

WEBOVÉ STRÁNKY DUKLY BRNO

DOKUMENTACE MATURITNÍHO PROJEKTU

Obsah

WEBOVÉ STRÁNKY DUKLY BRNO	3
Úvod.....	3
Použité technologie	4
Jazyky.....	4
Software.....	5
Ostatní služby	6
Front-End.....	7
Template	7
Aktuální struktura	8
CSS.....	9
Back-end.....	10
JavaScript	10
TypeScript	12
React.....	13
API	15
Postup spuštění.....	16
Závěr	18
Odkazy.....	19

Úvod

Tento document obsahuje informace o tvorbě, struktuře a funkci základu pro budoucí podobu webových stránek armádního sportovního oddílu Dukly Brno.

Pro toto téma jsem se rozhodl, neboť jakožto člen Dukly jsem si vědom špatného stavu zastaralých stránek, které dnes Dukla používá, a tak s tím je načasе něco udělat.

Použité technologie

Jazyky

HTML – HTML (HyperText Markup Language) je značkovací jazyk používaný pro vytváření webových stránek. Pomocí HTML kódu se definují strukturální prvky stránky, jako jsou nadpisy, odstavce, odkazy, obrázky a další prvky. Každý prvek je zapsán pomocí značek, které jsou uzavřeny v hranatých závorkách, například `<p>pro odstavec</p>`. HTML kód se interpretuje webovým prohlížečem a na základě něj se vytvoří vizuální podoba webové stránky.

CSS – CSS (Cascading Style Sheets) je jazyk používaný pro popis vzhledu webových stránek, oddělený od samotného obsahu stránky v HTML. CSS určuje, jaké barvy, fonty, velikosti, pozice a další vlastnosti mají jednotlivé prvky HTML na stránce. Soubor CSS může být připojen k jedné nebo více stránkám, což umožňuje aplikovat stejný styl na všechny stránky webového projektu. CSS je velmi užitečný pro oddělení obsahu a vzhledu, což umožňuje změny vzhledu stránek s minimálním vlivem na obsah.

JavaScript – JavaScript je skriptovací programovací jazyk, který se používá pro vytváření interaktivních webových stránek. Skripty napsané v JavaScriptu se provádějí na straně klienta (v prohlížeči) a mohou reagovat na uživatelské akce, například kliknutí na tlačítko nebo odeslání formuláře. JavaScript umožňuje dynamicky měnit obsah, styl a chování webových stránek a může být použit pro různé účely, jako jsou validace formulářů, animace, interaktivní prvky, zobrazení a skrývání obsahu a mnoho dalšího. JavaScript se také často používá pro tvorbu webových aplikací a herních prvků.

TypeScript – TypeScript je programovací jazyk vyvinutý společností Microsoft, který rozšiřuje funkce jazyka JavaScript. TypeScript umožňuje statickou typizaci proměnných, což znamená, že při tvorbě kódu musí být proměnným přiřazeno určité datové typy, což zlepšuje srozumitelnost kódu a snižuje počet chyb. TypeScript také poskytuje možnost rozšiřování kódu pomocí objektově orientovaného programování, dědičnosti a rozhraní. Kód v TypeScriptu je kompilován na kód JavaScriptu, který může být spuštěn v prohlížeči. TypeScript je často používán pro vývoj středně velkých a velkých aplikací, ať už webových nebo desktopových, a pro tvorbu komplexních aplikací ve frameworku Angular.

Software

VS Code – VS Code (Visual Studio Code) je multiplatformní integrované vývojové prostředí (IDE) vyvinuté společností Microsoft. VS Code nabízí podporu pro mnoho programovacích jazyků a technologií a poskytuje uživatelské rozhraní, které umožňuje vývojářům psát, testovat a ladit kód s vysokou efektivitou. Mezi jeho funkce patří například syntaktické zvýrazňování, automatické dokončování kódu, integrace s verzovacími systémy, jako je Git, a mnoho dalších rozšíření a pluginů, které lze snadno nainstalovat a používat. VS Code je oblíbeným nástrojem mezi vývojáři díky své rychlosti, jednoduchosti použití a podpoře široké škály jazyků a technologií.

React.js – React.js (nebo jen React) je open-source knihovna pro tvorbu uživatelského rozhraní (UI) webových aplikací. Byla vyvinuta společností Facebook a je nyní udržována společností Facebook a komunitou. React umožňuje vytváření komponent, které mohou být znovupoužity v různých částech aplikace a které se chovají podle určitých podmínek. React používá vlastní syntaxi nazývanou JSX (JavaScript XML), která umožňuje vkládat HTML do kódu JavaScriptu. React také umožňuje snadné propojení se stavem aplikace, což umožňuje snadné vytváření interaktivních aplikací. React je často používán pro vývoj moderních webových aplikací, jako jsou sociální sítě, e-shopy, zpravodajské portály a mnoho dalšího.

.NET CORE – .NET Core je open-source platforma pro vývoj aplikací, kterou vyvinula společnost Microsoft. Tato platforma podporuje vývoj aplikací pro různé operační systémy, včetně Windows, Linux a macOS. .NET Core poskytuje běhové prostředí, knihovny a nástroje pro tvorbu webových, desktopových a cloudových aplikací. .NET Core umožňuje vývojářům psát kód v různých programovacích jazycích, jako jsou C#, F# nebo Visual Basic, a poskytuje mnoho funkcí a nástrojů pro efektivní vývoj, testování a nasazování aplikací. .NET Core je často používán pro tvorbu moderních webových aplikací, mikroslužeb a cloudových řešení, a je oblíbeným nástrojem mezi vývojáři díky své flexibilitě, výkonu a široké podpoře komunity.

Ostatní služby

AzureDevops – Azure DevOps je cloudová platforma pro správu a sledování softwarových projektů. Tato platforma poskytuje nástroje pro řízení celého softwarového vývojového cyklu, včetně správy kódu, správy projektů, plánování a sledování úloh, testování a nasazování aplikací. Azure DevOps je navržen tak, aby byl jednoduchý a přehledný pro týmy vývojářů a umožňuje snadnou spolupráci a koordinaci mezi členy týmu. Mezi jeho hlavní funkce patří například integrace s verzovacími systémy, jako je Git, sledování problémů a chyb, automatizace procesů, kontinuální integrace a nasazování (CI/CD), a mnoho dalších nástrojů pro efektivní řízení a koordinaci projektů. Azure DevOps je oblíbeným nástrojem mezi vývojáři díky své široké škále funkcí, integraci s dalšími nástroji a jeho cloudovému charakteru, který umožňuje přístup k projektům odkudkoliv a kdykoliv.

Microsoft Azure – Microsoft Azure je cloudová platforma, kterou vyvinula a spravuje společnost Microsoft. Azure poskytuje širokou škálu služeb, které umožňují uživatelům vytvářet, testovat, nasazovat a spravovat aplikace v cloudu. Tyto služby zahrnují virtuální stroje, úložiště dat, nástroje pro analýzu dat, umělou inteligenci, internet věcí (IoT), kontejnerové aplikace a mnoho dalších. Azure také umožňuje uživatelům snadnou integraci s dalšími technologiemi a nástroji, jako jsou Microsoft Office 365, Visual Studio a GitHub. Azure má velmi širokou podporu a komunitu, a umožňuje uživatelům flexibilitu, výkon a škálovatelnost potřebnou pro tvorbu moderních aplikací. Azure je jedním z největších cloudových poskytovatelů na světě a je oblíbeným nástrojem mezi vývojáři a firmami všech velikostí, kteří chtějí využít výhod cloudových technologií pro své aplikace a služby.

WrapBootstrap – WrapBootstrap.com je online tržiště, které poskytuje předplatitelům širokou škálu hotových šablon a témat pro Bootstrap framework. Na této platformě mohou uživatelé najít a zakoupit hotové šablony a témata pro své webové projekty, což jim může ušetřit čas a náklady na vývoj vlastních řešení. Každá šablona nebo téma na WrapBootstrap.com je tvořena profesionálními designéry a vývojáři, a poskytuje uživatelům moderní a funkční uživatelské rozhraní, které je možné snadno přizpůsobit. WrapBootstrap.com umožňuje uživatelům také prodávat své vlastní hotové šablony a témata ostatním uživatelům platformy a tím si vydělávat peníze.

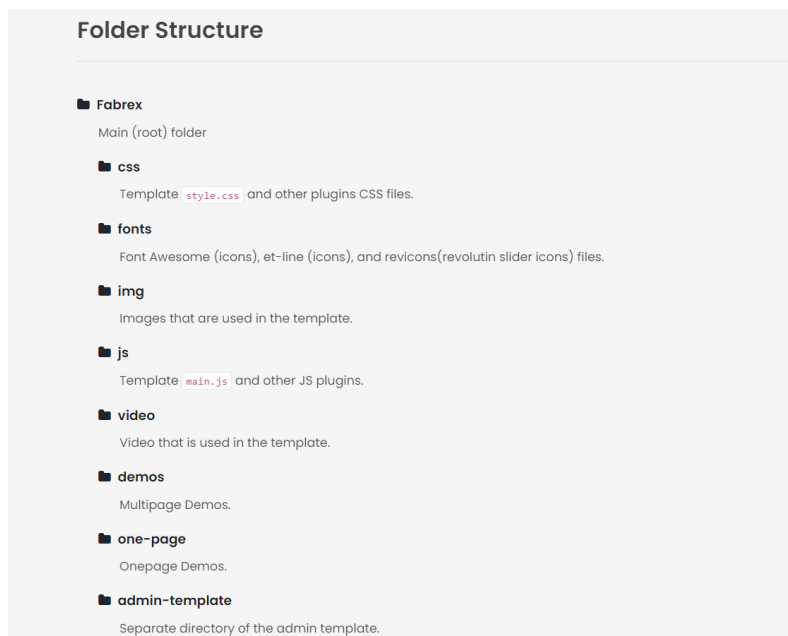
GitHub – GitHub je webová platforma pro správu verzí a sdílení kódu, která umožňuje vývojářům spolupracovat na projektech a sledovat historii změn v kódu. Uživatelé mohou vytvářet a ukládat projekty (tzv. „repositories“), přidávat do nich soubory a provádět změny v kódu. GitHub také poskytuje nástroje pro správu problémů a žádostí o spolupráci, umožňuje automatizaci procesů pomocí služeb a integruje se s mnoha dalšími nástroji používanými při vývoji softwaru. Je často využíván otevřenými projekty a komunitami, které se snaží vytvářet software v duchu spolupráce a sdílení.

Node.js – Node.js je open-source prostředí pro běh JavaScriptu mimo webový prohlížeč. Umožňuje vývojářům psát serverovou stranu aplikací v jazyce JavaScript, což zjednodušuje vývoj webových aplikací a usnadňuje sdílení kódu mezi frontendem a backendem. Node.js používá asynchronní model událostí a nepřerušovaný běh, což umožňuje zpracovávat velké množství požadavků současně a snižuje reakční dobu aplikace. Node.js má velkou komunitu a mnoho rozšíření (tzv. „packages“) v balíčkovacím systému npm, což zvyšuje produktivitu a usnadňuje vývoj aplikací.

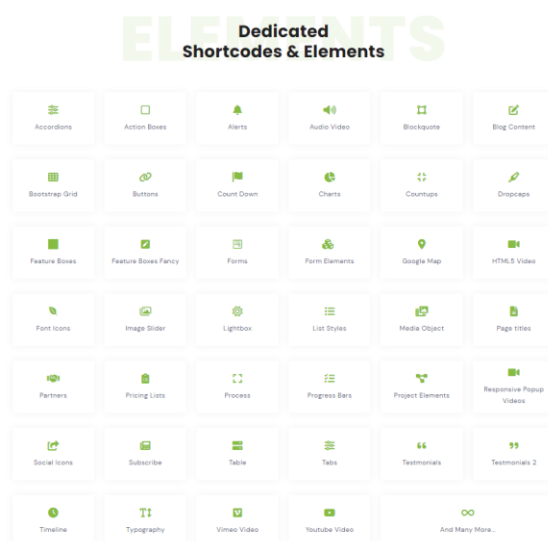
Front-End

Template

Projekt je založen na FABREX-Template od autora ChitrakootWeb. Tato šablona obsahuje velké množství předpřipravených stránek a jejich komponent.



Také ale umožňuje jednoduché upravení, přispůsobení i tvoření s předpřipravenými pluginy a styly.



Template byl zakoupen na stránce wrapbootstrap.com (odkaz přímo na template v úseku „Odkazy”).

Aktuální struktura

Stránky aktuálně obsahují 6 dostupný stránek a jednu 404 stránku hlásící chybu pro případ potřeby.

1. Home (index.html)

Domovská stránka obsahující pouze jednoduchý úvod. Co je Dukla Brno a čím se zabývá.

2. News (news.html)

Stránka v aktuální podobě nemá žádný významný obsah, je ale připravena pro využití. Novinky typu výsledků z posledních závodů, konání soustředění a tak dále zde budou vyvěšovány.

3. History (history.html)

Zde se lidé mohou dozvědět o historii Dukly Brno, minulých výsledků a jejím vývoji v minulých letech.

4. Track Cycling (track.html)

Na této stránce se mohou lidé dozvědět více o dráhové cyklistice, a především jejím sprinterském odvětví.

5. Team (team.html)

Samotný tým, jak závodníci, tak lidé okolo, jsou náplní a zde se dozvíte kdo jsou.

6. Contact (contact.html)

Výpis kontaktních informací.

CSS

Soubory v CSS jazyce slouží k určení vzhledových vlastností jako barva pozadí, font, velikost a barva písma, zarovnání obsahu, velikost a pozici obrázků a mnoho dalšího.

Například sekce s názvem „Project grid“ v použitém souboru „styles-5.css“ umožňuje jednoduché rozprostírání textu po stránce.

.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1	.col-md-1
.col-md-2		.col-md-2		.col-md-2		.col-md-2		.col-md-2		.col-md-2		
.col-md-3			.col-md-3			.col-md-3			.col-md-3			
.col-md-4				.col-md-4				.col-md-4				
.col-md-5					.col-md-5					.col-md-2		
.col-md-6						.col-md-6						
.col-md-1	.col-md-11											
.col-md-2		.col-md-10										
.col-md-3			.col-md-9									
.col-md-4				.col-md-8								
.col-md-5					.col-md-7							

Struktura CSS souborů vypadá následovně. Většina souborů se týká pouze jednotlivých component se, kterými sdílí název. Za vyzdvižení stojí soubor „styles-5.css“, který udává podobu většiny složek stránek.

```
app \my-app
├── node_modules
├── public
├── css
│   ├── plugins
│   │   ├── animate.css
│   │   ├── bootstrap.min.css
│   │   ├── datetimepicker.css
│   │   ├── default.css
│   │   ├── fontawesome-all.min.css
│   │   ├── lightgallery.css
│   │   ├── magnific-popup.css
│   │   ├── nav-menu.css
│   │   ├── owl.carousel.css
│   │   ├── owl.theme.default.css
│   │   ├── prism.css
│   │   ├── themify-icons.css
│   │   └── xzoom.css
│   ├── quform
│   │   └── css
│   │       ├── base.css
│   │       └── js
│   └── rev_slider
│       ├── layers.css
│       ├── navigation.css
│       ├── settings.css
│       ├── typewriter.css
│       ├── custom.css
│       ├── index.css
│       ├── plugins.css
│       ├── styles-5.css
│       └── styles-5.css.map
```

Back-end

JavaScript

JavaScript je nezbytnou součástí, která určuje funkci a chování stránky.

Například následující úsek mění ikony v horní části stránek, aby se přizpůsobily měnící se barvě horního navigačního baneru při scrolování.

```
/*-----  
02. Sticky Header  
-----*/  
  
$window.on('scroll', function() {  
    var scroll = $window.scrollTop();  
    var logoinner = $(".navbar-brand img");  
    var logodefauult = $(".navbar-brand.logodefauult img");  
    var logowhite = $(".navbar-brand.logowhite img");  
    var logo2 = $(".navbar-brand.logo2 img");  
    var logo4 = $(".navbar-brand.logo4 img");  
    var logo5 = $(".navbar-brand.logo5 img");  
    var logo6 = $(".navbar-brand.logo6 img");  
    var logo7 = $(".navbar-brand.logo7 img");  
    var logo8 = $(".navbar-brand.logo8 img");  
    var logoD = $(".navbar-dukla_1 img");  
    if (scroll <= 50) {  
        $(".header").removeClass("scrollHeader").addClass("fixedHeader");  
        logoinner.attr('src', 'img/logos/dukla_logo_1.png');  
        logodefauult.attr('src', 'img/logos/dukla_logo_1.png');  
        logowhite.attr('src', 'img/logos/logo-white.png');  
        logo2.attr('src', 'img/logos/logo-2-light.png');  
        logo4.attr('src', 'img/logos/logo-4.png');  
        logo5.attr('src', 'img/logos/logo-5-light.png');  
        logo6.attr('src', 'img/logos/logo-6.png');  
        logo7.attr('src', 'img/logos/logo-7.png');  
        logo8.attr('src', 'img/logos/logo-8.png');  
        logoD.attr('src', 'img/logos/dukla_logo_1.png');  
    }  
    else {  
        $(".header").removeClass("fixedHeader").addClass("scrollHeader");  
        logoinner.attr('src', 'img/logos/dukla_logo_1.png');  
        logodefauult.attr('src', 'img/logos/dukla_logo_1.png');  
        logowhite.attr('src', 'img/logos/logo-white.png');  
        logo2.attr('src', 'img/logos/logo-2-dark.png');  
        logo4.attr('src', 'img/logos/logo-4.png');  
        logo5.attr('src', 'img/logos/logo-5-dark.png');  
        logo6.attr('src', 'img/logos/logo-6.png');  
        logo7.attr('src', 'img/logos/logo-7.png');  
        logo8.attr('src', 'img/logos/logo-8.png');  
        logoD.attr('src', 'img/logos/dukla_logo_1.png');  
    }  
});
```

Popis technologie (ze zdrojů na internetu):

JavaScript je vysokoúrovňový, dynamicky typovaný programovací jazyk, který je široce používán pro vývoj webových aplikací. Byl vytvořen v roce 1995 Brendanem Eichem, a nyní je udržován konsorciem ECMA International pod názvem ECMAScript. JavaScript je skriptovací jazyk, který běží přímo v internetovém prohlížeči a umožňuje vývojářům vytvářet interaktivní a dynamické webové stránky.

Hlavní rysy JavaScriptu:

- Dynamický a slabě typovaný: JavaScript umožňuje proměnným měnit typ během provádění kódu, což zvyšuje jeho flexibilitu.
- Objektově orientovaný: JavaScript používá prototypovou dědičnost pro práci s objekty a jejich vlastnostmi.
- Event-driven: JavaScript pracuje s událostmi, které umožňují reagovat na akce uživatele, jako je kliknutí nebo stisknutí klávesy.
- Multi-paradigmaticý: JavaScript podporuje různé programovací styly, jako je imperativní, deklarativní, objektově orientovaný a funkcionální programování.
- Asynchronní a neblokující I/O: JavaScript podporuje asynchronní operace, které umožňují zpracování úkolů na pozadí, aniž by blokovaly hlavní vlákno.

JavaScript běží v prostředí prohlížeče a interaguje s HTML a CSS, což jsou další dva pilíře webového vývoje. JavaScript lze také použít na serverové straně prostřednictvím platformy Node.js, která umožňuje spouštění JavaScriptu mimo prohlížeč.

Zde jsou některé důležité zdroje pro JavaScript:

- Oficiální stránky ECMAScript: <https://www.ecma-international.org/publications/standards/Ecma-262.htm>
Zde se nachází oficiální specifikaci ECMAScript, která popisuje jazyk JavaScript.
- MDN Web Docs: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
MDN Web Docs poskytuje rozsáhlou dokumentaci a návody pro JavaScript, HTML, CSS a další webové technologie.
- Stack Overflow: <https://stackoverflow.com/questions/tagged/javascript>
Stack Overflow je komunitní web pro otázky a odpovědi, kde jsou k dispozici řešení problémů a diskuze týkající se JavaScriptu.
- GitHub: <https://github.com/trending/javascript>
GitHub je platforma pro vývojáře, kde se nachází řada JavaScript projektů a knihoven.

TypeScript

Při převodu statických stránek na react aplikaci jsem využil možnost kódovat v TypeScriptu, proto přidávám popis této „varianty“ JavaScriptu.

Popis technologie (ze zdrojů na internetu):

TypeScript je open-source jazyk, který byl vyvinut a je udržován společností Microsoft. TypeScript je nadstavba nad JavaScriptem, která přidává statické typy a další pokročilé funkce, které zlepšují spolehlivost, čitelnost a efektivitu vývoje kódu. TypeScript si zachovává veškerou funkcionalitu JavaScriptu, což umožňuje snadný přechod mezi oběma jazyky.

Několik klíčových vlastností TypeScriptu zahrnuje:

- **Statické typování:** TypeScript umožňuje definovat statické typy pro proměnné, parametry, návratové hodnoty funkcí atd. Statické typování zlepšuje kontrolu kódu během vývoje, protože chyby typů lze odhalit dříve, ještě před spuštěním kódu.
- **Třídy a rozhraní:** TypeScript rozšiřuje JavaScript o třídy a rozhraní, což umožňuje použít objektově orientované programování (OOP) a zlepšit znovupoužitelnost, abstrakci a čitelnost kódu.
- **Generics:** TypeScript podporuje generické typy, které umožňují vytvářet obecné funkce nebo třídy, které mohou pracovat s různými typy dat. Generics zvyšují flexibilitu kódu a zlepšují jeho znovupoužitelnost.
- **Dekorátory:** TypeScript umožňuje použití dekorátorů pro přidávání metadat nebo měnění chování tříd, metod, vlastností nebo parametrů. Dekorátory jsou užitečné pro implementaci funkcí, jako jsou závislosti na injekci nebo pro přidávání chování během běhu.
- **Transpilace¹ do JavaScriptu:** TypeScript je transpilován do čistého JavaScriptu, což znamená, že výsledný kód může být spuštěn v jakémkoli prostředí, které podporuje JavaScript. Transpilace zajišťuje kompatibilitu se staršími prohlížeči a prostředími.

TypeScript je vhodný pro projekty všech velikostí a je obzvláště užitečný pro velké a komplexní projekty, kde zlepšení kontroly kódu, abstrakce a znovupoužitelnosti mají významný dopad na kvalitu a efektivitu vývoje. Díky integraci se stávajícími nástroji a knihovnami je TypeScript široce používán ve vývoji webových aplikací.

Oficiální stránku projektu jsou: <https://www.typescriptlang.org/>.

Tato stránka nabízí mnoho informací o jazyce TypeScript, včetně dokumentace, návodů, zdrojového kódu a dalších zdrojů pro vývojáře.

¹ Transpilace – proces převodu kódu napsaného v jednom programovacím jazyce na kód v jiném programovacím jazyce (TypeScript je převáděn do JavaScriptu)

React

React je software, který umožňuje rendering obsahu stránek pomocí komponent, které tvoří na základě obsahu JSON souborů, které tak slouží jako náhrada pro jinak potřebnou databázi.

Popis technologie (ze zdrojů na internetu):

React je otevřená knihovna pro vývoj uživatelských rozhraní, kterou vytvořil tým firmy Meta. Byl navržen s cílem usnadnit vytváření složitých, responzivních a efektivních uživatelských rozhraní (UI) v JavaScriptu. Jedná se o jednu z nejpopulárnějších knihoven pro vývoj webových aplikací.

Hlavní rysy Reactu:

- Deklarativní přístup: React usnadňuje vývoj UI tím, že umožňuje vývojářům popsat, jak by měla aplikace vypadat, aniž by museli řešit, jak toho dosáhnout. Tento přístup zlepšuje čitelnost kódu a zjednodušuje jeho údržbu.
- Komponentový model: React zavádí koncept komponent, které jsou stavebními bloky aplikace. Komponenty jsou znovupoužitelné, což usnadňuje rychlý vývoj a údržbu kódu.
- Virtuální DOM²² (VDOM): React používá virtuální reprezentaci DOM, která zajišťuje efektivní aktualizaci skutečného DOM. VDOM umožňuje Reactu zjistit, jaké změny je třeba provést, a poté aktualizovat pouze relevantní části DOM, což zlepšuje výkon aplikace.
- Unidirekcionální tok dat: React prosazuje unidirekcionální tok dat pomocí konceptu stavů (*state*) a vlastností (*props*). Tento přístup zjednodušuje logiku aplikace a zlepšuje její předvídatelnost.
- Podpora pro server-side rendering (SSR): React umožňuje vykreslovat aplikace na serveru, což zlepšuje dobu načítání stránky a podporuje SEO.
- Společenství a ekosystém: React má silné společenství vývojářů a širokou škálu nástrojů, knihoven a modulů, které usnadňují vývoj a nasazení aplikací.
- Kompatibilita s dalšími technologiemi: React lze snadno integrovat s dalšími populárními technologiemi, jako jsou *Redux* (pro správu stavu aplikace) nebo *React Router* (pro navigaci mezi stránkami).

²² DOM (Document Object Model) – programovací rozhraní, které reprezentuje strukturu a obsah dokumentu (např. HTML nebo XML) jako stromovou strukturu objektů. Tím umožňuje programátorům přístup, manipulaci a modifikaci jednotlivých prvků a jejich vlastností v dokumentu.

Při použití Reactu vývojáři často vytvářejí složené komponenty, které zahrnují logiku, stav a vlastnosti. Tyto komponenty mohou být funkční nebo třídní, přičemž funkční komponenty jsou často preferovány

Některé z nejdůležitějších zdrojů pro React na webu:

- Oficiální webové stránky Reactu: <https://reactjs.org/>
Obsahují oficiální dokumentaci, návody a další zdroje související s Reactem.
- GitHub repozitář Reactu: <https://github.com/facebook/react>
Obsahuje zdrojový kód Reactu, chybové hlášení a příspěvy od komunity.
- React na Stack Overflow: <https://stackoverflow.com/questions/tagged/reactjs>
Stack Overflow je komunitní web pro otázky a odpovědi, kde je k dispozici mnoho diskusí a řešení problémů souvisejících s Reactem.
- React na Reddit: <https://www.reddit.com/r/reactjs/>
Uvedený odkaz je subreddit pro React, kde můžete najít nejnovější zprávy, články a diskuze týkající se Reactu.
- React na YouTube: https://www.youtube.com/results?search_query=reactjs
YouTube je skvělý zdroj pro návody, kurzy a další video obsah týkající se Reactu.
- Egghead.io: <https://egghead.io/technologies/react>
Egghead.io nabízí řadu krátkých, konkrétních lekcí o Reactu od zkušených vývojářů.
- React Training: <https://reacttraining.com/>
React Training poskytuje školení a workshopy, které vám pomohou získat praktické dovednosti v Reactu.
- Create React App: <https://github.com/facebook/create-react-app>
Create React App je oficiálně podporovaný způsob, jak vytvořit nový projekt Reactu s minimálním nastavením.
- React Router: <https://reactrouter.com/>
React Router je populární knihovna pro navigaci a správu URL v React aplikacích.
- Redux: <https://redux.js.org/>
Redux je populární knihovna pro správu stavu aplikací, která se často používá společně s Reactem

API

Zde je nutno zmínit že i když je API důležitou součástí projektu, nejedná se o čistě moji práci.

API jsem si nechal vytvořit „na zakázku“. Jde o tzv. Minimal API na verzi .NET 7.

Na příkladu rozhraní pro závodníky (riders) můžeme popsat způsob jeho fungování takto:

API endpointy pro závodníky:

- GET /riders: Vrátí seznam všech závodníků.
- GET /riders/{id:int}: Vrátí závodníka podle ID.
- POST /riders: Vytvoří nového závodníka z těla požadavku.
- PUT /riders/{id:int}: Aktualizuje závodníka podle ID s novými daty z těla požadavku.
- DELETE /riders/{id:int}: Smaže závodníka podle ID.

Popis technologie (ze zdrojů na internetu):

Minimal API je funkcí představenou v .NET 6, která umožňuje vytvářet webové API s menším množstvím kódu a snadnější konfigurací než klasické ASP.NET Core Web API. Minimal API se zaměřuje na zjednodušení a odstranění *boilerplate* kódu, což zlepšuje čitelnost a zrychluje vývoj.

Několik klíčových vlastností Minimal API v .NET 6 zahrnuje:

- Zjednodušená konfigurace: Minimal API používá nový přístup k nastavení a konfiguraci aplikace pomocí *WebApplicationBuilder* třídy, což umožňuje snadnou konfiguraci služeb, middleware a dalších nastavení.
- Přímé mapování endpointů: V Minimal API lze přímo mapovat HTTP metody (GET, POST, PUT, DELETE atd.) na metody C# pomocí MapXXX metod (např. MapGet, MapPost, MapPut, MapDelete). To eliminuje potřebu vytvářet samostatné kontrolery, což zjednodušuje kód a zlepšuje čitelnost.
- Lambda funkce pro zpracování požadavků: Minimal API umožňuje použití lambda funkcí pro zpracování HTTP požadavků, což vede ke kratšímu a čistšímu kódu.
- Implicitní model binding: Minimal API automaticky zpracovává parametry z URL, dotazu, hlaviček nebo těla požadavku a zjednodušuje tak práci s daty přicházejícími v požadavcích.
- Vestavěná podpora pro OpenAPI: Minimal API poskytuje vestavěnou podporu pro generování OpenAPI specifikace, která umožňuje snadnou integraci s nástroji, jako je Swagger, pro dokumentaci API a testování.

Minimal API je vhodné pro menší a středně velké projekty, kde je důraz na rychlost vývoje a jednoduchost. Pro větší projekty s komplexnějšími požadavky na architekturu může být vhodnější použít klasické ASP.NET Core Web API s kontrolery a více vrstvami abstrakce.

Postup spuštění

Začínáme s Create React App

Tento projekt byl vytvořen pomocí [<https://github.com/facebook/create-react-app>].

Dostupné skripty

V adresáři projektu můžete spustit následující příkazy:

`npm start`

Spustí aplikaci v režimu vývoje.\

Otevřete [<http://localhost:3000>], abyste ji viděli ve webovém prohlížeči.

Stránka se automaticky aktualizuje při provedení úprav.\

V konzoli také uvidíte případné chyby kódu.

`npm test`

Spustí testovací nástroj v interaktivním sledovacím režimu.\

Podívejte se na sekci [<https://facebook.github.io/create-react-app/docs/running-tests>] pro více informací.

`npm run build`

Vytvoří aplikaci pro produkční nasazení do složky `build`.

Správně seskupí React v produkčním režimu a optimalizuje sestavení pro nejlepší výkon.

Výsledný soubor je zmenšený a názvy souborů obsahují *hash*.

Vaše aplikace je připravena k nasazení!

Podívejte se na sekci [<https://facebook.github.io/create-react-app/docs/deployment>] pro více informací.

`npm run eject`

Poznámka: Toto je jednosměrná operace. Jakmile provedete „eject“, nemůžete se vrátit zpět!

Pokud nejste spokojeni s nástrojem pro sestavení a výběrem konfigurace, můžete kdykoli provést „eject“. Tento příkaz odstraní jedinou závislost na sestavení z vašeho projektu.

Místo toho zkopíruje všechny konfigurační soubory a tranzitivní závislosti (webpack, Babel, ESLint atd.) přímo do vašeho projektu, takže s nimi máte plnou kontrolu. Všechny příkazy kromě „eject“ budou stále fungovat, ale budou odkazovat na zkopírované skripty, abyste je mohli upravovat. V této fázi jste na vlastní pěst.

Nemusíte nikdy použít „eject“. Kurátovaná sada funkcí je vhodná pro malé a střední nasazení a nemusíte se cítit povinni tuto funkci používat. Rozumíme však tomu, že tento nástroj by nebyl užitečný, kdybyste jej nemohli přizpůsobit, až budete připraveni.

Závěr

V aktuálním stavu tento základ obsahuje strukturu, vzhled a možnost rozšíření. Postrádá ale redakční systém pro jednoduchou udržitelnost a podrobné informace, které má návštěvníkům poskytovat.

Projekt ve stavu, v jakém ho k maturitě dokládám se může zdát nedokončený. Moje práce nemá sloužit v aktuálním stavu jako hotový produkt, ale jako demonstrace, jak by mohly stránky vypadat, a to ze stránky frontendu i backendu. Po zakončení maturit plánuji přednést tuto ukázkou vedení oddílu a pracovat na stránkách dál a uvést je do reálného provozu.

Odkazy

Fabrex-Template: <https://wrapbootstrap.com/theme/fabrex-multipurpose-business-admin-WB0C89H44>

VS Code: <https://code.visualstudio.com/>

React.js: <https://react.dev/>

.NET CORE: <https://dotnet.microsoft.com/en-us/>

Dukla Brno: www.duklabrnosprint.cz

Git: <https://github.com/>

Referenční odkazy jiných týmů:

- Jumbo-Visma - <https://www.teamjumbovisma.com/>
- BEAT Cycling Club - <https://www.beatcycling.cc/>
- SOUDAL-QUICK STEP Team - <https://www.soudal-quickstepteam.com/en>
- ATT Investments Pro Cycling Team - <https://www.attcycling.cz/>