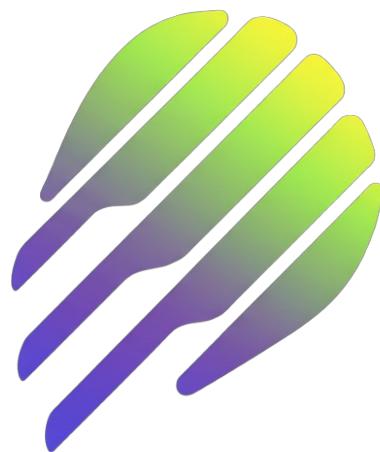


ZÁVĚREČNÁ STUDIJNÍ PRÁCE

dokumentace

Plutoa - Webová aplikace pro správu osobních financí



Autor: Tomáš Holeš
Obor: 18-20-M/01 INFORMAČNÍ TECHNOLOGIE
se zaměřením na počítačové sítě a programování
Třída: IT4
Školní rok: 2025/26

Poděkování

Rád bych poděkoval všem, kteří mě podporovali během tvorby této práce. Zvláštní poděkování patří mým učitelům za cenné rady a připomínky k projektu.

Prohlášení

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

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

V Opavě 15. 12. 2025

.....
Podpis autora

Abstrakt

Tato závěrečná práce se zabývá vývojem moderní webové aplikace Plutoa určené pro komplexní správu osobních financí. Aplikace je postavena na architektuře client-server s odděleným frontendem v React s TypeScriptem a backendem v Django s Django REST Framework. Hlavními funkcemi aplikace jsou sledování příjmů a výdajů s kategorizací, vytváření a monitoring rozpočtů, definování finančních cílů s vizualizací pokroku a pokročilé analytické nástroje včetně Financial Health Score. Aplikace nabízí intuitivní uživatelské rozhraní s interaktivními grafy, heatmap kalendářem a waterfall chartem pro vizualizaci cash flow. Bezpečnost je zajištěna JWT autentizací s automatickou rotací tokenů. Práce popisuje kompletní proces návrhu, implementace a testování aplikace včetně databázového schématu a API endpointů.

Klíčová slova

Osobní finance, webová aplikace, React, Django, REST API, TypeScript, rozpočty, analytika, JWT autentizace

Abstract

This thesis deals with the development of a modern web application Plutoa designed for comprehensive personal finance management. The application is built on a client-server architecture with a separated frontend in React with TypeScript and backend in Django with Django REST Framework. The main features include tracking income and expenses with categorization, budget creation and monitoring, defining financial goals with progress visualization, and advanced analytical tools including Financial Health Score. The application offers an intuitive user interface with interactive charts, heatmap calendar, and waterfall chart for cash flow visualization. Security is ensured by JWT authentication with automatic token rotation. The work describes the complete process of design, implementation, and testing of the application including database schema and API endpoints.

Keywords

Personal finance, web application, React, Django, REST API, TypeScript, budgets, analytics, JWT authentication

Obsah

Úvod	2
1 Analýza a návrh aplikace	3
1.1 Řešený problém	3
1.2 Návrh databázového schématu	3
1.2.1 Hlavní entity	4
1.3 Návrh architektury	4
2 Použité technologie	5
2.1 Backend technologie	5
2.2 Frontend technologie	5
3 Implementace backendu	6
3.1 Struktura Django projektu	6
3.2 Model uživatele	6
3.3 Autentizace	6
3.4 REST API endpointy	8
3.5 Analytické funkce	8
3.5.1 Financial Health Score	8
3.5.2 Trend Analysis	9
3.5.3 Insights	9
4 Implementace frontendu	11
4.1 Struktura React aplikace	11
4.2 Hlavní komponenty	11
4.2.1 App.tsx	11
4.2.2 AuthContext	11
4.2.3 Overview	11
4.2.4 Další klíčové stránky	12
4.3 Vizualizace dat	13
4.4 Responsivní design	14
5 Pokročilé funkce	15
5.1 Správa rozpočtů	15
5.2 Finanční cíle	15
5.3 Opakující se transakce	15
5.4 Import a export dat	16

5.5 Systém notifikací	16
Závěr	17

ÚVOD

Správa osobních financí je v dnešní době důležitější než kdy dříve. Mnoho lidí nemá přehled o tom, kam jejich peníze směřují, a často se dostávají do situací, kdy na konci měsíce zjistí, že utratili více, než původně plánovali. Existující aplikace pro správu financí na trhu často postrádají pokročilé analytické funkce nebo mají komplikované uživatelské rozhraní, které odrazuje běžné uživatele. Mnoho populárních aplikací jako Mint nebo YNAB není plně lokalizováno pro české prostředí.

Cílem této závěrečné práce je navrhnout a implementovat moderní webovou aplikaci **Plutoa** pro komplexní správu osobních financí. Aplikace kombinuje intuitivní uživatelské rozhraní s pokročilými analytickými nástroji včetně Financial Health Score, heatmap kalendáře a waterfall grafů pro vizualizaci cash flow. Dílčími cíli jsou implementace bezpečné JWT autentizace, vytvoření responsivního frontendu v Reactu s TypeScriptem a návrh databázového schématu optimalizovaného pro finanční data.

První kapitola se zabývá analýzou požadavků a návrhem aplikace včetně databázového schématu. Druhá kapitola popisuje použité technologie. Třetí kapitola rozebírá implementaci backendu včetně autentizace a API endpointů. Čtvrtá kapitola se věnuje implementaci frontendu a vizualizacím dat. Pátá kapitola představuje pokročilé funkce jako rozpočty, finanční cíle a systém notifikací. V závěru jsou shrnutы dosažené výsledky a nastíněny možnosti dalšího rozvoje.

1 ANALÝZA A NÁVRH APLIKACE

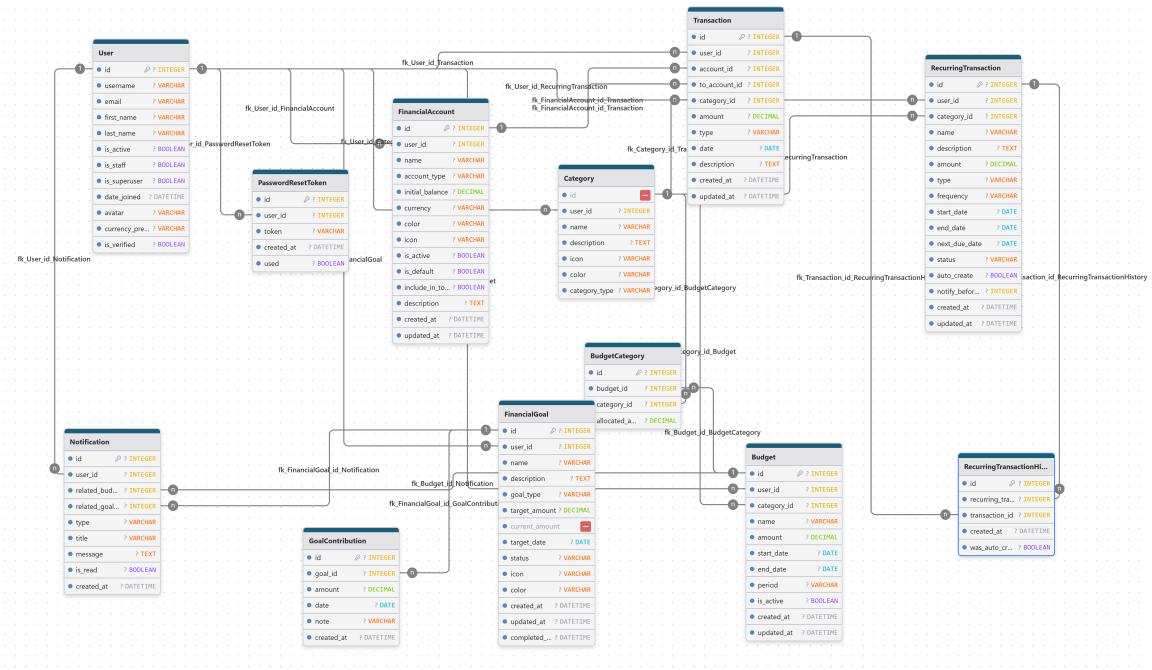
Návrh aplikace zahrnoval analýzu existujících řešení a vytvoření databázového schématu optimalizovaného pro rychlé dotazy a flexibilitu.

1.1 ŘEŠENÝ PROBLÉM

Existující aplikace pro správu financí (Mint, YNAB, Spendee) mají nedostatky v oblasti komplexní analytiky a vizualizace dat. Plutoa přináší pokročilé analytické nástroje včetně heatmap kalendáře, waterfall grafů a Financial Health Score.

1.2 NÁVRH DATABÁZOVÉHO SCHÉMATU

Databázové schéma bylo navrženo s ohledem na normalizaci, rychlosť dotazů a integritu dat. Aplikace využívá relační databázi s pěti hlavními entitami, které jsou propojeny cizími klíči.



Obrázek 1.1: ER diagram databázového schématu aplikace Plutoa

1.2.1 Hlavní entity

Tabulka 1.1: Přehled klíčových atributů databázových entit

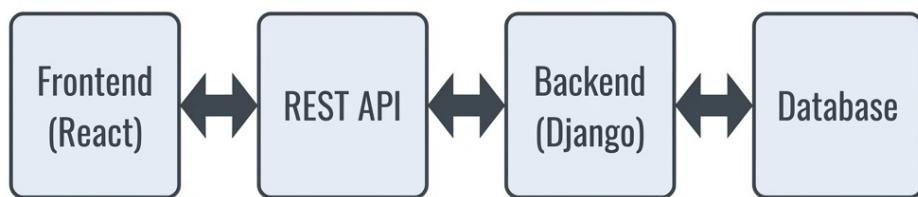
Entita	Klíčové atributy	Popis
User	username, email, currency_preference	Uživatelský účet s preferencemi
Account	name, type, balance, currency	Finanční účet (běžný, spořící...)
Category	name, icon, color, category_type	Kategorie pro třídění transakcí
Transaction	amount, type, date, category	Finanční transakce (příjem/výdaj)
Budget	name, amount, period, category	Rozpočet pro sledování výdajů
FinancialGoal	name, target_amount, target_date	Finanční cíl uživatele

Model **User** rozšiřuje Django AbstractUser o atributy jako avatar a preferovaná měna (výchozí CZK). Entita **Account** (FinancialAccount) reprezentuje reálné účty uživatele (běžný účet, hmotnost, spoření, investice) a sleduje jejich aktuální zůstatek. Každý uživatel má vlastní **Category** pro třídění transakcí s ikonou a barvou. Entita **Transaction** eviduje příjmy, výdaje a převody mezi účty. **Budget** umožňuje nastavit limity pro různá období (měsíční, roční, vlastní). **FinancialGoal** sleduje pokrok při dosahování finančních cílů.

1.3 NÁVRH ARCHITEKTURY

Byla zvolena architektura **client-server** s odděleným frontendem a backendem. Hlavní komponenty:

- **Frontend** – Single Page Application (SPA) v Reactu s TypeScriptem.
- **Backend** – REST API v Django s Django REST Framework.
- **Komunikace** – HTTP/JSON přes Axios s JWT autentizací.
- **Databáze** – SQLite pro vývoj, PostgreSQL pro produkci.



Obrázek 1.2: Diagram architektury aplikace Plutoa

2 POUŽITÉ TECHNOLOGIE

V následujících sekcích jsou popsány hlavní použité technologie, které byly zvoleny s ohledem na moderní trendy a škálovatelnost aplikace.

2.1 BACKEND TECHNOLOGIE

Backend aplikace je vybudován na frameworku **Django 5.2**, který byl zvolen pro svou rychlosť vývoje. Django poskytuje integrovaný ORM (Object-Relational Mapping) pro efektivní práci s databází a abstrakci nad SQL dotazy.

Pro tvorbu RESTful API byl využit **Django REST Framework (DRF) 3.16**. Tento toolkit umožňuje efektivní serializaci dat z Django modelů do formátu JSON a poskytuje nástroje pro správu viewsetů a routerů. Autentizace je zajištěna knihovnou **SimpleJWT 5.5**, která implementuje standard JSON Web Token (JWT) pro bezstavové ověřování uživatelů.

Jako databázový systém pro produkční prostředí slouží **PostgreSQL 16**, který nabízí pokročilé funkce a transakční integritu. Pro vývojové účely byla využita databáze **SQLite**.

2.2 FRONTEND TECHNOLOGIE

Klientská část je implementována jako Single Page Application (SPA) pomocí knihovny **React 19**.

Pro zvýšení kvality kódu byl zvolen jazyk **TypeScript 4.9**. Statické typování pomáhá předcházet chybám již v době komplikace a výrazně zlepšuje našeptávání (IntelliSense) v editoru kódu.

Vizualizace dat jsou realizovány pomocí knihovny **Recharts 3.4**, která je postavena na SVG elementech a poskytuje flexibilní sadu komponent pro tvorbu responsivních grafů (např. LineChart, BarChart, PieChart). Komunikaci se serverem zajišťuje **Axios 1.13**, promise-based HTTP klient, který zjednoduší práci s požadavky a umožňuje globální konfiguraci interceptorů pro práci s tokeny.

Pro konzistentní vzhled ikon v celé aplikaci je využita knihovna **Lucide React**, která poskytuje sadu moderních a lehkých SVG ikon.

3 IMPLEMENTACE BACKENDU

Tato kapitola popisuje implementaci serverové části aplikace. Backend je zodpovědný za zpracování dat, autentizaci uživatelů, business logiku a komunikaci s databází. Implementace je rozdělena do několika Django aplikací, z nichž každá má jasně definovanou odpovědnost.

3.1 STRUKTURA DJANGO PROJEKTU

Backend je rozdělen do Django aplikací: accounts, transactions, budgets, goals, analytics a notifications.

3.2 MODEL UŽIVATELE

Vlastní model uživatele rozšiřuje Django AbstractUser o pole jako avatar, currency_preference a is_verified. Toto rozhodnutí umožňuje flexibilitu pro budoucí rozšíření aplikace.

3.3 AUTENTIZACE

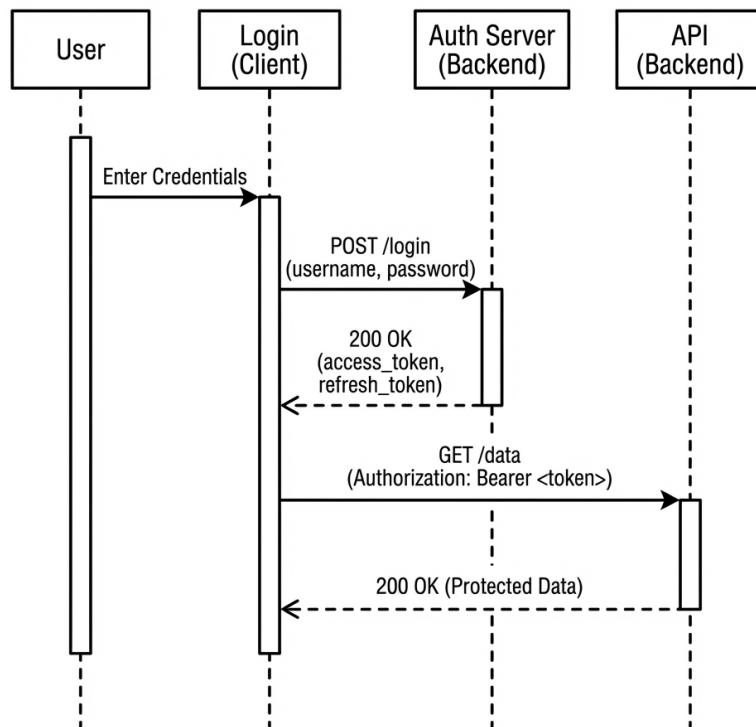
Autentizace je implementována pomocí JWT tokenů. Při přihlášení uživatel obdrží access token (krátkodobý, 5 minut) a refresh token (dlouhodobý, 7 dní). Access token se používá pro autorizaci API požadavků a je automaticky přikládán ke každému požadavku na chráněné endpointy.

Systém také podporuje obnovu zapomenutého hesla pomocí jednorázových tokenů (PasswordResetToken), které jsou zasílány na email uživatele a mají platnost 1 hodinu. Uživatelské rozhraní pro registraci a přihlášení je navrženo s důrazem na jednoduchost a bezpečnost.

The image shows two dark-themed user interface screens. The left screen is titled "Registrace" (Registration) and contains four input fields: "Uživatelské jméno" (User name), "Váš email" (Your email), "Heslo" (Password), and "Potvrďte heslo" (Confirm password). Below these is a large white button labeled "Registrovat se" (Register). At the bottom, it says "Už máte účet? Přihlásit se" (Already have an account? Log in). The right screen is titled "Přihlášení" (Login) and contains two input fields: "Uživatelské jméno" (User name) and "Heslo" (Password). It also includes a checkbox for "Zůstat přihlášen" (Stay logged in) and a link "Zapomenuté heslo?" (Forgot password?). Below these is a large white button labeled "Přihlásit se" (Log in). At the bottom, it says "Nemáte účet? Zaregistrujte se" (Don't have an account? Register).

Obrázek 3.1: Registrování nového uživatele

Obrázek 3.2: Přihlašovací formulář pro existující uživatele



Obrázek 3.3: Sekvenční diagram JWT autentizace

3.4 REST API ENDPOINTY

Aplikace poskytuje následující hlavní API endpointy:

Tabulka 3.1: Přehled API endpointů

Endpoint	Metoda	Popis
/api/accounts/login/	POST	Přihlášení uživatele
/api/accounts/register/	POST	Registrace uživatele
/api/accounts/users/me/	GET	Profil přihlášeného uživatele
/api/transactions/	GET, POST	Seznam a vytváření transakcí
/api/categories/	GET, POST	Seznam a vytváření kategorií
/api/budgets/	GET, POST	Seznam a vytváření rozpočtů
/api/goals/	GET, POST	Seznam a vytváření cílů
/api/analytics/overview/	GET	Analytický přehled
/api/analytics/health-score/	GET	Financial Health Score

3.5 ANALYTICKÉ FUNKCE

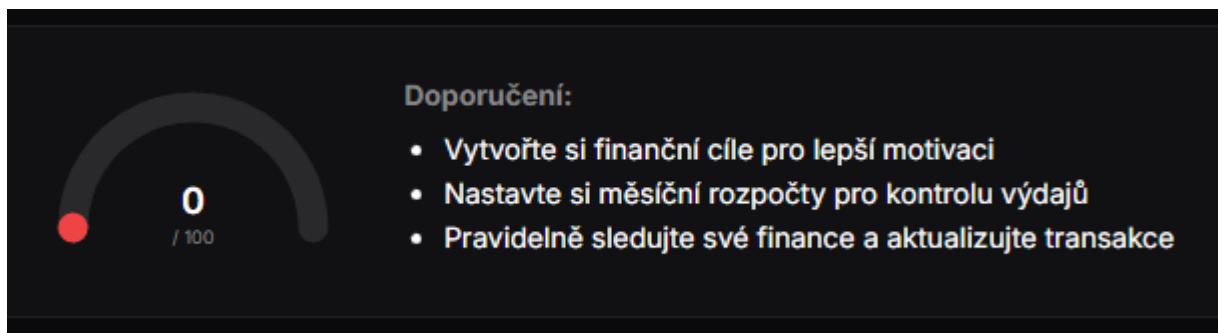
Backend poskytuje pokročilé analytické funkce.

3.5.1 Financial Health Score

Algoritmus pro výpočet finančního zdraví (0-100 bodů) zohledňuje:

- Poměr příjmů a výdajů (savings rate).
- Plnění rozpočtů.
- Pravidelnost příjmů.
- Diverzifikaci výdajů.

Výsledné skóre je vizualizováno barevným grafem a doplněno o konkrétní textová doporučení, jak finanční zdraví zlepšit (např. „Zvyšte úspory“ nebo „Pozor na překračování rozpočtů“).

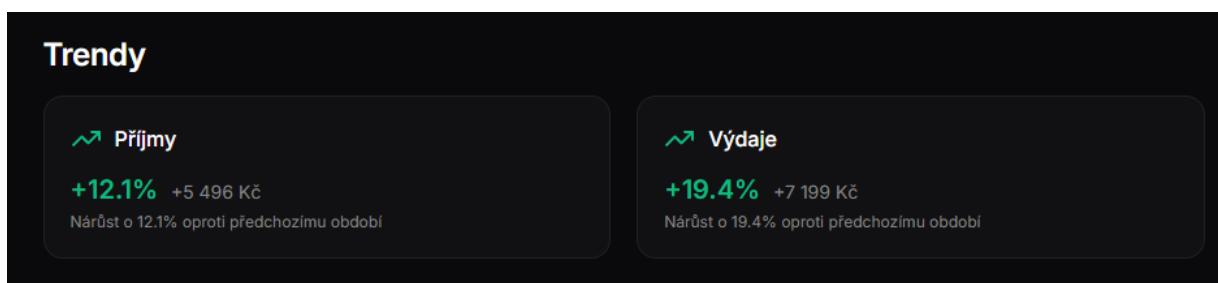


Obrázek 3.4: Detail Financial Health Score s rozpisem jednotlivých komponent

3.5.2 Trend Analysis

Analýza trendů v jednotlivých kategoriích porovnává aktuální období s předchozím a detektuje:

- Rostoucí trendy (výdaje se zvyšují).
- Klesající trendy (výdaje se snižují).
- Stabilní kategorie.

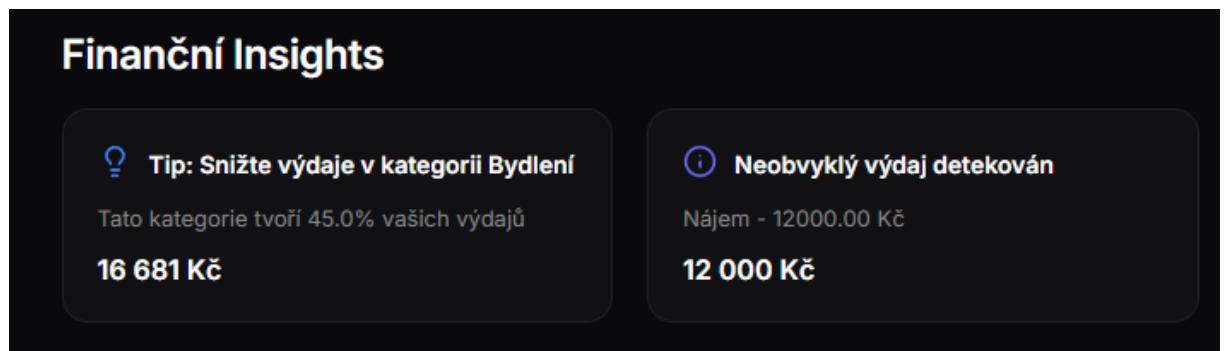


Obrázek 3.5: Analýza trendů výdajů podle kategorií

3.5.3 Insights

Modul Insights analyzuje transakční historii uživatele a generuje personalizované postřehy a rady. Systém automaticky detektuje:

- **Varování** – např. při neobvykle vysokých výdajích v kategorii.
- **Úspěchy** – pochvala za dodržení rozpočtu nebo vysokou míru úspor.
- **Tipy** – doporučení pro efektivnější hospodaření.



Obrázek 3.6: Ukázka personalizovaných finančních postřehů (Insights)

Tato funkce poskytuje uživateli okamžitou zpětnou vazbu a pomáhá mu lépe porozumět jeho finančním návykům.

4 IMPLEMENTACE FRONTENDU

Tato kapitola se věnuje implementaci klientské části aplikace. Frontend je zodpovědný za prezentaci dat uživateli, zpracování uživatelských vstupů a komunikaci s backendem. Díky použití Reactu je aplikace rychlá, responsivní a poskytuje plynulý uživatelský zážitek.

4.1 STRUKTURA REACT APLIKACE

Frontend je organizován do složek: components, contexts, services, styles a utils.

4.2 HLAVNÍ KOMPONENTY

4.2.1 App.tsx

Kořenová komponenta obsahuje providery (ThemeProvider, AuthProvider, ToastProvider) a React Router pro směrování mezi stránkami. Chráněné cesty jsou obaleny komponentou ProtectedRoute, která ověřuje přihlášení uživatele.

4.2.2 AuthContext

Context pro správu autentizace poskytuje:

- Stav přihlášeného uživatele.
- Metody pro přihlášení a odhlášení.
- Automatické obnovení session při refreshi stránky.

4.2.3 Overview

Hlavní dashboard zobrazuje:

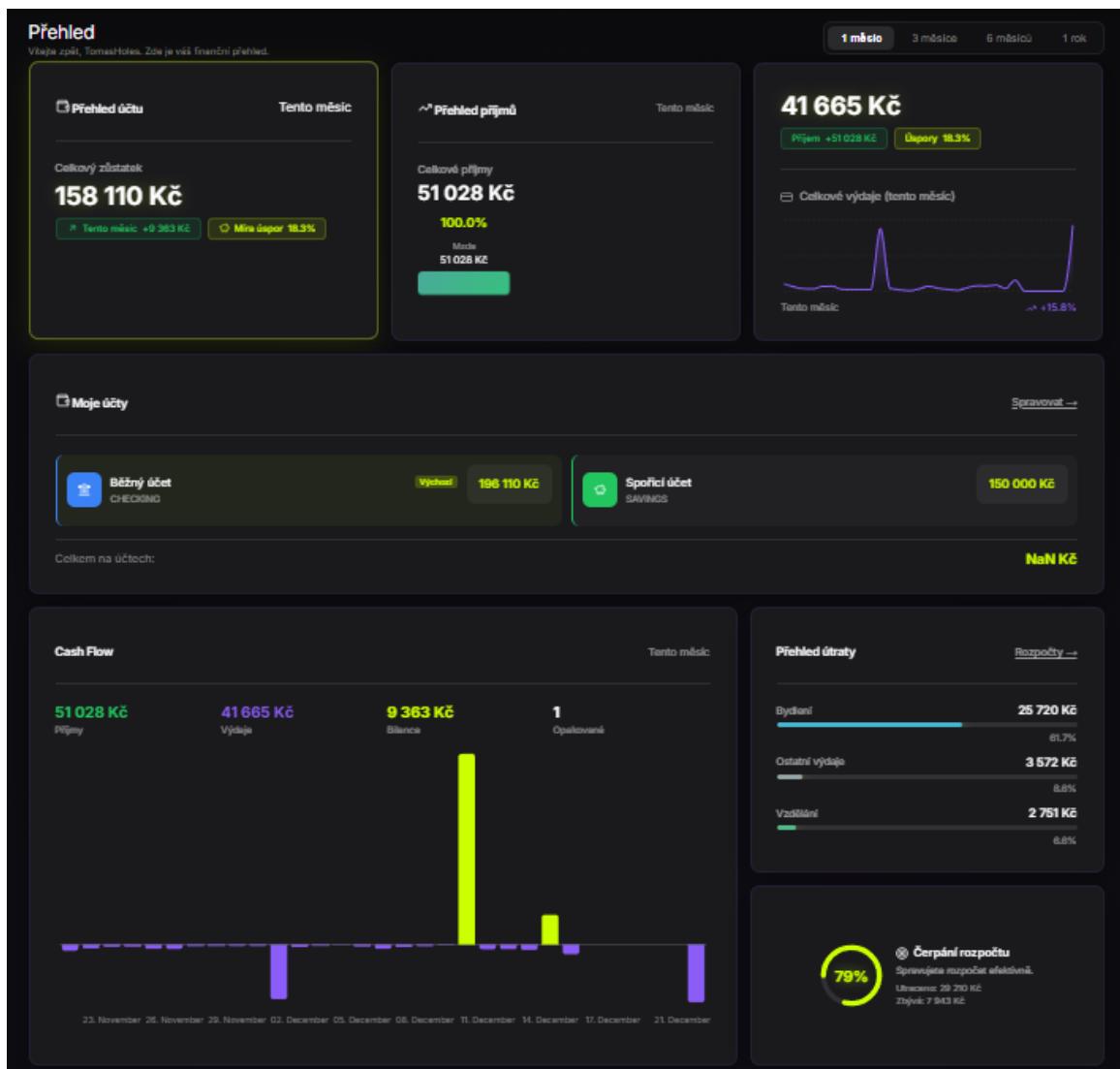
- Celkový zůstatek a měsíční bilanci.

- Přehled finančních účtů.
- Grafy příjmů vs výdajů.
- Rozpočty a jejich plnění.
- Top kategorie výdajů a příjmů.

4.2.4 Další klíčové stránky

Transactions umožňuje prohlížení historie transakcí s možností filtrování, vyhledávání a exportu.

Settings slouží pro správu uživatelského profilu (změna avatara, hesla), nastavení měny a správu vlastních kategorií.

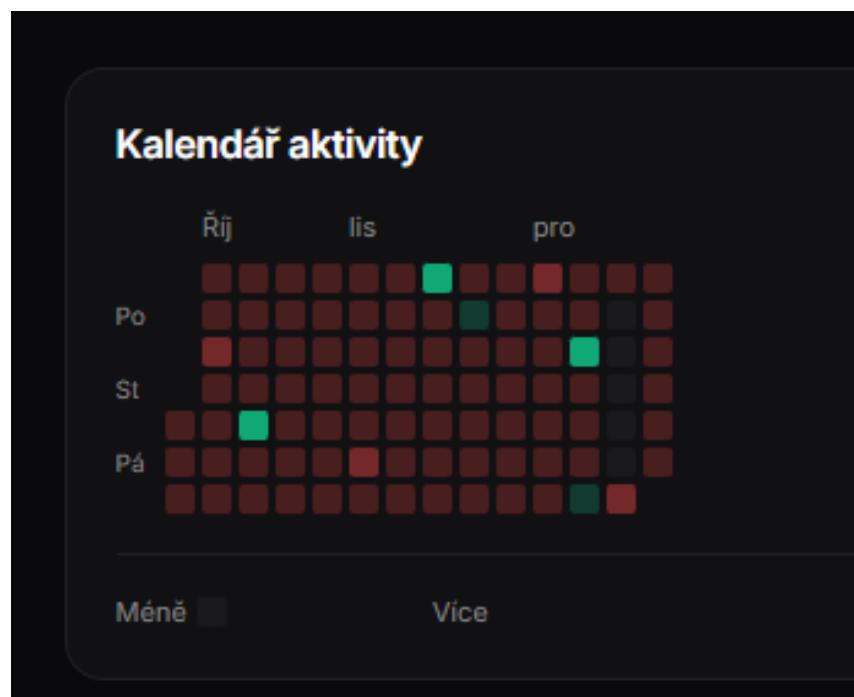


Obrázek 4.1: Hlavní dashboard aplikace Plutoa

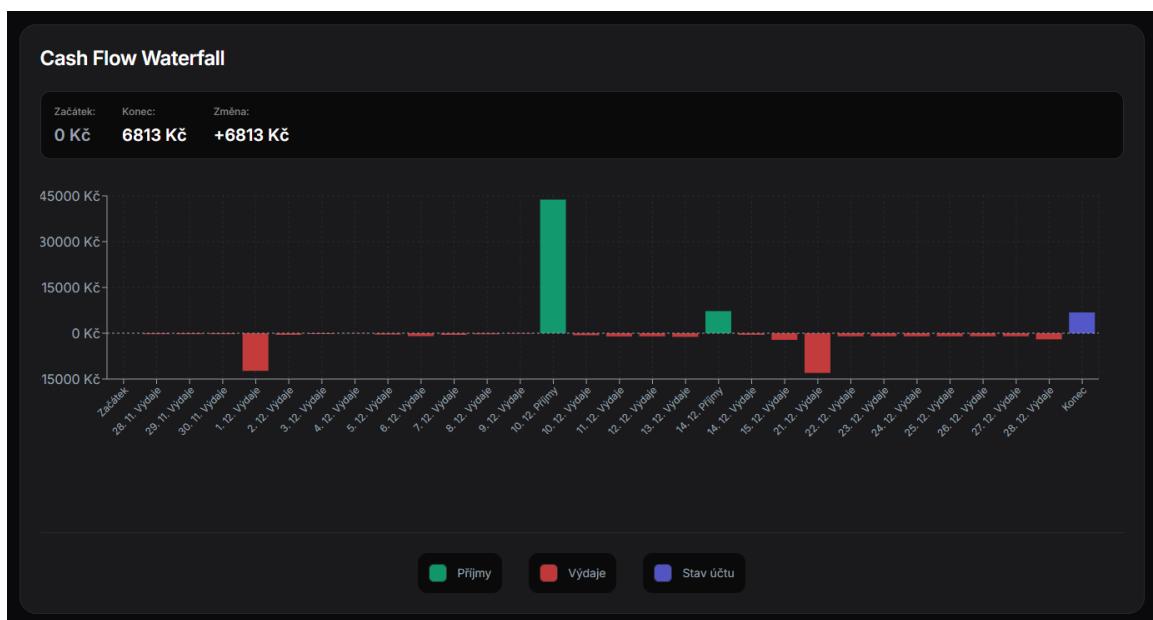
4.3 VIZUALIZACE DAT

Aplikace používá různé typy vizualizací:

- **Grafy příjmů a výdajů** – interaktivní sloupcové grafy s možností změny časového období.
- **Heatmap Calendar** – kalendář zobrazující denní aktivitu, inspirovaný GitHub grafem.
- **Waterfall Chart** – kaskádový graf cash flow ukazující vývoj zůstatku.
- **Category Pie Charts** – koláčové grafy rozložení výdajů a příjmů podle kategorií.



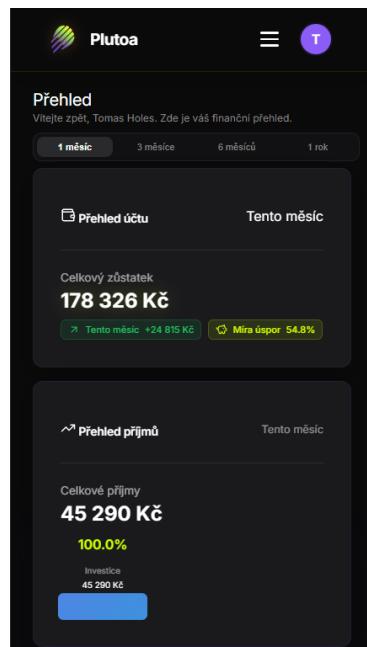
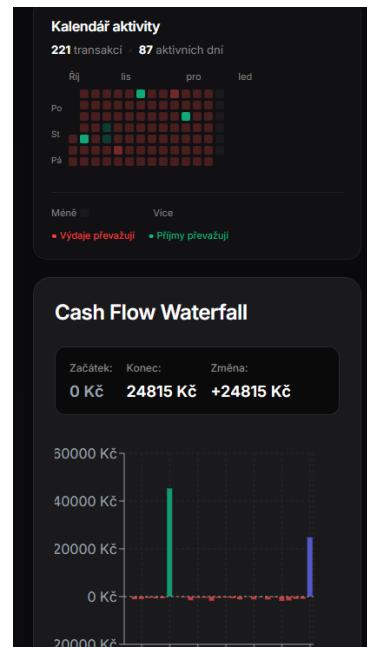
Obrázek 4.2: Heatmap kalendář zobrazující denní finanční aktivitu



Obrázek 4.3: Waterfall graf cash flow

4.4 RESPONSIVNÍ DESIGN

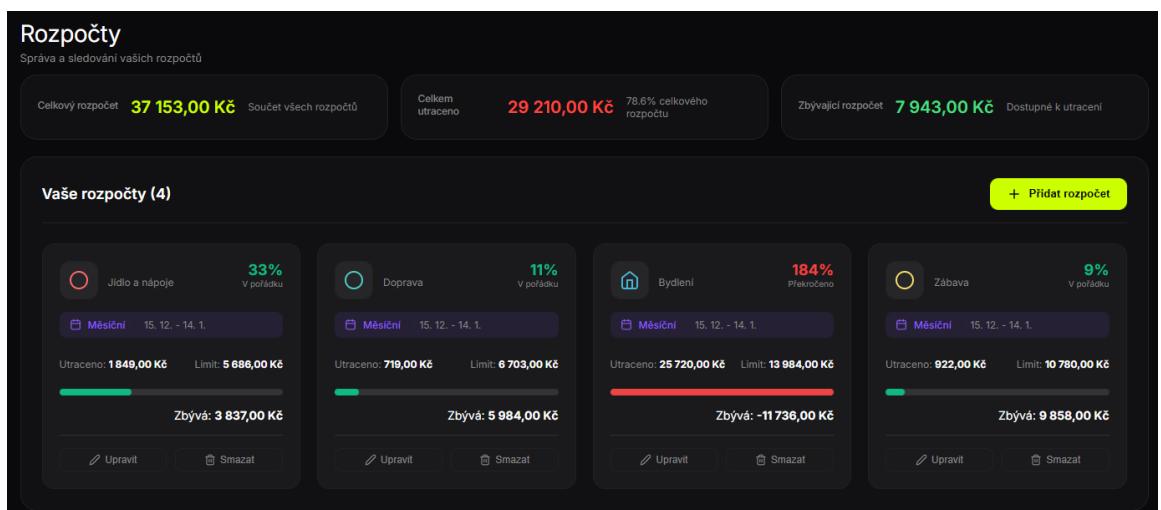
Aplikace je plně responsivní s podporou pro desktop, tablet i mobil. Rozhraní se automaticky přizpůsobuje velikosti obrazovky a poskytuje optimální uživatelský zážitek na všech zařízeních.

Obrázek 4.4: Mobilní verze
- hlavní dashboardObrázek 4.5: Mobilní verze
- analytika

5 POKROČILÉ FUNKCE

5.1 SPRÁVA ROZPOČTŮ

Rozpočty umožňují nastavit limity výdajů pro různá období a kategorie. Systém automaticky sleduje plnění a generuje upozornění při překročení 80%, 90% a 100% limitu.



Obrázek 5.1: Přehled rozpočtů s vizualizací plnění

5.2 FINANČNÍ CÍLE

Modul finančních cílů pomáhá spořit na konkrétní účely. Podporovány jsou různé typy cílů (úspory, splacení dluhu, nákup, nouzový fond, investice). Uživatel může přidávat příspěvky a sledovat pokrok.

5.3 OPAKUJÍCÍ SE TRANSAKCE

Modul opakujících se transakcí automatizuje evidenci pravidelných plateb:

- Různé frekvence (denně, týdně, měsíčně, čtvrtletně, ročně).

- Automatické vytváření transakcí.
- Upozornění před splatností.
- Možnost pozastavení nebo zrušení.

5.4 IMPORT A EXPORT DAT

Aplikace podporuje export dat do formátů **CSV** (vhodné pro tabulkové procesory jako Excel) a **JSON** (pro strojové zpracování). Uživatel si může zvolit časové období exportu. Podporován je také import transakcí z CSV souborů pro snadnou migraci dat.

5.5 SYSTÉM NOTIFIKACÍ

Systém notifikací automaticky upozorňuje na důležité události:

- **BUDGET_EXCEEDED** – překročení rozpočtu.
- **BUDGET_WARNING** – varování o blížícím se limitu.
- **RECURRING_DUE** – blížící se pravidelná platba.
- **GOAL_ACHIEVED** – dosažení finančního cíle.
- **GOAL_PROGRESS** – významný pokrok v cíli.
- **MONTHLY_SUMMARY** – měsíční přehled hospodaření.

ZÁVĚR

Cílem této závěrečné práce bylo vytvořit moderní webovou aplikaci pro správu osobních financí. Aplikace Plutoa nabízí kompletní sadu nástrojů pro sledování příjmů a výdajů, správu rozpočtů, definování finančních cílů a pokročilou analytiku s moderním uživatelským rozhraním.

HLAVNÍ PŘÍNOSY APLIKACE

- **Intuitivní uživatelské rozhraní** – moderní design s důrazem na přehlednost.
- **Pokročilé analytické nástroje** – Financial Health Score, heatmapy, waterfall grafy.
- **Bezpečná autentizace** – JWT tokeny s automatickou rotací.
- **Modulární architektura** – snadné rozšíření o nové funkce.
- **Lokalizace pro ČR** – podpora české měny a jazyka.

MOŽNOSTI DALŠÍHO ROZVOJE

Mezi plánované funkce patří integrace s bankovními API, mobilní aplikace pro iOS a Android, sdílení rozpočtů mezi členy rodiny a AI asistent pro personalizovaná doporučení.

LITERATURA

- [1] Django Software Foundation. *Django documentation* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://docs.djangoproject.com/>
- [2] Encode. *Django REST Framework* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://www.django-rest-framework.org/>
- [3] Meta Platforms. *React documentation* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://react.dev/>
- [4] Microsoft. *TypeScript documentation* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://www.typescriptlang.org/docs/>
- [5] Recharts. *Recharts - A composable charting library* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://recharts.org/>
- [6] Auth0. *JSON Web Tokens Introduction* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://jwt.io/introduction>
- [7] Simple JWT. *Simple JWT documentation* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://django-rest-framework-simplejwt.readthedocs.io/>
- [8] Axios. *Axios HTTP Client* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://axios-http.com/>
- [9] Lucide. *Lucide Icons* [online]. 2024 [cit. 2024-12-15]. Dostupné z: <https://lucide.dev/>

Seznam obrázků

1.1	ER diagram databázového schématu aplikace Plutoa	3
1.2	Diagram architektury aplikace Plutoa	4
3.1	Registrační formulář pro nové uživatele	7
3.2	Přihlašovací formulář pro existující uživatele	7

3.3	Sekvenční diagram JWT autentizace	7
3.4	Detail Financial Health Score s rozpisem jednotlivých komponent	9
3.5	Analýza trendů výdajů podle kategorií	9
3.6	Ukázka personalizovaných finančních postřehů (Insights)	10
4.1	Hlavní dashboard aplikace Plutoa	12
4.2	Heatmap kalendář zobrazující denní finanční aktivity	13
4.3	Waterfall graf cash flow	14
4.4	Mobilní verze - hlavní dashboard	14
4.5	Mobilní verze - analytika	14
5.1	Přehled rozpočtů s vizualizací plnění	15

Seznam tabulek

1.1	Přehled klíčových atributů databázových entit	4
3.1	Přehled API endpointů	8