|  |  |  |
| --- | --- | --- |
|  | | |
| **Závěrečná studijní práce**  **dokumentace** | | |
| **Webová aplikace pro Správu skladu** | | |
| David Beneš | | |
|  | | |
|  | |  |
| **Obor:** | 18-20-M/01 INFORMAČNÍ TECHNOLOGIE  se zaměřením na počítačové sítě a programování | |
| **Třída:**  **Školní rok:** | IT4  2024/2025 | |

**Poděkování**

*Rád bych poděkoval učitelům za jejich rady, které nám v hodině poskytly, jelikož mi s projektem dost pomohly.*

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

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

V Opavě 12. 1. 2025

*podpis autora práce*

**ABSTRAKT**

Tato aplikace, nazvaná **Sklad**, je určena pro správu skladových zásob a sledování operací s produkty. Umožňuje uživatelům efektivně spravovat dostupné položky, evidovat jejich množství, ceny a historii změn. Aplikace podporuje přidávání, odebírání a úpravu produktů přímo z webového rozhraní s přehledným zobrazením dat v kartách produktů.

Systém umožňuje **filtrování produktů** podle různých kritérií, jako je název, cena nebo dostupnost. K dispozici je také **historie operací**, která obsahuje vlastní filtr času dané operace. Uchovává záznamy o změnách ve skladu, včetně uživatele, který akci provedl. Data lze řadit a procházet s využitím moderního webového designu, který zajišťuje přehlednost a uživatelskou přívětivost.

Aplikace je navržena s důrazem na **automatizaci a jednoduchost použití**. Podporuje správu více uživatelských účtů a umožňuje přihlášení pomocí zabezpečeného autentizačního systému.

**Sklad** je určen pro **malé a střední podniky** i individuální uživatele, kteří chtějí mít **rychlý přehled o stavu svých zásob**. Pomáhá minimalizovat chyby v evidenci, zlepšuje správu produktů a přispívá k efektivnějšímu využití zdrojů.

****Klíčová slova:**** Správa skladu, evidence produktů, historie operací, webová aplikace, filtrování produktů, uživatelská autentizace.

**ABSTRACT**

This application, called **Sklad**, is designed for **inventory management and tracking product operations**. It allows users to efficiently manage available items, record their quantities, prices, and history of changes. The application supports adding, removing, and editing products directly from the web interface with a **clear card-based product display**.

The system enables **filtering products** based on various criteria, such as name, price, or availability. Additionally, it includes an **operation history** feature with a custom time filter for specific actions. It stores **records of warehouse changes**, including the user who performed the action. Data can be sorted and browsed using **modern web design**, ensuring clarity and user-friendliness.

The application is designed with a focus on **automation and ease of use**. It supports **multiple user accounts** and allows **secure authentication-based login**.

**Sklad** is intended for **small and medium-sized businesses** as well as **individual users** who want a **quick overview of their inventory status**. It helps **minimize errors in record-keeping**, improves **product management**, and contributes to **more efficient resource utilization**.

****Keywords:** Inventory management, product tracking, operation history, web application, product filtering, user authentication.**

# 

OBSAH

[Úvod 6](#__RefHeading___Toc6792_3228315617)

[1 Úvod do správy skladových zásob 7](#__RefHeading___Toc7485_3228315617)

[1.1 Aplikace pro správu skladových zásob 7](#__RefHeading___Toc7491_3228315617)

[1.2 Princip fungování aplikace Sklad 7](#__RefHeading___Toc7489_3228315617)

[1.3 Výhody a využití aplikace Sklad 7](#__RefHeading___Toc7487_3228315617)

[2 VYUŽITÉ TECHNOLOGIE 10](#__RefHeading___Toc7780_3228315617)

[2.1 Programovací jazyk 10](#__RefHeading___Toc7782_3228315617)

[2.2 Webový framework Django 10](#__RefHeading___Toc7790_3228315617)

[2.3 Databázový systém 10](#__RefHeading___Toc7788_3228315617)

[2.4 Frontend technologie 10](#__RefHeading___Toc7786_3228315617)

[2.5 Systém správy uživatelů a autentizace 11](#__RefHeading___Toc7784_3228315617)

[3 Způsoby řešení a použité postupy 12](#__RefHeading___Toc8682_3228315617)

[3.1 Zobrazení produktů skladem 12](#__RefHeading___Toc8686_3228315617)

[3.2 Historie operací 14](#__RefHeading___Toc8684_3228315617)

[3.3 Sledování zásob 15](#__RefHeading___Toc9143_3228315617)

[Popis funkce: 15](#__RefHeading___Toc9145_3228315617)

[3.4 Struktura projektu 16](#__RefHeading___Toc9864_3228315617)

[4 Výsledky řešení, výstupy, uživatelský manuál 17](#__RefHeading___Toc6782_3228315617)

[4.1 Výsledky řešení 17](#__RefHeading___Toc9862_3228315617)

[4.2 Výstupy aplikace 17](#__RefHeading___Toc9860_3228315617)

[4.3 Uživatelský Manuál 17](#__RefHeading___Toc9858_3228315617)

[1. Naklonujte repozitář: 17](#__RefHeading___Toc9882_3228315617)

[2. Vytvořte virtuální prostředí: 17](#__RefHeading___Toc9880_3228315617)

[3. Aktivujte virtuální prostředí: 17](#__RefHeading___Toc9878_3228315617)

[4. Nainstalujte závislosti: 18](#__RefHeading___Toc9876_3228315617)

[5. Spusťte aplikaci 18](#__RefHeading___Toc9866_3228315617)

[Používání aplikace: 18](#__RefHeading___Toc9872_3228315617)

[4.4 Splněné a nesplněné cíle 18](#__RefHeading___Toc9856_3228315617)

[Splněné cíle: Aplikace splňuje hlavní požadavky zadané na začátku projektu: 18](#__RefHeading___Toc9870_3228315617)

[Nesplněné cíle a návrhy na vylepšení: Během vývoje nebylo implementováno několik pokročilejších funkcí, které byly plánovány: 18](#__RefHeading___Toc9868_3228315617)

[Závěr 19](#__RefHeading___Toc6780_3228315617)

[Seznam použitýCH INFORMAČNÍCH ZDROJů 20](#__RefHeading___Toc6790_3228315617)

[Seznam příloh 21](#__RefHeading___Toc6788_3228315617)

Úvod

Efektivní správa skladových zásob je klíčovým faktorem pro optimalizaci logistických procesů a snížení provozních nákladů. Rostoucí počet produktů a jejich pohyb v rámci skladu vyžaduje přesné sledování množství, cen a provedených operací. Tato práce se zaměřuje na vývoj a implementaci aplikace ****Sklad****, která poskytuje přehledný a intuitivní systém pro správu produktů, jejich evidenci a historii změn.

Hlavním cílem práce je vytvoření nástroje, který umožní uživatelům efektivně spravovat skladové zásoby prostřednictvím moderního webového rozhraní. Aplikace umožňuje přidávání, úpravy a odebírání produktů, filtrování podle různých kritérií a sledování historie operací včetně identifikace uživatele, který změnu provedl.

Práce si klade za cíl nabídnout řešení vhodné jak pro malé a střední podniky, tak pro individuální uživatele, kteří chtějí mít lepší kontrolu nad svým skladem. Vývoj této aplikace představuje krok směrem k efektivnějšímu řízení skladových zásob a zároveň poskytuje uživatelsky přívětivé prostředí pro správu produktů a sledování operací.

# ****1 Úvod do správy skladových zásob****

## ****1.1 Aplikace pro správu skladových zásob****

Existuje mnoho systémů určených ke sledování a správě skladových zásob, které uživatelům umožňují evidovat produkty, monitorovat jejich dostupnost a sledovat historii operací. Mezi běžně používané nástroje patří ERP systémy jako SAP, Odoo nebo Microsoft Dynamics, které poskytují komplexní správu zásob v rámci podnikových procesů. Pro menší podniky a individuální uživatele jsou k dispozici jednodušší aplikace zaměřené na evidenci produktů a jejich pohyb. Tyto systémy obvykle obsahují základní funkce pro přidávání, úpravy a odstraňování produktů, ale často postrádají uživatelsky přívětivé rozhraní nebo možnosti detailního sledování změn v čase.

## ****1.2 Princip fungování aplikace Sklad****

Aplikace **Sklad** se zaměřuje na efektivní správu skladových zásob prostřednictvím intuitivního webového rozhraní. Umožňuje uživatelům přidávat nové produkty, upravovat jejich parametry (např. název, cenu, množství) a sledovat jejich historii. Systém automaticky zaznamenává každou změnu, včetně identifikace uživatele, který operaci provedl, a umožňuje filtrování záznamů podle času a dalších kritérií.

Data jsou prezentována v přehledných kartách produktů s možností řazení podle různých parametrů, jako je cena, množství nebo datum poslední úpravy. Historie operací nabízí detailní záznamy o pohybu produktů ve skladu, což umožňuje lepší kontrolu a minimalizaci chyb. Aplikace podporuje více uživatelských účtů s různými úrovněmi oprávnění a zabezpečeným přihlášením.

Aplikace je navržena s důrazem na automatizaci a přehlednost, čímž přispívá k efektivnímu řízení skladových zásob. Umožňuje rychlou orientaci ve stavu skladu, usnadňuje evidenci produktů a pomáhá uživatelům minimalizovat chyby a optimalizovat skladové operace.

## ****1.3 Výhody a využití aplikace Sklad****

Aplikace **Sklad** přináší řadu výhod, které ji činí užitečným nástrojem pro malé a střední podniky, e-shopy i jednotlivce spravující vlastní skladové zásoby.

**Klíčové výhody aplikace:**

* **Přehledná správa produktů** – umožňuje snadné přidávání, úpravy a mazání produktů přímo z webového rozhraní.
* **Historie operací** – detailně zaznamenává všechny změny, včetně času operace a uživatele, který ji provedl.
* **Filtrování a řazení dat** – poskytuje flexibilní nástroje pro vyhledávání produktů a operací podle různých parametrů.
* **Víceúrovňová správa uživatelů** – podporuje různé role uživatelů s odpovídajícími oprávněními.
* **Bezpečný přístup** – přihlašování přes zabezpečený autentizační systém chrání citlivá data.
* **Jednoduchost a automatizace** – minimalizuje administrativní zátěž a eliminuje běžné chyby při evidenci zásob.

Díky těmto funkcím aplikace **Sklad** pomáhá firmám efektivněji spravovat své skladové zásoby, optimalizovat logistické procesy a zajistit přesnost a aktuálnost dat.

# ****2 VYUŽITÉ TECHNOLOGIE****

## 2.1 Programovací jazyk

Pro vývoj aplikace **Sklad** byl zvolen programovací jazyk **Python**, který je ideální pro práci s webovými aplikacemi a databázovými systémy. Python umožňuje rychlý vývoj s přehledným a čitelným kódem, což přispívá k efektivitě implementace a snadné údržbě systému. Díky rozsáhlému ekosystému knihoven a frameworků je Python ideální volbou pro tvorbu moderních webových aplikací.

## ****2.2 Webový framework Django****

Aplikace je postavena na frameworku **Django**, což je robustní a bezpečný framework pro vývoj webových aplikací v Pythonu. Django poskytuje širokou škálu nástrojů pro správu databází, uživatelské autentizace a generování dynamických webových stránek. Díky svému **Model-View-Template (MVT)** architektonickému vzoru umožňuje Django efektivní oddělení logiky aplikace od prezentace dat.

Hlavní výhody využití Django v aplikaci **Sklad**:

* Vestavěná správa uživatelů a autentizace
* Automatizovaná správa databází pomocí ORM (Object-Relational Mapping)
* Možnost snadného rozšíření o další moduly a API
* Integrovaná ochrana proti běžným bezpečnostním hrozbám (CSRF, XSS, SQL Injection)

## ****2.3 Databázový systém****

Pro ukládání informací o produktech, skladových operacích a uživatelích aplikace byla použita **SQLite** – lehká relační databáze, která je součástí Django a umožňuje snadné nasazení bez nutnosti složité konfigurace. V budoucnu lze aplikaci snadno přizpůsobit pro **PostgreSQL** nebo **MySQL**, což zajistí lepší výkon a škálovatelnost při větším objemu dat.

## ****2.4 Frontend technologie****

Aplikace využívá **HTML, CSS a JavaScript** pro tvorbu uživatelského rozhraní. Styly jsou spravovány pomocí frameworku **Bootstrap**, který umožňuje responzivní design a moderní vzhled aplikace bez nutnosti rozsáhlého kódování CSS.

Hlavní výhody využití Bootstrapu:

* Rychlá a jednoduchá tvorba moderního UI
* Vestavěné komponenty, jako jsou tlačítka, formuláře a navigace

Pro dynamické prvky, jako je **automatické aktualizování seznamu produktů** nebo **vyhledávání v reálném čase**, je použit **JavaScript** a **AJAX**, což zvyšuje interaktivitu aplikace.

## ****2.5 Systém správy uživatelů a autentizace****

Django obsahuje vestavěný **autentizační systém**, který umožňuje správu uživatelských účtů, přihlašování a oprávnění. Uživatelé se mohou přihlašovat pomocí bezpečně uložených hesel a administrátor může nastavovat různé úrovně oprávnění pro jednotlivé role (například běžný uživatel vs. administrátor skladu).

# 3 Způsoby řešení a použité postupy

## 3.1 Zobrazení produktů skladem

def produkty\_skladem(request):  
 produkty = Produkt.objects.all()  
  
 # Inicializace aktivních filtrů  
 aktivni\_filtry = {}  
  
 # Filtrování podle názvu  
 nazev = request.GET.get('nazev', '').strip()  
 if nazev:  
 produkty = produkty.filter(nazev\_\_icontains=nazev)  
 aktivni\_filtry['Název'] = nazev  
  
 # Filtrování podle ceny  
 min\_cena = request.GET.get('min\_cena')  
 max\_cena = request.GET.get('max\_cena')  
 if min\_cena:  
 produkty = produkty.filter(cena\_\_gte=min\_cena)  
 aktivni\_filtry['Minimální cena'] = f"{min\_cena} Kč"  
 if max\_cena:  
 produkty = produkty.filter(cena\_\_lte=max\_cena)  
 aktivni\_filtry['Maximální cena'] = f"{max\_cena} Kč"  
  
 # Manipulace s množstvím při POST požadavku  
 if request.method == 'POST':  
 produkt\_id = request.POST.get('produkt\_id') # Získání ID produktu  
 mnozstvi = int(request.POST.get('mnozstvi', 0)) # Získání zadaného množství  
 produkt = get\_object\_or\_404(Produkt, id=produkt\_id)  
  
 # Aktualizace množství  
 produkt.mnozstvi += mnozstvi  
 produkt.save()  
  
 # Zpráva o úspěšné aktualizaci  
 messages.success(request, f"Produkt {produkt.nazev} byl aktualizován. Nové množství: {produkt.mnozstvi}.")  
 return redirect('produkty\_skladem')  
  
 # Předání produktů a aktivních filtrů do šablony  
 return render(request, 'sklad/produkty\_skladem.html', {  
 'produkty': produkty,  
 'aktivni\_filtry': aktivni\_filtry,  
 })

Tato funkce **produkty\_skladem** slouží k zobrazení všech produktů dostupných ve skladu. Uživatelé mohou produkty **filtrovat** podle různých kritérií, jako je **název** a **cenové rozmezí**. Kromě toho umožňuje **manipulaci s množstvím produktů**, tedy jejich přidávání prostřednictvím formuláře.

* **Filtrování produktů:**
  + **Název** – uživatel může zadat část názvu produktu a zobrazí se pouze odpovídající položky.
  + **Cenové rozmezí** – produkty lze filtrovat podle minimální a maximální ceny, což umožňuje rychlé nalezení produktů v požadovaném cenovém spektru.
* **Manipulace s množstvím:**
  + Pokud uživatel zadá určité množství a odešle formulář, množství vybraného produktu se **aktualizuje** v databázi.
  + Po úspěšné aktualizaci se zobrazí **potvrzující zpráva** o změně skladových zásob.

## **3.2 Historie operací**

def historie\_operaci(request):  
 # Načteme všechny operace z historie  
 historie = HistorieOperaci.objects.all()  
  
 # Získání parametrů filtrování z GET requestu  
 od\_datum = request.GET.get('od\_datum') # Parametr 'od\_datum' z GET  
 do\_datum = request.GET.get('do\_datum') # Parametr 'do\_datum' z GET  
  
 # Filtrování podle zadaných dat  
 if od\_datum:  
 od\_datum\_parsed = parse\_datetime(od\_datum) # Převod na datetime objekt  
 if od\_datum\_parsed:  
 historie = historie.filter(datum\_\_gte=od\_datum\_parsed) # Filtrovat záznamy po zadaném datu  
  
 if do\_datum:  
 do\_datum\_parsed = parse\_datetime(do\_datum) # Převod na datetime objekt  
 if do\_datum\_parsed:  
 historie = historie.filter(datum\_\_lte=do\_datum\_parsed) # Filtrovat záznamy do zadaného data  
  
 # Vrátíme historii operací do šablony  
 return render(request, 'sklad/historie.html', {'historie': historie})

*Funkce* ***historie\_operaci*** *slouží k zobrazení záznamů o všech změnách, které proběhly ve skladu. Tyto záznamy umožňují sledovat, jaké úpravy byly provedeny, kdo je provedl a kdy k nim došlo.*

* **Záznamy operací:**
  + Aplikace uchovává **historii změn** týkajících se skladových zásob.
  + Každý záznam obsahuje **datum operace**, typ změny (např. přidání nebo odebrání produktů) a další detaily.
* **Možnost filtrování:**
  + Uživatel může zobrazit operace **v určitém časovém období**.
  + Filtrace funguje pomocí **dvou parametrů**:
    - **od\_datum** – zobrazí všechny operace provedené **od zadaného data**.
    - **do\_datum** – zobrazí operace provedené **do zadaného data**.
  + Pokud jsou tyto parametry zadány, funkce je **převede na datetime objekt** a použije pro filtrování.
* **Uživatelská přívětivost:**
  + Historie operací je zobrazena ve **speciální šabloně**, kde se uživatelé mohou rychle orientovat.
  + To umožňuje **audit změn** a zajišťuje **přehlednost skladových pohybů**.

Tato funkce je klíčová pro **sledování změn** v systému a umožňuje lepší kontrolu nad správou skladových zásob.

## 3.3 Sledování zásob

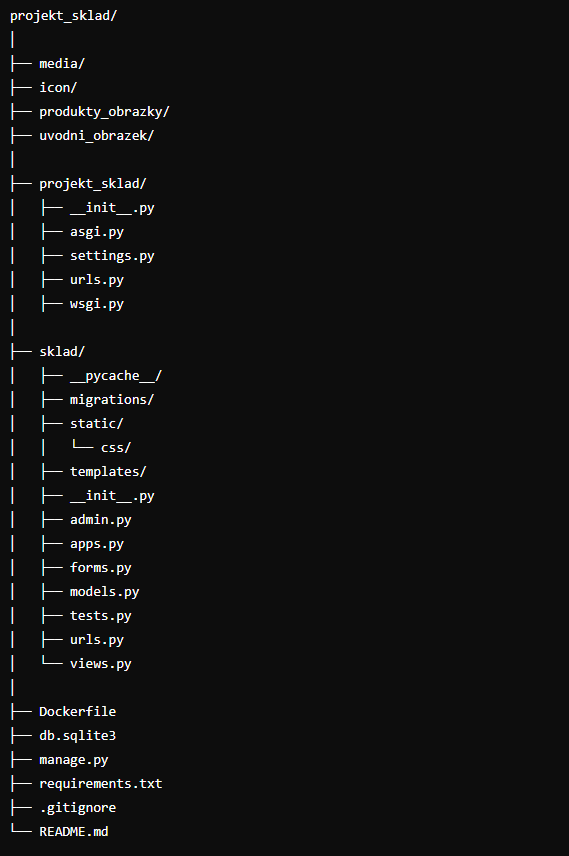
def sledovani\_zasob(request):  
 # Získání všech produktů  
 produkty = Produkt.objects.all()  
  
 # Filtrace produktů, které mají nulové množství  
 produkty\_nedostatek = produkty.filter(mnozstvi=0)  
  
 return render(request, 'sklad/sledovani\_zasob.html', {  
 'produkty': produkty,  
 'produkty\_nedostatek': produkty\_nedostatek,  
 })

Funkce **sledovani\_zasob** je určena pro správu a monitorování aktuálního stavu zásob ve skladové aplikaci. Umožňuje administrátorům rychle zjistit, které produkty jsou na skladě a které jsou vyprodány, čímž usnadňuje rozhodování o objednávkách a údržbě zásob.

### Popis funkce:

1. **Získání všech produktů**: Funkce začíná tím, že načte všechny produkty, které jsou uloženy v databázi, pomocí dotazu Produkt\_objects.all(). Tento dotaz získává všechny položky, které jsou v systému evidovány.
2. **Filtrace produktů s nulovým množstvím**: Funkce poté filtruje produkty, které mají nulové množství, tedy produkty, které již nejsou na skladě (množství=0). Tento krok je důležitý pro identifikaci produktů, které je potřeba objednat.
3. **Předání dat do šablony**: Funkce následně předává dvě sady produktů do šablony sledovani\_zasob.html. První sada obsahuje všechny produkty na skladě, zatímco druhá obsahuje pouze produkty, které mají nulové množství a nejsou k dispozici pro objednávky.

## 3.4 Struktura projektu



# 4 Výsledky řešení, výstupy, uživatelský manuál

## 4.1 Výsledky řešení

Aplikace byla navržena tak, aby efektivně podporovala správu skladových zásob a produktů. Hlavním cílem bylo zajistit snadnou a přehlednou evidenci produktů, sledování skladových pohybů a poskytování uživatelsky přívětivého rozhraní. Mezi hlavní funkce aplikace patří:

* **Správa skladových zásob:** Přidávání, upravování a mazání produktů v databázi.
* **Sledování pohybů na skladě:** Záznam operací, jako jsou příjmy a výdeje, s možností zpětného dohledání historie.
* **Generování přehledů:** Výpis aktuálních zásob včetně filtrů a třídění podle různých parametrů (název, množství, datum přidání apod.).
* **Notifikace:** Upozornění na nízké stavy zásob nebo expirované produkty.
* **Uživatelské role:** Oddělené přístupy pro administrátory a běžné uživatele.

## 4.2 Výstupy aplikace

Aplikace generuje následující typy výstupů:

* **Tabulka zásob:** Přehled aktuálních skladových položek s možností filtrování a třídění.
* **Historie operací:** Chronologický přehled všech příjmů, výdejů a dalších operací, včetně odpovědných osob.
* **Notifikační panel:** Přehled důležitých upozornění, jako jsou nedostatečné zásoby nebo produkty s blížící se expirací.

## 4.3 Uživatelský Manuál

**Spuštění aplikace:**  
Aplikaci lze spustit kliknutím na zástupce nebo přes vývojové prostředí:

### Naklonujte repozitář:

git clone https://github.com/d4veyyy/Zaverecny-Projekt-IT4.git

### Vytvořte virtuální prostředí:

python -m venv .env

### Aktivujte virtuální prostředí:

* + **Windows:** .\.env\Scripts\activate
  + **Linux/MacOS:** source .env/bin/activate

### Nainstalujte závislosti:

pip install -r requirements.txt

### Spusťte aplikaci

python manage.py runserver

### ****Používání aplikace:****

* **Přihlášení:** Přihlaste se pomocí zvolených přihlašovacích údajů.
* **Správa produktů:** V sekci „Správa produktů“ můžete přidávat, upravovat nebo mazat skladové položky.
* **Sledování pohybů:** Sledujte historii příjmů a výdejů v sekci „Historie operací“.
* **Export dat:** Exportujte data zásob nebo operací pomocí tlačítka „Exportovat“.
* **Nastavení:** Přizpůsobte aplikaci podle svých potřeb, například nastavte minimální skladové množství pro notifikace.

## 4.4 Splněné a nesplněné cíle

### ****Splněné cíle:**** Aplikace splňuje hlavní požadavky zadané na začátku projektu:

* **Správa produktů:** Možnost přidávat, upravovat a mazat produkty.
* **Sledování pohybů:** Přehledné zobrazení příjmů a výdejů s možností filtrace a exportu.
* **Notifikace:** Upozornění na nízké zásoby nebo blížící se expirace.
* **Role uživatelů:** Oddělené role pro administrátory a běžné uživatele.
* **Export dat:** Možnost exportovat skladové zásoby a historii operací.

### ****Nesplněné cíle a návrhy na vylepšení:**** Během vývoje nebylo implementováno několik pokročilejších funkcí, které byly plánovány:

1. **Rozšířené notifikace:**
   * Možnost zasílání e-mailových upozornění na nízké zásoby.
2. **Pokročilá analýza dat:**
   * Automatizované generování grafů nebo predikce zásob.
3. **Vylepšené GUI:**
   * Responsivní design pro lepší použití na mobilních zařízeních.

# Závěr

Cílem projektu bylo vytvořit aplikaci pro efektivní správu skladových zásob s využitím jazyka Python a frameworku Django. Aplikace umožňuje uživateli evidovat produkty, sledovat pohyby na skladě a generovat přehledné výstupy, jako je historie operací či export dat do CSV. Důraz byl kladen na uživatelskou přívětivost a možnost snadného použití v praxi.

Aplikace splnila hlavní cíle, včetně přehledné správy produktů, zaznamenávání skladových operací a upozornění na nízký stav zásob. Hlavním přínosem je jednoduché a efektivní řešení, které může být nasazeno ve skladových procesech malých a středních firem.

Do budoucna bych chtěl aplikaci rozšířit o nové funkce, jako jsou pokročilé notifikace, víceuživatelská podpora s rozdělením práv nebo integrace s jinými systémy, což v současné verzi představuje hlavní prostor pro zlepšení.

Odkaz na github : https://github.com/d4veyyy/Zaverecny-Projekt-IT4.git

# Seznam použitýCH INFORMAČNÍCH ZDROJů

[1] OpenAI. ChatGPT [online]. Poslední revize 20. 12. 2024 [cit. 2024-12-20]. Dostupné z: <https://chatgpt.com/>.

[2] Django Documentation [online]. Poslední revize 10. 1. 2025 [cit. 2025-01-10]. Dostupné z: <https://docs.djangoproject.com/en/stable/>.

[3] Bootstrap Documentation [online]. Poslední revize 10. 1. 2025 [cit. 2025-01-10]. Dostupné z: https://getbootstrap.com/docs/.

[4] Python Documentation [online]. Poslední revize 10. 1. 2025 [cit. 2025-01-10]. Dostupné z: <https://docs.python.org/3/>.