A Project Report on

"EXCELPRO"

Submitted in Partial Fulfillment of the Requirements for the award of the degree

Bachelor of Computer Applications

OF

BENGALURU CITY UNIVERSITY

SUBMITTED BY

SASITHAR M **R1811743**

GAIN MORE KNOWLEDGE
REACH GREATER HIEGHT
PRESIDENCY COLLEGE

PRESIDENCY COLLEGE

Kempapura, Hebbal, Bengaluru – 24

Re-accredited by NAAC with 'A+'

DEPARTMENT OF COMPUTER APPLICATIONS

PRESIDENCY COLLEGE

Kempapura, Hebbal, Bengaluru – 24

Re-accredited by NAAC with 'A+'

DEPARTMENT OF COMPUTER APPLICATIONS

GAIN MORE KNOWLEDGE
REACH GREATER HIEGHT
PRESIDENCY COLLEGE

CERTIFICATE

*This is to certify that **SASITHAR M** with Register No. **R1811743** has satisfactorily completed the Sixth Semester BCA Project titled "**EXCELPRO** ", as a partial fulfilment of the requirements for the award of the Degree in **Bachelor of Computer Applications**, awarded by **Bengaluru City University**, during the Academic Year **2020-2021.***

Project Guide: Mrs.Suma.SV

Head of Department
(Department of computer Application)

Examiners

1. ----------

2. ----------

Reg No: ----------

Examination Center: ----------

Date of the exam: ----------

\

Declaration

The project titled "**EXCELPRO**" developed by me in the partial fulfillment of Bengaluru City University. It is a systematic work carried by us under the guidance of **Mrs.Suma.SV,** Assistant Professor, Department of Computer Applications.

I, declare that this same project has not been submitted to any degree or diploma to the Bengaluru City University or any other Universities.

Name of the student: - **SASITHAR M**

Date: -

Signature

Acknowledgement

The development of software is generally bit complex and time-consuming task. The goal of developing the project "**EXCELPRO**" could not be archived without the encouragements of kindly helpful and supportive people. Here by we convey our sincere thanks for all of them.

I take this opportunity to express my gratitude to people who had been instrumental in the successful completion of this project.

I am thankful to our management trustee for providing us an opportunity to work and complete the project successfully.

I wish to express my thanks to our Principal **Dr. PRADEEP KUMAR SHINDE** for his support to the project work. I would like to acknowledge my gratitude to our HOD of Master of Computer Applications. **Dr. ALLI.A** for her encouragement and support. Without her encouragement and guidance this project would not have materialized.

The guidance and support received from our Internal Guide **Mrs.Suma.SV** who contributed to this project, was vital for the success of the project. We are grateful for their constant support and help.

INDEX

1. INTRODUCTION

1.1 Objective and Introduction:

In the recent times, keeping track of the progress in performance is highly necessary. The evolving ERP systems can make a significant change in the process of tracking and acknowledging a student's progress in performance and analyzing it in various ways. The system provides a comprehensive solution to the demand of examination result computation as well as student information and academic record management.

ExcelPro is a web-based application developed for colleges to analyze the result and keep track of students. The system is resilient and written with the flexibility to accommodate future needs. This system helps to calculate result fast so it optimizes the manpower. At a time, we can see all the years result in a single sheet and we are able to see the individual candidate's result separately.

The system provides a comprehensive solution to the demand of examination result, which is tedious and exorbitant. It facilitates computation of student information and academic record management in pictorial forms as well for getting instant idea.

Major Modules:

- Admin
- Faculty
- Students

Functionalities of Modules:

Admin can perform the following operations:

- Enroll Faculty and Students.
- Upload Mark list with Optical Character Recognition (scan result copies).
- CRUD Operations with student and faculty entries.
- Send notifications to students.

- Receive feedbacks from faculty and students.
- Can add college events.

Faculty will have the following privileges:

- Obtaining membership
- Sending announcements and notifications to students.
- Editing mark list of students.
- Changing profile password.
- Review, Assess and generate result Graphs.
- Can help students by viewing messages received from them and giving feedbacks.
- Can update college events.

Students will have the following privileges:

- Receive internal exam results.
- Receive marks weightage charts and result graphs.
- Receive announcements and notifications from admin and faculty.
- Change profile password.
- Send messages to faculty.
- Send feedback to admin.
- Edit Profile information.
- Change profile picture.

1.2 OUTLINE AND BRIEF DESCRIPTION OF THE ISSUES INVOLVED

The administrator has the full privilege in this application. The user with admin login can add new service details as when introduced in the market. The financial scenarios of clients are obtained and service booking plans and services are given. Also, on regular basis admin keeps checking the market trends and updates the service for the customers and generate the report.

The customers will have to register to do business through our web application. Only the authentic customers can make transactions and get their service status. The authenticity of customers is maintained through user name and password. Customers are facilitated with password change; they can register new password if their password is compromised. The Customers have options to view the new services and charges details and choose the services accordingly for their events.

2. SOFTWARE REQUIREMENT ANALYSIS

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Analysis begins when a developer begins a study of the program using existing system. During analysis, data are collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram, interviews, etc. Training, experience and common sense are required for collection of relevant information needed for the development of the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs.

2.1 PRESENT SYSTEM

In the present system, more time is required to analyze the students results and propagate the result data, solutions and feedbacks to the students. Automating database and result works can help in spending more time productively. The proposed system can help to increase productivity and generate precise result data. It can help the faculty to assist students and help them progress in a more productive way with less time and effort.

2.2 PROPOSED SYSTEM

The project can analyze and generate report of students based on the curriculum that represents student's academic performance. We have developed the system such that, it will automatically parse data onto the database, which will in return reduce time consumption of analysis of data. Previously, data used to be inserted manually to analyze result. But currently the project uses Optical Character Recognition APIs and supports excel(.xlsx) files for extraction of data. Visualization is provided to represent result data in graphical format. Various representation like pie chart, graph, etc. A more convenient way of viewing announcements, results and attendance records is created so as to increase productivity. Students can do more interaction with faculty for more productivity and feedbacks.

3. SOFTWARE REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform.

An SRS minimizes the time and effort required by developers to achieve desired goals and also minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs and human users in a wide variety of real-world situations. Parameters such as operating speed, response time, availability, portability, maintainability, footprint, security and speed of recovery from adverse events are evaluated.

The requirements of this web application (ems) developed by us is very well understood and documented initially when the process started. Accordingly, the hardware and software requirements are chosen. The same can be referred in the next section of this report. This web application runs in the Apache Tomcat Server, which can be accessed by web clients through the Internet from anywhere in the world. The SRS parameters are adopted in this system, with the detailed feasibility study. While developing, this system tested as the localhost, through the Mozilla Firefox browser. Also, few other popular web browsers are tested.

4. SOFTWARE AND HARDWARE CONFIGURATION

HARDWARE	
Processor	64-bit, Intel core i5
RAM	4GB
HDD	500GB
SOFTWARE	
CLIENT-SIDE TECHNOLOGIES	ANDROID, XML
SERVER-SIDE TECHNOLOGIES	FIREBASE, Apache POI
WEB SERVER	FIREBASE
PROGRAMMING LANGUAGE	JAVA
DATABASE	FIREBASE
IDE	ANDROID STUDIO 4.2.2

5. SOFTWARE PROFILE

5.1 ABOUT 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, mac OS and Linux based operating systems or as a subscription-based service in 2020. 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. On May 7, 2019.

Characteristics of Android Studio

- A flexible Gradle-based build system
- A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting your app

- Code templates and GitHub integration to help you build common app features and import sample code
- Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine

.2 ABOUT JAVA

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2012, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 10 million users . Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low level facilities than either of them. Java can be used to write applications and applets. A Java application is similar to any other high-level language program: It can only be compiled and then run on the same machine. An applet is compiled on one machine, stored on a server in binary, and can be sent to another machine over the Internet to be interpreted by a Java-aware browser. Java comes with a large library of ready-made classes and objects. The key difference between Java 1.0 and 1.1 was in this library. Similarly, Java 2.0 has a very much larger library for handling user interfaces (Swing by name) but only small changes to the core of the language.

5.3 ABOUT APACHE POI

Apache POI, a project run by the Apache Software Foundation, and previously a sub-project of the Jakarta Project, provides pure Java libraries for reading and writing files in Microsoft Office formats, such as Word, PowerPoint and Excel.

The name was originally an acronym for "Poor Obfuscation Implementation "referring humorously to the fact that the file formats seemed to be deliberately obfuscated, but poorly, since they were successfully reverse-engineered. This explanation – and those of the similar names for the various sub-projects – were removed from the official web pages in order to better market the tools to businesses who would not consider such humor appropriate. The original authors (Andrew C. Oliver and Marc Johnson) also noted the existence of the Hawaiian poi dish, made of mashed taro root, which had similarly derogatory connotations.

5.4 ABOUT FIREBASE

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google's infrastructure.

Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

Key Features

1. Authentication

It supports authentication using passwords, phone numbers, Google, Facebook, Twitter, and more. The Firebase Authentication (SDK) can be used to manually integrate one or more sign-in methods into an app.

2. Real-time database

Data is synced across all clients in real-time and remains available even when an app goes offline.

3. Hosting

Firebase Hosting provides fast hosting for a web app; content is cached into content delivery networks worldwide.

4. Test lab

The application is tested on virtual and physical devices located in Google's data centers.

5. Notifications

Notifications can be sent with firebase with no additional coding.

5.5 ABOUT XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The World Wide Web Consortium's XML 1.0 Specification of 1998 and several other related specifications—all of them free open standards—define XML.

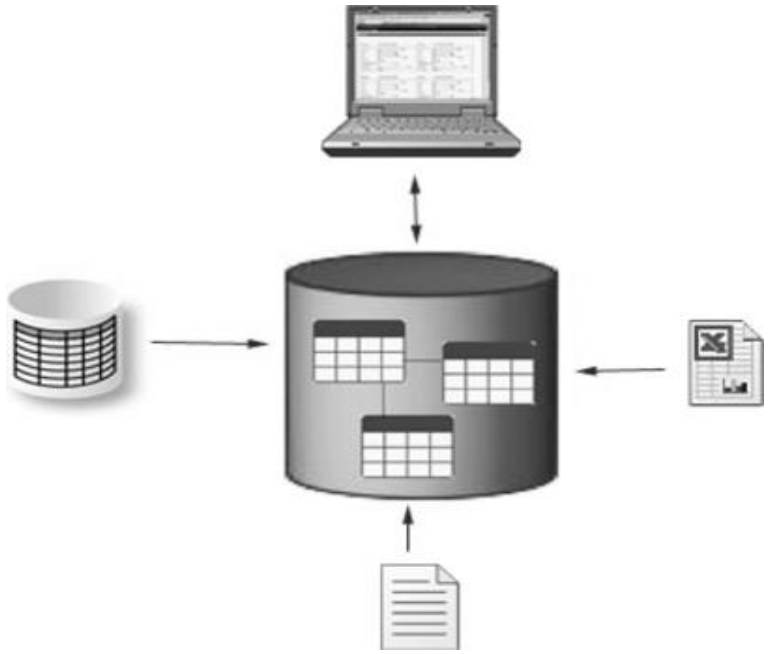
The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many applications programming interfaces (APIs) to aid the processing of XML data.

5.6 ABOUT EXCEL

Microsoft Excel is a spreadsheet developed by Microsoft for Windows, macOS, Android and iOS. It features calculation, graphing tools, pivot tables, and a macro programming language called Visual Basic for Applications (VBA). It has been a very widely applied spreadsheet for these platforms, especially since version 5 in 1993, and it has replaced Lotus 1-2-3 as the industry standard for spreadsheets. Excel forms part of the Microsoft Office suite of software.

5.7 DATABASE MANAGEMENT SYSTEM



A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. This is a software system that allows access to the data contained in the database. The primary goal of a DBMS is to provide an environment that is both convenient and efficient to use in storing and retrieving database information.

Functions of DBMS: -

DBMS is a general purpose software system that performs the following functions:-

- Defining a database.
- Constructing the database.
- Manipulating the database.

- Sharing database among various users.
- Protecting the database.
- Maintaining a database.

DBMS Architecture: -

DBMS architecture consists of three levels known as Three Schema Architecture. It is convenient tool with which the user can visualize the schema levels in a database system. It contains of the following three schemas: -

1. ***The Internal level*: -** This contains of an internal schema, which describes the physical storage structure of the database. It is the lowest level of abstraction. It does not hide the storage details. It contains the definition of the stored record, the method of representing the data fields and the access aids used. This internal schema uses a physical data model and describes the complete details of data storage and access paths for the database. It is also called the physical schema.

2. ***The Conceptual level*: -** This has a conceptual schema, which describes the structures of a database for a group of users. This schema hides the storage details from the user and it includes description of entities, data types, relationships, user operations and constraints. The description of data at this level is in a format independent of its physical representation.

3. ***The External level*: -** This has a number of external schemas or user views. Each external schema describes the part of the database that a particular user group is interested in and hides all the other details from this group. This is at a highest level of database absorption.

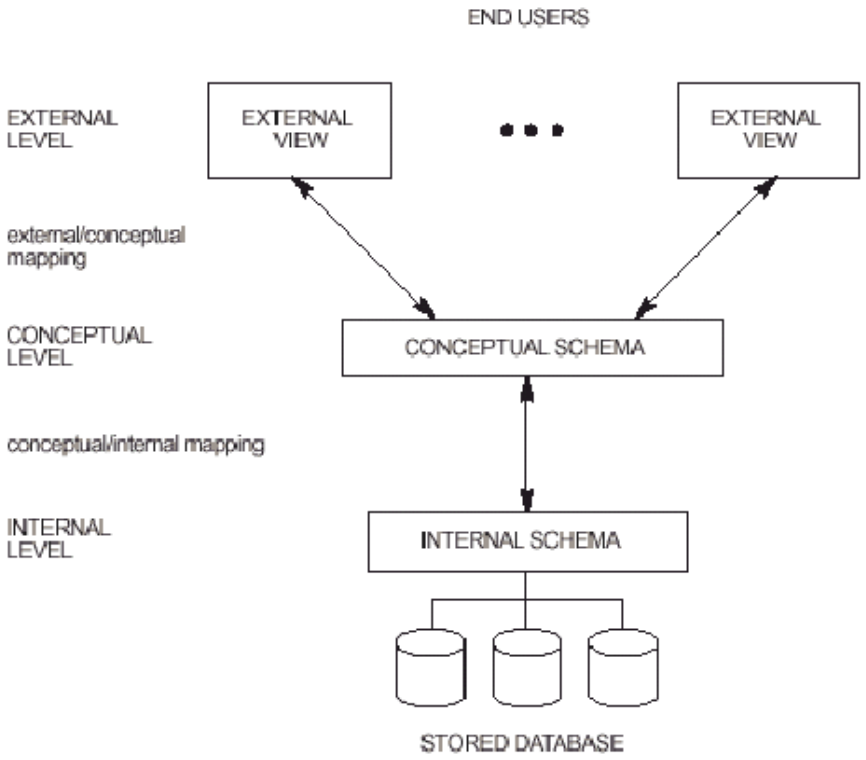


Fig: - Three schema DBMS architecture

Client-Server Architecture: -

The client/server architecture was developed to deal with computing environments in which a large number of PCs, workstations, file servers, printers, database servers, Web servers and other equipment are connected via a network. There are two client/server architecture: -

- Two-tier
- Three-tier
- N-tier

Two-Tier Client/Server Architecture: -

In two tier architecture, the software components are distributed over two systems: the client and the server. This architecture has two forms as: logical two-tier and physical two-tier.

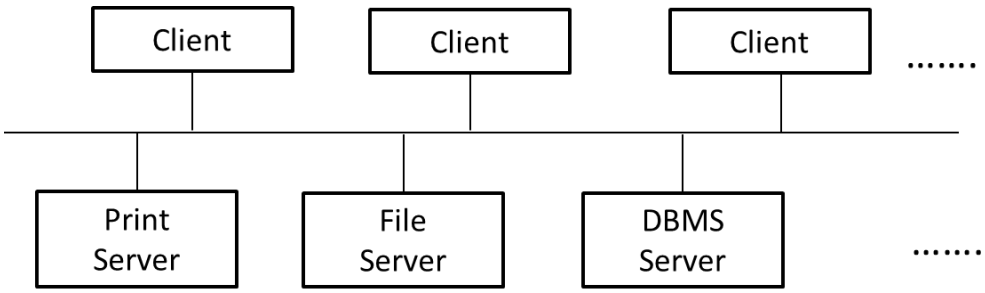


Fig: - Logical two-tier client/server architecture

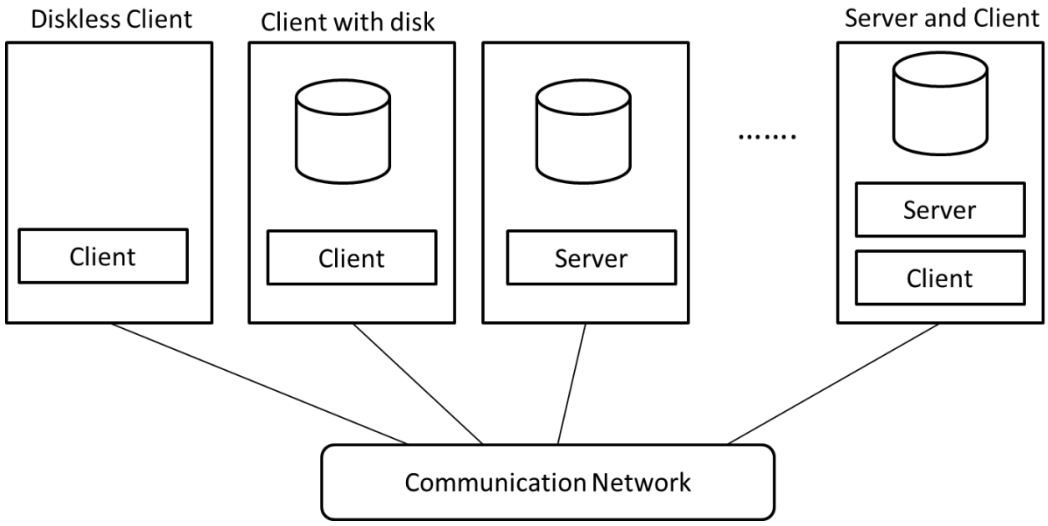


Fig: - Physical two-tier client/server architecture

The above figure shows the physical client/server architecture. Some machines like diskless workstations, or workstations/PCs with disks that have only client software

installed would be only client sites. Other machines would be dedicated servers. Some other machines would have both client and server functionality. In relational DBMS, user interface and application programs can run at client side. The query and transactions functionality are included on the server side. A client program can send query and transaction requests using the ODBC API which are then processed at the server site. The query results are sending back to the client program which can process or display the results needed.

Three-Tier Client/Server Architecture: -

With the emergence of World Wide Web, many web applications use the three-tier architecture. There is an intermediate layer between the client and the database server called the application server or the web server. The web server plays the intermediary role by storing business rules that are used to access data from the database server. It checks the client's credentials before forwarding a request to the database server. The intermediate server accepts requests from the client, process the request and sends database commands to the database server, and then acts as a conduit for passing processed data from the database server to the clients, where it may be processed further.

Fig: - Logical three-tier client/server architecture

N-Tier Architecture: -

In N-Tier architecture, the middle tier is allowed to have multiple application objects rather than a single application. Each of these application objects must have a well-defined interface which allows them to contact and communication with one another. An interface actually brings an idea of contract. That is, each object states through its interface that it will accept certain parameters and return a specific set of results.

Application objects use their interfaces to do business processing. With and N-Tier architecture, one can have multiple applications using a common set of business objects across an organization. This promotes the standardization of business practices by creating a single set of business functions for the entire organization to access. If a particular business rule changes, then changes have to be made to only the business object and if need, to its interface also.

Normalization: -

Normalization is a process during which unsatisfactory relation schemas are decomposed by breaking up their attributes into smaller relation schemas that possess desirable properties.

Normalization of data can be looked upon as a process of analyzing the given relation schemas based on their functional dependencies and primary keys to achieve the desirable properties of minimizing redundancy, insertion, deletion and update anomalies.

The different Normal Forms present in DBMS are: -

- ***First Normal Form:*** **-** It states that the domains of attributes must include only atomic values and that the value of any attribute in a tuple must be single value from the domain of that attribute. It disallows a set of values, a tuple of values, or a combination of both as an attribute value for a single tuple.

- ***Second Normal Form: -*** A relation is said to be in Second Normal Form if it is in 1NF and non-key attributes are fully functionally dependent on the key attribute(s). If the key has more than one attribute (composite key) then no non-key attributes should be functionally dependent upon a part of the key attributes.

- ***Third Normal Form: -*** A relation is said to be in Third Normal Form if it is in 2NF and no non-prime attributes of relation R is transitively dependent on the primary key.

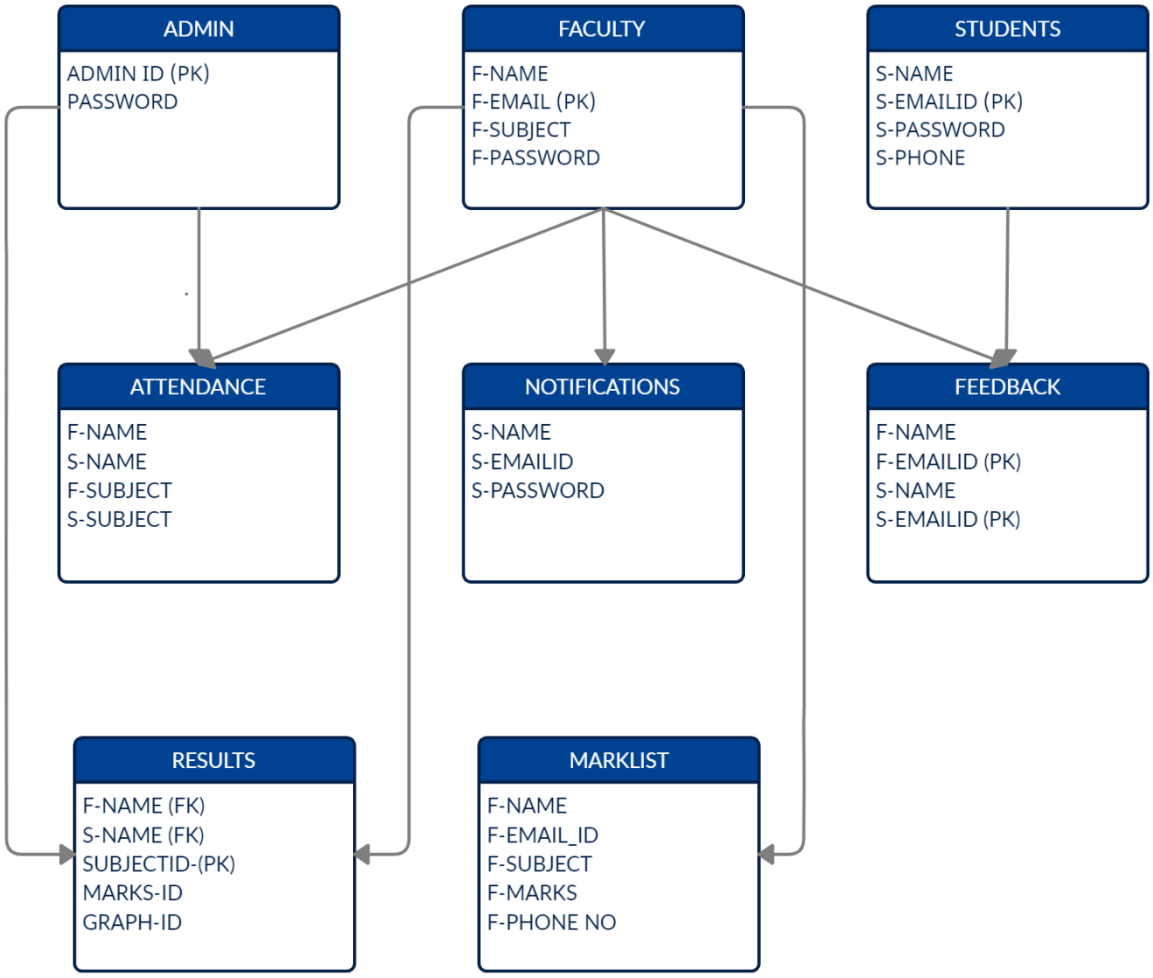
- ***Fourth Normal Form: -*** Under this, a record type should not contain two or more independent multi-valued facts about an entity. In addition, the record must satisfy third normal form.

- ***Fifth Normal Form: -*** It also deals with multi-valued facts. Here, the record must satisfy the fourth normal form.

- ***Boyce Codd Normal Form: -*** It a normal form used in database normalization. It is a slightly stronger version of the third normal form (3NF). A table is in Boyce–Codd normal form if and only if for every one of its nontrivial dependencies $X \rightarrow Y$, X is a super key—that is, X is either a candidate key or a superset thereof

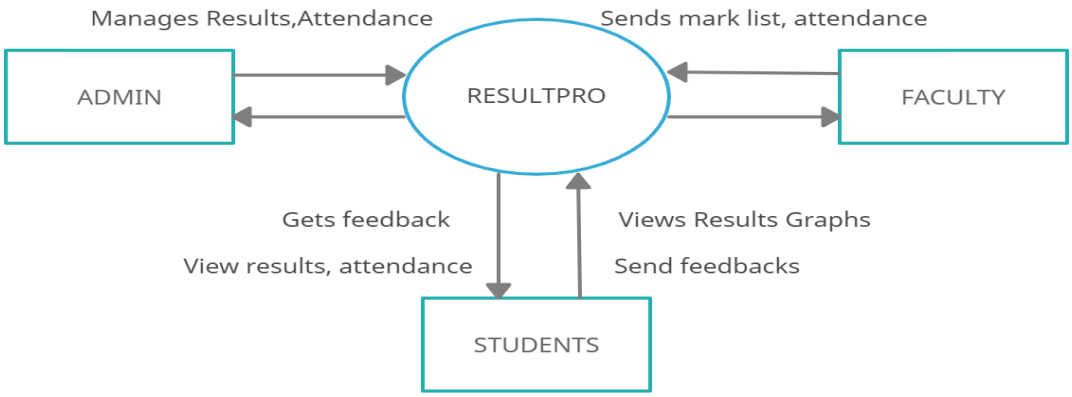
6. SYSTEM DESIGN

6.1 DATABASE SCHEMA DESIGN:

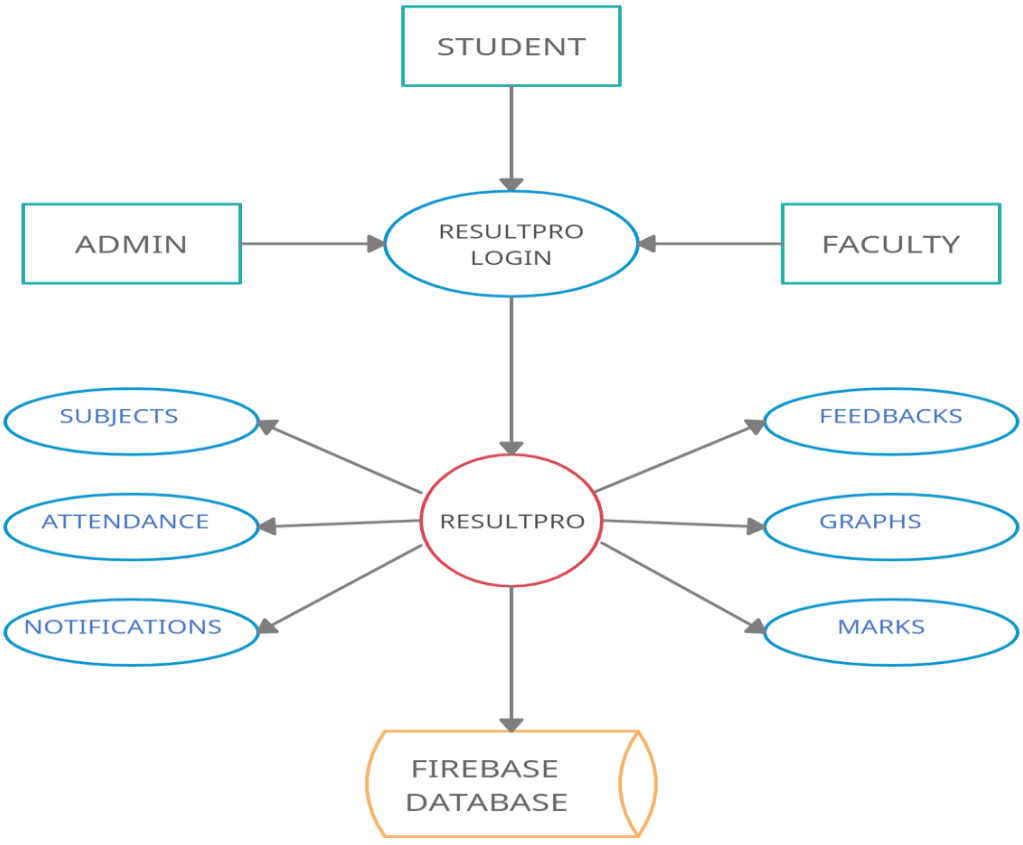


6.3 DATA FLOW DIAGRAM

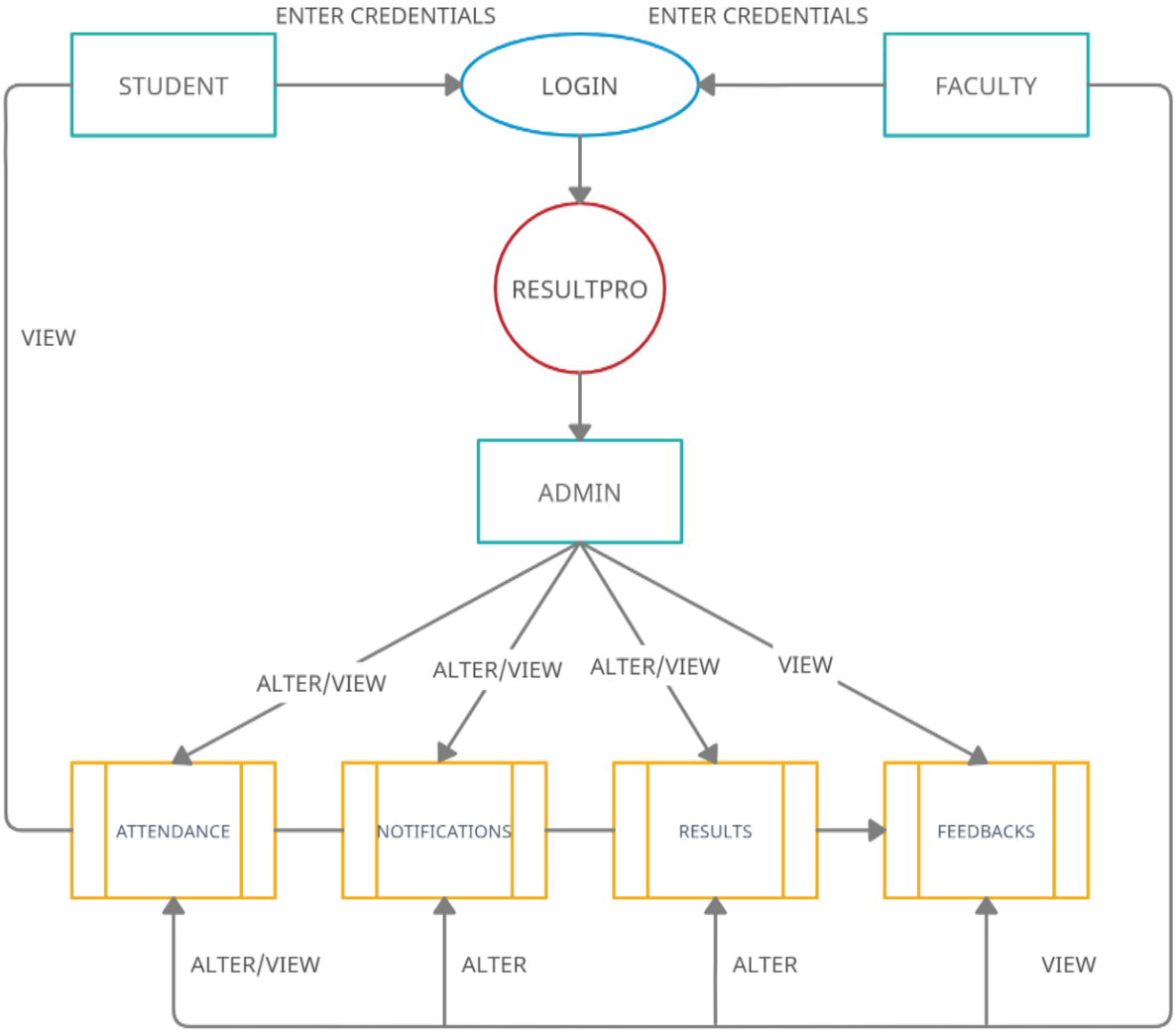
LEVEL 0:



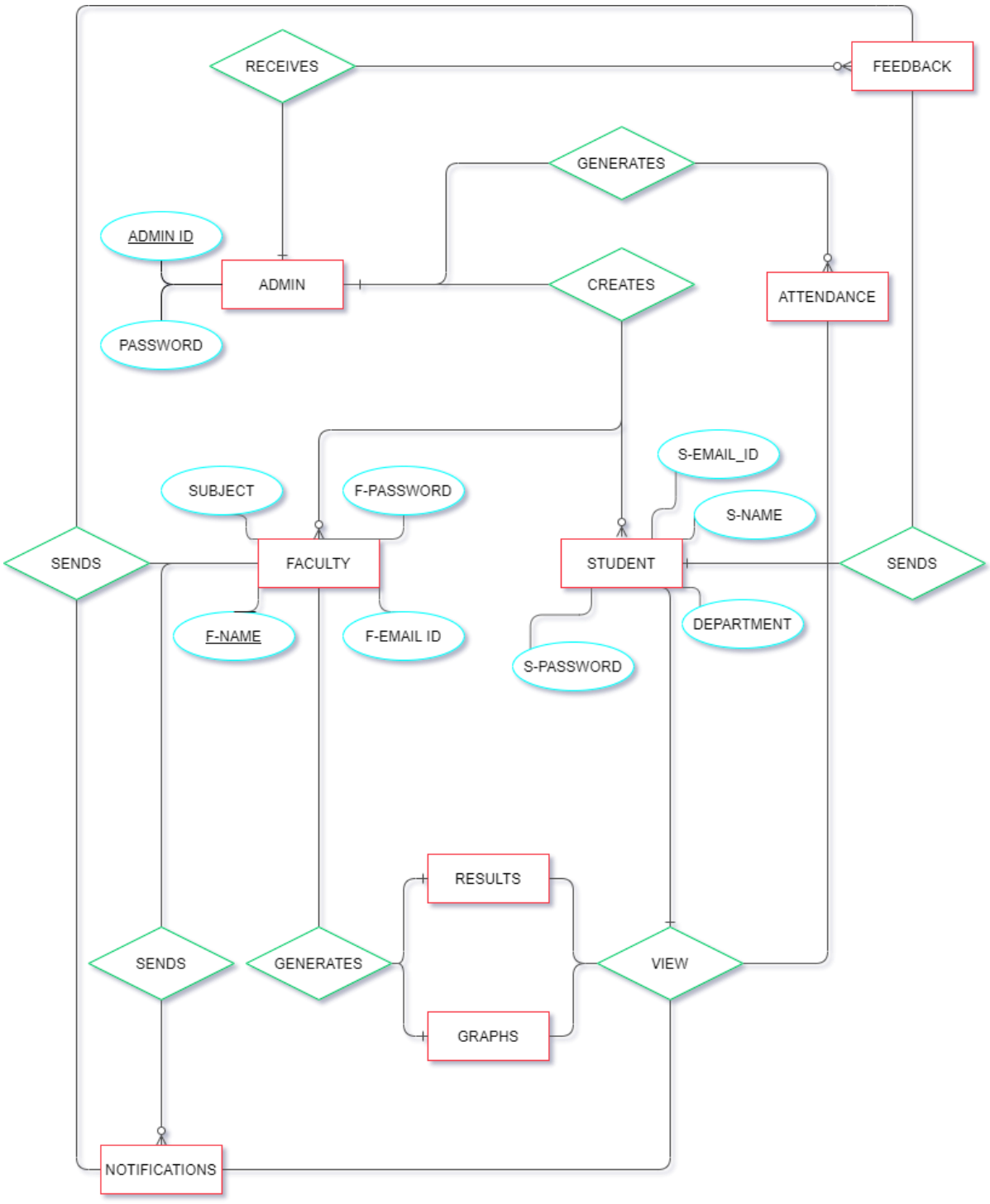
LEVEL 1:



LEVEL 2:


ENTER CREDENTIALS
ENTER CREDENTIALS
STUDENT
LOGIN
FACULTY
RESULTPRO
VIEW
ADMIN
ALTER/VIEW
ALTER/VIEW
ALTER/VIEW
VIEW
ATTENDANCE
NOTIFICATIONS
RESULTS
FEEDBACKS
ALTER/VIEW
ALTER
ALTER
VIEW

6.4 ER DIAGRAM



7. TESTING

7.1 CODING , TESTING AND IMPLEMENTATION

The main interfaces are:

- Web server and application server interface.
- Application server and Database server interface.

Check if all the interactions between these servers are executed properly. Errors are handled properly. If database or web server returns any error message for any query by application server then application server should catch and display these error messages appropriately to users. Check what happens if user interrupts any transaction in-between? Check what happens if connection to web server is reset in between?

7.2 TESTING AND ERRORS

Compatibility of our web site is very important testing aspect. See which compatibility test to be executed:

- Browser compatibility
- Operating system compatibility
- Mobile browsing
- Printing options

Browser compatibility:

Some applications are very dependent on browsers. Different browsers have different configurations and settings that your web page should be compatible with. Your web site coding should be cross browser platform compatible. If you are using java scripts or AJAX calls for UI functionality, performing security checks or validations then give more stress on browser compatibility testing of your web application. Test web application on different browsers like Internet explorer, Firefox, Netscape navigator, AOL, Safari, Opera, Chrome browsers with different versions.

OS compatibility:

Some functionality in your web application is may not be compatible with all operating systems. All new technologies used in web development like graphics designs, interface calls like different API's may not be available in all Operating Systems.

Test your web application on different operating systems like Windows, Unix, MAC, Linux, Solaris with different OS flavors.

7.3 SAMPLE TEST CASES DONE

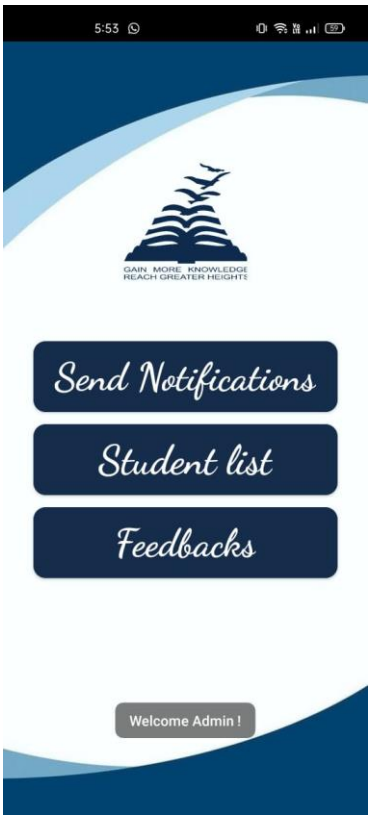
Sl.No.	Test Case Done	Action taken
1	Invalid user credentials entered	Redirected to error page. Customer of Admin main page not shown.
2	Incomplete form entry in the registration and few other pages.	Users are prompted to complete the page to proceed next.
3	Illegal data entry. E.g., amount entered in negative values. Phone numbers out of range, etc.,	All such errors are captured and prompted accordingly so that users will provide only the legal input to the system.
4	Invalid credit card number entry.	Captured and prompted the user to enter only the valid credit card number.
5	Incorrect Client ID entry	Error message notified to the Client ID.

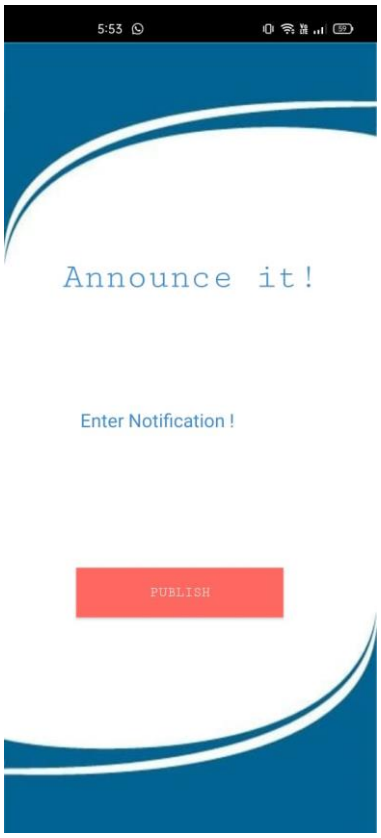
8.SCREENS


9:42 AM
ResultPro
Excel like a pro!

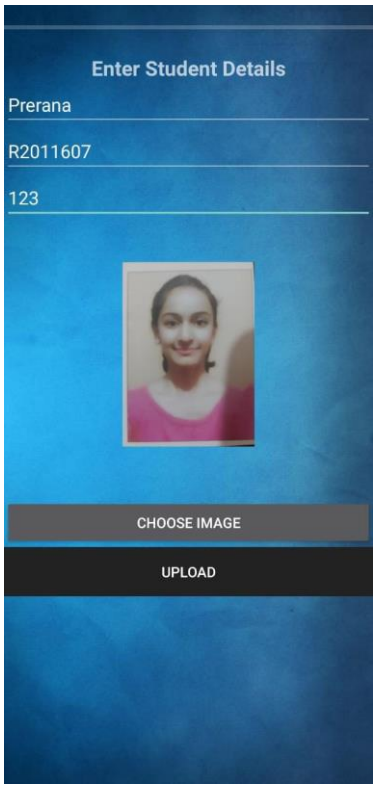

9:31 AM
ResultPro
Admin
Faculty
Student
Exit


5:54
GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
admin
....
Attempts Left
2
LOGIN
Wrong Credentials


5:53
GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Send Notifications
Student list
Feedbacks
Welcome Admin !

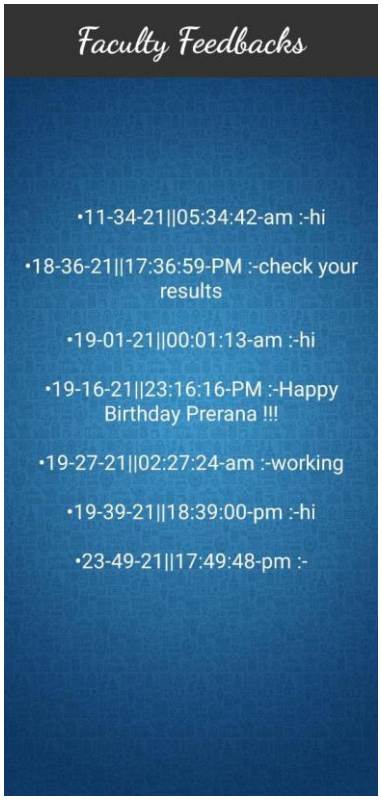

5:53
Announce it!
Enter Notification !
PUBLISH


5:53
GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Add Students
View All


Enter Student Details
Prerana
R2011607
123
CHOOSE IMAGE
UPLOAD

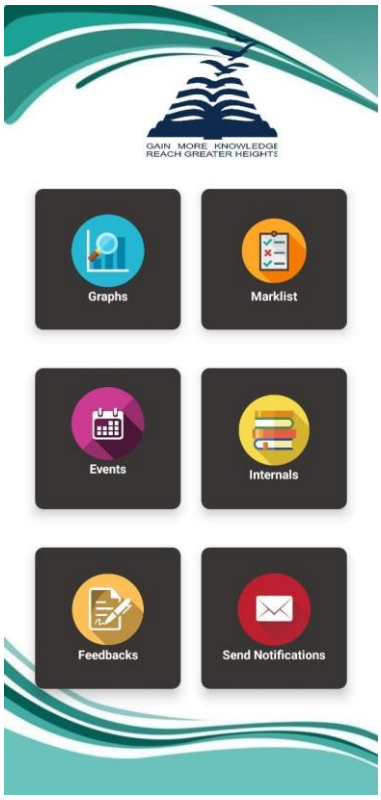

Student List
111
22/07/2021
111
sasi
22/07/2021
rr
22/07/2021
123
Item deleted
22/07/2021

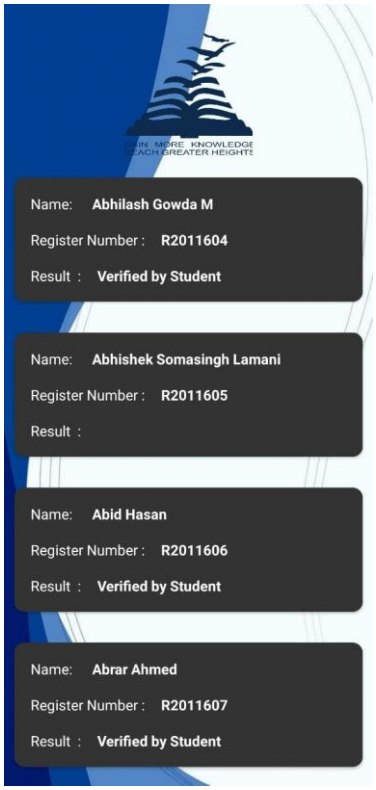

5:53
GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Faculty Feedback
Student Feedback


Faculty Feedbacks
•11-34-21||05:34:42-am :-hi
•18-36-21||17:36:59-PM :-check your results
•19-01-21||00:01:13-am :-hi
•19-16-21||23:16:16-PM :-Happy Birthday Prerana !!!
•19-27-21||02:27:24-am :-working
•19-39-21||18:39:00-pm :-hi
•23-49-21||17:49:48-pm :-


Student Feedbacks
•12-01-21||01:01:13-am :-Hi
•18-20-21||18:20:25-PM :-I'm satisfied with the class
•20-40-21||19:40:54-PM :-
•21-03-21||20:03:30-PM :-Hi Monisha and kaushik dgjmvnfdbvljdfbvndf
•21-05-21||14:05:41-pm :-g
•22-24-21||10:24:09-am :-


GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
prerana197@gmail.com
LOGIN


GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Graphs
Marklist
Events
Internals
Feedbacks
Send Notifications

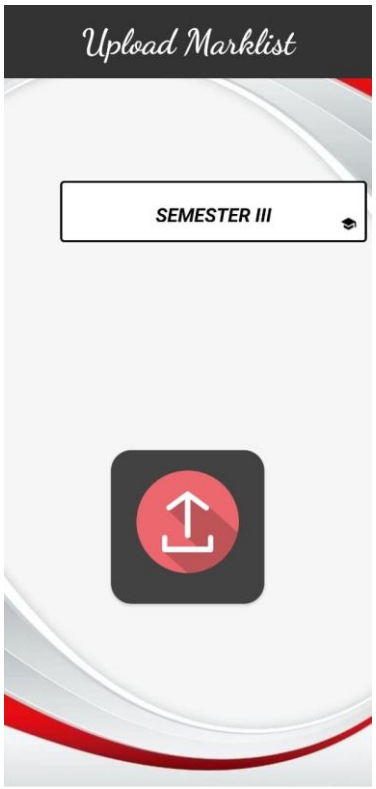

Name: Abhilash Gowda M
Register Number : R2011604
Result : Verified by Student
Name: Abhishek Somasingh Lamani
Register Number : R2011605
Result :
Name: Abid Hasan
Register Number : R2011606
Result : Verified by Student
Name: Abrar Ahmed
Register Number : R2011607
Result : Verified by Student

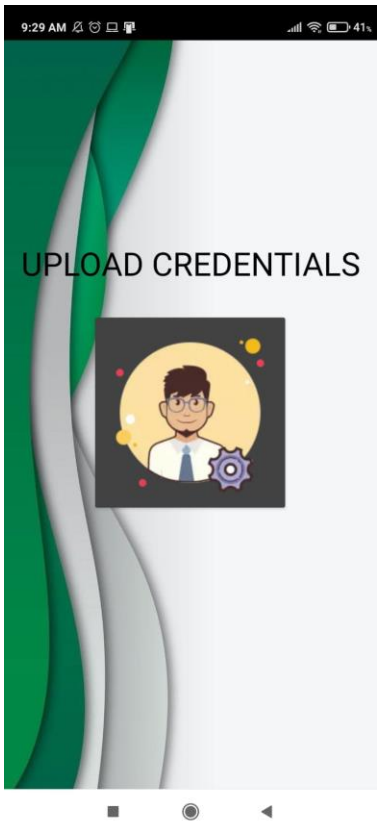

Faculty Feedback
Valuable Feedback
SEND FEEDBACK


GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS


BCA
B.Com
Upload Credentials


Announce it!
Enter Notification
PUBLISH

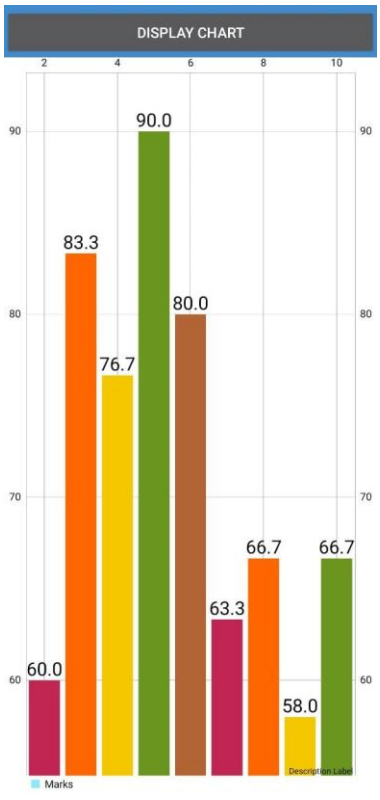

Upload Marklist
SEMESTER III


9:29 AM
UPLOAD CREDENTIALS


GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
R2011607
LOGIN


GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS
Graphs
Marks
Events
Subjects
Feedbacks
Notifications


Marks Card
SEMESTER I
NAME : Abrar Ahmed
REGISTER NUMBER : R2011607
Overall Percentage : 65.33334 %
GRADE : B+
OVERALL PERCENTAGE
1. Computer Organization 23.0
2. Indian Language 18.0
3. English 25.0
4. C Language 19.0
5. Office Automation 10.0
6. Discrete Mathematics 29.0
7. C LAB 10.0
8. Foundation Course 24.0
9. CC and EC 45.0
VERIFY

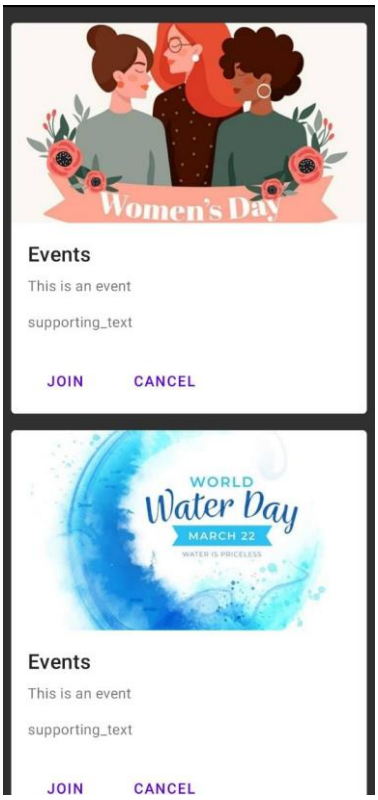

DISPLAY CHART
90.0
83.3
80.0
76.7
66.7
66.7
63.3
60.0
58.0
Marks

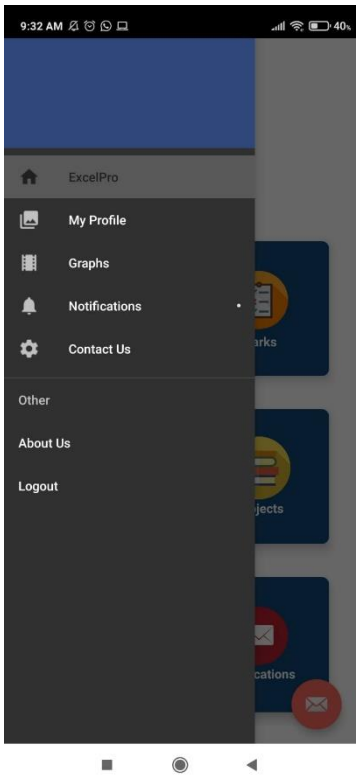
SUBJECT LIST

BCA
Bachelor of Computer Application

SEMESTER I

Sl.NO.	Paper Code	Subject Name
1	BCA 101T	Indian Language
2	BCA 102T	English
3	BCA 103T	Problem Solving Techniques using C
4	BCA 104T	Computer Organization
5	BCA 105T	Discrete Mathematics
6	BCA 103P	C Programming Lab
7	BCA 104P	Office Automation
8		Foundation Course
9		CC & EC





9. SAMPLE CODES

WELCOME.JAVA

```
package com.presidency.resultpro;

import androidx.cardview.widget.CardView;
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;
public class welcome extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);



        CardView c1 = findViewById(R.id.c1);
        c1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(welcome.this,AdminLogin.class);
                startActivity(intent);
            }
        });
        CardView c2 = findViewById(R.id.c2);
        c2.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            Intent intent = new Intent(welcome.this,FacultyLoginActivity.class);
            startActivity(intent);
        }
    });
    CardView c3 = findViewById(R.id.c3);
    c3.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(welcome.this,LoginActivity.class);
            startActivity(intent);
        }
    });
    CardView c4 = findViewById(R.id.c4);
    c4.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            int pid = android.os.Process.myPid();
       finishActivity(0);
       finishAndRemoveTask();
       finishAffinity();
       System.exit(0);

        }
    });


}
public void onBackPressed() {
// super.onBackPressed();
```

```
    // Not calling **super**, disables back button in current screen.
        Toast.makeText(getApplicationContext(),"Please Choose an
option!",Toast.LENGTH_SHORT).show();
    }
}
```

ACTIVITY_WELCOME.XML

```
<LinearLayout xmlns: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/wel9"
    android:orientation="vertical"
    tools:context=".welcome">

    <ImageView
        android:id="@+id/imageView8"
        android:layout_width="match_parent"
        android:layout_height="150dp"
        android:layout_marginTop="30dp"
        android:layout_marginBottom="36dp"
        android:src="@drawable/logo" />

    <androidx.cardview.widget.CardView
        android:id="@+id/c1"
        android:layout_width="290dp"
        android:layout_height="90dp"
        android:layout_gravity="center"
```

```
    android:layout_marginBottom="20dp"
    android:contextClickable="true"
    android:rotationY="0"
    android:textAlignment="center"
    app:cardBackgroundColor="#152d4b"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">

    <ImageView
        android:id="@+id/imageView4"
        android:layout_width="98dp"
        android:layout_height="wrap_content"
        android:src="@drawable/adminmenu" />

    <TextView
        android:id="@+id/viewmarks"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="cursive"
        android:text="@string/admin"
        android:textColor="#ffffff"
        android:textSize="38sp"
        android:textStyle="bold" />
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/c2"
    android:layout_width="290dp"
    android:layout_height="90dp"
```

```
    android:layout_gravity="center"
    android:layout_marginBottom="20dp"
    app:cardBackgroundColor="#152d4b"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">

    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="cursive"
        android:text="@string/faculty"
        android:textColor="#ffffff"
        android:textSize="40sp"
        android:textStyle="bold" />

    <ImageView
        android:id="@+id/imageView5"
        android:layout_width="102dp"
        android:layout_height="wrap_content"
        android:src="@drawable/facultymenu" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/c3"
    android:layout_width="290dp"
    android:layout_height="90dp"
    android:layout_gravity="center"
```

```
    android:layout_marginBottom="20dp"
    app:cardBackgroundColor="#152d4b"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginLeft="10dp"
        android:fontFamily="cursive"
        android:text="@string/student"
        android:textColor="#ffffff"
        android:textSize="40sp"
        android:textStyle="bold" />

    <ImageView
        android:id="@+id/imageView6"
        android:layout_width="70dp"
        android:layout_height="70dp"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="15dp"

        android:src="@drawable/studentmenu" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/c4"
```

```
    android:layout_width="146dp"
    android:layout_height="75dp"
    android:layout_gravity="center"
    android:layout_marginBottom="12dp"

    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">

    <TextView
        android:id="@+id/textView10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="cursive"
        android:text="@string/exit"
        android:textColor="#ffffff"
        android:textSize="40sp"
        android:textStyle="bold" />

  </androidx.cardview.widget.CardView>

</LinearLayout>
```

ADMIN LOGIN.JAVA

```
package com.presidency.resultpro;


import android.annotation.SuppressLint;
import android.app.Activity;
import android.content.Intent;
import android.graphics.Color;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class AdminLogin extends Activity {
    Button b1,b2;
    EditText ed1,ed2;
    ImageView img;
    TextView tx1,attempt;
    int counter = 3;
    @SuppressLint("SetTextI18n")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_login);
        b1 = findViewById(R.id.button25);
        ed1 = findViewById(R.id.editText25);
        ed2 = findViewById(R.id.editText27);
        attempt=findViewById(R.id.attempt);
        img=findViewById(R.id.img);
        tx1 = findViewById(R.id.textView11);
        tx1.setVisibility(View.GONE);
        attempt.setVisibility(View.GONE);

        b1.setOnClickListener(v -> {
            if(ed1.getText().toString().equals("admin") &&
```

```
            ed2.getText().toString().equals("admin")) {
            Toast.makeText(getApplicationContext(),
                "Welcome Admin !",Toast.LENGTH_SHORT).show();
            Intent intent = new Intent(AdminLogin.this,AdminDashboard.class);
            startActivity(intent);
        }else{
            Toast.makeText(getApplicationContext(), "Wrong
Credentials",Toast.LENGTH_SHORT).show();
            attempt.setVisibility(View.VISIBLE);
            tx1.setVisibility(View.VISIBLE);
            tx1.setBackgroundColor(Color.BLUE);
            counter--;
            tx1.setText(Integer.toString(counter));

            if (counter == 0) {
                b1.setEnabled(false);
            }
        }
    });
  }
}
```

ACTIVITY_ADMIN_LOGIN.XML

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns: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:fitsSystemWindows="true">

  <LinearLayout
```

```
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:layout_alignParentTop="true"
android:layout_marginTop="0dp"
android:background="@drawable/adminbg"
android:gravity="center"
android:orientation="vertical"
android:padding="@dimen/activity_horizontal_margin">

<ImageView
   android:id="@+id/img"
   android:layout_width="match_parent"
   android:layout_height="130dp"
   android:layout_marginTop="0dp"
   android:layout_marginBottom="50dp"
   android:visibility="visible"
   android:src="@drawable/presidencylogo" />

<EditText
   android:id="@+id/editText25"
   style="@android:style/Widget.EditText"
   android:layout_width="200dp"
   android:layout_height="wrap_content"
   android:layout_marginBottom="10dp"
   android:hint="Enter Name"
   android:inputType="text"
   android:textColor="#000000"
   android:textColorHint="#152d4b" />


<EditText
   android:id="@+id/editText27"
   style="@android:style/Widget.EditText"
   android:layout_width="200dp"
   android:layout_height="wrap_content"
   android:layout_marginBottom="10dp"
   android:hint="Enter Password"

   android:inputType="textPassword"
   android:textColor="#000000"
```

```
        android:textColorHint="#152d4b" />

    <TextView
        android:id="@+id/attempt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Attempts Left"
        android:textColor="@color/bg_main"
        android:textSize="24sp" />


    <TextView
        android:id="@+id/textView11"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:text="TextView"
        android:textSize="20sp" />
    <Button
        android:id="@+id/button25"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="20dip"
        android:background="#CC3F2F"
        android:text="@string/btn_login"
        android:textColor="@android:color/black" />

    <!-- Login Button -->
    <!-- Link to Login Screen -->
  </LinearLayout>

</RelativeLayout>
```

ADMIN DASHBOARD.JAVA

```
package com.presidency.resultpro;
```

```
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class AdminDashboard extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_admin_dashboard);
        CardView cg2=findViewById(R.id.c2);
        CardView cg1 = findViewById(R.id.cg1);
        CardView cg3 = findViewById(R.id.cdfeed);
        cg2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent3 = new Intent(AdminDashboard.this,
com.presidency.resultpro.view.studentlistview.class);
                startActivity(intent3);
            }
        });
        cg1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(AdminDashboard.this, admincircular.class);
                startActivity(intent);
            }
        });
        cg3.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(AdminDashboard.this, Feedbackchoice.class);
            startActivity(intent);
        }
    });
  }
}
```

ACTIVITY_ADMIN_DASHBOARD.XML

```
<LinearLayout xmlns: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/admindash"
  android:orientation="vertical">

  <ImageView
    android:id="@+id/imageView15"
    android:layout_width="match_parent"
    android:layout_height="116dp"
    android:layout_marginTop="70dp"
    android:layout_marginBottom="50dp"
    app:srcCompat="@drawable/presidencylogo" />


  <androidx.cardview.widget.CardView
    android:id="@+id/cg1"
    android:layout_width="300dp"
    android:layout_height="70dp"
```

```
    android:layout_gravity="center"
    android:layout_marginBottom="12dp"
    android:contextClickable="true"
    android:rotationY="0"
    android:textAlignment="center"
    app:cardBackgroundColor="@color/input_register_hint"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">
    <TextView
        android:id="@+id/textView7"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="cursive"
        android:text="Send Notifications"
        android:textColor="@color/white"
        android:textSize="38sp"
        android:textStyle="bold" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/c2"
    android:layout_width="300dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginBottom="12dp"
    app:cardBackgroundColor="@color/input_register_hint"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">
```

```
    <TextView
        android:id="@+id/textView8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="cursive"
        android:text="Student list"
        android:textColor="@color/white"
        android:textSize="40sp"
        android:textStyle="bold" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/cdfeed"
    android:layout_width="300dp"
    android:layout_height="70dp"
    android:layout_gravity="center"
    android:layout_marginBottom="12dp"
    android:contextClickable="true"
    android:rotationY="0"
    android:textAlignment="center"
    app:cardBackgroundColor="@color/input_register_hint"
    app:cardCornerRadius="12dp"
    app:cardElevation="2dp">
    <TextView
        android:id="@+id/textViewfeed"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="cursive"
        android:text="Feedbacks"
```

```
        android:textColor="@color/white"
        android:textSize="38sp"
        android:textStyle="bold" />
    </androidx.cardview.widget.CardView>

</LinearLayout>
```

ADMINCIRCULAR.JAVA

```
package com.presidency.resultpro;

import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import java.text.SimpleDateFormat;
import java.util.Date;

public class admincircular extends AppCompatActivity {

    EditText editText1;
    Button submit;
    DatabaseReference rootRef, demoRef;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_updatemarks);
        SimpleDateFormat sdf = new SimpleDateFormat("dd-mm-yy||HH:mm:ss-aa :");
        String currentDateandTime = sdf.format(new Date());
        submit=findViewById(R.id.btnSubmit);
        editText1=findViewById(R.id.etValue);
        // Database reference pointing to root of database
```

```
        rootRef = FirebaseDatabase.getInstance().getReference();
        // Database reference pointing to demo node
        demoRef = rootRef.child("ExcelPro").child("notifications").child(currentDateandTime);
        submit.setOnClickListener(v -> { String value = currentDateandTime+"-
"+editText1.getText().toString();
        // Push creates a unique id in database
            demoRef.setValue(value);
            Toast.makeText(getApplicationContext(),"Circular published
!",Toast.LENGTH_SHORT).show();
        });


    }


}
```

ACTIVITY_ADMINCIRCULAR.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns: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/notify"
    tools:context=".admincircular">

    <EditText
        android:id="@+id/etValue"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="95dp"
        android:layout_marginTop="97dp"
        android:layout_marginEnd="106dp"
        android:ems="10"
```

```
    android:hint="Enter Notification !"
    android:inputType="textPersonName"

    android:textColor="#000000"
    android:textColorHint="@color/bg_main"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />

<androidx.appcompat.widget.AppCompatButton
    android:id="@+id/btnSubmit"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="95dp"
    android:layout_marginTop="103dp"
    android:layout_marginEnd="116dp"
    android:layout_marginBottom="192dp"
    android:background="@color/btn_logut_bg"
    android:fontFamily="serif-monospace"
    android:text="Publish"
    android:textColor="@color/btn_login_bg"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etValue" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="111dp"
    android:layout_marginTop="208dp"
    android:layout_marginEnd="111dp"
    android:fontFamily="serif-monospace"
    android:text="Announce it!"
    android:textColor="@color/bg_main"
    android:textSize="34sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

EXCELUPLOAD.JAVA

```
package com.presidency.resultpro;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.FirebaseDatabase;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.FormulaEvaluator;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.HashMap;

public class ExcelUpload extends AppCompatActivity {
    //initialising the cellcount as 2
```

```
public static final int cellCount=12;
CardView sp,sem2,sem3,sem4,sem5,sem6;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_excel_upload);


    sp = findViewById(R.id.uploadmarklist1);

    sem2 = findViewById(R.id.uploadmarklist2);

    sem3 = findViewById(R.id.uploadmarklist3);

    sem4 = findViewById(R.id.uploadmarklist4);

    sem5 = findViewById(R.id.uploadmarklist5);

    sem6 = findViewById(R.id.uploadmarklist6);

    //click on excel to select a file
    String subjects[] ={"SEMESTER I","SEMESTER II","SEMESTER III","SEMESTER
IV","SEMESTER V","SEMESTER VI"};
    String myString="1";
    Spinner spinnersub=findViewById(R.id.spinnersubject);
    ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this,R.layout.spinnerlist,subjects);
    spinnersub.setAdapter(adapter);
    int spinnerPosition = adapter.getPosition(myString);
    spinnersub.setSelection(spinnerPosition);

    spinnersub.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
        @Override
        public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
            switch (position) {
                case 0:
                    sp.setVisibility(View.VISIBLE);
                    sem2.setVisibility(View.INVISIBLE);
```

```
        sem3.setVisibility(View.INVISIBLE);
        sem4.setVisibility(View.INVISIBLE);
        sem5.setVisibility(View.INVISIBLE);
        sem6.setVisibility(View.INVISIBLE);
        break;
    case 1:
        sp.setVisibility(View.INVISIBLE);
        sem2.setVisibility(View.VISIBLE);
        sem3.setVisibility(View.INVISIBLE);
        sem4.setVisibility(View.INVISIBLE);
        sem5.setVisibility(View.INVISIBLE);
        sem6.setVisibility(View.INVISIBLE);
        break;
    case 2:
        sp.setVisibility(View.INVISIBLE);
        sem2.setVisibility(View.INVISIBLE);
        sem3.setVisibility(View.VISIBLE);
        sem4.setVisibility(View.INVISIBLE);
        sem5.setVisibility(View.INVISIBLE);
        sem6.setVisibility(View.INVISIBLE);
        break;
    case 3:
        sp.setVisibility(View.INVISIBLE);
        sem2.setVisibility(View.INVISIBLE);
        sem3.setVisibility(View.INVISIBLE);
        sem4.setVisibility(View.VISIBLE);
        sem5.setVisibility(View.INVISIBLE);
        sem6.setVisibility(View.INVISIBLE);
        break;
    case 4:
        sp.setVisibility(View.INVISIBLE);
        sem2.setVisibility(View.INVISIBLE);
        sem3.setVisibility(View.INVISIBLE);
        sem4.setVisibility(View.INVISIBLE);
        sem5.setVisibility(View.VISIBLE);
        sem6.setVisibility(View.INVISIBLE);
        break;
    case 5:
        sp.setVisibility(View.INVISIBLE);
```

```
                sem2.setVisibility(View.INVISIBLE);
                sem3.setVisibility(View.INVISIBLE);
                sem4.setVisibility(View.INVISIBLE);
                sem5.setVisibility(View.INVISIBLE);
                sem6.setVisibility(View.VISIBLE);
                break;
            default:
                //Default image
                //image.setImageResource(R.drawable.item2);
                break;

        }
    }

    @Override
    public void onNothingSelected(AdapterView<?> parent) {

    }
});


sp.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(ActivityCompat.checkSelfPermission(ExcelUpload.this ,
Manifest.permission.READ_EXTERNAL_STORAGE)==
PackageManager.PERMISSION_GRANTED){
            selectfile();
        }
        else {
            ActivityCompat.requestPermissions(ExcelUpload.this,new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},101);
        }
    }
});

sem2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
```

```
            if(ActivityCompat.checkSelfPermission(ExcelUpload.this ,
Manifest.permission.READ_EXTERNAL_STORAGE)==
PackageManager.PERMISSION_GRANTED){
                selectfile();
            }
            else {
                ActivityCompat.requestPermissions(ExcelUpload.this,new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},102);
            }
        }
    });
  }
  //request for storage permission if not given
  @Override
  public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    if (requestCode == 101) {
      if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        selectfile();
      } else {
        Toast.makeText(ExcelUpload.this, "Permission Not granted",
Toast.LENGTH_LONG).show();
      }
    } else if (requestCode == 102) {
      if (grantResults[0] == PackageManager.PERMISSION_GRANTED) {
        selectfile();
      } else {
        Toast.makeText(ExcelUpload.this, "Permission Not granted",
Toast.LENGTH_LONG).show();
      }
    }
  }
  private void selectfile(){
    //select the file from the file storage
    Intent intent=new Intent(Intent.ACTION_OPEN_DOCUMENT);
    intent.setType("*/*");
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    startActivityForResult(Intent.createChooser(intent, "Select File"),102);
```

```
}
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 101) {
        if (resultCode == RESULT_OK) {
            String filepath = data.getData().getPath();
            //If excel file then only select the file
            if (filepath.endsWith(".xlsx") || filepath.endsWith(".xls")) {
                readfile(data.getData());
            }
            //else show the error
            else {
                Toast.makeText(this, "Please Select an Excel file to upload",
Toast.LENGTH_LONG).show();
            }
        }
    } else  if (requestCode == 102) {
        if (resultCode == RESULT_OK) {
            String filepath = data.getData().getPath();
            //If excel file then only select the file
            if (filepath.endsWith(".xlsx") || filepath.endsWith(".xls")) {
                readfilesem2(data.getData());
            }
            //else show the error
            else {
                Toast.makeText(this, "Please Select an Excel file to upload",
Toast.LENGTH_LONG).show();
            }
        }
    }
}

ProgressDialog dialog;
private void readfile(final Uri file)
{
    dialog=new ProgressDialog(this);
    dialog.setMessage("Uploading");
    dialog.setCanceledOnTouchOutside(false);
    dialog.show();
```

```
    AsyncTask.execute(new Runnable() {
        @Override
        public void run() {
            final HashMap<String ,Object> parentmap=new HashMap<>();

            try {
                XSSFWorkbook workbook;
                //check for the input from the excel file
                try (InputStream inputStream = getContentResolver().openInputStream(file)) {
                    workbook = new XSSFWorkbook(inputStream);
                }
                final String timestamp=""+System.currentTimeMillis();
                XSSFSheet sheet=workbook.getSheetAt(0);
                FormulaEvaluator
formulaEvaluator=workbook.getCreationHelper().createFormulaEvaluator();
                int rowscount=sheet.getPhysicalNumberOfRows();
                if(rowscount>0){
                    //check rowwise data
                    for (int r=0;r<rowscount;r++){
                        Row row=sheet.getRow(r);
                        if(row.getPhysicalNumberOfCells()==cellCount) {
                            //get cell data
                            String A = getCellData(row,0,formulaEvaluator);
                            String B = getCellData(row,1,formulaEvaluator);
                            String C = getCellData(row,2,formulaEvaluator);
                            String D = getCellData(row,3,formulaEvaluator);
                            String E = getCellData(row,4,formulaEvaluator);
                            String F = getCellData(row,5,formulaEvaluator);
                            String G = getCellData(row,6,formulaEvaluator);
                            String H = getCellData(row,7,formulaEvaluator);
                            String I = getCellData(row,8,formulaEvaluator);
                            String J = getCellData(row,9,formulaEvaluator);
                            String K = getCellData(row,10,formulaEvaluator);
                            String L = getCellData(row,11,formulaEvaluator);

                            //initialise the hashmap and put value of a and b into it
                            HashMap<String,Object> quetionmap=new HashMap<>();
                            quetionmap.put("SERIAL NUMBER",A);
                            quetionmap.put("registernumber",B);
```

```
                    quetionmap.put("name",C);
                    quetionmap.put("INDIAN LANGUAGE",D);
                    quetionmap.put("ENGLISH",E);
                    quetionmap.put("C LANGUAGE",F);
                    quetionmap.put("COMPUTER ORGANISATION",G);
                    quetionmap.put("DM",H);
                    quetionmap.put("C LAB",I);
                    quetionmap.put("OFFICE AUTOMATION",J);
                    quetionmap.put("FOUNDATION COURSE",K);
                    quetionmap.put("CCANDEC",L);
                    String id=B;
                    parentmap.put(id,quetionmap);
                }
                else {
                    dialog.dismiss();
                    Toast.makeText(ExcelUpload.this,"row no. "+(r+1)+" has incorrect
data",Toast.LENGTH_LONG).show();
                    return;
                }
            }
            //add the data in firebase if everything is correct
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    //add the data accourding to timestamp

FirebaseDatabase.getInstance().getReference().child("FirstSem3").updateChildren(parentmap
).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if(task.isSuccessful()){
                                dialog.dismiss();
                                Toast.makeText(ExcelUpload.this,"Uploaded
Successfully",Toast.LENGTH_LONG).show();
                            }else {
                                dialog.dismiss();
                                Toast.makeText(ExcelUpload.this,"Something went
wrong",Toast.LENGTH_LONG).show();
                            }
```

```
                    }
                });
            }
        });

    }
    //show the error if file is empty
    else {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                dialog.dismiss();
                Toast.makeText(ExcelUpload.this,"File is
empty",Toast.LENGTH_LONG).show();

            }
        });
        return;
    }
}
//show the error message if failed due to file not found
catch (final FileNotFoundException e){
    e.printStackTrace();
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(ExcelUpload.this,e.getMessage(),
Toast.LENGTH_LONG).show();
        }
    });
}
//show the error message if there is error in input outut
catch (final IOException e){
    e.printStackTrace();
    runOnUiThread(new Runnable() {
        @Override
        public void run() {

            Toast.makeText(ExcelUpload.this,e.getMessage(),
```

```
Toast.LENGTH_LONG).show();
                    }
                });
            }
        }
    });
  }

  private void readfilesem2(final Uri file)
  {
    dialog=new ProgressDialog(this);
    dialog.setMessage("Uploading");
    dialog.setCanceledOnTouchOutside(false);
    dialog.show();
    AsyncTask.execute(new Runnable() {
      @Override
      public void run() {

        final HashMap<String ,Object> parentmap=new HashMap<>();

        try {
          XSSFWorkbook workbook;
          //check for the input from the excel file
          try (InputStream inputStream = getContentResolver().openInputStream(file)) {
            workbook = new XSSFWorkbook(inputStream);
          }
          final String timestamp=""+System.currentTimeMillis();
          XSSFSheet sheet=workbook.getSheetAt(0);
          FormulaEvaluator
formulaEvaluator=workbook.getCreationHelper().createFormulaEvaluator();
          int rowscount=sheet.getPhysicalNumberOfRows();
          if(rowscount>0){
            //check rowwise data
            for (int r=0;r<rowscount;r++){
              Row row=sheet.getRow(r);
              if(row.getPhysicalNumberOfCells()==cellCount) {
                //get cell data
                String A = getCellData(row,0,formulaEvaluator);
                String B = getCellData(row,1,formulaEvaluator);
```

```
                    String C = getCellData(row,2,formulaEvaluator);
                    String D = getCellData(row,3,formulaEvaluator);
                    String E = getCellData(row,4,formulaEvaluator);
                    String F = getCellData(row,5,formulaEvaluator);
                    String G = getCellData(row,6,formulaEvaluator);
                    String H = getCellData(row,7,formulaEvaluator);
                    String I = getCellData(row,8,formulaEvaluator);
                    String J = getCellData(row,9,formulaEvaluator);
                    String K = getCellData(row,10,formulaEvaluator);
                    String L = getCellData(row,11,formulaEvaluator);

                    //initialise the hashmap and put value of a and b into it
                    HashMap<String,Object> quetionmap=new HashMap<>();
                    quetionmap.put("SERIAL NUMBER",A);
                    quetionmap.put("registernumber",B);
                    quetionmap.put("name",C);
                    quetionmap.put("INDIAN LANGUAGE",D);
                    quetionmap.put("ENGLISH",E);
                    quetionmap.put("C LANGUAGE",F);
                    quetionmap.put("COMPUTER ORGANISATION",G);
                    quetionmap.put("DM",H);
                    quetionmap.put("C LAB",I);
                    quetionmap.put("OFFICE AUTOMATION",J);
                    quetionmap.put("FOUNDATION COURSE",K);
                    quetionmap.put("CCANDEC",L);

                    String id=B;
                    parentmap.put(id,quetionmap);
                }
                else {
                    dialog.dismiss();
                    Toast.makeText(ExcelUpload.this,"row no. "+(r+1)+" has incorrect
data",Toast.LENGTH_LONG).show();
                    return;
                }
            }
            //add the data in firebase if everything is correct
            runOnUiThread(new Runnable() {
                @Override
```

```
                public void run() {
                    //add the data accourding to timestamp

FirebaseDatabase.getInstance().getReference().child("FirstSem3").updateChildren(parentmap
).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if(task.isSuccessful()){
                                dialog.dismiss();
                                Toast.makeText(ExcelUpload.this,"Uploaded
Successfully",Toast.LENGTH_LONG).show();
                            }else {
                                dialog.dismiss();
                                Toast.makeText(ExcelUpload.this,"Something went
wrong",Toast.LENGTH_LONG).show();
                            }
                        }
                    });
                }
            });

        }
        //show the error if file is empty
        else {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    dialog.dismiss();
                    Toast.makeText(ExcelUpload.this,"File is
empty",Toast.LENGTH_LONG).show();

                }
            });
            return;
        }
    }
    //show the error message if failed due to file not found
    catch (final FileNotFoundException e){
        e.printStackTrace();
```

```
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        Toast.makeText(ExcelUpload.this,e.getMessage(),
Toast.LENGTH_LONG).show();
                    }
                });
            }
            //show the error message if there is error in input outut
            catch (final IOException e){
                e.printStackTrace();
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {

                        Toast.makeText(ExcelUpload.this,e.getMessage(),
Toast.LENGTH_LONG).show();
                    }
                });
            }
        }
    });
  }
  private String getCellData(Row row, int cellposition, FormulaEvaluator
formulaEvaluator){
    String value="";
    //get cell fom excel sheet
    Cell cell=row.getCell(cellposition);
    switch (cell.getCellType()){

        case Cell.CELL_TYPE_BOOLEAN:
            return value+cell.getBooleanCellValue();
        case Cell.CELL_TYPE_NUMERIC:
            return value+cell.getNumericCellValue();
        case Cell.CELL_TYPE_STRING:
            return value+cell.getStringCellValue();
        default:
            return value;
    }
```

```
    }
}
```

ACTIVITY_EXCEL_UPLOAD.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns: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/marksup"
    android:orientation="vertical"
    app:layoutDescription="@xml/activity_excel_sp_scene">

    <TextView
        android:id="@+id/heading"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:background="@color/lbl_name"
        android:fontFamily="cursive"
        android:gravity="center"
        android:padding="15dip"
        android:text="Upload Marklist"
        android:textColor="@color/follow_button_light_color"
        android:textSize="34sp"
        android:textStyle="bold" />


    <Spinner
        android:id="@+id/spinnersubject"
        android:layout_width="300dp"
        android:layout_height="60dp"
        android:layout_marginStart="55dp"
        android:layout_marginTop="100dp"
        android:layout_marginEnd="56dp"
        android:background="@drawable/spinnerbg"
        android:clickable="true"
        android:forceDarkAllowed="false"
        android:gravity="center"
```

```
    android:spinnerMode="dialog"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/heading" />

<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist6"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
    <ImageView
        android:id="@+id/imageView182"
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist1"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
    <ImageView
        android:id="@+id/imageView18"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist2"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
    <ImageView
        android:id="@+id/imageView23"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist3"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
    <ImageView
        android:id="@+id/imageView183"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist4"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>
<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist5"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
```

```
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
    <ImageView
      android:id="@+id/imageView185"
      android:layout_width="100dp"
      android:layout_height="100dp"
      android:layout_gravity="center_horizontal"
      android:layout_marginTop="20dp"
      app:srcCompat="@drawable/upload" />

  </androidx.cardview.widget.CardView>


</androidx.constraintlayout.widget.ConstraintLayout>
```

FACULTYDASHBOARD.JAVA

```
package com.presidency.resultpro;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
public class FacultyDashboard extends AppCompatActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_faculty_dashboard);
    CardView marklist,cardfeed,cardnotify,cggraph,subcard2,events;
    marklist=findViewById(R.id.cdmarks);
    cardfeed=findViewById(R.id.cardfeed);
    cardnotify=findViewById(R.id.cardnotify);
    events=findViewById(R.id.attendancecardfaculty);
    cggraph=findViewById(R.id.cggraph);
    subcard2=findViewById(R.id.subcardupload);
```

```
marklist.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,choosedept.class);
        startActivity(intent);
    }
});
cardfeed.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,FeedbackTeachers.class);
        startActivity(intent);
    }
});
cardnotify.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,teachercircular.class);
        startActivity(intent);
    }
});
cggraph.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,graphDash.class);
        startActivity(intent);
    }
});
subcard2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,resultcard.class);
        startActivity(intent);
    }
});
events.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,EventsActivity.class);
```

```
            startActivity(intent);
        }
    });
  }
}
```

ACTIVITY_EXCEL_UPLOAD.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns: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/marksup"
   android:orientation="vertical"
   app:layoutDescription="@xml/activity_excel_sp_scene">

   <TextView
      android:id="@+id/heading"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:background="@color/lbl_name"
      android:fontFamily="cursive"
      android:gravity="center"
      android:padding="15dip"
      android:text="Upload Marklist"
      android:textColor="@color/follow_button_light_color"
      android:textSize="34sp"
      android:textStyle="bold" />
   <Spinner
      android:id="@+id/spinnersubject"
      android:layout_width="300dp"
      android:layout_height="60dp"
      android:layout_marginStart="55dp"
      android:layout_marginTop="100dp"
      android:layout_marginEnd="56dp"
      android:background="@drawable/spinnerbg"
      android:clickable="true"
```

```
    android:forceDarkAllowed="false"
    android:gravity="center"
    android:spinnerMode="dialog"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/heading" />

<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist6"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">

    <ImageView
        android:id="@+id/imageView182"
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist1"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
```

```
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">

    <ImageView
        android:id="@+id/imageView18"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist2"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">

    <ImageView
        android:id="@+id/imageView23"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist3"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">

    <ImageView
        android:id="@+id/imageView183"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist4"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">
```

```
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />
</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/uploadmarklist5"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginStart="55dp"
    android:layout_marginTop="60dp"
    android:layout_marginEnd="56dp"
    android:layout_marginBottom="36dp"
    app:cardCornerRadius="20dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/spinnersubject">

    <ImageView
        android:id="@+id/imageView185"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="20dp"
        app:srcCompat="@drawable/upload" />

</androidx.cardview.widget.CardView>


</androidx.constraintlayout.widget.ConstraintLayout>
```

ADDSTUDENTS.JAVA

```
package com.presidency.resultpro.view;
```

```
import android.content.ContentResolver;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.webkit.MimeTypeMap;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.Continuation;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.google.firebase.storage.StorageTask;
import com.google.firebase.storage.UploadTask;
import com.presidency.resultpro.Model.Teacher;
import com.presidency.resultpro.R;
import com.squareup.picasso.Picasso;

public class AddStudents extends AppCompatActivity {

    private static final int PICK_IMAGE_REQUEST = 1;
    private Button chooseImageBtn;
    private Button uploadBtn;
    private EditText nameEditText;
    private EditText descriptionEditText;
    private ImageView chosenImageView;
    private ProgressBar uploadProgressBar;
     public static Uri mImageUri;
    private StorageReference mStorageRef;
    private DatabaseReference mDatabaseRef;
```

```
private StorageTask mUploadTask;
StorageReference fileReference;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add_students);

    chooseImageBtn = findViewById(R.id.button_choose_image);
    uploadBtn = findViewById(R.id.uploadBtn);
    nameEditText = findViewById(R.id.nameEditText);
    descriptionEditText = findViewById(R.id.descptxt);
    chosenImageView = findViewById(R.id.chosenImageView);
    uploadProgressBar = findViewById(R.id.progress_bar);

    mStorageRef = FirebaseStorage.getInstance().getReference("Student_uploads");
    mDatabaseRef = FirebaseDatabase.getInstance().getReference("credentials");

    chooseImageBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            openFileChooser();
        }
    });

    uploadBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mUploadTask != null && mUploadTask.isInProgress()) {
                Toast.makeText(AddStudents.this, "An Upload is Still in Progress",
Toast.LENGTH_SHORT).show();
            } else {
                uploadFile();
            }
        }
    });
}

private void openFileChooser() {
```

```
    Intent intent = new Intent();
    intent.setType("image/*");
    intent.setAction(Intent.ACTION_GET_CONTENT);
    startActivityForResult(intent, PICK_IMAGE_REQUEST);
  }

  @Override
  protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == PICK_IMAGE_REQUEST && resultCode == RESULT_OK
        && data != null && data.getData() != null) {
      mImageUri = data.getData();

      Picasso.get().load(mImageUri).into(chosenImageView);
    }
  }

  private String getFileExtension(Uri uri) {
    ContentResolver cR = getContentResolver();
    MimeTypeMap mime = MimeTypeMap.getSingleton();
    return mime.getExtensionFromMimeType(cR.getType(uri));
  }

  private void uploadFile() {
    if (mImageUri != null) {
      fileReference = mStorageRef.child(nameEditText+ "." +
getFileExtension(mImageUri));


      uploadProgressBar.setVisibility(View.VISIBLE);
      uploadProgressBar.setIndeterminate(true);

      mUploadTask = fileReference.putFile(mImageUri);
      mUploadTask.continueWithTask(new Continuation<UploadTask.TaskSnapshot,
Task<Uri>>() {
        @Override
        public Task<Uri> then(@NonNull Task<UploadTask.TaskSnapshot> task) throws
Exception {
```

```
            if (!task.isSuccessful()) {
                throw task.getException();
            }

            return fileReference.getDownloadUrl();
        }
    }).addOnCompleteListener(new OnCompleteListener<Uri>() {
        @Override
        public void onComplete(@NonNull Task<Uri> task) {
            if (task.isSuccessful()) {
                Uri taskResult = task.getResult();
                Teacher upload = new Teacher(nameEditText.getText().toString().trim(),
                        taskResult.toString(),
                        descriptionEditText.getText().toString());
                String uploadId = nameEditText.getText().toString();
                mDatabaseRef.child(uploadId).setValue(upload);

                 uploadProgressBar.setVisibility(View.INVISIBLE);
                openImagesActivity();
            }
            }

    })

            .addOnFailureListener(new OnFailureListener() {
                @Override
                public void onFailure(@NonNull Exception e) {
                    uploadProgressBar.setVisibility(View.INVISIBLE);
                    Toast.makeText(AddStudents.this, e.getMessage(),
Toast.LENGTH_SHORT).show();
                }
            });


    } else {
        Toast.makeText(this, "You haven't selected any file!",
Toast.LENGTH_SHORT).show();
    }
```

```
    }

    private void openImagesActivity() {
        Intent intent = new Intent(this, studentlistview.class);
        startActivity(intent);
    }
}
```

ACTIVITY_ADD_STUDENTS.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns: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/addstudentsbg"
    android:orientation="vertical">

    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentStart="true"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:orientation="vertical">
            <ProgressBar
                android:id="@+id/progress_bar"
                style="@style/Widget.AppCompat.ProgressBar.Horizontal"
                android:layout_width="match_parent"
                android:layout_height="52dp" />
            <TextView
                android:id="@+id/headerTextView"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="Enter Student Details"
```

```
        android:textSize="34sp"
        android:textStyle="bold" />

    <EditText
        android:id="@+id/nameit"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:hint="Name"
        android:inputType="text"
        android:textColor="#ffffff"
        android:textColorHint="#ffffff" />
    <EditText
        android:id="@+id/nameEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Register Number"
        android:inputType="text"
        android:minLines="3"
        android:textColor="#ffffff"
        android:textColorHint="#ffffff" />


    <EditText
        android:id="@+id/descptxt"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Password"
        android:inputType="textPersonName"
        android:textColor="#ffffff"
        android:textColorHint="#ffffff" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">
        <ImageView
            android:id="@+id/chosenImageView"
            android:layout_width="match_parent"
            android:layout_height="180dp"
```

```
            android:layout_marginTop="40dp"
            android:layout_marginBottom="50dp" />
        <Button
            android:id="@+id/button_choose_image"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="CHOOSE IMAGE" />
    </LinearLayout>
    <Button
        android:id="@+id/uploadBtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="center|center_horizontal"
        android:background="@color/input_login"
        android:gravity="center"
        android:text="Upload"
        android:textColor="@color/white" />
  </LinearLayout>

 </ScrollView>

</LinearLayout>
```

VIEWALLSTUDENTS.JAVA

```
package com.presidency.resultpro.view;

import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.ProgressBar;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
```

```
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
import com.presidency.resultpro.Adapter.RecyclerAdapter;
import com.presidency.resultpro.Model.Teacher;
import com.presidency.resultpro.R;
import java.util.ArrayList;
import java.util.List;

public class viewallstudents extends AppCompatActivity implements
RecyclerAdapter.OnItemClickListener{

    private RecyclerView mRecyclerView;
    private RecyclerAdapter mAdapter;
    private ProgressBar mProgressBar;
    private FirebaseStorage mStorage;
    private DatabaseReference mDatabaseRef;
    private ValueEventListener mDBListener;
    private List<Teacher> mTeachers;
    public static Uri imageuri;


    private void openDetailActivity(String[] data){
        Intent intent = new Intent(this, DetailsActivity.class);
        intent.putExtra("NAME_KEY",data[0]);
        intent.putExtra("DESCRIPTION_KEY",data[1]);
        intent.putExtra("IMAGE_KEY",data[2]);
        startActivity(intent);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate ( savedInstanceState );
        setContentView ( R.layout.activity_items );
        mRecyclerView = findViewById(R.id.mRecyclerView);
        mRecyclerView.setHasFixedSize(true);
```

```
        mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
        mProgressBar = findViewById(R.id.myDataLoaderProgressBar);
        mProgressBar.setVisibility(View.VISIBLE);
        mTeachers = new ArrayList<> ();
        mAdapter = new RecyclerAdapter (viewallstudents.this, mTeachers);
        mRecyclerView.setAdapter(mAdapter);
        mAdapter.setOnItemClickListener(viewallstudents.this);
        mStorage = FirebaseStorage.getInstance();
        mDatabaseRef = FirebaseDatabase.getInstance().getReference("credentials");

        mDBListener = mDatabaseRef.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(DataSnapshot dataSnapshot) {

                mTeachers.clear();

                for (DataSnapshot teacherSnapshot : dataSnapshot.getChildren()) {
                    Teacher upload = teacherSnapshot.getValue(Teacher.class);
                    upload.setKey(teacherSnapshot.getKey());
                    mTeachers.add(upload);
                }
                mAdapter.notifyDataSetChanged();
                mProgressBar.setVisibility(View.GONE);
            }

            @Override
            public void onCancelled(DatabaseError databaseError) {
                Toast.makeText(viewallstudents.this, databaseError.getMessage(),
Toast.LENGTH_SHORT).show();
                mProgressBar.setVisibility(View.INVISIBLE);
            }
        });

    }
    public void onItemClick(int position) {
        Teacher clickedTeacher=mTeachers.get(position);
        String[]
teacherData={clickedTeacher.getRegisternumber(),clickedTeacher.getPassword(),clickedTea
cher.getImageUrl()};
```

```
        openDetailActivity(teacherData);
    }

    @Override
    public void onShowItemClick(int position) {
        Teacher clickedTeacher=mTeachers.get(position);
        String[]
teacherData={clickedTeacher.getRegisternumber(),clickedTeacher.getPassword(),clickedTea
cher.getImageUrl()};
        openDetailActivity(teacherData);
    }

    @Override
    public void onDeleteItemClick(int position) {
        Teacher selectedItem = mTeachers.get(position);
        final String selectedKey = selectedItem.getKey();

        StorageReference imageRef =
mStorage.getReferenceFromUrl(selectedItem.getImageUrl());


        imageRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
            @Override
            public void onSuccess(Uri uri) {
            imageuri=uri;
            }
        });

        imageRef.delete().addOnSuccessListener(new OnSuccessListener<Void> () {
            @Override
            public void onSuccess(Void aVoid) {
                mDatabaseRef.child(selectedKey).removeValue();
                Toast.makeText(viewallstudents.this, "Item deleted",
Toast.LENGTH_SHORT).show();
            }
        });

    }
    protected void onDestroy() {
```

```
        super.onDestroy();
        mDatabaseRef.removeEventListener(mDBListener);
    }

}
```

ACTIVITY_STUDENTLIST.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns: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:orientation="vertical"
    tools:context=".view.viewallstudents">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Student List"
        android:textAlignment="center"
        android:textAppearance="@style/TextAppearance.AppCompat.Large"
        android:textColor="#ffffff"
        android:textStyle="bold" />

    <ProgressBar
        android:id="@+id/myDataLoaderProgressBar"
        style="?android:attr/progressBarStyleHorizontal"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:indeterminate="true"
        android:indeterminateBehavior="cycle"
        android:visibility="gone" />


    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/mRecyclerView"
```

```
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="0.5" />

</LinearLayout>
```

UPLOADCRED.JAVA

```
package com.presidency.resultpro;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import androidx.core.app.ActivityCompat;
import android.Manifest;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.database.FirebaseDatabase;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.FormulaEvaluator;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.xssf.usermodel.XSSFSheet;
import org.apache.poi.xssf.usermodel.XSSFWorkbook;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.InputStream;
import java.util.HashMap;
```

```
public class uploadcred extends AppCompatActivity {
    //initialising the cellcount as 2
    public static final int cellCount=2;
    CardView upcred;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_uploadcred);
        upcred = findViewById(R.id.cred);


        //click on excel to select a file
        String subjects[] ={"SEMESTER I","SEMESTER II","SEMESTER III","SEMESTER
IV","SEMESTER V","SEMESTER VI"};
        upcred.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if(ActivityCompat.checkSelfPermission(uploadcred.this ,
Manifest.permission.READ_EXTERNAL_STORAGE)==
PackageManager.PERMISSION_GRANTED){
                    selectfile();
                }
                else {
                    ActivityCompat.requestPermissions(uploadcred.this,new
String[]{Manifest.permission.READ_EXTERNAL_STORAGE},101);
                }
            }
        });

    }
    //request for storage permission if not given
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions,
@NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        if(requestCode==101){
            if(grantResults[0]==PackageManager.PERMISSION_GRANTED){
                selectfile();
```

```
        }else {
            Toast.makeText(uploadcred.this,"Permission Not
granted",Toast.LENGTH_LONG).show();
        }
      }
   }
   private void selectfile(){
      //select the file from the file storage
      Intent intent=new Intent(Intent.ACTION_OPEN_DOCUMENT);
      intent.setType("*/*");
      intent.addCategory(Intent.CATEGORY_OPENABLE);
      startActivityForResult(Intent.createChooser(intent, "Select File"),102);
   }

   protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
      super.onActivityResult(requestCode, resultCode, data);
      if(requestCode==102){
         if(resultCode==RESULT_OK){
            String filepath=data.getData().getPath();
            //If excel file then only select the file
            if(filepath.endsWith(".xlsx") || filepath.endsWith(".xls")){
               readfile(data.getData());
            }
            //else show the error
            else {
               Toast.makeText(this,"Please Select an Excel file to
upload",Toast.LENGTH_LONG).show();
            }
         }
      }
   }

   ProgressDialog dialog;
   private void readfile(final Uri file)
   {
      dialog=new ProgressDialog(this);
      dialog.setMessage("Uploading");
      dialog.setCanceledOnTouchOutside(false);
      dialog.show();
```

```
    AsyncTask.execute(new Runnable() {
        @Override
        public void run() {

            final HashMap<String ,Object> parentmap=new HashMap<>();

            try {
                XSSFWorkbook workbook;
                //check for the input from the excel file
                try (InputStream inputStream = getContentResolver().openInputStream(file)) {
                    workbook = new XSSFWorkbook(inputStream);
                }
                final String timestamp=""+System.currentTimeMillis();
                XSSFSheet sheet=workbook.getSheetAt(0);
                FormulaEvaluator
formulaEvaluator=workbook.getCreationHelper().createFormulaEvaluator();
                int rowscount=sheet.getPhysicalNumberOfRows();
                if(rowscount>0){
                    //check rowwise data
                    for (int r=0;r<rowscount;r++){
                        Row row=sheet.getRow(r);
                        if(row.getPhysicalNumberOfCells()==cellCount) {
                            //get cell data
                            String A = getCellData(row,0,formulaEvaluator);
                            String B = getCellData(row,1,formulaEvaluator);
                            //initialise the hashmap and put value of a and b into it
                            HashMap<String,Object> quetionmap=new HashMap<>();
                            quetionmap.put("registernumber",A);
                            quetionmap.put("password",B);

                            String id=A;


                            parentmap.put(id,quetionmap);
                        }
                        else {
                            dialog.dismiss();
                            Toast.makeText(uploadcred.this,"row no. "+(r+1)+" has incorrect
data",Toast.LENGTH_LONG).show();
```

```
                    return;
                }
            }
            //add the data in firebase if everything is correct
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    //add the data accourding to timestamp

FirebaseDatabase.getInstance().getReference().child("credentials").updateChildren(parentma
p).addOnCompleteListener(new OnCompleteListener<Void>() {
                        @Override
                        public void onComplete(@NonNull Task<Void> task) {
                            if(task.isSuccessful()){
                                dialog.dismiss();
                                Toast.makeText(uploadcred.this,"Credentials Uploaded
Successfully default pwd : 123",Toast.LENGTH_LONG).show();
                            }else {
                                dialog.dismiss();
                                Toast.makeText(uploadcred.this,"Something went
wrong",Toast.LENGTH_LONG).show();
                            }
                        }
                    });
                }
            });

        }
        //show the error if file is empty
        else {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    dialog.dismiss();
                    Toast.makeText(uploadcred.this,"File is
empty",Toast.LENGTH_LONG).show();

                }
            });
```

```
                return;
            }
        }
        //show the error message if failed due to file not found
        catch (final FileNotFoundException e){
            e.printStackTrace();
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    Toast.makeText(uploadcred.this,e.getMessage(),
Toast.LENGTH_LONG).show();
                }
            });
        }
        //show the error message if there is error in input outut
        catch (final IOException e){
            e.printStackTrace();
            runOnUiThread(new Runnable() {
                @Override
                public void run() {

                    Toast.makeText(uploadcred.this,e.getMessage(),
Toast.LENGTH_LONG).show();
                }
            });
        }
      }
    });
  }
  private String getCellData(Row row, int cellposition, FormulaEvaluator
formulaEvaluator){
    String value="";
    //get cell fom excel sheet
    Cell cell=row.getCell(cellposition);
    switch (cell.getCellType()){

      case Cell.CELL_TYPE_BOOLEAN:
        return value+cell.getBooleanCellValue();
      case Cell.CELL_TYPE_NUMERIC:
```

```
            return value+cell.getNumericCellValue();
        case Cell.CELL_TYPE_STRING:
            return value+cell.getStringCellValue();
        default:
            return value;
    }
  }

}
```

ACTIVITY_UPLOADCRED.XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns: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/bgfacultynotify"
   tools:context=".uploadcred">

   <androidx.cardview.widget.CardView
      android:id="@+id/cred"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginStart="129dp"
      android:layout_marginTop="292dp"
      android:layout_marginEnd="130dp"
      android:layout_marginBottom="292dp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.0"
      tools:srcCompat="@drawable/adminmenu">
      <ImageView
         android:id="@+id/imageView20"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
```

```
        app:srcCompat="@drawable/adminmenu" />
    </androidx.cardview.widget.CardView>
    <TextView
        android:id="@+id/textView12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="132dp"
        android:layout_marginTop="225dp"
        android:layout_marginEnd="132dp"
        android:layout_marginBottom="48dp"
        android:text="UPLOAD CREDENTIALS"
        android:textColor="#000000"
        android:textSize="34sp"
        app:layout_constraintBottom_toTopOf="@+id/cred"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

FACULTYDASHBOARD.JAVA

```
package com.presidency.resultpro;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;

public class FacultyDashboard extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_faculty_dashboard);
        CardView marklist,cardfeed,cardnotify,cggraph,subcard2,events;
        marklist=findViewById(R.id.cdmarks);
        cardfeed=findViewById(R.id.cardfeed);
```

```
cardnotify=findViewById(R.id.cardnotify);
events=findViewById(R.id.attendancecardfaculty);
cggraph=findViewById(R.id.cggraph);
subcard2=findViewById(R.id.subcardupload);
marklist.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,choosedept.class);
        startActivity(intent);
    }
});

cardfeed.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,FeedbackTeachers.class);
        startActivity(intent);
    }
});

cardnotify.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,teachercircular.class);
        startActivity(intent);
    }
});

cggraph.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,graphDash.class);
        startActivity(intent);
    }
});
subcard2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent intent = new Intent(FacultyDashboard.this,resultcard.class);
```

```
            startActivity(intent);
        }
    });
    events.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent(FacultyDashboard.this,EventsActivity.class);
            startActivity(intent);
        }
    });
  }
}
```

ACTIVITY_FACULTY_DASHBOARD.XML

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns: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/fnotification"
    android:orientation="vertical"
    android:weightSum="10">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:layout_marginBottom="10dp"
        android:layout_weight="2"
        android:background="@android:color/transparent">

        <ImageView
            android:layout_width="300dp"
            android:layout_height="400dp"
            android:layout_gravity="center_horizontal"

            android:layout_marginLeft="50dp"
            android:layout_marginTop="30dp"
            android:background="@android:color/transparent"
```

```
        android:foregroundGravity="clip_horizontal|center"
        android:src="@drawable/presidencylogo" />

</RelativeLayout>

<GridLayout
    android:id="@+id/mainGrid"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="7"
    android:alignmentMode="alignMargins"
    android:columnCount="2"
    android:columnOrderPreserved="false"
    android:padding="20dp"
    android:rowCount="3">

    <!-- ROW 1  -->
    <!-- COLUMN 1  -->
    <androidx.cardview.widget.CardView
        android:id="@+id/cggraph"
        android:layout_width="0dp"
        android:layout_height="30dp"
        android:layout_rowWeight="1"
        android:layout_columnWeight="1"
        android:layout_marginLeft="10dp"
        android:layout_marginRight="10dp"
        android:layout_marginBottom="37dp"
        android:background="#4F6E9C"
        app:cardBackgroundColor="#3A3535"
        app:cardCornerRadius="8dp"
        app:cardElevation="8dp">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal|center_vertical"
            android:layout_margin="16dp"
            android:orientation="vertical">
            <ImageView
                android:id="@+id/imageView7"
```

```
            android:layout_width="70dp"
            android:layout_height="70dp"
            app:srcCompat="@drawable/graph2" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Graphs"
            android:textAlignment="center"
            android:textColor="#ffffff"
            android:textSize="12sp"
            android:textStyle="bold" />
    </LinearLayout>
</androidx.cardview.widget.CardView>
<!-- COLUMN 2  -->
<androidx.cardview.widget.CardView
    android:id="@+id/cdmarks"
    android:layout_width="0dp"
    android:layout_height="30dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="37dp"
    android:background="#4F6E9C"
    app:cardBackgroundColor="#3A3535"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp">
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center|center_horizontal|center_vertical"
        android:layout_margin="16dp"
        android:orientation="vertical">
        <ImageView
            android:id="@+id/imageView13"
            android:layout_width="70dp"
            android:layout_height="70dp"
            app:srcCompat="@drawable/marks2" />
```

```
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Marklist"
            android:textAlignment="center"
            android:textColor="#ffffff"
            android:textSize="12sp"
            android:textStyle="bold" />
    </LinearLayout>

</androidx.cardview.widget.CardView>


<!-- ROW 2  -->
<!-- COLUMN 1  -->

<androidx.cardview.widget.CardView
    android:id="@+id/attendancecardfaculty"
    android:layout_width="0dp"
    android:layout_height="30dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="37dp"
    android:background="#4F6E9C"
    app:cardBackgroundColor="#3A3535"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal|center_vertical"
        android:layout_margin="16dp"
        android:orientation="vertical">

        <ImageView
```

```
            android:layout_width="70dp"
            android:layout_height="70dp"
            android:layout_gravity="center_horizontal"
            android:src="@drawable/events"
            app:srcCompat="@drawable/events2" />

        <TextView
            android:layout_width="84dp"
            android:layout_height="29dp"
            android:layout_gravity="center"
            android:text="Events"
            android:textAlignment="center"
            android:textColor="#ffffff"
            android:textSize="12sp"
            android:textStyle="bold" />

    </LinearLayout>

</androidx.cardview.widget.CardView>

<!-- COLUMN 2  -->
<androidx.cardview.widget.CardView
    android:id="@+id/subcardupload"
    android:layout_width="0dp"
    android:layout_height="30dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="37dp"
    android:background="#4F6E9C"
    app:cardBackgroundColor="#3A3535"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal|center_vertical"
```

```
        android:layout_margin="16dp"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView9"
            android:layout_width="68dp"
            android:layout_height="68dp"
            android:background="@drawable/book2"
            app:srcCompat="@drawable/book2" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Internals"
            android:textAlignment="center"
            android:textColor="#ffffff"
            android:textSize="12sp"
            android:textStyle="bold" />
    </LinearLayout>
</androidx.cardview.widget.CardView>

<!-- ROW 3  -->
<!-- COLUMN 1  -->
<androidx.cardview.widget.CardView
    android:id="@+id/cardfeed"
    android:layout_width="0dp"
    android:layout_height="30dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="37dp"
    android:background="#4F6E9C"
    app:cardBackgroundColor="#3A3535"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp">

    <LinearLayout
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal|center_vertical"
        android:layout_margin="16dp"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/imageView12"
            android:layout_width="70dp"
            android:layout_height="70dp"
            app:srcCompat="@drawable/papers2" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="Feedbacks"
            android:textAlignment="center"
            android:textColor="#ffffff"
            android:textSize="12sp"
            android:textStyle="bold" />

    </LinearLayout>

</androidx.cardview.widget.CardView>
<!-- COLUMN 2  -->
<androidx.cardview.widget.CardView
    android:id="@+id/cardnotify"
    android:layout_width="0dp"
    android:layout_height="30dp"
    android:layout_rowWeight="1"
    android:layout_columnWeight="1"
    android:layout_marginLeft="10dp"
    android:layout_marginRight="10dp"
    android:layout_marginBottom="37dp"
    android:background="#4F6E9C"
    app:cardBackgroundColor="#3A3535"
    app:cardCornerRadius="8dp"
    app:cardElevation="8dp">
```

```
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal|center_vertical"
            android:layout_margin="16dp"
            android:orientation="vertical">
            <ImageView
                android:id="@+id/imageView10"
                android:layout_width="70dp"
                android:layout_height="70dp"
                android:layout_gravity="center"
                app:srcCompat="@drawable/notification" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="Send Notifications"
                android:textAlignment="center"
                android:textColor="#ffffff"
                android:textSize="12sp"
                android:textStyle="bold" />

        </LinearLayout>

    </androidx.cardview.widget.CardView>

  </GridLayout>

</LinearLayout>
```

PERSON.JAVA

```
package com.presidency.resultpro;

public class person
{
  // Variable to store data corresponding
```

```
// to firstname keyword in database
private String name;

// Variable to store data corresponding
// to lastname keyword in database
private String registernumber;

// Variable to store data corresponding
// to age keyword in database
private String status;

// Mandatory empty constructor
// for use of FirebaseUI
public person() {}

// Getter and setter method
public String getName()
{
   return name;
}
public void setName(String name)
{
   this.name = name;
}
public String getRegisternumber()
{
   return registernumber;
}

public void setRegisternumber(String registernumber)
{
   this.registernumber = registernumber;
}
public String status()
{
   return status;
}
public String getStatus()
{
```

```
        return status;
    }
    public void setStatus(String status)
    {
        this.status = status;
    }
}
```

PERSONADAPTER.JAVA

```
package com.presidency.resultpro;

import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import androidx.annotation.NonNull;
import androidx.recyclerview.widget.RecyclerView;
import com.firebase.ui.database.FirebaseRecyclerAdapter;
import com.firebase.ui.database.FirebaseRecyclerOptions;


// FirebaseRecyclerAdapter is a class provided by
// FirebaseUI. it provides functions to bind, adapt and show
// database contents in a Recycler View
public class personAdapter extends FirebaseRecyclerAdapter<
        person,personAdapter.personsViewholder> {

    public personAdapter(
            @NonNull FirebaseRecyclerOptions<person> options)
    {
        super(options);
    }

    // Function to bind the view in Card view(here
    // "person.xml") iwth data in
    // model class(here "person.class")
    @Override
    protected void
    onBindViewHolder(@NonNull personsViewholder holder,
```

```
                int position, @NonNull person model)
{

    // Add firstname from model class (here
    // "person.class")to appropriate view in Card
    // view (here "person.xml")
    holder.firstname.setText(model.getName());

    // Add lastname from model class (here
    // "person.class")to appropriate view in Card
    // view (here "person.xml")
    holder.lastname.setText(model.getRegisternumber());

    // Add age from model class (here
    // "person.class")to appropriate view in Card
    // view (here "person.xml")
    holder.age.setText(model.getStatus());
}

// Function to tell the class about the Card view (here
// "person.xml")in
// which the data will be shown
@NonNull
@Override
public personsViewholder
onCreateViewHolder(@NonNull ViewGroup parent,
                   int viewType)
{
    View view
            = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.person, parent, false);
    return new personAdapter.personsViewholder(view);
}

// Sub Class to create references of the views in Crad
// view (here "person.xml")
class personsViewholder
        extends RecyclerView.ViewHolder {
    TextView firstname, lastname, age;
```

```
    public personsViewholder(@NonNull View itemView)
    {
        super(itemView);

        firstname
                = itemView.findViewById(R.id.firstname);
        lastname = itemView.findViewById(R.id.lastname);
        age = itemView.findViewById(R.id.age);
    }
  }
}
```

STUDENTRESULTS.JAVA

```
package com.presidency.resultpro;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import android.widget.TableLayout;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.cardview.widget.CardView;
import com.firebase.client.Firebase;
import com.firebase.client.FirebaseError;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.Query;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;
public class StudentResults extends Activity {
   static DatabaseReference rootRef;
   static DatabaseReference demoRefnum;
```

```
    static DatabaseReference cnsverify;
    static DatabaseReference officear;
    static DatabaseReference dmr;
    static DatabaseReference indianlanguage;
    static DatabaseReference english;
    static DatabaseReference name;
    static DatabaseReference clanguage;
    static DatabaseReference comporg;
    static DatabaseReference clab;
    static DatabaseReference foundationc;
    static DatabaseReference cc;
    static DatabaseReference demoRefmarks;
    private ArrayList<String> Userlist;
    TextView markstxt;
    static Button verify,ovbtn;
    public static float dmpercent, cccc, officepercent, englishpercentage, comppercentage,
clab1, ccc, foundationpercent, cpercentage, langpercent;
    public static TextView percentage, grade,comp, lang, English, cl, officepercentage, Dm,
clabpercent11, found, ccnec, regnum, fetchedtextname, fetchedtext4, fetchedtext10, officea,
dm, fetchedtext5, fetchedtext666, fetchedtext111, fetchedtext7, fetchedtext8;
    TableLayout resulttable;
    CardView detailcard;
    static String output10000, output100001, output1, output1x, output1ppp, output4, output5,
output10055, output666, output7, output8;

   static String register;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        Firebase.setAndroidContext(this);
        setContentView(R.layout.activity_student_results);
        resulttable = findViewById(R.id.resulttable);
        comp = findViewById(R.id.comp);
        grade = findViewById(R.id.grade);
        percentage = findViewById(R.id.percentage);
        ovbtn = findViewById(R.id.ovbtn);
        English = findViewById(R.id.English);
        cl = findViewById(R.id.cl);
```

```
officepercentage = findViewById(R.id.officepercentage);
Dm = findViewById(R.id.Dm);
clabpercent11 = findViewById(R.id.clabpercent11);
found = findViewById(R.id.found);
ccnec = findViewById(R.id.ccnec);
detailcard = findViewById(R.id.detailcard);
fetchedtextname = findViewById(R.id.fetchedtextname);
fetchedtext4 = findViewById(R.id.fetchedtext4);
fetchedtext5 = findViewById(R.id.fetchedtext5);
fetchedtext666 = findViewById(R.id.fetchedtext666);
fetchedtext111 = findViewById(R.id.fetchedtext111);
fetchedtext7 = findViewById(R.id.fetchedtext7);
fetchedtext8 = findViewById(R.id.fetchedtext8);
fetchedtext10 = findViewById(R.id.fetchedtext10);
regnum = findViewById(R.id.regnum);
lang = findViewById(R.id.lang);
officea = findViewById(R.id.officea);
dm = findViewById(R.id.dm);
verify = findViewById(R.id.verifybtn);
resulttable.setVisibility(View.INVISIBLE);
detailcard.setVisibility(View.INVISIBLE);
rootRef = FirebaseDatabase.getInstance().getReference();
demoRefnum =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(LoginActivity.usern
ame).child("registernumber");
Query regno = demoRefnum.equalTo(register);
String demoRefnums = demoRefnum.toString();
resulttable.setVisibility(View.VISIBLE);
detailcard.setVisibility(View.VISIBLE);
if (LoginActivity.username.isEmpty()) {
   ///Toast.makeText(getApplicationContext(), "Empty Field",
///Toast.LENGTH_SHORT).show();

} else {

   getit();

}
String subjects[] ={"SEMESTER I","SEMESTER II","SEMESTER III","SEMESTER
```

```
IV","SEMESTER V","SEMESTER VI"};
        String myString="1";
        Spinner spinnersub=findViewById(R.id.spinnersubject);
        ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this,R.layout.spinnerlist,subjects);
        spinnersub.setAdapter(adapter);
        int spinnerPosition = adapter.getPosition(myString);
        spinnersub.setSelection(spinnerPosition);

        spinnersub.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id)
{
                switch (position) {
                    case 0:
                        resulttable.setVisibility(View.VISIBLE);

                        break;
                    case 1:
                        resulttable.setVisibility(View.INVISIBLE);
                        grade.setText("");
                        percentage.setText("");

                        break;
                    case 2:
                        resulttable.setVisibility(View.INVISIBLE);
                        grade.setText("");
                        percentage.setText("");
                        break;
                    case 3:
                        resulttable.setVisibility(View.INVISIBLE);
                        grade.setText("");
                        percentage.setText("");
                        break;
                    case 4:
                        resulttable.setVisibility(View.INVISIBLE);
                        grade.setText("");
                        percentage.setText("");
                        break;
```

```
                case 5:
                    resulttable.setVisibility(View.INVISIBLE);
                    grade.setText("");
                    percentage.setText("");
                    break;
                default:
                    resulttable.setVisibility(View.INVISIBLE);
                    grade.setText("");
                    percentage.setText("");
                    break;
            }
        }
        @Override
        public void onNothingSelected(AdapterView<?> parent) {

        }
    });
    verify.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(getApplicationContext(), "Verified Successfully!",
Toast.LENGTH_LONG).show();
            String value = "Verified by Student";
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(LoginActivity.usern
ame).child("status").setValue(value);

        }
    });

}
public static void getit() {


    register = LoginActivity.username;
    final Firebase firebaseRef = new Firebase("https://resultpro-29f21-default-
rtdb.firebaseio.com/FirstSem3").child(register).child("registernumber");

///Toast.makeText(getApplicationContext(),register,///Toast.LENGTH_SHORT).show();
    firebaseRef.addListenerForSingleValueEvent(new
```

```
com.firebase.client.ValueEventListener() {
        @Override
        public void onDataChange(com.firebase.client.DataSnapshot dataSnapshot) {
           if (dataSnapshot.exists()) {

               /////Toast.makeText(getApplicationContext(), "im in",
///Toast.LENGTH_SHORT).show();
               cnsverify =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("statu
s");

               indianlanguage =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("IND
IAN LANGUAGE");
               name =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("nam
e");
               english =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("EN
GLISH");
               clanguage =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("C
LANGUAGE");
               comporg =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("CO
MPUTER ORGANISATION");
               dmr =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("DM
");
               clab =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("C
LAB");
               officear =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("OFF
ICE AUTOMATION");
               foundationc =
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("FO
UNDATION COURSE");
               cc =
```

```
FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("CCANDEC");
        demoRefmarks = FirebaseDatabase.getInstance().getReference().child("FirstSem3").child(register).child("registernumber");


        demoRefmarks.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot10000) {


                output10000 = dataSnapshot10000.getValue().toString();

                if (output10000 != null) {
                    //Set the textview to the output string
                    regnum.setText(output10000);


                } else {

                    ///Toast.makeText(getApplicationContext(), "Invalid register number", ///Toast.LENGTH_SHORT).show();

                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register number", ///Toast.LENGTH_SHORT).show();

            }
        });

        name.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot10000) {
```

```
                    output100001 = dataSnapshot10000.getValue().toString();
                    if (output10000 != null) {
                        //Set the textview to the output string
                        fetchedtextname.setText(output100001);

                    } else {


                        ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                    }
                }

                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();

                }
            });

            indianlanguage.addListenerForSingleValueEvent(new ValueEventListener() {


                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {


                    output1 = dataSnapshot.getValue().toString();

                    if (output1 != null) {
                        //Set the textview to the output string
                        fetchedtext111.setText(output1);
                        langpercent = (Float.parseFloat(output1) / 30) * 100;
                        String langg = Float.toString(langpercent);

                       lang.setText(langg);
```

```
                } else {


                    ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
                // ///Toast.makeText(getApplicationContext(), "Please Enter Correct
Register number", ///Toast.LENGTH_SHORT).show();
            }
        });

        officear.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {


                output1x = dataSnapshot.getValue().toString();

                if (output1x != null) {
                    //Set the textview to the output string
                    officea.setText(output1x);
                    officepercent = (Float.parseFloat(output1x) / 15) * 100;
                    String off = Float.toString(officepercent);
                    officepercentage.setText(off);
                } else {


                    ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                }
            }
```

```
                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();
                }
            });

            dmr.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot45) {


                    output1ppp = dataSnapshot45.getValue().toString();

                    if (output1ppp != null) {
                        //Set the textview to the output string
                        dm.setText(output1ppp);
                        dmpercent = (Float.parseFloat(output1ppp)) / 50 * 100;
                        String dms = Float.toString(dmpercent);
                        Dm.setText(dms);
                    } else {

                        ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                    }
                }

                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();
                }
            });


            english.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot10) {
```

```
                    output4 = dataSnapshot10.getValue().toString();
                    if (output4 != null) {

                        //Set the textview to the output string

                        fetchedtext4.setText(output4);
                        englishpercentage = (Float.parseFloat(output4) / 30) * 100;
                        String eng = Float.toString(englishpercentage);
                        English.setText(eng);
                    } else {

                        ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                    }
                }

                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();

                }
            });
            clanguage.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot5) {


                    output5 = dataSnapshot5.getValue().toString();

                    if (output5 != null) {
                        //Set the textview to the output string

                        fetchedtext10.setText(output5);
                        cpercentage = (Float.parseFloat(output5) / 30) * 100;
                        String cp = Float.toString(cpercentage);
```

```
                    cl.setText(cp);

                } else {


                    ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                }
            }


            @Override
            public void onCancelled(@NonNull DatabaseError error) {

            }
        });


        comporg.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot555) {


                output10055 = dataSnapshot555.getValue().toString();

                if (output10055 != null) {
                    //Set the textview to the output string

                    fetchedtext5.setText(output10055);
                    comppercentage = (Float.parseFloat(output10055) / 30) * 100;
                    String com = Float.toString(comppercentage);
                    comp.setText(com);
                } else {


                    ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();
```

```
            }
        }


        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register number", ///Toast.LENGTH_SHORT).show();

        }
    });


    clab.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot666) {


            output666 = dataSnapshot666.getValue().toString();
            if (output666 != null) {

                //Set the textview to the output string

                fetchedtext666.setText(output666);
                clab1 = (Float.parseFloat(output666) / 15) * 100;
                String cll = Float.toString(clab1);
                clabpercent11.setText(cll);
            } else {


                ///Toast.makeText(getApplicationContext(), "Invalid register number", ///Toast.LENGTH_SHORT).show();

            }
        }


        @Override
        public void onCancelled(@NonNull DatabaseError error) {
```

```
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();

                }
            });


            foundationc.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot7) {


                    output7 = dataSnapshot7.getValue().toString();

                    if (output7 != null) {
                        //Set the textview to the output string

                        fetchedtext7.setText(output7);
                        foundationpercent = ((Float.parseFloat(output7)) / 30) * 100;
                        String foundd = Float.toString(foundationpercent);
                        found.setText(foundd);
                    } else {


                        ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                    }
                }


                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();

                }
            });
```

```
            cc.addListenerForSingleValueEvent(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot8) {


                    output8 = dataSnapshot8.getValue().toString();

                    if (output8 != null) {
                        //Set the textview to the output string

                        fetchedtext8.setText(output8);
                        cccc = (Float.parseFloat(output8) / 50) * 100;
                        String ccc = Float.toString(cccc);
                        ccnec.setText(ccc);
                    } else {


                        ///Toast.makeText(getApplicationContext(), "Invalid register number",
///Toast.LENGTH_SHORT).show();

                    }
                }


                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                    ///Toast.makeText(getApplicationContext(), "Please Enter Correct Register
number", ///Toast.LENGTH_SHORT).show();

                }


            });

        } else {

            ///Toast.makeText(getApplicationContext(),"Invalid Register
Number",///Toast.LENGTH_SHORT).show();
```

```
            }


        }


        @Override
        public void onCancelled(FirebaseError firebaseError) {
            ///Toast.makeText(getApplicationContext(),"Invalid Register
Number",///Toast.LENGTH_SHORT).show();
        }

    });

ovbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        float overall = ((Float.parseFloat(lang.getText().toString()) +
Float.parseFloat(English.getText().toString()) + Float.parseFloat(comp.getText().toString())
+ Float.parseFloat(ccnec.getText().toString()) + Float.parseFloat(found.getText().toString())
+ Float.parseFloat(cl.getText().toString()) + Float.parseFloat(officea.getText().toString()) +
Float.parseFloat(Dm.getText().toString()) +
Float.parseFloat(clabpercent11.getText().toString())) / 9);
        percentage.setText(Float.toString(overall) + " %");


        if (overall > 90) {

            grade.setText("O-outstanding");


        } else if (overall > 80 && overall < 90) {

            grade.setText("A+ Exemplary");

        }
```

```
    else if(overall>70&&overall<80)
    {
        grade.setText("A First Class Distinction");
    }

else if(overall>60&&overall<70){
        grade.setText("B+");
    }

else if(overall>50&&overall<60){
            grade.setText("B");
        }

    }
});
    }

}
```

10. CONCLUSION AND FUTURE ENHANCEMENTS

CONCLUSION:

This project can highly increase the productivity of students, faculties, and teachers. Its advanced security and precise information can be of great use. It can store a lot of information precisely on the database. Edition, updating and deletion of the data are not tedious anymore. Further, advancements like displaying the information in the form of graphics and charts are also implemented.

FUTURE ENHANCEMENTS:

- Individual chat with faculty by students.
- Online Study materials.
- Attendance tracking.
- Accurate Time Table.
- Fees management.

11. BIBLIOGRAPHY

1) Books Referred:

- Android App Development for Dummies by Michael Burton
- Head First Android Development: A Brain-Friendly Guide by Dawn Griffiths
- Head First Java: A Brain-Friendly Guide by Kathy Sierra and Bert Bates

2) Websites referred:

- www.stackoverflow.com
- www.youtube.com
- www.w3school.com
- www.pub.dev
- http://en.wikipedia.org
- http://www.geeksforgeeks.org/
- http://medium.com/androiddevelopers