CheckMe – Todo App Documentation

App Name: Godefroid CheckMe
Platform: Flutter
State Management: Riverpod
Version: 1.0.0
Purpose: A user-friendly Todo app designed to manage tasks, supporting features like adding, editing, deleting, marking tasks as completed, and switching themes.

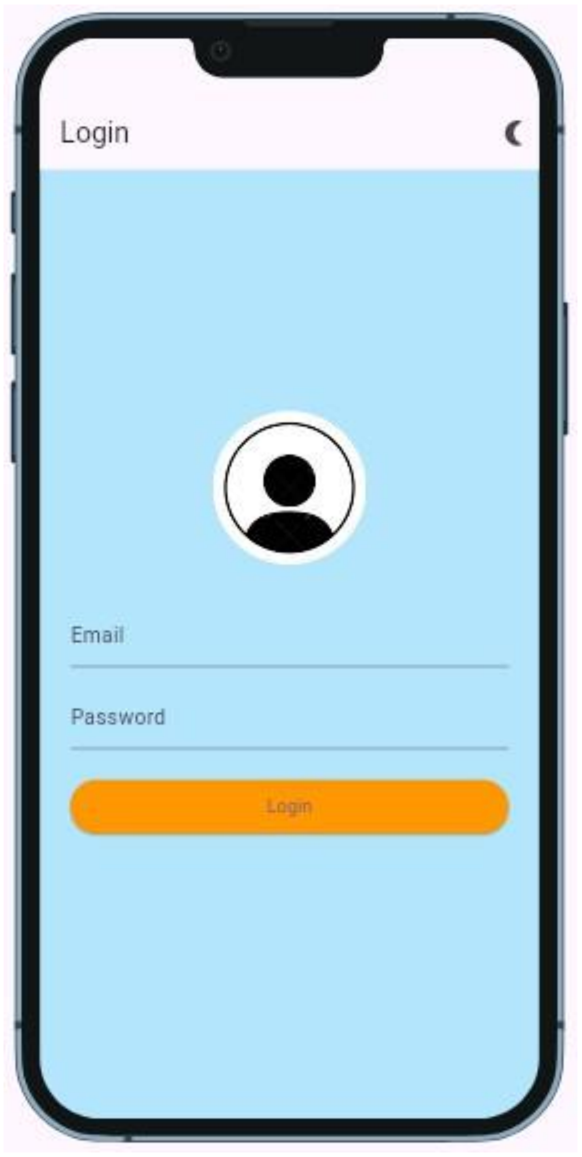
Project Overview

CheckMe is a Todo List application that allows users to manage tasks efficiently. It features functionalities such as adding, editing, deleting, and marking tasks as complete. The app is built using **Flutter** and follows best practices for state management, design, and Firebase integration.

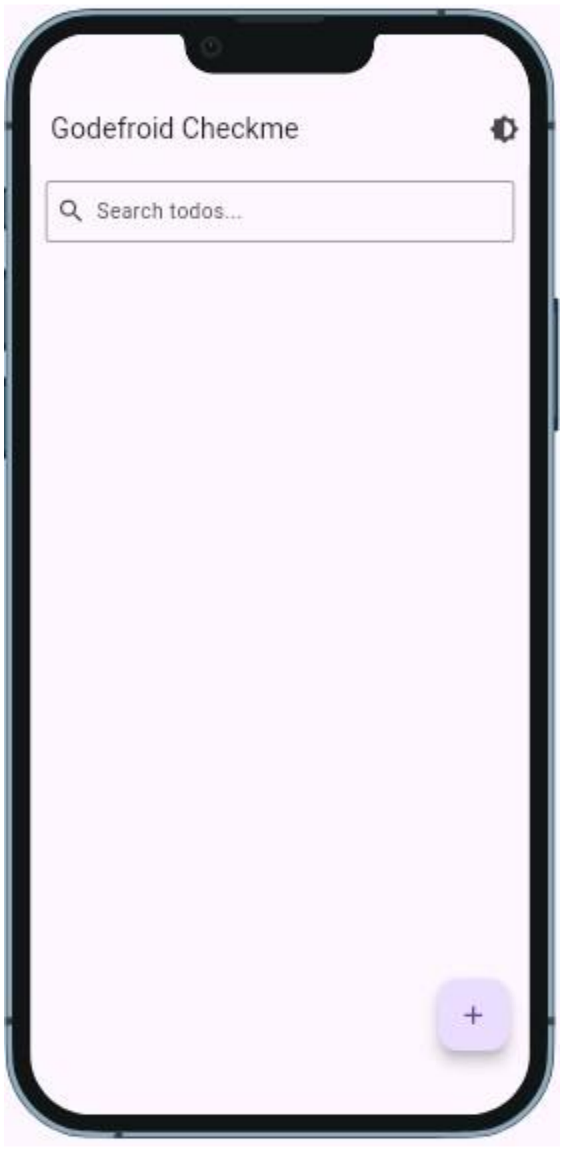
This app is designed for beginners, providing hands-on experience with common Flutter tools and libraries, including state management with Riverpod, Firebase Firestore integration for data storage, and responsive design.

Key Features

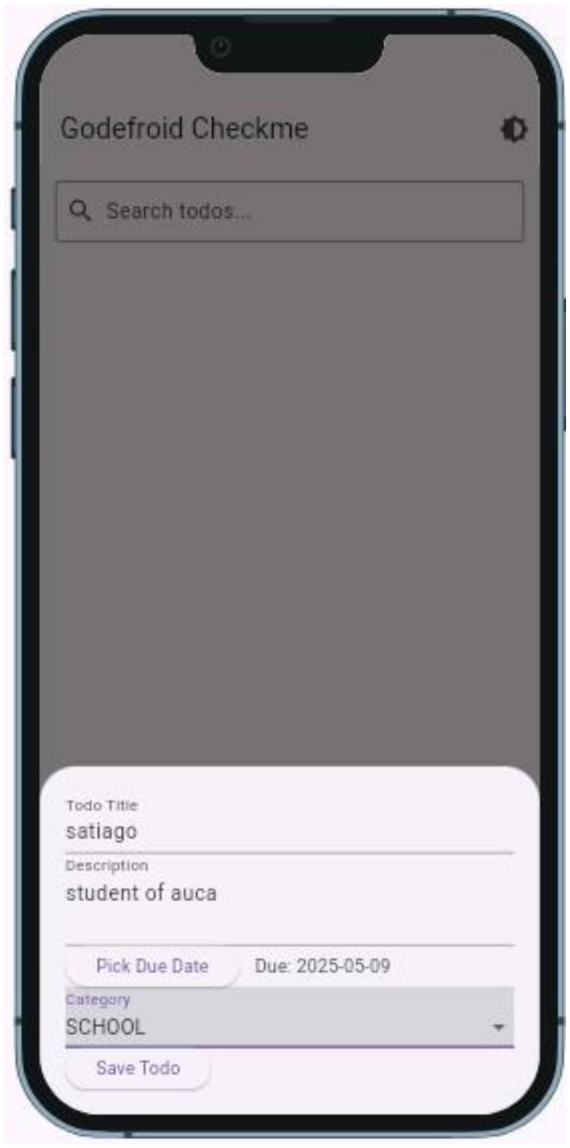
1. **Login Screen:**
    - Allows users to sign in using their email and password.
    - Includes input validation for email and password fields.
    - Users are redirected to the home screen upon successful login.



2. **Home Screen (Todo Dashboard):**
    - Displays a personalized welcome message.
    - Shows a list of tasks with checkboxes to mark them as complete.
    - Tasks can be added, edited, and deleted.

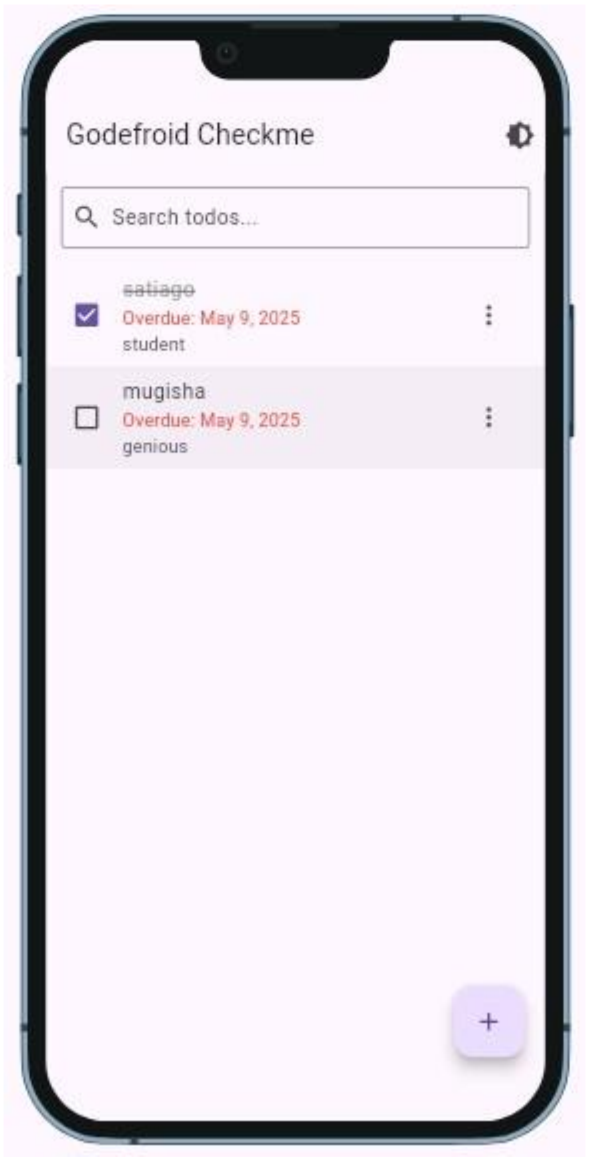


3. **Add Todo Feature:**
   - A floating action button (FAB) allows users to add new tasks.
   - Users can enter a task title and optional description.
   - Tasks are saved using **setState** for local state management.



Todo Title
satiago
Description
student of auca
Due: 2025-05-09
Category
SCHOOL

4. **Mark as Done:**
    - Tasks can be marked as complete using a checkbox.
    - Completed tasks are visually distinguished (strikethrough or faded).

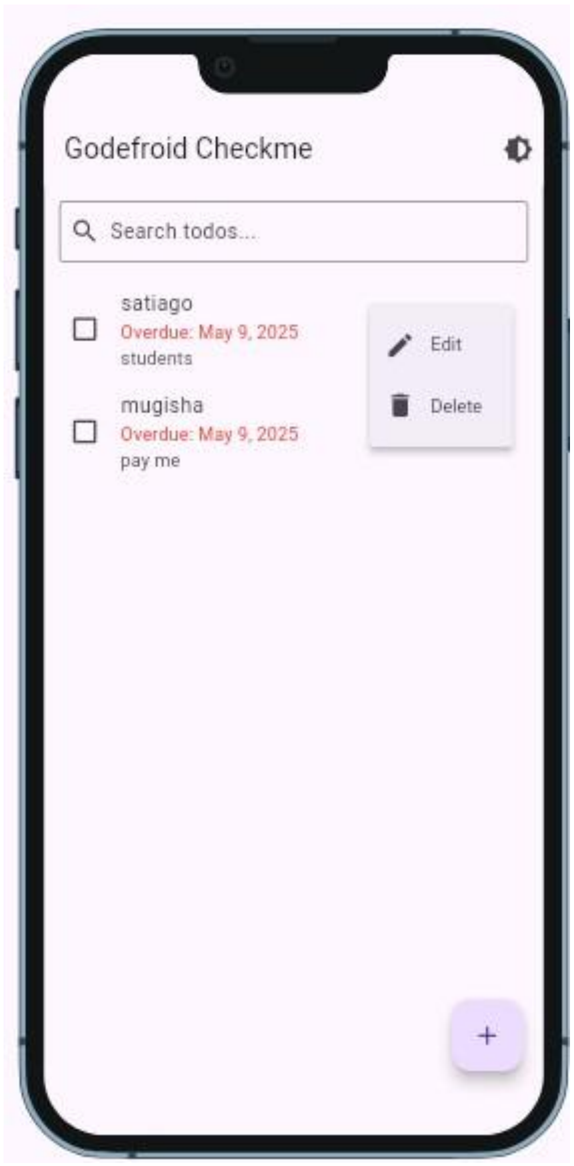

Godefroid Checkme

Search todos...

~~satiago~~
Overdue: May 9, 2025
student

mugisha
Overdue: May 9, 2025
genious

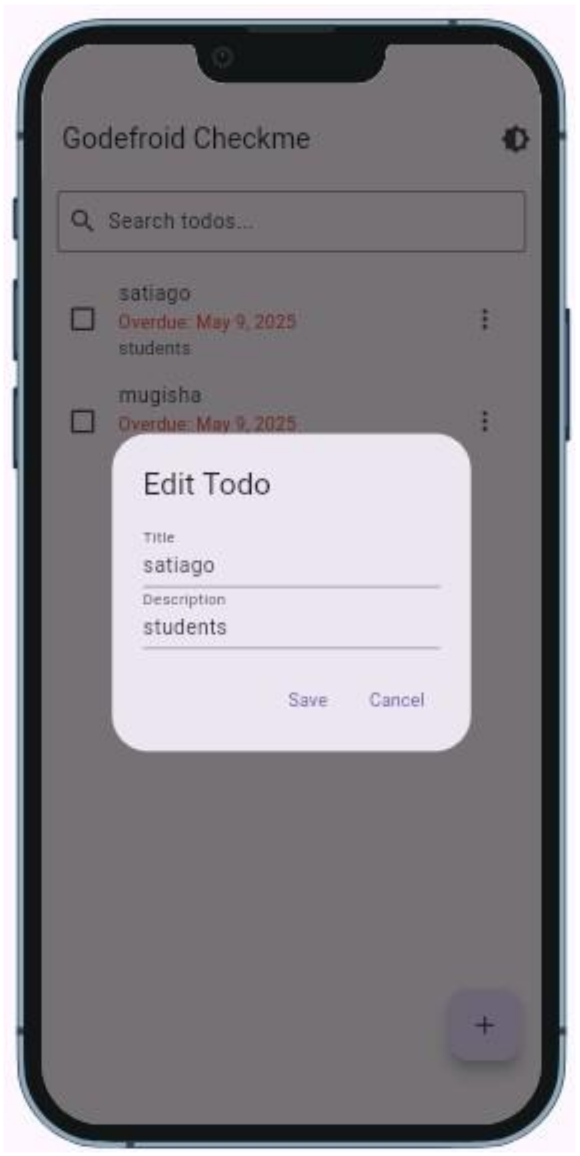
5. **Delete Todo:**
    - Tasks can be deleted by swiping or long-pressing them.
    - Users can confirm deletions for better user experience.



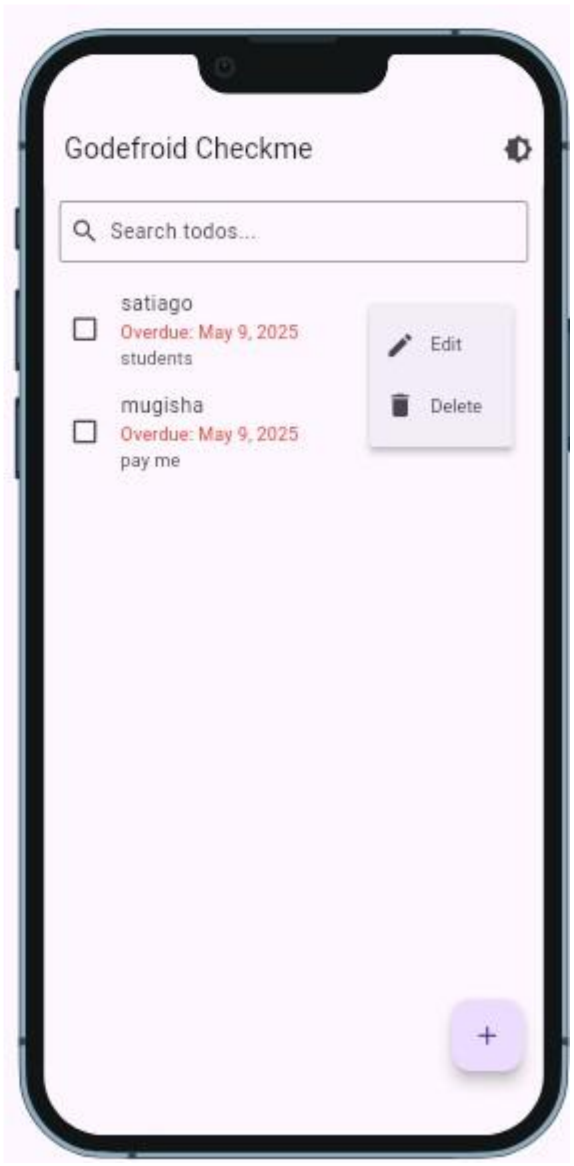
satiago
Overdue: May 9, 2025
students

mugisha
Overdue: May 9, 2025
pay me

6. **Edit Todo (Pen Icon):**
    - Users can edit the details of an existing task (title, description).
    - The **pen icon** next to a task triggers the edit functionality.



7. **Menu Dots (More Options):**
   - The menu dots allow users to access more options for each task, including deleting it.

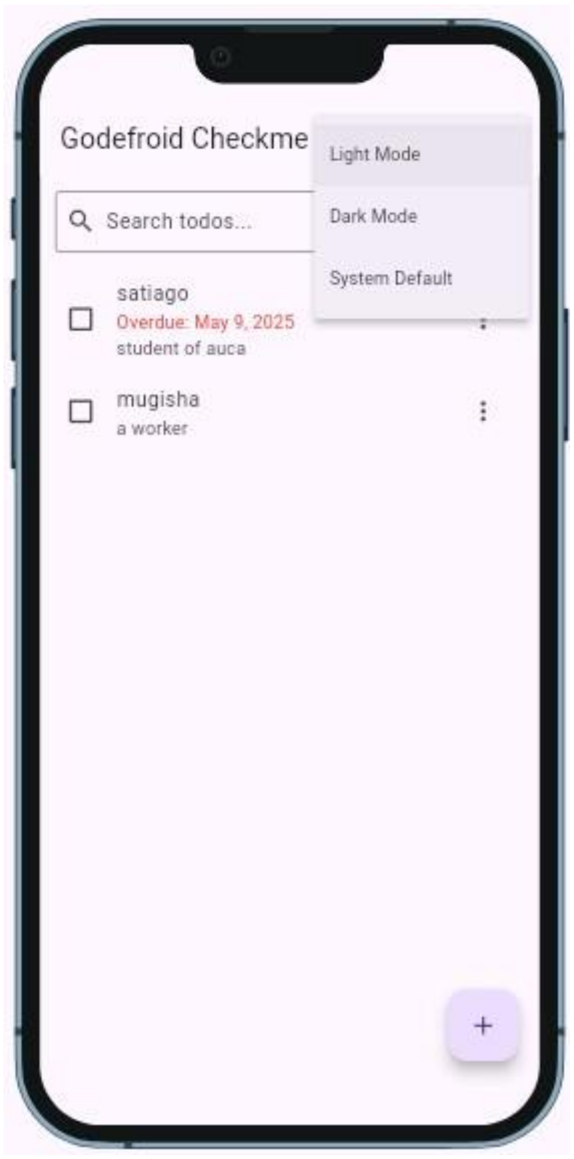

Godefroid Checkme

Search todos...

satiago
Overdue: May 9, 2025
students

mugisha
Overdue: May 9, 2025
pay me

8. **Theme Support:**
    - Toggle between Light and Dark themes with a button in the AppBar.


Godefroid Checkme

Search todos...

satiago
Overdue: May 9, 2025
student of auca

mugisha
a worker

9. **Due Date:**
   - Users can set a due date for each task.
   - Tasks show an “Overdue” notification if the due date passes.

UML Design Overview – *Godefroid CheckMe: Smart Todo App*

Title:

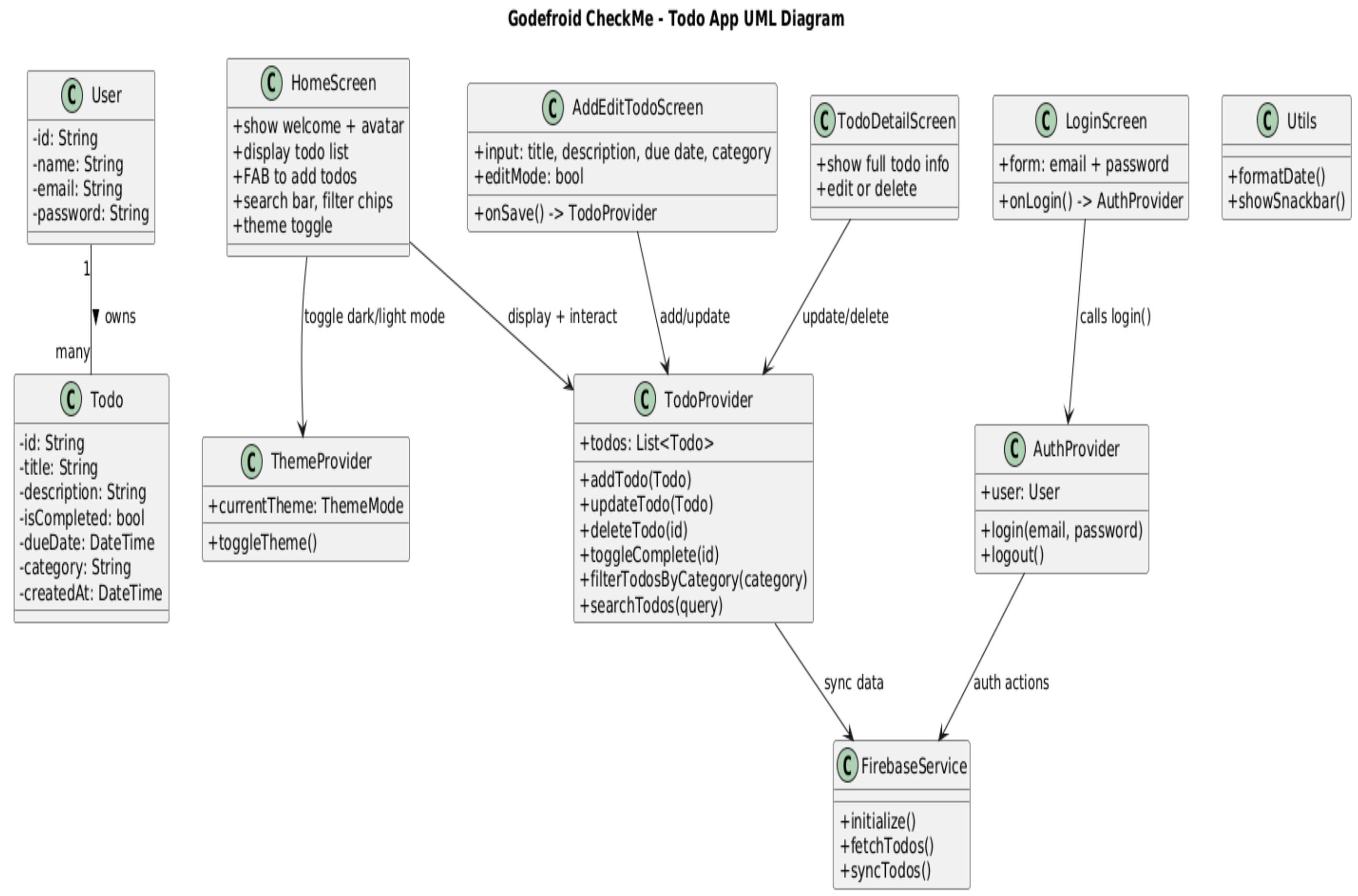
System Architecture & UML Design for Godefroid CheckMe

Description:

This UML class diagram represents the complete architecture and logical flow of the **Godefroid CheckMe** mobile application — a smart Todo app built using Flutter and Riverpod. The diagram captures the relationships between the major components such as user authentication, todo management, UI screens, and state management.

The app uses a layered structure where:

- **Model classes** (User, Todo) represent the data schema.
- **State management** (with AuthProvider, TodoProvider, ThemeProvider) handles logic and business rules.
- **UI screens** (e.g., LoginScreen, HomeScreen, AddEditTodoScreen) interface with providers to display and manipulate data.
- **Utilities and services** (like FirebaseService and Utils) handle backend interaction and helper functions.



App Structure

1. Login Screen:

The login screen allows users to authenticate. It uses:

- **TextFormField** widgets for email and password inputs.
- **ElevatedButton** for login action.
- **CircleAvatar** for user avatar.
- **Form validation** to ensure correct input.
- **Navigation** to the home screen upon successful login.

2. Home Screen (Todo Dashboard):

The home screen is the main interface for managing tasks. It includes:

- A **ListView** to display todos.
- **Checkbox** to mark todos as complete.
- **FloatingActionButton** to open a form for adding new todos.
- **Menu options (dots)** to delete tasks.
- **Task cards** for displaying title and description.

3. Todo Details Screen:

- A detailed view for editing and viewing tasks.
- Shows a task's title, description, creation date, and edit button.

State Management:

The app utilizes **Riverpod** for state management, which offers a more flexible and scalable approach compared to traditional setState. Riverpod enables managing both local and global states.

- **Global State:** To manage the overall theme (Light/Dark mode) and task data (todos).

- **Local State:** To manage individual screens and their states, such as text field input and task completion.

Firebase Integration:

The app integrates Firebase for storing tasks. Specifically, it uses:

- **firebase_core:** Initializes Firebase services.
- **cloud_firestore:** Connects to Firestore, where todos are saved and retrieved in real-time.

Dependencies Used:

```
dependencies:
  flutter:
    sdk: flutter
  flutter_riverpod: ^2.4.0
  google_fonts: ^6.1.0
  intl: ^0.19.0
  flutter_hooks: ^0.20.3
  firebase_core: ^2.18.0
  cloud_firestore: ^4.15.1
  cupertino_icons: ^1.0.8
  device_preview: 1.2.0
  uuid: ^3.0.4
```

Description of Dependencies:

1. **flutter_riverpod:** For state management.
2. **google_fonts:** To easily use custom Google fonts.

3. **intl:** For formatting locale-sensitive data like dates and numbers.
4. **flutter_hooks:** For enhancing code reusability and state management with hooks.
5. **firebase_core:** Firebase initialization.
6. **cloud_firestore:** For real-time data storage and retrieval.
7. **cupertino_icons:** For adding iOS-style icons.
8. **device_preview:** To preview the app on various devices during development.
9. **uuid:** For generating unique identifiers for todos.

How to Set Up the Project:

Step 1: Clone the Repository

Clone the project to your local machine using Git.

git clone <repo_url>

Step 2: Install Dependencies

Open the project directory and run the following command to install dependencies:

flutter pub get

Step 3: Firebase Setup

1. Create a Firebase project at Firebase Console.
2. Set up Firestore Database and enable Firestore in your Firebase project.
3. Download the google-services.json file and place it in the android/app directory.
4. Initialize Firebase in your Flutter app by adding firebase_core and cloud_firestore dependencies in pubspec.yaml.

Step 4: Run the App

After the setup is complete, you can run the app with the following command:

```
flutter run
```

Step 5: Build for Release

To build the app for release, run the following:

```
flutter build apk
```

Or for iOS:

```
flutter build ios
```

Conclusion:

The **CheckMe Todo App** is a feature-rich yet beginner-friendly app that helps users manage their tasks with ease. It demonstrates a wide variety of skills such as:

- **State management** using Riverpod.
- **Firebase Firestore integration** for real-time data storage.
- **UI design principles** with Flutter.
- **User experience enhancements** such as theming and task management.

This app serves as an excellent starting point for developers looking to get into Flutter and build functional, visually appealing apps with modern features.