
<br><br>
Best Regards, <br>
BCI Campus <br>
Negombo<br>
Administration  </p>

</body>
</html>
```

HeloMail.php (Model)

```
<?php

namespace App\Mail;

use Illuminate\Bus\Queueable;
use Illuminate\Mail\Mailable;
use Illuminate\Queue\SerializesModels;

class HelloMail extends Mailable
{
    use Queueable, SerializesModels;

    public $fname;
    public $user_id;
    public $password;

    public function __construct($fname, $user_id, $password)
    {
        $this->fname = $fname;
        $this->user_id = $user_id;
        $this->password = $password;
    }
```

```
    public function build()
    {
        return $this->view('auth.hello')
                    ->with([
                        'email' => $this->fname,
                        'user_id' => $this->user_id,
                        'password' => $this->password,
                    ]);
    }
}
```