**Пояснительная записка к проекту “Трекер Финансов”**

**Автор: Евгения Аршинова**

***Идея проекта:*** приложение для учёта финансов, с созданием визуализации внесённых счетов и предоставлением их краткого анализа нейросетью.

***Структура проекта:***

* **data –** директория с моделями SQLAlchemy;
* **db** – директория с базой данных пользователей и их счетов;
* **forms** – директория с Flask формами;
* **static:**
  + - **css** – стили;
    - **img** – картинки;
* **templates** – html файлы страниц;
* **ai.py** – генерация оценки расходов с помощью api DeepSeek;
* **config.py** – файл с константами;
* **main.py** – файл запускающий веб-приложение;
* **requirements.txt** – библиотеки проекта;
* **utils.py** – файл с абстрактными функциями.

***Использованные технологии и библиотеки:***

Flask, SQLAlchemy, asyncio, aiohttp, pyChart.JS, requests.

***Основные страницы:***

**/** – стартовая страница, которая перебрасывает пользователя на домашнюю страницу, если пользователь авторизован, в противном случае – требует от пользователя авторизации.

**/login** – страница авторизации.

**/register** – страница регистрации.

**/home** – домашняя страница, где в таблице показаны все финансы пользователя, отсортированные по возрастанию даты. На неё можно попасть, нажав на кнопку **Home** в навигационной панели сверху.

**/add\_account** – страница, на которой пользователь вносит свои доходы или расходы.

**/accounts/<id>** – страница, на которой пользователь редактирует свои внесённые данные.

**/account\_delete/<id>** – страница для удаления данных пользователя.

**/period** – страница, на которой пользователь выбирает период времени для создания визуализации своих данных

**/dashboard/<start\_date>/<end\_date>** – страница, на которой показана визуализация финансовых показателей пользователя, а также их анализ нейросетью.

**/get\_ai\_summary** – конечная страница для получения результата запроса к api нейросети.

**/profile** – страница профиля пользователя. На неё можно попасть, нажав на имя пользователя в навигационной панели сверху

**/edit\_profile** – страница редактирования профиля.

**/logout** – страница выхода из профиля.

***Основные решения:***

Проект реализован при помощи фреймворка Flask. Вся информация (пользователи и их финансы) записывается в базу данных **database.sqlite** по ORM технологии. В веб-приложении используется api DeepSeek, урезанная модель которого установлена локально на компьютере.

Для того, чтобы начать пользоваться сайтом “Трекер Финансов” пользователю сначала нужно зарегистрироваться (за это отвечает функция **register**) или авторизоваться (функция **login**).

После авторизации пользователь попадает на страницу **/home**, где ему представлены все его записи в таблице, отсортированные по возрастанию. Пользователь может добавлять свои финансы и указывать их параметры: тип (доходы или расходы), категорию, дату (веб-приложение не даёт пользователю добавлять данные на даты в будущем) и сумму. Добавление записей осуществляется с помощью Flask формы **AddAccountForm** и функции **add\_account**. Изменения в финансы вносятся с помощью той же Flask формы (**AddAccountForm**) и функции **edit\_account**. Записи удаляются функцией **account\_delete**. Можно также удалить все записи сразу, нажав на кнопку **Remove all data**.

При нажатии на **Dashboard** на навигационной панели сверху, открывается страница **/period**. После выбора периода времени на данной странице, происходит переадресация на **/dashboard**, где по финансам пользователя создаётся их визуализация (соотношение доходов и расходов, сортировка сумм доходов/расходов по категориям и т. д). Помимо графиков пользователю также представляется краткий анализ его финансов нейросетью DeepSeek (за это отвечает асинхронная функция **ai\_review\_async**). Функция специально сделана асинхронной, чтобы пользователю не нужно было ждать ответа DeepSeek перед просмотром визуализаций своих доходов и расходов. Таким образом, пока ответ от api ещё не получен, пользователь видит текст об ожидании результата анализа и все построенные графики и диаграммы. После получения ответа текст ожидания сразу же сменяется на текст-анализ DeepSeek. Визуализации строятся с помощью библиотеки **pyChart.JS**, которая является обёрткой библиотеки на JS (Chart.js).

На странице **/profile** отображается профиль пользователя, а именно его имя, фамилия, возраст, почта и аватар. Если при регистрации пользователь не установил себе аватар, то ему присваивается стандартный. Изменение профиля происходит на странице **/edit\_profile**, где можно изменить все данные пользователя, включая пароль. Выход из профиля осуществляется нажатием кнопки **Log out**.

***Скриншоты программы***

















