

RUST_B

FIIT STU

Projekt na tému: „Riadenie vlastných financií“

Tím “Two Bits”:

Nazar Biriukov

Yuliia Kyrylenko

Introduction

Riadenie vlastných financií je desktopová aplikácia vytvorená s použitím Rustu a frameworku Iced GUI. Jej hlavným cieľom je pomôcť používateľom spravovať ich osobné financie jednoduchým a efektívnym spôsobom. Používatelia môžu pridávať transakcie, kategorizovať výdavky a zobrazovať súhrny svojich finančných aktivít. Aplikácia podporuje overovanie používateľov a poskytuje štruktúrovaný spôsob vizualizácie a zaznamenávania finančných údajov v priebehu času.

Requirements

User registration and authentication with secure password storage

Ability to create, edit, and delete transactions

Categorization of transactions using expense tags

Association of transactions with dates, amounts, and comments

Display of transaction history

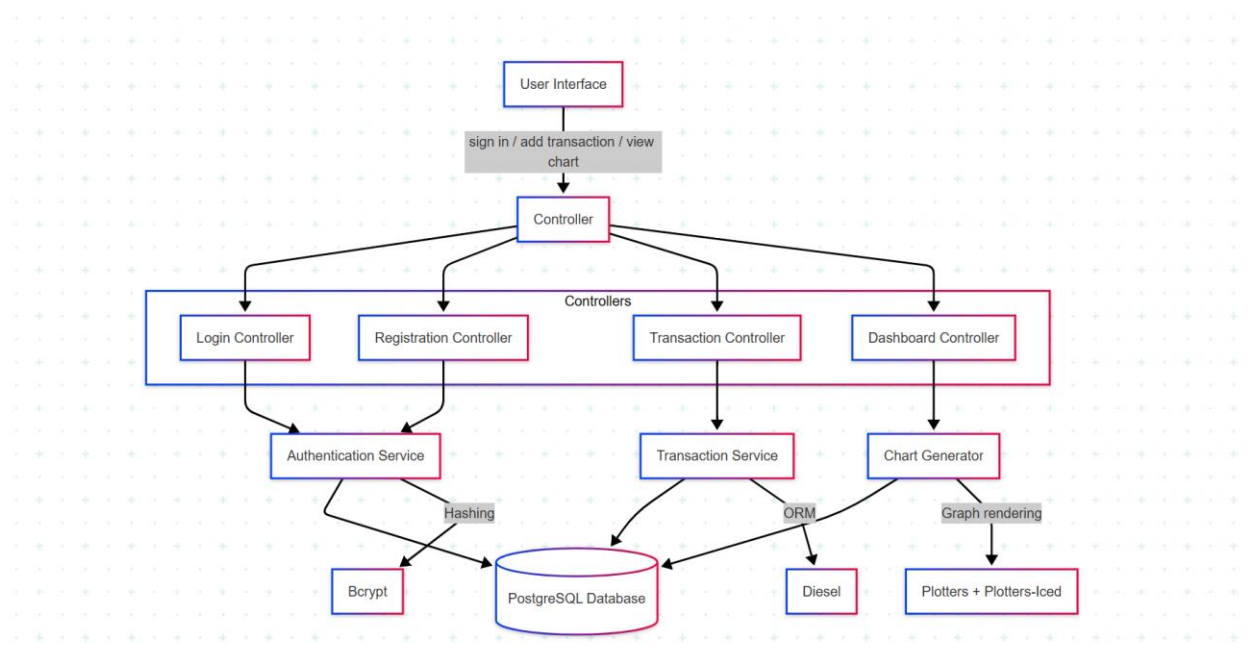
Visualization of financial data

Use of a relational database

Safe handling of user credentials

Clear module separation

Design diagram



Dependencies

Iced - Basic GUI framework na zobrazenie používateľského rozhrania

iced_aw - Rozšírenia Iced, používané pre date_picker a ďalšie widgety

chrono - Spracovanie dátumu a času; podpora serializácie

serde - Serializácia a deserializácia údajov

dotenv - Načítanie premenných prostredia zo súboru .env

bcrypt - Hashovanie používateľských hesiel

diesel - ORM na interakciu s databázou (používa sa s PostgreSQL)

Postgres - Natívny ovládač PostgreSQL

plotters - Generovanie tabuliek a grafov

plotters-iced - Vloženie grafov do rozhrania Iced

Design Choices

Používanie rozhrania Iced:

Iced bol zvolený ako rámec používateľského rozhrania, pretože poskytuje deklaratívny model používateľského rozhrania pripomínajúci Elm/React a dobre zapadá do ekosystému Rust.

Používanie Diesel ako ORM:

Diesel sme si vybrali na interakciu s databázou vďaka jeho statickému typovaniu a overovaniu dotazov počas kompilácie, čo znižuje počet chýb počas behu.

Kód je organizovaný podľa princípu MVC, čo zvyšuje čitateľnosť a zjednodušuje údržbu.

Bezpečné ukladanie hesiel:

Na bezpečné uloženie hesiel používateľov sme použili knižnicu bcrypt. Umožňuje bezpečné hashovanie hesiel

Evaluation

Celkovo bol projekt zaujímavou a obohacujúcou skúsenosťou, najmä v kontexte učenia sa jazyka Rust. Tu je niekoľko pozorovaní na tejto ceste:

Čo sa podarilo:

Práca s databázou prostredníctvom programu Diesel sa ukázala ako spoľahlivá a predvídateľná. Bezpečnosť typov v jazyku Rust výrazne znížila pravdepodobnosť chýb, najmä pri tvorbe dotazov SQL.

Grafické používateľské rozhranie pomocou Iced sa ukázalo ako stabilné a umožňovalo rýchle vytváranie vizuálnych prvkov.

Ťažkosti:

Práca s formátom dátumu

Zobrazenie s možnosťou posúvania

Integrácia grafov prostredníctvom plotrov-iced si vyžadovala dodatočné úsilie a experimentovanie s formátovaním údajov.

Skúsenosti s jazykom Rust v porovnaní s inými jazykmi:

Rust-analyzer veľmi pomáha pri písaní projektov

V porovnaní s jazykom Python je oveľa ťažšie implementovať grafy, ale zároveň je oveľa jednoduchšie zabrániť únikom pamäte.