**50-Day Python & Django Learning Plan**

**Week 1-2: Python Fundamentals**

**Goal**: Learn Python basics and practice coding.

| **Day** | **Topics to Study** | **Projects/Tasks** |
| --- | --- | --- |
| 1 | Python Basics: Variables, Data Types, Input/Output | Write a program to take user input and display it. |
| 2 | Python Basics: Strings, String Methods | Build a program to manipulate strings (e.g., reverse a string). |
| 3 | Python Basics: Lists, Tuples, Dictionaries | Create a program to store and retrieve data using lists/dictionaries. |
| 4 | Python Basics: Loops (for, while) | Write a program to print numbers 1-100 using loops. |
| 5 | Python Basics: Conditionals (if, elif, else) | Build a program to check if a number is odd or even. |
| 6 | Python Basics: Functions | Write reusable functions for basic math operations. |
| 7 | Python Basics: Error Handling (try, except) | Create a program to handle invalid user input. |

**Week 3-4: Intermediate Python**

**Goal**: Learn OOP, file handling, and work on small projects.

| **Day** | **Topics to Study** | **Projects/Tasks** |
| --- | --- | --- |
| 8 | Object-Oriented Programming (OOP): Classes and Objects | Create a class for a "Car" with attributes like color and speed. |
| 9 | OOP: Inheritance, Polymorphism | Extend the "Car" class to create a "Truck" class. |
| 10 | File Handling: Reading/Writing Files | Write a program to read and write data to a text file. |
| 11 | Modules and Libraries (e.g., math, random) | Build a program to generate random numbers. |
| 12 | Working with APIs (e.g., requests library) | Fetch data from a public API (e.g., weather data). |
| 13 | Mini Project: To-Do List App | Build a CLI-based to-do list app. |
| 14 | Mini Project: Weather App | Create a program to fetch and display weather data using an API. |

**Week 5-6: Django Fundamentals**

**Goal**: Learn Django basics and build a simple web application.

| **Day** | **Topics to Study** | **Projects/Tasks** |
| --- | --- | --- |
| 15 | Django Setup: Installation, Project Structure | Set up a Django project and run the development server. |
| 16 | Django Models: Creating Database Models | Create a model for a "Blog" app with fields like title and content. |
| 17 | Django Views: Function-Based Views | Write views to display blog posts. |
| 18 | Django Templates: HTML Templates, Template Tags | Create templates to render blog posts. |
| 19 | Django URLs: URL Routing | Set up URLs for the blog app. |
| 20 | Django Admin Panel: Managing Data | Register the "Blog" model in the admin panel. |
| 21 | Mini Project: Simple Blog Website | Build a blog where users can view posts. |

**Week 7-8: Intermediate Django**

**Goal**: Learn advanced Django concepts and build a portfolio website.

| **Day** | **Topics to Study** | **Projects/Tasks** |
| --- | --- | --- |
| 22 | Django Forms: Creating and Handling Forms | Add a form to create blog posts. |
| 23 | Django Static Files: CSS, JavaScript, Images | Style the blog website using CSS. |
| 24 | Django User Authentication: Login, Logout, Signup | Add user authentication to the blog. |
| 25 | Django Relationships: One-to-Many, Many-to-Many | Add a "Comment" model to the blog. |
| 26 | Django Deployment: Deploying to Heroku or PythonAnywhere | Deploy the blog website. |
| 27 | Mini Project: Portfolio Website | Build a personal portfolio website with Django. |
| 28 | Mini Project: Add Blog Feature to Portfolio | Integrate the blog app into your portfolio. |

**Week 9-10: Advanced Django**

**Goal**: Build a complex project and learn deployment.

| **Day** | **Topics to Study** | **Projects/Tasks** |
| --- | --- | --- |
| 29 | Django REST Framework: Building APIs | Create an API for the blog app. |
| 30 | Django Testing: Writing Unit Tests | Write tests for the blog app. |
| 31 | Django Advanced Templates: Template Inheritance | Refactor templates to use inheritance. |
| 32 | Django Security: CSRF, XSS Protection | Implement security features in the blog app. |
| 33 | Django Caching: Improving Performance | Add caching to the blog app. |
| 34 | Mini Project: E-Commerce Website (Part 1) | Set up models for products, categories, and orders. |
| 35 | Mini Project: E-Commerce Website (Part 2) | Add user authentication and a shopping cart. |
| 36 | Mini Project: E-Commerce Website (Part 3) | Implement payment integration (e.g., Stripe). |
| 37 | Mini Project: E-Commerce Website (Part 4) | Deploy the e-commerce website. |

**Week 11-12: Final Projects & Review**

**Goal**: Build a capstone project and review concepts.

| **Day** | **Topics to Study** | **Projects/Tasks** |
| --- | --- | --- |
| 38 | Capstone Project: Social Media App (Part 1) | Set up models for users, posts, and comments. |
| 39 | Capstone Project: Social Media App (Part 2) | Add user authentication and profile pages. |
| 40 | Capstone Project: Social Media App (Part 3) | Implement features like liking and sharing posts. |
| 41 | Capstone Project: Social Media App (Part 4) | Deploy the social media app. |
| 42 | Review Python Basics | Solve Python coding challenges on platforms like LeetCode or HackerRank. |
| 43 | Review Django Basics | Rebuild the blog app from scratch. |
| 44 | Review Advanced Django | Refactor the e-commerce website for better performance. |
| 45 | Practice Deployment | Deploy all projects to a cloud platform (e.g., Heroku, AWS). |
| 46 | Practice Debugging | Debug errors in your projects. |
| 47 | Practice Testing | Write unit tests for all projects. |
| 48 | Final Project: Build a Unique Web App | Create a unique project combining all learned concepts. |
| 49 | Final Project: Polish and Deploy | Add final touches and deploy the project. |
| 50 | Celebrate! 🎉 | Review your progress and celebrate your achievements. |

**Tips for Success**

1. **Consistency**: Dedicate at least **2-3 hours daily** to learning and coding.
2. **Practice**: Build small projects to reinforce concepts.
3. **Ask for Help**: Use forums like Stack Overflow or Reddit if you’re stuck.
4. **Track Progress**: Use Excel to mark completed tasks and track your progress.

Let me know if you’d like an Excel template for this plan! 😊