1. What is an API (Application Programming Interface)?

* An API is a set of rules that allows different software applications to communicate and exchange data with each other.

1. Types of APIs: REST, SOAP.

* REST APIs use HTTP and are lightweight and flexible. SOAP APIs use XML and are more secure but heavier.

1. Why are APIs important in web development?

* APIs allow applications to communicate with each other, integrate third-party services, and build scalable systems.

1. Understanding project requirements.

* Before development, understanding what features the project needs helps plan the backend structure and APIs.

1. Setting up the environment and installing necessary packages.

* Install Python, Django, and Django REST Framework using pip install django djangorestframework.

1. What is Serialization?

* Serialization converts complex data (like QuerySets) into native Python data types or JSON for API responses.

1. Converting Django QuerySets to JSON.

* Use Django REST Framework serializers or serializers.serialize('json', queryset) to convert data.

1. Using serializers in Django REST Framework (DRF).

* Serializers in DRF help convert model instances to JSON and validate input data from API requests.

1. HTTP request methods (GET, POST, PUT, DELETE).

* GET retrieves data, POST creates, PUT updates, and DELETE removes data on the server.

1. Sending and receiving responses in DRF

* DRF uses Response objects to send JSON data, often with status codes like 200 OK or 400 Bad Request.

1. Understanding views in DRF: Function-based views vs Class-based views

* Function-based views use decorators; class-based views offer more structure and reuse with DRF generic views.

1. Defining URLs and linking them to views.
2. Use path() in urls.py to connect endpoints to specific view functions or classes.
3. Adding pagination to APIs to handle large data sets

* DRF provides pagination classes like PageNumberPagination to limit data per page and reduce load.

1. Configuring Django settings for database, static files, and API keys.

* Edit settings.py to add DB config, set STATIC\_URL, and store API keys securely using os.environ.

1. Setting up a Django REST Framework project.

* Create a Django project, add 'rest\_framework' in INSTALLED\_APPS, and configure URLs and views.

1. Implementing social authentication (e.g., Google, Facebook) in Django.

* Use packages like django-allauth or social-auth-app-django for social login integrations.

1. Sending emails and OTPs using third-party APIs like Twilio, SendGrid.

* Use SendGrid for email and Twilio for SMS OTPs by installing their SDKs and calling their APIs.

1. REST principles: statelessness, resource-based URLs, and HTTP methods for CRUD operations.

* REST follows stateless calls, uses nouns in URLs for resources, and maps HTTP methods to CRUD.

1. What is CRUD, and why is it fundamental to backend development?

* CRUD stands for Create, Read, Update, Delete—core actions to manage data in databases.

1. Difference between authentication and authorization.

* Authentication verifies identity; authorization checks permissions for actions after login

1. Implementing authentication using Django REST Framework’s token-based system.

* Install rest\_framework.authtoken, add it to INSTALLED\_APPS, and use TokenAuthentication in settings.

1. Introduction to OpenWeatherMap API and how to retrieve weather data.

* Sign up, get an API key, and use HTTP GET requests like /weather?q=City&appid=KEY.

1. Using Google Maps Geocoding API to convert addresses into coordinates.

* Use /geocode/json endpoint with address and API key to get latitude and longitude.

1. Introduction to GitHub API and how to interact with repositories, pull requests, and issues.

* GitHub REST API allows access to repos, PRs, and issues using token-authenticated HTTP requests.

1. Using Twitter API to fetch and post tweets, and retrieve user data.

* Use Twitter’s API with OAuth to send tweets, get timeline data, or access user info.

1. Introduction to REST Countries API and how to retrieve country-specific data.

* REST Countries API provides country details like capital, population, and flag via GET requests

1. Using email sending APIs like SendGrid and Mailchimp to send transactional emails.

* SendGrid and Mailchimp APIs allow sending automated or bulk emails using simple POST requests.

1. Introduction to Twilio API for sending SMS and OTPs.

* Twilio API allows sending SMS and OTPs using your Twilio phone number and authentication token.

1. Introduction to integrating payment gateways like PayPal and Stripe.

* Use their SDKs or APIs to securely handle payments, refunds, and customer billing.

1. Using Google Maps API to display maps and calculate distances between locations.

* Google Maps API provides map rendering and distance calculation using coordinates and markers.