**RANI CHANNAMMA UNIVERSITY, BELAGAVI**

****

**B.V.V. SANGHA’S**

**BASAVESHWARA SCIENCE COLLEGE,**

**BAGALKOTE - 587103**



**Department of Computer Science**

**CERTIFICATE**

This is to certify that **Shravan H. Jampanagoudra, Riteshsingh Killedar and Vaishakh Koti** has satisfactorily completed the Disseration work on **“ SMART ATTENDANCE”** for the partial fulfilment of Bachelor of Computer Application for the VI semester prescribed by the Rani Channamma University, Belagavi during the year 2023-24.

Internal Guide

Shri. Girish. M. K

Co-Ordinator

Smt. Akshata. Akkamaradi

HOD

Shri. P. S. Akkimaradi

Principal

Shri. S. M. Gaonkar

Reg. No: U15JP21S0215

Reg. No: U15JP21S0186

Reg. No: U15J021S0278

Examiners

1. 2

**Index**

|  |  |  |
| --- | --- | --- |
| **Sl No** | **Contents** | **Page No** |
| 1 | Abstract | 3 |
| 2 | Introduction   * System Introduction | 4 |
| 3 | Existing System | 5 |
| 4 | Proposed System | 6 |
| 5 | Modules   * User Management Module * Fingerprint Capture and Matching Module * Attendance Management Module * Session Management Module | 7-8 |
| 6 | System Specification   * Software Requirements * Hardware Requirements | 9 |
| 7 | ER Diagram | 10 |
| 8 | DFD Diagram | 11 |
| 9 | UseCase Diagram and UML Diagram | 12 |
| 10 | Testing and Results   * Functional Testing * Performance Testing * Usability Testing | 13 |
| 11 | Tools And Technologies | 14 |
| 12 | Demo Code | 15-38 |
| 13 | Screenshots | 39-40 |
| 14 | Conclusion | 41 |
| 15 | Future Enhancements | 42 |
| 16 | References | 43 |

1. **Abstract**

The "Smart Attendance" system is a biometric-based attendance tracking solution designed specifically for managing lab session attendance in educational institutions. This system leverages fingerprint recognition technology to provide an efficient, secure, and accurate method of recording student attendance.

The system consists of three types of users: administrators, staff, and students. Administrators have comprehensive control over the system, including the ability to manage subjects, staff, and students, as well as view and edit all attendance records. Staff members can monitor and manage attendance records specific to their sessions, while students can register and update their profiles independently.

The "Smart Attendance" system not only enhances the accuracy of attendance records but also reduces the administrative burden on educators, ensuring that the process is both streamlined and user-friendly. This project was developed as a standalone solution, with a focus on improving the overall efficiency of attendance management in lab settings.

1. **System Introduction**

The "Smart Attendance" system is designed specifically for educational labs, where accurate and timely attendance tracking is crucial. The system operates using a SecuGen fingerprint scanner to capture and verify student identities. With three distinct user roles—admin, staff, and students—the system ensures that only authorized individuals can access specific functionalities. Admins have comprehensive control over the system, allowing them to manage subjects, staff, and student records. Staff members can monitor attendance for their sessions, while students can easily register and participate in the attendance process using their fingerprints. This system enhances both the accuracy of attendance records and the overall efficiency of attendance management in lab settings.

1. **Existing System**

Traditional attendance systems in educational labs primarily rely on manual methods such as roll calls, sign-in sheets. These methods are often inefficient, time-consuming, and susceptible to errors or manipulation. Manual roll calls can lead to inaccuracies, especially in large classes, and require valuable class time to complete. Sign-in sheets may be prone to forgery or loss, while ID card scanning, although more automated, still depends on the physical presence of cards, which can be lost or forgotten by students.

Moreover, these systems lack real-time data processing and often require additional administrative effort to compile and maintain records. There is also a higher risk of data inconsistencies due to human error during data entry. Overall, traditional systems fall short in terms of accuracy, security, and efficiency, making them inadequate for modern educational environments, particularly in lab settings where precise tracking is essential.

1. **Proposed System**

The "Smart Attendance" system is introduced to overcome the limitations of traditional attendance tracking methods by providing a more efficient, accurate, and secure solution tailored specifically for lab environments. This system leverages biometric fingerprint recognition technology to ensure that attendance is recorded accurately and in real-time, reducing administrative workload and enhancing overall data integrity.

1. **Modules**
   1. **User Management Module:**
   * Purpose: This module is responsible for managing user accounts across the system, including administrators, staff, and students.
   * Features:
     + Registration: Allows admins and staff to register themselves and students. Students can also self-register.
     + Login: Provides authentication for users to access their respective dashboards.
     + Profile Management: Enables users to update their profile details, such as name, email, password, and other relevant information.
   1. **Fingerprint Capture and Matching Module:**

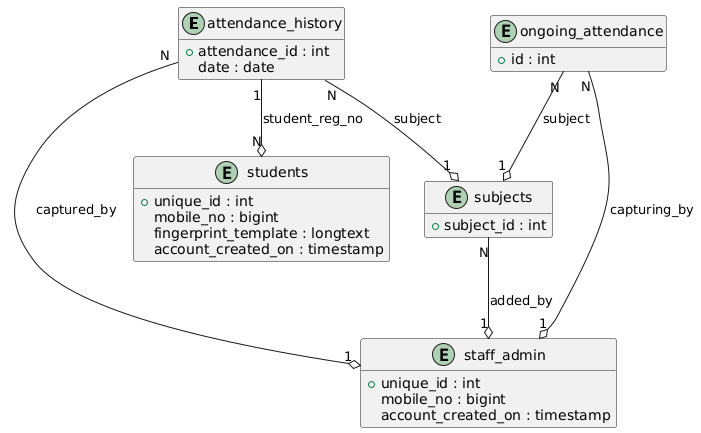
* Purpose: Central to the biometric attendance system, this module captures and verifies student identities using fingerprint data.
* Features:
  + - Fingerprint Enrollment: During registration, the system captures and stores the fingerprint template of each student.
    - Attendance Verification: During lab sessions, students' fingerprints are captured and matched against stored templates to mark attendance.
  1. **Attendance Management Module:**
* Purpose: Handles the creation, tracking, records.
* Features:
* Ongoing Attendance: Tracks active attendance sessions, recording which staff member is capturing attendance and which students have been marked present.
* Attendance History: Once a session is complete, attendance data is stored, allowing for future reference and reporting.
  1. **Session Management Module:**
* Purpose: Manages the lifecycle of lab attendance sessions.
* Features:
* Session Initiation: Staff or admins can start a new attendance session, selecting the relevant subject and batch.
* Session Monitoring: The module keeps track of ongoing sessions, ensuring that attendance is accurately captured.
* Session Finalization: At the end of a lab session, attendance records are finalized and stored in the system's database.

These modules work together to provide a comprehensive solution for managing lab attendance, ensuring accuracy and ease of use for all users.

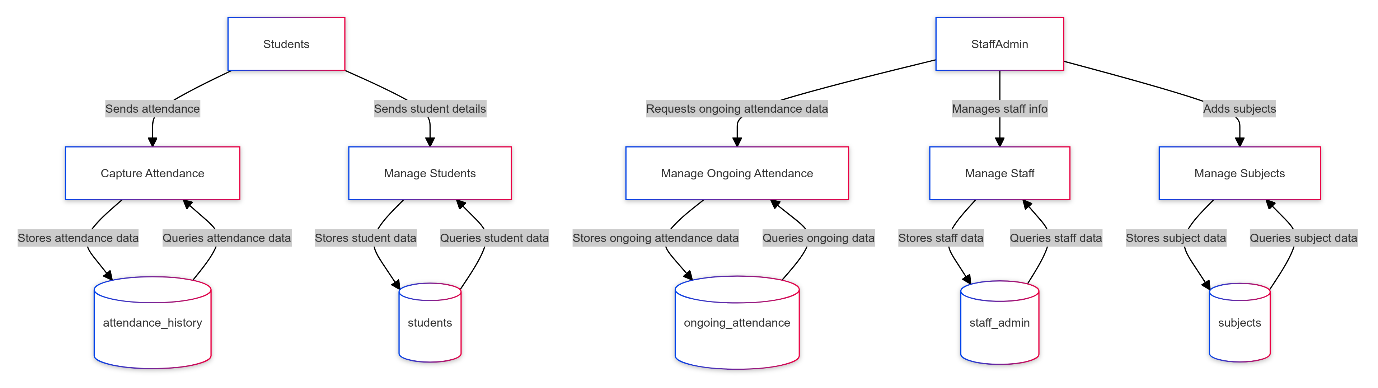
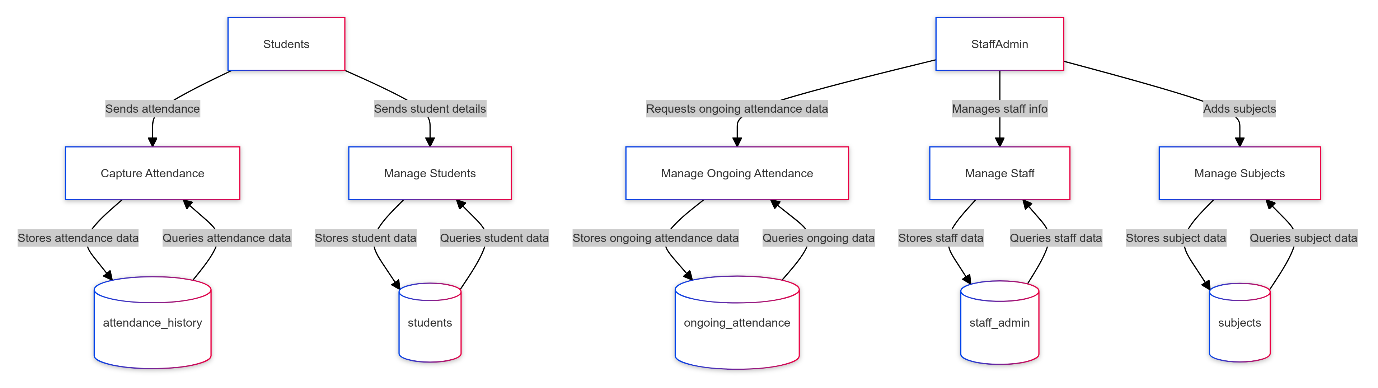
1. **System Specification**
   1. **Software Requirements:**

* Operating System: Windows 10 or later (for compatibility with SecuGen fingerprint scanner drivers)
* Web Server: XAMPP (Apache, MySQL, PHP)
* Database: MySQL 5.7 or later
* Programming Languages:
  + Frontend: HTML, CSS, JavaScript, Bootstrap
  + Backend: PHP
  + Database Management: MySQL
  + Web Browser: Google Chrome, Mozilla Firefox (latest versions)
  + SecuGen SDK: Required for fingerprint scanning and integration with the web application
  + Code Editor: Visual Studio Code, Sublime Text, or any other preferred IDE
  + Version Control: Git (optional, for version tracking and collaboration)
  1. **Hardware Requirements:**
* Processor: Intel Core i3 or higher
* RAM: 4 GB or more
* Hard Disk: Minimum 500 GB HDD or SSD
* Display: 1366x768 resolution or higher
* Fingerprint Scanner: SecuGen Hamster Pro 20 or any compatible fingerprint scanner
* Other Peripherals: Standard keyboard and mouse

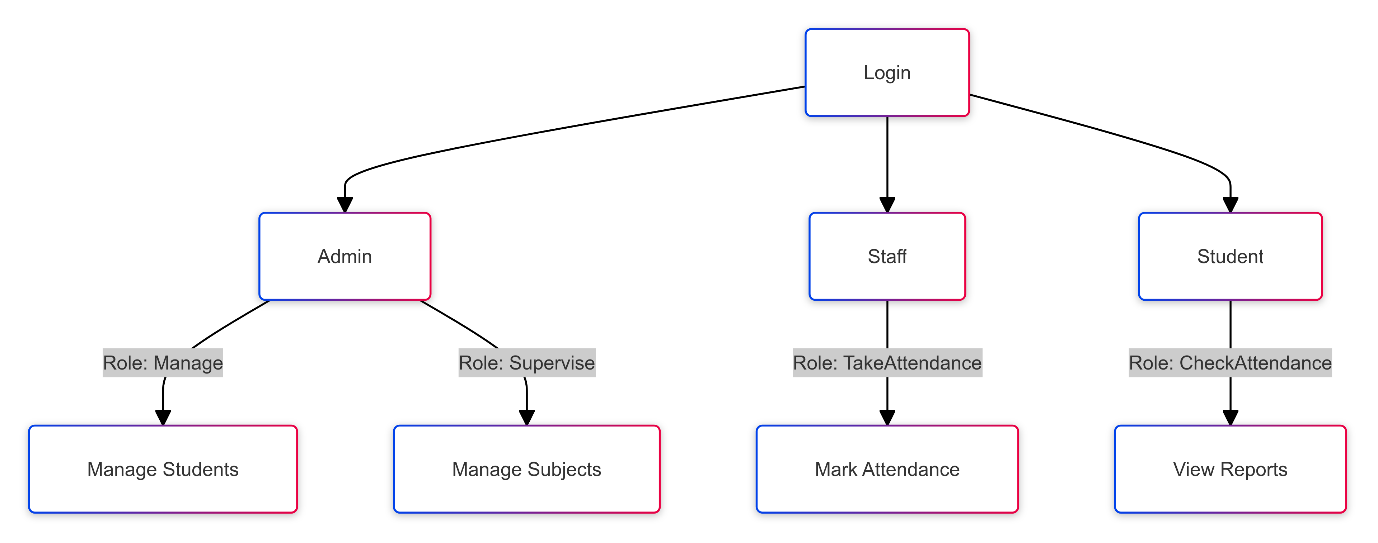
1. **ER(Entity Relationship) Diagram**

****

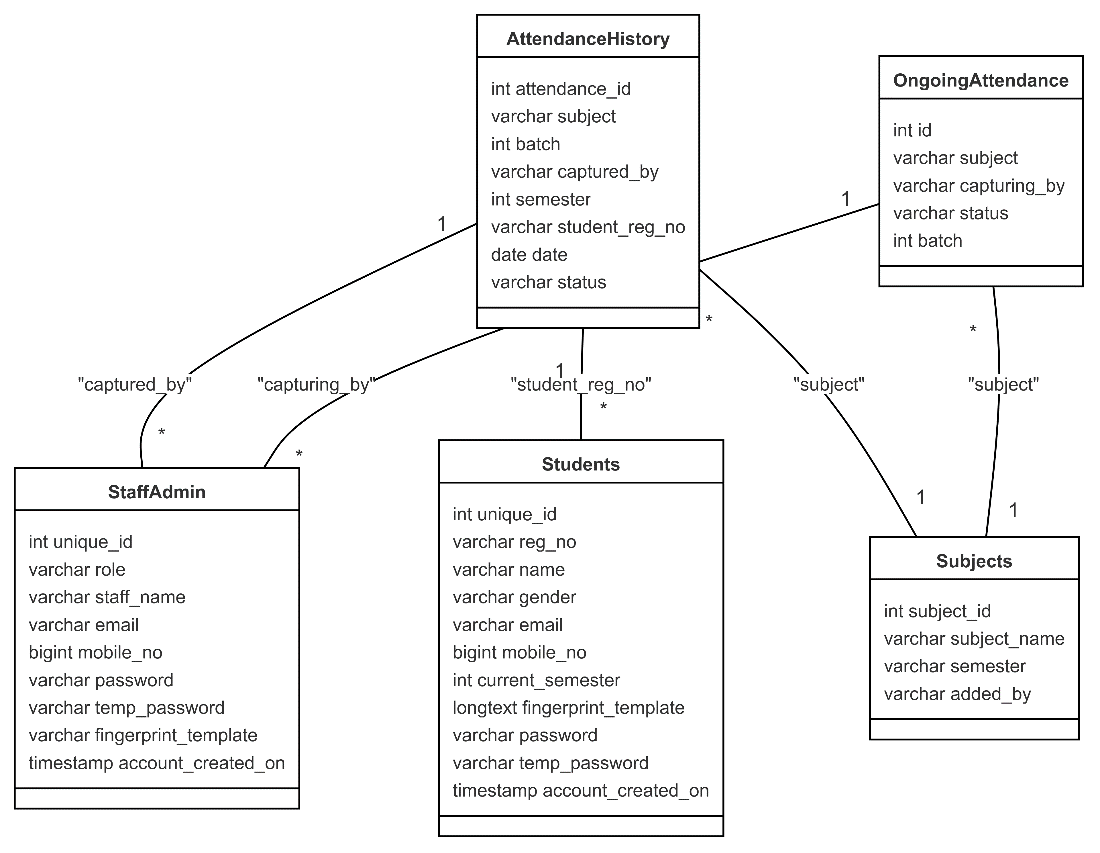
1. **DFD(Data Flow Diagram)**



1. **Use Case Diagram and UML Diagram**
   1. **Use Case Diagram**

****

* 1. **UML Diagram**

****

1. **Testing and Results**
   1. **Functional Testing**

* **User Registration**
* **Test Case**: Registration for Students, Staff, and Admins.
* **Results**: All types of user registrations (Student, Staff, Admin) were successfully completed. Temporary passwords were sent, and users were able to update their details correctly.
* **Attendance Capture**
* **Test Case**: Capturing attendance using fingerprint authentication.
* **Results**: Fingerprint matching was accurate in most tests. Attendance records were correctly updated in the attendance\_history table. The system handled session management (start, end, cancel) effectively.
* **Data Management**
* **Test Case**: Management of attendance data and subjects.
* **Results**: Attendance records and subject information were accurately managed. Admins were able to add, view, and edit subjects as required.
  1. **Performance Testing.**
* **Security Testing**
* **Test Case**: Encryption of sensitive data and access controls.
* **Results**: Sensitive data, including passwords and fingerprint templates, were encrypted. Access controls were properly enforced based on user roles.
  1. **Usability Testing**
* **User Interface**
* **Test Case**: Ease of use and navigation.
* **Results**: Users found the interface intuitive and easy to navigate. Feedback during interactions was clear and effective.

1. **Tools and Technologies**

* **Web Development Technologies**
* **HTML**: Used for structuring the web pages and defining the content on the front end.
* **CSS**: Applied for styling the web pages, including layout, colors, and fonts.
* **JavaScript**: Implemented for creating dynamic and interactive elements on the web pages.
* **Bootstrap**: Utilized for responsive design and to quickly develop the layout with pre-designed components.
* **Backend Technologies**
* **PHP**: Employed for server-side scripting to handle form submissions, manage user sessions, and interact with the database.
* **MySQL**: Used as the database management system for storing and managing user data, attendance records, and other application data.
* **Database**
* **MySQL**: The relational database system used for creating and managing the project's database, including tables like students, staff\_admin, attendance\_history, ongoing\_attendance, and subjects.
* **Fingerprint Authentication**
* **SecuGen Fingerprint Scanner**: Hardware used to capture and scan fingerprints for authentication purposes.
* **SecuGen Driver and WebAPI Software**: Provides the necessary drivers and API to interface with the SecuGen fingerprint scanner, allowing the integration of fingerprint authentication into the web application.
* **Development Tools**
* **Visual Studio Code**: A popular code editor used for writing and managing the project’s code.
* **phpMyAdmin**: A web-based tool for managing MySQL databases, used for database administration and management tasks.

1. **Demo Code**

**User Registration Code**

<?php include 'config/config.php'; ?>

<?php include 'common/header.php';

if(isset($\_SESSION['Student'])){

header("Location:index.php");

exit();

}

?>

<?php

use PHPMailer\PHPMailer\PHPmailer;

use PHPMailer\PHPMailer\Exception;

require 'phpmailer/src/Exception.php';

require 'phpmailer/src/PHPMailer.php';

require 'phpmailer/src/SMTP.php';

if(isset($\_POST['capture'])){

$email = strtolower($\_POST['student\_email']);

$reg\_no = strtoupper($\_POST['registration\_no']);

$fingerprint\_template = $\_POST['fingerprint\_template'];

if($fingerprint\_template == ''){

$message = 'Please Scan Fingerprint';

}

else if($email == '' OR $reg\_no == ''){

$message = 'e-Mail or Registration fields are required';

}

else{

$check\_student = $connection->prepare("SELECT \* FROM `students` WHERE reg\_no = ? OR email = ?");

$check\_student->execute([$reg\_no, $email]);

if($check\_student->rowCount() > 0)

{

$message = "Student with Registation or E-Mail already exists.";

}

else{

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$charactersLength = strlen($characters);

$randomString = '';

for ($i = 0; $i < 6; $i++) {

$randomString .= $characters[random\_int(0, $charactersLength - 1)];

}

$email\_message = '

<h3>Smart Attendance</h3>

<h5">Student Registration</h5>

<p class="fs-3">

Dear student,<br>

Your password is : <b>'.$randomString.'</b> use thsi password to login.<br><br>

Team Smart Attendance.

</p>';

$mail = new PHPMailer(true);

try{

$mail->isSMTP();

$mail->Host = 'smtp.gmail.com';

$mail->SMTPAuth = true;

$mail->Username = 'iamwilliamsmith07@gmail.com';

$mail->Password = 'bplcpnotbaavazdy';

$mail->SMTPSecure = 'ssl';

$mail->Port = 465;

$mail->setFrom('iamwilliamsmith07@gmail.com');

$mail->addAddress($email);

$mail->isHTML(true);

$mail->Subject = 'Welcome to Smart Attendance System';

$mail->Body = $email\_message;

$mail->send();

}

catch(Exception $e){

echo $e;

}

$insert\_student = $connection->prepare("INSERT INTO `students`(reg\_no, email, fingerprint\_template, password, temp\_password) VALUES (?, ?, ?, ?, ?)");

$insert\_student->execute([$reg\_no, $email, $fingerprint\_template, $randomString, $randomString]);

if($insert\_student){

$message = 'Password has been sent to your e-Mail id. Please login and update your academic details.<br><br>

<b>Note : </b>Please connect to the same Wi-Fi network and access the update page after login.';

}

else{

$message = 'Password has been sent through email but failed to insert it into database.<br><br>

<b>Note : </b>Please scan and register yourself again.';

}

}

}

}

?>

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Message from Smart-Attendance</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<p><?= $message; ?></p>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Ok</button>

</div>

</div>

</div>

</div>

<?php

?>

<div class="mt-3 col-md-3 p-5 p-lg-2 mx-auto">

<div class="container-sm bg-white rounded shadow-sm border mx-auto ">

<div class="p-4 pb-1">

<h3 class="fw-bold text-center mb-0 text-black">Smart<b class="bg-green ms-2 p-1 pt-0 text-dark rounded">Attendance</b></h3>

<h6 class="text-center text-black mb-1 mt-2 p-0 ">Student Registration</h6>

</div>

<form action="" id="reg\_form" method="post" class="p-3" style="font-size: 0.9rem;">

<?php

if(isset($message)){

?>

<script>

var myModal = new bootstrap.Modal(document.getElementById("exampleModal"));

myModal.show();

</script>

<?php

}

?>

<div class="row">

<div class="col-md mb-3">

<label class="form-label text-black mb-1">Email</label>

<input type="email" id="student\_email" placeholder="iamstudent@example.com" class="form-control form-control-sm" name="student\_email" required>

</div>

</div>

<div class="row">

<div class="col-md mb-3">

<label class="form-label text-black mb-1">University Reg. No</label>

<input type="text" id="reg\_no" placeholder="UXXXXXXSXXXX" class="form-control form-control-sm" name="registration\_no" required>

</div>

</div>

<div class="row">

<div class="col-sm-12">

<p class="p-1 m-0" id="status"><b>Scan Status</b> : Not Scanned.</p>

<input type="hidden" name="fingerprint\_template" id="fingerprint\_template" required>

<input type="hidden" name="capture">

</div>

</div>

<div class="row">

<div class="col-sm-12">

<a onclick="captureFP()" class="btn mb-3 btn-warning col-sm-12 text-black" >Scan now</a>

</div>

</div>

<div class="row">

<div class="col-sm-12">

<p class="p-1 m-0"><b>Note</b> : Please check you mail and submit all details.</p>

</div>

</div>

</form>

</div>

</div>

<?php include 'common/footer.php';?>

**Adding New Subject and Starting Lab Session**

<?php include '../config/config.php'; ?>

<?php include 'common/header.php';?>

<?php

if(isset($\_SESSION['Staff']) || isset($\_SESSION['Admin'])){

if(isset($\_SESSION['Staff'])){

$logged\_in\_staff = $\_SESSION['Staff'];

}

else{

$logged\_in\_staff = $\_SESSION['Admin'];

}

}

else{

header('Location:../index.php');

}

$select\_logged\_in\_user = $connection->prepare("SELECT \* FROM `staff\_admin` WHERE unique\_id = ?");

$select\_logged\_in\_user->execute([$logged\_in\_staff]);

$logged\_in\_data = $select\_logged\_in\_user->fetch(PDO::FETCH\_ASSOC);

if(isset($\_POST['addSub'])){

$semester = $\_POST['semester'];

$subject = $\_POST['subject\_name'];

$insert\_subject = $connection->prepare("INSERT INTO `subjects` (subject\_name, semester, added\_by) VALUES (?, ?, ?)");

if($insert\_subject->execute([$subject, $semester, $logged\_in\_data['staff\_name']])){

$message = $subject.' '. 'Subject Added Successfully for Semester'.' '. $semester;

}

}

if(isset($\_POST['start\_session'])){

$subject\_id = strtoupper($\_POST['session\_subject']);

$capturing\_by = $\_POST['logged\_user'];

$batch = $\_POST['batch'];

$status = 'Capturing';

$insert\_into\_ongoing = $connection->prepare("INSERT INTO `ongoing\_attendance` (subject, capturing\_by, status, batch) VALUES (?, ?, ?, ?)");

$insert\_into\_ongoing->execute([$subject\_id, $capturing\_by, $status, $batch]);

if($insert\_into\_ongoing){

header('Location:captureAttendance.php');

}

}

?>

<div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="exampleModalLabel">Message from Smart-Attendance</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body">

<p><?= $message; ?></p>

</div>

<div class="modal-footer">

<button type="button" class="btn m-0 btn-secondary" data-bs-dismiss="modal">Ok</button>

</div>

</div>

</div>

</div>

<?php

?>

<?php

if(isset($message)){

?>

<script>

var myModal = new bootstrap.Modal(document.getElementById("exampleModal"));

myModal.show();

</script>

<?php

}

?>

<div class="container text-center mt-3 justify-content-center">

<div class="mx-auto col-sm-3">

<h3 class="text-center">Add new</h3>

<form method="post" id="addSub" style="display:none;">

<div class="row">

<div class="col-md mb-3">

<label>Semester</label>

<select name="semester" class="form-select form-select-sm p-1 shadow-sm bg-white rounded" required>

<option value="0">Select Semester</option>

<option value="1">I</option>

<option value="2">II</option>

<option value="3">III</option>

<option value="4">IV</option>

<option value="5">V</option>

<option value="6">VI</option>

</select>

</div>

<div class="col-md mb-3">

<label>Subject Name</label>

<input id="subjectadd" type="text" placeholder="Subject Name" name="subject\_name" class="form-control mb-2 form-control-sm" required>

</select>

</div>

</div>

<input id="addBtn" type="submit" name="addSub" class="btn mb-2 btn-sm btn-success">

</form>

<button onclick="showAddSubject()" type="submit" id="changeBtn" class="btn mb-2 btn-sm btn-primary">Add Subject</button>

</div>

</div>

<div class="container text-center mt-3 justify-content-center">

<div class="mx-auto col-sm-3">

<h3 class="text-center">Enter Session Details</h3>

<div class="row">

<form method="post" onclick="this.form.submit()">

<div class="col-md mb-3">

<label>Semester</label>

<select name="session\_semester" onchange='this.form.submit()' class="form-select form-select-sm p-1 shadow-sm bg-white rounded" required>

<option value="0">Select Semester</option>

<option value="1">I</option>

<option value="2">II</option>

<option value="3">III</option>

<option value="4">IV</option>

<option value="5">V</option>

<option value="6">VI</option>

</select>

</div>

</form>

<form method="post" action="">

<div class="col-md mb-3">

<label>Select Subject</label>

<select name="session\_subject"class="form-select form-select-sm p-1 shadow-sm bg-white rounded" required>

<option value="0">Select Subject</option>

<?php

$select\_subject = $connection->prepare("SELECT \* FROM `subjects` WHERE semester = ?");

$select\_subject->execute([$\_POST['session\_semester']]);

while($select\_sub = $select\_subject->fetch(PDO::FETCH\_ASSOC)){

?>

<option value="<?= $select\_sub['subject\_name'];?>"><?= $select\_sub['subject\_name'];?></option>

<?php

}

?>

</select>

</div>

<div class="col-md mb-3">

<label>Select Batch</label>

<select name="batch" class="form-select form-select-sm p-1 shadow-sm bg-white rounded" required>

<option value="0">Select Batch</option>

<option value="1">Batch 1(Day 1)</option>

<option value="2">Batch 2(Day 1)</option>

<option value="3">Batch 3(Day 2)</option>

<option value="4">Batch 4(Day 2)</option>

<option value="5">Batch 5(Day 3)</option>

<option value="6">Batch 6(Day 3)</option>

</select>

</div>

<input type="hidden" name="logged\_user" value="<?= $logged\_in\_data['staff\_name']; ?>">

<input id="addBtn" type="submit" name="start\_session" class="btn mb-2 btn-sm btn-warning" value="Start Session">

</form>

</div>

</div>

</div>

<script>

function showAddSubject(){

var add\_sub = document.getElementById("addSub");

var changeBtn =document.getElementById('changeBtn');

if(add\_sub.style.display == 'block'){

add\_sub.style.display = 'none';

changeBtn.setAttribute('class', 'btn mb-2 btn-sm btn-primary');

changeBtn.textContent = 'Add Subject';

}

else{

add\_sub.style.display = 'block';

changeBtn.setAttribute('class', 'btn mb-2 btn-sm btn-danger');

changeBtn.textContent = 'Cancel';

}

}

</script>

<?php include 'common/footer.php';?>

**Capturing Attendance and Matching Fingerprints**

<?php include '../config/config.php'; ?>

<?php include 'common/header.php'; ?>

<?php

date\_default\_timezone\_set("Asia/Calcutta");

$today\_date = date('Y-m-d');

if(isset($\_SESSION['Staff']) || isset($\_SESSION['Admin'])){

    if(isset($\_SESSION['Staff'])){

        $logged\_in\_staff = $\_SESSION['Staff'];

    }

    else{

        $logged\_in\_staff = $\_SESSION['Admin'];

    }

}

else{

    header('Location:../index.php');

}

$select\_logged\_in\_user = $connection->prepare("SELECT \* FROM `staff\_admin` WHERE unique\_id = ?");

$select\_logged\_in\_user->execute([$logged\_in\_staff]);

$logged\_in\_data = $select\_logged\_in\_user->fetch(PDO::FETCH\_ASSOC);

$ongoing\_attendance = $connection->prepare("SELECT \* FROM `ongoing\_attendance` WHERE capturing\_by = ? AND status = 'Capturing' ");

$ongoing\_attendance->execute([$logged\_in\_data['staff\_name']]);

$ongoing\_attendance\_data = $ongoing\_attendance->fetch(PDO::FETCH\_ASSOC);

$batch = $ongoing\_attendance\_data['batch'];

    $select\_subject = $connection->prepare("SELECT \* FROM `subjects` WHERE subject\_name = ?");

    $select\_subject->execute([$ongoing\_attendance\_data['subject']]);

    $select\_sub = $select\_subject->fetch(PDO::FETCH\_ASSOC);

    $session\_semester = $select\_sub['semester'];

    $select\_from\_students\_temp = $connection->prepare("SELECT \* FROM `students` WHERE current\_semester = ? LIMIT 1");

    $select\_from\_students\_temp->execute([$session\_semester]);

    if($select\_from\_students\_temp->rowCount() <=0){

        echo "<script type='text/javascript'>alert('No Students present or registered for this semester yet.,,');window.location.href='sessionDetails.php';</script>";

    }

    $fetched\_from\_students = $select\_from\_students\_temp->fetch(PDO::FETCH\_ASSOC);

    $university\_reg\_no\_temp = mb\_strimwidth($fetched\_from\_students['reg\_no'], 0, 9, '');

    if(isset($\_POST['registration'])){

        $no = $\_POST['registration'];

        $real\_reg\_no = $university\_reg\_no\_temp.$no;

        $select\_real\_student = $connection->prepare("SELECT \* FROM `students` WHERE reg\_no = ? AND current\_semester = ? LIMIT 1");

        $select\_real\_student->execute([$real\_reg\_no, $session\_semester]);

        $real\_student\_data = $select\_real\_student->fetch(PDO::FETCH\_ASSOC);

    }

    if(isset($\_POST['cancel'])){

        $delete\_ongoing = $connection->prepare("DELETE FROM `ongoing\_attendance` WHERE capturing\_by = ? AND status = 'Capturing'");

        $delete\_ongoing->execute([$logged\_in\_data['staff\_name']]);

        $delete\_history = $connection->prepare("DELETE FROM `attendance\_history` WHERE captured\_by = ? AND status = 'Not Confirmed'");

        $delete\_history->execute([$logged\_in\_data['staff\_name']]);

        if($delete\_ongoing){

            echo "<script type='text/javascript'>alert('Capturing attendance data deleted');window.location.href='sessionDetails.php';</script>";

        }

    }

    if(isset($\_POST['stop'])){

        $delete\_ongoing = $connection->prepare("UPDATE`ongoing\_attendance` WHERE capturing\_by = ? AND status = 'Captured' WHERE status = 'Capturing'");

        $delete\_ongoing->execute([$logged\_in\_data['staff\_name']]);

        $delete\_history = $connection->prepare("UPDATE `attendance\_history` SET status = 'Confirmed' WHERE captured\_by = ? AND status = 'Not Confirmed'");

        $delete\_history->execute([$logged\_in\_data['staff\_name']]);

        if($delete\_ongoing){

            echo "<script type='text/javascript'>alert('Capturing attendance Completed');window.location.href='sessionDetails.php';</script>";

        }

    }

?>

        <div class="p-3" id="capture">

            <div class="container mx-auto rounded  bg-white align-items-center p-2">

                <h3 class="fw-bold mb-0 text-center text-black text-bold mb-3">Capturing Attendance</h3>

                <h6 class="fw-bold mb-0 text-start text-black text-bold mt-3 mb-0 p-0 ms-4">Attendance Details</h6>

                <div class="row justify-content-start ps-4 pt-3">

                    <div class="col-sm-4">

                        <p class="text-bold text-start text-bold"><b>Subject</b>&emsp;&emsp; &nbsp;: <?= $select\_sub['subject\_name']; ?></p>

                        <p class="text-bold text-start text-bold"><b>Date</b>&emsp; &emsp; &ensp;&nbsp;&nbsp; : <?= $today\_date; ?></p>

                        <p class="text-bold text-start text-bold"><b>Time from</b>&nbsp; &nbsp; :  <?= date("h:iA"); ?></p>

                        <p class="text-bold text-start text-bold"><b>Time to</b>&emsp;&emsp;&nbsp; : ----- </p>

                    </div>

                    <div class="col-sm-4">

                        <p class="text-bold text-start text-bold"><b>Semester</b>&ensp;&ensp;&nbsp; : Semester <?= $session\_semester; ?></p>

                        <p class="text-bold text-start text-bold"><b>Admin</b>&emsp;&emsp;&ensp; : <?= $logged\_in\_data['staff\_name']; ?></p>

                        <p class="text-bold text-start text-bold"><b>Mode</b>&emsp; &ensp; &ensp;&nbsp; : Scanning Fingerprint</p>

                        <form method="post">

                            <input type="submit" name="cancel" class="btn btn-outline-danger" value="Cancel Attendance">

                            <input type="submit" name="stop" class="btn btn-outline-success" value="End Session">

                        </form>

                    </div>

                    <div class="col-sm-4">

                        <h3 class="fw-bold text-center mb-0 text-black">Give Attendance</h3>

                    <form action="" method="POST" class="p-3" style="font-size: 0.9rem;">

                        <div class="row">

                            <div class="col-sm-6 mb-3">

                                <label class="form-label text-black mb-1">Registration No.</label>

                                <input type="number" id="reg\_no" value="" onchange="this.form.submit()" placeholder="000(Last Three Digits)" class="form-control form-control-sm" name="registration" required>

                            </div>

                        </div>

                    </form>

                        <div class="row">

                            <div class="col-sm-12">

                                <button type="submit" onclick="captureFinger()" class="btn mb-3 btn-warning text-black" value="" name="give\_attendance">Capture Finger</button>

                            </div>

                        </div>

                    </div>

                </div>

            </div>

        </div>

        <form method="post" id="returned\_value\_form">

            <input type="hidden" name="returned\_value" id="returned\_value" value="">

        </form>

<script>

    function captureFinger(){

        var template\_1 = '';

        template\_1 = '<?= $real\_student\_data['fingerprint\_template'];?>';

        var uri = "https://localhost:8443/SGIFPCapture";

        var xmlhttp = new XMLHttpRequest();

        xmlhttp.onreadystatechange = function(){

            if (xmlhttp.readyState == 4 && xmlhttp.status == 200){

                fingerpring\_object = JSON.parse(xmlhttp.responseText);

                if(fingerpring\_object.ErrorCode == 0){

                    var template\_2 = fingerpring\_object.TemplateBase64;

                    matchScore(template\_1, template\_2);

                }

                else if(fingerpring\_object.ErrorCode == 55){

                    alert('Connect the scanner and try again...');

                }

                else{

                    alert('Check wether driver installed correctly...');

                }

            }

             else if (xmlhttp.status == 404) {

                console.log('Error page not found');

            }

        }

        xmlhttp.open("POST", uri, true);

        xmlhttp.send();

        xmlhttp.onerror = function () {

            console.log("failed");

        }

    }

    function matchScore(template\_1, template\_2) {

        if (template\_1 == "" || template\_2 == "") {

            alert("Please scan two fingers to verify!!");

            return;

        }

        var uri = "https://localhost:8443/SGIMatchScore";

        var xmlhttp = new XMLHttpRequest();

        xmlhttp.onreadystatechange = function () {

            if (xmlhttp.readyState == 4 && xmlhttp.status == 200){

                fpobject = JSON.parse(xmlhttp.responseText);

                if(fpobject.ErrorCode == 0){

                    if(fpobject.MatchingScore > 70){

                        var returning\_form =document.getElementById('returned\_value\_form');

                        var returning\_data =document.getElementById('returned\_value');

                        returning\_data.value = template\_1;

                        returning\_form.submit();

                    }

                    else{

                        console.log(fpobject.MatchingScore);

                    }

                }

            }

            else if (xmlhttp.status == 404){

                alert("Page not FOund!!");

            }

        }

        var params = "template1=" + encodeURIComponent(template\_1);

        params += "&template2=" + encodeURIComponent(template\_2);

        xmlhttp.open("POST", uri, false);

        xmlhttp.send(params);

    }

</script>

<?php

if(isset($\_POST['returned\_value'])){

    $returned\_value = $\_POST['returned\_value'];

    $presented\_student = $connection->prepare("SELECT \* FROM `students` WHERE fingerprint\_template = ?");

    $presented\_student->execute([$returned\_value]);

    $presented\_student\_date = $presented\_student->fetch(PDO::FETCH\_ASSOC);

    $insert\_attendance = $connection->prepare("INSERT INTO `attendance\_history` (subject, batch, captured\_by, semester, student\_reg\_no, date) VALUES (?, ?, ?, ?, ?, ?)");

    $insert\_attendance->execute([$ongoing\_attendance\_data['subject'], $batch, $logged\_in\_data['staff\_name'], $session\_semester, $presented\_student\_date['reg\_no'], $today\_date]);

}

?>

<?php

$display\_attendance = $connection->prepare("SELECT \* FROM `attendance\_history` WHERE batch = ? AND captured\_by = ? AND date = ? AND status = 'Not Confirmed'");

$display\_attendance->execute([$batch, $logged\_in\_data['staff\_name'], $today\_date]);

if($display\_attendance->rowCount() < 0){

    echo 'No sutdents present';

}

else{

    ?>

    <div class="container">

        <div class="row justify-content-start">

    <?php

    while($attendance\_data = $display\_attendance->fetch(PDO::FETCH\_ASSOC))

    {

        $select\_student\_to\_display = $connection->prepare("SELECT \* FROM `students` WHERE reg\_no = ?");

        $select\_student\_to\_display->execute([$attendance\_data['student\_reg\_no']]);

        while($student\_display\_data = $select\_student\_to\_display->fetch(PDO::FETCH\_ASSOC)){

            ?>

                <div class="card bg-warning me-2 mt-1" style="width: 8.6rem;">

                    <div class="p-2">

                        <img class="rounded-circle mx-auto" width="100px" height="100px" src="../profile.jpg">

                        <p class="text-bold text-start p-0 m-0 text-bold" style="font-size: 0.8rem;"><b><?= $student\_display\_data['reg\_no'] ;?></b></p>

                        <p class="text-bold text-start text-bold" style="font-size: 0.8rem;"><?= $student\_display\_data['name'] ;?></p>

                    </div>

                </div>

            <?php

        }

    }

    ?>

        </div>

    </div>

    <?php

}

?>

<?php include 'common/footer.php';?>

**Changing Password**

<?php include 'config/config.php'; ?>

<?php include 'common/header.php';

if(isset($\_SESSION['Student']) ){

    $select\_logged\_in\_user = $connection->prepare("SELECT \* FROM `students` WHERE reg\_no = ?");

    $select\_logged\_in\_user->execute([$\_SESSION['Student']]);

    $logged\_in\_data = $select\_logged\_in\_user->fetch(PDO::FETCH\_ASSOC);

    $table = 'students';

    $access = 'reg\_no';

    $user = $logged\_in\_data['reg\_no'];

}

else if(isset($\_SESSION['Staff']) || isset($\_SESSION['Admin'])){

    if(isset($\_SESSION['Staff'])){

        $logged\_in\_staff = $\_SESSION['Staff'];

    }

    else{

        $logged\_in\_staff = $\_SESSION['Admin'];

    }

    $select\_logged\_in\_user = $connection->prepare("SELECT \* FROM `staff\_admin` WHERE unique\_id = ?");

    $select\_logged\_in\_user->execute([$logged\_in\_staff]);

    $logged\_in\_data = $select\_logged\_in\_user->fetch(PDO::FETCH\_ASSOC);

    $table = 'staff\_admin';

    $access = 'unique\_id';

    $user = $logged\_in\_data['unique\_id'];

}

else{

    header('Location:login.php');

}

?>

<?php

if(isset($\_POST['update\_password'])){

    $password = $\_POST['student\_password'];

    $update\_student = $connection->prepare("UPDATE `$table` set password = ? WHERE $access = ?");

    if($update\_student->execute([$password, $user])){

        $message = 'Password updated successfully.';

        if($\_SESSION['Student']){

            header('Location:index.php');

        }

        else{

            header('Location:staff/index.php');

        }

    }

}

?>

    <div class="modal fade" id="exampleModal" tabindex="-1" aria-labelledby="exampleModalLabel" aria-hidden="true">

        <div class="modal-dialog">

            <div class="modal-content">

                <div class="modal-header">

                    <h5 class="modal-title" id="exampleModalLabel">Message from Smart-Attendance</h5>

                    <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

                </div>

                <div class="modal-body">

                    <p><?= $message; ?></p>

                </div>

                <div class="modal-footer">

                    <button type="button" class="btn m-0 btn-secondary" data-bs-dismiss="modal">Ok</button>

                </div>

            </div>

        </div>

    </div>

<?php

?>

<div class="mt-3  col-md-3 p-5 p-lg-2 mx-auto">

    <div class="container-sm bg-white rounded shadow-sm border mx-auto ">

        <div class="p-4 pb-1">

            <h3 class="fw-bold text-center mb-0 text-black">Change Password</h3>

        </div>

        <form action="" id="reg\_form" method="post" class="p-3" style="font-size: 0.9rem;">

        <?php

        if(isset($message)){

            ?>

            <script>

            var myModal = new bootstrap.Modal(document.getElementById("exampleModal"));

            myModal.show();

            </script>

            <?php

            }

        ?>

            <div class="row">

                <div class="col-md mb-3">

                    <label class="form-label text-black mb-1">Password</label>

                    <input type="password" id="student\_password" placeholder="New Password" onchange="onChange()" class="form-control form-control-sm" name="student\_password" required>

                </div>

            </div>

            <div class="row">

                <div class="col-md mb-3">

                    <label class="form-label text-black mb-1">Confirm Password</label>

                    <input type="text" id="confirm\_password" placeholder="Confirm Password" onchange="onChange()" class="form-control form-control-sm" name="confirm\_password" required>

                </div>

            </div>

            <div class="row">

                <div class="col-sm-12">

                    <input type="submit" class="btn mb-3 btn-warning  col-sm-12 text-black" value="Update" name="update\_password">

                </div>

            </div>

        </form>

    </div>

</div>

<script>

    function onChange() {

  const password = document.querySelector('input[name=student\_password]');

  const confirm = document.querySelector('input[name=confirm\_password]');

  if (confirm.value === password.value) {

    confirm.setCustomValidity('');

  } else {

    confirm.setCustomValidity('Passwords do not match');

  }

}

</script>

<?php include 'common/footer.php';?>

**JavaScript to Capture Fingerprint**

    function captureFP(){

        var uri = "https://localhost:8443/SGIFPCapture";

        var xmlhttp = new XMLHttpRequest();

        xmlhttp.onreadystatechange = function(){

            if (xmlhttp.readyState == 4 && xmlhttp.status == 200){

                fingerpring\_object = JSON.parse(xmlhttp.responseText);

                if(fingerpring\_object.ErrorCode == 0){

                    var fingerprint\_template = fingerpring\_object.TemplateBase64;

                    document.getElementById('status').innerHTML = "<b>Scan Status</b> : Scanned successfully.";

                    document.getElementById('fingerprint\_template').value = fingerprint\_template;

                    var form =document.getElementById('reg\_form');

                    form.submit();

                }

                else if(fingerpring\_object.ErrorCode == 55){

                    alert('Connect the scanner and try again...');

                }

                else{

                    alert('Check wether driver installed correctly...');

                }

                }

            else if (xmlhttp.status == 404) {

                console.log('Error page not found');

                }

            }

            xmlhttp.open("POST", uri, true);

            xmlhttp.send();

            xmlhttp.onerror = function () {

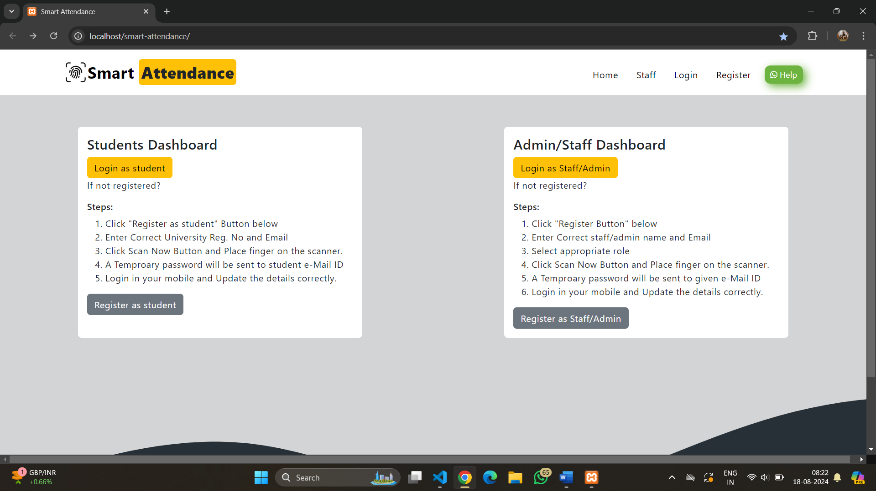
                console.log("failed");

            }

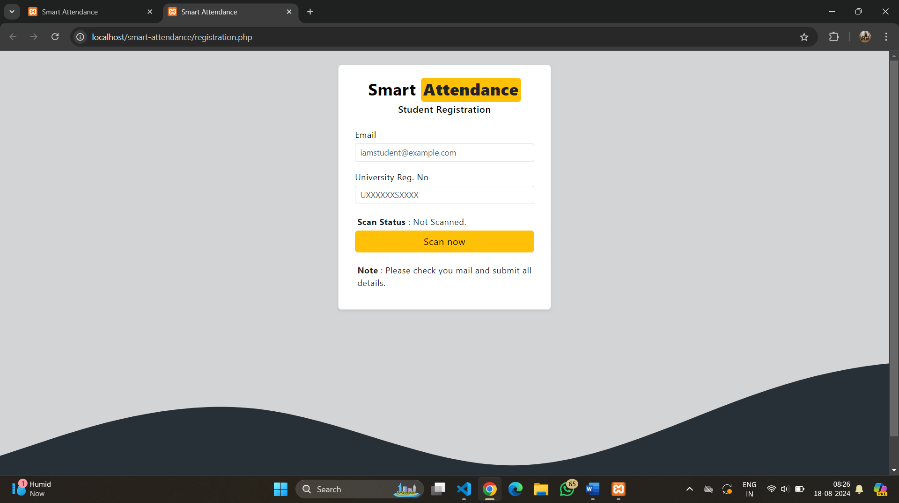
    }

1. **Screenshots**

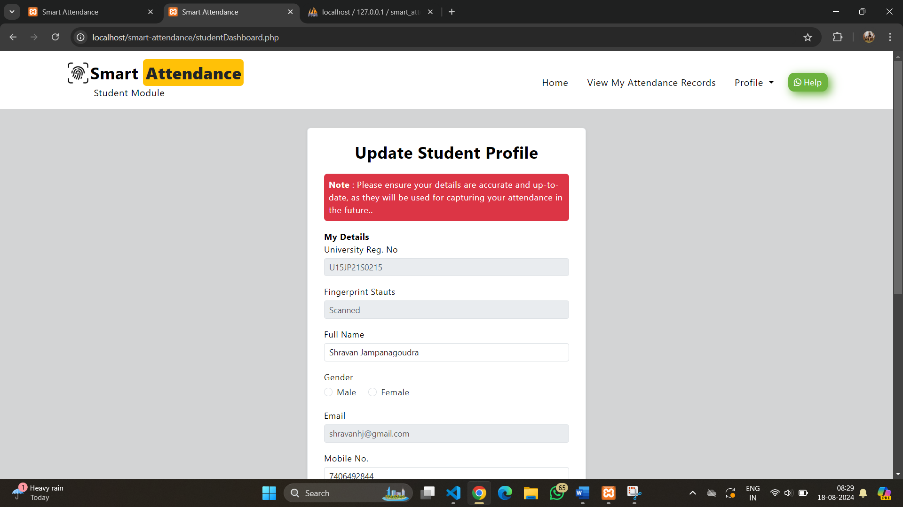
**Picture 13.1 Index Page**

****

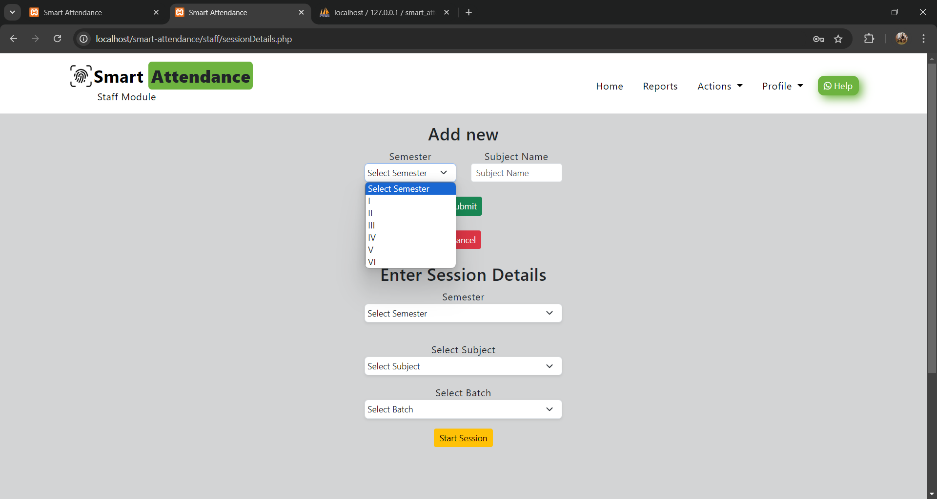
**Picture 13.2 Student Registration Page**

****

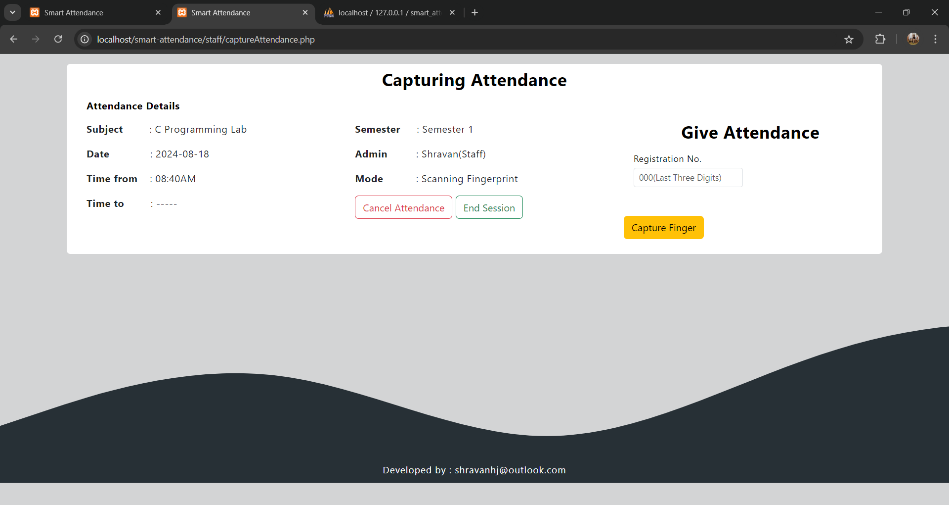
**Picture 13.3 Student Dashboard/Update Page**

****

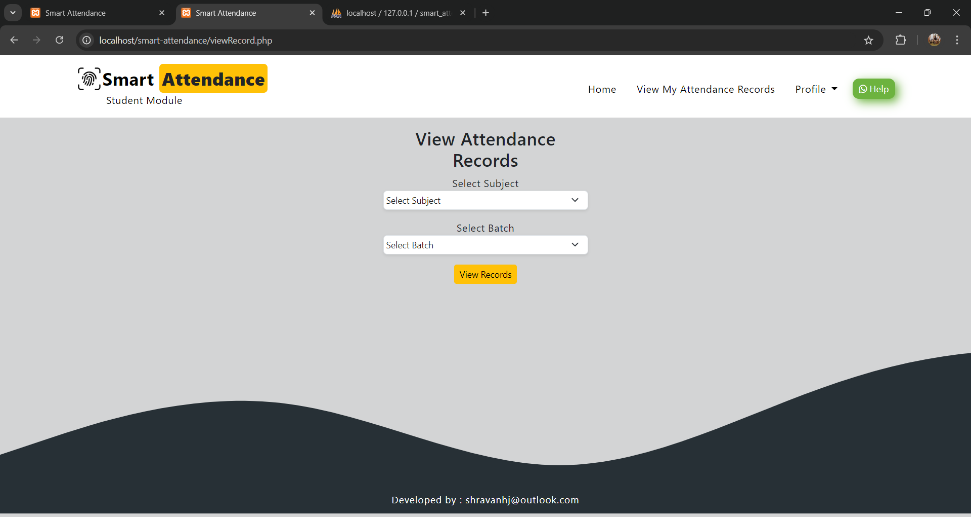
**Picture 13.4 Adding Subject and Lab Session Details**

****

**Picture 13.5 Capturing Student Attendance**

****

**Picture 13.6 Viewing Attendance Records**

****

1. **Conclusion**

The Smart Attendance system has successfully met its objectives, delivering a reliable and efficient solution for managing lab attendance through biometric authentication. The project demonstrates the effective application of modern web technologies and biometric systems to address common challenges in attendance management.

1. **Future Enhancements**
2. **Real-time Notifications**

* **Absentees and Late Arrivals**: Implement a notification system that alerts administrators and students about absentees and late arrivals. This feature will ensure timely communication and enable prompt action in case of attendance issues.

1. **Mobile App Support**

* **Convenient Access**: Develop a mobile application to provide students and teachers with easy access to attendance records, notifications, and other relevant features. The mobile app will enhance usability and ensure that users can interact with the system from their smartphones or tablets.

1. **Student Shortage List**

* **View Shortage List**: Integrate a feature to generate and view a list of students who are at risk of falling short of attendance requirements. This will allow administrators and teachers to monitor attendance patterns more effectively and take proactive measures to address potential issues.

These enhancements aim to improve the Smart Attendance system's functionality, user engagement, and overall effectiveness in managing attendance within educational institutions.

1. **References**
2. **Stack Overflow**. (n.d.). *Questions and Answers on Web Development*. Retrieved from <https://stackoverflow.com>
   * A valuable resource for developers to find solutions to coding issues, discuss best practices, and share knowledge on web development technologies used in the Smart Attendance project.
3. **SecuGen Corporation**. (n.d.). *SecuGen Fingerprint Scanner*. Retrieved from <https://www.secugen.com>
   1. Provides detailed information about SecuGen fingerprint scanners, including specifications and software necessary for integrating biometric authentication into the Smart Attendance system.
4. **Bootstrap Documentation**. (n.d.). *Bootstrap*. Retrieved from <https://getbootstrap.com>
   1. Official documentation for Bootstrap, offering guidelines and resources for implementing responsive design and user interface components used in the Smart Attendance project.