



JAI SHRIRAM ENGINEERING COLLEGE



(An Autonomous Institution)

(Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai)

Recognized by UGC & Accredited by NAAC and NBA for CSE & ECE)

Dharapuram Road, Avinashipalayam, Tirupur – 638 660.

Academic Year 2024-2025 (Odd Semester)

NM1026 – ANDRIOD APPLICATION DEVELOPMENT

A FLEXIBLE EMAIL CLIENT APP

PROJECT REPORT

Submitted by

RISHITTHA MJ

711222243044

RITHYA M

711222243045

ROOBIKA R

711222243047

SHARMILA G

711222243050

SOWMIYA K

711222243053

of

BACHELOR OF TECHNOLOGY in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

JAI SHRIRAM ENGINEERING COLLEGE

TIRUPPUR –638 660

DECEMBER -2024

BONAFIDE CERTIFICATE

Certified that this course report “**A FLEXIBLE EMAIL CLIENT APP**” is the Bonafide work of “**RISHITTHA MJ, RITHYA M, ROOBICA R, SHARMILA G, SOWMIYA K**” who carried out the course work under my supervision.

SIGNATURE

Ms. B. Revathi M.E.,
Assistant Professor

COURSE INCHARGE

Artificial Intelligence and Data Science
Jai Shriram Engineering college,
Tirupur -638 660

SIGNATURE

Ms. P. Mallika M.E.,
Assistant Professor [Sr. G]

HEAD OF THE DEPARTMENT

Artificial Intelligence and Data Science
Jai Shriram Engineering college,
Tirupur -638 660

The candidate with the University Register no 711222243044, 711222243045, 711222243047, 711222243050, 711222243053 was examined by us in the project viva voce examination held on.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

A FLEXIBLE EMAIL CLIENT APP

ABSTRACT:

The Flexible Email Client App is designed to make email management easier, faster, and more secure. It brings all your email accounts together in one place and helps you stay organized with features like smart filters, priority inbox, and tagging. The app works offline, so you can access your emails anytime, and it protects your data with strong security measures like encryption and multi-factor authentication. You can personalize the app with different themes, layouts, and connect it to tools like calendars and task managers. It also uses AI to help you write, schedule, and reply to emails quickly. With its simple design and advanced features, this app is perfect for anyone looking for a better way to manage their emails.

INTRODUCTION:

The Flexible Email Client App is a modern solution for managing emails efficiently and securely. It brings all your email accounts into one easy-to-use platform, helping you stay organized and productive. With smart tools like filters, priority inboxes, and tagging, it simplifies email management and saves time. The app is designed for everyone, offering customizable layouts, themes, and integration with tools like calendars and task managers. It also works offline and ensures your data is protected with advanced security features. Whether for work, school, or personal use, the Flexible Email Client App makes handling emails easier and more convenient.

HARDWARE REQUIREMENTS :

DEVELOPMENT SYSTEM

- Processor: Intel Core i5 or equivalent (minimum)
- RAM: 8 GB (minimum)
- Storage: 256 GB SSD (minimum)

TESTING DEVICE

- Android Smartphone or tablet (Android 8.0 or higher)
- Minimum 2 GB RAM and 16 GB internal storage

SOFTWARE REQUIREMENTS:

- Java and Kotlin (for Android development)
- SQL (for database integration)
- Firebase SDK (for authentication and cloud features)

TOOLS USED AND THEIR VERSIONS

- Android Studio (Ladybug)
- Java Development Kit (JDK 11)
- Firebase Real-time Database
- Git

PROJECT OBJECTIVES

1. Unified Email Management:

Enable users to connect and manage multiple email accounts from different providers in a single interface. Support seamless switching between accounts without requiring separate logins. Provide cross-platform compatibility for desktop, mobile, and web applications.

2. Enhanced Organization and Productivity:

The project is implement smart filters to automatically categorize emails based on user-defined rules or AI suggestions. Introduce a priority inbox that highlights important emails and minimizes clutter. Provide tools for quick email actions, such as one-click replies, snoozing, and archiving. Include powerful search functionality for easy retrieval of emails using keywords, filters, and advanced queries.

3. Customizability and Integration:

Offer customizable themes and layouts to suit individual preferences and workflows. Allow integration with third-party productivity tools like Google Calendar, Microsoft Teams, or Slack for seamless scheduling and collaboration. Support plugging extensions to add new functionalities, such as task management or cloud storage access. Provide a dashboard view for consolidating emails, tasks, and appointments in one place.

4. Provide Secure and Scalable Data Handling:

Incorporate end-to-end encryption to ensure email content is secure during transmission and storage. Use multi-factor authentication to prevent unauthorized access to accounts. Implement real-time anti-phishing and spam detection to safeguard users from malicious attacks.

CORE FUNCTIONALITIES OF THE PROJECT

➤ Multi-Account Management:

Connect and manage multiple email accounts (e.g., Gmail, Outlook, Yahoo) within a single platform. Provide easy switching between accounts without logging in separately.

➤ Email Organization:

To ensure Smart filters for automatic categorization of emails (e.g., promotions, social, primary). Priority inbox to highlight important messages. Custom tagging and labeling for personalized organization. Quick actions like snoozing, archiving, and marking emails as read or unread.

➤ AI-Powered Assistance:

Suggest email responses based on content using natural language processing (NLP). Enable smart scheduling for sending emails at optimal times. Predictive text suggestions to compose emails faster.

➤ Personalization and Customization:

Custom themes, layouts, and inbox views tailored to user preferences. Widget integration for adding third-party tools like task managers and calendars. Adjustable notification settings for email alerts based on priorities or categories. Fortify for Diabetic survey.

➤ Enhanced Security and Privacy:

Access previously downloaded emails and drafts without an internet connection. Automatically sync emails across devices when reconnected to the internet. Advanced search functionality with keyword filters, tags, and sender/recipient fields. Saved searches for recurring queries. Provide a consistent experience across desktop, mobile, and web apps. Ensure compatibility with major operating systems like Windows, macOS, iOS, and Android. Intuitive design for easy navigation and management. Context-aware menus and drag-and-drop functionality for improved

SUMMARY OF THE DEVELOPMENT

The development of the Flexible Email Client App is a comprehensive process aimed at delivering a cutting-edge, user-friendly, and secure email management solution. The project begins with extensive research and user requirement analysis to identify pain points and define key features such as multi-account management, advanced filtering, and personalized customization. The design phase focuses on creating an intuitive interface with responsive layouts and user-centric workflows, incorporating feedback from prototypes and usability testing. During development, modern technologies and frameworks are employed to ensure efficiency, scalability, and cross-platform compatibility for web, desktop, and mobile versions. Advanced features, including AI-powered tools for composing, sorting, and scheduling emails, as well as integrations with calendars, task managers, and cloud storage, are implemented to enhance productivity. Robust security measures like end-to-end encryption, multi-factor authentication, and anti-phishing tools are integrated to protect user data. Rigorous testing phases, including functional, performance, and security testing, ensure the app's reliability and efficiency under various conditions.

CODING

➤ Dependencies used

```
androidx.core:core-ktx:1.15.0
androidx.lifecycle:lifecycle-runtimektx:2.8.7
androidx.activity:activity-compose:1.9.3
androidx.compose.ui:ui
androidx.compose.ui:ui-tooling preview
androidx.room:room-common:2.6.1
androidx.room:room-ktx:2.6.1
com.google.firebase:firebase-auth:23.1.0
junit:junit:4.13.2
androidx.test.ext:junit:1.2.1
androidx.test.espresso:espresso-core:3.6.1
androidx.compose.ui:ui-test-junit4
androidx.compose.ui:ui-tooling
androidx.compose.ui:ui-test manifest
```

➤ **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" >

    <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:supportRtl="true"
        android:theme="@style/Theme.EmailApplication"
        tools:targetApi="31" >
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="@string/title_activity_register"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".MainActivity"
            android:exported="false"
            android:label="MainActivity"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".ViewMailActivity"
            android:exported="false"
            android:label="@string/title_activity_view_mail"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".SendMailActivity"
            android:exported="false"
            android:label="@string/title_activity_send_mail"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.EmailApplication" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

➤ **Build.gradle(project)**

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}
android {
    namespace 'com.example.emailapplication'
    compileSdk 35
    defaultConfig {
        applicationId "com.example.emailapplication"
        minSdk 21
        targetSdk 35
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        vectorDrawables {
            useSupportLibrary true
        }
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    kotlinOptions {
        jvmTarget = '1.8'
    }
    buildFeatures {
        compose true
    }
    composeOptions {
        kotlinCompilerExtensionVersion '1.2.0'
    }
    packagingOptions {
        resources {
            excludes += '/META-INF/{AL2.0,LGPL2.1}'
        }
    }
}
dependencies {
    implementation 'androidx.core:core-ktx:1.15.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.8.7'
    implementation 'androidx.activity:activity-compose:1.9.3'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
```

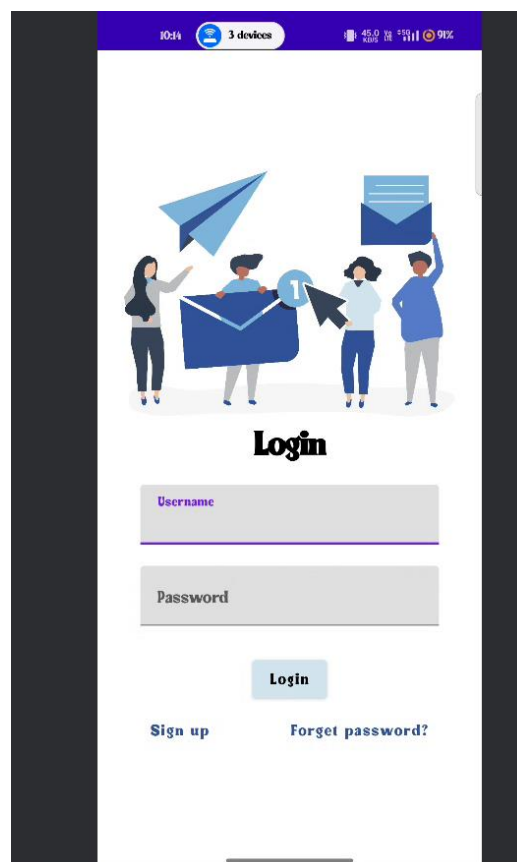


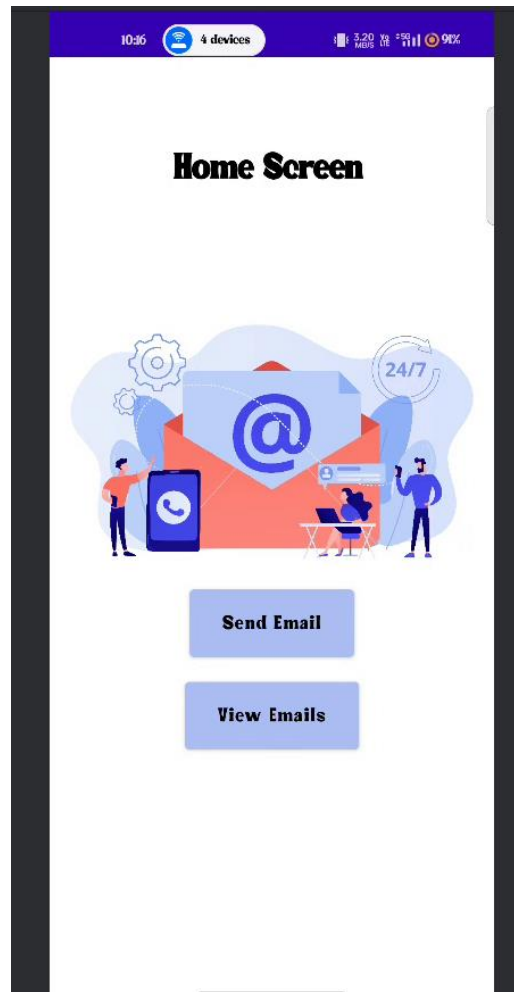
```

implementation 'androidx.room:room-common:2.6.1'
implementation 'androidx.room:room-ktx:2.6.1'
testImplementation 'junit:junit:4.13.2'
androidTestImplementation 'androidx.test.ext:junit:1.2.1'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.6.1'
//noinspection GradleDependency
androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_ui_version"
debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"
}
targetCompatibility JavaVersion.VERSION_1_8
}
kotlinOptions { jvmTarget = '1.8'
}
buildFeatures { compose true
}
composeOptions { kotlinCompilerExtensionVersion '1.2.0'
}
packagingOptions { resources {
excludes += '/META-INF/{AL2.0,LGPL2.1}'
}
}
}

```

SCREENSHOTS:





CONCLUSION:

In conclusion, The Flexible Email Client App successfully addresses the challenges of modern email management by providing a unified, secure, and user-friendly platform. Through its innovative features, including multi-account integration, AI-powered tools, advanced organization options, and robust security measures, the app streamlines email communication and boosts productivity for users. Its customizable interface and seamless integration with third-party tools further enhance user experience, making it adaptable to diverse needs. The development process, from meticulous planning and design to rigorous testing and deployment, ensures the app's reliability, scalability, and alignment with user expectations. Regular updates and maintenance keep the app secure, relevant, and equipped with the latest features to meet evolving demands.