Google Developer Groups
On Campus SVIT, Vasad

🎯 Welcome to Build With AI 2.0! 🚀

💡 **Let's Learn, Build & Innovate Together!** 🚀
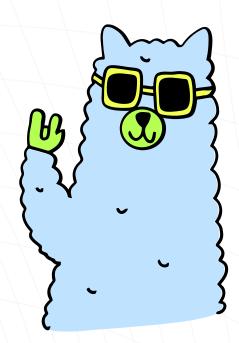
Agenda

- Introduction to Flutter
- API vs Packages
- Project Setup
- Basics
- Package Integration
- Hand-on Session
- Q&A and Next Steps

01

What is Flutter?

Key Features

Cross-platform

Single codebase
for all
(iOS, Android, Web,
Desktop)

Efficient

natively
compiled
applications

Hot Reload

Fast
development
with hot reload

02 APIs vs Packages

APIs vs Packages

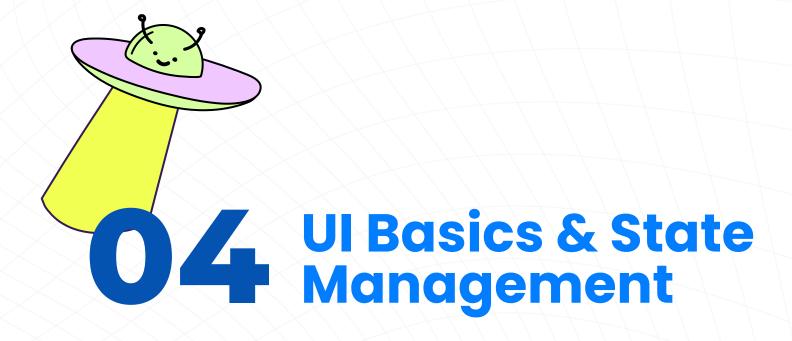| API	Packages
A set of rules for communication between applications	A pre-built library that simplifies interaction with an API or service.
You manually write code to make HTTP requests and handle responses	You use pre-built functions provided by the package.
Requires more code and error handling (e.g., HTTP status codes).	Abstracts complexity, making it easier to use.
e.g. Fetching data from Gemini API by making http requests	e.g. Using google_generative_ai package to interact with Google's AI

03 Setting Up the Flutter Project

Setting Up the Flutter Project

- Creating a new Flutter project
- Project structure overview
  - pubspec.yaml: Manages dependencies and assets.
  - android and ios: Platform-specific code.
  - test: For writing tests.
  - lib: Contains the main Dart code, we'll work mainly in the lib folder.

04 UI Basics & State Management

UI Basics

- Everything in Flutter is a widget (Text, Button, Image, etc.).
- Widgets are composable – you can nest them to create complex layouts.
- Two types of widgets:
  - StatelessWidget (UI doesn't change)
  - StatefulWidget (UI changes dynamically)

State Management

- State = Data that can change in an app
- Common state management approaches:
  - setState()
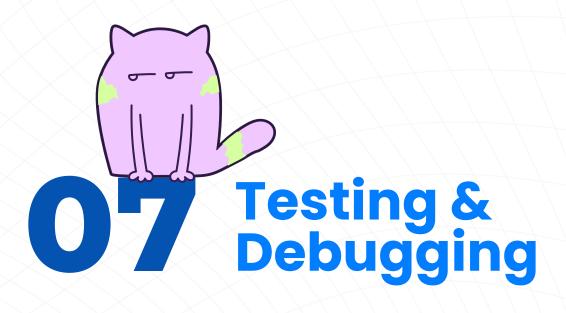 - Provider, Riverpod, BLoC

05 Packages

Packages

- What are Flutter packages?
- Where to find packages? pub.dev
- Commonly used packages
- How to install a package
- Run flutter pub get to install

06 Coding the App

Coding the App

- Home Screen
- Input Field – Add a TextField for user input
- Chat Bubbles – Display messages in a ListView
- Gemini Integration – Send messages and display AI responses
- Clearing the History - if time permits (DIY Session)
- Alternate Way of Integrating Gemini - if time permits

07 Testing & Debugging

Testing & Debugging

- Use print() to check values.

- Handle errors with try-catch to prevent crashes.

- Use Flutter DevTools to inspect UI and logs.

Thanks!

Any Questions?