**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**

**Jnana Sangama, Belagavi-590010**

|  |
| --- |
|  |

**A MINI PROJECT REPORT ON**

**FETCHING MEMES USING VOLLEY LIBRARY**

**Submitted in partial fulfillment for the requirements for the 6th semester.**

**BACHELOR OF ENGINEERING**

**IN**

**INFORMATION SCIENCE AND ENGINEERING**

**For the Academic Year 2022 - 2023**

Submitted by:

|  |  |
| --- | --- |
| **M SHAKTHI RAJ** | **1MV21IS400** |
| **PAVAN KUMAR B** | **IMV21IS401** |
| **SANDEEP RC** | **1MV21IS404** |

**Under the guidance of**

**Mr. Vitesh Babu M**

**Assistant Professor, Department of ISE**

|  |
| --- |
|  |
| **DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**  **SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY**  **HUNASAMARANAHALLI,**  **BENGALURU-562157** |

**SIR M. VISVESVARAYA INSTITUTE OF TECHNOLOGY**

Krishnadevaraya Nagar, International Airport Road, Hunasmaranahalli, Bengaluru – 562157

**DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING**

|  |
| --- |
|  |

**CERTIFICATE**

It is certified that the **MINI PROJECT** entitled **"FETCHING MEMES USING VOLLEY LIBRARY "** is carried out by **1MV21IS400 – M SHAKTHI RAJ , 1MV21IS401 – PAVAN KUMAR B, 1MV21IS404 – SANDEEP RC** bonafide **Students** of **Sir M Visvesvaraya Institute of Technology** in partial fulfillment for the 6th semester for the award of the Degree of Bachelor of Engineering in Information Science and Engineering of the **Visvesvaraya Technological University, Belagavi** during the academic year **2022-2023**.It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the course of Bachelor of Engineering.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name & Signature of  Guide |  | Name & Signature of  HOD |  | Name & Signature of Principal |
| **Mr. Vitesh Babu M**  Assistant Professor  Dept. Of ISE,  Sir MVIT  Bengaluru - 562157 |  | **Dr. G. C. Bhanu Prakash**  HOD,  Dept. Of ISE,  Sir MVIT  Bengaluru - 562157 |  | **Prof. S G Rakesh**  Principal,  Sir MVIT  Bengaluru – 562157 |

External Examination:

Name of Examiner Signature with Date

1)

2)

**DECLARATION**

We hereby declare that the entire project work embodied in this dissertation has been carried out by us and no part has been submitted for any degree or diploma of any institution previously.

Place Bengaluru :

Date:

|  |
| --- |
| **M SHAKTHI RAJ**  **(1MV21IS400)**  **PAVAN KUMAR B**  **(1MV21IS401)**  **SANDEEP RC**  **(1MV21IS404)** |

**Signature of Student**

**ACKNOWLEDGMENT**

It gives us immense pleasure to express our sincere gratitude to the management of **Sir M. Visvesvaraya Institute of Technology,** Bengaluru for providing the opportunity and the resources to accomplish our project work in their premises.

On the path of learning, the presence of an experienced guide is indispensable and we would like to thank our guide **Mr. Vitesh Babu M, Assistant Professor**, Dept. of SE, for her invaluable help and guidance.

Heartfelt and sincere thanks to **Dr. G. C. Bhanu Prakash,** HOD, Dept. of ISE, for his suggestions, constant support and encouragement.

We would also like to convey our regards to **Prof. S. G. Rakesh ,** Principal,   
Sir MVIT for providing us with the infrastructure and facilities needed to develop our project.

We would also like to thank the staff of Department of Information Science and Engineering and lab-in-charges for their co-operation and suggestions. Finally, we would like to thank all our friends for their help and suggestions without which completing this project would not have been possible.

**ABSTRACT**

Fetching Memes Using Volley Library is a mobile application project that revolves around the generation and sharing of random memes sourced from Reddit, utilizing the Volley for seamless data retrieval. The project aims to provide users with an entertaining and engaging experience by presenting them with a diverse collection of memes at the press of a button and enabling easy sharing to various platforms.

This project developed using Android Studio and Java, with XML utilized for frontend

design.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **SL No** | **Chapters** | **Page No** |
| **1** | **Introduction** |  |
| **1.1 About Android Studio** |
| **1.2 About the Project** |
| **2** | **System Overview** |  |
| **2.1 Problem Definition** |
| **2.2 Key Features and Benefits** |
| **2.3 Service Provided to the User** |
| **3** | **Flow of Control** |  |
|  | **3.1 Flow chart** |
| **4** | **System Requirement & Specification** |  |
| **4.1 Hardware Requirements** |
| **4.2 Software Requirements** |
| **5** | **Implementation** |  |
| **5.1 Importing Required Libraries** |
| **5.2 Implementing Methods** |
| **6** | **Source Code** |  |
| **6.1 XML and JAVA CODE** |
| **7** | **Screen Shots** |  |
| **8** | **Conclusion** |  |
| **9** | **References** |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Description** | **Page No** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**CHAPTER – 1:**

**INTRODUCTION**

**1.1 Introduction to Android Studio**

Android Studio is the official integrated development environment (IDE) for Android app development. It is specifically designed to streamline the process of building, testing, and debugging Android applications. Android Studio provides developers with a comprehensive set of tools, libraries, and features that aid in creating high-quality Android apps.

Java is a widely used programming language for Android app development, and it serves as the primary language for building Android applications in Android Studio. Java offers a robustand object-oriented approach to software development, making it suitable for creating complex and scalable Android apps

**1.2 Introduction to Fetching Meme Sharing Using Volley**

Fetching Memes Using Volley Library is a project that focuses on creating an Android application for users to share memes seamlessly by leveraging the capabilities of the Volley . This project aims to provide an engaging and user-friendly experience for meme enthusiasts, allowing them to discover, explore, and share memes with ease.

**1.2.1 Introduction to Volley Library:**

Volley is an open-source networking library developed by Google, specifically designed for efficient communication between Android applications and web services. It simplifies the process of making network requests and handles the management of network connections, caching, and request scheduling. Volley is widely used in Android app development to streamline network operations and improve the overall performance of applications.

**1.2.2 Introduction to Glide Library:** Glide is a popular open-source image loading and caching library for Android.

**CHAPTER - 2:**

**SYSTEM OVERVIEW**

The Fetching Memes using Volley Library project is an Android application designed to provide. Users with a seamless and enjoyable experience in discovering, sharing, and engaging with memes. The system consists of different components working together to facilitate meme browsing, sharing, and social interaction.

**User Interface (UI):**

The user interface component provides an intuitive and user-friendly interface for users to interact with the app. It includes screens for meme browsing, sharing, personalization options, and social engagement features. The UI is designed to be visually appealing, responsive, and easy to navigate.

**Meme Repository:**

The meme repository component manages the collection of memes available in the app.It sources memes from reliable platforms or APIs and stores them in a structured manner. The repository ensures that the meme collection is up to date, diverse, and organized, allowing users to access a wide range of memes.

**Volley Integration:**

The Volley integration component handles the networking aspects of the app. It utilizes the Volley library to efficiently communicate with web services and fetch meme data. The integration component handles network requests, manages connections, and performs data parsing to retrieve memes from the designated sources.

**Meme Sharing:**

The meme sharing component enables users to share memes seamlessly to various platforms.It integrates with the device's sharing functionality or provides custom sharing options within the app. Users can easily share memes with friends, family, or on social media platforms, expanding the reach and impact of the shared content.

**Personalization and Preferences:**

The personalization and preferences component allows users to customize their meme-sharing experience. It includes features such as meme category selection, saving favorite memes, and following specific meme sources. These customization options enhance user engagement and cater to individual preferences.

**Social Engagement:**

The social engagement component promotes user interaction and community building. It includes features like comment sections, likes, and user profiles. Users can engage with each other by commenting on memes, participating in discussions, and forming connections based on shared humor and interests.

**Error Handling and Feedback:**

The error handling and feedback component ensures smooth app operation and user satisfaction. It incorporates error handling mechanisms to handle network errors, notify users of any issues, and provide informative feedback. This component enhances user experience by addressing potential glitches and keeping users informed.

**Data Management and Caching:**

The data management and caching component optimize data storage and retrieval. It manages local caching of meme data to reduce network usage and improve app performance. Caching allows for faster loading times and offline access to previously fetched memes, enhancing the overall user experience. In summary, the Meme Sharing using Volley system integrates various components to provide users with a seamless, personalized, and engaging meme-sharing experience. Through an intuitive user interface, efficient networking, meme repository management, social engagement features, and customization options, the system aims to deliver a delightful and entertaining platform for users to discover, share, and connect through memes.

**2.1 Problem Definition**

The problem this meme sharing app aims to address is the need for a convenient and entertaining way to access and share memes from various sources. Memes have become an integral part of online culture, providing a humorous and relatable form of content that brings joy and laughter to people. However, finding and sharing memes often involves searching through multiple platforms or websites, which can be time-consuming and inefficient.

**2.2 Key Features and Benefits.**

**Seamless Meme Sharing:** Effortlessly share memes to various platforms directly from the app.

**Extensive Meme Collection:** Access a diverse range of memes, including popular and trending ones.

**Efficient Networking:** Optimize network communication for smooth browsing and sharing.

**Error Handling and Feedback:** Robust error handling and informative feedback for a user-friendly experience.

**Customization and User Preferences:** Personalize meme categories and save favorite memes.

**Support for Multiple Platforms:** Share memes to social media, messaging apps, and more.

**Stay Updated:** Get access to the latest meme trends and keep the content fresh.

**Engaging User Experience:** Enhance the meme-sharing experience and foster a sense of community among users.

**2.3 Service Provided to the User.**

**Diverse Meme Collection:** Users have access to a vast and diverse collection of memes from reliable sources. This ensures a constant supply of fresh and entertaining content for users to explore and share.

**Effortless Meme Sharing:** The app simplifies the process of sharing memes by providing a seamless sharing feature. Users can easily share their favorite memes to various platforms directly from within the app, enhancing their ability to engage with others and spread laughter.

**Personalized Meme Experience:** The app offers customization options that allow users to personalize their meme-sharing experience. They can customize meme categories, create personalized meme collections, or follow specific meme sources based on their interests and preferences.

**Real-Time Updates and Trending Memes:** The app keeps users up to date with the latest meme trends and ensures that the meme collection is constantly updated. Users can discover and share the most popular and trending memes, staying in the loop of the meme culture.

**User-Friendly Interface and Smooth Browsing:** The app provides a user-friendly interface that makes browsing and discovering memes a seamless experience. Users can navigate through the app effortlessly, enjoying a smooth and intuitive browsing experience.

**Engaging Community Interaction:** The app fosters a sense of community and interaction among users. Users can engage with each other by sharing memes, commenting on posts, and participating in meme-related discussions, creating a vibrant and engaging community around humor and shared interests.

**Error Handling and Feedback:** The app ensures proper error handling and provides informative feedback to users in case of any issues during the meme-sharing process. This ensures that users are informed and can take necessary actions to resolve any problems.

**Multi-Platform Sharing and Social Integration:** The app supports sharing memes to multiple platforms, including social media networks and messaging app.

**CHAPTER-3:**

**FLOW OF CONTROL**

**3.1 FLOW CHART**

A picture containing diagram, text, screenshot, plan

Description automatically generated

Fig 3.1 Flow Chart

**CHAPTER – 4:**

**SYSTEM REQUIREMENTS AND SPECIFICATION**

**4.1 HARDWARE REQUIREMENTS**

* Windows 8/8.1/[Recommended:10 or higher]
* Android Device [Recommended] or a Android Emulator
* 8GB RAM
* 64 Bit Operating System

**4.2 SOFTWARE REQUIREMENTS**

* Android Studio [IDE]
* Java Development Kit [JDK]
* Android SDK
* Volley Library and Glide Library.

**Internet Connectivity:** The meme sharing app heavily relies on internet connectivity to fetch memes from the API and enable seamless sharing. Therefore, users will require a stable internet connection, either through Wi-Fi or mobile data, to access and use the app effectively.

**CHAPTER – 5:**

**IMPLEMENTATION.**

**5.1 Importing Required Library**

Import the necessary libraries and packages required for the implementation, including the Volley library, Android-specific packages, and JSON-related packages.

**5.2 Implementing Methods**

**onCreate Method**:In the onCreate method, initialize the ActivityMainBinding and set the app's content view.Implement the getMeme() method to fetch and display the first meme.Set click listeners for the "Next" and "Share" buttons.

**getMeme Method:**Within the getMeme() method, specify the URL of the API endpoint from which the memes will be fetched.Show the loader and hide the meme image view to indicate the loading process.

Create a RequestQueue using Volley's newRequestQueue() method.

Create a JsonObjectRequest with a GET request, passing the URL, null for the request body, a response listener, and an error listener.

In the response listener, retrieve the URL of the meme from the JSON response and use Glide library to load the image into the memeImageView.

Hide the loader and show the meme image view.

**shareMeme Method:**The shareMeme() method is called when the "Share" button is clicked.Use the getBitmapFromView() method to convert the memeImageView into a Bitmap object.

Call the shareImageAndText() method and pass the Bitmap object.

**shareImageAndText Method:**The shareImageAndText() method is responsible for creating an Intent and sharing the meme image.

Create a URI for the image using the getImageToShare() method.

Create an Intent with the ACTION\_SEND action.

Add the image URI to the Intent as an extra using the EXTRA\_STREAM key.

Set the MIME type of the content to "image/png".

Start the activity with Intent.createChooser() to allow the user to select an app for sharing.

**getImageToShare Method:**The getImageToShare() method is responsible for saving the meme image to a file and returning the URI.

Create a file directory for storing images using the getCacheDir() method.

Create a File object within the imageFolder directory.

Create a FileOutputStream to write the image data to the file.

Compress the image as PNG and write it to the FileOutputStream.

Close the FileOutputStream.

Use FileProvider.getUriForFile() to get a content URI for the file.

Return the URI.

**getBitmapFromView Method:**The getBitmapFromView() method converts a view into a Bitmap object.

Create a Bitmap object with the dimensions of the view.

Create a Canvas object with the Bitmap.

Draw the view on the canvas.

Return the resulting Bitmap.

**CHAPTER – 6:**

**SOURCE CODE**

**6.1 XML CODE**

<?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="@color/background"

tools:context=".MainActivity">

<TextView

android:id="@+id/title"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp"

android:text="MemeGaalata"

android:textColor="@color/white"

android:textSize="26sp"

android:textStyle="bold"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

/>

<ImageView

android:id="@+id/meme\_image"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_margin="16dp"

app:layout\_constraintBottom\_toTopOf="@+id/share"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/title" />

<ProgressBar

android:id="@+id/loader"

android:layout\_width="80dp"

android:layout\_height="80dp"

android:indeterminate="true"

android:indeterminateTintMode="src\_atop"

android:indeterminateTint="@color/purple\_200"

app:layout\_constraintTop\_toTopOf="@+id/meme\_image"

app:layout\_constraintStart\_toStartOf="@+id/meme\_image"

app:layout\_constraintEnd\_toEndOf="@+id/meme\_image"

app:layout\_constraintBottom\_toBottomOf="@+id/meme\_image"

/>

<Button

android:id="@+id/share"

android:layout\_width="0dp"

android:layout\_height="72dp"

android:backgroundTint="@color/button\_color"

android:text="SHARE"

android:textSize="18sp"

android:layout\_margin="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toStartOf="@+id/next"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintHorizontal\_chainStyle="packed"

app:layout\_constraintStart\_toStartOf="parent" />

<Button

android:id="@+id/next"

android:layout\_width="0dp"

android:layout\_height="72dp"

android:backgroundTint="@color/purple\_200"

android:text="NEXT"

android:textSize="18sp"

android:layout\_margin="16dp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toEndOf="@+id/share" />

</androidx.constraintlayout.widget.ConstraintLayout>

**6.2 JAVA CODE**

package com.example.memeapp;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.content.FileProvider;

import android.content.Intent;

import android.graphics.Bitmap;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.drawable.Drawable;

import android.net.Uri;

import android.os.Bundle;

import android.os.FileObserver;

import android.view.View;

import android.widget.Toast;

import com.android.volley.Request;

import com.android.volley.RequestQueue;

import com.android.volley.Response;

import com.android.volley.VolleyError;

import com.android.volley.toolbox.JsonObjectRequest;

import com.android.volley.toolbox.Volley;

import com.bumptech.glide.Glide;

import com.example.memeapp.databinding.ActivityMainBinding;

import org.json.JSONException;

import org.json.JSONObject;

import java.io.File;

import java.io.FileOutputStream;

public class MainActivity extends AppCompatActivity {

ActivityMainBinding binding;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

binding = ActivityMainBinding.inflate(getLayoutInflater());

setContentView(binding.getRoot());

getMeme();

binding.next.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

getMeme();

}

});

binding.share.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

shareMeme();

}

});

}

private void getMeme() {

String url = "https://meme-api.com/gimme/programminghumor";

binding.loader.setVisibility(View.VISIBLE);

binding.memeImage.setVisibility(View.GONE);

RequestQueue que = Volley.newRequestQueue(this);

JsonObjectRequest jsonObjectRequest = new JsonObjectRequest

(Request.Method.GET, url, null, new Response.Listener<JSONObject>() {

@Override

public void onResponse(JSONObject response) {

try {

String imgUrl = response.getString("url");

Glide.with(getApplicationContext())

.load(imgUrl)

.into(binding.memeImage);

binding.loader.setVisibility(View.GONE);

binding.memeImage.setVisibility(View.VISIBLE);

} catch (JSONException e) {

e.printStackTrace();

}

}

}, new Response.ErrorListener() {

@Override

public void onErrorResponse(VolleyError error) {

// TODO: Handle error

Toast.makeText(MainActivity.this, error.getMessage(), Toast.LENGTH\_SHORT).show();

}

});

que.add(jsonObjectRequest);

}

private void shareMeme() {

Bitmap image = getBitmapFromView(binding.memeImage);

shareImageAndText(image);

}

private void shareImageAndText(Bitmap image) {

Uri uri = getImageToShare(image);

Intent intent = new Intent(Intent.ACTION\_SEND);

intent.putExtra(Intent.EXTRA\_STREAM,uri);

intent.setType("image/png");

startActivity(Intent.createChooser(intent, "Share Image Via :"));

}

private Uri getImageToShare(Bitmap image) {

File imageFolder = new File(getCacheDir(),"images");

Uri uri = null;

try{

imageFolder.mkdirs();

File file = new File(imageFolder,"meme.png");

FileOutputStream outputStream = new FileOutputStream(file);

image.compress(Bitmap.CompressFormat.PNG,100,outputStream);

outputStream.flush();

outputStream.close();

uri = FileProvider.getUriForFile(this,"com.example.shareImage.fileProvider",file);

}

catch (Exception e){

Toast.makeText(this, ""+e.getMessage(), Toast.LENGTH\_SHORT).show();

}

return uri;

}

private Bitmap getBitmapFromView(View view){

Bitmap returnedBitmap = Bitmap.createBitmap(view.getWidth(),view.getHeight(),Bitmap.Config.ARGB\_8888);

Canvas canvas = new Canvas(returnedBitmap);

Drawable background = view.getBackground();

if(background != null){

background.draw(canvas);

}else{

canvas.drawColor(Color.WHITE);

}

view.draw(canvas);

return returnedBitmap;

}

}

**CHAPTER – 7:**

**SCREENSHOTS**

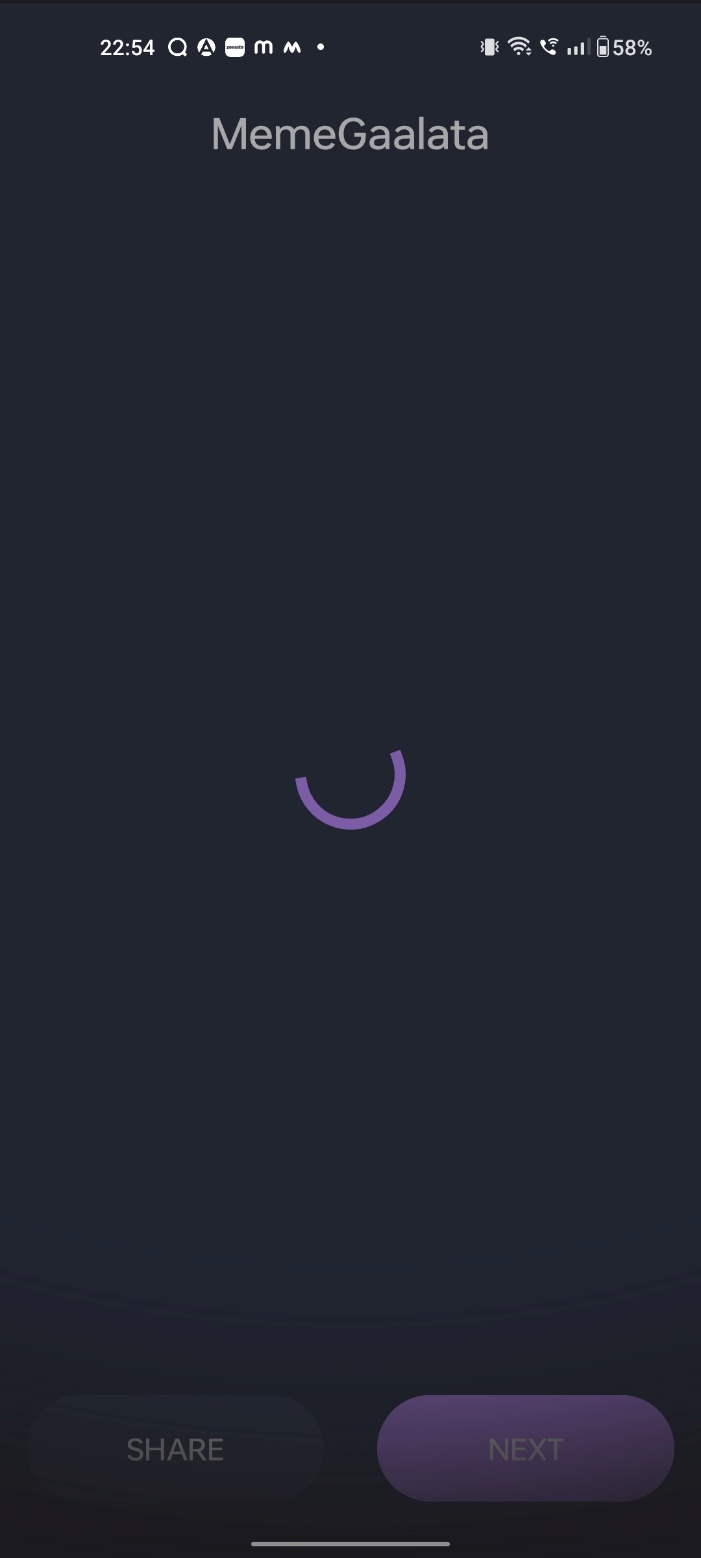
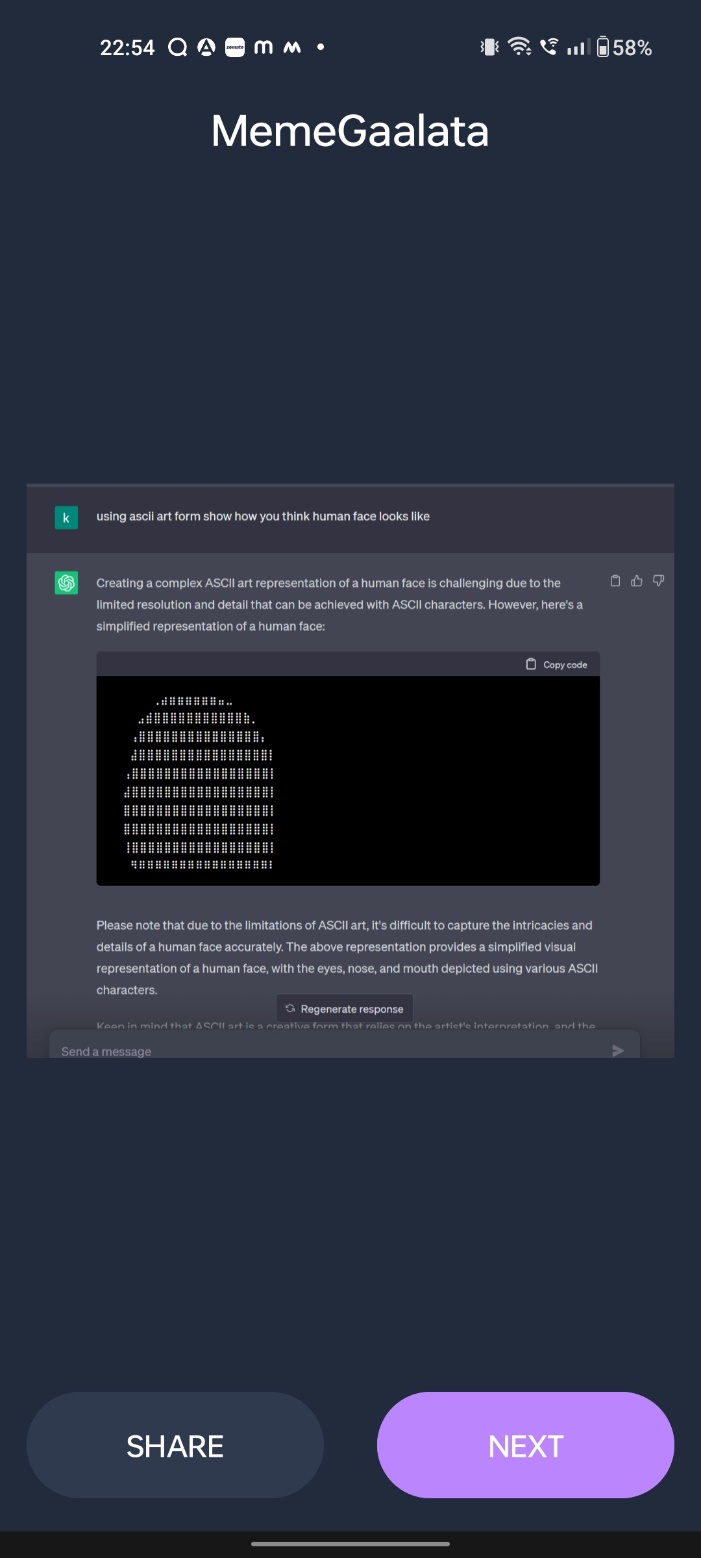
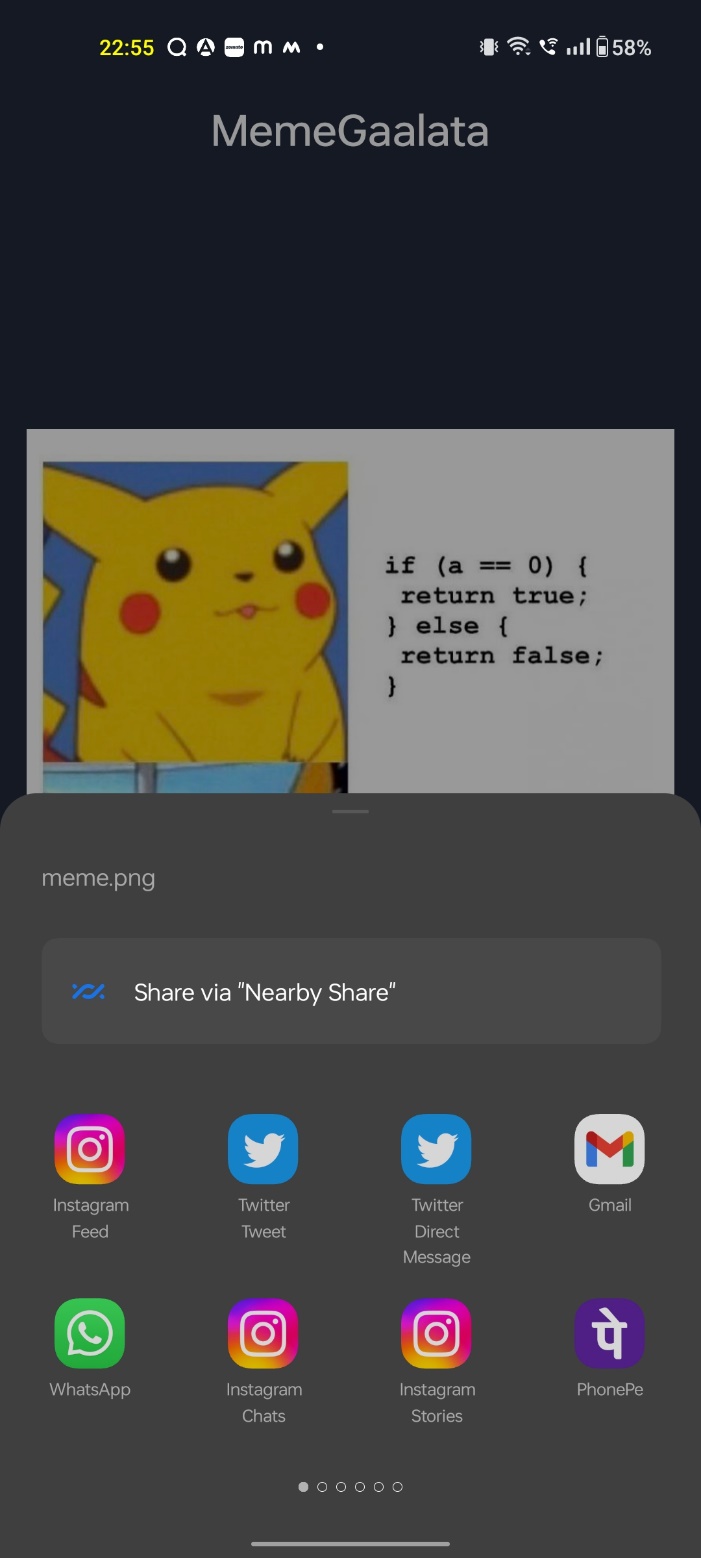
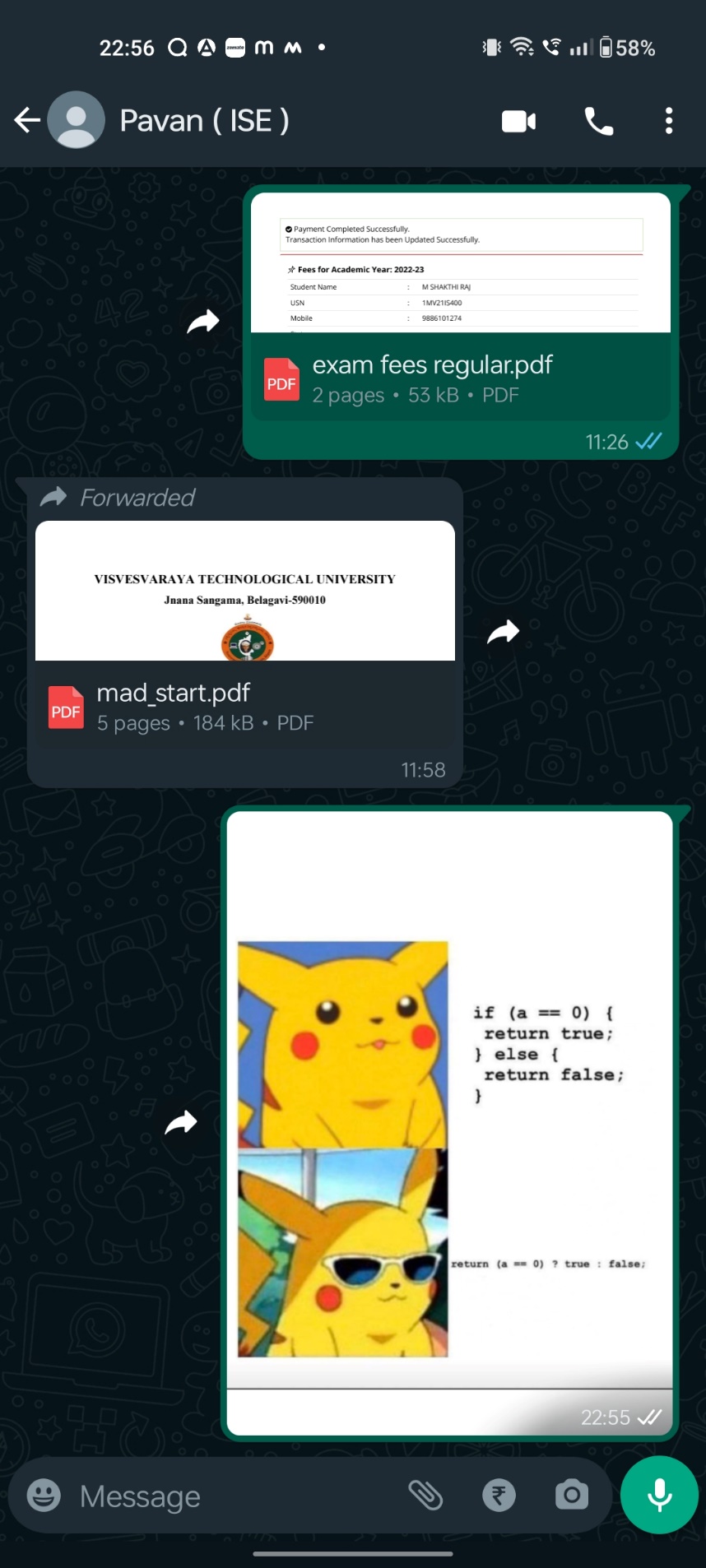


Fig 7.1 Screenshot







**CHAPTER – 8:**

**CONCLUSION**

In conclusion, the meme sharing app developed using Android Studio and Java provides users with a convenient and enjoyable way to access, browse, and share memes. The app leverages the Volley API to fetch random memes from reliable sources like Reddit, ensuring a diverse and up-to-date selection of content.

The key features of the app include a user-friendly interface, the ability to fetch and display memes with the click of a button, and seamless sharing functionality. Users can easily navigate through memes, discover new and trending content, and share their favorite memes with friends and on various platforms.

The app caters to the users' need for humor, stress relief, and a sense of entertainment by providing a wide range of hilarious and relatable memes. It aims to bring laughter and joy to users' lives, serving as a digital companion for those seeking a quick escape or a mood-lifter.

By utilizing the capabilities of Android Studio, Java, and the Volley API, the app delivers a smooth and efficient user experience. The code implementation incorporates network requests, image loading, bitmap manipulation, and sharing functionality to ensure seamless operation and optimal performance.

In a world where memes have become a significant part of online culture, the meme sharing app provides a centralized platform for meme enthusiasts to enjoy,

explore, and share humorous content. It fosters a sense of community, encourages social interactions, and enhances the overall user experience.

With its engaging features and seamless sharing capabilities, the meme sharing app is poised to become a go-to source for meme lovers, providing endless laughter and entertainment at the touch of a button

**CHAPTER – 9:**

**REFERENCES**

**Books:**

**Java Programming for Android Developers for Dummies:**

Author: Burd, B.

Publisher: Wiley

**Fundamentals of Creating a Great UI/UX:**

Author: Creative Tim

Publisher: Creative Tim

**Websites**

**GeeksForGeeks:**

Website: www.geeksforgeeks.org

**TutorialsPoint:**

Website: www.tutorialspoint.com

**w3Schools:**

Website: www.w3schools.com