



# WHATSAPP CLONE PROJECT

DONE BY :

Saipriyadarshini R

Pooja Sri J

Vedhalakshmi M

Swetha R



# Introduction:

We will be walking you through the process of building a user interface similar to WhatsApp using Flutter, a popular framework for building beautiful and performant mobile applications. We'll explore the key components of the app, understand how Flutter helps us achieve this, and discuss potential future enhancements.



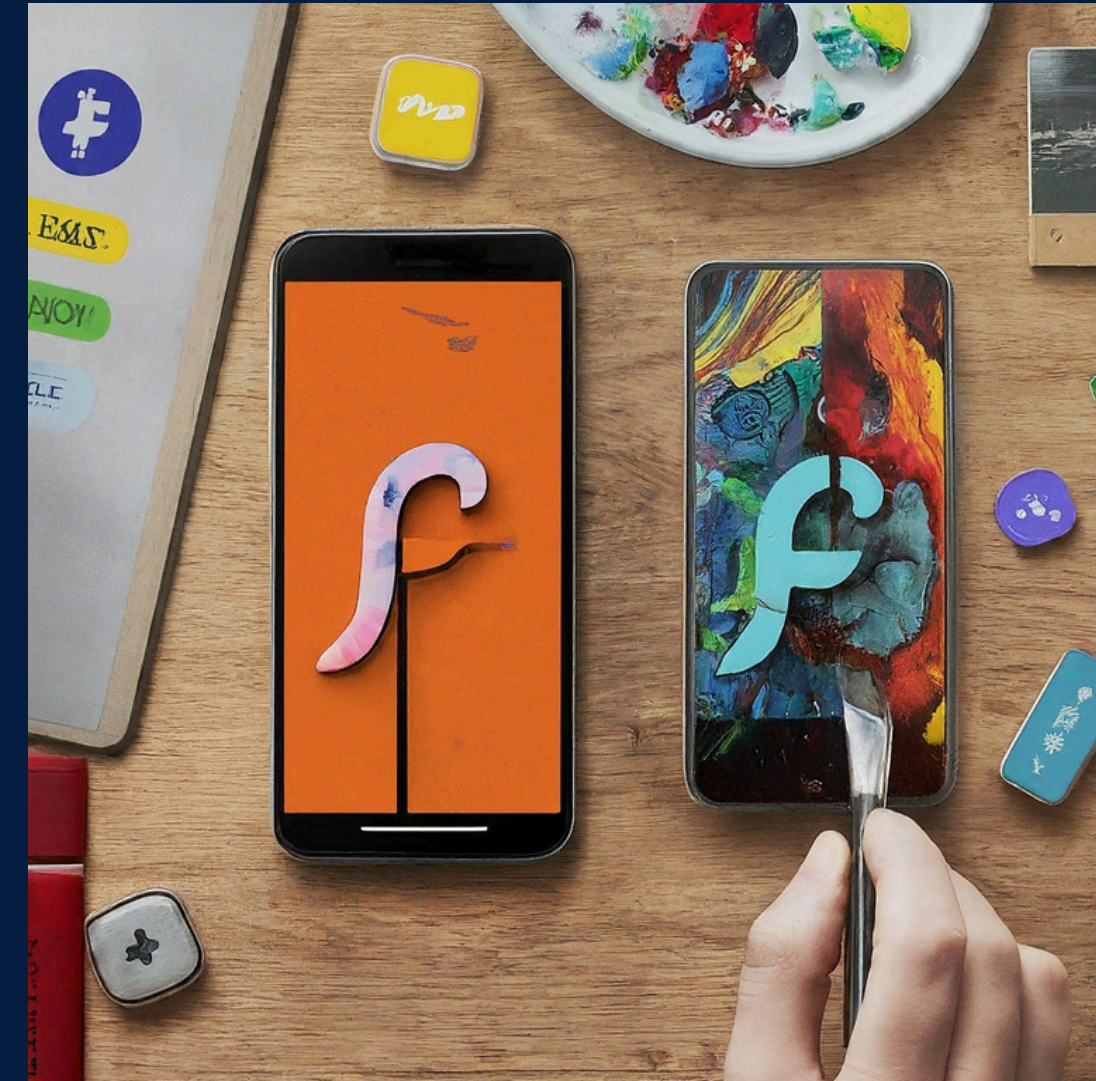
# Flutter:

- Flutter is an open-source UI framework created by Google.
- It allows developers to build beautiful, native-looking apps for mobile, web, desktop, and embedded devices using a single codebase.
- Flutter uses the Dart programming language, known for its simplicity and fast development cycles.



# Key Benefits Of Flutter:

- **Cross-platform development:** Write code once and deploy it to multiple platforms (iOS, Android, web, etc.) with minimal changes.
- **Hot reload:** See your code changes reflected in the app instantly, speeding up the development process.
- **Rich widgets:** Flutter provides a wide range of built-in widgets for creating complex UI elements with ease.
- **Customizability:** You have complete control over the UI and can create highly customized experiences.





# Why Build a WhatsApp-like UI?

- WhatsApp is one of the most popular messaging apps globally.
- Building a similar UI provides valuable learning experiences for Flutter developers.
- We'll explore concepts like state management, building layouts with widgets, and handling user interaction.
- This project serves as a foundation for building more complex mobile applications in the future.



# Imports and Main Function:

- import 'package:flutter/material.dart'; This line imports the Flutter framework, which is essential for building user interfaces for mobile, web, desktop, and more.
- void main(){ ... } This is the entry point of the application. The runApp function is called here, which takes a Widget as an argument (in this case, const WhatsAppApp()).

## WhatsAppApp Class:

- class WhatsAppApp extends StatelessWidget { ... } This class defines the main application widget. It inherits from StatelessWidget, meaning its appearance doesn't change dynamically.
- const WhatsAppApp({super.key}); The constructor creates an instance of WhatsAppApp. The super.key argument is a unique identifier for the widget.
- @override Widget build(BuildContext context){ ... } This method builds the UI for the WhatsAppApp. It returns a MaterialApp widget, which is the foundation for most Flutter applications.

# MaterialApp Configuration:

- **title: 'WhatsApp':** This sets the application's title, which appears in the app switcher on mobile devices.
- **theme:** This defines the app's visual theme, including the primary color (primaryColor: const Color(0xff075E54)) set to the familiar WhatsApp green.
- **appBarTheme:** This configures the app bar's appearance, applying the same green background color.
- **home:** const WhatsAppHomePage(): This sets the WhatsAppHomePage as the initial screen displayed in the app.

## WhatsAppHomePage Class:

- **class WhatsAppHomePage extends StatefulWidget { ... }:** This class defines the widget for the WhatsApp home page. It inherits from StatefulWidget because its content might change over time.
- **const WhatsAppHomePage({super.key});:** The constructor creates an instance of WhatsAppHomePage.
- **@override State<WhatsAppHomePage> createState() => \_WhatsAppHomePageState();:** This method returns the state object (\_WhatsAppHomePageState) associated with the WhatsAppHomePage.

# WhatsAppHomePageState Class:

- class WhatsAppHomePageState extends State<WhatsAppHomePage> with SingleTickerProviderStateMixin { ... }: This class manages the state of the WhatsAppHomePage. It inherits from the parent State class and SingleTickerProviderStateMixin, which is required to use the TabController.
- late final TabController tabController;: This declares a TabController variable, which will be used to manage the tabs in the app bar.
- @override void initState() { ... }: This method is called when the widget is first created. It initializes the \_tabController with three tabs (length: 3) and uses this (the current state object) for synchronization (vsync: this).
- @override void dispose() { ... }: This method is called when the widget is disposed of. It's important to release resources like the \_tabController to avoid memory leaks.
- @override Widget build(BuildContext context) { ... }: This method builds the UI for the WhatsAppHomePage. It returns a Scaffold widget, which provides the basic app structure with an app bar, body, and floating action button.



# Scaffold Structure:

- **AppBar**: This defines the app bar with the following elements:
- **title**: `const Text('WhatsApp')`: Sets the app bar title.
- **actions**: These are buttons displayed on the right side of the app bar, currently non-functional icons for search and more options.
- **bottom**: `TabBar`: This creates a tab bar at the bottom of the app bar.
- **controller**: `_tabController`: Binds the tab bar to the previously created `_tabController`.
- **tabs**: This defines the three tabs using `Tab` widgets, each with an icon (`Icon(Icons.chat)`, `Icon(Icons.group)`, `Icon(Icons.call)`) representing chats, groups, and calls.
- **body**: `TabBarView`: This widget displays content based on the selected tab. It's connected to the `_tabController`.
- **children**: This list contains a widget for each tab:
- **ChatList()**: This widget (defined later) represents the chat list for the first tab.
- **Text('Groups')**: Placeholder text for the groups tab (needs to be replaced with a `GroupsList` widget).
- **Text('Calls')**: Placeholder text for the calls tab (needs to be replaced with a `CallsList` widget).
- **floatingActionButton**: This creates a floating action button at the bottom right corner, currently

# Conclusion:

- We've explored the key concepts of building a WhatsApp-like UI with Flutter.
- We've learned about Flutter's core components and how they can be used to create a visually appealing and functional mobile app.
- This project serves as a stepping stone for building more complex mobile applications in the future.