

Notes & Password Manager

Overview

Objective

You will be creating an android application which helps to take notes at any time and helps to generate a very strong password for the security of your account and makes them available whenever you need. It's going to have two major functionalities at the same time.

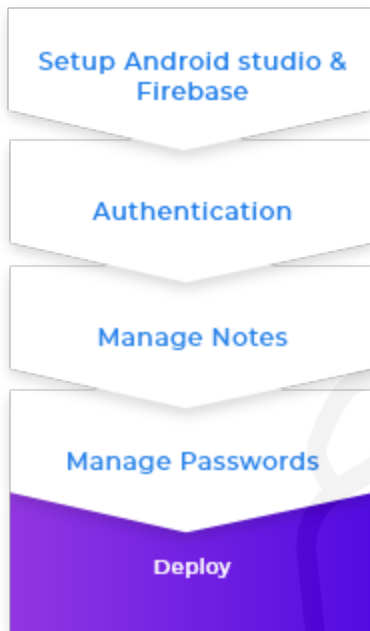
Project Context

Password managing & note-taking are one of the most overlooked virtual activities in today's digital age. Various companies are constantly adding new features to note-taking and password management. Have you ever considered building your own note-taking and password managing application?

So, instead of having and maintaining two different apps in your system, we'll be developing a single app which can perform both of these activities.

Project Stages

The project consists of the following stages:



High-Level Approach

- The first task, once we get the development environment ready, will be to set up **Android Studio & Firebase**.
- Once we have everything in place, we can start off with creating the application, which will basically start with the login authentication.
- Next up is the home page building. In this project, we'll be keeping it simple by showing two options i.e Notes & Password.
- In our project, we'll need to manage three states: **Note** (to manage notes data), **Password** (to manage password data), and **user** (for managing the details of the currently logged-in user).
- For showing all the notes or passwords, we'll be using **Listviews or recycleviews**.
- Handling our database and authentication needs to be supported and we'll be using **Firebase** for the same. Basically, the database will be used to store the login information for the users, but the resource can be used for storing notes and password information as well.
- Successful implementation of the above requirements will lead to the completion of the core implementation of our application. Next up, deploy!

Primary goals

- Create a login page for user login.
- Create a home page to display notes & password options.

- Create a note page to display all notes.
- Create a password page to display all passwords.
- Add functionalities like add note or password, delete note or password, and update note (if required).

Task 1

Environment & Firebase setup

Before the start of any development procedure, we need to set up the environment according to our application needs. Then connect your Android Studio with Firebase.

Requirements

- Install Android Studio on your machine.
- Install and set up JDK.
- Create a new Android project.
- Like any typical application, the source code of Java should be in a java folder and the source code of XML should be in a res folder.
- Connect with Firebase.

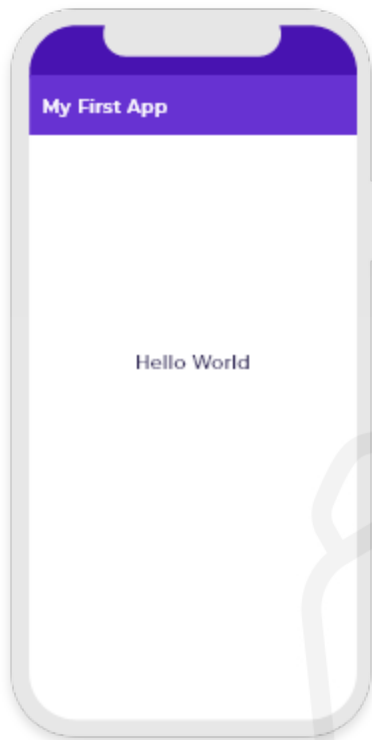
References

- [Download Android Studio](#)
- [Download JDK](#)
- [Create a new Android project](#)
- [Creating a virtual Android device on your machine](#)
- [A beginners guide to Gradle](#)
- [Run your Android application](#)
- [Connect your Android App to Firebase](#)

Expected Outcome

The main objective of this milestone is to make sure that you have the required development environment in place.

On completion of the above requirements, run the application using the virtual device and the end result should be as shown in the picture below.



Task 2

Setting up Firebase and Authentication

Firebase is a great service provided by Google for configuring the backend of any application with all the general necessities like database preparation, authentication using various methods, etc. In this milestone, we'll be preparing our database and setting up authentication using email and password.

[**Note:** Use the references provided to implement the following requirements.]

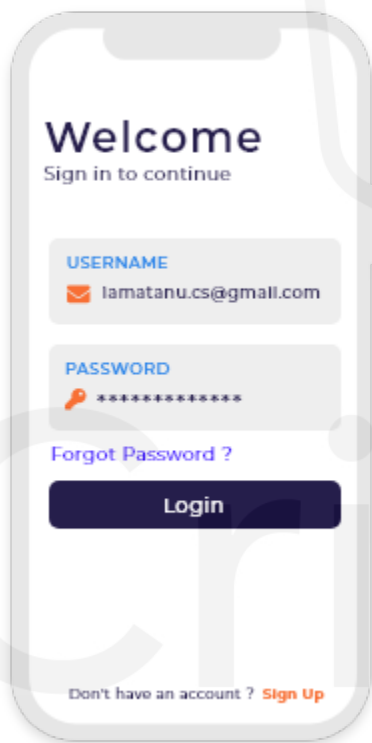
Requirements

- Setup sign-in method using Email/Password.
- Declare the dependency for the Firebase Authentication Android library in your module (app-level) Gradle file (usually app/build.gradle).

```
dependencies {  
    // Import from the Firebase platform  
    implementation  
    platform('com.google.firebase:firebase-bom:26.3.0')
```

```
// Declare the dependency for the Firebase Authentication library
// When using the BoM, you don't specify versions in Firebase
library dependencies
    implementation 'com.google.firebase:firebase-auth'
}
```

- To use an authentication provider, you need to enable it in the Firebase console. Go to the Sign-in Method page in the Firebase Authentication section to enable Email/Password sign-in and any other identity providers you want for your app.
- Android has Material Components, which helps in building our frontend.
- Create a new activity named Login. Style the activity so that it looks similar to the one shown below.



[**Note:** You are free to work on your own design. The above image can be used as a reference.]

References

- [Material Components for Android](#)

- [Get Started with Firebase Authentication on Android](#)

Expected Outcome

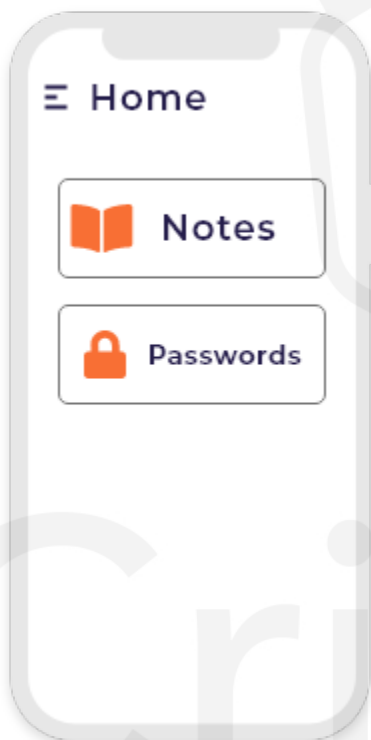
You should be able to develop the Login feature of your application and also integrate Firebase authentication for the same.

Task 3

Creating home activity

Now we will create an activity where there will be two options **note** and **password**.

- Style the activity so that it looks similar to the one shown below.



[**Note:** The above image might give you some inspiration to work on your own design.]

Creating note activity

Now we will create one more activity through which all the notes will be managed and besides add, delete and update functionality should work.

Requirements

- We need a couple of icons like the navigation icon, note icon, and password icon.

- Create a new component called App-Bar in note activity.
- Add the necessary code in the xml and java files so that all the work is done accurately as mentioned earlier.

References

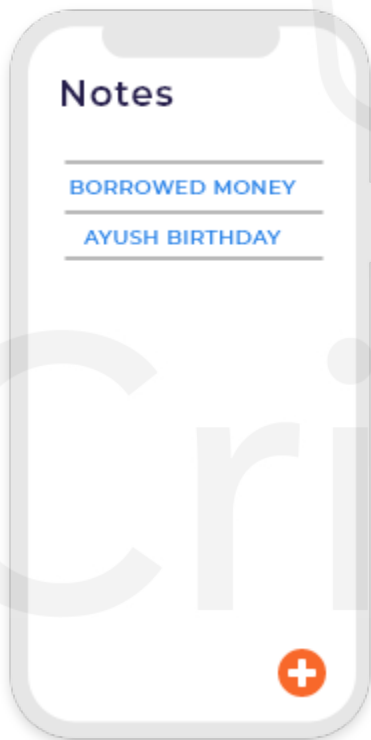
- [Material Components for Android](#)
- [Read and Write Data into Firebase Database](#)

Tip

You can use Listview or Recycleview to show notes as mentioned earlier. Using Listview or Recycleview, you can only show the title of the notes and after clicking on those titles, the rest of the functionality can be done.

Expected Outcome

On completion of this milestone, the note manager of your application should look similar to the one shown below.



Task 4

Creating password activity

Now we will create one more activity through which all the passwords will be managed and besides add and delete functionality should work feasibly.

Requirements

- We need a couple of icons like the password icon and the add icon.
- Create a new component called App-Bar in password activity.
- Add the necessary code in the xml and java files so that all the work is done accurately as mentioned earlier.
- Create a new function for generating password

```
public static String generateRandomPassword(int len) {  
    String chars = "0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz!"  
        + "jklmnopqrstuvwxyz!@#$%&";  
    Random rnd = new Random();  
    StringBuilder sb = new StringBuilder(len);  
    for (int i = 0; i < len; i++)  
        sb.append(chars.charAt(rnd.nextInt(chars.length())));  
    return sb.toString();  
}
```

- You should create a new function that will measure the strength of any password.

References

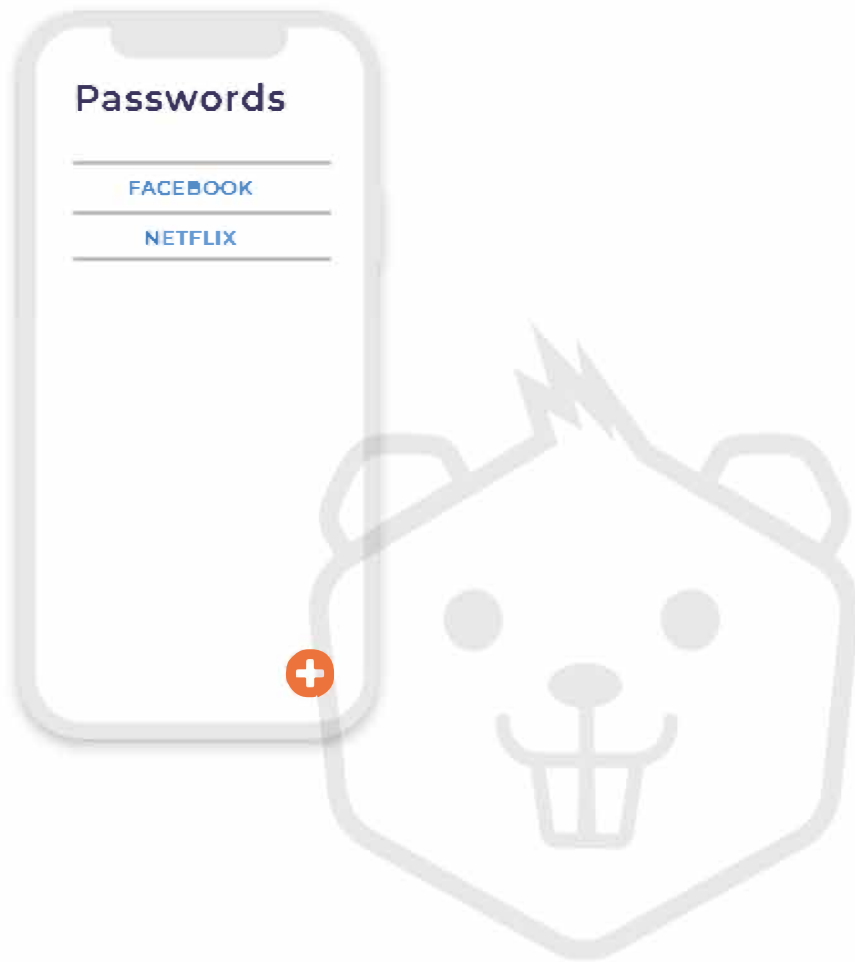
- [Material Components for Android](#)
- [Read and Write Data into Firebase Database](#)

Tip

You can use Listview or Recycleview to show passwords like notes as mentioned earlier. Using Listview or Recycleview you can only show the field name of the passwords and after clicking on those field names, the rest of the functionality can be done.

Expected Outcome

On completion of this milestone, the password manager of your application should look similar to the one shown below.



Crio.Do