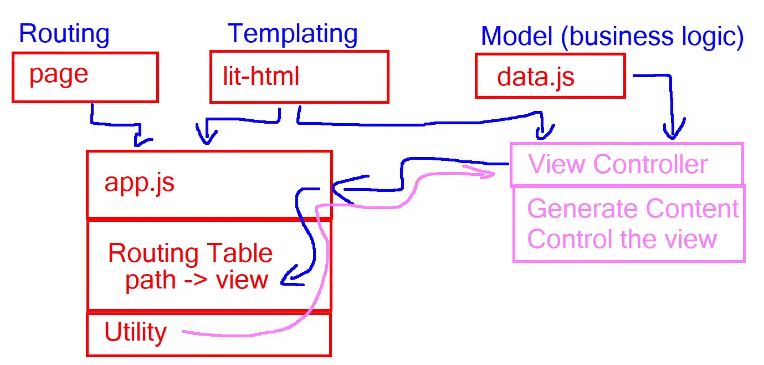
Изпит (max точки 100 + 5 bonus)

  
1.Сваляме ресурси – softuni или judge

2. Отваряме VSC с code .

3. В терминал   
- npm install  
- npm run start  
- npm run test

- cd server

- node server.js

4. Copy api.js в src/api  
 (e2e.test.js 🡪describe.only)   
5. Създаваме app.js в src

6. Слагаме в index.html   
<script src=”/src/app.js” type=”module”></script>

7. Слагаме в app.js

import { render } from ’../node\_modules/lit-html/lit-html.js’;   
import page from ‘../node\_modules/page/page.mjs’;

import { logout as apiLogout } from ’./api /data.js’;

import { homePage } from ’./views /home.js’;

import { loginPage } from ’./views /login.js’;

import { registerPage } from ’./views /register.js’;

const main = document.querySelector(‘main’);

document.getElementById(‘logoutBtn’).addEventListener(‘click’, logout);

page(‘/’, decorateContext, homePage);

page.start();

function decorateContext(ctx, next){

ctx.render = (content) => render(content, main);

ctx.setUserNav = setUserNav;

next();

}

function setUserNav() {

const email = sessionStorage.getItem(‘email);

if (email != null) {

document.querySelector(‘div.profile > span’).textContent = ‘Welcome, ${ email }’;

document.querySelector(‘.user’).style.display = ‘’;

document.querySelector(‘.guest).style.display = ‘none’;

} else {

document.querySelector(‘.user’).style.display = ‘none’;

document.querySelector(‘.guest).style.display = ‘’;

}

}

async function logout() {

await apiLogout();

seUserNav();

page.redirect(‘/’);

}

8.Създаваме папка views в src

9.Създаваме home.js  
import { html } from ’../../node\_modules/lit-html/lit-html.js’;

const homeTemplate = () => html`  
 Welcome Page от index.html

в href=”/login”  
 в href=”/register”  
`;

export async function homePage (ctx) {

ctx.render(homeTemplate());

}

10.В index.html променяме линковете на навигацията   
от href=”#” на   
href=”/”, href=”/catalog”, href=”/create”, href=”/login”, href=”/register”, href=”/profile”,

за logout 🡪 id=”logoutBtn” href=”javascript:void(0)”

11. Създаване на data.js в src/api

import \* as api from ‘./api.js’;

api.settings.host = ‘http://localhost:3030’;

export const login = api. login;

export const register = api. register;

export const logout = api. logout;

// Application-specific requests

export async function getMemes() {

return await api.get(host + ‘/data/memes?sortBy=\_createOn%20desc’);

}

export async function createMemes(meme) {

return await api.post(host + ‘/data/memes’, meme);

}

12. login.js като home.js

import { html } from ’../../node\_modules/lit-html/lit-html.js’;

const loginTemplate = (onSubmit) => html`  
 Login Page от index.html

на <form @submit=${onSubmit} id=”login-form”>  
`;  
export async function loginPage (ctx) {

ctx.render(homeTemplate(onSubmit));

}

async function onSubmit(event) {

event.preventDefault();

const formData = new FormData(event.target);  
 const email = formData.get(‘email’).trim();

const password = formData.get(‘password’) .trim();;

await login(email, password);

ctx.setUserNav();

ctx. page.redirect(‘/catalog’);

}

13. register.js като login.js

Валидация при onSubmit

if (!username || ! !email || !password || !repeatPass || !gender) {

return alert(‘All fields are required!’);

}

if (password != repeatPass) {

return alert(‘Passwords don\’t match!’);

}

await login(username, email, password, gender);

ctx.setUserNav();

ctx. page.redirect(‘/catalog’);