****

**PROJECT REPORT**

**OF**

**MINOR PROJECT**

**ON**

**ATTENDIFY-ATTENDANCE MANAGEMENT APPLICATION**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE**

**OF**

**BACHELOR OF TECHNOLOGY**

**IN**

**Computer Science and Engineering**

**Submitted by:**

**Name: Daksh Verma**

**Roll No.: 12101049**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**PUNJABI UNIVERSITY**

**PATIALA – 147002**

**ABSTRACT**

In the dynamic academic environment of colleges and universities, effective attendance management plays a pivotal role in fostering student success. Attendify, an innovative Android application, addresses this crucial aspect by providing college students with a comprehensive tool to manage their attendance seamlessly. This report delves into the key features, functionalities, and potential impact of Attendify, highlighting its significance in enhancing academic performance and empowering students to take control of their learning journey.

Attendify's user-centric design prioritizes simplicity and intuitive navigation, ensuring effortless interaction even for tech-averse users. The implementation of Attendify is expected to yield significant benefits for college students, including improved academic performance, reduced absenteeism, and enhanced engagement in classroom activities.

Its user-friendly design, comprehensive features, and potential impact make it an invaluable asset for students seeking to optimize their college experience and achieve their full potential.

**TABLE OF CONTENTS**

**CHAPTERS Page No.**

**CH-1. INTRODUCTION 4-10**

1.1. Introduction to Technologies Used 4

1.2. What is Android 5-7

1.3. Brief introduction to the Android Studio 7-10

**CH-2. WORK UNDERTAKEN 11-15**

2.1. Installation and setup of JDK 11-15

2.2. Downloading and installing Android Studio 15

**CH-3. RESULTS 16-19**

3.1. Table of Project’s Screenshots 16-18

3.2. Features of the Project 19

**CH-4. CONCLUSION AND FUTURE SCOPE 20**

4.1. Conclusion 20

4.2. Future Scope 20

**CH-1. INTRODUCTION**

**1.1. Introduction to Technologies Used**

**1.1.1. XML**

XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax.

There are many languages based on XML, including XHTML, MathML, SVG, RSS, and RDF. You can also define your own.

Structure of an XML document

The whole structure of XML and XML-based languages is built on tags.

**XML declaration**

XML - declaration is not a tag. It is used for the transmission of the meta-data of a document.

<?xml version="1.0" encoding="UTF-8"?>

**Attributes**

version - Used version XML in this document.

encoding - Used encoding in this document.

**Example XML :-**

<?xml version="1.0" encoding="UTF-8"?>

<message>

<warning>

Hello World

</warning>

</message>

**1.1.2. Java**

Java is a most popular, object-oriented, widely used programming language and platform that is utilized for Android development, web development, artificial intelligence, cloud applications, and much more. Java is a class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is intended to let application developers write once, and run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java was first released in 1995 and is widely used for developing applications for desktop, web, and mobile devices. Java is known for its simplicity, robustness, and security features, making it a popular choice for enterprise-level applications.

Every line of code that runs in Java must be inside a class. A class should always start with an uppercase first letter. The name of the java file **must match** the class name. When saving the file, save it using the class name and add ".java" to the end of the filename.

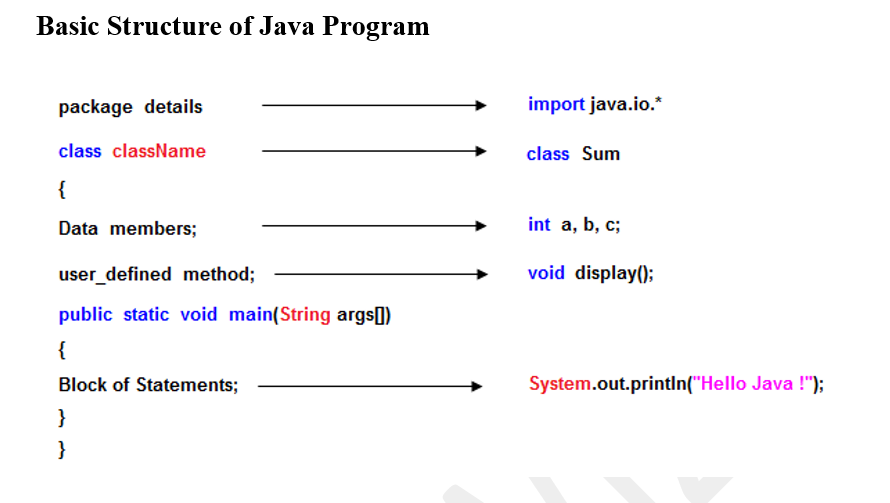
The main () method public static void **main** (String [] args) is required in every Java program

Any code inside the main () method will be executed first.

A Java program typically consists of the following elements:

* **Documentation section**: optional comments that describe the purpose and functionality

of the program.

* **Package declaration**: optional statement that specifies the package name of the program.
* **Import statements**: optional statements that import other classes or packages that are used by the program.
* **Interface section**: optional section that defines interfaces that are implemented by the program or its classes.
* **Class definition**: mandatory section that defines one or more classes that make up the program. Each class has a name, variables, and methods.
* **Main method**: mandatory method that is the entry point of the program. It has the signature public static void main (String [] args) and contains the logic of the program.

**Advantages of Java:**

**1. Simple:** Java is a simple programming language since it is easy to learn and easy to

understand. Its syntax is based on C++, andit uses automatic garbage collection; therefore,

we don't need to remove the unreferenced objects from memory. Javahas also removed the

features like explicit pointers, operator overloading, etc., making it easy to read and write.

**2. Object-Oriented**: Java uses an object-oriented paradigm, which makes it more practical.

Everything in Java is an object which takes care of both data and behaviour. Java uses object-

oriented concepts like object, class, inheritance, encapsulation, polymorphism, and

abstraction.

**3. Secured:** Java is a secured programming language because it doesn't use Explicit pointers.

Also, Java programs run inside thevirtual machine sandbox. JRE also provides a class loader,

which is used to load the class into JVM dynamically. It separates the class packages of the

local file system from the ones that are being imported from the network.

**4. Robust:** Java is a robust programming language since it uses strong memory management.

We can also handle exceptions through the Java code. Also, we can use type checking to

make our code more secure. It doesn't provide explicit pointers so that the programmer cannot

access the memory directly from the code.

**5. Platform independent:** Java code can run on multiple platforms directly, I.e., we need not

compile it every time. It is right once, runs anywherelanguage (WORA) which can be

converted into byte code at the compile time. The byte code is a platform-independent code

that can run on multiple platforms.

**6. Multi-Threaded**: Java uses a multi-threaded environment in which a bigger task can be

converted into various threads and runseparately. The main advantage of multi-threading is

that we need not provide memory to every running thread.

**1.2. What is Android**

**1.2.1. Android**

Android is a mobile operating system based on a modified version of the Linux kernel and other open-source software, designed primarily for touchscreen mobile devices such as smartphones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, though its most widely used version is primarily developed by Google. It was unveiled in November 2007, with the first commercial Android device, the HTC Dream, being launched in September 2008.

## At its core, the operating system is known as Android Open Source Project (AOSP) and is free and open-source software (FOSS) primarily licensed under the Apache License. However most devices run on the proprietary Android version developed by Google, which ship with additional proprietary closed-source software pre-installed, most notably Google Mobile Services (GMS) which includes core apps such as Google Chrome, the digital distribution platform Google Play, and the associated Google Play Services development platform. Firebase Cloud Messaging is used for push notifications. While AOSP is free, the "Android" name and logo are trademarks of Google, which imposes standards to restrict the use of Android branding by "uncertified" devices outside their ecosystem.

## 1.2.2. Android Components

**1. Activities**

Activities are said to be the presentation layer of our applications. The UI of our application is built around one or more extensions of the Activity class. By using Fragments and Views, activities set the layout and display the output and also respond to the user’s actions. An activity is implemented as a subclass of class Activity.

**2. Services**

Services are like invisible workers of our app. These components run at the backend, updating your data sources and Activities, triggering Notification, and also broadcast Intents. They also perform some tasks when applications are not active. A service can be used as a subclass of class Service

**3. Content Providers**

It is used to manage and persist the application data also typically interacts with the SQL database. They are also responsible for sharing the data beyond the application boundaries. The Content Providers of a particular application can be configured to allow access from other applications, and the Content Providers exposed by other applications can also be configured. A content provider should be a sub-class of the class ContentProvider.

**4. Broadcast Receivers**

They are known to be intent listeners as they enable your application to listen to the Intents that satisfy the matching criteria specified by us. Broadcast Receivers make our application react to any received Intent thereby making them perfect for creating event-driven applications

**5. Intents**

It is a powerful inter-application message-passing framework. They are extensively used throughout Android. Intents can be used to start and stop Activities and Services, to broadcast messages system-wide or to an explicit Activity, Service or Broadcast Receiver or to request action be performed on a particular piece of data.

**6. Widgets**

These are the small visual application components that you can find on the home screen of the devices. They are a special variation of Broadcast Receivers that allow us to create dynamic, interactive application components for users to embed on their Home Screen.

**7. Notifications**

Notifications are the application alerts that are used to draw the user’s attention to some particular app event without stealing focus or interrupting the current activity of the user. They are generally used to grab user’s attention when the application is not visible or active, particularly from within a Service or Broadcast Receiver. Examples: E-mail popups, Messenger popups, etc.

**1.3. Brief introduction to the Android Studio**

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013, at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. At the end of 2015, Google dropped support for Eclipse ADT, making Android Studio the only officially supported IDE for Android development.

Android Studio provides many excellent features that enhance productivity when building Android apps, such as:

* A blended environment where one can develop for all Android devices
* Apply Changes to push code and resource changes to the running app without restarting the app
* A flexible Gradle-based build system
* A fast and feature-rich emulator
* GitHub and Code template integration to assist you to develop common app features and import sample code
* Extensive testing tools and frameworks
* Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine, and many more.
* Provides GUI tools that simplify the less interesting parts of app development.
* Easy integration with real time database ‘firebase’.

**System Requirements**

* Microsoft Windows 7/8/10 (32-bit or 64-bit)
* 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator) 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
* 1280 x 800 minimum screen resolution

**CH-2. WORK UNDERTAKEN**

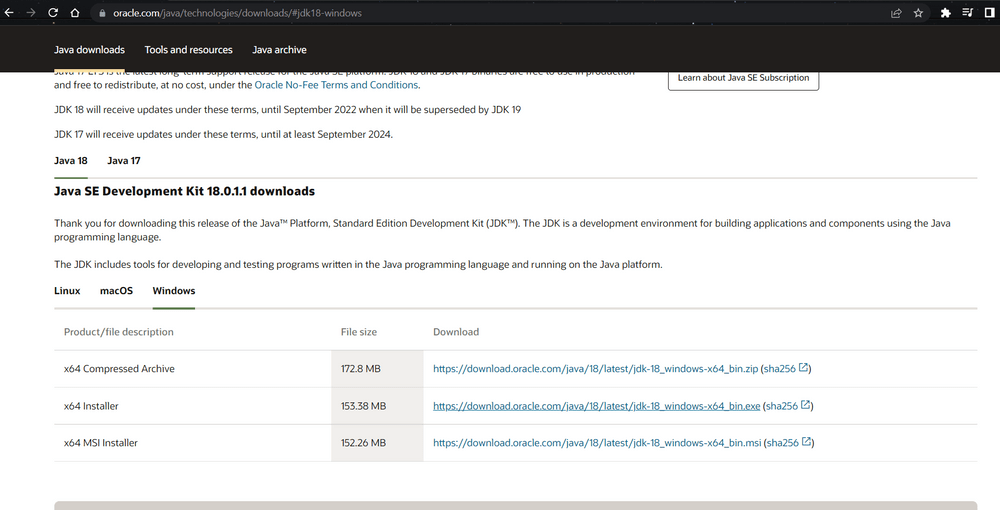
**2.1. Installation and setup of JDK**

**2.1.1. Installation and setup of JDK**

**Step 1:** Download and Install Java Development Kit (JDK)

The very first step is to download the Oracle Java Development Kit (JDK) from the Official

Oracle Website. For that, Head over to the Official Website.



After the download is complete, proceed to install the JDK by following the bootstrapped steps.

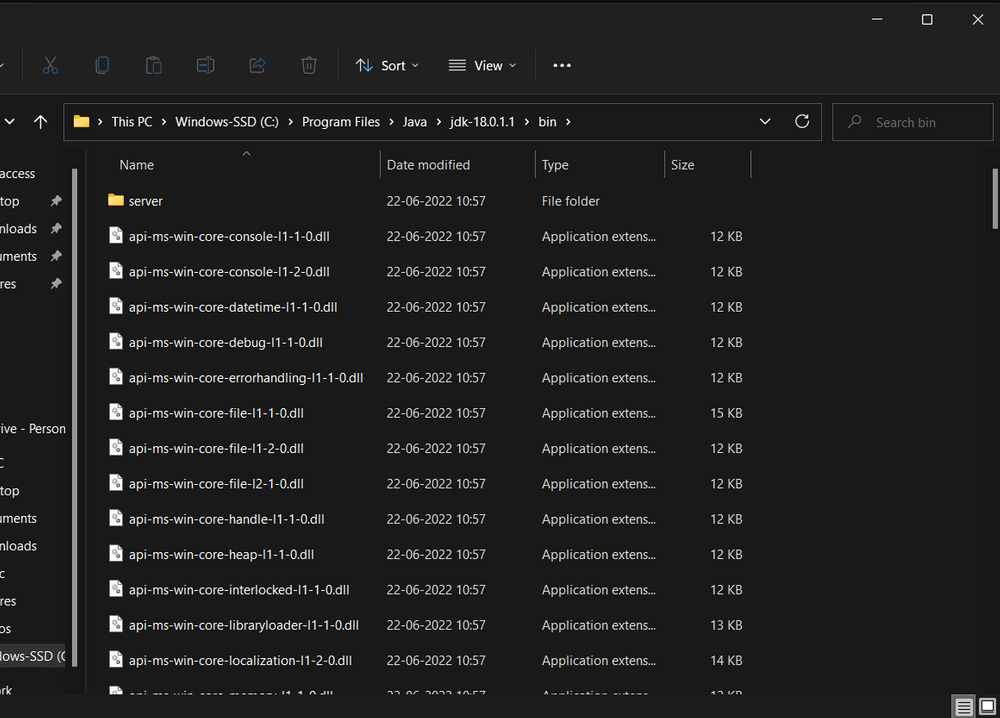


**Step 2:** Configure Environment Variables

After the installation is complete, we have to configure environment variables to notify the

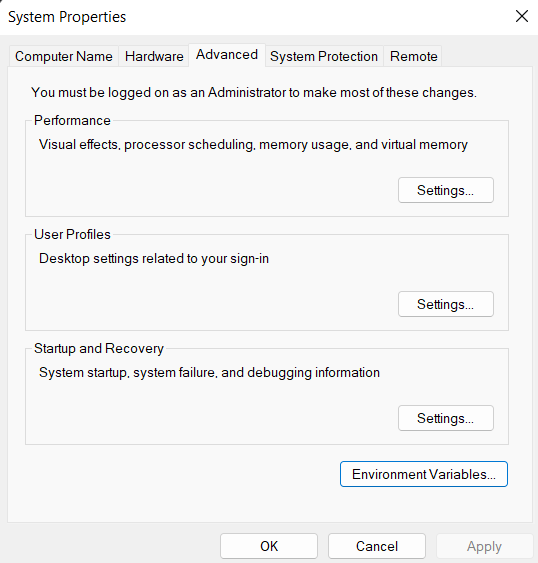
system about the directory in which JDK files are located. Proceed to C:\Program

Files\Java\jdk- {Your JDK version}\bin

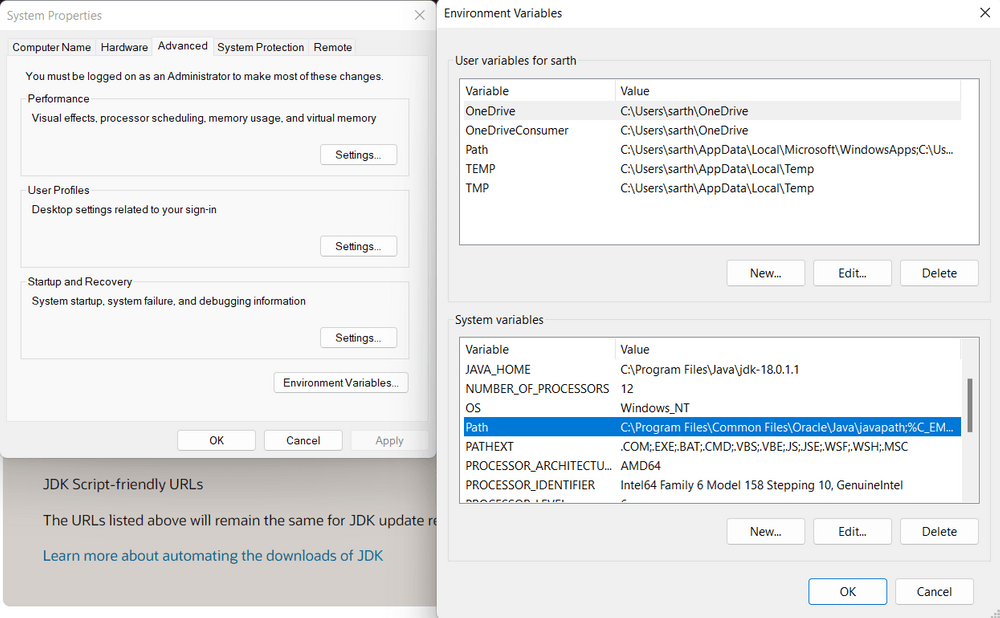


To set the Environment Variables, you need to search Environment Variables in the Task Bar

and click on "Edit the system environment variables".



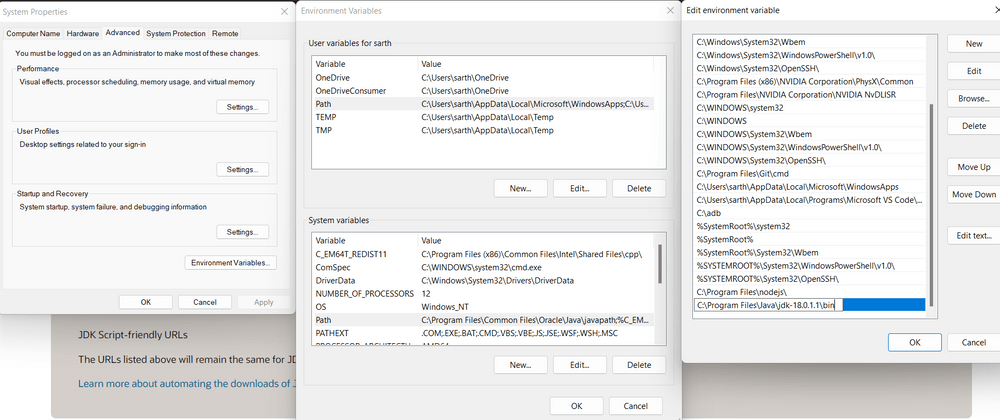
Under the Advanced section, Click on "Environment Variables".



Under System variables, select the "Path" variable and click on "Edit". Click on "New" then

paste the Path Address i.e. C:\Program Files\Java\jdk-{YOUR\_JDK\_VERSlON}\bin. Click on

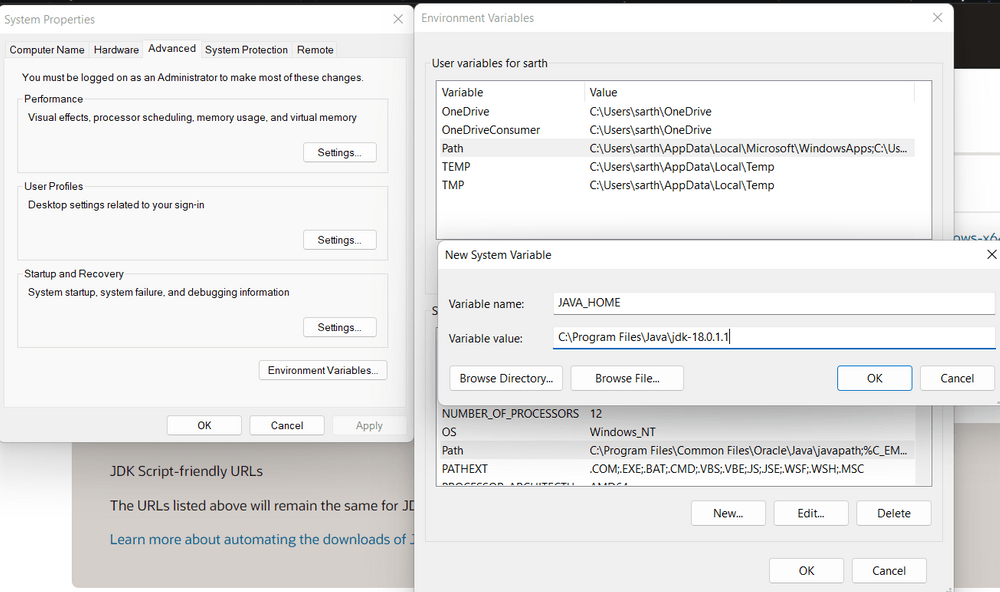
"0K".



Now, in the Environment Variables dialogue, under System variables, click on "New" and then

under Variable name: JAVA\_HOME and Variable value: paste address i.e.

C:\ProgramFiles\Java\jdk-{YOUR\_JDK\_VERSION}.Click on 0K=>0K=>0K.

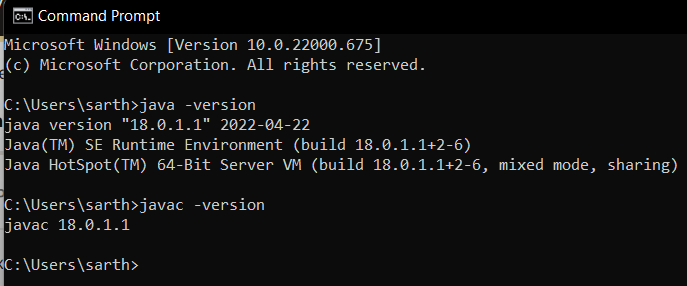


**Step 3:** Check the Java Version

Open Command Prompt and enter the following commands

java -version

javac -version



**2.2. Downloading and Installing Android Studio**

**Step 1** **:** Go to the <https://developer.android.com/studio> to download android studio

**Step 2 :** Go to the folder where the downloaded file is present

**Step 3 :** Double click on .exe file to install Android Studio

**Step 4 :** The installation dialog will appeared as shown below



**Step 5:** Click on Next => Next => Next => Install

**Step 6:** After installation is completed ,it will automatically download the required components

**CH-3. RESULTS**

**3.1. Table of Project’s Screenshots**

3.1.1. Splash Screen

3.1.2. Main Screen

3.1.3. Add Dialog

3.1.4. Main Screen after adding Subject

3.1.5. Change in percentage after changing Present and Absent Count

3.1.6. Options Dialog

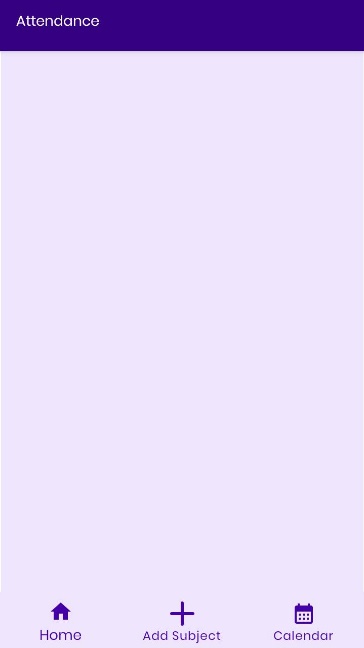
3.1.7. Delete Dialog

3.1.8. Subject data after Reset

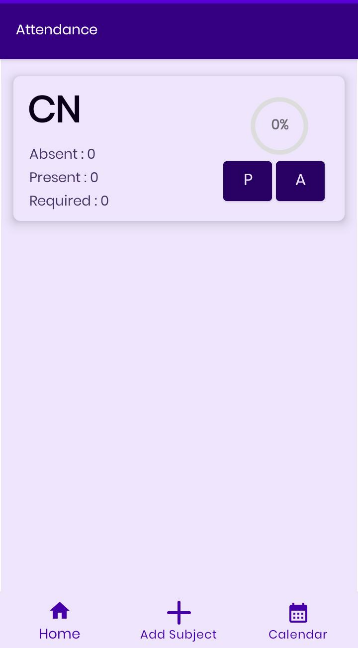
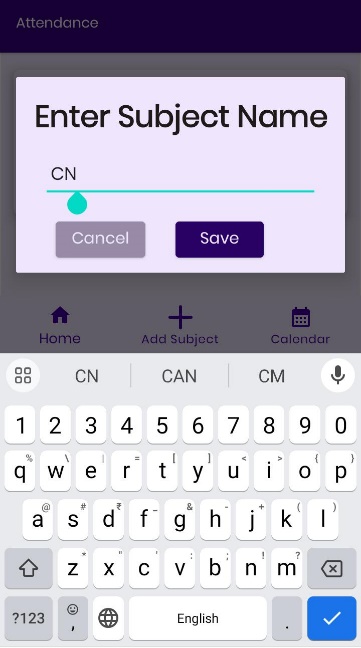
3.1.9. Edit Dialog

3.1.10. History Page

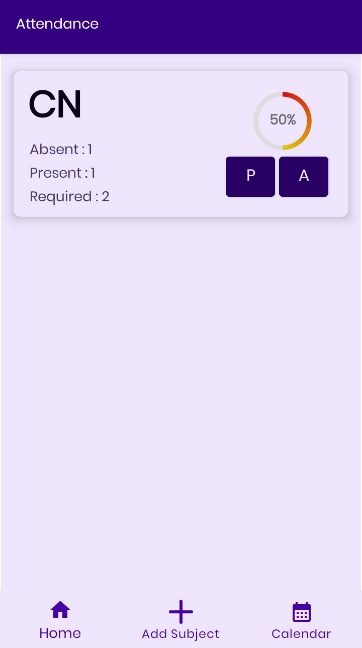
**PROJECT’S SCREENSHOTS**



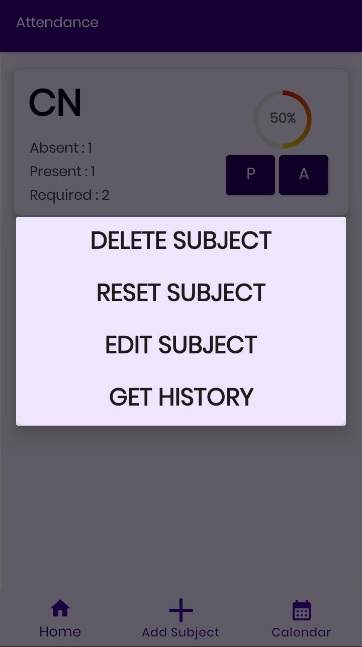
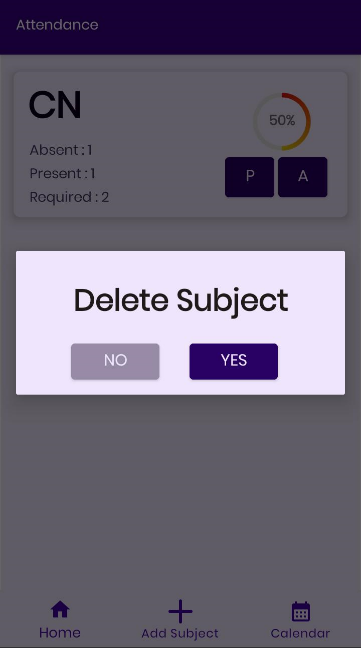
*3.1.1. Splash Screen 3.1.2. Main Screen*

**

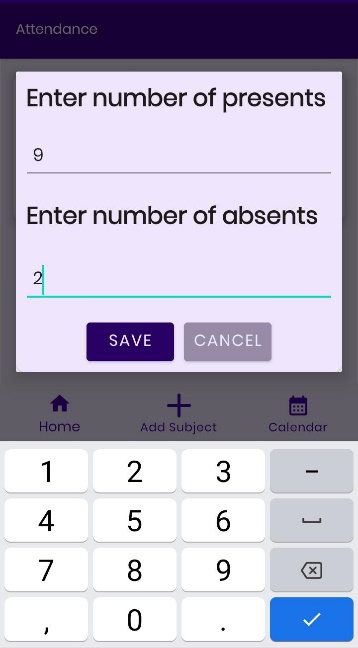
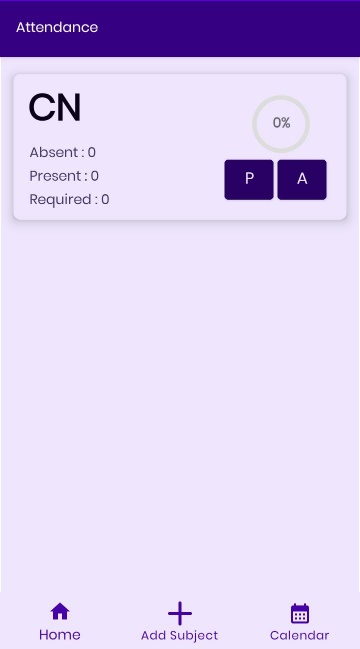
*3.1.3. Add Dialog 3.1.4. Main Screen after adding Subject*

**

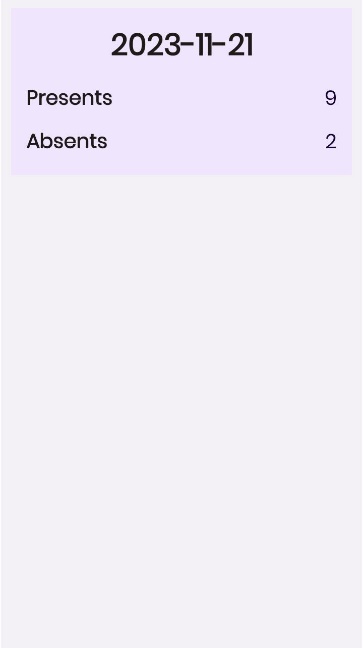
*3.1.5. Change in percentage after changing Present and Absent Count*



*3.1.6. Options Dialog 3.1.7.Delete Dialog*

**

*3.1.8. Subject data after Reset 3.1.9. Edit Dialog*

**

*3.1.10. History Page*

**3.2. Features of the Project**

* Simple and User-Friendly Interface
* Compatibility with wide range of Android Devices

(Min. Android Version : Android Oreo(V8.1.0))

* Visualized Presentation of Attendance percentage of each subject
* Automatic Calculation for no. of lectures need to be required to achieve 75% attendance
* Integrated shortcut to open Calendar

**CH-4. CONCLUSION AND FUTURE SCOPE**

**4.1. Conclusion**

The above report highlights about the technologies learnt during the development of the project. The task of developing the report also helps in skill development of the students. After working on the project, I understand how the things actually work in the real-life projects such as how the UI elements respond, storage of data in the database etc. The Minor Project Development task also helps the students in exploring the new opportunities and technologies and choose a career field according to their interests.

**4.2 Future Scope**

Android development has a bright future, with several emerging technologies and trends that are poised to shape the landscape in the coming years. Here are some of the key areas to watch:

**1. Artificial Intelligence (AI) and Machine Learning (ML)**

AI-powered features like personalized recommendations, predictive analytics, and intelligent assistants are becoming increasingly common, and developers are finding new and innovative ways to integrate these technologies into their apps.

**2. Internet of Things (IoT)**

The IoT is rapidly expanding, and Android devices are becoming increasingly important hubs for connecting and interacting with IoT devices. Developers are creating apps that allow users to control and manage their smart homes, wearable devices, and other IoT devices, making their lives easier and more connected.

**3. Augmented Reality (AR) and Virtual Reality (VR)**

AR apps are overlaying digital information onto the real world, while VR apps are creating immersive experiences that transport users to new worlds.

**5. Wearables and Foldable Devices**

Wearables like smartwatches and fitness trackers are providing new opportunities for app development, while foldable devices are introducing new challenges and opportunities for app design and development.

And Much More..