Mobile Operating System
Innovative Assignment

By: 21BCE103 and 21BCE135

Wedding Planning Application

Why our app?
A wedding planning app is essential because it simplifies the complex and often overwhelming process of organizing a wedding. It provides a centralized platform where couples can manage budgets, track guest lists, schedule tasks, and communicate with vendors, all in one place. By offering tools like budget calculator, checklists, and AI features, it ensures that every detail is accounted for and reduces the stress of planning, allowing couples to focus more on enjoying their special day. Additionally, the app offers convenience and accessibility, making it easier to plan anytime, anywhere.

Flow of our application:

1. User Authentication (Login and Sign-Up)

- Login and Sign-Up: Users land on a page where they can either sign up for a new account or log in to an existing one.
- Input: Username, email, password.
- Technology: Java for logic, XML for UI design.

2. Bride and Groom Details

- Input Form: Enter the names of the bride and groom.
- Use Case: Store these names to personalize the wedding plan throughout the app.

3. Wedding Planning

- Location Selection: Users can select the location from various options (e.g., beach, garden, hall).
- Dress and Suit Options: Choose attire options for both bride and groom.
- Estimated Cost: Display the estimated cost based on the choices made.

4. Catering Services

- Catering Options: Users can select from available caterers.
- Cost Calculation: Enter the number of guests and cost per plate. The app calculates and displays the total catering cost.

5. Invite Generation

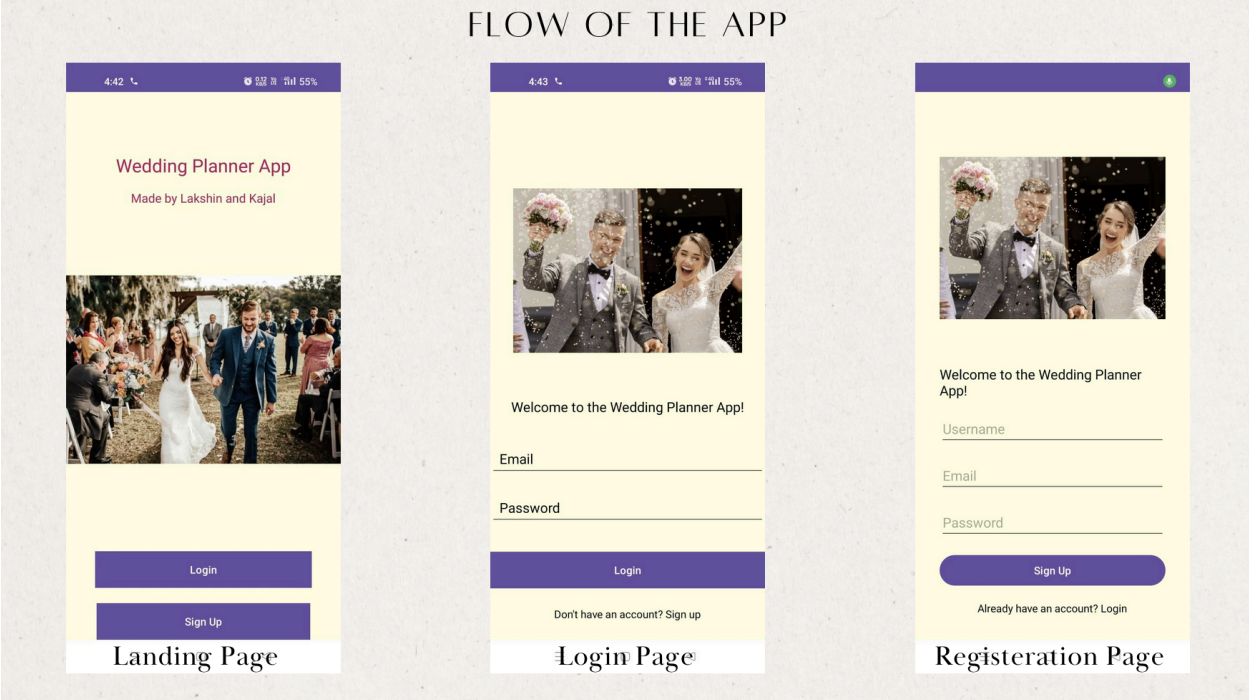
- Photo Upload: Users upload photos of the bride, groom, and guest names.
- PDF Invite: Generate a wedding invite and allow users to download it as a PDF.

6. AI Wedding Suggestions (Gemini Integration)

- Input: Users provide a color scheme and wedding theme.
- AI Suggestions: Using Gemini AI, the app suggests floral arrangements, theme names, and décor ideas based on user inputs.

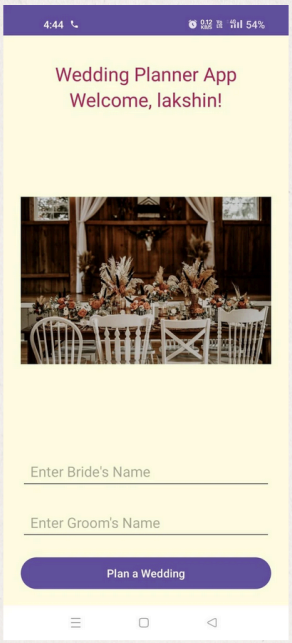
Each of these sections can be implemented independently, but they will come together to provide a seamless experience for the user.

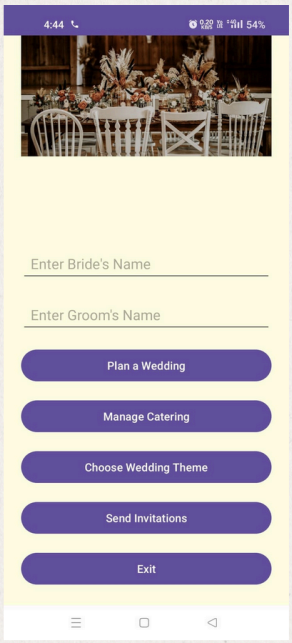
Screenshots:



FLOW OF THE APP

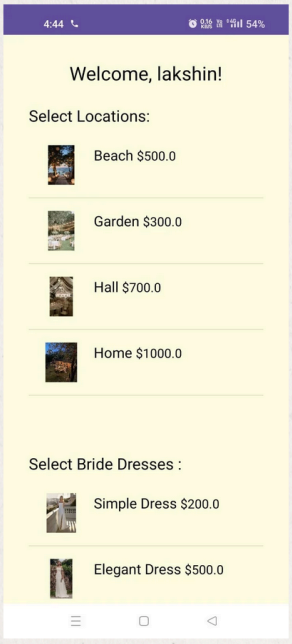
Managing a Wedding

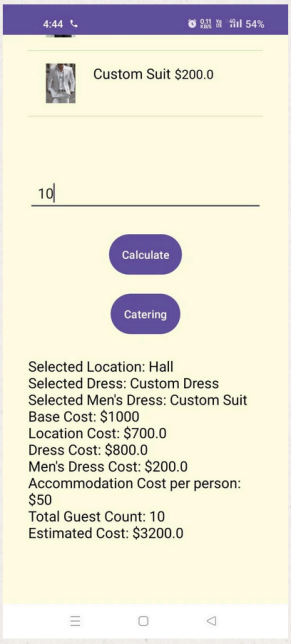




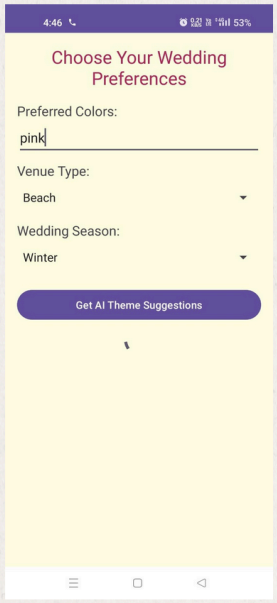
FLOW OF THE APP

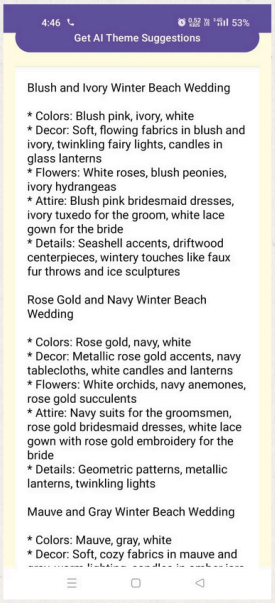
Planning a Wedding



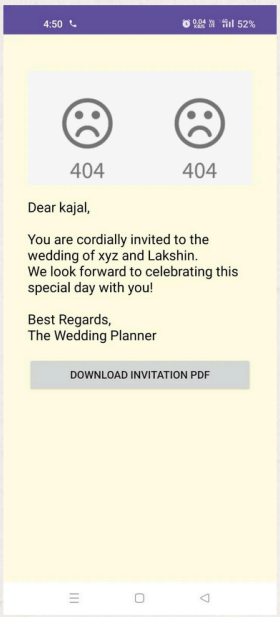


AI Wedding Planning Feature

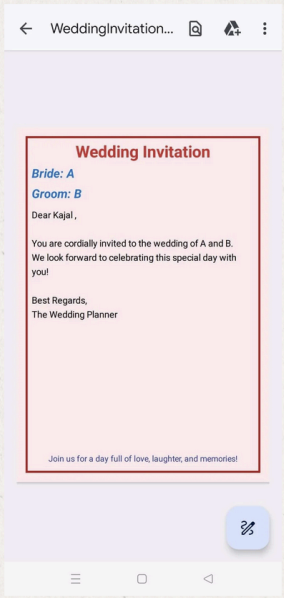




FLOW OF THE APP



Wedding Invitation and it's PDF generation



MainActivity.java

```
package com.example.mos_innovative;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

        Button buttonLogin = findViewById(R.id.buttonLogin);
        Button buttonSignUp = findViewById(R.id.buttonSignUp);

        buttonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, LoginActivity.class);
                startActivity(intent);
            }
        });

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

AndroidManifest.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">

<!-- Add permissions here -->
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<!-- Add any other permissions your app requires -->

<application
    android:allowBackup="true"
    android:dataExtractionRules="@xml/data_extraction_rules"
    android:fullBackupContent="@xml/backup_rules"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/Theme.MOS_Innovative"
    tools:targetApi="31">

    <!-- Main Activity -->
    <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>

    <!-- Login Activity -->
    <activity
        android:name=".LoginActivity"
        android:exported="true" />

    <!-- Sign Up Activity -->
    <activity
        android:name=".SignUpActivity"
        android:exported="true" />

    <!-- Home Activity -->
    <activity
```

```
            android:name=".HomeActivity"
            android:exported="true" />

        <!-- Planner Activity -->
        <activity
            android:name=".PlannerActivity"
            android:exported="true" />

        <!-- Catering Activity -->
        <activity
            android:name=".CateringActivity"
            android:exported="true" />

        <!-- Invitations Activity -->
        <activity
            android:name=".InvitationsActivity"
            android:exported="true" />

        <!-- Wedding Theme Activity -->
        <activity
            android:name=".WeddingThemeActivity"
            android:exported="true" /> <!-- Add this line -->

    </application>

</manifest>
```

HomeActivity.java

```
package com.example.mos_innovative;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class HomeActivity extends AppCompatActivity {

    private static final String PREFS_NAME = "UserPrefs"; // Name of the SharedPreferences file

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

        // Get the username passed from LoginActivity or retrieved from SharedPreferences
        SharedPreferences sharedPreferences = getSharedPreferences(PREFS_NAME,
MODE_PRIVATE);
        String username = sharedPreferences.getString("username", "User"); // Default to "User" if
not found

        // Display the username on the HomeActivity
        TextView welcomeText = findViewById(R.id.welcomeText);
        welcomeText.setText("Welcome, " + username + "!");

        // Initialize EditText fields for bride and groom names
        EditText editTextBrideName = findViewById(R.id.editTextBrideName);
        EditText editTextGroomName = findViewById(R.id.editTextGroomName);

        // Initialize buttons
        Button buttonPlanWedding = findViewById(R.id.buttonPlanWedding);
        Button buttonCatering = findViewById(R.id.buttonCatering);
        Button buttonInvitations = findViewById(R.id.buttonInvitations);
        Button buttonWeddingTheme = findViewById(R.id.buttonWeddingTheme); // New button
        Button buttonExit = findViewById(R.id.buttonExit);

        // Set up onClickListener for Plan Wedding button
        buttonPlanWedding.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Navigate to PlannerActivity
                startActivity(new Intent(HomeActivity.this, PlannerActivity.class));
            }
        });
```

```
        // Set up onClickListener for Catering button
        buttonCatering.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Navigate to CateringActivity
                startActivity(new Intent(HomeActivity.this, CateringActivity.class));
            }
        });

        // Set up onClickListener for Wedding Theme button
        buttonWeddingTheme.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Navigate to WeddingThemeActivity
                startActivity(new Intent(HomeActivity.this, WeddingThemeActivity.class));
            }
        });

        // Set up onClickListener for Invitations button
        buttonInvitations.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get bride's and groom's names from EditText
                String brideName = editTextBrideName.getText().toString().trim();
                String groomName = editTextGroomName.getText().toString().trim();

                // Check if names are not empty
                if (brideName.isEmpty() || groomName.isEmpty()) {
                    Toast.makeText(HomeActivity.this, "Please enter both names",
Toast.LENGTH_SHORT).show();
                    return;
                }

                // Create intent to start InvitationsActivity
                Intent intent = new Intent(HomeActivity.this, InvitationsActivity.class);
                intent.putExtra("BRIDE_NAME", brideName);
                intent.putExtra("GROOM_NAME", groomName);
                startActivity(intent);
            }
        });
```

```
        // Set up onClickListener for Exit button
        buttonExit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(HomeActivity.this, "Logging out...",
Toast.LENGTH_SHORT).show();
                // Finish the activity and go back to LoginActivity
                finish();
            }
        });
    }
}
```

LoginActivity.java

```
package com.example.mos_innovative;

import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class LoginActivity extends AppCompatActivity {

    private static final String PREFS_NAME = "UserPrefs"; // Name of the SharedPreferences file
    private EditText editTextEmail, editTextPassword;
    private Button buttonLogin;
    private TextView textViewSignUp;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_login); // Set the layout for the activity

        // Initialize UI elements
        editTextEmail = findViewById(R.id.editTextEmail);
        editTextPassword = findViewById(R.id.editTextPassword);
        buttonLogin = findViewById(R.id.buttonLogin);
        textViewSignUp = findViewById(R.id.textViewSignUp);

        // Set click listener for the Login button
        buttonLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get the input values
                String email = editTextEmail.getText().toString().trim();
                String password = editTextPassword.getText().toString().trim();

                // Retrieve stored user data from SharedPreferences
                SharedPreferences sharedPreferences = getSharedPreferences(PREFS_NAME,
MODE_PRIVATE);
                String storedEmail = sharedPreferences.getString("email", null);
                String storedPassword = sharedPreferences.getString("password", null);

                // Validate input
                if (email.isEmpty() || password.isEmpty()) {
                    Toast.makeText(LoginActivity.this, "Email and password cannot be empty",
Toast.LENGTH_SHORT).show();
                    return;
                }

                // Check if the input matches the stored data
                if (email.equals(storedEmail) && password.equals(storedPassword)) {
                    Toast.makeText(LoginActivity.this, "Login successful!",
Toast.LENGTH_SHORT).show();
                    Intent intent = new Intent(LoginActivity.this, HomeActivity.class);
                    intent.putExtra("USERNAME", storedEmail); // Pass email to HomeActivity
                    startActivity(intent);
                    finish(); // Close the LoginActivity
                } else {
                    // Display an error message if the login fails
```

```
                Toast.makeText(LoginActivity.this, "Invalid email or password",
Toast.LENGTH_SHORT).show();
            }
        }
    });

    // Set click listener for the Sign Up TextView
    textViewSignUp.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Navigate to signup activity
            startActivity(new Intent(LoginActivity.this, SignUpActivity.class));
        }
    });
  }
}
```

SignUpActivity.java

```
package com.example.mos_innovative;

import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;

public class SignUpActivity extends AppCompatActivity {

  private static final String PREFS_NAME = "UserPrefs"; // Name of the SharedPreferences file

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_signup); // Set the layout for the activity
```

```
        // Initialize UI elements
        EditText editTextUsername = findViewById(R.id.editTextUsername);
        EditText editTextEmail = findViewById(R.id.editTextEmail);
        EditText editTextPassword = findViewById(R.id.editTextPassword);
        Button buttonSignUp = findViewById(R.id.buttonSignUp);
        TextView textViewLogin = findViewById(R.id.textViewLogin);

        // Set click listener for the Sign Up button
        buttonSignUp.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get the input values
                String username = editTextUsername.getText().toString().trim();
                String email = editTextEmail.getText().toString().trim();
                String password = editTextPassword.getText().toString().trim();

                // Save the user data in SharedPreferences
                SharedPreferences sharedPreferences = getSharedPreferences(PREFS_NAME,
MODE_PRIVATE);
                SharedPreferences.Editor editor = sharedPreferences.edit();
                editor.putString("username", username);
                editor.putString("email", email);
                editor.putString("password", password);
                editor.apply(); // Commit the changes

                // Show a Toast message
                Toast.makeText(SignUpActivity.this, "User signed up successfully!",
Toast.LENGTH_SHORT).show();

                // Optionally, navigate to login activity
                startActivity(new Intent(SignUpActivity.this, LoginActivity.class));
                finish(); // Close the SignUpActivity
            }
        });

        // Set click listener for the Login TextView
        textViewLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```
            // Navigate to login activity
            startActivity(new Intent(SignUpActivity.this, LoginActivity.class));
        }
    });
  }
}
```

WeddingThemeActivity.java

```
package com.example.mos_innovative;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ProgressBar;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import com.example.mos_innovative.GeminiPro;
import com.example.mos_innovative.ResponseCallback;

public class WeddingThemeActivity extends AppCompatActivity {

    private EditText inputColors;
    private Spinner inputVenue, inputSeason;
    private TextView themeSuggestionText;
    private Button btnGetTheme;
    private ProgressBar progressBar; // ProgressBar to show loading state

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

        inputColors = findViewById(R.id.inputColors);
        inputVenue = findViewById(R.id.inputVenue);
```

```
        inputSeason = findViewById(R.id.inputSeason);
        themeSuggestionText = findViewById(R.id.themeSuggestionText);
        btnGetTheme = findViewById(R.id.btnGetTheme);
        progressBar = findViewById(R.id.progressBar); // Initialize your ProgressBar

        btnGetTheme.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Get user inputs
                String colors = inputColors.getText().toString();
                String venue = inputVenue.getSelectedItem().toString();
                String season = inputSeason.getSelectedItem().toString();

                // Validate inputs
                if (colors.isEmpty()) {
                    Toast.makeText(WeddingThemeActivity.this, "Please enter your color preferences",
Toast.LENGTH_SHORT).show();
                    return;
                }

                // Call method to get AI theme suggestions
                getAIThemeSuggestions(colors, venue, season);
            }
        });
    }

    // Method to get AI-based theme suggestions using Gemini API
    private void getAIThemeSuggestions(String colors, String venue, String season) {
        // Create query string
        String query = String.format("Suggest wedding themes using colors: %s, venue: %s,
season: %s.", colors, venue, season);

        GeminiPro geminiPro = new GeminiPro();
        progressBar.setVisibility(View.VISIBLE); // Show the progress bar

        // Clear previous suggestion text
        themeSuggestionText.setText("");

        // Call the GeminiPro method to get the response
        geminiPro.getResponse(query, new ResponseCallback() {
```

```
        @Override
        public void onResponse(String response) {
            // Remove asterisks from the response
            response = response.replace("**", ""); // Remove asterisks

            // Display the AI-generated theme suggestion
            themeSuggestionText.setText(response);
            themeSuggestionText.setVisibility(View.VISIBLE);
            progressBar.setVisibility(View.GONE); // Hide the progress bar
        }

        @Override
        public void onError(Throwable throwable) {
            // Handle API failure
            Toast.makeText(WeddingThemeActivity.this, "API call failed: " +
throwable.getMessage(), Toast.LENGTH_LONG).show();
            progressBar.setVisibility(View.GONE); // Hide the progress bar
        }
    });
  }
}
```

InvitationsActivity.java

```
package com.example.mos_innovative;
import android.graphics.drawable.BitmapDrawable;
import android.graphics.Typeface;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.pdf.PdfDocument;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.view.View;
import android.widget.Button;
```

```
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.ContextCompat;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

public class InvitationsActivity extends AppCompatActivity {

    private TextView invitationTextView;
    private EditText guestNameEditText;
    private ImageView brideImageView, groomImageView;
    private String brideName;
    private String groomName;

    private String selectedImageType; // To store whether bride or groom image is selected
    private boolean isBrideImageSelected = false; // Flag to check if bride image is selected
    private boolean isGroomImageSelected = false; // Flag to check if groom image is selected

    private static final int PICK_IMAGE_REQUEST = 1;

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

        // Get bride's and groom's names from intent
        brideName = getIntent().getStringExtra("BRIDE_NAME");
        groomName = getIntent().getStringExtra("GROOM_NAME");

        // Initialize UI elements
        invitationTextView = findViewById(R.id.invitationTextView);
        guestNameEditText = findViewById(R.id.guestNameEditText);
        brideImageView = findViewById(R.id.brideImageView);
        groomImageView = findViewById(R.id.groomImageView);
```

```
        Button sendInvitationButton = findViewById(R.id.sendInvitationButton);
        Button selectBrideImageButton = findViewById(R.id.selectBrideImageButton);
        Button selectGroomImageButton = findViewById(R.id.selectGroomImageButton);

        // Set up onClickListener for send invitation button
        sendInvitationButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String guestName = guestNameEditText.getText().toString();

                // Check if both images are selected
                if (isBrideImageSelected && isGroomImageSelected) {
                    String invitationMessage = generateInvitationMessage(guestName);
                    displayInvitation(invitationMessage);
                } else {
                    Toast.makeText(InvitationsActivity.this, "Please select both images before sending
the invitation.", Toast.LENGTH_SHORT).show();
                }
            }
        });

        // Set up onClickListener for selecting bride image
        selectBrideImageButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectedImageType = "bride"; // Set the selected image type
                openImageChooser();
            }
        });

        // Set up onClickListener for selecting groom image
        selectGroomImageButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                selectedImageType = "groom"; // Set the selected image type
                openImageChooser();
            }
        });
    }
```

```
    // Method to generate invitation message
    private String generateInvitationMessage(String guestName) {
        return "Dear " + guestName + ",\n\n" +
                "You are cordially invited to the wedding of " + brideName + " and " + groomName +
".\n" +
                "We look forward to celebrating this special day with you!\n\n" +
                "Best Regards,\n" +
                "The Wedding Planner";
    }

    // Method to open image chooser
    private void openImageChooser() {
        Intent intent = new Intent(Intent.ACTION_PICK,
MediaStore.Images.Media.EXTERNAL_CONTENT_URI);
        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) {
            Uri imageUri = data.getData();
            if (imageUri != null) {
                // Check which image is being set
                if (selectedImageType.equals("bride")) {
                    brideImageView.setImageURI(imageUri);
                    isBrideImageSelected = true; // Set the flag to true
                } else {
                    groomImageView.setImageURI(imageUri);
                    isGroomImageSelected = true; // Set the flag to true
                }
            }
        }
    }


private void createPdf(String invitationMessage) {
    // Create a new PdfDocument
```

```
    PdfDocument pdfDocument = new PdfDocument();

    // Create a page description with a wider page size for better flexibility
    PdfDocument.PageInfo pageInfo = new PdfDocument.PageInfo.Builder(500, 700, 1).create();

    // Start a new page
    PdfDocument.Page page = pdfDocument.startPage(pageInfo);

    // Get the canvas of the page
    Canvas canvas = page.getCanvas();

    // Set up paint for the background
    Paint backgroundPaint = new Paint();
    backgroundPaint.setColor(Color.parseColor("#FFEBEE")); // Light pinkish background
    canvas.drawRect(0, 0, pageInfo.getPageWidth(), pageInfo.getPageHeight(), backgroundPaint);

    // Set up paint for title text
    Paint titlePaint = new Paint();
    titlePaint.setColor(Color.parseColor("#D32F2F")); // Dark red for title
    titlePaint.setTextSize(32);
    titlePaint.setTypeface(Typeface.create(Typeface.DEFAULT_BOLD, Typeface.BOLD));
    titlePaint.setTextAlign(Paint.Align.CENTER);

    // Draw the title
    canvas.drawText("Wedding Invitation", pageInfo.getPageWidth() / 2, 60, titlePaint);

    // Set up paint for the invitation message
    Paint messagePaint = new Paint();
    messagePaint.setColor(Color.BLACK);
    messagePaint.setTextSize(18);
    messagePaint.setTextAlign(Paint.Align.LEFT);

    // Draw the bride and groom names in a special font style
    Paint coupleNamePaint = new Paint();
    coupleNamePaint.setColor(Color.parseColor("#1976D2")); // Blue color for names
    coupleNamePaint.setTextSize(24);
    coupleNamePaint.setTypeface(Typeface.create(Typeface.DEFAULT,
Typeface.BOLD_ITALIC));

    // Draw the couple names
```

```
    canvas.drawText("Bride: " + brideName, 30, 100, coupleNamePaint);
    canvas.drawText("Groom: " + groomName, 30, 140, coupleNamePaint);

    // Set up the text for the message with word wrap and margins
    float marginLeft = 30;
    float marginRight = pageInfo.getPageWidth() - 30;
    float yPosition = 180; // Start below the couple's names

    // Split the invitation message into lines and wrap text
    String[] lines = invitationMessage.split("\n");
    for (String line : lines) {
        // Handle word wrapping if the text is too wide
        String[] words = line.split(" ");
        StringBuilder currentLine = new StringBuilder();
        for (String word : words) {
            float textWidth = messagePaint.measureText(currentLine + word + " ");
            if (textWidth < (marginRight - marginLeft)) {
                currentLine.append(word).append(" ");
            } else {
                canvas.drawText(currentLine.toString(), marginLeft, yPosition, messagePaint);
                yPosition += messagePaint.getTextSize() + 10; // Add spacing
                currentLine = new StringBuilder(word).append(" ");
            }
        }
        canvas.drawText(currentLine.toString(), marginLeft, yPosition, messagePaint);
        yPosition += messagePaint.getTextSize() + 10; // Add spacing between lines
    }



    // Draw a decorative border
    Paint borderPaint = new Paint();
    borderPaint.setColor(Color.parseColor("#B71C1C")); // Dark red for border
    borderPaint.setStyle(Paint.Style.STROKE);
    borderPaint.setStrokeWidth(4);
    canvas.drawRect(20, 20, pageInfo.getPageWidth() - 20, pageInfo.getPageHeight() - 20,
borderPaint);

    // Draw footer text with a different font and style
    Paint footerPaint = new Paint();
```

```
    footerPaint.setColor(Color.parseColor("#303F9F")); // Dark blue for footer text
    footerPaint.setTextSize(16);
    footerPaint.setTextAlign(Paint.Align.CENTER);
    canvas.drawText("Join us for a day full of love, laughter, and memories!",
pageInfo.getPageWidth() / 2, pageInfo.getPageHeight() - 40, footerPaint);

    // Finish the page
    pdfDocument.finishPage(page);

    // Save the document to device storage
    String directoryPath =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWNLOADS).t
oString();
    File file = new File(directoryPath, "WeddingInvitation.pdf");

    try {
        pdfDocument.writeTo(new FileOutputStream(file));
        Toast.makeText(this, "PDF saved to Downloads folder!", Toast.LENGTH_LONG).show();
    } catch (IOException e) {
        e.printStackTrace();
        Toast.makeText(this, "Error saving PDF: " + e.getMessage(),
Toast.LENGTH_LONG).show();
    }

    // Close the document
    pdfDocument.close();
}

    // Method to display the invitation message and images
    private void displayInvitation(String invitationMessage) {
        // Hide all other UI elements
        guestNameEditText.setVisibility(View.GONE);
        brideImageView.setVisibility(View.VISIBLE);
        groomImageView.setVisibility(View.VISIBLE);
        findViewById(R.id.selectBrideImageButton).setVisibility(View.GONE);
        findViewById(R.id.selectGroomImageButton).setVisibility(View.GONE);
        findViewById(R.id.sendInvitationButton).setVisibility(View.GONE);

        // Set the invitation message
        invitationTextView.setText(invitationMessage);
```

```
        invitationTextView.setVisibility(View.VISIBLE);

        // Show the download PDF button
        Button downloadPdfButton = new Button(this);
        downloadPdfButton.setText("Download Invitation PDF");
        downloadPdfButton.setVisibility(View.VISIBLE);

        // Add the button to the layout
        LinearLayout layout = findViewById(R.id.invitationLayout); // Ensure 'invitationLayout'
exists in your XML
        layout.addView(downloadPdfButton);

        // Set the download PDF button's click listener
        downloadPdfButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                createPdf(invitationMessage);
            }
        });
    }
}
```

CateringActivity.java

```
package com.example.mos_innovative;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class CateringActivity extends AppCompatActivity {

    private EditText guestCountEditText, foodCostEditText;
    private Button calculateCateringButton;
    private TextView totalCostTextView; // Add a TextView to display the total cost
```

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

        guestCountEditText = findViewById(R.id.editTextGuestCount);
        foodCostEditText = findViewById(R.id.editTextFoodCost);
        calculateCateringButton = findViewById(R.id.buttonCalculateCatering);
        totalCostTextView = findViewById(R.id.textViewTotalCost); // Initialize the TextView

        calculateCateringButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String guestCountString = guestCountEditText.getText().toString();
                String foodCostString = foodCostEditText.getText().toString();

                if (guestCountString.isEmpty() || foodCostString.isEmpty()) {
                    Toast.makeText(CateringActivity.this, "Please fill all fields",
Toast.LENGTH_SHORT).show();
                    return;
                }

                int guestCount = Integer.parseInt(guestCountString);
                double foodCost = Double.parseDouble(foodCostString);

                double totalCateringCost = guestCount * foodCost;

                // Set the calculated cost to the TextView
                totalCostTextView.setText("Total Catering Cost: $" + totalCateringCost);
            }
        });
    }
}
```

SpinnerItem.java

```
package com.example.mos_innovative;
```

```
public class SpinnerItem {
    private String name;
    private double cost;
    private int imageResId; // Resource ID for the image

    // Constructor
    public SpinnerItem(String name, double cost, int imageResId) {
        this.name = name;
        this.cost = cost;
        this.imageResId = imageResId;
    }

    // Getter for name
    public String getName() {
        return name;
    }

    // Getter for cost
    public double getCost() {
        return cost;
    }

    // Getter for image resource ID
    public int getImageResId() {
        return imageResId;
    }

    @Override
    public String toString() {
        return name; // This will be shown in the ListView
    }
}
```

ResponseCallback.java

```
package com.example.mos_innovative;

public interface ResponseCallback {
    void onResponse(String response);
    void onError(Throwable throwable);
}
```

CustomAdapter.java

```
package com.example.mos_innovative;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

import java.util.List;

public class CustomAdapter extends ArrayAdapter<SpinnerItem> {

    private Context context;
    private List<SpinnerItem> items;

    public CustomAdapter(Context context, List<SpinnerItem> items) {
        super(context, R.layout.list_item, items); // Use the item layout and the list of items
        this.context = context;
        this.items = items;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        // Use ViewHolder pattern for better performance
        ViewHolder holder;

        if (convertView == null) {
            convertView = LayoutInflater.from(context).inflate(R.layout.list_item, parent, false);
            holder = new ViewHolder();
            holder.imageView = convertView.findViewById(R.id.itemImage);
            holder.textViewName = convertView.findViewById(R.id.itemName);
            holder.textViewCost = convertView.findViewById(R.id.itemCost);
            convertView.setTag(holder); // Store the holder with the view
        } else {
            holder = (ViewHolder) convertView.getTag(); // Reuse the holder
        }
```

```
        // Get the current item
        SpinnerItem currentItem = getItem(position);

        // Set the image and text
        if (currentItem != null) {
            holder.imageView.setImageResource(currentItem.getImageResId());
            holder.textViewName.setText(currentItem.getName());
            holder.textViewCost.setText("$" + currentItem.getCost());
        }

        return convertView;
    }

    // ViewHolder class to hold the views
    static class ViewHolder {
        ImageView imageView;
        TextView textViewName;
        TextView textViewCost;
    }
}
```

GeminiPro.java

```
package com.example.mos_innovative;

import com.google.ai.client.generativeai.GenerativeModel;
import com.google.ai.client.generativeai.java.GenerativeModelFutures;
import com.google.ai.client.generativeai.type.BlockThreshold;
import com.google.ai.client.generativeai.type.Content;
import com.google.ai.client.generativeai.type.GenerateContentResponse;
import com.google.ai.client.generativeai.type.GenerationConfig;
import com.google.ai.client.generativeai.type.HarmCategory;
import com.google.ai.client.generativeai.type.SafetySetting;
import com.google.common.util.concurrent.FutureCallback;
import com.google.common.util.concurrent.Futures;
import com.google.common.util.concurrent.ListenableFuture;
```

```
import java.util.Collections;
import java.util.concurrent.Executor;

public class GeminiPro {
   public void getResponse(String query, ResponseCallback callback) {
      GenerativeModelFutures model = getModel();

      Content content = new Content.Builder().addText(query).build();
      Executor executor = Runnable::run;

      ListenableFuture<GenerateContentResponse> response = model.generateContent(content);
      Futures.addCallback(response, new FutureCallback<GenerateContentResponse>() {
         @Override
         public void onSuccess(GenerateContentResponse result) {
            String resultText = result.getText();
            callback.onResponse(resultText);
         }

         @Override
         public void onFailure(Throwable throwable) {
            throwable.printStackTrace();
            callback.onError(throwable);
         }
      }, executor);

   }
   private GenerativeModelFutures getModel() {
      String apiKey = BuildConfig.apiKey;

      SafetySetting harassmentSafety = new SafetySetting(HarmCategory.HARASSMENT,
            BlockThreshold.ONLY_HIGH);

      GenerationConfig.Builder configBuilder = new GenerationConfig.Builder();
      configBuilder.temperature = 0.9f;
      configBuilder.topK = 16;
      configBuilder.topP = 0.1f;
      GenerationConfig generationConfig = configBuilder.build();

      GenerativeModel gm = new GenerativeModel(
            "gemini-pro",
```

```
            apiKey,
            generationConfig,
            Collections.singletonList(harassmentSafety)
        );

        return GenerativeModelFutures.from(gm);
    }
}
```

BuildConfig.java

```
package com.example.mos_innovative;

public class BuildConfig {
    public static String apiKey = "Enter_Gemini_api_key_here";
}
```