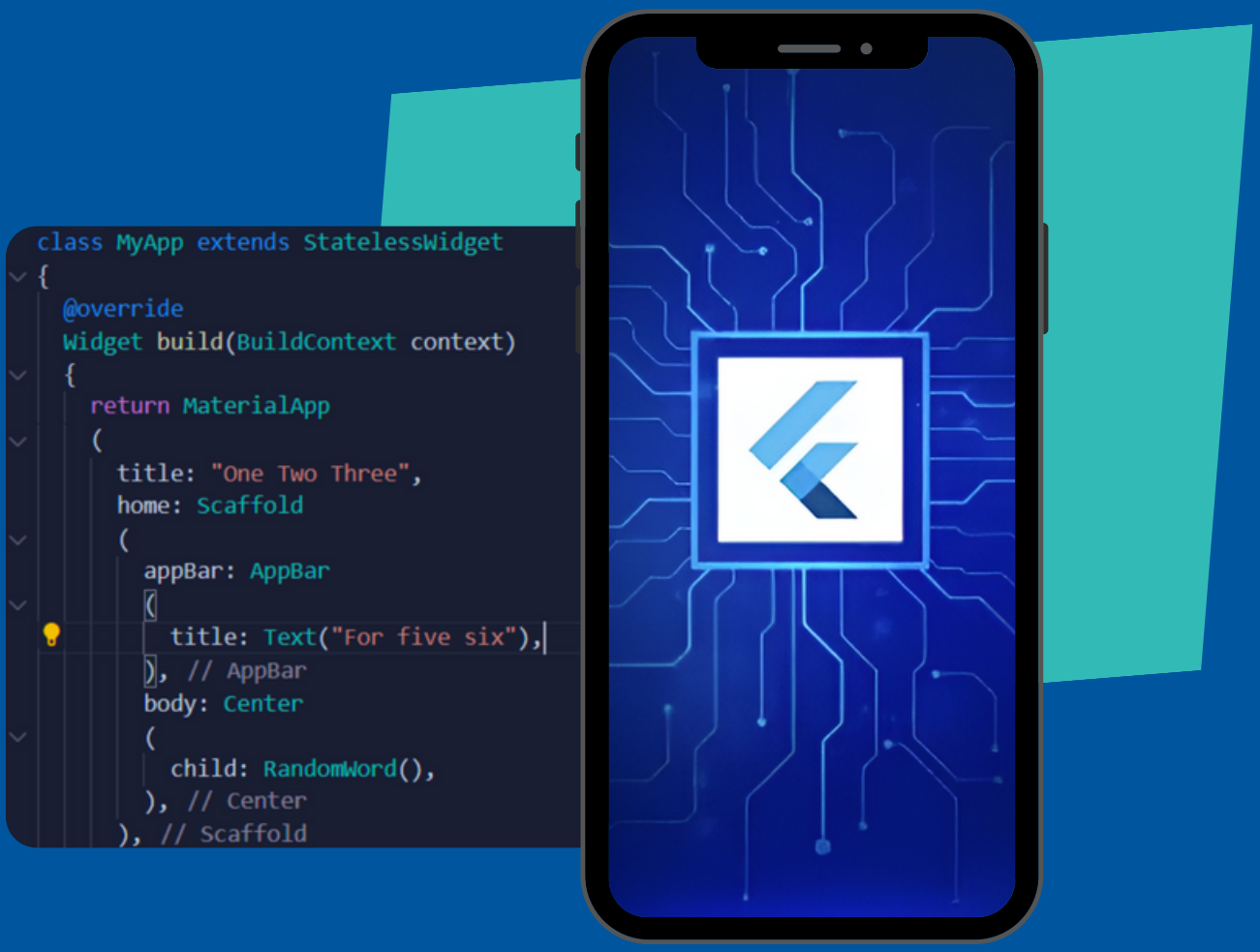
Flutter Syllabus

For Apprenticeship Training Program



Flutter AI ML

HINDTECH
LEARNING POINT

Skill India
कौशल भारत-कुशल भारत

N·S·D·C
National
Skill Development
Corporation
Transforming the skill landscape

Flutter is an open-source UI software development framework created by Google. It is designed for building natively compiled applications for mobile, web, and desktop from a single codebase. Flutter has gained significant popularity among developers due to its distinctive features and advantages.

FEATURES OF FLUTTER:

- Single Codebase, Multi-Platform: Write code once and run it on iOS, Android, Web, and desktop.
- Expressive UI Framework: Create beautiful and customizable user interfaces with a rich set of widgets.
- Fast Performance: Flutter compiles to native ARM code for near-native performance.
- Hot Reload: See instant code changes in the app during development.
- Rich Ecosystem: Access a wide range of packages and plugins from pub.dev.
- Dart Programming Language: Use Dart, a language with a strong static type system.

BASIC SESSION:

- Welcome To the Course
- What is Flutter?
- Flutter Uses Dart!
- One Code Base Multiple Platform
- Project Creation & Setting Up a Code Editor for Flutter Development
- Flutter Setup - Overview
- Windows Setup
- macOs Setup
- Running a First Flutter App
- Understanding Material Design
- About The Course

DURATION: 2 DAYS

OUTCOME:

1. **Introduction to Flutter and Dart:** Learn the basics of Flutter and Dart.
2. **Cross-Platform Development:** Understand how to use one codebase for multiple platforms.
3. **Setup and Configuration:** Set up Flutter on Windows and macOS, and configure a code editor.
4. **Practical Application:** Run your first Flutter app and understand Material Design principles.

PROJECT SESSION 2: BUILDING FLUTTER APPS:

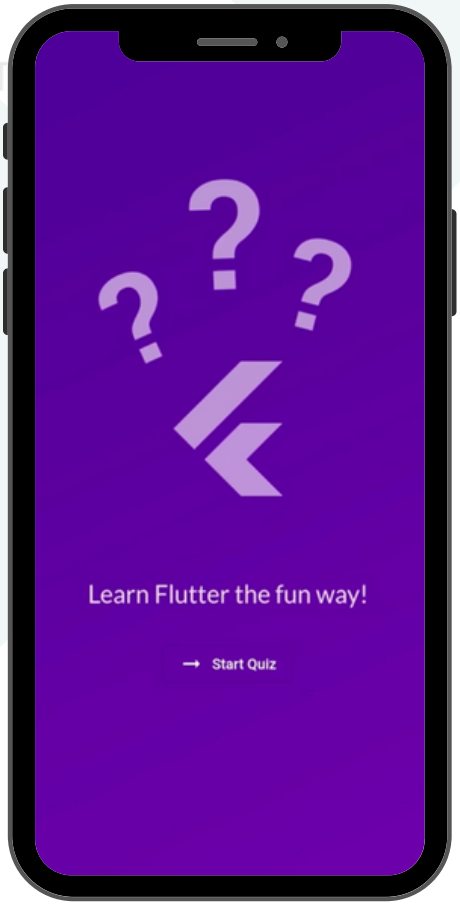
- You'll create Flutter apps, set up your environment, structure code, and build custom widgets. Gain hands-on experience for feature-rich apps.

FLUTTER & DART BASICS I - GETTING A SOLID FOUNDATION [ROLL DICE APP]

- Module Introduction
- Analysing A New Flutter Project
- From Dart To Machine Code
- How Programming Languages Work
- Starting From Scratch: Understanding Functions
- Importing Features From Packages
- How Flutter Apps Start
- Understanding Widgets
- Using a First Widget & Passing Values to Functions
- Positional & Named Arguments
- Deep Dive: Position & Named Arguments
- Combining Multiple Widgets
- Understanding "const" Values
- Building More Complex Widget Trees
- Understanding Value Types
- Configuring Widgets & Understanding Objects
- Working with "Configuration Objects" (Non-Widget Objects)
- Generics, Lists & Adding Gradient Colours
- How To Configure Widgets & Objects
- Practice: Styling Text
- Onwards to Custom Widgets: Why Do You Need Them?
- Understanding Classes
- Building Custom Widgets
- Working with Constructor Functions
- Splitting Code Across Files
- Practice: Create a Custom Widget
- Introducing Variables
- Variables & Types - Combining Two Key Concepts
- "final" & "const" - Special Kinds Of "Variables"
- Instance Variables (Properties) & Configurable Widgets
- Practice: Reusable Widgets & Constructor Functions
- Displaying Images & Using Multiple Constructor Functions
- Adding Buttons & Using Functions As Values
- Styling Buttons & Working with Padding
- How NOT To Build Interactive Widgets
- Introducing Stateful Widgets
- Generating Random Numbers
- Module Summary

FLUTTER & DART BASICS II - FUNDAMENTAL DEEP DIVE [QUIZ APP]

- Module Introduction
- A Challenge For You!
- Challenge Solution 1/2 - Creating a Widget
- Challenge Solution 2/2 - Working with More Widgets
- Adding Icons to Buttons
- Adding Transparency to Widgets
- Repetition & Exercise: Adding a Stateful Widget
- Rendering Content Conditionally
- Accepting & Passing Functions as Values
- The "initState" Method
- Deep Dive: Flutter's (Stateful) Widget Lifecycle
- Using Ternary Expressions & Comparison Operators
- Understanding "if" Statements
- Using "if" Statements In Lists
- if Statements & Comparison Operators
- Adding a Data Model & Dummy Data
- Configuring a Column
- Creating a Reusable, Custom Styled Button
- Accessing List Elements & Object Properties
- Mapping Lists & Using the Spread Operator
- Alignment, Margin & Padding
- Mutating Values in Memory
- Managing The Questions Index As State
- More on Button Styling
- Using Third-Party Packages & Adding Google Fonts
- Passing Data via Functions Across Widgets
- More Conditions
- Getting Started with the Results Screen
- Passing Data to the Results Screen
- Introducing Maps & "for" Loops
- Using "for" Loops In Lists
- Note: A Typo In The Next Lecture
- Accessing Map Values & Using "Type Casting"
- Combining Columns & Rows
- Expanded To The Rescue!
- Filtering & Analysing Lists
- Making Content Scrollable with SingleChildScrollView
- Beyond the Basics: Optional, Important Dart Features
- Module Summary



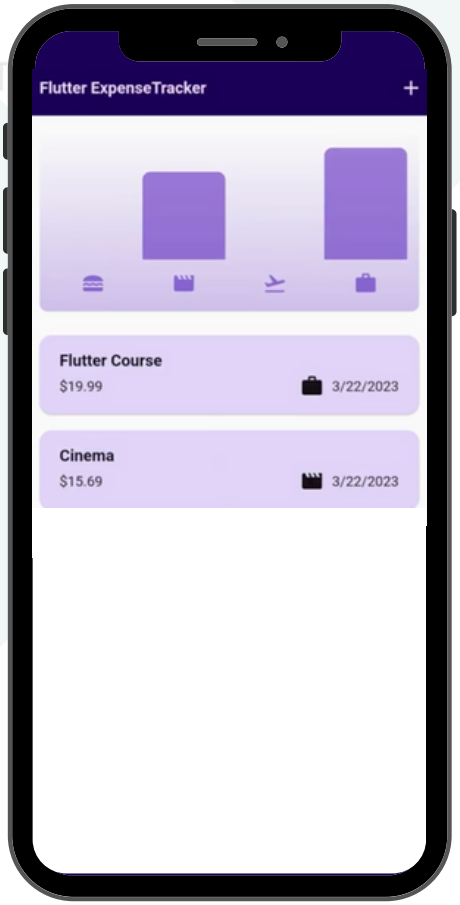
DEBUGGING FLUTTER APPS

- Module Introduction
- The Starting Project & A Problem

- Understanding Error Messages
- Debugging Apps & Using "Debug Mode"
- Working with the Flutter DevTools
- Running the App on Real iOS or Android Devices

ADDING INTERACTIVITY, MORE WIDGETS & THEMING [EXPENSE TRACKER APP]

- Module Introduction
- Starting Setup & Repetition Time!
- Adding an Expense Data Model with a Unique ID & Exploring Initializer Lists
- Introducing Enums
- Creating Dummy Data
- Efficiently Rendering Long Lists with ListView
- Using Lists Inside Of Lists
- Creating a Custom List Item with the Card & Spacer Widgets
- Using Icons & Formatting Dates
- Setting an AppBar with a Title & Actions
- Adding a Modal Sheet & Understanding Context
- Handling User (Text) Input with the TextField Widget
- Getting User Input on Every Keystroke
- Letting Flutter do the Work with TextEditingController
- Time to Practise: Adding a New Input
- Exercise Solution
- Closing The Modal Manually
- Showing a Date Picker
- Working with "Futures" for Handling Data from the Future
- Adding a Dropdown Button
- Combining Conditions with AND and OR Operators
- Validating User Input & Showing an Error Dialog
- Saving New Expenses
- Creating a Fullscreen Modal
- Using the Dismissible Widget for Dismissing List Items
- Showing & Managing "Snackbars"
- Flutter & Material 3
- Getting Started with Theming
- Setting & Using a Colour Scheme
- Setting Text Themes
- Using Theme Data in Widgets
- Important: Adding Dark Mode
- Adding Dark Mode
- Using Another Kind of Loop (for-in)
- Adding Alternative Constructor Functions & Filtering Lists



- Adding Chart Widgets
- Module Summary

BUILDING RESPONSIVE & ADAPTIVE USER INTERFACES [EXPENSE TRACKER APP]

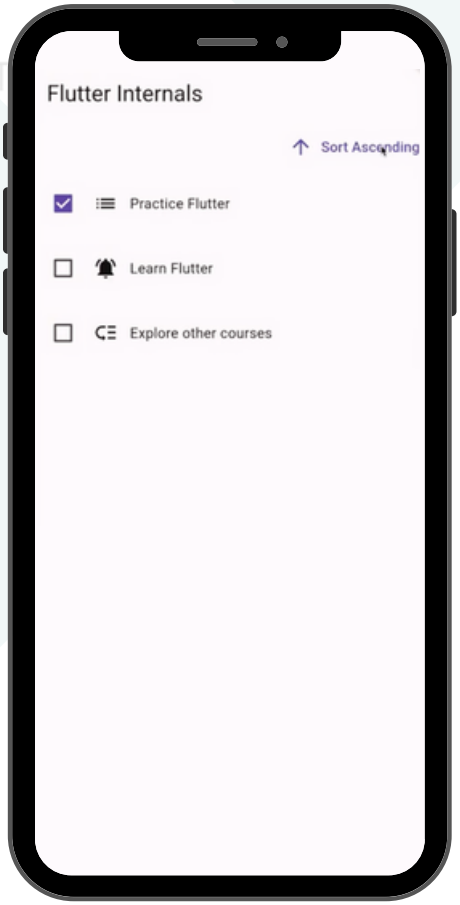
- Module Introduction
- What is "Responsiveness"?
- Locking the Device Orientation
- Updating the UI based on the Available Space
- Understanding Size Constraints
- Handling to Screen Overlays like the Soft Keyboard
- Understanding "Safe Areas"
- Using the LayoutBuilder Widget
- Building Adaptive Widgets
- Module Summary

FLUTTER & DART INTERNALS [TODO APP]

- Module Introduction
- Three Trees: Widget Tree, Element Tree & Render Tree
- How The UI Is Updated
- Refactor & Extract Widgets To Avoid Unnecessary Builds
- Understanding Keys - Setup
- Which Problem Do Keys Solve?
- Understanding & Using Keys
- Mutating Values in Memory & Making Sense of var, final & const
- Module Summary

BUILDING MULTI-SCREEN & NAVIGATING BETWEEN SCREENS [MEALS APP]

- Module Introduction
- Project Setup
- Using a GridView
- Widgets vs Screens
- Displaying Category Items on a Screen
- Making any Widget Tappable with InkWell
- Adding Meals Data
- Loading Meals Data Into a Screen
- Adding Cross-Screen Navigation
- Passing Data to the Target Screen
- Introducing the Stack Widget
- Improving the MealItem Widget
- Adding Navigation to the MealDetails Screen
- Improving the MealDetails Screen
- Adding Tab-based Navigation
- Passing Functions Through Multiple Layers of Widgets (for State Management)



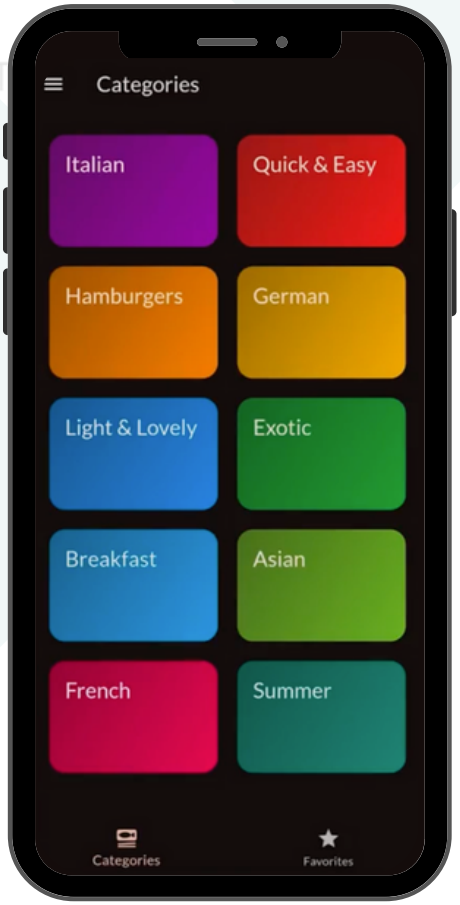
- Adding a Side Drawer
- Closing the Drawer Manually
- Adding a Filter Item
- Replacing Screens (Instead of Pushing)
- Adding More Filter Options
- Replacing WillPopScope with PopScope
- Returning Data When Leaving a Screen
- Reading & Using Returned Data
- Applying Filters
- An Alternative Navigation Pattern: Using Named Routes
- Module Summary

MANAGING APP - WIDE STATE [MEALS APP]

- Module Introduction
- What's The Problem?
- Installing the Solution: Riverpod
- How State Management with Riverpod Works
- Creating a Provider
- Using a Provider
- Creating a More Complex Provider with StateNotifier
- Using the FavoritesProvider
- Triggering a Notifier Method
- Getting Started with Another Provider
- Combining Local & Provider-managed State
- Outsourcing State Into The Provider
- Connecting Multiple Providers With Each Other (Dependent Providers)
- Swapping The "Favorite Button" Based On Provider State
- Module Summary
- "riverpod" vs "provider" - There are many Alternatives!

ADDING ANIMATION - [MEALS APP]

- Module Introduction
- Setup & Understanding Explicit vs Implicit Animations
- Explicit Animations: Adding an Animation Controller
- Explicit Animations: Playing the Animation with AnimatedBuilder
- Fine Tuning Explicit Animations
- Getting Started with Implicit Animations
- Configuring Implicit Animations
- Adding Multi-Screen Transitions
- Module Summary

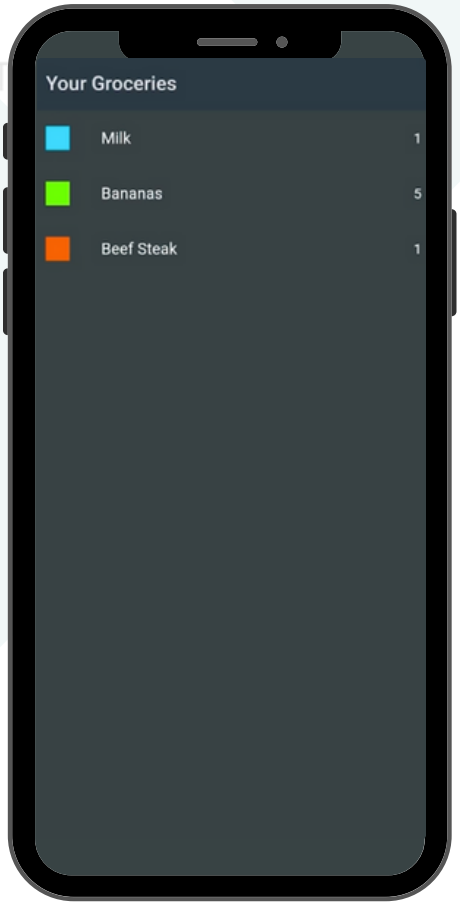


HANDLING USER INPUT & WORKING WITH FORMS - [SHOPPING LIST APP]

- Module Introduction
- Setup & A Challenge For You
- Challenge Solution 1 - Building & Using Models
- Challenge Solution 2 - Building the List UI
- Adding a "New Item" Screen
- The Form & TextFormField Widgets
- A Form-aware Dropdown Button
- Adding Buttons to a Form
- Adding Validation Logic
- Getting Form Access via a Global Key
- Extracting Entered Values
- Passing Data Between Screens
- Final Challenge Solution
- Module Summary

CONNECTING A BACKEND & SENDING HTTP REQUESTS [SHOPPING LIST APP]

- Module Introduction
- What's a Backend? And Why Would You Want One?
- What Is HTTP & How Does It Work?
- Setting Up a Dummy Backend (Supabase)
- Adding the http Package
- Sending a POST Request to the Backend
- Working with the Request & Waiting for the Response
- Fetching & Transforming Data
- Avoiding Unnecessary Requests
- Managing the Loading State
- Error Response Handling
- Sending DELETE Requests
- Handling the "No Data" Case
- Better Error Handling
- Module Summary
- Using the FutureBuilder Widget



USING NATIVE DEVICE FEATURES (E.G. CAMERA) [FAVOURITE PLACES APP]

- Module Introduction
- Setup & A Challenge For You!
- Adding a Place Model (Challenge Solution 1/6)
- Adding a "Places" Screen (Challenge Solution 2/6)
- Adding an "Add Place" Screen (Challenge Solution 3/6)
- Adding "riverpod" & A Provider (Challenge Solution 4/6)

- Adding Places with Provider & Displaying Places (Challenge Solution 5/6)
- Adding a "Place Details" Screen (Challenge Solution 6/6)
- Adding a "Pick an Image" Input
- Installing the "Image Picker" Package
- Using the Device Camera For Taking Pictures
- Adding the Picked Image to the Model & "Add Place" Form
- Previewing the Picked Image
- Important: "location" Package & Android
- Adding the "location" Package & Starting with the "Get Location" Input Widget
- Getting the User's Current Location
- Using the Flutter Maps API - Setup
- Using Google's Geocoding API
- Storing the Location Data in the Model
- Displaying a Location Preview Map Snapshot via Flutter
- Using the Picked Location in the Form
- Outputting the Location Data
- Installing & Configuring the Flutter Maps Package
- Adding a "Map" Screen
- Displaying the Picked Place on a Dynamic Map
- Handling Map Taps for Selecting a Location Manually
- Using the Map Screen in the "Add Place" Form
- Installing Packages for Local (On-Device) Data Storage
- Storing the Picked Image Locally
- Storing Place Data in a (On-Device) SQL Database
- Loading Data from the SQL Database
- Using a FutureBuilder for Loading Data
- Module Summary
- Adding Your Own Native Code

PUSH NOTIFICATION & MORE : BUILDING A [CHAT APP WITH FLUTTER & SUPABASE]

- Module Introduction
- App & Supabase Setup
- Adding an Authentication Screen
- Adding Buttons & Modes to the Authentication Screen
- Validating User Input
- Supabase CLI & SDK Setup 1/2
- FlutterFire Configuration
- Supabase CLI & SDK Setup 2/2

- Signing Users Up
- Logging Users In
- Showing Different Screens Based On The Authentication State
- Adding a Splash Screen (Loading Screen)
- Adding User Logout
- Image Upload: Setup & First Steps
- Adding a User Image Picker Widget
- Using the ImagePicker Package
- Managing The Selected Image In The Authentication Form
- Uploading Images To Supabase
- Showing a Loading Spinner Whilst Uploading
- Adding a Remote Database: Firestore Setup
- Sending Data to Firestore
- Storing a Username
- Adding ChatMessages & Input Widgets
- A Note About Reading Data From Firestore
- Sending & Reading Data To & From Firestore
- Loading & Displaying Chat Messages as a Stream
- Styling Chat Message Bubbles
- Push Notifications - Setup & First Steps
- Requesting Permissions & Getting an Address Token
- Testing Push Notifications
- Working with Notification Topics
- Sending Push Notifications Automatically via Cloud Functions
- Module Summary

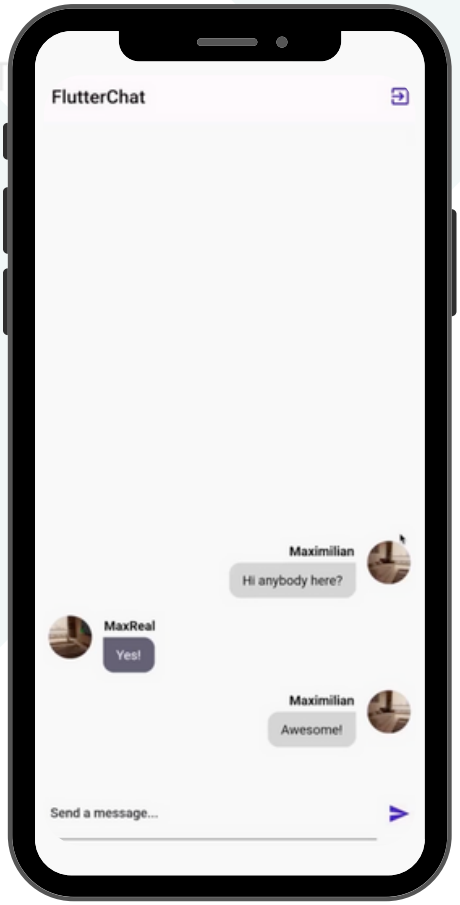
NEXT STEPS & ROUNDUP

- Publishing Android Apps
- Course Roundup

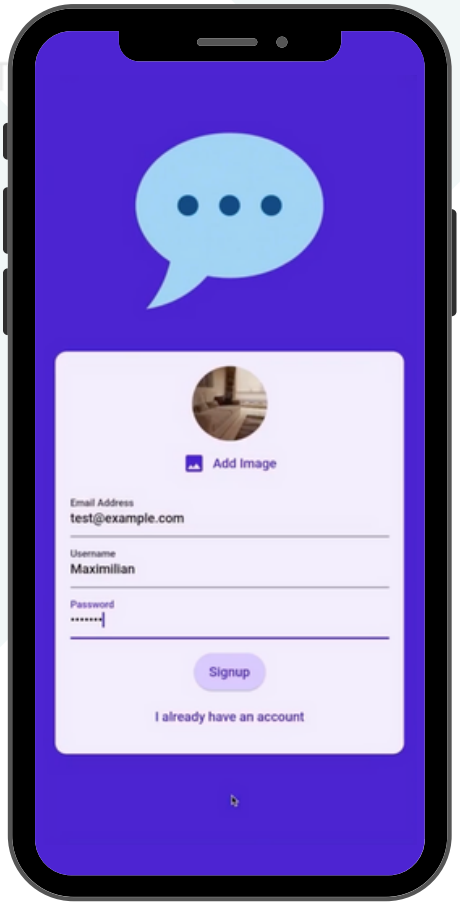
DURATION: 20 WEEKS

OUTCOME:

1. **Understand Flutter Project Structure:** Learn the basics of a Flutter project, from setup to code structure.
2. **Master Dart Fundamentals:** Gain a solid understanding of Dart programming language, including functions, variables, and data types.

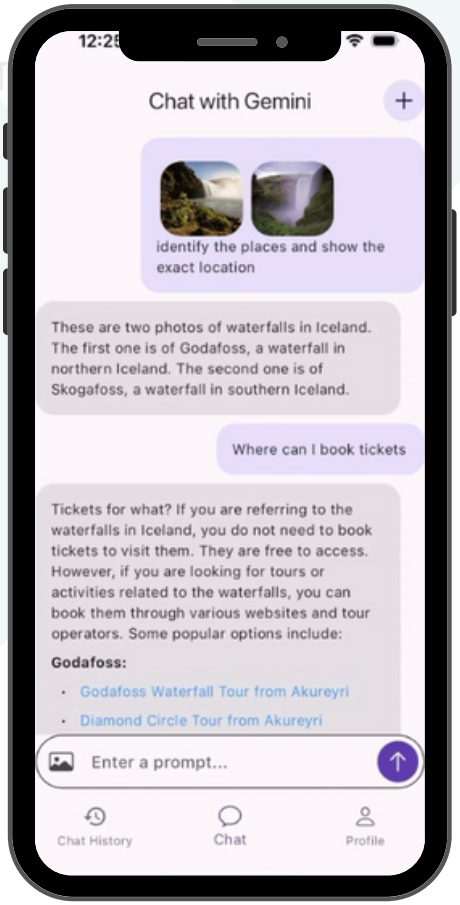


- **Work with Widgets:** Learn about different widgets in Flutter, how to use them, and build complex widget trees.
- **Create Custom Widgets:** Understand the importance of custom widgets, how to build them, and reuse them across your app.
- **State Management:** Get introduced to stateful widgets and learn how to manage state in a Flutter app.
- **Debugging Skills:** Develop skills to debug Flutter apps using error messages, debug mode, and Flutter DevTools.
- **Adding Interactivity:** Learn to add interactive elements like buttons, input fields, and handle user interactions.
- **Theming and Styling:** Understand how to apply themes, styles, and dark mode to your Flutter app.
- **Responsive UI Design:** Build responsive and adaptive user interfaces that work well on different screen sizes and orientations.
- **Multi-Screen Navigation:** Learn to navigate between multiple screens and pass data across them.
- **Advanced State Management with Riverpod:** Implement complex state management using Riverpod.
- **Animations:** Add explicit and implicit animations to enhance the user experience in your Flutter app.
- **Form Handling:** Work with forms, input validation, and manage form data in Flutter.
- **Backend Integration:** Connect your Flutter app to a backend service, send HTTP requests, and handle responses.
- **Native Device Features:** Utilise native device features like the camera and location services in your Flutter app.
- **Push Notifications:** Implement push notifications and work with Supabase for authentication and real-time data.
- **App Deployment:** Learn the process of publishing your Flutter app to the Google Play Store.



FLUTTER USING GEMINI API - [AI CHATBOT APP]

- Introduction to Chatbot Development
- Overview of Gemini API and its capabilities
- Working with Gemini API
- Understanding Gemini API request/response format
- Implementing chat functionality with streaming responses
- Error handling and API key management securely
- Creating the Chat UI
- Design beautiful and modern chat bubbles (user & bot)
- Scrollable message list with animations
- Display typing indicators and message timestamps
- Custom Widgets for Chat Components
- Chat bubble widget
- Input field widget with send button
- Animated loader widget
- State Management using Riverpod
- Manage conversation history
- Loading states, message updates, and UI reactivity
- Adding Interactivity and UX Enhancements
- Typing indicator using animation
- Auto-scroll to latest message
- Copy, delete, or long-press chat options
- Theming and Styling
- Apply light/dark theme toggle
- Use Google Fonts, gradients, and shadows for modern design
- Multi-Screen Navigation
- Splash screen, onboarding, and main chat screen
- Using Local Storage
- Animations for Chat UX
- Responsive design using LayoutBuilder & MediaQuery
- Load history on app start
- Page transitions and button effects using AnimatedContainer, AnimatedSwitcher
- Supabase Auth (Google/Email login)
- Notify users of new bot responses or scheduled messages
- Write widget tests for chat components
- Use Flutter DevTools and error logs for debugging
- Create release build, sign APK, and test on real device



FLUTTER APPRENTICESHIP TRAINING
SYLLABUS PROGRESS TRACKER

- ◯ **BASIC SESSION**

PROJECTS

- ◯ ROLL DICE APP
- ◯ QUIZ APP
- ◯ EXPENSE TRACKER APP
- ◯ TODO APP
- ◯ MEALS APP
- ◯ SHOPPING LIST APP
- ◯ FAVOURITE PLACES APP
- ◯ CHAT APP
- ◯ AI Chatbot App

NOTES

The Premier Institute For Summer, Apprenticeship Training

Hindtech Learning Point

A Trusted IT Institute Brand In Lucknow

Admission Open For

Summer Training
CSE/IT
Diploma, B.Tech/B.E, BCA, MCA
45 - 50 Days Program

Apprenticeship Training
CSE/IT
B.Tech, BCA, MCA, PGDCA, Dip.
6 to 8 Months Program

Industrial Training
CSE/IT
B.Tech, BCA, MCA, PGDCA, Dip.
3 to 5 Months Program

UI/UX Designing
CSE/IT or interested student
B.Tech, BCA, MCA, PGDCA, Dip.
45 - 50 Days Program

Number ONE

Excellent Performance by Hindtech Learning Point in IT Training

Skill India
कौशल भारत-कुशल भारत

N·S·D·C National Skill Development Corporation
Transforming the skill landscape

AI ML

Flutter MERN Java .NET php AI Machine Learning

Since 2020, 1000+ students placed in top IT firms | Parmar Plaza, BKT, Bargadi, UP 226201 | Call: 6307738600 / 7905320279 | www.hindtechlearningpoint.com

HINDTECH
IT SOLUTIONS

Training	Duration	Amount
Flutter Apprenticeship Training Program	6 to 8 Months	₹21,000

www.hindtechitsolutions.com
hr@hindtechitsolutions.com
+91 7905320279

www.hindtechlearningpoint.com
hindtechlearningpointlko@gmail.com
+91 6307738600

Building No 10/703, Ground Floor, near Arvindo Park Road, Sector 10, Indira Nagar, Lucknow, Uttar Pradesh 226016

Branch 1

PARMAR PLAZA, 1st Floor, Bakshi Ka Talab, Bargadi Magath,
Uttar Pradesh 226201

Branch 2

Scan Me:

Scan this QR code for details

HINDTECH
LEARNING POINT

Skill India
कौशल भारत-कुशल भारत

N·S·D·C National Skill Development Corporation
Transforming the skill landscape

Hindtech Learning Point Brochure Update: June 2024-25. Details are applicable from this date.
We offer placement assistance to all eligible trainees.