Freelancing with Python Django - Class Outline

Duration= 4 months - (20000 tk)
Total class = 32 classes
Course duration = 64 hours (4 Months)

Class 1: Orientation Class.

Class 2: Introduction to Django & HTML

- Overview of Django framework and its use in web development.
- Basic HTML structure: tags, headings, paragraphs, links, images, and forms.
- Build a simple HTML page.

Class 3: HTML + CSS Styling

- CSS basics: selectors, properties, and values.
- Flexbox and Grid layout systems.
- Style an **e-commerce product card** (image, title, price, button).

Class 4: Basic JavaScript for DOM & Interactivity

- DOM manipulation: `getElementById`, `querySelector`.
- Event listeners: click, submit, etc.
- Build an **“Add to Cart”** button with a dynamic cart counter.

Class 5: Building the E-Commerce Homepage(Ready Mate Template Customization)

- Find suitable frontend template for project.
- Template Modification.

Class 6: Python Basics & Setup

- Installing Python & IDE (VS Code / PyCharm / etc.)
- Writing & running Python scripts
- Variables & data types (int, float, str, bool)
- Input/output (`print()`, `input()`)
- Basic operators (+, -, *, /, %, //, **)

💡 Mini Task: Write a script that takes user input (name & age) and prints a welcome message.

Class 7: Control Flow

- Conditional statements (`if`, `elif`, `else`)
- Comparison & logical operators
- Indentation rules in Python
- Nested conditions

💡 Mini Task: Write a program that asks for a number and prints whether it's positive, negative, or zero.

Class 8: Loops & Collections

- `for` and `while` loops
- `break` and `continue`
- Lists, tuples, and sets
- Iterating over collections

💡 Mini Task: Write a program that prints all even numbers from 1–50 using a loop.

Class 9: Dictionaries & Functions

- Dictionary basics (key-value pairs)
- CRUD operations with dictionaries
- Functions (`def`, return values, parameters)
- Scope of variables (local vs global)

💡 Mini Task: Create a function that takes a dictionary of students and marks, and prints the name of the student with the highest mark.

Class 10: Python OOP Basics (For Django Models)

- Introduction to classes & objects
- Attributes & methods
- `__init__` method
- Example: Representing a `Student` as a class

💡 Mini Task: Create a `Book` class with attributes `title`, `author`, and `price`. Add a method `discounted_price()` that returns the price after a 10% discount.

Class 11: File Handling & Django-Oriented Concepts

- Reading & writing text files (`open`, `with`)
- Exception handling (`try`, `except`)
- Python virtual environments (very basic intro)
- How Python connects with Django (why we need all these basics)

💡 Mini Task: Write a program that saves a user’s details (name, email) into a file, then reads and prints it.

Class 12: Django Setup & First Project

- Install Django & virtual environment
- Create project & run server
- Project structure explained

💡 Task: Create a project and run it.

Class 13: Apps & URL Routing

- What are apps in Django?
- Create `store` app
- Setup `urls.py` for project & app
- First simple view

💡 Task: Create `/products/` route that shows "Products Page".

Class 14: Templates & Static Files

- Template rendering (`render()`)
- Template inheritance (`base.html`)
- Using static files (CSS/JS/images)

💡 Task: Create homepage with navbar + footer using inheritance.

Class 15: Models & Database

- Create `Product` model (`name`, `price`, `desc`, `image`)
- Migrations (`makemigrations`, `migrate`)
- Django admin setup

💡 Task: Add products via admin.

Class 16: Displaying Products

- Queryset basics (`objects.all()`)
- Passing data to templates
- Looping in templates

💡 Task: Show product list in `/products/`.

Class 17: Product Details Page

- Dynamic URLs (`path('<int:id>/')`)
- `get_object_or_404`
- Product detail template

💡 Task: Show product details at `/products/1/`.

Class 18: User Authentication (Signup & Login)

- Django built-in auth system
- User registration
- Login/logout functionality

💡 Task: Create `/signup/` & `/login/`.

Class 8: User Authentication (Profiles & Restrictions)

- Restrict cart & checkout for logged-in users
- User profile basics
- Logout & redirect

💡 Task: Restrict checkout to logged-in users only.

Class 9: Shopping Cart (Session-Based)

- Use sessions for cart
- Add product to cart
- View cart items

💡 Task: Add "Add to Cart" button to product detail.

Class 10: Cart Management

- Increase/decrease quantity
- Remove item from cart
- Cart total calculation

💡 Task: Implement "Remove" button in cart.

Class 11: Checkout Process

- Checkout form (shipping details)
- Save order data in DB
- Connect cart → order

💡 Task: Create `Order` model with user & address.

Class 12: Order History

- Show user orders in profile
- Display order details
- Track order status (pending, confirmed, delivered)

💡 Task: Show logged-in user’s past orders.

Class 13: Admin Customization

- Customize product admin list view
- Add categories to products
- Manage orders from admin

💡 Task: Add category filter for products in admin.

Class 14: Final Project Review & Polishing

- Improve templates (Bootstrap for styling)
- Add search bar & product filtering
- Add messages (success/error notifications)
- Project walkthrough (end-to-end flow)

💡 Task: Polish UI & test full workflow (Browse → Cart → Checkout → Orders).

12. Django Setup & First Steps

- Install Django and create a new project.
- Create your first Django app.

13. Django Models & Database

- Create `Product` and `Category` models.
- Configure database settings.

- Use Django Admin to add and manage products.

14. Django Views & URLs

- Create product list and product detail views.
- Map views to URLs.

15. Django Templates & Static Files

- Render dynamic product data using templates.
- Integrate CSS and JavaScript from earlier classes.

16. User Authentication

- Implement user signup, login, and logout.
- Restrict checkout access to logged-in users.

17. Shopping Cart Logic in Django

- Use Django sessions to store cart data.
- Add, remove, and update product quantities in the cart.

18. Checkout & Orders

- Create an `Order` model.
- Build a checkout form and order summary page.

19. Search & Filters

- Implement search by product name.
- Add category-based filters.

20. Additional Features (Optional)

- Product reviews and ratings.
- Image uploads for products.

21. Final Project Wrap-up & Q&A

- Debugging tips.
- Next steps for further learning.

22

23

24

25

26. problem solving

27 - 32 classes - freelancing
cv
linkedin
fiverr
upwork