

LIST OF FIGURES

1. Figure 1.1 Use Case Diagram for Website.
2. Figure 1.2 Use Case Diagram for Application.
3. Figure 3.1 Class Diagram of I – Attend.
4. Figure 3.2 Activity Diagram of I – Attend.
5. Figure 3.3 Sequence Diagram of I – Attend.
6. Figure 3.4 Data Flow Diagram of I – Attend.
7. Figure 6.1 Flow Chart of I – Attend.
8. Figure 6.2 Login Page for Admin.
9. Figure 6.3 Credentials Page for Admin.
10. Figure 6.4 QR Code Generator for Student by Admin.
11. Figure 6.5 Mark Attendance by Admin.
12. Figure 6.6 Account Creation to App by Student.
13. Figure 6.7 Log in To App by Student.
14. Figure 6.8 Face Authentication to App by Student.
15. Figure 6.9 Attendance Marked by App.

CHAPTER 1

INTRODUCTION

1.1 Introduction

I-Attend is a smart attendance recording and maintaining software, that uses both Web and App interfaces to record and maintain attendance using modern technologies such as face recognition, geolocation tracing, QR codes.

I-Attend automates the task of attendance recording and maintaining up to large extent. It reduces the time required to capture attendance and also reduces the burden from the one taking the attendance.

This software is intended to be used by the teachers or professors of Schools and colleges, but can also be used by other organizations after applying some modifications and changes on the software.

1.2 Motivation

The following are some of the motivation points of I-Attend:

- **Saves time:** The software automates the task of attendance recording, which saves time for both the teachers/professors and the students. This can also lead to more time for teaching and learning.
- **Reduces errors:** Manual attendance recording can be prone to errors, but with I-Attend's use of modern technologies such as face recognition and geolocation tracing, it can reduce errors in attendance records.
- **Increases efficiency:** I-Attend can handle a large number of students, which can increase the efficiency of attendance recording and maintaining.
- **Enhances security:** The use of face recognition and QR codes in attendance recording can enhance security and prevent unauthorized attendance recording.
- **Adaptability:** Although the software is intended for schools and colleges, it can be modified and used by other organizations, which can increase its adaptability and usefulness.

1.3 Objective

The objectives of I-Attend are:

- To automate the attendance recording process for teachers or professors of schools and colleges, which can save time and reduce the burden of taking attendance.
- To use modern technologies such as face recognition, geolocation tracing, and QR codes to improve the accuracy and efficiency of attendance recording and maintaining.
- To provide both web and app interfaces for attendance recording, which can enhance accessibility and convenience for both teachers and students.
- To handle a large number of students, which can increase the efficiency and effectiveness of attendance recording and maintaining.
- To provide a secure and reliable system for attendance recording that prevents unauthorized attendance recording and reduces errors in attendance records.
- To be adaptable to other organizations after applying modifications and changes, which can increase its usefulness and versatility.

1.4 Analysis

Analysis of I-Attend:

- **Features:** I-Attend is a software that provides modern features such as face recognition, geolocation tracing, and QR codes to automate the attendance recording process. It also provides both web and app interfaces for recording attendance and maintaining records.
- **Benefits:** The use of I-Attend can reduce the time and burden required to take attendance manually. It can also increase the accuracy and efficiency of attendance recording, and provide a secure and reliable system for maintaining records. Additionally, the software can handle a large number of students and be adapted to other organizations after modifications.
- **Target users:** The primary target users of I-Attend are teachers and professors of schools and colleges, but the software can also be used by other organizations with modifications.

- **Implementation:** I-Attend uses modern technologies such as face recognition, geolocation tracing, and QR codes to automate the attendance recording process. The implementation of these technologies can ensure accurate and efficient attendance recording.
- **Potential drawbacks:** While I-Attend has many benefits, there may be potential drawbacks such as the need for reliable internet connectivity and the cost of implementing the software. Additionally, some students may have concerns about privacy and the use of facial recognition technology.
- **Competition:** There may be competition in the market for attendance recording software, which may impact the adoption and success of I-Attend. The software will need to provide unique and useful features to stand out in the market.

1.4.1 Functional Requirements

The following are some functional requirements of I-Attend:

- **User Management:** The software should allow the teacher or professor to create and manage student accounts, as well as add and manage other teachers or professors who can access the system.
- **Attendance Recording:** The software should allow the teacher or professor to record attendance using different methods such as face recognition, geolocation tracing, and QR codes. The software should also be able to handle a large number of students.
- **Attendance Reports:** The software should be able to generate attendance reports for individual students, classes, or periods. The reports should be easily accessible and customizable.
- **Notifications:** The software should notify the teacher or professor of any unusual attendance patterns or discrepancies.
- **User Interface:** The software should provide an easy-to-use interface for both teachers/professors and students. The interface should be accessible through both web and app interfaces.
- **Security:** The software should ensure the privacy and security of student data, including attendance records. The software should also provide protection against unauthorized access.

- **Integration:** The software should be able to integrate with other systems used by the school or college, such as student information systems.
- **Customization:** The software should allow for customization to meet the specific needs of the school or college.

1.4.2 Non-Functional Requirements

The following are some non-functional requirements of I-Attend:

- **Performance:** The software should be fast and responsive, with a low response time, especially during peak usage times.
- **Scalability:** The software should be able to handle a large number of students and teachers, and should be scalable to accommodate future growth.
- **Reliability:** The software should be reliable and available at all times, with a high uptime percentage.
- **Usability:** The software should have a user-friendly interface that is easy to navigate and use.
- **Security:** The software should be secure and protect against unauthorized access, data breaches, and cyber-attacks.
- **Compatibility:** The software should be compatible with a variety of devices and platforms, including mobile devices.
- **Maintainability:** The software should be easy to maintain and update, with minimal downtime for updates and upgrades.
- **Accessibility:** The software should be accessible to users with disabilities, including those who use assistive technologies.
- **Data Privacy:** The software should comply with data privacy regulations and ensure the confidentiality of student data.

1.4.3 Use Case Diagram

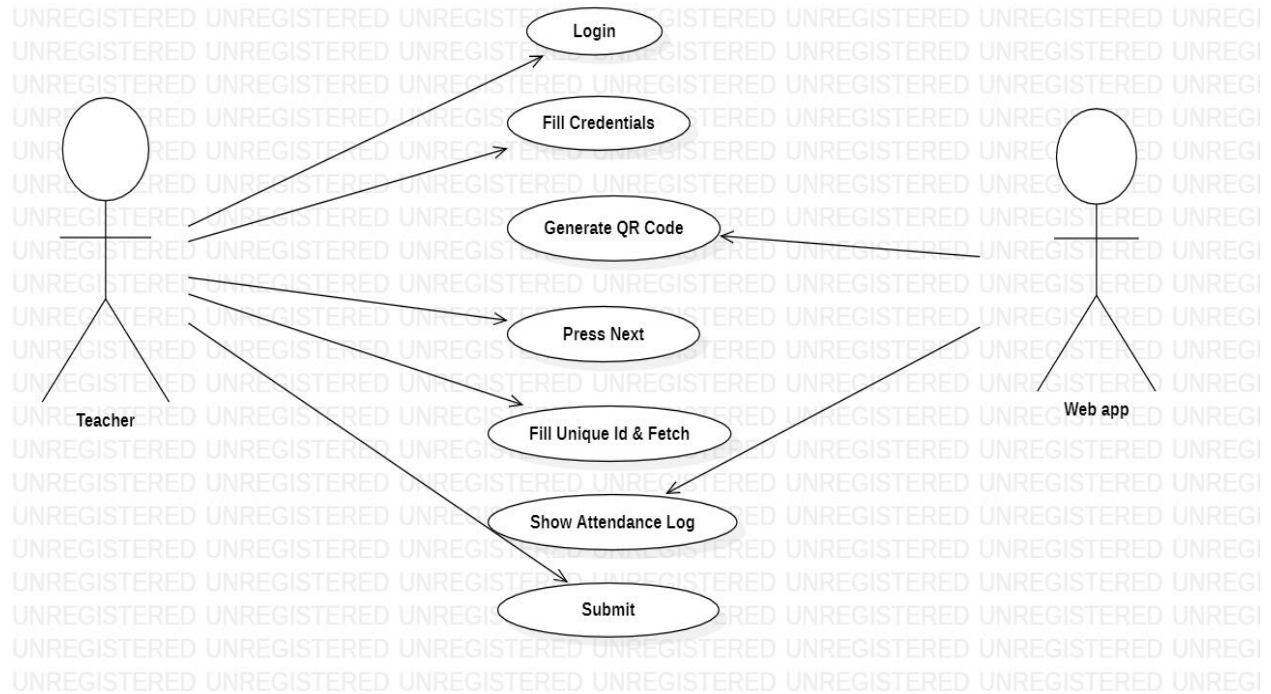


Figure 1.1 Use Case Diagram for Website.

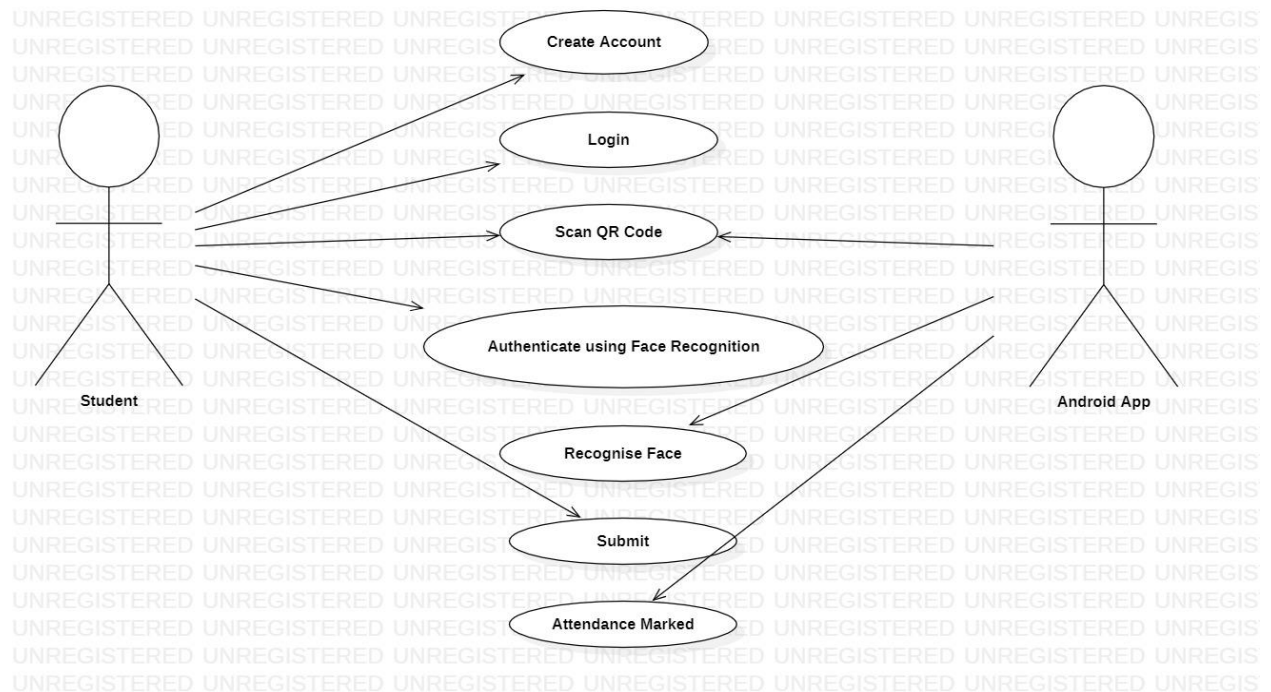


Figure 1.2 Use Case Diagram for Application.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Problem Statement

The problem that I-Attend aims to address is the manual and time-consuming task of taking attendance in schools and colleges. The traditional method of attendance taking involves manually calling out names or roll numbers, which is not only time-consuming but also prone to errors and fraud. Additionally, teachers and professors often have to spend a significant amount of time recording attendance and maintaining attendance records, which can be a burden on their workload. I-Attend solves these problems by automating the attendance recording process and reducing the burden on teachers and professors. It also eliminates the possibility of errors and fraud in attendance records by using modern technologies such as face recognition, geolocation tracing, and QR codes.

2.2 Background and Related Work

2.2.1 Introduction

As for the background work on attendance recording software, traditional methods include manual sign-in sheets, barcode scanning, or RFID tags. However, these methods can be time-consuming, inefficient, and prone to errors. As technology has advanced, new methods such as facial recognition, geolocation tracing, and QR codes have been developed to improve the accuracy and efficiency of attendance tracking.

Several software solutions have been developed in recent years to automate attendance tracking. Some popular examples include Tardy, TeacherKit, and Edsby. These solutions often include features such as cloud-based data storage, mobile compatibility, and customizable attendance tracking settings. However, I-Attend sets itself apart by combining multiple modern technologies to create a comprehensive attendance tracking solution.

2.2.2 Background Work

Automated attendance systems have been in use for several years, but they have primarily relied on barcode scanning, RFID technology, or biometric scanning. However, these methods have their own limitations, such as the need for additional hardware, limited accuracy, and high implementation costs.

With the advancements in facial recognition and geolocation tracing technology, I-Attend offers a more efficient and accurate alternative to traditional attendance-taking methods. Several other attendance management systems have also emerged in recent years, such as Attendance Bot, Prodigy, and Kronos, but I-Attend distinguishes itself with its use of multiple modern technologies for attendance recording and maintenance.

Additionally, with the COVID-19 pandemic causing a shift towards online and hybrid learning models, the need for remote attendance management systems has increased. I-Attend offers a solution for remote attendance taking, as it can be accessed through both web and app interfaces.

2.3 Solution Approach

The solution approach of I-Attend includes the following:

- **Web and App Interfaces:** I-Attend provides both web and app interfaces for attendance recording and maintenance. This allows users to access the software through their preferred platform, making it more convenient for them.
- **Modern Technologies:** The software uses modern technologies such as face recognition, geolocation tracing, and QR codes to record and maintain attendance. These technologies automate the task of attendance recording, making it more accurate and efficient.
- **Time Reduction:** I-Attend reduces the time required to capture attendance by automating the process. This helps teachers and professors to focus more on teaching, rather than spending time on manual attendance recording.
- **Reduced Burden:** The software reduces the burden from the one taking the attendance, as it automates the task. This helps teachers and professors to be more efficient and productive.
- **Customizability:** I-Attend can be customized and modified to suit the needs of different organizations. This makes it more flexible and adaptable to different environments.

Overall, the solution approach of I-Attend focuses on using modern technologies to automate the task of attendance recording and maintenance, while also providing a convenient and customizable platform for users.

CHAPTER 3

DESIGN (UML AND DATA MODELING)

3.1 UML Modeling

Unified Modeling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to **visualize** the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

UML is not a programming language; it is rather a visual language. We use UML diagrams to portray the behaviour and structure of a system. UML helps software engineers, businessmen and system architects with modelling, design and analysis. The Object Management Group (OMG) adopted Unified Modelling Language as a standard in 1997. It's been managed by OMG ever since. International Organization for Standardization (ISO) published UML as an approved standard in 2005. UML has been revised over the years and is reviewed periodically.

3.1.1 Class Diagram

Class diagrams: Class diagrams are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has an appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in their context. It describes various kinds of objects and the static relationship between them.

The main purpose to use class diagrams are:

This is the only UML that can appropriately depict various aspects of the OOPs concept.

Proper design and analysis of applications can be faster and efficient.

It is the base for deployment and component diagram.

There are several software available that can be used online and offline to draw these diagrams Like Edraw max, lucid chart, etc. There are several points to be kept in focus while drawing the class diagram. These can be said as its syntax:

Each class is represented by a rectangle having a subdivision of three compartments name, attributes, and operation.

There are three types of modifiers that are used to decide the visibility of attributes and operations.

+ is used for public visibility (for everyone)

is used for protected visibility (for friend and derived)

– is used for private visibility (for only me)

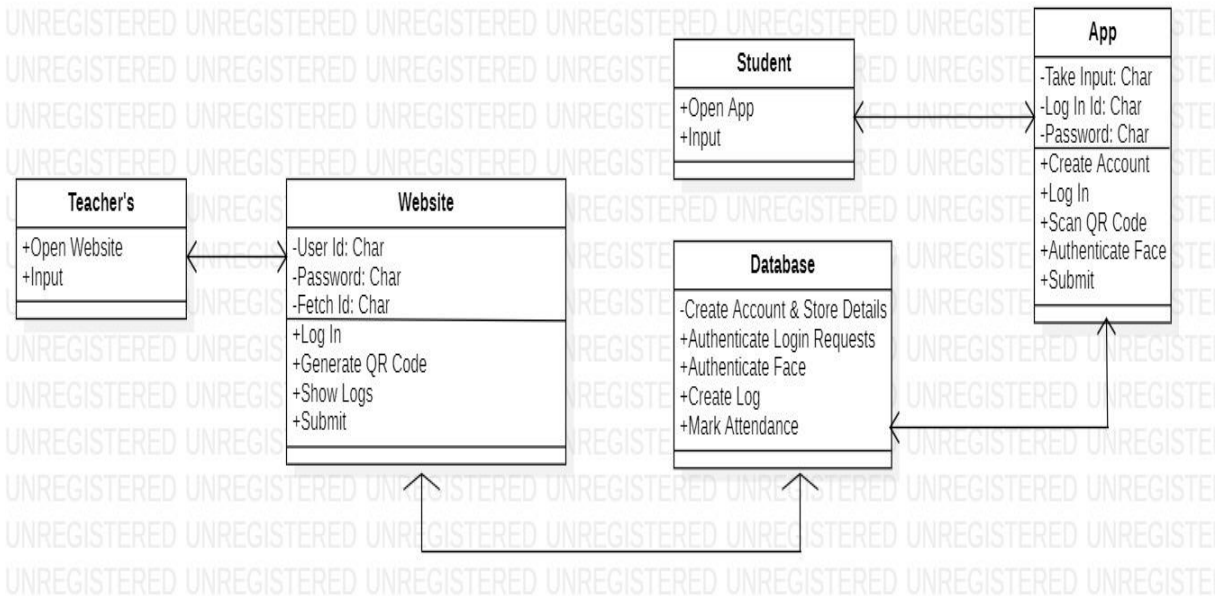


Figure 3.1 Class Diagram of I – Attend

3.1.2 Activity Diagram

We use **Activity Diagrams** to illustrate the flow of control in a system and refer to the steps involved in the execution of a use case. We model sequential and concurrent activities using activity diagrams. So, we basically depict workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. We describe or depict what causes a particular event using an activity diagram. UML models basically three types of diagrams, namely, structure diagrams, interaction diagrams, and behaviour diagrams. An activity diagram is a **behavioural diagram** i.e., it depicts the behaviour of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed. We can depict both sequential processing and concurrent processing of activities using an activity diagram. They are used in business and process modelling where their primary use is to depict the dynamic aspects of a system. An activity diagram is very **similar to a flowchart**. So let us understand if an activity diagrams or flowcharts are any different:

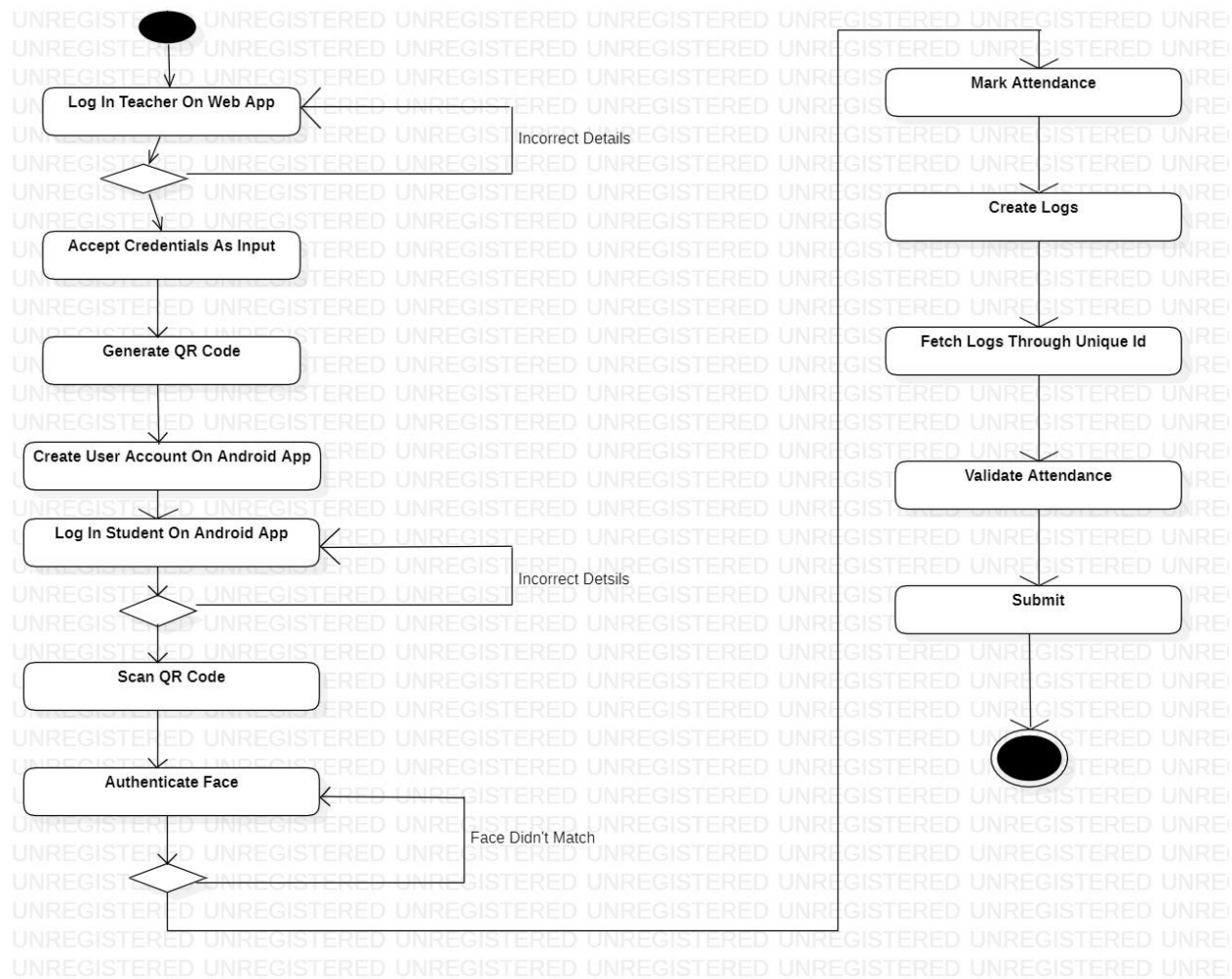


Figure 3.2 Activity Diagram of I – Attend.

3.1.3 Sequence Diagram

Unified Modelling Language (UML) is a modeling language in the field of software engineering which aims to set standard ways to visualize the design of a system. UML guides the creation of multiple types of diagrams such as interaction, structure and behaviour diagrams.

A **sequence diagram** is the most commonly used **interaction** diagram.

Interaction diagram – An interaction diagram is used to show the **interactive behaviour** of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system.

Sequence Diagrams – A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

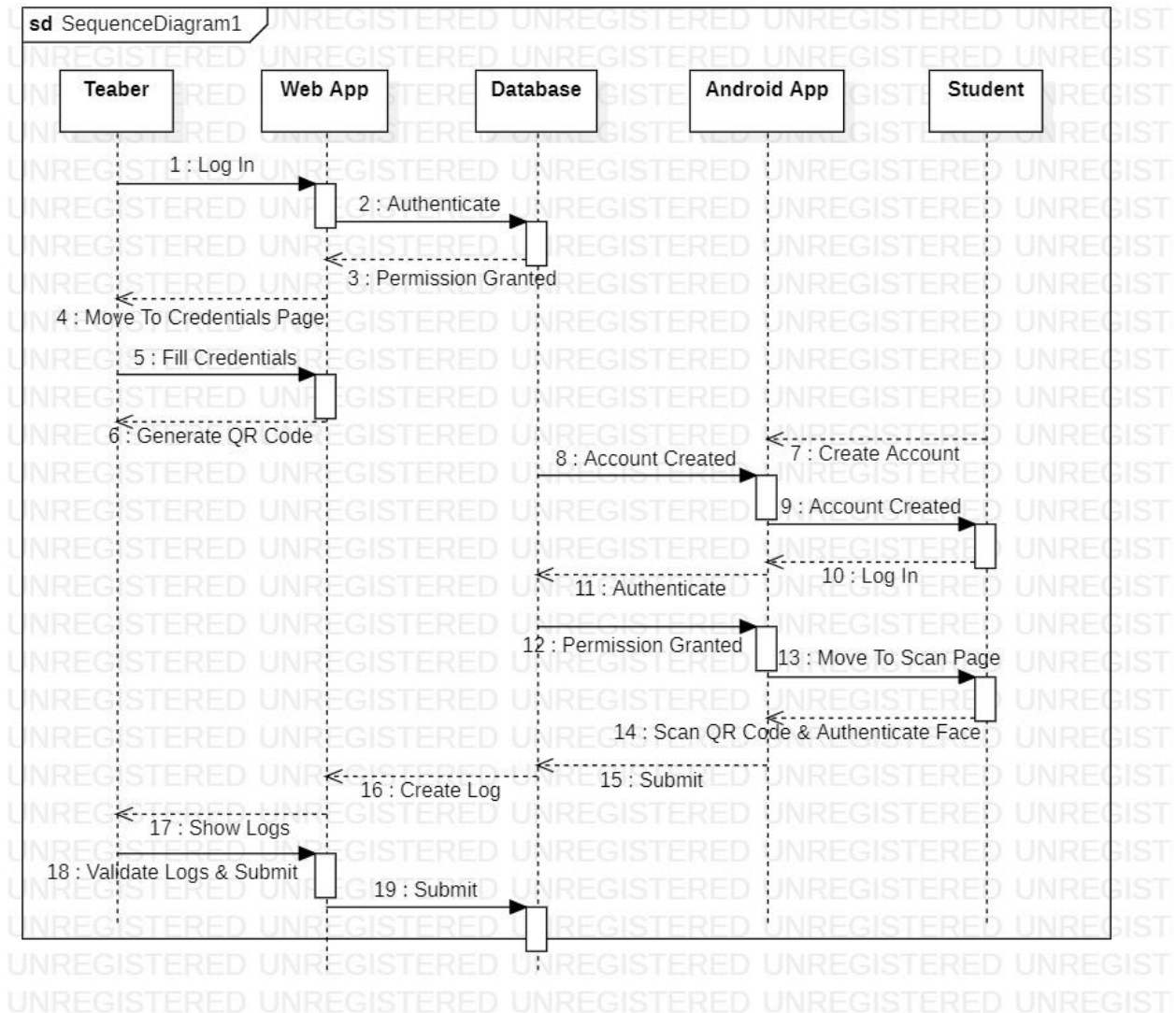


Figure 3.3 Sequence Diagram of I – Attend.

3.2 Data Modeling

Data modeling is the process of creating a visual representation of either a whole information system or parts of it to communicate connections between data points and structures. The goal is to illustrate the types of data used and stored within the system, the relationships among these data types, the ways the data can be grouped and organized and its formats and attributes.

Data models are built around business needs. Rules and requirements are defined upfront through feedback from business stakeholders so they can be incorporated into the design of a new system or adapted in the iteration of an existing one.

Data can be modelled at various levels of abstraction. The process begins by collecting information about business requirements from stakeholders and end users. These business rules are then translated into data structures to formulate a concrete database design. A data model can be compared to a roadmap, an architect's blueprint or any formal diagram that facilitates a deeper understanding of what is being designed.

Data modeling employs standardized schemas and formal techniques. This provides a common, consistent, and predictable way of defining and managing data resources across an organization, or even beyond.

Ideally, data models are living documents that evolve along with changing business needs. They play an important role in supporting business processes and planning IT architecture and strategy. Data models can be shared with vendors, partners, and/or industry peers.

3.2.1 Data Flow Diagram

Data flow diagram is the starting point of the design phase that functionally decomposes the requirements specification. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. A DFD describes what data flow rather than how they are processed, so it does not hardware, software and data structure.

A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). A data flow diagram (DFD) is a significant modeling technique for analysing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. A DFD can be referred to as a Process Model.

The data flow diagram is a graphical description of a system's data and how to Process transform the data is known as Data Flow Diagram (DFD). Unlike details flow chart, DFDs

don't supply detail descriptions of modules that graphically describe a system's data and how the data interact with the system.

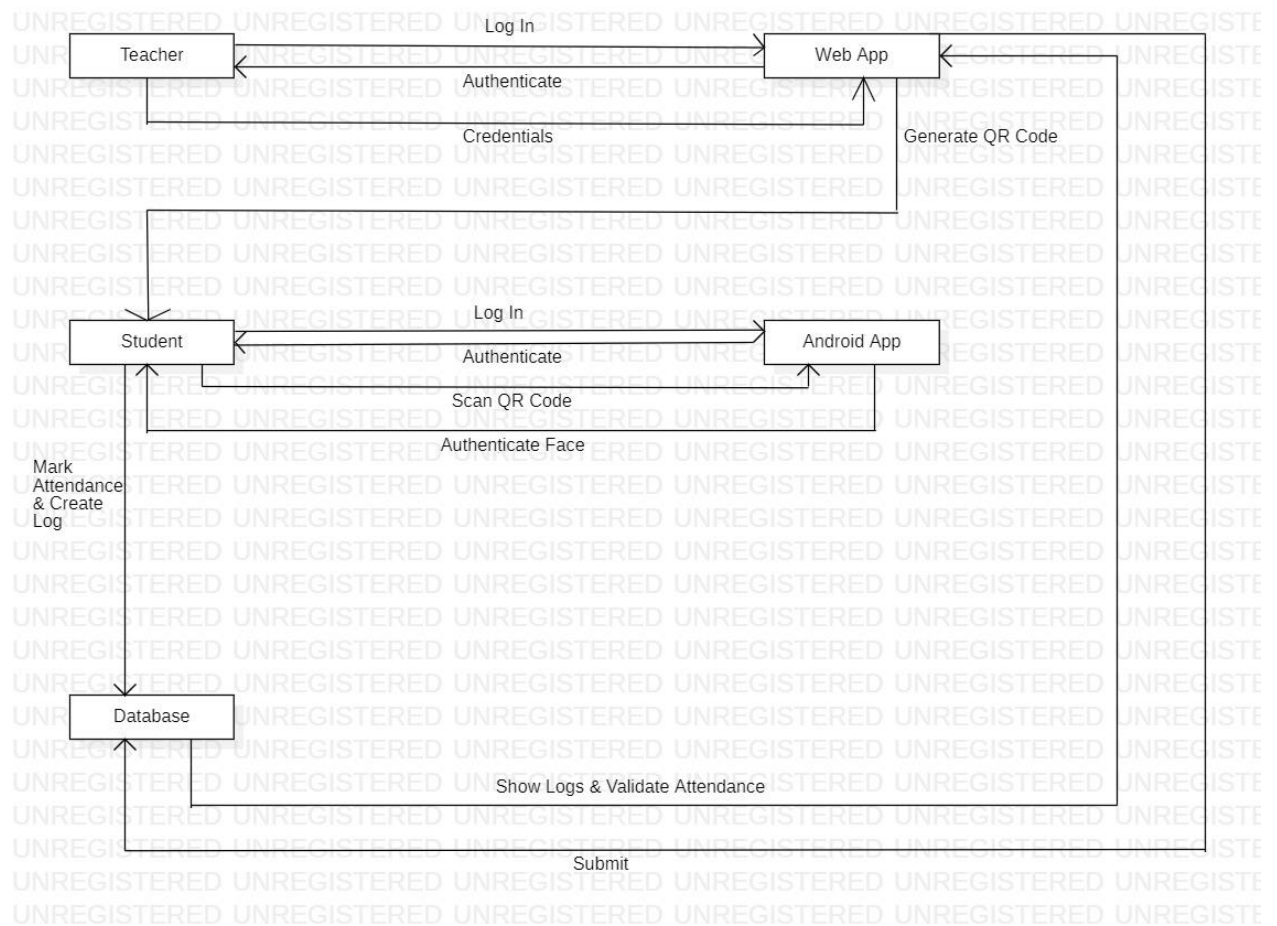


Figure 3.4 Data Flow Diagram of I – Attend.

CHAPTER 4

IMPLEMENTATION

4.1 Tools Used

The specific tools used in I-Attend will depend on the implementation details of the software, but based on the description provided, it is likely that the following tools or technologies could be used:

- **Face recognition software:** This could be a commercial software package or a custom-built solution using open-source libraries such as OpenCV.
- **Geolocation tracking software:** This could be implemented using GPS technology or by using Wi-Fi triangulation to determine a user's location.
- **QR code scanner:** This could be implemented using a smartphone camera or a dedicated QR code scanner.
- **Web interface:** The web interface could be built using a variety of web development tools such as HTML, CSS, JavaScript, and a server-side scripting language such as PHP or Python.
- **Mobile app interface:** The mobile app interface could be built using native mobile development tools such as Swift for iOS or Kotlin for Android, or using cross-platform development tools such as React Native or Flutter.

4.2 Technologies

I-Attend uses various modern technologies for attendance recording and maintenance, including:

- **Face recognition technology:** This technology uses facial recognition algorithms to identify students or attendees and mark their attendance.
- **Geolocation tracing:** It uses GPS-enabled devices to track the location of the attendees and record attendance based on their location.

- **QR codes:** This technology uses QR codes to mark attendance. Each student or attendee has a unique QR code, and they have to scan it to mark their attendance.
- **Web and app interfaces:** I-Attend uses both web and app interfaces for attendance recording and maintenance, providing a flexible and easy-to-use platform.
- **Firebase storage:** The attendance data is stored in the Firebase, making it easily accessible and secure.

4.3 Testing

To ensure the quality and reliability of I-Attend, various types of testing can be used. Here are some examples:

- **Functional testing:** This type of testing ensures that the software meets the functional requirements as specified in the design document. This can include testing features such as attendance recording, maintenance, and reporting, as well as user authentication and authorization.
- **Performance testing:** This type of testing ensures that the software performs well under various conditions, such as heavy load and high traffic. This can include testing the speed of attendance recording and retrieval, as well as the response time of the system.
- **Security testing:** This type of testing ensures that the software is secure and can protect sensitive data such as attendance records and user credentials. This can include testing for vulnerabilities such as SQL injection, cross-site scripting, and other types of attacks.
- **Usability testing:** This type of testing ensures that the software is easy to use and understand for the intended users. This can include testing the user interfaces of both the web and app interfaces, as well as conducting user surveys and interviews to gather feedback on the usability of the software.
- **Compatibility testing:** This type of testing ensures that the software is compatible with different platforms and devices, such as different web browsers and mobile devices.

Overall, a comprehensive testing strategy is crucial for ensuring the quality and reliability of I-Attend, and the above testing types are just a few examples of what can be included in such a strategy.

4.4 User Guide

- **Installation:** First, install the software either on your computer or mobile device.
- **Set up:** Once the software is installed, set it up according to your requirements, such as adding the names of your students or employees.
- **Select the mode of attendance recording:** Select the mode of attendance recording, which could be face recognition, geolocation tracing, QR codes, or any other mode supported by the software.
- **Start recording attendance:** Start the attendance recording process by selecting the appropriate class or group and marking the attendance.
- **Generate reports:** Once the attendance is recorded, generate reports for individual students or employees, or for the entire class or group.
- **Analyse the attendance data:** Analyse the attendance data to identify patterns and trends, and take appropriate actions to improve attendance.

It is important to note that the exact steps to use I-Attend may vary depending on the specific features and functionalities of the software. Therefore, it is recommended to refer to the user guide or documentation provided by the software vendor for detailed instructions.

CHAPTER 5

PROJECT PLAN

5.1 Project Planning:

- **Define the project scope and objectives:** Clearly define what the project aims to achieve and what functionalities are required in the software.
- **Identify stakeholders:** Identify all the stakeholders involved in the project, including end-users, administrators, IT team, and other stakeholders.
- **Define project timeline:** Develop a timeline for the project that outlines when key milestones are expected to be achieved.
- **Develop a project team:** Determine the team members who will be working on the project and their roles and responsibilities.
- **Allocate resources:** Allocate resources such as budget, hardware, software, and other required resources.
- **Determine project risks:** Identify potential risks to the project and develop a risk management plan.
- **Define project metrics:** Determine the metrics that will be used to measure the success of the project.
- **Develop a project schedule:** Develop a detailed project schedule that outlines all tasks required to complete the project and the timeline for each task.
- **Execute the project plan:** Execute the plan while monitoring progress against the project schedule and metrics.
- **Evaluate the project:** Evaluate the project to ensure that it meets the project scope and objectives, and identify opportunities for improvement.

5.2 Effort and Cost estimation

Estimating the effort and cost for developing a software like I-Attend can be a complex task as it depends on various factors such as the size of the development team, the scope of the project, the complexity of the features, the time required for testing and debugging, and the cost of the required hardware and software.

Here are some of the factors that can affect the effort and cost estimation for I-Attend:

- **Size of the development team:** The size of the development team required to develop I-Attend will depend on the scope of the project and the time frame for completion. A larger team can help to reduce the development time, but it will also increase the cost.
- **Scope of the project:** The scope of the project will determine the features that need to be included in the software. The more complex the features, the more effort will be required to develop them.
- **Complexity of the features:** Features like face recognition, geolocation tracing, and QR codes are complex and require a significant amount of effort to implement. The complexity of the features will also impact the testing and debugging effort.
- **Time required for testing and debugging:** Testing and debugging are critical parts of software development, and they require a significant amount of effort. The time required for testing and debugging will depend on the complexity of the features and the size of the development team.
- **Cost of hardware and software:** Developing a software like I-Attend requires specific hardware and software resources. The cost of these resources can impact the overall cost of the project.

Based on these factors, the effort and cost estimation for I-Attend can range from moderate to high, depending on the specific requirements of the project. It is important to perform a detailed analysis of the requirements and the resources needed to develop the software to get an accurate estimate of the effort and cost.

CHAPTER 6

PROJECT SCREENSHOT

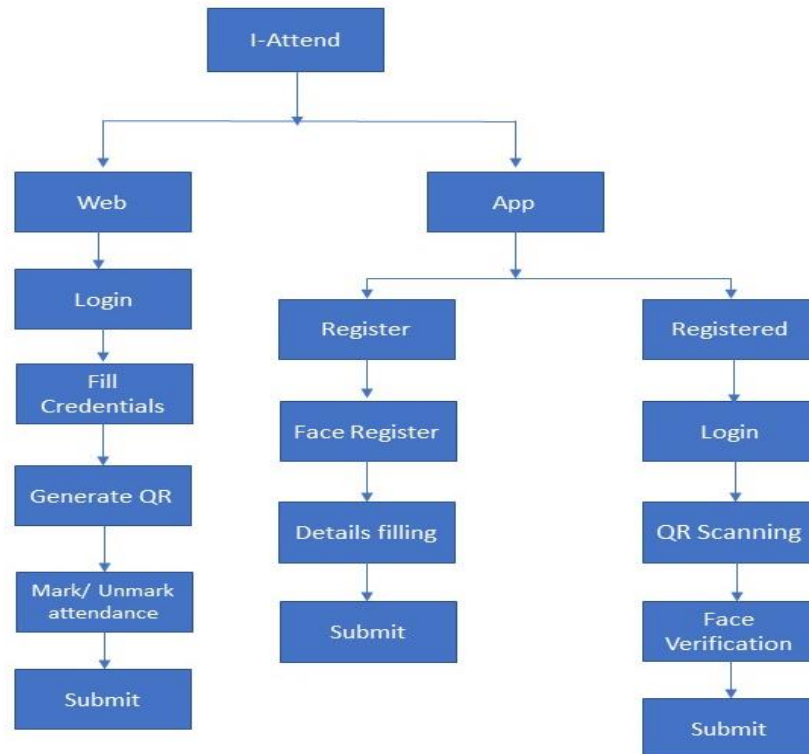


Figure 6.1 Flow Chart of I – Attend.

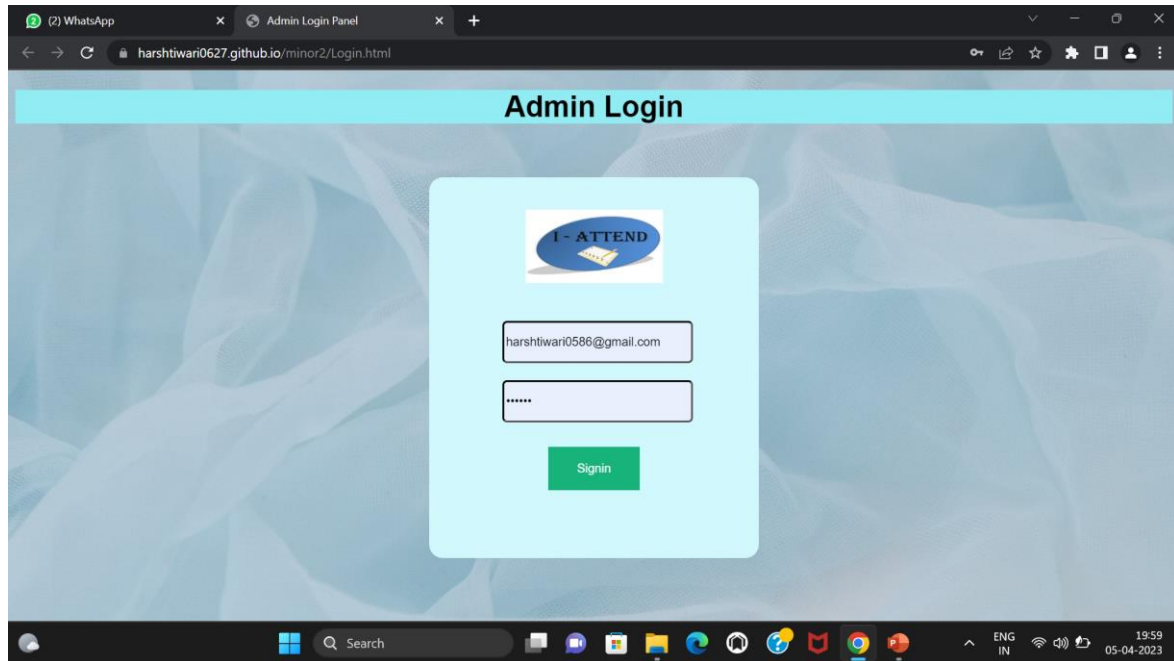


Figure 6.2 Login Page for Admin.

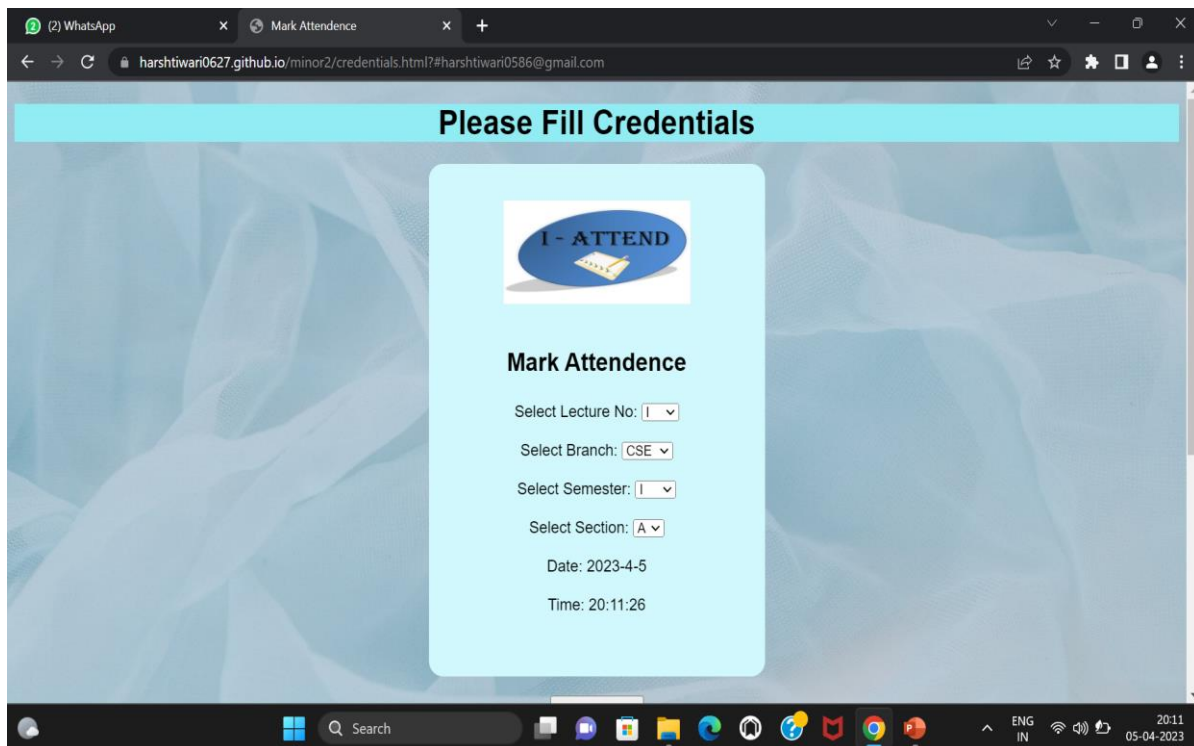


Figure 6.3 Credentials Page for Admin.

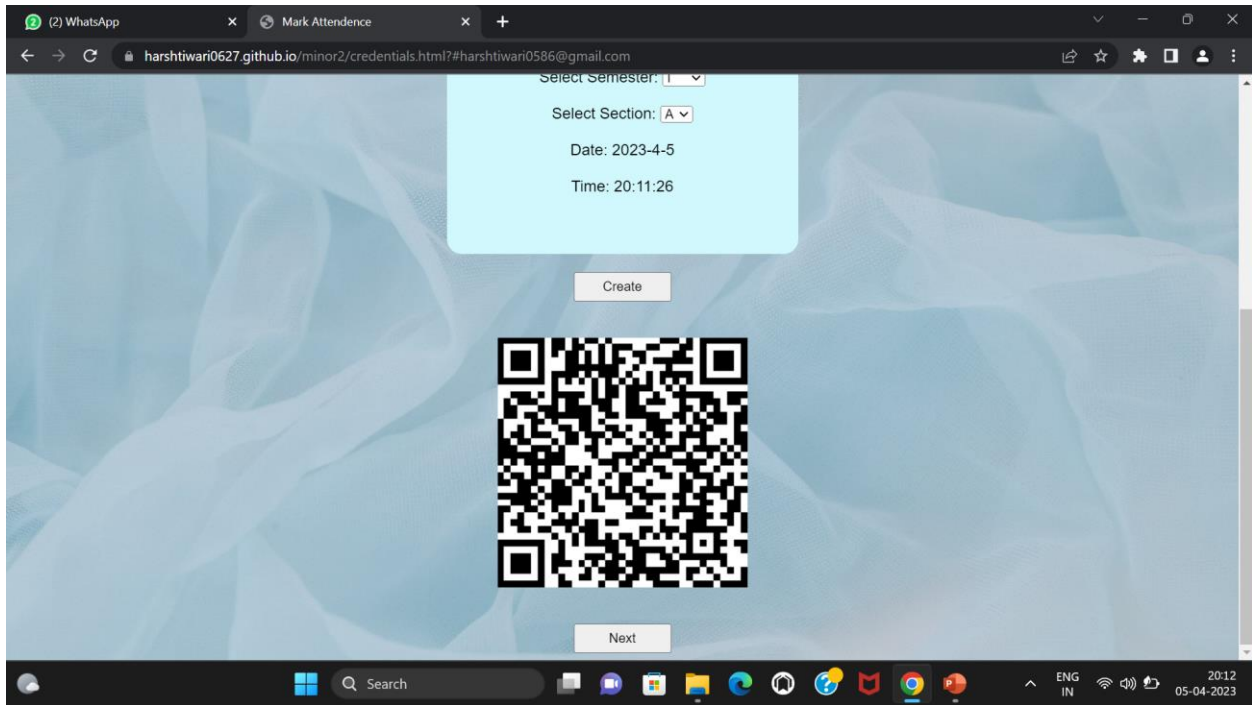
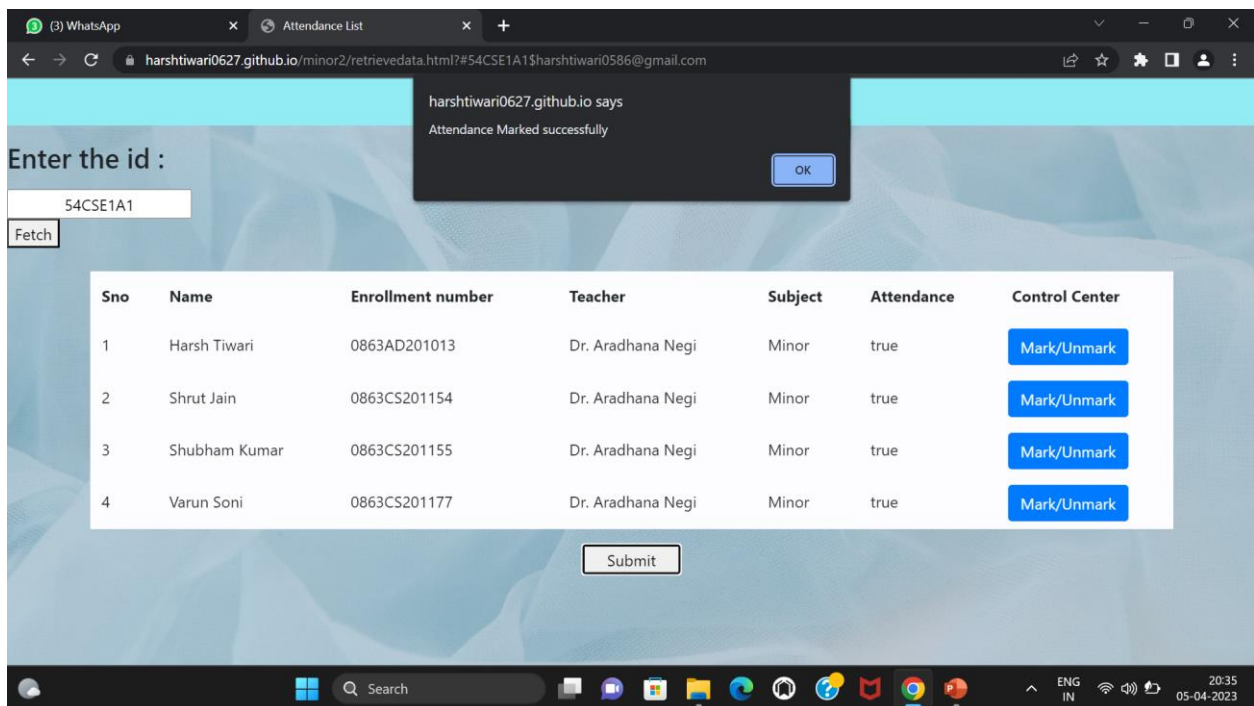


Figure 6.4 QR Code Generator for Student by Admin.



6.5 Mark Attendance by Admin.

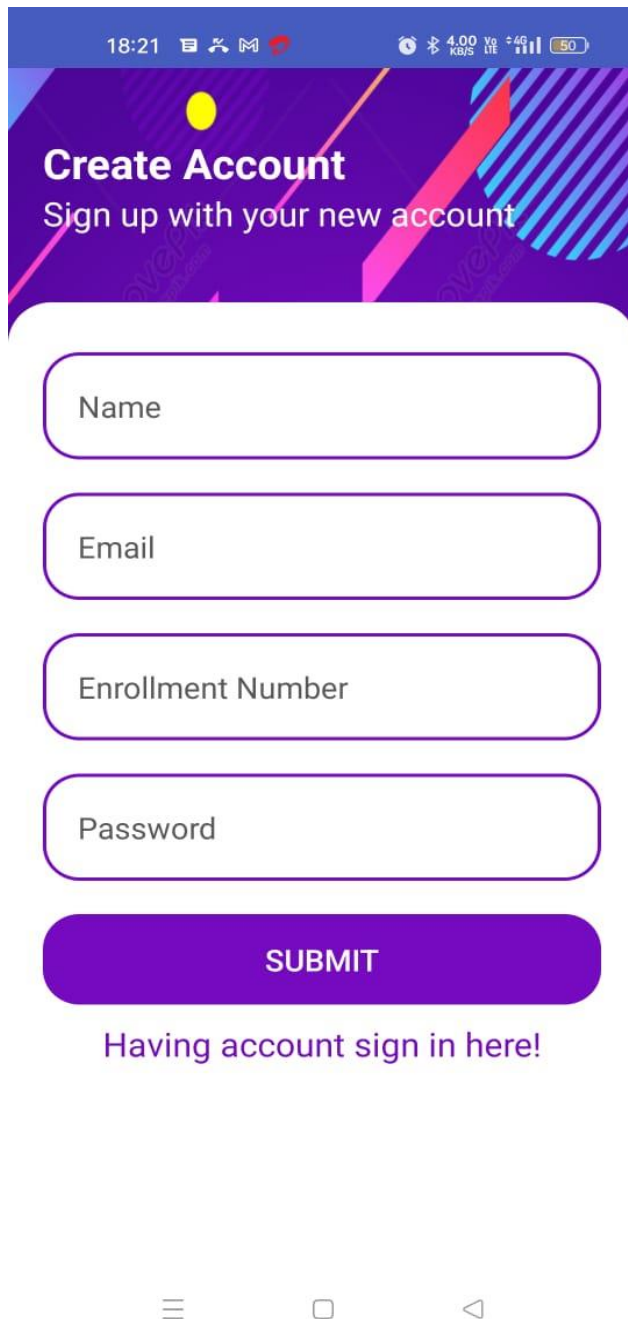


Figure 6.6 Account Creation to App by Student.

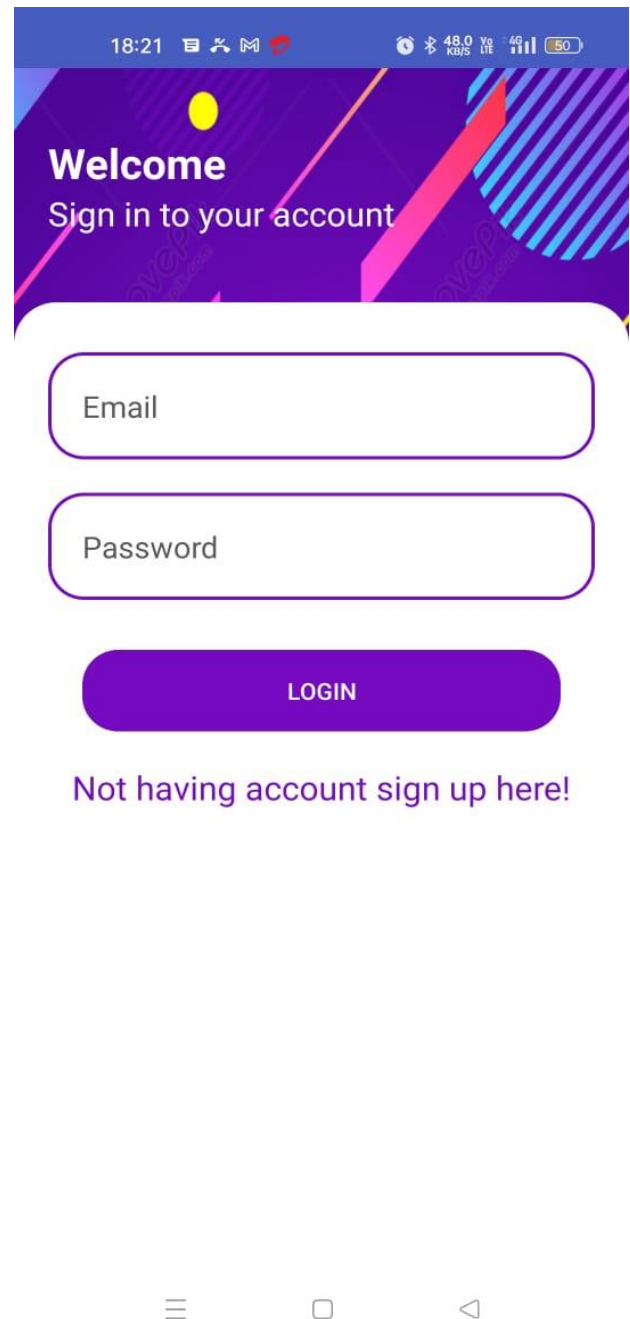


Figure 6.7 Log in To App by Student

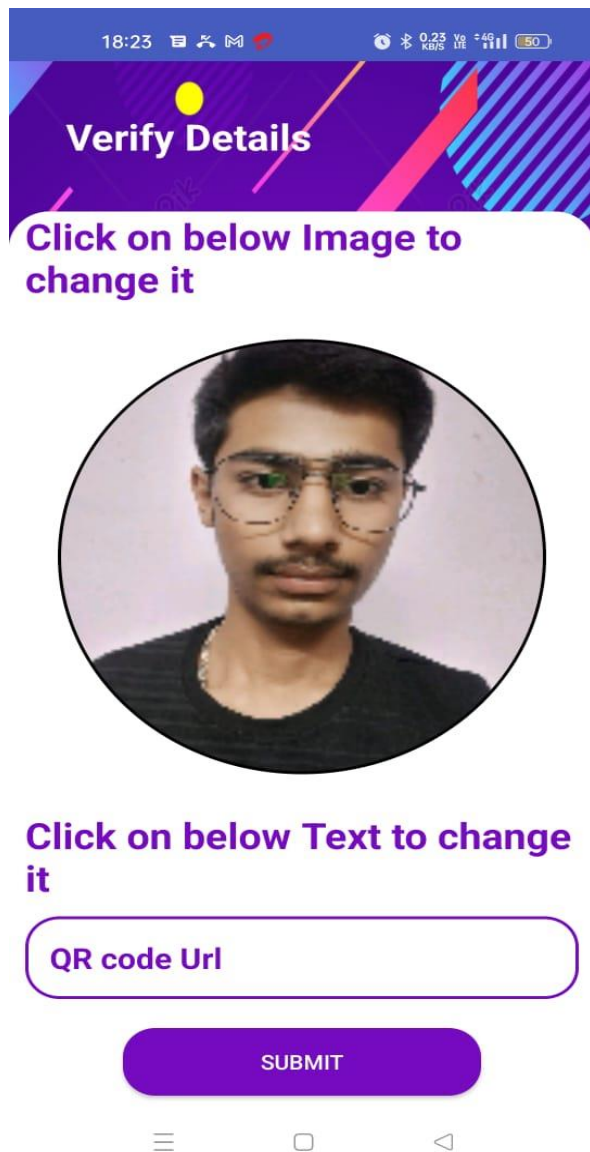


Figure 6.8 Face Authentication to App by Student

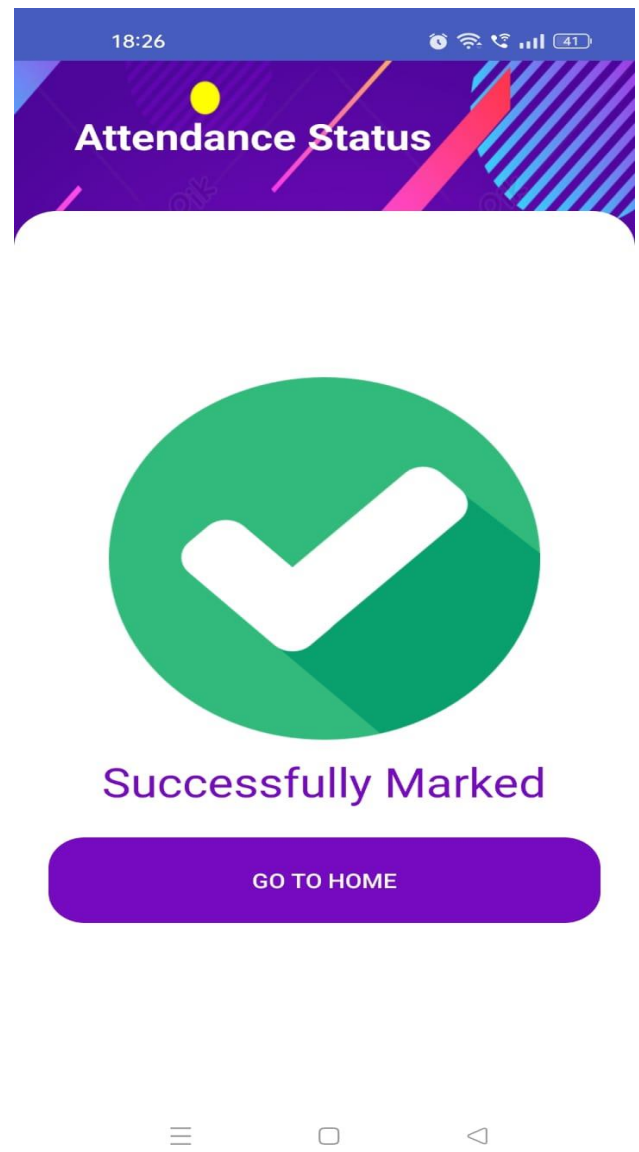


Figure 6.9 Attendance Marked by APP

CHAPTER 7

CONCLUSION/ FUTURE SCOPE

7.1 Conclusion

The Project successfully delivered on all requirements specification specified by the user. Care was ensured during the design to make sure data integrity is maintained and to avoid all forms of redundancies associated with the data. This service has a very user-friendly interface for both Android app and Website, which is using JavaScript to fetch data about the Attendance details. This project can be further modified for using the geolocation service for verifying that the client is inside the premises or not. The technical document that is provided in the report of this project will help developers understand the internal workings of this project.

7.1.1 Objective Assessment

After building the Project, we achieved these objectives:

- I-Attend uses modern AI Technologies to capture and record attendance.
- It automates the task of attendance capturing, and thus reducing the efforts of the one taking the attendance and the one attending the class or the session.
- It also reduces the attendance capturing time of the teacher or the one taking the attendance.
- It is less prone to the false present or false absent, marked by mistake, of any candidate or the student.

The android application can be used by the users and admin can use the website for generation unique QR code and can fetch attendance details from the same.

7.1.2 Limitation and Challenges

We have some limitation which are as follows:

- We cannot identify the correct geolocation of the candidate, i.e., if he/she is physically attending the class or not. Although we can check whether he/she is in the organization premises or not with GPS but exact location of the classroom cannot be determined, for now.

7.2 Future Scope

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

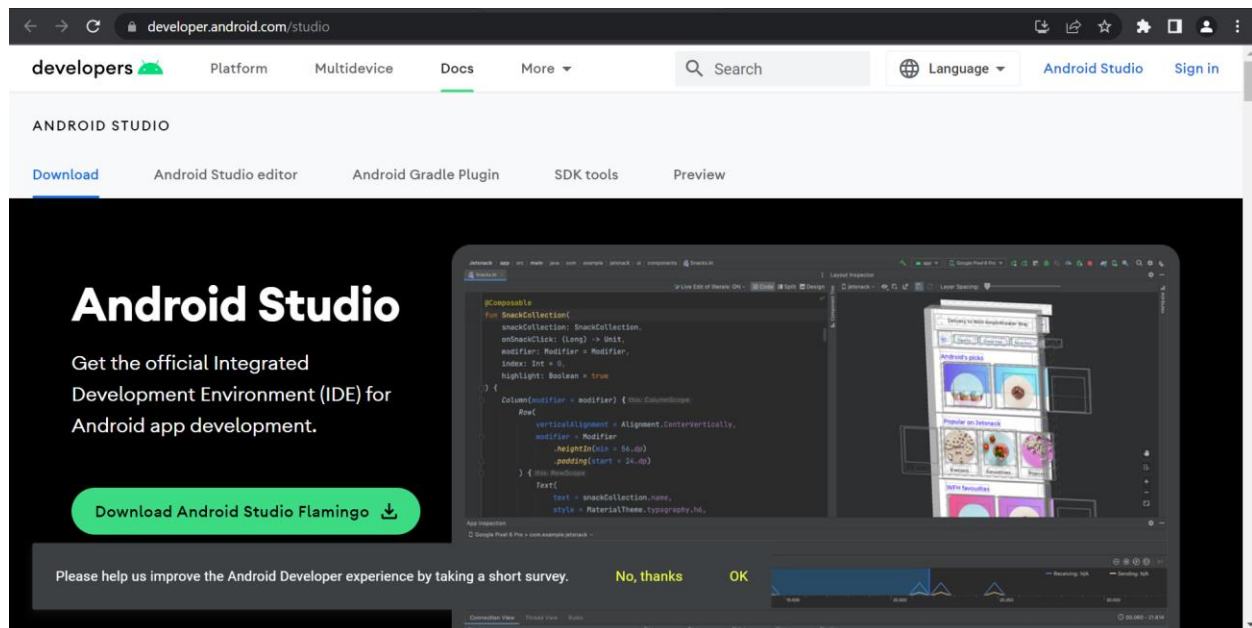
- We can make use of geolocation for verifying whether the user is inside the premises or not.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.

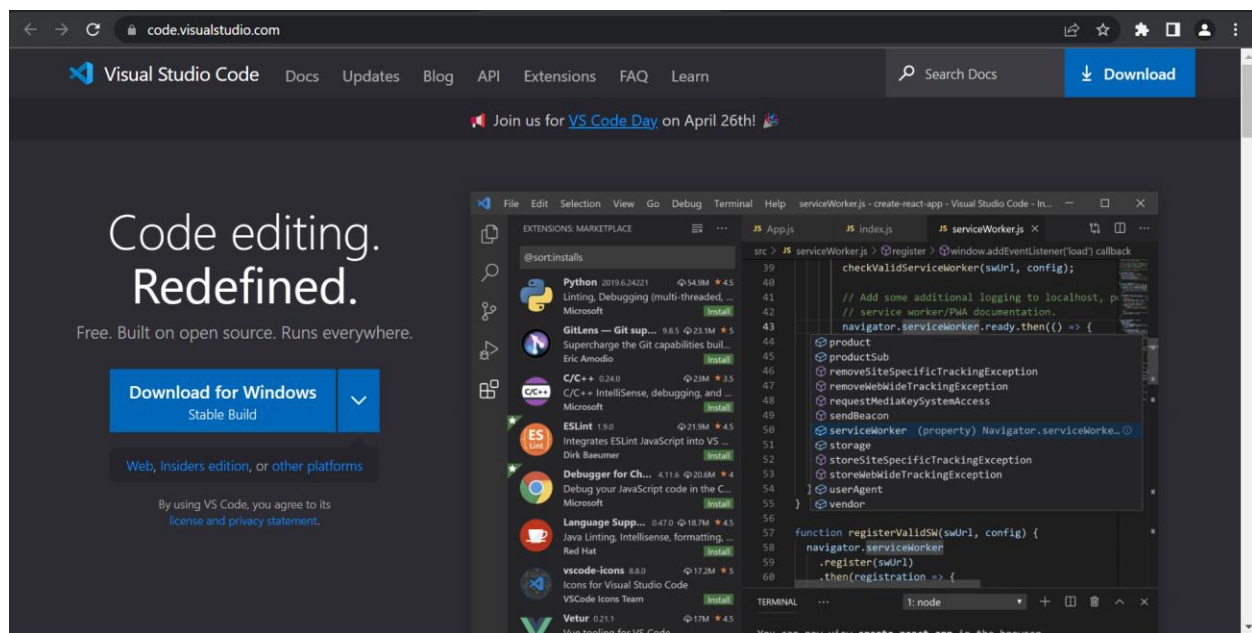
In the last we would like to thank all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is developed here by underlining success of process.

Appendix – A

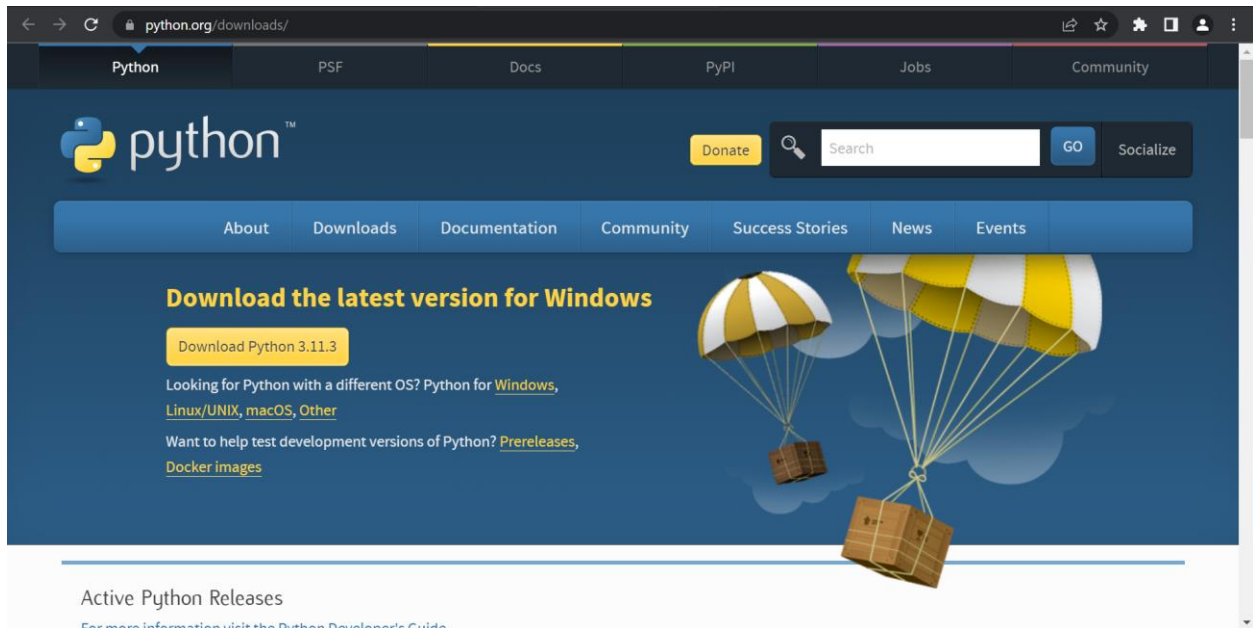
Android Studio: [Download Android Studio & App Tools - Android Developers](https://developer.android.com/studio)



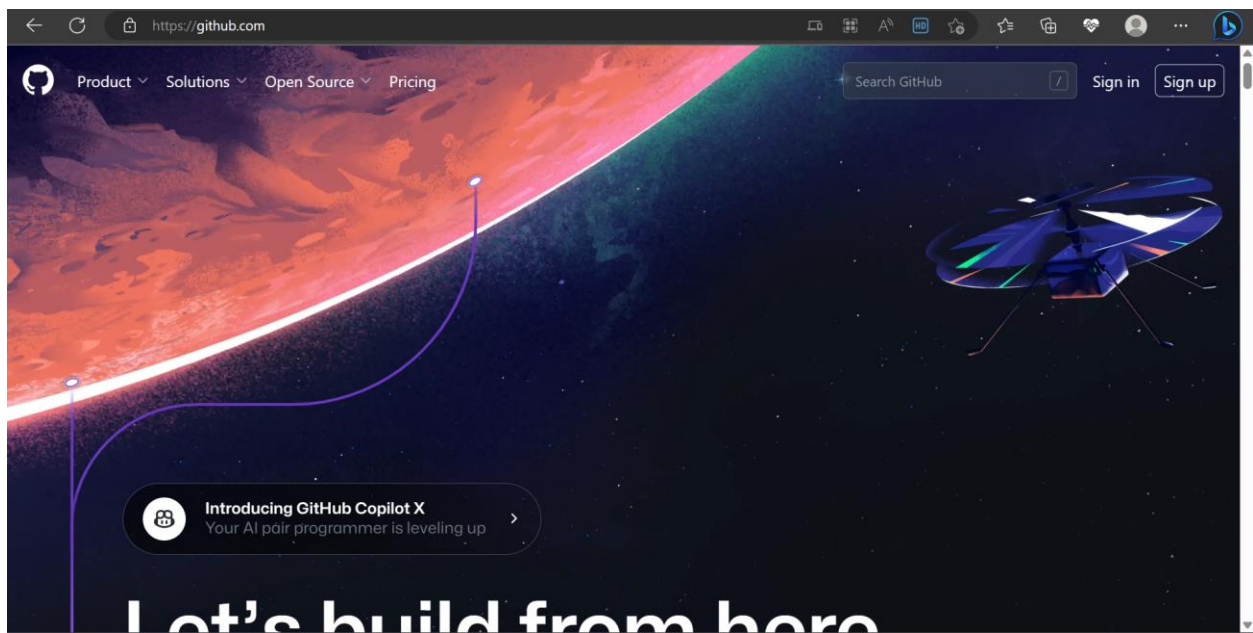
Visual Studio: [Visual Studio Code - Code Editing. Redefined](https://code.visualstudio.com)



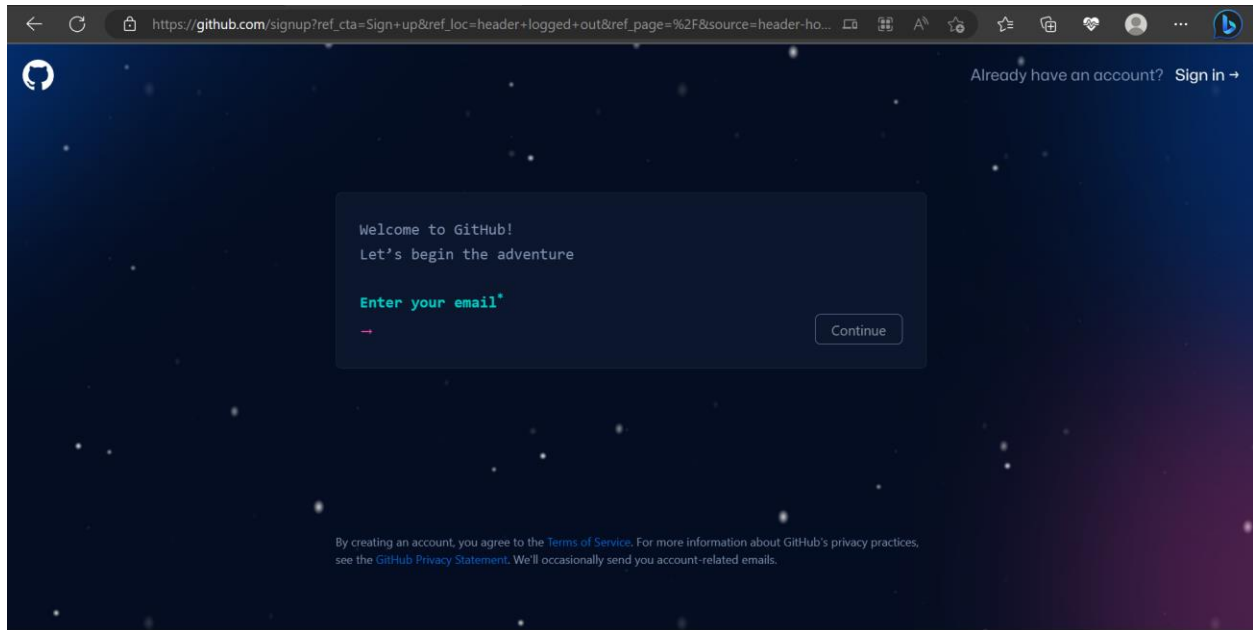
Python: [Download Python | Python.org](https://python.org/downloads/)



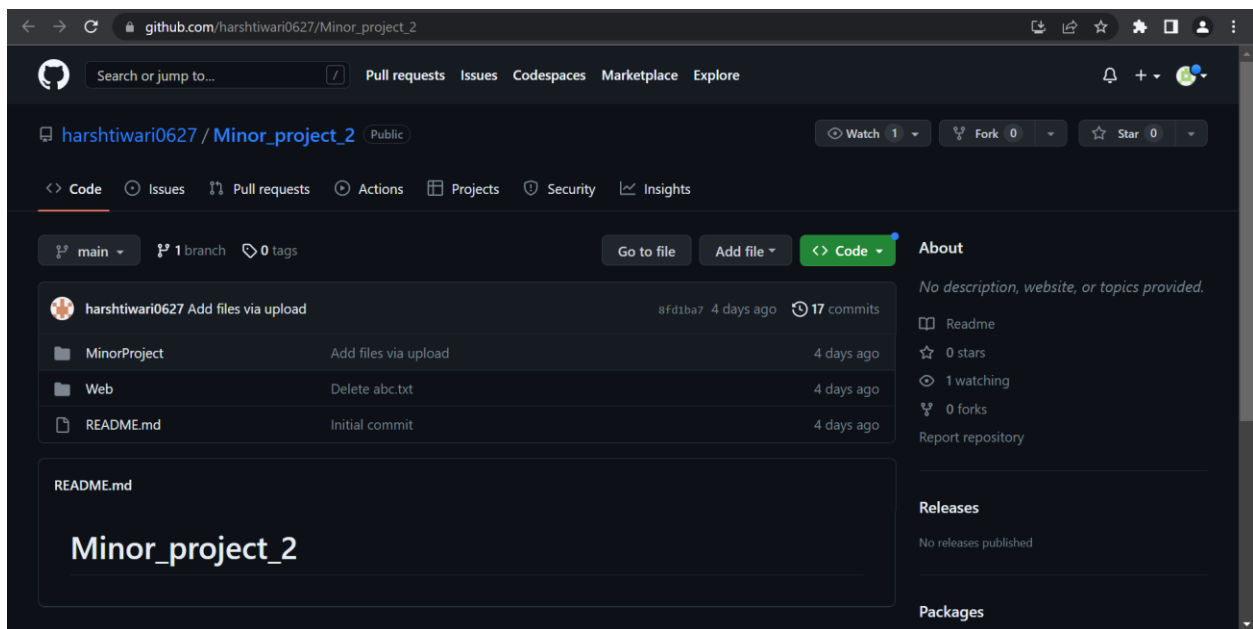
GitHub: [GitHub: Let's build from here · GitHub](https://github.com)



GitHub Sign in Page: [Join GitHub · GitHub](https://github.com)



GitHub Project Repository: [Visit here](#)



Android App Link: [Download here](#)

