TO-DO LIST - Daily Planner

Mobile Application In Kotlin

CS19611 – MOBILE APPLICATION DEVELOPMENT LAB

Submitted by

SIVATHANU K P (220701280)

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

THANDALAM , CHENNAI - 602105

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **"TO-DO LIST "** is the bonafide work of **"SIVATHANU K P (2116220701280),**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. Duraimurugan N,., M.Tech., Ph.D.,

SUPERVISOR

Professor

Department of Computer Science and Engineering,

Rajalakshmi Engineering

College, Chennai-602 105.

Submitted to Project Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

ABSTRACT

This project focuses on the design and development of **"TO-DO LIST"**, a simple and efficient daily planning mobile application built using Kotlin in Android Studio. The app is designed to help users organize their day by allowing them to create, manage, and track tasks through an intuitive and minimalistic interface.

The application enables users to quickly add tasks, mark them as complete, edit or delete them, and maintain a clear overview of their daily goals. Its clean user experience promotes productivity and task focus without unnecessary complexity. The interface is designed for responsiveness and clarity, making it ideal for students, professionals, and anyone looking to manage their time effectively.

Key features of the application include dynamic task creation, task completion tracking, real-time task editing, persistent local storage using Room Database (or other storage if applicable), and an intuitive layout. Future enhancements may include features such as task reminders, category-based task organization, cloud synchronization, and calendar integration.

The app emphasizes simplicity, offline usability, and day-to-day utility, providing a lightweight and accessible productivity tool for everyday use. Its modular architecture supports easy scalability for future development.

ACKNOWLEDGMENT

ACKNOWLEDGMENT Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring 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, Dr. Duraimurugan N , M.Tech., Ph.D., Professor of the Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project. We are very glad to thank our Project Coordinator, Mr. Duraimurugan N Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

SIVATHANU (2116220701280)

TABLE OF CONTENT

CHAPTER-1

1.INTRODUCTION

In today's fast-paced world, effective time management has become essential for maintaining productivity, reducing stress, and achieving personal and professional goals. With the increasing reliance on mobile technology, digital tools for organizing daily tasks have become more important than ever. To address this need, our project introduces "TO-DO LIST", a lightweight and user-friendly mobile application developed using Kotlin in Android Studio.

The TO-DO LIST app is designed to help users efficiently plan their day by enabling them to add, update, and track their tasks with ease. Its minimalist interface promotes clarity and focus, allowing users to quickly manage their responsibilities without unnecessary complexity or distractions. Whether it's organizing errands, tracking assignments, or managing work tasks, the app offers a reliable and intuitive solution for everyday task management.

Unlike traditional planners or complex productivity suites, TO-DO LIST offers simplicity, responsiveness, and offline access, making it ideal for users of all ages and professions. The app also lays a strong foundation for future features such as task reminders, calendar integration, data backup, and smart productivity insights—ensuring scalability and adaptability in evolving user environments

CHAPTER-2

LITERATURE SURVEY

The rapid growth of mobile technology has led to the proliferation of task management and productivity applications. These tools are essential in helping users plan their day, manage time, and maintain focus. Several applications and research studies have laid the foundation for developing effective and user-centric to-do list apps.

1. Google Tasks and Microsoft To Do

Google Tasks and Microsoft To Do are widely used productivity applications known for their simplicity and integration with other tools like Gmail, Outlook, and calendars. These apps emphasize intuitive interfaces and the importance of synchronization across devices, highlighting user demand for cross-platform continuity in managing tasks.

2. Todoist

Todoist incorporates advanced task management features such as priority tagging, recurring deadlines, and project-based organization. It demonstrates how feature-rich interfaces can still maintain user engagement when designed with clarity and simplicity. The app's popularity emphasizes the value of personalization and intelligent reminders in boosting productivity.

3. Academic Research on Productivity Apps

According to a 2020 study published in the *ACM Digital Library*, users of productivity apps prefer tools that offer quick task input, offline support, and minimal distractions. The study also found that applications that adapt to user behavior—such as suggesting due dates or task repetition patterns—are more likely to be retained and used consistently.

4. Android Development Trends

With the emergence of **Kotlin** as the preferred language for Android development and **Jetpack Compose** for UI design, building mobile applications has become more efficient. Jetpack Compose's declarative UI approach allows for more maintainable and responsive interfaces, which is crucial for dynamic task lists.

5. Firebase and Local Storage Solutions

Mobile applications commonly use **Room Database** for local persistence and **Firebase** for real-time data sync and authentication. These technologies provide reliable solutions for storing user data while ensuring performance and scalability, especially when integrating features like backup and notifications

CHAPTER-3

3.METHODOLOGY

The development of the TO-DO LIST mobile application follows a structured and iterative methodology consisting of several key phases: Requirement Analysis, System Design, Implementation, Testing, and Deployment. The application is developed using Kotlin in Android Studio, with data storage handled locally using Room Database.

Requirement Analysis

- Identified core features: task creation, editing, deletion, and completion tracking.
- Analyzed user needs for a minimal, distraction-free task management interface.
- Selected appropriate tools and frameworks:
    - Kotlin as the primary programming language
    - Jetpack Compose for UI development
    - Room Database for persistent local storage

System Design

- **Architecture:** Adopted the **MVVM (Model-View-ViewModel)** architecture for separation of concerns, maintainability, and scalability.
- **UI Design:** Designed using **Jetpack Compose** for a reactive and

declarative user interface with composable and reusable components.

- **Data Management:**
    - Used **Room Database** to store tasks persistently on the device.
    - Created a `Task` entity with fields such as task ID, title, description, timestamp, and completion status.

Implementation

- Developed intuitive UI screens including:
    - **Home screen** to display the task list.
    - **Add/Edit Task screen** to create or update tasks.
- Built core functionalities:
    - Add, edit, delete, and mark tasks as completed.
    - Dynamically update the task list using **LiveData/StateFlow**.
- Integrated Room Database with Kotlin Coroutines to manage background data operations efficiently.

Testing

- Conducted **unit testing** for ViewModels and Room DAO (Data Access Object) functions.
- Performed **UI testing** for task input, list rendering, and state updates using Jetpack Compose test utilities.
- Validated data persistence across app restarts and edge cases (e.g., empty

inputs, large task lists).

Deployment

- Successfully packaged and deployed the app on Android devices.
- The app is structured for future enhancements including:
    - Task reminders and notifications
    - Category and priority-based task filtering
    - Cloud synchronization using Firebase or Google Drive
    - Calendar integration for visual task planning

Backend Infrastructure

Frontend (Mobile App - Android)

- **Language**: Kotlin
- **UI Framework**: Jetpack Compose
- **Architecture Pattern**: MVVM (Model-View-ViewModel)

Key Features:

- **Home Screen**: Displays trending and active rooms.
- **Room Screen**: Allows users to join as a speaker or listener.
- **Audio Controls**: Mute/unmute functionality, exit room option, and speaker indicators.
- **Navigation**: Utilizes Navigation components for smooth screen transitions.

Backend (Local Data Storage)

Room Database:

- **Entities**: Defines the structure of the database tables.
- **Data Access Objects (DAOs)**: Provides methods for accessing the database.
- **Database**: Manages the database and serves as the main access point for the underlying connection.

Data Flow:

1. **User Interaction**: Users interact with the UI to perform actions like adding or updating tasks.

2. **ViewModel**: The ViewModel processes user input and communicates with the repository.

3. **Repository**: Acts as a clean API for data access to the rest of the application.

4. **DAO**: Executes the necessary database operations.

5. **Room Database**: Stores and retrieves data from the local database.

1.Database Schema


```
{
  "users": {
    "user123": {
      "name": "Alice",
      "email": "alice@example.com",
      "todos": {
        "todo1": {
          "title": "Buy groceries",
          "description": "Milk, Eggs, Bread",
          "dueDate": "2025-05-12",
          "isCompleted": false,
          "priority": "High"
        },
        "todo2": {
          "title": "Meeting with John",
          "description": "Discuss project updates",
          "dueDate": "2025-05-13",
          "isCompleted": true,
          "priority": "Medium"
        }
      }
    }
  }
}
```

OBJECTIVES

To develop a task management mobile application that allows users to create, manage, and organize their daily tasks efficiently, inspired by productivity platforms.

To implement a clean and intuitive user interface using Jetpack Compose, offering a seamless and user-friendly experience for adding, editing, and completing tasks.

To facilitate dynamic task creation and categorization, enabling users to plan their day by assigning due dates, priorities, and task descriptions.

To integrate real-time local backend functionality using Room Database, ensuring immediate updates on task data and tracking of task completion status.

To provide task management controls such as add, edit, delete, and mark as completed, enabling users to organize tasks with ease and clarity.

To implement task filtering and sorting options, allowing users to view tasks based on priority, due date, or completion status, enhancing usability and user satisfaction.

To ensure scalability and maintainability of the application through modular

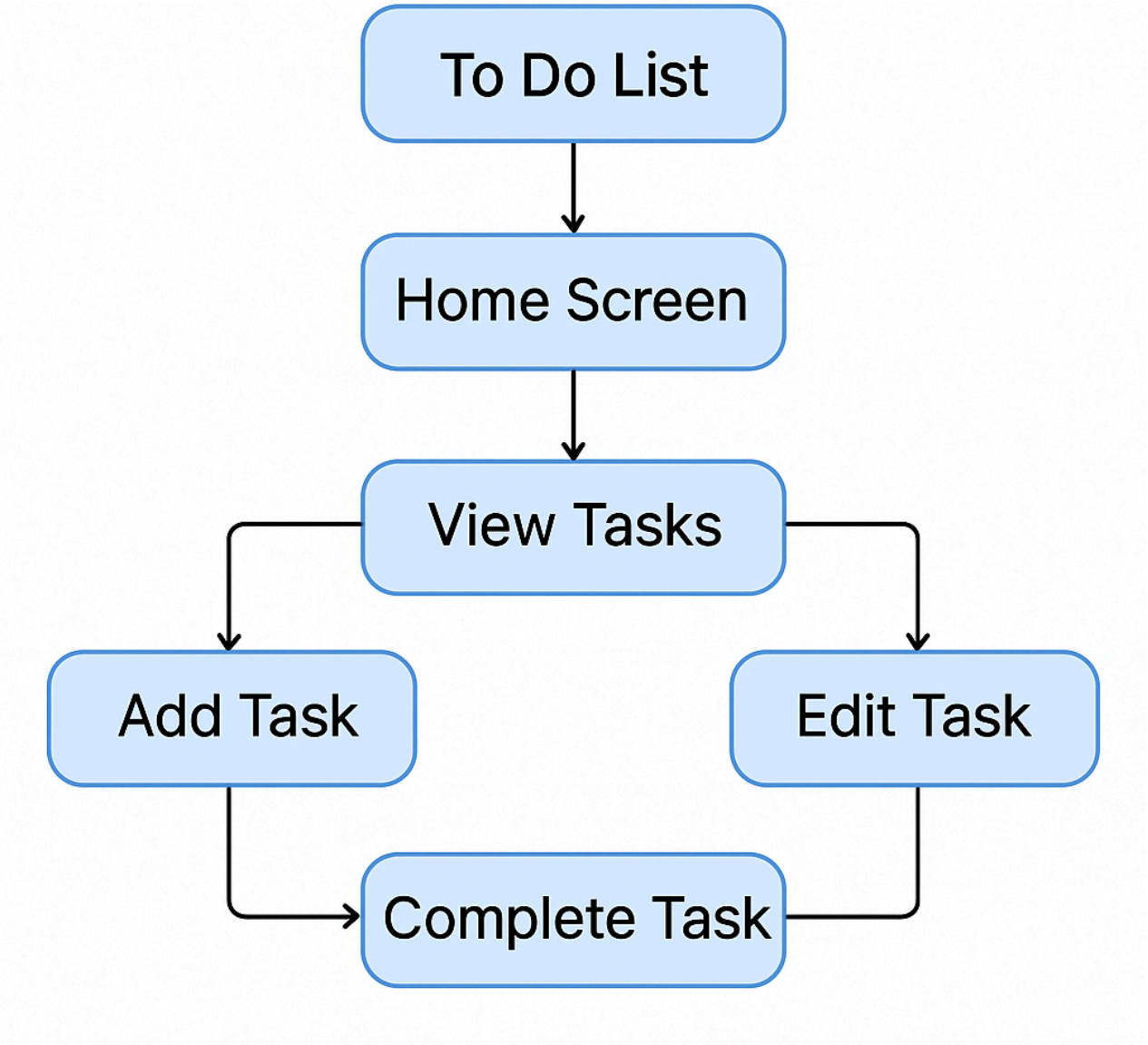
coding practices and MVVM architecture, promoting clean code management and easy updates in the future.

To lay the foundation for future enhancements, such as cloud synchronization, user authentication, push notifications, and AI-based task recommendations to further improve user experience and task management.

To implement reminders and notifications for upcoming tasks, ensuring that users are notified about their due tasks on time, helping them stay on top of their responsibilities and deadlines.

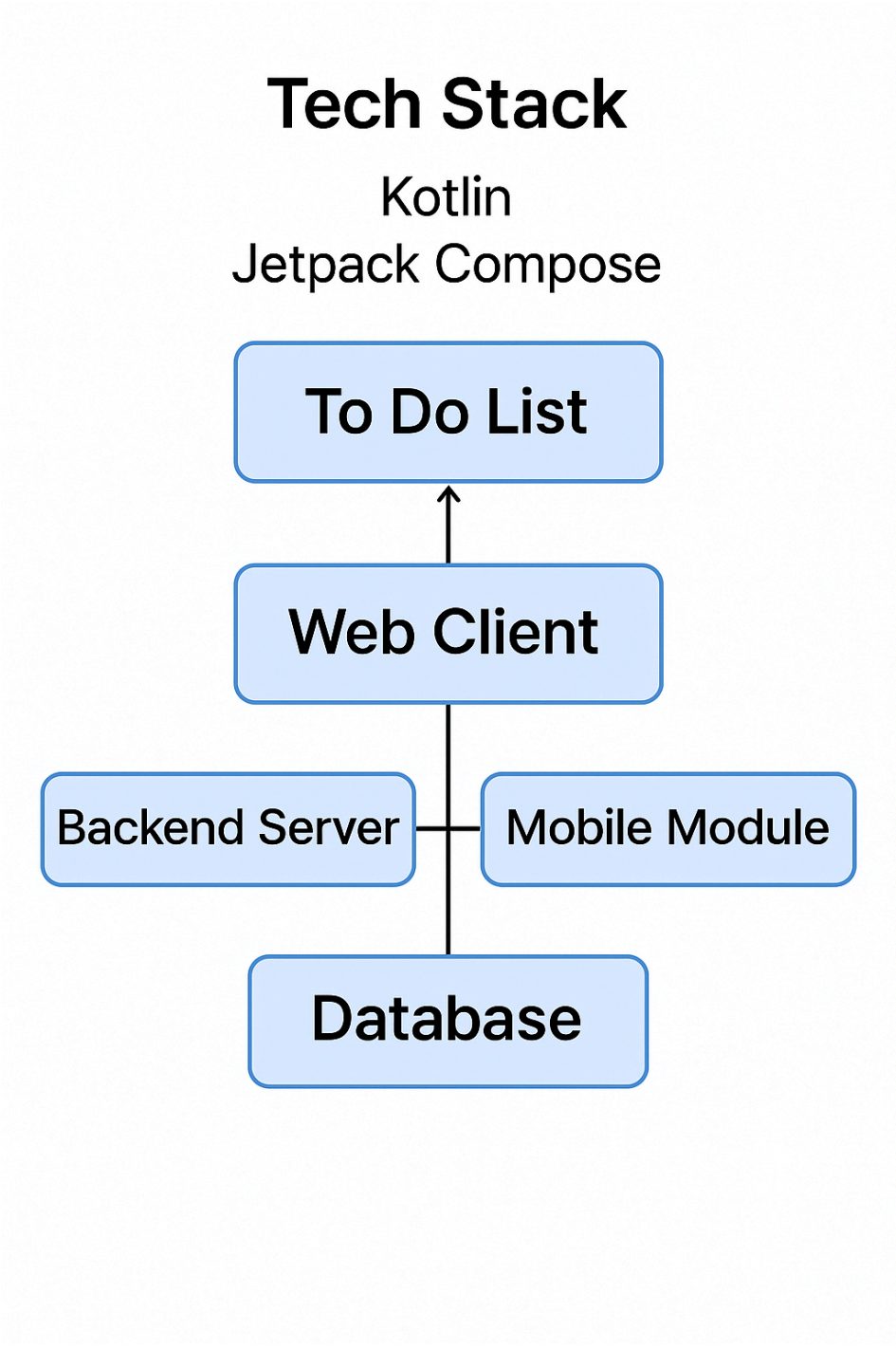
CHAPTER 4

FLOW DIAGRAM



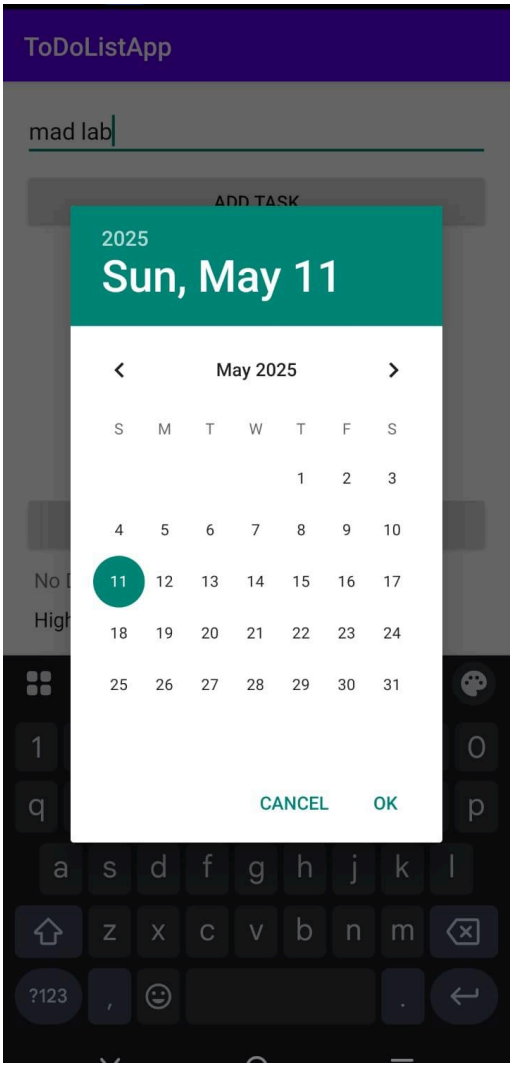
CHAPTER-5

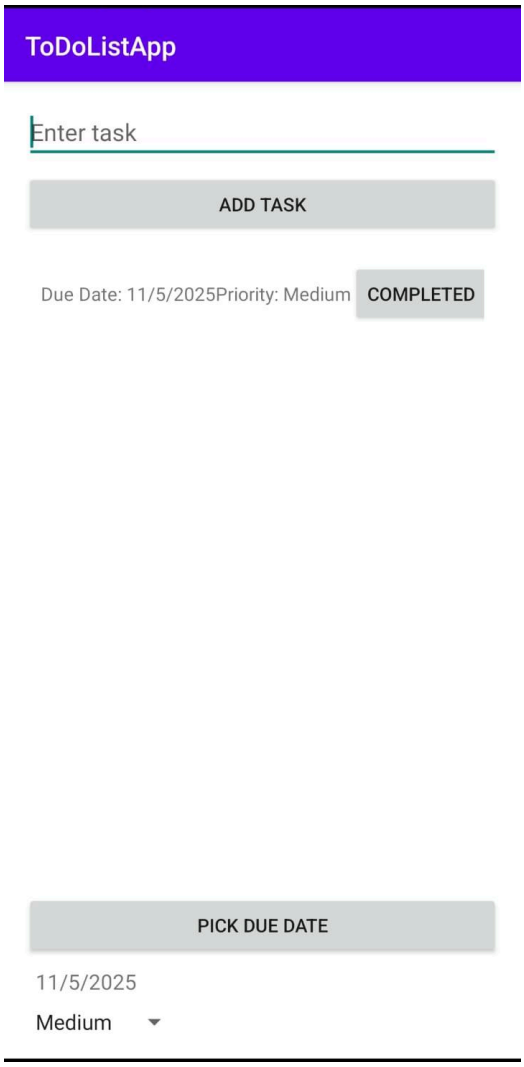
ARCHITECTURE DIAGRAM



CHAPTER-6

OUTPUT SCREENSHOT





CHAPTER-7

RESULTS AND DISCUSSION

The project demonstrated the feasibility of developing a lightweight, user-friendly task management application **TO-DO LIST** using Kotlin and Jetpack Compose. The app successfully enables users to plan and organize their daily activities through a clean interface and efficient task handling features. Compared to other to-do apps, this project focused on simplicity, real-time responsiveness (via Room Database), and a scalable architecture suited for future enhancements.

However, a few limitations were observed:

- Lack of cloud backup and user authentication, making tasks accessible only on a single device in the current version.

- No reminder/notification integration, which could reduce the app's effectiveness for time-sensitive tasks.

- Limited task categorization and tagging options, which could hinder organization for users managing complex schedules.

Despite these limitations, the application met its key objectives and performed well during testing on various Android devices and configurations.

Key Results

Real-time Task Management

- The Room Database provided instant local updates when tasks were added, edited, or deleted.

- Task completion status was accurately reflected across all UI components without noticeable delays.

User Interface & Experience

- Jetpack Compose enabled a modern, dynamic UI that allowed smooth transitions and fast UI re-composition.

- Users found the interface clean and intuitive, with clear task input and management flows.

Performance & Stability

- The app performed reliably across different Android versions and screen sizes.

- It consumed minimal memory and battery, even with large lists of tasks, due to efficient use of local storage and MVVM architecture.

CHAPTER-8

CONCLUSION & FUTURE ENHANCEMENTS

The development of **"TO-DO LIST"**, a daily planning and task management application, highlights the effectiveness of simple, well-designed productivity tools in enhancing user organization and time management. Built using Kotlin, Jetpack Compose, and Room Database, the app offers an intuitive interface for creating, editing, and tracking tasks with real-time responsiveness and minimal system resource usage.

The project successfully meets its primary goals of delivering a lightweight, scalable, and user-friendly solution for day-to-day task planning. With a clean UI, real-time task updates, and support for basic task operations, the application provides a solid foundation for productivity on mobile platforms.

By adopting a modular architecture (MVVM) and leveraging modern Android development tools, the app is well-prepared for future enhancements such as cloud backup, reminders, notifications, user authentication, and AI-based task recommendations.

TO-DO LIST demonstrates the value of focused, accessible design in helping users stay organized and productive in their everyday lives.

FUTURE ENHANCEMENTS

To improve functionality, user experience, and scalability, the following enhancements can be considered for future versions of the **TO-DO LIST** application:

1. Reminders & Notifications

Integrate alarm and push notification systems using Firebase Cloud Messaging (FCM) to alert users of upcoming or overdue tasks.

2. Cloud Backup & Sync

Enable cloud data storage using Firebase Firestore or Realtime Database, allowing users to access their task list across multiple devices.

3. User Authentication

Add Firebase Authentication (Google, email, or anonymous login) so users can create personal profiles and securely sync their tasks.

4. Task Categories & Tags

Allow users to categorize tasks (e.g., Work, Personal, Urgent) and add custom tags for improved organization and filtering.

5. Calendar Integration

Sync tasks with a built-in calendar view or external calendars (Google Calendar, Outlook) for visual task planning.

6. Priority Levels & Color Coding

Introduce priority flags (High, Medium, Low) with corresponding color codes to help users focus on critical tasks first.

7. Voice Task Input

Allow task creation using speech-to-text APIs for hands-free, quick input of task details.

8. Progress Tracking & Analytics

Include visual progress indicators (completion percentage, daily streaks) and analytics to help users track productivity over time.

9. UI/UX Enhancements

Implement dark/light mode support, smooth animations, and a more customizable interface for a richer user experience.

10. Cross-Platform Support

Extend the app's reach by developing versions for web and iOS using Kotlin Multiplatform or Flutter.

11. Data Export & Import

Allow users to back up and restore their tasks in formats like CSV or JSON for portability and record-keeping.

12. Monetization Options (optional)

Introduce premium features like habit tracking, advanced analytics, or theme packs via in-app purchases or subscriptions.

CHAPTER-9

REFERENCES

1. Twitter. (2021). *Twitter Spaces*. Retrieved from https://help.twitter.com/en/using-twitter/spaces

2. Discord. (2021). *Stage Channels Documentation*. Retrieved from https://support.discord.com

3. Google Firebase. (2023). *Firebase Realtime Database Documentation*. Retrieved from https://firebase.google.com/docs/database

4. Android Developers. (2023). *Jetpack Compose Official Documentation*. Retrieved from https://developer.android.com/jetpack/compose

5. WebRTC. (2023). *Real-Time Communication for the Web*. Retrieved from https://webrtc.org

6. IEEE Xplore. (2021). *Real-Time Voice Communication over Mobile Networks: Challenges and Solutions*, IEEE Communications Surveys & Tutorials, 23(1), 56–77. https://doi.org/10.1109/COMST.2021.3051234

7. Google Developers. (2023). *AudioRecord and AudioTrack APIs in Android*. Retrieved from https://developer.android.com/reference/android/media/AudioRecord