

# Hospital Flutter App Documentation

## Overview

The Hospital Flutter App is a mobile application built using the Flutter framework. It provides functionalities related to managing patient records in a hospital setting. The app allows users to perform various tasks, including viewing a list of patients, adding new patients, editing patient details, and more.

## Table of Contents

- [Getting Started](#)
- [Application Structure](#)
- [Routes and Navigation](#)
- [Components and Widgets](#)
- [Data Handling](#)

## Getting Started

To get started with the app, ensure that you have the Flutter SDK installed on your development machine. You can clone the app's repository from GitHub and use a compatible code editor (such as Visual Studio Code) to open the project.

```
git clone <repository-url>
cd hospital_flutter_app
```

Install the required dependencies using:

```
flutter pub get
```

Run the app using:

```
flutter run
```

## Application Structure

The app is structured into several key components:

- **lib:** Contains the main source code of the application.
  - **data:** Includes data-related functionality, such as API interactions and repository classes.
  - **presentation:** Contains UI-related code, including pages and widgets.
  - **routes.dart:** Defines the app's routing logic using the AppRouter class.

## Routes and Navigation

The app uses a navigation system to switch between different pages. The routing logic is managed by the `AppRouter` class defined in `routes.dart`. This class generates routes based on route settings and arguments.

Routes defined in the app include:

- `/`: Root route displaying a list of patients using `PatientListWidget`.
- `/login`: Login page for authentication using `LoginPage`.
- `/patientDetails`: Patient details page displaying patient information using `PatientDetailsWidget`.
- `/addPatient`: Page for adding new patients using `AddPatientWidget`.
- `/editPatient`: Page for editing patient details using `EditPatientWidget`.

## Components and Widgets

The app's UI is composed of various widgets and components:

- `PatientListWidget`: Displays a list of patients fetched from the API. Allows navigation to patient details and adding patients.
- `PatientDetailsWidget`: Shows detailed information about a selected patient. Provides options to edit and delete the patient.
- `AddPatientWidget`: Provides a form for adding new patient records. Validates input and sends data to the repository.
- `EditPatientWidget`: Allows editing existing patient details. Similar to `AddPatientWidget` but pre-populates fields with patient data.
- `LoginPage`: Provides a login screen for user authentication.

## Data Handling

Data management is handled through the `PatientRepository` class, which interacts with the API using the `ApiService` class. The repository class is responsible for fetching, adding, updating, and deleting patient records. Data models like `Patient` define the structure of patient records.

## Conclusion

The Hospital Flutter App provides an intuitive user interface for managing patient records in a hospital setting. Developers can extend and enhance the app's functionality by building upon the existing codebase and implementing new features.

For any questions or assistance, contact the project maintainers or refer to Flutter's official documentation.