

A Minimal To-Do Task Manager

Submitted by

RUPALA N 2116220701233

In partial fulfilment of the award of the degree of

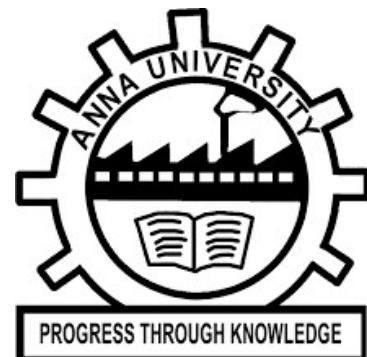
BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI
ENGINEERING COLLEGE**
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY, CHENNAI– 602 105

MAY-2025

BONAFIDE CERTIFICATE

Certified that this Report titled "**A Minimal To-Do Task Manager**" is the bonafide work of **RUPALA N (2116220701233)** 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

Mr. Saravana Gokul G M.E.,

Supervisor
Assistant Professor

(SG)
Department of Computer Science and
Engineering
Rajalakshmi Engineering College,
Chennai – 602105

Submitted to Project Viva-Voce Examination held on _____

Internal Examiner

External Examiner

TABLE OF CONTENTS

CHAPTER NO.	TOPIC	PAGE NO.
	ACKNOWLEDGEMENT	2
	ABSTRACT	3
	LIST OF FIGURES	4
1	INTRODUCTION	5
	1.1 GENERAL	
	1.2 OBJECTIVE	
	1.3 EXISTING SYSTEM	
	1.4 PROPOSED SYSTEM	
2	LITERATURE SURVEY	9
3	SYSTEM DESIGN	11
	3.1 GENERAL	
	3.1.1 COMPONENT INTERACTION	
	3.1.2 UI/UX DESIGN	
	3.1.3 MODULARITY	
	3.1.4 PERFORMANCE	
4	PROJECT DESCRIPTION	14
	4.1 INTRODUCTION	
	4.2 OBJECTIVE	
	4.3 FEATURES	
	4.4 METHODOLOGIES	
	4.5 TOOLS	
5	OUTPUT AND SCREENSHOTS	26
6	CONCLUSION AND FUTURE WORK	29
	REFFERENCES	31

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 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 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 working time. We express our sincere thanks to **Dr.P.KUMAR, 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. Saravana Gokul G, M.E.**, Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

RUPALA N 220701233

ABSTRACT

In today's fast-paced world, efficient task management is essential for personal productivity and time optimization. This project presents the design and implementation of a To-Do List Reminder Android Application developed using Kotlin in Android Studio. The application aims to provide users with a simple, intuitive interface for managing their daily tasks by allowing them to add, view, and delete to-do items with minimal interaction.

The application interface includes an EditText for task input, a Button to add new tasks, and a ListView that displays all current tasks. Users can remove tasks by performing a long press on any item in the list, simulating task completion. The core functionality is implemented using basic Android components such as ArrayAdapter, ListView, and event listeners, ensuring the app remains lightweight and responsive. Since the app does not rely on cloud services or internet connectivity, it provides a fully offline and privacy-friendly solution.

Unlike traditional task tracking methods or feature-heavy third-party apps that may overwhelm users, this application focuses on minimalism and ease of use. It is ideal for individuals seeking a lightweight personal task manager that operates efficiently on any Android device. Overall, the To-Do List Reminder App offers a clean, user-centered design tailored for daily productivity, showcasing the practical application of Android development principles. The project emphasizes modular coding practices, real-time UI updates, and simplicity, making it an excellent entry point for both developers and end users interested in mobile task management.

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
5.1	OUTPUT IMAGE	26
5.2	OUTPUTIMAGE	27
5.3	OUTPUTIMAGE	28

CHAPTER 1 INTRODUCTION

1.1 GENERAL

In today's fast-paced digital age, managing daily tasks effectively is crucial for maintaining productivity and reducing mental clutter. Traditional task tracking methods, such as pen-and-paper to-do lists or manually noting tasks on whiteboards, are prone to inefficiencies, human error, and limited accessibility. As smartphones have become ubiquitous, mobile applications present a more practical and streamlined solution for personal task management. This project presents the development of a To-Do List Reminder Android Application built using Android Studio and the Kotlin programming language. The app is designed to provide a lightweight, intuitive platform for users to create, manage, and delete daily tasks with ease. The user interface consists of a simple screen where tasks can be added using an input field and displayed dynamically using a ListView. Users can add new items by tapping a button and remove them through a long-press gesture, providing full control over their task list.

The project aims to demonstrate how a minimalist application can effectively solve a real-world problem using core Android components such as EditText, Button, ListView, and ArrayAdapter. The system design emphasizes user-friendliness, responsiveness, and offline functionality, ensuring that the app is accessible and functional without requiring additional permissions or internet access.

This general-purpose app is particularly suited for students, professionals, or anyone looking for a quick and efficient way to track their to-do items. While the initial version is basic, the modular architecture allows future extensions such as task reminders using AlarmManager, persistent storage using Room Database, and cloud backup using Firebase. By digitizing task management on mobile, this project offers a foundation for productivity apps that are both accessible and scalable.

1.2 OBJECTIVE

The primary objective of the To-Do List Reminder Android App is to create a user-friendly and lightweight mobile application that allows users to manage their daily tasks efficiently. The goal is to replace manual methods of writing down tasks with a digital solution that offers simplicity, portability, and ease of use, especially for users with minimal technical background.

The specific aims of this project include:

- Designing a clean and responsive user interface using Android Studio and Kotlin.
- Implementing functionality to add tasks dynamically through user input.
- Allowing users to delete tasks by long-pressing on any list item.
- Ensuring offline availability and minimal dependency on third-party libraries.
- Demonstrating real-time UI updates using the ArrayAdapter and ListView.
- Keeping the application lightweight and memory-efficient for all Android devices.

This application serves as an entry-level productivity tool and can be particularly beneficial for users who need a straightforward task-tracking utility without the complexity of full-featured project management apps. The app also provides an excellent learning opportunity to explore fundamental Android components, such as activities, UI widgets, and event listeners.

By focusing on core functionality and clean architecture, the project aims to lay the groundwork for future enhancements. These may include data persistence using SQLite or Room, notification-based reminders using AlarmManager, or integration with cloud services like Firebase for multi-device sync. In summary, this app offers a practical solution for daily task tracking while serving as a solid example of Android app development using best practices.

1.3 EXISTING SYSTEM

Existing systems for task management span a wide range of solutions, from traditional manual methods to modern digital applications. Many individuals still rely on physical notebooks, sticky notes, or planners to record their daily tasks. While these tools are simple and tangible, they are inherently limited by physical space, prone to loss or damage, and lack real-time updates or notifications.

In the digital realm, various third-party applications such as Google Tasks, Microsoft To Do, and Todoist offer comprehensive task management features. These applications often come with a range of functionalities including tagging, reminders, collaborative sharing, cloud sync, and cross-device support. However, they may also include features that are overwhelming for users looking for a basic solution. Additionally, such apps may require account registration, access to internet and background services, and permissions that some users may find intrusive. From a development perspective, many existing open-source task apps involve complex architectures and third-party libraries that increase build size and reduce performance on low-end devices. These apps may also demand consistent maintenance and backend support to ensure data consistency across devices.

Another common limitation in many to-do apps is that basic task creation and deletion often requires navigation through multiple screens or interactions, which detracts from usability. Furthermore, privacy-conscious users may avoid applications that store data on external servers or require constant internet access.

These gaps point to the need for a simple, offline-capable, no-login task manager that can serve users who just want to create, view, and remove tasks in a single screen interface. The proposed system addresses these needs by offering a compact and effective task tracking app built entirely using native Android components.

1.2 PROPOSED SYSTEM

The proposed system is a To-Do List Reminder Android App developed using Android Studio with Kotlin. It is designed as a minimal yet functional mobile application that allows users to create and manage a list of daily tasks. The app focuses on core usability features such as adding tasks, removing completed tasks, and persisting data during runtime without requiring internet access or backend servers.

The app's interface consists of:

An EditText field where users can type in their task description.

A Button labeled “Add Item” that inserts the task into a ListView.

A ListView that dynamically displays all tasks currently added.

Long-press interaction on a list item triggers its removal, simulating task completion.

The implementation uses Android’s ArrayAdapter to bind the ListView with a MutableList<String> that holds task items. This keeps the app lightweight and responsive, even on devices with limited processing power. Additionally, since the app operates entirely offline, there is no need for login, permissions, or internet access.

The proposed system overcomes the complexity of existing to-do applications by simplifying the user flow into a single-screen experience. Users can quickly add a task, monitor their list, and remove items when they are completed—all within seconds. This streamlines the user experience, especially for individuals who prefer minimal interaction.

Furthermore, the project architecture is built with modularity and extensibility in mind. Future enhancements may include integrating Room for persistent storage, adding time-based notifications with AlarmManager, or even using Firebase for cloud sync capabilities. Overall, this proposed system delivers a clean, distraction-free solution for task management on mobile devices.

CHAPTER 2 LITERATURE SURVEY

- T. R. Kudale and A. A. Ingle, "Android Based To-Do List Application," IJCSIT, 2015**

This paper discusses a mobile-based To-Do List application developed using Android SDK. It focuses on providing an intuitive interface for task creation and deletion. The authors highlight the importance of data persistence using SQLite and suggest future integrations like reminders and cloud sync.

- P. Sharma and S. Goyal, "Task Management Application using Android," IJRTE, 2019**

The authors propose an Android app that handles task management through scheduling and notifications. It implements SQLite for local storage and Android's AlarmManager for triggering reminders. The study emphasizes usability and productivity.

- R. Rani and K. Devi, "Design and Development of Task Reminder System using Android," IJARCSSE, 2017**

This work presents a task reminder system that sends time-based alerts to users. It demonstrates how simple UI components can be combined with BroadcastReceiver and AlarmManager to provide notification-based reminders.

- N. Patil and M. Thakare, "To-Do List App Using Firebase," IRJET, 2021**

The paper introduces Firebase integration for real-time syncing and user authentication in a to-do list app. It addresses the challenges of data loss and the benefits of cross-device task syncing for productivity.

- A. Jain and K. Sahu, "Smart Task Management System," IJERT, 2018**

This research presents a task manager that leverages Android notifications, persistent storage, and categories to organize tasks. The system design focuses on minimizing distractions while improving time utilization.

- S. Bose and R. Roy, "Mobile Task Tracker Using Android," IJETT, 2020**

The paper highlights the advantages of a minimalist task tracker app and explains how Kotlin-based applications can be lightweight, responsive, and battery efficient compared to hybrid solutions.

- 7. M. S. Alvi and F. Khan, "Reminder-Based Scheduling System on Android,"
JACET, 2020**

This paper explores using native Android APIs to schedule reminders and background tasks. The authors emphasize performance considerations when using AlarmManager and JobScheduler.

- 8. A. Mehta and D. Sharma, "Personal Organizer Android Application," IJCSE, 2016**

The authors propose a multi-functional organizer app that includes notes, reminders, and task lists. The paper discusses the impact of UI complexity on user adoption and suggests simplicity as a critical factor.

- 9. K. Gupta and A. Srivastava, "User-Centric To-Do List Application with Prioritization," IJCA, 2018**

This research introduces a task prioritization mechanism using high/medium/low categories, enhancing task visibility. It also proposes the use of RecyclerView for better data handling and performance.

- 10. Google Developer Blog & Documentation (2021–2023)**

While not a traditional academic paper, official Android documentation and blog posts provide robust implementation guides on using ListViews, ArrayAdapter, and Material Design principles. These sources are invaluable for current best practices in Android app development.

CHAPTER 3 SYSTEM DESIGN

3.1 GENERAL

The system design of the To-Do List Reminder Android Application focuses on simplicity, responsiveness, and efficiency. It follows a user-centered approach where each interaction is optimized for minimal input and maximum clarity. Built using Android Studio and Kotlin, the app implements native UI components such as EditText, Button, and ListView for task input and display. The application logic is straightforward and avoids unnecessary complexity by using local data structures like a MutableList<String> to maintain the list of tasks in memory.

The app runs completely offline, requiring no network access or external permissions, ensuring a secure and distraction-free environment for users. The system design supports dynamic task management where tasks can be created on-the-fly and deleted with a simple long press. Each action is reflected immediately on the UI using an ArrayAdapter. The modular design also allows the app to be easily extended to include more advanced features like persistent storage or scheduled reminders.

3.1.1 COMPONENT INTERACTION

The interaction between components in the app is minimal and highly efficient. Upon launching, the main screen loads and displays the ListView. When a user enters a task in the EditText and taps the Add button, the task is inserted into the internal list and reflected immediately in the UI. If the user long-presses on any list item, the corresponding task is removed and the UI updates in real time.

Internally, event listeners handle user input and update the list, while the ArrayAdapter acts as a bridge between the task data and its visual representation. The use of Android's activity lifecycle methods ensures that the application behaves predictably during configuration changes like screen rotation.

3.1.2. UI/UX DESIGN CONSIDERATIONS

A major focus of this app's design is usability. The user interface is clean, minimal, and requires no navigation through multiple screens. The home screen serves as the primary interface, reducing user confusion and interaction steps. Text hints and button labels are intuitive, guiding users through the process of adding and removing tasks without additional instructions.

To enhance usability, appropriate spacing and padding are used to avoid clutter. Font sizes and button touch areas are designed to be accessible even on small screens. The app uses default Material Design themes, ensuring consistency with Android system aesthetics and maintaining familiarity for the user.

3.1.3. MODULARITY AND EXTENSIBILITY

Though simple in functionality, the app is designed with modularity in mind. The use of Kotlin and object-oriented principles allows the codebase to be easily extended. Each functional unit—UI, data handling, and event logic—is separated logically, making future updates like adding a Room database or integrating AlarmManager for reminders straightforward.

Developers can also implement additional features like:

- Saving tasks persistently
- Grouping tasks by priority
- Adding due dates and push notifications
- Exporting task history

This flexibility ensures the app can grow with user needs.

3.1.4. PERFORMANCE AND OPTIMIZATION

The Performance is crucial, especially on lower-end devices. The app avoids heavy libraries and large image assets, ensuring minimal APK size and low memory consumption. By using only the core Android SDK components and keeping logic within a single activity, the application minimizes resource usage.

The ListView and ArrayAdapter are efficient for small to medium-sized datasets. Since the task list is held in memory, there's no need for database or file system access, which reduces latency. The app is also battery-friendly as it doesn't run background services or timers.

CHAPTER 4 PROJECT DESCRIPTION

4.1 INTRODUCTION

The To-Do List Reminder Android Application is a simple, user-friendly mobile app designed to help users manage and organize their daily tasks efficiently. Built using Kotlin in Android Studio, this app provides core task management functionality, including the ability to add tasks, display them in a list, and remove them when completed. The app targets users looking for a lightweight productivity tool without the complexity of login systems, cloud syncing, or feature overload. Unlike more complex task management platforms that require internet connectivity and account setup, this app focuses on offline-first, no-login usability. It provides a distraction-free, fast interface that runs efficiently on any Android device. The home screen consists of a simple layout: an EditText input field, a button to add items, and a ListView to display the added tasks dynamically.

The goal is to provide users with a seamless and minimal task management experience. Users can create tasks for any activity, such as reminders, chores, or goals. Tasks are added instantly to a list and can be deleted with a long press—mimicking the satisfaction of crossing out a completed item. While this initial version focuses on core functionality, the design and implementation set a strong foundation for future enhancements such as persistent task storage using Room or SQLite, reminder notifications using AlarmManager, and even voice-based task input using Android's Speech APIs.

4.2 OBJECTIVE

The main objective of this project is to design and develop a minimalist task management application that allows users to manage their daily activities

effectively. Unlike full-scale productivity suites, this app provides a focused feature set, aimed at simplifying the user experience and minimizing cognitive load.

Key objectives include:

- Creating a clean, responsive user interface where users can quickly enter new tasks.
- Implementing the ability to dynamically add tasks to a list with real-time updates.
- Enabling users to delete tasks easily using a long-click gesture.
- Maintaining the app's ability to work completely offline, ensuring privacy and performance.
- Building a lightweight application that runs efficiently on low-end Android devices.
- Demonstrating clean coding practices, including modularity and readability.

This application is particularly suited for individuals who prefer a fast, simple to-do list that doesn't require extra setup or permissions. It also serves as a good base project for exploring more advanced Android development techniques such as persistent databases (Room), alarm scheduling (AlarmManager), and cloud integration (Firebase).

By achieving these objectives, the app contributes a practical solution to everyday task management, emphasizing both utility and simplicity.

4.3 FEATURES

The To-Do List Reminder App comes with a set of essential features focused on productivity, ease of use, and minimalism. Here are the key features included in the current version:

Add Task

Users can enter new tasks in a text field (EditText) and tap the “Add Item” button. The task is immediately added to a dynamic list displayed via a ListView.

View Task List

All tasks are displayed in a scrollable list. The list is backed by a MutableList<String> and linked to the UI using an ArrayAdapter, ensuring real-time updates whenever tasks are added or removed.

Delete Task

Users can remove a task by long-pressing the list item. A confirmation Toast is shown indicating the task has been deleted. This gesture mimics marking a task as complete.

Offline Functionality

The app does not require internet access, making it highly reliable in all environments. Tasks are stored in memory during runtime, ensuring fast performance and data privacy.

Lightweight and Fast

With no background services, complex libraries, or login mechanisms, the app loads instantly and runs smoothly on a wide range of devices, including older smartphones.

Modular Design for Future Expansion

The app architecture allows easy addition of new features like task persistence using Room, scheduled reminders with AlarmManager, and Firebase cloud sync.

Together, these features provide a clean, functional app ideal for quick and efficient task management.

4.4 Methodology (With Detailed Steps & Codes)

The development of the To-Do List Reminder App followed an iterative and modular development process. The key stages of the methodology are outlined below:

Step 1 – Requirements Gathering

The first step involved understanding the target users and their need for a basic, fast, and functional task management app. Simplicity, performance, and ease of use were prioritized.

Step 2 – Design and Architecture

The app was planned using a single-activity architecture with minimal UI components to enhance simplicity. The logic was separated into small, maintainable functions.

Step 3 – Development

The app was developed in Android Studio using Kotlin. UI elements such as EditText, Button, and ListView were laid out using XML. Task data was managed using ArrayAdapter.

Step 4 – Event Handling

Click and long-click listeners were implemented to handle user interactions. addButton adds the task, and a long-press on a list item triggers its removal.

Step 5 – Testing

The app was tested on multiple devices and emulators to ensure functionality, performance, and responsiveness. Various input types and edge cases (empty input,

multiple deletions) were handled gracefully.

Step 6 – Optimization

Unnecessary dependencies were removed to keep the APK size low. UI responsiveness was checked for all screen sizes.

This stepwise approach ensured that the app was built efficiently and is now ready for future improvements..

CODE:

ACTIVITY_MAIN.XML:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <!-- EditText for new item input -->
    <EditText
        android:id="@+id/newItemEditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter a new to-do item"
        android:inputType="text" />
```

```
<!-- Button to add the new item to the list -->
<Button
    android:id="@+id/addButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Add Item" />

<!-- ListView to display the to-do items -->
<ListView
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:divider="@android:color/darker_gray"
    android:dividerHeight="1dp"
    android:paddingTop="20dp" />

</LinearLayout>
```

MAINACTIVITY.KT:

```
package com.example.todoList233

import android.os.Bundle
import android.widget.Button
import android.widget.EditText
```

```
import android.widget.ListView
import android.widget.ArrayAdapter
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var addButton: Button
    private lateinit var newItemEditText: EditText
    private lateinit var listView: ListView
    private lateinit var adapter: ArrayAdapter<String>

    private val todoList = mutableListOf<String>()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        addButton = findViewById(R.id.addButton)
        newItemEditText = findViewById(R.id.newItemEditText)
        listView = findViewById(R.id.listView)

        // Set up adapter and connect it with the ListView
        adapter = ArrayAdapter(this, android.R.layout.simple_list_item_1, todoList)
        listView.adapter = adapter

        // Add new item to the list
        addButton.setOnClickListener {
            val newItem = newItemEditText.text.toString()
```

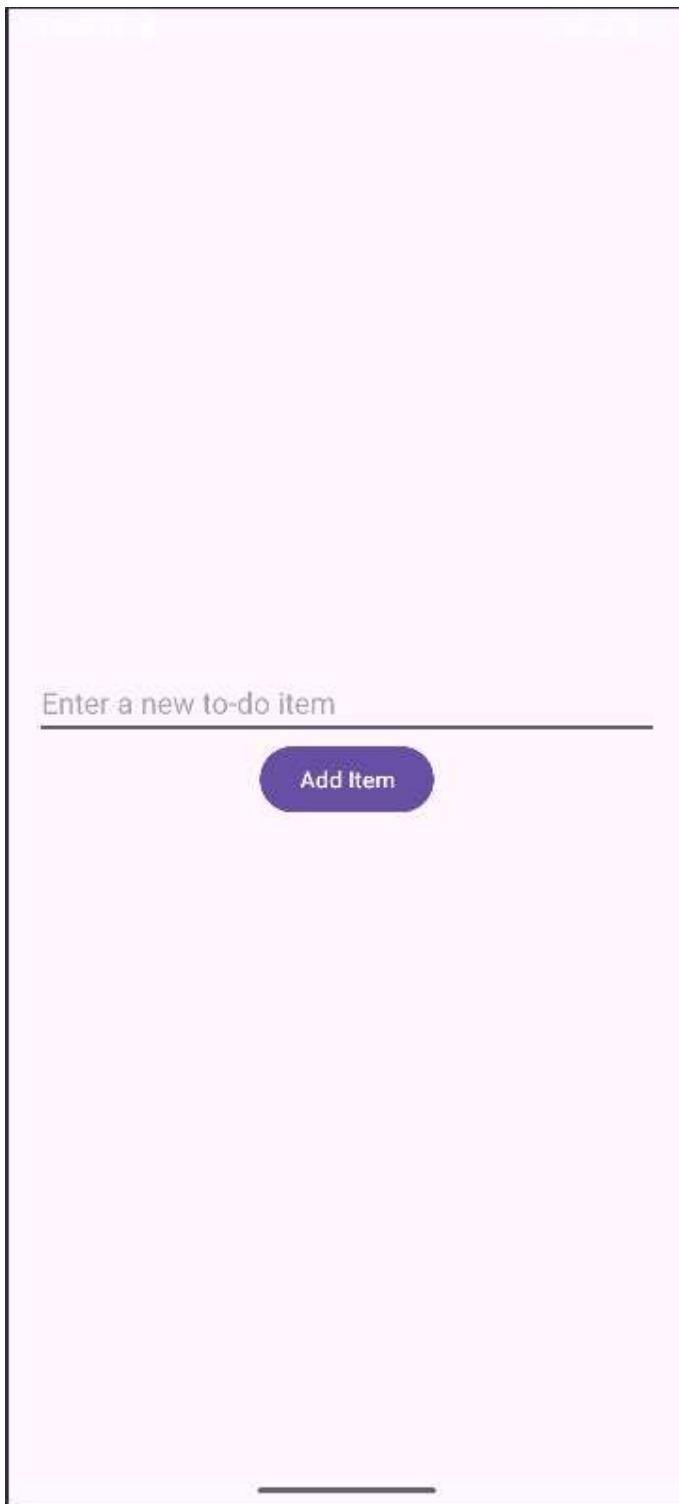
```
    if (newItem.isNotEmpty()) {
        todoList.add(newItem)
        adapter.notifyDataSetChanged()
        newItemEditText.text.clear()
    }
}

// Long click to delete an item
listView.setOnItemLongClickListener { _, _, position, _ ->
    val itemToDelete = todoList[position]
    todoList.removeAt(position)
    adapter.notifyDataSetChanged()
    Toast.makeText(this, "$itemToDelete deleted", Toast.LENGTH_SHORT).show()
    true // Return true to indicate the long click was handled
}
}
```

4.5 Tools & Technologies Used

Tool/Technology	Purpose
Android Studio	Integrated Development Environment (IDE) for Android
Kotlin	Programming language for app development
XML	Used for designing the UI layout
ArrayAdapter	Used to link the list data with the ListView
EditText	Widget for user input of task text
Button	Triggers the addition of new tasks
ListView	Displays the list of tasks in a scrollable view
Toast	Provides user feedback on actions like deletion
Logcat & Emulator	Used for debugging and real-time app testing

CHAPTER 5 OUTPUT AND SCREENSHOTS



Study for exam

Add Item

Enter a new to-do item

Add Item

Play at evening



Study for exam deleted

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION

The To-Do List Reminder Android Application has been successfully developed and tested using Android Studio and Kotlin. The application meets its primary goal of providing users with a clean and efficient interface to manage daily tasks. With the ability to add and delete tasks dynamically, the app ensures that users can quickly jot down reminders and remove them once completed, without navigating through multiple screens or dealing with complex features.

The app's functionality is focused on simplicity and offline usability. Unlike other task management apps that require logins, internet connectivity, or external dependencies, this application provides a seamless offline experience, with all task data stored in-memory during runtime. This approach makes the app lightweight and suitable for use on a wide range of Android devices.

The real-time UI update mechanism via ArrayAdapter, combined with intuitive interaction like long-press deletion, ensures a smooth user experience. Through this project, core Android development concepts such as event listeners, layout design, and UI feedback mechanisms were effectively demonstrated and implemented.

In conclusion, the project successfully showcases the power of minimalist design in mobile app development and highlights how essential features can be implemented without relying on heavy frameworks or third-party services. The foundation laid by this app allows for multiple extensions, and the modular architecture ensures that future improvements can be incorporated with ease.

6.2 FUTURE ENHANCEMENTS

While the current version of the app fulfills its core objective, several enhancements can be considered to improve user experience and scalability:

- **Persistent Storage:** Implementing Room or SQLite to save tasks even after app restarts.
- **Reminder Notifications:** Using AlarmManager to alert users about tasks at specific times.
- **Cloud Sync:** Integrating Firebase to allow cross-device task syncing and user accounts.
- **Dark Mode:** Adding UI themes to enhance accessibility and battery efficiency.
- **Task Categories:** Enabling users to organize tasks by priority or type (e.g., Work, Personal).
- **Data Export:** Supporting CSV/PDF exports for users who wish to keep track of completed tasks.
- **Voice Input:** Integrating voice-to-text APIs to allow task entry using speech.

These features will elevate the application from a basic tool to a more robust and feature-rich productivity app while retaining the ease-of-use that defines its core.

REFERENCES

- T. R. Kudale and A. A. Ingle, “Android Based To-Do List Application,” IJCSIT, 2015.
- P. Sharma and S. Goyal, “Task Management Application using Android,” IJRTE, 2019.
- R. Rani and K. Devi, “Design and Development of Task Reminder System using Android,” IJARCSSE, 2017.
- N. Patil and M. Thakare, “To-Do List App Using Firebase,” IRJET, 2021.
- A. Jain and K. Sahu, “Smart Task Management System,” IJERT, 2018.
- S. Bose and R. Roy, “Mobile Task Tracker Using Android,” IJETT, 2020.
- M. S. Alvi and F. Khan, “Reminder-Based Scheduling System on Android,” JACET, 2020.
- A. Mehta and D. Sharma, “Personal Organizer Android Application,” IJCSE, 2016.
- K. Gupta and A. Srivastava, “User-Centric To-Do List Application with Prioritization,” IJCA, 2018.

