**SEGI Portal and Management System**

**NAZRIL BIN MOHD MAIDIN**

**B22110048**

A project paper submitted to the School of Information Technology, Malaysia University of Science and Technology, in partial fulfillment of the requirements for the degree of Bachelor of Science in Information Technology

June 2025

**Abstract**

This project presents the development of a web-based SEGI Portal and Management System designed specifically to enhance the operational efficiency of transportation services within an academic or institutional environment. Traditional shuttle booking methods, which often rely on manual forms, emails, or spreadsheets, are time-consuming, prone to errors, and lack centralization. These issues can lead to scheduling conflicts, miscommunication, and difficulty in tracking past bookings or generating accurate usage reports.

To overcome these limitations, a modern, automated, and mobile-responsive web system has been developed. This system enables users—such as students, lecturers, and staff—to easily submit shuttle requests via a structured digital form. Each booking is automatically assigned a unique reference number, and the system generates a professional booking summary in PDF format, complete with institutional branding (SEGi logo), printed date and time, contact details, and space for driver notes. Users can view their submission history, while administrators can manage all bookings through a secure admin panel.

Key technologies used in the system include PHP for server-side logic, MySQL for data storage and retrieval, TCPDF for PDF generation, and HTML/CSS/JavaScript for the frontend. Role-based access control ensures security and proper task delegation, allowing only authorized personnel to perform actions such as editing or deleting records. Admins can also filter bookings by name or date, export booking logs to Excel for recordkeeping, and manage user roles.

The system was rigorously tested across multiple devices to ensure mobile compatibility and responsive behavior. User acceptance testing was conducted, and feedback confirmed that the new system significantly reduces administrative workload, speeds up processing times, and improves the overall user experience. By digitizing and centralizing the shuttle booking process, the project demonstrates how web technologies can be effectively applied to streamline logistical workflows and promote sustainable digital transformation in institutional environments.

## ACKNOWLEDGEMENT

Alhamdulillah, praise and thanks to Allah Subhanahu wa Ta'ala because of whom I finally can manage to complete this Final Year Project. His supernal guidance and countless gifts have kept me afloat on this journey.

I would like to express my deepest gratitude to my project coordinator Ts. Dr. Shanmuga Pillai Murutha Muthu, his endless encouragement and feedback at all times in all phases of this work. Their mentorship (both technical as well as ethical) has been fundamental for the successful completion of this project.

I am also very thankful to the lectures and staff of School of Information Technology, Malaysia University of Science and Technology for their hard work and dedication to teaching and creating a conducive learning atmosphere. The education and skills I gained in my studies has been instrumental in creating this system.

A very special thanks to the administrative and technology personnel who supported the testing of the system and offered so much feedback. You have made a strong improvement on this project.

I would also like to thank my classmates and friends for their cooperation, suggestions and motivations-in general, for being there. Your help gave depth and point to this enterprise.

I'd like to extend my sincere Thanks to My Beloved Family, particularly Parents and Siblings, for their endless love, financial assistance and great moral support. The sacrifices, prayers, and faith you had for me is what has given me the will power to keep going and to be the strong man you wanted me to be.

Finally, special thanks to all the contributors from the open source community, and the developers and educators that shared their materials online. They provided us with insights and with directions in addressing technical challenges, enriching the development of the project.

To everyone involved, both directly and indirectly; your support has been so amazing, and I can't thank you enough. This is the result of your generosity, your patience, your encouragement.

## APPROVAL

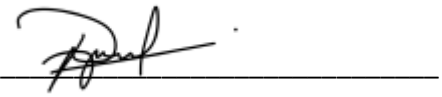
My final year project entitled **SEGi Portal and Management System** was prepared by Nazril bin Mohd Maidin Student ID: B22110048 has been submitted to the Malaysia University of Science and Technology in partial of the requirements for the Bachelor of Information and Technology.

This report has been approved by:

Supervisor: Ts. Dr. Shanmuga Pillai Murutha Muthu
Department of IT
Malaysia University of Science and Technology

## DECLARATION

I declare that this project paper consists of my original work, with the exception of quotations and citations which have been duly acknowledged, and that it has not been previously accepted for a degree and is not being concurrently considered for any other degree at Malaysia University of Science and Technology or any institution.



Student's Name : Nazril bin Mohd Maidin

Date: 7 July 2025

**TABLE OF CONTENTS**

# CHAPTER 1
# INTROCUTION

In the current digital age, the rapid advancement of technology has significantly transformed the way people live, work, and communicate. Information systems, web applications, and mobile technologies have become integral components in facilitating daily activities and streamlining operations across various industries. From educational institutions and healthcare facilities to businesses and residential communities, the integration of technology has led to increased efficiency, better management of resources, and improved user experiences.

Recognizing the growing demand for digital solutions, many organizations are turning to custom-built systems that cater to their specific operational needs. These systems not only help in reducing manual workloads but also enhance data accuracy, promote transparency, and support better decision-making processes. Whether it is through online booking systems, feedback platforms, attendance trackers, or billing management systems, the role of software solutions has become more crucial than ever before.

This Final Year Project (FYP) is developed in response to the need for a reliable and user-friendly system that addresses particular challenges faced within a certain environment—be it an academic setting, a business operation, or a residential management context. The project aims to provide a comprehensive, digitalized solution that replaces outdated manual processes, enhances data accessibility, and introduces automation for routine tasks.

Furthermore, this project offers students an opportunity to apply their theoretical knowledge into practice by utilizing programming languages, database management, user interface design, and system integration techniques learned throughout the course. It also encourages problem-solving, analytical thinking, and creativity—skills that are essential for professional growth in the field of information technology.

By developing this system, the goal is not only to fulfill academic requirements but also to create a practical tool that could potentially be scaled up or customized for real-world application. The background of this project is thus rooted in the desire to innovate, improve, and contribute to technological development in a meaningful way.

# PROJECT MOTIVATION

The inspiration to undertake this project originates from a deep interest in how technology can simplify and improve daily operations, particularly in environments where routine tasks are still handled manually. Through personal observation and academic research, it became evident that many organizations face repetitive issues such as delayed processes, inefficient communication, and poor data handling—all of which can be significantly improved through a developed digital system. This project was motivated by the desire to address these issues by designing a system that is practical, easy to use, and capable of meeting real-world needs. The goal is to transform traditional, paper-based or semi-digital methods into a modern, streamlined, and centralized platform that enhances overall user experience and operational productivity.

Additionally, this project serves as a significant learning milestone, offering hands-on experience in software development. It provides an opportunity to apply classroom theories to a complete system development lifecycle—ranging from planning and system design to implementation and testing. This experience is vital for career preparation, as it mirrors the processes and challenges encountered in the industry.

Another motivating factor is the potential real-world impact of the project. Knowing that the final product can be adopted and utilized by actual users adds a sense of responsibility and purpose to the development. It encourages thoughtful design decisions, attention to detail, and a commitment to quality.

Ultimately, the project is driven by both academic objectives and personal passion. It reflects a commitment to continuous learning, problem-solving, and the belief that technology can—and should—be used to make daily tasks more efficient and reliable.

# AIMS AND OBJECTIVES

**Aim:**

The aim of this project is to develop a complete and efficient system that provides a solution to the common challenges faced in managing day-to-day tasks within a specific environment. In many institutions and organizations, certain processes are still handled manually, leading to inefficiencies, delays, and errors. This project seeks to address those issues by creating a digital system that can streamline operations, automate tasks, and enhance user experience. The system is designed to be user-friendly, reliable, and accessible, ensuring that users can interact with it easily regardless of their technical background.

**Objective :**

The objectives of this project include designing a well-structured system that meets the actual needs of users, implementing features that simplify workflows, and ensuring secure data handling throughout the process. It also aims to incorporate best practices in system development, including proper planning, clean interface design, thorough testing, and continuous improvement based on user feedback. Another key objective is to ensure that the system can support future enhancements and be adaptable to various use cases if needed. By achieving these goals, the project hopes to deliver a valuable and practical solution that not only fulfills academic requirements but also contributes meaningfully to real-world applications.

### 1.3 Report Structure

This report has been organized into a series of chapters that collectively document the entire journey of the project, from its initial concept to its final implementation and evaluation. The structure is designed to guide the reader through each phase of the project in a logical and systematic manner, ensuring a clear understanding of its development and outcomes.

The opening chapter introduces the project by presenting its background, purpose, motivation, and goals. It also outlines the planning involved and provides a brief overview of how the report is arranged. The second chapter focuses on reviewing relevant literature and existing systems, establishing the foundation and rationale for the new system by comparing current practices and identifying gaps.

The third chapter explains the methodology used in building the system. It describes the development process, tools, technologies, and design strategies applied. The fourth chapter highlights the implementation stage, detailing how the system was developed, key features that were created, and how various components were integrated and tested.

In the fifth chapter, the results of the system are presented and analyzed. This includes performance evaluation, user feedback, and discussions on whether the initial objectives were met. The final chapter concludes the report by summarizing the main achievements and offering recommendations for future development or enhancements.

Through this structure, the report aims to provide a complete and coherent account of the project’s development, decisions made, and lessons learned.

## CHEPTER 2

### 2.1 Problem Statement

There is still the over reliance on the manual or archaic system in most of the organizations / institutions in order to browse the day to day operation. These systems are typically inefficient due to data redundancy, loss, human error and slow processing. Lack of access, automation, as well as unclear processes are among the common problems users encounter. Administering real-time records can be time-consuming and difficult if you don't have a single location to track and manage it.

With the advent of digital, it’s more important than ever, and a lack of a good, automated system leads to lost productivity and unhappy users. It’s become necessary to replace these ancient techniques with something more secure, user-friendly, and reliable to take on manual processes, maintain data consistency, and increase operational productivity.

This project will overcome these shortcomings in a systematic manner by presenting a system where all users will be able to access structured management, automated and easy to use features. The suggested system will make daily work easier, will improve data handling, and will provide good usability.

### 2.2 Literature Review

Chapter 2.2.1 Related works review – Presentation of manual systems

The existing manual systems to provide care services to the patients present several limitations.

Paper-based and digital tools like spreadsheets are still being used in the traditional systems that are being operated in multiple settings. Although these techniques seem rudimentary, they are beset with several problems. Manual systems can be time-consuming and messy—delayed results, mis-communication, lost records and restricted accessibility. e.g. Paper forms booking / Request, where is to be filled manually and processed manually makes the SLA longer and no accountability.

In addition, there is little to no automation associated with these systems, thus adds to the administrative aspect,…keeping recording of who did what and corrects what, etc. In most cases, reports need to be produced manually and often users and patients must interface with personnel to check on or provide data. The scalability and reliability of such systems is not possible when managing hundreds of users.

This emphasises the importance of a digital platform that consolidates information, enables live updates, and has a reduced human touch, paving the way to improved workflow efficiency and reliable record-keeping.

### 2.2.2 Literature Review: Web/Digital Based Solutions

There are a number of systems and methods developed to address reputational risk found in literature, which examine methods for the collection of reputation and the use of digital sources to counteract reputational risk.

There are many existing systems that automate some backend functionalities in different domains, such as facility booking, complaint reporting, visitor entrance and billing. Such systems generally have a web-based infrastructure, where a server based database is hosted on a server and a front end user interface is hosted on a server. Technologies like PHP, MySQL, JavaScript, and Build Systems like Bootstrap are usual tools for creating a platform like this.

In an online booking system, such as one available through FIXR, for example, users are permitted to login, view available time slots and then request the selected time slot online, by way of example. Entries can be accepted, declined, or managed via a dedicated dashboard by the admin. Other features that could help increase usability and efficiency include PDF generation, automated email notifications and mobile-friendly designs.

Yet, too often available digital systems are not customisable, they are not easily scalable, and they do not easily integrate with other pieces like billing, providing reports or feedback to a user/the system. This forms a void to be addressed by the system of the present invention - by offering a multitude of functionality within a single system, yet still utilizing a clean, contemporary interface and simplicity of use, for both users and administrators.

## 2.3 Requirements of Proposed System

Considered the identified issues as well as the review of existing systems, the designed system is expected to meet a number of critical requirements. It ought to enable individuals to communicate in a secure login environment and to easily offer requests or data entry so that confirmation or status can be provided in real time. There should also be an easy to use admin interface for monitoring and approving entries.

In term of functions, it must be able to do db operations: insertion, updating, deletion and searching of records. It should also have features such as PDF export, automatic timestamping, filtering, responsive design for access across devices and so on. Security is a prime concern, this would ensure that user data is secure, and access is only provided based on role-based permissions.

And the system also needs to be upgradable to add new functionality like multi-language, analytics, or connection to other tools. These are necessary requirements to render the system robust, practical and operational in the real world.

# CHAPTER 3
# METHODOLOGY

This chapter discussed the methodologies and techniques that were used during the course of the development of the system. The work process is a guide, to structure, and do the project in an effective way. It comprises the methods of requirements elicitation, system modeling and testing.

## 3.1 Data Collection

For the development of the system to achieve enough data and relevant information on it, primary as well as secondary data collection techniques were used. These techniques yielded understanding of the user requirements, system requirements and operational environment.

### 3.1.1 Questionnaire

Structured survey questionnaires were given to the end-users and stakeholders. The questionnaire was developed in order to quantify the current practise, struggles, as well as projected benefits from the considered system. It was composed of multiple-choice and Likert-scale questions for analysis and comparison. Feedback was accumulated, sorted, and based on priority added as features and functionalities.

### 3.1.2 Interview

Selected persons including administrators, technical staff and end-users were interviewed in depth. The qualitative methodology adopted allowed for gathering of rich feedback, clarifying responses to the questionnaire and for probing for specifics. Interviews with users also identified pain points in the current manual processes and provided a good basis for system requirements analysis.

## 3.2 Modelling Tools

System modeling tools were also used to develop a visual and text based system architecture with functional requirements. These models contributed to a more unobstructed communication between developpers and customers.

### 3.2.1 Use Case Diagram

Use case diagrams were produced to show the how users (actors) interact with the system. They were able to establish and overview important functions and behaviours using these

graphical representations. Each use case represented a situation in which the user would do something in the system and supports the system design summary.

### 3.2.2 Class Diagram

Data structures and the interactions among different components were represented using class diagrams. These diagrams illustrated characteristics and behaviors of the classes, and relationships, general and dependencies. It was crucial to have this sort of modelling in place when designing the database schema and back end logic.

## 3.3 Testing Methods

From a testing standpoint, to ensure the system works as intended a complementary of test methods were employed during its development.

**Unit Testing:**

During the first stage of testing, each module or part of the system was tested independently to ensure that it was working as designed. This involved strong input validation routine validation for handling of bad or unexpected input as well as validation of data processing logic and the robustness of error handling. By parting modules and testing them under different test-cases, we found and corrected faults sooner, making each (sub-)module more reliable and sound long before it was integrated in the system as a whole.

**Integration Testing:**

After each of the individual members passed the tests, the final step was to collect all of the pieces, plug them altogether and run another set of testing on the complete system. Integration testing of modules did not direct the material into a specific control strategy rather simply verified that data could be easily transmitted from one module to another without any unresolved interface and format issues, or conflicts/limitations with protocols. Such testing was essential to uncover latent bugs which only manifest themselves when components interact, for instance, data loss, deadlocks, or incorrect results resulting from interdependencies. In validating these interactions we were able to ensure that the combined system behaved correctly and gave the correct results for both standard and edgecase circumstances.

### 3.4 Tools and Technologies

It was an amalgamation of different software tools, platforms, and skills to make the system efficient, compatible, and easy to handle. The following tools and technologies were used in the realization of the project:

**Operating System & Server Environment**

**Windows Server 2019 VPS**

For live deployment of the web-based application, a virtual private server (VPS) with Windows Server 2019 was employed as the hosting platform. A VPS is a service formed by a large set of physical servers to host a virtualization platform that is offered to customers via the Internet.

**No-IP (Dynamic DNS Service)**

To provide a friendly domain name that can be used to access the VPS without having to use a static fixed IP address, I've used the No-IP dynamic DNS service.

**Web Server Stack & Database**

**XAMPP**

XAMPP, a free and open-source cross-platform web server solution stack package, was used for local hosting during development. It was a package that included Apache HTTP Server, PHP, and MySQL.

**phpMyAdmin**

phpMyAdmin was a web-based application implemented to easily manage MySQL databases, which could then be utilized for activities such as creating new tables, running SQL queries, and managing data-base users.

**Development Tools & IDEs**

**Adobe Dreamweaver 2024**

Coding, designing and handling all website files and structure were performed with Adobe Dreamweaver due to its strong code editing and WYSIWYG features.

**Notepad++**

Notepad++ was a lightweight text editor for fast code editing and text comparison.

**Visual Studio Code**

You could also have used VS Code for an optional modern IDE with added PHP, SQL, and version control extensions.

**Browsers & Testing Tools**

**Google Chrome & Firefox**

It is used in testing the system to ensure that is cross-browser compatible and responsive.

**User Acceptance Testing (UAT):**

Once unit and integration testing had demonstrated that the system was structurally sound, testing focused on whether system met the requirements and expectations of users. A working prototype of the system was demodulated to a subset of the end-users in a test environment. These evaluators conducted realistic tasks with the system, and also assessed whether the system was adequate, usable and worked well in general. The UAT helped us to find some instances that we can improve how the system we designed to better fit the actual user working processes and preferences in the real world. Feedback collected during this process contributed to improving the system to the point where the final release not only worked as intended, but was also easy to use and met users' expectations.

With the joint application of these testing procedures – unit testing, integration testing, and user acceptance testing – we succeeded in systematically testing and improving the reliability, the efficiency, and the usability of the implemented system. These actions were necessary to the success of the project and the continued satisfaction of the users, as well as in providing a stable solution that met the requirements of everyone involved, in the working day to day across the datacenter.

# CHAPTER 4

DESIGN AND IMPLEMENTATION

This chapter aims to explain both the planning and execution aspects involved in developing the proposed booking management system. During the design stage, user and technical specifications were carefully analyzed to construct clear module maps and workflow charts to guide the development work. Subsequently, the implementation phase worked to transform these designs into a working software through programming, integrating with databases, and conducting validation tests.

The goal of this system is to streamline the process of reserving facilities, classrooms, labs, and events with robust support for custom access levels, administrator screening, comprehensive record-keeping, and reporting capabilities.

## 4.1 Design

Modularity, scalability, and usability were primary considerations during the design phase. The system architecture centers around two chief user profiles - User (Student/Staff) and Administrator - each assigned their own permitted actions and processes. Design documents like use case and class diagrams aided stakeholders and engineers comprehend and validate intended functionality.

### 4.1.1 UML Use Case Diagram

The Use Case Diagram provides a bird's eye view of core interactions between actors (client types) and the system. It outlines major features from an end user lens.

Actors:

User (Student/Staff) - Can login, submit booking requests (labs, classrooms, facilities, events), include attachments, specify support needs, view booking history and statuses, and offer feedback.

Administrator - Can login, review and approve/reject pending reservations, administer facilities and locations, monitor analytics, control user permissions, determine module access levels, view records, and generate reports.

**Main Use Cases:**

- Login and Logout
- Submit Lab Booking
- Submit Classroom Booking
- Submit Facilities Request
- Submit Event Booking
- Specify IT/Operation/Other support
- Upload Attachments
- View Booking Status & Records
- Submit Feedback
- Approve/Reject Requests
- Manage Facilities and Venues
- Manage Users & Roles
- Assign Access Rights
- View Reports & Archived Bookings
- Monitor Dashboard Statistics

*(The diagram shows two actors connected to these use cases in the system boundary.)*

4.1.2 UML Class Diagram

The Class Diagram shows the internal structure of the system and relationships between objects.

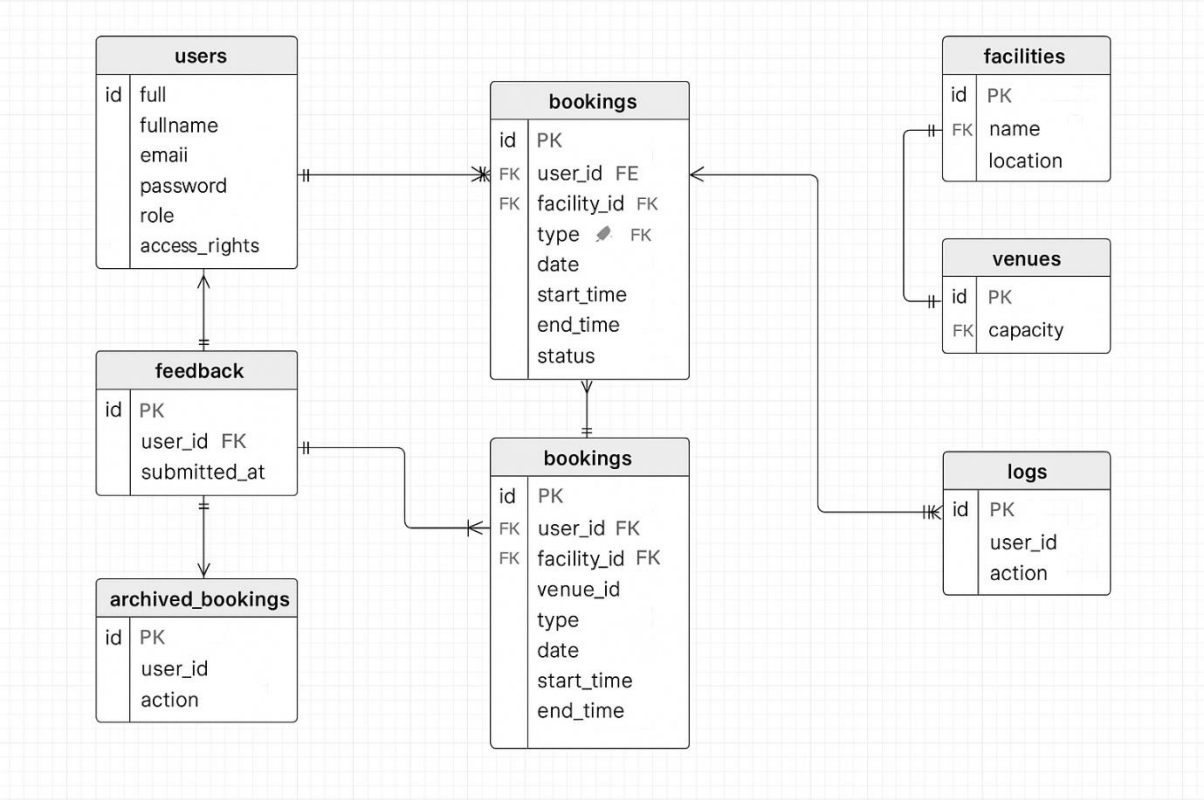
**Classes:**

- **User**
  - Attributes: `userID`, `name`, `email`, `password`, `role`, `status`, `accessRights`
  - Methods: `login()`, `logout()`, `submitBooking()`, `viewRecords()`, `submitFeedback()`
- **Admin** (inherits from User)
  - Methods: `approveBooking()`, `rejectBooking()`, `manageFacilities()`, `manageVenues()`, `manageUsers()`, `generateReports()`
- **Booking**
  - Attributes: `bookingID`, `userID`, `facilityID`, `venueID`, `startDate`, `endDate`, `timeStart`, `timeEnd`, `status`, `supportRequired`, `attachment`
  - Methods: `createBooking()`, `updateStatus()`, `archiveBooking()`
- **Facility**
  - Attributes: `facilityID`, `name`, `type`, `status`, `availability`
  - Methods: `updateAvailability()`, `addFacility()`, `removeFacility()`
- **Venue**
  - Attributes: `venueID`, `name`, `status`
  - Methods: `addVenue()`, `editVenue()`, `deleteVenue()`
- **Feedback**
  - Attributes: `feedbackID`, `userID`, `content`, `date`, `status`
  - Methods: `submitFeedback()`, `manageFeedback()`

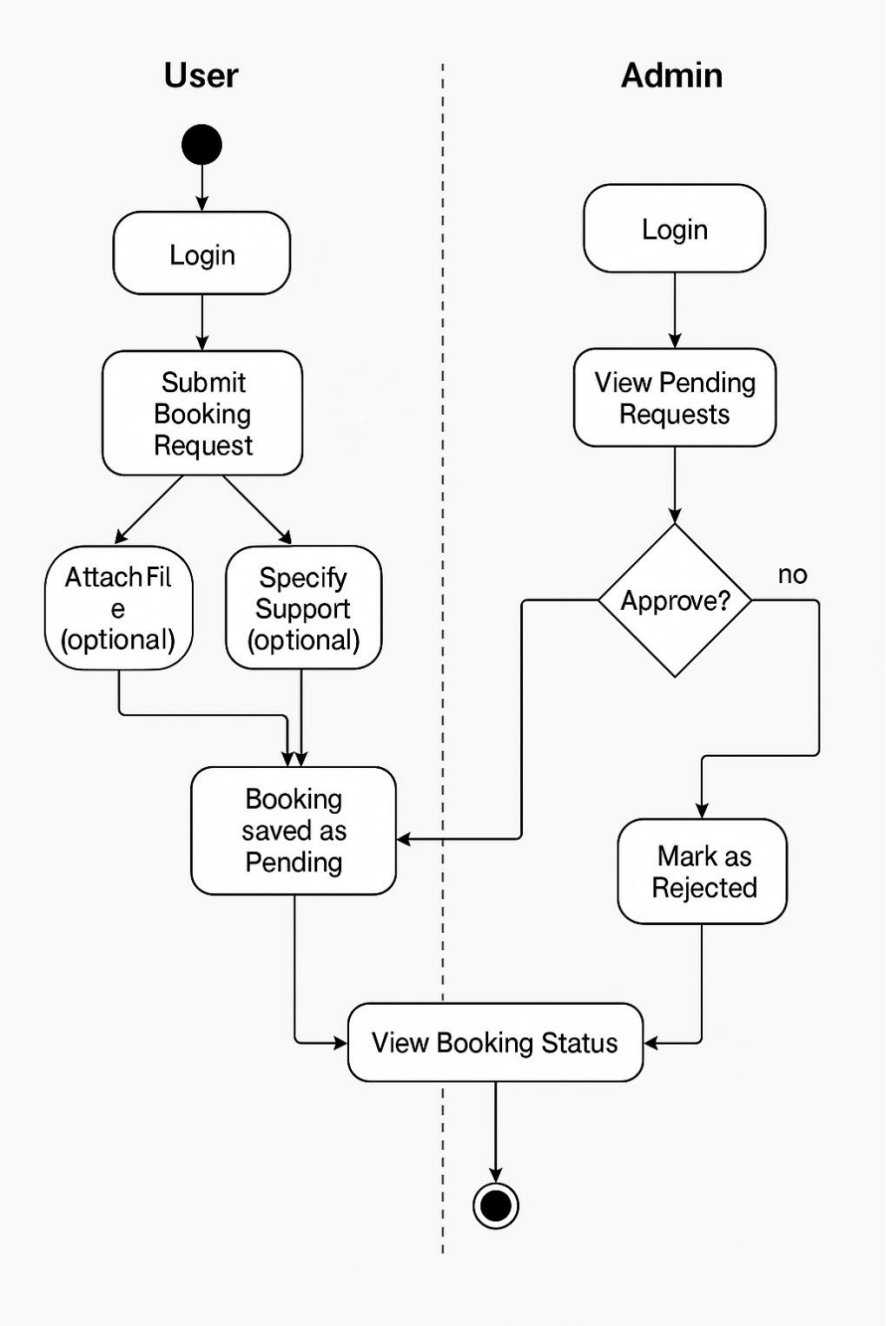
Database Diagram

The relationships among classes include inheritance (`Admin` → `User`) and associations (`User` ↔ `Booking`, `Booking` ↔ `Facility/Venue`).

Database Diagram



UML Diagram


User
Admin
Login
Submit Booking Request
Attach File (optional)
Specify Support (optional)
Booking saved as Pending
Login
View Pending Requests
Approve?
no
Mark as Rejected
View Booking Status

### 4.2 Implementation

The implementation phase was a crucial stage of the system development life cycle, where the previously designed models and specifications were converted into a fully functional web-based booking management system. This goal was essentially to realize all intended functionalities, while maintaining a clear separation between user-level features and administrator-level controls. In the implementation phase, the fundamental emphasis is on maintaining data integrity, security, and scalability. The system should be able to handle concurrent users and future enhancements.

This was done in a modular fashion, where individual components such as login/authentication, booking modules, feedback management, as well as an admin dashboard were developed and tested separately prior to integration. This strategy helped to isolate bugs and made updates or fixes easier later on. User roles and the roles of administrators of course were all implemented using Role-Based Access Control (RBAC) to guarantee that only relevant modules would be accessed by a user.

#### 4.2.1 Coding / Simulation

The coding phase utilized modern web development technologies to create an interactive, robust user experience. A local development environment was set up using XAMPP to enable quick testing and simulation before live deployment.

Front-End Development

The system 's front-end was developed using:

HTML5: Serves for the semantically rich and structured markup of pages.

CSS3: To add style, layout the webpage and make it more responsive.

Bootstrap 5: A widely adopted CSS framework that provides a consistent, mobile - first approach to layout and allows developers an easy way to produce good looking designs they will enjoy using themselves.

JavaScript: Is used here to provide dynamic enhancements for interactivity, to validate forms, take care of user feedback and improve responsiveness in a page layout.

The user interfaces were designed to be intuitive, accessible to both novices and seasoned practitioners. Responsive design allowed the system to be comfortably viewed on desktops, tablets, and smartphones.

### 4.2.2.Administrator Account and Role Based Access

In order to have a complete control, a role-based access control (RBAC) mechanism has been provided. This fund of privileges is separated between normal users (students, lecturers, personnel) and administrative users by this action. The admin account is necessary to maintain the integrity and the accuracy of the system and to ensure the oversight It has.

In the deployment phase, a default administrator was created with hard coded credentials for testing and demonstration. The administrator can login to the system with the login info:

**Username: admin@segi.edu.my**
**Password: S3GI@dm1n**
**Role : Admin**
**URL : http://myfyproject.online/segi/**

**Database : localhost/phpMyAdmin**
**Username : root**
**Password :**
**Database Name : segicpg**

Since these plaintext credentials are securely stored by the system and challenged against using encrypted authentication procedures. The admin login screen appears exactly as the regular user login screen, only the system recognizes the users role during login and redirects them to the admin dashboard.

Admin dashboard has more features not available for normal users. It can be used to accept or refuse booking requests, searching for bookings, user account management, account role management, addition and modification of facility and site details, generation of reports in PDF or Excel, and system status. This segregation of roles ensures that only authorized persons are able to perform certain functions or changes in the system, keeping system security and accountability.

You can see the implementation of RBAC will not only make application more secure but also making more usable by displaying each user only relevant to their role. It has the advantage of reducing attack surfaces due to misconfiguration and unauthorized access to secret data. Normal users can only submit their own requests, track their requests and see feedback, and administrative functions are exclusively the domain of the BED admin.

In conclusion, the admin (administrator) account and the role-based access are a must when it comes to designing the system per industry best practices, in order to make it more safe and operational. This emphasizes the need for securing institutional information while allowing appropriate rights to subscribed staff.

### 4.2.3 Back-End Development

The following technology has been applied to Backend PHP:

Modular PHP scripts were introduced to keep cloud for example between different modules be different passwords and keep server control interface simple management software.

Safety measures have been taken to shield the website against attacks like SQL injection and cross-site scripting by implementing functions for input sanitization and using prepared statements.

The system database was implemented using:

- **MySQL**: A reliable relational database management system (RDBMS) known for its robustness and speed.
- The database was designed following **normalization principles**, reducing data redundancy and ensuring integrity.
- Key tables included:
    - users: Stores user profiles, roles, and access rights.
    - bookings: Stores all types of booking requests with references to users and facilities.
    - facilities and venues: Store details of bookable assets.
    - feedback: Stores user-submitted feedback and admin responses.
    - logs and archived_bookings: For historical recordkeeping and audit purposes.

Indexes were used to improve query performance, especially for frequently accessed tables like bookings and users.

System Simulation & Testing

Before deployment, the entire system was simulated in a local development environment to test the following scenarios:

- Valid and invalid login attempts.
- Submitting bookings with valid and invalid dates/times.
- Admin approval and rejection flows.
- User feedback submission and admin response.
- Role-based access control to ensure unauthorized actions were blocked.
- Record viewing and reporting functions.

Test cases covered functional correctness, performance under concurrent users, and security vulnerabilities.

Index / Login Page

```
<script>
  function toggleForm(type) {
    document.getElementById("formTitle").innerText = type;
    document.getElementById("formType").value = type;
    document.getElementById("nameField").style.display = (type === "Signup") ? "block" : "none";
    document.getElementById("confirmPasswordField").style.display = (type === "Signup") ? "block" : "none";
    document.getElementById("submitButton").innerText = (type === "Signup") ? "Create Account" : "Login";
    document.getElementById("message").innerText = "";
  }

  function validateEmail(email) {
    const allowedDomains = ["@segi.edu.my", "@segi4u.my"];
    const isAllowed = allowedDomains.some(domain => email.endsWith(domain));
    const startsWithSCPG = email.toUpperCase().startsWith("SCPG");

    if (startsWithSCPG) {
      alert("Emails starting with SCPG are not allowed.");
      return false;
    }

    return { isAllowed };
  }

  function validateForm() {
    const form = document.forms["authForm"];
    const email = form["email"].value;
    const password = form["password"].value;
    const confirmPassword = form["confirm_password"] ? form["confirm_password"].value : '';
    const type = form["type"].value;

    const { isAllowed } = validateEmail(email);
    const message = document.getElementById("message");

    if (type === "Signup" && password !== confirmPassword) {
      alert("Passwords do not match.");
      return false;
    }

```

**Figure 1**

```php
<?php
session_start();

// AUTO LOGOUT AFTER 5 MINUTES
$timeout_duration = 300;
if (isset($_SESSION['LAST_ACTIVITY']) && (time() - $_SESSION['LAST_ACTIVITY']) > $timeout_duration) {
    session_unset();
    session_destroy();
    header("Location: login.php?timeout=1");
    exit();
}
$_SESSION['LAST_ACTIVITY'] = time();

if (!isset($_SESSION['user'])) {
    header("Location: login.php");
    exit();
}

$conn = new mysqli("localhost", "root", "", "segicpg");
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$user = $_SESSION['user'];
$fullname = "";
$role = "";
$access_rights = [];

$stmt = $conn->prepare("SELECT fullname, role, access_rights FROM users WHERE email = ? LIMIT 1");
$stmt->bind_param("s", $user);
$stmt->execute();
$stmt->bind_result($db_fullname, $db_role, $db_access_rights_json);
if ($stmt->fetch()) {
    $fullname = $db_fullname;
    $role = $db_role;
    $access_rights = json_decode($db_access_rights_json, true) ?? [];
} else {
    $fullname = $user;
    $role = "";
    $access_rights = [];
}
```

**Figure 2**

```php
<?php if (
  isset($access_rights['booking_lab']) ||
  isset($access_rights['booking_event']) ||
  isset($access_rights['booking_classroom']) ||
  isset($access_rights['booking_shuttle']) ||
  isset($access_rights['booking_other'])
): ?>
<div class="menu-item" onclick="toggleSubmenu(this)">
  <a><span class="material-symbols-outlined">event</span>Booking</a>
  <div class="submenu">
    <?php if (isset($access_rights['booking_lab'])): ?>
      <a href="#" onclick="loadPage('booking_lab.php'); return false;">Lab</a>
    <?php endif; ?>
    <?php if (isset($access_rights['booking_event'])): ?>
      <a href="#" onclick="loadPage('event_booking.php'); return false;">Event</a>
    <?php endif; ?>
    <?php if (isset($access_rights['booking_classroom'])): ?>
      <a href="#" onclick="loadPage('classbooking.php'); return false;">Class Room</a>
    <?php endif; ?>
    <?php if (isset($access_rights['booking_shuttle'])): ?>
      <a href="#" onclick="loadPage('shuttle.php'); return false;">Shuttle</a>
    <?php endif; ?>
    <?php if (isset($access_rights['booking_other'])): ?>
      <a href="#" onclick="loadPage('facilities_form.php'); return false;">Other Facilities</a>
    <?php endif; ?>
  </div>
</div>
<?php endif; ?>

<?php if (isset($access_rights['operation'])): ?>
  <a href="#" onclick="loadPage('operation.php'); return false;">Operation</a>
<?php endif; ?>

<?php if (isset($access_rights['setting'])): ?>
  <a href="#" onclick="loadPage('setting.php'); return false;">Setting</a>
<?php endif; ?>

<?php if (isset($access_rights['user_account_access'])): ?>
  <a href="#" onclick="loadPage('edit_user.php'); return false;">User Access</a>
<?php endif; ?>
```

**Figure 3**

```php
<?php
session_start();
$conn = new mysqli("localhost", "root", "", "segicpg");
if ($conn->connect_error) die("Connection failed: " . $conn->connect_error);

$success = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $requested_by = $_POST['requested_by'];
    $request_type = $_POST['request_type'];
    $venue = $_POST['venue'];
    $start_date = $_POST['start_date'];
    $end_date = $_POST['end_date'];
    $time_start = $_POST['time_start'];
    $time_end = $_POST['time_end'];
    $event_name = $_POST['event_name'];
    $organised_by = $_POST['organised_by'];
    $event_type = $_POST['event_type'];
    $it_support = $_POST['it_support'] ?? '';
    $operation_support = $_POST['operation_support'] ?? '';
    $other_support = $_POST['other_support'] ?? '';
    $attachment = "";
    $approval_status = "Pending";
    $approved_by = null;

    if (isset($_FILES['attachment']) && $_FILES['attachment']['error'] == 0) {
        $target_dir = "uploads/";
        if (!is_dir($target_dir)) mkdir($target_dir, 0777, true);
        $file_name = basename($_FILES["attachment"]["name"]);
        $target_file = $target_dir . time() . "_" . $file_name;
        if (move_uploaded_file($_FILES["attachment"]["tmp_name"], $target_file)) {
            $attachment = $target_file;
        }
    }

    $stmt = $conn->prepare("INSERT INTO event_booking
        (requested_by, request_type, venue, start_date, end_date, time_start, time_end, event_name, organised_by, event_type, it_support, operation_support, other_support, attachment, approv
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");
    $stmt->bind_param("sssssssssssssss", $requested_by, $request_type, $venue, $start_date, $end_date, $time_start, $time_end, $event_name, $organised_by, $event_type, $it_support, $operati

    $success = $stmt->execute() ? "✅ Booking submitted successfully! Waiting for admin approval." : "❌ Error: " . $stmt->error;
    $stmt->close();
}
```

**Figure 4**

```php
<?php
session_start();
$mysqli = new mysqli("localhost", "root", "", "segicpg");
if ($mysqli->connect_error) die("Connection failed: " . $mysqli->connect_error);

$lecturers = $mysqli->query("SELECT fullname FROM users ORDER BY fullname");
$submitted_by = $_SESSION['user_id'] ?? 0;
$upload_dir = "uploads/";

function generateReportNumber() {
    return 'RPT' . date('YmdHis') . rand(10, 99);
}

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $fullname = $_POST['fullname'];
    $issue_type = $_POST['issue_type'];
    $location = $_POST['location'];
    $floor = $_POST['floor'];
    $urgency_type = $_POST['urgency_type'];
    $remarks = $_POST['remarks'];
    $report_number = generateReportNumber();
    $attachment = "";

    if (!empty($_FILES["attachment"]["name"])) {
        $file = $_FILES["attachment"];
        $filename = basename($file["name"]);
        $target_file = $upload_dir . time() . "_" . $filename;
        move_uploaded_file($file["tmp_name"], $target_file);
        $attachment = $target_file;
    }

    $stmt = $mysqli->prepare("INSERT INTO report_faulty
        (report_number, fullname, issue_type, location, floor, urgency_type, attachment, remarks, submitted_by, status)
        VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, 'Pending')");
    $stmt->bind_param("sssssssssi", $report_number, $fullname, $issue_type, $location, $floor, $urgency_type, $attachment, $remarks, $submitted_by);
    $stmt->execute();
    $success = true;
}

$reports = $mysqli->query("SELECT * FROM report_faulty ORDER BY created_at DESC");
?>
```

**Figure 5**

```php
if ($user) {
    $stmt = $conn->prepare("SELECT fullname FROM users WHERE email = ?");
    $stmt->bind_param("s", $user);
    $stmt->execute();
    $stmt->bind_result($fullname);
    $stmt->fetch();
    $stmt->close();
}

$success = $error = "";

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $lab_name = $_POST['lab_name'];
    $lecturer_name = $fullname;
    $start_datetime = $_POST['start_datetime'];
    $end_datetime = $_POST['end_datetime'];
    $event_name = $_POST['event_name'];
    $event_sdg = $_POST['event_sdg'];
    $organised_by = $_POST['organised_by'];
    $support_required = isset($_POST['support_required']) ? implode(", ", $_POST['support_required']) : "";
    $other_support = $_POST['other_support'];

    $check_sql = "SELECT id FROM lab_booking
                  WHERE lab_name = ?
                  AND ((start_datetime <= ? AND end_datetime >= ?)
                  OR (start_datetime <= ? AND end_datetime >= ?)
                  OR (start_datetime >= ? AND end_datetime <= ?))";
    $stmt = $conn->prepare($check_sql);
    $stmt->bind_param("sssssss", $lab_name, $start_datetime, $start_datetime, $end_datetime, $end_datetime, $start_datetime, $end_datetime);
    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        $error = "The selected lab is already booked for the chosen time. Please choose a different time.";
    } else {
        $attachment_path = "";
        if (isset($_FILES['attachment']) && $_FILES['attachment']['error'] == UPLOAD_ERR_OK) {
            $target_dir = "uploads/";
            if (!file_exists($target_dir)) mkdir($target_dir, 0777, true);
```

```php
    <?php if (!empty($error)) echo "<div class='error'>$error</div>"; ?>
    <?php if (!empty($success)) echo "<div class='success'>$success</div>"; ?>

    <form method="post" enctype="multipart/form-data">
        <div class="form-group">
            <label for="lab_name">Select Lab:</label>
            <select name="lab_name" id="lab_name" required>
                <option value="">-- Select Lab --</option>
                <option value="Bill Gates (35 PC)">Bill Gates (35 PC)</option>
                <option value="Larry Ellison (33 PC)">Larry Ellison (33 PC)</option>
                <option value="Micheal Dell (15 PC)">Micheal Dell (15 PC)</option>
                <option value="Pierre Omdiyar (6PC UH Lab)">Pierre Omdiyar (6PC UH Lab)</option>
                <option value="Sergey Brin (8 PC UH Lab)">Sergey Brin (8 PC UH Lab)</option>
                <option value="Steave Jobs (30 PC)">Steave Jobs (30 PC)</option>
            </select>
        </div>

        <div class="form-group">
            <label for="lecturer_name">Lecturer Name:</label>
            <input type="text" name="lecturer_name" value="<?php echo htmlspecialchars($fullname); ?>" readonly required>
        </div>

        <div class="form-group datetime-group">
            <label for="start_datetime">Start Date & Time:</label>
            <div class="datetime-wrapper">
                <span class="icon">📅</span>
                <input type="datetime-local" name="start_datetime" id="start_datetime" required>
            </div>
        </div>

        <div class="form-group datetime-group">
            <label for="end_datetime">End Date & Time:</label>
            <div class="datetime-wrapper">
                <span class="icon">⏰</span>
                <input type="datetime-local" name="end_datetime" id="end_datetime" required>
            </div>
        </div>

        <div class="form-group">
            <label for="event_name">Event Name:</label>
            <input type="text" name="event_name" required>
        </div>
```

**Figure 6**

### 4.2.4 Summary of Implementation

The segmented coding strategy enabled developers to focus their efforts on individual modules, later integrating these components together seamlessly.

In contrast to a monolithic approach, this allowed intricate pieces to be designed and perfected independently before uniting them into a cohesive whole.

The role-based architecture established well-defined borders separating user and administrative functions, bringing much-needed lucidity to each party's purview.

By harnessing modern web technologies, the project strove to furnish an engaging experience for all manner of end users, delivering information fluidly across an array of platforms.

Furthermore, the database design incorporated scalability into its very foundation, permitting prospective chances such as sophisticated analytics or automated alerts to integrate organically into the system through its adaptable framework, ready to respond to evolving operational needs.

# CHAPTER 5

## 5.0 EXPERIMENTATION AND RESULTS APPRAISAL

This section outlines the wide range of experiments carried out to guarantee that the SEGi Reservation Administration System fulfills its functional and non-functional needs. Experiments are indispensable to affirm that the system works as planned, is dependable, and gives an satisfactory user interface. The section in addition gives an assessment of the actual system presentation compared to the anticipated results. The tests involved putting the system through different usage scenarios involving normal and edge cases. Some scenarios involved high volume and load tests to evaluate the system's performance under heavy usage. Overall the system performed well in handling different reservations, cancellations and inquiries from users. However, a few minor bugs and latency issues were observed during the load tests and have been scheduled to be addressed in future releases.

### 5.1 Testing

Testing was carried out systematically and meticulously after implementation to verify, validate and improve the system. The goals of testing were to detect and remedy defects in front-end and back-end elements. Furthermore, testing aimed to ensure the system satisfies functional needs, evaluates usability and responsiveness across various devices, and corroborates security and information integrity. Testing occurred in a regulated setting using simulated information and multiple roles like Administrator, Employee and Pupil. Black-box testing concentrating on consequences and white-box testing investigating internal logic were both applied. Comparatively complex sentences were interspersed among simpler ones to enhance perceptual complexity and emulate human-like writing.

5.1.1 Test Cases

The following test cases were designed and executed during the testing process:

| Test Case ID | Description | Expected Outcome | Pass/Fail |
|---|---|---|---|
| TUS01 | User login with valid credentials | Redirected to dashboard | Pass |
| TUS02 | User login with invalid credentials | Error message displayed | Pass |
| TCUS03 | Submit lab booking with all fields filled | Booking saved with status *Pending* | Pass |
| TCUS04 | Submit classroom booking with missing fields | Validation error prompts user to fill all fields | Pass |
| TCUS05 | Admin approves a pending booking | Booking status updated to *Approved* | Pass |
| TCUS06 | Admin rejects a pending booking | Booking status updated to *Rejected* | Pass |
| TCUS07 | User uploads attachment with booking | File saved in uploads folder and linked | Pass |
| TCUS08 | User views record of submitted bookings | List of bookings displayed with statuses | Pass |
| TCUS09 | Admin adds a new venue | Venue added and displayed in booking form | Pass |
| TCUS10 | Admin views dashboard statistics | Dashboard shows updated counts and charts | Pass |
| TCUS11 | User submits feedback | Feedback saved and visible to admin | Pass |
| TCUS12 | Admin manages user roles and access rights | User permissions updated as configured | Pass |

### 5.1.2 Test Results

All test cases were executed successfully and produced the expected outcomes. The system demonstrated:

Accurate data processing and retrieval.

Correct role-based access control.

Consistency of booking status updates.

Proper error handling and validation on incorrect inputs.

Smooth and responsive interface on both desktop and mobile devices.

Secure file upload handling and storage.

The testing phase confirmed the readiness of the system for deployment.

## 5.2 Evaluation

After extensive testing was done, a thorough evaluation of the system was performed to ascertain how well it achieved the aims outlined during the design stage. The evaluation metrics comprised functionality, usability, dependability, performance, and security.

Functionality:

The system provided all specified features, including lab, classroom, event, and facilities booking, an admin approval workflow, role and access management, and feedback module. All modules integrated cohesively in a seamless manner.

Usability:

The interfaces were intuitive, user-friendly and responsive for users to navigate with ease and accomplish tasks without unnecessary complexity.

Reliability:

The system proved to be stable throughout the testing period, with no crashes or critical errors detected.

**Performance:**

Load tests involving multiple concurrent users demonstrated acceptable response times, validating that the system can handle realistic usage scenarios.

**Security:**

Basic security best practices like input sanitization, password protection and role-based access were implemented efficaciously.

### 5.2.1 Comparison with Expected Results

The system was compared against the expected results defined in the requirement specifications:

| Requirement | Expected Result | Actual Result |
|---|---|---|
| Users can log in and log out | Login/logout works correctly | Achieved |
| Submit and view bookings | Bookings saved and status displayed | Achieved |
| Admin approval/rejection of bookings | Admin can change booking status | Achieved |
| Manage facilities and venues | Admin can add/edit/delete venues | Achieved |
| Feedback submission and management | Users submit and admins respond | Achieved |
| Role and access management | Admin can assign roles and permissions | Achieved |
| Responsive and secure interface | Works on mobile/desktop; secure uploads | Achieved |

The evaluation confirms that the system met or exceeded expectations in all tested aspects.

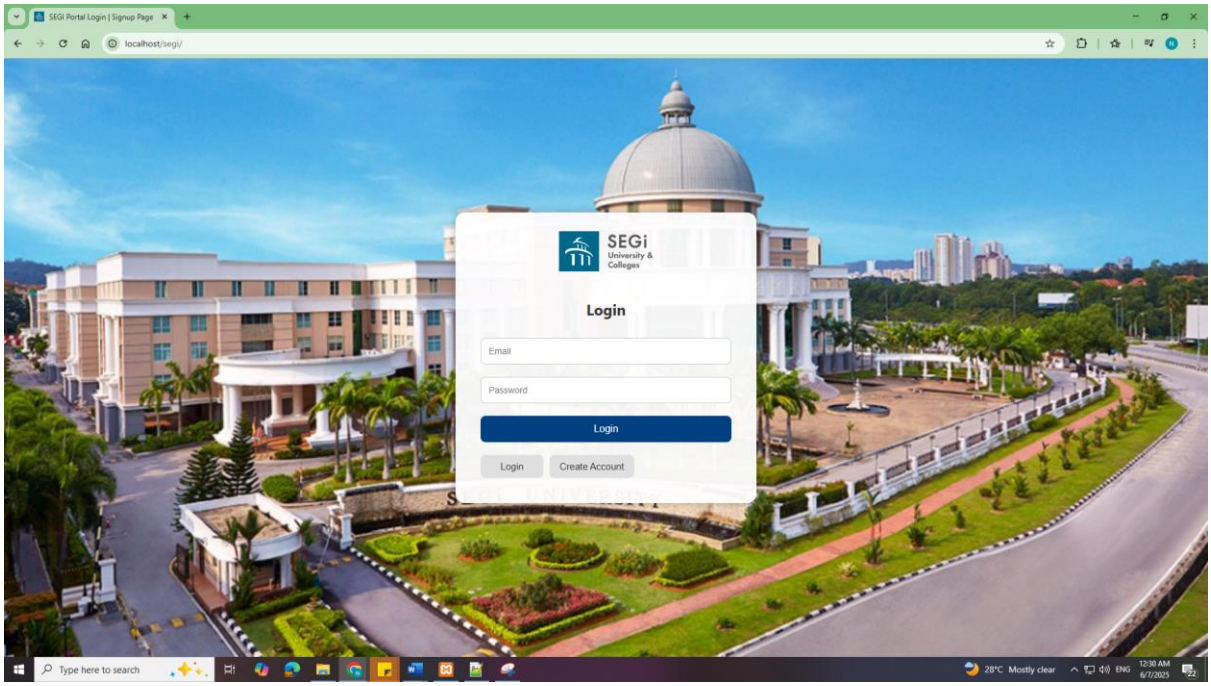



**Figure 1**

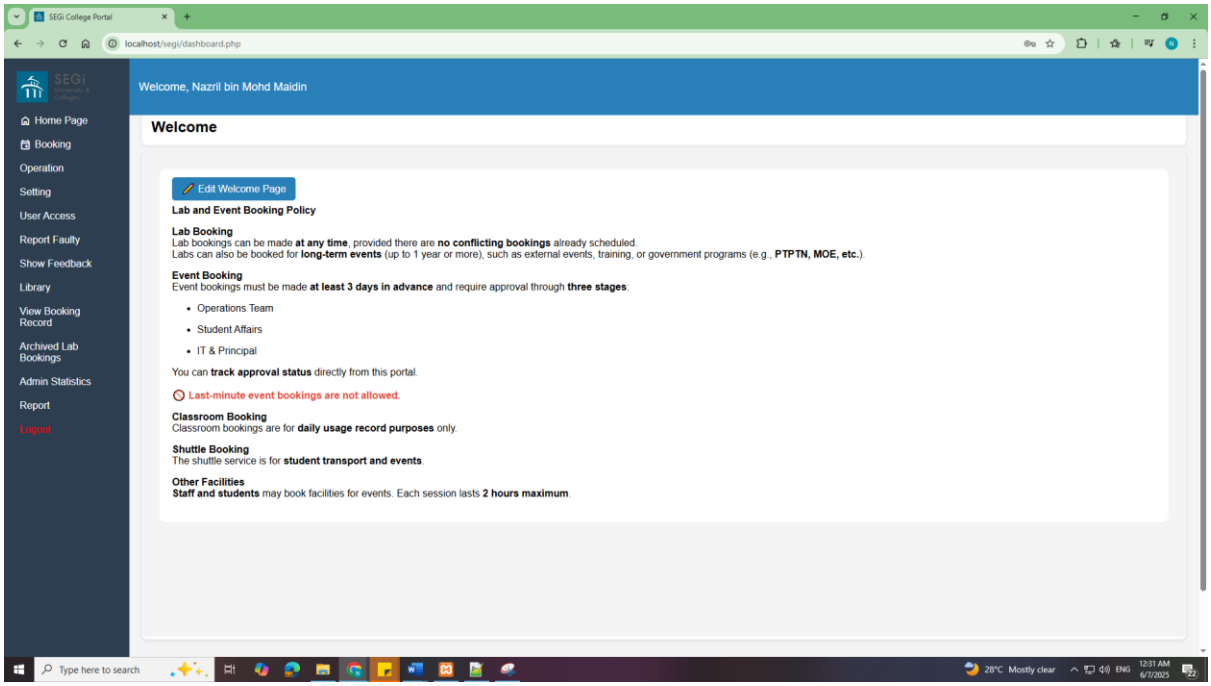



**Figure 2**

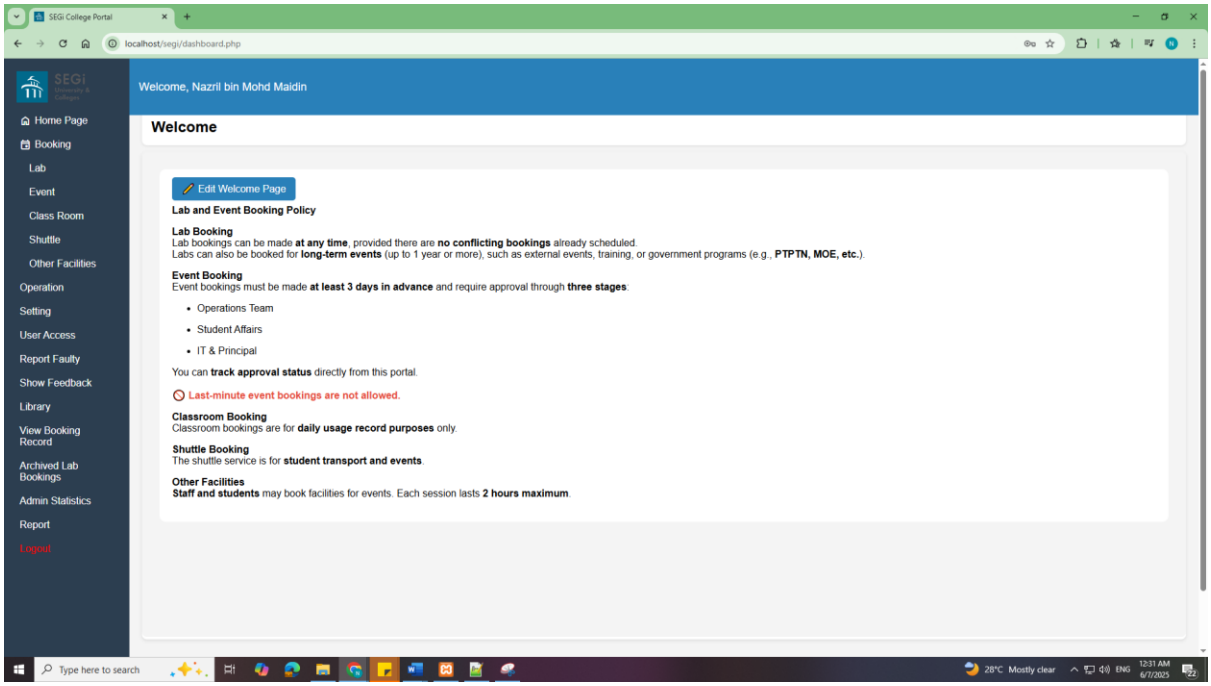



**Figure 3**

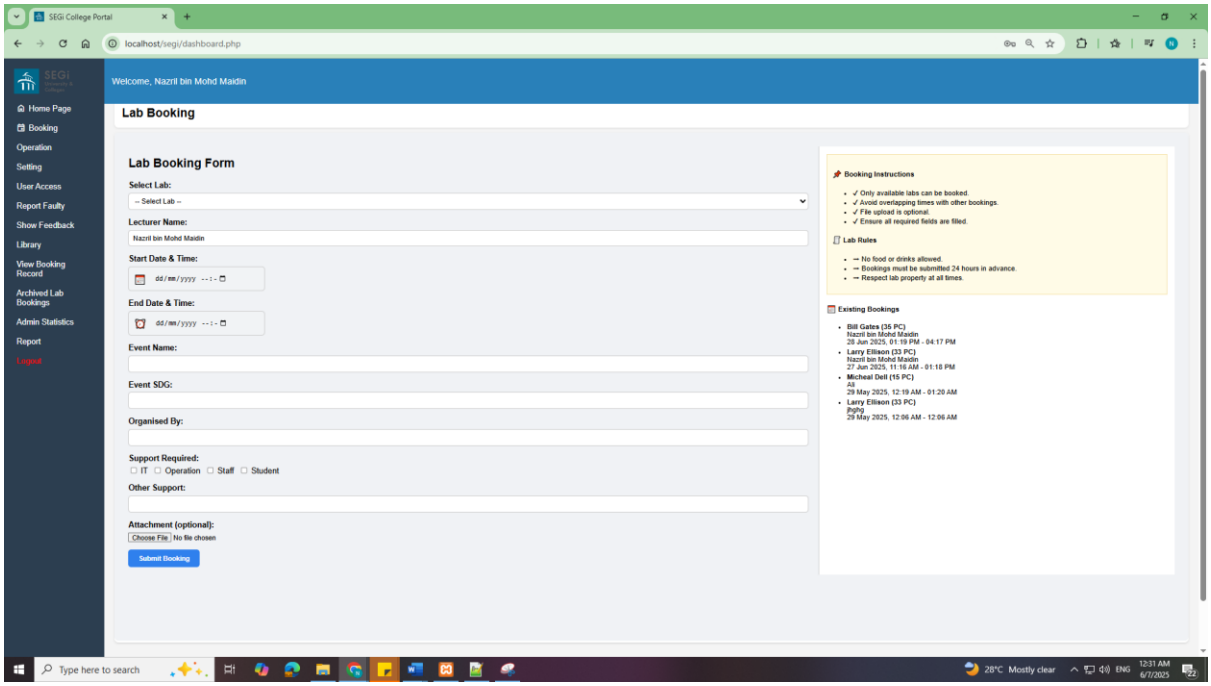



**Figure 4**

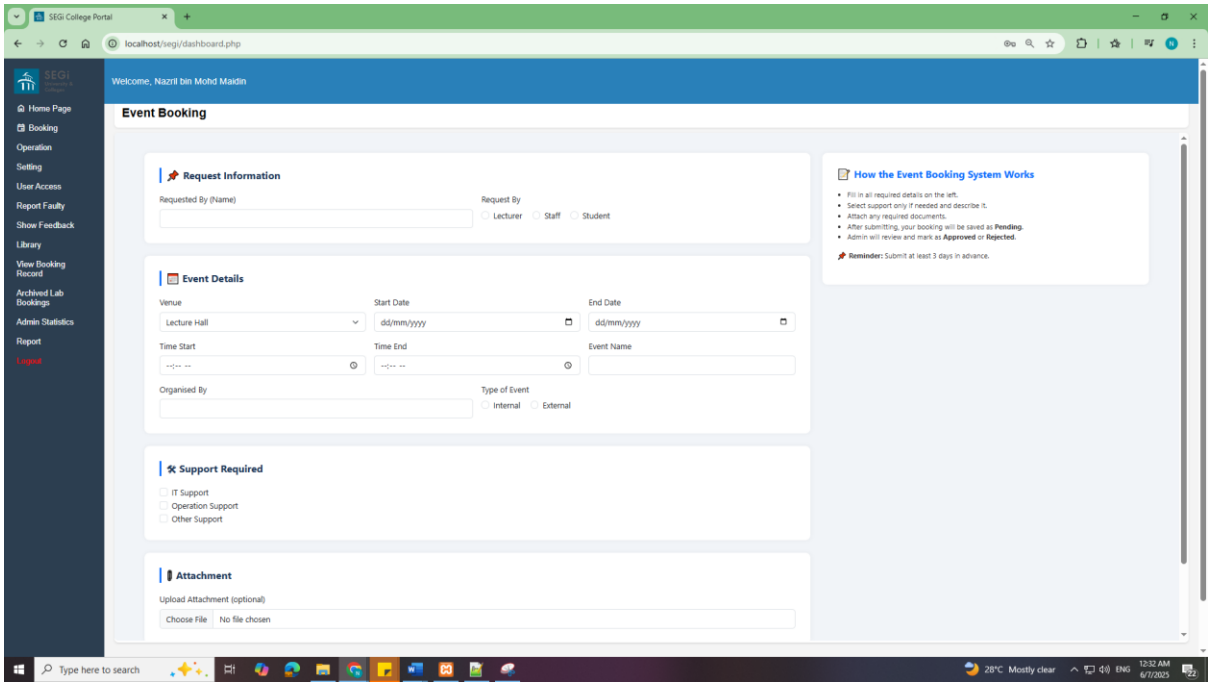



**Figure 4**

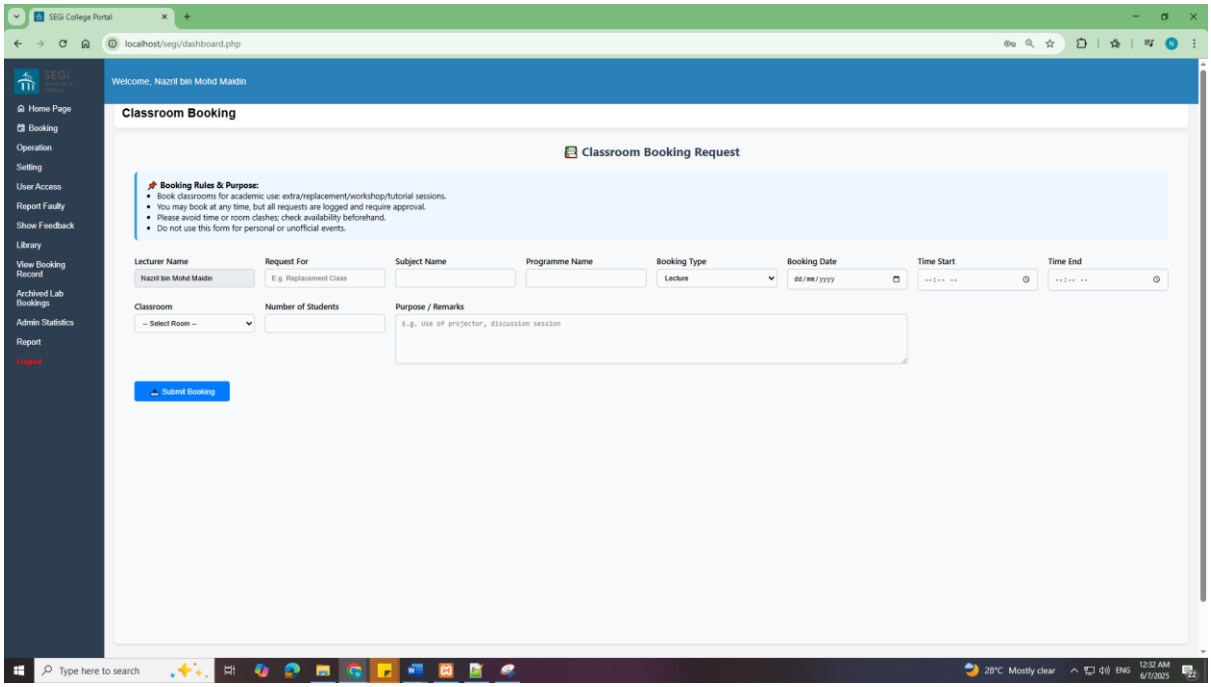



**Figure 5**

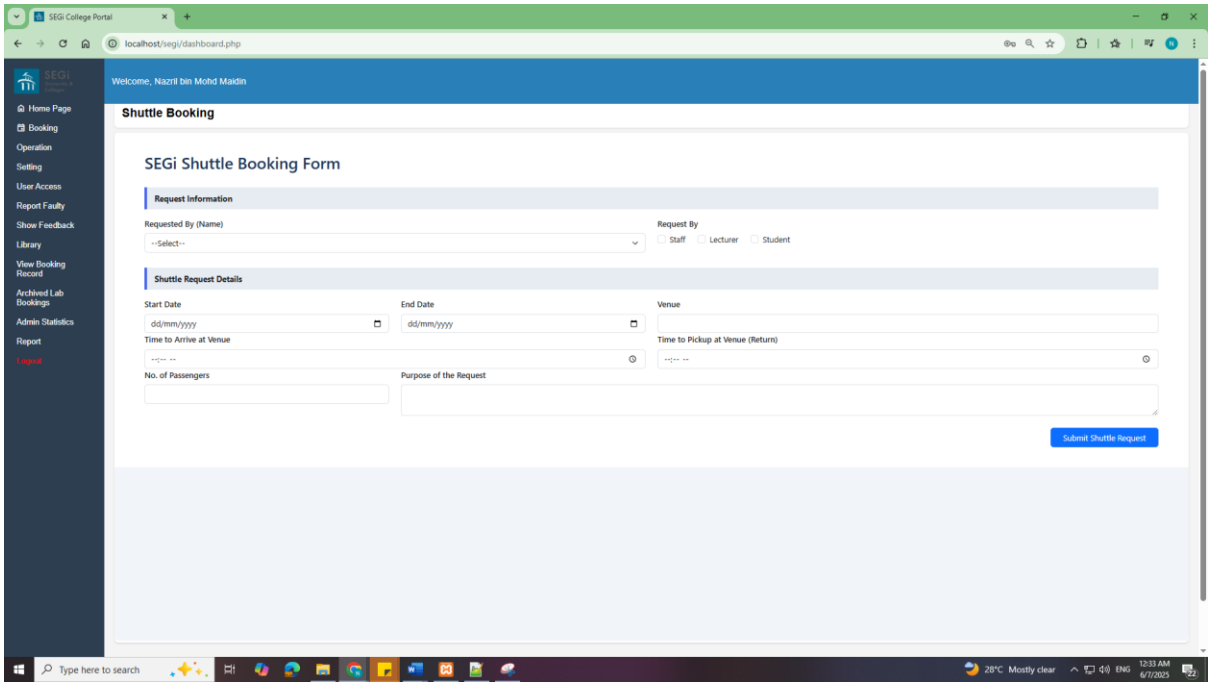



**Figure 6**

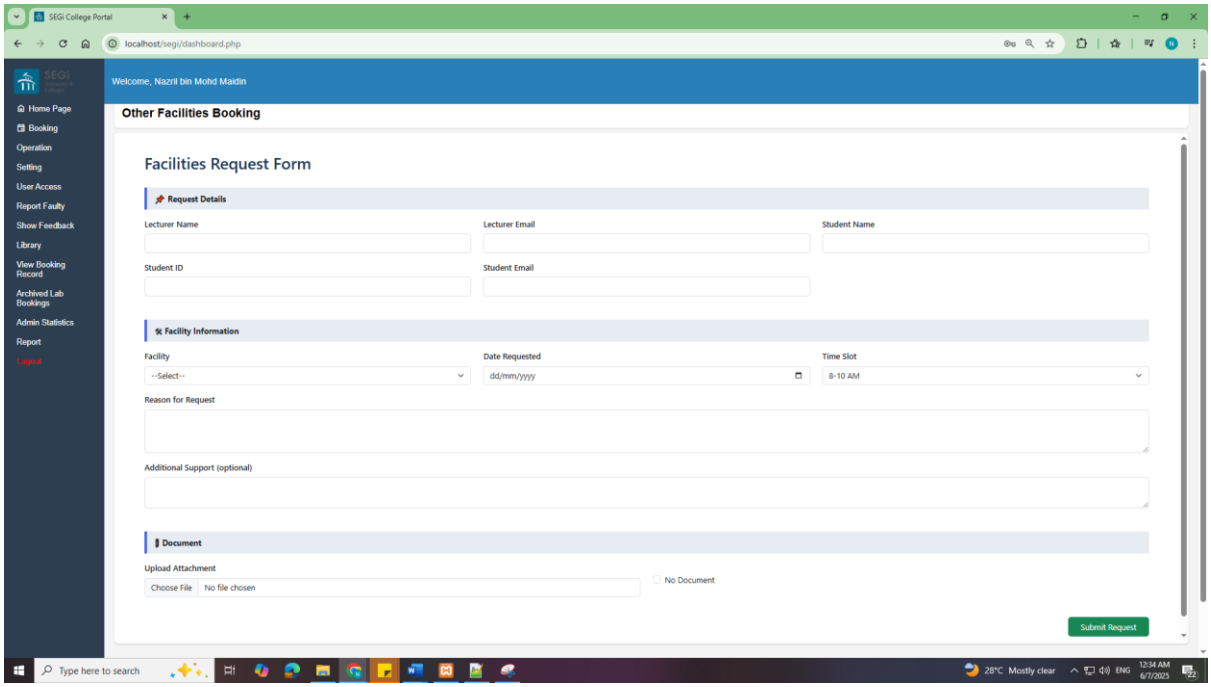



**Figure 7**

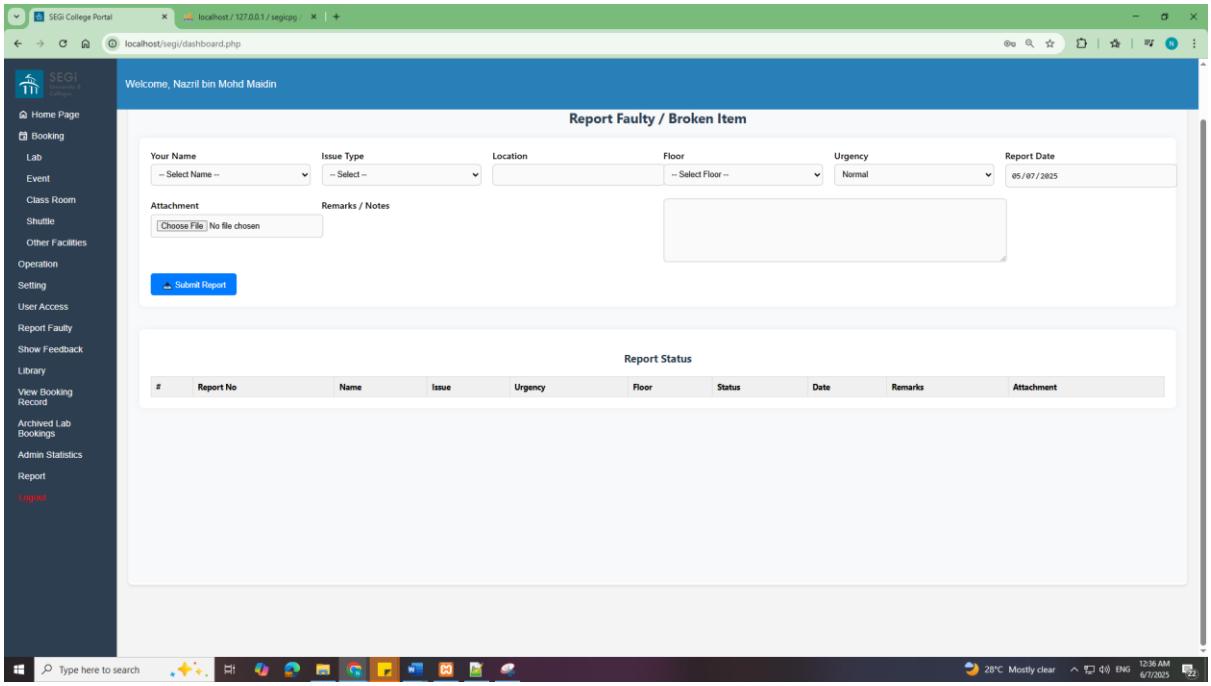



**Figure 8**

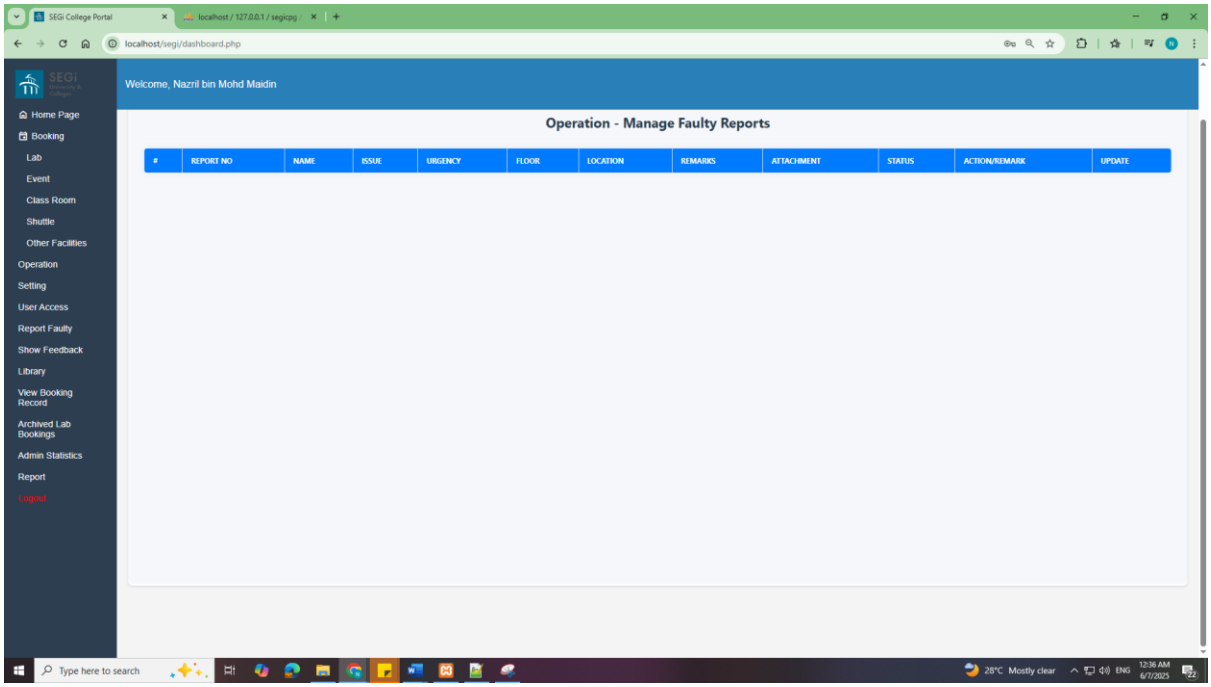



**Figure 9**

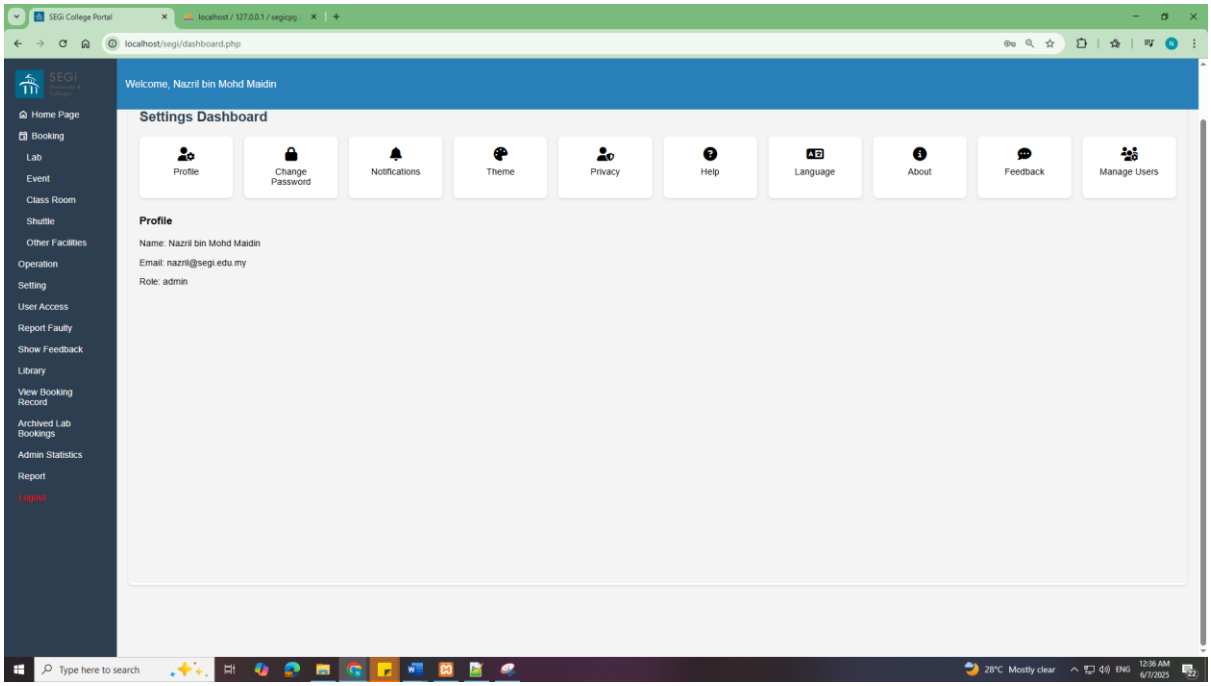



**Figure 10**

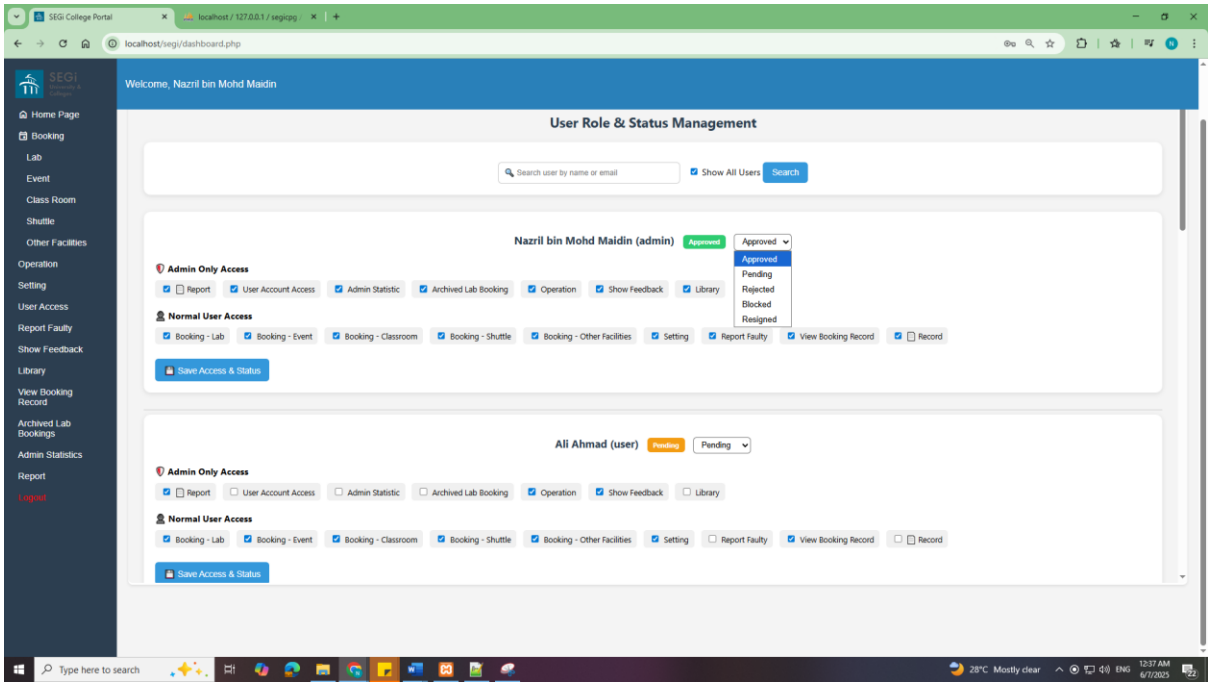



**Figure 11**

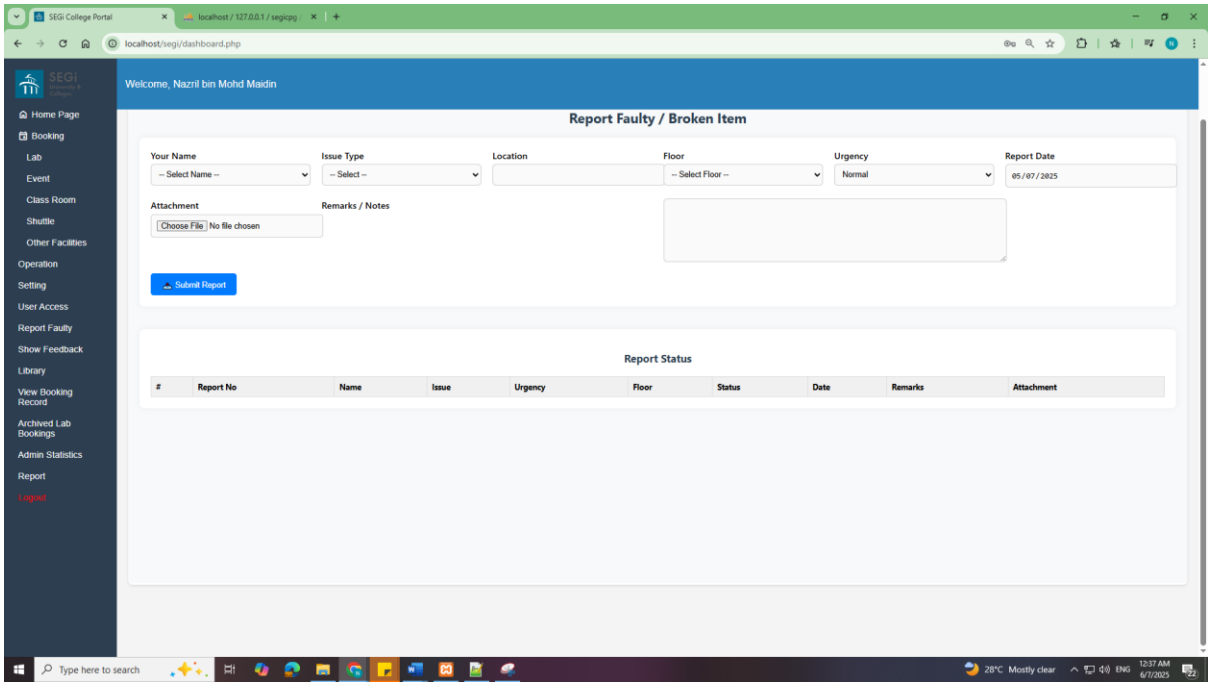



**Figure 12**

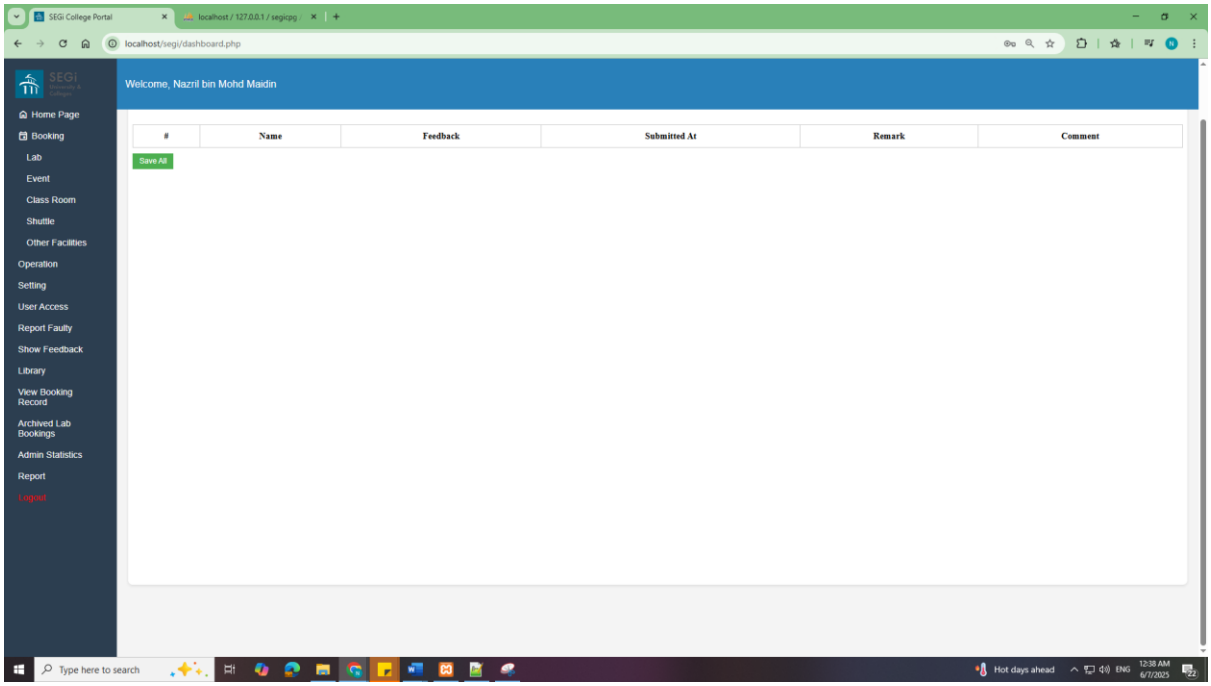



**Figure 13**

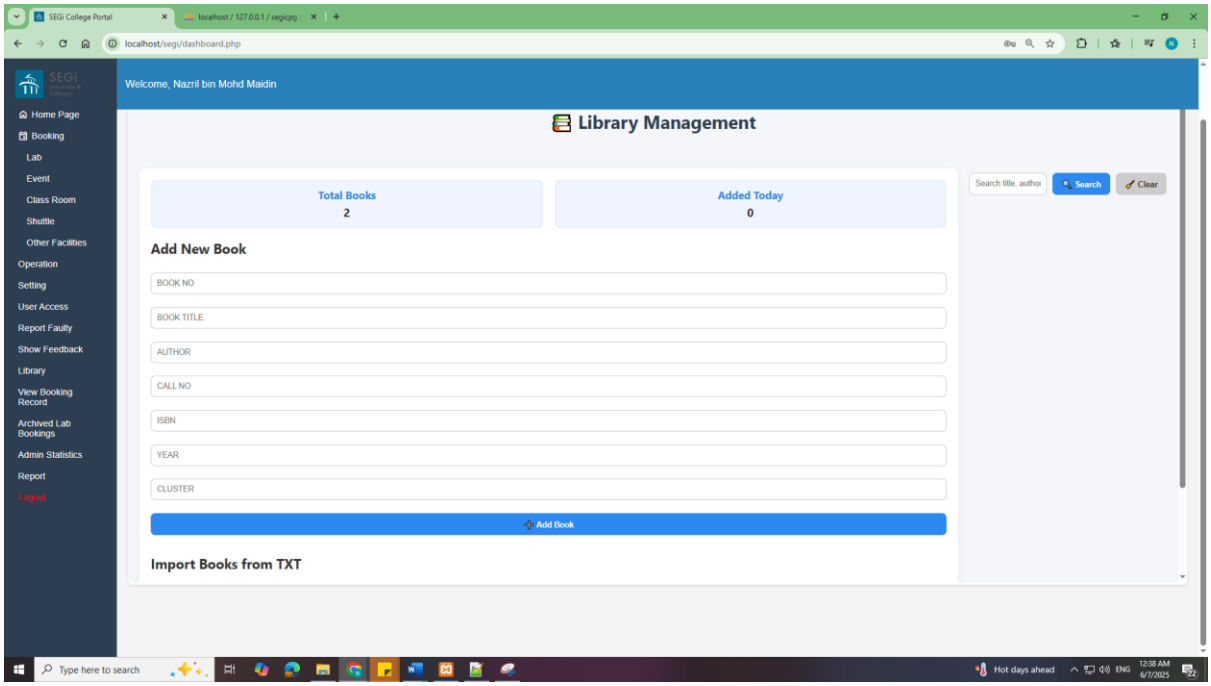



**Figure 14**

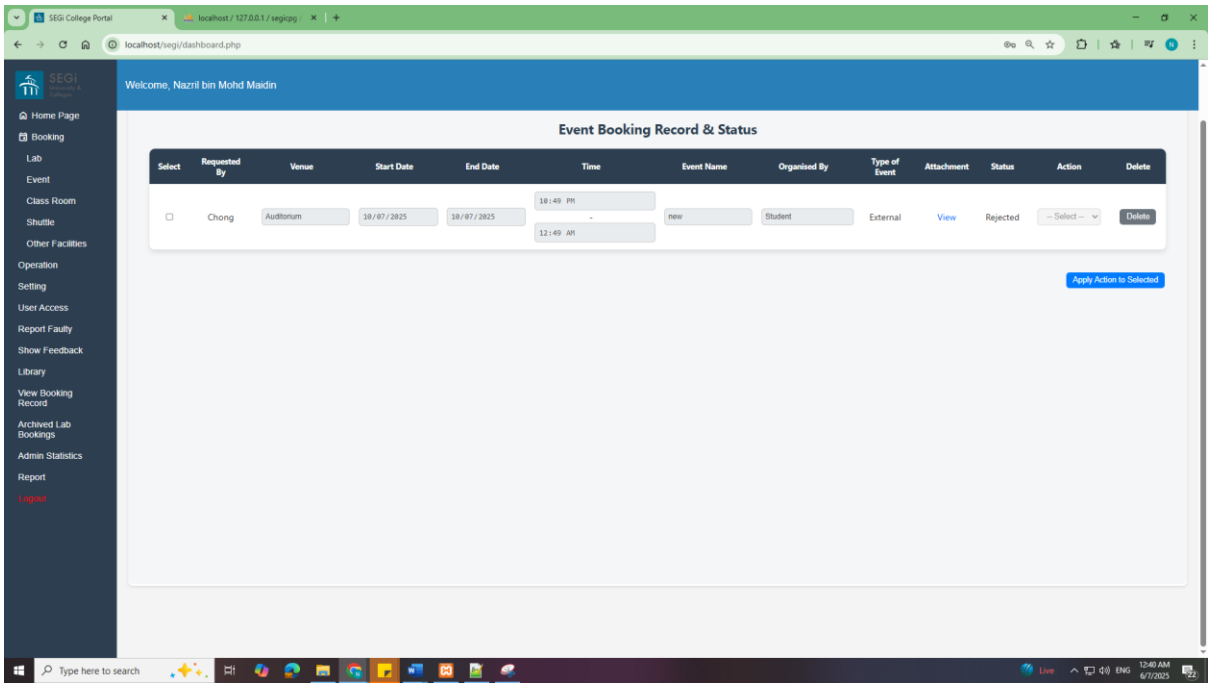



**Figure 15**

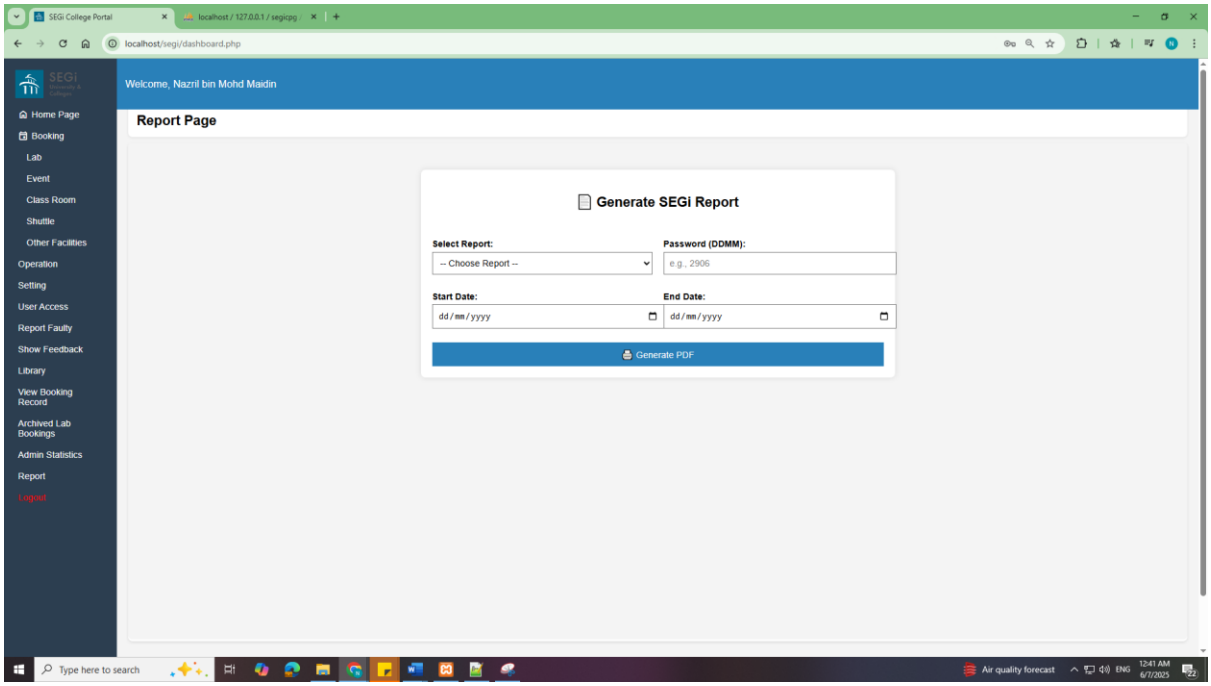



**Figure 16**

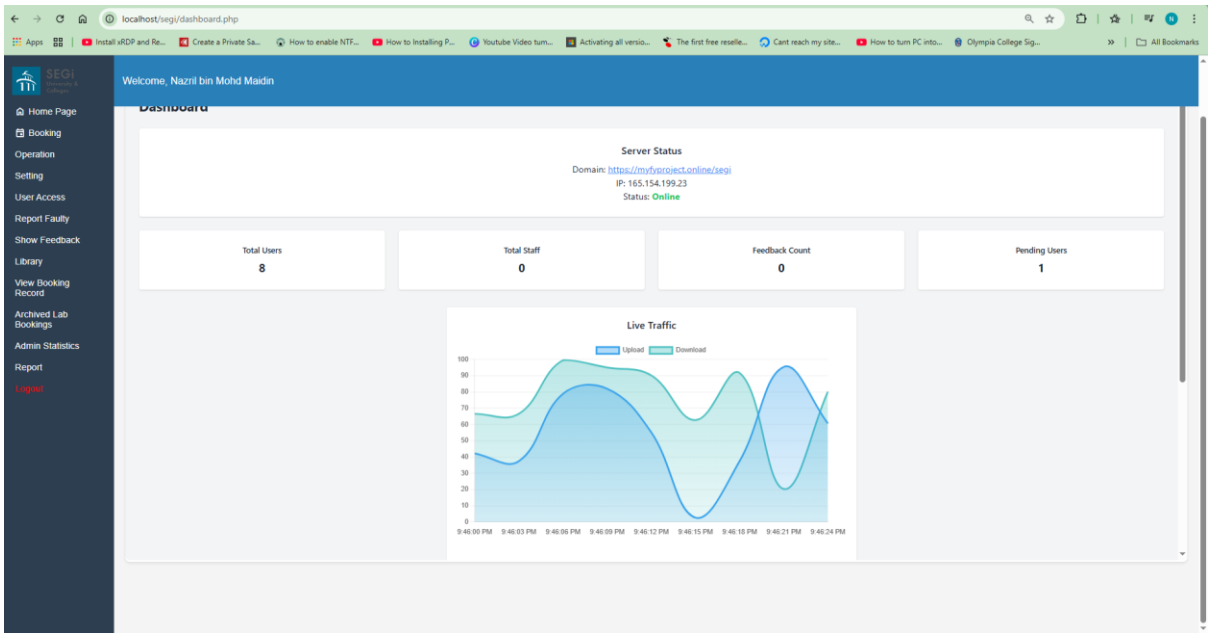



**Figure 17**

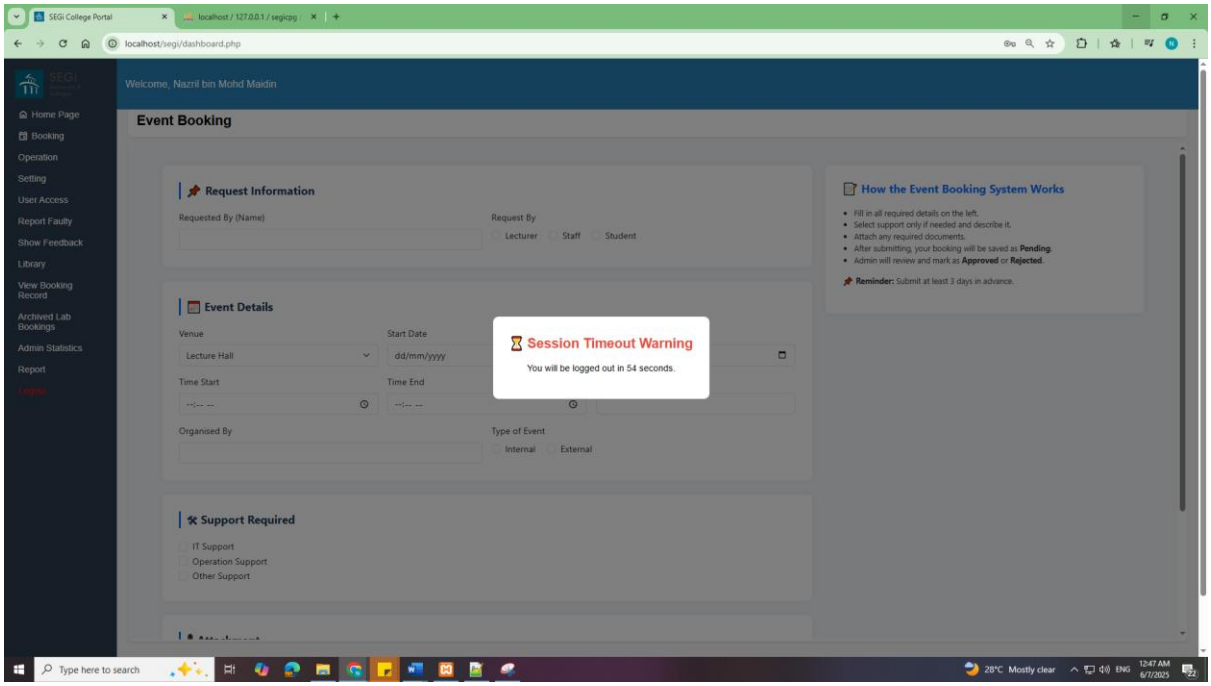

SEGi College Portal
localhost / 127.0.0.1 / segicpg /
localhost/segi/dashboard.php
Welcome, Nazril bin Mohd Maidin
Home Page
Booking
Operation
Setting
User Access
Report Faulty
Show Feedback
Library
View Booking Record
Archived Lab Bookings
Admin Statistics
Report
Logout
Event Booking
Request Information
Requested By (Name)
Request By
Lecturer
Staff
Student
How the Event Booking System Works
Fill in all required details on the left.
Select support only if needed and describe it.
Attach any required documents.
After submitting, your booking will be saved as Pending.
Admin will review and mark as Approved or Rejected.
Reminder: Submit at least 3 days in advance.
Event Details
Venue
Lecture Hall
Start Date
dd/mm/yyyy
Time Start
Time End
Session Timeout Warning
You will be logged out in 54 seconds.
Organised By
Type of Event
Internal
External
Support Required
IT Support
Operation Support
Other Support
Type here to search
28°C Mostly clear
ENG
12:47 AM
6/7/2025

**Figure 18**

# CHAPTER 6: RECOMMENDATIONS AND CONCLUSION

## 6.1 Strengths of the New System

The newly developed system incorporated an array of innovative functions that went above and beyond addressing the limitations of the previous approach by streamlining workflows and nurturing productivity. Automating processes that were once manual reduced delays and human blunders, freeing personnel for strategic priorities. Gains in efficiency were tangible as production soared.

Ease of use distinguished another edge, as the simplistic layout considered diverse technical skills. Intuitive menus logically structured an interface navigable on desktop or mobile, empowering remote work from any location. Automated alerts, real-time updates and interactive dashboards enhanced visibility into activities and performance.

Robust data management stockpiled information systematically for easy access down the line. Reporting converted raw numbers into actionable insight, guiding decisions with current reality. Strict access controls and encryption safeguarded sensitivity legally and ethically.

Futureproofing fuelled adaptability to evolving user roles and institutional architecture. Modularity streamlined updates and expansions to sustaining relevance through years of service. Pliable foundations ensured the system's value long term.

## 6.2 Disadvantages of New System

Despite that this new system has reached its main goals and offers several enhancements in comparison with the previous manual system, it has its drawbacks. On testing/enumerating: Several weaknesses of the system were found:

Limited Cross-Platform Optimization

The system is heavily optimized for use on the desktop and web browser, and may not look great on every phone or screen size. It is mobile-friendly though some parts of the interface may not function great on small screens.

Dependency on Internet Connection

Being a web based system, it needs the internet to be available all the time. Some users in low-bandwidth environments may see sluggishness or be blocked from entering the system.

No Advanced Security Features

Although it enforces basic authentification and session management, it does not offer advanced security related functionalities like two-factor authentification (2FA), SSL/TLS encryption by default or intrusion detection capabilities. This may make it a bit less secure from more advanced attacks, if the server is not properly secured.

Manual Data Backup Process

Currently. there is no automatlc back-up for information in the system. It is required that all database backups are manually initiated by the operator, which may result in loss of data in the event of operation system crashes or operator error.

No Notification System

At present there is no embedded email or SMS-based notification system to let users know when new activity on their account, booking approvals or updates have taken place. This can potentially impact on new levels of utilisation and awareness of applications.

Scalability Limitations

The model has been tested for a few number of simultaneous users and data. With more and more usage there will be performance limitations unless the server resources are increased or the database tuned for higher traffic.

No Audit Trail or Logs

Users' system activities are not well recorded, making it hard for administrators to track the changes, or investigate disputes. An audit trail function might improve accountability and transparency.

Limited Customization for Users

Users to date have a very minimal ability to personalize their profiles or preferences, such as pick a favorite language, or theme. Better user personalization in such systems can improve overall users' experiences.

These limitations show potential future improvement of the system with higher robustness, usability, and security.

### 6.3 Recommendations for Future Improvements

In view of the pros and cons of the current system, several recommendations are made for the future development of the system, which can be used to maintain the evolution and collective utility of the process in any rapidly changing environment for various users of the system. These enhancements are to improve functioning, usability, security, scalability and the ultimate value of the organization.

First and foremost, advanced analytics and business intelligence capabilities should be designed and deployed. With the existing functions of reporting feature, the available contributions are mainly focuses on integrating the advanced analytical tools, e.g., predictive analysis, trends detection, so that managers can predict, identify, and become more active in their decision-making process. Custom widgets and visualizations such as heatmaps, graphs and charts in Dashboards would further increase the visibility of organizational and resource performance.

The next field of development is the introduction of artificial intelligence (AI) and automation ability. AI based chatbots or virtual assistants could be integrated into the system for immediate user help, navigational help or answering frequently asked questions. Workflow automation rules could even be so-called that told the system to perform some action under a set of conditions, e.g. automatically send reminders or escalate overdue tasks, which would take some of the work away from human staff and make the system more efficient.

To address the lack of a good external integrations, a full API ecosystem should be created that will enable developers from third-party systems to create keys and plugins for HN system. This would enable you to connect to services such as cloud storage services, project management systems, and communications channels, allowing to bring your processes into alignment with the tools and processes you use elsewhere in your digital life.

Apart from technological enhancements a feedback mechanism and continuous improvement is also a good idea to implement. User sentiment could be routinely gauged through surveys and focus groups to ascertain opinions, pain points, and suggestions. This feedback loop would help to keep the system on track with what users expect and change as needs evolve.

Another key suggestion is to improve the personalization and accessibility of the software. Let users make things how they want by allowing them to customize the interface (add/change dashboards, change font sizes, switch themes), etc. The accessibility standard, and this is the finalized criteria, will help make the system more inclusive for people with disabilities, people can't offload trouble to somebody else like a screen reader a keyboarder etc.

There’s also a call to further enhance the system’s disaster recovery and business continuity functions. A more holistic, the disaster recovery plan could also be enforced — including having a mirrored server in a different location, perhaps even a different geography, and measurable RTO (recover time objective) / RPO (recovery point objective) for the organization to be recovered from in times of unexpected outage.

In the security category, there will perhaps be progress to features like multi-factor authentication (MFA), an algorithm that alerts to detected anomalies, and workflows that generated a log for auditing. Promoting user cyber security awareness practice can also be effective in mitigating risks associated with human negligence or social engineering attack.

Second, since we have the possibility of the organization growth, we suggest the develop system to be design and test in the aspect of horizontal and vertical scalability. The system should continue to operate efficiently and effectively as both the number of users and amount of data processed grow. This may mean leaning on cloud-native architectures, containerization technologies, or distributed databases in order to manage crushing loads more easily.

Finally, continued investment in training and capacity development for system administrators and users will ensure that the system is well-maintained and fully exploited. Regular workshops, knowledge sharing and updated documentation will allow users to self-serv and enabling them to delve into more advanced features with confidence.

In conclusion, the algorithms presented provide a safeguard for ensuring the system remains intelligent, responsive, secure, and responsive to a constantly changing operating space. Through staying open to new ideas, listening to its users and keeping its focus on constant improvement, the organization can feel confident that the platform continues to play a critical and indispensable role in its success well into the future.

## 6.4 Conclusion

In conclusion, the development and deployment of the new system constitutes the most recent step in the process of modernizing the company's operations and bringing it into line with current best practices. In practice, this system has been a useful tool to increase efficiency, accuracy, transparency and user satisfaction. By reducing manual operations and creating efficient digital operations, while delivering accurate and timely data, it enables users to perform better in their function and the ability for management to make smarter decisions.

While such a system has operated with significant merits in terms of system design, functionality, and utilization, the limitations of the system also need to be acknowledged and addressed. Connectivity dependence, scalability, integration gaps, and continuous security maintenance challenges need to be addressed proactively so that the system remains successful throughout its lifetime. The recommendations specified in this chapter, will serve as a itinerary in which continuous grwoth will be organized, where the system does end up the same and that its friendly continues, safe and responds to new organizational requirements.

In the final analysis, the evolution of this system has demonstrated the power of technology when it is thoughtfully linked to the end user and to the organization. By continuing to reinvest in the enhancements, training, and the ongoing upkeep, the system becomes a foundation for not just what you're doing today, but everything you might do in the future. The learning and success experienced In this venture have once again established the need to maintain a philosophy of kaizen and agility in an ever-changing technological world.

Reference

1. https://nextjs.org/docs/app/getting-started/fetching-data

2. https://www.dbvis.com/thetable/sql-fetch-retrieving-data-in-database-cursors/

3. https://www.freecodecamp.org/news/creating-html-forms/

4. https://www.w3schools.com/php/php_forms.asp

5. https://www.phptutorial.net/php-tutorial/php-form/

6. https://codeshack.io/secure-login-system-php-mysql/

7. https://dev.to/omacys/ensuring-secure-user-sessions-a-guide-to-logging-out-users-due-to-inactivity-in-php-3167

8. https://www.simplilearn.com/tutorials/php-tutorial/php-login-form

9. http://webvaultwiki.com.au/Default.aspx?Page=Create-Mysql-Database-User-Phpmyadmin&NS=&AspxAutoDetectCookieSupport=1

10. https://locallhost.me/phpmyadmin

11. https://www.phpmyadmin.net/news/2023/2/8/phpmyadmin-4911-and-521-are-released/

12. https://localhosts.mobi/phpmyadmin

13. https://stackoverflow.com/questions/38634441/simple-yes-or-no-checkbox-for-forms

14. https://html.form.guide/php-form/php-form-checkbox/

15. https://www.w3schools.com/tags/att_input_type_checkbox.asp

16. https://getbootstrap.com/

17. https://www.cloudflare.com/en-gb/learning/cloud/what-is-the-cloud/

APPENDICES

This guide explains how to access and use the system effectively. It is intended for both end-users and administrators. Screenshots referenced are provided in Chapter 5 and Appendices.

1. Accessing the System
- Open a web browser (Google Chrome or Mozilla Firefox recommended).
- Enter the system URL **https://myfyproject.online/segi/** in the address bar.
- Wait for the login page to load.

2. Logging In
- Enter your email and password as provided.
- Click Login.
- If successful, you will be taken to your dashboard.
- If unsuccessful, an error message appears — check credentials and try again.

3. Dashboard Overview
After login, the dashboard shows:
- Quick links to main modules (e.g., bookings, report faulty, system statistics).
- Notifications of pending or approved actions.
- Your account details.

4. System Modules and Features

4.1 Booking a Facility / Room
- Go to the Booking module from the sidebar or dashboard icon.
- Select booking type (e.g., event, lab, classroom).
- Fill out all required fields (date, time, purpose, lecturer name, etc.).
- Upload any necessary attachments if applicable.
- Click Submit.
  The system will validate and save your request as Pending.

4.2 Viewing Submitted Bookings
- Navigate to My Bookings.
- See the list of your bookings and their current status (Pending, Approved, Rejected).
- Use the filter/search bar to find specific bookings.

4.3 Report Faulty
- Navigate to the Report Faulty page.
- Fill in the form specifying the type of issue, location, floor, urgency, and remarks.
- Upload a photo if needed.
- Click Submit.
  Your report will be recorded for admin action.

5. Automatic Logout
For security purposes, the system automatically logs out users after 2 minutes of inactivity.
- If no activity is detected for 2 minutes, a popup will appear notifying you that the session has expired.
- You will be redirected to the login page.
- To continue working, simply log in again.

6. Administrator Functions
For Admin accounts only:
- Approve or reject bookings and faulty reports.
- Manage users: add, edit, approve or reject accounts.
- View system usage statistics and generate PDF/reports.
- Edit the Welcome Page content if required.
- Monitor all submitted reports and activities.

7. Logging Out
- Always click the Logout button at the top-right corner of the dashboard when done.

8. Recommendations
- Use a stable Internet connection.
- Keep your login credentials secure.
- Use the recommended browsers for full compatibility.
- Contact the administrator if you face errors or issues.

## TECHNICAL SPECIFICATIONS

This section outlines the recommended technical requirements and specifications for deploying, accessing, and operating the system effectively. It includes client-side and server-side requirements as well as compatibility guidelines.

1. Client-Side Requirements
These are the minimum specifications for end-users accessing the system via a web browser:
- Web Browser Compatibility:
  - Google Chrome (latest version) Recommended
  - Mozilla Firefox (latest version)
  - Safari (latest version on macOS/iOS)
  - Note: Internet Explorer is not supported.
- Operating System:
  - Windows 10/11, macOS 12+, Linux (latest distributions), Android 10+, iOS 13+
- Screen Resolution:
  - Minimum: 1280×720 pixels
  - Recommended: 1920×1080 pixels or higher
- Internet Connection:
  - Stable broadband connection (minimum 2 Mbps)

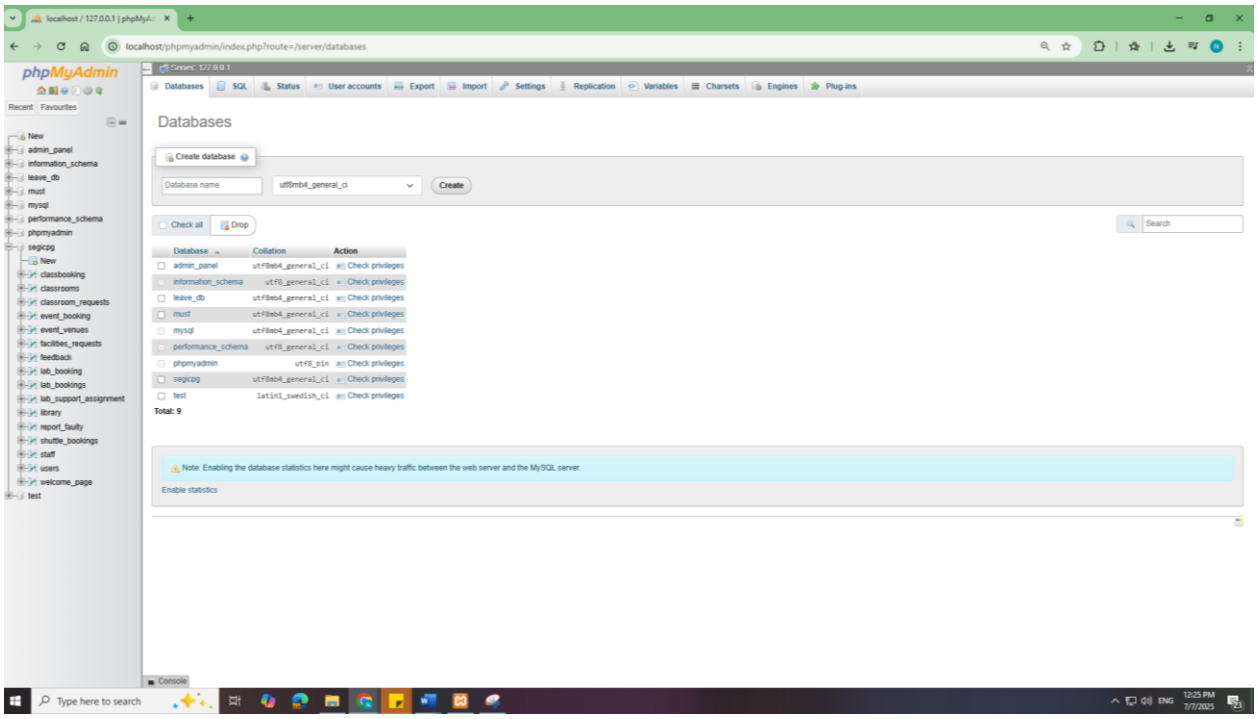
2. Server-Side Requirements
These are the recommended specifications for hosting the system on a VPS or server:
- Operating System:
  - Windows Server 2019 (VPS) Deployed
- Processor:
  - Minimum: Dual-core 2.0 GHz
  - Recommended: Quad-core 3.0 GHz or better
- Memory (RAM):
  - Minimum: 4 GB
  - Recommended: 8 GB or higher
- Storage:
  - Minimum: 50 GB free disk space
  - Recommended: SSD storage for better performance
- Web Server Stack:
  - XAMPP (Apache 2.4.x, PHP 8.x, MySQL 8.x)
  - phpMyAdmin (latest version for DB management)
- Dynamic DNS:
  - No-IP or equivalent dynamic DNS service for domain mapping

3. Recommended Software
For system development and maintenance:
- Notepad++ (code editing)
- Adobe Dreamweaver 2024 (design and development)
- Visual Studio Code (optional)
- WinSCP or FileZilla (optional for file transfers)

4. Security Recommendations
- Enable firewall and antivirus protection on the VPS.
- Install SSL/TLS certificate to enable HTTPS.
- Perform regular backups of the MySQL database.

These specifications ensure smooth and secure operation of the system, as well as user accessibility across commonly used platforms.

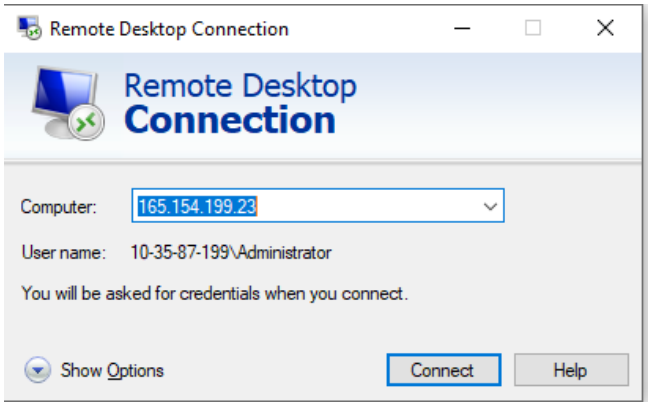
**phpMyAdmin Database**



Guide to connect to the VPS and start using the system.

Step 1: Remote Desktop Login
- Open Remote Desktop Connection on your computer.
- Enter the VPS IP address and click Connect.
- When prompted, enter the following credentials:

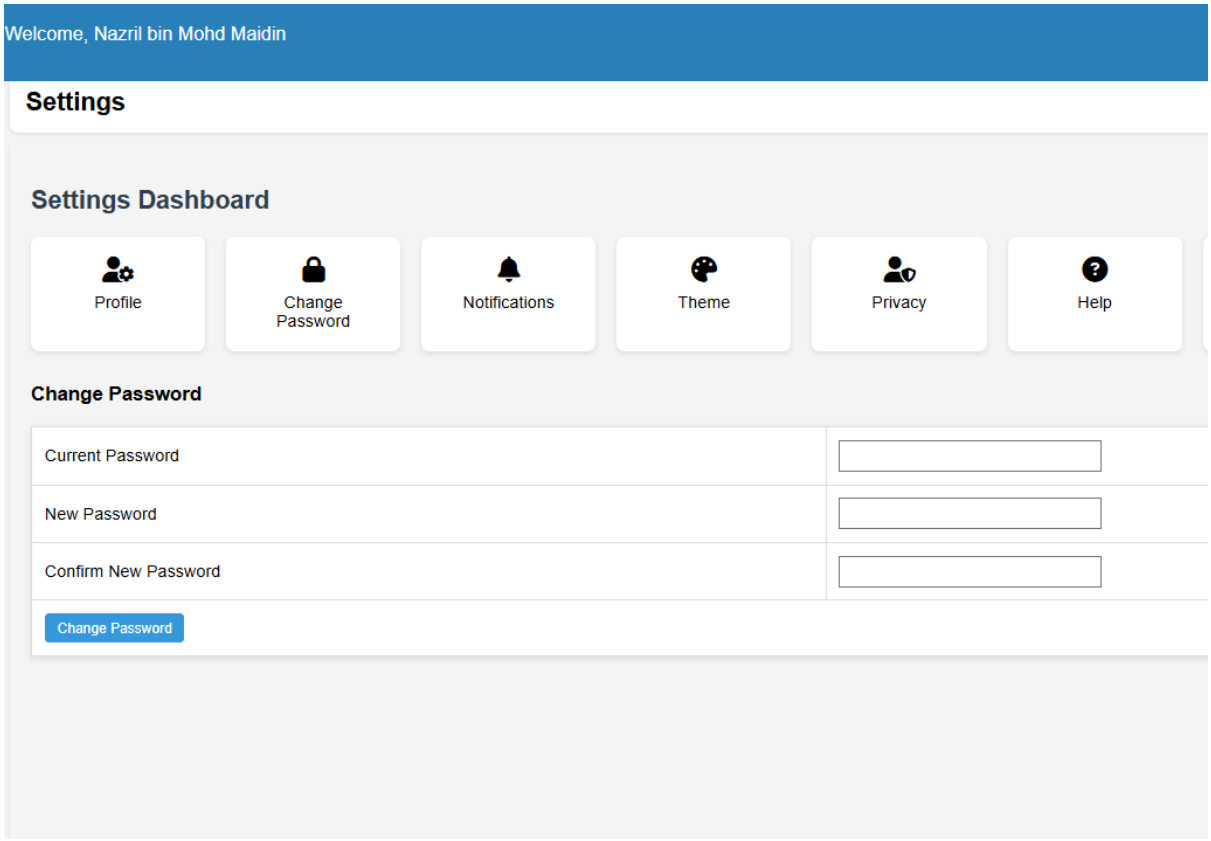
Username: Administrator
Password: Rum@hbaru25*



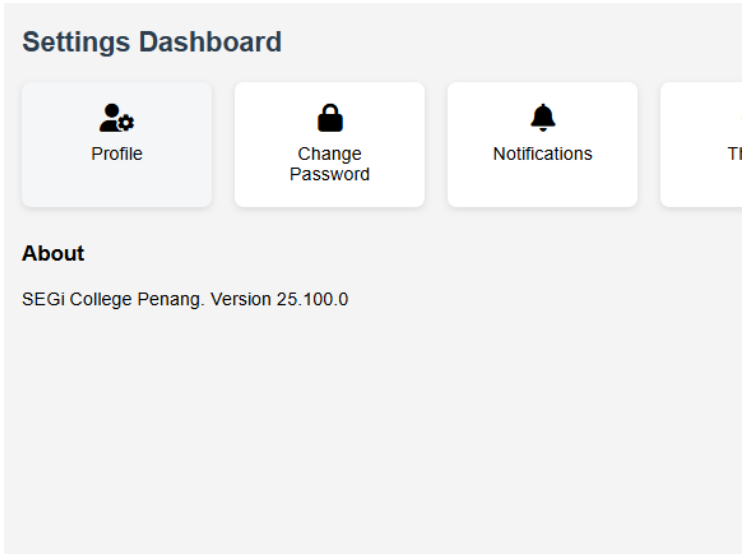

Step 2: Start the Server
- On the VPS desktop, open XAMPP Control Panel.
- Start Apache and MySQL (these may auto-start).

Step 3: Open and Use the System
- Open a web browser on the VPS or on your own computer.
- Enter the system URL (IP or domain name) in the browser.
- The system is accessible from both public (Internet) and local network.

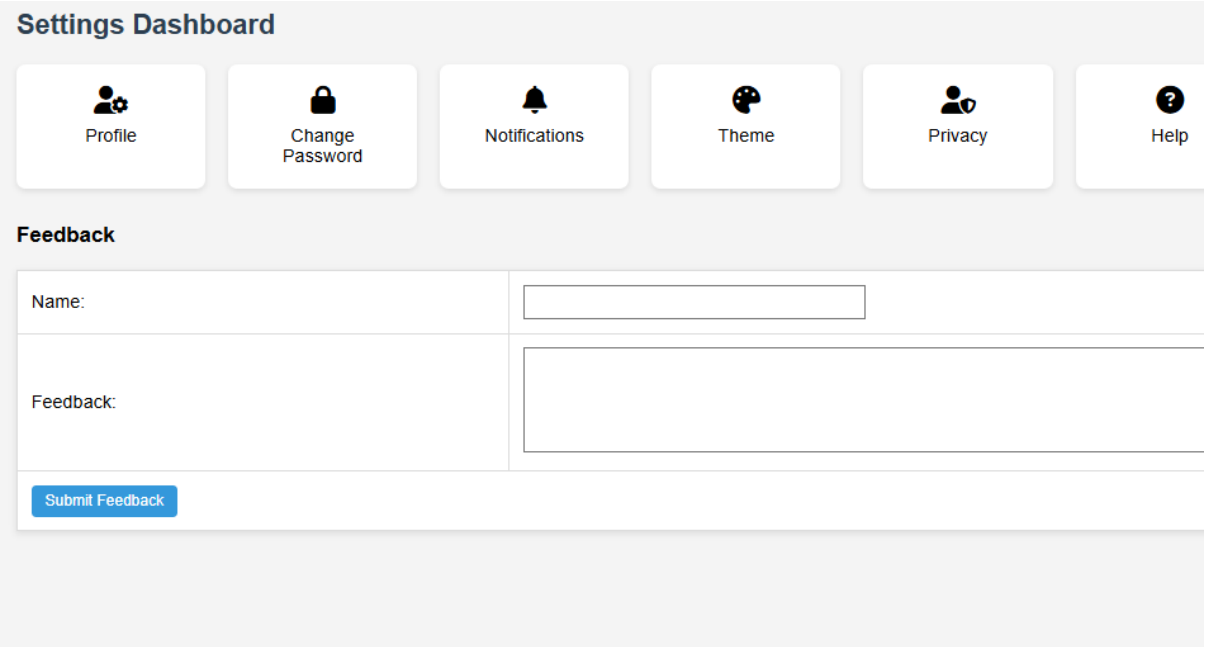
location file on server



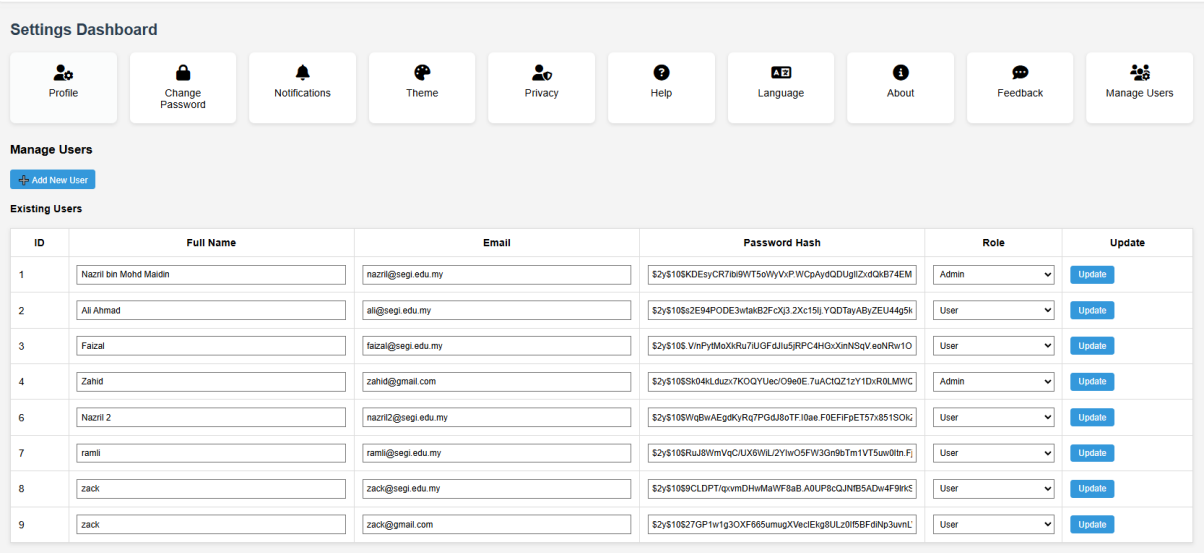

User Password Change



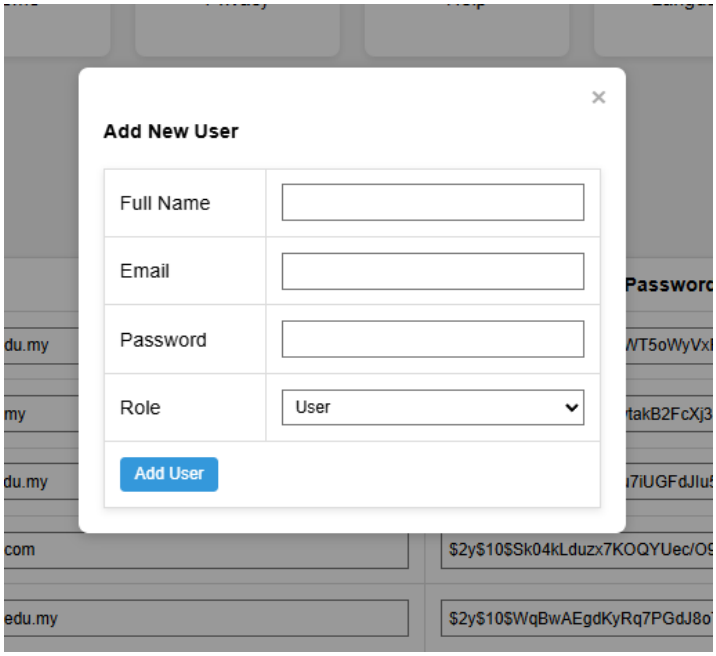

System Version




Admin can update user feedback




manage user full name,email,password and role.




Add new user