FINANCE MANAGEMENT APP : MONEY MATTERS

OVERVIEW

The Finance Management App is an Android application designed to help users track their personal finances and manage their expenses efficiently. It enables users to log daily expenditures, assign categories (like groceries, utilities, dining, and transportation), and gain a comprehensive view of their spending patterns. The app's interface is intuitive, making it easy for users to add, edit, and delete expense entries. By leveraging local storage through SQLite or Room databases, the app allows users to access their data offline while ensuring security and privacy. Additionally, it includes features like data visualization, where users can generate graphical reports (such as pie charts and bar graphs) to better understand their financial habits over time. The app also supports secure user authentication to protect sensitive data. With functionalities like expense filtering, report generation, and potentially data export options, this app serves as a reliable tool for budgeting and managing personal finances, catering to both casual users and those with more detailed financial tracking needs.

OBJECTIVE

The primary goal of this project is to design and develop a Finance Management App that helps users efficiently track and manage their expenses. The app aims to provide a seamless and intuitive way to log daily expenditures, categorize spending, and monitor financial health over time. This application is particularly useful for individuals who wish to maintain a budget, avoid overspending, and gain insights into their financial habits.

FEATURES

- User authentication: Secure user registration and login with encrypted passwords to protect user data
- Expense management: Users can record expenses with descriptions, amounts, dates, and categories
- Categorization: Users can organize expenses into categories like groceries, travel, dining, healthcare, etc., for better clarity
- Expense search and filtering: Users can filter expenses based on dates, categories, or amounts to review specific spending
- Data visualization: The app generates visual reports (e.g., pie charts, bar graphs) that illustrate spending habits
- Local storage (offline access): The app uses a local database to store user data securely, allowing offline access to expense records.
- Export data: Optionally, users may be able to export their expense data as CSV files for external analysis.

FUNCTIONALITIES

- Registration and login system: Users can create accounts with their email and password. Authentication ensures data privacy.
- Add new expense entries by specifying details like amount, category, date, and description.
- Edit or update existing entries if needed.
- Delete incorrect or unnecessary records.

- Display all expenses in a list with options to sort by date, category, or amount.
- Graphical representation of expenses to provide a clear overview.
- Reports and analysis: Generate monthly or custom reports that highlight spending patterns to help users identify areas where they can cut costs.

TESTING & VALIDATION

- Unit testing: Validate individual modules (like user authentication, expense entry, and editing) to ensure they work correctly.
- UI/UX testing: Confirm that the app's interface is user friendly and free of navigation bugs.
- Integration testing: Ensure that the components interact smoothly. For example, the expense entry module should seamlessly integrate with the database and reporting sections.
- Performance testing: Check the app's performance, especially when handling large amounts of data or using graphical reports.
- Security testing: Ensure user data is protected, especially during the login process and data storage.

DETAILED STEPS FOR IMPLEMENTATION

Step 1: Project setupDownload and install Android StudioOpen the project and sync dependencies using build.gradle.

Step 2: Database configurationSet up SQLite or Room database for efficient storage and retrieval of dataCreate tables for users, expenses, and categories.

Step 3: User authentication module Implement registration, login, and password encryption using secure methods.

Step 4: Expense management module Develop forms for adding and editing expense entries Integrate functionalities for viewing and filtering expenses.

Step 5: Reporting and analytics Create graphical reports using libraries (like MP AndroidChart) to visualize data.

Step 6: Testing and debugging Conduct thorough testing on different Android devices to ensure compatibility Fix any identified bugs and optimize the code.

Step 7: Final deployment Generate a release build and deploy the app to the Google Play Store or distribute it as an APK file.

ADDITIONAL REQUIREMENTS

- Android Studio (latest version recommended).
- JDK (Java Development Kit).
- An Android device or emulator for testing.
- AndroidX libraries for compatibility.
- Libraries for graphical charts (like MPAndroidChart).
- Authentication dependencies if using Firebase or a similar service.

CODE (AndroidManifest.xml)

```
<?xml version="1.0" encoding="utf  :8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.expensestracker"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses  :sdk
        android:minSdkVersion="21"
        android:targetSdkVersion="33" />

    <application
        android:allowBackup="true"

android:appComponentFactory="androidx.core.app.CoreComponentF
actory"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:debuggable="true"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="MONEY MATTERS"
        android:supportsRtl="true"
        android:testOnly="true"
        android:theme="@style/Theme.ExpensesTracker" >
```

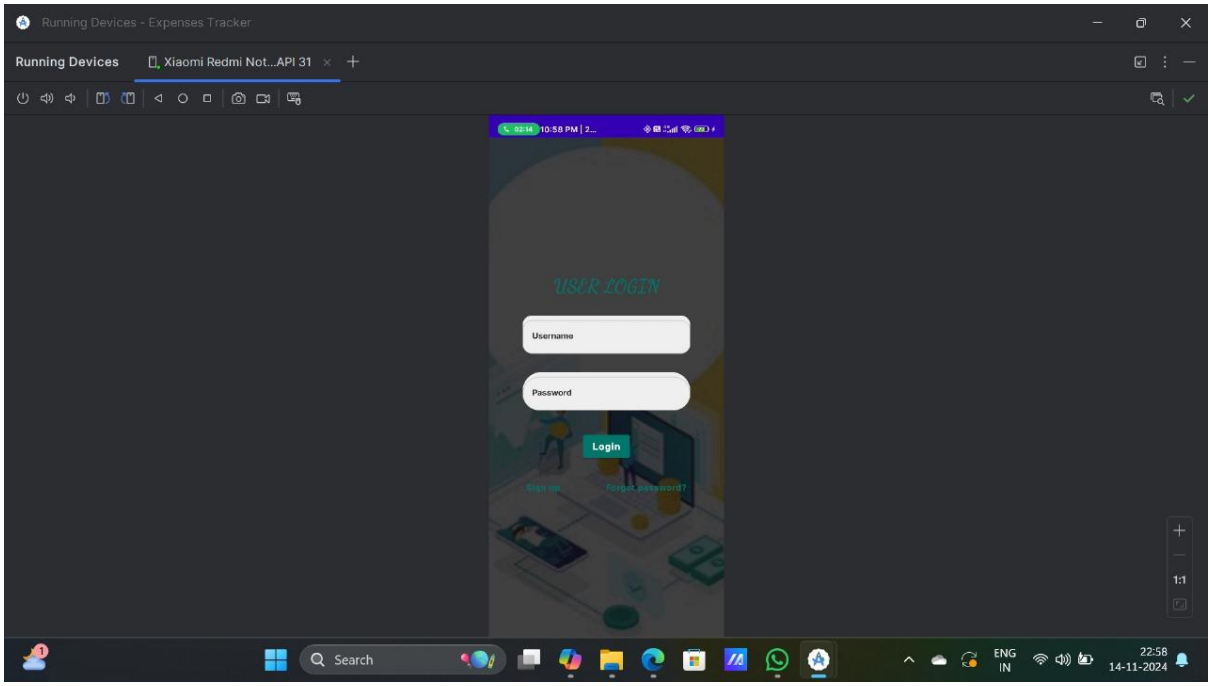
```
        <activity

android:name="com.example.expensestracker.RegisterActivity"
            android:exported="false"
            android:label="@string/title_activity_register"
            android:theme="@style/Theme.ExpensesTracker" />
        <activity
            android:name="com.example.expensestracker.MainActivity"
            android:exported="false"
            android:label="MainActivity"
            android:theme="@style/Theme.ExpensesTracker" />
        <activity

android:name="com.example.expensestracker.ViewRecordsActivity"
            android:exported="false"
            android:label="@string/title_activity_view_records"
            android:theme="@style/Theme.ExpensesTracker" />
        <activity

android:name="com.example.expensestracker.SetLimitActivity"
            android:exported="false"
            android:label="@string/title_activity_set_limit"
            android:theme="@style/Theme.ExpensesTracker" />
        <activity
```

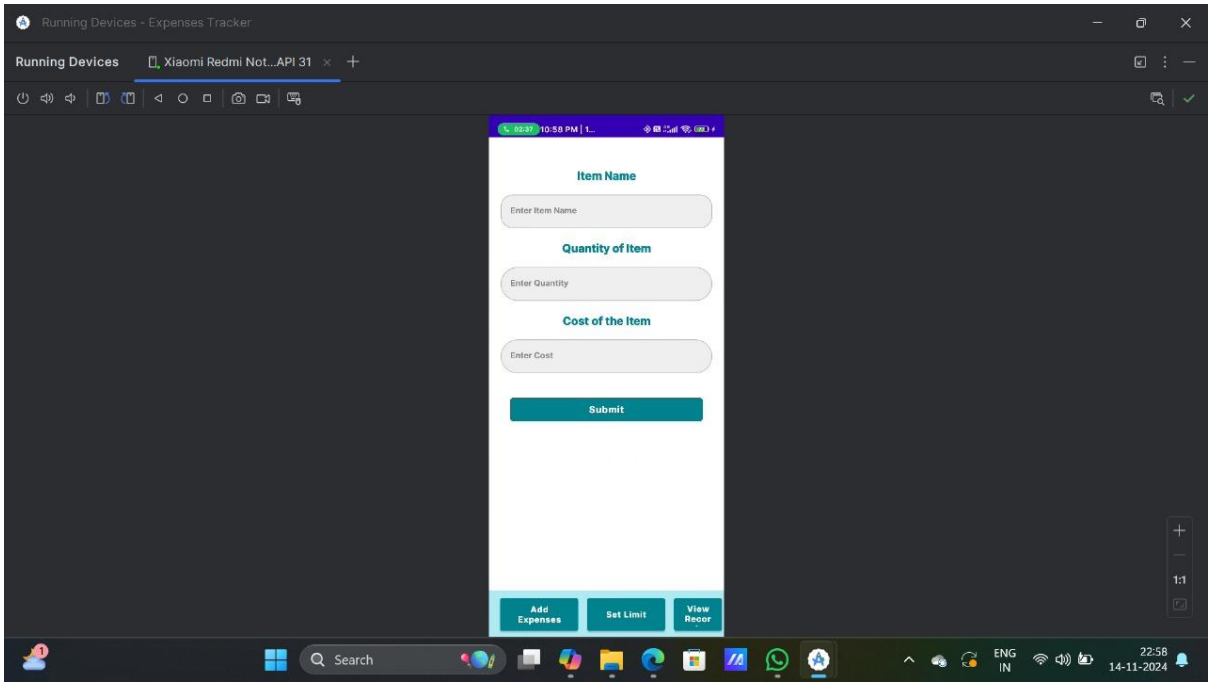
```
android:name="com.example.expensestracker.AddExpensesActivity"
            android:exported="false"
            android:label="@string/title_activity_add_expenses"
            android:theme="@style/Theme.ExpensesTracker" />
        <activity
            android:name="com.example.expensestracker.LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.ExpensesTracker" >
            <intent  :filter>
                <action android:name="android.intent.action.MAIN" />

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

        <service

android:name="androidx.room.MultiInstanceInvalidationService"
            android:directBootAware="true"
            android:exported="false" />

        <activity
```

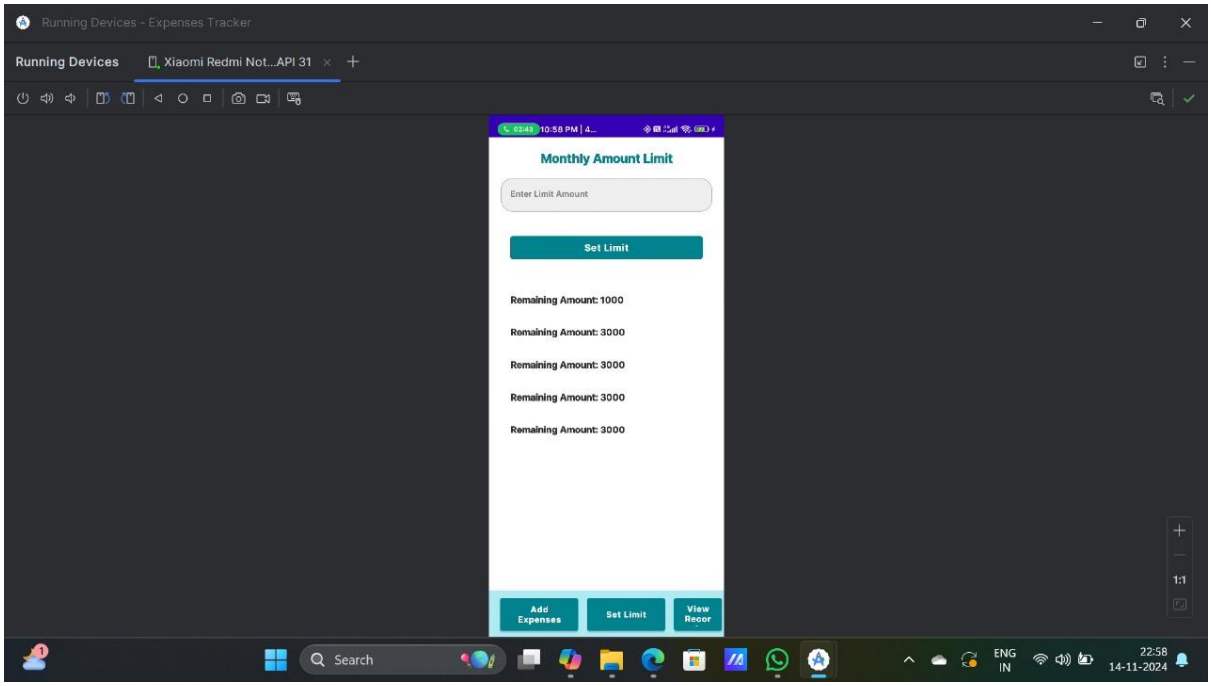
```
            android:name="androidx.compose.ui.tooling.PreviewActivity"
            android:exported="true" />
        <activity
            android:name="androidx.activity.ComponentActivity"
            android:exported="true" />

        <provider
            android:name="androidx.startup.InitializationProvider"

android:authorities="com.example.expensestracker.androidx  :startup"
            android:exported="false" >
            <meta  :data

android:name="androidx.profileinstaller.ProfileInstallerInitializer"
                android:value="androidx.startup" />
        </provider>

        <receiver

android:name="androidx.profileinstaller.ProfileInstallReceiver"
            android:directBootAware="false"
            android:enabled="true"
            android:exported="true"
            android:permission="android.permission.DUMP" >
            <intent  :filter>
```

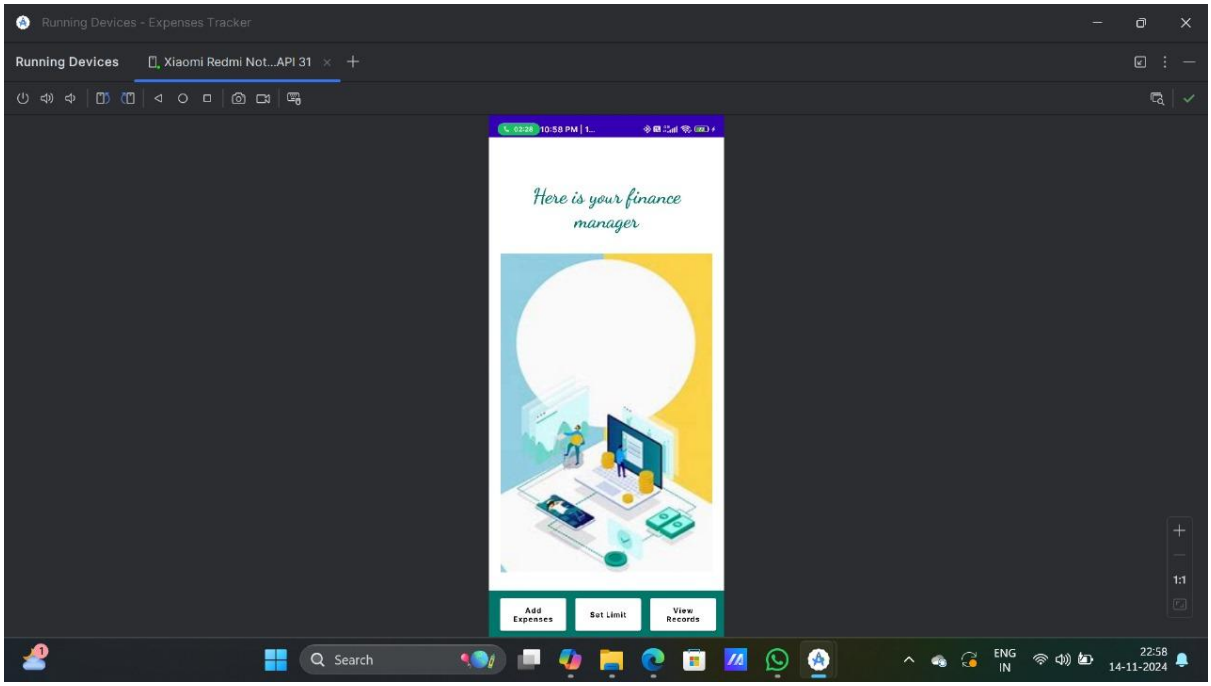
```
            <action
android:name="androidx.profileinstaller.action.INSTALL_PROFILE"
/>
        </intent  :filter>
        <intent  :filter>
            <action
android:name="androidx.profileinstaller.action.SKIP_FILE" />
        </intent  :filter>
      </receiver>
   </application>

</manifest>
```

OUTPUT:


Running Devices - Expenses Tracker
Running Devices
Xiaomi Redmi Not...API 31
USER LOGIN
Username
Password
Login
Sign up
Forgot password?
Search
ENG
IN
22:58
14-11-2024


Running Devices - Expenses Tracker
Running Devices
Xiaomi Redmi Not...API 31
Item Name
Enter Item Name
Quantity of Item
Enter Quantity
Cost of the Item
Enter Cost
Submit
Add Expenses
Set Limit
View Recor
Search
ENG
IN
22:58
14-11-2024


Running Devices - Expenses Tracker
Running Devices
Xiaomi Redmi Not...API 31
10:58 PM | 4...
Monthly Amount Limit
Enter Limit Amount
Set Limit
Remaining Amount: 1000
Remaining Amount: 3000
Remaining Amount: 3000
Remaining Amount: 3000
Remaining Amount: 3000
Add Expenses
Set Limit
View Recor
Search
ENG IN
22:58 14-11-2024


Running Devices - Expenses Tracker
Running Devices
Xiaomi Redmi Not...API 31
10:58 PM | 1...
Here is your finance manager
Add Expenses
Set Limit
View Records
Search
ENG IN
22:58 14-11-2024

DEMO VIDEO:

https://drive.google.com/file/d/15ueVJ_zWUYJK4Jk5mHxIL2j77GlWON9/view?usp=sharing

GITHUB LINK:

https://github.com/Madhu-SenthilKumar/MONEY-MATTERS.git

CONCLUSION:

Developing an expense tracker app in Android Studio provided a practical opportunity to explore the core concepts of mobile app development, including user interface design, database management, and seamless user interaction. The app effectively allows users to track their daily expenses, manage budgets, and gain insights into their spending habits, promoting better financial planning. By utilizing Android Studio, we leveraged tools and features like SQLite databases, RecyclerView, and Material Design components to create an intuitive and efficient application.This project not only enhanced my technical skills but also demonstrated the real-world impact of mobile applications in simplifying everyday tasks. Moving forward, additional features like cloud synchronization, data export options, and personalized analytics could further enhance the app's functionality, making it even more robust and user-friendly.