***ACKNOWLEDGEMENT***

***“ Gratitude makes sense of our past, brings peace for today and creates a vision for tomorrow”.***

***So, we expressed our gratitude to all those people without whose support, encouragement, guidance and co-operation this project would not have been completed***

***We would like to take privilege to thank all those who have guided and inspired us in completion of the project.***

***Firstly, we would like to express my heartfelt thanks to our respected Principal prof. Manjunath Muttalgerisir for his keen interest and valuable inputs throughout the course of this work.***

***Secondly, we are thankful to our respected Teacher Prof. Manjunath Muttalgeri for giving us opportunity to carry out the project.***

***Last but not least I sincerely thank my beloved parents and family who were very cooperative and supported us morally during the course***

**DECLARATION**

We,Ms vaishnavi sinnur(20U10442) and Ms Srushti Goudar (20U10440)

student of BCA 5th

semester Oxford College of Computer Applications,

hereby declare

that dissertation

**OXFORD COLLEGE”**

Submitted by us to the Department of Computer Applications, Karnataka University Dharwad in partial fulfilment of BCA program is a Bonafide one. This is not submitted earlier to any other University or Institution for the award of any Degree certificate or published any time.

Ms Vaishnavi sinnur(20U10442

Ms Srushti Goudar (20U10440)

*.*

Introduction

This project is based on the Android in which we can store and retrieve all the information regarding the College Management system details.

Database is a structured format. So if we store in the database, we can retrieve that particular information by giving a command directly. There is no process for installing database for mobile-phone applications. Whenever the database is needed, then only the database is created through coding. In this project there is a feasibility to change and delete the data which is not required.

The application we are developing is about “**Oxford College”**. It is an android application used to find out the Information about websites, notes, Syllabus, Question papers, and also notification from college.

## Existingsystem v/s proposedsystem.

##### **Existingsystem:**

### The main problem projected is that pupil’s particulars are reported manually in distinct records, which is a laborious job. Handling and updating these records manually increase the chances of mistakes. It takes a lot of time and needs many employees to accomplish the task. It even lacks security and disability to produce various types of reports.

### They imply a high entry barrier for educators and students who have to learn the new tools. This is a well-known problem related to the usage of IT solutions as education-supporting elements.

##### Proposedsystem:

The benefits of college management system for the employee is they can create any kind of certificate easily using this system. They can easily retrieve all information related to student and faculty. This system gives easy approach to find the detail information for any student/faculty. Using this college management system it is very easy to handle all functionality of college. This system is beneficial for both students and faculty as they can get all previous or current information when they need.

## Objective and purpose.

##### **Objective:**

This is a web oriented application allows us to access the whole information about thecollege, staffs, students, facilities etc. This application provides a virtual tour of Campus.Here we will get the latest information about the students .The main objective of this application is to provide a college information like website,erp login, notes, syllabus, notification, old question papers. This application gives a very flexible environment to the mobile user because now a days students are more addicted to a mobile so they can get information about a college easily.

**Puropose**

The college management system helps Educational Institutions especially colleges in various ways, such as storing data, maintain student profiles, analyzing administrative and

maintaining the Questions papers Syllabus and notes etc.

## Literature survey

Mobile application can be one of the best ways to keep consumers engaged with as they are on the move. With the increase in demand for smartphones and efficiency of wireless network, the demand for mobile application has increased incredibly.

## Hardware and Software Requirements

**Hardware Requirement:**

➢ **Computer:** A computer with a 64-bit operating system, either Windows, MacOS, or Linux. A laptop or desktop computer with at least 4 GB of RAM is recommended.

➢ **Processor:** A dual-core processor with at least 2 GHz clock speed.

➢ **Hard Disk Space**: At least 4 GB of free hard disk space is required to install Android Studio and the Android SDK.

➢ **Graphics Card:** A graphics card that supports OpenGL 2.0 or higher.

➢ **Monitor:** A monitor with a screen resolution of at least 1280x800 pixels.

➢  **Internet Connection**: A reliable and fast internet connection is required to download updates and new components from the Android SDK Manager

**Software Requirements:**

SOFTWARE REQUIREMENTS:

➢ **Android Studio**: As the primary development platform, Android Studio will be used to design and develop the mobile application.

➢ **Java**: Java will be the programming language used to write the code for the mobile application in Android Studio.

➢ **Android SDK (Software Development Kit):**

The Android SDK will be necessary to build and run the mobile application on an Android device or emulator.

➢ Android emulator or physical Android device:

**Android System Architecture**

In android operating system, there are four layers. Android [1] has its own libraries; it is helpful for developing and designing any application of android platform.These libraries are written in C/C++. Linux kernel is the 1st layer which is written in C. Linux also helps to wrap the application. The unveiling of the Android platform on 5 November 2007 was announced with the founding of the Open Handset Alliance, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices. When released in 2008, most of the Android platform will be made available under the Apache free-software and open source license. Currently Android represents 31.2 percent of the U.S Smartphone market. Android has a large community of developers writing application programs. There are currently over 150,000 apps available for Android. Android

**Features of Android**

Application Framework enabling reuse and replacement of components.

* Dalvik Virtual Machine optimized for mobile devices.
* Integrated browser based on the open source Web Kit engine.
* SQLite for structured data storage.
* Media support for common audio, video and still image formats (MPEG4, H.264, MP3, AAC, AMR,JPG,PNG,GIF).
* GSM Telephony (hardware dependent).
* Bluetooth,EDGE,3G, and Wi-Fi(hardware dependent)

**Structure:**

Android is a widely anticipated open source operating system for mobile devices that provides a base operating system, an application middleware layer, a Java software development kit (SDK), and collection of system applications. Android mobile application development is based on Java language codes, as it allows developers to write codes in the Java language.



**Android Application:**

Android will ship with a set of core applications including an email client, SMS program,

calendar, maps, browser, contacts, and others. All applications are written using the Java

programming language.

**Application Framework:**

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

• A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser.

• Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data.

• A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files.

• A Notification Manager that enables all applications to display custom alerts in the status bar.

• An Activity Manager that manages the lifecycle of applications and provides a common navigation back stack.

**Libraries:**

Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:

System C library - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices

Media Libraries - based on Packet Video's Open CORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.

Surface Manager - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications.

LibWebCore - A modern web browser engine which powers both the Android browser and an embeddable web view.

SGL - the underlying 2D graphics engine.

3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware

3D acceleration (where available) or the included, highly optimized 3D software rasterizer.

Free Type - bitmap and vector font rendering

**Android Runtime**:

Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language. Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool. The Dalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

**Linux Kernel:**

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

The android SDK consists of mainly the following:

• Activity

• Service

• Broadcast receiver

• Content provider

**Activity:**

An activity is a user interface concept. An activity usually represents a single screen in

your application. It generally contains one or more views, but it doesn’t have to. An

Activity is pretty much like it sounds something that helps the user do one thing and

that one thing could be viewing data, creating data, or editing data. Most Android applications

have several activities within them.

**Service:**

Services in Android resemble services you see in Windows or other platforms they’rebackground processes that can potentially run for a long time. Android defines two types of services: local services and remote services. Local services are components that are only accessible by the application that is hosting the service. Conversely, remote services are services that are meant to be accessed remotely by otherapplications running on the device.

**Broadcast receiver:**

Android again drops a message on the main queue of the package process in which the registered. Receiver is to be invoked from. The main thread will come around to that message at a later time to invoke the receiver. The main thread does the work for abroadcast receiver as well. If the main thread is busy responding to a menu action, thebroadcast receiver will have to wait until the main thread gets freed up.

Content provider:

Android uses a concept called content providers for abstracting data into services. ThisIdea of content providers makes data sources look like REST-enabled data providers, Such as web sites. In that sense, a content provider is a wrapper around data. A SQLite Database on an Android device is an example of a data source that you can encapsulate Into a content provider.

**XML**

• XML stands for eXtensible Markup Language.

• XML was designed to store and transport data.

• XML was designed to be both human- and machine-readable.

• XML plays an important role in many different IT systems.

• XML is often used for distributing data over the Internet.

• It is important (for all types of software developers!) to have a good understandin

**JSON**

JSON: JavaScript Object Notation.

JSON is a syntax for storing and exchanging data.

JSON is text, written with JavaScript object notation.

When exchanging data between a browser and a server, the data can only be text

**Gradle**

Gradle is a build tool for Android Studio that provides a flexible and powerful way to manage the build process for Android applications. Some key features of Gradle in the context of Android Studio include:

1. **Dependency management:**

Gradle makes it easy to manage dependencies, including downloading and integrating libraries and other components.

1. **Build variants:**

Gradle allows you to define multiple build variants, such as different builds for different flavors of your app or different builds for different build types (e.g., debug, release).

1. **Incremental builds:**

Gradle uses an incremental build system, meaning that it only builds the parts of your project that have changed, making builds faster and more efficient.

1. **Custom build logic:**

Gradle provides a flexible and powerful scripting language for defining custom build logic, allowing you to tailor the build process to your specific needs

**Software Requirement Specification(SRS) Documnet**

**1. Introduction**

1.1 Purpose The purpose of this document is to specify the requirements for an online vehicle parking system that allows users to find, book, and manage parking slots and communicate with an admin.

1.2 Scope The scope of this project includes the following features: user login and registration, finding and getting details of question papers, notes and syllabus and notification, user settings. The admin will have the ability to add, update, and delete the database which he doesn’t need

1.3 Stakeholders The stakeholders for this project include users, administrators, and the development team.

**2. Functional Requirements**

2.1 User Login and Registration Users must be able to create an account and log in to the system. They must also be able to reset their password if they forget it.

2.2 Finding Scope The scope of this project includes the following features: user login and registration, finding and getting details of question papers, notes and syllabus and notification, user settings. The admin will have the ability to add, update, and delete the database which he doesn’t nee

2.3 Managing Favorites Users must be able to add parking slots to a list of favorites for easy future access. They must also be able to remove parking slots from the favorites list.

2.4 User Settings Users must be able to update their personal information, such as their name, email, and password.

2.7 Admin: Add database must be able to add new database to the system.

2.8 Admin Update the database must be able to modify the information for existing database.

2.9 Admin Delete database must be able to remove database from the system.

**3. Non-Functional Requirement**

3.1 Performance The system must respond to user requests in a timely manner, with response times of less than 5 seconds.

3.2 Reliability The system must be highly reliable, with a maximum downtime of 4 hours per year.

3.3 Usability The system must be easy to use, with a clear and straightforward user interface.

3.4 Security The system must be secure, with proper authentication and authorization procedures in place to protect sensitive data

**4. User Interfaces**

4.1 Login and Registration The login and registration pages will have fields for the user to enter their email and password.

4.2 Finding the scope of this project includes the following features: user login and registration, finding and getting details of question papers, notes and syllabus and notification, user settings. The admin will have the ability to add, update, and delete the database which he doesn’t nee

4.3 Managing Favorites The favorites page will display a list of the user's favorite parking slots. The user can add or remove parking slots from the list.

4.4 User Settings The user settings page will allow the user to update their personal Information

**Non-Functional Requirements**

**Reliability**

The factors needed to establish the software expected reliability are

• The user inputs should be valid and within the given range.

• Normal Termination of program

**Security**

• One user cannot access the other user’s account.

• Only the authenticated user is allowed to log in.

**Maintainability**

• The system shall ensure that the data is protected from unauthorized access.

• The application uses minimum number of interactions with the server which results in

overall high performance of the application.

• The Application is operational 24 hours a day, 7 days a week.

• Application code is easy to debug and extend for new features if needed

**System Design**

• The conceptual model that is used to define the structural behavior and the detailed view of the system is nothing but the system architecture and the basic task of the system architecture is to give the overall idea of the project and the system architecture does the same thing. In the below diagram we have mentioned the detailed view of what is admin.

• The Data Flow Diagram (DFD) is a vital demonstrating instrument. It is a Graphical apparatus utilized for show framework necessities in a graphical structure. It is otherwise called "bubble outline". It's motivation is explaining framework prerequisites and distinguishing significant change that will become programs in framework plan. DFD addresses the progression of the information among measure and various changes in the framework.

• It addresses the useful conditions inside a framework. It addresses yield esteems in a calculation are gotten from input esteems. The information stream diagram shows intelligent progression of the information. It gives data about the intelligent construction of the framework. It matches information data to different cycles of the framework. Study of project through Classical Model

• A system model is a systematic approach towards software development. Before any building is built it is necessary to make the drawing of that building, similar is the case with any software. Here also before any coding of the software begins, it is necessary to make the model of the software.

• A model basically specifies all the steps to be followed during the system development. This makes the task of the engineer quite simple and helps him build software, which is less error prone.

• Tasks accomplished by system modeling are:

• Define the processes that serve the needs of the view under consideration.

• Represent the behavior of the processes and the assumptions on which behavior is based.

• Explicitly define both exogenous and endogenous input to the model.

• Represent all linkages that will enable the engineer to better understand the viewA crucial phase in the System Development Life Cycle (SDLC)

• Is the successful implementation of the new system design? Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be completely new, replacing an existing manual or automated system or it may be major modification to an existing system. In either case, proper implementation becomes necessary so that a reliable system based on the requirements of the organization can be provided.

• Successful Implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it. It has been observed that even the best system cannot show good results if the analyst managing the implementation does not attend to every important detail.

• This is the area where the system analyst needs to work most carefully

• The proposed system is very simple to implement. As specified prior, the required Hardware and Software are sufficient for implementing this software. The implementation process is quite similar to most software’s. So the implementations of this software become easy.

**Data Flow Diagram (Level 0)**

Android

Application

**End user**

**(Student)**

API

**Log user**

**Modules**

**Call for forget password and Queries**

**Get Info**

**Select Module**

**Login**

**Buy Android**

**Application**

**Register**

**Install it on**

**Device**

**Retrieve Information**

**Run**

**Run in Background**

• Data flow-oriented techniques advocate that the major data items handled by a system must be first identified and then the processing required on these data items to produce the desired outputs should be determined. The DFD (also called as bubble chart) is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on these data, and the output generated by the system. It was introduced by De Macro (1978), Gane and Sarson (1979).

**Use Case Diagram:**

**LOGIN**

**Notification**

**Syllabus**

**Question papers**

**Notes**

**ERP login**

**Website**

**Manifest Code**

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools">

<uses-permission android:name="android.permission.INTERNET"/>

<application

android:allowBackup="true"

android:dataExtractionRules="@xml/data\_extraction\_rules"

android:fullBackupContent="@xml/backup\_rules"

android:icon="@mipmap/ic\_launcher"

android:label="OXFORD COLLEGE"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.OxfordCollege"

tools:targetApi="31">

<activity

android:name=".loginActivity"

android:exported="true">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

</application>

**Manifest Debug:**

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

<manifest xmlns:android="http://schemas.android.com/apk/res/android">

<application>

<activity

android:name=".loginActivity"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".registerActivity"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".forgotPassword"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".MainActivity"

android:exported="false">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".erpLogin"

android:exported="false">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".notes"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".BCA1stSem"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".BCA2ndSem"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".BCA3rdSem"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".BCA4thSem"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".BCA5thSem"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".BCA6thSem"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".question\_papers"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".syllabus"

android:exported="true">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

<activity

android:name=".notification"

android:exported="false">

<meta-data

android:name="android.app.lib\_name"

android:value="" />

</activity>

</application>

</manifest>

**Login Activity**

package com.vaishu.oxfordcollege;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.text.method.PasswordTransformationMethod;

import android.util.Log;

import android.view.MotionEvent;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.android.material.textfield.TextInputEditText;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class loginActivity extends AppCompatActivity {

EditTextLoginEmail;

EditTextLoginPassword;

booleanpasswordVisible;

Button btnLogin;

TextViewRegisterHere,ForgotPassword;

FirebaseAuthfirebaseAuth;

@SuppressLint("MissingInflatedId")

@Override

protected void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_login);

LoginEmail=findViewById(R.id.Email);

LoginPassword=findViewById(R.id.password);

RegisterHere=findViewById(R.id.RegisterHere);

ForgotPassword=findViewById(R.id.forgotPass);

btnLogin=findViewById(R.id.LoginBtn);

firebaseAuth=FirebaseAuth.getInstance();

btnLogin.setOnClickListener(view ->{

loginUser();

});

RegisterHere.setOnClickListener(view ->{

startActivity(new Intent(loginActivity.this,registerActivity.class));

});

ForgotPassword.setOnClickListener(view -> {

startActivity(new Intent(loginActivity.this, forgotPassword.class));

});

}

private void loginUser(){

String email = LoginEmail.getText().toString();

String password = LoginPassword.getText().toString();

if(TextUtils.isEmpty(email)){

LoginEmail.setError("Email cannot be empty");

LoginEmail.requestFocus();

} else if (TextUtils.isEmpty(password)) {

LoginPassword.setError("password cannot be empty");

LoginPassword.requestFocus();

}else {

firebaseAuth.signInWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if(task.isSuccessful()){

Toast.makeText(loginActivity.this, "user logged in successful", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(loginActivity.this,MainActivity.class));

}else{

Toast.makeText(loginActivity.this, "Login Error" + task.getException().getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

}

}

}

**Register Activity**

package com.vaishu.oxfordcollege;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.internal.InternalTokenProvider;

public class registerActivity extends AppCompatActivity {

EditTextRegEmail;

EditTextRegPassword; Button btnRegister;

TextViewLoginHere;

FirebaseAuthfirebaseAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_register);

RegEmail = findViewById(R.id.RegEmail);

RegPassword = findViewById(R.id.password);

btnRegister = findViewById(R.id.RegisterBtn);

LoginHere =findViewById(R.id.LoginHere);

firebaseAuth = FirebaseAuth.getInstance();

btnRegister.setOnClickListener(view -> {

createUser();

});

LoginHere.setOnClickListener(view ->{

startActivity(new Intent(registerActivity.this,loginActivity.class));

});

} private void createUser(){

String email = RegEmail.getText().toString();

String password = RegPassword.getText().toString();

if(TextUtils.isEmpty(email)){

RegEmail.setError("Email cannot be empty");

RegPassword.requestFocus();

} else if (TextUtils.isEmpty(password)) {

RegEmail.setError("password cannot be empty");

RegPassword.requestFocus();

}else {

firebaseAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if(task.isSuccessful()){

Toast.makeText(registerActivity.this, "user registered successful", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(registerActivity.this,loginActivity.class));

}else{

Toast.makeText(registerActivity.this, "Registration error" + task.getException().getMessage() , Toast.LENGTH\_SHORT).show();

}

}

});

}

}

}

**Main Activity:**

package com.vaishu.oxfordcollege;

import android.content.Intent;

import android.net.Uri;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View;

import androidx.appcompat.app.AppCompatActivity;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.google.firebase.ktx.Firebase;

import androidx.cardview.widget.CardView;

public class MainActivity extends AppCompatActivity implements View.OnClickListener {

FirebaseAuthfirebaseAuth;

CardViewcardWebsite;

CardViewcardERPLogin;

CardViewcardNotes;

CardViewcardQuestion\_papers;

CardViewcardSyllabus;

CardViewcardNotification;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

firebaseAuth = FirebaseAuth.getInstance();

cardWebsite = findViewById(R.id.cardWebsite);

cardERPLogin = findViewById(R.id.cardERPLogin);

cardNotes = findViewById(R.id.cardNotes);

cardQuestion\_papers = findViewById(R.id.cardQuestion\_paper);

cardSyllabus = findViewById(R.id.cardSyllabus);

cardNotification = findViewById(R.id.cardNotification);

//Adding click listener to the cards

cardWebsite.setOnClickListener(this);

cardERPLogin.setOnClickListener(this);

cardNotes.setOnClickListener(this);

cardQuestion\_papers.setOnClickListener(this);

cardSyllabus.setOnClickListener(this);

cardNotification.setOnClickListener(this);

} @Override

public void onClick(View v) {

Intent i;

switch (v.getId()) {

case R.id.cardWebsite:

gotoUrl("https://www.oxfordcollege.edu.in/");

break;

case R.id.cardERPLogin:

i = new Intent(this, erpLogin.class);

startActivity(i);

break;

case R.id.cardNotes:

i = new Intent(this, notes.class);

startActivity(i);

break;

case R.id.cardQuestion\_paper:

i = new Intent(this, question\_papers.class);

startActivity(i);

break;

case R.id.cardSyllabus:

i = new Intent(this, syllabus.class);

startActivity(i);

break;

case R.id.cardNotification:

i = new Intent(this, notification.class);

startActivity(i);

break;

default:

break;

}

}@Override

protected void onStart() {

super.onStart();

FirebaseUser user = firebaseAuth.getCurrentUser();

if (user == null) {

startActivity(new Intent(MainActivity.this, loginActivity.class));

}

} private void gotoUrl(String s) {

Uri uri = Uri.parse(s);

startActivity(new Intent(Intent.ACTION\_VIEW, uri));

}

}

**Forget Password:**

package com.vaishu.oxfordcollege;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.FirebaseAuth;

public class forgotPassword extends AppCompatActivity {

private Button ForgotBtn;

private EditTexttxtEmail;

private String email;

private FirebaseAuth auth;

@SuppressLint("MissingInflatedId")

@Override

protected void onCreate(Bundle savedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_forgot\_password); auth=FirebaseAuth.getInstance();

txtEmail=findViewById(R.id.forEmail);

ForgotBtn=findViewById(R.id.forgotBtn);

ForgotBtn.setOnClickListener(view -> {

validateData();

});

} private void validateData() {

email=txtEmail.getText().toString();

if(email.isEmpty()){

txtEmail.setError("required");

}else{

forgotPass();

}

}private void forgotPass() {

auth.sendPasswordResetEmail(email)

.addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if(task.isSuccessful()){

Toast.makeText(forgotPassword.this, "check your email", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(forgotPassword.this,loginActivity.class));

finish();

}else{

Toast.makeText(forgotPassword.this, "Error" +task.getException().getMessage(), Toast.LENGTH\_SHORT).show();

}

}

});

}

}

**XML CODE:**

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

<ScrollViewxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".question\_papers">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<com.google.android.material.button.MaterialButton

android:id="@+id/BCA1stSem"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="55dp"

android:layout\_marginTop="40dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="20dp"

android:background="@color/dark\_blue"

android:backgroundTint="@color/dark\_blue"

android:text="BCA 1st sem"

android:textStyle="bold"

android:textColor="@color/white"

android:textColorHint="@color/white"/>

<com.google.android.material.button.MaterialButton

android:id="@+id/BCA2ndSem"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="55dp"

android:layout\_marginTop="40dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="20dp"

android:background="@color/dark\_blue"

android:backgroundTint="@color/dark\_blue"

android:text="BCA 2nd sem"

android:textStyle="bold"

android:textColor="@color/white"

android:textColorHint="@color/white"/>

<com.google.android.material.button.MaterialButton

android:id="@+id/BCA3rdSem"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="55dp"

android:layout\_marginTop="40dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="20dp"

android:background="@color/dark\_blue"

android:backgroundTint="@color/dark\_blue"

android:text="BCA 3rd sem"

android:textStyle="bold"

android:textColor="@color/white"

android:textColorHint="@color/white"/>

<com.google.android.material.button.MaterialButton

android:id="@+id/BCA4thSem"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="55dp"

android:layout\_marginTop="40dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="20dp"

android:background="@color/dark\_blue"

android:backgroundTint="@color/dark\_blue"

android:text="BCA 4th sem"

android:textStyle="bold"

android:textColor="@color/white"

android:textColorHint="@color/white"/>

<com.google.android.material.button.MaterialButton

android:id="@+id/BCA5thSem"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="55dp"

android:layout\_marginTop="40dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="20dp"

android:background="@color/dark\_blue"

android:backgroundTint="@color/dark\_blue"

android:text="BCA 5th sem"

android:textStyle="bold"

android:textColor="@color/white"

android:textColorHint="@color/white"/>

<com.google.android.material.button.MaterialButton

android:id="@+id/BCA6thSem"

android:layout\_width="300dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="55dp"

android:layout\_marginTop="40dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="20dp"

android:background="@color/dark\_blue"

android:backgroundTint="@color/dark\_blue"

android:text="BCA 6th sem"

android:textStyle="bold"

android:textColor="@color/white"

android:textColorHint="@color/white"/>

</LinearLayout>

</ScrollView>

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

<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://s

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".erpLogin">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="266dp"

android:background="@drawable/clg"

android:layout\_marginBottom="25dp"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="\* User Name"

android:layout\_marginHorizontal="30dp"

android:textColor="@color/black"

android:textStyle="bold"/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="30dp"

android:layout\_marginBottom="30dp"/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="\* Password"

android:layout\_marginHorizontal="30dp"

android:textColor="@color/black"

android:textStyle="bold"/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="30dp"

android:inputType="textPassword"

android:layout\_marginBottom="30dp"/>

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginHorizontal="30dp"

android:layout\_marginBottom="25dp"

android:text="Login"

android:textStyle="bold"/>

</LinearLayout>chemas.android.com/tools"

Vaishnavi: <?xml version="1.0" encoding="utf-8"?>

<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/cut\_card\_background\_2"

tools:context=".forgotPassword">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/ForgotTxt"

android:text="Forgot Password"

android:textColor="@color/white"

android:textSize="25dp"

android:layout\_margin="80dp"

android:gravity="center"

android:textStyle="bold"/>

<EditText

android:id="@+id/forEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/ForgotTxt"

android:layout\_marginStart="3dp"

android:layout\_marginTop="3dp"

android:layout\_marginEnd="3dp"

android:layout\_marginBottom="3dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_perm\_identity\_24"

android:drawablePadding="20dp"

android:hint="Email"

android:padding="20dp"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:textStyle="bold" />

<com.google.android.material.button.MaterialButton

android:id="@+id/forgotBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/forEmail"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="90dp"

android:layout\_marginTop="51dp"

android:layout\_marginEnd="30dp"

android:layout\_marginBottom="70dp"

Vaishnavi: android:background="@drawable/roundbtn"

android:backgroundTint="@color/cardview\_dark\_background"

android:gravity="center"

android:text="Forgot Password"

android:textColor="@color/black"

android:textStyle="bold"

app:backgroundTint="@color/white" />

</RelativeLayout>

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

<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/cut\_card\_background\_2"

tools:context=".loginActivity">

TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/logIn"

android:text="Login"

android:textColor="@color/white"

android:textSize="35dp"

android:layout\_margin="80dp"

android:gravity="center"

android:textStyle="bold"/>

<EditText

android:id="@+id/Email"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/logIn"

android:layout\_marginStart="3dp"

android:layout\_marginTop="3dp"

android:layout\_marginEnd="3dp"

android:layout\_marginBottom="3dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_perm\_identity\_24"

android:drawablePadding="20dp"

android:hint="Email"

android:padding="20dp"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:textStyle="bold" />

<EditText

android:id="@+id/password"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" android:layout\_below="@+id/Email"

android:layout\_marginStart="3dp"

android:layout\_marginTop="3dp"

android:layout\_marginEnd="3dp"

android:layout\_marginBottom="3dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_info\_24"

android:drawablePadding="20dp"

android:hint="password"

android:inputType="textPassword"

android:drawableRight="@drawable/baseline\_visibility\_off\_24"

android:paddingRight="10dp"

android:longClickable="false"

android:padding="20dp"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:textStyle="bold" />

<com.google.android.material.button.MaterialButton

android:id="@+id/LoginBtn"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/forgotPass"

android:layout\_centerHorizontal="true"

android:layout\_marginStart="20dp"

android:layout\_marginTop="20dp"

android:layout\_marginEnd="20dp"

android:layout\_marginBottom="20dp"

android:background="@drawable/roundbtn"

android:backgroundTint="@color/cardview\_dark\_background"

android:text="LOGIN"

app:backgroundTint="@color/white"

android:textColor="@color/black"

android:textStyle="bold"

/><TextView

android:id="@+id/tv"

android:layout\_width="200dp"

android:layout\_height="40dp"

android:layout\_centerInParent="true"

android:text="Not registered yet"

android:textStyle="bold"

android:textSize="20dp"

android:gravity="center"

android:layout\_below="@id/LoginBtn"/>

<TextView

android:id="@+id/RegisterHere"

android:layout\_width="400dp"

android:layout\_height="60dp"

android:layout\_below="@id/tv"

android:layout\_centerInParent="true"

android:layout\_marginLeft="-48dp"

android:layout\_marginTop="-17dp"

android:gravity="center"

android:text="Register here"

android:textColor="@color/white"

android:textSize="20dp"

android:textStyle="bold" /><TextView

android:id="@+id/forgotPass"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="forgot password"

android:layout\_alignParentRight="true"

android:textColor="@color/white"

android:textStyle="bold"

android:gravity="center"

android:textSize="20dp"

jm</RelativeLayout>

**TESTING**

Testing is a process of executing a program with the interest of finding an error. A good test is one that has high probability of finding the yet undiscovered error. Testing should systematically uncover different classes of errors in a minimum amount of time with a minimum amount of efforts. Two classes of inputs are provided to test the process

1. A software configuration that includes a software requirement specification, a design specification and source code.

2. A software configuration that includes a test plan and procedure, any testing tool and test cases and their expected results.

**1. Unit Testing**

Unit test comprises of a set test performed by an individual program prior to the integration of the unit into large system. A program unit is usually the smallest free functioning part of the whole system. Module unit testing should be as exhaustive as possible to ensure that each representation handled by each module has been tested. All the units that makeup the system must be tested independently to ensure that they work as required. During unit testing some errors were raised and all of them were rectified and handled well. The result was quite satisfactory and it worked well.

**2. Integration Testing**

Integration testing is a system technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested modules and build a program structure that has been dictated by design. Bottom-up integration is the traditional strategy used to integrate the components of a software system into functioning whole. Bottom-up integration consists of unit test followed by testing of the entire system. A sub-system consists of several modules thatcommunicated with other defined interface. The system was done the integration testing. All the modules were tested for their compatibility with other modules. They test was almost successful. All the modules coexisted very well, with almost no bugs. All the modules were encapsulated very well so as to not hamper the execution of other modules.

**3. Validation Testing**

After validation testing, software is completely assembled as a package, interfacing errors that have been uncovered and corrected and the final series of software test; the validation test begins. Steps taken during software design and testing can greatly improve the probability of successful integration in the larger system. System testing is actually a series of different tests whose primary purpose is to fully exercise the compute –based system.

### Test roles by Vaishnavi Sinnur:

|  |  |  |
| --- | --- | --- |
| Error | Correction | Output |
| Enter valid information in new student page | Debugged the error | Expected output |
| Login page error | Connected the code properly with the page | Expected output |

### Test roles by SrushtiGoudar:

|  |  |  |  |
| --- | --- | --- | --- |
| Textcase | Error | Correction | Output |
| 1. | College Image Display | Debugging the code | Expected Output |
| 2. | Staff details | Worked on it | Expected Output |

**The devices in which it is Tested:**

## Vivo V23

## Redmi note 7 pro

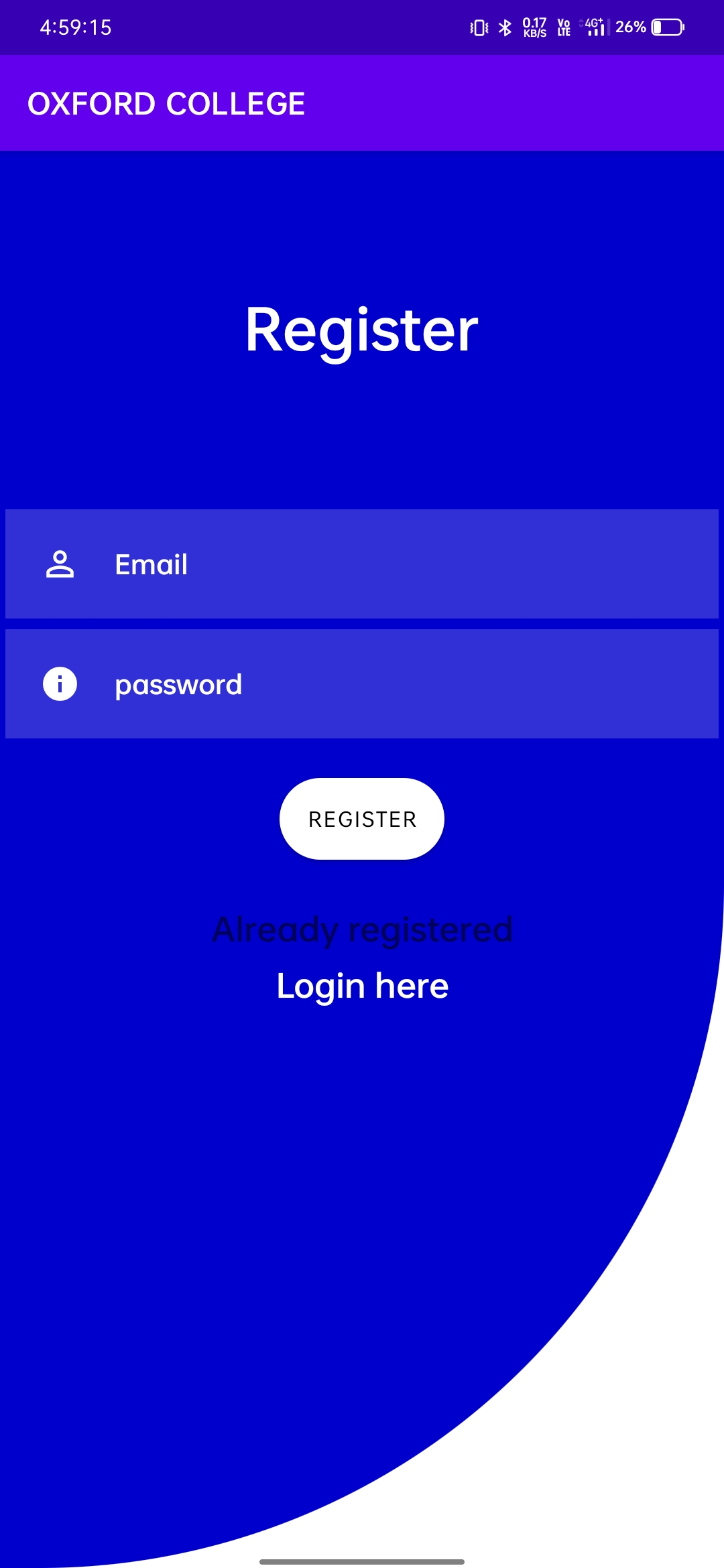
## Samsung Galaxy A9(2018)

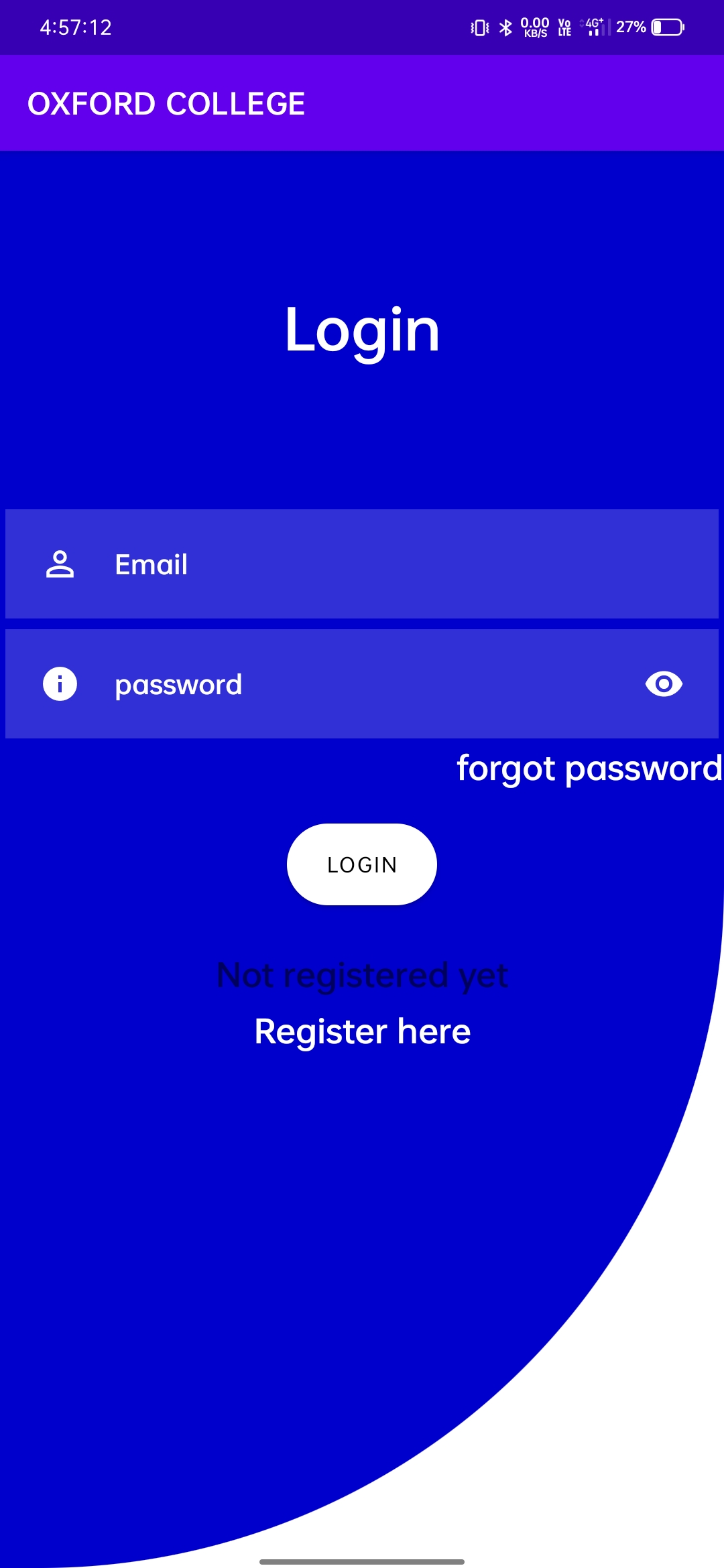
## Oppo F19s

## Redmi note 10 pro

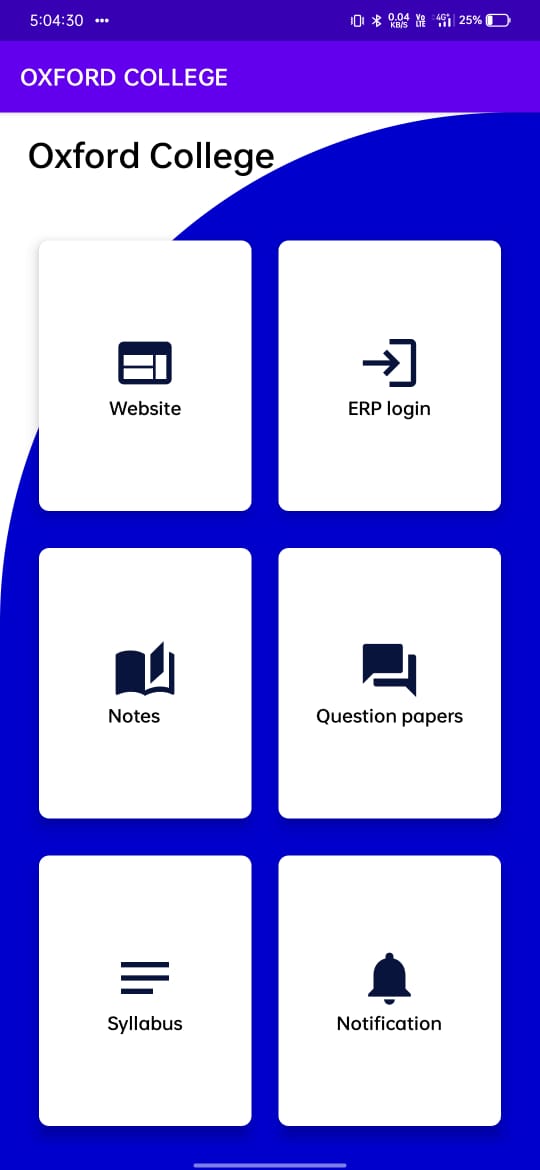
## Project execution screenshots

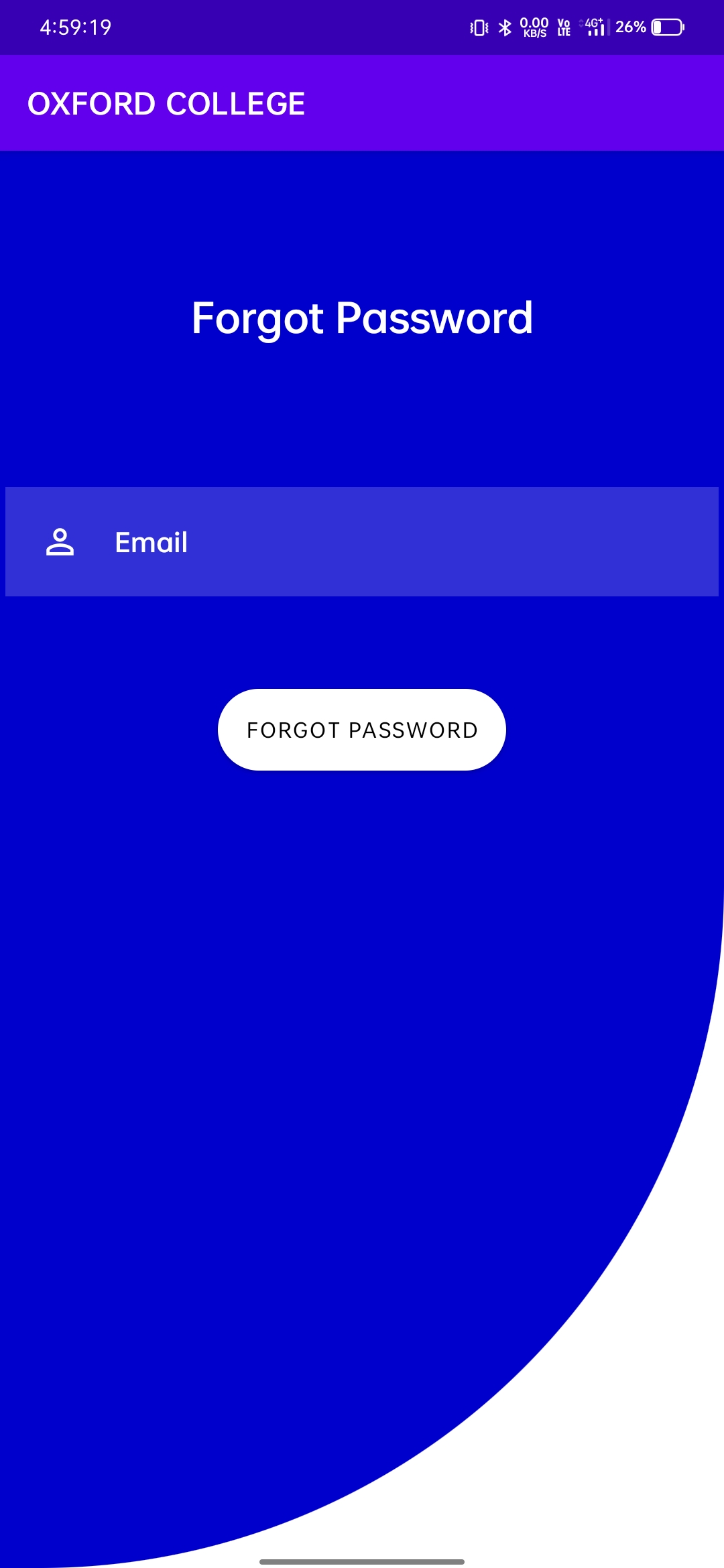
1. **LOGIN PAGE : 2. REGISTER PAGE:**



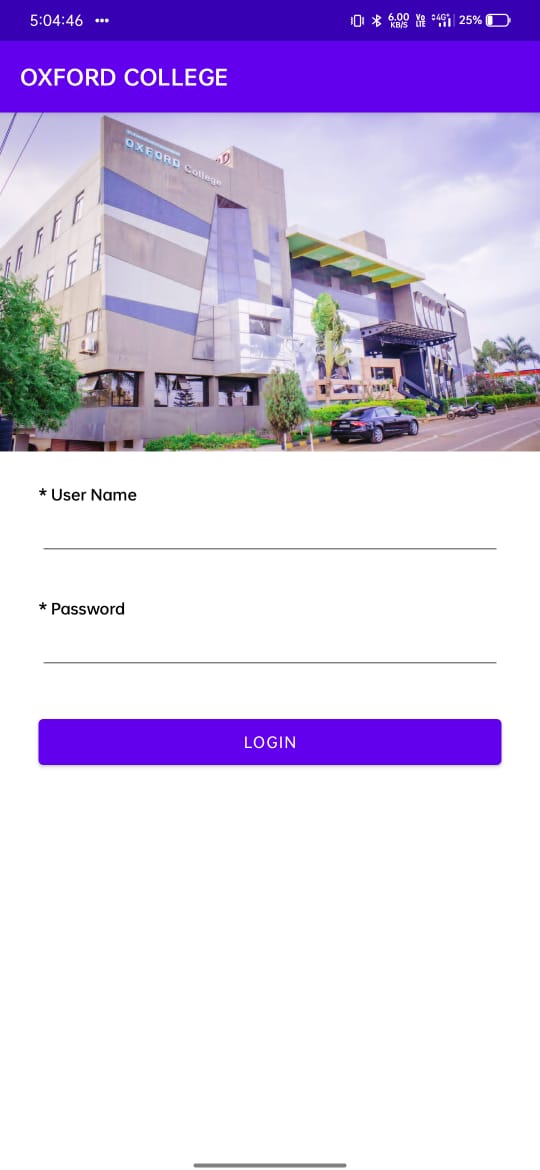


**3.Forgot Password 4.HomePage**



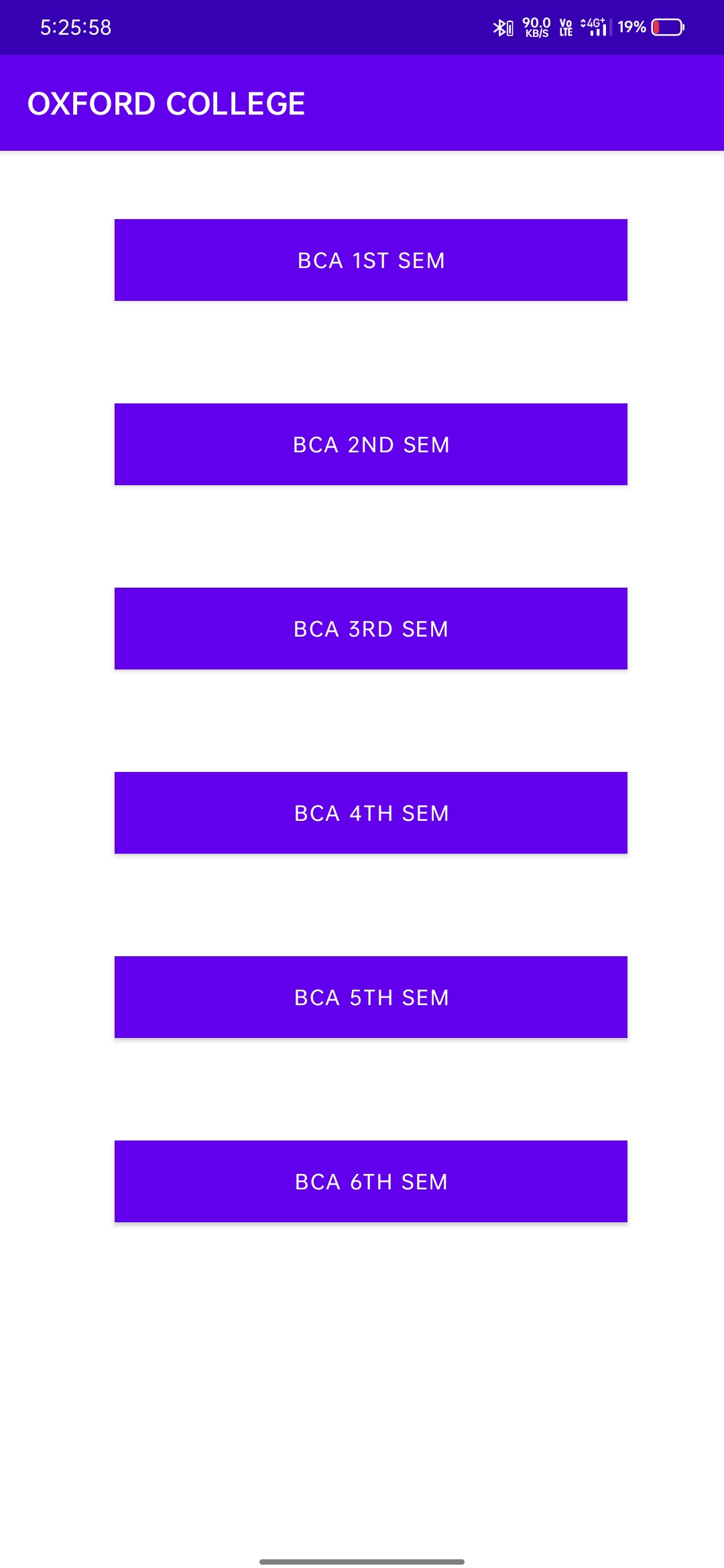


**4. WEBSITE PAGE: 5. ERP LOGIN:**





**6.NOTES, SYLLABUS, QUESTIONPAPERS**



**Limitations:**

### 1.The student result management system is prone to hacks.

### 2.Must require good signal streangth.

### 3.Extensive modules and features make it difficult for a user to utilize the application.

### 4.Minor technical glitches and issues.

### 5.Works only on Android os but not on iosplatfrom.

#### Conclusion:

#### Android studio is the best IDE in the market for android development. It increases the productivity with its great set of tools and plugins.

#### It is easy to use quick to learn with effective debugging features

#### It has got a lot of features like Emulator, Plugins, Android SDK, Code shortcuts(hints)

#### Future enhancement:

In future we are willing to fulfil the database

When new idea comes into brain , we are going to implement it and provide the update to the user

We are going to add more features

**Bibliography.**

1. "SQLite: A Comprehensive Guide" by Grokking Android (Website)

2. "Developing Android Apps with Java" by Google (Website)

3. "SQLite Tutorial for Android" by Ray Wenderlich (Website)

4. "Java for Android: Developing Mobile Apps" by John Horton (Book)

5. "Professional Android App Development with Java" by Reto Meier (Book)

6. "The Complete Android & Java Developer Course - Build 21 Apps" by Udemy (Online Course)

7. "Building Mobile Apps with SQLite and Java" by Sitepoint (Website)

8. OpenAI (chatgpt).