

Mastering flutter application development

Day 1: Dart Programming

Objective: Build a strong foundation in Dart programming language.

1. Introduction to Dart

- Overview of Dart: Why Dart for Flutter?
- Setting up the Dart environment.
- Running a Dart program.

2. Dart Basics

- Variables, Data types, and Operators.
- Control flow statements (if-else, loops, etc.)
- Dart Lists, Sets and Map
- Functions and Lambdas.

3. Object-Oriented Programming in Dart

- Classes and Objects.
- Inheritance, and Encapsulation.
- Abstract classes

4. Advance topics

- Dart Packages
- Dart Libraries
- Dart Generators
- Dart Async

5. Lab

- Write and debug basic Dart programs.
- Create a mini-project: A simple "To-Do List" CLI application.

Day 2: Flutter Fundamentals

Objective: Introduce the fundamentals of Flutter, focusing on building a strong foundation for creating mobile apps.

1. Introduction to Flutter

- Overview of Flutter framework
- Understanding Flutter architecture (Widgets, Dart)
- Installation and setup (Flutter SDK, Android Studio/VS Code)

2. Widgets Basics

- StatelessWidget and StatefulWidget
- Building UI using basic widgets (Text, Image, Container, Row, Column, Scaffold)
- Introduction to Material Design and Cupertino Widgets
- More about widgets
 1. TextField
 2. Buttons
 3. Stack
 4. Checkbox
 5. Radio Button

3. Dart Basics

- Syntax, variables, and data types
- Functions and control flow
- Classes and object-oriented programming in Dart

4. Lab:

Objective: Create a simple "Hello World" app and customize it with basic widgets and Material Design components.

Day 3: Intermediate UI, form and Navigation

Objective: Develop skills to create more complex UIs and implement navigation.

1. Advanced Widgets

- ListView, GridView, and dynamic lists
- Alert Dialogs
- Card, and Stack
- Tabbar
- Snackbar
- Slider
- Bottom Navigation Bar
- Slivers

2. Flutter Forms

3. Navigation and Routing

- Navigation basics (Navigator.push and Navigator.pop)
- Named routes and route management

4. Lab:

Objective: Build an app that displays a list of items with navigation to a detail screen for each item using named routes.

Day 4: Networking and APIs

Objective: Explore handling external data sources.

1. Consuming RESTful APIs

- Basics of HTTP requests (GET, POST, PUT, DELETE)
- Consuming RESTful APIs using http package
- Parsing JSON in Flutter

2. Database Integration

- SQLite
- Saving and retrieving data locally

3. Lab:

Objective: Create an app that fetches data from an API, displays it in a list, and allows saving items locally using SQLite.

Day 5: Animations, Error Handling, and Security

Objective: Learn and apply best practices for writing secure and error-free Flutter apps.

1. Animations

- Implicit animations (AnimatedContainer, AnimatedOpacity)
- Explicit animations using AnimationController

2. Error Handling

- Using try-catch and error widgets
- Logging errors and monitoring

3. Security

- Secure storage of sensitive data
- Input validation
- API security and token management (how to consume token-based APIs)

4. Lab:

Objective: Enhance the app from Day 4 to include proper error handling for network calls

Day 6: Advanced Flutter - Optimization, Performance and Best Practices

Objective: Focus on improving app performance and integrating location-based services.

1. Performance Optimization

- Analyzing performance using Flutter DevTools
- Lazy loading lists with ListView.builder
- Reducing widget rebuilds

2. Best Practices in Flutter Development

- Code organization and modularization
- Reusable widgets and class components
- Theming and styling for consistency

Day 7: Advanced Flutter - Google Maps SDK

1. Google Maps SDK

- Setting up and integrating Google Maps SDK in Flutter
- Adding markers and customizing map styles

2. Working with Geolocation

- Accessing user location.
- Drawing paths and routes on maps.

3. Lab:

Objective: Build an app that displays a Google Map, shows the user's current location, and allows adding custom markers.

Day 8: Advanced Topics and Project Development

Objective: Work on a complete project incorporating all concepts, best practices, and advanced topics.

1. Debugging and Testing

- Debugging techniques
- Writing unit and widget tests

2. Final Project Development

- Design and build a complete app with the following features:
 - Authentication using secure token storage
 - API integration for data fetching and updating
 - Local data storage using SQLite

3. App deployment

- Preparing for deployment (signing APKs, creating bundles).

4. Lab:

Objective: Complete a capstone project involving user authentication, API consumption, local data storage.