

GraphQL

Question 1. What is GraphQL? How is it different from REST API?

Answer:

GraphQL is a query language for APIs developed by Facebook that allows clients to request only the data they need.

Difference between GraphQL and REST:

GraphQL	REST
Single endpoint	Multiple endpoints
Client chooses data	Server decides data
Strongly typed schema	No strict typing
Reduces over-fetching	Can cause over/under-fetching

Question 2. What are the key concepts of GraphQL?

Answer:

The main key concepts are:

- Schema – Defines structure of data and types.
- Query – Used to fetch data.
- Mutation – Used to modify data (Add, Update, Delete).
- Resolver – Function that returns data for queries/mutations.
- Type – Defines structure of data (e.g., type User).

Question 3. What are the benefits of GraphQL?

Answer:

- Fewer API calls
- Strongly typed queries
- Fetch only required data
- Better performance
- Ideal for frontend applications

Question 4. What is Apollo Client?

Answer:

Apollo Client is a GraphQL client library used in React applications to connect to a GraphQL server. It helps in:

- Sending queries
- Handling mutations
- Managing cache
- Handling loading and error states

Question 5. What is ApolloProvider?

Answer:

ApolloProvider connects the React application with the Apollo Client. It wraps the root component and allows all child components to access GraphQL data.

Question 6. What is useQuery hook?

Answer:

useQuery is a hook provided by Apollo Client to fetch data from a GraphQL API.

It returns:

- loading → Shows loading state
- error → Shows error state
- data → Contains fetched data

Example:

```
const { loading, error, data } = useQuery(GET_USERS);
```

Question 7. How do you handle loading and error states in GraphQL?

Answer:

```
if (loading) return <p>Loading...</p>;
```

```
if (error) return <p>Error occurred!</p>;
```

Question 8. What is useMutation hook?

Answer:

useMutation is used to modify data in GraphQL.

It is used for:

- Add data
- Update data

- Delete data

Example:

```
const [addUser] = useMutation(ADD_USER);
```

Question 9. What is Optimistic UI Update?

Answer:

Optimistic UI update means updating the UI immediately before the server confirms the change.

It improves user experience by showing instant results.

Question 10. Explain GraphQL Schema with example.

Answer:

Schema defines types and structure of API.

Example:

```
type User {  
  id: ID!  
  name: String!  
  email: String!  
}
```
