Spørgsmål 6

Question

Show how to implement authentication and authorisation in a Web application.
Show how to use JWT as authentication in a Web API.
Explain how to publish a Web application to the production environment whether on local server or a cloud based hosting.

Authentication and Authorisation

Authentication and authorisation are two different terms but is mostly used in a similar situation.

1. Authentication referes to having to ensure a person is who they are. ex. by verifying credentials such as email and password.
2. Authorisation referes to specifying access to the authenticated user. ex. you can't view your lecture documents on blackboard, if you aren't authenticated(logged in), even if you have the direct link. This is to ensure that potential important info isn't leaked.
    1. This can also be extended further upon. By specifying roles, that have limited priveliges. This would potentially end up being catagorised as an RBAC system.
3.

JWT

1. JSON Web Token is a standard that defines a way for secure transmission between parties.
2. JWT focuses on signed tokens which is a way to verify integrity.
3. Common usecase is authorisation, ex. after login, the token is sent with the request header as a bearer token. The token will be verified for every specified route to ensure that the user got authorisation to access.
4.

Heroku

Heroku is a platform that enables a developer to deploy and host a web application. By using the Heroku CLI an app can be deployed in seconds.

```
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp> heroku create
Creating app... done, ⬀ fierce-island-03042
https://fierce-island-03042.herokuapp.com/ | https://git.heroku.com/fierce-island-03042.git
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp>
```

```
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp> git push heroku main
Enumerating objects: 27, done.
Counting objects: 100% (27/27), done.
Delta compression using up to 12 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (27/27), 12.92 KiB | 3.23 MiB/s, done.
Total 27 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:        NPM_CONFIG_LOGLEVEL=error
remote:        NODE_ENV=production
remote:        NODE_MODULES_CACHE=true
remote:        NODE_VERBOSE=false
remote:
remote: -----> Installing binaries
remote:        engines.node (package.json):  unspecified
remote:        engines.npm (package.json):   unspecified (use default)
remote:
remote:        Resolving node version 12.x...
remote:        Downloading and installing node 12.20.1...
remote:        Using default npm version: 6.14.10
remote:
remote: -----> Installing dependencies
remote:        Installing node modules
remote:        added 85 packages in 1.836s
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:        - node_modules
remote:
remote: -----> Pruning devDependencies
remote:        audited 86 packages in 1.179s
remote:
remote:        2 packages are looking for funding
remote:          run `npm fund` for details
remote:
remote:        found 0 vulnerabilities
remote:
remote:
remote: -----> Build succeeded!
```

```
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp> heroku ps:scale web=1
Scaling dynos... done, now running web at 1:Free
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp>
```

Here you tell heroku that there should always be a working instant running.

```
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp> heroku open
PS C:\Users\abdul\OneDrive\Documents\Coding\ITWeb\Authentication\authentication_sampleapp>
```

1. In essence deploying is the process of making your appilcation both accesible and useable on non-development-environment.
2. As an example when using heroku it's important to set the environmental port variable, because that's what heroku uses when deploying.
3. Other than that connection strings to databases and potentially tokens to JWT authentication should be taken out of the code and placed in enviromental variables as well.
    1. if they are saved within the code, unautherised people can possibly get their hands on them and delete the content of the db or get security clearence.

4. If a developer wants to host something locally the important thing is making your network accsesible from anywhere, which means creating firewall rules that enables extern clients to listen to specific ips and ports.
    1. This can create a security risk, which is why it is recommended to host on a cloud based service(remote server).

APP url:
https://ittweb-gruppe18-workoutwebsite.herokuapp.com/login

API url:
https://ittweb-gruppe18-workoutapi.herokuapp.com/

Gyldig Login til Angular App:

---

mail: jens@test.dk
password: test
