Month 1: Introduction to Python and Web Technologies

Week 1: Introduction to Python

- **Installation & Environment Setup**:
    - Installing Python and IDE setup
    - Introduction to Python shell and IDE
- **Python Basics**:
    - Variables, keywords, and identifiers
    - Data types (string, integer, float, boolean)
    - Basic operations
- **Control Flow**:
    - Conditional statements (if, else, elif)
    - Looping statements (for, while)
    - Break, continue, pass

Week 2: Python Functions and Object-Oriented Programming (OOP)

- **Functions**:
    - Function definition, arguments, and return values
    - Types of functions: normal, lambda, recursive
    - Variable scope and local/global variables

- **Object-Oriented Programming**:
    - Classes and objects
    - Inheritance, polymorphism, method overriding
    - Access specifiers (public, private, protected)

Week 3: Python Advanced Concepts

- **Advanced Data Structures**:
    - Lists, sets, tuples, dictionaries
    - List comprehensions, dict comprehensions
    - Working with data slicing
- **Exception Handling**:
    - Try, except, finally
    - Custom exceptions and assertions
- **File Handling**:
    - Reading and writing files
    - JSON handling and serialization (Pickle)

Week 4: Introduction to Web Technologies

- **HTML5 Basics**:
    - Structure of HTML
    - Tags, attributes, and metadata
    - Basic webpage structure and common tags (div, p, h1, a, img)
- **CSS3 Basics**:
    - Introduction to CSS
    - Box model, padding, margin, borders
    - Styling text, colors, and backgrounds
    - Positioning elements (relative, absolute, fixed)
- **Responsive Web Design**:
    - Mobile-first design principles
    - Introduction to **Bootstrap** for responsive layouts

Month 2: Front-End Development and Flask Framework

Week 5: JavaScript Basics

- **Introduction to JavaScript**:
    - Variables, data types, operators
    - Control structures (if, switch, loops)

    - Functions, events, and DOM manipulation

- **Arrays and Objects**:
    - Arrays: Definition, methods, iteration
    - Objects: Properties, methods, and construction
    - JSON handling in JavaScript

Week 6: Advanced JavaScript

- **JavaScript Advanced Features**:
    - Closures, callbacks, and promises
    - Asynchronous programming (AJAX)
    - Error handling in JavaScript (try-catch)
- **JavaScript in the Browser**:
    - DOM manipulation (creating, modifying, and deleting elements)
    - Event handling (click, keypress, submit)
    - Form validation using JavaScript

Week 7: Flask Framework Basics

- **Introduction to Flask**:
    - Flask installation and setup
    - Basic structure of a Flask app

    - Routing, rendering HTML templates

- **Flask Forms and HTTP Methods**:
    - Creating forms in Flask (GET, POST)
    - Working with form data
    - Handling user input and validation
- **Flask Templates**:
    - Jinja templating engine
    - Dynamic content rendering
    - Template inheritance and blocks

Week 8: Flask Advanced Topics

- **Flask Database Integration**:
    - Introduction to databases (SQL vs NoSQL)
    - Using SQLite/PostgreSQL with Flask
    - Introduction to **SQLAlchemy** ORM for database operations
- **Authentication in Flask**:
    - User login and registration system
    - Session management (cookies)
    - Password hashing and salting

- **Flask Deployment**:
    - Deploying Flask app to **Heroku**
    - Setting environment variables and configuration

Month 3: Django Framework and Database Integration

Week 9: Introduction to Django Framework

- **Django Basics**:
    - Setting up Django project and app
    - Django MVC (Model-View-Controller) architecture
    - URL routing and views
- **Django Templates and Static Files**:
    - Rendering dynamic content with templates
    - Linking static files (CSS, JavaScript, images)
    - Template inheritance in Django

Week 10: Django Models and Forms

- **Django Models**:
    - Creating models and defining fields
    - Database migrations in Django

    - Querying the database with Django ORM

- **Django Forms**:
    - Creating forms in Django
    - Built-in form fields and validation
    - Customizing forms and using model forms
- **Django Admin Panel**:
    - Customizing the Django admin interface
    - Adding models to the admin

Week 11: Django Authentication and REST API

- **Django Authentication**:
    - User registration, login, and logout
    - Session and cookie management in Django
    - Password management (hashing, resetting)
- **Django REST Framework**:
    - Introduction to building APIs with Django
    - Serializers, views, and models in DRF
    - JWT (JSON Web Token) Authentication
    - Building a simple API for CRUD operations

Week 12: Working with Databases in Django

- **Database Management**:
    - Introduction to relational databases (MySQL/PostgreSQL)
    - Using Django ORM for database operations
    - Advanced queries with Django ORM
- **Database Normalization**:
    - Concepts of database normalization
    - Normal forms (1NF, 2NF, 3NF)
    - Relationships between models (One-to-one, One-to-many, Many-to-many)

Month 4: Advanced Topics, Testing, and Final Project

Week 13: Testing and Debugging

- **Introduction to Testing**:
    - Importance of testing in software development
    - Writing unit tests in Python using **unittest** and **pytest**
    - Writing tests for Flask/Django apps
- **Debugging Tools**:
    - Using debuggers in Python (pdb)

- Flask/Django logging and error handling
- Testing APIs with Postman

Week 14: Cloud Deployment and Advanced Topics

- **Deployment in the Cloud**:
  - Hosting applications on **Heroku**, **AWS**, or **DigitalOcean**
  - Configuring databases in the cloud
  - Setting up CI/CD pipelines (e.g., GitHub Actions)
- **Advanced Python Concepts**:
  - Asynchronous programming in Python (asyncio, Flask async)
  - Real-time communication with WebSockets (Flask-SocketIO, Django Channels)
  - Introduction to Docker for containerization

Week 15: Full-Stack Project - Flask

- **Build a Full-Stack Application**:
  - Create a simple Flask-based project (e.g., a blog or e-commerce site)
  - Integrate Flask with MySQL/PostgreSQL
  - Implement CRUD operations and user authentication
- **Project Deployment**:

    - Deploy the full-stack application to **Heroku** or **AWS**
    - Connect the app to the cloud database

Week 16: Full-Stack Project - Django

- **Build a Full-Stack Application**:
    - Create a Django-based project (e.g., a task manager or social media app)
    - Implement Django REST API for backend
    - Connect the Django app with a PostgreSQL database
- **Project Deployment**:
    - Deploy the Django application to **Heroku** or **AWS**
    - Set up production-ready settings for Django