**Task: Create a GraphQL Layer for an Existing REST API**

**Objective:**

Your goal is to create a GraphQL service that connects to an existing REST API and exposes the same data using GraphQL queries.

This will help you understand how GraphQL works and how it can be used to interact with existing REST services.

**1. Define a GraphQL Schema**

* Create a schema that defines the types (like User, Post, etc.) and queries (like get all users, get post by ID, etc.).
* These types and queries should match the data you get from the REST API.

**2. Create Resolvers**

* Resolvers are functions that run when someone asks for data using a GraphQL query.
* Each resolver should call the appropriate REST API endpoint to get the data needed for the query or field being requested.
* Include both Query Resolvers and Field Resolvers:

Query Resolvers: Used for top-level queries (like getting a list of users or a single post).

Field Resolvers: Used for nested fields (for example, fetching the user details inside a post).

**3. Add proper error handling**:

* If the REST API returns an error or fails to respond, the resolver should catch the error.
* Return a meaningful error message so the client knows what went wrong (e.g., “User not found” or “Failed to fetch data”).\

**4. Test Your GraphQL API**

* Make sure your GraphQL server runs correctly.
* Use a tool (like GraphQL UI) to write sample queries and test the results.

**5. Write Tests**

* Write Unit Tests to test individual resolver functions using mock data.
* Write Functional Tests to test the overall GraphQL API by sending real queries and checking if the correct data is returned.