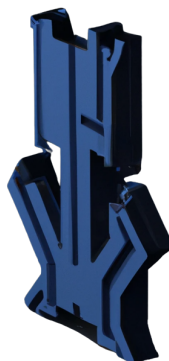


# **ZÁVĚREČNÁ STUDIJNÍ PRÁCE**

## **dokumentace**

### **Webová aplikace na zobrazení 3D modelů pomocí Aframe**



**Autor:** Ondřej Vícha  
**Obor:** 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  
se zaměřením na počítačové sítě a programování  
**Třída:** IT4  
**Školní rok:** 2023/24

## **Poděkování**

Chtěl bych poděkovat panu učiteli Mgr. Markovi Lučnému za užitečné rady při tvorbě projektu.

## **Prohlášení**

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a uvedl veškeré použité informační zdroje.

Souhlasím, aby tato studijní práce byla použita k výukovým a prezentačním účelům na Střední průmyslové a umělecké škole v Opavě, Praskova 399/8.

V Opavě 1. 1. 2024

.....  
Podpis autora

## **Abstrakt**

Tento projekt se zaměřuje na vytvoření webové aplikace, která umožní snadné zobrazení a stahování 3D modelů. Základem tohoto projektu je databáze MySQL a framework Django který nabízí různé vestavěné šablony. Databáze MySQL slouží jako úložiště pro 3D modely a další související informace. Díky kombinaci těchto technologií je cílem vytvořit webovou aplikaci, která bude nejen vizuálně atraktivní, ale také jednoduše ovladatelná pro uživatele, kteří chtějí zobrazovat a stahovat 3D modely.

## **Klíčová slova**

Django, MySQL, 3D, webová stránka, modely, databáze.. ...

# Obsah

<b>Úvod</b>	<b>5</b>
<b>1 Teoretické a metodické základy</b>	<b>6</b>
1.1 Záměr . . . . .	6
1.2 Architektura databázových aplikací . . . . .	6
1.3 Počáteční zkušenosti . . . . .	7
1.4 Využité technologie . . . . .	7
1.4.1 Django . . . . .	7
1.4.2 MySQL . . . . .	7
1.4.3 Bootstrap 4.4.1 . . . . .	8
1.4.4 Aframe.js . . . . .	8
<b>2 Použité postupy při práci</b>	<b>9</b>
2.1 Založení projektu . . . . .	9
2.2 Databázový model . . . . .	9
2.2.1 Vztah modelu s kategoriemi . . . . .	10
2.2.2 Vztah uživatele s profilem . . . . .	10
2.2.3 Vztah modelu a uživatele . . . . .	10
2.3 Autentizace a autorizace . . . . .	11
2.4 Operace s uživatelským účtem . . . . .	12
2.4.1 Úprava profilu . . . . .	12
2.4.2 Změna hesla . . . . .	13
2.5 Přidání modelu . . . . .	14
2.5.1 Filtrace . . . . .	15
2.5.2 Editace . . . . .	16
2.5.3 Zobrazení na stránce . . . . .	17
<b>3 Zobrazení modelu</b>	<b>18</b>
3.1 Aframe . . . . .	18
3.2 Scripty . . . . .	18
3.2.1 Základní zobrazení . . . . .	18
3.2.2 Script pro zmenšení . . . . .	19
3.2.3 Vizualizace . . . . .	19
<b>A Spot diagramy a další</b>	<b>23</b>

# ÚVOD

Cílem bylo sestavení webové aplikace se zobrazením a stáhnutím 3D modelů co nejpřehledněji. Aplikace by měla obsahovat registraci a přihlášení uživatelů pro nahrání 3D modelů i s možností hodnocení modelů a zobrazení jiných uživatelů. Uživatel by měl mít možnost upravit si vlastní profil o různé informace a taky přidat nebo upravit různé informace o vlastních modelech. Všechny modely jsou přehledně na jedné stránce kde je možnost filtrace modelů. Celá webová stránka by měla být responzivní, aby fungovala na jiných zařízeních.

Nepřihlášený uživatel bude mít možnost pouze registrace, přihlášení nebo zobrazení a stáhnutí modelů zatímco přihlášený uživatel bude moci kromě těchto funkcí také hodnotit modely ostatních a také nahrávat a upravovat své vlastní modely.

V rámci projektové dokumentace se zaměřuji na postup tvorby konkrétní aplikace. Začínám analýzou databázového modelu v úvodní části. Následně se věnuji správě uživatelských účtů, implementaci formulářů pro přidání a editaci dat a vytváření pohledů pro prezentaci modelů a uživatelů. V závěrečné části se pak soustředím na design samostatné vizuální stránky a její přizpůsobení pro různá zařízení.

# 1 TEORETICKÉ A METODICKÉ ZÁKLADY

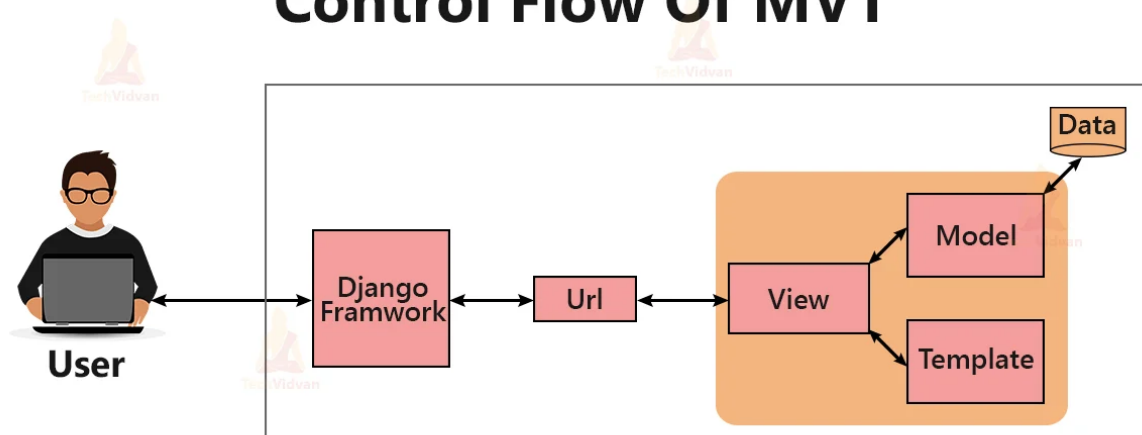
## 1.1 ZÁMĚR

Webové aplikace pro zobrazení 3D modelů poskytují prostor pro interaktivní prezentaci třírozměrných objektů přes internet. Tyto aplikace mohou být využívány v různých odvětvích, včetně designu, vzdělávání, průmyslu, umění a mnoha dalších. Webové aplikace ve kterých je možné si zobrazit 3D modely je několik, ale většina z nich je placená a složitá. To je pár z mála faktorů proč jsem si tenhle projekt vybral udělat.

## 1.2 ARCHITEKTURA DATABÁZOVÝCH APLIKACÍ

Framework Django nabízí 2 architektury. Architektura MVC a MVT. V tomto projektu využívám architekturu MVT protože v případě MVC musí programátoři napsat veškerý kód specifický pro ovládací prvky, zatímco v případě MVT se o část týkající se kontroléru stará samotný framework.

### Control Flow Of MVT



Obrázek 1.1: Logika MVT

Model bude sloužit jako rozhraní pro vaše data. Je zodpovědný za údržbu dat. Je to logická struktura dat, která stojí za celou aplikací, a je reprezentována databází (zpravidla relační databází, jako je MySQL, Postgres).

Zobrazení je uživatelské rozhraní - to, co vidíte v prohlížeči při vykreslování webové stránky. Je reprezentováno soubory HTML/CSS/Javascript.

Šablona se skládá ze statických částí požadovaného výstupu HTML a také z určité speciální syntaxe popisující, jak bude vložen dynamický obsah.

## **1.3 POČÁTEČNÍ ZKUŠENOSTI**

Během třetího ročníku jsem se věnoval tvorbě databázových aplikací v rámci frameworku Django a manipulaci obsahu v databázi MySQL. Zaměřil jsem se na využití generických funkcí a implementaci uživatelského systému v souladu s nativními možnostmi, které Django poskytuje. Cílem bylo vytvořit funkční aplikaci s minimálním úpravami standardních prvků frameworku. Pro správu databáze jsem preferoval nástroj phpMyAdmin, který mi umožnil snadné provádění operací nad databází přes webové rozhraní. Tyto dovednosti vytvořily pevný základ pro mé další projekty v oblasti databázového vývoje.

## **1.4 VYUŽITÉ TECHNOLOGIE**

### **1.4.1 Django**

Django je svobodný a open-source webový framework napsaný v programovacím jazyce Python. Byl vyvinut pro podporu rychlého a efektivního vývoje webových aplikací. Django poskytuje komplexní sadu nástrojů a funkcí, které usnadňují vývoj webových projektů.

### **1.4.2 MySQL**

MySQL je open-source relační databázový systém, který poskytuje efektivní a spolehlivé ukládání a správu dat. Jedná se o jeden z nejpopulárnějších databázových systémů, a to zejména ve světě webového vývoje. MySQL podporuje mnoho jazyků programování, a je běžně používán pro ukládání dat webových aplikací, obchodních systémů, a dalších typů softwaru.

### **1.4.3 Bootstrap 4.4.1**

Bootstrap je open-source framework pro vývoj responsivních a mobilně přizpůsobivých webových stránek. Bootstrap poskytuje sadu nástrojů, stylů a komponent, které zjednodušují a urychlují proces vývoje webových aplikací. Bootstrap je napsán v jazyce HTML, CSS a JavaScript. Verze Bootstrap 4.4.1 je konkrétní verze tohoto frameworku. Každá verze Bootstrap obsahuje aktualizace, opravy chyb a nové funkce oproti předchozím verzím.

### **1.4.4 Aframe.js**

A-Frame je open-source framework pro vytváření a vývoj virtuální reality (VR) a 3D scén na webu. Je postaven na jazyce JavaScript a běží v prohlížeči, což umožňuje vytvářet interaktivní a prostorové zážitky přímo ve webovém prostředí. A-Frame bylo vyvinuto společností Mozilla a je distribuováno pod licencí MIT. A-Frame je vhodný pro tvorbu různých typů projektů, včetně virtuálních prohlídek, interaktivních edukativních aplikací, her a dalších 3D prostorových zážitků na webu.

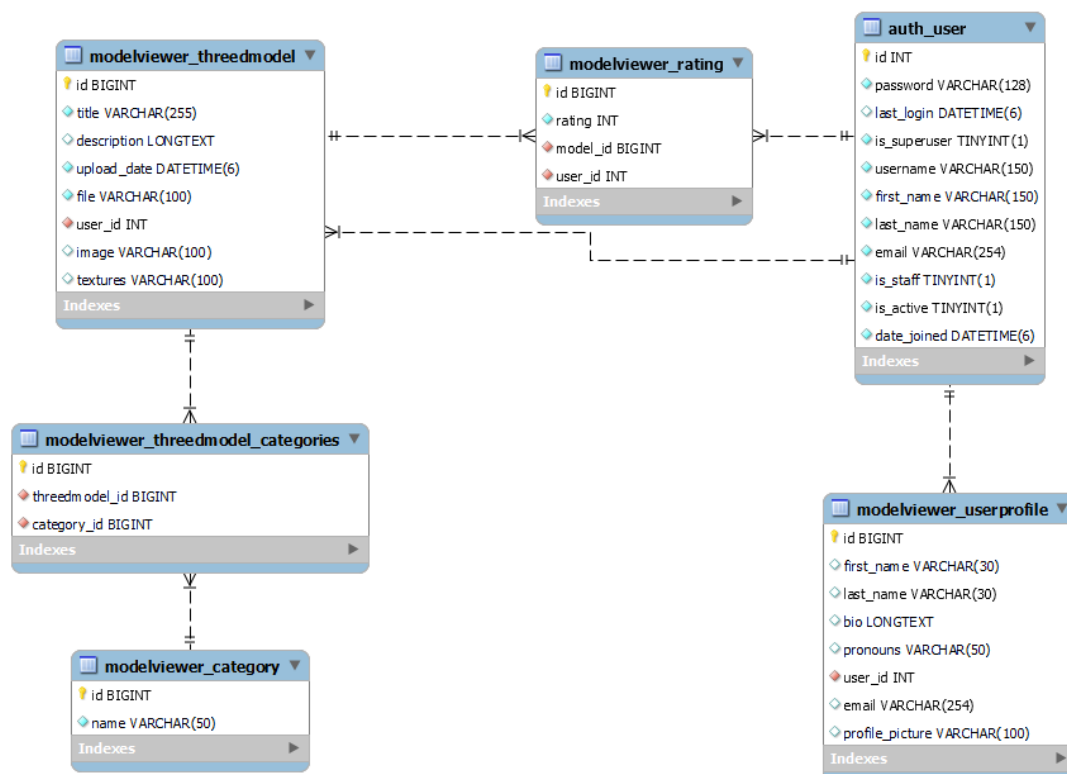


## 2 POUŽITÉ POSTUPY PŘI PRÁCI

### 2.1 ZALOŽENÍ PROJEKTU

Celý projekt byl dělán v Pycharm. Prvním krokem bylo založení si Django projektu který má v sobě vestavěnou pouze SQLite databázi a proto bylo potřeba, po delším uvažování, změnit databázi na MySQL kvůli zabezpečení a funkčnosti. Následně stačilo vytvořit v projektu vlastní aplikaci a měl jsem základní kostru stránky a mohl jsem se pustit do vytváření modelů.

### 2.2 DATABÁZOVÝ MODEL



Obrázek 2.1: ER diagram mé webové aplikace vytvořený pomocí MySQL Workbench

### **2.2.1 Vztah modelu s kategoriemi**

Nejdůležitější je tabulka na nahrání a popis 3D modelů u kterého si i ve vztahu N:1 můžeme vybrat více kategorií pro jeden model díky tabulce která obsahuje cizí klíče a tím funguje jako spojovací tabulka mezi nahráním 3D modelu a přidáním kategorií.

### **2.2.2 Vztah uživatele s profilem**

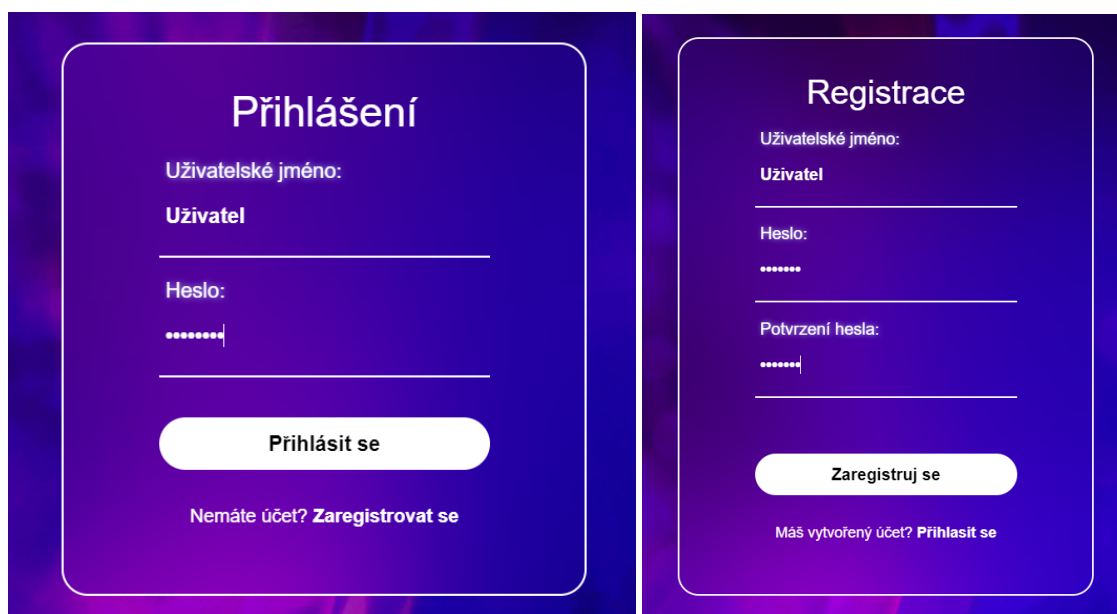
Součástí diagramu je také tabulka uživatele která zajišťuje registraci, přihlášení a pravomoc uživatele. Vztah N:1 zajišťuje aby uživatelé měli jen jeden profil který můžou libovolně upravovat.

### **2.2.3 Vztah modelu a uživatele**

Jeden model může být propojen s více hodnocením ale každé hodnocení může být propojeno pouze s jedním modelem. Modely můžou být propojeny s jedním uživatelem, ale každý uživatel může být propojen pouze s jedním modelem. Jeden uživatel může mít mnoho hodnocení, ale každé hodnocení patří pouze jednomu uživateli.

## 2.3 AUTENTIZACE A AUTORIZACE

Uživatelský systém, který řídí přihlašování a registraci uživatelů, je klíčovým prvkem každé aplikace, a je nezbytné zajistit, aby uživatelé měli přístup pouze k těm akcím, které neporušují integritu a chod aplikace. Pro řešení této problematiky jsem se rozhodl využít vestavěný model uživatele v rámci frameworku Django, konkrétně balíčku `django-contrib-auth-models`. Tento balíček poskytuje robustní implementaci uživatelského modelu nazvaného `User`, který slouží k reprezentaci uživatelů v systému.



The image displays two side-by-side web forms for user authentication, set against a dark blue background with a subtle pattern. The left form, titled 'Přihlášení' (Login), features a white rounded rectangle containing the title, a label 'Uživatelské jméno:' followed by a text input field with the placeholder 'Uživatel', a label 'Heslo:' followed by a password input field with masked characters, a 'Přihlásit se' button, and a link 'Nemáte účet? Zaregistrovat se'. The right form, titled 'Registrace' (Registration), also in a white rounded rectangle, includes the title, a label 'Uživatelské jméno:' followed by a text input field with the placeholder 'Uživatel', a 'Heslo:' label followed by a password input field, a 'Potvrzení hesla:' label followed by a second password input field, a 'Zaregistruj se' button, and a link 'Máš vytvořený účet? Přihlasit se'.

Obrázek 2.2: Formulář na přihlášení a registraci

Při využití modelu `User` z balíčku `django-contrib-auth-models` získávám celou řadu funkcionalit, která usnadňuje správu uživatelů. Tento model obsahuje atributy a metody, které umožňují bezpečné ukládání hesel, ověřování přihlašovacích údajů a mnoho dalšího. Díky této integraci mohu snadno implementovat přihlašování a registraci s minimálním úsilím a s důrazem na bezpečnostní standardy.

Jedním z klíčových prvků je také řízení přístupových práv uživatelů. Django-contrib-auth-models nabízí robustní systém oprávnění, který umožňuje definovat, které části aplikace může uživatel přistupovat a jaké akce může provádět. To zajišťuje, že uživatelé mají pouze oprávnění, která jsou nezbytná pro správný chod aplikace, a minimalizuje riziko možných neoprávněných činností.

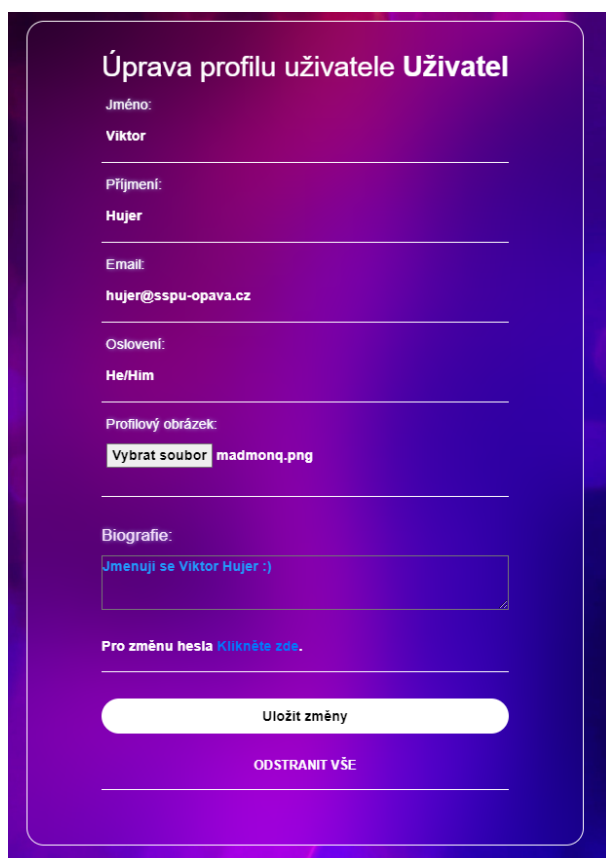
## 2.4 OPERACE S UŽIVATELSKÝM ÚČTEM

Uživatelé jsou umožněny základní operace s jejich účtem.

- Přidání osobních informací
- Přidání profilového obrázku který se zobrazí v seznamu uživatelů
- Přidání emailu
- Změna hesla

Aby k těmto funkcím měl uživatel snadný přístup, vytvořil jsem v navigační liště bloky, které se zobrazí pouze pokud je uživatel přihlášený a odkazují na podstránky řešící dané funkce.

### 2.4.1 Úprava profilu



Úprava profilu uživatele **Uživatel**

Jméno:  
Viktor

Příjmení:  
Hujer

Email:  
hujer@sspu-opava.cz

Oslovení:  
He/Him

Profilový obrázek:  
Vybrat soubor madmonq.png

Biografie:  
Jmenuji se Viktor Hujer :)

Pro změnu hesla [Klikněte zde.](#)

Uložit změny

ODSTRANIT VŠE

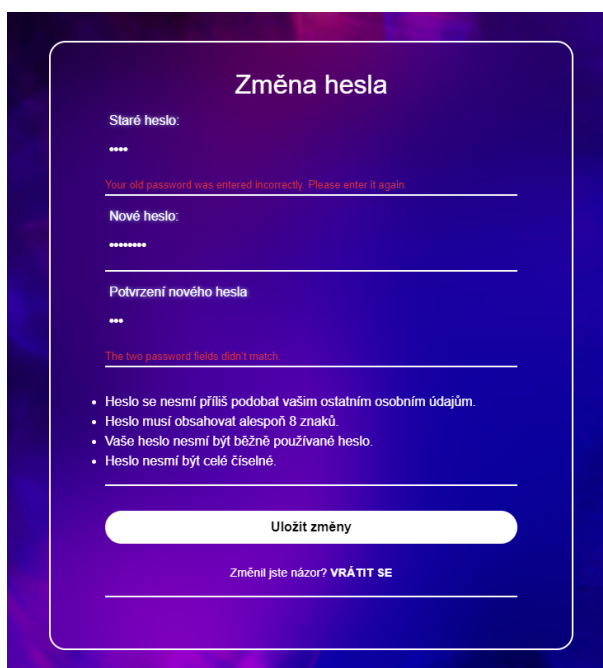
Obrázek 2.3: Formulář úpravu profilu

Tento formulář slouží k shromažďování klíčových informací o uživateli, přičemž zahrnuje jeho osobní údaje, jako jsou jméno, příjmení, e-mailová adresa, preferovaná rodná zaměření, profilový obrázek a stručná biografie. Formulář není pouze prostředkem pro shromažďování

informací, ale také představuje možnost zlepšení uživatelské interakce a začlenění jednotlivců na naší platformě. Kromě toho formulář obsahuje praktickou možnost změny hesla, což přispívá k bezpečnosti účtu uživatele. Odkaz na další formulář o změně hesla umožňuje uživatelům snadno a bezpečně aktualizovat svá přihlašovací údaje podle svých potřeb a preferencí.

## 2.4.2 Změna hesla

Uživatel by měl mít jednoduchý a snadný přístup k formuláři. Toto by mělo být realizováno prostřednictvím jasně označeného odkazu nebo tlačítka ve vhodné části uživatelského rozhraní.



The image shows a web form for changing a password. The title is 'Změna hesla'. It contains three input fields: 'Staré heslo:' (Old password), 'Nové heslo:' (New password), and 'Potvrzení nového hesla' (Confirm new password). Each field has a placeholder of four asterisks. There are two error messages in red text: 'Your old password was entered incorrectly. Please enter it again.' and 'The two password fields didn't match.' Below the fields is a list of password requirements in Czech: 'Heslo se nesmí příliš podobat vašim ostatním osobním údajům.', 'Heslo musí obsahovat alespoň 8 znaků.', 'Vaše heslo nesmí být běžně používané heslo.', and 'Heslo nesmí být celé číselné.' At the bottom, there is a large white button labeled 'Uložit změny' and a link labeled 'Změnil jste názor? VRÁTIT SE'.

Obrázek 2.4: Formulář na změnu hesla

Formulář pro změnu hesla je klíčovým prvkem pro uživatelskou bezpečnost a pohodlí. Při jeho implementaci je důležité dbát na bezpečnostní standardy a poskytnout uživatelům jasný a intuitivní proces změny hesla.

## 2.5 PŘIDÁNÍ MODELU

Uživatel bude mít možnost přidat do systému vlastní 3D model a obohatit jej o různé parametry a podrobnosti, které poskytnou uživatelům lepší představu o daném modelu. Tato možnost umožní uživatelům efektivněji procházet a vyhledávat mezi dostupnými 3D modely, protože budou schopni využít filtry a specifikovat požadované parametry při hledání.

Databáze následně prezentuje tyto modely ostatním uživatelům, kteří mohou prohlížet, hodnotit a stahovat modely dle svých potřeb. Uživatelská interakce s modelem je klíčová, protože umožňuje tvůrcům získat zpětnou vazbu od komunity a vylepšit své modely na základě reálných potřeb uživatelů. Pro uživatele, kteří nahráli model, je k dispozici přehled vlastních modelů, což umožňuje snadný přístup k informacím, aktualizacím a případným budoucím úpravám.

Důležitým a klíčovým prvkem v konfiguraci modelu jsou definované kategorie, které slouží k jeho zařazení do specifických typů. Tímto způsobem můžeme efektivně filtrovat a organizovat modely podle konkrétních charakteristik. V rámci této konfigurace jsou k dispozici několik základních kategorií, které umožňují přesnější a cílenější třídění.

Významným parametrem je také přípona souboru s modelem, která je klíčová pro správnou integraci s Aframe. Pro optimální kompatibilitu a funkčnost je nutné, aby modely používaly přípony `.glb` nebo `.gltf`, neboť Aframe pracuje výhradně s těmito typy souborů. Tím je zajištěno plynulé a bezproblémové propojení modelu s danou platformou.

Kromě technických specifikací je také důležité pamatovat na estetický a vizuální prvek prezentace projektu. Fotka, která slouží k prezentaci modelu, hraje klíčovou roli při zaujímání uživatelů. Tato fotografie je také zobrazena v náhledu modelu pro všechny uživatele, což přispívá k atraktivitě a přehlednosti prezentovaného obsahu.

## 2.5.1 Filtrace

Vytvoření funkcí ve `views.py` poskytuje uživateli široké možnosti filtrování a řazení modelů v aplikaci. S pomocí těchto funkcí může uživatel přesně specifikovat požadovaná kritéria pro nalezení konkrétních modelů.



Obrázek 2.5: Filtrování

```
user_uploaded_models = ThreeDModel.objects.filter(user=request.user)
categories = Category.objects.all()

category_id = request.GET.get('category')
upload_date = request.GET.get('upload_date')
model_name = request.GET.get('model_name')
sort_by = request.GET.get('sort_by')

if category_id:
    user_uploaded_models = user_uploaded_models.filter(categories__id=category_id)

if upload_date:
    user_uploaded_models = user_uploaded_models.filter(upload_date=upload_date)

if model_name:
    user_uploaded_models = user_uploaded_models.filter(Q(title__icontains=model_name))

if sort_by == 'newest':
    user_uploaded_models = user_uploaded_models.order_by('-upload_date')
elif sort_by == 'oldest':
    user_uploaded_models = user_uploaded_models.order_by('upload_date')
return render(request, 'user/profile.html', {'user_uploaded_models':
    user_uploaded_models, 'categories': categories})
```

Tahle funkce v `views.py` zajistí filtraci v zobrazení vlastních modelů ...

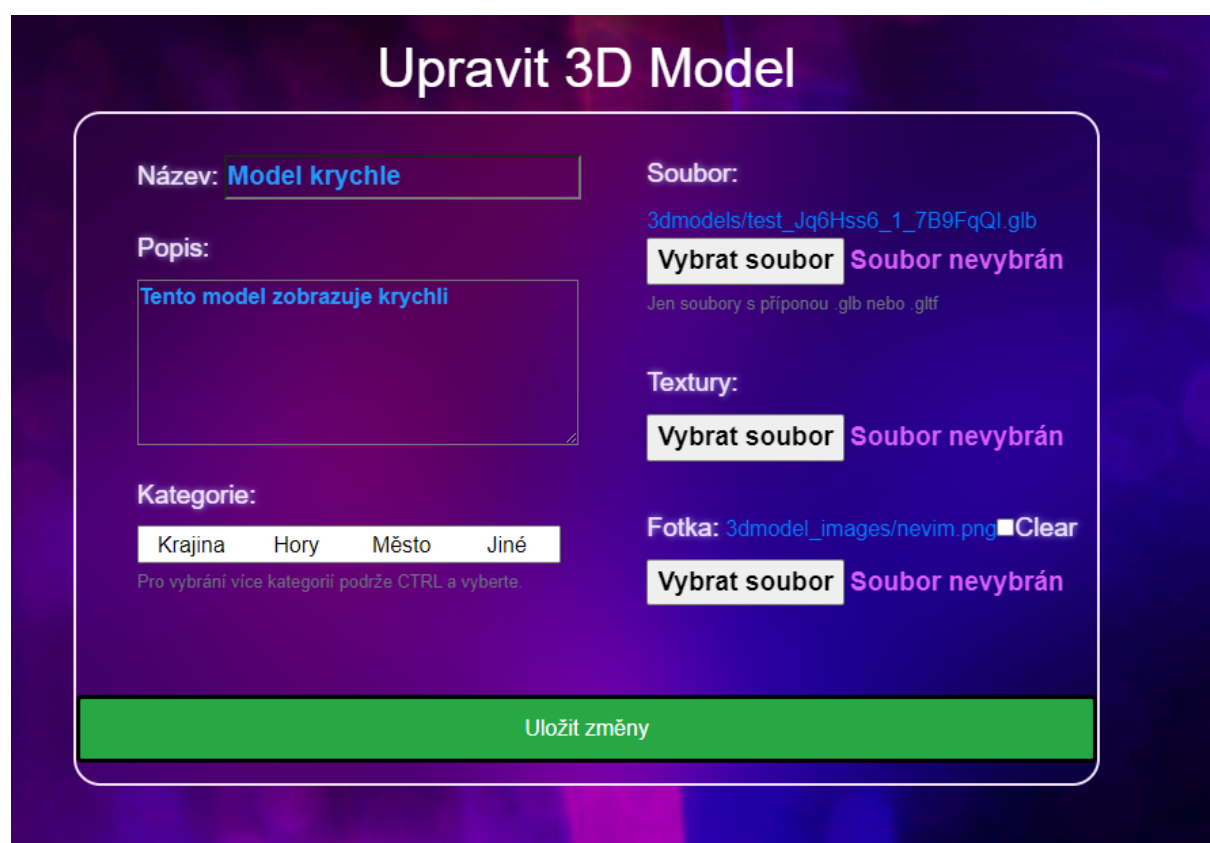
Filtraci lze provádět podle různých parametrů, včetně jména modelu, data nahrání a kategorií. To umožňuje uživateli rychle a efektivně najít požadované modely v rozsáhlé databázi. Díky této funkcionalitě má uživatel možnost upravit zobrazení výsledků podle svých individuálních preferencí a potřeb.

Kromě toho je uživatel schopen řadit modely podle data nahrání, a to buď od nejstarších po nejnovější, nebo naopak. Tato možnost řazení umožňuje uživateli organizovat výsledky ve způsobu, který nejlépe odpovídá jeho požadavkům a usnadňuje rychlé porovnání modelů.

Celkově vzato, díky implementaci těchto funkcí ve views.py uživatel získává vysokou flexibilitu a přesnost při procházení a vyhledávání modelů v aplikaci. To zlepšuje uživatelskou zkušenost a umožňuje snadné a intuitivní používání aplikace pro správu a prohlížení modelů.

## 2.5.2 Editace

Uživateli je poskytnuta jedinečná příležitost upravit svůj nahraný model dle svých představ a potřeb. Pro usnadnění této úpravy mu je zpřístupněn interaktivní formulář, ve kterém má možnost měnit a přizpůsobovat různé části modelu podle svých specifických požadavků.



**Upravit 3D Model**

Název:

Popis:

Kategorie: ☐ Krajina ☐ Hory ☐ Město ☐ Jiné  
Pro vybrání více kategorií podržte CTRL a vyberte.

Soubor:   
 **Soubor nevybrán**  
Jen soubory s příponou .glb nebo .gltf

Textury:  **Soubor nevybrán**

Fotka:    
 **Soubor nevybrán**

Obrázek 2.6: Formulář editace

Tento formulář zobrazuje uživateli obsah, který měl původně nahraný v různých částech modelu, takže má jasný přehled o tom, co již bylo vytvořeno. Každá část, kterou uživatel rozhodne upravit, je snadno dostupná pro editaci. Může měnit text, parametry a další klíčové prvky podle svého uvážení.

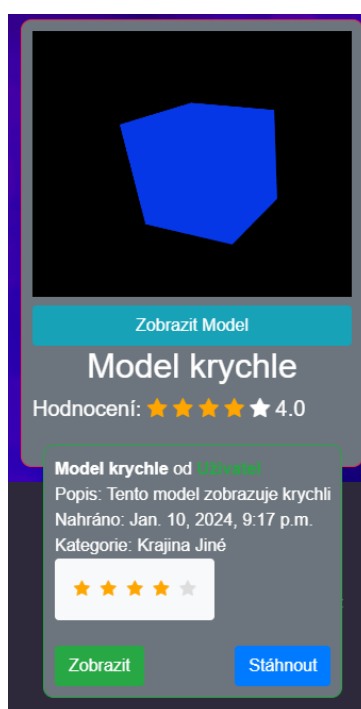
Tato interaktivní možnost úpravy umožňuje uživateli vylepšit a personalizovat svůj model, aby co nejlépe odpovídal jeho potřebám a preferencím. Tímto způsobem má uživatel plnou



kontrolu nad tím, jakým směrem se vyvíjí jeho vlastní model, a může ho dále zdokonalovat podle svých aktuálních požadavků.

### 2.5.3 Zobrazení na stránce

Preview modelu je způsobem prezentace, který využívá karty k zobrazení klíčových informací o každém modelu. Každá karta obsahuje vizuální obrázek, název a průměrné hodnocení daného modelu, což umožňuje uživatelům rychle získat přehled o dostupných modelech. Při najetí myší na konkrétní kartu se automaticky zobrazí podrobnější informace, což zahrnuje údaje jako jméno uživatele, který model nahrál, popis samotného modelu a datum, kdy byl model nahrán do systému.



Obrázek 2.7: Zobrazení modelů

Tato funkce nabízí uživatelům příležitost získat hlubší vhled do každého modelu, a to vše na jednom uživatelském rozhraní. Kromě toho je implementována funkce, která umožňuje hodnocení modelu 1-5 hvězdičkami, pokud uživatel, který hodnotí, není tím samým tvůrcem modelu. Tato bezpečnostní opatření zajistí, že hodnocení jsou poskytována pouze uživateli, u kterých není střet zájmů a mohou poskytnout objektivní zpětnou vazbu.

## 3 ZOBRAZENÍ MODELU

### 3.1 AFRAME

Zobrazení modelů bylo uděláno pomocí Aframe, tahle jednoduchá možnost nabízí vytvořit zobrazení modelů z obyčejného souboru HTML, aniž by bylo nutné cokoli instalovat. Jediná důležitá věc je přidat do svojí stránky nebo aplikace script s odkazem na knihovnu Aframe a pak už můžeme upravovat a přidávat věci podle vlastního uvážení.

```
<script src="https://aframe.io/releases/1.5.0/aframe.min.js"></script>
```

### 3.2 SCRIPTY

#### 3.2.1 Základní zobrazení

Základní kód na zobrazení modelu, který zatím řeší pouze jednoduché textury i s scriptem který zmenší velikost modelu tak aby nebyl příliš veliký v zobrazení:

```
<a-scene class="aframebox" embedded>
  <a-assets>
    <a-asset-item id="model" src="{_model.file.url_}"></a-asset-item>
  </a-assets>
  <a-light type="ambient" position="1_1_1" intensity="1"></a-light>
  <a-entity gltf-model="#model" modify-materials
    max-dimensions position="0_0_0" rotation="0_0_0"/>
  <a-camera id="camera" position="5_1.5_5" rotation="0_0_0" wasd-controls>
    <a-cursor></a-cursor>
  </a-camera>
</script>
</script>
</a-scene>
```

Tenhle kód zajistí pozici, kameru a základní funkci zobrazení konkrétního modelu, další funkce kterými chceme zlepšit interaktivitu modelu píšeme do <script> <script>...

### 3.2.2 Script pro zmenšení

```
AFRAME.registerComponent('max-dimensions', {
  init: function () {
    var maxDimensions = { x: 1, y: 1, z: 1 };
    this.el.addEventListener('model-loaded', function () {
      var modelDimensions = this.getObject3D(
        'model').geometry.boundingBox.getSize();
      var scaleFactor = Math.min(
        maxDimensions.x / modelDimensions.x,
        maxDimensions.y / modelDimensions.y,
        maxDimensions.z / modelDimensions.z
      );
      this.setAttribute('scale', scaleFactor
        + ' ' + scaleFactor + ' ' + scaleFactor);
    });
  }
});
```

### 3.2.3 Vizualizace

## ZÁVĚR

Věřím, že jsem ti spolu se šablonou poskytl několik tipů, jak napsat práci. At' už jde o úplné začátky s  $\text{\LaTeX}$ em. Či ukázkou toho, co vše s ním zvládneš. Pokud bys měl k šabloně libovolné dotazy, rouhodně se na mě obrať.  $\text{\LaTeX}$ tvé práci dodá určitou krásu, tak doufám, že ti dodá sebevědomí a uspěješ při soutěži. A i kdyby ne vzpomeň si, kolik ses toho musel naučit a hned uvidíš o jaký kus ses posunul.

## LITERATURA

- [1] DOKULIL Jakub. *Šablona pro psaní SOČ v programu L<sup>A</sup>T<sub>E</sub>X* [Online]. Brno, 2020 [cit. 2020-08-24]. Dostupné z: [https://github.com/Kubiczek36/SOC\\_sablona](https://github.com/Kubiczek36/SOC_sablona)
- [2] OETIKER, Tobias, Hubert PARTL, Irene HYNA, Elisabeth SCHEGL, Michal KOČER a Pavel SÝKORA. *Ne příliš stručný úvod do systému LaTeX2e* [online]. 1998 [cit. 2020-08-24]. Dostupné z: <https://www.jaroska.cz/elearning/informatika/typografie/lshort2e-cz.pdf>
- [3] *Wikibooks: LaTeX* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2020-08-24]. Dostupné z: <https://en.wikibooks.org/wiki/LaTeX>
- [4] *TeX - LaTeX Stack Exchange* [online]. Stack Exchange, 2020 [cit. 2020-09-01]. Dostupné z: <https://tex.stackexchange.com>
- [5] *Střední škola průmyslová a umělecká Opava* [online]. [cit. 2023-11-11]. Dostupné z: <https://www.sspu-opava.cz>
- [6] *Citace PRO* [online]. Citace.com, 2020 [cit. 2020-08-31]. Dostupné z: <https://www.citacepro.com>
- [7] BORN, Max a Emil WOLF. *Principles of optics: electromagnetic theory of propagation, interference and diffraction of light*. 7th (expanded) edition. Reprinted with corrections 2002. 15th printing 2019. Cambridge: Cambridge University Press, 2019. ISBN 978-0-521-64222-4.

## Seznam obrázků

1.1	Logika MVT . . . . .	6
2.1	ER diagram mé webové aplikace vytvořený pomocí MySQL Workbench . . . .	9
2.2	Formulář na přihlášení a registraci . . . . .	11

2.3	Formulář úpravu profilu . . . . .	12
2.4	Formulář na změnu hesla . . . . .	13
2.5	Filtrování . . . . .	15
2.6	Formulář editace . . . . .	16
2.7	Zobrazení modelů . . . . .	17

## Seznam tabulek

## **PŘÍLOHA A SPOT DIAGRAMY A DALŠÍ**