

InternshipApp Documentation

1. Project Overview

InternshipApp is an Android application developed as part of an internship project to demonstrate proficiency in Android development. The app showcases core Android features, including networking with Retrofit, UI components like `RecyclerView` and `CardView`, and navigation between activities. The primary functionality is displaying a list of posts fetched from the JSONPlaceholder API in a `RecyclerView`, with additional screens for user profile, about information, and advanced features.

- **Package Name:** `com.apexplanet.internshipapp`
- **Target SDK:** API 34 (Android 14)
- **Minimum SDK:** API 21 (Android 5.0 Lollipop)
- **Language:** Java
- **Build System:** Gradle (Kotlin DSL, `build.gradle.kts`)
- **IDE:** Android Studio
- **API Endpoint:** `https://jsonplaceholder.typicode.com/posts`

2. Project Objectives

- Fetch and display a list of posts (title and body) from the JSONPlaceholder API using Retrofit.
- Implement a `RecyclerView` with `CardView` items for an aesthetically pleasing UI.
- Provide navigation to multiple activities (`Profile`, `About`, `Advanced Features`) using `CardView`-based buttons.
- Ensure compatibility with modern Android versions (up to API 34) and older devices (down to API 21).
- Use AndroidX libraries and follow best practices for Android development.

3. Project Structure

The project is organized into the standard Android Studio structure, with 27 files across Java source code, XML layouts, and Gradle configuration. Below is the directory structure and key files:

```

internshipapp/
├─ app/
│   └─ src/
│       └─ main/
│           └─ java/
│               └─ com/apexplanet/internshipapp/
│                   └─ AboutActivity.java
│                   └─ AdvancedFeaturesActivity.java
│                   └─ DataActivity.java
│                   └─ FinalProjectActivity.java
│                   └─ PostAdapter.java
│                   └─ ProfileActivity.java
│                   └─ network/
│                       └─ ApiService.java
│                       └─ NetworkManager.java
│                       └─ Post.java
│                   └─ res/
│                       └─ layout/
│                           └─ activity_about.xml
│                           └─ activity_advanced_features.xml
│                           └─ activity_data.xml
│                           └─ activity_final_project.xml
│                           └─ activity_profile.xml
│                           └─ item_post.xml
│                       └─ values/
│                           └─ colors.xml
│                           └─ strings.xml
│                           └─ themes.xml
│                   └─ AndroidManifest.xml
│               └─ build.gradle.kts
├─ build.gradle.kts
├─ settings.gradle.kts
├─ gradle/
│   └─ wrapper/
│       └─ gradle-wrapper.jar
│       └─ gradle-wrapper.properties
├─ gradle.properties
└─ local.properties

```

Key Files

- **Java Classes:**

- `FinalProjectActivity.java`: Main entry point with `CardView` buttons for navigation to other activities.
- `DataActivity.java`: Displays a `RecyclerView` of posts fetched via `Retrofit`.
- `PostAdapter.java`: Custom adapter for the `RecyclerView`, binding `Post` data to `item_post.xml`.
- `ProfileActivity.java`: Displays user profile information.
- `AboutActivity.java`: Shows information about the app.
- `AdvancedFeaturesActivity.java`: Placeholder for advanced features.
- `network/Post.java`: Model class for posts (id, userId, title, body).
- `network/ApiService.java`: `Retrofit` interface for API calls.
- `network/NetworkManager.java`: Singleton for `Retrofit` instance management.

- **XML Layouts:**

- `activity_final_project.xml`: Layout for `FinalProjectActivity` with navigation cards.
- `activity_data.xml`: Layout for `DataActivity` with `RecyclerView`, `ProgressBar`, and `Button`.
- `item_post.xml`: Item layout for `RecyclerView`, using `CardView` with title and body `TextView`s.
- `activity_profile.xml`, `activity_about.xml`, `activity_advanced_features.xml`: Layouts for respective activities.

- **Resources:**
 - `colors.xml`: Defines colors (e.g., `text_primary`, `text_secondary`).
 - `strings.xml`: String resources for the app.
 - `themes.xml`: App theme configuration.
- **Configuration:**
 - `AndroidManifest.xml`: Declares activities, permissions, and app metadata.
 - `app/build.gradle.kts`: Module-level build script with dependencies and Android configuration.
 - `build.gradle.kts`: Project-level build script with repositories and Gradle plugin.
 - `settings.gradle.kts`: Includes the app module.
 - `gradle-wrapper.properties`: Specifies Gradle version (8.0).

4. Setup Instructions

Prerequisites

- **Android Studio**: Latest stable version (e.g., Android Studio Koala or newer).
- **JDK**: Java 8 (configured in Android Studio).
- **Internet Connection**: Required for downloading dependencies and fetching API data.
- **Emulator/Device**: Android device or emulator running API 21 (Android 5.0) or higher.

Installation Steps

1. Clone or Create Project:

- If starting fresh, create a new Android Studio project with package name `com.apexplanet.internshipapp`.
- If using existing files, place all 27 files in the correct directories as outlined above.
- Alternatively, unzip the project folder to `C:\Users\akavi\AndroidStudioProjects\internshipapp`.

2. Configure Gradle Files:

- **Project-level `build.gradle.kts`:**

```
plugins {
    id("com.android.application") version "8.1.0" apply false
}
allprojects {
    repositories {
        google()
        mavenCentral()
    }
}
tasks.register("clean", Delete::class) {
    delete(rootProject.buildDir)
}
```

- Save as `build.gradle.kts` in the project root.
- **Module-level `app/build.gradle.kts`:**

```

plugins {
    id("com.android.application")
}
android {
    namespace = "com.apexplanet.internshipapp"
    compileSdk = 34
    defaultConfig {
        applicationId = "com.apexplanet.internshipapp"
        minSdk = 21
        targetSdk = 34
        versionCode = 1
        versionName = "1.0"
        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            isMinifyEnabled = false
            proguardFiles(getDefaultProguardFile("proguard-android-optimize.txt"), "proguard-rules.pro")
        }
    }
    compileOptions {
        sourceCompatibility = JavaVersion.VERSION_1_8
        targetCompatibility = JavaVersion.VERSION_1_8
    }
}
dependencies {
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.9.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("androidx.recyclerview:recyclerview:1.3.2")
    implementation("androidx.cardview:cardview:1.0.0")
    implementation("com.squareup.retrofit2:retrofit:2.9.0")
    implementation("com.squareup.retrofit2:converter-gson:2.9.0")
    testImplementation("junit:junit:4.13.2")
    androidTestImplementation("androidx.test.ext:junit:1.1.5")
    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")
}

```

- Save as app/build.gradle.kts.

o **Settings:**

```
include(":app")
```

- Save as settings.gradle.kts.

o **Gradle Wrapper:**

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.0-bin.zip
```

- Save as gradle/wrapper/gradle-wrapper.properties.

3. Place Source Files:

- o Place Java files in app/src/main/java/com/apexplanet/internshipapp/ and app/src/main/java/com/apexplanet/internshipapp/network/.
- o Place layout files in app/src/main/res/layout/.
- o Place resource files (colors.xml, strings.xml, themes.xml) in app/src/main/res/values/.

- Place `AndroidManifest.xml` in `app/src/main/`.

4. Sync Project:

- Open the project in Android Studio.
- Click “Sync Project with Gradle Files” (elephant icon or `File > Sync Project with Gradle Files`).
- Ensure an internet connection for dependency downloads.

5. Build and Run:

- Clean the project: `Build > Clean Project`.
- Rebuild: `Build > Rebuild Project`.
- Run on an emulator or device (API 21 or higher) via `Run > Run 'app'`.

5. Functionality

Main Features

1. Home Screen (`FinalProjectActivity`):

- Displays `CardView` buttons for navigation to:
 - `DataActivity` (view posts).
 - `ProfileActivity` (user profile).
 - `AboutActivity` (app info).
 - `AdvancedFeaturesActivity` (placeholder for future features).
- Uses `activity_final_project.xml` for layout.

2. Data Screen (`DataActivity`):

- Features a `RecyclerView` to display posts fetched from `https://jsonplaceholder.typicode.com/posts`.
- Includes a “Load Data” button to trigger the API call and a `ProgressBar` for loading state.
- Uses `PostAdapter` to bind post data to `item_post.xml`.
- Shows a toast message on success (“Data loaded successfully!”) or failure (“Network error”).
- Layout: `activity_data.xml`.

3. Post Adapter (`PostAdapter`):

- Custom `RecyclerView.Adapter` that binds `Post` objects to `item_post.xml`.
- Each post is displayed in a `CardView` with two `TextViews` for title and body.
- Supports dynamic data updates via `setPosts()`.

4. Networking:

- Uses `Retrofit 2.9.0` to fetch posts via `NetworkManager` and `ApiService`.
- `Post` model maps JSON data (`id, userId, title, body`).
- Handles network errors gracefully with toast messages.

5. Other Screens:

- `ProfileActivity`: Displays user profile information (placeholder).
- `AboutActivity`: Shows app details (placeholder).
- `AdvancedFeaturesActivity`: Reserved for future enhancements.

User Flow

1. Launch the app to `FinalProjectActivity`.
2. Click the “Data” card to navigate to `DataActivity`.
3. In `DataActivity`, click “Load Data” to fetch and display posts in the `RecyclerView`.
4. Navigate to `Profile`, `About`, or `Advanced Features` via other cards in `FinalProjectActivity`.

6. Dependencies

The project uses the following dependencies (defined in `app/build.gradle.kts`):

- **AndroidX:**
 - `androidx.appcompat:appcompat:1.6.1`: App compatibility library.
 - `androidx.constraintlayout:constraintlayout:2.1.4`: Flexible layout manager.
 - `androidx.recyclerview:recyclerview:1.3.2`: For RecyclerView in DataActivity.
 - `androidx.cardview:cardview:1.0.0`: For CardView in FinalProjectActivity and PostAdapter.
- **Material Design:**
 - `com.google.android.material:material:1.9.0`: Material components for UI.
- **Networking:**
 - `com.squareup.retrofit2:retrofit:2.9.0`: HTTP client for API calls.
 - `com.squareup.retrofit2:converter-gson:2.9.0`: JSON parsing for Retrofit.
- **Testing:**
 - `junit:junit:4.13.2`: Unit testing.
 - `androidx.test.ext:junit:1.1.5`: Android test framework.
 - `androidx.test.espresso:espresso-core:3.5.1`: UI testing.

7. Key Artifacts

Below are the key files provided or referenced during development, with their artifact IDs:

- **Configuration:**
 - `app/build.gradle.kts` (artifact_id: `c77fd83b-6494-4840-90a2-f054cbaf23bd`): Module build script with namespace and dependencies.
 - `build.gradle.kts` (artifact_id: `6d99c184-e5bd-4535-8d8e-a3cccd6979c3`): Project-level build script.
 - `AndroidManifest.xml` (artifact_id: `1b53db1e-ece1-491c-854a-c20ccde9bb4b`): App manifest with activities and permissions.
- **Java Classes:**
 - `DataActivity.java` (artifact_id: `7b18f3d6-e40e-462d-8c36-9a3e60030244`): Displays posts in RecyclerView.
 - `PostAdapter.java` (artifact_id: `4eb32e58-fd84-449d-ba33-8d74ecdf56ed`): Adapter for RecyclerView.
- **Layouts:**
 - `item_post.xml` (artifact_id: `0e2230a3-aebf-4c07-94a7-9ad8f6a00705`): CardView layout for post items.
 - `activity_data.xml` (artifact_id: `b940edf9-8397-4024-95e1-b00818d4ec00`): Layout for DataActivity.
- **Resources:**
 - `colors.xml` (artifact_id: `650192a6-6686-4f82-9b65-5107afcf006a`): Defines `text_primary` and `text_secondary`.

8. Troubleshooting

Common Issues and Fixes

1. Error: Unresolved reference (recyclerview, cardview, retrofit):

- **Cause:** Incorrect dependency syntax in `app/build.gradle.kts`.
- **Fix:** Ensure dependencies are defined correctly:

```
implementation("androidx.recyclerview:recyclerview:1.3.2")
implementation("androidx.cardview:cardview:1.0.0")
implementation("com.squareup.retrofit2:retrofit:2.9.0")
```

- Sync project: File > Sync Project with Gradle Files.

2. Error: Namespace not specified:

- **Cause:** Missing namespace in app/build.gradle.kts.
- **Fix:** Add to android block:

```
namespace = "com.apexplanet.internshipapp"
```

3. Error: Cannot find symbol: class PostAdapter:

- **Cause:** Missing or misplaced PostAdapter.java.
- **Fix:**
 - Ensure PostAdapter.java is in app/src/main/java/com/apexplanet/internshipapp/.
 - Verify package: package com.apexplanet.internshipapp;.
 - Sync and rebuild project.

4. Error: Resource not found (e.g., @color/text_primary):

- **Cause:** Missing color definitions in colors.xml.
- **Fix:** Create colors.xml in app/src/main/res/values/:

```
<resources>
    <color name="text_primary">#FF000000</color>
    <color name="text_secondary">#FF666666</color>
</resources>
```

5. Error: Network failure in DataActivity:

- **Cause:** No internet connection or API downtime.
- **Fix:**
 - Ensure device/emulator has internet access.
 - Verify API endpoint: <https://jsonplaceholder.typicode.com/posts>.
 - Check NetworkManager.java for correct base URL.

General Troubleshooting

• Sync Issues:

- Ensure internet connection.
- Invalidate caches: File > Invalidate Caches / Restart.
- Check Gradle version in gradle-wrapper.properties:

```
distributionUrl=https\://services.gradle.org/distributions/gradle-8.0-bin.zip
```

• Build Errors:

- Clean project: Build > Clean Project.
- Rebuild: Build > Rebuild Project.
- Review "Build" tab for specific errors.

• Layout Issues:

- Use Layout Inspector (View > Tool Windows > Layout Inspector) to debug RecyclerView or CardView rendering.
- Verify IDs in item_post.xml match PostAdapter.java.

9. Testing

Test Cases

1. Navigation:

- Verify clicking each `CardView` in `FinalProjectActivity` navigates to the correct activity (`DataActivity`, `ProfileActivity`, etc.).

2. Data Loading:

- In `DataActivity`, click "Load Data" and confirm:
 - `ProgressBar` appears during loading.
 - Posts display in `RecyclerView` with title and body.
 - Toast shows "Data loaded successfully!".

3. Error Handling:

- Disconnect internet, click "Load Data", and verify error toast ("Network error").

4. UI Rendering:

- Ensure `CardView` items in `RecyclerView` render with proper styling (corner radius, elevation).
- Verify text colors in `item_post.xml` (e.g., `?attr/colorPrimary`).

Testing Tools

- **Unit Tests:** Use JUnit (`junit:junit:4.13.2`) for unit tests in `app/src/test/`.
- **UI Tests:** Use Espresso (`androidx.test.espresso:espresso-core:3.5.1`) for UI tests in `app/src/androidTest/`.
- **Emulator/Device:** Test on multiple API levels (API 21, 30, 34) to ensure compatibility.

10. Future Enhancements

- **Offline Support:** Cache posts using Room database for offline access.
- **Pagination:** Implement endless scrolling with Paging 3 library for large datasets.
- **UI Improvements:** Add animations (e.g., `RecyclerView` item animations) and custom themes.
- **Advanced Features:** Expand `AdvancedFeaturesActivity` with features like push notifications or local storage.
- **Testing:** Add comprehensive unit and UI tests for all activities.

11. Deployment

To deploy the app:

1. Generate a signed APK:
 - Build > Generate Signed Bundle/APK.
 - Create a keystore and sign the APK.
2. Test the APK on a device.
3. Distribute via Google Play Store or sideloading.

12. Notes

- **Project Evolution:** The project faced issues with Gradle configuration (`build.gradle.kts`), `namespace`, `RecyclerView`, `CardView`, `Retrofit`, and `PostAdapter`. All were resolved by providing correct files and configurations.
- **ZIP File:** To share the project, create a ZIP file containing all 27 files in their respective directories. Ensure `app/build.gradle.kts`, `item_post.xml`, and `PostAdapter.java` are included.
- **Documentation Artifact:** This document is provided as an artifact for reference and can be saved as `README.md` in the project root.

13. Contact

For issues or questions, contact the developer or refer to Android documentation:

- Android Developer Guide: developer.android.com (<https://developer.android.com>)
- Retrofit: square.github.io/retrofit (<https://square.github.io/retrofit>)
- JSONPlaceholder: jsonplaceholder.typicode.com (<https://jsonplaceholder.typicode.com>)