TulaBy Project

Supervisor:

Rania Ramadan

Faculty of Computer and Artificial Intelligence at Sohag University

January , 2024

Project Team Members

Mostafa Maher Ahmed

Role: Front-End Developer

Hadeer Elsayed Ahmed

Role: Back-End Developer

Janet Gad Elseed Attia

Role: UI/UX Designer

Mohamed Ahmed Noureldeen

Role: Front-End Developer

Mostafa Adel ElSayed

Role: Back-End Developer

Hesham Mohamed Ahmed

Role: Mobile-Developer

Mostafa Abdelhakeem Mohamed

Role: Quality Assurance Engineer

Fouad Tharwat Hanna

Role: Mobile-Developer

Acknowledgements

We express our deep gratitude and appreciation to those who agreed to participate in this project, for their time expended and courage in sharing their insights with a fledgling student. It is to them that we are most indebted, and we can only hope that the product of our collaboration benefits each one as much as we benefited from the process.

First, we must thank **Prof. Elnomery Alam zanaty** faculty Dean for facilitating the educational process throughout the last 4 years and making sure for providing all necessary tangible and mental basics needed for our success.

second,we want to thank **Dr. Rania Ramadan** for her support and guidance throughout the course of our project.

AOT	Ahead-of-time
Css	Cascading Style Sheets
DFD	Data Flow Diagram
Dom	Document Object Model
ERD	Entity Relationship Diagram
HTML	Hypertext Markup Language
JIT	Just-In-Time
JSX	javaScript XML
OTP	One time password
RDBMS	Relational Database Management System
PDF	Portable Document Format
SQL	Structured Query Language
UI	User Interface
UML	Unified Modeling Language
XML	Extensible Markup Language

Abstract

TuLaBy is a complete solution for dealing with the challenges that administrators face when it comes to keeping track of attendance and participation in lectures and labs. The platform improves communication between students, teachers, assistant teachers, and college administrators. **TuLaBy** allows users to easily stay updated with the latest news and announcements from the college, making communication better. Students can simply log when they attend or are absent, and teachers and assistant teachers can efficiently keep track and update attendance records, including bonuses.

The system makes it easier for teachers to assign lectures and for assistant teachers to manage labs, along with registering students for their chosen courses. **TuLaBy** provides accurate reports on student attendance and performance, giving teachers, assistant teachers, and administrators the information they need to make informed decisions.

Built using modern technologies like ReactJS for the web application, PHP-Laravel for the back-end, MySQL for the database, and Flutter for the mobile application, the platform ensures good performance and an easy-to-use interface. **TuLaBy** promises an easy and user-friendly experience for college students, teachers, and assistant teachers, encouraging interaction and communication.

With these features, **TuLaBy** provides a complete solution for improving attendance and participation management, making the overall experience better for students and faculty in the college environment.

Contents

List of Figures

List of Tables

Part I

Theoretical

Chapter 1

Introduction

1.1 Overview:

The **TuLaBy** project is a comprehensive solution designed to meet the specific needs of students, professors, teaching assistants, and administrators within academic institutions. It aims to revolutionize traditional methods of attendance recording, information access, and administrative tasks by providing a centralized and efficient platform. The project focuses on enhancing user experience, improving communication channels, and streamlining administrative processes to create a more conducive learning environment.

1.2 About TuLaBy Project:

TuLaBy, which stands for "Technology for Unified Learning and Administration", represents a holistic approach to academic management. The project encompasses a wide range of features tailored to meet the diverse needs of users, including dynamic attendance recording, personal information management, and real-time access to academic data. **TuLaBy** integrates seamlessly with existing academic systems and technologies, providing a user-friendly interface for students and faculty members.

1.3 Challenges We Have:

Academic institutions face various challenges, including manual attendance recording, disparate information systems, and time-consuming administrative tasks. Traditional attendance management methods are prone to errors and inefficiencies, leading to discrepancies in student records and grades. Limited communication channels between students, professors, and administrators hinder collaboration and information sharing, resulting in disjointed academic experiences.

1.4 TuLaBy Motivation:

The motivation behind **TuLaBy** is to address these challenges and create a more efficient and cohesive academic environment. The project aims to streamline attendance

management processes, provide valuable insights into student attendance patterns, and empower users with timely information and communication tools. **TuLaBy** seeks to enhance academic management practices and improve overall educational outcomes by leveraging technology and innovation.

Unified Platform and Device Integration: Imagine a seamless experience across web and mobile applications, providing users with a unified platform for effortless access and management.

Technological Advancements and User-Centric Design: Web and mobile applications have undergone continuous improvements since the advent of devices. Our platform exemplifies these advancements by serving as a creative tool that not only redefines the purpose of devices but also ensures technology remains accessible and user-friendly. Developers continually strive to enhance these applications, delivering the most convenient functionalities for users to simplify and enjoy their tasks.

Precision, High-Quality, and Future-Ready Flexibility: TuLaBy is crafted with precision and high quality, providing a flexible framework for future feature additions without modifying existing functionalities. The platform's adaptability ensures smooth integration of upcoming enhancements without altering the core aspects that have already been perfected.

1.5   TuLaBy Contribution:

This project is meticulously designed to meet the specific needs of students, professors, teaching assistants, and administrators, providing tailored benefits to each user group.

Student-Centric Features: Students enjoy the convenience of dynamically recording attendance for lectures and labs, eliminating the need for manual intervention by instructors. The platform offers an intuitive interface for students to access and manage their personal information, view details about upcoming lectures and labs, and keep track of attendance records and grades.

Professorial Efficiency: Professors benefit from streamlined attendance management, allowing them to effortlessly record attendance for their lectures. The system empowers professors with comprehensive insights into student attendance, enabling them to switch attendance states and assign bonus marks efficiently. Professors can finalize grades for students enrolled in the subjects they teach and receive timely information about upcoming lectures.

Teaching Assistant's Role: Teaching assistants play a vital role in teaching specific subjects and efficiently managing attendance for the associated labs. Their responsibilities include recording attendance, managing attendance properties, and staying informed about upcoming labs they are designated to conduct.

Administrative Control and System Management: Administrators exercise comprehensive control over the system, facilitating the efficient addition and management of students, professors, and teaching assistants. The administrative dashboard simplifies tasks such as adding subjects, managing professors and teaching assistants, and allocating lectures and labs. Administrators can post critical news and announcements, ensuring all users stay informed.

1.6   Outline of the Documentation:

The documentation provides a detailed overview of **TuLaBy**'s objectives, features, and implementation strategies. It includes user guides, tutorials, and best practices for maximizing the benefits of **TuLaBy** within academic settings. Additionally, the documentation outlines future enhancements and developments planned for **TuLaBy**, ensuring its continued relevance and effectiveness in the rapidly evolving educational landscape.

Chapter 2

Problem Description

2.1 Overview

Many colleges and universities face challenges in efficiently monitoring student attendance and absences during lectures and labs. Despite technological progress, Many educational institutions still adhere to outdated paper-based systems, requiring manual attendance tracking. This traditional method raises various problems:

- **Time and Effort Drain:** Traditional attendance systems demand the physical presence of students, resulting in significant time wastage as each student's name is individually called for attendance registration. This not only consumes valuable time but also places a substantial burden on instructors, contributing to the overall time and effort drain during every study session.

- **Potential for Errors:** Manual registration increases the chance of mistakes, both in recording attendance and in carefully organizing paper-based data.

- **Management Challenges:** Professors and teaching assistants grapple with the management of substantial data volumes manually when using a paper-based system.

- **Loss of Information:** Relying on paper often leads to information loss, either due to physical misplacement or challenges in accessibility.

In response to this challenge, our project introduces an intelligent and modern system that simplifies attendance and absence tracking processes. This initiative aims to

enhance the educational experience for both students and instructors by replacing traditional methods with a more efficient and accurate approach, fostering student engagement and overall educational improvement.

2.2 Solution

2.2.1 Introduction of the New Feature:

To elevate the learning experience and streamline attendance and absence management processes, the **TuLaBy** project introduces an exceptional feature designed to simplify the lives of teachers and students while enhancing the efficiency of teaching operations.

New Feature: Unified Tracking of Attendance and Absences for All Courses in One Place

With the integration of this feature, administrators and teachers can effortlessly track attendance and absences for all courses from a centralized location, reducing complexity and saving a considerable amount of effort and time. The system allows instant and seamless access to attendance and absence information for each student and teacher.

2.2.2 Benefits of This Feature:

1. **Time and Effort Savings:** Simplifies the tracking of attendance and absences for courses, minimizing effort in each study session.

2. **Greater Accuracy:** Diminishes the likelihood of manual errors and ensures higher accuracy in recording attendance and absences.

3. **Flexible Management:** Facilitates individual management of attendance and absences for each course, accommodating the unique needs of each teacher and student.

4. **Improved Access and Visibility:** Offers immediate access to attendance and absence information, enhancing communication and improving data clarity.

5. **Modern Technology:** Leverages cutting-edge technologies for user-friendly interactions and compatibility with technological advancements.

2.2.3 Secure Data Retention:

The TuLaBy project prioritizes data security, adhering to the highest standards to ensure confidentiality and integrity.

1. **Secure Backups:** Regular backups are systematically created and securely stored.

2. **Access Levels:** Organized access levels guarantee privacy and security.

3. **Protection Against Electronic Threats:** Incorporates modern techniques to safeguard the system from electronic threats.

4. **Compliance with Privacy Legislation: TuLaBy** complies with all legislation and regulations about privacy and personal data protection.

5. **Security Updates:** Regular reviews and updates of security procedures ensure alignment with the latest developments in information security.

These comprehensive security measures reflect TuLaBy's commitment to ensuring the safety and integrity of data, establishing a dependable and secure environment for educational information. The addition of this feature aims to enhance the educational process and simplify the lives of all stakeholders in the university environment.

Chapter 3

System Analysis and Design

3.1 Overview

System analysis and design is a structured process of examining, modeling, and improving an organization's information systems to ensure they meet its needs effectively and efficiently. It involves studying and understanding the existing system, identifying areas for improvement, and designing a new system or making enhancements to the current system to address those needs.

System analysis and design typically includes the following key activities:

1. **Requirements gathering:** Understanding and documenting the needs and objectives of the organization and its stakeholders.

2. **System modeling:** Creating visual representations (e.g., data flow diagrams, entity-relationship diagrams) of the current and proposed systems to analyze and communicate the system's structure and processes.

3. **System design:** Developing detailed specifications for the new or enhanced system, including software, hardware, databases, interfaces, and user interfaces.

4. **Implementation:** Building, testing, and deploying the new system or system improvements.

5. **Maintenance and evaluation:** Monitoring and maintaining the system to ensure it continues to meet the organization's needs, as well as evaluating its performance and making further improvements as necessary.

System analysis and design aims to improve business processes, increase efficiency, reduce costs, enhance decision-making capabilities, and support organizational goals through the effective use of technology and information

systems. It is a crucial aspect of information technology and plays a vital role in ensuring that systems align with the organization's strategic objectives.

3.2 Scenario: Introducing TuLaBy - Student Attendance Management Platform

In the digital landscape of higher education, **TuLaBy** emerges as a a comprehensive platform dedicated to managing student attendance within universities. Tailored to meet the diverse needs of academic institutions, **TuLaBy** streamlines attendance tracking and Management processes, ensuring efficient oversight of student participation across various academic endeavors.

Platform Functionality:-

- **Database Architecture: TuLaBy**'s database architecture is meticulously designed to accommodate the complex ecosystem of student attendance Management. It encompasses a multitude of entities including students, professors, lectures, sessions, groups, and instructors, each with unique roles and responsibilities within the educational framework.

- **Student Profile Management:** Students registered on **TuLaBy** provide detailed personal information including identification code, Name, Personal Identity Kind, Nationality, Religion, Gender, birth, Picture, Academic Email, and contact details(Email, Address, Phone), and academic levels. This comprehensive profile system facilitates targeted communication and personalized support services for students.

- **Professor and Instructors Integration: TuLaBy** seamlessly integrates professors and Instructors into its platform, enabling efficient allocation of subjects(Id, number of lectures, number of hours, level, and Title), groups, lectures, and sessions. Professor profiles capture essential information such as identification code, Name, Personal Identity, educational levels, Personal Identity Kind, Nationality, Religion, Gender, Birth, Picture, Academic Email, and contact details(Email, Address, Phone), empowering instructors to engage with students and oversee actively academic activities.

- **Lecture and Session Tracking:** Lectures(Code, Date, Tilte, Active) and sessions(Code, Date, Tilte, Active) form the cornerstone of the academic experience within **TuLaBy**'s ecosystem. Lectures are assigned to specific subjects, with each lecture contributing to the holistic learning journey of students. Sessions, conducted by instructors, provide additional learning opportunities in Groups and are meticulously tracked to ensure comprehensive attendance recording.

- **Group Allocation and Registration: TuLaBy** facilitate the seamless allocation of students to various academic groups, streamlining communication and administrative processes within the university ecosystem. Group details, including title, and Level.

- Professors teach multiple subjects, and each subject can be taught by multiple professors. Lectures are given by professors, with each professor delivering multiple lectures. Every lecture is associated with one subject, and each subject may have several lectures.

- Students enroll in specific subjects within a group and track total marks, attend lectures, and monitor bonuses and attendance for their enrolled subjects.

- Instructors teach various subjects to different groups, and a subject can be taught to multiple groups by different instructors. Each group is assigned specific subjects taught by specific instructors.

- Sessions are allocated to specific subjects and groups, and instructors conduct these sessions. Each session is conducted by one instructor. Students attend multiple sessions and monitor bonuses and attendance for each session they attend.

- **Attendance Recording and Bonus Allocation:** Both professors and instructors leverage **TuLaBy**'s robust attendance recording system and bonus to track student participation during lectures and sessions. Active engagement is incentivized through bonus allocation, fostering a culture of accountability and academic excellence within the university community.

As universities embrace digital transformation, TuLaBy stands at the forefront of innovation, empowering institutions to redefine the student experience and foster a culture of continuous learning and growth. Through its intuitive interface and comprehensive feature set, TuLaBy revolutionizes student attendance

Management, paving the way for enhanced engagement, accountability, and academic success.
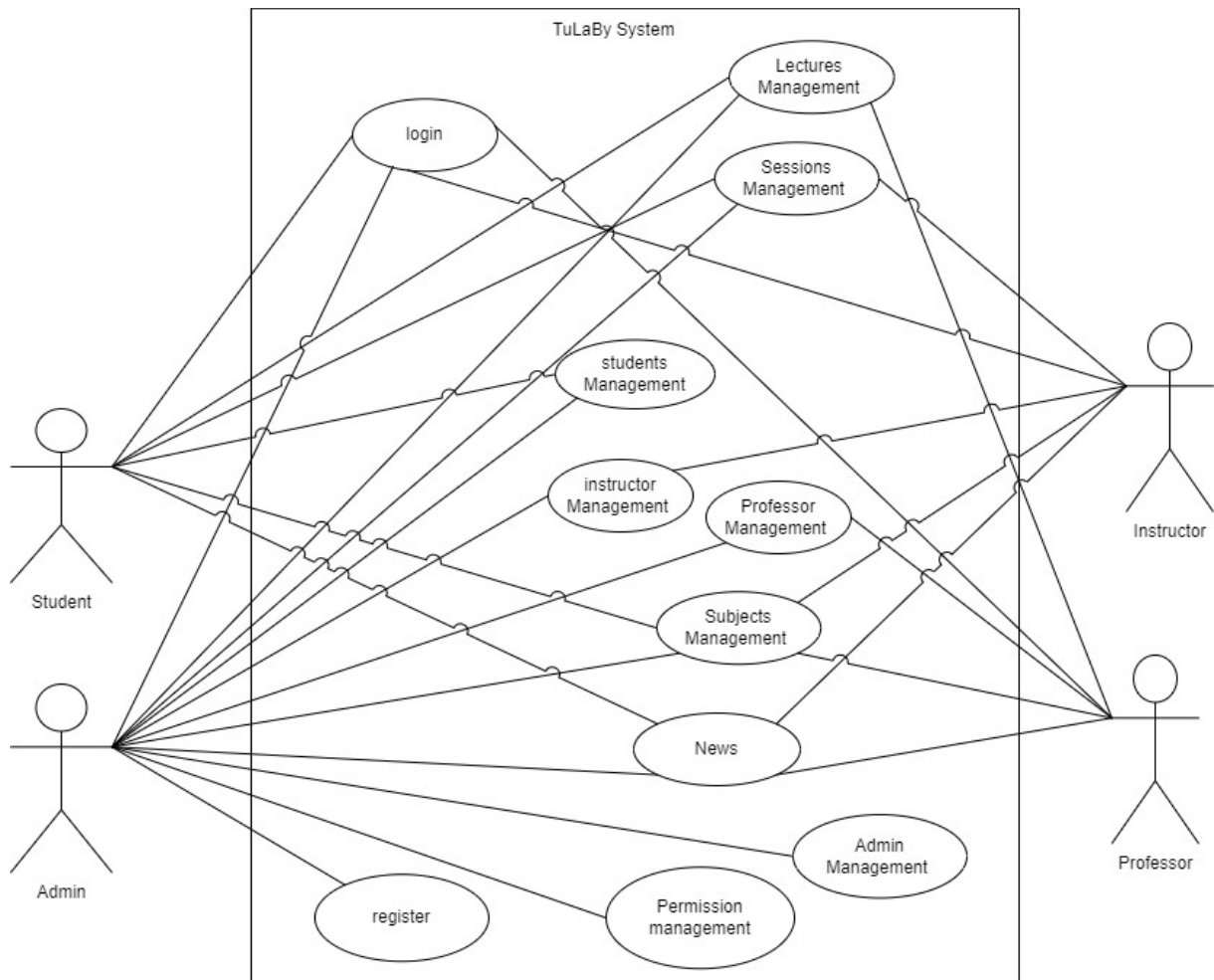
3.3　System Analysis

System analysis examines an organization's current information systems to understand how they work and identify areas for improvement. It involves studying the existing system's components, processes, and interactions to determine its strengths and weaknesses. The goal of system analysis is to gather and analyze information about the organization's needs, objectives, and requirements to propose solutions for enhancing or replacing the current system. This process typically involves gathering user requirements, modeling system components and processes, and identifying opportunities for improvement through the use of technology and information systems. System analysis is a critical step in the overall system development life cycle and is essential for ensuring that new or enhanced systems effectively meet the needs of the organization and its stakeholders.

3.3.1　Use Case Diagrams

Use Case Diagrams are a type of behavioral UML diagram that depicts the interactions between a system and its users or other systems. They illustrate the functionality of a system by showing the relationships between actors (users, systems, or external entities) and use cases (specific tasks or functions performed by the system). Use case diagrams help to identify and define the system's requirements and provide a high-level view of the system's functionality.

Use Case Diagrams Components:

- Actor

- Use cases (Goals)

- System

- Relationships

Figure 3.3.1: Use Case Diagram.

3.3.2 Interface To Describe Use Case Diagram

System	Student Management system
Use case	login
actor	student
data	After clicking on the login button. The student should insert his code and Password. If the student inserts valid data, the login will be completed and the student will access the application. If the student inserts an invalid code an error message will appear asking to enter a valid code. If the student inserts an invalid Password, an error message will appear asking to enter a valid Password.

Table 3.3.1: login use case for students.

System	Student Management system
Use case	login
actor	instructor
data	After clicking on the login button. The instructor should insert his code and Password. If the instructor inserts valid data the login will be completed and the instructor will access the application. If the instructor inserts an invalid code, an error message will appear asking to enter a valid code. If the instructor inserts an invalid Password, an error message will appear asking to enter a valid Password.

Table 3.3.2: Login use case for instructor

System	Student Management system
Use case	login
actor	Professor
data	After clicking on the login button. The Professor should insert his code and Password. If the Professor inserts valid data the login will be completed and the Professor will access the application. If the Professor inserts an invalid code, an error message will appear asking to enter a valid code. If the Professor inserts an invalid Password, an error message will appear asking to enter a valid Password.

Table 3.3.3: login use case for Professor.

System	Student Management system
Use case	login
actor	admin
data	After clicking on the login button. The admin should insert his email and password. If the admin inserts valid data the login will be completed and the admin will access the application. If the admin inserts an invalid email, an error message will appear asking to enter a valid email. If the admin inserts an invalid password, an error message will appear asking to enter the Valid Password.

Table 3.3.4: login use case for admin

System	Student Management system
Use case	Lecture Management
actor	admin
data	After correct login, the admin manages Lectures such as (adding, deleting, update) all data related to the lecture.

Table 3.3.5: Lecture Management use case for admin

System	Student Management system
Use case	Subjects Management
actor	Admin
data	Admin can manage Subjects (Add, Delete, Update).

Table 3.3.6: Subjects Management use case for Admin.

System	Student Management system
Use case	Subjects Management
actor	Instructor
data	Instructor Can Teach Many Subjects

Table 3.3.7: Subjects Management use case for Instructor.

System	Student Management system
Use case	Subjects Management
actor	Student
data	Students can Assign To Subjects.

Table 3.3.8: Subjects Management use case for students.

System	Student Management system
Use case	Subjects Management
actor	Professor
data	Professor Can Teach Many Subjects.

Table 3.3.9: Subjects Management use case for Professor.

System	Student Management system
Use case	Lecture Management
actor	student
data	After the correct login, the student can manage lectures by recording his Attendance for lectures.

Table 3.3.10: Lecture Management use case for students.

System	Student Management system
Use case	Sessions Management
actor	student
data	After the correct login, the student can manage sessions by recording his Attendance for sections.

Table 3.3.11: Sessions Management use case for students.

System	Student Management system
Use case	Student Management
actor	student Management
data	After the correct login, the student can manage his profile.

Table 3.3.12: Student Manage use case for students.

System	Student Management system
Use case	Lecture Management
actor	Professor
data	After correct login, the Professor manages Lectures by recording student Attendance and tracking student participation during lectures.

Table 3.3.13: Lecture Management use case for Professor.

System	Student Management system
Use case	Professor Management
actor	Professor
data	After the correct login, the Professor can manage His profile.

Table 3.3.14: Professor Management use case for Professor

System	Student Management system
Use case	Sessions Management
actor	instructor
data	After correct login, the instructor manages sessions by recording student Attendance and tracking student participation during sessions.

Table 3.3.15: Sessions Management use case for instructor

System	Student Management system
Use case	News
actor	Professor
data	The Professor can see the all news.

Table 3.3.16: News Management use case for Professor

System	Student Management system
Use case	Instructor Management
actor	instructor
data	After the correct login, the instructor can manage the His Profile

Table 3.3.17: Instructor Management use case for instructor

System	Student Management system
Use case	News
actor	instructor
data	The Instructor can see all the news.

Table 3.3.18: News use case for instructor

System	Student Management system
Use case	Register
actor	admin
data	After clicking on the register button admin should insert The Identification information For The New User. And data will be saved in the database.

Table 3.3.19: Register use case for admin

System	Student Management system
Use case	News
actor	student
data	Students can see all the news.

Table 3.3.20: News use case for students.

System	Student Management system
Use case	Student Management
actor	admin
data	After the correct login, the admin can manage the student (add, delete, edit).

Table 3.3.21: Student Management use case for admin

System	Student Management system
Use case	Professor Management
actor	admin
data	After the correct login, the admin can manage the Professor (add, delete, edit).

Table 3.3.22: Professor Management use case for admin

System	Student Management system
Use case	Instructor Management
actor	admin
data	After the correct login, the admin can manage the Instructor (add, delete, edit).

Table 3.3.23: Instructor Management use case for admin

System	Student Management system
Use case	News
actor	admin
data	After the correct login, the admin can manage News (publish, edit, delete)

Table 3.3.24: News use case for admin

System	Student Management system
Use case	Permission Management
actor	admin
data	After the correct login, the admin can manage permission(Grant, Deny) For students, professors, Instructors, or other users.

Table 3.3.25: Permission Management use case for admin

System	Student Management system
Use case	Sessions Management
actor	admin
data	After correct login, the admin manages sessions such as (adding, deleting, update) all data related to sessions.

Table 3.3.26: Sessions Management use case for admin

System	Student Management system
Use case	Admin Management
actor	admin
data	After the correct login, the admin can manage the Admins (add, delete, edit).

Table 3.3.27: Admins Management use case for admin

3.3.3 Data Flow Diagrams

Data flow diagrams (DFDs) are a type of visual representation that illustrates the data flow within a system. They are used to model the processes, data stores, and data flows within a system or organization. DFDs depict how data moves from input to processing to output and can be used to represent the flow of information within a business process or software system. They are often used in the analysis and design of information systems to understand and document the flow of data and the processes that manipulate it.

Data flow diagram Components:-

- Processes

- Data Flows

- Datastore

- External entities

3.3.4 Levels of Data Flow Diagrams

3.3.5 Context Level:

The context level of a data flow diagram (DFD) provides an overview of the entire system or organization, showing the interactions between the system and its external entities. It represents the highest level of abstraction and serves as a starting point for understanding the flow of data in the system. The context level DFD typically only shows a single process representing the entire system, along with external entities such as users, other systems, or data sources, and the data flows between them.

In essence, the context-level DFD is a high-level representation that helps stakeholders understand how the system interacts with its environment and the overall flow of data

into and out of the system. It does not go into detail about the internal processes and data stores within the system but rather focuses on the interactions between the system and its external entities.
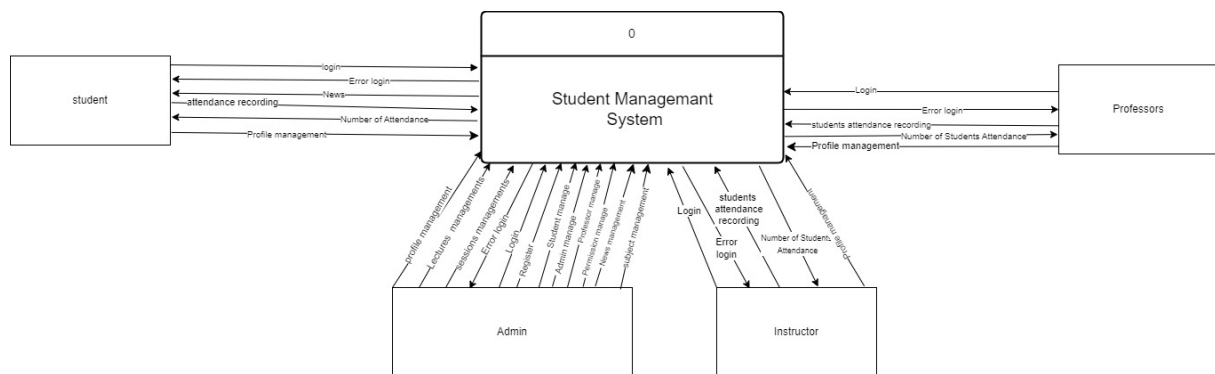

Figure 3.3.2: Data Flow Diagram: Context Level

3.3.6   Level 0:

The level 0 data flow diagram (DFD) is the next level of detail below the context level DFD. It provides a more detailed view of the system by breaking down the single process from the context level into sub-processes and showing the data flows between them. At level 0 DFD, the system is represented as a single process and the external entities are shown interacting with this process through data flows. The Level 0 DFD typically includes multiple processes, each representing a specific function or sub-process within the system. It also shows the data stores that hold and maintain data within the system, as well as the data flows between processes, data stores, and external entities.

In summary, the level 0 DFD provides a more detailed and structured view of the system's processes and data flows compared to the context-level DFD, making it a valuable tool for understanding the overall structure and flow of data within the system.
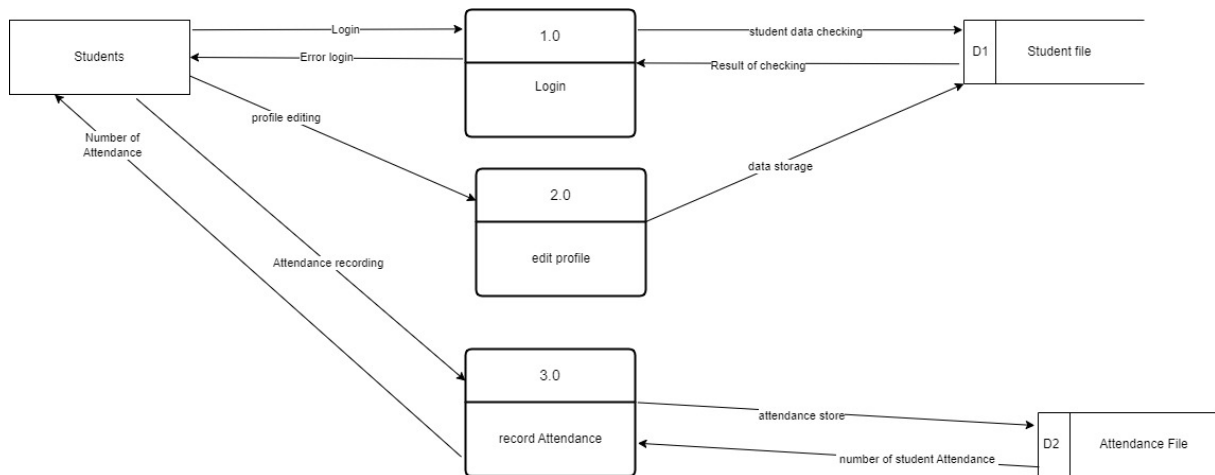
Figure 3.3.3: Data Flow Diagram: Level 0 (Student Diagram)
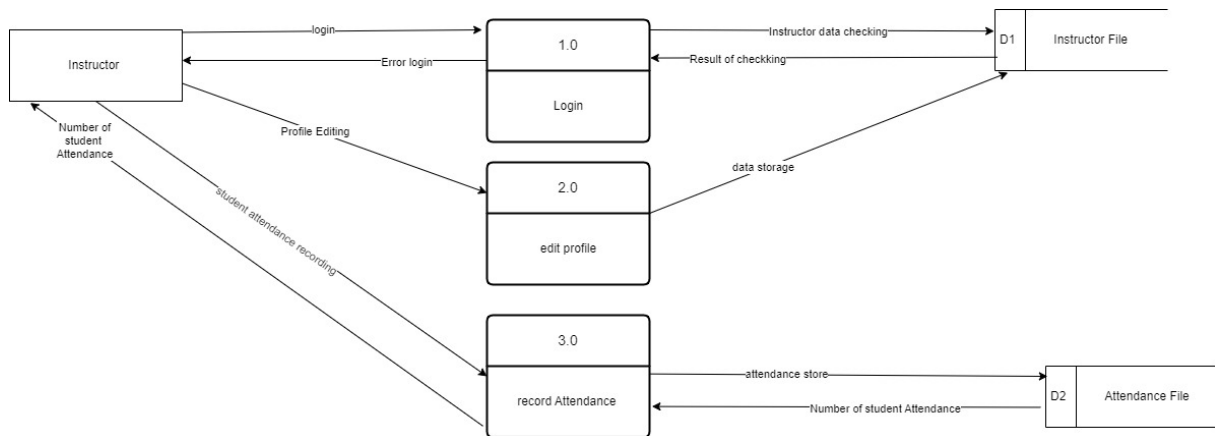
Figure 3.3.4: Data Flow Diagram: Level 0 (Instructor Diagram)
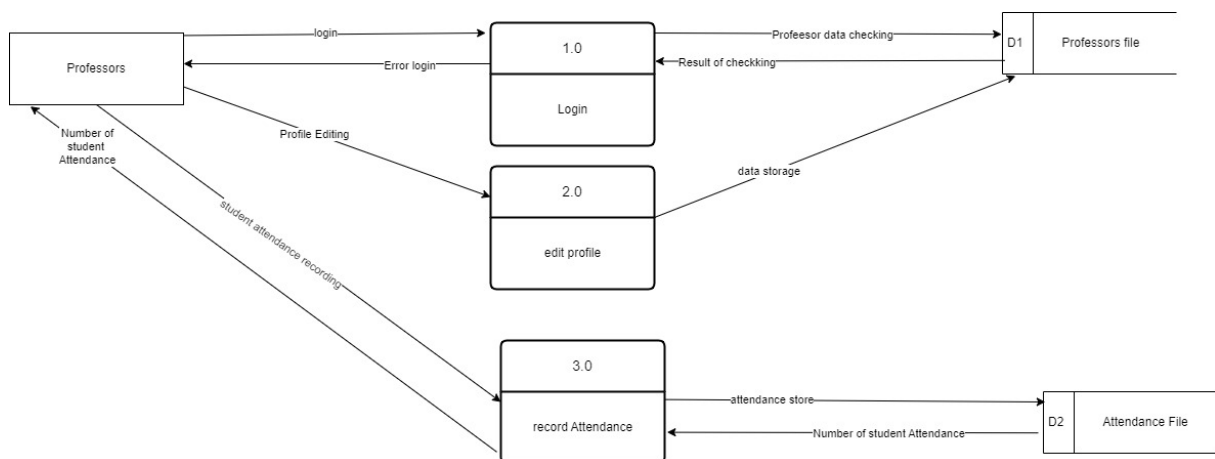
Figure 3.3.5: Data Flow Diagram: Level 0 (Professor Diagram)
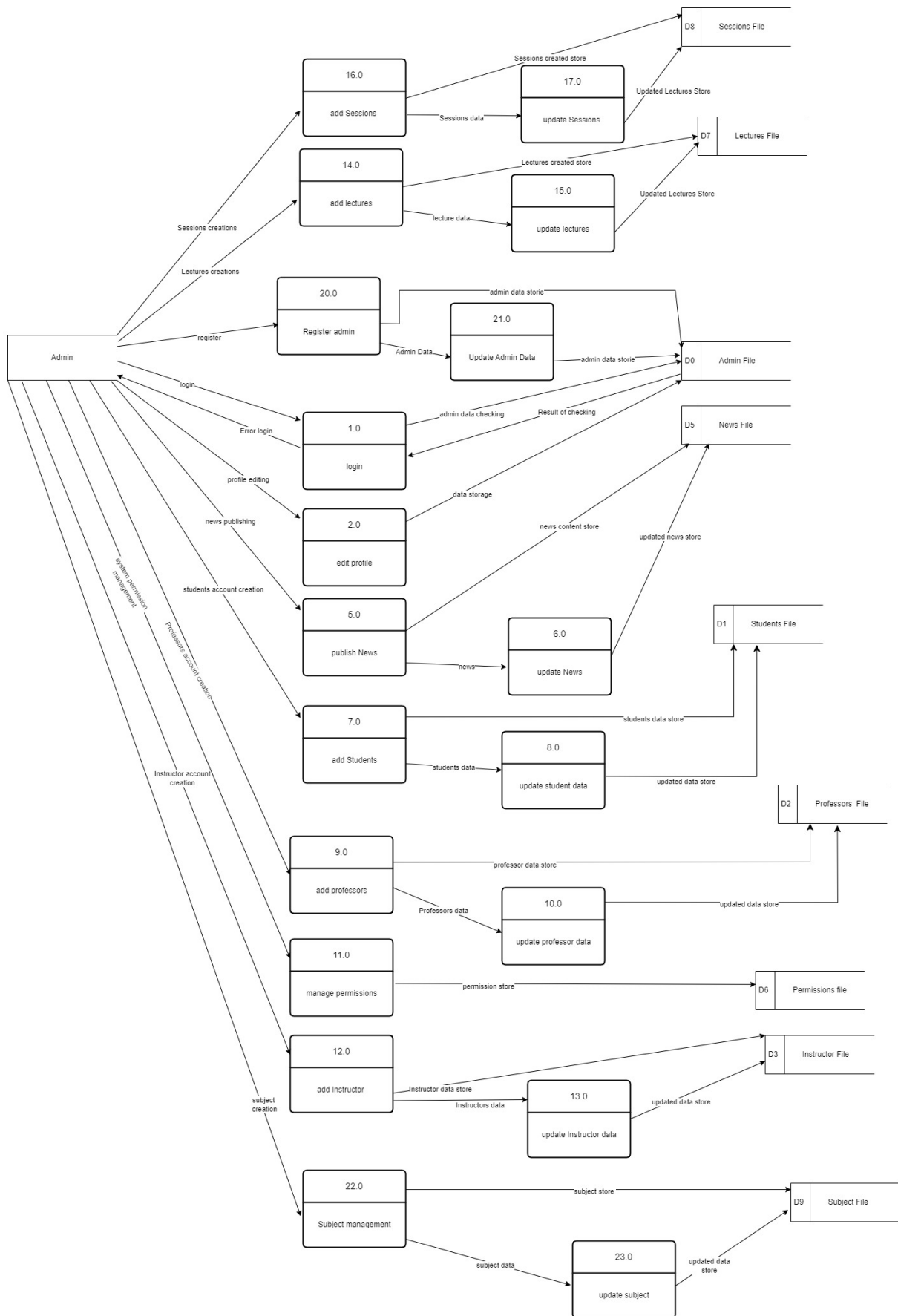
Figure 3.3.6: Data Flow Diagram: Level 0 (Admin Diagram)

3.4   System Design

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves creating a detailed blueprint or plan for how a system will be implemented, including the technical specifications, hardware and software requirements, and the overall structure of the system.

System design takes into account the functional and non-functional requirements of the system, as well as considerations such as scalability, reliability, performance, security, and maintainability. During system design, the high-level requirements gathered during system analysis are translated into a detailed design that can be used as a basis for system development. This includes designing the system's user interface, databases, data storage and retrieval mechanisms, networking infrastructure, and overall system architecture.

The goal of system design is to create a comprehensive and coherent plan that guides the implementation and development of the system while ensuring that it meets the needs and expectations of its users. System design is a critical phase in the system development life cycle, as it provides the technical specifications and guidance necessary for building and implementing the system.

It serves as a bridge between system analysis and system implementation, ensuring that the requirements identified during analysis are effectively translated into a working system that fulfills the organization's objectives.

3.4.1   Entity Relationship Diagram (ERD)

An entity-relationship diagram (ERD) is a visual representation of the entities (such as people, objects, concepts, or events) and the relationships between them within a system or database. ERDs are commonly used in the field of database design to illustrate the logical structure of a database. They depict how different entities relate to each other and the nature of those relationships. In an ERD, entities are represented as rectangles, and relationships between entities are depicted using lines connecting them. The lines indicate the type of relationship (e.g., one-to-one, one-to-many, many-to-many) and may include additional information such as cardinality (how many instances of one entity can be related to how many instances of another entity). Attributes of entities (i.e., the specific pieces of data that describe each entity) are often included in the rectangles representing the entities in the diagram. Overall, an ERD provides a clear and concise

way to visualize the structure of a database and its various components, helping database designers, developers, and stakeholders understand how different elements of the systems are related to each other.

Entity Relationship Diagram Components:

- Entities.

- Relationships.

- Attributes.
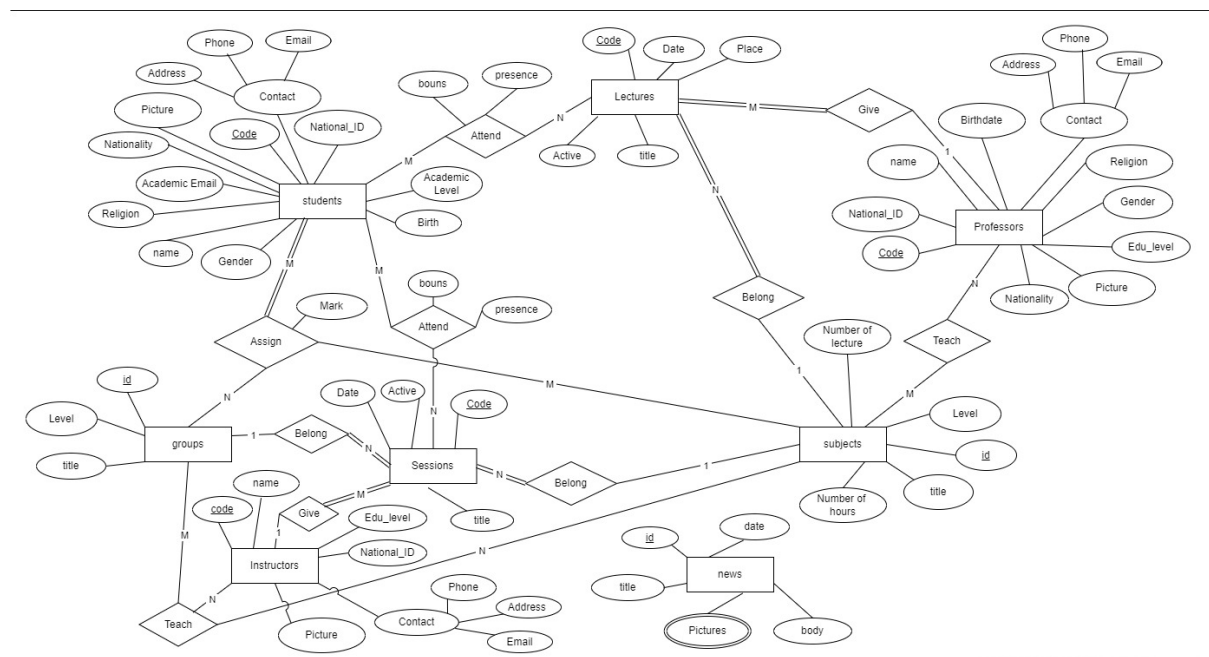
Figure 3.4.1: ERD Diagram

Entities that we will need:

- Students

- Professors

- Instructors

- Lectures

- Sessions

- Subjects

- Group

- News

Chapter 4

project Tools

4.1 Overview

Our project represents a convergence of frontend, backend, UI, and mobile development tools to create a comprehensive solution. Frontend technologies like HTML, CSS, JavaScript, and React provide an interactive and visually attractive user interface. The PHP and MySQL backend ensures robust data handling and processing. UI tools like Figma contribute to the aesthetics,while flutter has expanded our reach to mobile platforms.. Latex aids in creating professional documentation. Together, these tools form the backbone of our project, symbolizing a commitment to innovation and excellence.

4.2 Developing tools

The process of developing computer software or a set of programs to accomplish the many functions that a business requires is known as application development, applications assist organizations to automate operations and boost productivity by doing everything from calculating monthly budgets to arranging sales reports, gathering requirements, designing prototypes, testing, implementation, and integration are all processes in the app development process.

And because of rapid advances in technology, now you almost can use any programming language to develop any kind of application, of course, there is the best choice for every application, but you still can choose any developing tools to develop the specific application.

Tools that will be used in developing our project

- UI/UX Design.

- Front-end.

- Back-end.

- Mobile APP.

4.2.1   UI/UX Design

Figma has emerged as an indispensable tool in our project's UI/UX design workflow, revolutionizing the way we conceptualize, collaborate, and implement design elements. As a cloud-based design platform, Figma transcends traditional design software by providing a seamless and collaborative environment accessible to the entire team, regardless of geographical location.

One of the key advantages of Figma is its real-time collaboration feature, allowing multiple team members to work simultaneously on a project. This fosters a dynamic and interactive design process where designers, developers, and stakeholders can contribute in real time, share feedback, and iterate swiftly. This collaborative approach has significantly streamlined our design-to-development pipeline, reducing communication barriers and enhancing efficiency. The versatility of Figma is evident in its ability to facilitate the end-to-end design process. From wireframing and prototyping to high-fidelity mockups, Figma offers a comprehensive suite of design capabilities. Its intuitive interface, coupled with an extensive library of design assets and components, empowers our team to create visually appealing and consistent user interfaces.

Furthermore, Figma's responsive design features ensure that our UI/UX designs seamlessly adapt to various screen sizes and devices, a crucial aspect in today's diverse digital landscape.

4.2.2   Front-End

HTML or HyperText Markup Language, is the standard markup language used to create and design web pages. It forms the backbone of content structure on the World Wide Web, defining the elements and their arrangement within a webpage. HTML uses a system of tags, each enclosed in angle brackets, to describe different elements on a web page.

CSS or Cascading Style Sheets, is a stylesheet language used to describe the presentation and formatting of a document written in HTML or XML. In simpler terms, CSS is responsible for styling the visual presentation of web pages, allowing developers to control the layout, colors, fonts, and other design aspects.

JAVASCRIPT JavaScript is a versatile and widely used programming language that enables the creation of dynamic and interactive content on websites. Originally developed to enhance the interactivity of web pages, JavaScript has evolved into a powerful and versatile language that is now used for both front-end and back-end web development.

JavaScript is an essential component of modern web development, providing the means to create interactive user interfaces, handle user input, communicate with servers, and perform various other tasks that enhance the user experience on the web.

BOOTSTRAP   is a popular open-source front-end framework that facilitates the development of responsive, mobile-first web projects. It was created by Twitter and is now maintained by the open-source community. Bootstrap provides a set of pre-designed HTML, CSS, and JavaScript components, as well as a responsive grid system, helping developers create consistent and visually appealing web applications with ease.

REACT   is an open-source JavaScript library for building user interfaces, particularly for single-page applications where user interactions and updates need to be handled efficiently. Developed and maintained by Facebook, React allows developers to create reusable UI components and manage the state of these components efficiently and pre-

dictably.

React is widely used for building modern web applications, and its componen-t-based approach and efficient rendering make it a powerful tool for creating interactive and dynamic user interfaces.

4.2.3 BACK-END

PHP is a scripting programming language widely used in the development of web applications and their interaction with databases. PHP was developed to run on the server side and used to generate and automatically update dynamic web pages. PHP is an open-source programming language, available free of charge for use.

PHP allows for faster and easier generation of dynamic web pages by interacting with databases to obtain and display necessary information on web pages. Additionally, it enables the generation of image files, videos, PDFs, and other dynamic content on the web.

PHP is extensively used in the development of modern web applications, including business applications, e-commerce, and social websites. It is also used in other fields such as web-based game development and mobile applications.

MYSQL is developed, distributed, and supported by Oracle Corporation. MySQL is a database system used on the web it runs on a server.

MySQL is ideal for both small and large applications. It is very fast, reliable, and easy to use. It supports standard SQL. MySQL can be compiled on several platforms. The data in MySQL is stored in tables. A table is a collection of related data, and it consists of columns and rows. Databases are useful when storing information categorically.

WHY TO USE MySQL:-

- Leading open source RDBMS

- Ease of use – No frills

- Fast

- Robust

- Security

- Multiple OS support

- Free

- Technical support

LARAVEL   is a web application framework for developing dynamic web applications using PHP. Laravel helps developers develop web applications faster and more efficiently, as they include many features and helper tools such as routing system, database system, data validation, session control, and more. Laravel is one of the foundational programming languages that any web developer should learn.

The Programs used:-

- **XAMPP** is a cross-platform web server that is free and open-source. XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl. XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver.

- **Visual Studio Code** is a code editor in layman's terms. Visual Studio Code is "a predator that helps the programmer write code, helps in debugging and corrects the code using the intelli-sense method ". In normal terms, it facilitates users to write the code efficiently.

4.2.4   Mobile APP

DART is a general-purpose programming language developed by Google. It was first announced in 2011 and has since gained popularity for building web, mobile, and server applications. Dart is designed with a focus on simplicity, productivity, and performance. It serves as the primary language for developing applications using the Flutter framework.

Dart has found success in building mobile applications, particularly with Flutter, and it continues to evolve as both the language and the Flutter framework are actively developed and supported by the community and Google.

FLUTTER is an open-source UI software development toolkit created by Google. It is used for building natively compiled applications for mobile, web, and desktop from a single codebase. Flutter was first introduced in 2018 and has gained popularity for its ability to provide a consistent and high-performance user experience across different platforms.

Flutter has become a popular choice for developers looking to create cross–platform applications with a consistent and native-like user experience. It is particularly well-suited for projects where a unified codebase and rapid development are important considerations.

4.3   Another tools

4.3.1   latex

LaTeX is a typesetting system and document preparation software widely used for the communication and publication of scientific and mathematical documents due to its excellent handling of complex equations, figures, and references.

To create a LaTeX document, you typically write your content in a plain text file with a .tex extension. The document is then compiled using a LaTeX compiler to produce a PDF or other output formats. Many online platforms, such as Overleaf, provide collaborative LaTeX editing environments.

4.4 Git

Git is a distributed version control system that helps developers manage and track changes to source code during software development. It was created by Linus Torvalds in 2005 and has become a standard tool in the software development industry.

4.4.1 GitHub

GitHub is a web-based platform that provides hosting for software development and collaboration using Git. It was founded in 2008 by Tom Preston-Werner, Chris Wanstrath, and PJ Hyett. GitHub is widely used for hosting open-source projects, as well as private repositories for individual developers and organizations.

Chapter 5

System Design

5.1 Web Design

5.1.1 Home Page
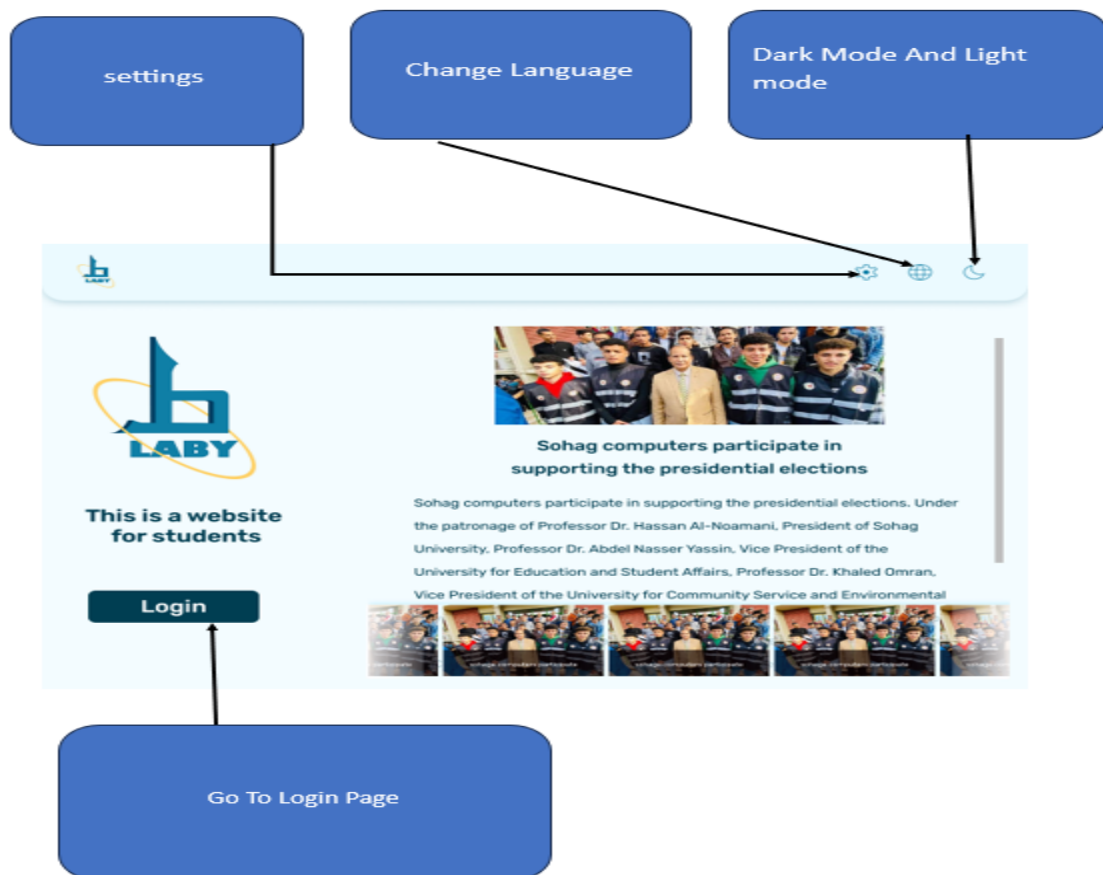
Figure 5.1.1: Home page

5.1.2   Login Page
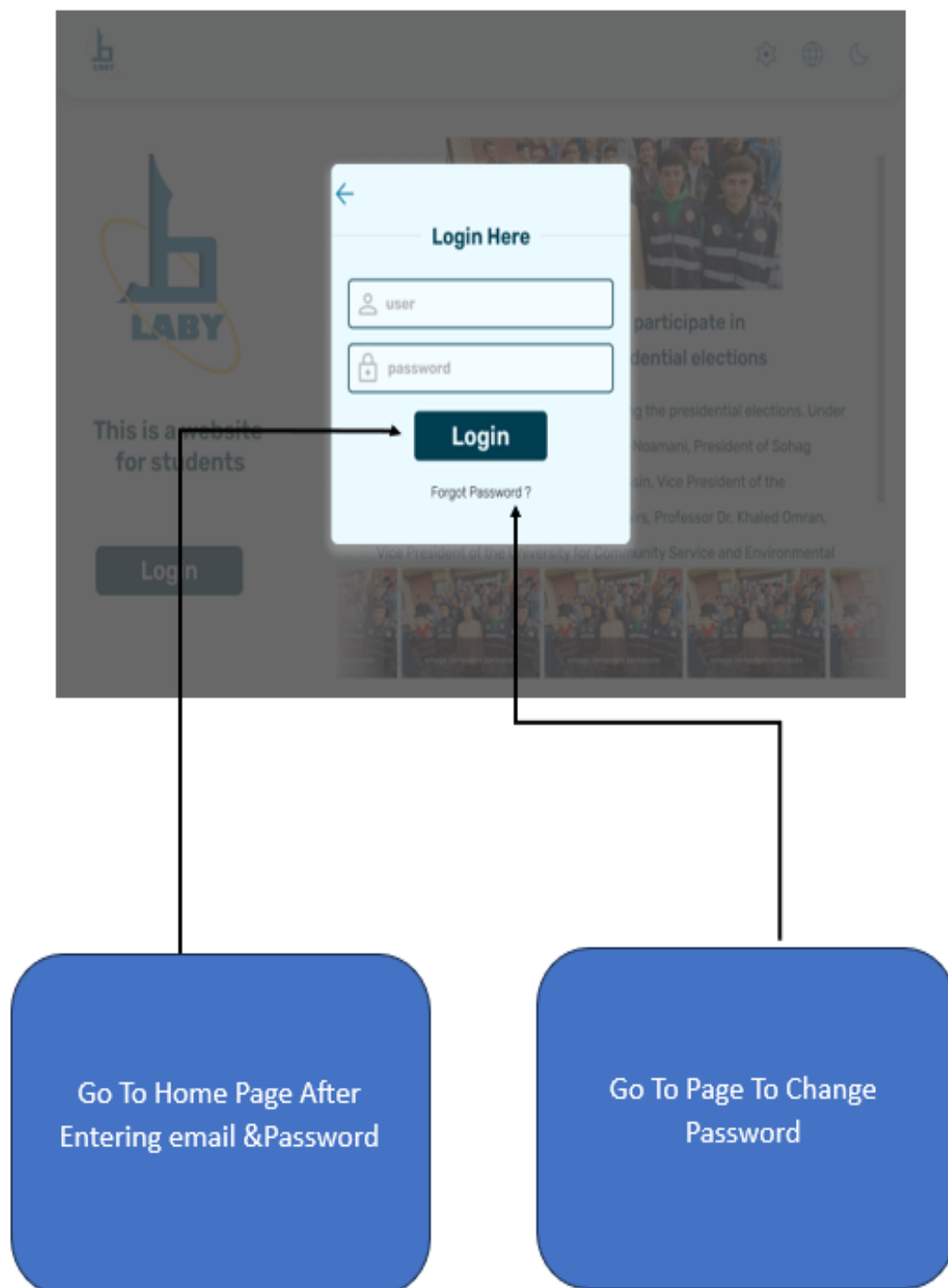
Figure 5.1.2: Login page

5.1.3 After Login Page
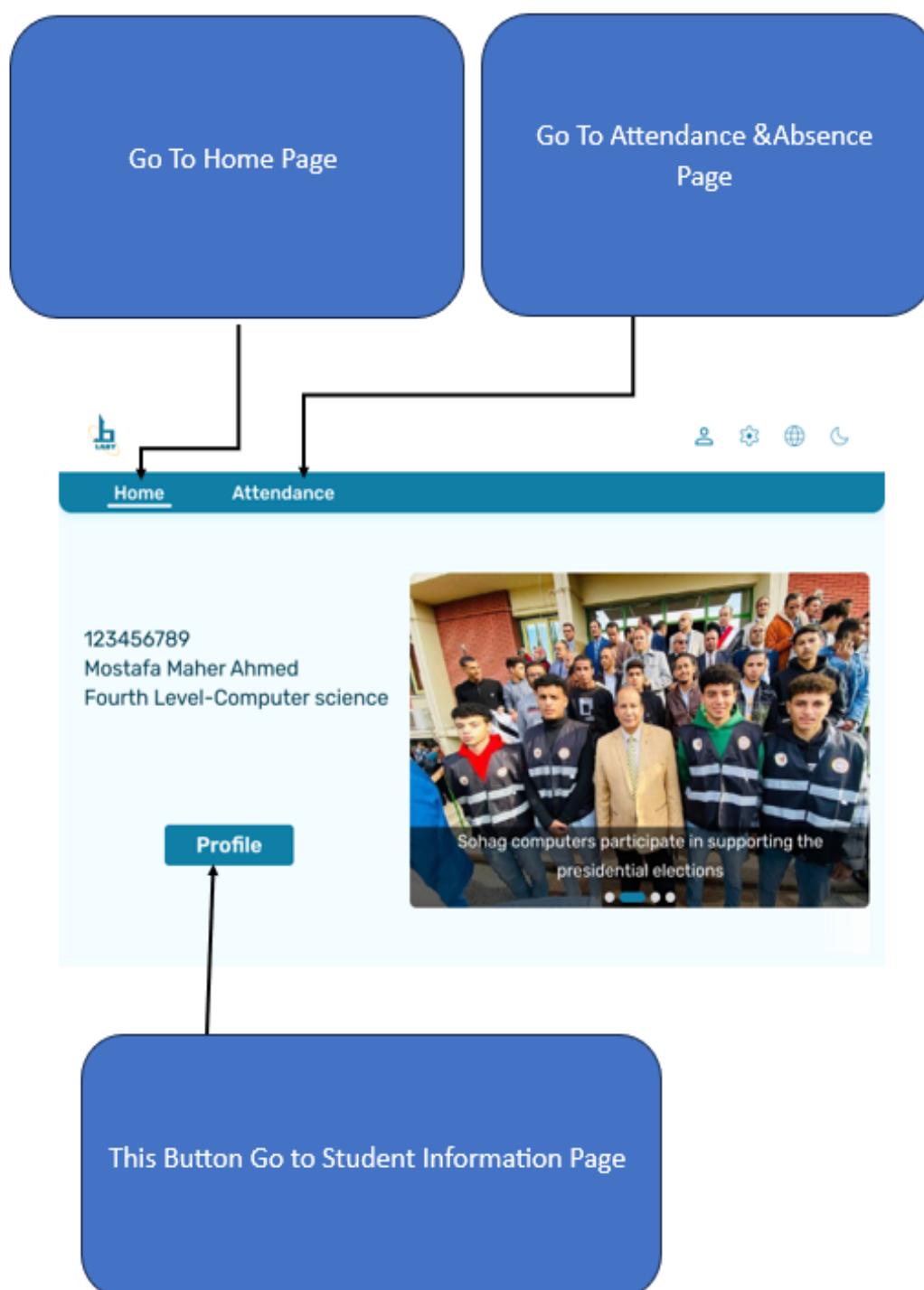
Figure 5.1.3: After login page

5.1.4 Profile Page
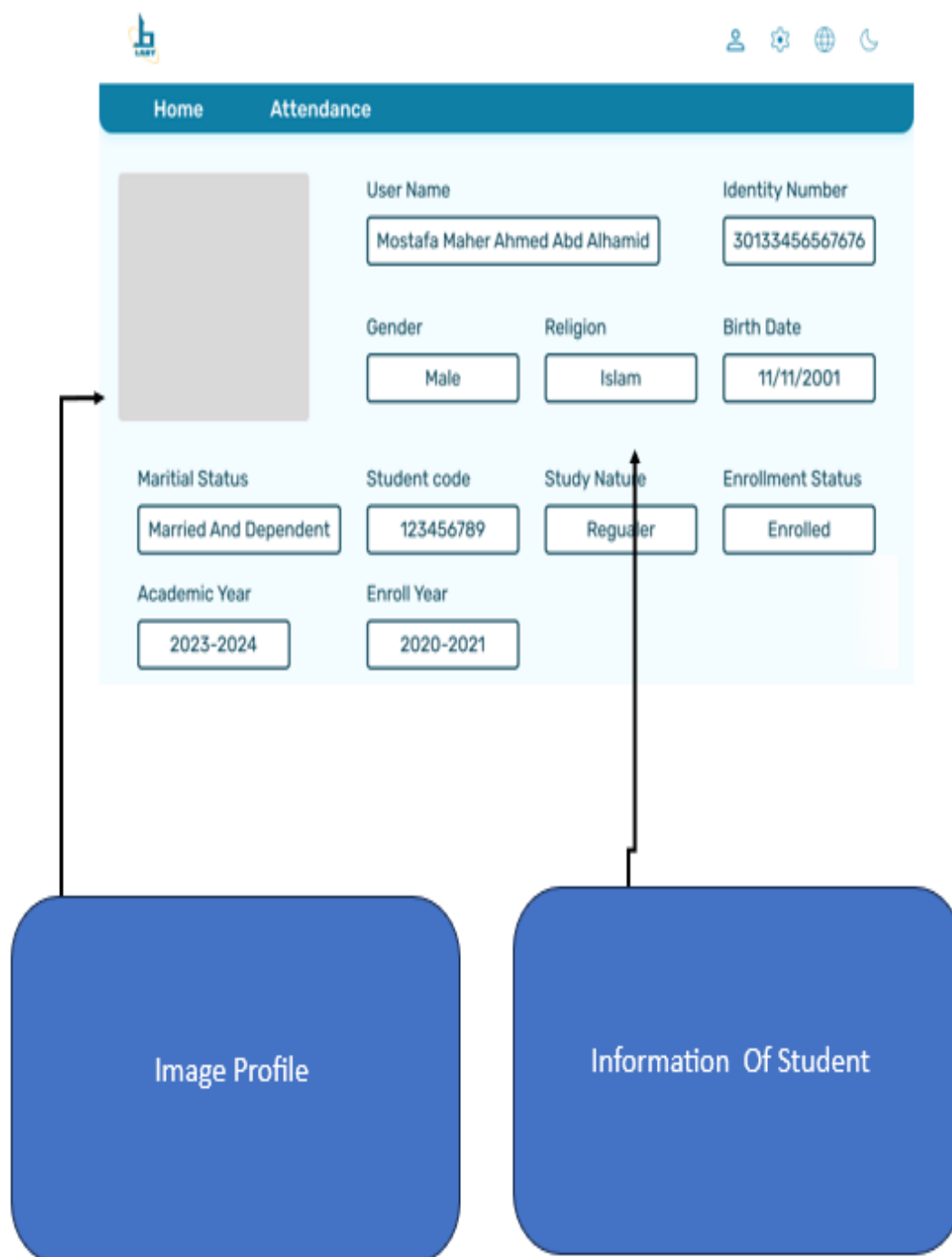
Figure 5.1.4: Profile page

5.1.5 News Page
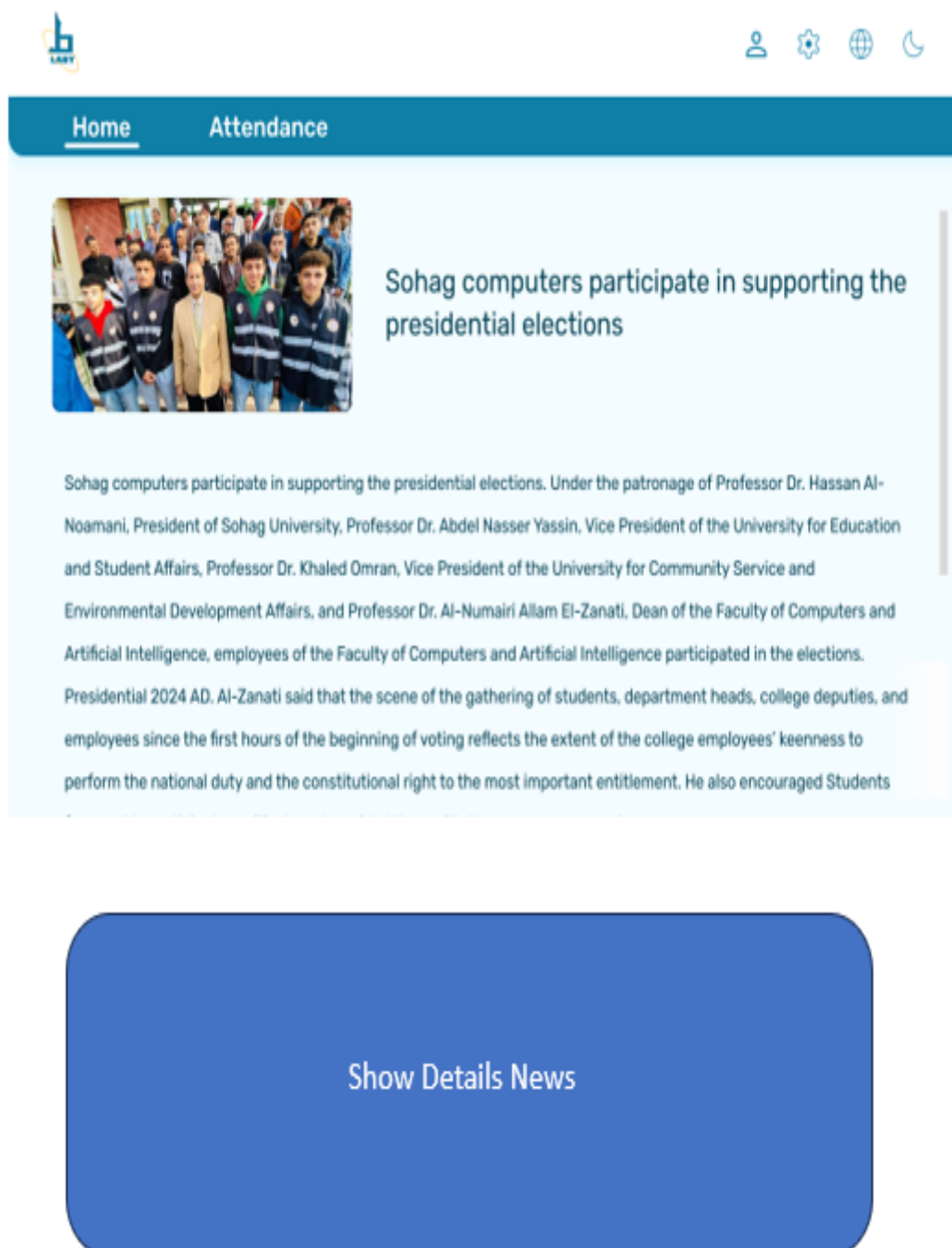
Figure 5.1.5: News page

5.1.6 Attendance Pages
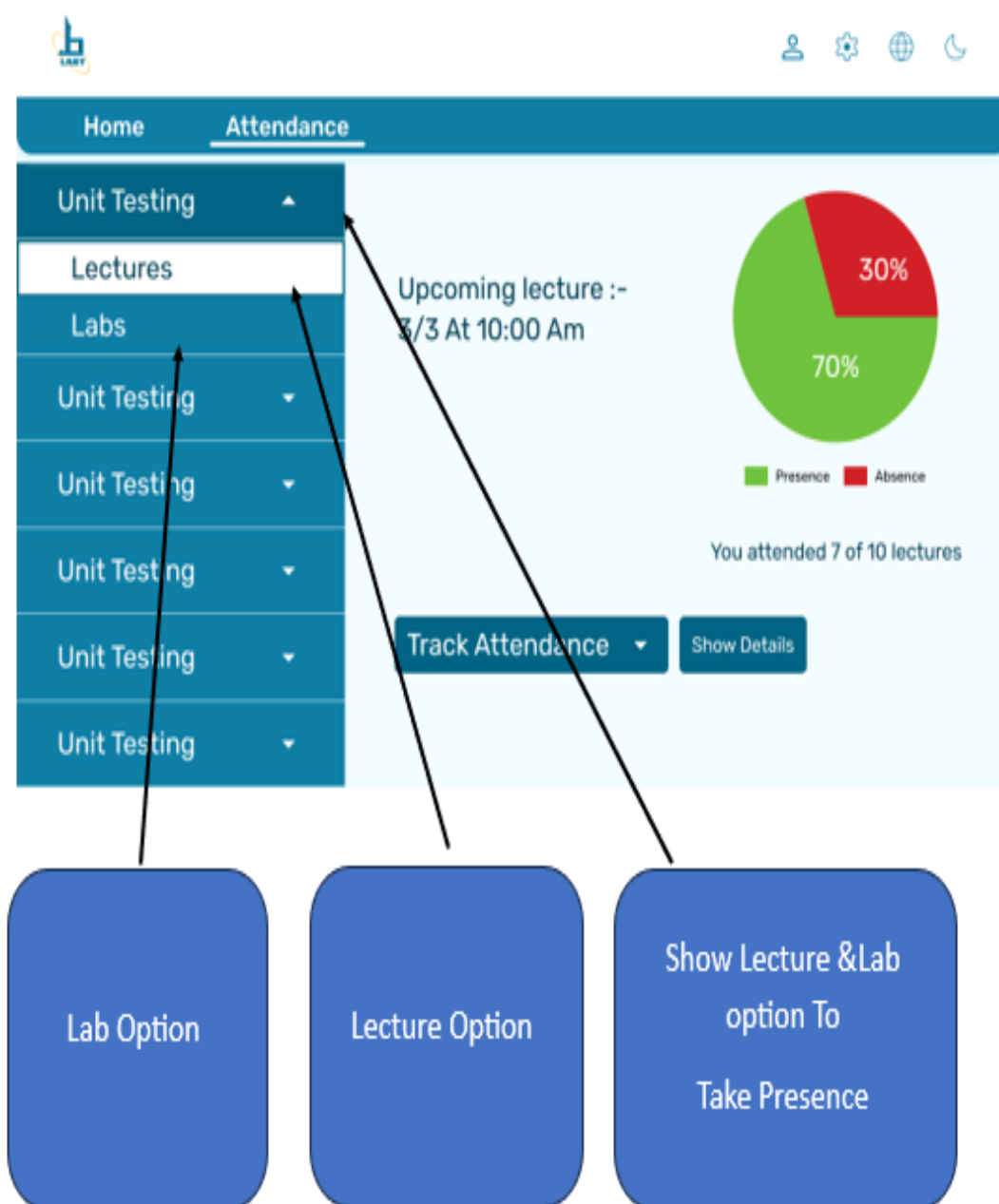
Figure 5.1.6: Attendance page
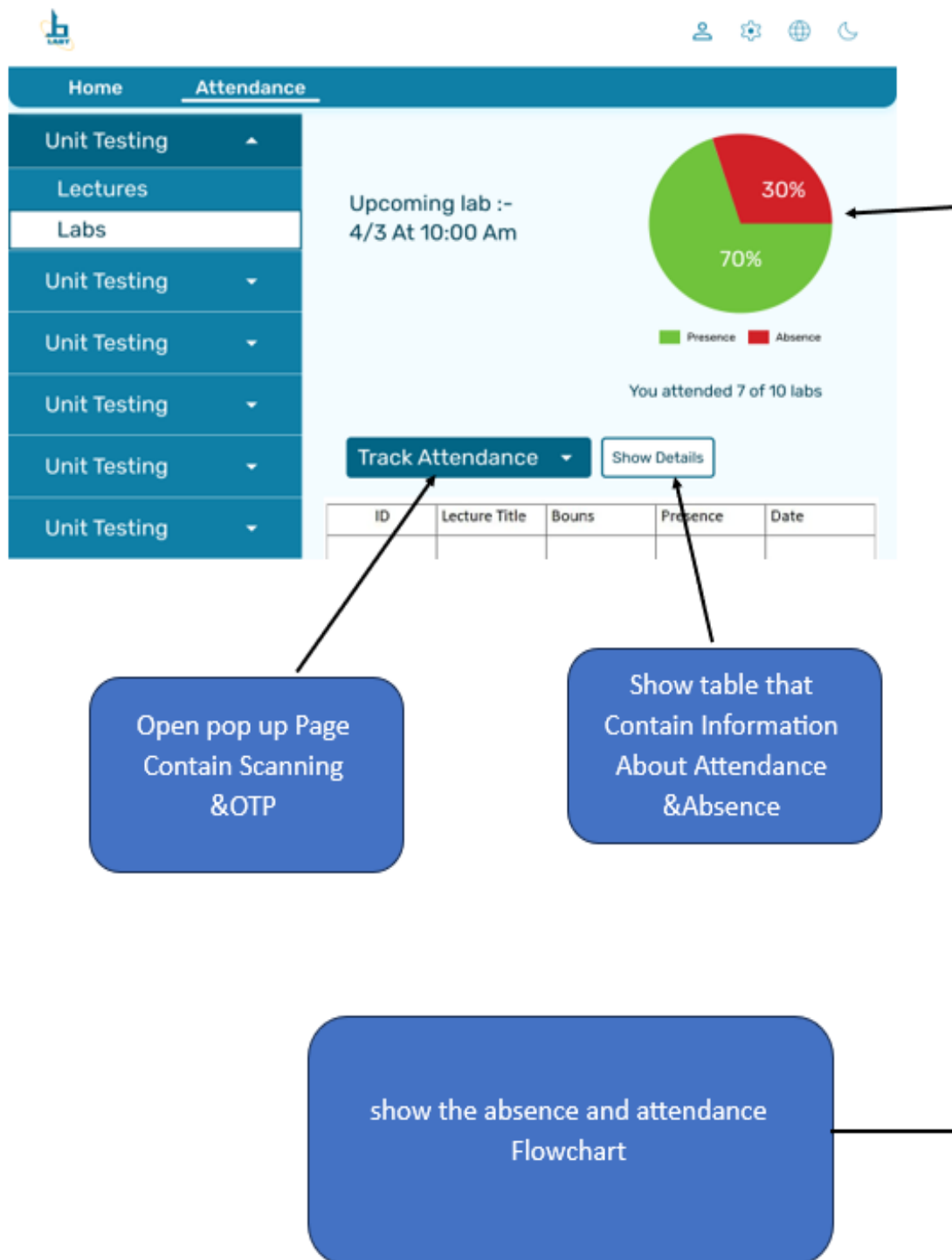
Figure 5.1.7: Attendance page
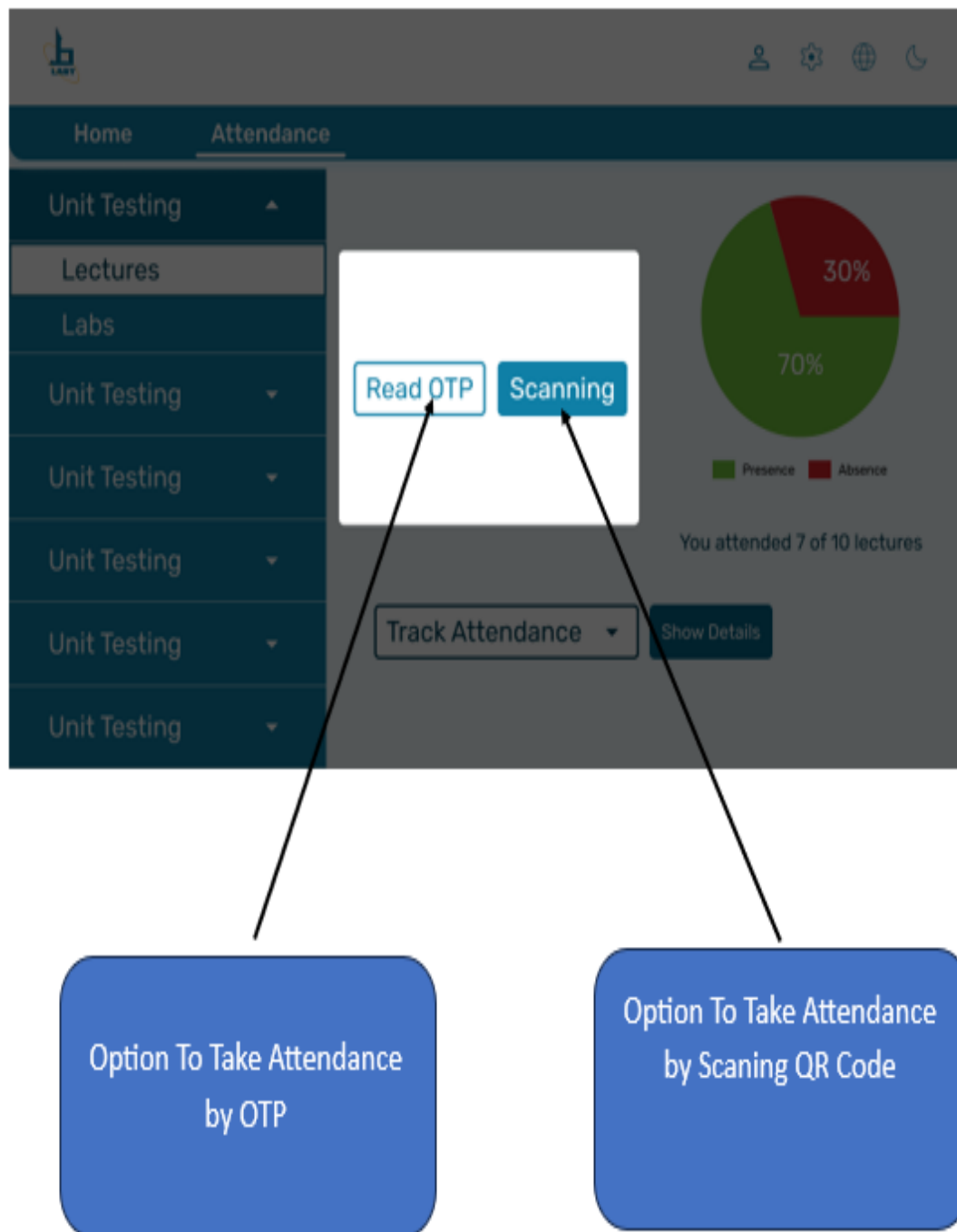
Figure 5.1.8: Attendance page
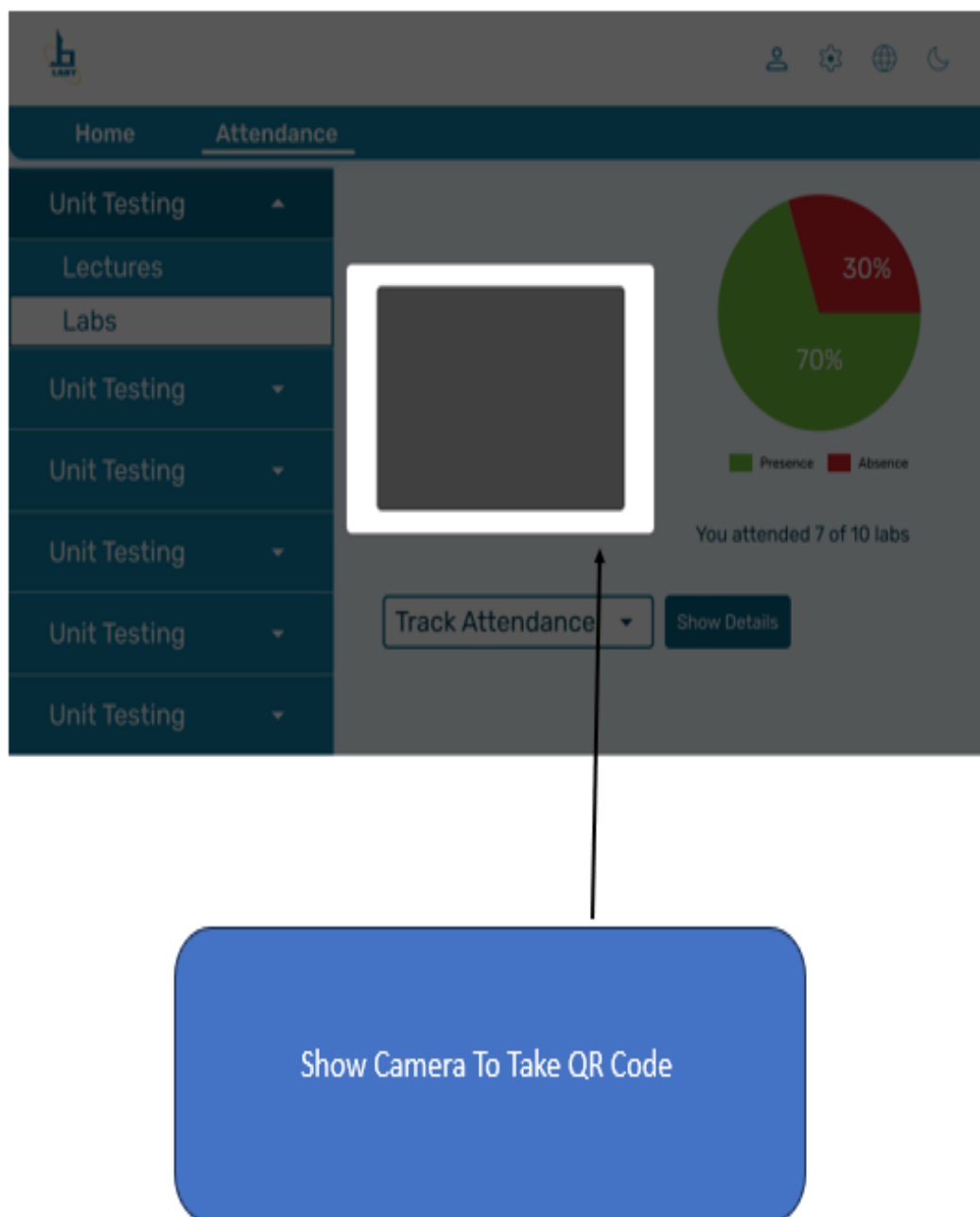
Figure 5.1.9: Attendance page

Figure 5.1.10: Attendance page

5.2 Mobile Design
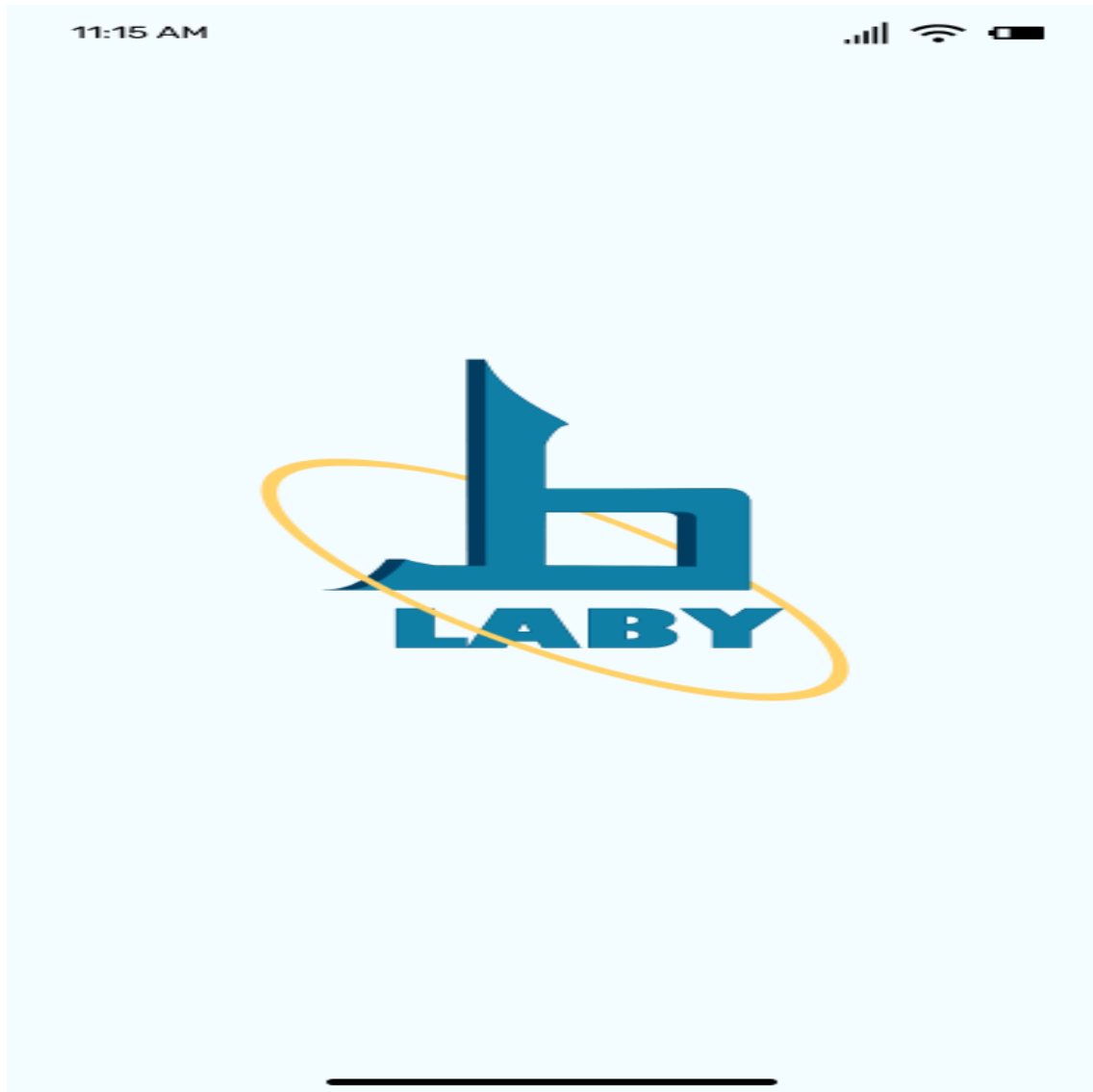
Figure 5.2.1: Loading Page
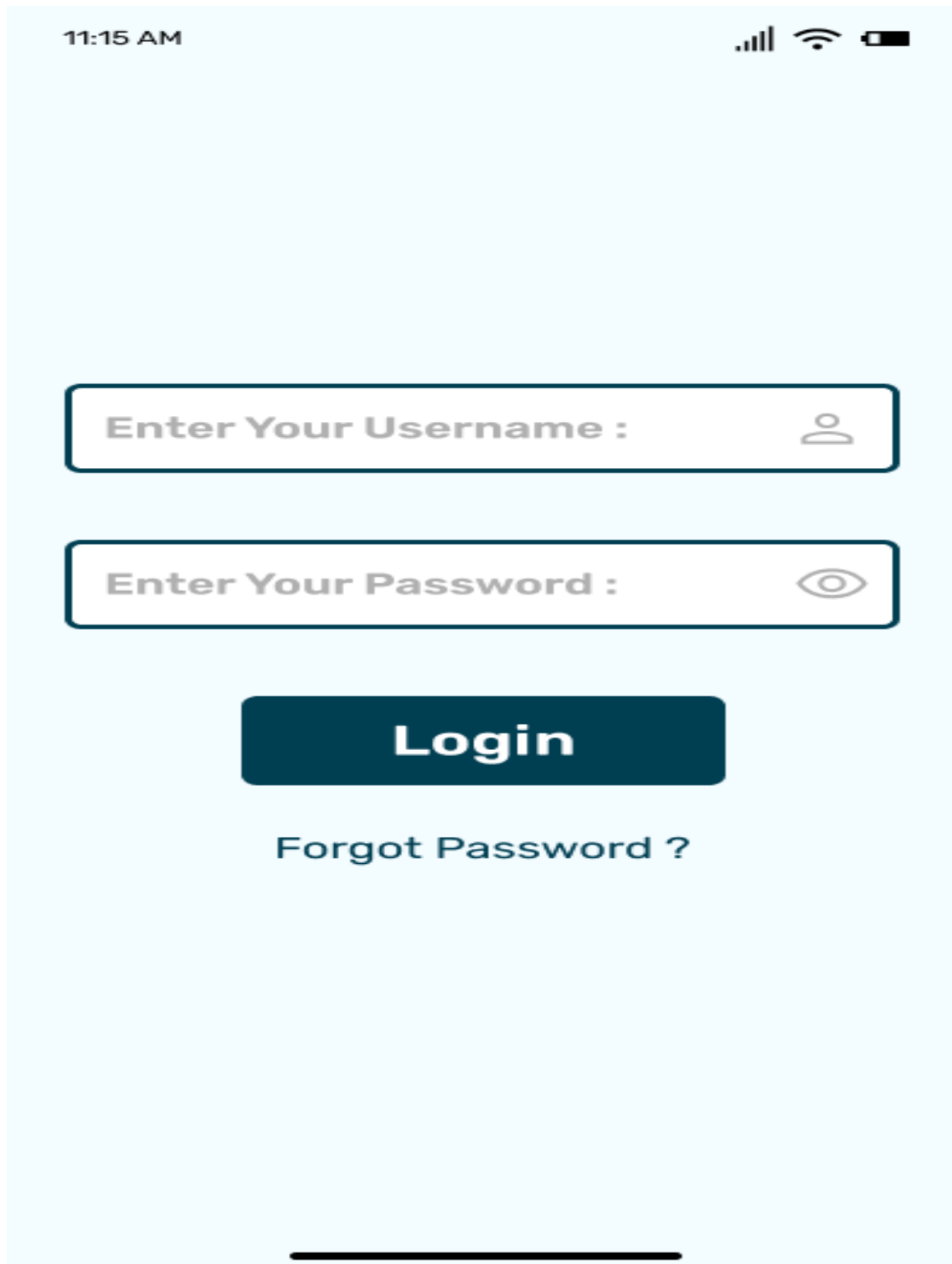
Figure 5.2.2: Login Page
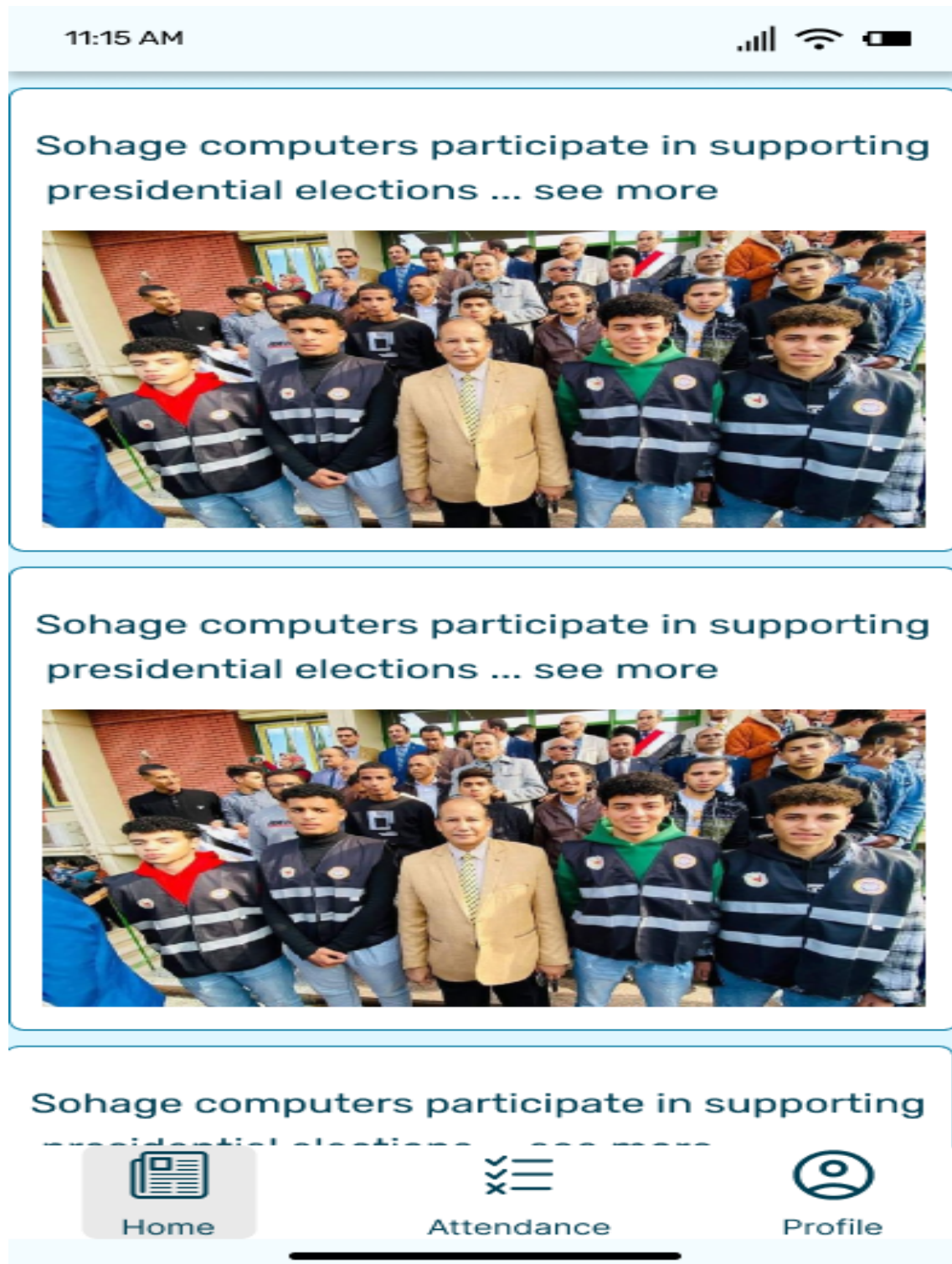
Figure 5.2.3: Home Page
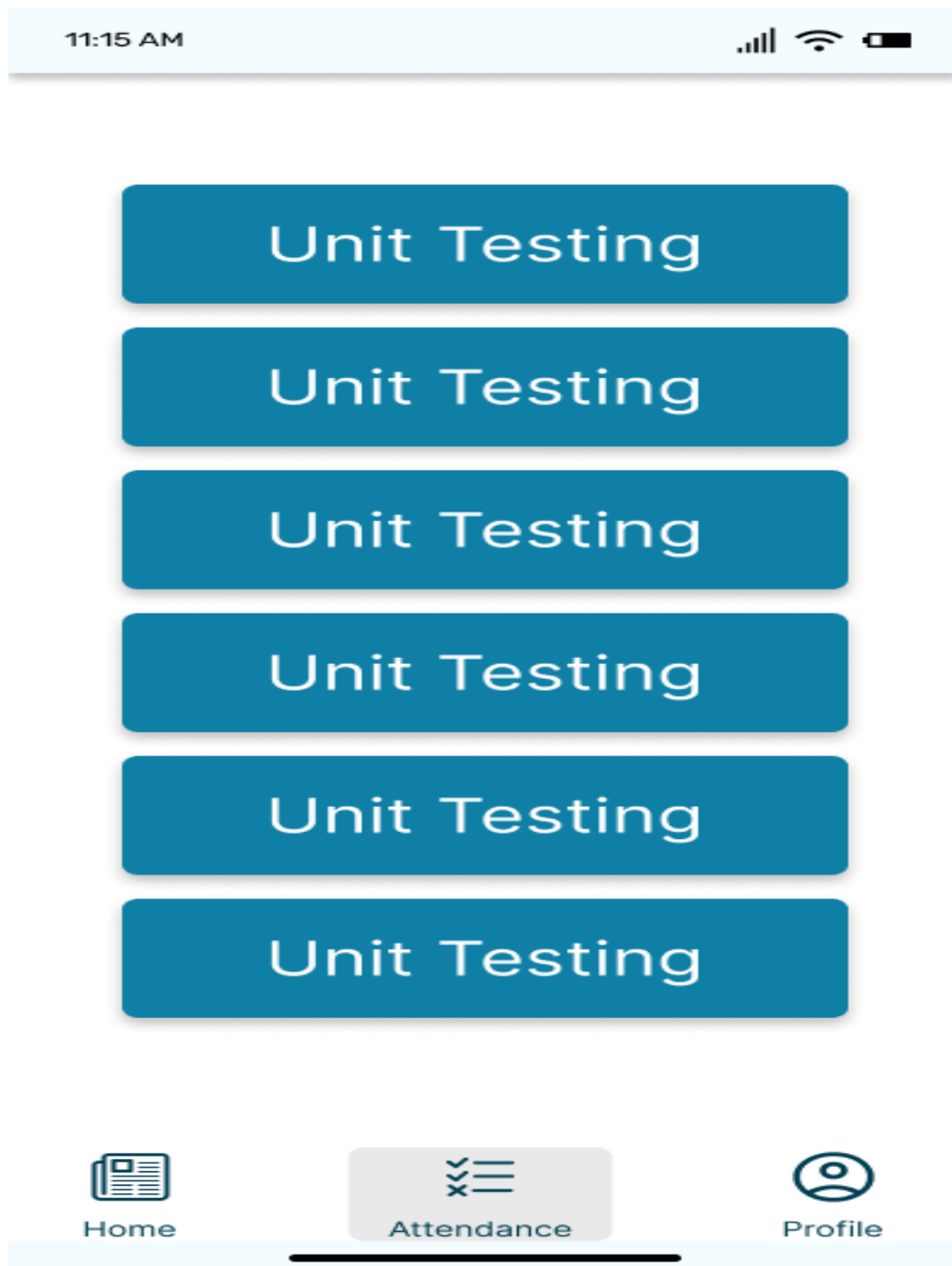
Figure 5.2.4: Page Contains the Subjects
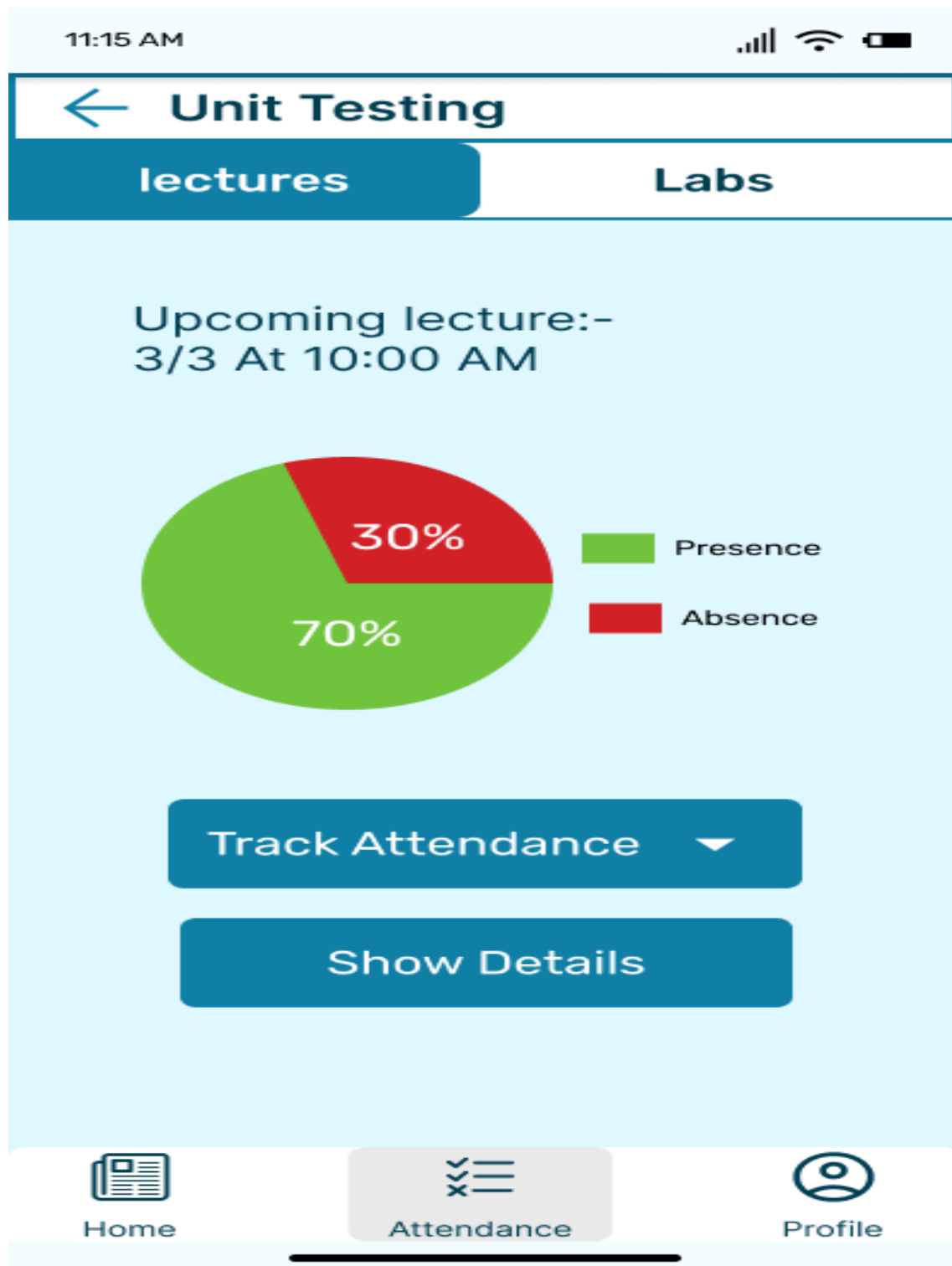
Figure 5.2.5: Page to register attendance or Show Details
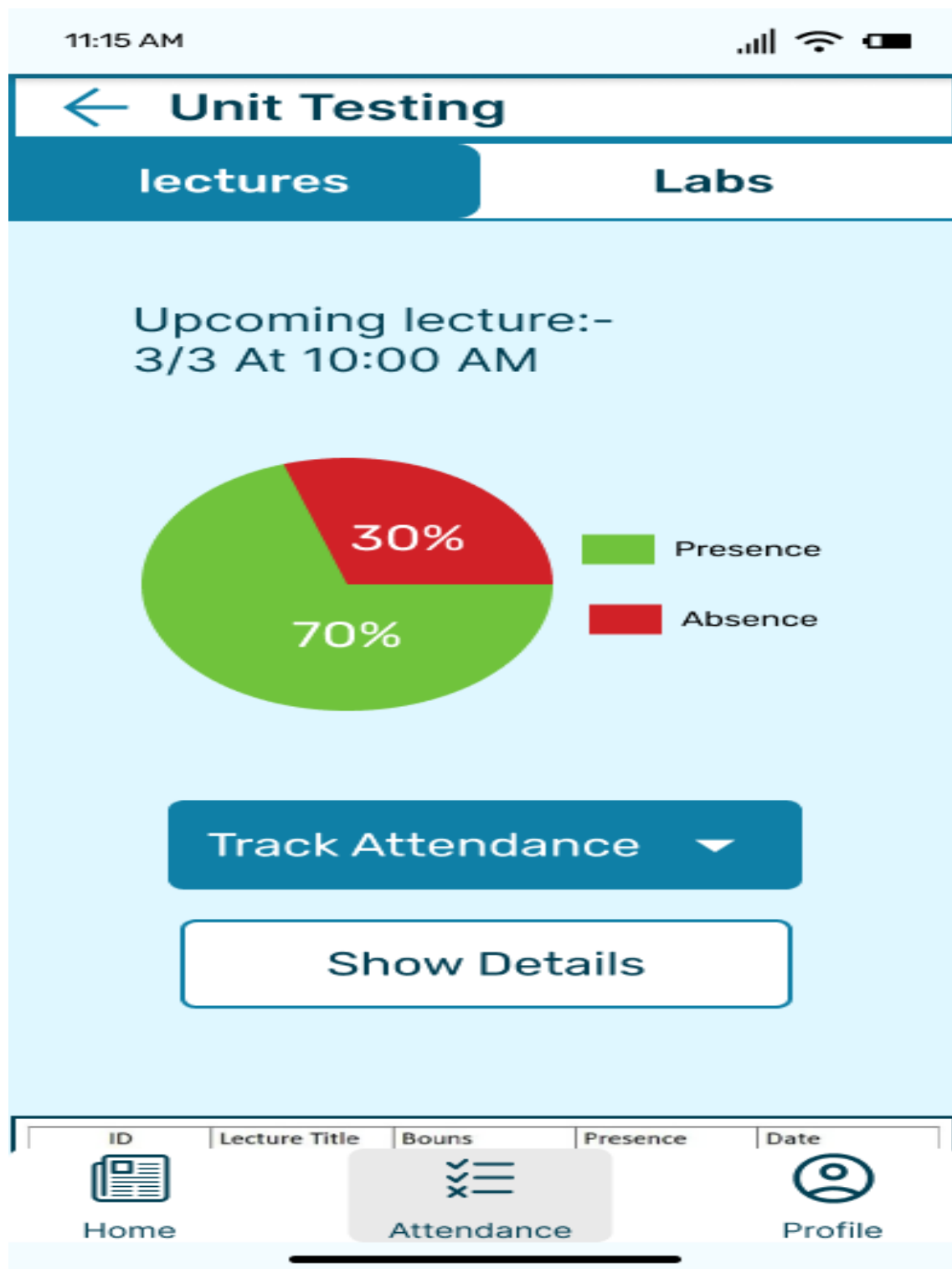
Figure 5.2.6: After You Click on Show Details Button, a Table with previous Attendance Records will appear
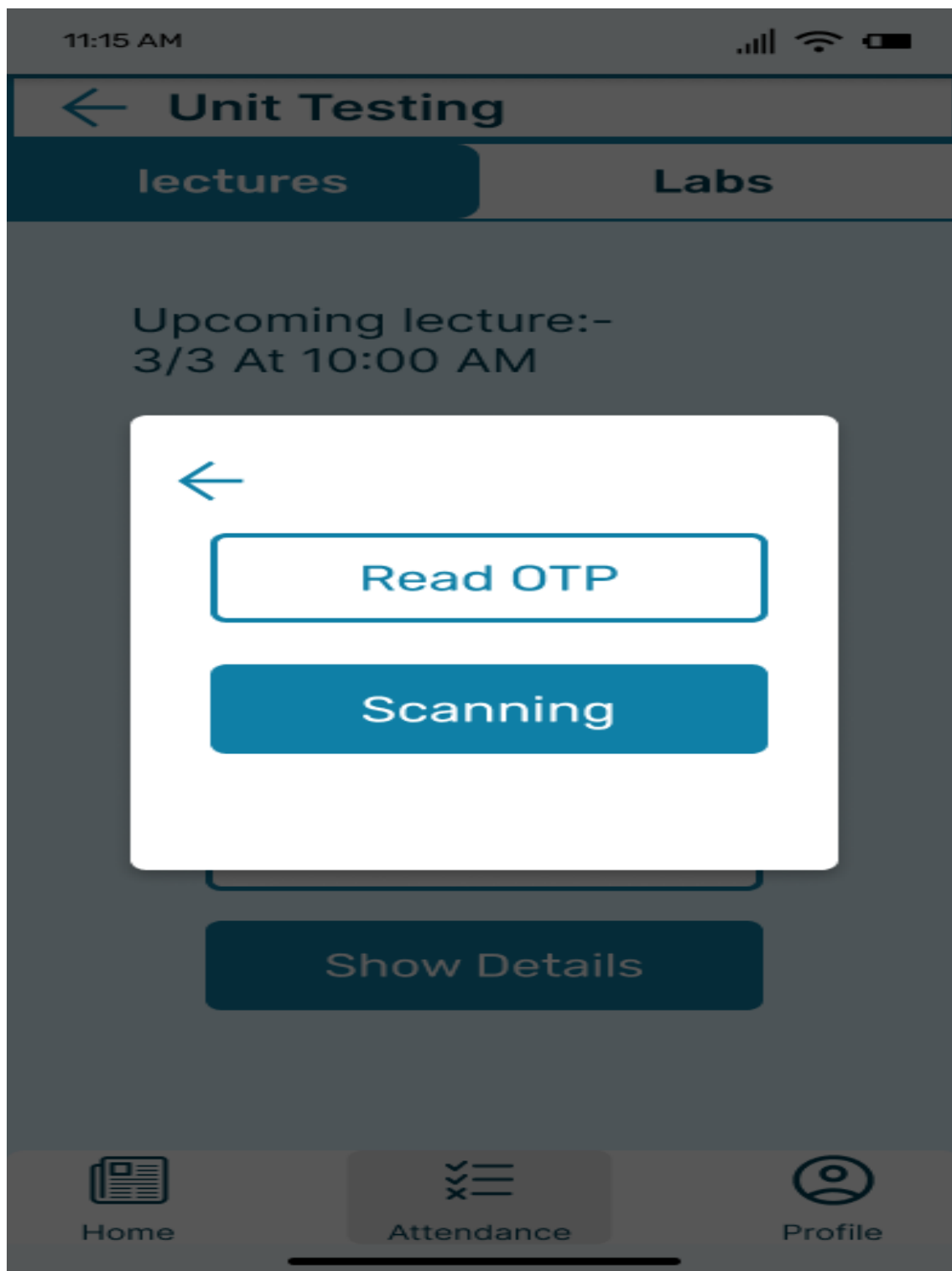
Figure 5.2.7: After You Click Track Attendance Pop up With Various Options to Register your Attendance.
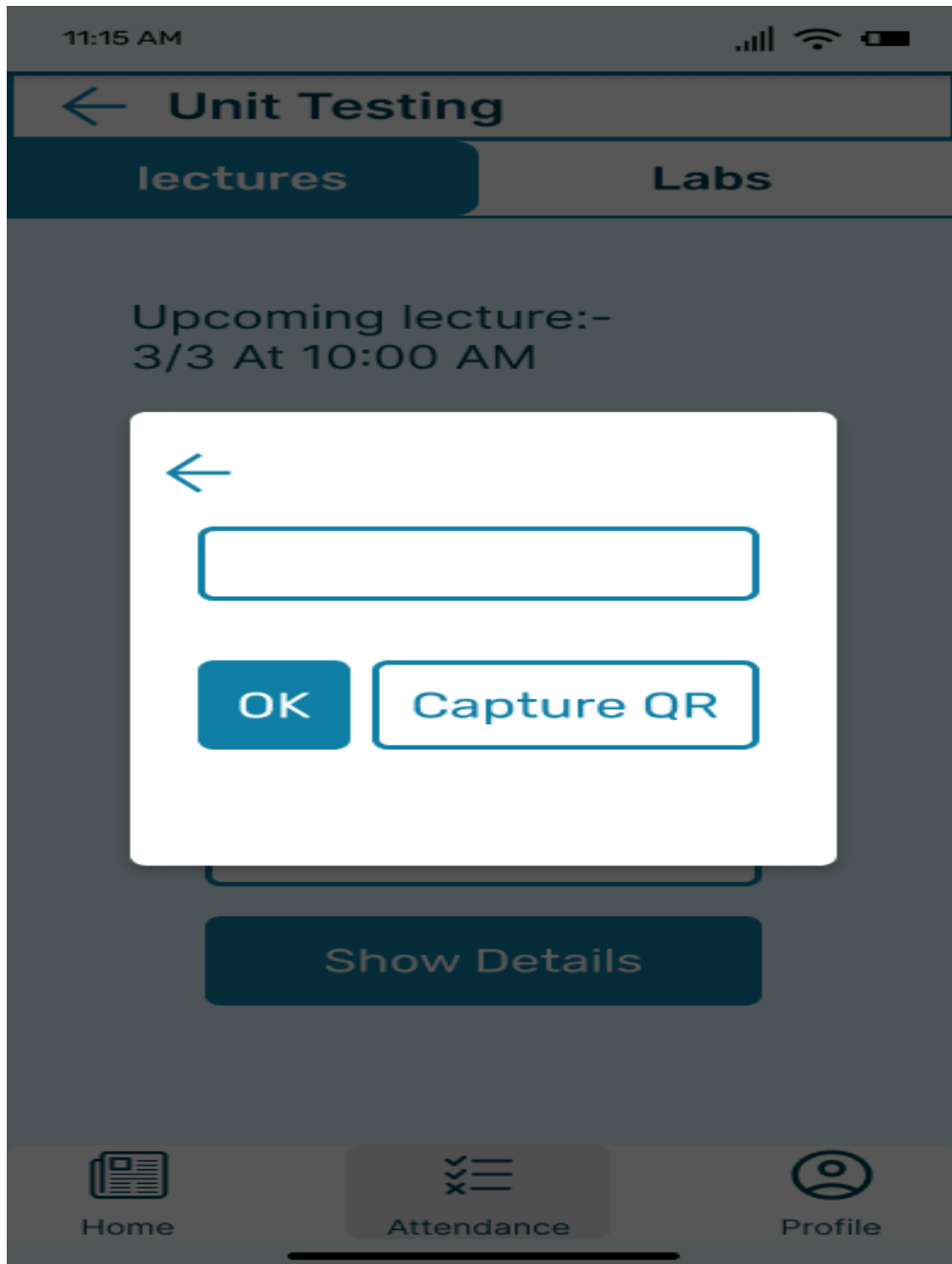
Figure 5.2.8: This Page Will Appear When You Click Take OTP

Figure 5.2.9: Profile Page

Figure 5.2.10: Data About User

Figure 5.2.11: Data About User

Chapter 6

Conclusion

TuLaBy is an accurately designed platform tailored for students, professors, teaching assistants, and administrators in academic institutions.

It offers tailored benefits such as students being able to record attendance, access to personal information related to the educational process, view schedules, and track grades effortlessly.

Professors and teaching assistants benefit from streamlined attendance management, insightful analytics, and grade finalization. Teaching assistants efficiently manage lab attendance and properties.

Administrators have comprehensive control over users, subjects, and announcements.

TuLaBy offers an easy experience across web and mobile applications, prioritizing accessibility and user-friendliness. It provides a flexible framework for future enhancements without compromising existing functionalities, exemplifying precision, quality, and adaptability in academic management.

Bibliography

[1] Dr. Mohamed El Desouki, Fundamentals of Database
https://www.youtube.com/@DesoukiEgypt

[2] Dr. Ahmed Younes, Book Of Fundamentals Of Database Systems.

[3] Dr. Seham Elew, Data Flow Diagrams And System Analysis.

[4] Stack overflow "https://stackoverflow.com/"

[5] Overleaf "https://www.overleaf.com"

[6] Github "https://github.com/"

[7] Draw.io "https://app.diagrams.net/"