

FLUTTER APP DEVELOPMENT

FUNDAMENTALS | UI DESIGN & LAYOUTS | STATE MANAGEMENT
API INTEGRATION | APP DEPLOYMENT

REAL SKILLS
FOR MODERN
APP DEVELOPMENT
CAREERS

EARN
&
LEARN



DURATION

Skill Diploma - 4 months

Internship - 2 month

1 year plan is also available, along with a professional company experience certificate.



PRACTICAL

ORIENTED
CURRICULUM



ABOUT THE COURSE

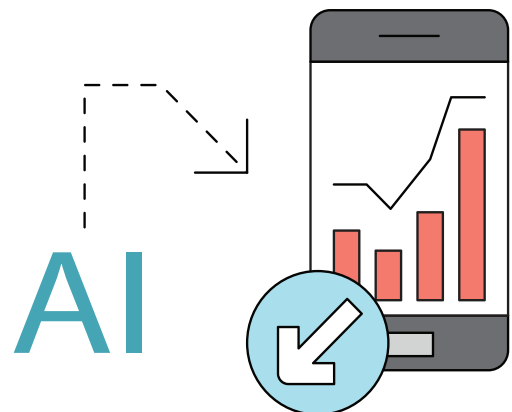
Our Flutter App Development course is an expert-designed program that covers fundamental to advanced concepts of application development using the Flutter framework, an open-source UI software development kit made by Google. This course offers you the skills and knowledge related to Flutter needed to build high-quality, visually engaging and cross-platform mobile applications.

Throughout the course, we provide an interactive learning experience with an opportunity to work on live projects guided by professionals in the industry. By the end of this course, students will be fully equipped to design, develop, test and deploy mobile applications for both Android and iOS platforms using a single codebase, saving time and effort while delivering professional-grade results.



AI TOOLS INTEGRATED

This course is updated with the latest AI tools used by professionals in the industry. Students will learn to use these tools throughout the course to speed up development, improve code quality and bring creative ideas to life faster.



AUTOCOMPLETE: GITHUB COPILOT

GitHub Copilot is an AI-powered coding assistant developed by GitHub and OpenAI. It helps developers write Dart and Flutter code faster by suggesting entire lines or blocks of code in real time. Students will learn to use Copilot inside VS Code to reduce repetitive typing, catch errors early and explore smarter ways to write clean code

UI GENERATION: FLUTTERFLOW

FlutterFlow is a visual development platform that allows developers to build Flutter applications using a drag-and-drop interface powered by AI. Students will use FlutterFlow to quickly prototype app screens, generate production-ready Flutter code and speed up the UI building process without writing everything from scratch.

DESIGN IDEAS: GOOGLE GEMINI

Google Gemini is Google's advanced AI model that helps developers and designers brainstorm ideas, generate content, explain concepts and suggest design approaches. Students will use Gemini to get instant design inspiration, understand complex Flutter concepts in simple language and explore creative UI/UX ideas for their projects.

DESIGN IMPORT: FIGMA

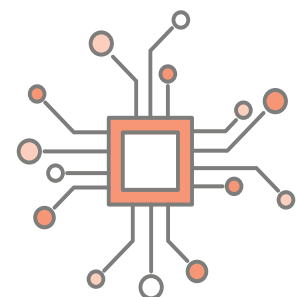
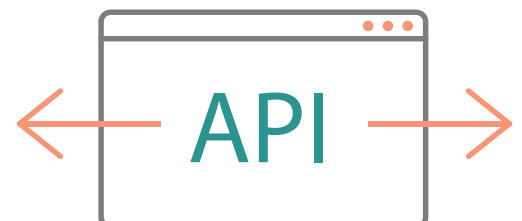
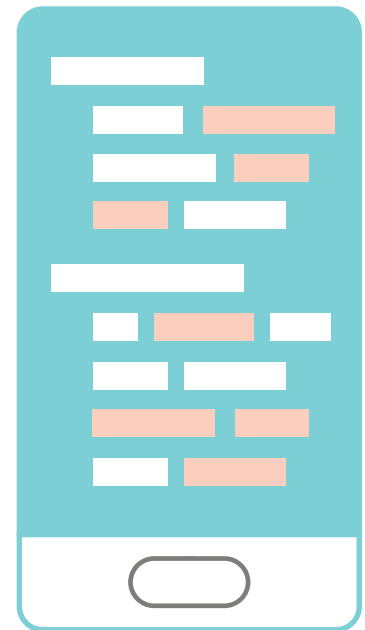
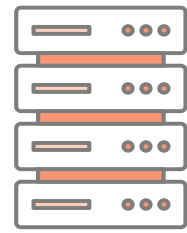
Figma is the industry-standard design tool used by UI/UX designers worldwide. Students will learn how to create or import app designs from Figma and translate them accurately into Flutter widgets and layouts.

CODING: ANTIGRAVITY

Antigravity is an AI coding assistant that helps developers build features rapidly by generating code snippets, solving bugs and suggesting optimizations. Students will use Antigravity to accelerate their development workflow and learn best practices while building their Flutter projects.

WHAT YOU WILL LEARN :

- Introduction to Flutter
- What is Flutter
- Why Flutter
- Building Blocks of a Flutter Application
- Flutter vs Other Frameworks
- Flutter CLI
- Flutter Widgets and UI Design
- Introduction to Widgets
- Stateless vs Stateful Widgets
- Layouts (Row, Column, Stack, Container, Expanded, Flexible, Card)
- Styling and Theming
- Material Design Widgets
- Cupertino Widgets
- State Management in Flutter
- Introduction to State Management
- setState() and Stateful Widgets
- Riverpod for State Management
- Managing Global State
- Flutter Navigation and Routing
- Introduction to Navigation
- Named Routes
- Navigation with Arguments
- Nested Navigation
- Navigation with Tabs
- Forms and Validation in Flutter
- Introduction to Forms
- Creating Forms
- Form Validation
- Custom Validators
- Handling Form Submission
- API Integration



AUTHENTICATION AND AUTHORIZATION

Students will learn how to implement secure authentication in Flutter apps. This includes token-based authentication using JWT, storing tokens securely on the device, sending authenticated requests and handling session expiry and logout.

CONNECTING FLUTTER WITH APIS

Students will learn how to make HTTP requests from Flutter using packages like http and Dio. They will handle GET, POST, PUT and DELETE requests, parse JSON responses into Dart models and display real data inside their Flutter app screens.

- Testing in Flutter
- Unit Testing
- Testing functions and logic
- Widget Testing
- Testing UI behavior
- Integration Testing
- Testing complete app flows

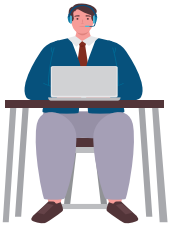
INTERNSHIP – 2 MONTHS

The internship is a 2-month full-time industry experience that gives students the opportunity to apply everything they have learned in a real professional environment.

- Real-world Flutter project development under senior developer guidance
- Contributing to production-ready apps with real users
- Agile workflows (sprints, standups, collaboration)
- Code reviews and GitHub workflow
- AI tools in real development pipelines
- Internship Certificate + Letter of Recommendation



EXIT PROFILES



FLUTTER DEVELOPER

Build cross-platform apps using Flutter and Dart



MOBILE APP DEVELOPER

Develop apps for Android and iOS using a single codebase



FRONT-END DEVELOPER

Focus on UI, UX and responsive design



SOFTWARE DEVELOPER

Work on application logic, features and system design



WHY TECHNO DOT ACADEMY ?

Techno Dot Academy is a career-focused IT training institute dedicated to building job-ready professionals through practical, industry-oriented learning. With expert mentors, real-time project exposure, and a curriculum aligned to current market demands, the academy helps students develop strong technical and problem-solving skills. Techno Dot Academy also emphasizes career growth by providing guidance, support, and placement assistance, making it a trusted choice for those looking to start or advance their journey in the IT field.



**Guaranteed
Placement Assistance
& Internship**



**Strongly Endorsed
by a Reputed
IT Firm**



**Scholarships/
Earn While Learn
Option**



INDIA

TECHNO DOT ACADEMY

Kinfra Techno Industrial Park, Kakkanchery, Calicut

University PO, Kerala 673635

+91 807 844 0115

info@techno.academy

UAE

TECHNO DOT ACADEMY

Sharjah Research Technology And

Innovation Park, UAE

+971 525646337

info@techno.academy

www.techno.academy

PROGRAMMING FOUNDATIONS

- What is Programming
- Programming Languages Overview
- Why Python
- How Computers Execute Code
- Compiler vs Interpreter
- Algorithms
- Python Installation & Setup
- IDEs & Editors
- Errors & Debugging
- Syntax Basics

CONTROL FLOW

- Conditional Statements (if, elif, else)
- Nested Conditions
- Logical Decision Making
- Loops (for, while)
- Loop Control (break, continue, pass)
- Nested Loops

DATA STRUCTURES

- Strings
- Lists
- Tuples
- Sets
- Dictionaries

PYTHON BASICS (CORE SYNTAX)

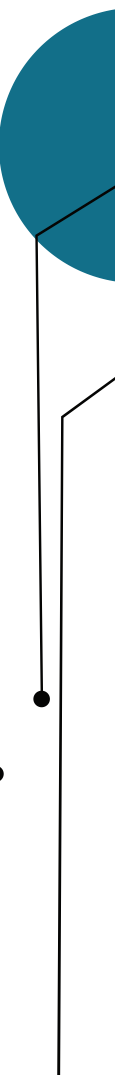
- Variables
- Data Types
- Type Casting
- Operators (Arithmetic, Assignment)
- Comparison Operators
- Logical Operators
- Input & Output
- String Formatting
- Comments

FUNCTIONS

- Function Definition & Calling
- Parameters & Arguments
- Return Values
- Default & Keyword Arguments
- *args and **kwargs
- Scope (Local & Global)
- Lambda Functions
- Recursion
- Nested Functions

OBJECT-ORIENTED PROGRAMMING (OOP)

- Classes & Objects
- Attributes & Methods
- Constructor (`__init__`)
- Encapsulation
- Abstraction
- Inheritance
- Polymorphism



ADVANCED OOP & ERROR HANDLING

- Class vs Instance Variables
- Static & Class Methods
- Magic Methods (`__str__`, `__repr__`)
- Composition vs Inheritance
- Exception Handling (`try`, `except`, `finally`)
- Custom Exceptions
- Raising Errors

MODULES & PYTHONIC CODING

- Modules & Imports
- Packages
- Virtual Environments
- pip & Dependency Management
- Iterators & Generators
- List/Dict/Set Comprehensions
- `map()`, `filter()`, `reduce()`
- Clean Code & PEP 8

PROJECTS & APPLICATIONS

- Calculator Application
- Number Guessing Game
- Student Management System
- To-Do List Application
- Bank Account System
- Library Management System
- Inventory Management
- Expense Tracker
- Portfolio & GitHub

FILE HANDLING & OS INTERACTION

- File Operations (`read`, `write`, `append`)
- File Modes
- CSV Handling
- JSON Handling
- OS Module
- File Paths & System Operations

PROMPT ENGINEERING BASICS

- How to ask AI correctly:
 - Clear instructions
 - Provide context
 - Ask step-by-step solutions

DJANGO CURRICULUM

WEB & DJANGO FOUNDATIONS

- What is Django
- Web Applications
- Client-Server Architecture
- HTTP Basics
- MVT Architecture
- Request-Response Lifecycle
- Virtual Environments
- Django Installation
- Project & App Structure

PROJECT SETUP & CONFIGURATION

- settings.py
- INSTALLED_APPS
- MIDDLEWARE
- URLs & Routing
- Views (FBV)
- HTTP Methods (GET, POST)
- Dynamic URLs
- Environment Settings

TEMPLATES & STATIC FILES

- Django Templates
- Template Rendering
- Template Tags & Filters
- Template Inheritance
- Includes
- Static Files (CSS, JS, Images)

MODELS & ORM

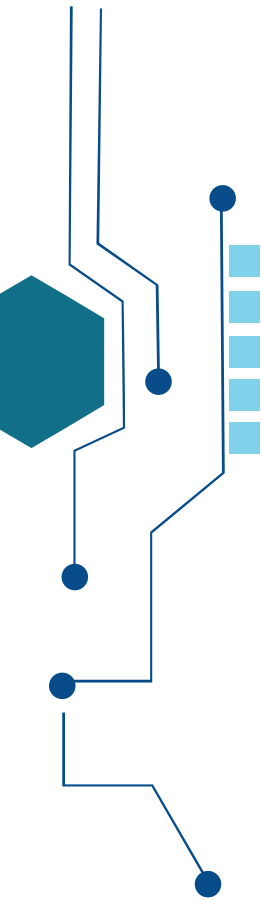
- Models
- ORM
- Migrations
- Django Shell
- CRUD Operations
- Model Relationships

DJANGO ADMIN PANEL

- Admin Setup
- Model Registration
- Admin Customization
- Admin Forms
- Inline Models
- Admin Permissions

FORMS & USER INPUT

- Django Forms
- Form Validation
- ModelForm
- CRUD with Forms
- File Uploads
- Media Configuration



AUTHENTICATION & AUTHORIZATION

- User Model
- Authentication
- Login & Logout
- Signup
- Permissions
- Groups & Roles

MIDDLEWARE & ADVANCED ORM

- Middleware
- Custom Middleware
- Query Optimization
- select_related
- prefetch_related
- Aggregation
- Annotation
- F & Q Objects

TESTING & DEBUGGING

- Django Testing
- TestCase
- View Testing
- Form Testing
- Debugging Tools
- Logging
- Error Handling

CLASS-BASED VIEWS (CBV)

- CBV Basics
- TemplateView
- ListView
- DetailView
- CreateView
- UpdateView
- DeleteView
- Mixins

FILE HANDLING & SECURITY

- Media Files
- File Uploads
- Validation
- Security (CSRF, XSS, SQL Injection)
- Caching
- Performance Optimization

AI TOOLS FOR DJANGO DEVELOPERS

- GitHub Copilot
 - Auto-generate Django models, views, serializers
 - Speed up CRUD development
- ChatGPT
 - Debug Django errors
 - Explain ORM queries
- Google Gemini
 - Alternative for logic explanation & documentation
- DeepSeek
 - Strong for backend logic & optimization



DEPLOYMENT & PRODUCTION

- API Design & Architecture
- CRUD APIs
- Authentication Integration
- Filtering & Pagination
- Testing & Documentation
- Performance Optimization
- Deployment Preparation

PROJECT ARCHITECTURE & CAPSTONE

- App Modularization
- Django Signals
- Service Layer
- Enterprise Structure
- Capstone Project
- Testing & Optimization

