Period 3 GraphQL

Note: This description is too big for a single exam-question. It will be divided up into separate questions for the exam

Explain shortly about GraphQL, its purpose and some of its use cases

With GraphQL we get to fetch ONLY the data we need(Under Over fetching). As read by the name it's a query language. And you get everything in a single endpoint, where the query defines the amount you get.

https://github.com/Groenbek/Period3/tree/main/startcode/src/graphql

Explain some of the Server Architectures that can be implemented with a GraphQL backend

https://www.howtographql.com/basics/3-big-picture/

What is meant by the terms over- and under-fetching in GraphQL, compared to REST

When fetching there is chance of getting irrelevant data or not getting all the data you want. With GraphQL you get to chose specifically what you want and need

Overfetching - Fetching more data than necessary

Underfetching - Fetching too little data

Explain shortly about GraphQL's type system and some of the benefits we get from this

https://www.howtographql.com/basics/1-graphql-is-the-better-rest/

With GraphQLs type system the frontend and backend can work separately thanks to the schemes and the already known types.

Explain shortly about GraphQL Schema Definition Language, and provide examples of schemas you have defined.

The schemas we're working with have queries and mutations which can be seen at

https://github.com/Groenbek/Period3/blob/main/startcode/src/graphql/schema.ts

Provide examples demonstrating data fetching with GraphQL. You should provide examples both running in a Sandbox/playground and examples executed in an Apollo Client

Simply run the startcode and the frontend

Backend - https://github.com/Groenbek/Period3/tree/main/startcode

Frontend -

https://github.com/Groenbek/Period3/tree/main/apollo-client-for-lynda-server

https://github.com/Groenbek/Period3/tree/main/apolloClientWithLogin

Provide a number of examples demonstrating; creating, updating and deleting with Mutations. You should provide examples both running in a Sandbox/playground and examples executed in an Apollo Client.

https://github.com/Groenbek/Period3/blob/main/apollo-client-for-lynda-server/src/components/AddFriend.tsx

To view GraphQL, run the staretcode and the apollo frontend

Explain the Concept of a Resolver function, and provide a number of simple examples of resolvers you have implemented in a GraphQL Server.

A resolver function is a function that runs the code stated when a query or mutation is called from GraphQL.

https://github.com/Groenbek/Period3/blob/main/startcode/src/graphql/resolvers.ts

Explain the benefits we get from using a library like Apollo-client, compared to using the plain fetch-API

Benefits are it allows us to remotely and locally manage data with GraphQL. It fetches, caches and modify data.

In an Apollo-based React Component, demonstrate how to perform GraphQL Queries, including:

- Explain the purpose of ApolloClient and the ApolloProvider component

  ApolloProvider allows all our react hooks to use the same ApolloClient. This means no lifting state is needed

- Explain the purpose of the gql-function (imported from graphql-tag)

  It defines a mutation or query. Which can be executed later.

- Explain Custom Hooks used by your Client Code

  Our data is sent to a hook through the ApolloProvider

- 🟥 Explain and demonstrate the caching features built into Apollo Client

  https://github.com/Groenbek/Period3/blob/main/apolloClientWithLogin/src/components/AllFriends.tsx

  Line 24

In an Apollo-based React Component, demonstrate how to perform GraphQL Mutations?

They can be found here

https://github.com/Groenbek/Period3/blob/main/apolloClientWithLogin/src/components/AddFriend.tsx

Demonstrate and highlight important parts of a "complete" GraphQL-app using Express and MongoDB on the server-side, and Apollo-Client on the client.

https://github.com/Groenbek/Period3/blob/main/startcode/src/graphql/resolvers.ts

https://github.com/Groenbek/Period3/blob/main/startcode/src/graphql/schema.ts

https://github.com/Groenbek/Period3/blob/main/startcode/src/app.ts

Client:

https://github.com/Groenbek/Period3/blob/main/apolloClientWithLogin/src/components/AddFriend.tsx

https://github.com/Groenbek/Period3/blob/main/apolloClientWithLogin/src/components/AllFriends.tsx

https://github.com/Groenbek/Period3/blob/main/apolloClientWithLogin/src/components/App.tsx