UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2025 Klára Kahounová

Univerzita Pardubice
Fakulta elektrotechniky a informatiky

Využití webscrapingu pro vizualizaci dat

Bakalářská práce

2025 Klára Kahounová

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	**Klára Kahounová**
Osobní číslo:	**I22105**
Studijní program:	**B0688A140009 Informační technologie**
Téma práce:	**Využití webscrapingu pro vizualizaci dat.**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování

Cílem práce bude vytvořit jednoduchou aplikaci, která umožní uživatelům provádět datovou analýzu na základě dat získaných pomocí web scrapingu z veřejně dostupného zdroje. Teoretická část bude obsahovat: podrobný popis techniky web scraping, včetně nástrojů a knihoven; etické aspekty web scrapingu (autorská práva, ochrana osobních údajů). Praktická část bude obsahovat: výběr datového zdroje, které budou sloužit jako základ pro vizualizaci; implementaci web scrapingu: detailní popis implementace web scrapingu v programovacím jazyku Python; návrh uživatelského rozhraní.

Rozsah pracovní zprávy: **min. 30**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

HAJBA, Gábor László. Website scraping with Python: using Beautifulsoup and Scrapy. New York: Apress, [2018]. ISBN 978-1-4842-3924-7.
Responsive web design with HTML5 and CSS: develop future-proof responsive websites using the latest HTML5 and CSS techniques

Vedoucí bakalářské práce: **Ing. Soňa Neradová, Ph.D.**
Katedra informačních technologií

Datum zadání bakalářské práce: **15. prosince 2024**
Termín odevzdání bakalářské práce: **16. května 2025**

L.S.

prof. Ing. Petr Doležel, Ph.D. v.r.
děkan

Ing. Jan Panuš, Ph.D. v.r.
vedoucí katedry

V Pardubicích dne 28. února 2025

Prohlašuji:

Práci s názvem Využití webscrapingu pro vizualizaci dat jsem vypracovala samostatně. Veškeré literární prameny a informace, které jsem v práci využila, jsou uvedeny v seznamu použité literatury.

Byla jsem seznámena s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 7/2019 Pravidla pro odevzdávání, zveřejňování a formální úpravu závěrečných prací, ve znění pozdějších dodatků, bude práce zveřejněna prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15. 05. 2025

Klára Kahounová v.r.

PODĚKOVÁNÍ

Mé poděkování patří Ing. Soně Neradové Ph.D. za velmi vstřícný přístup, podporu a praktické rady. Velké díky také patří mému příteli, kamarádkám a rodině za trpělivost a podporu v průběhu mého studia.

ANOTACE

Tato bakalářská práce se zabývá využitím web scrapingu pro automatizovaný sběr dat z webových stránek a jejich následné zpracování a prezentaci. V teoretické části jsou popsány principy web scrapingu, jeho využití v praxi, dostupné nástroje a právní a etické souvislosti. Praktická část práce se zaměřuje na návrh a implementaci webové aplikace, která automatizovaně získává informace o akčních produktech obchodního řetězce Penny Market, ukládá je do databáze a zobrazuje prostřednictvím webového rozhraní. Aplikace umožňuje sledování cenového vývoje, porovnání produktů a nabízí přehledné uživatelské rozhraní. Cílem práce je ukázat možnosti praktického nasazení web scrapingu při zpracování veřejně dostupných informací a reflektovat technické, právní a návrhové aspekty spojené s vývojem takové aplikace.

KLÍČOVÁ SLOVA

Web scraping, datová analýza, webová aplikace

TITLE

Leveraging web scraping for data visualization

ANNOTATION

This bachelor's thesis focuses on the use of web scraping for the automated collection of data from websites and its subsequent processing and presentation. The theoretical part describes the principles of web scraping, its practical applications, available tools, and legal and ethical considerations. The practical part is dedicated to the design and implementation of a web application that automatically collects information about promotional products from the Penny Market retail chain, stores the data in a database, and displays it through a web interface. The application enables users to track price trends, compare products, and provides a clear and user-friendly interface. The goal of this thesis is to demonstrate the practical use of web scraping for processing publicly available information and to reflect on the technical, legal, and design aspects associated with the development of such an application.

KEYWORDS

Web scraping, data analysis, web application

OBSAH

SEZNAM ILUSTRACÍ A TABULEK

SEZNAM ZKRATEK

AJAX - Asynchronous JavaScript and XML

API - Application Programming Interface

CDP - Chrome DevTools Protocol

CFAA - Computer Fraud and Abuse Act

CSS - Cascading Style Sheets

CSV – Comma-Separated Values

DMCA - Digital Millennium Copyright Act

DOM - Document Object Model

EHP - Evropský hospodářský prostor

EU - Evropská unie

GDPR - General Data Protection Regulation

HTML - HyperText Markup Language

HTML5 - HyperText Markup Language version 5

HTTP - HyperText Transfer Protocol

HTTPS - HyperText Transfer Protocol Secure

JS - JavaScript

JSON - JavaScript Object Notation

JSON-RPC - JavaScript Object Notation – Remote Procedure Call

ORM - Object-Relational Mapping

UI - User Interface

URL - Uniform Resource Locator

WHATWG - Web Hypertext Application Technology Working Group

XML - eXtensible Markup Language

Xpath - XML Path Language

XSS - Cross-Site Scripting

TERMINOLOGIE

Web scraping: Technika pro automatizované získávání dat z webových stránek.

API (Application Programming Interface): Rozhraní, které umožňuje komunikaci mezi různými softwarovými aplikacemi.

ÚVOD

Moderní společnost generuje obrovské množství dat, jejichž dostupnost a správné využití se stává klíčovým faktorem v řadě oborů. Mnohá z těchto dat jsou veřejně přístupná na webových stránkách, avšak často ve formátu nevhodném pro přímé zpracování. Web scraping představuje techniku, která umožňuje automatizované získávání těchto dat a jejich převod do strukturované podoby, čímž umožňuje jejich další analýzu, zobrazení a efektivní využití.

Tato bakalářská práce se zaměřuje na využití web scrapingu v kontextu práce s informacemi získanými z veřejně dostupných online zdrojů. V úvodních kapitolách jsou popsány základní principy scrapingu, jeho vývoj, běžné způsoby využití a přehled relevantních nástrojů. Hlavní pozornost je věnována programovacímu jazyku Python a knihovnám Beautiful Soup, Scrapy a Selenium, které jsou široce používané pro automatizovaný sběr webových dat. Dále jsou uvedeny alternativy v jiných programovacích jazycích a analyzovány právní a etické aspekty této techniky, včetně problematiky ochrany osobních údajů a respektování pokynů definovaných souborem robots.txt.

Praktická část práce je věnována návrhu a implementaci webové aplikace, která slouží k automatickému získávání a prezentaci informací o akčních nabídkách obchodního řetězce Penny Market. Aplikace kombinuje metodu web scrapingu s přístupem přes aplikační rozhraní a výsledná data ukládá do databáze, ze které jsou následně zobrazována prostřednictvím webového rozhraní. Popsána je architektura systému, návrh datového modelu, jednotlivé aplikační moduly i vizuální rozhraní určené pro koncového uživatele.

Cílem práce je ukázat možnosti praktického nasazení web scrapingu při zpracování veřejně dostupných informací a současně reflektovat technické, právní a návrhové aspekty spojené s vývojem takové aplikace.

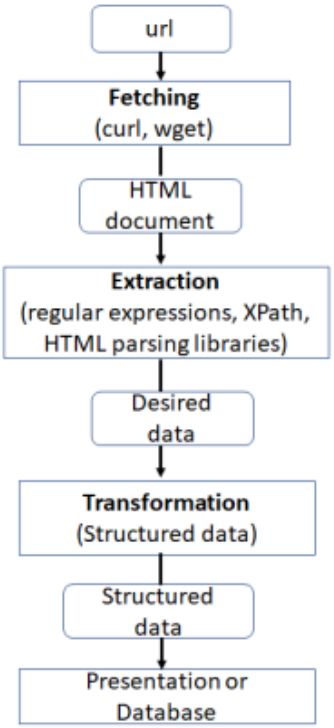
1 Představení web scrapingu

1.1 Web scraping

Web scraping je účinná metoda pro získávání nestrukturovaných dat z webových stránek a jejich převod do strukturovaného formátu, který lze uložit do databáze a následně analyzovat. Tato technika se také označuje jako extrakce webových dat, web data scraping, web harvesting nebo screen scraping. Web scraping spadá do oblasti data miningu a jeho hlavním cílem je získat informace z webových stránek a převést je do přehledné podoby, například tabulek, databází nebo souborů CSV (Comma-Separated Values). Pomocí této metody lze shromažďovat data, jako jsou ceny produktů, akciové kurzy, tržní analýzy a podrobnosti o produktech. [1]

1.2 Proces web scrapingu

Průběh web scrapingu je rozdělen do tří hlavních etap, jak ilustruje Obrázek 1. Nejprve musí být dostupná cílová webová stránka obsahující relevantní informace, což se označuje jako etapa získávání. Tato činnost se realizuje prostřednictvím protokolu HTTP (HyperText Transfer Protocol), což je standardní metoda pro přenos požadavků a odpovědí mezi klientem a serverem. Webové prohlížeče využívají obdobné techniky k načítání obsahu stránek. V této fázi lze využít knihovny jako curl nebo wget, které odešlou HTTP GET požadavek na cílovou adresu (URL - Uniform Resource Locator) a jako odpověď obdrží HTML (HyperText Markup Language) dokument. Po získání HTML dokumentu by měla být vybrána klíčová data. V tomto kroku, který se označuje jako etapa extrakce, se aplikují regulární výrazy, knihovny pro parsování HTML a dotazy XPath (XML Path Language). XPath slouží k vyhledávání informací v datových strukturách. Jakmile zůstanou pouze potřebná data, mohou být převedena do strukturované podoby pro prezentaci či archivaci. Na základě těchto uložených údajů lze dále informace analyzovat. [2]



Obrázek 1: Proces web scrapingu [2]

1.3 Historie web scrapingu

Web scraping se začal vyvíjet společně s internetem od vzniku World Wide Webu v roce 1989. První webový prohlížeč WorldWideWeb vznikl v roce 1990 a o tři roky později se objevily první webové roboty, například JumpStation, které automatizovaly indexování webových stránek. V roce 2000 společnosti Salesforce a eBay představily API (Application Programming Interface), které umožnilo vývojářům přistupovat k veřejně dostupným datům bez nutnosti ručního scrapování. Roku 2004 byla vydána Python knihovna Beautiful Soup, která zjednodušila extrakci informací z HTML stránek. V roce 2006 se web scraping stal přístupnějším i pro běžné uživatele díky vizuálním nástrojům, jako byla Web Integration Platform 6.0 od Kapow Software. Dnes existuje mnoho nástrojů, které umožňují automatizovaný sběr dat, a web scraping se běžně využívá v různých oblastech, od analýzy trhu po monitorování konkurence.[3]

1.4 Využití web scrapingu

V dnešní době se dá web scraping uplatnit v mnoha odvětvích pro různé účely. Nejčastější odvětví jsou:

- Sledování změn cen – monitoring cen produktů a jejich srovnávání mezi různými e-shopy.

- Scrapování kontaktních údajů – extrakce e-mailových adres nebo telefonních čísel.
- Sběr recenzí produktů – analýza uživatelských recenzí a zpětné vazby.
- Shromažďování realitních nabídek – scrapování seznamů nemovitostí z realitních webů.
- Monitorování počasí – automatizovaný sběr meteorologických dat.
- Detekce změn na webových stránkách – sledování aktualizací obsahu na konkrétních stránkách.
- Analýza sociálních médií – scrapování příspěvků pro analýzu veřejného mínění a trendů.
- Indexování webu pro vyhledávače – Google, Bing a další vyhledávače pravidelně scrapují metadata webových stránek pro tvorbu výsledků vyhledávání.[4].

2. Nástroje pro web scraping

Jedním z nejpoužívanějších jazyků pro web scraping je Python, a to díky své jednoduché syntaxi, rozsáhlé knihovně nástrojů a silné komunitní podpoře. Python nabízí efektivní řešení pro základní i pokročilé úlohy – od extrakce dat z HTML po zpracování dynamického obsahu a paralelní zpracování požadavků.

Mezi nejvýznamnější nástroje pro scraping patří Beautiful Soup pro jednoduchou extrakci dat z HTML a XML (eXtensible Markup Language), Scrapy jako výkonný framework pro škálovatelné aplikace a Selenium pro práci s dynamickými stránkami prostřednictvím ovládání reálného prohlížeče. Každý z těchto nástrojů má specifické výhody v závislosti na charakteru cílové webové stránky a požadavcích na scraping. V následujících podkapitolách jsou podrobněji představeny jednotlivé nástroje, včetně jejich vlastností a způsobu instalace.

2.1 Python

Python je vysoce úrovňový, interpretovaný programovací jazyk, který je známý svou jednoduchostí, čitelností a všestranností. Díky své srozumitelné syntaxi a dynamickému typování je Python ideální volbou pro širokou škálu úloh, od jednoduchých skriptů po komplexní aplikace a systémy. Jazyk podporuje objektově orientované programování, ale zároveň umožňuje i procedurální a funkcionální přístup, což z něj činí flexibilní nástroj vhodný pro různé programovací přístupy.

Python je navržen tak, aby výrazně urychlil proces vývoje aplikací a skriptů. Oproti jazykům jako C, C++ nebo Java, které vyžadují kompilaci a složitější syntaxi, Python umožňuje vývojářům psát méně kódu pro dosažení stejných cílů, což zvyšuje produktivitu. Programy v Pythonu bývají kratší a přehlednější díky vestavěným vysokoúrovňovým datovým typům, jako jsou seznamy (listy), slovníky (dictionaries) nebo množiny (sets). Strukturování kódu se v Pythonu provádí pomocí odsazení namísto složených závorek, což přispívá k lepší čitelnosti a srozumitelnosti programů.

Python je interpretovaný jazyk, což znamená, že kód je vykonáván přímo bez nutnosti předchozí kompilace. To umožňuje interaktivní práci s interpretem, kde mohou programátoři snadno testovat jednotlivé části kódu, navrhovat funkce a rychle iterovat při vývoji aplikací. Díky tomu je Python oblíbený nejen mezi začátečníky, ale i mezi profesionály, kteří oceňují jeho schopnost zrychlit vývojový cyklus.

Python disponuje rozsáhlou standardní knihovnou, která pokrývá široké spektrum oblastí, včetně práce se soubory, sítí, systémových volání, regulárních výrazů, webových služeb, vícevláknového zpracování a dalších. Vedle standardní knihovny je dostupné i obrovské množství modulů a balíčků třetích stran, které lze snadno instalovat pomocí nástroje pip.

Python je vhodný pro různé aplikační oblasti, včetně webového vývoje, vědeckého výpočtu, datové analýzy, automatizace, strojového učení, umělé inteligence, tvorby her a mnoho dalších.

Díky těmto vlastnostem je Python jedním z nejpopulárnějších a nejpoužívanějších programovacích jazyků současnosti a představuje ideální volbu jak pro začínající programátory, tak pro zkušené vývojáře při řešení širokého spektra úloh. [5]

2.1.1 BeautifulSoup

Beautiful Soup je knihovna pro syntaktickou analýzu, která slouží k extrakci dat z dokumentů ve formátu HTML, XML a dalších značkovacích jazyků. Umožňuje automatizované získávání specifických informací z webových stránek, které neposkytují přímé rozhraní pro export dat. Pomocí této knihovny lze extrahovat požadovaný obsah, odstranit značkovací jazyk a dále zpracovat čistá data. Součástí knihovny Beautiful Soup je rozsáhlá dokumentace zaměřená na popis jednotlivých funkcí a možností jejího využití. Nejsnazší postup pro instalaci knihovny Beautiful Soup představuje využití správce balíčků pip. V případě, že pip není k dispozici, je vhodné nejprve provést základní postup instalace Python modulů. Po úspěšném zprovoznění správce pip se instalace knihovny provede prostřednictvím následujícího příkazu zadaného v příkazovém řádku. [6]

pip install beautifulsoup4

Obrázek 2 zobrazuje úvodní blok kódu v jazyce Python, který slouží k přípravě web scrapingu ze stránek obchodního řetězce Penny Market. Nejprve jsou importovány potřebné knihovny BeautifulSoup a requests, které umožňují načítání a zpracování HTML obsahu. V proměnné url je uložen odkaz na stránku s aktuálními nabídkami. Pomocí metody requests.get() se provede HTTP GET požadavek a získá se obsah stránky. Ten je následně zpracován pomocí knihovny BeautifulSoup s využitím HTML parseru, čímž vznikne objekt soup, určený pro snadné vyhledávání konkrétních prvků v HTML kódu. Na závěr je inicializován prázdný seznam products, do kterého budou později ukládány informace o nabízených produktech. Tento kód tedy představuje přípravnou fázi pro samotné získávání dat ze zvoleného webu.

```
from bs4 import BeautifulSoup
import requests

url = 'https://www.penny.cz/nabidky'
response = requests.get(url)
soup = BeautifulSoup(response.text, 'html.parser')
products = []
```

Obrázek 2: Použití knihovny Beautiful Soup pro extrakci akčních produktů

2.1.2 Scrapy

Tento framework je v současnosti považován za jedno z nejkomplexnějších řešení pro automatizované získávání dat z webových stránek, neboť je schopen zvládnout i složité úlohy spojené se scrapingem bez nutnosti rozsáhlých úprav. Scrapy poskytuje funkce pro ukládání webových stránek do mezipaměti a podporuje implementaci paralelního zpracování dat, což výrazně zvyšuje efektivitu a rychlost procesu. [7] Proces instalace frameworku je obdobný jako v případě knihovny Beautiful Soup.

pip install scrapy

2.1.3 Selenium

Selenium představuje výkonný nástroj pro web scraping, jenž byl původně navržen pro účely testování webových aplikací. V současnosti se však hojně využívá i v situacích, kdy je nezbytné načíst webovou stránku v přesné podobě, jaká je prezentována uživateli v reálném prohlížeči. Selenium umožňuje automatizované ovládání prohlížečů za účelem načtení webových stránek, extrakce potřebných dat, pořizování snímků obrazovky či validace prováděných interakcí na webu. Selenium neobsahuje vlastní webový prohlížeč, a proto vyžaduje ke svému spuštění integraci s externím prohlížečem. Například při spuštění Selenia s prohlížečem Firefox se uživateli zobrazí nové okno Firefoxu, ve kterém Selenium realizuje akce definované ve zdrojovém kódu. [8] Selenium využívá podobný instalační postup jako předchozí nástroje.

pip install selenium

2.2 Porovnání jednotlivých nástrojů

Tabulka 1 poskytuje srovnání klíčových vlastností a charakteristik nástrojů používaných pro web scraping, konkrétně Beautiful Soup, Scrapy a Selenium. Pro každý nástroj jsou uvedeny následující kritéria: klasifikace nástroje, výkonnost, podpora JavaScriptu, oblast použití, závislost na prohlížeči, hlavní výhoda a hlavní nevýhoda. Toto srovnání má za cíl usnadnit výběr vhodného nástroje v závislosti na specifických požadavcích projektu web scrapingu.

Tabulka 1: Porovnání nástrojů pro web scraping

Kritérium	**Beautiful Soup**	**Scrapy**	**Selenium**
Klasifikace nástroje	Knihovna pro parsování HTML	Framework pro web scraping a crawling	Nástroj pro automatizaci prohlížeče a testování UI (User Interface)
Výkonnost	Průměrná (vhodná pro menší a střední úlohy)	Vysoká (efektivní asynchronní zpracování)	Nízká (ovlivněná latencí prohlížeče a vykonáváním JS)
Podpora JavaScriptu (JS)	Nepodporuje zpracování JavaScriptu	Nepodporuje zpracování JavaScriptu	Plná podpora (emulace uživatelského chování)
Oblast použití	Rychlé zpracování statického HTML	Rozsáhlé scrapingové projekty, webové crawlery	Automatizace dynamických webů a testování uživatelského rozhraní
Závislost na prohlížeči	Nezávislá na prohlížeči	Nezávislá na prohlížeči	Vyžaduje instalaci a použití reálného prohlížeče
Hlavní výhoda	Jednoduchost, rychlá integrace s HTTP knihovnami	Vysoká škálovatelnost, podpora rozsáhlého crawlování	Možnost testovat aplikaci ve skutečném prostředí prohlížeče
Hlavní nevýhoda	Neschopnost zpracovat dynamický obsah (JS)	Nadbytečná komplexita pro jednoduché skripty	Nízká rychlost a vyšší systémové nároky

2.3 Alternativy v jiných jazycích

Přestože je Python v oblasti web scrapingu dominantní volbou, podobné možnosti nabízí i další populární programovací jazyky. Pro některé z nich existují knihovny nebo nástroje, které umožňují získávání a zpracování dat z webových stránek. Následující podkapitoly představují nejvýznamnější knihovny pro scraping dat v jazycích Java, JavaScript a C#, které poskytují srovnatelné funkcionality s nástroji dostupnými v Pythonu.

2.3.1 Java

Nejčastěji využívaná knihovna pro extrakci dat v jazyce Java se nazývá Jsoup. Je určená pro zpracování HTML dokumentů. Nabízí přehledné API pro stahování dat z URL adres, extrakci a úpravu dat a využívá přitom metody DOM (Document Object Model) podle specifikace HTML5 (HyperText Markup Language version 5) a CSS (Cascading Style Sheets) selektory.

Implementuje HTML standard podle specifikace WHATWG (Web Hypertext Application Technology Working Group) a provádí parsování HTML do DOM stromu, který odpovídá tomu, co generují moderní prohlížeče.

Jsoup umožňuje načíst HTML ze zdrojů, jako jsou URL, soubory nebo řetězce, a následně nad tímto obsahem provádět operace vyhledávání a extrakce dat. Dále umožňuje manipulaci s HTML prvky, jejich atributy a textovým obsahem.

Součástí knihovny je také funkce pro čištění obsahu zaslaného uživatelem pomocí safelistu, čímž se předchází XSS (Cross-Site Scripting) útokům. Výstupem je vždy upravené a přehledné HTML.

Jsoup je navržen tak, aby si poradil se všemi typy HTML dokumentů, od plně validního kódu až po nevalidní „tag-soup“, a vždy vytvořil smysluplný parsovací strom. [9]

2.3.2 JavaScript

Puppeteer je open-source knihovna pro Node.js, která poskytuje vysoce abstraktní rozhraní pro automatizaci a ovládání prohlížečů založených na Chromiu. Využívá protokol Chrome DevTools (CDP) k odesílání JSON-RPC (JavaScript Object Notation – Remote Procedure Call) zpráv za účelem inspekce, ladění a profilování těchto prohlížečů. Po instalaci si Puppeteer automaticky stáhne a používá prohlížeč „Chrome for Testing“, což je odlehčená verze Chromia určená primárně pro testovací účely. Kromě toho umožňuje ovládání i dalších prohlížečů založených na Chromiu, jako jsou například Google Chrome nebo Microsoft Edge. Skripty

využívající Puppeteer API, psané v jazyce JavaScript nebo TypeScript, odesílají CDP zprávy prohlížeči, který následně prostřednictvím protokolu HTTP(S) komunikuje s testovanou webovou aplikací. [10]

2.3.3 C#

HTML Agility Pack je knihovna pro .NET, která slouží k zpracování a úpravu HTML dokumentů. Je často využívána při web scrapingu, protože umožňuje efektivní navigaci a extrakci dat ze strukturovaného i nevalidního HTML kódu. Nabízí funkce pro práci s DOM modelem a podporuje využití XPath dotazů. Díky této knihovně lze snadno analyzovat HTML soubory bez nutnosti ručního parsování. [11]

3. Právní a etické aspekty

Webscraping, jako nástroj pro získávání dat z internetu, představuje nejen technickou výzvu, ale také vyvolává řadu právních a etických problémů. Tyto otázky se týkají jak samotného procesu sběru dat, tak i jeho následného využívání. I když webscraping přináší významné výhody v oblasti analýzy dat a výzkumu, je nezbytné zajistit, aby tento proces byl prováděn v souladu s platnými právními předpisy a etickými normami. Organizace a jednotlivci, kteří webscraping využívají, musí postupovat tak, aby jejich činnost neporušovala právní předpisy a etické zásady, a to jak ve vztahu k ochranným mechanismům webových stránek, tak k ochraně soukromí jednotlivců.

3.1 Etika

Webscraping je efektivní metoda získávání dat, avšak jeho použití musí být vyváženo odpovědným přístupem k technickým a etickým aspektům. Jedním z hlavních rizik je nadměrné zatížení serveru, které může vést ke zpomalení jeho odezvy nebo dokonce k dočasné nedostupnosti služby. Intenzivní odesílání požadavků v krátkém čase může neúmyslně způsobit narušení provozu webové stránky, a proto je nezbytné omezit frekvenci požadavků a přizpůsobit rozsah scrapingu tak, aby byla zachována dostupnost služby pro ostatní uživatele.

Dodržování technických pravidel je základním předpokladem etického webscrapingu. Klíčovou roli hraje soubor robots.txt, který stanovuje pravidla pro automatizovaný přístup k webovému obsahu. I když jeho existence nebrání technicky scrapingu, jeho ignorování může být považováno za neetické chování. Dalšími osvědčenými postupy jsou zavedení časových prodlev mezi požadavky a provádění scrapingu v době nižšího zatížení serveru, což minimalizuje riziko negativních dopadů na jeho výkon.

Vedle technických aspektů je nutné zohlednit i ochranu soukromí. Kombinací a analýzou získaných dat může dojít k neúmyslnému narušení soukromí jednotlivců, a proto je nezbytné důsledně anonymizovat osobní údaje. Další je scraping obsahu z webových stránek, které obsahují ochranné mechanismy proti automatizovanému sběru dat. V těchto situacích je nutné zvážit nejen právní důsledky, ale také širší společenský dopad takového jednání.

Etika webscrapingu není pouze otázkou dodržení formálních pravidel, ale zahrnuje i odpovědné posouzení konkrétního kontextu. Situační etika hraje významnou roli při rozhodování o tom, zda a jak data získávat, přičemž je nutné zohlednit jak technické, tak etické dopady na provozovatele webu a jeho uživatele. Zodpovědný přístup k webscrapingu vyžaduje nejen

technická opatření ke snížení zátěže serveru, ale také pečlivé zvážení jeho širších důsledků, zejména v souvislosti s veřejným zájmem a ochranou digitálních zdrojů. [12]

3.2 Robots.txt

Soubor robots.txt je jednoduchý textový dokument bez HTML struktury, který je uložen na webovém serveru stejně jako běžné webové soubory. Lze jej obvykle zpřístupnit přidáním přípony /robots.txt za doménu. Tento soubor zpravidla nebývá přímo odkazován na webových stránkách, a proto na něj běžný uživatel nenarazí. Naproti tomu pro webové crawlery je jeho přítomnost klíčová a často jej vyhledávají jako první. Robots.txt slouží jako sada pokynů, které určují, jakým způsobem by se měly automatizované programy, tzv. boti, na daném webu chovat. Je standardní součástí mnoha webových stránek a jeho hlavním úkolem je řízení činnosti legitimních (tzv. „dobrých“) botů, jako jsou například vyhledávací roboti. Tito boti respektují pravidla uvedená v souboru robots.txt, zatímco škodliví (tzv. „špatní“) boti tato pravidla často záměrně ignorují nebo je naopak využívají k identifikaci skrytých částí webu. Je důležité zdůraznit, že soubor robots.txt má pouze doporučující charakter a nemá právní nebo technickou pravomoc vynucovat jeho dodržování. Poskytuje tedy pouze „návod“, kterým by se boti měli řídit. [13]

3.3 GDPR a ochrana osobních údajů

Obecné nařízení o ochraně osobních údajů (General Data Protection Regulation – GDPR) č. 2016/679 je legislativní akt Evropské unie (EU), který upravuje ochranu a zpracování osobních údajů jednotlivců v rámci EU a Evropského hospodářského prostoru (EHP). Hlavními cíli GDPR jsou poskytnutí kontroly jednotlivcům nad tím, jak jsou jejich osobní údaje využívány, a sjednocení pravidel pro ochranu osobních údajů v rámci všech členských států EU, což usnadňuje firmám orientaci v právních předpisech.

Osobně identifikovatelné informace zahrnují veškeré údaje umožňující identifikaci konkrétní osoby. Patří mezi ně například jméno, e-mailová adresa, telefonní číslo, poštovní adresa, číslo kreditní karty, bankovní údaje, IP adresa, datum narození, fotografie, videozáznamy, zvukové záznamy, lékařské záznamy nebo informace o zaměstnání. GDPR přísně reguluje zpracování těchto údajů, pokud se týkají fyzických osob žijících v EU a EHP.

Podle článku 6 GDPR existuje šest právních základů pro zpracování osobních údajů. Prvním je souhlas subjektu údajů, pokud tento subjekt výslovně udělil svolení ke konkrétnímu účelu zpracování. Druhým je plnění smlouvy, kdy je zpracování nezbytné pro splnění smluvních závazků mezi správcem a subjektem údajů. Třetím základem je splnění právní povinnosti, která

správci údajů vyplývá z právních předpisů. Čtvrtým důvodem může být ochrana životně důležitých zájmů fyzické osoby, například v případech zdravotní péče. Pátý právní základ představuje veřejný zájem nebo výkon pravomoci svěřené správci. Poslední možností je oprávněný zájem správce, pokud tento zájem nepřevažuje nad právy a svobodami subjektu údajů.

Pokud je cílem web scrapingu získání pouze neosobních údajů, například textových recenzí pro účely výzkumu a vývoje, nařízení GDPR se na takovou činnost nevztahuje. V případě zpracování osobních údajů, jako jsou jména, věk, geografická poloha nebo jiné identifikátory, je však nutné zajistit soulad s GDPR. Zatímco právní povinnosti, veřejný zájem a ochrana životně důležitých zájmů jsou obvykle jednoznačné, právní základ v podobě oprávněného zájmu může být složitější na prokázání. Z tohoto důvodu je pro většinu organizací obtížné spoléhat se na tento právní základ při provádění web scrapingu. [14]

3.4 Případy soudních sporů o webscraping

Spor mezi společnostmi LinkedIn a hiQ Labs představuje klíčový právní precedent v oblasti web scrapingu a přístupu k veřejně dostupným datům na internetu. LinkedIn, jakožto profesionální sociální síť, dlouhodobě implementuje ochranná opatření proti automatizovanému sběru dat a denně blokuje přibližně 95 milionů pokusů o scrapování. Společnost hiQ Labs, založená v roce 2012, se specializovala na analýzu pracovních trendů a zaměstnaneckých dovedností. Za tímto účelem systematicky scrapovala veřejné uživatelské profily na LinkedIn a získaná data využívala pro své analytické produkty, které poskytovala obchodním klientům.

LinkedIn původně s hiQ udržoval jistou míru spolupráce – jeho zástupci se účastnili odborných konferencí organizovaných hiQ, kde měli možnost se seznámit s jeho produkty a metodologií. Nicméně v roce 2017 LinkedIn oficiálně vyzval hiQ k ukončení web scrapingu s odvoláním na porušení smluvních podmínek platformy a možnou nezákonnost tohoto jednání podle federálního zákona o počítačových podvodech a zneužívání (CFAA), zákona DMCA, kalifornského trestního zákoníku a principů obecného práva. Současně LinkedIn implementoval technická opatření k omezení přístupu hiQ k jeho serverům.

HiQ se proti těmto opatřením ohradil a podal žalobu proti LinkedIn, v níž požadoval soudní ochranu proti omezení přístupu k veřejným údajům. Tvrdil, že LinkedIn zneužívá své dominantní postavení k vytlačení konkurence z trhu analytických nástrojů. Tento případ

vyvolal širší diskuzi o právní regulaci web scrapingu, ochraně digitálních dat a právu na přístup k veřejně dostupným informacím na internetu. [15]

4 Cíl aplikace a požadavky na systém

Tato kapitola se zaměřuje na vymezení hlavního cíle aplikace a specifikuje požadavky, na systém pro zajištění jeho funkčnosti, efektivity a udržitelnosti.

4.1 Cíl aplikace

Cílem aplikace je zajistit spolehlivý a efektivní sběr dat o akčních nabídkách obchodního řetězce Penny Market, a to kombinací technik web scrapingu a přístupu přes API.

Aplikace má sloužit k automatizované extrakci klíčových informací o produktech, jako jsou název, popis, cena, platnost akce a obrázky produktů. Tyto údaje jsou strukturovány do jednotného formátu vhodného pro další zpracování.

Hlavní důraz je kladen na použití metody web scrapingu jako primárního nástroje pro získávání dat z veřejně dostupného webového rozhraní. V případech, kdy není možné některá data touto metodou získat, je využit alternativní způsob prostřednictvím neveřejného API s povoleným přístupem. Tento přístup umožňuje komplexní pokrytí dostupných informací, minimalizuje ztráty dat a zvyšuje spolehlivost řešení.

Výstup aplikace představuje webové rozhraní zaměřené na zobrazení akčních nabídek produktů. U jednotlivých položek jsou dostupné informace o produktu, přičemž součástí je také modul datové analýzy umožňující sledování cenového vývoje a porovnání s podobnými produkty. Aplikace dále zahrnuje rozhraní s úplným přehledem nasbíraných dat, včetně možnosti vyhledávání a filtrování produktů. Návrh klade důraz na modularitu, přehlednost a rozšiřitelnost s cílem usnadnit budoucí rozvoj funkcionality a integraci s dalšími datovými zdroji.

4.2 Funkční požadavky

Funkční požadavky představují základní stavební kámen navrhované aplikace a popisují konkrétní operace, úkoly a interakce, které má systém umožňovat. Zahrnují zejména činnosti přímo související s hlavním účelem aplikace, tedy sběrem, zpracováním, uchováváním a prezentací dat o akčních nabídkách.

4.2.1 Sběr a extrakce dat o produktech

- Aplikace musí automatizovaně extrahovat klíčové informace o produktech z webových stránek Penny Marketu, jako jsou název, popis, cena, platnost akce a obrázky produktů.

- V případě, že není možné získat data pomocí web scrapingu, musí být k dispozici možnost přepnout aplikaci na přístup prostřednictvím API.

4.2.2 Uchovávání dat v databázi

- Aplikace musí zajistit, že v databázi nebudou uchovávány duplicitní informace.
- Historické záznamy o změnách produktů musí být uchovávány v samostatné tabulce. Tato historie musí být dostupná pro pozdější analýzy a nesmí být upravována ani odstraňována bez oprávněného důvodu.

4.2.3 Zobrazení informací o produktech

- Webové rozhraní musí poskytovat zobrazení detailních informací o produktech, včetně názvu, popisu, ceny, platnosti akce a obrázků.

4.2.4 Datová analýza a porovnání produktů

- Aplikace musí obsahovat modul datové analýzy, který umožní:
    - Sledování cenového vývoje vybraných produktů v průběhu času.
    - Porovnání produktů na základě názvu, množství a ceny. Porovnání produktů musí určit, který produkt je výhodnější.
    - Zobrazení kalendáře, který bude obsahovat informace o trvání slev pro každý produkt, včetně minulých, aktuálních a plánovaných akcí.

4.2.5 Vyhledávání produktů

- Aplikace musí umožnit uživatelům vyhledávat produkty podle názvu produktu.
- Vyhledávání musí být intuitivní a rychlé, s možností zúžit výběr podle cenového rozsahu.

4.2.6 Zobrazení seznamu nascrapovaných produktů

- Aplikace musí umožnit zobrazit seznam všech nascrapovaných produktů na samostatné stránce.
    - Tento seznam musí obsahovat název produktu, cenu, platnost akce a možnost přístupu k detailu každého produktu.

4.2.7 Uživatelské rozhraní

- Aplikace musí mít přehledné a intuitivní uživatelské rozhraní, které umožní snadné interakce s aplikací, jako je vyhledávání produktů, analýza cenového vývoje a prohlížení historických dat.

4.3 Nefunkční požadavky

Nefunkční požadavky definují kvalitu, stabilitu a technické vlastnosti systému, které nejsou přímo spojeny s jednotlivými funkcemi, ale významně ovlivňují jeho použitelnost, udržitelnost a bezpečnost. Patří sem aspekty jako výkon při zpracování dat, odolnost vůči chybám, efektivita zvolených technik scrapování nebo bezpečnostní opatření bránící selhání systému.

4.3.1 Spolehlivost systému

- V případě selhání při načítání dat nedochází k ukončení programu s chybou. Namísto toho je vrácen prázdný seznam a uživatel je informován informační zprávou.

4.3.2 Podpora více metod pro získávání dat

- Aplikace umožňuje volbu získávání dat prostřednictvím parametru, který určuje, zda budou data načítána pomocí web scrapingu nebo prostřednictvím API.

4.3.3 Efektivita z hlediska výkonnosti

- Varianta využívající API je z hlediska rychlosti a efektivity výhodnější než web scraping, jelikož přímo pracuje se strukturovanými daty ve formátu JSON (JavaScript Object Notation) a eliminuje potřebu parsování HTML dokumentů.

4.3.4 Bezpečnostní aspekty

- Program nevykonává žádné rizikové operace, jako je spuštění cizího kódu, či přímá manipulace se souborovým systémem nebo databázemi, čímž je minimalizováno riziko výskytu bezpečnostních zranitelností.

5 Architektura a struktura aplikace

Tato kapitola se zabývá celkovým návrhem architektury systému, popisuje jednotlivé vrstvy aplikace, jejich vzájemné vztahy a strukturu zdrojových souborů, na nichž je aplikace postavena.

5.1 Architektura aplikace

Aplikace je koncipována jako webová aplikace, jejímž cílem je automatizovaně sbírat, ukládat a zobrazovat akční nabídky z webových stránek obchodního řetězce Penny Market. Celá architektura je postavena na principu oddělení jednotlivých vrstev a komponent tak, aby bylo dosaženo modularity, přehlednosti a snadné rozšiřitelnosti.

Z hlediska architektonického návrhu lze aplikaci rozdělit do tří hlavních částí:

- Získávání dat – získávání dat buď pomocí web scrapingu nebo z API.
- Datové úložiště – ukládání a správa dat pomocí relační databáze SQLite.
- Webová prezentace – zobrazení informací uživateli prostřednictvím webového rozhraní.

Každá část systému je samostatná a komunikace mezi nimi je zprostředkována prostřednictvím Flask frameworku, který tvoří páteř celé aplikace.

5.1.2 Struktura a komponenty aplikace

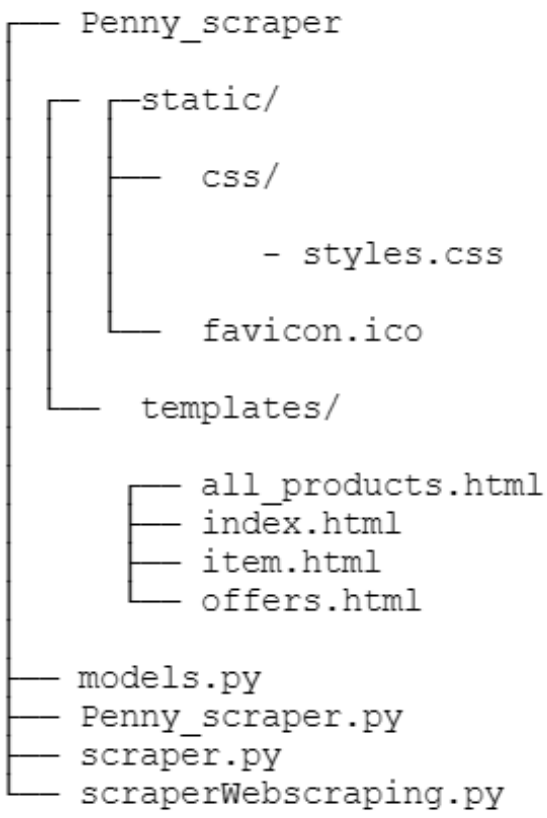
Aby byla výsledná aplikace udržitelná, přehledná a snadno rozšiřitelná, byla při jejím návrhu zvolena architektura založená na oddělení jednotlivých vrstev s jasně definovanými odpovědnostmi. Z hlediska organizace projektu bylo cílem vytvořit takovou strukturu, která bude reflektovat funkční komponenty systému a zároveň umožní jejich jednoduchou správu a další vývoj.

Architektura aplikace se skládá ze čtyř hlavních komponent: modul pro získávání dat (scraper), databázová vrstva, aplikační logika a uživatelské rozhraní. Každá komponenta je implementována pomocí samostatných souborů a složek, jejichž podrobné vysvětlení je uvedeno níže.

Adresářová struktura projektu

Zdrojové soubory aplikace jsou umístěny v kořenovém adresáři Penny_scraper, který tvoří základní organizační rámec celého systému.

Struktura projektu, znázorněná na Obrázku 3, byla navržena s důrazem na srozumitelnost, modulárnost a snadnou údržbu. Každá složka a soubor v rámci této struktury má jasně vymezenou roli a odpovídá specifické funkční vrstvě aplikace.



Obrázek 3: Adresářová struktura aplikace

Moduly pro získávání dat

Moduly pro získávání dat tvoří klíčovou komponentu aplikace, jejichž úkolem je automatizovaný sběr aktuálních informací o produktech z webové stránky obchodního řetězce Penny Market. Získaná data následně vstupují do databázové vrstvy, kde jsou uložena, a dále slouží jako podklad pro vizualizaci, analýzu cenového vývoje.

Sběr dat je realizován dvěma na sobě nezávislými přístupy: parsingem HTML kódu pomocí web scrapingu a voláním API, které poskytuje doplňkové informace.

- **scraperWebscraping.py**

  Tento modul využívá metodu web scrapingu implementovanou pomocí knihovny BeautifulSoup. Principem této techniky je parsování HTML dokumentu, ve kterém jsou pomocí selektorů identifikovány specifické elementy obsahující cílová data (např.

název produktu, cena, platnost akce). Výhodou tohoto přístupu je možnost přímého získání všech informací, které jsou prezentovány uživatelům na stránce, bez závislosti na dostupnosti aplikačního rozhraní.

Tato metoda je považována za primární, neboť umožňuje extrahovat komplexní informace přímo ze struktury stránky a zajišťuje vyšší míru kontroly nad zpracovávaným obsahem. Přesto je třeba počítat s potenciální citlivostí na změny v HTML struktuře, které mohou vyžadovat aktualizaci selektorů.

- **scraper.py**
  Tento modul využívá neveřejné aplikační rozhraní poskytované webem Penny Market, které umožňuje přístup k doplňkovým datům ve formátu JSON. API poskytuje strukturované informace, jež lze zpracovat bez nutnosti analýzy HTML, a představuje efektivní způsob rozšíření datového modelu o další vlastnosti, jako jsou např. obrázky nebo doplňkové informace o produktu.

  Tento přístup je vhodná zejména pro doplnění či ověření údajů získaných pomocí web scrapingu a současně přináší výhody v podobě rychlosti zpracování a menší náchylnosti ke změnám.

Výsledkem obou procesů je naplnění databázových tabulek informacemi o produktech, jako je název, množství, cena (běžná i s věrnostní kartou), období platnosti akce a další doprovodné údaje.

Tyto informace jsou následně využívány aplikační logikou pro zobrazení aktuálních nabídek, porovnávání cen, generování cenových grafů a predikcí budoucích akcí.

Databázová vrstva

Získaná data jsou následně ukládána do databáze, která plní roli úložiště pro jejich další zpracování a vizualizaci. Použitá databáze SQLite je lehké a samostatně fungující řešení, které nevyžaduje instalaci ani správu samostatného serveru a dobře se hodí pro menší webové aplikace.

- **models.py**
  Tento soubor definuje datové modely pomocí SQLAlchemy, která v rámci objektově-relačního mapování (ORM) umožňuje pracovat s databází ve formě Python objektů. Obsahuje dvě klíčové datové třídy:

- Product reprezentuje základní informace o jednotlivých produktech, jako je název, množství, měrná jednotka, krátký popis, URL identifikátor (slug) a případně obrázek. Součástí modelu je také relace na třídu PriceHistory, díky níž je ke každému produktu možné přiřadit více cenových záznamů.
- PriceHistory uchovává historické záznamy o cenách produktů. Každý záznam obsahuje informace o platnosti akce, cenách s a bez věrnostní karty, základní měrné jednotce, základním množství pro přepočet ceny a cenách za standardizovanou jednotku. Každý záznam je navázán na příslušný produkt prostřednictvím cizího klíče.

- **penny_offers.db**
  Samotný databázový soubor SQLite, který vzniká při prvním spuštění aplikace. Uchovává veškerá scrapovaná data, a to ve formátu navrženém v modelu výše. Je možné jej snadno zálohovat, kopírovat nebo analyzovat pomocí nástrojů pro SQLite.

Řízení aplikační logiky pomocí Flasku

Kompletní zpracování dat, jejich zpřístupnění uživatelskému rozhraní a zajištění vzájemné komunikace mezi jednotlivými komponentami má na starosti aplikační logika, která běží pod webovým frameworkem Flask.

Penny_scraper.py

Jedná se o hlavní spustitelný soubor celé aplikace. Obsahuje:

- definici všech rout (např. /offers, /all_products, /item/<id>, atd.),
- inicializaci databáze a napojení modelů,
- import scraperů pro získávání dat přes web scraping nebo API,
- logiku pro zpracování uživatelských požadavků a jejich výstup do HTML šablon pomocí render_template().

Soubor zároveň spojuje jednotlivé části systému dohromady a umožňuje jejich součinnost. Zajišťuje například automatické spuštění scrapování při načtení hlavní stránky nebo uložení získaných dat do databáze. Díky tomu tvoří páteř celého backendu aplikace.

Uživatelské rozhraní

Frontendová část aplikace je tvořena kombinací HTML šablon a kaskádových stylů. Jejím účelem je přehledná a interaktivní prezentace dat koncovému uživateli.

Složka static

Složka obsahuje statické zdroje, které se na straně klienta nemění.

- static/css/styles.css – definuje vzhled aplikace včetně barev, fontů, rozvržení prvků a responzivity.
- favicon.ico – ikona webové aplikace zobrazovaná v záložce prohlížeče.

Složka templates

Tato složka obsahuje HTML šablony, které jsou zpracovávány pomocí systému Jinja2, integrovaného ve frameworku Flask. Umožňuje generovat dynamický obsah prostřednictvím proměnných, cyklů a podmínek. Každý soubor zde odpovídá konkrétní funkci aplikace.

- **index.html** – úvodní stránka s popisem projektu.
- **offers.html** – výpis aktuálních slev s obrázky, množstvím a cenami.
- **all_products.html** – stránka pro vyhledávání a filtrování všech dostupných produktů v databázi.
- **item.html** – detailní stránka konkrétního produktu, zobrazující jeho cenovou historii ve formě grafu, podobné produkty a predikci dalších slevových období.

5.2 Analýza scrapované webové stránky

Aby bylo možné získat potřebná data o akčních nabídkách, je nutné provést analýzu cílové webové stránky, ze které budou informace extrahovány. V tomto případě se jedná o oficiální web obchodního řetězce Penny Market, konkrétně o stránku s přehledem aktuálních slevových akcí. Tato analýza slouží jako základ pro návrh funkčního scraperu, který dokáže automaticky vyhledat, rozpoznat a extrahovat požadované informace bez ručního zásahu.

5.2.1 Popis scrapované stránky

Scraping je prováděn ze stránky https://www.penny.cz/nabidky, která obsahuje seznam produktů v aktuální akční nabídce. Každá položka v seznamu zobrazuje základní údaje o produktu včetně:

- názvu produktu,
- množství,
- ceny bez a s věrnostní kartou,

- platnosti slevy,
- obrázku.

Tato stránka je dynamicky generovaná a jednotlivé produkty jsou zobrazeny v rámci HTML prvků typu <li>, přičemž pro každý produkt je použita stejná struktura a třídy CSS, což usnadňuje automatizované zpracování. Stránka je navržena tak, aby byla přehledná pro běžné uživatele, a proto jsou data prezentována ve vizuálně strukturované a srozumitelné podobě.

Přestože stránka obsahuje některé prvky generované JavaScriptem, klíčová data jsou dostupná přímo ve statickém HTML. Díky tomu lze scraping realizovat bez potřeby nástrojů typu Selenium, které simulují chování skutečného prohlížeče. Pro potřeby projektu je stránka ideálním kandidátem pro použití klasického HTML scrapingu.

5.2.2 Příklad struktury dat na stránce

Pro úspěšné extrahování dat je zásadní pochopit konkrétní strukturu HTML dokumentu. Níže, na obrázku 4, je ukázka části HTML kódu, která odpovídá jedné položce akční nabídky:

```
<li class="ws-product-item-base" data-product-url="grilovaci-klobasa">
    <img src="https://example.com/img/klobasa.jpg" alt="Grilovací klobása">
    <h3 class="ws-product-title">Grilovací klobása</h3>
    <ul class="ws-product-information__piece-description">
        <li>500 g</li>
    </ul>
    <div class="ws-product-price-type__value">69,90 Kč</div>
    <div class="ws-product-price-type__value">59,90 Kč</div>
    <div class="ws-product-price-validity">
        <div>od 15.04.2025</div>
        <div>do 21.04.2025</div>
    </div>
</li>
```

Obrázek 4: Ukázka jedné položky akční nabídky [16]

Z této struktury je patrné:

- Název produktu je umístěn uvnitř značky <h3> s třídou ws-product-title.
- Množství produktu je v <ul> s třídou ws-product-information__piece-description.
- Ceny se nachází v opakujících se s třídou ws-product-price-type__value, kde první je běžná cena a druhá cena s věrnostní kartou.
- Informace o platnosti akce je dostupná v , která obsahuje dvě podznačky – první s datem začátku a druhá s datem konce.

5.3 Získávání dat: Web scraping vs. API

Získávání dat představuje jednu z klíčových součástí celé aplikace. Právě kvalita, úplnost, aktuálnost a přesnost získaných informací přímo ovlivňují výslednou použitelnost systému, jeho funkčnost i celkový přínos pro uživatele.

Cílem bylo navrhnout takové řešení, které bude schopno spolehlivě a opakovaně získávat všechny klíčové informace z veřejného webového rozhraní obchodního řetězce Penny Market, a to i v případě, že dojde ke změnám v interní infrastruktuře webu nebo jeho obsahu.

Jako hlavní technika byla zvolena metoda web scrapingu, která umožňuje automaticky extrahovat data přímo z HTML struktury webové stránky. Tato metoda byla vybrána především kvůli tomu, že zachycuje přesně to, co je zobrazováno běžným návštěvníkům stránek, a tím zajišťuje nejvyšší míru shody s reálnou nabídkou.

Jako doplňkové řešení byl do systému začleněn také přístup k aplikačnímu rozhraní, které poskytuje informace ve strukturované podobě. Obě metody pracují nezávisle na sobě a jejich použití lze přepínat pomocí parametru v požadavku. Tento modulární přístup zajišťuje vyšší spolehlivost celého systému, zvyšuje jeho odolnost vůči změnám a zároveň umožňuje flexibilně reagovat na různé scénáře – ať už jde o dočasnou nedostupnost některého zdroje, nebo o doplňování či ověřování údajů mezi dvěma odlišnými datovými kanály.

5.3.1 Web scraping jako primární nástroj sběru dat

Metoda web scrapingu tvoří základní mechanismus pro získávání dat v této aplikaci. Vzhledem k zadání práce i povaze samotného cílového webu byla tato metoda zvolena jako primární nejen kvůli své nezávislosti na neveřejných službách, ale především díky schopnosti zachytit a extrahovat přesně ta data, která jsou prezentována koncovému uživateli. Aplikace tímto způsobem získává skutečný obsah stránky.

Web scraping v tomto řešení nefunguje pouze jako jednorázový nástroj na „vytažení“ dat, ale jako stabilní a opakovatelný proces, který lze spustit kdykoliv podle potřeby. Díky pravidelnému spouštění přímo při načtení domovské stránky aplikace dochází k automatizované aktualizaci databáze o nové nebo upravené produkty.

V rámci implementace byla zvláštní pozornost věnována tomu, aby byl systém schopen přizpůsobit se i případným změnám v HTML struktuře webu. Součástí návrhu je proto také kontrola dostupnosti dat a validace, která minimalizuje riziko ukládání neúplných nebo chybných údajů. Díky své otevřenosti a přímému přístupu k obsahu stránky je web scraping

v této aplikaci nejen technickým jádrem sběru dat, ale i nástrojem pro dosažení vysoké míry přesnosti a relevance získaných informací.

5.3.2 Použití API jako doplňkového zdroje

Druhou možností sběru dat je přístup k neveřejnému aplikačnímu rozhraní serveru Penny Market. Tato možnost slouží jako doplněk k hlavní scrapovací metodě a umožňuje získat potřebná data v situacích, kdy scraping není dostupný, je neúplný nebo kdy je potřeba efektivně ověřit správnost již uložených údajů.

Na rozdíl od metody web scrapingu poskytuje API přístup k rozšířenému množství informací, které na samotné stránce běžně chybí nebo nejsou přímo dostupné. Patří mezi ně například identifikátor slug pro detailní přístup k produktům, doplňující informace o produktu, odkaz na obrázek produktu nebo ceny za standardizované množství. API tak poskytuje přesně definovaná data, která by při použití samotného web scrapingu bylo obtížné získat nebo by vyžadovala další logiku pro extrakci a interpretaci.

Přestože API nabízí cenné informace a rychlejší přenos dat, jeho využití zůstává záměrně na pozadí. Hlavním důvodem je skutečnost, že se nejedná o oficiálně dokumentované rozhraní určené pro externí aplikace, což přináší riziko nečekaných změn nebo úplné nedostupnosti. Tato metoda je určena pro případy, kdy uživatel nemůže získat potřebná data prostřednictvím web scrapingu.

Z architektonického hlediska je API integrováno jako volitelná součást, kterou lze zvolit prostřednictvím parametru method. Díky tomu je celé řešení modulární a snadno rozšiřitelné, jak pro případné přepínání mezi metodami, tak pro jejich budoucí úpravy. Přítomnost této alternativní metody významně zvyšuje spolehlivost aplikace a rozšiřuje její možnosti v oblasti kontroly, porovnávání a validace dat.

5.4 Ukládání dat do databáze

Aby mohla aplikace efektivně pracovat s daty získanými z webového rozhraní, je nezbytné zajistit jejich bezpečné a přehledné uchování. Databáze zde plní úlohu úložiště, které uchovává detailní záznamy o produktech a jejich cenové historii. Současně poskytuje strukturovaný základ pro vyhledávání, analýzu a vizualizaci. Následující část se věnuje návrhu databázového řešení a popisuje, jakým způsobem jsou jednotlivá data zpracovávána a ukládána.

5.4.1 Výběr databázového systému

Pro účely této aplikace byl zvolen databázový systém SQLite, který představuje jednoduché, ale velmi efektivní řešení pro menší až střední webové projekty. SQLite je tzv. „embedded“ databáze – nevyžaduje instalaci žádného samostatného serveru, protože veškerá data jsou uchovávána v jednom souboru přímo na disku. Tato vlastnost zjednodušuje nasazení i správu celé aplikace, zejména v prostředí bez požadavků na paralelní přístup více uživatelů.

Mezi výhody SQLite patří také její rychlost, malá velikost výsledného souboru, nezávislost na operačním systému a kompatibilita s mnoha programovacími jazyky a frameworky. Z těchto důvodů byla vyhodnocena jako ideální volba pro tuto bakalářskou práci, která je primárně zaměřena na získávání a analýzu dat z jednoho zdroje bez potřeby víceúrovňových oprávnění nebo vícevláknového přístupu k databázi.

Veškerá data – tedy jak základní informace o produktech, tak jejich cenová historie – jsou ukládána do jediného souboru s příponou .db. V této aplikaci je databázový soubor pojmenován penny_offers.db a je automaticky vytvořen při prvním spuštění aplikace.

Aplikace využívá knihovnu SQLAlchemy jako ORM vrstvu pro práci s databází. Tento přístup umožňuje snadnou správu datových modelů a zajišťuje nezávislost na konkrétním typu databázového systému. I když je aktuálně použita databáze SQLite, je možné ji v budoucnu nahradit robustnějším řešením, jako je PostgreSQL nebo MySQL, bez nutnosti zásadních změn v aplikační logice.

5.4.2 Definice a vytvoření databázové struktury

Struktura databáze je specifikována v souboru models.py pomocí knihovny SQLAlchemy, která umožňuje definovat datové modely v podobě tříd jazyka Python. Každá třída představuje jednu tabulku v databázi a jednotlivé atributy těchto tříd odpovídají konkrétním sloupcům v dané tabulce. Tento přístup zajišťuje srozumitelnost a jednotný způsob správy dat napříč celou aplikací.

Po spuštění aplikace dochází automaticky k vytvoření celé databázové struktury. Tento proces zajišťuje volání metody db.create_all(), která na základě deklarovaných tříd vygeneruje odpovídající schéma databáze. Vzhledem k tomu, že není nutné tvořit databázové tabulky ručně, odpadá riziko syntaktických chyb a současně se zjednodušuje správa a případná rozšíření databázového modelu.

Použití objektově-relačního mapování v tomto případě přináší řadu výhod. Umožňuje definovat relace mezi entitami na úrovni kódu, nastavovat omezující podmínky a určovat jedinečnost záznamů na základě kombinací více sloupců. Díky tomu je databázová vrstva aplikace navržena s důrazem na integritu dat, flexibilitu a možnost jejího budoucího rozšiřování bez zásahu do aplikační logiky.

5.4.3 Návrh tabulek a jejich vzájemné vztahy

Aplikace využívá dvě hlavní tabulky: Product a PriceHistory. Tyto tabulky odpovídají dvěma zásadním entitám – produktům a jejich historickým cenovým záznamům.

product:

Tato tabulka obsahuje základní informace o každém produktu:

- jednoznačný identifikátor id,
- název produktu (name),
- volitelný popis (descriptionShort),
- identifikátor slug
- množství (amount),
- zkratku měrné jednotky (volumeLabelShort),
- odkaz na obrázek (image).

price_history:

Tato tabulka obsahuje záznamy o historických cenách, které se vztahují ke konkrétnímu produktu. Každý záznam obsahuje:

- jednoznačný identifikátor id,
- cizí klíč product_id odkazující na tabulku product,
- základní měrná jednotka, ke které se cena vztahuje (price_baseUnitLong),
- množství, ke kterému se vztahuje přepočtená cena (price_basePriceFactor),
- datum začátku platnosti akce (price_validityStart),
- datum konce platnosti akce (price_validityEnd),

- běžná cena bez věrnostní karty (price_regular_value),
- cena s Penny kartou, pokud je dostupná (price_loyalty_value),
- přepočtená jednotková cena bez karty (price_regular_perStandardizedQuantity),
- přepočtená jednotková cena s kartou (price_loyalty_perStandardizedQuantity).

Mezi oběma tabulkami existuje vztah jedna ku mnoha – každý produkt může mít více cenových záznamů. Tato relace je definována pomocí cizího klíče v tabulce price_history, který odkazuje na primární klíč v tabulce product.

Celý návrh odpovídá principům databázové normalizace, zejména druhé a třetí normální formě. Díky tomu nedochází k redundantnímu ukládání stejných informací a jednotlivé části dat jsou snadno spravovatelné a analyzovatelné.

Aby se zabránilo duplicitám, je nad tabulkou price_history definováno kombinované unikátní omezení nad několika sloupci – konkrétně product_id, price_validityStart, price_validityEnd, price_regular_value a price_loyalty_value. Tím je zajištěno, že ke stejnému produktu nemůže být uložen stejný cenový záznam vícekrát.

5.4.4 Získávání, validace a uložení dat

Po získání dat prostřednictvím web scrapingu nebo aplikačního rozhraní dochází k jejich zpracování v rámci datové vrstvy systému. Tato fáze zahrnuje validaci vstupních údajů, ověření jejich konzistence s již uloženými daty a následné rozhodnutí o jejich vložení jako nových záznamů, případně aktualizaci existujících položek v databázi.

Zjištění existence produktu

Nejprve se ověřuje, zda daný produkt již v databázi existuje. Identifikace je založena na kombinaci názvu a množství. Pokud produkt neexistuje, je vytvořen nový záznam v tabulce product. Pokud již existuje, ověří se, zda nedošlo ke změně dalších atributů (např. descriptionShort, image, slug), a v případě potřeby jsou tyto hodnoty aktualizovány.

Kontrola cenové historie

Následně se ověřuje, zda již neexistuje identický záznam v tabulce price_history. Pokud ne, je vytvořen nový záznam s informacemi o platnosti, ceně, přepočtu na jednotku apod.

Formátování a validace

Před uložením do databáze jsou hodnoty upraveny do standardizovaného formátu. Datum je převedeno do českého zápisu dd.mm.yyyy, ceny jsou převedeny na textový formát s desetinnou čárkou, jednotky jsou převedeny na jednotný formát a chybějící hodnoty doplněny výchozími.

Součástí logiky je i validace vstupních dat. Pokud chybí některý klíčový údaj, jako například cena nebo období platnosti, záznam není uložen. Díky tomu se snižuje riziko poškození integrity dat a zvyšuje kvalita výstupů.

5.5 Implementace jednotlivých souborů

Funkčnost celé aplikace je rozdělena mezi několik klíčových souborů, z nichž každý plní specifickou úlohu v rámci jednotlivých vrstev systému. Kód je strukturován tak, aby odpovídal zásadám modularity, čitelnosti a oddělení zodpovědností. Backendová logika je implementována v jazyce Python za využití frameworku Flask a rozšíření SQLAlchemy. Uživatelské rozhraní je tvořeno pomocí HTML šablon s podporou šablonovacího systému Jinja2. V této kapitole jsou jednotlivé soubory podrobně popsány včetně jejich účelu a vzájemných vazeb.

5.5.1 Implementace aplikační logiky v jazyce Python

Aplikační logika zajišťuje propojení mezi uživatelským rozhraním, databází a zdroji dat. Stará se o načítání, zpracování a ukládání informací i jejich předávání do šablon pro zobrazení ve webovém prostředí. V následujících podkapitolách jsou představeny jednotlivé soubory, které tuto logiku zajišťují, včetně jejich účelu a klíčových funkcí.

Penny_scraper.py

Tento soubor představuje hlavní vstupní bod celé aplikace. Obsahuje definici Flask aplikace, konfiguraci databáze, import modelů a jednotlivých modulů pro sběr dat a také definice všech dostupných webových rout. Kromě toho zajišťuje i zpracování uživatelských požadavků a jejich předání do HTML šablon.

Po spuštění dochází k inicializaci aplikace ve frameworku Flask a k propojení s databází prostřednictvím knihovny SQLAlchemy. Součástí je i volání metody db.create_all(), které zajistí vytvoření všech potřebných tabulek v databázi, pokud ještě neexistují.

Mezi definované routy patří:

- / – úvodní stránka s automatickým spuštěním scrapingu,

- /scrape – explicitní spuštění scrapování s možností volby metody (web/api),
- /show_data – náhled aktuálně získaných dat určený primárně pro vývojové a testovací účely
- /offers – zobrazení nascrapovaných produktů s aktuálními i nadcházejícími slevami,
- /all_products – zobrazení všech uložených produktů s možností filtrování,
- /item/<int:product_id> – detailní zobrazení vybraného produktu,
- /search – AJAX (Asynchronous JavaScript and XML) endpoint pro vyhledávání podle názvu a cenového rozsahu.

Soubor rovněž obsahuje pomocné funkce pro formátování dat, jako jsou format_price() a format_date(), které upravují surové vstupy do standardizované podoby. Funkce convert_to_base_unit() slouží k přepočtu množství a jednotek na základní měrné hodnoty, což umožňuje výpočet jednotkových cen a jejich porovnání mezi různými produkty.

models.py

Soubor models.py definuje strukturu databáze pomocí datových tříd Product a PriceHistory, které odpovídají dvěma hlavním tabulkám. Využívá přitom knihovnu SQLAlchemy pro objektově-relační mapování. Každá třída obsahuje jednotlivé atributy, jejich datové typy, vazby mezi tabulkami a případné omezení.

Soubor zajišťuje:

- jednoznačné přiřazení cenové historie ke konkrétnímu produktu,
- uchování všech historických cen v čase,
- přípravu dat pro načítání do rozhraní bez nutnosti psaní SQL dotazů.

Díky použití ORM vrstvy je možné pracovat s databází přímo jako s Python objekty, což výrazně usnadňuje správu dat i jejich integraci do logiky aplikace.

scraperWebscraping.py

Tento soubor obsahuje implementaci hlavní metody web scrapingu pomocí knihovny BeautifulSoup. Jeho úkolem je načíst HTML stránku s akčními nabídkami a z její struktury extrahovat relevantní informace o jednotlivých produktech. Výběr datových prvků je realizován

pomocí metod find() a find_all(), které umožňují cílené vyhledávání HTML elementů na základě jejich tagu a hodnot atributu class.

Mezi klíčové části tohoto souboru patří:

- odeslání HTTP požadavku a načtení HTML obsahu cílové stránky,
- identifikace jednotlivých produktových bloků a extrakce jejich názvu, množství, cen a dalších atributů,
- získání cenových údajů a jejich zpracování s rozlišením mezi běžnou cenou a cenou s věrnostní kartou,
- převod množství a jednotek na standardizovaný formát pomocí mapovacích funkcí,
- sestavení seznamu strukturovaných slovníků s kompletními daty připravenými k uložení do databáze.

Tento modul umožňuje získat všechna potřebná data z veřejně přístupného webového rozhraní bez nutnosti přístupu k aplikačnímu serveru nebo použití pokročilých automatizačních nástrojů.

scraper.py

Získávání dat prostřednictvím tohoto alternativního modulu probíhá pomocí přímé komunikace s aplikačním rozhraním Penny Marketu. Na rozdíl od web scrapingu, který pracuje s HTML strukturou, je zde využito API, jež poskytuje potřebná data ve formátu JSON. To umožňuje přístup ke konkrétním informacím o produktech bez nutnosti parsovat webový obsah.

V rámci této metody probíhá:

- načtení identifikátoru categorySlug z výchozího rozhraní,
- sestavení požadavku na příslušný API endpoint s využitím daného parametru,
- získání kompletní sady dat o produktech v JSON struktuře,
- zpracování vybraných atributů jako je název, množství, jednotka, obrázek, ceny a platnost akce,
- vytvoření seznamu datových záznamů ve formátu připraveném k dalšímu uložení.

Použití tohoto přístupu se v aplikaci uplatňuje jako alternativa ke klasickému web scrapingu. Díky předem strukturovaným datům z API je možné efektivně doplňovat nebo validovat

informace uložené v databázi, a to zejména v případech, kdy webová stránka některé údaje nezobrazuje nebo zobrazuje nesprávně.

5.5.2 Implementace prezentační vrstvy pomocí HTML šablon

Prezentační vrstva aplikace je tvořena soubory HTML šablon, které slouží k dynamickému generování webových stránek pomocí systému Jinja2. Každá šablona odpovídá jedné konkrétní části rozhraní a je přímo propojena s určitou routou ve Flask aplikaci. Do šablon jsou vkládány speciální značky, které umožňují pracovat s předanými daty, například zobrazit hodnotu proměnné, projít seznam produktů nebo vykreslit obsah pouze při splnění určité podmínky.

Při návrhu šablon je využito moderních principů HTML5, který díky zavedení sémantických značek umožňuje přehlednější a strukturálně čistší kód. Tyto prvky přispívají nejen k lepší čitelnosti, ale i ke zlepšení přístupnosti a optimalizaci pro vyhledávače. HTML5 rovněž podporuje multimediální obsah a lepší správu formulářových vstupů, což činí výsledné webové rozhraní kompatibilní s moderními zařízeními. [17]

Každá z následujících šablon plní specifickou funkci v rámci uživatelského rozhraní aplikace:

index.html

Tato šablona slouží jako úvodní stránka aplikace. Obsahuje základní popis projektu a odkazy na další části systému, zejména na zobrazení aktuálních slev a přehled všech nascrapovaných produktů. Její struktura je jednoduchá a je zaměřená na seznámení se s projektem.

offers.html

Šablona slouží k výpisu všech produktů, které mají v databázi nastavenou aktuální nebo budoucí platnost slevy. Produkty jsou zobrazeny ve formě dlaždic s přehledem základních údajů jako je název, množství, cena s a bez věrnostní karty, platnost akce a obrázku. Každá položka obsahuje odkaz na detailní zobrazení produktu.

item.html

Tato šablona zajišťuje zobrazení detailních informací o jednom konkrétním produktu. Uživatel zde vidí název, popis, množství, jednotku, obrázek a aktuální nebo poslední známé ceny, včetně rozlišení mezi běžnou cenou a cenou s Penny kartou. Pokud databáze obsahuje dostatek historických záznamů, je navíc vypočten předpokládaný termín další slevy na základě pravidelnosti předchozích akcí.

Součástí stránky je graf vývoje cen v čase, vytvořený pomocí knihovny Chart.js, a vizualizace slevových období v kalendáři pomocí knihovny FullCalendar.

Stránka dále nabízí srovnání s podobnými produkty a tabulku s porovnáním jednotkových cen, kde jsou výhodnější nabídky vizuálně zvýrazněny. Tím poskytuje uživateli nástroje pro snadné rozhodnutí o nákupu na základě aktuálních i historických dat.

all_products.html

Tato šablona slouží k zobrazení všech produktů dostupných v databázi bez ohledu na jejich aktuální slevový status. Uživatel má možnost vyhledávat produkty podle názvu a omezit výsledky pomocí cenového filtru s posuvníky. Produkty se načítají dynamicky pomocí AJAX dotazů na endpoint /search, což umožňuje vyhledávání a filtrování bez opětovného načítání celé stránky.

Výsledky vyhledávání jsou prezentovány ve formě seznamu, přičemž u každého produktu jsou uvedeny hlavní informace a možnost přejít na jeho detailní zobrazení.

Použití šablon umožňuje vytvářet strukturované a přehledné webové rozhraní, které reaguje na různé typy dat a uživatelských interakcí. Díky jasně oddělené prezentační vrstvě lze upravovat vzhled jednotlivých částí bez zásahu do aplikační logiky. Tato architektura podporuje flexibilní rozvoj systému a zároveň zajišťuje konzistentní způsob zobrazování informací v celé aplikaci.

6 Uživatelská dokumentace

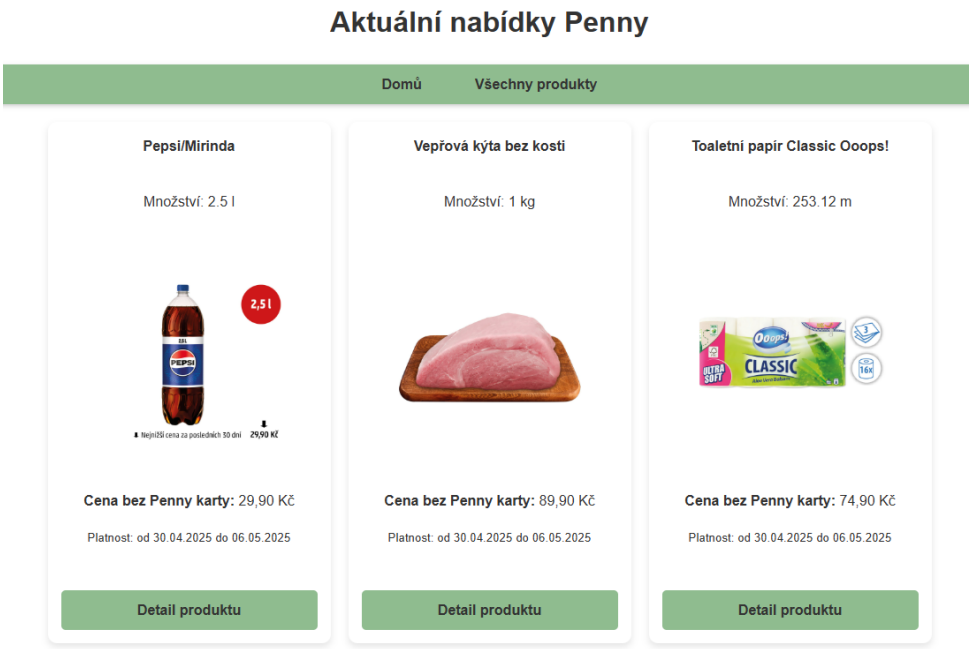
Tato kapitola slouží jako průvodce používáním webové aplikace z pohledu koncového uživatele. Popisuje, jak se v aplikaci orientovat, jaké funkce jsou dostupné a jakým způsobem lze získat přístup k požadovaným informacím. Uživatel je seznámen se všemi stránkami systému, jejich obsahem a ovládacími prvky, které umožňují pohodlné procházení aktuálních i historických dat o produktech.

6.1 Práce s aplikací

Po spuštění aplikace se v prohlížeči otevře úvodní stránka, která slouží jako rozcestník ke všem hlavním částem systému. Stránka obsahuje název projektu, krátký popis a navigační odkazy na sekci s aktuálními nabídkami a na přehled všech produktů, které byly uloženy do databáze.

Obrázek 5: Ukázka úvodní stránky

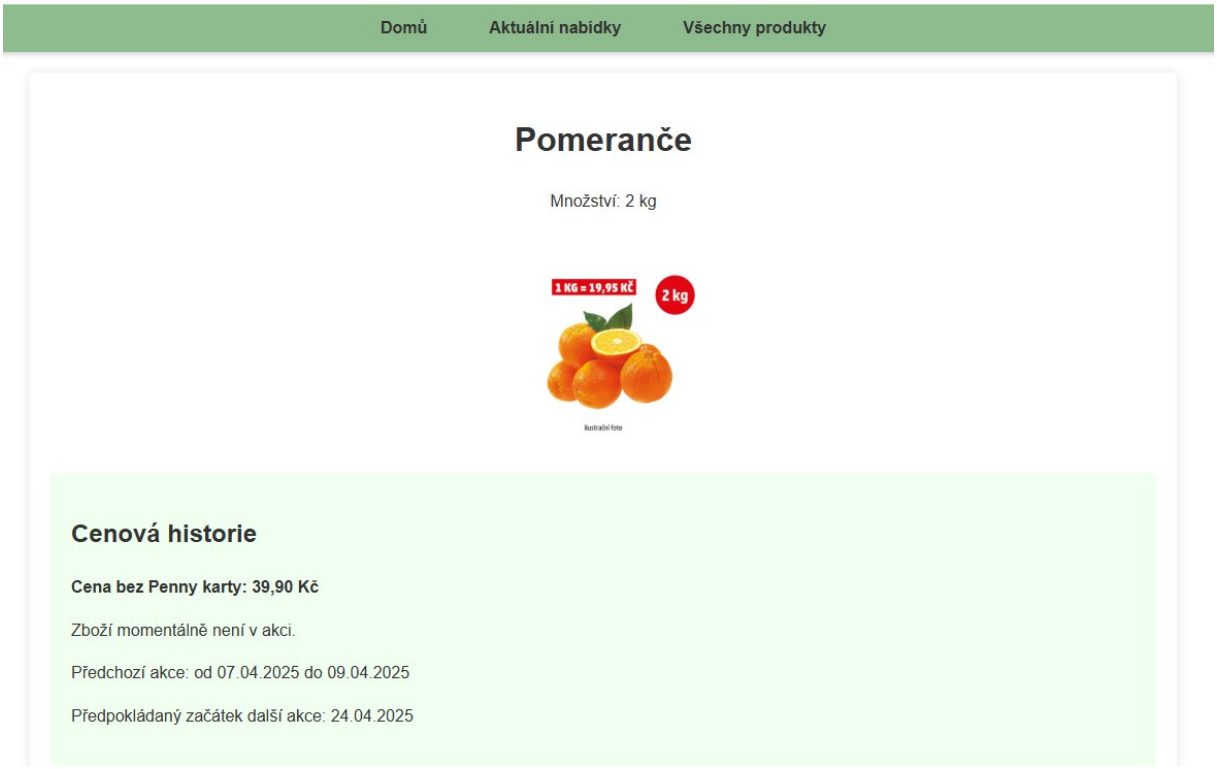
Sekce Aktuální nabídky slouží k přehlednému zobrazení produktů, u nichž byla detekována sleva, která právě platí nebo teprve začne. Každý produkt je zobrazen jako samostatná dlaždice s obrázkem, názvem, množstvím, cenou a obdobím platnosti akce. Uživatel může kliknutím na tlačítko nebo obrázek přejít na podrobnosti o konkrétním produktu.



Obrázek 6: Ukázka stránky s akčními produkty

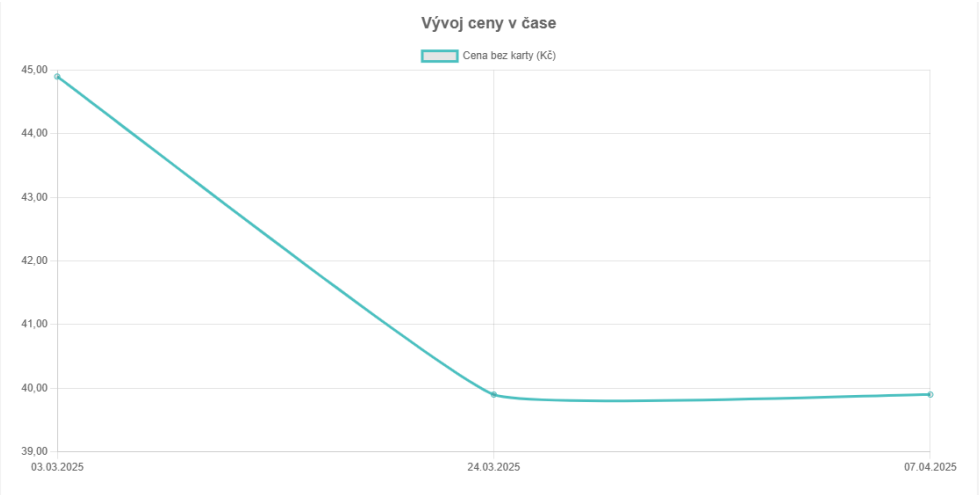
Po kliknutí na konkrétní produkt se zobrazí stránka s detailem produktu. V horní části stránky je zobrazen název produktu, jeho stručný popis a množství s jednotkou. Uživatel má rovněž k dispozici ilustrační obrázek, který usnadňuje vizuální orientaci.

Pod hlavními informacemi se nachází přehled cenové historie. Uživatel je informován o aktuální ceně bez i s Penny kartou, případně o tom, zda je produkt právě v akci. Zobrazena je i předchozí sleva a pokud to datová historie umožňuje zobrazí se i odhad další akční nabídky.



Obrázek 7: Ukázka detailní stránky – informace o produktu

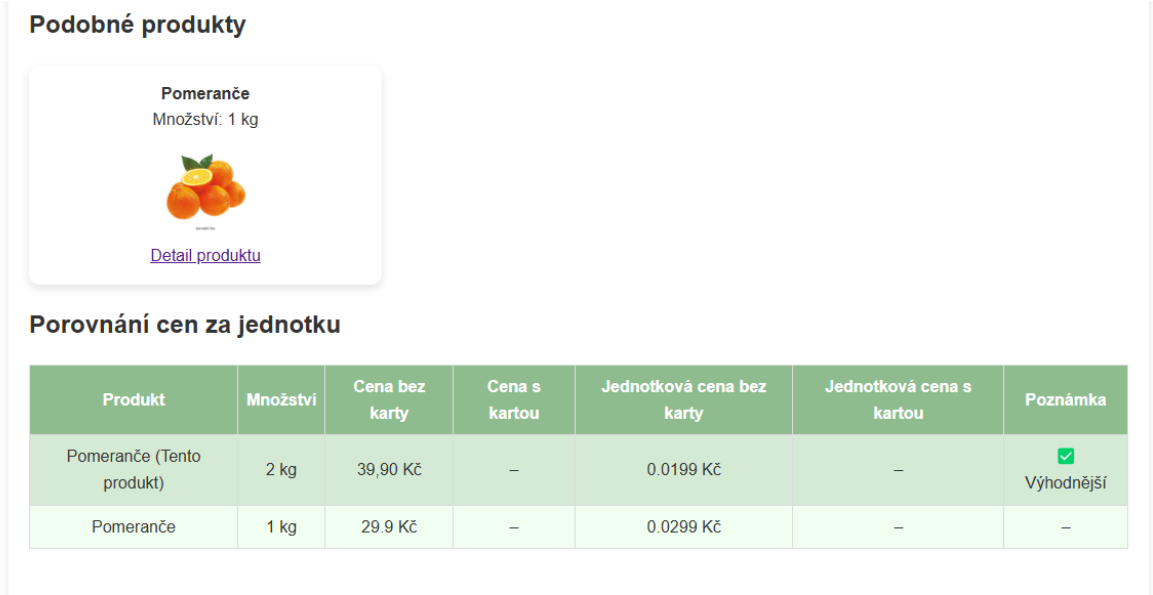
Pro snadnější přehled o vývoji cen je k dispozici graf, který zobrazuje, jak se měnila cena daného produktu v průběhu času. Uživatel zde uvidí jednotlivá období slev a odpovídající ceny, ať už běžné nebo s Penny kartou. Díky tomu si může snadno udělat představu, kdy byl produkt naposledy levnější nebo zda jeho cena dlouhodobě roste, klesá nebo kolísá.



Obrázek 8: Ukázka detailní stránky – graf cenového vývoje

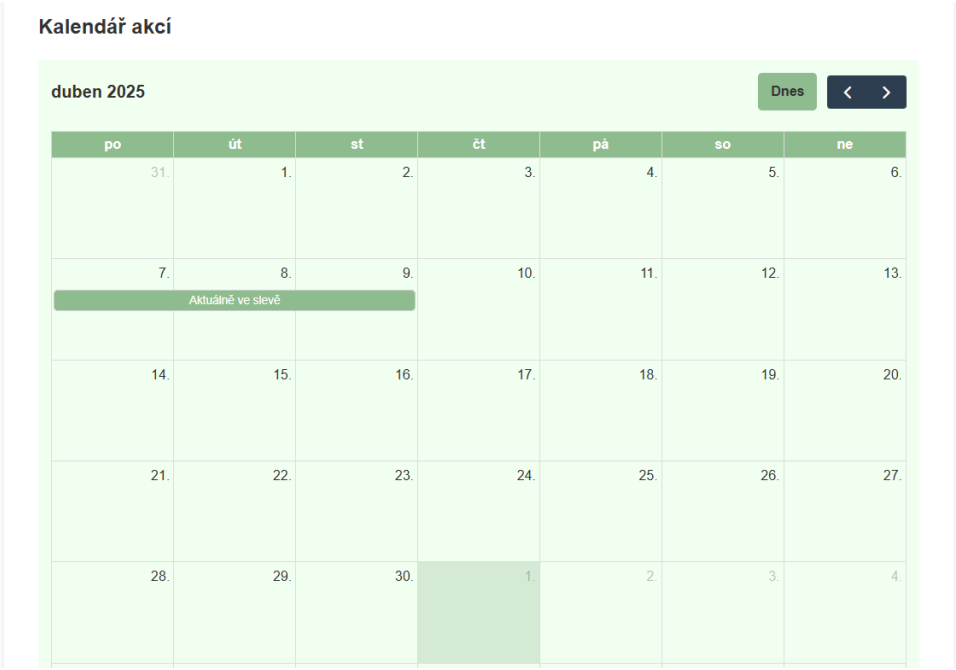
Dále aplikace automaticky vyhledá produkty se stejným názvem, ale odlišným množstvím. Tyto produkty jsou zobrazeny jako dlaždice s obrázkem, množstvím a tlačítkem pro přechod na jejich vlastní detail.

Důležitou funkcí stránky je také možnost porovnání jednotkových cen. Nabídky, které jsou cenově nejvýhodnější, jsou přehledně označeny barevným podkladem a ikonou, což uživateli usnadňuje rozhodování o nákupu.



Obrázek 9: Ukázka detailní stránky – porovnání s podobnými produkty

Na konec stránky je umístěn kalendář, který přehledně zobrazuje dobu trvání všech slev včetně těch, které již proběhly, právě probíhají nebo jsou plánovány do budoucna. Uživatel má možnost listovat mezi jednotlivými měsíci a kdykoliv se vrátit k aktuálnímu datu pomocí tlačítka „Dnes“.

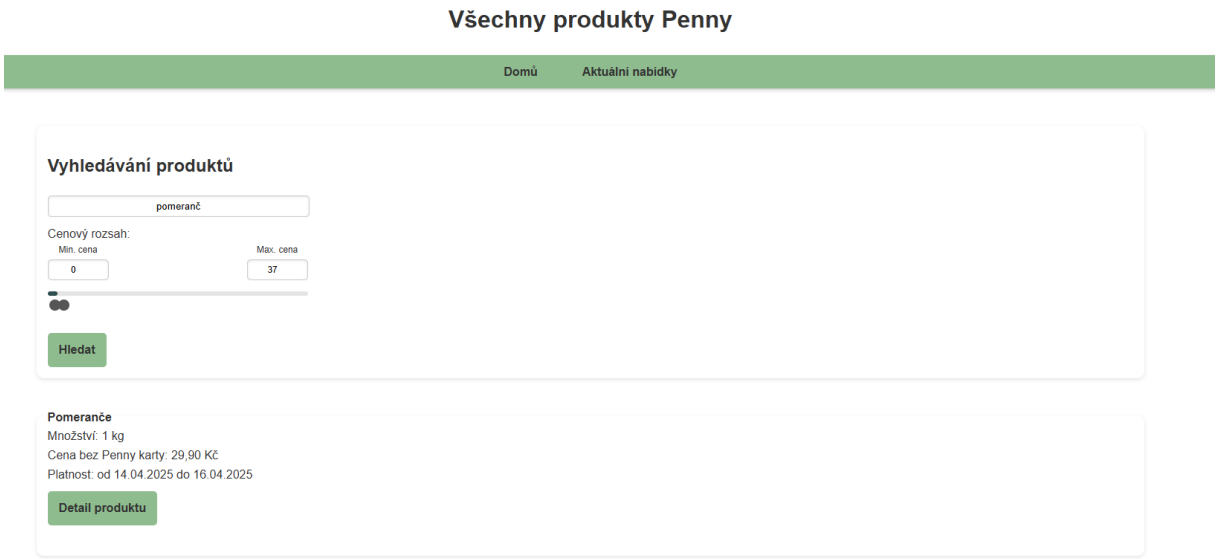


Obrázek 10: Ukázka detailní stránky – kalendář akcí

Tato stránka slouží k přehlednému procházení všech produktů, které byly v minulosti zařazeny do akčních nabídek. V horní části se nachází vyhledávací pole, do kterého lze zadat název produktu. Pod ním je cenový filtr, pomocí kterého si uživatel může nastavit minimální a maximální cenu produktů, které ho zajímají.

Po kliknutí na tlačítko Hledat se zobrazí seznam odpovídajících produktů. U každé položky jsou viditelné klíčové informace jako název, množství, cena bez Penny karty, případně i cena s kartou a období, kdy byl produkt v akci. Pokud chce uživatel zjistit více, může kliknout na tlačítko Detail produktu a otevře se stránka s podrobnějšími informacemi.

Celá stránka je navržena tak, aby se s ní snadno pracovalo. Uživatel může pohodlně vyhledávat, filtrovat a prohlížet si jednotlivé produkty bez obnovování stránky.



Obrázek 11: Ukázka stránky se všemi produkty – filtrování

ZÁVĚR

Tato bakalářská práce se zabývala využitím web scrapingu pro automatizovaný sběr dat z webových stránek obchodního řetězce a jejich následné zpracování a prezentaci uživatelům prostřednictvím webové aplikace. V teoretické části byly popsány principy této techniky, její využití v praxi, dostupné nástroje a také právní a etické souvislosti, které je při jejím použití nutné zohlednit.

Na základě těchto poznatků byla navržena a implementována webová aplikace, která umožňuje uživatelům získávat aktuální informace o akčních produktech, ukládat je do databáze a přehledně je zobrazovat. Systém nabízí také základní analytické funkce, jako je sledování cenového vývoje a porovnání produktů.

Během vývoje a testování aplikace byl kladen důraz na šetrný přístup k serverům obchodního řetězce. Byly nastaveny intervaly mezi požadavky a respektována pravidla uvedená v souboru robots.txt. V průběhu práce nedošlo k nadměrnému zatížení cílových serverů ani k narušení jejich provozu.

Výsledné řešení je navrženo modulárně a umožňuje snadné rozšíření o další datové zdroje nebo nové funkce. Mezi hlavní omezení patří závislost na struktuře cílových webových stránek, což může v případě jejich změny vyžadovat úpravu scrapovací logiky. Dalším limitem je aktuálně omezený počet podporovaných obchodních řetězců.

Práce naplnila stanovené cíle a ukázala, že web scraping zůstává i v současném prostředí relevantním a prakticky využitelným přístupem ke zpracování veřejně dostupných informací. Výsledná aplikace poskytuje pevný základ pro další rozvoj, například rozšíření o notifikace na nové akce, pokročilejší analytické nástroje nebo podporu dalších obchodních řetězců. Projekt zároveň přispěl k rozvoji mých znalostí v oblasti webových technologií, datové analýzy a etického přístupu k automatizovanému sběru dat.

POUŽITÁ LITERATURA

[1] SIRISURIYA, S.C.M. de S. A Comparative Study on Web Scraping. Online. In: *Proceedings of 8th International Research Conference, KDU*. Ratmalana, Sri Lanka, 2015, s. 135-139. Dostupné z: http://192.248.104.6/bitstream/handle/345/1051/com-059.pdf?sequence=1&isAllowed=y&fbclid=IwZXh0bgNhZW0CMTEAAR1kk1yUBUorzeJXQg0VfgJj9JpzI9PVgtN1WPwvkFr20hL4JhTJkqweLxU_aem_kxqRJEHFjSPMlBb33DY9Ig. [cit. 2025-05-03].

[2] KHDER, Moaiad Ahmad. Web Scraping or Web Crawling: State of Art, Techniques, Approaches and Application: State of Art, Techniques, Approaches and Application. Online. *International Journal of Advances in Soft Computing and its Applications*. 2021, vol. 13, no. 3, s. 145–168. ISSN 20748523. Dostupné z: https://doi.org/10.15849/IJASCA.211128.11. [cit. 2025-05-04].

[3] BARRETT, Ansel. *What is Web Scraping and How Does It Work*. Online. Octoparse. 2018. Dostupné z: https://www.octoparse.com/blog/web-scraping-introduction. [cit. 2025-05-04].

[4] ZHAO, Bo. Web Scraping. Online. In: SCHINTLER, Laurie A. a MCNEELY, Connie L. (ed.). *Encyclopedia of Big Data*. Cham: Springer International Publishing, 2017, s. 1-3. ISBN 978-3-319-32001-4. Dostupné z: https://doi.org/10.1007/978-3-319-32001-4_483-1. [cit. 2025-05-04].

[5] *The Python Tutorial*. Online. Python. C2001-2025. Dostupné z: https://docs.python.org/3/tutorial/index.html. [cit. 2025-05-04].

[6] WIERINGA, Jeri. Intro to Beautiful Soup. Online. *Programming Historian*. 2012. Dostupné z: https://doi.org/10.46430/phen0008. [cit. 2025-05-03].

[7] HAJBA, Gábor László. *Website scraping with Python: using Beautifulsoup and Scrapy*. New York: Apress, [2018]. ISBN 978-1-4842-3924-7.

[8] MITCHELL, Ryan. *Web Scraping with Python: Collecting More Data from the Modern Web*. 2. O'Reilly Media, 2018. ISBN 9781491985526.

[9] *Jsoup*. Online. C2009 - 2025. Dostupné z: https://jsoup.org. [cit. 2025-05-04].

[10] BERTOLINO, Antonia; FARIA, João Pascoal a LAGO, Patricia, SEMINI, Laura (ed.). *Quality of Information and Communications Technology*. Online. Springer Nature, 2024. ISBN 9783031702457. Dostupné z: https://books.google.cz/books?id=pY0gEQAAQBAJ. [cit. 2025-05-04].

[11] *How to use HTML Agility Pack ?* Online. GeeksforGeeks. Dostupné z: https://www.geeksforgeeks.org/how-to-use-html-agility-pack/. [cit. 2025-05-04].

[12] LUSCOMBE, Alex; DICK, Kevin a WALBY, Kevin. Algorithmic thinking in the public interest: navigating technical, legal, and ethical hurdles to web scraping in the social sciences: navigating technical, legal, and ethical hurdles to web scraping in the social sciences. Online. *Quality & Quantity*. 2022, vol. 56, no. 3, s. 1023-1044. ISSN 0033-5177. Dostupné z: https://doi.org/10.1007/s11135-021-01164-0. [cit. 2025-05-04].

[13] *What is robots.txt? | How a robots.txt file works*. Online. CLOUDFLARE, INC. Cloudflare. C2025. Dostupné z: https://www.cloudflare.com/learning/bots/what-is-robots-txt/. [cit. 2025-05-04].

[14] BARRETT, Ansel. *GDPR Compliance In Web Scraping*. Online. Octoparse. 2021. Dostupné z: https://www.octoparse.com/blog/gdpr-compliance-in-web-scraping. [cit. 2025-05-04].

[15] UNITED STATES COURT OF APPEALSFOR THE NINTH CIRCUIT. *HIQ LABS, INC. V. LINKEDIN CORPORATION, No. 17-16783*. Online. 2022. Dostupné také z: https://law.justia.com/cases/federal/appellate-courts/ca9/17-16783/17-16783-2022-04-18.html.

[16] PENNY MARKET S.R.O. *Akční nabídka*. Online. PENNY MARKET S.R.O. Penny. Dostupné z: https://www.penny.cz/nabidky. [cit. 2025-05-04].

[17] FRAIN, Ben. *Responsive web design with HTML5 and CSS: develop future-proof responsive websites using the latest HTML5 and CSS techniques*. 4. Packt Publishing. ISBN 9781803242712.

SEZNAM PŘÍLOH