Guide2Code - Django Roadmap

🟢 Phase 1: Beginner Level

📚 Topics to Learn:

1. Introduction to Django (What is Django? Why use it?)
2. Setting Up Django (Installing Django, Virtual Environments)
3. Django Project Structure (Understanding the MVT Pattern)
4. Creating and Running a Django App (manage.py, settings.py)
5. Django Models (Defining Models, Migrations, ORM Basics)
6. Django Views & Templates (Rendering HTML, Template Tags)
7. Django Forms (Handling Forms, Form Validation)
8. URL Routing & Path Converters (urls.py, path(), re_path())
9. Static Files & Media Handling (Serving CSS, JS, and Images)
10. Handling User Authentication (Login, Logout, User Model)

🛠 Beginner Project Ideas:

- **Personal Blog Website** – Create and manage blog posts
- **To-Do List App** – Add, edit, and delete tasks
- **Weather App** – Fetch weather data using an API
- **Simple Contact Form** – Store user messages in the database
- **Notes App** – Save, edit, and delete personal notes

🟡 Phase 2: Intermediate Level

📚 Topics to Learn:

1. Django Admin Panel Customization (Registering Models, Admin Actions)
2. Django ORM Deep Dive (QuerySets, Filtering, Aggregation)
3. Django Rest Framework (DRF) (Building RESTful APIs)
4. User Authentication & Authorization (Custom User Models, JWT)
5. Django Class-Based Views (CBVs) (Generic Views, Mixins)
6. Working with Databases (PostgreSQL, MySQL, SQLite)
7. Email Sending in Django (SMTP, SendGrid, Celery for Background Tasks)

8. File Uploading & Management (Handling Images and Documents)
9. Deployment Basics (Gunicorn, Nginx, Heroku, DigitalOcean)
10. Caching & Performance Optimization (Redis, Memcached)

🛠 Intermediate Project Ideas:

- **E-commerce Website** – Product listings, cart, and checkout system
- **Job Portal** – Post and apply for jobs
- **Online Polling System** – Create and manage online polls
- **Expense Tracker** – Track personal or business expenses
- **API for a Mobile App** – Build a REST API using Django REST Framework

🔴 Phase 3: Advanced Level

📚 Topics to Learn:

1. Advanced Django Rest Framework (Throttling, Versioning, Permissions)
2. Django Channels (Real-Time Communication with WebSockets)
3. Microservices with Django (Decoupled Architecture, Async Processing)
4. GraphQL with Django (Using Graphene, Queries, Mutations)
5. Containerization & DevOps (Dockerizing Django Apps, Kubernetes)
6. Serverless Django (Deploying on AWS Lambda, Google Cloud Functions)
7. Payment Integration (Stripe, PayPal, Razorpay)
8. Full-Text Search (ElasticSearch, Haystack, Whoosh)
9. Advanced Security (Django Security Middleware, Rate Limiting)
10. Continuous Integration & Deployment (CI/CD with GitHub Actions, Jenkins)

🛠 Advanced Project Ideas:

- **Social Media Platform** – User profiles, posts, comments, and likes
- **AI-Powered Chatbot** – Integrate OpenAI for automated chat responses
- **Online Learning Platform** – Courses, quizzes, and student tracking
- **Real-Time Stock Market Dashboard** – Display stock prices with WebSockets
- **Multi-Tenant SaaS Application** – A scalable SaaS app with Django