Learn Django

Phase 1: Prerequisites (3 days)

1. Python basics:
    - Variables, data types, loops, conditional statements, functions, modules
    - Familiarize yourself with Python 3.x
2. HTML, CSS, and JavaScript basics:
    - Understand web development fundamentals
3. Install Python, pip, and a code editor (e.g., PyCharm, VSCode)

Phase 2: Django Fundamentals (2 weeks)

1. Introduction to Django:
    - Overview, features, and advantages
    - Install Django and create a new project
2. Django Core Concepts:
    - Models, views, templates, URLs, and

databases

- Understand MTV (Model-Template-View) architecture

3. Django ORM (Object-Relational Mapping):
   - Interact with databases using Django's ORM
4. Django Admin Interface:
   - Understand and customize the admin dashboard
5. Basic Django App:
   - Build a simple app (e.g., todo list, blog)

Phase 3: Django Advanced Topics (2 weeks)

1. User Authentication and Authorization:
   - Implement user registration, login, and permissions
2. Forms and Validation:
   - Handle user input with Django forms

3. File Uploads and Static Files:
    - Manage file uploads and static assets
4. Django REST Framework (DRF):
    - Build RESTful APIs
5. Advanced Database Concepts:
    - Query optimization, indexing, and transactions

Phase 4: Project Development and Deployment (3 weeks)

Project: "EduConnect" - A Social Learning Platform

Description: EduConnect is a web application that connects students, teachers, and institutions for collaborative learning, resource sharing, and community building.

Features:

1. User Authentication:
    - Student, Teacher, and Admin roles
    - Registration, Login, and Password Recovery
2. Profiles and Networking:
    - User profiles with bio, interests, and skills
    - Follow/Unfollow, Friend requests, and Connections
3. Course Management:
    - Create, Edit, and Delete courses
    - Assign teachers, students, and resources
4. Resource Sharing:
    - Upload, Share, and Download files (docs, videos, images)
    - Resource categorization and tagging
5. Discussion Forums:
    - Course-specific discussion boards
    - Threaded comments, replies, and

notifications

6. Event Calendar:
    - Create, Edit, and Delete events
    - Event reminders and notifications
7. Blogging:
    - Article creation, editing, and publishing
    - Category, tagging, and commenting system
8. Search and Filtering:
    - Search users, courses, resources, and events
    - Filter by categories, tags, and dates
9. Admin Dashboard:
    - User management, analytics, and reporting
    - Course, resource, and event management

Technical Requirements:

1. Django 4.x

2. Python 3.x
3. Database: PostgreSQL or MySQL
4. Frontend: HTML, CSS, JavaScript, Bootstrap
5. APIs: Django REST Framework (DRF) for mobile app integration
6. Deployment: Heroku or AWS

Phase 4 Tasks:

1. Design and implement the database schema
2. Create models, views, templates, and URLs for each feature
3. Implement user authentication and authorization
4. Develop the frontend using HTML, CSS, and JavaScript
5. Integrate APIs for mobile app connectivity
6. Test and debug the application

7. Deploy the application on Heroku or AWS

Learning Outcomes:

1. Comprehensive understanding of Django's MTV architecture
2. Practical experience with Django's ORM, authentication, and authorization
3. Familiarity with frontend development using HTML, CSS, and JavaScript
4. Knowledge of API development using DRF
5. Experience with deployment on cloud platforms

Tips and Variations:

1. Add features like gamification, badges, or rewards
2. Integrate payment gateways for course purchases

3. Implement video conferencing or live streaming
4. Use machine learning for personalized recommendations

This project encompasses various aspects of Django development, allowing you to apply concepts learned throughout the roadmap. Good luck with your project!