TALLINNA TEHNIKAÜLIKOOL
School of Information Technologies

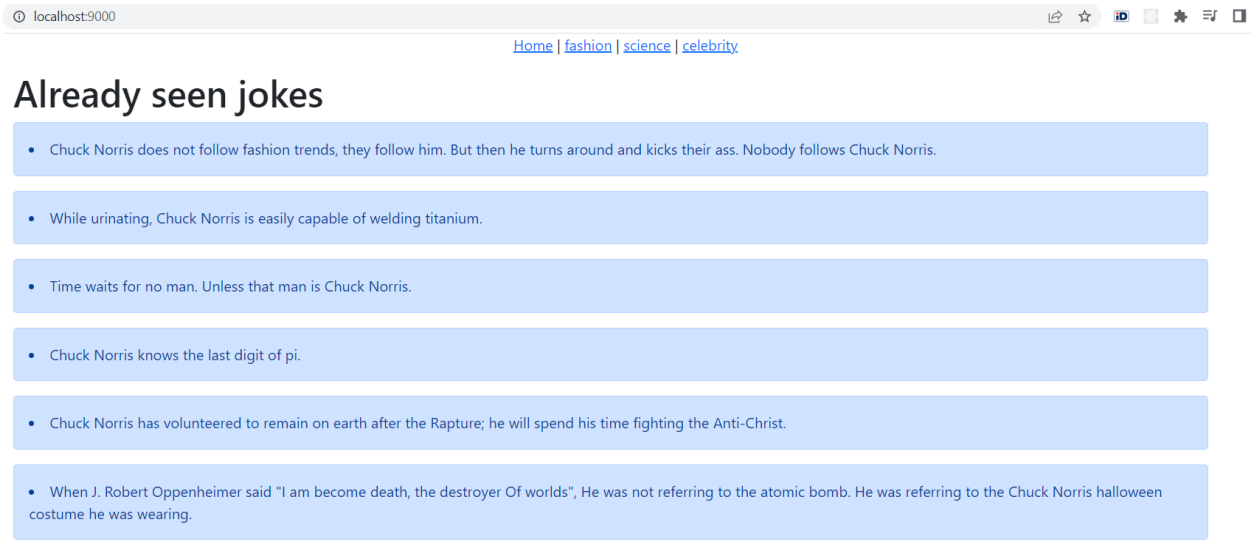
Ahmed Abdullajev 192816IADB

HOMEWORK 2 LEG 1 – CHUCKNORRIS APP AURELIA

JavaScript

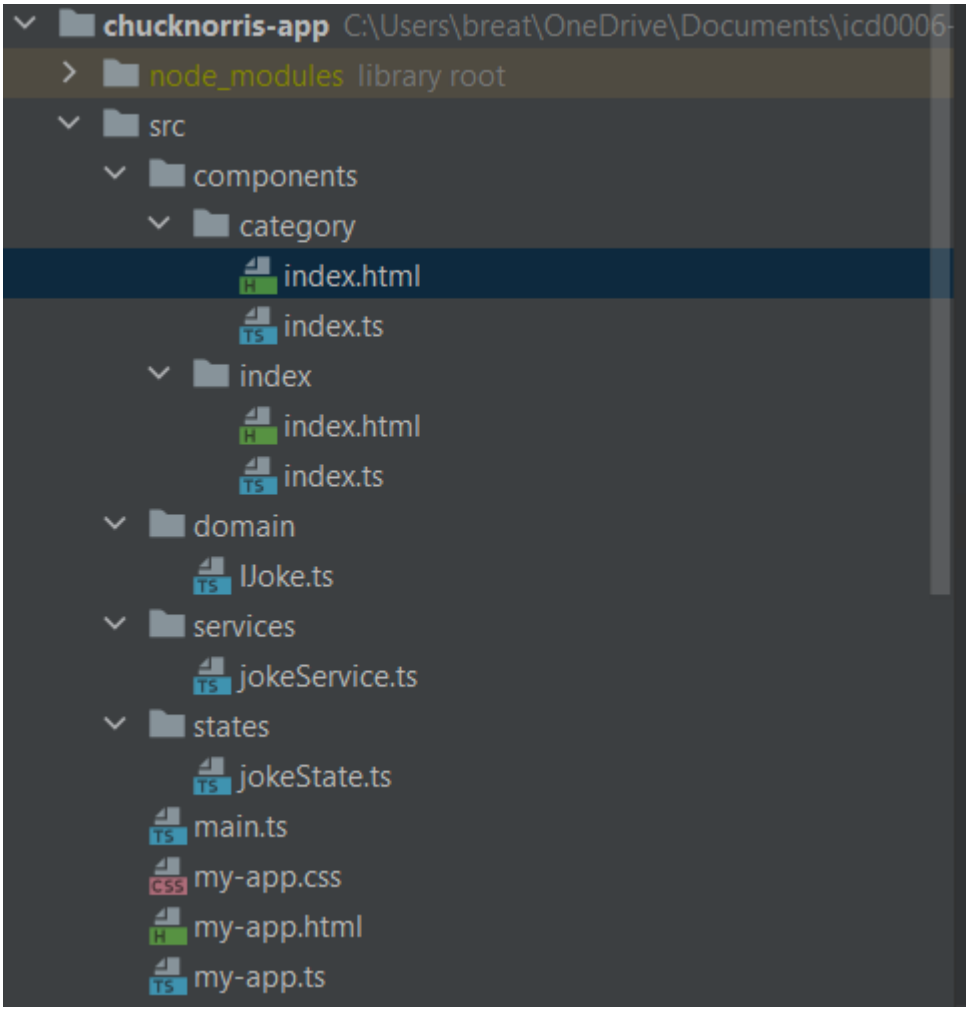
TALLINN 2022

SCREENSHOOTS



You can check my code below or in my school repository:
https://gitlab.cs.ttu.ee/ahabdu/icd0006-21-22-s

PROJECT STRUCTURE



MY-APP.TS

```
import { ILogger, route} from 'aurelia';
import {JokeService} from "./services/jokeService";
@route({
  routes: [
    {
      id: 'main',
      path: '',
      component: import('./components/index'),
      title: 'Main',
    },
    {
      id: 'results',
      path: '/category/:cat',
      component: import('./components/category'),
      title: 'Results',
    },

  ]
})
export class MyApp{
  private categories : string[] = [];
  constructor(
      @ILogger private logger: ILogger,
      private JokeService: JokeService
  ) {
    this.logger = logger.scopeTo("MyApp");
  }
  async created(){
     let data = await this.JokeService.getRandomCategories(3);
     this.categories = data;
  }

}
```

Category/index

```
import {ILogger, IRouter, inject, Params, IRouteViewModel} from 'aurelia';
import {IJoke} from "../../domain/IJoke";
import {JokeService} from "../../services/jokeService";
import {JokeState} from "../../states/jokeState";

@inject()
export class Index {
    private state: JokeState;

    private tempJokes: IJoke[] = [];
    private category : string = "No category";
    constructor(
        @IRouter private router: IRouter,
        @ILogger private logger: ILogger,
        private JokeService: JokeService,
        JokeState: JokeState) {
        this.state = JokeState;
    }
    async load(params: Params) {
        this.category = params['cat']
        this.tempJokes = []
        for (let i = 0; i < 5; i++){
            let joke = await this.getJokes();
            this.tempJokes.push(joke)
            if (this.checkInList(joke) === true) {
                this.state.addJoke(joke)
            }
        }
    }
    async getJokes(){
       return  await this.JokeService.getJoke(this.category);
    }
    checkInList(data : IJoke){
        for (let i = 0; i < this.state._jokes.length; i++){
            if(this.state._jokes[i].id == data.id){
                return false;
            }
        }
        return true;
    }

}
```

HTML

```
<div class="container">
<h1>${category}</h1>
<div class="alert alert-primary" role="alert">
    <ul>
    <template repeat.for="joke of tempJokes">
        <li class="">${joke.value}</li>
    </template>
    </ul>
</div>
</div>
```

Index/index (main page with already seen jokes)

```
import {EventAggregator, ILogger, inject, IRouter} from 'aurelia';
import {JokeState} from "../../states/jokeState";

@inject()
export class Index {
    private state: JokeState;
    constructor(
        @IRouter private router: IRouter,
        @ILogger private logger: ILogger,
        JokeState: JokeState) {
        this.state = JokeState;
        this.logger = logger.scopeTo("Index");
    }

    attached() {
        this.logger.debug("attached");
    }

    async go(event: PointerEvent) {

        event.preventDefault();
        this.logger.debug("GO");
        await this.router.load('results/somevalue',
            {
                title: 'My product',
                queryParams: {
                    bar: 'one'
                }
            }
        );
    }
}
```

HTML

```
<template>
  <div class="container">
  <h1>Already seen jokes</h1>
    <h1>
      ${state._jokes.length ? '' : 'No jokes'}
    </h1>
  <template repeat.for="item of state._jokes">
    <div class="alert alert-primary" role="alert">
      <li class="">${item.value}</li>
    </div>
  </template>
  </div>
</template>
```

IJoke domain

```
export interface IJoke{
    categories: string[];
    created_at: string;
    icon_url: string;
    id: string;
    updated_at: string;
    url: string;
    value: string;
}
```

services/jokeService.ts

```
import {HttpClient, inject} from "aurelia";
import {IJoke} from "../domain/IJoke";

@inject()
export class JokeService {
    constructor(private httpClient: HttpClient) {
    }
    private async getCategories(url : string) : Promise<string[]>{
        return await this.httpClient.get(url, { cache: "no-store" })
            .then((data) =>{
                return data.json();
            }).catch((err) => {
                console.warn(err)
            })
    }
    async getRandomCategories(amount: number) : Promise<string[]>{
        let array : string[] = await
this.getCategories("https://api.chucknorris.io/jokes/categories");
        let shuffled = array.sort(()=>{return 0.5 - Math.random()});
        return shuffled.slice(0, amount);
    }
    async getJoke(cat: string): Promise<IJoke> {
        const url = "https://api.chucknorris.io/jokes/random?category="
        return await this.httpClient.get(url + cat, { cache: "no-store" })
            .then((data) =>{
                return data.json();
            })
            .catch((err)=>{
                return err;
            });
    }
}
```

States/jokeState.ts

```
import {IJoke} from "../domain/IJoke";


export class JokeState{
    _jokes: IJoke[] = []

    getJokes = () => this._jokes;

    addJoke(joke : IJoke): void{
        this._jokes.push(joke);
    }
}
```