Государственное бюджетное общеобразовательное учреждение города Москвы «Школа№1542»

**ИНДИВИДУАЛЬНЫЙ ПРОЕКТ**

**Проект «Веб-приложение «Трекер привычек» на Python»**

**Выполнил: Акамсин Артемий Романович**

**Ученик 10 «Я» класса**

**Направление «Программирование»**

**Руководитель: Ушинская Ольга Борисовна­­­**

Москва, 2024/25 учебный год

**ВВЕДЕНИЕ**

Актуальность проекта.

Современные технологии играют важную роль в управлении личной эффективностью и организацией времени. Проблема формирования и поддержания полезных привычек актуальна для многих людей, так как привычки напрямую влияют на качество жизни, продуктивность и достижение целей. Однако, несмотря на наличие различных приложений и инструментов, многие из них сложны в использовании, перегружены функционалом или требуют подписки. Создание собственного веб-приложения для отслеживания привычек на Python с использованием Flask позволит предложить удобный и интуитивно понятный интерфейс для пользователя.

Проблема, которую я перед собой ставил.

Создать приложение с регистрацией и входом по логину. Сделать так, чтобы пользователь мог легко добавлять привычки и наблюдать за их прогрессом.Благодаря таблице, пользователи видели бы свой прогресс и процент выполненных задач. Приложение поможет фиксировать выполнение действий, что минимизирует вероятность забыть о поставленных целях.

Оригинальность проекта.

В отличие от многих готовых решений, проект полностью управляем пользователем — можно добавить абсолютно любые привычки. Использование таблиц для анализа выполнения привычек выделяет приложение среди аналогов. Возможность самостоятельного запуска приложения на сервере или локально делает его привлекательным для пользователей, которые хотят избежать подписок и рекламы.

Таким образом, проект сочетает в себе практическую пользу, образовательную ценность (в области Python и Flask) и потенциал для дальнейшего развития.

Цели проекта: Создать функциональное и удобное веб-приложение "Трекер привычек" на Python с использованием Flask, которое поможет пользователям формировать, отслеживать и анализировать свои привычки, мотивируя их на достижение целей и повышение личной эффективности.

Задачи проекта:

1. Выбрать вид сайта
2. Познакомиться с фреймворком Flask и другими
3. Познакомиться с языком html
4. Познакомиться с созданием баз данных и Sqlite3
5. Создать структуру бд
6. Создать разметку для сайта
7. Создать серверную часть сайта

**ГЛАВА 1**

В современном обществе управление временем и самоорганизация становятся ключевыми факторами успеха. Формирование полезных привычек способствует улучшению здоровья, повышению продуктивности и достижению долгосрочных целей. Однако многие люди сталкиваются с трудностями в их внедрении и поддержании. В этой связи цифровые инструменты, такие как трекеры привычек, оказываются крайне полезными для пользователей, стремящихся к улучшению своей жизни.

Существующие приложения, например Habitica, Loop и другие, предоставляют функционал для отслеживания привычек, однако многие из них имеют ограничения, такие как недостаточная гибкость настройки, платные функции или сложный интерфейс. Это создает необходимость разработки более доступных и персонализированных решений.

Первое с чего нужно начать делать сайт – определиться с его видом. Вот виды сайтов с которыми я ознакомился:

**Сайт-визитка** - представляет собой небольшой сайт, который содержит базовую информацию о компании, услуги и контакты. Обычно имеет от одной до пяти страниц и служит для начального представления в интернете.

**Корпоративный сайт** - создан для представления компании, ее деятельности, проектов и ценностей. На таких сайтах часто размещают новости, информацию о команде, вакансии и формы обратной связи.

**Интернет-магазин** - специализированная платформа для продажи товаров и услуг. Такие сайты включают каталог, корзину для покупок, систему оплаты и доставки.

**Информационный портал** - сайт, содержащий большое количество новостей, статей или тематических материалов. Обычно предназначен для широкой аудитории и может охватывать разные темы или быть узкоспециализированным.

**Блог** - личный или корпоративный сайт, где публикуются статьи, заметки или новости. Основной упор делается на тексты, фотографии и видео, часто используется для личного брендинга.

**Лендинг (одностраничник)** - одностраничный сайт, обычно созданный для продвижения конкретного продукта, услуги или акции. Отличается ярким дизайном и высокой концентрацией информации.

**Социальная сеть - п**латформа, предназначенная для общения пользователей, обмена контентом и создания сообществ. Примеры: Facebook, Instagram.

**Образовательный сайт** - содержит курсы, уроки, лекции или тесты. Основное предназначение — обучение и развитие навыков пользователей.

**Форум** - сайт для общения на тематические темы, где пользователи могут создавать обсуждения, задавать вопросы и делиться опытом.

**Развлекательный сайт -** платформа, ориентированная на развлечения: игры, видео, мемы, музыка или другой контент для отдыха и досуга

Веб приложение - интерактивный сайт с функциональностью, направленной на выполнение конкретной задачи. В отличие от простых информационных сайтов, веб-приложения позволяют пользователям взаимодействовать с системой: вводить данные, анализировать прогресс, получать визуализацию в виде графиков и таблиц.

Ресурсы - <https://altblog.ru/vidy_sajtov/> ; <https://timeweb.com/ru/community/articles/kakie-tipy-sajtov-byvayut>

Для реализации моего проекта лучше всего подошел последний вариант – веб-приложение. Для его создания я изучил работу с фреймворком Flask и flask\_login, sqlalchemy, werkzeug, flask\_wtf, wtforms. Всего я использовал сотни разных десятки разных функций этих библиотек, однако среди них можно выделить основные:

Flask() – для создания экземпляра

app.secret\_key = “” – создание секретного ключа для защиты сессий и данных

@app.route("/") – маршрут на главную страницу сайта

request.form – обработка данных из форм

render\_template(“”) – отражение страницы сайта

redirect – перенаправка на другую страницу

send\_file – скачать файл

class Name(SqlAlchemyBase): - модель базы данных

db\_sess = db\_session.create\_session() – создание сессии

generate\_password\_hash(password) – хеширование пароля

check\_password\_hash(password) – проверка хеша и пароля

class Name(flask\_wtf.FlaskForm) – форма

И многие другие функции.. Информацию я брал из курсов Яндекс Лицея: Основы программирования на python, и Основы промышленного программирования на python.

Страницы сайтов — это файлы с текстом, размеченным на языке HTML. Эти файлы, будучи загруженными посетителем на его компьютер, обрабатываются браузером и выводятся на его средство отображения (монитор, экран КПК, принтер или синтезатор речи). Затем я изучил как работать с html, css и javasript (для разметки сайта и придания красочности).

Главная информация по этим языкам:

### ****HTML (основы разметки веб-страницы):****

1. **Теги структурирования:**
   * <html>, <head>, <body> — основа страницы.
   * <div>, <span> — контейнеры для элементов.
   * <header>, <footer>, <section>, <article> — семантические теги для структуры.
2. **Контентные теги:**
   * <h1>–<h6> — заголовки.
   * <p> — параграфы.
   * <a href="..."> — ссылки.
   * <img src="..."> — изображения.
   * <ul>, <ol>, <li> — списки (нумерованные и маркерные).
   * <table> и <tr>, <td>, <th> — таблицы.
3. **Формы:**
   * <form>, <input>, <button>, <textarea> — создание интерактивных форм.
   * <select>, <option> — выпадающие списки.
   * Атрибуты форм: placeholder, required, type.
4. **Мета-данные:**
   * <meta charset="UTF-8"> — кодировка.
   * <title> — название страницы.
   * <link rel="stylesheet" href="..."> — подключение CSS.

**CSS (стилизация):**

1. **Основы стилей:**
   * Селекторы: element, .class, #id, комбинированные селекторы.
   * Свойства: color, background-color, font-size, margin, padding, border.
2. **Расположение элементов:**
   * display: block, inline-block, flex, grid.
   * position: static, relative, absolute, fixed.
   * Выравнивание: text-align, align-items, justify-content.
3. **Работа с цветами и шрифтами:**
   * Цвета: HEX, RGB, HSL.
   * Шрифты: font-family, font-weight, line-height.
4. **Анимации и переходы:**
   * transition, animation, @keyframes.
5. **Работа с медиазапросами:**
   * Адаптивность: @media (max-width: 768px).

### ****JavaScript (основы интерактивности):****

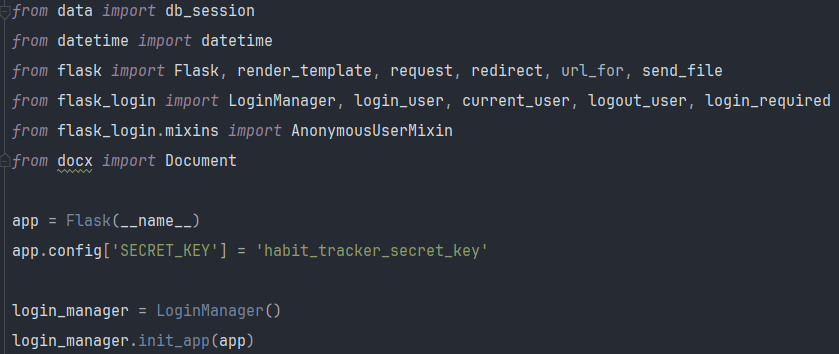
1. **Переменные и типы данных:**
   * let, const, var.
   * Типы данных: строки, числа, массивы, объекты.
2. **Работа с DOM:**
   * Поиск элементов: document.querySelector(), getElementById().
3. **Функции и методы:**

* Создание функций: function() {}, стрелочные функции () => {}.
* Работа с массивами: .map(), .filter(), .forEach().

Ресурсы: https://htmlbook.ru/ ; <https://learn.javascript.ru/> ; <https://html5book.ru/osnovy-css/>

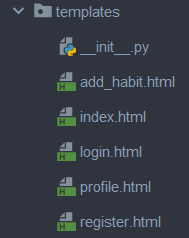
**ГЛАВА 2**

Итак, после изучения всех библиотек и языков, я определился с ide – Pycharm Community Edition 2024. Я выбрал Pycharm, так как им легко пользоваться и именно на нём пишется множество проектов на python. Первый файл, который я создал – main.py в корневой папке проекта. В этом файле будет почти вся серверная часть проекта.



*Иллюстрация 1. Импорты и основные элементы*

Изначально я сделал экземпляр моего проекта (app), секретный ключ (для защиты данных) и login\_manager (для логики профилей).



*Иллюстрация 2. Папка templates*

Вот за что отвечает каждый файл:

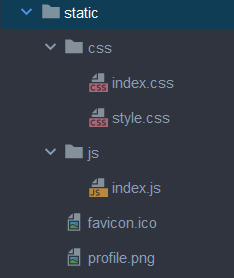
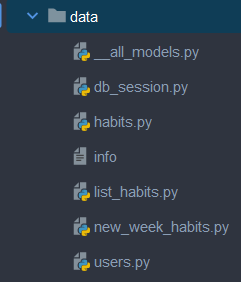
1 add\_habit.html – добавление новой привычки

2 index.html – основная страница сайта

3 login.html – вход по логину

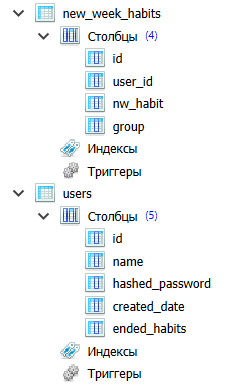
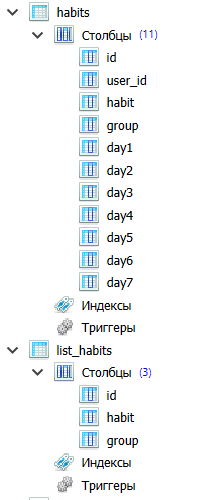
4 register.html – регистрация

5 profile.html – профиль



*Иллюстрация 3. Папки data и static.*

Далее я создал папку data. В ней находятся файлы моделей базы данных, инициализации базы данных и информация о привычках (добавляется ли завершенная в новую неделю). Затем я сделал папку static – со статичными файлами. favicon.ico – логотип сайта, я нарисовал с помощью Paint, profile.png – картинка для кнопки профиля. Саму базу данных я положил в отдельную папку db/habit\_tracker.db Я сделал её с помощью SQliteStudio.



*Иллюстрация 4. База данных*

В ней находятся 4 таблицы:

habits – привычки пользователя

list\_habits – все привычки из выпадающего списка

new\_week\_habits – привычки, которые появятся на следующей неделе

users – пользователи

Функции основного файла (main.py)

1. load\_user(user\_id) – получение user\_id пользователя из базы данных.

2. get\_user\_habits(user\_id) – получение привычек пользователя.

3. index(group=’Все привычки’) – загрузка основной страницы

4. add\_new\_habit() – переход на страницу с добавлением привычки

5. habit\_add() – добавление привычки

6. download\_table() – скачивание таблицы

7. end\_habit() – завершить привычку

8. delete\_habit() – удалить привычку

9. rename\_habit() – переименовать привычку

10. login() – вход

11. register() - регистрация

12. logout() – выход из профиля

13. is\_adding\_to\_new\_week() – добавление привычки в новую неделю

14. create\_new\_week() –новая неделя

15. main() – запуск сайта

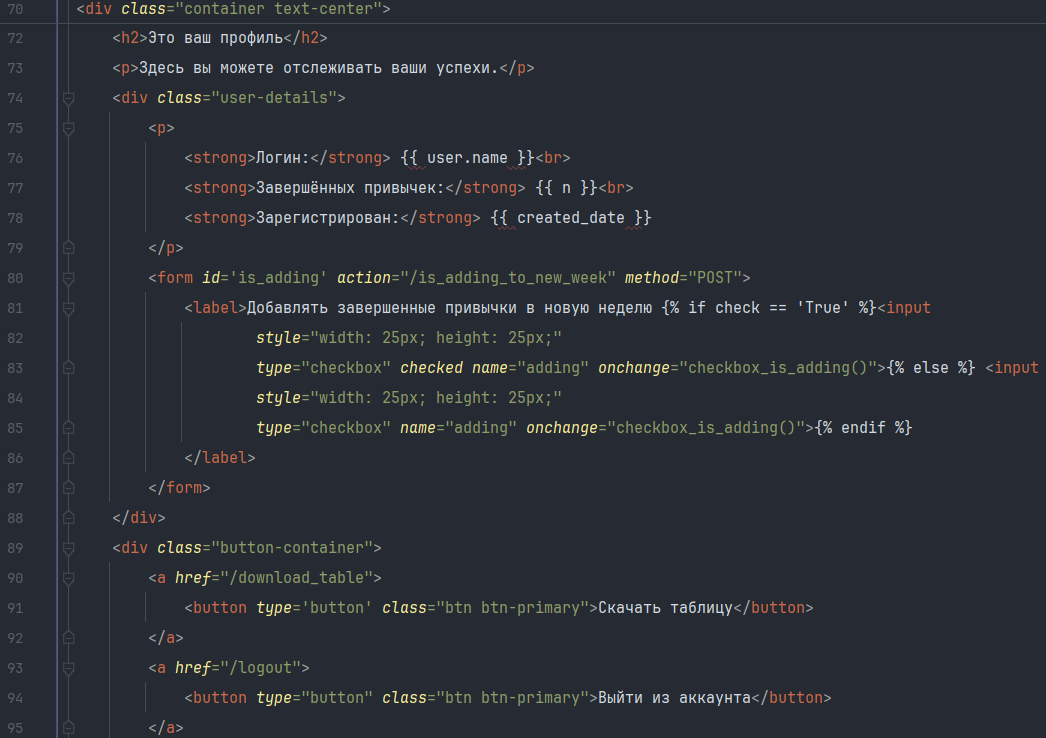
**Страница профиля:**

Логин, количество завершённых привычек, дата регистрации – берется из users – name, ended\_habits, created\_date

Галочка для добавления завершённых привычек в новую неделю. Я сделал это с помощью checkbox, его положение (true/false) хранится в файле data/info

Кнопка “Скачать таблицу” – я реализовал ее с помощью Document() из docx и send\_file()

Кнопка “Выйти из аккаунта” – тригерит функцию logout() и выходит из профиля



*Иллюстрация 5. profile.html*

**Главная страница.**

Изначально я хотел делать фиксирование результатов в другой форме, но пришел к выводу, что организация таблицы будет гораздо проще и понятней пользователю. Исходя из этой мысли я сделал такую таблицу: в шапке находится группа привычек (нажимая на неё выпадает список групп), 7 колонок с днями недели, процент завершённости привычки (количество дней, в которых она завершена / 7 \* 100). В первой колонке находятся те привычки, которые принадлежат выбранной группе. В остальных клетках по умолчанию пусто, а при нажатии появляется галочка (сделал с помощью checkbox) и изменяется последняя колонка (процент завершённости).

Под таблицей есть кнопка добавления привычки и начала новой недели.

Как работает кнопка “Следующая неделя”.

Если в профиле галочка “Добавлять завершенные привычки в следующую неделю” включена, то таблица очищается и в неё добавляются все привычки, которые были к начале последней. Иначе, в неё добавляются только те привычки, которые остались под конец последней.

Кнопка добавления привычки перенаправляет на страницу сайта /add\_habit. На ней нужно выбрать группу из списка, к которой будет принадлежать привычка. Затем, либо выбрать привычку из списка, либо написать свою. Весь список основных привычек, из которых может выбрать пользователь находится в бд в таблице list\_habits.

При нажатии на название привычки появляется окошко с возможностью изменить или удалить эту привычку.

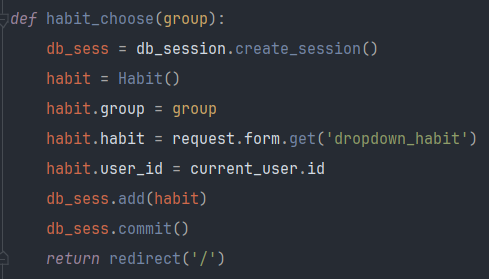
**Страница добавления привычки.**

Она состоит из трёх компонентов:

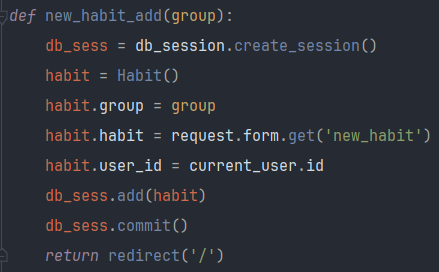


*Иллюстрация 6. Выбор группы*

В выбранную группу добавится новая привычка.

****

*Иллюстрация 7. Добавление привычки из выпадающего списка*



*Иллюстрация 8. Добавление своей привычки*

**ГЛАВА 3**

**Инструкция по пользованию сайтом.**

1. При первом входе сразу попросят зарегистрироваться (если уже есть профиль – кнопка войти)
2. После регистрации переход на главную страницу.
3. Чтобы добавить привычку нужно:
   * Выбрать группу для неё из выпадающего списка
   * Выбрать привычку из выпадающего списка (соответствует выбранной группе) или написать название своей привычки.
4. Чтобы пометить выполненную привычку нужно нажать на ячейку сегодняшнего дня – в последней колонке изменится процент завершения
5. Чтобы завершить привычку (выполнена 7 дней подряд) нужно нажать на появившуюся светящуюся кнопку.
6. Чтобы изменить или удалить привычку нужно нажать на неё, а затем на кнопку действия
7. Чтобы начать новую неделю – кнопка “Начать новую неделю” (при выключенной галочке в профиле – в новой неделе будут только незавершённые привычки, при включённой – все привычки этой недели)
8. Чтобы попасть на страницу профиля – кнопка в правом верхнем углу.
9. Галочка добавления завершённых привычек в новую неделю.
10. Кнопка скачать таблицу – скачивает в формате docx таблицу с привычками.
11. Кнопка выйти из аккаунта – выходит из аккаунта.

**Тестирование продукта.**

**Перегрузка сервера и базы данных.**

Я добавил очень много привычек и на главной странице начал бездумно ставить и убирать галочки. После сотни поставленных галочек и очень редким задержкам сайта я понял – мой сайт отлично справляется с перегрузкой и прошёл первый тест.

**Совместимость.**

Я проверил совместимость с тремя устройствами – windows 10 pro, windows 8, macbook pro 12. Дизайн ни на одном из устройств не слетел, а значит сайт прошел тест.

**ЗАКЛЮЧЕНИЕ**

В ходе реализации проекта был создан функциональный и удобный веб-сервис для отслеживания привычек. Основной целью было разработать приложение, которое позволит пользователям регистрироваться, добавлять свои привычки и следить за прогрессом выполнения. Эти цели были успешно достигнуты благодаря созданию интуитивно понятного интерфейса, таблицы для отображения прогресса и расчёта процента выполнения задач.

Проект выполнил заявленные задачи:

1. Определён тип веб-сайта, подходящий для поставленных целей.
2. Изучены фреймворк Flask, язык HTML, основы создания баз данных и работа с SQLite.
3. Разработана структура базы данных для хранения информации о пользователях и привычках.
4. Созданы серверная и клиентская части сайта, а также пользовательский интерфейс.

Созданное приложение решает проблему отсутствия систематизации в формировании привычек, помогая пользователям фиксировать их выполнение. Это минимизирует вероятность забыть о поставленных целях и способствует личной эффективности.

Продукт имеет потенциал для дальнейшего развития: можно добавить визуализацию статистики, возможность делиться достижениями с другими пользователями и интеграцию с другими платформами. Первоначальные отзывы пользователей указывают на удобство и полезность системы, что подтверждает успешность реализации проекта.

**ИСТОЧНИКИ ИНФОРМАЦИИ**

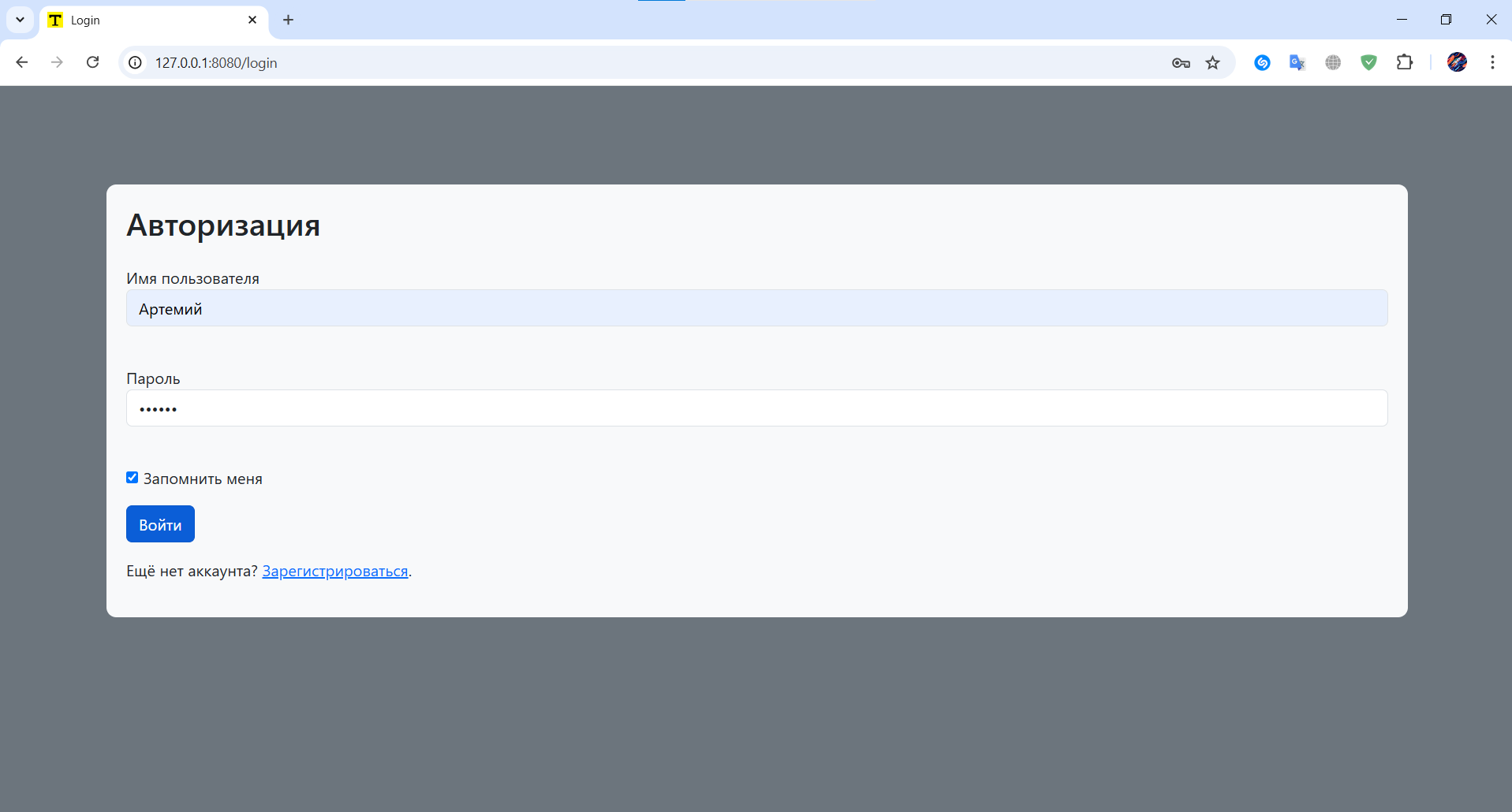
**Электронные ресурсы**

1. Влад Мержевич, Справочник по html: https://htmlbook.ru/html
2. Илья кантор, Современный учебник JavaScript: <https://learn.javascript.ru/>
3. Елена Назарова, Основы CSS: <https://html5book.ru/osnovy-css/>
4. Кирилл Рамирас, Какие типы сайтов бывают – полная классификация с примерами: <https://altblog.ru/vidy_sajtov/>
5. Добро пожаловать в Flask: <https://runebook.dev/ru/docs/flask/>
6. Александр, Мега-Учебник Flask Глава 4: База данных (издание 2024): https://habr.com/ru/articles/807371/

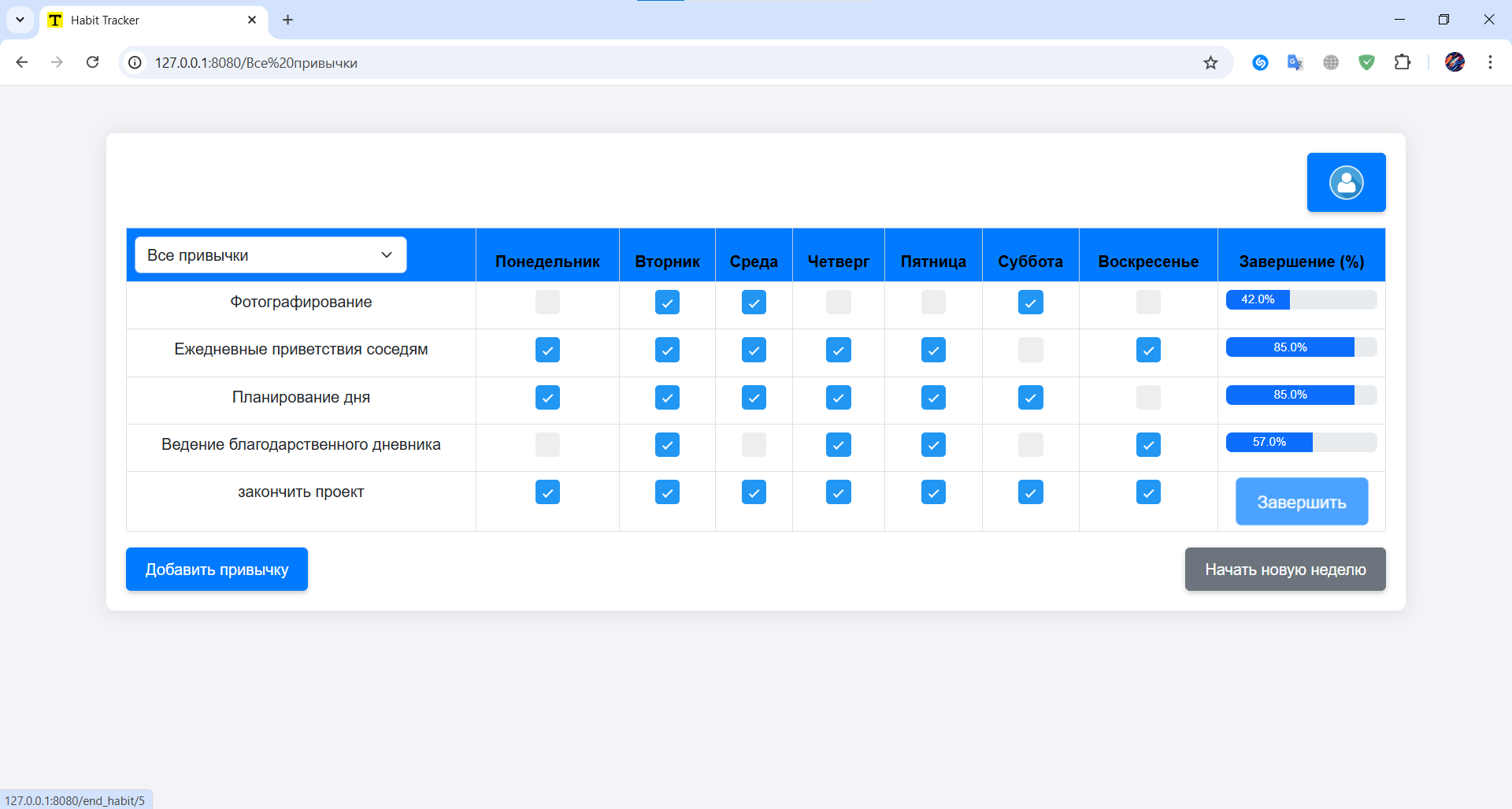
**ПРИЛОЖЕНИЯ**

****

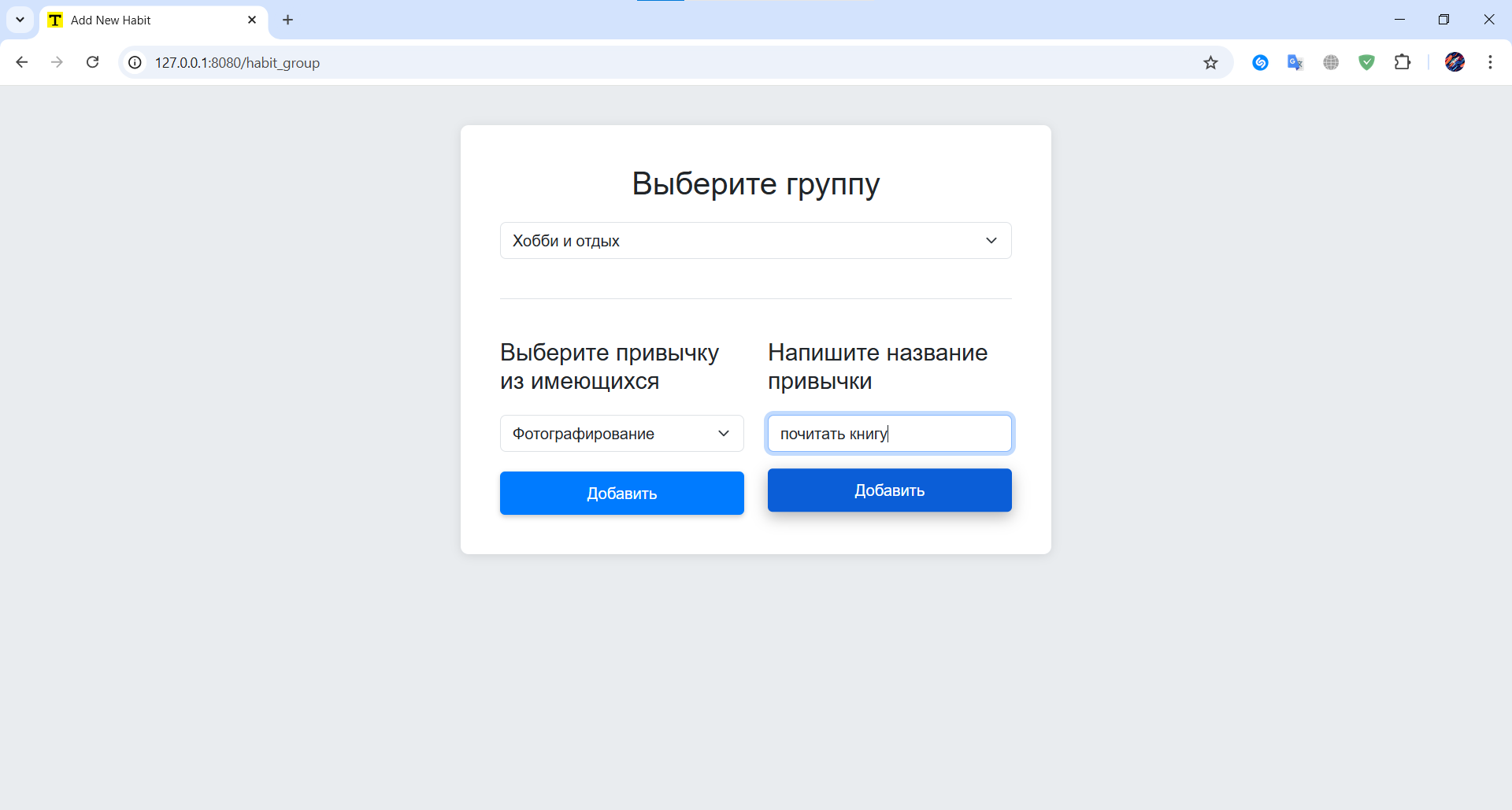
*Иллюстрация 9. Страница регистрации*

****

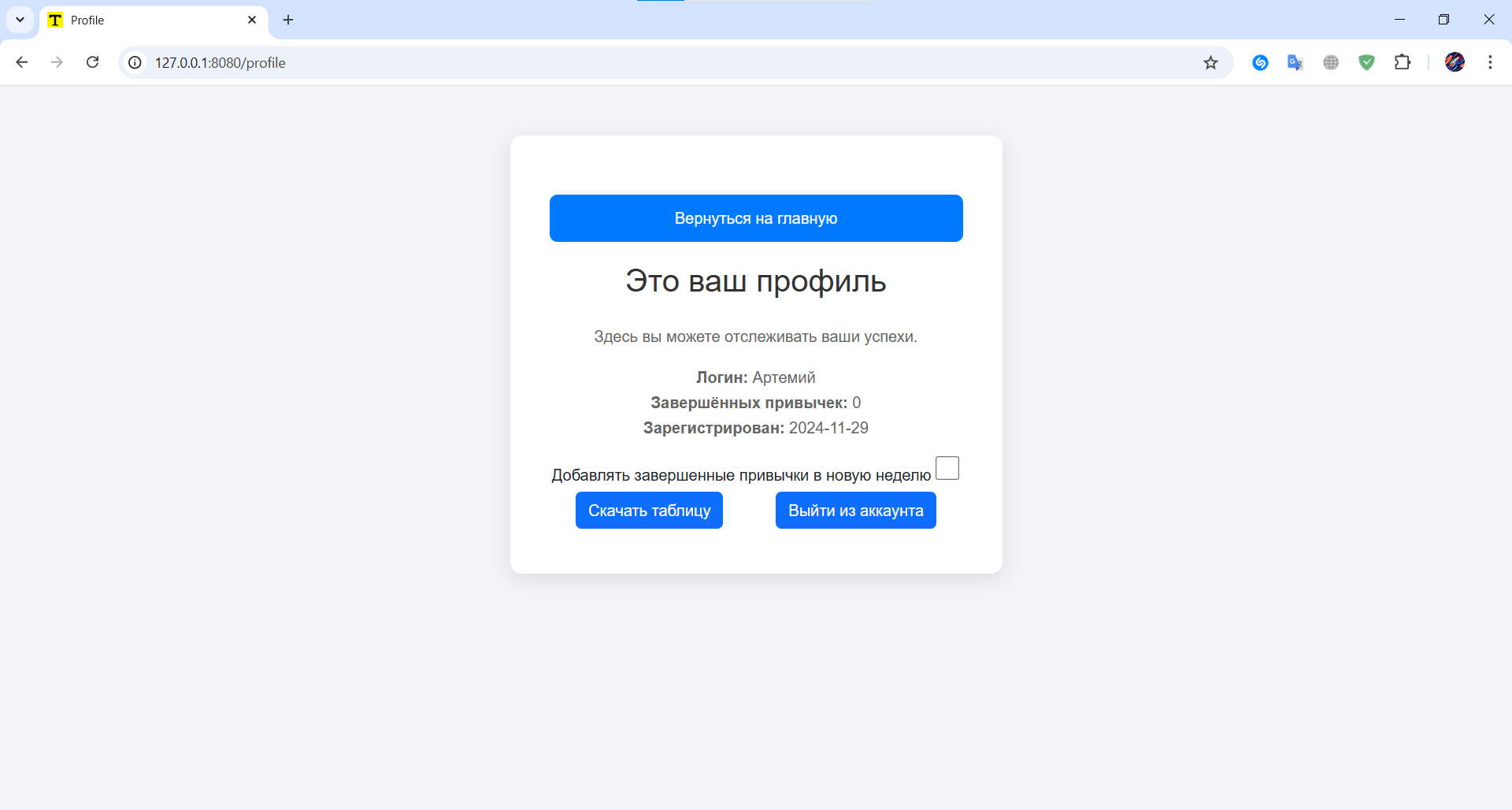
*Иллюстрация 10. Страница входа в профиль*

****

*Иллюстрация 11. Главная страница*

****

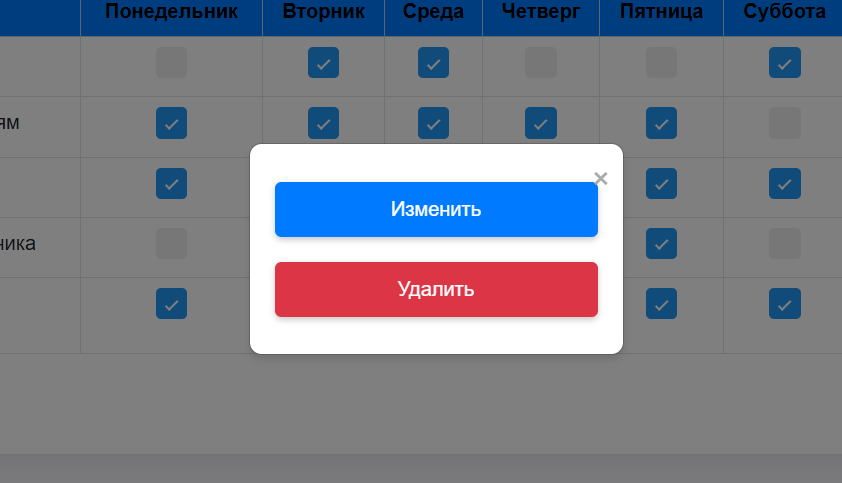
*Иллюстрация 12. Страница добавления привычек*

****

*Иллюстрация 13. Страница профиля*

****

*Иллюстрация 14. Скаченная таблица, в формате docx*

**

*Иллюстрация 15. Окно изменения или удаления привычки*