# STUDY PLANNER APP

**A MINI PROJECT REPORT**

***Submitted by***

**LITHAN G (220701143)**

***in partial fulfillment for the course***

**CS19611 – MOBILE APPLICATION DEVELOPMENT LABORATORY**

***of the degree of***

**BACHELOR OF ENGINEERING**

**in**

**COMPUTER SCIENCE AND ENGINEERING**



**RAJALAKSHMI ENGINEERING COLLEGE**

**RAJALAKSHMI NAGAR**

**THANDALAM**

**CHENNAI – 602 105**

**MAY 2025**

# BONAFIDE CERTIFICATE

Certified that this Report titled **"Study Planner App"** is the bonafide work of **LITHAN G (2116220701143)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.


**SIGNATURE**

**Dr. P. Kumar M.E., Ph.D.**
Head of the Department
Professor
Department of Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai – 602105

**SIGNATURE**

**Mr. Karthik V, M.E.**
Supervisor
Assistant Professor
Department of Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai – 602105


Submitted to Project Viva-Voce Examination held on _____

**Internal Examiner** **External Examiner**

## ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S.Meganathan, B.E, F.I.E.,** our Vice Chairman **Mr. Abhay Meganathan, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) Thangam Meganathan, Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N.Murugesan, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P.Kumar, M.E., Ph.D.,** Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. Karthik V, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, **Mr. B.Bhuvaneswaran, M.E.,** Assistant Professor (SG), Department of Computer Science and Engineering for his useful tips during our review to build our project.

**LITHAN G (220701143)**

# ABSTRACT

The Smart Study App is an Android-based application designed to enhance students' academic productivity by streamlining study schedule management. Leveraging Jetpack Compose for a modern user interface, Room Library for efficient data persistence, and MVVM architecture for scalability, the app provides a robust platform for organizing study-related activities. Key features include a dashboard for tracking progress and upcoming tasks, subject and task management for structured goal-setting, and study session scheduling to foster consistent learning routines. Targeted at students seeking an intuitive tool to optimize their study habits, the app supports Android devices with a minimum SDK version 21, ensuring broad accessibility. This project demonstrates the integration of contemporary Android development practices to deliver a user-centric, efficient, and maintainable study management solution.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1
# INTRODUCTION

## 1.1 GENERAL

In the dynamic landscape of modern education, students juggle numerous academic responsibilities, including managing subjects, assignments, and study schedules. The Smart Study App emerges as a vital tool to address these challenges, offering an intuitive and efficient platform to enhance academic productivity. Built using advanced Android development technologies, the app leverages Jetpack Compose for a sleek, responsive user interface, Room Library for robust data storage, and MVVM architecture for scalability and maintainability. This combination ensures a seamless experience tailored to the needs of students striving for academic excellence. The app's core features include a dashboard for monitoring study progress, subject management for organizing coursework, task assignment for goal-setting, and study session scheduling to promote consistent learning habits. Designed to support Android devices with a minimum SDK version 21, the Smart Study App ensures broad accessibility, catering to a diverse user base. By simplifying the complexities of study planning, the app empowers students to focus on learning, reduce procrastination, and achieve their academic goals. This project exemplifies the application of contemporary software development practices to create a user-centric solution that bridges the gap between

technology and effective study management, fostering success in an increasingly competitive academic environment.

## 1.2 OBJECTIVE

1. **Organize Study Schedules**: Enable students to efficiently manage their academic workload by creating and categorizing subjects, assigning tasks, and scheduling study sessions within an intuitive mobile application.
2. **Track Academic Progress**: Empower students to monitor their study progress and upcoming tasks through a clear, interactive dashboard, promoting motivation and accountability.
3. **Build Consistent Study Habits**: Support students in developing disciplined study routines by facilitating the planning of dedicated study sessions tailored to their subjects and academic objectives.
4. **Ensure Accessibility and Usability**: Provide students with a user-friendly, accessible tool compatible with Android devices (minimum SDK 21), ensuring seamless study management anytime, anywhere.

## 1.3 EXISTING SYSTEM

The academic landscape is supported by various study management tools, but many existing systems fall short in delivering a comprehensive, user-centric experience. Traditional methods, such as physical planners and notebooks, are widely used but lack flexibility, real-time tracking, and accessibility. Digital alternatives, including generic to-do list apps like Todoist and note-taking platforms like Notion, offer task management but often lack specialized features for academic scheduling. Dedicated study apps, such as My Study Life and StudyBlue, provide subject organization and study planning, yet they frequently suffer from outdated interfaces, limited customization, or complex navigation, which can hinder user adoption. Additionally, many of these tools do not integrate progress tracking with task and session management seamlessly, leaving students to juggle multiple platforms to monitor their academic goals. Most existing systems are either web-based or lack robust offline capabilities, limiting accessibility for students with inconsistent internet access. Furthermore, few apps leverage modern Android development frameworks like Jetpack Compose or MVVM architecture, resulting in less responsive interfaces and scalability issues. While some tools offer reminders and calendars, they often fail to foster consistent study habits through tailored session planning. The Smart Study App aims to address these gaps by combining an intuitive interface, comprehensive study

management features, and modern technology to provide a unified, accessible, and efficient solution for students, overcoming the limitations of fragmented or outdated systems.

## 1.4 PROPOSED SYSTEM

The Smart Study App is proposed as a modern, comprehensive solution to revolutionize study management for students, addressing the shortcomings of existing tools. Built using Jetpack Compose, Room Library, and MVVM architecture, the app ensures a responsive, scalable, and maintainable platform. Unlike traditional planners or generic task apps, it offers a tailored academic experience with four core features: a dashboard for real-time progress tracking, subject management for organizing coursework, task assignment for goal-setting, and study session scheduling for consistent learning routines. The dashboard provides visual insights into completed tasks and upcoming deadlines, enhancing motivation. Subject and task management allow students to categorize and prioritize their workload efficiently, while session scheduling supports recurring study plans with reminders to foster discipline. Designed for Android devices with a minimum SDK version 21, the app ensures broad accessibility, including offline functionality for uninterrupted use. By leveraging Room Library, it guarantees secure, local data persistence, eliminating reliance on constant internet connectivity. The use of Jetpack Compose delivers a sleek, intuitive interface,

reducing the learning curve compared to apps with outdated designs. Unlike fragmented systems like My Study Life or Notion, the Smart Study App integrates all study-related functions into a single platform, minimizing the need for multiple tools. This proposed system aims to empower students with a user-centric, technology-driven solution to optimize their academic productivity and achieve their educational goals effectively.

# CHAPTER 2

## 2.1 LITERATURE SURVEY

The rapid integration of smartphones into daily life has significantly altered students' learning habits, often leading to a perceived decline in traditional study practices. Studies highlight that excessive smartphone use correlates with reduced attention spans and increased procrastination, contributing to a loss of disciplined learning habits. A 2017 study found that the mere presence of smartphones reduces cognitive capacity, diverting focus from academic tasks. Surveys, such as one by Meier (2017), link frequent smartphone checking to decreased well-being, further disrupting study routines. These findings suggest that habitual smartphone use fosters distractions, undermining deep learning practices like sustained reading or note-taking.

Conversely, research indicates smartphones can enhance learning when used purposefully. A 2022 study on primary school students showed that high smartphone use groups outperformed peers in academic tasks, suggesting that targeted educational apps can improve engagement. The study employed surveys and statistical analyses, revealing positive correlations between smartphone behavior and learning effectiveness. Similarly, a 2024 study on higher education students found that smartphones improve interaction competency and self-efficacy, boosting academic performance when integrated into learning activities. These results emphasize the potential of mobile learning (m-learning) to foster collaborative and flexible study environments.

However, the literature reveals mixed outcomes. Kates et al. (2018) conducted a meta-analysis showing a negative correlation between smartphone use and academic achievement ($r = -0.16$), particularly when devices are used for non-educational purposes. Surveys often rely on self-reported data, which may introduce biases, as noted in a 2020 review by Amez and Baert. This highlights the need for controlled studies to validate findings.

Mobile learning apps, like the proposed Smart Study App, address these challenges by channeling smartphone use toward structured academic tasks. Research by Alrasheedi et al. (2015) underscores that user-friendly mobile apps enhance student satisfaction and engagement. Surveys indicate that features like task tracking and reminders help students regain disciplined study habits.

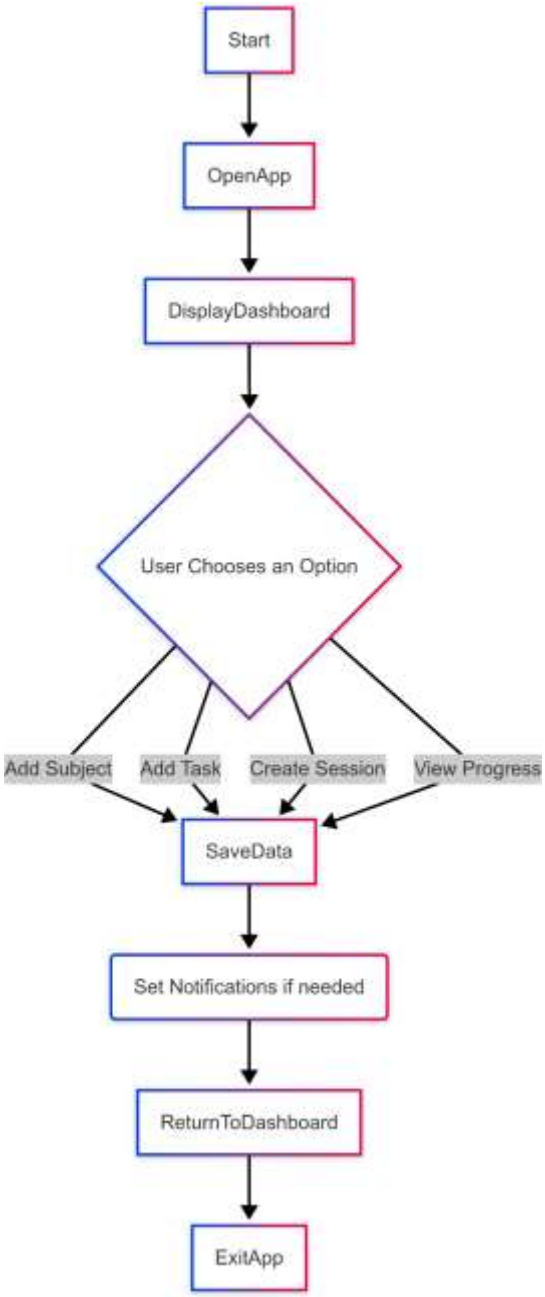
The literature also emphasizes the importance of pedagogical support. A 2015 EDUCAUSE survey found that while 86% of students own smartphones, only 50% use them daily for schoolwork, indicating a gap in guided integration. Instructors' reluctance to incorporate smartphones, due to distraction concerns, further limits their educational potential.

# CHAPTER 3
# SYSTEM DESIGN

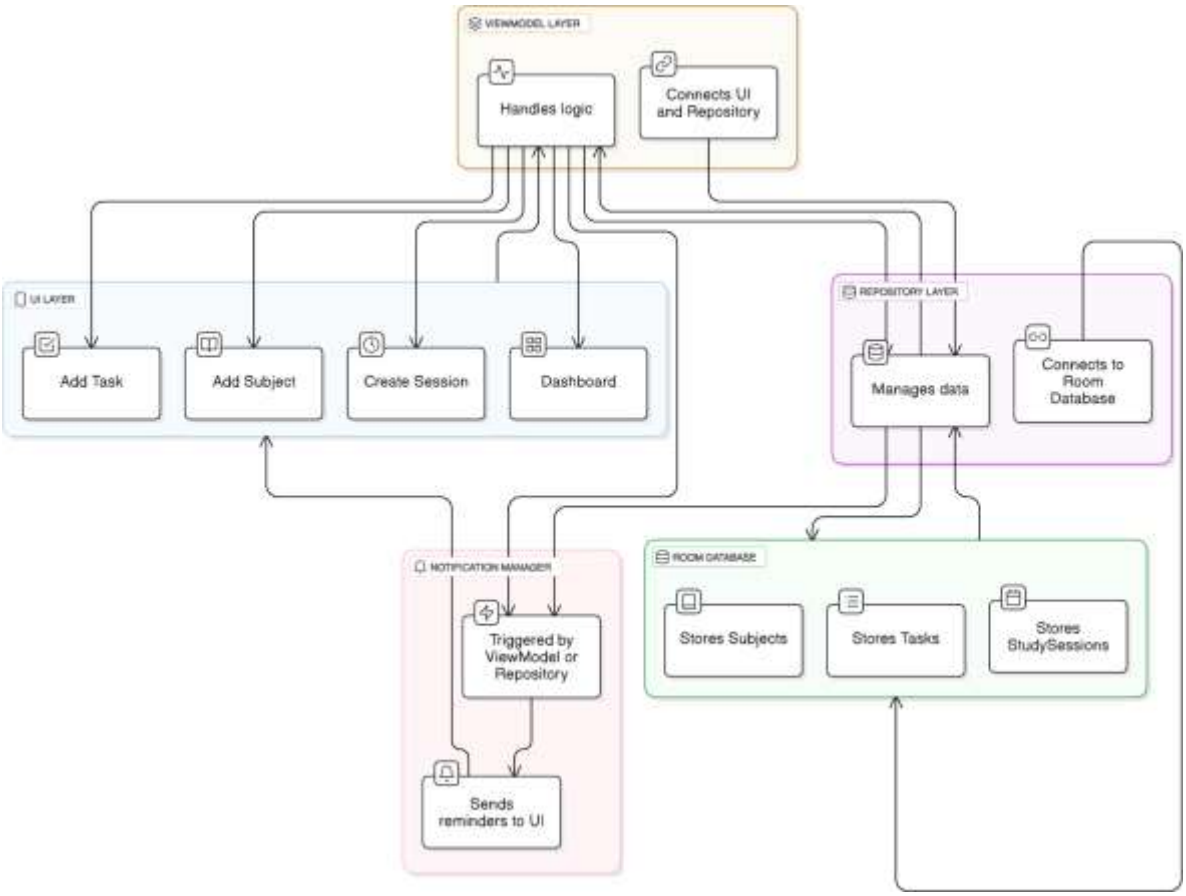
## 3.1 GENERAL

The Smart Study App's system design leverages Jetpack Compose for a modern, responsive UI, Room Library for efficient local data persistence, and MVVM architecture for scalability and maintainability. The app comprises four modules: a dashboard for progress tracking, subject management, task assignment, and study session scheduling. Data flows from UI components to ViewModels, which interact with Room database entities for CRUD operations. Coroutines manage asynchronous tasks, ensuring smooth performance. Offline functionality enhances accessibility on Android devices (minimum SDK 21). The modular design separates concerns, facilitating future enhancements while maintaining robust performance and a seamless user experience for effective study management.

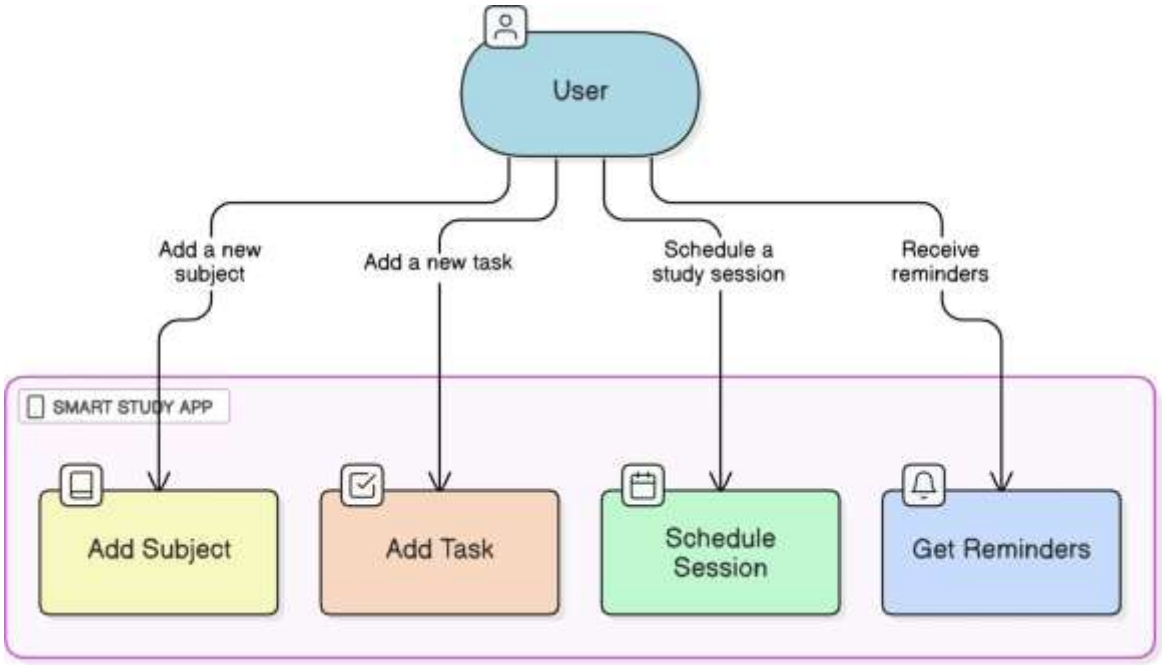
### 3.1.1 SYSTEM FLOW DIAGRAM



## 3.1.2 ARCHITECTURE DIAGRAM



### 3.1.3 USE CASE DIAGRAM



# CHAPTER 4

# PROJECT DESCRIPTION

## 4.1 METHODOLOGY

The development of the Smart Study App follows a structured and systematic approach to ensure that each phase of the project is well-executed and meets the desired objectives. The methodology used for this project is the **Agile Development Process**, which emphasizes iterative progress, flexibility, and constant feedback. The development process is divided into key phases, each contributing to the successful realization of the app. Below are the phases of the project development cycle:

### 1. Requirement Gathering

The first step in developing the Smart Study App was gathering and understanding the requirements. This phase involved:

- **Identifying Target Users**: The app is designed for students who need to manage their study schedules efficiently.
- **Defining Key Features**: Based on user needs, the primary features were identified, including subject management, task creation, study session scheduling, and progress tracking.
- **Technical Requirements**: Discussions were held on the use of Android Studio, Jetpack Compose for UI development, and Room Library for local database management. The app's minimum SDK version was set to 21, and Kotlin was chosen as the programming language.
- **Feasibility Analysis**: The technological stack was assessed to ensure compatibility and scalability.

### 2. Design

The design phase involved creating the visual and interaction design for the app. This phase focused on:

- **User Interface Design**: Using **Jetpack Compose**, an intuitive and visually appealing UI was designed. The dashboard was created to give users a clear overview of their study progress and upcoming tasks. The screens for adding subjects, tasks, and scheduling sessions were designed to be user-friendly and easy to navigate.

- **Database Design**: The app utilizes **Room Library** to manage local data, such as subjects, tasks, and study sessions. The database schema was created to ensure efficient storage and retrieval of data.

- **Wireframing**: Low-fidelity wireframes were created to plan the layout and functionality of each screen before moving to high-fidelity designs.

**3. Development**

The development phase focused on the actual coding of the Smart Study App. The following components were developed:

- **User Interface Implementation**: The design was translated into UI components using **Jetpack Compose**, providing a modern and responsive interface across various device screen sizes.

- **Database Integration**: **Room Library** was used to implement the local database for storing subjects, tasks, and study sessions. Data persistence was ensured with appropriate DAO (Data Access Object) methods for CRUD operations.

- **MVVM Architecture**: The app was built following the **Model-View-ViewModel (MVVM)** architecture to separate concerns and make the codebase more maintainable and testable. This structure helped in managing UI-related data efficiently.

- **Task Management & Session Scheduling**: Features for task creation, assignment to subjects, and scheduling study sessions were implemented using Kotlin and integrated with the database.

**4. Testing**

Once the development phase was completed, comprehensive testing was carried out to ensure the app's functionality and stability. Testing included:

- **Unit Testing**: Individual components, such as database operations and business logic, were tested using Kotlin-based unit tests to ensure they worked as expected.

- **UI Testing**: Automated UI tests were performed using tools like **Jetpack Compose Testing** to check for UI consistency, navigation flow, and responsiveness.

- **User Acceptance Testing (UAT)**: The app was tested by a small group of target users (students) to get feedback on usability and ensure that the features met their needs.

**5. Deployment**

The final phase of the project involved deploying the Smart Study App for real-world use. This included:

- **Optimization**: The app was optimized for performance to ensure fast load times and smooth transitions between screens.

- **Publishing**: The app was prepared for release on the **Google Play Store**. This involved setting up the app's metadata, such as the app description, screenshots, and icons, and uploading the APK file.

- **Post-Deployment Support**: After deployment, the app was monitored for any issues or bugs, and updates were made based on user feedback to improve the app's functionality and performance.

### 4.1.1 MODULES

The Smart Study App is structured into several well-defined modules to ensure maintainability, scalability, and a smooth user experience. Each module is responsible for a specific aspect of the application, enabling clear separation of concerns.

#### 1. User Interface (UI) Module

Handles the design and layout of the app using **Jetpack Compose**, providing a responsive and modern UI.

- Dashboard Screen: Displays study statistics, progress, and upcoming tasks/sessions.
- Subject Screen: Enables users to create, update, and delete subjects.
- Task Screen: Allows users to add, assign, and track study-related tasks.
- Session Scheduler: Lets users create study sessions for each subject and view them in an organized format.

#### 2. Database Module

Manages local data storage using the **Room Library**, ensuring data persistence and fast access.

- Entities: Data models like `Subject`, `Task`, and `StudySession`.
- Room Database: Stores user data efficiently with schema relationships.

- DAO Interfaces: Provide data manipulation methods (CRUD operations).

### 3. Business Logic (ViewModel) Module

Implements the **MVVM architecture** to manage app logic and lifecycle-aware data flow.

- ViewModels: Handle business logic and interact with the repository.
- LiveData / StateFlow: Automatically update the UI when the data changes.

### 4. Repository Module

Serves as a mediator between ViewModels and the data layer.

- Data Source Abstraction: Simplifies database interaction.
- Data Flow Management: Provides a consistent and reusable API for accessing app data.

### 5. Notification Module

Ensures users are reminded of important tasks and study sessions.

- Task Alerts: Notifies users about upcoming deadlines.
- Session Reminders: Sends notifications for scheduled study sessions.

### 6. Settings Module

Allows users to configure app preferences for a personalized experience.

- Theme Options: Switch between light and dark mode.

- Notification Preferences: Enable or disable specific alerts.
- General App Settings: Customize additional behaviors based on user needs.

### 7. Testing and Debugging Module

Focuses on application stability through structured testing.

- Unit Testing: Validates core logic and database operations.
- UI Testing: Verifies layout and navigation consistency.
- Debugging Tools: Implements logs and error monitoring for faster issue resolution.

### Features

The Smart Study App offers a variety of user-focused features designed to simplify study planning and boost productivity.

- Dashboard Overview: Quickly view study progress, pending tasks, and upcoming sessions in one place.
- Subject Management: Create and manage multiple subjects to organize your academic areas.
- Task Tracking: Add, edit, and monitor tasks related to each subject for goal-oriented study.
- Session Scheduling: Plan study sessions per subject to build and maintain a consistent study habit.
- Smart Reminders: Get timely notifications for tasks and sessions to stay on track.

- Customizable Settings: Personalize the app with theme options and notification controls.

# CHAPTER 6

# CONCLUSION

## 6.1 GENERAL

The Smart Study App was developed with the primary objective of helping students manage their academic activities more effectively. By integrating key functionalities such as subject creation, task assignment, and session scheduling, the app addresses the common challenges faced by students in organizing their study routines. Utilizing modern Android development tools like Jetpack Compose for the UI, Room Library for local data storage, and the MVVM architectural pattern, the app ensures a clean, responsive, and maintainable codebase.

The modular structure of the app allows for flexibility and scalability, making it easier to enhance or introduce new features in the future. Features like real-time notifications and a visual dashboard enable users to stay updated and motivated in their academic journey. The testing and debugging process further ensured the app's stability and user-friendliness.

In conclusion, the Smart Study App successfully delivers a streamlined platform for students to plan, track, and maintain their study schedules. It lays a strong foundation for future enhancements, such as cloud sync and advanced analytics, to further support students in achieving their educational goals.

# APPENDICES

## APPENDIX A: SOURCE CODE SNIPPETS

### A.1 Launch background.xml

```
<?xml version="1.0" encoding="utf-8"?>
<vectorxmlns:android="http://schemas.android.com/apk/res/android"
        android:width="108dp"
        android:height="108dp"
        android:viewportWidth="108"
        android:viewportHeight="108">
        <path
        android:fillColor="#3DDC84"
        android:pathData="M0,0h108v108h-108z" />
        <path
        android:fillColor="#00000000"
        android:pathData="M9,0L9,108"
        android:strokeWidth="0.8"
        android:strokeColor="#33FFFFFF" />
        <path
        android:fillColor="#00000000"

         />
```

**A.2 Main Activity.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">

<uses-permission
android:name="android.permission.FOREGROUND_SERVICE" />

<uses-permission
android:name="android.permission.FOREGROUND_SERVICE_SPECIAL_USE"
/>

<uses-permission android:name="android.permission.POST_NOTIFICATIONS"
/>

<application

android:name=".MyApplication"

android:allowBackup="true"

android:dataExtractionRules="@xml/data_extraction_rules"

android:fullBackupContent="@xml/backup_rules"

android:icon="@drawable/studying"

android:label="@string/app_name"

android:roundIcon="@drawable/studying"

android:supportsRtl="true"

android:theme="@style/Theme.SmartStudy"

tools:targetApi="31">
```

<activity

android:name=".MainActivity"

android:exported="true"

android:label="@string/app_name"

android:theme="@style/Theme.SmartStudy">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

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

</intent-filter>

</activity>

<service

android:name=".presentation.screens.session.SessionTimerService"

android:exported="false"

android:foregroundServiceType="specialUse" />

</application>

</manifest>

**A.3 Build.gradle.kts**

plugins {

alias(libs.plugins.android.application)

alias(libs.plugins.jetbrains.kotlin.android)

```
id("com.google.dagger.hilt.android")
id("com.google.devtools.ksp")
}
android {
namespace = "com.example.smartstudy"
compileSdk = 34
defaultConfig {
applicationId = "com.example.smartstudy"
minSdk = 24
targetSdk = 34
versionCode = 1
versionName = "1.0"
testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
vectorDrawables {
useSupportLibrary = true
}
}
buildTypes {
release {
isMinifyEnabled = false
proguardFiles(
```

```
getDefaultProguardFile("proguard-android-optimize.txt"),
"proguard-rules.pro"
)
}
}
compileOptions {
isCoreLibraryDesugaringEnabled = true
sourceCompatibility = JavaVersion.VERSION_11
targetCompatibility = JavaVersion.VERSION_11
}
kotlinOptions {
jvmTarget = "11"
}
buildFeatures {
compose = true
}
composeOptions {
kotlinCompilerExtensionVersion = "1.5.3"
}
packaging {
resources {
```

```
excludes += "/META-INF/{AL2.0,LGPL2.1}"
}
}
}
dependencies {
implementation(libs.androidx.core.ktx)
implementation(libs.androidx.lifecycle.runtime.ktx)
implementation(libs.androidx.activity.compose)
implementation(platform(libs.androidx.compose.bom))
implementation(libs.androidx.ui)
implementation(libs.androidx.ui.graphics)
implementation(libs.androidx.ui.tooling.preview)
implementation(libs.androidx.material3)
testImplementation(libs.junit)
androidTestImplementation(libs.androidx.junit)
androidTestImplementation(libs.androidx.espresso.core)
androidTestImplementation(platform(libs.androidx.compose.bom))
androidTestImplementation(libs.androidx.ui.test.junit4)
```

**A.4 launcher round.xml**

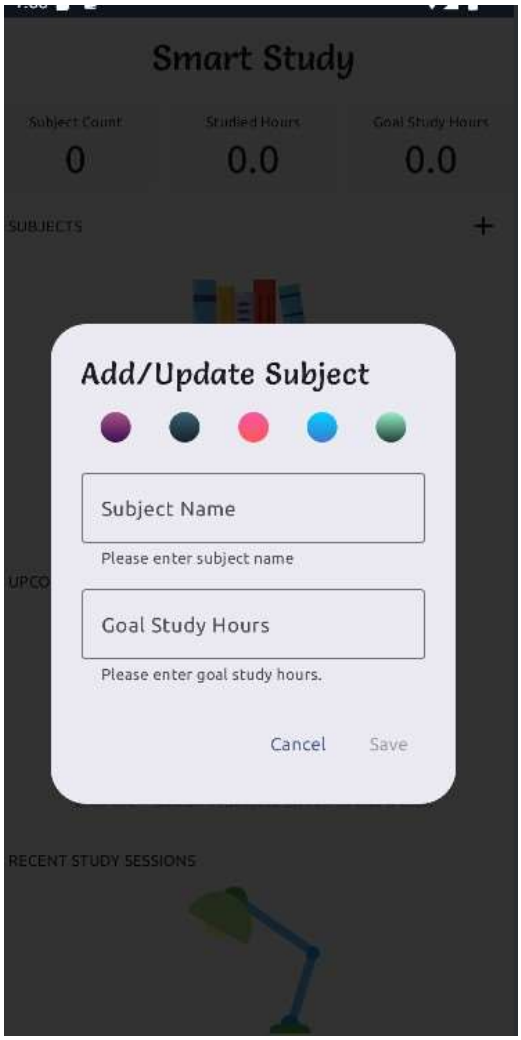
```
<?xml version="1.0" encoding="utf-8"?>
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
<background android:drawable="@drawable/ic_launcher_background" />
<foreground android:drawable="@drawable/ic_launcher_foreground" />
<monochrome android:drawable="@drawable/ic_launcher_foreground" />
</adaptive-icon>
```

## APPENDIX B: APPLICATION SCREENSHOTS

### 1. STUDY DASHBOARD


### 2. STUDY SESSION

## 3. ADD SUBJECT




## 4. TRACK TIME

## REFERENCES


1. Android Developers. (n.d.). *Jetpack Compose Overview*. Retrieved from https://developer.android.com/jetpack/compose

2. Android Developers. (n.d.). *Room Persistence Library*. Retrieved from https://developer.android.com/training/data-storage/room

3. Google. (n.d.). *Guide to app architecture*. Retrieved from https://developer.android.com/jetpack/guide

4. Kotlin Foundation. (n.d.). *Kotlin Programming Language*. Retrieved from https://kotlinlang.org/

5. MindOrks. (2021). *MVVM Architecture Android Tutorial for Beginners*. Retrieved from https://mindorks.com/post/mvvm-architecture-android-tutorial-for-beginners-step-by-step-guide

6. Vogella. (n.d.). *Introduction to Android Development with Kotlin*. Retrieved from https://www.vogella.com/tutorials/AndroidKotlin/article.html

7. GitHub. (n.d.). *Sample Android MVVM Projects with Jetpack Compose*. Retrieved from https://github.com/topics/mvvm-architecture