Spørgsmål 2

Question

Explain and show examples of how to design and implement a Web API with Node.js that includes data persistence to a database. Explain the principles for how to design a RESTful Web API. And discuss pros and cons of a RESTful Web API vs GrapQL.

GraphQL

- GraphQL stands for Graph Query Language
- It was created by facebook due to the amount of API calls being called from different endpoints To retrieve the needed data. A lot of extra unecessary data got retrieved too.
- The idea was therefore to create an API that can get the specific data by requesting a query.
- The API maps out the different data from the database into a graph hence the name Graph
- GraphQL Is really fast but is verry complex to implement.
- It is not suited for samller application
- GraphQL is a specification, which means there isn't a right or wrong way to implement it.
- The only requirements are that the quesries and schemas represent the GraphQL specification

Pros

- Very fast and flexible

Cons

- Complex implementation
- Only viable in bigger applications

RESTAPI

- REST is not a standard but an architectual style
- Created as a simpler alternative to SOAP and WSDL
- Helps prevent Swamp of POX(Plain Old XML)
- 4 basic architectual design principels creates REST
    - HTTP MEthods exclusivly
        - GET,POST,PUT,DELETE(CRUD)
        - The body should carry resources and not instructions on what functions to call, The verb handles that

- A non-RESTful API:

```
GET /adduser?name=Robert HTTP/1.1
```

- A RESTful API:

```
POST /users HTTP/1.1
Content-Type: application/json
{name : Robert}
```

    -
  - Being stateless(also cacheable)
    - Server should not know the state of the client on its own
    - Everything staterelated should be handled on the client and the resulting data retrieved from the server in a normal fasion.
    - Resources can be cached in the client so the server should depict how long it can be cached
  - Calling the directory endpoints in the form of URIs
    - Referes to the URL which depicts which resource to be retrieved
  - Transfer media should be something like XML,JSON or other Internet media type content

Pros

- Widely used a lot of information on the internet
- Frameworks such as .NET provides autogeneration of the core implementation. Reduces implementation time.
- Easy to use in all sizes of applications
- Server can send information about resource cachability to the client, which means that in some cases the client doesn't need to go to the server to retrieve a cached request.
- Allows serverside authentication to verify the request

Cons

- The stateless design means more logic needs to be implemented into the client.

Explain, Design and Implement WEB API with Node.js

- A WEB API is a endpoint that applications can communicate with to retrieve data. does not return html but returns data
- When sending a request a response will be returned.
  - Response should send a status code and data
- When requesting data, the request is based on the URI, which contains parameters that singles out the needed information

- When designing a web API the calls mostly gets seperated in two, the definition of the API call and a controller that contains the logic linked to the defined call.
- Designing an API it is important to be aware of CORS. CORS is a safety measure created to prevent one domain to retrive resources from another domain. By defining in the header that cross origin request is allowed the API call can go through
- When designing a WEB API Swagger is a testing tool that can help test it efficiently.
- ***CODE SAMPLE***