

# SKILL DIPLOMA IN **PYTHON DJANGO** **ANGULAR FULLSTACK**

LEARN FULL STACK,  
LAND YOUR DREAM  
JOB

EARN  
&  
LEARN



**django**

python

  
Angular JS

## DURATION

Skill Diploma - 5 months

Internship - 1 month

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

ADMISSION

OPEN

 **technodot**  
ACADEMY OF R&D&I



1800 1236212 | 7558 837333

[www.techno.academy](http://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)

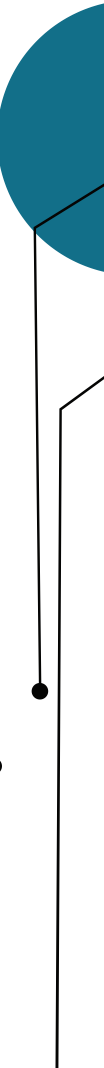
- Variables
- Data Types
- Type Casting
- Operators (Arithmetic, Assignment)
- Comparison Operators
- Logical Operators
- Input & Output
- String Formatting
- Comments

# 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

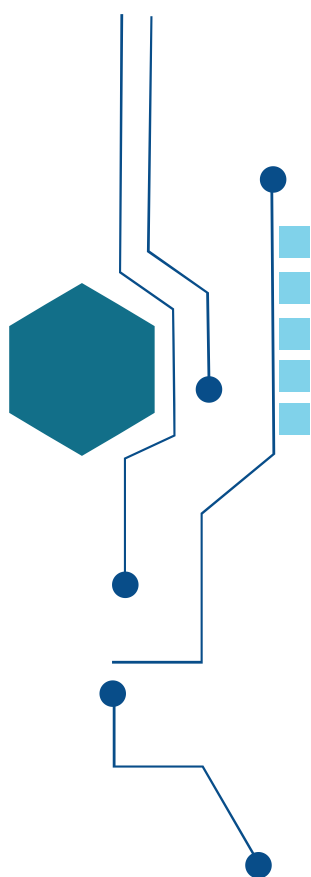
- Models
- ORM
- Migrations
- Django Shell
- CRUD Operations
- Model Relationships

## 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

- Production Setup
- Gunicorn
- Nginx
- Static & Media Deployment
- Docker
- CI/CD
- Monitoring & Scaling

## PROJECT ARCHITECTURE & CAPSTONE

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

# DRF / API CURRICULUM

## API & REST FUNDAMENTALS

- What is API
- API Lifecycle
- REST Architecture
- HTTP Methods
- HTTP Status Codes
- JSON Format
- API Testing (Postman)
- Introduction to DRF

## SERIALIZERS

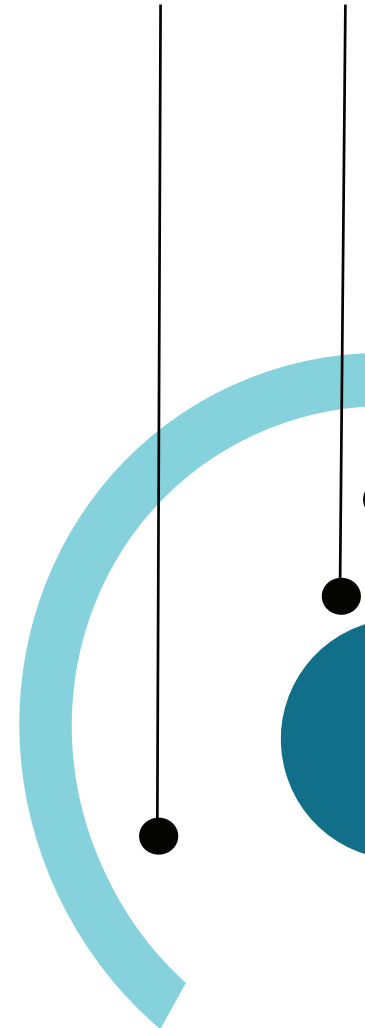
- Serializer Basics
- ModelSerializer
- Serialization & Deserialization
- Validation
- create() & update()
- Serializer Context
- Nested Serializers
- SerializerMethodField

## API VIEWS

- APIView
- GET, POST, PUT, PATCH, DELETE
- Function-Based API Views (@api\_view)
- Request & Response Objects
- Error Handling
- API Response Structure

## GENERIC VIEWS & VIEWSETS

- GenericAPIView
- Mixins
- ListCreateAPIView
- RetrieveUpdateDestroyAPIView
- ViewSets
- ModelViewSet
- Routers
- Custom Actions



## AUTHENTICATION

- Authentication vs Authorization
- Session Authentication
- Token Authentication
- JWT Authentication
- Login & Logout
- User Registration
- Auth Best Practices

## ADVANCED SERIALIZATION & FILE APIS

- Nested Create & Update
- Dynamic Fields
- Custom Representation
- File Upload APIs
- Image Upload
- Advanced Validation
- Global Exception Handling

## API DOCUMENTATION & TESTING

- Swagger / OpenAPI
- API Documentation
- API Testing
- TestCase & APIClient
- CORS
- API Security

## API PERFORMANCE & OPTIMIZATION

- Query Optimization
- select\_related
- prefetch\_related
- Throttling
- Rate Limiting
- Caching
- API Versioning

## PERMISSIONS & AUTHORIZATION

- Built-in Permissions
- Custom Permissions
- Object-Level Permissions
- Filtering
- Searching
- Ordering
- Pagination

## AI TOOLS FOR API DEVELOPERS

- GitHub Copilot
  - Auto-generate serializers, views, ViewSets
  - Speed up CRUD API development
- ChatGPT
  - Generate DRF APIs
  - Debug errors (400, 401, 403, 500)
  - Explain serializers & ORM
- Google Gemini
  - Alternative for API design & logic explanation
- DeepSeek
  - Strong for backend-heavy logic & optimization
- Claude

Best for:

- Large codebase understanding
- Writing clean APIs
- Documentation & architecture
- Antigravity IDE
  - AI-integrated coding environment
  - Faster development workflow

# PRACTICAL AI USAGE IN DRF

## API GENERATION

- Generate:
- Serializers
- ViewSets
- Routers
- Pagination & Filters

## AI FOR AUTHENTICATION SYSTEMS

- Generate:
  - JWT login/register APIs
  - Token refresh logic
  - Role-based access

## API OPTIMIZATION USING AI

- Improve:
  - Query performance
  - Serializer efficiency
  - Reduce DB hits

## API OPTIMIZATION USING AI

- Suggest:
  - `select_related` / `prefetch_related`
  - Caching strategies

## AI FOR TESTING

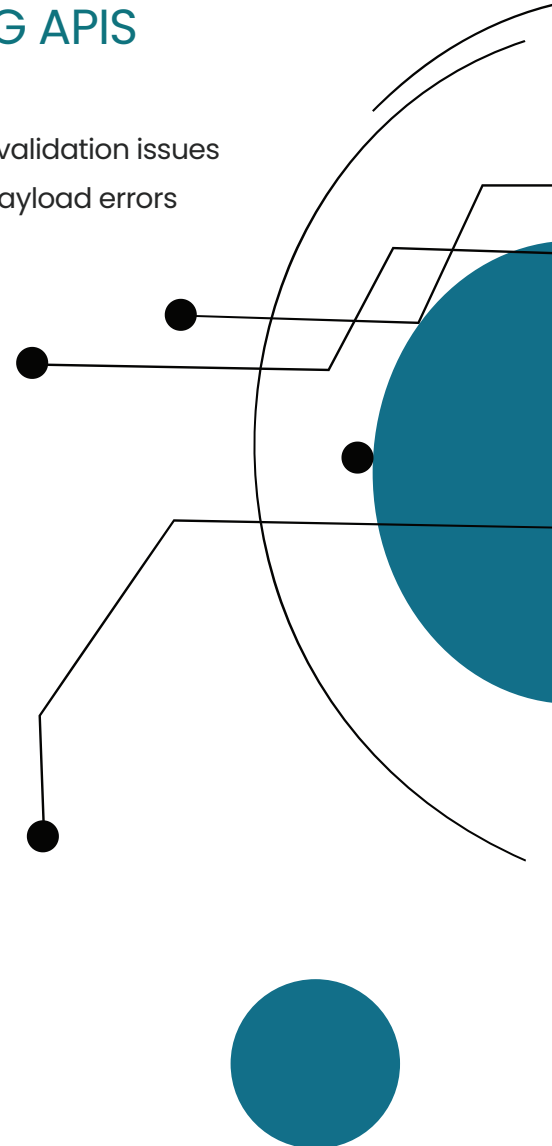
- Generate
  - Test cases
  - APIClient tests
  - Edge case scenarios

## DEBUGGING APIS

- Fix:
  - 400 Bad Request
  - 401 Unauthorized
  - 403 Forbidden
  - 500 Server Errors

## DEBUGGING APIS

- Analyze:
  - Serializer validation issues
  - Request payload errors



# CAPSTONE PROJECT & PRODUCTION

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

## POSTGRESQL / SQL CURRICULUM

### DATABASE FUNDAMENTALS

- Data & Databases
- Structured vs Unstructured Data
- Types of Databases (SQL vs NoSQL)
- What is SQL
- PostgreSQL Introduction
- Database Tools (psql, pgAdmin)
- Database & Schema Concepts

### SQL CORE (CRUD)

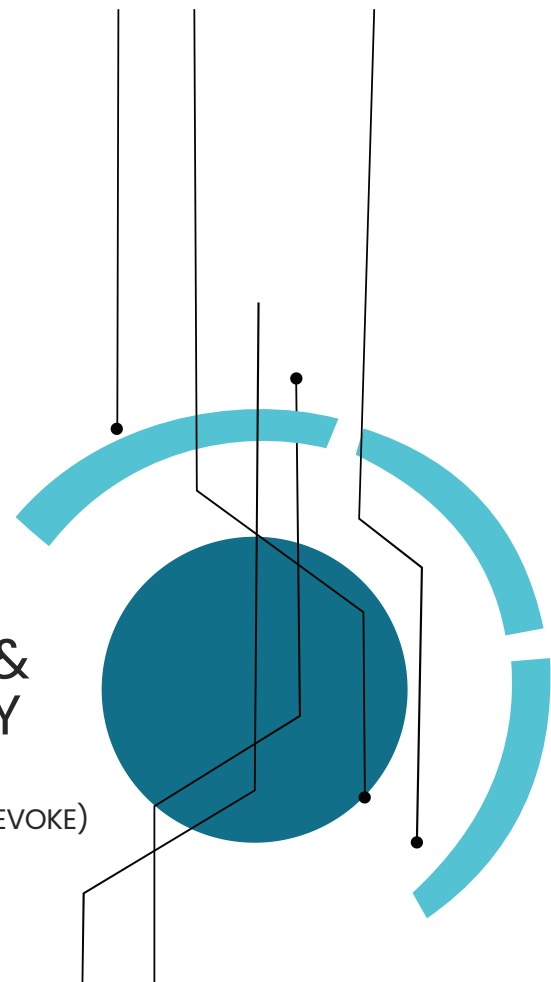
- Data Types
- CREATE TABLE
- INSERT
- SELECT
- UPDATE
- DELETE
- NULL Handling
- ALTER TABLE

### FILTERING & CONDITIONS

- WHERE Clause
- Comparison Operators
- Logical Operators
- LIKE & ILIKE
- IN & BETWEEN
- ORDER BY
- LIMIT & OFFSET
- DISTINCT

### CONSTRAINTS & DATA INTEGRITY

- Roles & Users
- Permissions (GRANT, REVOKE)
- Schema Security
- Row-Level Security
- Backup & Restore
- Production Setup
- Monitoring



## RELATIONSHIPS & JOINS

- One-to-One
- One-to-Many
- Many-to-Many
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN
- Self JOIN
- Subqueries

## AGGREGATIONS & GROUPING

- COUNT, SUM, AVG, MIN, MAX
- GROUP BY
- HAVING
- Aggregation with JOINS
- CASE WHEN
- Window Functions
- ROW\_NUMBER, RANK

## PERFORMANCE & OPTIMIZATION

- Indexes
- Query Optimization
- EXPLAIN & ANALYZE
- Views
- Materialized Views
- Sequences & SERIAL
- UUID
- JSONB & Arrays

## TRANSACTIONS & ADVANCED FEATURES

- Transactions (BEGIN, COMMIT, ROLL-BACK)
- ACID Properties
- Isolation Levels
- Locking
- Functions
- Stored Procedures
- Triggers
- Audit Logging

## SECURITY & BACKUP

- Roles & Users
- Permissions (GRANT, REVOKE)
- Schema Security
- Row-Level Security
- Backup & Restore
- Production Setup
- Monitoring



# ADVANCED POSTGRESQL & APPLICATIONS

- PostgreSQL with Django
- JSON & JSONB
- Full-Text Search
- Partitioning
- Foreign Data Wrappers
- Advanced Query Patterns

# ANGULAR

## (HTML + CSS)

### 1: HTML BASICS

- Web fundamentals, HTML structure
- Tags, elements, attributes
- Text formatting, lists
- Links, images

### 3: CSS BASICS

CSS syntax, selectors

Colors, units, typography

Box model

## (JAVASCRIPT CORE)

### 5: JS BASICS

V- ariables, data types

- Operators, conditionals

### 7: JS ADVANCED

- Objects, destructuring
- Array methods (map, filter, reduce)
- Strings, template literals

# CAPSTONE & PRODUCTION

- Database Design
- Normalization
- ER Diagrams
- Complex Queries
- Reporting
- Optimization
- Security
- Docker & Deployment

## 2: HTML ADVANCED

- Tables, forms, inputs
- Media (audio/video/iframe)
- Semantic HTML
- Accessibility basics

## 4: CSS LAYOUT

- Flexbox
- Grid
- Positioning
- Responsive design (media queries)

## 6: JS FUNCTIONS & LOOPS

- Functions, scope
- Loops (for, while)
- Arrays basics

## 8: DOM & ASYNC JS

1. DOM manipulation
2. Events
3. setTimeout / setInterval
4. Promises, fetch, async/await

# ANGULAR

## 9: FRONTEND FRAMEWORK INTRODUCTION

- 1. SPA concept
- Why frameworks?
- Angular overview
- Project setup:
- Angular (CLI)

## 11: DATA BINDING & DIRECTIVES

- Interpolation
- Property & event binding
- Structural directives (\*ngIf, \*ngFor)

## 13: ROUTING & API

- Angular routing
- HTTPClient
- API integration

## AI USAGE :

1. Debugging help
2. Concept explanation

## 10: ANGULAR BASICS

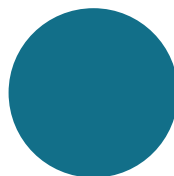
- Angular architecture
- TypeScript basics
- Components, templates

## 12: FORMS & SERVICES

- Template-driven forms
- Reactive forms
- Services & dependency injection

## 14: ADVANCED ANGULAR

- RxJS basics
- State management intro
- Guards (auth)
- Mini project



# 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](http://www.techno.academy)