Autonomous Institute, Affiliated to **JNTUH,**
Approved by **AICTE,** Accredited by **NAAC** with **A**++ Grade
Kacharam, Shamshabad, Hyderabad- 501 218,
Telangana, India.

COURSE END PROJECT

Student Performance evaluation App

MOBILE APPLICATION DEVELOPMENT(A8606)

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE OF ENGINEERING

SUBMITTED BY

23881A05F8 K. JYOTHI

UNDER THE GUIDANCE OF
Mr. B. J. VENKATA VARMA
Assistant Professor

Autonomous Institute, Affiliated to **JNTUH,**
Approved by **AICTE,** Accredited by **NAAC** with **A**++ Grade
Kacharam, Shamshabad, Hyderabad- 501 218,
Telangana, India.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the short technical report work entitled “title” carried out by **Name: K. JYOTHI, Rollno: 23881A05F8** towards **A8606 – MOBILE APPLICATION DEVELOPMENT** course and submitted to **Department of Computer science & Engineering,** in partial fulfilment of the Requirements for the award of degree of Bachelor of Technology during the academic year 2025-2026.

Name & signature of the Instructors	**Name & signature of the HOD**
Mr. B. J. Venkata Varma Assistant Professor	Dr. Ramesh Karnati ` Associate Professor, HoD, CSE-I

VARDHAMAN
COLLEGE OF ENGINEERING

Autonomous Institute, Affiliated to **JNTUH,**
Approved by **AICTE,** Accredited by **NAAC** with **A**++ Grade
Kacharam, Shamshabad, Hyderabad- 501 218,
Telangana, India.

Years of Vibrant Vardhaman

DEPARTMENTOFCOMPUTER SCIENCE AND ENGINEERING
Course End Project
MOBILE APPLICATION DEVELOPMENT(A8606)

CheckList

S.No.	Content	Status(✓/ ×)
1	Abstract	✓
2	Problem Statement	✓
3	Objectives	✓
4	Algorithm	✓
5	Flowchart	✓
6	Sourcecode	✓
7	Resultsand Discussion	✓
8	Conclusion	✓
9	Future scope	✓
10	References	✓

Signature of the Supervisor
Mr. B. J Venkata Varma
Assistant Professor

1. ABSTRACT

The **Student Performance Evaluation App** is a mobile-based solution designed to help educational institutions, teachers, students, and parents efficiently monitor and improve academic outcomes. It collects key data such as marks, attendance, assignments, and behavioral records to generate real-time performance reports with interactive visualizations like charts and graphs. With its user-friendly interface and secure data management, teachers can record grades and provide feedback effortlessly, while students can track their progress over time, identify strengths and weaknesses, and receive personalized recommendations for improvement. The app also offers features like automated report generation, attendance alerts, goal tracking, and parent access for transparent communication. By integrating analytics and smart evaluation tools, the app enhances decision-making for educators and fosters self-assessment among students, ultimately promoting better learning outcomes and streamlined academic management.the Student Performance Evaluation App simplifies academic tracking, promotes transparency, and supports data-driven improvement, making it an effective tool for enhancing student success and communication between teachers, students, and parents.

2. PROBLEMSTATEMENT

Monitoring and evaluating student performance through manual methods is often time-consuming, susceptible to errors, and lacks real-time visibility, making it challenging for teachers to detect learning gaps early. Students receive delayed or limited feedback, while parents remain uninformed due to ineffective communication channels. To address these issues, there is a growing need for a centralized digital solution that enables real-time performance tracking, automated assessment, and intelligent data analytics. This project proposes the development of a mobile application that simplifies grade and attendance management, strengthens communication between teachers, students, and parents, and delivers visual reports along with personalized recommendations. By doing so, it promotes continuous improvement and significantly enhances learning outcomes.

3. OBJECTIVES

a. To develop a mobile application for evaluating student academic performance.

b. To automate the calculation of grades, averages, and overall performance.

c. To provide graphical representations of performance through charts and graphs.

d. To allow students to track progress and identify areas of improvement.

e. To provide secure storage of student data with user authentication.

f. To generate performance reports for teachers and management.

g. To enhance student-teacher interaction and improve academic efficiency.

h. To enable attendance tracking and integrate it into performance evaluation.

i. To notify students and parents about important updates such as low performance, missed assignments, or upcoming tests.

j. To allow teachers to provide personalized feedback and study recommendations.

k. To support multi-role access for teachers, students, parents, and administrators.

l. To facilitate data export and backup for academic record management.

4. ALGORITHM

Step 1: Start the application and load the login page.

Step 2: Authenticate user (Teacher/Student).

Step 3: If Teacher:

- Input student data marks, attendance, assignments).
- Calculate total marks, grades, and average performance.
- Store data securely in the database.

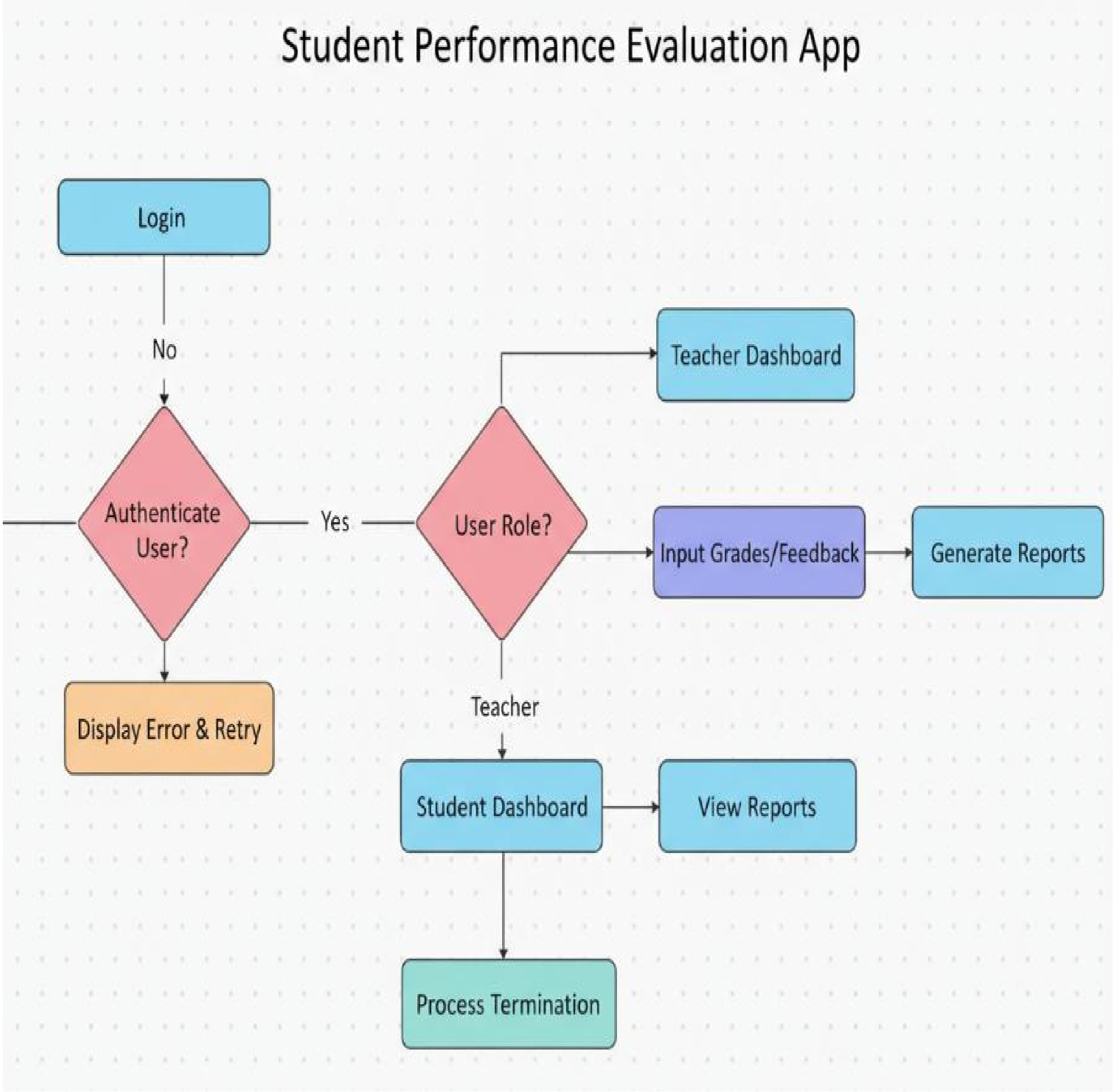
Step 4: If Student:

- View individual performance dashboard.
- Access grades, attendance, and teacher feedback.
- Generate graphical performance reports.

Step 5: Allow data export as PDF or report generation.

Step 6: End process.

5. FLOWCHART


Student Performance Evaluation App
Login
No
Authenticate User?
Yes
User Role?
Teacher Dashboard
Input Grades/Feedback
Generate Reports
Display Error & Retry
Teacher
Student Dashboard
View Reports
Process Termination

6. SOURCECODE

//activity_main.xml

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

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

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

android:layout_width="match_parent"

android:layout_height="match_parent"

android:orientation="vertical"

android:gravity="center"

android:padding="20dp"

android:background="#E8EEF1"

tools:context=".MainActivity">

<TextView

android:id="@+id/titleText"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Student Performance Evaluation"

android:textSize="20sp"

android:textStyle="bold"

android:textColor="#000"

android:layout_marginBottom="20dp"

/>

<EditText

android:id="@+id/editName"
```

```
android:layout_width="match_parent"

android:layout_height="wrap_content"

android:hint="Enter Student Name"

android:inputType="textPersonName"

android:layout_marginBottom="10dp"

android:background="@android:drawable/edit_text"

android:padding="10dp"/>

<EditText

android:id="@+id/editMarks"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:hint="Enter Marks"

android:inputType="number"

android:layout_marginBottom="20dp"

android:background="@android:drawable/edit_text"

android:padding="10dp"/>

<Button

android:id="@+id/btnSubmit"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:text="Submit"

android:background="#6A1B9A"

android:textColor="#FFF"

android:layout_marginBottom="10dp"/>

<Button
```

```
android:id="@+id/btnView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="View Performance"
android:background="#6A1B9A"
android:textColor="#FFF"
android:layout_marginBottom="20dp"/>
<TextView
android:id="@+id/tvResult"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text=""
android:textSize="16sp"
android:gravity="center"/>
</LinearLayout>
```

//AndroidMenifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools">
<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="project"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
```

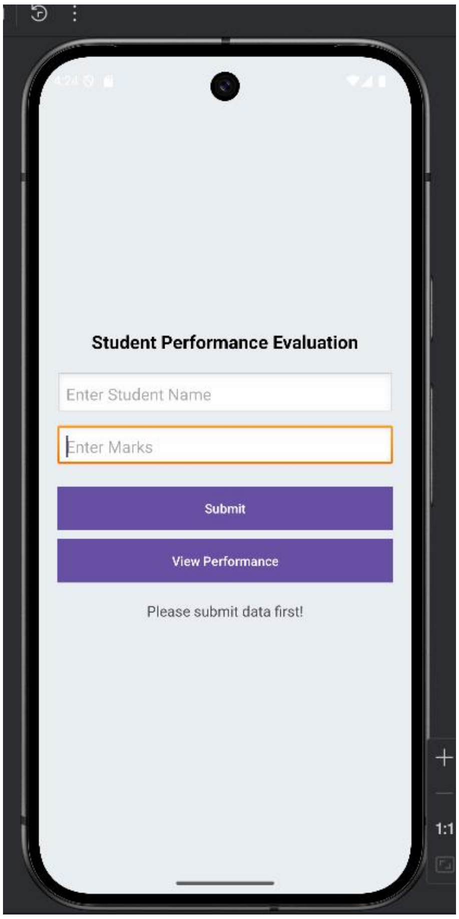
```
android:theme="@style/Theme.Project">
<activity android:name=".MainActivity"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

//MainActivity.java

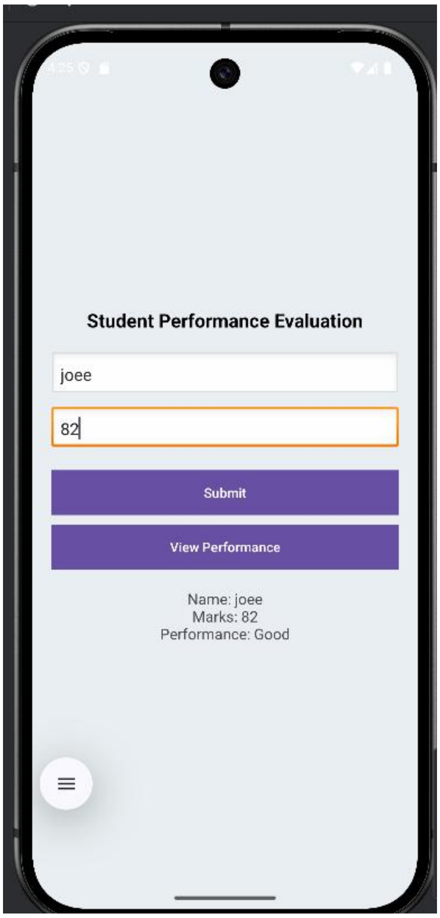
```
package com.example.project;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
EditText editName, editMarks;
Button btnSubmit, btnView;
TextView tvResult;
String studentName = "";
int marks = 0;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
editName = findViewById(R.id.editName);
editMarks = findViewById(R.id.editMarks);
btnSubmit = findViewById(R.id.btnSubmit);
btnView = findViewById(R.id.btnView);
tvResult = findViewById(R.id.tvResult);
btnSubmit.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
studentName = editName.getText().toString().trim();
String marksStr = editMarks.getText().toString().trim();
if (studentName.isEmpty() || marksStr.isEmpty()) {
Toast.makeText(MainActivity.this, "Please enter both fields",
Toast.LENGTH_SHORT).show();
return;
}
try {
marks = Integer.parseInt(marksStr);
Toast.makeText(MainActivity.this, "Data Submitted!", Toast.LENGTH_SHORT).show();
} catch (NumberFormatException e) {
Toast.makeText(MainActivity.this, "Enter valid marks", Toast.LENGTH_SHORT).show();
}
}
});
btnView.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if (studentName.isEmpty()) {
tvResult.setText("Please submit data first!");
} else {
String performance;
if (marks >= 90) performance = "Excellent";
else if (marks >= 75) performance = "Good";
else if (marks >= 50) performance = "Average";
else performance = "Needs Improvement";
tvResult.setText("Name: " + studentName + "\nMarks: " + marks + "\nPerformance: " + performance);
}
}
});
}
}
```

7. RESULTSANDDISCUSSION



Please submit data first!



joee

82

```
Name: joee
Marks: 82
Performance: Good
```

The project successfully demonstrates the integration of basic Android UI components (EditText, Button, TextView) with backend logic using Java.

Key discussion points include:

- The app ensures **input validation**, preventing empty or invalid entries.
- The **performance evaluation logic** is simple yet effective, suitable for beginner-level Android projects.
- The **real-time feedback** (using Toast messages and TextView updates) enhances user interaction.
- The application runs smoothly on devices with different screen sizes due to the use of responsive layout and match_parent attributes.
- The app design follows **Material Design color principles**, maintaining readability and usability.

However, for future enhancements, the following improvements can be considered:

- Storing student data permanently using **SQLite Database** or **Firebase**.
- Adding a **Performance History Screen** to view past evaluations.
- Implementing **Graphical Representation** (bar or pie charts) of student performance.
- Enabling **multiple subject inputs** and calculating averages.

8. CONCLUSION

The Student Performance Evaluation app is a beginner-friendly Android project that assesses students' performance based on their marks. Users input a student's name and scores, and the app validates the data before classifying performance as Excellent, Good, Average, or Needs Improvement. It uses core Android components like EditText, Button, and TextView, along with Java-based event handling and conditional logic. The simple, intuitive design makes it accessible to all users and serves as a strong foundation for learning Android development and educational app design.

9. FUTURE SCOPE

- Integrate **AI-based performance prediction** for student outcomes.
- Include **notifications and reminders** for pending assignments or low attendance.
- Add **multi-school or institutional support** for centralized evaluation.
- Enable **integration with online learning platforms** for automatic grad fetching.
- Develop **cross-platform support** for iOS, Android, and web.
- Provide role-based access for **students, teachers, parents, and administrators**.
- Introduce **gamification elements** like badges, streaks, or reward levels to encourage consistent performance.
- Enable **real-time chat or feedback channels** between teachers and students.

Support **offline usage with automatic data sync** when internet connectivity is restored.

10. REFERENCES

1. Android Developers Documentation – https://developer.android.com/docs.

2. Mobile Application Development, Reema Thareja, Oxford University Press, 2021

3. “Mobile App Development for Education” – Journal of Educational Technology, 2022.

4. Kotlin Programming Official Documentation – https://kotlinlang.org/docs.

5. SQLite Database for Android Apps – https://www.sqlite.org.