Wichterlovo gymnázium, Ostrava-Poruba, příspěvková organizace, Čs. exilu 669/16, Ostrava-Poruba, 708 00



**E-shop – webová aplikace**

Ostrava, 2025

Filip Ševčuk

Vedoucí práce: Mgr. Monika Pinkasová

# Čestné prohlášení

Prohlašuji, že jsem svou maturitní práci vypracoval samostatně a na základě literatury a pramenů uvedených v seznamu použitých zdrojů.

V Ostravě dne:

Popis:

# Obsah

[**Čestné prohlášení 2**](#_ol4rlwrz0qo7)

[**Obsah 3**](#_3keiwcvdfn3u)

[**1. Úvod 5**](#_cg7if5kjhbej)

[1. 1. Proč zrovna e-shop? 5](#_ijpg0kc0rph4)

[1. 2. Výběr technologií 5](#_awf21o390uz0)

[1. 3. Cíle práce 5](#_h8jbtfu5qliu)

[**2. Popis aplikace 7**](#_y8p1lmn0okxs)

[2. 1. Hlavička (Header) 7](#_jnwbpheane4m)

[2. 2. Patička (Footer) 8](#_4oy4s6wmtji3)

[2. 3. Domovská stránka 8](#_p5sxwbhk64xk)

[2. 4. Stránka produktů (Products) 8](#_62b449860ndn)

[2. 5. Přihlášení a registrace 9](#_oiug7rlvhr13)

[2. 6. Košík 10](#_hzz95j728p3g)

[2. 7. Objednávkový formulář 11](#_1evq4089asez)

[2. 8. Uživatelský účet (Profile) 12](#_74bmo9cp5061)

[2. 9. Závěr popisu aplikace 12](#_gv17365j0ywx)

[**3. Technologie a nástroje 13**](#_hs4szbi2hras)

[3. 1. React 13](#_3ctks5fyhjgz)

[3. 2. Firebase 13](#_uu7j6ceujwl1)

[3. 3. JavaScript 14](#_c8h5z5vv607h)

[3. 4. HTML a CSS 14](#_tg3wu718zq4o)

[3. 5. Google Maps API 14](#_noz8q6bp7nog)

[3. 6. React Icons 15](#_ypnse4rcghnb)

[3. 7. Git a GitHub 15](#_mxobfrvfwqyp)

[3. 8. Netlify 15](#_efzx3k5sled0)

[3. 9. Visual Studio Code (VS Code) 15](#_dj4axv38qkx)

[3. 10. Závěr použitých technologií 16](#_1wfg7um80thy)

[**4. Vývojový proces 17**](#_w0g8wgovw5ge)

[4. 1. Počáteční myšlenka a plánování 17](#_3wrn77ndq12g)

[4. 2. Návrh aplikace 18](#_y2hgdebnllv8)

[4. 3. Implementace 18](#_8we9afdjcn6e)

[4. 4. Testování 19](#_49pleu530p88)

[4. 5. Nasazení 20](#_j522uv1meajy)

[4. 6. Závěr vývojového procesu 20](#_7iml8klmz9fk)

[**5. Organizace kódu v Reactu 21**](#_hu0k291e6quw)

[5. 1. Komponenty (pages) 21](#_rk3nkkx9g7hr)

[5. 2. Stránky (pages) 21](#_nhcamip1odtq)

[5. 3. Propojení v App.js 21](#_u6w575msvooj)

[5. 4. Ostatní soubory a složky 21](#_je5fev1re09d)

[**6. Struktura databáze (Firebase Firestore) 22**](#_tyjjtau6rdh7)

[**7. Ukázka kódu konkrétní funkce 23**](#_56lqgxd8yg2h)

[7. 1. Deklarace 24](#_uk8qzrqh00m1)

[7. 2. Toastify 24](#_fdmqeed577jz)

[7. 3. Ověření 24](#_7anh954nru6v)

[7. 4. Přidání produktu do košíku 24](#_vmi4w7qwh6em)

[7. 5. Uložení do databáze 25](#_4oq29kppe0u6)

[7. 6. Aktualizace ikony košíku 25](#_6xexrkigfvkc)

[**8. Závěr 26**](#_tgjj7iuq0auo)

[**9. Spuštění aplikace 27**](#_agdis56qnz5y)

[**10. Zdroje 28**](#_h6ym9tqx1zir)

# 

# 

# 1. Úvod

Cílem této maturitní práce je vytvořit funkční webovou aplikaci – internetový obchod (e-shop). Na základě svého zájmu ve sbírání mečů a jiných zbraní jsem se rozhodl web stylizovat do této tématiky a nazvat jej “Sword World”.

E-shop je navržen tak, aby nabízel uživatelům přehledný a intuitivní způsob nákupu produktů, zároveň však slouží jako platforma pro mé další vzdělávání a zdokonalování v oblasti webového vývoje.

## 1. 1. Proč zrovna e-shop?

Myšlenka vytvořit e-shop vznikla již dříve, když jsem se poprvé seznámil s programováním. Můj úplně první projekt ve škole byl jednoduchý webový obchod vytvořený pomocí HTML a CSS. Tehdy jsem ještě neměl dostatečné znalosti, a proto výsledek nebyl příliš působivý. Rozhodl jsem se tedy tento projekt znovu oživit a výrazně vylepšit pomocí moderních technologií, jako je JavaScript, databáze a další nástroje.

## 1. 2. Výběr technologií

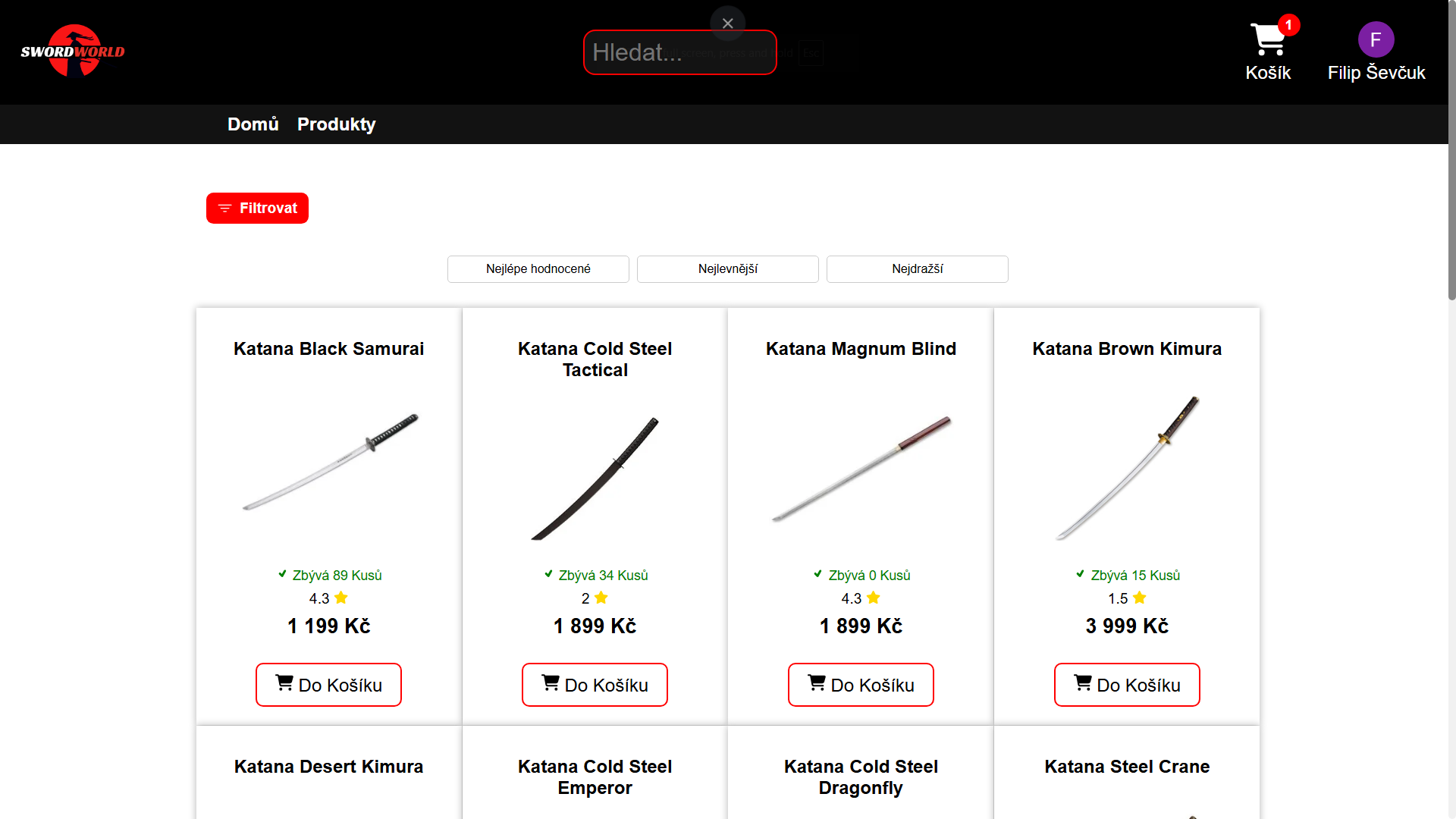
Pro vývoj aplikace jsem zvolil React, protože se mi líbí jeho modularita, jednoduchá syntaxe a možnost vytvářet dynamické uživatelské rozhraní. Kromě Reactu jsem použil Firebase pro ukládání dat a správu produktů, Google Maps API pro zobrazení lokace obchodu a React Icons pro jednoduché SVG obrázky. Celý projekt je hostován službou Netlify a verzován pomocí Git a GitHubu.

## 1. 3. Cíle práce

Hlavním cílem této práce je vytvořit plně funkční e-shop, který splňuje následující kritéria:

* Uživatelsky přívětivé rozhraní
* Možnost prohlížení produktů, jejich filtrování a přidávání do košíku
* Integrace s externími službami (Google Maps, Firebase)
* Responzivní design pro zobrazení na různých zařízeních

# 2. Popis aplikace



Aplikace „Sword World“ je moderní webový obchod specializující se na prodej mečů a dalších zbraní. Cílem aplikace je poskytnout uživatelům jednoduchý a intuitivní způsob prohlížení a nákupu produktů, zároveň však nabízí funkce, jako je filtrování produktů, hodnocení a recenze, správa historie objednávek a integrace s Google Maps pro výběr doručovací adresy.

## 2. 1. Hlavička (Header)

Na každé stránce se nachází hlavička, která obsahuje:

* Logo (vlevo): Kliknutím na logo je uživatel přesměrován na domovskou stránku.
* Vyhledávací pole (uprostřed): Uživatel může zadat název produktu, a aplikace zobrazí relevantní výsledky.
* Ikonka košíku (vpravo): Přesměruje uživatele na stránku košíku.
* Ikonka uživatelského profilu (vedle košíku): Zobrazuje uživatelské jméno a přesměruje na stránku uživatelského účtu.

Na mobilních zařízeních se hlavička přizpůsobí a některé prvky (jako vyhledávací pole a ikonky) jsou přesunuty do burger menu (ikona s třemi vodorovnými čarami). Kliknutím na burger menu se zobrazí rozbalovací nabídka s těmito možnostmi.

## 2. 2. Patička (Footer)

V patičce na každé stránce uživatel najde kontaktní informace (e-mail, telefon) a odkazy na důležité profily tvůrce a majitele e-shopu.

## 2. 3. Domovská stránka

Domovská stránka obsahuje stručný popis e-shopu a jeho zaměření. K tomu se zde také nachází bannerový obrázek stylizovaný do tématiky obchodu. Zde je možné také stáhnout tuto dokumentaci po kliknutí na modrý odkaz s nápisem “dokumentace”.

## 2. 4. Stránka produktů (Products)

Stránka produktů zobrazuje všechny dostupné produkty. Každý produkt je prezentován pomocí:

* Obrázku
* Názvu
* Ceny
* Množství na skladě
* Průměrného hodnocení.
* Tlačítka pro přidání do košíku.

Uživatel může produkty filtrovat podle:

* Ceny (rozsah od–do).
* Barvy
* Štítků (např. ostré)

Produkty lze také řadit podle:

* Nejlevnějších
* Nejdražších
* Nejlépe hodnocených

Kliknutím na produkt je uživatel přesměrován na detailní stránku produktu, kde najde podrobný popis produktu, možnost přidat recenzi nebo přečíst recenze od ostatních uživatelů.

## 2. 5. Přihlášení a registrace

Nepřihlášení uživatelé mohou prohlížet produkty, ale pro přidání zboží do košíku je zákazník vyzván k registraci nebo přihlášení.

Pokud nepřihlášený uživatel klikne na košík nebo ikonku účtu, je přesměrován na přihlašovací stránku.

Na přihlašovací stránce se může uživatel přihlásit pomocí e-mailu a hesla nebo účtu Google. Nebo se registrovat přes jméno, příjmení, e-mail a heslo.

## 2. 6. Košík

V košíku uživatel vidí vybrané produkty spolu s jejich množstvím. Má možnost upravit množství (přidat nebo odebrat kusy). Dole vidí celkovou cenu objednávky.

Pokud je uživatel spokojen, klikne na tlačítko „Pokračovat“, které otevře objednávkový formulář.

## 2. 7. Objednávkový formulář

## 

Zde uživatel vyplní:

* E-mail
* Jméno a příjmení
* Způsob doručení
* Způsob platby
* Adresu pro doručení, kterou může vybrat přímo na mapě (integrovaná mapa Google Maps).

Po úspěšném vyplnění formuláře může uživatel objednávku dokončit. Informace o objednávce jsou uloženy do databáze majitele e-shopu.

## 2. 8. Uživatelský účet (Profile)

Přihlášení uživatelé mají přístup ke stránce „Účet“, kde mohou zobrazit historii svých objednávek.

## 2. 9. Závěr popisu aplikace

Aplikace „Sword World“ kombinuje jednoduchost a funkčnost, což z ní dělá skvělý nástroj pro nákup mečů či jiných zbraní online. Díky funkcím, jako je filtrování produktů, hodnocení a integrace s Google Maps, poskytuje uživatelům příjemný a bezproblémový nákupní zážitek.

# 3. Technologie a nástroje

V této kapitole jsou popsány všechny technologie a nástroje, které byly použity pro vývoj a provoz této webové aplikace. Každá technologie je stručně vysvětlena, včetně jejího účelu a způsobu použití v projektu.

## 3. 1. React

React je open-source JavaScriptová knihovna pro tvorbu uživatelských rozhraní (UI). Byla vyvinuta společností Facebook a je široce používána pro vytváření dynamických a interaktivních webových aplikací. React je založen na komponentovém modelu, což znamená, že aplikace je rozdělena na menší, samostatné části (komponenty), které lze snadno znovu použít a kombinovat.

React jsem zvolil pro svou modularitu, jednoduchou syntaxi a vysokou výkonnost. Díky virtuálnímu DOM (Document Object Model) minimalizuje počet operací potřebných pro aktualizaci uživatelského rozhraní, což zlepšuje rychlost aplikace.

Všechny stránky a prvky aplikace (např. hlavička, patička, produktové karty) jsou vytvořeny jako React komponenty. React také umožňuje snadnou správu stavu aplikace pomocí React Hooks (např. useState, useEffect).

## 3. 2. Firebase

„Firebase je platforma pro vývoj aplikací, kterou vlastní a provozuje společnost Google. Nabízí celou řadu nástrojů a služeb pro vývojáře, které pokrývají různé aspekty vývoje aplikací, od databází a autentizace až po analýzy a cloudové funkce.“ (PEACH-DEV, [cit. 2025-04-19]).

Firebase jsem zvolil pro svou jednoduchost a škálovatelnost. Umožňuje rychlé nasazení aplikace a poskytuje všechny potřebné nástroje na jednom místě.

**Firebase Firestore:** Slouží jako databáze pro ukládání informací o produktech, uživatelích a objednávkách. Firestore je NoSQL databáze, která umožňuje ukládat data ve formě dokumentů a kolekcí.

**Firebase Authentication:** Zajišťuje autentizaci uživatelů. Uživatelé se mohou přihlásit pomocí e-mailu a hesla nebo přes Google účet.

## 3. 3. JavaScript

JavaScript je programovací jazyk, který se používá pro vytváření interaktivních webových stránek. Je základem pro většinu moderních webových aplikací.

JavaScript je nezbytný pro práci s Reactem a pro přidání interaktivity do aplikace.

JavaScript je používán pro logiku aplikace, jako je manipulace s daty, řízení stavu a komunikace s Firebase.

## 3. 4. HTML a CSS

HTML (HyperText Markup Language) a CSS (Cascading Style Sheets) jsou základní technologie pro vytváření webových stránek. HTML slouží pro strukturu stránky, zatímco CSS se používá pro stylování.

Tyto technologie jsou nezbytné pro vytvoření základní struktury a vzhledu aplikace.

HTML a CSS byly použity pro vytvoření layoutu stránek, stylování komponent a zajištění responzivního designu.

## 3. 5. Google Maps API

Google Maps API je služba, která umožňuje integraci map do webových aplikací. Poskytuje funkce pro zobrazení map, vyhledávání míst a výběr adres.

Tuto službu jsem zvolil pro svou jednoduchost a přesnost. Umožňuje uživatelům vybrat doručovací adresu přímo na mapě.

Google Maps API je integrováno do objednávkového formuláře, kde uživatelé mohou vybrat adresu pro doručení.

## 3. 6. React Icons

React Icons je knihovna, která poskytuje ikony pro React aplikace. Obsahuje ikony z populárních knihoven, jako jsou Font Awesome, Material Design a další.

Tuto knihovnu jsem zvolil pro svou jednoduchou integraci s Reactemt a širokou škálu dostupných ikon.

React Icons byly použity pro ikony v mnohých částech aplikace pro jednoduché grafické rozhraní.

## 3. 7. Git a GitHub

Git je systém pro správu verzí, který umožňuje vývojářům sledovat změny v kódu a spolupracovat na projektech. GitHub je platforma pro hostování a sdílení kódu pomocí Gitu.

Tyto služby jsem zvolil pro správu verzí kódu a usnadnění spolupráce.

Veškerý kód aplikace je verzován pomocí Gitu a uložen na GitHubu, což umožňuje snadné sledování změn a návrat k předchozím verzím.

## 3. 8. Netlify

Netlify je platforma pro hosting a nasazení webových aplikací. Nabízí jednoduché nasazení aplikací přímo z GitHubu a dalších zdrojů. Netlify jsem zvolil pro svou jednoduchost, rychlost nasazení a možnost bezplatného využívání. Aplikace je hostována na této platformě, což umožňuje snadnou distribuci a možnost rychlých aktualizací.

## 3. 9. Visual Studio Code (VS Code)

Visual Studio Code je bezplatný, open-source vývojový editor od Microsoftu, který podporuje širokou škálu programovacích jazyků a nástrojů. Je vysoce přizpůsobitelný a nabízí množství rozšíření pro zvýšení produktivity vývoje.

Visual Studio Code jsem zvolil, jelikož jsem zvyklý s tímto programem pracovat již dlouhou dobu. Má všechny potřebné funkce, intuitivní rozhraní a snadnou integraci různých potřebných rozšíření.

## 3. 10. Závěr použitých technologií

Všechny použité technologie a nástroje byly vybrány s ohledem na jejich funkčnost, jednoduchost a vhodnost pro daný projekt. React a Firebase tvoří jádro aplikace, zatímco další nástroje, jako jsou Google Maps API, React Icons a Netlify, přidávají důležité funkce a usnadňují vývoj a provoz.

# 

# 4. Vývojový proces

Vývoj aplikace „Sword World“ jsem rozdělil do několika fází, které zahrnovaly plánování, návrh, implementaci, testování a nasazení. Tato kapitola popisuje, jak jsem se k projektu dostal, jak jsem postupoval při vývoji a jak jsem aplikaci testoval.

## 4. 1. Počáteční myšlenka a plánování

Myšlenka vytvořit e-shop s tématikou zbraní a mečů vznikla na základě mého osobního zájmu. Již dříve jsem v rámci školního projektu vytvořil jednoduchý web zaměřen na tuto tématiku pomocí HTML a CSS. Ten byl ovšem velmi omezený především funkcionalitou, ale i designem. Rozhodl jsem se proto tento projekt znovu oživit a výrazně vylepšit pomocí moderních technologií.

Proč e-shop?

E-shop mi přišel jako ideální projekt, na kterém se mohu naučit řadu nových dovedností – od práce s databázemi přes možnosti Reactu až po integraci externích služeb.

Před zahájením vývoje jsem prostudoval několik existujících e-shopů, jako jsou *Alza.cz* a *Knifestock.cz*, abych získal představu o tom, jak by mohl e-shop vypadat a fungovat.

## 4. 2. Návrh aplikace

Před samotným kódováním jsem si vytvořil hrubý návrh aplikace, který zahrnoval:

* Strukturu stránek: Domovská stránka, stránka produktů, detail produktu, košík, přihlášení/registrace, uživatelský účet.
* Komponenty: Hlavička, patička, produktové stránky, filtry, formuláře.
* Datový model: Jak budou data (produkty, uživatelé, objednávky) uložena v databázi. (Více v kapitole 7)

## 4. 3. Implementace

Vývoj aplikace probíhal postupně, krok za krokem:

Nastavení projektu:

* Vytvořil jsem nový React projekt pomocí nástroje Create React App.
* Nastavil jsem Firebase pro ukládání dat a autentizaci uživatelů.
* Přidal jsem základní závislosti, jako jsou React Icons a React Router pro navigaci.

Vytvoření komponent:

Každá stránka nebo důležitá část byla implementována jako samostatná React komponenta. Například: Header, Footer, Cart, Login.

Integrace s Firebase:

Produkty jsem uložil do Firestore databáze a zařídil aby byly načítány dynamicky do aplikace. Uživatelskou autentizaci jsem vyřešil pomocí Firebase Authentication.

Přidání funkcionalit:

* Filtrování a řazení produktů.
* Přidávání produktů do košíku a správa množství.
* Integrace Google Maps API pro výběr doručovací adresy.
* Kontrola validity objednávky ze strany databáze

Stylování a responzivita:

Aplikaci jsem nastyloval pomocí CSS a přizpůsobil ji pro mobilní zařízení.

Některé prvky (např. vyhledávací pole) jsem pro mobilní zařízení přesunul do burger menu. Celý web jsem se rozhodl stylizovat do barevné kombinace černé, červené a bílé barvy.

## 4. 4. Testování

Testování probíhalo v několika fázích:

Vlastní testování:

Během vývoje jsem průběžně testoval jednotlivé funkce a opravoval nalezené chyby.

Zaměřil jsem se na to, aby vše správně fungovalo jak ze strany uživatele, tak ze strany serveru a databáze.

Testování s kamarády:

Po dokončení základních funkcí jsem aplikaci nasdílel několika kamarádům, kteří mi poskytli zpětnou vazbu. Jejich připomínky mi pomohly vylepšit uživatelské rozhraní a opravit drobné chyby.

Finální testování:

Před nasazením aplikace jsem provedl komplexní testování všech funkcí na různých zařízeních (PC, mobil, tablet).

## 4. 5. Nasazení

Po dokončení vývoje a testování jsem se rozhodl aplikaci nahrát na GitHub a hostovat pomocí Netlify.

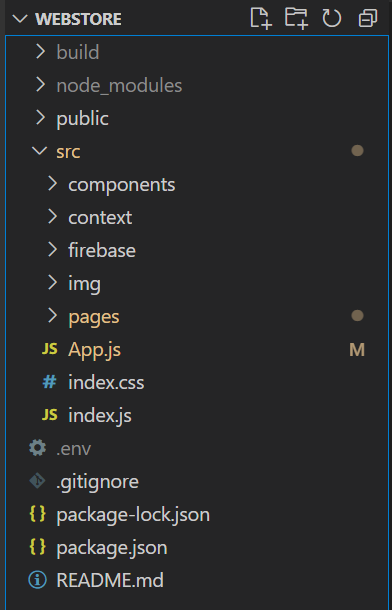
Nasazení proběhlo bez problémů a aplikace je nyní dostupná online.

## 4. 6. Závěr vývojového procesu

Vývoj této aplikace byl pro mě velkou výzvou, ale zároveň skvělou příležitostí naučit se nové dovednosti. Postupoval jsem podle jasného plánu, který zahrnoval návrh, implementaci, testování a nasazení. Díky zpětné vazbě od kamarádů se mi podařilo vylepšit aplikaci a zajistit, aby byla uživatelsky přívětivá a funkční.

# 

# 5. Organizace kódu v Reactu

Aplikace je strukturována podle standardních konvencí Reactu s jasným rozdělením na stránky a komponenty.

## 5. 1. Komponenty (pages)

Každá komponenta (např. Filters) má svůj vlastní soubor .js a přidružený .css soubor pro styly. Například:

Filters.js – definuje vzhled a chování karty filterů.

Filters.css – obsahuje styly pouze pro tuto komponentu.

## 5. 2. Stránky (pages)

Hlavní stránky (např. Products.js) využívají komponenty a definují logiku konkrétní stránky. Routování mezi stránkami je řešeno v App.js pomocí react-router-dom.

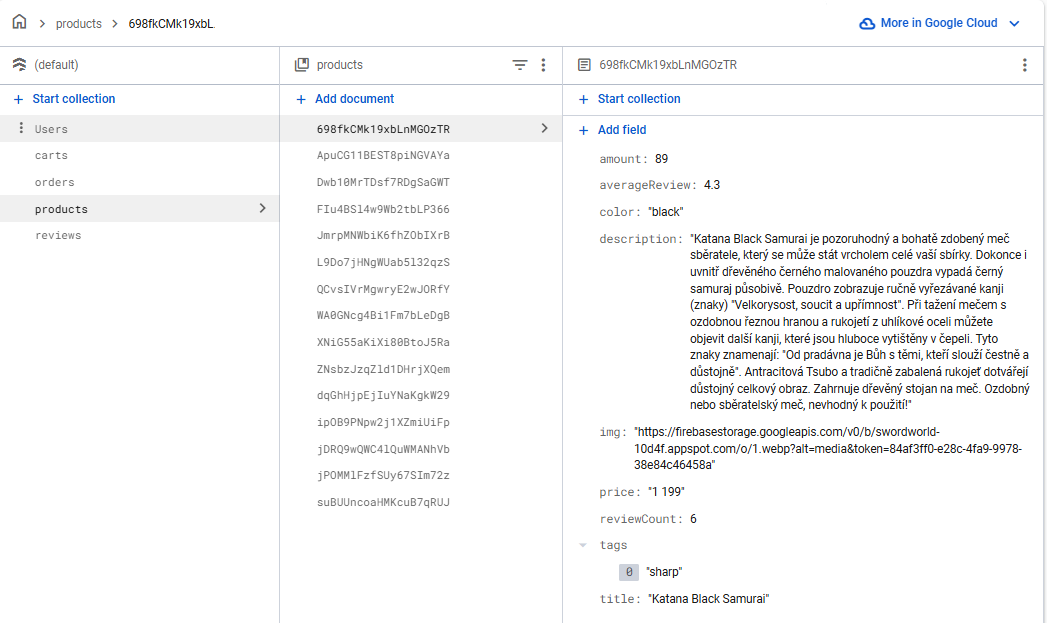
## 5. 3. Propojení v App.js

App.js slouží jako kořenový soubor, kde jsou definovány všechny cesty (routes) a globální prvky (např. společný Header a Footer).

## 5. 4. Ostatní soubory a složky

* Složka *img* obsahuje některé obrázky, které není nutné ukládat do databáze. např. ikonku.
* Složka *context* obsahuje vlastní React Hooks. V mém případě jsou zde funkce pro ověření zda je uživatel přihlášen
* Složka *firebase* obsahuje soubory propojující celý projekt s databází
* soubor *.env* obsahuje veškeré API klíče, které se dále nesdílí
* soubor .gitignore obsahuje seznam všech souborů, které jsou gitem ignorovány. Jedná se např. o *.env* soubor

# 6. Struktura databáze (Firebase Firestore)

Databáze je organizována do kolekcí a dokumentů s následující strukturou:

Každá kolekce obsahuje několik dokumentů, které dále obsahují konkrétní atributy a data. V mém případě se jedná o 5 kolekcí, přičemž dokumenty v některých jsou pevně stanoveny (např. *products* s daty o produktech), jiné jsou přepisovány na základě akcí uživatelů (např. při dokončení objednávky). Pro omezení redundance jsou potřebné dokumenty propojeny pomocí primárního klíče (ID).

# 7. Ukázka kódu konkrétní funkce

V této části podrobněji rozepíšu logiku funkce, která zajišťuje přidání produktů do košíku. Tuto funkci jsem se rozhodl popsat, jelikož je esenciální pro fungování celého projektu a dá se na ní ukázat mnoho věcí.

// přidání produktů do košíku

const addToCart = async (product) => {

if (!user) {

toast.dismiss()

toast.error("Pro přidání do košíku se přihlašte");

return;

}

if (product.amount > 0){

// provede se jen pokud je amount (zbývající kusy) > 0

const updatedProductsInCart = [...productsInCart];

const existingProductIndex = updatedProductsInCart.findIndex(p => p.id === product.id);

if (existingProductIndex !== -1) {

updatedProductsInCart[existingProductIndex].numberOfItems += 1;

} else {

updatedProductsInCart.push({ ...product, numberOfItems: 1 });

}

setProductsInCart(updatedProductsInCart);

// uloží kosik do databaze

const userCartDoc = doc(projectFirestore, "carts", user.uid);

await setDoc(userCartDoc, { cartItems: updatedProductsInCart }, { merge: true });

setInCart(inCart + 1)

toast.dismiss()

toast.success("Přidáno do košíku")

}

else{

toast.dismiss()

toast.error("zboží momentálně nemáme na skladě")

}

}

## 7. 1. Deklarace

Celá funkce addToCart je deklarována jako asynchronní (async), jelikož obsahuje operaci zápisu do databáze pomocí setDoc, která probíhá asynchronně. Použití await zajistí, že zápis do databáze proběhne správně a až poté bude možné pokračovat v dalším kódu.

## 7. 2. Toastify

V kódu je mnohokrát využita knihovna Toastify pro zobrazování uživatelských oznámení. To umožňuje okamžitou zpětnou vazbu na provedené akce.

* toast.dismiss()

před každým novým oznámením se smažou již zobrazené notifikace, aby nedocházelo k jejich hromadění.

* toast.error()

zobrazí červenou chybovou zprávu.

* toast.success()

zobrazí zelenou zprávu při úspěšném přidání produktu.

## 7. 3. Ověření

Funkce obsahuje několik kontrol pro zajištění správné funkčnosti:

Ověření přihlášení – pokud není uživatel přihlášen *(!user)*, zobrazí se chybová hláška a funkce se ukončí.

Ověření dostupnosti produktu – pokud je hodnota *product.amount* větší než 0, produkt je možné přidat do košíku, jinak se zobrazí oznámení o nedostupnosti.

## 7. 4. Přidání produktu do košíku

* Pokud produkt splňuje výše uvedená ověření, dojde k jeho přidání do košíku:
* Nejprve se vytvoří kopie aktuálního stavu košíku *(updatedProductsInCart)*.
* Funkce *findIndex* se použije k nalezení, zda již daný produkt v košíku existuje:
* Pokud ano, zvýší se jeho množství *(numberOfItems).*
* Pokud ne, přidá se nový produkt s *numberOfItems* nastaveným na 1.
* Aktualizovaný seznam produktů se následně uloží do stavu *(setProductsInCart)*.

## 7. 5. Uložení do databáze

Pro zachování stavu košíku i po obnovení stránky je nutné jej uložit do databáze Firestore. To se provádí pomocí:

const userCartDoc = doc(projectFirestore, "carts", user.uid);

await setDoc(userCartDoc, { cartItems: updatedProductsInCart },

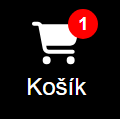
{ merge: true });

Použití *{ merge: true* } zajistí, že se aktualizují pouze položky v košíku bez přepsání celého dokumentu.

## 7. 6. Aktualizace ikony košíku

Proměnná *inCart* se používá k zobrazení malé notifikační ikonky u košíku, která indikuje počet položek v košíku. Po přidání produktu se její hodnota zvýší o 1:

setInCart(inCart + 1)



To umožňuje vizuální zpětnou vazbu uživateli bez nutnosti obnovovat stránku.

# 8. Závěr

V rámci této maturitní práce jsem vytvořil funkční e-shop, který nabízí uživatelům intuitivní prostředí pro nákup zbraní a mečů. Během vývoje jsem se naučil pracovat s moderními technologiemi, jako jsou React, Firebase a Netlify, což mi pomohlo rozšířit mé znalosti v oblasti webového vývoje.

Projekt splnil svůj účel a naplnil stanovené cíle. Aplikace umožňuje registraci a přihlášení uživatelů, přidávání produktů do košíku, správu objednávek a integraci s Google Maps API. Vývoj tohoto e-shopu mi poskytl cenné zkušenosti s návrhem databáze, správou uživatelských dat a optimalizací uživatelského rozhraní.

Do budoucna bych rád projekt rozšířil o další funkcionality, jako je pokročilejší vyhledávání produktů, doporučovací systém na základě uživatelských preferencí a implementace platební brány pro přímé platby online, případně přímé propojení s reálnými doručovacími službami.

Tato práce mi umožnila nejen zlepšit mé technické dovednosti, ale také pochopit celý proces vývoje webových aplikací – od počátečního návrhu až po finální nasazení. Doufám, že mé poznatky a zkušenosti získané během tohoto projektu mi pomohou v dalším studiu i budoucí profesní kariéře.

# 9. Spuštění aplikace

aplikaci je možné nalézt na adrese:

[*https://sword-world.netlify.app/*](https://sword-world.netlify.app/)

zdrojový kód je dostupný na drese:

[*https://github.com/BafilaCz/Sword-World*](https://github.com/BafilaCz/Sword-World)

# 

# 

# 

# 10. Zdroje

ŠETEK, David. React - posuňte své webové aplikace na nový level. [online]. YouTube, 2022. Dostupné z:

[*https://www.youtube.com/playlist?list=PLQ8x\_VWW6Akua8I5spV8nHIWlG6\_tX6dx*](https://www.youtube.com/playlist?list=PLQ8x_VWW6Akua8I5spV8nHIWlG6_tX6dx)

ŠETEK, David. React 2 - pojďme vyvíjet pokročilejší aplikace. [online]. YouTube, 2023. Dostupné z:

[*https://www.youtube.com/playlist?list=PLQ8x\_VWW6Aktmp\_v1eYcCBbO5GovP\_sHl*](https://www.youtube.com/playlist?list=PLQ8x_VWW6Aktmp_v1eYcCBbO5GovP_sHl)

ŠETEK, David. React 3 - vícestránkový web a napojení na databázi firebase. [online]. YouTube, 2023. Dostupné z:

[*https://www.youtube.com/playlist?list=PLQ8x\_VWW6AkvURLFWrOZy1e8cZuyx\_xN8*](https://www.youtube.com/playlist?list=PLQ8x_VWW6AkvURLFWrOZy1e8cZuyx_xN8)

Knifestock. *Zdroj obrázků použitých v databázi aplikace*. [online]. Přístup dne 4. 8. 2024. Dostupné z: <https://www.knifestock.cz/>