**Course Instructor: Ms. Mariam Memon  
Semester: 6th  
Year: 3rd  
Department: Software Engineering, MUET Jamshoro**

**MAD CEP Report(Functional To Do List)**

**Submitted By**

***22SW33 & 22SW124***

**A report with following sections:**

**• Real World Problem Identification**

**• Proposed Solution**

**• Responsive User Interfaces (Screenshots of your app on different screens & platforms)**

**• Data Storage (With justification for using a particular database)**

**• [Optional Section] APIs/Packages/Plug-ins (if used with justifications for using them).**

**• Issues and Bugs Encountered and Resolved during Development**

## ****1. Real World Problem Identification****

In today’s fast-paced lifestyle, people often struggle to manage daily tasks efficiently. Missing deadlines, forgetting important chores, and poor prioritization lead to productivity loss. Many task management apps are either too complex or require a paid subscription.

Therefore, there is a need for a **simple, functional, and responsive To-Do List app** that helps users easily manage their tasks, set priorities, and stay organized without requiring login or complex setup.

## ****2. Proposed Solution****

The **Functional To-Do List App** was developed to help users efficiently organize and manage their daily tasks. The app allows users to:

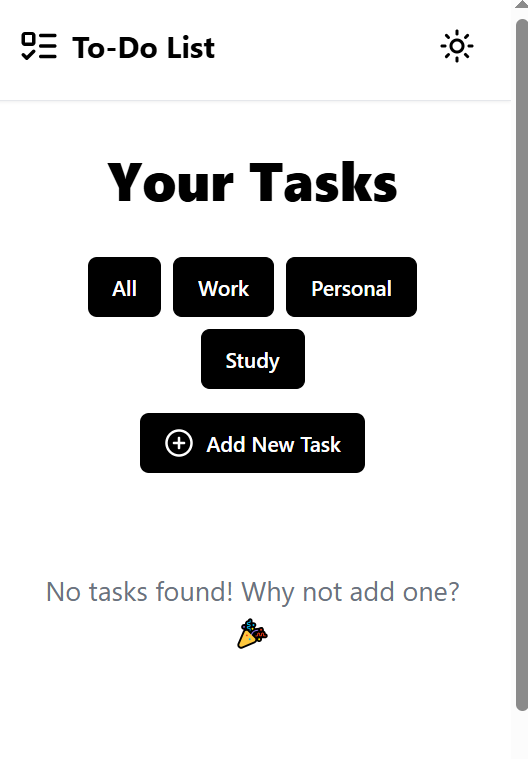
* Add, edit, and delete tasks.
* Mark tasks as completed or pending.
* Categorize tasks (e.g., personal, work, urgent).
* Store data locally so that tasks persist even after the app is closed.
* Provide a clean, responsive interface for both mobile phones and tablets.

This solution was implemented using **Flutter**, ensuring **cross-platform compatibility** for both Android and iOS devices.

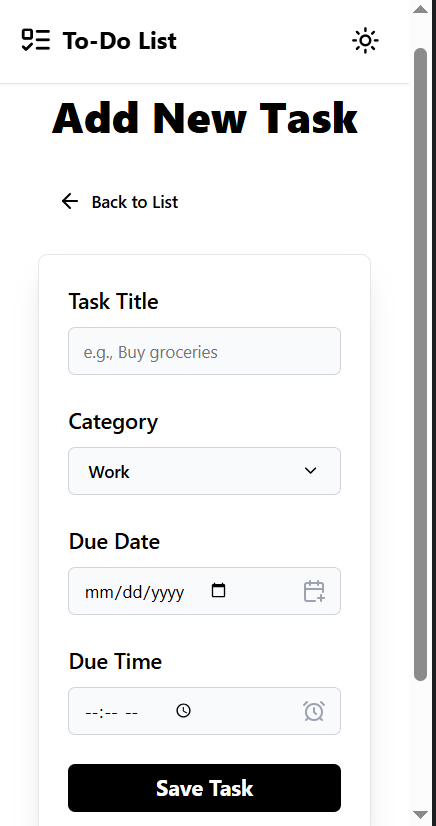
## ****3. Responsive User Interfaces****

The app’s UI was designed using **Flutter’s responsive layout widgets** like Expanded, Flexible, and MediaQuery to maintain consistent scaling across various screen sizes and platforms.

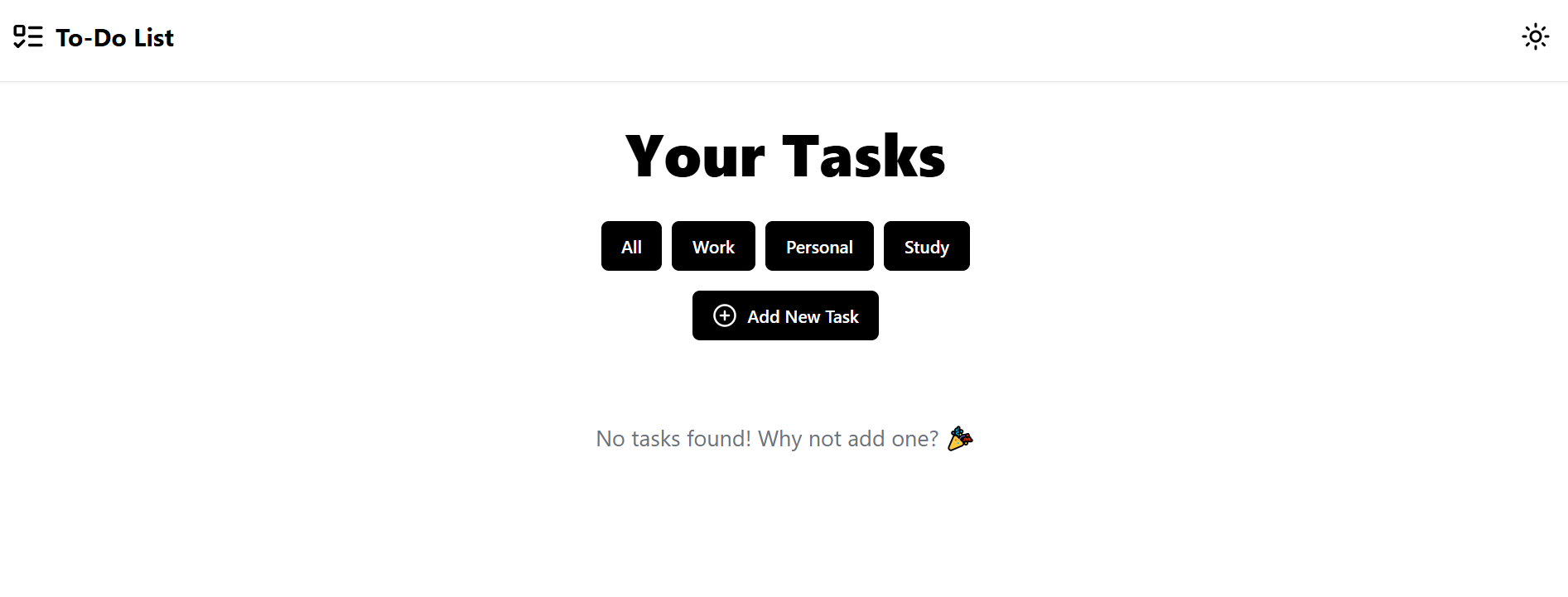
## ****Mobile Responsive (Homepage)­****



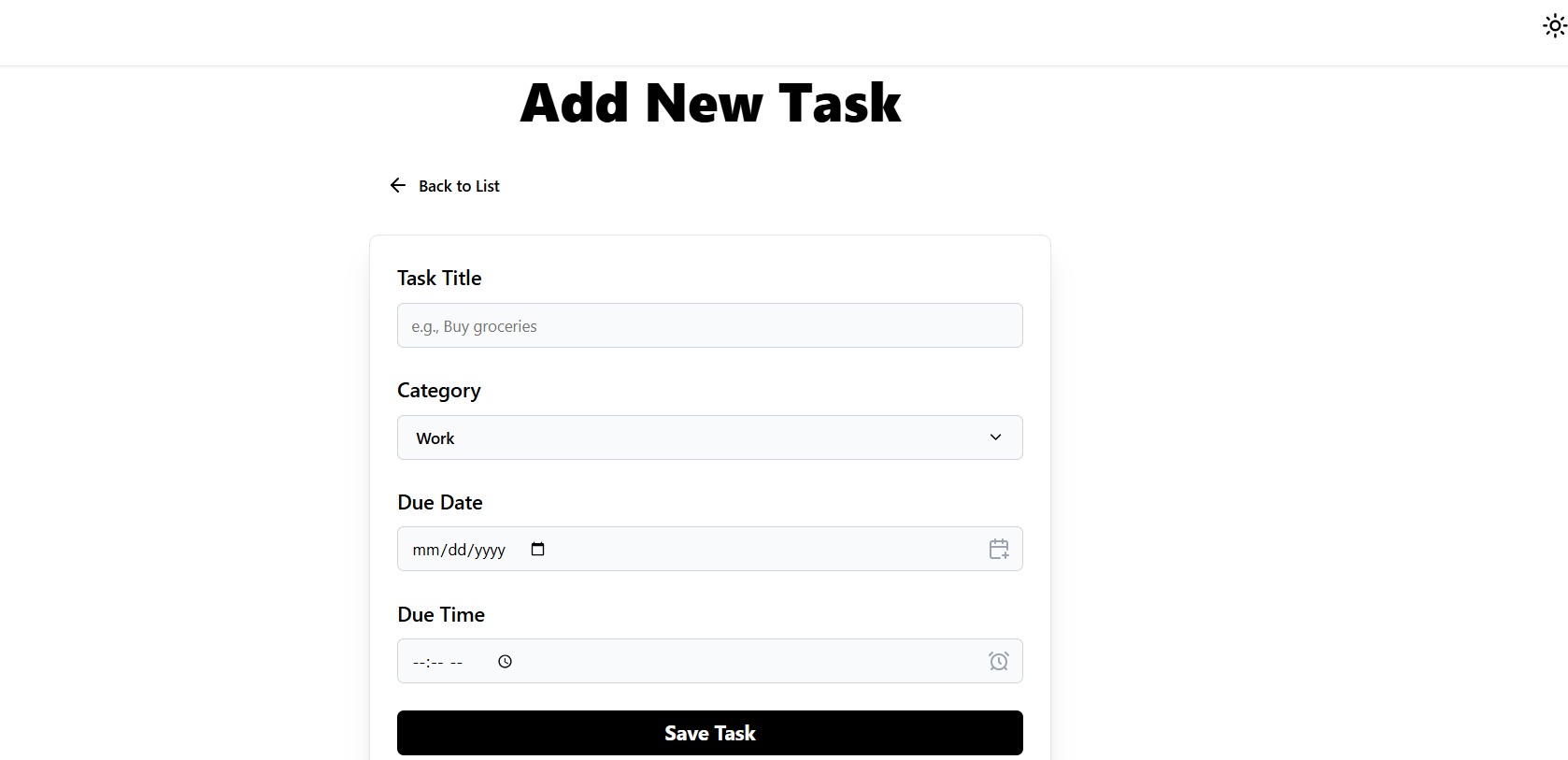
## ****(navigation page)­****



## ****Laptop Responsive****



## ****(navigation page)­****



## ****4. Data Storage****

The app uses **Firebase Firestore** as its cloud-based database.

### ****Justification for Firebase:****

* **Real-time Synchronization:** Any update (add, delete, edit) instantly reflects across all connected devices.
* **Scalability:** Firebase can handle multiple users without affecting performance.
* **Security:** Provides built-in authentication and data rules for safe data access.
* **Cross-Platform Support:** Firebase integrates seamlessly with both Android and iOS apps built on Flutter.

**Collection: tasks**

**├── Document ID: task\_001**

**│ ├── title: "Buy groceries"**

**│ ├── description: "Milk, eggs, bread"**

**│ ├── isCompleted: false**

**│ ├── createdAt: 2025-10-22**

## ****5. APIs/Packages/Plug-ins Used****

**Packages and Plugins:**

* cloud\_firestore: Firebase Firestore database integration.
* firebase\_core: Initializes Firebase in the app.
* provider: For efficient state management.
* flutter\_slidable: Adds slide-to-delete functionality.
* intl: Formats timestamps and dates.

**Justification:**

* These packages are official, reliable, and lightweight.
* Ensure efficient performance and minimal latency during data fetch operations.
* Simplify data flow and state updates across the app’s widgets.

## ****6. Issues and Bugs Encountered & Resolved****

| ****Issue**** | ****Cause**** | ****Resolution**** |
| --- | --- | --- |
| Data not showing after adding task | Delay in Firebase read operation | Added async/await and real-time listener using StreamBuilder |
| Task duplication in Firestore | Duplicate document IDs | Used autoID() method to generate unique document IDs |
| UI distortion on tablet screens | Fixed pixel-based design | Replaced with MediaQuery and Flexible layout |
| Dark mode color conflicts | Inconsistent theme data | Applied consistent ThemeData configuration |
| Network delay during sync | Firebase latency | Added loading indicator using CircularProgressIndicator |

## ****GitHub Link****

🔗 **https://github.com/Feroz-33/Functional-To-do-/tree/main**