Яндекс.Лицей

**Проект WebServer + API**

**Веб-приложение для контроля финансов “WebFinUp”**

**Работу выполнил:**

Ученик 11T класса

ГБОУ Образовательный центр “Протон”

Семенов Александр Алексеевич

**Научный руководитель:**

Федоров Кирилл Евгеньевич

Москва 2022

**Проблема**

1. В наше время трудно контролировать свои финансы, которые у большинства людей находятся на разных счетах в разных банках.
2. По статистике финансовая подушка безопасности есть у каждого третьего Россиянина, но большинство просто хранит деньги, не преумножая их. Многим людям трудно следить за своими расходами.

**Цель**

Разработать удобное веб-приложение для контроля финансов.

**Задачи (Ход работы)**

1. Изучить созданные финансовые веб-приложения и выявить их плюсы и минусы;
2. Разработать функции для работы с базой данных SQLite;
3. Разработать интерфейс веб-приложения с использованием шаблонов Bootstrap;
4. Привязать html-шаблоны к веб-страницам, создаваемым в микрофреймворке Flask, организовать вывод данных из sql-функций на веб-страницы и получение с них других данных от пользователя обратно в базу данных.

**Aрхитектура веб-приложения**

Ссылка на облачную платформу контроля версий с исходным кодом программы: <https://github.com/20alex20/WebFinUp>

Python-файлы расположены в корневой папке проекта и включают в себя четыре файла, общим объёмом более тысячи строк кода:

1. sql.py (456 строк кода)

Данный файл предназначен для работы с базой данных. Через функции get\_data и do\_query производятся запросы к базе данных db.sqlite3. Помимо двух основных функций реализовано 43 вспомогательных функций для выполнения конкретных запросов к базе данных.

Перечень вспомогательных функций:

register, login, edit\_about\_me, logout, delete\_my\_account, add\_category, get\_categories, edit\_category, delete\_category, add\_deposit\_category, get\_deposit\_categories, edit\_deposit\_category, delete\_deposit\_category, add\_purchase, get\_purchase, edit\_purchase, delete\_purchase, add\_bank\_account, get\_bank\_accounts, edit\_bank\_account, delete\_bank\_account, add\_deposit, get\_deposits, edit\_deposit, delete\_deposit, get\_purchase\_deposit, get\_sum, get\_sum\_deposits, get\_sum\_purchases, get\_category\_name, get\_deposit\_category\_name, get\_bank\_acc\_name, get\_all

1. csv\_xlsx.py (38 строк кода)

Данный файл предназначен для экспорта данных в xlsx (функция export\_xlsx) и csv (собранных в zip) (функция export\_csv).

1. charts.py (120 строк кода)

Данный файл предназначен для формирования наглядных графиков. Основной функцией является функция "charts", которая строит графики четырёх разных типов (круговая диаграмма, столбчатая диаграмма, гистограмма с накоплением, график с маркерами). За тип графика отвечает параметр "mode", параметр "type" - флаг, обозначающий построение графиков расходов (значение True) или графика доходов (значение False). Параметр "time\_mode" отвечает за выбор построения по годам/месяцам/дням (значения 1, 2, 3 соответственно). Параметр "args" служит для передачи даты старта и даты конца периода, за который требуется построить график.

1. index.py (424 строк кода)

Это основной файл программы, который отвечает за вывод информации на экран, получение данных от пользователя и их обработку.

Функция login\_web отвечает за страницу входа (/login), в которую встроена форма для ввода логина и пароля, за которую отвечает класс Login. Функция register\_web отвечает за страницу регистрации (/register), в которую встроена форма для ввода логина, ФИО и пароля, за которую отвечает класс Register. После входа или регистрации данные о нем сохраняются в cookie-файлы. Функция logout\_web отвечает за выход пользователя из системы, т. е. за очистку файлов cookie. Функция index отвечает за главную страницу (/index), на которой отображается сумма на всех счетах, сумма доходов за месяц и сумма расходов за месяц, а также история операций по убыванию даты (но можно сделать сортировку и по другим критериям). Функция bank\_accounts отвечает за страницу со списком всех счетов пользователя (/bank\_accounts), а функция categories – за страницу со списком всех категорий доходов и расходов (/bank\_accounts). Функция analytics отвечает за страницу с построением четырёх видов графиков (/analytics). Функции income и expense отвечают за страницы добавления доходов/расходов через формы (/income, /expense), за которые отвечают классы Income, Expense. Функции income\_category и expense\_category отвечают за страницы добавления и изменения категорий доходов/расходов через формы (/income\_category, /expense\_category), за которые отвечают классы IncomeCategory, ExpenseCategory.

**Выводы**

Разработано веб-приложение с обширным функционалом, которое поможет людям контролировать свои финансовые потоки и эффективно ими управлять. Оно позволяет смотреть общий баланс, контролировать расходы, подсчитывать доходы, анализировать и делать выводы о своей стратегии управления личными финансами.

**Список источников**

1. Финансы

<https://play.google.com/store/apps/details?id=ru.innim.my_finance&hl=ru&gl=US>

2. Журнал расходов

<https://play.google.com/store/apps/details?id=com.vitvov.profit&hl=ru&gl=US>

3. Учет расходов и доходов. Семейный бюджет

<https://play.google.com/store/apps/details?id=ru.artem_rumyantsev.financialarchitect&hl=ru&gl=US>

4. Moneon — мой бюджет доходов и расходов, финансы

<https://play.google.com/store/apps/details?id=hk.moneon&hl=ru&gl=US>

5. Дзен-мани: учет расходов и доходов

<https://play.google.com/store/apps/details?id=ru.zenmoney.androidsub&hl=ru&gl=US>

6. Библиотека Matplotlib для отображения графиков

<https://matplotlib.org/>

7. Фреймворк Flask для создания веб-приложений на языке программирования Python

<https://flask.palletsprojects.com/en/2.1.x/>