

Dokumentace klauzurní práce

[Odkaz](#)

Vypracoval: Hanuš Valenta

Obsah

1.	Úvod	3
2.	Co je v dokumentu uvedeno	3
3.	Popis použitých technologií.....	4
3.1	HTML	4
3.2	CSS	4
3.3	JavaScript	4
3.4	NodeJS	5
3.5	Three.js	5
4.	Použitý editor	6
4.1	VSCodium	6
4.2	Výhody VSCodium	6
5.	Popis adresářů	7
5.1	Portfolio	7
5.2	Assets.....	7
5.3	node_modules.....	7
6.	Popis souborů	8
6.1	Blank.html	8
6.2	Cursor.js	8
6.3	Experiences.html	8
6.4	GhostCursor.js.....	8
6.5	index.html.....	8
6.6	Index.js.....	8
6.7	Inspiration.html	8
6.8	Main.js	9
6.9	package.json a package-lock.json.....	11
6.10	Projects.html	11
6.11	Story.html	11
6.12	Typing.js.....	12
7.	Zdroje.....	14

1. Úvod

V dnešní době je považováno za nezbytné, aby vlastní portfolio bylo vlastněno každým, od umělce po vývojáře. Tato portfolio slouží k prezentaci technických schopností a také k seznámení s již existujícími pracemi daného člověka. U vývojářů je portfolio využíváno k ukázce jejich znalostí programování, práce s různými technologiemi a dovedností při řešení problémů. Tímto způsobem může být umožněno potenciálním zaměstnavatelům udělat si představu o jejich schopnostech a přístupu k práci.

2. Co je v dokumentu uvedeno

V dokumentu jsou uvedeny informace o využitých technologiích a jejich následném praktickém využití při stavbě portfolio. Podrobně je popsáno, jakým způsobem byly tyto technologie aplikovány v různých fázích vývoje a jakým přínosem byly pro celkovou funkčnost a design portfolio. Zároveň je věnována pozornost popisu jednotlivých adresářů a souborů, které byly vytvořeny v rámci projektu. Každý adresář a soubor je detailně popsán, včetně jejich specifických funkcí a způsobu, jakým přispívají k celkové struktuře a organizaci projektu. Díky těmto informacím je možné lépe porozumět nejen technickým aspektům portfolio, ale také jeho praktickému použití a údržbě.

3. Popis použitých technologií

3.1 HTML

HTML je zkratka pro Hypertext Markup Language a je základním jazykem pro tvorbu webových stránek. Používá se k definování struktury a obsahu stránek pomocí různých značek a elementů. [1]

3.2 CSS

CSS, zkratka z anglického názvu Cascading Style Sheets, je jazyk používaný pro stylování webových stránek. Jeho hlavním účelem je definovat vzhled a prezentaci obsahu HTML dokumentů. CSS umožňuje nastavovat různé vlastnosti elementů, jako je barva, velikost, poloha, písmo a mnoho dalšího. [2]

3.3 JavaScript

JavaScript je primárně známý jako skriptovací jazyk pro webové stránky, kde je používán k interaktivitě, dynamickému obsahu a manipulaci s HTML a CSS. S jeho pomocí je možné reagovat na uživatelské interakce, validovat formuláře, vytvářet animace, dynamicky měnit obsah stránek a mnoho dalšího. [3]

```
○ ○ ○  
  
1 main = document.getElementsByTagName("Header")[0];  
2  
3 if(!navigator.userAgentData.mobile) {  
4     document.addEventListener("mousemove", () => {  
5         main.style.boxShadow = (event.clientX - window.innerWidth / 2) / -15 + "px " +  
6         (event.clientY - window.innerHeight / 2) / -15 + "px 0 #fff";  
7     });  
8 }
```

1

¹ Ukázka javascript kódu z index.js

3.4 NodeJS

Node.js je runtime prostředí, které je využíváno pro vývoj serverových aplikací. Je postaveno na JavaScriptovém engine, který umožňuje běh JavaScriptu mimo webový prohlížeč.

Umožňuje práci s asynchronními operacemi díky událostmi řízené architektuře, čímž je dosaženo vysoké škálovatelnosti a efektivity při zpracování velkého množství současných požadavků. Node.js poskytuje různé moduly pro práci s http. [4]

3.5 Three.js

Three.js je open-source JavaScriptová knihovna, která umožňuje tvorbu a zobrazování 3D grafiky přímo ve webovém prohlížeči. Je postavena na WebGL, což je standardní webová API pro práci s 3D grafikou v prohlížeči, a umožňuje vývojářům vytvářet působivé 3D scény a interaktivní vizualizace bez nutnosti použití pluginů či externích programů. [5]

```
○ ○ ○

1 import * as THREE from 'three';
2
3 const scene = new THREE.Scene();
4 const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
5 const renderer = new THREE.WebGLRenderer();
6 renderer.setSize(window.innerWidth, window.innerHeight);
7 document.body.appendChild(renderer.domElement);
8
9 const geometry = new THREE.BoxGeometry();
10 const material = new THREE.MeshBasicMaterial({ color: 0xff0000 });
11 const cube = new THREE.Mesh(geometry, material);
12 scene.add(cube);
13
14 function animate() {
15     requestAnimationFrame(animate);
16     cube.rotation.x += 0.01;
17     cube.rotation.y += 0.01;
18     renderer.render(scene, camera);
19 }
20 animate();
```

² Ukázka základní node.js scény

4. Použitý editor

4.1 VSCodium

VSCodium je open-source distribuce Visual Studio Code, která je poskytována bez integrovaných služeb Microsoftu, jako je například Telemetry. To znamená, že VSCodium je úplně stejný jako Visual Studio Code, ale neobsahuje žádný kód, který by sledoval nebo sbíral informace o uživatelském chování. [6]

4.2 Výhody VSCodium

VSCodium nabízí širokou škálu funkcí pro vývojáře, včetně podpory pro mnoho programovacích jazyků, integrovaných nástrojů pro ladění a správu verzí a rozsáhlého ekosystému rozšíření. VSCodium je ideální volbou pro ty, kteří hledají výkonný a flexibilní textový editor, který respektuje jejich soukromí a bezpečnost.

5. Popis adresářů

5.1 Portfolio

Zde je celý projekt a jeho soubory, spolu s nezbytnými knihovnami.

5.2 Assets

V několika adresářích jsou umístěny veškeré potřebné grafiky, ikony, fonty a modely, které jsou nezbytné pro funkci webu.

(Soubory jsou k dispozici ke stažení na stránce "Files", využívané fonty jsou umístěny ve složce "Fonts", fotografie, ikony a favikony stránky najdete ve složce "Images", a 3D modely zobrazované na stránce jsou umístěny ve složce "Models".)

5.3 node_modules

Všechny knihovny používané pro Node.js spolu s jejich skripty jsou umístěny v této složce.

6. Popis souborů

6.1 Blank.html

Tento soubor není dostupný z hlavní stránky, ale je důležitý jako předloha pro budoucí vytváření podstránek, přičemž slouží jako jejich vzor.

6.2 Cursor.js

Zde je umístěn JavaScript pro alternativní kurzor, který je tvořen z několika kruhů, které se aktualizují.

6.3 Experiences.html

Popis předchozích zkušeností.

6.4 GhostCursor.js

Soubor obsahuje JavaScript kód, který vytváří alternativní kurzor. Tento kurzor generuje několik zpomalených kopií s latencí za účelem dosažení efektu sekání.

6.5 index.html

Hlavní stránka obsahuje krátké uvítání a mapu stránky pro pozdější navigaci.

6.6 Index.js

JavaScript v souboru index.html je určen k vytvoření pohybujícího se dropshadow na základě lokace kurzoru na stránce.

```
○○○  
  
1 main = document.getElementsByTagName("Header")[0];  
2  
3 if(!navigator.userAgentData.mobile) {  
4   document.addEventListener("mousemove", () => {  
5     main.style.boxShadow = (event.clientX - window.innerWidth / 2) / -15 + "px " +  
6     (event.clientY - window.innerHeight / 2) / -15 + "px 0 #fff";  
7   });  
7 }
```

3

6.7 Inspiration.html

Stránka obsahuje inspiraci, která byla využita při designování webu, a zároveň zahrnuje poděkování autorům použitých knihoven.

³ JavaScript hledá element s ID "header" a poté aktualizuje dropshadow.

6.8 Main.js

Soubor pro knihovnu three.js vytváří 3D scénu v pozadí obsahující náhodně generované hvězdy, které jsou náhodně rozmístěny daleko od kamery, a několik objektů, které jsou prezentovány vedle textu. Vše je konvertováno do ASCII textu pro estetické účely.

○ ○ ○

```
1 const scene = new THREE.Scene();
2 const camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);
3
4 camera.position.z = 5;
5
6 const canvas = document.getElementById("display-canvas");
7 const renderer = new THREE.WebGLRenderer({ canvas });
8 const effect = new AsciiEffect(renderer, ' .:-+*=%@#', { invert: true });
```

4

○ ○ ○

```
1 const heartShape = new THREE.Shape()
2   .moveTo(5, 5)
3   .bezierCurveTo(5, 5, 4, 0, 0, 0)
4   .bezierCurveTo(-6, 0, -6, 7, -6, 7)
5   .bezierCurveTo(-6, 11, -3, 15.4, 5, 19)
6   .bezierCurveTo(12, 15.4, 16, 11, 16, 7)
7   .bezierCurveTo(16, 7, 16, 0, 10, 0)
8   .bezierCurveTo(7, 0, 5, 5, 5, 5);
```

5

⁴ Konstanty pro scénu, kameru a efekt ASCII jsou vytvořeny v JavaScriptu.

⁵ Na hlavní stránce je zobrazené srdce vytvořené pomocí bodů, které jsou propojeny do tvaru srdce.

○ ○ ○

```
1 function createStar() {
2   const geometry = new THREE.SphereGeometry(0.1, 24, 24);
3   const material = new THREE.MeshBasicMaterial({ color: 0xffffff });
4   const star = new THREE.Mesh(geometry, material);
5   const x = Math.random() * 200 - 100;
6   const y = Math.random() * 200 - 100;
7   const z = -Math.random() * 500 - 50;
8   star.position.set(x, y, z);
9   scene.add(star);
10 }
```

6

○ ○ ○

```
1 function moveCamera() {
2   const t = document.body.getBoundingClientRect().top;
3   camera.position.y = t * 0.003;
4   dirLight.position.copy(camera.position);
5 }
6 document.body.onscroll = moveCamera;
7
8 function rotateObjects() {
9   heartMesh.rotation.x += 0.01;
10  heartMesh.rotation.y += 0.01;
11  verticalMesh.rotation.x += 0.01;
12  verticalMesh.rotation.y += 0.01;
13  cubeMesh.rotation.x += 0.01;
14  cubeMesh.rotation.y += 0.01;
15 }
16
17 function animate() {
18   requestAnimationFrame(animate);
19   rotateObjects();
20   effect.render(scene, camera);
21 }
```

7

⁶ Funkce pro vytvoření hvězdy, náhodné její rozmístění a vložení do scény jsou definovány v JavaScriptu.

⁷ Funkce pro pohyb kamerou synchronizovaný s posouváním stránky a pravidelně volaná funkce pro rotaci objektů ve scéně a generování nových snímků.

6.9 package.json a package-lock.json

Oba soubory, package.json a package-lock.json, jsou důležité pro správu závislostí Node.js aplikace. package.json obsahuje metadata a seznam knihoven aplikace, zatímco package-lock.json udržováním přesného stavu těchto knihoven pomáhá zajistit konzistenci mezi různými instalacemi aplikace. [7]

6.10 Projects.html

Představení předchozích projektů a vytvořených prací.

6.11 Story.html

Informace o autorovi s jeho životopisem a událostmi v něm.Style.css

Soubor obsahující veškeré informace o vzhledu stránky, včetně animací a responzivity závislosti na rozlišení obrazovky

```
○ ○ ○  
  
1 #cursor {  
2   display: inline-block;  
3   background-color: white;  
4   color: white;  
5   animation: blink 1s infinite;  
6 }  
7  
8 @keyframes blink {  
9   0%, 100% {  
10     opacity: 1;  
11   }  
12   50% {  
13     opacity: 0;  
14   }  
15 }
```

⁸ Soubor s veškerými informacemi ohledně vzhledu stránky animacemi a responzivitě v závislosti na ohledu podle rozlišení.

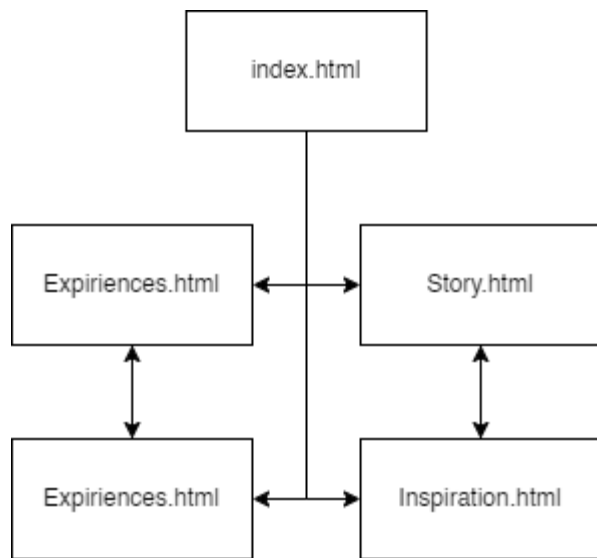
6.12 Typing.js

Soubor postupně doplňující charaktery do jména pro vytvoření efektu psaní.

```
○ ○ ○  
  
1 const text = "Valenta Hanuš";  
2   const typingSpeed = 100;  
3  
4   let index = 0;  
5  
6   function typeText() {  
7     if (index < text.length) {  
8       document.getElementById('typed-text').innerHTML += text.charAt(index);  
9       index++;  
10      setTimeout(typeText, typingSpeed);  
11    }  
12  }  
13  
14  typeText();
```

⁹ JavaScript, který postupně doplňuje jeden znak každých 100 milisekund.

7. Schéma webu



8. Zdroje

[1] WIKIEPDIA. HTML. Online. 2003, 29. května 2024. Dostupné z: <https://en.wikipedia.org/wiki/HTML>. [cit. 2024-06-01].

[2] WIKIEPDIA. CSS. Online. 2001, 6. května 2024. Dostupné z: <https://en.wikipedia.org/wiki/CSS>. [cit. 2024-06-01].

[3] WIKIEPDIA. JavaScript. Online. 2001, 18. dubna 2024. Dostupné z: <https://en.wikipedia.org/wiki/JavaScript>. [cit. 2024-06-01].

[4] WIKIEPDIA. Node.js. Online. 2009, 2. června 2024. Dostupné z: <https://en.wikipedia.org/wiki/Node.js>. [cit. 2024-06-01].

[5] WIKIEPDIA. Three.js. Online. 2012, 11. ledna 2024. Dostupné z: <https://en.wikipedia.org/wiki/Three.js>. [cit. 2024-06-01].

[6] TIM O'BRIEN. VSCodium. Online. 2019, 8. ledna 2024. Dostupné z: <https://vscode.com/#why>. [cit. 2024-06-01].

[7] EDWARD THOMSON. Package.json. Online. 2019, 22. září 2020. Dostupné z: <https://docs.npmjs.com/cli/v6/configuring-npm/package-json>. [cit. 2024-06-01].