

DELTA – Střední škola informatiky a ekonomie, s.r.o. Ke Kamenci 151, Pardubice

TRADER

Platforma pro obchodování s 3D modely

Příjmení, jméno	Grafek Filip
Studijní obor	Informační technologie
Třída:	4B
Školní rok	2024/2025

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení: *Filip Grafek*
Pro školní rok: *2025/2026*
Třída: *4. B.*
Obor: *Informační technologie 18-20-M/01*

Téma práce: *Komunitní e-shop 3D modelů a jejich textur*
Vedoucí práce: *ak. mal. Daniel Václavík*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem maturitního projektu je vytvoření *komunitního e-shopu zaměřeného na prodej 3D modelů a textur*. E-shop bude zprostředkovávat prodej 3D modelů a bude současně fungovat i jako cloudové úložiště jednotlivých tvůrců.

- Logo projektu
- Grafický návrh a design stránky
- Vytvoření databáze cloudu
- Implementace addonu Three.js pro 3D rozhraní
- Přidání platební brány
- Marketing projektu

Součástí práce bude i dokumentace k návrhům, postupům a popis použitých programů a technologií.

Rozsah práce:

- Tištěná dokumentace /min. 25 – 35 normostran/
- Grafické řešení projektu
- Implementace databáze a cloudu
- Programování webové aplikace
- Realizace marketingu na sociálních sítích

Hodnocení:

Projekt bude hodnocen na základě odborné úrovně, celkového provedení, co se týká programátorské stránky, jazykové kultury a úspěšnosti prezentace
Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- Září: Hotový grafický design
- Říjen: Implementace Three.js a cloudu
- Listopad až prosinec: Programování projektu

- Leden: Implementace platební brány
- Únor: Řešení legislativy a marketingu
- Březen: Dokončení písemné části a finalizace projektu

Prohlášení

Prohlašuji, že jsem maturitní projekt vypracoval(a) samostatně, výhradně s použitím uvedené literatury.

V Pardubicích

Podpis:

Poděkování

Upřímně děkuji vedoucímu mého maturitního projektu ak. mal. Danielovi Václavíkovi za odborné vedení, trpělivost a cenné připomínky při zpracování tohoto maturitního projektu.

Dále děkuji DELTA – Střední škole informatiky a ekonomie, s.r.o. za poskytnuté zázemí, technické vybavení a kvalitní vzdělání v oboru informačních technologií, které mi umožnilo projekt realizovat.

Anotace

Tato maturitní práce se zabývá návrhem a implementací webové aplikace Trader, která slouží jako komunitní eshop s 3D modely. Cílem aplikace je umožnit uživatelům sdílet, prodávat a nakupovat 3D modely v přehledném a bezpečném prostředí, přičemž využívá cloudové úložiště S3 pro správu souborů a knihovnu Three.js pro interaktivní 3D vizualizaci modelů přímo v prohlížeči. Trader řeší potřebu centralizovaného tržiště pro 3D modely, podporuje komunitní interakci a nabízí funkce jako správa uživatelských profilů, obchodování, notifikace a bezpečné transakce.

Aplikace je postavena jako full-stack řešení s využitím moderních technologií: Next.js a TypeScript na frontendu, Prisma ORM s databází PostgreSQL na backendu, integrací S3 pro ukládání souborů a knihovnou Three.js pro zobrazení 3D modelů. Uživatelské rozhraní je navrženo s důrazem na responzivitu a uživatelskou přívětivost pomocí Tailwind CSS.

Práce zahrnuje analýzu existujících tržišť s 3D modely, návrh datového modelu, implementaci klíčových funkcí včetně správy modelů, obchodních transakcí, notifikací, uživatelských účtů, bezpečné autentifikace a interaktivního zobrazení 3D modelů. Součástí je také testování a závěrečné zhodnocení projektu.

Klíčová slova

webová aplikace, komunitní eshop, 3D modely, Next.js, TypeScript, Prisma, PostgreSQL, S3, Three.js, Tailwind CSS, cloudové úložiště, obchodování, notifikace, vizualizace, full stack, maturitní project

Abstract

This graduation project focuses on the design and implementation of Trader, a community-driven e-shop for 3D models. The application enables users to share, sell, and purchase 3D models in a secure and user-friendly environment, utilizing S3 cloud storage for file management and the Three.js library for interactive 3D visualization of models directly in the browser. Trader addresses the need for a centralized marketplace for 3D models, supports community interaction, and offers features such as user profile management, trading, notifications, and secure transactions.

The application is built as a full-stack solution using modern technologies: Next.js and TypeScript on the frontend, Prisma ORM with a PostgreSQL database on the backend, S3 integration for file storage, and Three.js for 3D model rendering. The user interface is designed with an emphasis on responsiveness and usability using Tailwind CSS.

The project includes an analysis of existing 3D model marketplaces, data model design, implementation of key features including model management, trading transactions, notifications, user accounts, secure authentication, and interactive 3D model visualization, as well as testing and final project evaluation.

Keywords

web application, community e-shop, 3D models, Next.js, TypeScript, Prisma, PostgreSQL,

S3, Three.js, Tailwind CSS, cloud storage, trading, notifications, visualization, full stack, graduation project

Obsah

1. Úvod.....	1
2. Popis projektu.....	1
2.1 Cíle a funkčnost.....	1
2.2 Uživatelské role.....	1
2.3 Use Case diagram	1
3. Technologie a architektura.....	3
3.1 Přehled technologií.....	3
3.2 Architektura aplikace	3
3.3 Adresářová struktura	4
4. Databázový model.....	5
4.0 Class diagram.....	5
4.1 Tabulka User	5
4.2 Tabulka Model.....	6
4.3 Tabulka OffModel.....	6
4.4 Další tabulky.....	7
5. API rozhraní	8
5.1 Autentizace	8
5.2 Správa modelů.....	8
5.3 Obchodování (OffModels)	10
5.4 Nákupy a platby	10
5.5 Přátelé, dary a oznámení.....	10
6. Komponenty frontendu	12
6.1 Stránky (Pages).....	12
6.2 Klíčové komponenty	12
6.3 Three.js integrace.....	13
7. Systém autentizace	14
7.1 Registrace a přihlášení.....	14
7.2 Autorizace API.....	14
7.3 Odhlášení.....	14
8. Obchodní systém a platby	14
8.1 Tok nákupu	14
8.2 Stripe integrace.....	14

8.3 S3 logika při nákupu	16
9. Systém přátel a oznámení	17
9.1 Přátelství.....	17
9.2 Dar modelů	17
9.3 Oznámení.....	17
10. Nasazení a konfigurace.....	18
10.1 Požadavky.....	18
10.2 Proměnné prostředí (.env.local).....	18
10.3 Instalace a spuštění.....	18
11. Bezpečnost	19
12. Závěr	19

1. Úvod

Trader je webová platforma určená pro nahrávání, správu a obchodování s 3D modely ve formátu GLB/GLTF. Systém umožňuje uživatelům vytvářet osobní knihovny 3D modelů, sdílet vyrenderované obrázky, navázat sociální kontakty a obchodovat s modely prostřednictvím integrace platního brány Stripe.

Projekt vznikl jako maturitní projekt v oboru informatiky s cílem prokázat znalosti moderních webových technologií, návrhu databáze, cloudové infrastruktury a tvorby plnohodnotné full-stack aplikace.

2. Popis projektu

2.1 Cíle a funkčnost

Platforma Trader nabízí následující hlavní funkcionalitu:

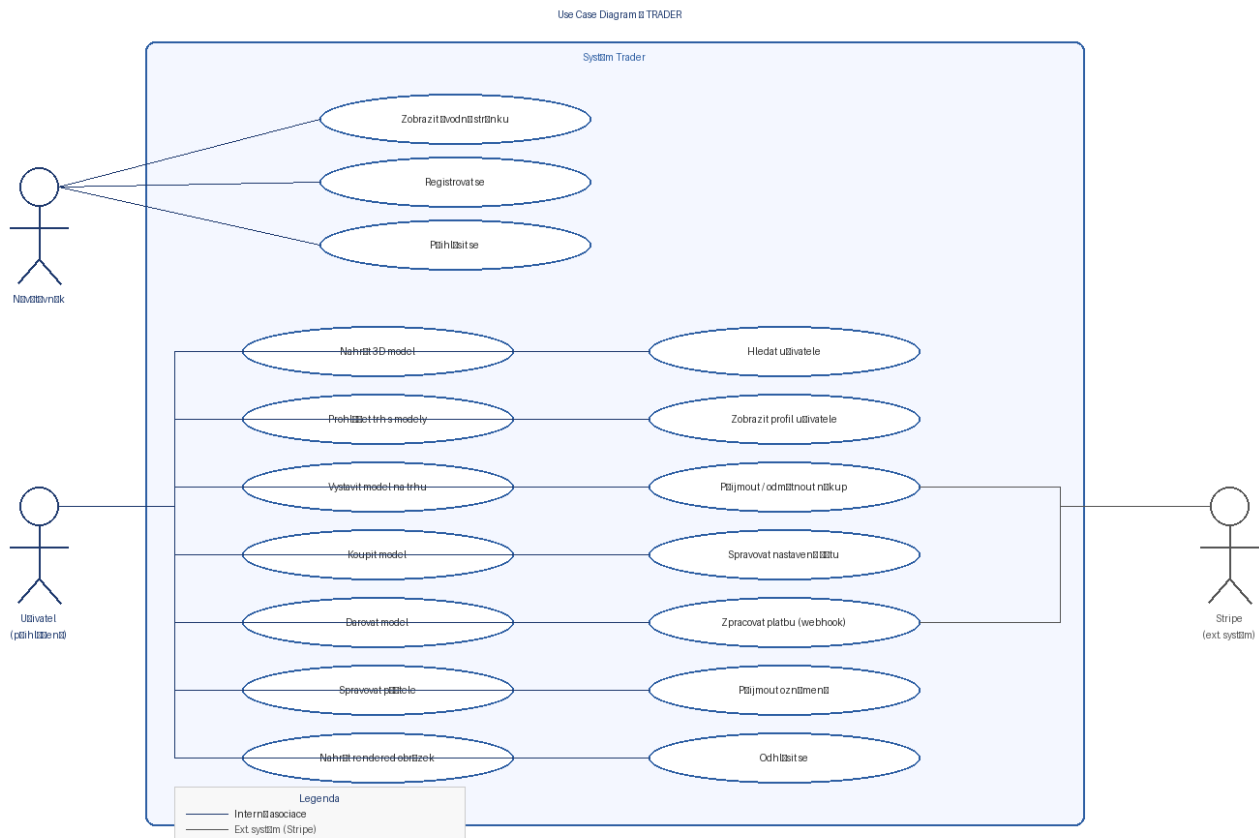
- Nahrávání a správa 3D modelů (formáty GLB, GLTF, OBJ, FBX) s uložením v Amazon S3
- Interaktivní prohlížení modelů pomocí Three.js s ovládáním kamerou (OrbitControls)
- Obchodování s modely – vystavování modelů na trhu s cenami, kategoriemi a popisem
- Platní systém integrovaný prostřednictvím Stripe Checkout
- Nahrávání a správa rendered obrázků (PNG, JPEG a další formáty)
- Systém přátel – posílání žádostí o přátelství, přijímání a rušení
- Dary modelů – možnost posílání 3D modelů jiným uživatelům jako dar
- Oznámení v reálném čase – notifikace o přátelích, darech, nákupech
- Profily uživatelů s ukázkou jejich modelů a rendered obrázků
- Vyhledávání uživatelů podle přezdívky

2.2 Uživatelské role

Systém rozlišuje jediný typ uživatele (autentizovaný), který může plnit jak roli kupujícího, tak roli prodávajícího. Neautentizovaní návštěvníci vidí pouze úvodní stránku.

2.3 Use Case diagram

Následující diagram znázorňuje hlavní případy užití systému Trader pro jednotlivé typy aktérů.



Obr. 1 – Use Case diagram systému Trader

3. Technologie a architektura

3.1 Přehled technologií

Vrstva	Technologie	Verze / poznámka
Frontend	Next.js (App Router)	v16.2.1
Jazyk	TypeScript	v5.x
Stylování	Tailwind CSS + inline styles	v4
3D vykreslování	Three.js + GLTFLoader + OrbitControls	v0.181
Backend / API	Next.js Route Handlers (API routes)	App Router
ORM	Prisma	v6.19
Databáze	PostgreSQL	dle hostitele
Cloudové úložiště	Amazon S3	AWS SDK v3
Autentizace	HTTP-only cookies (JSON payload)	vlastní implementace
Hesla	bcryptjs	v3
Platby	Stripe Checkout + Webhooks	v21
Dodání	Vercel / Node.js ≥ 20.9	

3.2 Architektura aplikace

Aplikace využívá architektonický vzor Full-Stack monorepo s Next.js App Routerem. Vrstvy komunikují následovně:

- Klientské komponenty (React) volá REST API endpointy přes `fetch()`.
- API Route Handlers (v složce `/app/api/`) komunikují s PostgreSQL databází prostřednictvím Prisma ORM.
- Soubory (3D modely, obrázky) jsou uloženy v Amazon S3 a přístupné přes předpodepované URL (Presigned URL s platností 1 hodina).
- Nahrávání probíhá přímo z prohlížeče na S3 pomocí presigned PUT URL (obdán serverů a zrychlení).

3.3 Adresářová struktura

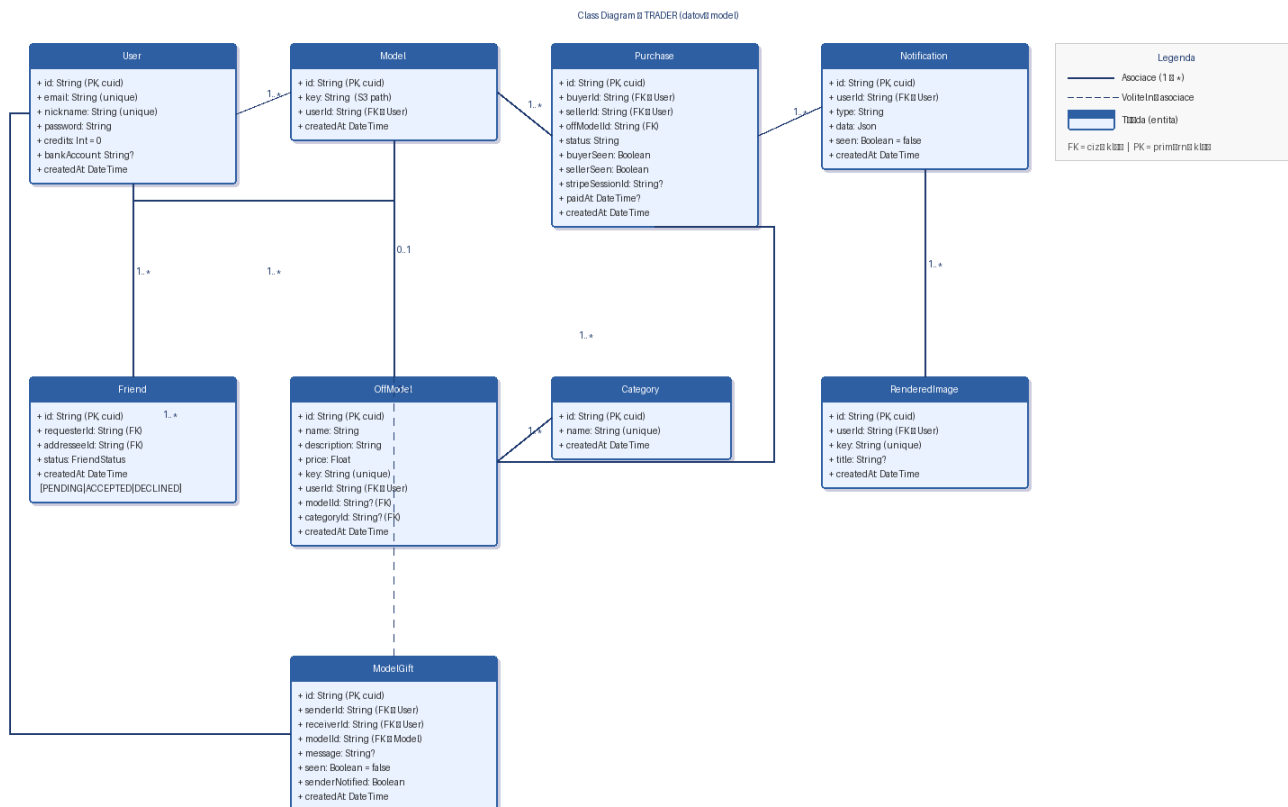
Složka / soubor	Popis
/app/api/	Všechny backend API endpointy (Route Handlers)
/app/dashboard/	Stránky obchodního trhu (Trading)
/app/lobby/	Stránky osobní knihovny modelů
/app/profile/[nickname]/	Profil uživatele (dynamická trasa)
/app/settings/	Nastavení účtu
/app/authPage/	Přihlašovací a registrační stránka
/components/	Sdílené React komponenty
/lib/prisma.ts	Singleton Prisma klient
/lib/S3.ts	Singleton AWS S3 klient
/prisma/schema.prisma	Schéma databáze
/prisma/migrations/	Migrační soubory databáze

4. Databázový model

Databáze PostgreSQL je popsána pomocí Prisma schématu. Systém obsahuje následující tabulky:

4.0 Class diagram

Následující class diagram znázorňuje všechny entity databáze a jejich vztahy dle Prisma schématu.



Obr. 2 – Class diagram (datový model) systému Trader

4.1 Tabulka User

Hlavní entita reprezentující uživatele systému.

Pole	Typ	Popis
id	String (CUID)	Primární klíč
email	String (unique)	Unikátní e-mail
nickname	String (unique)	Unikátní přezdívka

Pole	Typ	Popis
password	String	Hash hesla (bcrypt)
credits	Int	Interní kredity (default 0)
bankAccount	String?	IBAN účtu pro příjem plateb
createdAt	DateTime	čas registrace

4.2 Tabulka Model

Reprezentuje 3D model uložený v Amazon S3.

Pole	Typ	Popis
id	String (UUID)	Primární klíč
userId	String (FK)	Vlastník modelu
key	String	Klíč objektu v S3 (cesta k souboru)
createdAt	DateTime	čas nahrání

4.3 Tabulka OffModel

Model vystavovaný k obchodování („výloha“). Unikátní klíč zajistí, že jeden soubor v S3 může mít jen jedno aktivní nabídkové vypsání.

Pole	Typ	Popis
id	String (UUID)	Primární klíč
userId	String (FK)	Prodávající
key	String (unique)	S3 klíč (shoduje se s Model.key)
name	String	Název nabídky
description	String	Popis nabídky
price	Float	Cena v EUR
categoryId	String? (FK)	Volitelná kategorie
modelId	String? (FK)	Odkaz na Model záznam
createdAt	DateTime	čas vypsání

4.4 Další tabulky

Tabulka	Popis
Category	Kategorie pro třídění OffModel (unikátní název)
Purchase	Záznam o nákupu – stav (PENDING/ACCEPTED/DECLINED), Stripe session, časy
RenderedImage	Rendered obrázky uživatele uložené v S3
Friend	Přátelství – obousměrné se stavy PENDING/ACCEPTED/DECLINED
ModelGift	Dar modelu jinému uživateli s volitelným vzkazem
Notification	Obecné notifikace (např. zrušení přátelství)

5. API rozhraní

Všechna API jsou implementována jako Next.js Route Handlers v adresáři /app/api/. Autentizace probíhá přes HTTP-only cookie „user“ obsahující JSON s id a nickname uživatele.

5.1 Autentizace

Endpoint	Metoda	Popis
POST /api/signUp	POST	Registrace nového uživatele (nickname, email, heslo)
POST /api/login	POST	Přihlášení emailem nebo přezdívkou
GET /api/getUserId	GET	Vrátí ID aktuálního uživatele z cookie
GET /api/settings	GET	Data uživatele (bez hesla)
PUT /api/settings	PUT	Aktualizace profilu, hesla nebo IBAN
POST /api/findUser	POST	Vyhledání uživatele dle emailu nebo přezdívky
GET /api/searchUsers	GET	Full-text hledání přezdívek (?q=)

5.2 Správa modelů

Endpoint	Metoda	Popis
POST /api/getUploadUrl	POST	Generuje presigned PUT URL pro nahrání do S3
POST /api/addObject	POST	Uloží záznam modelu do DB po nahrání
GET /api/getMyModels	GET	Seznam modelů přihlášeného uživatele s presigned URL
GET /api/models?key=	GET	Detail jednoho modelu dle S3 klíče
DELETE /api/models?key=	DELETE	Smaže model ze S3 i z DB
GET /api/getModelById?id=	GET	Model dle interního ID
GET /api/getSignedUrl?key=	GET	Presigned GET URL pro přehled

5.3 Obchodování (OffModels)

Endpoint	Metoda	Popis
POST /api/offModelTrade	POST	Vypsání modelu na trh (vloží OffModel záznam)
GET /api/getOffModels	GET	Seznam všech nabídek (s filtrem categoryId)
GET /api/offModels?key=	GET	Detail nabídky dle S3 klíče
DELETE /api/offModels?key=	DELETE	Odebere model z trhu
POST /api/updateTrade	POST/PUT	Aktualizace názvu, popisu, ceny nabídky
GET /api/getCategories	GET	Seznam všech kategorií
POST /api/getCategories	POST	Vytvoří nebo vyhledá kategorii dle názvu

5.4 Nákupy a platby

Endpoint	Metoda	Popis
POST /api/purchase	POST	Kupující iniciuje nákup (stav PENDING)
PUT /api/purchase	PUT	Prodávající přijme nebo odmítne nákup
POST /api/stripe/checkout	POST	Vytvoří Stripe Checkout Session
POST /api/stripe/webhook	POST	Webhook handler – přijme platbu a přesune model
GET /api/seller/bankAccount?userId=	GET	Ověří, zda má prodávající nastaven IBAN

5.5 Přátelé, dary a oznámení

Endpoint	Metoda	Popis
POST /api/friends	POST	Pošle žádost o přátelství

Endpoint	Metoda	Popis
GET /api/friends	GET	Seznam přátel nebo stav s konkrétním uživatelem (?checkId=)
DELETE /api/friends	DELETE	Zruší přátelství a pošle oznámení
PUT /api/friends/respond	PUT	Přijme nebo odmítně žádost
POST /api/giftModel	POST	Připraví dar (vrátí presigned URL pro kopírování)
POST /api/giftModel/confirm	POST	Potvrdí dar po nahrání souboru příjemci
GET /api/notifications	GET	Soupis všech aktivních oznámení
POST /api/notifications/seen	POST	Označí oznámení jako zobrazeno

6. Komponenty frontendu

6.1 Stránky (Pages)

Cesta	Popis
/	Hlavní marketingová stránka s popisem platformy
/authPage	Přihlášení a registrace (postupný formulář)
/lobby	Osobní knihovna modelů a rendered obrázků
/lobby/addObject	Formulář pro nahrání 3D modelu
/lobby/addRenderedImage	Formulář pro nahrání obrázku
/lobby/addObjectTrade	Nastavení a vypsání modelu na trh
/lobby/objectViewPage?key=	Detail modelu s 3D prohlížečem a správou
/dashboard	Trh s dostupnými nabídkami (s filtrem kategorií)
/dashboard/offModelViewPage?key=	Detail nabídky s možností nákupu
/dashboard/purchaseSuccess	Potvrzovací stránka po úspěšné platbě
/profile/[nickname]	Veřejný profil uživatele (modely + obrázky)
/profile/[nickname]/sendModel	Výběr a odeslání daru
/settings	Nastavení profilu, hesla a bankovního účtu

6.2 Klíčové komponenty

Komponenta	Popis
MyObjects	Mřížka s 3D modely a rendered obrázky v osobní knihovně s filtrováním a lightboxem
OffModels	Mřížka nabídek na trhu s postranním panelem kategorií
ObjectView (lobby)	Plnoformátový 3D prohlížeč s OrbitControls a správou nabídky
ObjectView (dashboard)	Nákupní pohled na nabídku s víceetapovým nákupním tokem
NotificationBell	Zvonek s počtem nesmítených oznámení a panelem akcí
SearchBar	Fulltext hledání uživatelů s dropdown výsledky

Komponenta	Popis
ProfileCard	Profil uživatele s přáteli, modely a rendered obrázky
SendModelPage	Výběr modelu pro dar s XHR progress barem
FriendsList	Rozbalovací seznam přijatých přátel s navigací na profily
SettingsPage	Tabové nastavení (Profil / Bezpečnost / Platby)
Auth	Postupný přihlašovací formulář s typewriter efektem
PurchaseSuccess	Potvrzovací stránka s auto-přesměrováním do knihovny

6.3 Three.js integrace

Pro vykreslování 3D modelů se používá knihovna Three.js s GLTFLoader a OrbitControls. Každý model má vlastní WebGLRenderer, scenu a kameru, které jsou správně likvidovány při odstranění komponenty. Pro detail modelu jsou zapnuté OrbitControls pro plnou interakci (rotace, zoom, přesun).

7. Systém autentizace

7.1 Registrace a přihlášení

Systém implementuje vlastní autentizaci bez externích knihoven (bez NextAuth apod.).
Postup:

- Uživatel zadá e-mail nebo přezdívku – systém ověří existenci účtu.
- Heslo je hashováno pomocí bcryptjs (salt rounds = 10) před uložením do DB.
- Po úspěšném přihlášení je nastaven HTTP-only cookie „user“ s hodnotou {id, nickname}.
- Cookie má platnost 7 dní a je nastavená jako Secure v produkci.

7.2 Autorizace API

Každý chráněný endpoint parsuje cookie z příchozího requestu a verifikuje existenci uživatele v databázi. Při chybějící nebo neplatné cookie je vrácen HTTP status 401 Unauthorized.

7.3 Odhlášení

Odhlášení se provádí na straně klienta nastavením cookie max-age=0, což ji okamžitě inval iduje. Následně dojde k přesměrování na hlavní stránku.

8. Obchodní systém a platby

8.1 Tok nákupu

Nákupní tok probíhá ve více krocích:

Krok 1: Kupující klikne na „Koupit“ – vytvoří se záznam Purchase ve stavu PENDING.

Krok 2: Kupující klikne na „Zaplatit přes Stripe“ – server vytvoří Stripe Checkout Session.

Krok 3: Kupující je přesměrován na Stripe platební stránku.

Krok 4: Po úspěšné platbě Stripe volá webhook endpoint.

Krok 5: Webhook zkopíruje soubor v S3 do složky kupujícího a vytvoří záznam Model.

Krok 6: Purchase je označen jako ACCEPTED a kupující dostává oznámení.

8.2 Stripe integrace

Stripe je integrován pomocí oficiální Node.js knihovny. Jedná se o sandbox (demo) verzi. Platby probíhají v EUR. Při vypršení Stripe Session je Purchase automaticky označen jako DECLINED. Prodejce musí mít nastavený IBAN, aby platba mohla být převedena.

8.3 S3 logika při nákupu

Soubor originalu zůstává v S3 pod klíčem prodávajícího. Při nákupu je soubor zkopírován pod novým klíčem (s UUID prefixem) do složky kupujícího (models/{buyerId}/). Idempotencí je zajištěno, že dvakrát uplatněný webhook nevytvoří duplicitní záznamy.

9. Systém přátel a oznámení

9.1 Přátelství

Přátelství je obousměrné a uložené v tabulce Friend s enum stavy PENDING, ACCEPTED a DECLINED. Unikátní kombinace (requesterId, addresseeId) zabráňuje duplicitním žádostem. Zrušení přátelství generuje notifikaci pro druhou stranu.

9.2 Dar modelů

Odeslat model jako dar probíhá na straně klienta přes XHR, který zobrazuje postup nahrávání. Mechanismus:

- Odesilatel získá presigned GET URL pro zdrojový soubor a presigned PUT URL pro cílový klíč.
- Klient stáhne soubor a nahrává ho pod novým klíčem do složky příjemce.
- Po nahrání se volá /api/giftModel/confirm, který vytvoří záznam Model pro příjemce.
- Příjemce vidí dar v oznámeních a může ho potvrdit (dismiss).

9.3 Oznámení

Endpoint GET /api/notifications dává všechna aktivní oznámení do jednoho pole:

- friend_request – čekající žádosti o přátelství
- model_gift – přijaté, dosud nepotvrzené dary
- purchase_request – čekající nákupní žádosti pro prodávajícího
- purchase_update – výsledek nákupu pro kupujícího
- friend_removed – oznámení o zrušení přátelství

Oznámení jsou na frontendu aktualizovaná každých 30 sekund komponentou NotificationBell.

10. Nasazení a konfigurace

10.1 Požadavky

- Node.js \geq 20.9.0
- PostgreSQL databáze (libovolný hostitel)
- Amazon S3 bucket s povoleným CORS pro PUT operace z prohlížeče
- Stripe účet s nastavenými webhook endpointy

10.2 Proměnné prostředí (.env.local, .env)

Proměnná	Popis
DATABASE_URL	PostgreSQL connection string
AWS_REGION	Region S3 bucketu (např. eu-central-1)
AWS_ACCESS_KEY_ID	Klíč IAM uživatele s přístupem k S3
AWS_SECRET_ACCESS_KEY	Tajný klíč IAM uživatele
AWS_S3_BUCKET	Název S3 bucketu
STRIPE_SECRET_KEY	Tajný klíč Stripe API
STRIPE_WEBHOOK_SECRET	Webhook signing secret ze Stripe Dashboard
NEXT_PUBLIC_APP_URL	Veřejná URL aplikace (pro Stripe redirect)

10.3 Instalace a spuštění

Instalace závislostí:

```
npm install
```

Spuštění Prisma migraci:

```
npx prisma migrate deploy
```

Spuštění vývojového serveru:

```
npm run dev
```

Sestavení pro produkci:

```
npm run build && npm start
```

11. Bezpečnost

Oblast	Opatření
Hesla	Hashována pomocí bcryptjs se salt rounds = 10
Session	HTTP-only cookie – JavaScriptový kód nemá přístup
S3 přístup	Presigned URL s časovým omezením 1 hodina
Nahrávání	Cesta v S3 vždy obsahuje userId – zabraňuje přepsání cizích souborů
Webhook	Podpis Stripe webhookových událostí ověřován STRIPE_WEBHOOK_SECRET
Autorizace API	Každý endpoint ověřuje cookie a přístupová práva
Idempotence	Opakované webhook volání nevytvoří duplicitu (kontrola stavu Purchase)
CORS S3	Povoleno pouze pro PUT z domény aplikace

12. Závěr

Projekt Trader je plnohodnotná full-stack webová aplikace demonstrující pokročilé znalosti v oblasti moderního webového vývoje. Kombinuje Next.js App Router, TypeScript, Prisma ORM, Amazon S3, Three.js a Stripe do jednoho, funkčního produktu.

Klíčové výzvy při vývoji zahrnovaly:

- Správné nakládání s WebGL kontexty při renderování více 3D modelů současně
- Implementace vícekrokového nákupního toku s Stripe a idempotencí
- Návrh S3 klíčové struktury pro bezpečné oddělení dat uživatelů
- Real-time notificační systém bez WebSocketů (polling)
- Synchronizace stavu mezi DB a S3 při darech a nákupech
- Ukládání modelů do S3 na frontendu, kvůli maximální velikosti posílání přes API na vercelu

Platforma je připravena k nasazení na Vercel s externí PostgreSQL databází a splňuje všechny požadavky na maturitní projekt v oblasti IT.

13. Seznam použitých zdrojů

- [1] Next.js – dokumentace frameworku [online]. Vercel, 2026. Dostupné z: <https://nextjs.org/docs>
- [2] TypeScript – dokumentace jazyka [online]. Microsoft, 2026. Dostupné z: <https://www.typescriptlang.org/docs>
- [3] Tailwind CSS – dokumentace frameworku [online]. Tailwind Labs, 2026. Dostupné z: <https://tailwindcss.com/docs>
- [4] Prisma ORM – dokumentace [online]. Prisma, 2026. Dostupné z: <https://www.prisma.io/docs>
- [5] Supabase – dokumentace platformy [online]. Supabase, 2026. Dostupné z: <https://supabase.com/docs>
- [6] bcrypt – npm package [online]. npm Registry, 2026. Dostupné z: <https://www.npmjs.com/package/bcrypt>
- [5] React – dokumentace knihovny [online]. Meta Platforms, 2026. Dostupné z: <https://react.dev>
- [7] PostgreSQL – dokumentace databáze [online]. PostgreSQL Global Development Group, 2026. Dostupné z: <https://www.postgresql.org/docs>