

DELTA – Střední škola informatiky a ekonomie s.r.o.

Ke Kamenci 151, PARDUBICE



## TVORBA WEBOVÝCH STRÁNEK & ŘEŠENÍ DATABÁZE

Martin Macicha

4.A

Obor č. 18: Informatika

2020/21

# **Prohlášení**

Prohlašuji, že jsem maturitní projekt vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

V Pardubicích dne 29.3. ....

Martin Macicha

## **Anotace**

Cílem práce je vytvoření webového portfolia firmy Rehamza s.r.o, společně s řešením jednoduchého informačního systému pro správu softwarových produktů včetně databáze uživatelů. Součástí teoretické práce bude porovnání softwarových nástrojů pro tvorbu jednoduchých informačních systémů.

Informační systém umožní uživateli vstup do systému, stažení softwarových produktů a správu zakoupených licencí.

Součástí stránek bude i podrobný katalog softwarových řešení s možností stažení produktových materiálů.

## **Klíčová slova**

Webová stránka; databáze; NextJs; Firebase;

## Obsah

1	Úvod.....	4
2	Teoretická část .....	4
2.1	Nuxt.js .....	4
2.2	ASP.NET.....	5
2.2.1	ASP.NET WebForms.....	5
2.2.2	ASP.NET MVC .....	5
2.3	Next.js .....	6
2.4	Moje volba – Next.js .....	6
3	Použité technologie.....	6
3.1	React.....	6
3.2	Firebase .....	6
3.2.1	Firebase Auth.....	6
3.3	Cloud Firestore.....	7
3.4	Firebase Admin SDK .....	7
3.5	Semantic UI React.....	7
4	Použité nástroje.....	7
4.1	Visual Studio Code .....	7
4.1.1	Prettier.....	8
5	Vlastní řešení .....	8
5.1	Návrh databáze.....	8
5.1.1	Licence.....	8
5.1.2	Uživatel.....	9
5.2	Databázový systém.....	9
5.3	Webová stránka.....	10
5.3.1	Hlavní stránka .....	10
5.3.2	Responsivita webové stránky.....	11
5.3.3	Stránka produktů.....	12
5.3.4	Přihlašovací stránka .....	13
5.3.5	Uživatelský profil .....	13
6	Závěr .....	14
7	References.....	<b>Error! Bookmark not defined.</b>

# 1 ÚVOD

Společnost Rehamza s.r.o. byla založena v říjnu roku 2017 a jejím zakladatelem je Ing. Pavel Čáp, který patří mezi odborníky v programování a vývoji elektronických zařízení. Společnost REHAMZA s.r.o. vznikla jako první krok realizace nápadu jejích společníků zpřístupnit nejmodernější léčebné a rehabilitační metody širokému okruhu pacientů.

Hlavní ideou jejich produktů je tendence nutit pacienty hrát si, bavit se, konat něco příjemného a zábavného a současně se dostaví sekundární efekty, kterými jsou monitoring aktivity jednotlivého pacienta a též dosažení pokroku v léčbě konkrétního hendikepu. Velice významným aspektem práce s jejich produkty je zpětná vazba, která se dostává lékařům, kdy lékař získává detailní přehled o tom, jak pacient „cvičí“ a jaké mu cvičení přináší výsledky. Všechny tyto funkce se neustále zdokonalují, a to tak, jak se jim dostává pozitivních ohlasů od odborníků i pacientů, kteří s našimi výrobky denně pracují. [1]

## 2 TEORETICKÁ ČÁST

V teoretické části své maturitní práce porovnám nástroje pro vytváření informačních systémů. Při porovnávání se budu zaměřovat hlavně na kontext použití a jiné okolnosti.

### 2.1 Nuxt.js

Nuxt.js je open source aplikační framework založený na Vue.js, Node.js, Webpacku a Babel.js. Je inspirován Reactím Frameworkem Next.js. Nuxt.js odstraňuje detaily distribuce kódu pro server a klienta, díky čemuž se vývojář může soustředit na vývoj aplikace. Cílem Nuxt.js je být dostatečně flexibilní na to, aby byl použitelný jako hlavní základnu projektu.

Jedna z největších výhod Nuxt.js je, že zjednodušuje vytváření univerzálních aplikací. Univerzální aplikace je termín používaný pro popis JavaScript kódu, který procesuje kód na straně klienta i serveru. Dobrý příklad univerzální aplikace je takzvaná SPA (Single Page Application) což je stránka která se dynamicky přepisuje podle toho, jak se na ní uživatel pohybuje.

Univerzální aplikace předkreslí aplikaci nejprve na webovém serveru, pošle načtené HTML jako odpověď na požadavek prohlížeče, což ulehčuje Googlu indexování vaší stránky a vylepšuje SEO. Vytváření univerzálních aplikací může být složité, protože musíte nastavovat hodně věcí na straně serveru a klienta. Nuxt.js tento problém řeší tak, že kód pro front-end a back-end je sdílený, takže se dá jednoduše soustředit na logiku aplikace. [2]

## 2.2 ASP.NET

ASP.NET je součástí .NET Frameworku pro tvorbu webových aplikací a služeb. Je nástupcem technologie ASP (Active Server Pages). Ačkoliv je ASP.NET odvozen od starší technologie pro vývoj webů ASP, obě technologie jsou velmi odlišné. ASP.NET je založen na CLR (Common Language Runtime), který je sdílen mezi všemi aplikacemi postavenými na .NET Frameworku. Programátoři tak mohou realizovat své projekty v jakémkoliv jazyce podporujícím CLR, jimi jsou například Visual Basic.NET, C#, C++.

Aplikace založené ASP.NET mají být rychlejší, díky tomu že jsou předkompilovány do jednoho a více DLL souborů, na rozdíl od skriptovacích jazyků, kde se musí stránky opakovaně analyzovat při každém vstupu. [3]

### 2.2.1 ASP.NET WebForms

ASP.NET WebForms ulehčuje programátorům přechod od programování Windows aplikací do prostředí webu. Stránky jsou poskládány z objektů, ovládacích prvků, které jsou podobné ovládacím prvkům ve Windows aplikacích. Při tvorbě webových stránek je tedy možné používat ovládací prvky jako jsou tlačítka (Button), nápisy (Label) a další. Těmto ovládacím prvkům lze přiřadit vlastnosti, zachytávat na nich události a další. Tak jako se Windows Forms kreslí do formulářů na obrazovku, ovládací prvky ASP.NET WebForms produkují HTML kód, který tvoří část stránky poslané klientovi. [4]

### 2.2.2 ASP.NET MVC

Oblíbený návrhový vzor, který se používá zejména na webu, ačkoli byl původně používán na desktopech. Základní myšlenkou MVC architektury je oddělení logiky od výstupu, což by mělo řešit problém „zapleteného kódu“, kdy se v jednom souboru vyskytují logické operace a zároveň vykreslování výstupu. MVC tak míří pro lepší udržovatelnost kódu.

Toho se snaží MVC dosáhnout tak, že celou aplikaci rozděluje na komponenty 3 typů, jimiž jsou Model, View a Controller, od toho MVC. Komponenta je v tomto případě označení třídy, děděné z abstraktní třídy Mode, View a Controller.

**Model** obsahuje vše, co se týká logiky aplikace. Příkladem jsou třeba výpočty, databázové dotazy, validace a podobně. Jeho funkce spočívá v přijetí parametrů zvenčí a vydávání dat ven. Model nemá tušení, odkud data přišla a ani jak budou data zformátována při výstupu

**View** se stará o zobrazení výstupu uživateli. Nejčastěji se jedná o phtml šablonu, obsahující HTML stránku a tagy nějakého značkovacího jazyka, který umožňuje vkládání proměnných do šablon. View obsahuje minimální množství logiky, která je pro výpis potřebná.

**Controller** je chybějící prvek z trojice, který má za úkol osvětlit funkčnost celého vzoru. Jedná se tedy o jakéhosi prostředníka, se kterým všechno komunikuje. Propojuje mezi sebou jednotlivé komponenty a drží pohromadě celý systém. [5]

## 2.3 Next.js

Next.js je Open-source Reactí framework který podporuje vykreslování ze strany serveru. Jde o „předvykreslení“ webové stránky používající JS kód na straně webového serveru. Uživatel dostane nejprve částečně vykreslenou stránku, která se mu zobrazí skoro okamžitě. Až během toho, co se uživatel na stránce orientuje, pošle Next uživateli JavaScript, který jeho prohlížeč zpracuje a provede tak interaktivní prvky.

Díky tomu že se celá stránka vykresluje už na serveru je web přívětivější pro vyhledavače kvůli SEO. Next.js také podporuje Hot Reload což usnadňuje proces vývoje, protože se stránka automaticky obnoví pokaždé, když zjistí změnu souborů na disku. Stránky se vykreslují pouze s knihovny a JavaScriptem které jsou potřeba. Celá aplikace je rozdělena Next.js na několik různých zdrojů. Například pokud jedna ze stránek používá knihovnu Axios, často používaný HTTP klient, bude ho mít pouze ta jediná stránka ve svém balíčku. [6]

## 2.4 Moje volba – Next.js

Pro tvorbu svého maturitního projektu jsem si vybral **Next.js**. Samotný framework láká už svojí podrobně napsanou dokumentací, ve které je důkladně popsáno, jak všechno funguje. Mým důvodem pro vybrání Next.js je využití Reactu. Je to knihovna, kterou jsme probírali při výuce, takže nebylo tak náročné oživit si základy a začít psát. Líbí se mi princip komponent a úkladně rozděleného kódu, což Next.js společně s Reactem umožňuje. Jako krajní důvod je to, že si u vývojářů tohoto frameworku můžete nechat zadarmo stránku jednoduše nasadit. Takže máte jak vývojářskou verzi aplikace, tak i verzi která funguje a jede.

## 3 POUŽITÉ TECHNOLOGIE

### 3.1 React

Front-endová JavaScriptí knihovna pro tvorbu webů a jejich uživatelského rozhraní. Reactí aplikace jsou rozkouskované do komponent – pomocí tříd nebo funkcí. Je vhodný pro práci s měnícími se daty.

### 3.2 Firebase

Platforma pro vývoj mobilních a webových aplikací. Funguje na principu Backend-as-a-service, což znamená že poskytuje předpřipravené funkce jako je například Firebase Authentication, Firebase Firestore a Firebase Cloud Storage.



#### 3.2.1 Firebase Auth

Firestore poskytuje back-endové služby pro správu uživatelů. Podporuje autorizaci hesly, telefonními čísly a přihlašování přes Google a Facebook.

### 3.3 Cloud Firestore

Flexibilní a škálovatelná NoSQL databáze která přehledně rozděljuje data (docs) do kolekcí. Firestore funguje „realtime“ což znamená že na stránce můžete mít nastavený listener který aktualizuje data na stránce při každé změně v databázi. Stránka tak může ukazovat přesná data.

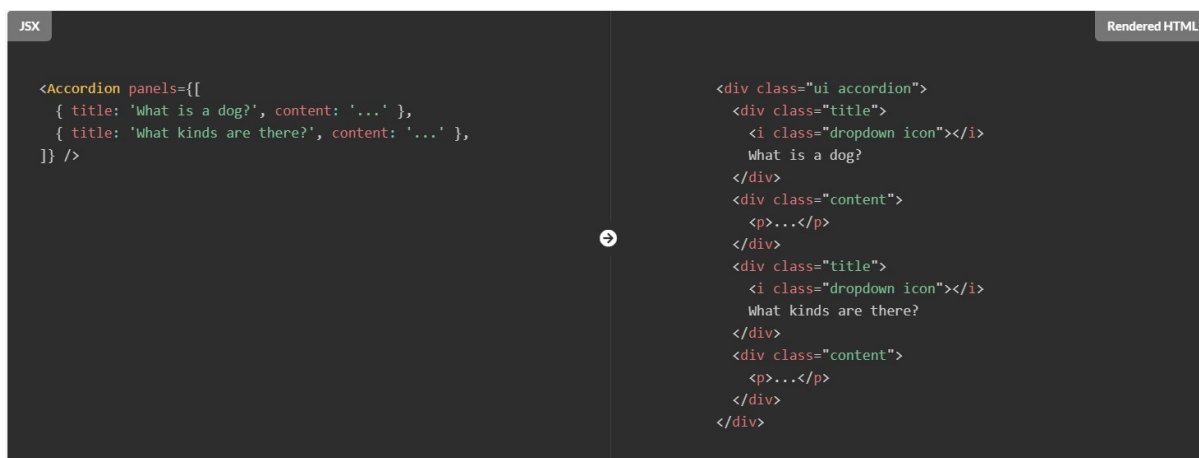
### 3.4 Firebase Admin SDK

Umožňuje číst a zapisovat do databáze s plnými administrátorskými právy. Generuje a kontroluje Firebase autorizační tokeny.

### 3.5 Semantic UI React

Semantic UI React je oficiální Reactí integrace Semantic UI, což je framework, který napomáhá vytvářet responsivní layouts s pomocí uživatelsky přívětivého a stručného HTML. HTML třídy Semantic UI jsou tak dobře čitelné a lehce upravovatelné.

Použití Semantic UI React může vypadat například takto:



```
JSX                                     Rendered HTML

<Accordion panels={[
  { title: 'What is a dog?', content: '...' },
  { title: 'What kinds are there?', content: '...' },
]} />

<div class="ui accordion">
  <div class="title">
    <i class="dropdown icon"></i>
    What is a dog?
  </div>
  <div class="content">
    <p>...</p>
  </div>
  <div class="title">
    <i class="dropdown icon"></i>
    What kinds are there?
  </div>
  <div class="content">
    <p>...</p>
  </div>
</div>
```

Obrázek 1 - Příklad použití Semantic UI

## 4 POUŽITÉ NÁSTROJE

### 4.1 Visual Studio Code

Visual Studio Code je editor kódu vyvíjený společností Microsoft. Používám jej, protože obsahuje podporu pro GitHub, zvýrazňuje syntaxi kódu a je zcela zdarma. Do programu můžete také nainstalovat spoustu doplňujících pluginů které pomáhají s efektivností programu.



## 4.1.1 Prettier

Prettier je formátovač kódu který zaručuje že celý kód má konzistentní styl.

Prettier v akci může vypadat například takto:

```
foo(reallyLongArg(), omgSoManyParameters(), IShouldRefactorThis(), isThereSeriouslyAnotherOne());
```

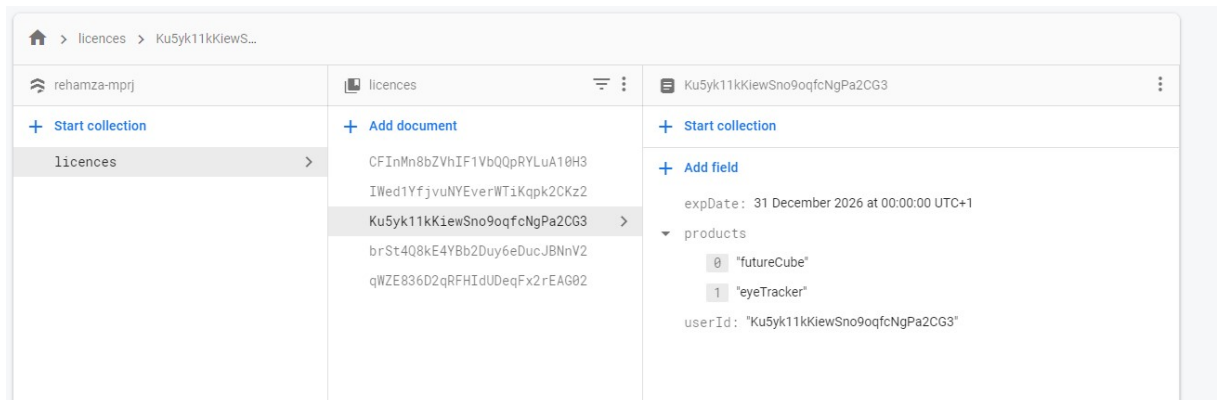
Obrázek 2 - Kód před použitím Prettier

```
foo(  
  reallyLongArg(),  
  omgSoManyParameters(),  
  IShouldRefactorThis(),  
  isThereSeriouslyAnotherOne()  
);
```

Obrázek 3 - Automatické formátování Prettier

## 5 VLASTNÍ ŘEŠENÍ

### 5.1 Návrh databáze



Obrázek 4 - Návrh databáze

#### 5.1.1 Licence

Každá licence musí obsahovat určité informace, aby mohla být zpracována systémem.

- **userId** – Uživatelské ID je unikátní identifikátor uživatelů. Pomocí uživatelského ID lze uživatli zobrazit které produkty vlastní
- **products** – Pole produktů, které v sobě drží, na které produkty se licence vztahuje.
- **expDate** – Datum kdy licence vyprší; umožňuje uživateli na profilu zobrazit za jak dlouho bude licence deaktivována

## 5.1.2 Uživatel

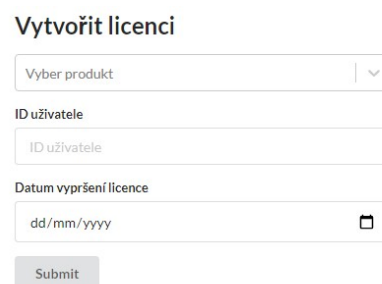
Jak už bylo zmíněno v použitých technologiích, uživatele spravuji s pomocí Firebase Auth. Každý uživatel obsahuje data jako je například email uživatele nebo jestli má uživatel administrátorská práva. Uživatel obsahuje také uživatelské ID, pomocí kterého uživateli přiřazují licence.

## 5.2 Databázový systém

Celý databázový systém je přístupný pouze uživateli s administrátorskými právy. Uživatelé bez administrátorských práv do databázového systému nevidí, a nedostanou se tam ani pomocí přímého odkazu na administrátorský dashboard.

Po načtení dashboardu stránka zkontroluje, zda uživatel zrovna přihlášený disponuje administrátorskými právy. Zde má velkou výhodu vykreslování ze strany serveru – pokud uživatel nemá dostatečná práva stránka se uživateli nenačte. Pokud má uživatel administrátorská práva, stránka načte dvě komponenty:

- Formulář **Vytvořit licenci** – Jak už z názvu vypovídá, formulář slouží k zapisování licencí do databáze. Aby se licence zapsala musí admin zadat *produkty*, *ID uživatele* a *Datum kdy licence vyprší*. Po kliknutí na tlačítko *Submit*. Si stránka ještě jednou zkontroluje, zda má uživatel dostatečná práva a až poté licenci vytvoří.



Obrázek 5 - Licenční formulář

- Tabulka uživatelů – Po úvodní kontrole práv stránka načte tabulku licencí, ve které se zobrazují všichni zaregistrovaní uživatelé a informace o nich. Všechny licence jsou uloženy ve Firebase Firestore databázi ve formě dokumentů.

Email	Uid	Datum posledního přihlášení	Je admin?	Licence
test@test.cz	IWed1YfjuNYEverWTiKqpk2CKz2	10.11.2020	Není admin	balancePad,futureCube
admin@admin.cz	Ku5yk11kKiew5no9oqfcNgPa2CG3	21.2.2021	Je admin	futureCube,eyeTracker
zerotwo@poggers.megumin	qWZE836D2qRFHldUDeqFx2rEAG02	20.12.2020	Není admin	futureCube,eyeTracker

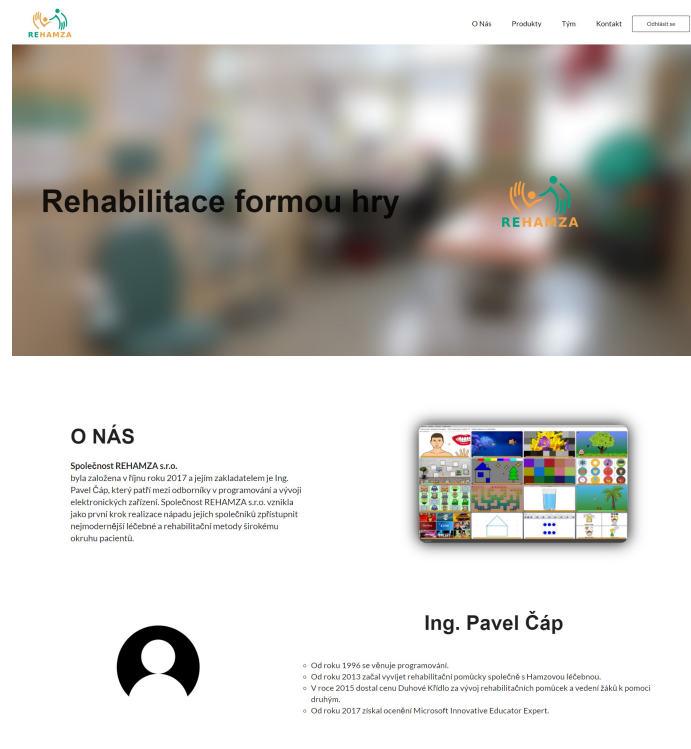
Obrázek 6 - Tabulka uživatelů

## 5.3 Webová stránka

Tato sekce dokumentace popisuje webovou stránku z grafického hlediska.

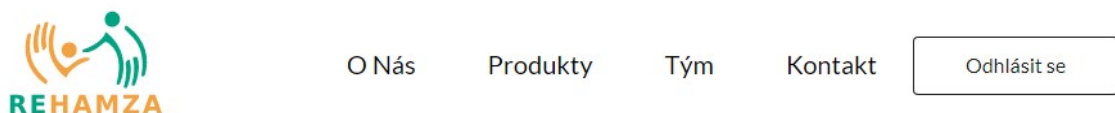
### 5.3.1 Hlavní stránka

Cílem hlavní stránky je informovat návštěvníka o Rehamze, její historii a teamu který za ní stojí. Stránka je pro přehlednost rozdělena do sekcí. Jsou na ní pouze stručné informace.



Obrázek 7 - Hlavní stránka

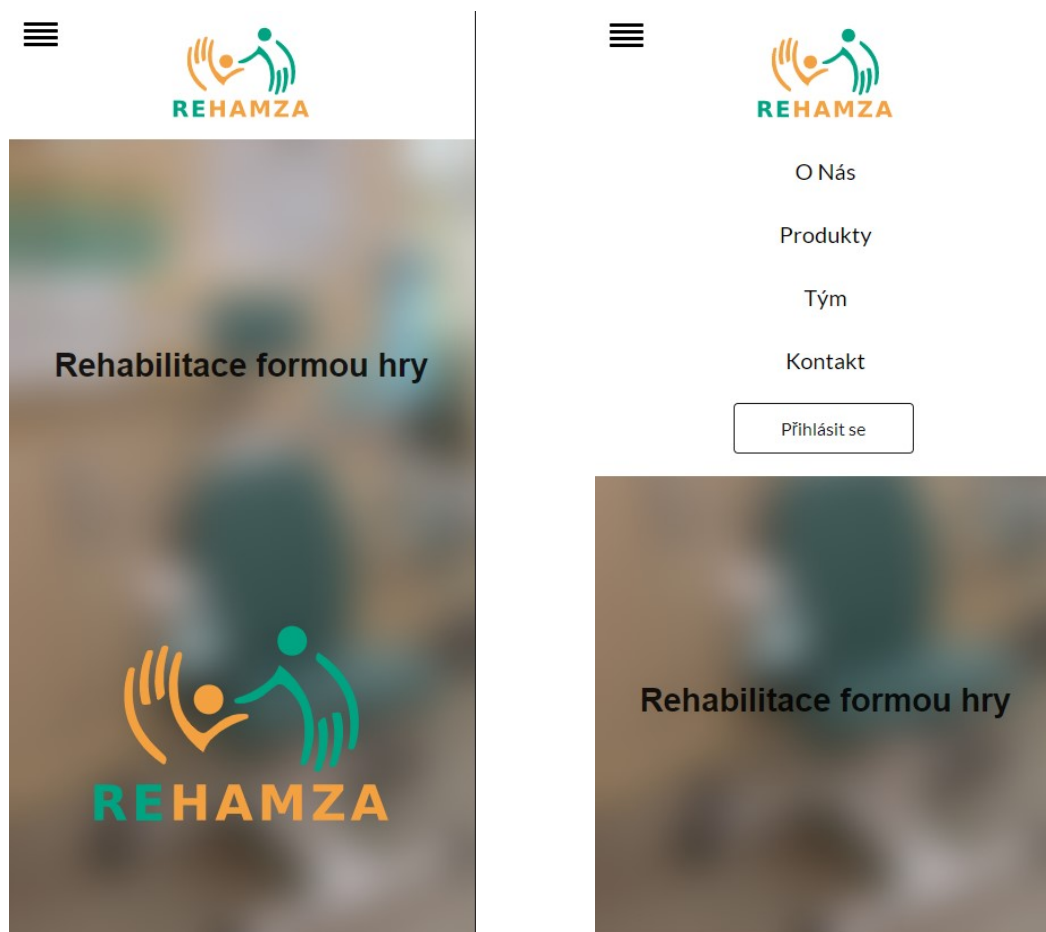
Součástí celé webové stránky je navigační menu, které slouží k pohybu po stránce. V navigačním menu se také nachází tlačítko na přihlášení, které mění stav podle toho, jestli je uživatel přihlášený nebo ne.



Obrázek 8 - Navigační menu

### 5.3.2 Přizpůsobivost webové stránky

Celá stránka včetně navigačního menu je plně responsivní. Je tomu tak protože stránka očekává hodně návštěvníků, kteří používají smartphone. Z horizontálního navigačního menu se stane vertikální, které se dá rozbalit.



Obrázek 9 - Ukázka přizpůsobivosti

### 5.3.3 Stránka produktů

Stránka produktů slouží jako katalog rehabilitačních pomůcek Rehamzy. Je rozdělená na části, takže je při malém počtu jednoduché produkty přidávat. Jak už z předchozí věty vyplývá, produkty jsou zatím na stránce zasazeny natvrdo.

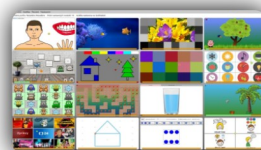


O Nás   Produkty   Tým   Kontakt  

## NAŠE PRODUKTY

### Rehamza Software

Jedná se o softwarové řešení pro podporu rehabilitace. Skládá se ze sady programů, které navzájem komunikují a společně vytváří rehabilitační prostředí pro pacienta i lékaře. Software obsahuje velké množství cvičebních modulů. Software umožňuje individuální nastavení pro široké spektrum diagnóz.



### Future Cube

Speciální barevná kostka, která slouží jako prostorový bezdrátový ovladač. Toto čidlo má i zvukovou a vibrační vazbu. Její předností je snadný úchop a bezdrátové nabíjení. Vhodné pro pacienty s poruchou hrubé motoriky, stability, u dětí s různou formou DMO, po kraniotraumatu a jiných úrazech, u dětí s poruchou koncentrace, senzorky. Kontraindikace nejsou známy. Je zde důležitá zraková kontrola a schopnost porozumění zadaným úkolům.

### Motivační robot

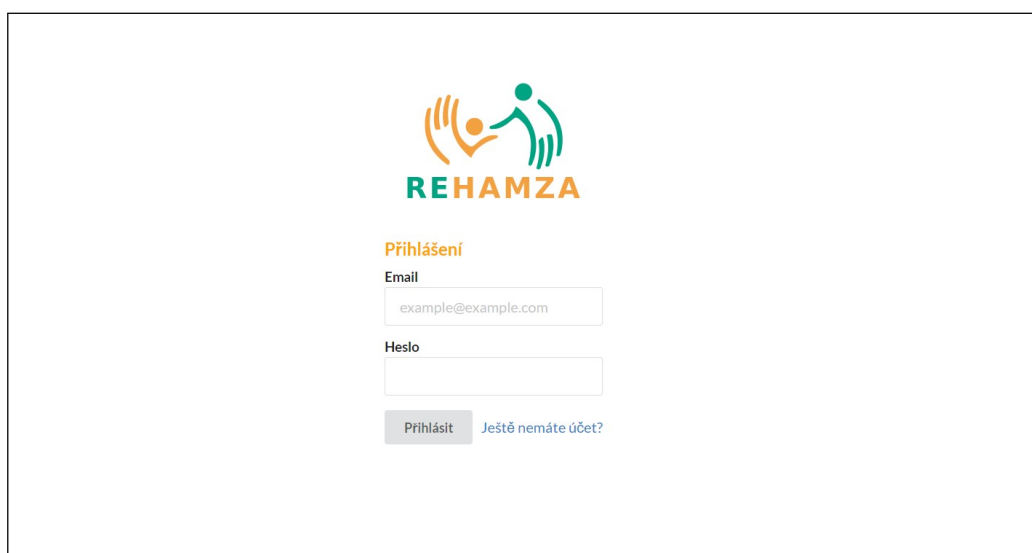
Jedná se o malého humanoidního robota, který je bezdrátově propojen s počítačem PC. Robot se dokáže naučit libovolné cviky a posléze je opakovat. Rehabilitační pracovník pomocí manipulace s robotem ukáže, jak cviky provádět a po spuštění programu robot cviky opakuje. Tento projekt je určený především dětským pacientům pro zvýšení motivace ke cvičení a zároveň umožňuje terapeutovi sledovat a upravovat kvalitu prováděných cviků.



Obrázek 10 - Stránka produktů

### 5.3.4 Přihlašovací stránka

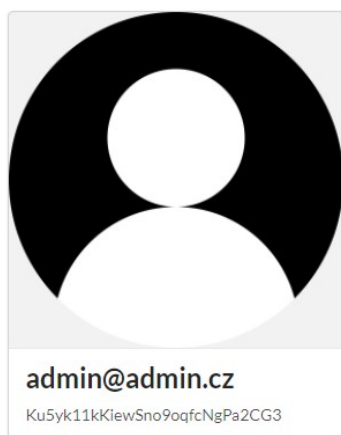
Po kliknutí na tlačítko přihlásit následuje přesunutí na přihlašovací stránku. Pro přihlášení musí uživatel vyplnit svůj email a heslo. Pro přihlášení uživatele používám funkci *signInWithEmailAndPassword* z Firebase Auth. Pokud uživatel zatím nemá účet, je pod přihlašovacím formulářem odkaz (*Ještě nemáte účet?*) na stránku pro vytvoření účtu. Na stránce pro registraci je pouze změněný název formuláře, použitá funkce a je přehozený odkaz, aby se uživatel dostal zpět na přihlašovací stránku.



Obrázek 11 - Přihlašovací formulář

### 5.3.5 Uživatelský profil

Uživatelský profil je pro běžného uživatele důležitá stránka. Zobrazují se mu zde důležité informace o něm, jako je například jeho e-mail. Pro můj projekt je uživatelský profil důležitý, protože umožňuje uživateli zobrazit jeho [vlastněné licence](#), a [kdy mu vyprší](#).



#### Vámi vlastněná licence

futureCube,eyeTracker

Do vypršení licence zbývá 2115 dní

Obrázek 12 - Informace zobrazené na profilu

## 6 ZÁVĚR

V teoretické části dokumentace jsem porovnal různé nástroje na vytváření informačních systémů. Na základě jejich porovnání jsem vybral vhodný nástroj pro tvorbu mého informačního systému.

Výsledkem práce je webová stránka, která slouží jako informační portál. Uživatelům, co mají dostatečná práva, je umožněn přístup do informačního systému, kde můžou spravovat licence. Správa licencí je hlavní a nejdůležitější část mého projektu. V celém systému byla použita moderní technologie Next.js. Stránka je vytvořena tak aby byla plně přizpůsobivá na mobilní zařízení pro pohodlí uživatelů.

## 7 REFERENCE

- [1] Rehamza s.r.o., „Rehamza s.r.o.“ 2015. [Online]. Available: <http://www.rehamza.cz/#>.
- [2] D. Sozo, „Medium,“ 19 March 2018. [Online]. Available: <https://medium.com/vue-mastery/10-reasons-to-use-nuxt-js-for-your-next-web-application-522397c9366b>.
- [3] „Wikipedia,“ Microsoft, [Online]. Available: [https://en.wikipedia.org/wiki/ASP.NET\\_MVC](https://en.wikipedia.org/wiki/ASP.NET_MVC).
- [4] Microsoft, „Microsoft,“ Microsoft, [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/web-forms/what-is-web-forms>.
- [5] „Wikipedia,“ [Online]. Available: <https://en.wikipedia.org/wiki/Next.js>.
- [6] D. Čápka, „ITnetwork.cz,“ [Online]. Available: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>.

Obrázek 1 - Příklad použití Semantic UI.....	7
Obrázek 2 - Kód před použitím Prettier.....	8
Obrázek 3 - Automatické formátování Prettier.....	8
Obrázek 4 - Licenční formulář.....	9
Obrázek 5 - Tabulka uživatelů.....	9
Obrázek 6 - Hlavní stránka .....	10
Obrázek 7 - Navigační menu .....	10
Obrázek 8 - Ukázka přizpůsobivosti.....	11
Obrázek 9 - Stránka produktů.....	12
Obrázek 10 - Přihlašovací formulář.....	13
Obrázek 11 - Informace zobrazené na profilu .....	13