2 . To compress data in a Django CRUD view, you can use Django's built-in compression middleware. This middleware can automatically compress response content for browsers

3. Query parameters included in the URL and allows you to pass additional information to a view in Django. They are commonly used for filtering and searching a view.

4. HttpResponse is a class provided by Django for generating HTTP responses. It allows you to return a response with content, status code, and headers. The renderers used by the Response class can't be used in django to use response I need to serialize the data into primitive datatypes before creating the Response object.

5. Class-based views (CBVs) and function-based views (FBVs) are two ways to define views in Django. Function-based views are Python functions. They take a request as an argument and return an HTTP response. They are easy to use for one specific HTTP method. Class-based views are classes. They have methods to handle various HTTP methods. They are good to use for more complex views, especially when the same view needs to respond to different HTTP methods.

6. To make an FBV handle multiple HTTP methods, I can use conditional statements to each request . for example, I can use if statement to see which request is it .

7. when I create an object without his primary key the `save()` function determines to create a new object. the `save()` function update an existing object when I retrieve an existing object from the database and the primary key is already set.

8. `full\_clean()` method is good To handle errors when the validation didn't work when saving a model instance .This function allows to catch validation errors before attempting to save the object, ensuring that you don't save an invalid object to the database.

9. this line performs validation on the serializer instance and raises an exception if the data being validated is not valid. this way is better than the original way because it allows to catch validation errors without having to write custom logic for error handling.