

1. Цели проекта

Целью проекта “Система учета финансов” является предоставление пользователю удобного инструмента для учета личных финансов.

2. Технологии, которые используются для реализации проекта

- DB: PostgreSQL
- Backend: Java (Spring, Hibernate, SpringBoot)
- Frontend: Vue.js

3. Функциональный объем проекта.

- Пользователь может зарегистрироваться в системе, передав свои данные (логин, пароль, имя), тем самым защитив свои данные
- Пользователь может ввести бюджет (вручную регистрируя свои доходы и расходы)
- Пользователь может создавать категории расходов (передавая имя категории)
- Пользователь может удалить категорию расходов
- Пользователь может установить лимит трат для каждой категории (передавая число в форму ввода)
- Пользователь может удалить лимит трат для выбранной категории
- Пользователь может найти желаемые операции по заданным фильтрам (дата, объем трат/поступлений)
- Пользователь может получить статистику по операциям за заданный период
- Пользователь может изучить свои траты/поступления исходя из наглядных диаграмм (круговая диаграмма распределения бюджета по категориям расходов, круговая диаграмма доходов по категориям поступления, столбцовая диаграмма трат по месяцам, столбцовая диаграмма по разнице трат и поступлений по месяцам)
- Пользователь может вручную добавлять напоминания о предстоящих платежах
- Пользователь имеет возможность выбрать вид напоминаний
- Система должна уметь уведомлять пользователя о предстоящих платежах (из возможных вариантов: при помощи рассылки на почту/бота в telegram)

- Пользователь может получить доступ к предварительно отобранным образовательным материалам по финансовой грамотности

4. Нефункциональные требования проекта

- Безопасность (в приложении используется Spring Security with JWT, что гарантирует безопасность данных пользователя)
- Интуитивно понятный пользовательский интерфейс
- Производительность (сбор статистических данных, формирование отчетов с помощью JasperTemplates) операции(статистика) выполняются в асинхронном режиме, не блокируя основной поток исполнения)
- Доступность (система доступна с любого устройства с доступом в интернет и совместимым браузером (с установленным обновлением не позднее 2012 г.), благодаря размещению на веб ресурсах)
- Совместимость (система совместима с другими системами благодаря API) (к примеру, система автоматического учета финансовых операций)

5. Архитектура проекта

- БД - PostgreSQL
- Микросервис - Статистика (для подсчета сложной статистики, применимо для будущей реализации)
- Микросервис - Core
- Микросервис - Frontend

- Диаграмма архитектуры

