Eksamen Rapport

Kandidat Nummer: 30

For this exam, I used React and Tailwind frontend, and a .NET/C# backend connected to an SQLite database. While I installed Tailwind, I didn't fully utilize it for every part of the project. Instead, I explored multiple approaches to achieve the desired results and relied on a mix of techniques to bring the project together. The application focuses on managing data in three main categories: Trump Thoughts, Trump Merchandise, and Trump Staff.

Frontend

On the frontend, I used React for building the user interface and Tailwind CSS for responsive styling. Additionally, I used an `index.css` file to handle standard styling throughout the project. This file was crucial for defining background colors, fonts, and consistent design elements across the application. On some pages, like the info page, I included traditional CSS for more specific styling needs. This combination allowed me to experiment with various approaches to design and layout.

I designed the app to be accessible and usable across devices like phones, tablets, and desktops. I aimed to follow Universal Design principles, such as ensuring semantic HTML, high color contrast for readability, and providing alt text for all images. I also tried to apply the BEM methodology for naming classes wherever possible to maintain clear and organized code. The React app communicates with the backend using Axios to perform CRUD operations, enabling users to create, read, update, and delete data dynamically.

Backend

On the backend, I created a Web API using .NET/C#, which connects to an SQLite database to store the application's data. Each category (TrumpThoughts, TrumpMerch, TrumpStaff) has its own dedicated controller to handle CRUD operations, ensuring that the logic for each section remains clear and modular. Additionally, I implemented an image upload feature for merchandise, allowing users to add product visuals directly to the application. (I mixed using the database and localstorage)

Reflection

This project was a great learning experience and a true challenge. I explored different methods and often went back and forth between approaches to figure out the best solutions. Balancing new techniques with what I already knew pushed me to improve my coding skills. The end result is a functional and responsive application that showcases everything I've learned so far. This process also taught me the importance of flexibility and problem-solving in development.