TECHPATASHALA

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

BY

P.Amulya (22501A05D7)

Under the Guidance of

Mr.B.Vishnu Vardhan ,B.Tech,M.Tech

Assistant Professor

PRASAD V POTLURI SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Permanently affiliated to JNTU :: Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 Certified Institution)

Kanuru, Vijayawada - 520007

(2024-25)

PRASAD V POTLURI

SIDDHARTHA INSTITUTE OF TECHNOLOGY

(Permanently affiliated to JNTU :: Kakinada, Approved by AICTE)

(An NBA & NAAC A+ accredited and ISO 9001:2015 certified institution)

Kanuru, Vijayawada – 520007

CERTIFICATE

This is to certify that the project report titled **"TECHPATASHALA"** is the bona fide work of **P.Amulya (22501A05D7)** in partial fulfilment of completing the Academic project in Mobile App Development Lab during the academic year 2024-25.

Signature of the Incharge **Signature of the HOD**

1.ABSTRACT

In today's digital era, access to educational resources and digital literacy skills is essential for success in education, employment, and daily life. However, marginalized communities often lack structured learning resources and the necessary guidance to develop these skills. **TechPatashala** addresses this challenge by offering an interactive and user-friendly mobile application that empowers individuals to explore digital literacy concepts and engage with educational content.

The primary objective of TechPatashala is to deliver educational content in a **document (PDF) format** that users can download and view through related PDF viewing applications. This approach ensures that learners have access to offline learning resources that they can refer to at their convenience. Through a cloud-based system powered by Firebase, uploaded content is securely stored and made available for download in PDF format, enhancing accessibility and promoting independent learning.

Key Features of TechPatashala:

- **Downloadable Document Format:** Users can download educational content in PDF format for offline access.
- **User Role Management:** Viewers can access content, while uploaders can contribute new educational material.
- **Content Security and Cloud Storage:** All content is securely stored using Firebase, ensuring seamless access and content reliability.

By fostering an environment where users can engage with quality educational resources, TechPatashala aims to bridge the digital divide and empower individuals to navigate the digital world confidently.

1.1. SDG JUSTIFICATION REPORT

SDG Mapped: SDG 4 – Quality Education, SDG 8 – Decent Work and Economic Growth

1.1.1 How This Project Supports SDG 4 & SDG 8

TechPatashala directly contributes to achieving SDG 4 and SDG 8 by fostering innovation, improving accessibility, and enhancing collaboration through:

- **Quality Education (SDG 4)**: The project enhances learning opportunities by providing accessible educational resources, interactive learning tools, and skill-building activities.
- **Decent Work & Economic Growth (SDG 8)**: It creates professional development opportunities, bridges the gap between academia and industry, and promotes entrepreneurship.
- **Enhanced Engagement**: Encourages participation in educational programs, workshops, and networking events, fostering career growth and skill development.
- **Efficient Management & Scalability**: Enables efficient organization and execution of educational and professional events, with potential for broader adoption across institutions and industries.

2.INTRODUCTION

In an increasingly digital world, access to structured educational content and digital literacy skills is crucial for success in education, employment, and daily life. However, many underprivileged communities lack the resources and opportunities needed to develop these essential skills. **TechPatashala** aims to address this gap by offering an interactive and user-friendly mobile application that provides downloadable educational content in document (PDF) format.

The platform serves two primary user roles:

- **Viewers:** Can browse and download educational content in PDF format for offline access.
- **Uploaders:** Can create, manage, and share content related to digital literacy and computer science topics.

TechPatashala ensures that all uploaded content is securely stored in the cloud using **Firebase**, allowing users to access learning material anytime and from any device. By providing educational content in a downloadable PDF format, TechPatashala empowers learners with the flexibility to engage with content offline, promoting continuous learning and skill development.

TechPatashala follows a structured system architecture that includes **user management, content management, and document generation modules**. Users can interact with the platform through a clean and intuitive interface that facilitates smooth navigation and content management. The cloud-based storage ensures that all content is securely accessible and up-to-date, enhancing the reliability and performance of the application.

By empowering individuals with easy access to relevant educational resources, **TechPatashala** strives to bridge the digital divide and create opportunities for skill enhancement and knowledge acquisition. Through continuous improvement and user engagement, TechPatashala remains committed to fostering an inclusive learning environment that benefits underserved communities.

3. OBJECTIVES AND SCOPE OF THE PROJECT

Objectives

- **Provide Downloadable Educational Content** in **document (PDF) format** for offline access through compatible PDF viewing applications.
- **Empower Content Creators** by allowing uploaders to create, manage, and share digital literacy content.
- **Ensure Secure and Reliable Content Management** using **Firebase** to securely store user information and uploaded content.
- **Offer an Intuitive and User-Friendly Interface** to enhance the user experience for both viewers and uploaders.
- **Promote Digital Literacy Among Underserved Communities** by offering accessible and structured learning resources.

Scope of the Project

- **Deliver Content in Document-Based Format** that can be downloaded as PDFs, ensuring offline accessibility.
- **Manage User Roles** with two primary roles: **Viewers** (browse and download) and **Uploaders** (create and manage content).
- **Store Content Securely in Firebase** to ensure real-time synchronization and seamless access from any device.
- **Facilitate Seamless Content Management** allowing uploaders to update or modify their contributions.
- **Enable Offline Learning** by allowing users to download content for future reference in environments with limited connectivity.

4. SOFTWARE USED

In the development of **TechPatashala**, the following software technologies were utilized to ensure a seamless, responsive, and user-friendly experience:

Frontend Technologies

XML (Extensible Markup Language):

- XML was used to design the user interface (UI) for the Android application.
- It defined the layout of various elements such as buttons, text fields, and navigation components, ensuring a clean and intuitive design.

Java:

- Java was used to handle the logic and functionality of the application.
- It ensured smooth integration between UI components and backend services while providing event-driven interaction with the application.

Android Studio:

- Android Studio was the primary IDE used for developing and testing the application.
- It offered various tools and emulators that facilitated efficient debugging and optimization of the app's performance.

Backend Technologies

Firebase:

- Firebase was used for managing backend operations, including user authentication, real-time database management, and cloud storage.
- It provided a secure environment to handle user login and content uploads.

Firebase Realtime Database:

- The Realtime Database stored all application-related data, including user profiles and uploaded content.
- It allowed data synchronization across devices in real-time.

Design and Documentation Tools

Canva:

- Canva was used for designing the use case diagrams and user interface.
- It provided visually appealing designs that aligned with the project's branding.

Microsoft Word:

- Microsoft Word was used for creating detailed project documentation.
- It ensured clarity and consistency in maintaining records, reports, and structured documentation for the project.

Version Control

GitHub:

- GitHub was used for version control and collaboration among the development team.
- It ensured that all code changes were tracked and managed efficiently.

5. PROPOSED MODEL

The TechPatashala platform follows a well-structured model to facilitate interactive learning and content management by providing downloadable educational content in document (PDF) format. The model is designed to offer a seamless experience for both viewers and uploaders while ensuring content security and accessibility.

1. User Management & Authentication

- Secure login and registration system using Firebase Authentication to ensure that only authorized users can upload and manage content.
- Password protection and encrypted data storage enhance the security of user information, ensuring safe and reliable access.

2. Content Browsing & Downloading

- Viewers can explore educational content related to digital literacy and technology.
- Users can download content in document (PDF) format for offline access using compatible PDF viewing applications.

3. Content Creation & Management

- Uploaders can create, manage, and delete their own content related to digital literacy and technology.
- Content is categorized by relevant topics, making it easier for viewers to search, access, and download the material.

4. Cloud-Based Storage & Accessibility

- All content and user data are securely stored in a Firebase Realtime Database, ensuring real-time synchronization and seamless access from any device.
- The cloud-based approach eliminates the risk of data loss and allows users to download content anytime, anywhere.

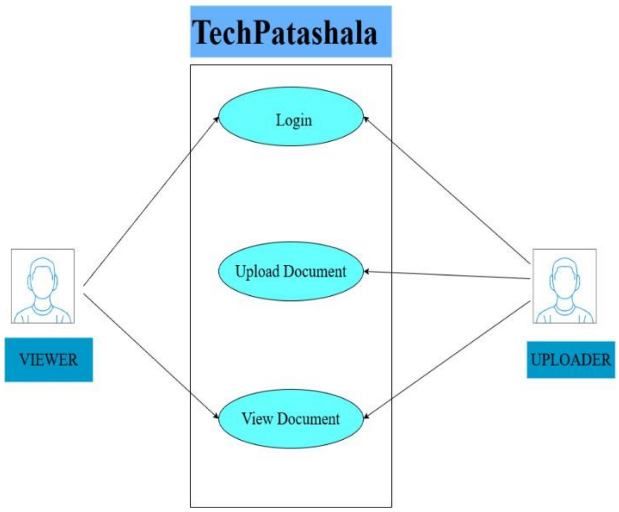
5. User-Friendly Interface

- The application features a visually appealing and intuitive interface that enhances the user experience.
- Simple navigation ensures that both viewers and uploaders can easily interact with the platform without any complexity.

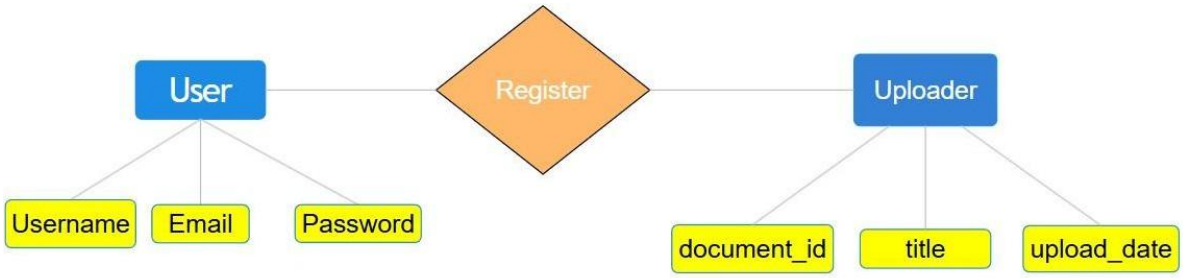
6. Content Review & Update

- Uploaders can update their content after submission, allowing modifications and improvements without creating duplicate entries.
- The platform provides an edit option where users can modify content descriptions **and** update tags as needed.

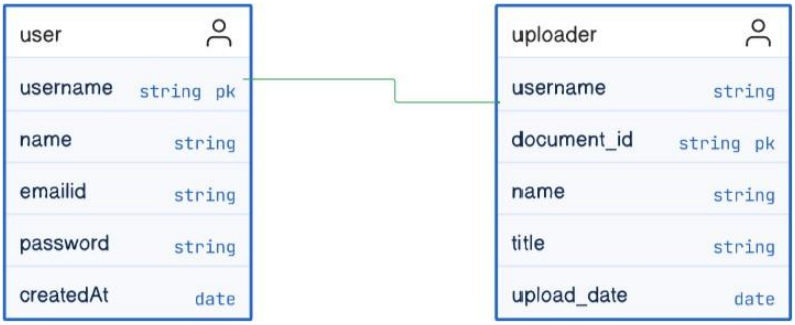
Use Case Diagram



E – R Diagram



Database Schema



6. SAMPLE CODE

MainActivity.java

```
package com.example.techpatashala;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import com.google.firebase.FirebaseApp;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        FirebaseApp.initializeApp(this);
        Button btnSignIn = findViewById(R.id.btn_signin);
        Button btnSignUp = findViewById(R.id.btn_signup);
        btnSignIn.setOnClickListener(view -> {
            Intent intent = new Intent(MainActivity.this, SignInActivity.class);
            startActivity(intent);
        });
        btnSignUp.setOnClickListener(view -> {
            Intent intent = new Intent(MainActivity.this, SignUpActivity.class);
            startActivity(intent);
        });
    }
}
```

Activity_main.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:gravity="center"
android:orientation="vertical"
android:padding="20dp"
android:background="#FFFACD">
<Button
   android:id="@+id/btn_signin"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginTop="88dp"
   android:backgroundTint="#FF9800"
   android:padding="12dp"
   android:text="Sign In"
   android:textColor="#FFFFFF"
   android:textSize="18sp"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintHorizontal_bias="1.0"
   app:layout_constraintStart_toStartOf="parent"
   app:layout_constraintTop_toBottomOf="@+id/imageView2" />
<Button
   android:id="@+id/btn_signup"
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_marginTop="20dp"
   android:backgroundTint="#4CAF50"
   android:padding="12dp"
   android:text="Sign Up"
   android:textColor="#FFFFFF"
   android:textSize="18sp"
   app:layout_constraintEnd_toEndOf="parent"
   app:layout_constraintHorizontal_bias="0.0"
   app:layout_constraintStart_toStartOf="parent"
```

```
        app:layout_constraintTop_toBottomOf="@+id/btn_signin" />
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="350dp"
        android:layout_height="266dp"
        android:layout_marginTop="64dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/logo_img" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

SignUpActivity.java

```
package com.example.techpatashala;
import android.content.Intent;
import android.os.Bundle;
import android.text.method.HideReturnsTransformationMethod;
import android.text.method.PasswordTransformationMethod;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
public class SignUpActivity extends AppCompatActivity {
    private boolean isPasswordVisible = false;
    private DatabaseReference fdb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_sign_up);
        fdb = FirebaseDatabase.getInstance("https://techpatashala-d6821-default-
rtdb.firebaseio.com/").getReference("users");
        EditText edtUsername = findViewById(R.id.edt_username);
        EditText edtEmail = findViewById(R.id.edt_email);
        EditText edtPassword = findViewById(R.id.edt_password);
        EditText edtConfirmPassword = findViewById(R.id.edt_confirm_password);
        ImageButton btnShowPassword = findViewById(R.id.btn_show_password);
        Button btnRegister = findViewById(R.id.btn_register);
        btnShowPassword.setOnClickListener(view -> {
            if (!isPasswordVisible) {
        edtPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
edtConfirmPassword.setTransformationMethod(PasswordTransformationMethod.getInstance
());
                btnShowPassword.setImageResource(R.drawable.baseline_remove_red_eye_24);
            } else {
    edtPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
edtConfirmPassword.setTransformationMethod(HideReturnsTransformationMethod.getInsta
nce());
                btnShowPassword.setImageResource(R.drawable.baseline_visibility_off);
            }
            isPasswordVisible = !isPasswordVisible;
        });
        btnRegister.setOnClickListener(view -> insertUser(edtUsername.getText().toString(),
edtEmail.getText().toString(), edtPassword.getText().toString()));
    }
    public void insertUser(String username, String email, String password) {
        User user = new User(username, email, password);
        fdb.push().setValue(user)
                .addOnSuccessListener(new OnSuccessListener<Void>() {
                    @Override
                    public void onSuccess(Void unused) {
                        Toast.makeText(SignUpActivity.this, "User Registered Successfully!",
Toast.LENGTH_SHORT).show();
```

```
                startActivity(new Intent(SignUpActivity.this, UserPageActivity.class));
                finish();
            }
        }).addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                Toast.makeText(SignUpActivity.this, "Registration Failed!",
Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

Activity_sign_up.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:orientation="vertical"
    android:padding="20dp"
    android:gravity="center"
    android:background="#FFFACD">
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="168dp"
        android:text="Sign Up"
        android:textColor="#4CAF50"
        android:textSize="24sp"
        android:textStyle="bold"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.432"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<EditText
    android:id="@+id/edt_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="36dp"
    android:background="@android:drawable/edit_text"
    android:hint="Username"
    android:padding="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView2" />
<EditText
    android:id="@+id/edt_email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:background="@android:drawable/edit_text"
    android:hint="Email"
    android:padding="10dp"
    app:layout_constraintTop_toBottomOf="@+id/edt_username"
    tools:layout_editor_absoluteX="20dp" />
<EditText
    android:id="@+id/edt_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:background="@android:drawable/edit_text"
    android:hint="Password"
    android:inputType="textPassword"
    android:padding="10dp"
```

```
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edt_email" />
  <EditText
    android:id="@+id/edt_confirm_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@android:drawable/edit_text"
    android:hint="Confirm Password"
    android:inputType="textPassword"
    android:padding="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/edt_password" />
  <ImageButton
    android:id="@+id/btn_show_password"
    android:layout_width="50dp"
    android:layout_height="46dp"
    android:background="?attr/selectableItemBackground"
    android:src="@drawable/baseline_remove_red_eye_24"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="@+id/edt_password" />
  <Button
    android:id="@+id/btn_register"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="128dp"
    android:backgroundTint="#4CAF50"
    android:padding="12dp"
    android:text="Register"
```

```
        android:textColor="#FFFFFF"
        app:layout_constraintTop_toBottomOf="@+id/edt_confirm_password"
        tools:layout_editor_absoluteX="5dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

SignInActivity.java

```
package com.example.techpatashala;
import android.content.Intent;
import android.os.Bundle;
import android.text.method.HideReturnsTransformationMethod;
import android.text.method.PasswordTransformationMethod;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
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;
public class SignInActivity extends AppCompatActivity {
    private boolean isPasswordVisible = false;
    private DatabaseReference fdb;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_in);
        fdb = FirebaseDatabase.getInstance("https://techpatashala-d6821-default-
rtdb.firebaseio.com/").getReference("users");
        EditText edtUsername = findViewById(R.id.edt_username);
        EditText edtPassword = findViewById(R.id.edt_password);
        ImageButton btnShowPassword = findViewById(R.id.btn_show_password);
```

```
        Button btnSubmit = findViewById(R.id.btn_submit);
        btnShowPassword.setOnClickListener(view -> {
            if (isPasswordVisible) {
        edtPassword.setTransformationMethod(PasswordTransformationMethod.getInstance());
                btnShowPassword.setImageResource(R.drawable.baseline_remove_red_eye_24);
            } else {
edtPassword.setTransformationMethod(HideReturnsTransformationMethod.getInstance());
                btnShowPassword.setImageResource(R.drawable.baseline_visibility_off);
            }
            isPasswordVisible = !isPasswordVisible;
        });
        btnSubmit.setOnClickListener(view -> {
            String username = edtUsername.getText().toString().trim();
            String password = edtPassword.getText().toString().trim();
            if (username.isEmpty() || password.isEmpty()) {
                Toast.makeText(SignInActivity.this, "Please enter username and password",
Toast.LENGTH_SHORT).show();
            } else {
                verifyUser(username, password);
            }
        });
    }
    private void verifyUser(String username, String password) {
        fdb.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                boolean userFound = false;
                for (DataSnapshot userSnapshot : snapshot.getChildren()) {
                    String dbUsername = userSnapshot.child("username").getValue(String.class);
                    String dbPassword = userSnapshot.child("password").getValue(String.class);
                    if (dbUsername != null && dbPassword != null &&
dbUsername.equals(username) && dbPassword.equals(password)) {
                        userFound = true;
                        break;
```

```
                }
            }
            if (userFound) {
                Toast.makeText(SignInActivity.this, "Login Successful!",
Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(SignInActivity.this, UserPageActivity.class);
                intent.putExtra("username", username);
                startActivity(intent);
                finish();
            }else {
                Toast.makeText(SignInActivity.this, "Wrong Credentials! Try Again.",
Toast.LENGTH_SHORT).show();
            }
        }
        @Override
        public void onCancelled(@NonNull DatabaseError error) {
            Toast.makeText(SignInActivity.this, "Database Error! Try Again.",
Toast.LENGTH_SHORT).show();
        }
    });
  }
}
```

Activity_singn_in.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:orientation="vertical"
   android:padding="20dp"
   android:gravity="center"
```

```
android:background="#FFFACD">
<TextView
    android:id="@+id/textView"
    android:layout_width="125dp"
    android:layout_height="33dp"
    android:layout_marginTop="204dp"
    android:text="Sign In"
    android:textAlignment="center"
    android:textColor="#FF9800"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
<EditText
    android:id="@+id/edt_username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="32dp"
    android:background="@android:drawable/edit_text"
    android:hint="Username"
    android:padding="10dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
<EditText
    android:id="@+id/edt_password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:background="@android:drawable/edit_text"
    android:hint="Password"
    android:inputType="textPassword"
```

```
        android:padding="10dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edt_username" />
    <ImageButton
        android:id="@+id/btn_show_password"
        android:layout_width="61dp"
        android:layout_height="41dp"
        android:layout_marginTop="16dp"
        android:background="?attr/selectableItemBackground"
        android:src="@drawable/baseline_visibility_off"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.954"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/edt_username" />
    <Button
        android:id="@+id/btn_submit"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:backgroundTint="#FF9800"
        android:padding="12dp"
        android:text="Submit"
        android:textColor="#FFFFFF"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="1.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/btn_show_password"
        app:layout_constraintVertical_bias="0.121" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

NewUploadActivity.java

```
package com.example.techpatashala;
import android.app.ProgressDialog;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.gms.tasks.OnFailureListener;
import com.google.android.gms.tasks.OnSuccessListener;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.storage.FirebaseStorage;
import com.google.firebase.storage.StorageReference;
public class NewUploadActivity extends AppCompatActivity {
    private static final int PICK_PDF_REQUEST = 1;
    EditText edtTitle;
    TextView txtFileSelected;
    Button btnSelectFile, btnUpload;
    Uri fileUri;
    ProgressDialog progressDialog;
    DatabaseReference databaseReference;
    StorageReference storageReference;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_new_upload);
        edtTitle = findViewById(R.id.edtTitle);
        txtFileSelected = findViewById(R.id.txtFileSelected);
```

```
        btnSelectFile = findViewById(R.id.btnSelectFile);
        btnUpload = findViewById(R.id.btnUpload);
        storageReference = FirebaseStorage.getInstance().getReference("upload");
        databaseReference = FirebaseDatabase.getInstance("https://techpatashala-d6821-
default-rtdb.firebaseio.com/").getReference("upload");
        progressDialog = new ProgressDialog(this);
        progressDialog.setMessage("Uploading...");
        btnSelectFile.setOnClickListener(v -> {
            Intent intent = new Intent(Intent.ACTION_GET_CONTENT);
            intent.setType("application/pdf");
            startActivityForResult(Intent.createChooser(intent, "Select PDF"),
PICK_PDF_REQUEST);
        });
        btnUpload.setOnClickListener(v -> {
            if (fileUri != null && !edtTitle.getText().toString().isEmpty()) {
                uploadFileToFirebase();
            } else {
                Toast.makeText(NewUploadActivity.this, "Please select a file and enter a title",
Toast.LENGTH_SHORT).show();
            }
        });
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == PICK_PDF_REQUEST && resultCode == RESULT_OK && data
!= null && data.getData() != null) {
            fileUri = data.getData();
            txtFileSelected.setText("File Selected: " + fileUri.getLastPathSegment());
        }
    }
    private void uploadFileToFirebase() {
        progressDialog.show();
```

```
        String title = edtTitle.getText().toString().trim();
        String fileName = title + "_" + System.currentTimeMillis() + ".pdf";
        StorageReference fileReference = storageReference.child(fileName);
        fileReference.putFile(fileUri)
                .addOnSuccessListener(taskSnapshot ->
fileReference.getDownloadUrl().addOnSuccessListener(uri -> {
                    saveFileToDatabase(title, "pdf", uri.toString());
                }))
                .addOnFailureListener(e -> {
                    progressDialog.dismiss();
                    Toast.makeText(NewUploadActivity.this, "Upload failed: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
                });
    }
    private void saveFileToDatabase(String title, String type, String fileUrl) {
        String uploadId = databaseReference.push().getKey();
        Upload upload = new Upload(title, type, fileUrl);
        assert uploadId != null;
        databaseReference.child(uploadId).setValue(upload)
                .addOnSuccessListener(aVoid -> {
                    progressDialog.dismiss();
                    Toast.makeText(NewUploadActivity.this, "File uploaded successfully!",
Toast.LENGTH_SHORT).show();
                    resetUI();
                })
                .addOnFailureListener(e -> {
                    progressDialog.dismiss();
                    Toast.makeText(NewUploadActivity.this, "Failed to save file data",
Toast.LENGTH_SHORT).show();
                });
    }
    private void resetUI() {
        edtTitle.setText("");
        txtFileSelected.setText("No file selected");
```

```
        fileUri = null;
    }
}
```

Activity_new_upload.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:padding="16dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <EditText
        android:id="@+id/edtTitle"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter Title"
        android:padding="10dp"
        android:textSize="16sp" />
    <TextView
        android:id="@+id/txtFileSelected"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="No file selected"
        android:textSize="14sp"
        android:textColor="#757575"
        android:layout_marginTop="8dp" />
    <Button
        android:id="@+id/btnSelectFile"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Select PDF"
        android:layout_marginTop="12dp" />
    <Button
        android:id="@+id/btnUpload"
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Upload PDF"
        android:layout_marginTop="12dp" />
</LinearLayout>
```

User.java

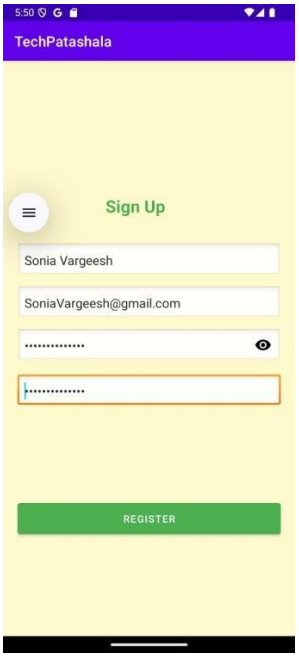
```
package com.example.techpatashala;
public class User {
    private String username, email, password;
    public User() {}
    public User(String username, String email, String password) {
        this.username = username;
        this.email = email;
        this.password = password;
    }
    public String getUsername() {
        return username;
    }
    public String getEmail() {
        return email;
    }
    public String getPassword() {
        return password;
    }
}
```

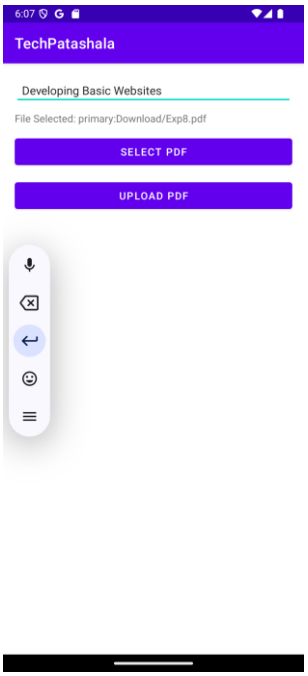
Upload.java

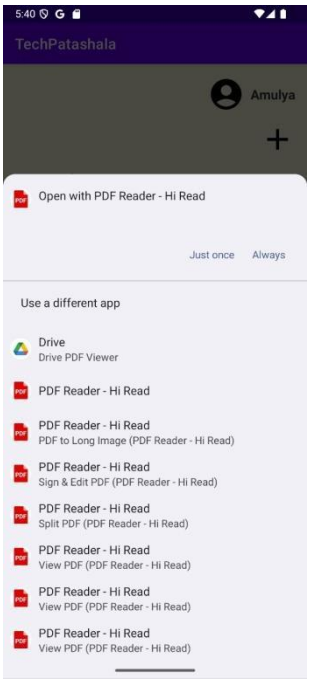
```
package com.example.techpatashala;
public class Upload {
    private String title;
    private String type;
    private String fileUrl;
    public Upload() {}
```

```
    public Upload(String title, String type, String fileUrl) {
        this.title = title;
        this.type = type;
        this.fileUrl = fileUrl;
    }
    public String getTitle() {
        return title;
    }
    public String getType() {
        return type;
    }
    public String getFileUrl() {
        return fileUrl;
    }
}
```

7.RESULT/OUTPUT SCREEN SHOTS







8. CONCLUSION AND FUTURE SCOPE

TechPatashala serves as an effective platform designed to promote digital literacy by providing accessible and structured learning resources for underprivileged communities. By leveraging modern technologies such as Firebase for authentication and real-time database management, TechPatashala ensures a seamless experience for users who seek to enhance their digital literacy skills.

Through this project, we aimed to simplify the process of learning and content sharing by providing features such as content browsing, secure cloud storage, and downloadable documents (PDFs). These functionalities empower users to explore educational content and allow uploaders to share their knowledge effectively without encountering unnecessary complexities.

Additionally, TechPatashala enhances learning by offering an intuitive and user-friendly interface that allows viewers to browse and download content efficiently while enabling uploaders to manage their contributions effortlessly. The platform's emphasis on content quality ensures that users engage with reliable and relevant information, ultimately fostering a culture of continuous learning and knowledge-sharing.

Future Scope

- **Support for multimedia content** with video and audio uploads.
- **Enable multilingual content** to cater to diverse audiences.
- **Introduce gamified learning modules** to enhance engagement.
- **Allow offline access** to downloaded content for uninterrupted learning.
- **Implement user analytics** to improve content relevance and user experience.

9.REFERENCES

1. **Android Studio:** Used as the primary IDE for developing, testing, and debugging the TechPatashala application.
   Link : https://developer.android.com/studio

2. **Firebase Console:** Used to manage authentication, real-time database, and cloud storage for the TechPatashala application.
   Link : https://firebase.google.com/

3. **Canva:** Used to design the logo for the TechPatashala application.
   Link : https://www.canva.com/design/DAGgXAWjli4/IocCIHsaNlaPxf-kn_rNTQ/view?utm_content=DAGgXAWjli4&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h5f83acbe06