**ДИПЛОМНАЯ РАБОТА**

**на тему:**

**«Реализация серверной части сайта (backend)**

**для бизнеса на примере финансового аналитика»**

Исполнитель:

Карасева Анастасия

Дмитриевна

*Telegram: https://t.me/karasevaad*

*2024 г.*

ОГЛАВЛЕНИЕ

[1. ВВЕДЕНИЕ 3](#_Toc185447411)

[2. ОПИСАНИЕ ПРОЕКТА 4](#_Toc185447412)

[2.1. Проектирование 4](#_Toc185447413)

[2.2. Разработка проекта 5](#_Toc185447414)

[2.3. Анализ и интерпретация результатов 7](#_Toc185447415)

[ЗАКЛЮЧЕНИЕ 13](#_Toc185447416)

[ПРИЛОЖЕНИЯ 14](#_Toc185447417)

[Приложение 1 14](#_Toc185447418)

[Приложение 2 15](#_Toc185447419)

# 1. ВВЕДЕНИЕ

***Цель проекта:*** разработка серверной части сайта (backend) для бизнеса на примере сайта для действующего финансового аналитика.

Сайт — это лицо компании, недорогой, но эффективный способ рекламы. Наличие собственного сайта существенно повышает имидж организации в глазах потенциальных клиентов и партнёров. Это виртуальный офис, работающий круглосуточно и без выходных. В любой момент пользователь может зайти и получить нужную ему информацию.

Backend-разработка — это создание бизнес-логики цифрового продукта. Бэкенд помогает сделать так, чтобы сервисы правильно обрабатывали запросы пользователей. Любой запрос, который совершает пользователь, передаётся на сервер, запрос обрабатывается, а затем ответ отправляется обратно.

Frontend- и backend-разработчики вместе создают полноценное цифровое решение. Если бэкэнд управляет всем, что скрыто «за сценой», то фронтенд отвечает за визуальную часть и пользовательский опыт.   
Фронтенд — это витрина магазина, бэкенд — это склад и касса, которые обеспечивают всю работу.

В рамках реализации данного проекта будут решены ***следующие задачи:***

* разработка серверной части сайта (backend) с использованием Framework Django**;**
* минимальная настройка фронтенда с инспользлванием html и css;
* оценка полученного результата и формулирование выводов по проделанной работе.

Фреймворк (Framework) – программная платформа, представляющая собой набор инструментов, ускоряющих разработку приложений.

Django — это фреймворк для быстрой разработки сайтов и приложений на Python. Это значит, что с ним можно будет собрать готовый сайт или веб-приложение быстрее, проще и аккуратнее, чем писать весь код самому с нуля.

На Django основано множество сайтов, используемых самым активным образом, в частности, Instagram и Pinterest. Django зародился в издательской среде, поэтому неудивительно, что данный фреймворк используется на таких сайтах, как The Washington Post и Smithsonian Magazine.

Django зарекомендовал себя как надёжный выбор для создания веб-приложенийблагодаря широкому набору функций, сильной поддержке сообщества и масштабируемости. Согласно статистике BuiltWith, в настоящее время более 1,8 миллиона веб-сайтов используют Django.

На сегодняшний день специалистов, обладающих навыками разработки сервисной части сайтов на Django, недостаточно, чтобы удовлетворить потребности клиентов в полном объеме.

Все вышеизложенное подтверждает актуальность выбранной темы проекта.

***Результат проекта*** *–* готовый к использованиюbackend сайта для финансового аналитика.

# 2. ОПИСАНИЕ ПРОЕКТА

## 2.1. Проектирование

***Этапы выполнения проекта:***

1. Выбор архитектуры сайта;

2. Разработка прототипа;

3. Доработка деталей;

4. Добавление docstring для классов и функций и файла зависимостей (requirement.txt);

5. Итоговая проверка функционала сайта в ручном режиме;

6. Загрузка результата на GitHub, подготовка README.

***Основные требования:***

1. Наличие нескольких страниц с возможностью перехода между ними;
2. Наполнение соответствующих страниц услугами и общей информацией;
3. Работа ссылок-переходов на сторонние сайты;
4. Аутентификация пользователя;
5. Реализация функционала корзины.

***Технические требования:***

Бэкенд:

* Язык – Python;
* Фреймворк – Django;
* База данных – SQLite.

Фронтенд:

* HTML;
* CSS.

Проект выполнен на Windows 10.

## 2.2. Разработка проекта

После принятия решения об архитектуре сайта (этап 1), разработка (этап 2) осуществлялась следующим образом:

1. Начало работы:

1.1. Открытие PyCharm и создание проекта Pycharm.

Возможно создать проект напрямую в командной строке и затем открыть через PyCharm, чтобы избежать проблемы с импортами (подчеркиваются красным).

Но для целей настоящего проекта выбран путь через PyCharm. Импорты выделяются красным, но успешно работают;

1.2. Установка фреймворка Django командой «pip install django» в Terminal;

1.3. Создание проекта командой «django-admin startproject WebsiteForBusiness» в Terminal;

1.4. Создание приложения:

- переход в папку проекта командой «cd WebsiteForBusiness» в Terminal;

- создание приложения командой «python manage.py startapp financial\_analyst» в Terminal;

- подключение приложения в файле settings.py;

1.5. Создание базы данных с системными таблицами командой «python manage.py migrate» в Terminal;

1. Написание моделей (models.py) и их дальнейшая миграция:

2.1. Модель Service (Услуги);

2.2. Модель CartItem (Корзина);

2.3. Миграция с помощью команд в Terminal (в папке проекта Django):

«python manage.py makemigrations»;

«python manage.py migrate»;

2.4. Проверка создания базы данных с таблицами в SQLite;

3. Создание админ-панели, супер-пользователя и наполнение базы данных через админ-панель:

3.1. Админ-панель:

- проверка пути в urls.py;

- подключение моделей в файле admin.py;

3.2. Регистрация администратора командой «python manage.py createsuperuser» в Terminal;

3.3. Переход на сайт, вход в админ-панель и добавление данных в таблицу Услуги;

4. Работа с файлами css и html:

4.1. Стилизация страниц с применением собственного шаблона css и доработанного под себя шаблона-примера css (ссылка на шаблон: https://devdevout.com/css/css-list-styles);

4.2. Оформление страниц в html (применение include и block для наследования);

4.3. Вывод объектов из базы данных в html и наполнение страниц в html;

4.4. Добавление пагинации для страницы с услугами;

5. Создание форм регистрации и входа в файле form.py (файл добавлен в папку приложения);

6. Наполнение представления – view.py:

6.1. Создание функций, которые принимают запросы пользователей, обрабатывают их и возвращают шаблоны;

6.2. Согласно функционалу, дополнение функций контентом, пагинатором, формами, http-запросами. В частности, настройка работы корзины (добавление, удаление), регистрации, входа и выхода пользователя.

7. Наполнение urls.py маршрутами.

После всех этапов разработки реализованы этапы 3-6, изложенные в пункте 2.1.

Результат проекта получен в полном объеме.

Файловая структура проекта и перечень необходимых библиотек представлены в приложениях 1 и 2 соответственно.

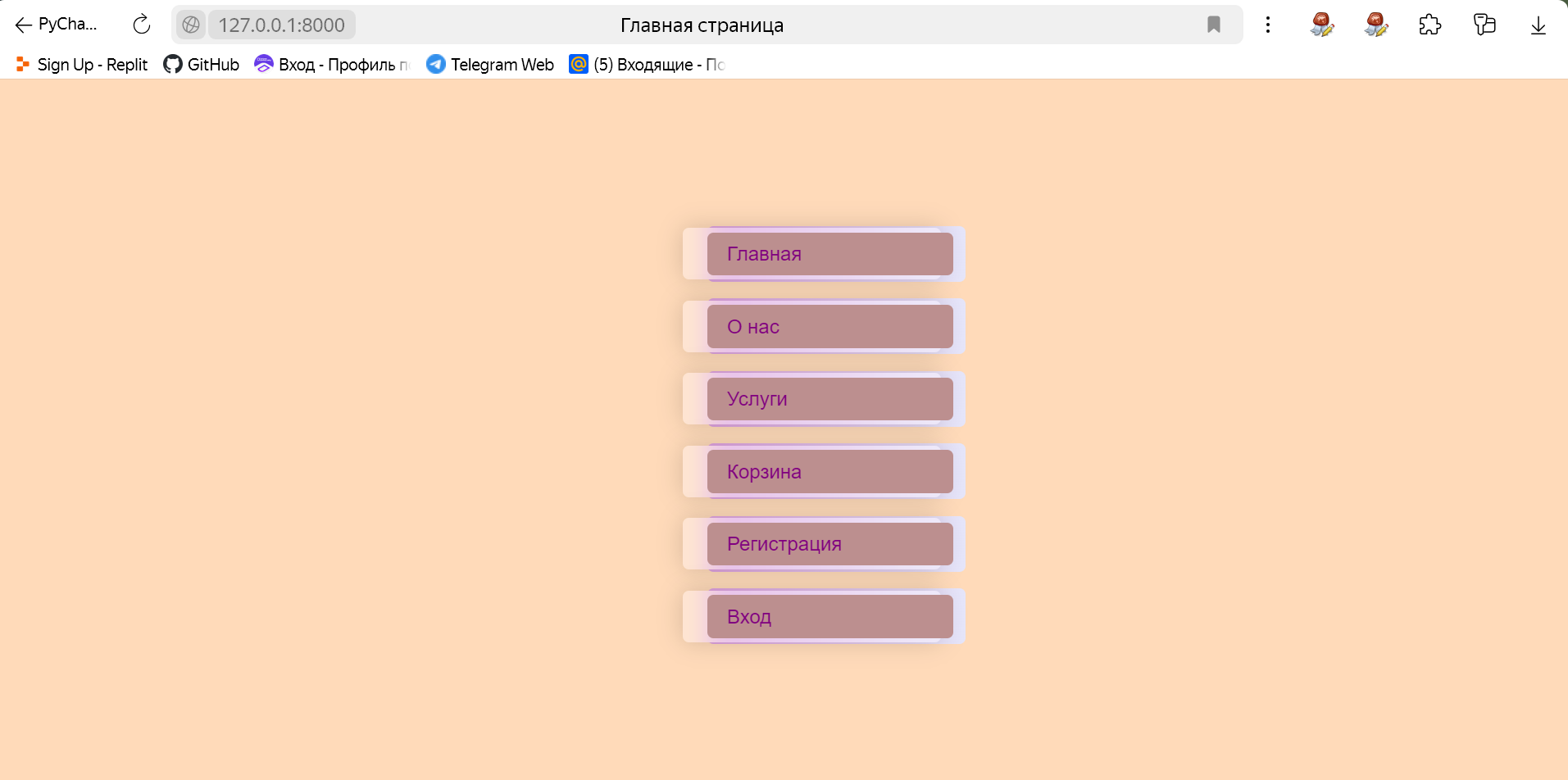
## 2.3. Анализ и интерпретация результатов

***Демонстрация работы (структура сайта)***

Сайт включает следующие страницы:

1. Главная страница.

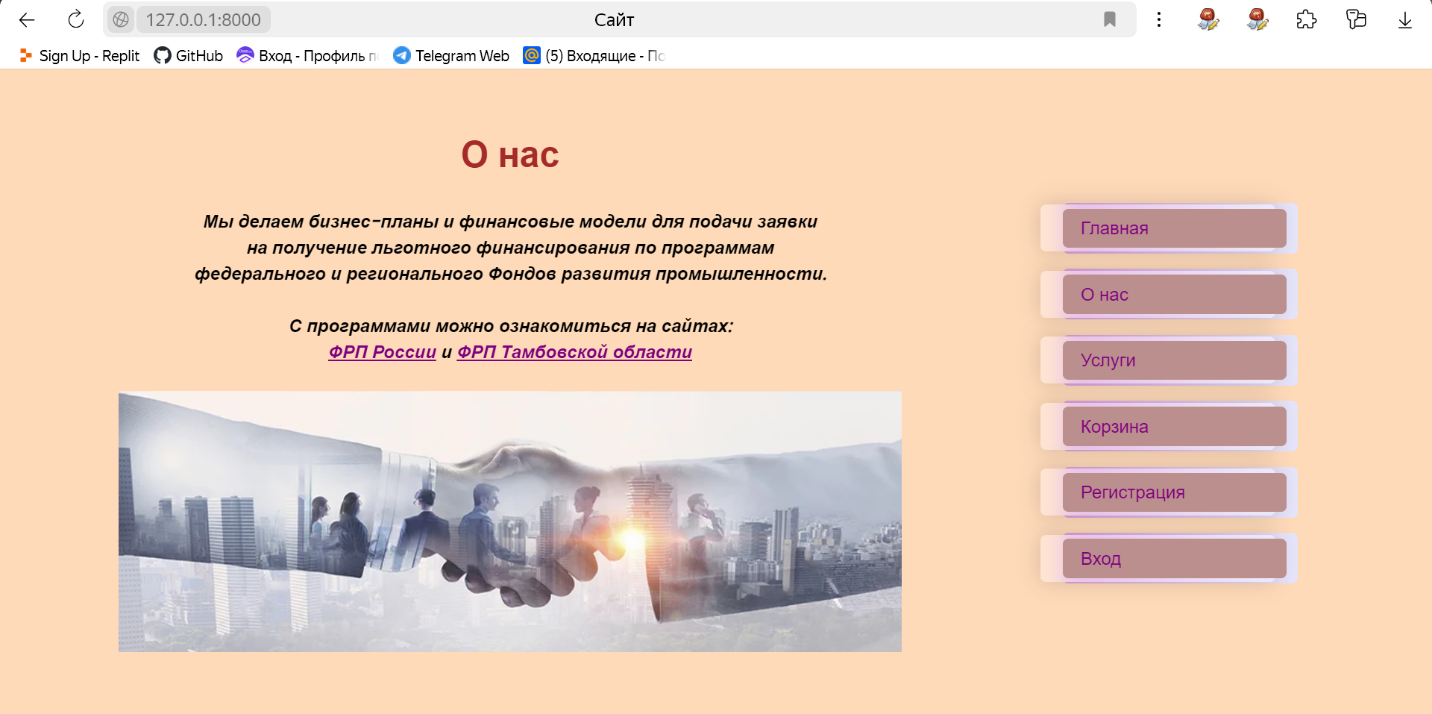
На странице расположены стилизованные кнопки с переходами по другим страницам сайта. Доступ к главной странице открыт всем.



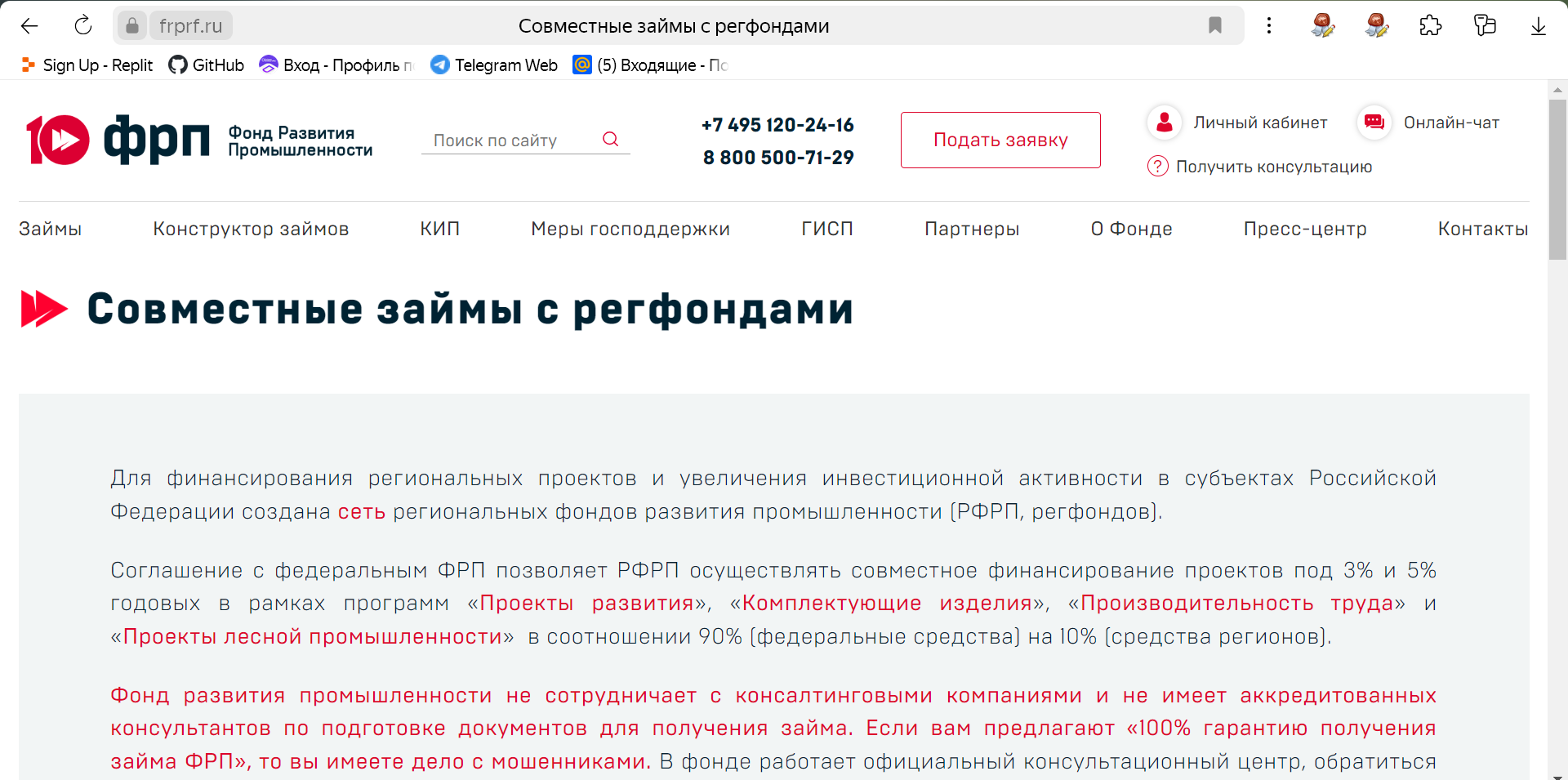
*Рис. 1. Главная страница*

2. О нас.

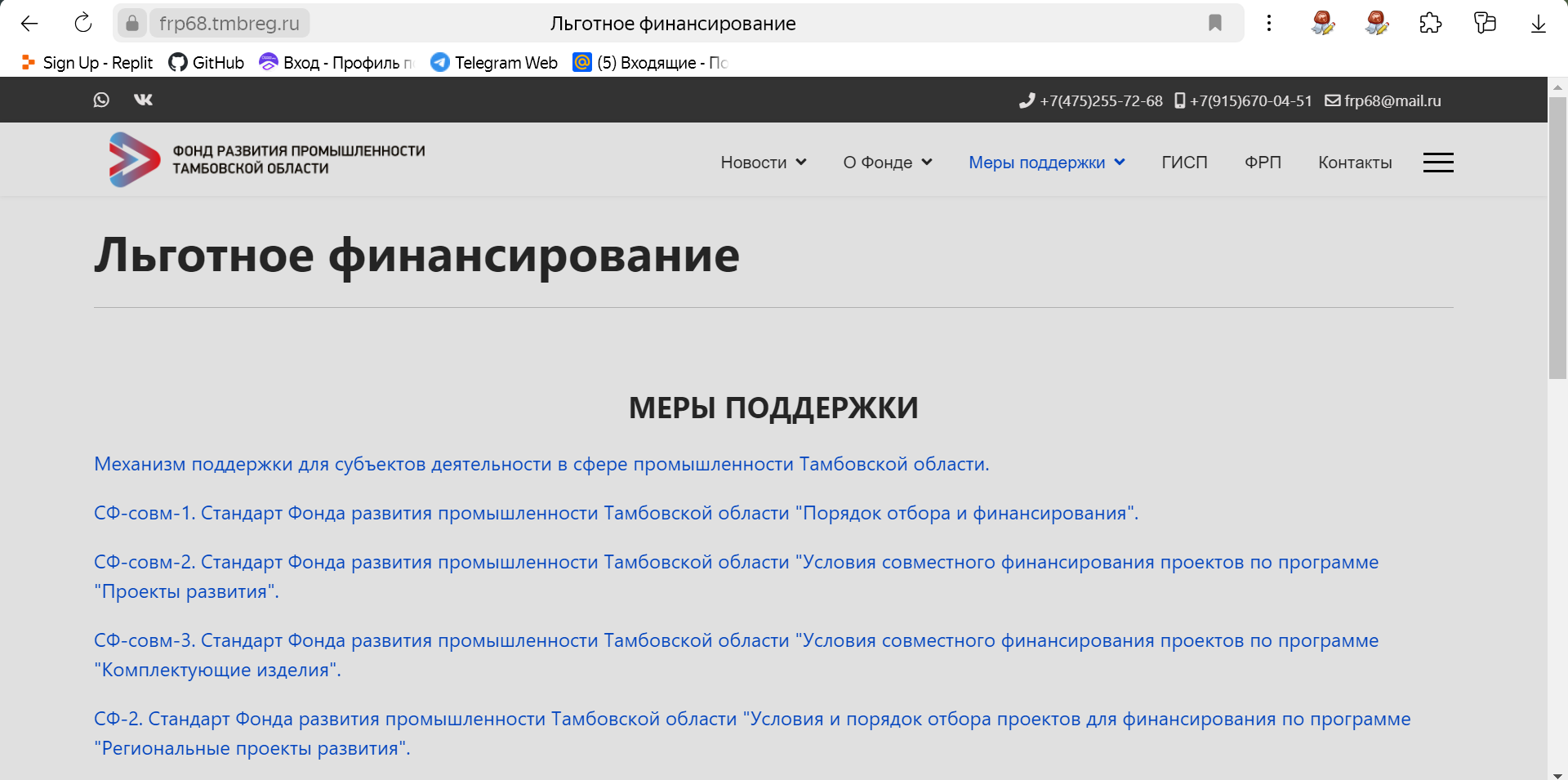
На странице расположена общая информация о компании и услугах. Есть ссылки на другие сайты и общее меню (такое же, как на главной странице, но справа). Доступ к странице открыт всем.



*Рис. 2. О нас*



*Рис. 3. Переход на страницу ФРП России*



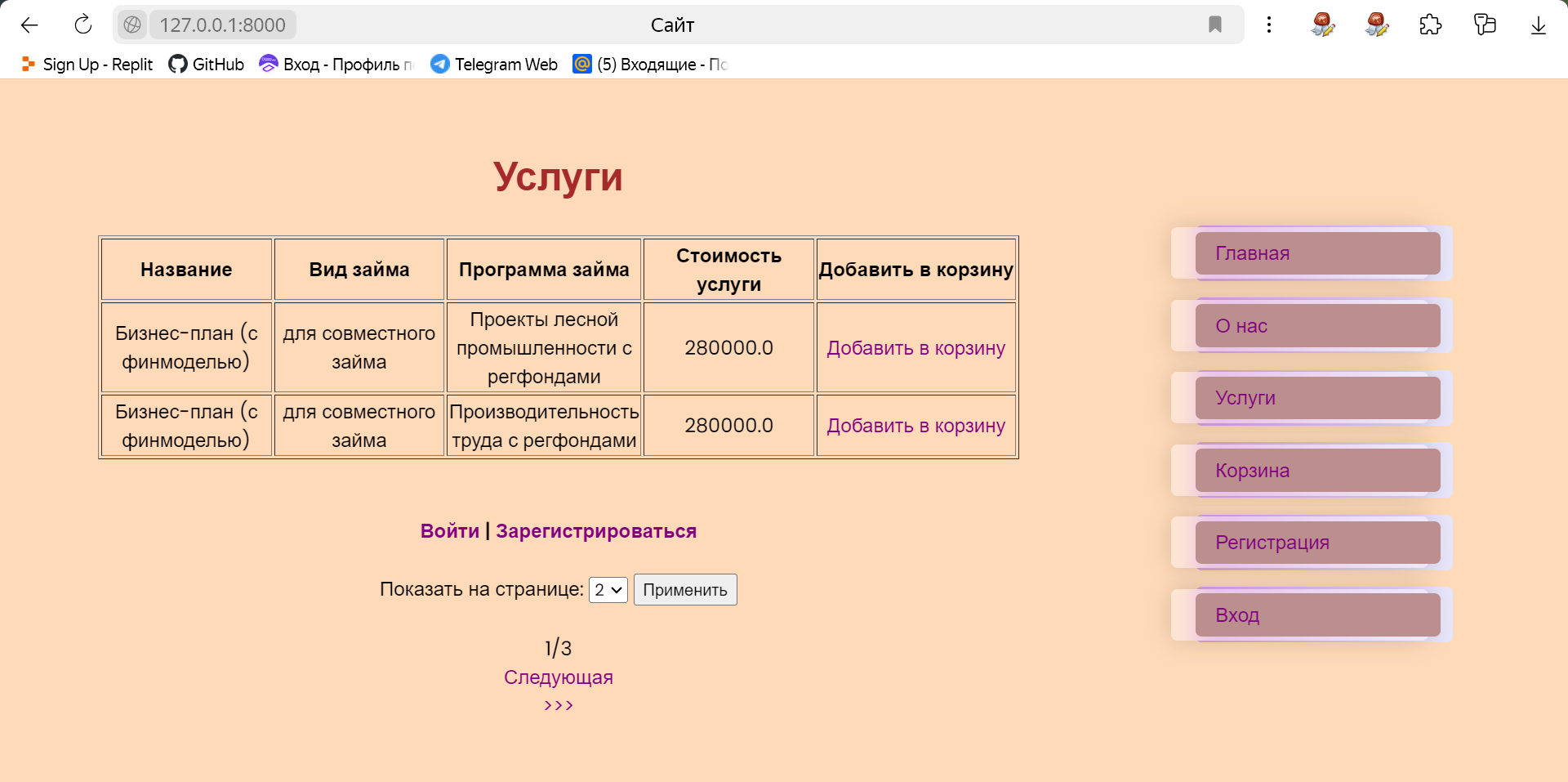
*Рис. 4. Переход на страницу ФРП Тамбовской области*

3. Услуги.

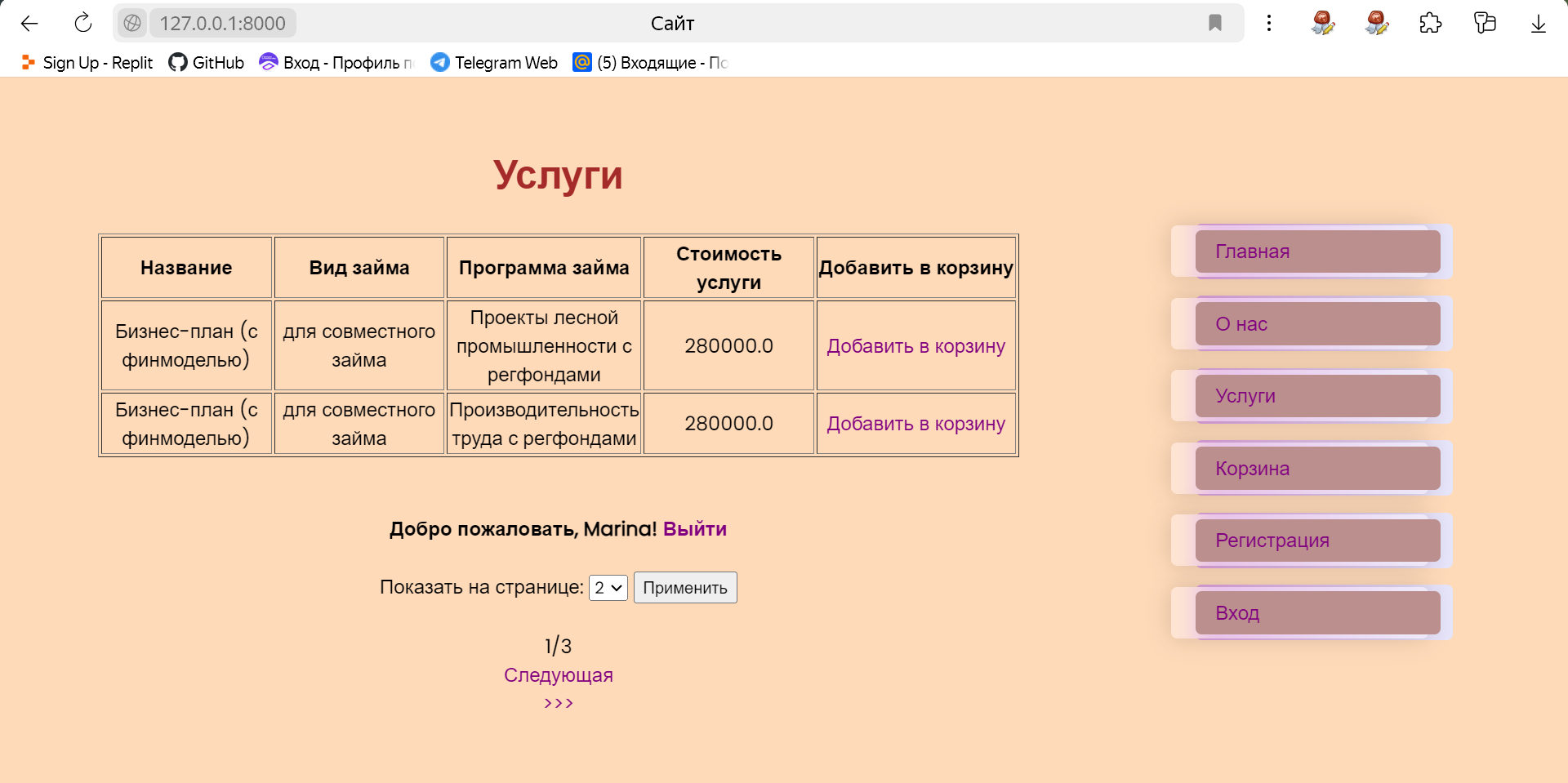
На странице в виде таблицы представлены оказываемые услуги (реализован пагинатор для перехода по страницам, а также механизм выбора количества элементов на странице - 2 или 5). Просмотр таблицы открыт всем пользователям, однако – напротив услуг в таблице есть кнопки «Добавить в корзину». Данные кнопки работают только для авторизованных пользователей, при нажатии неавторизованным пользователем он будет перекинут на страницу «Вход».

Войти или зарегистрироваться также можно на этой странице, как и выйти потом.

Есть общее меню (такое же, как на главной странице, но справа).



*Рис. 5. Услуги (неавторизованный пользователь)*



*Рис. 6. Услуги (авторизованный пользователь)*

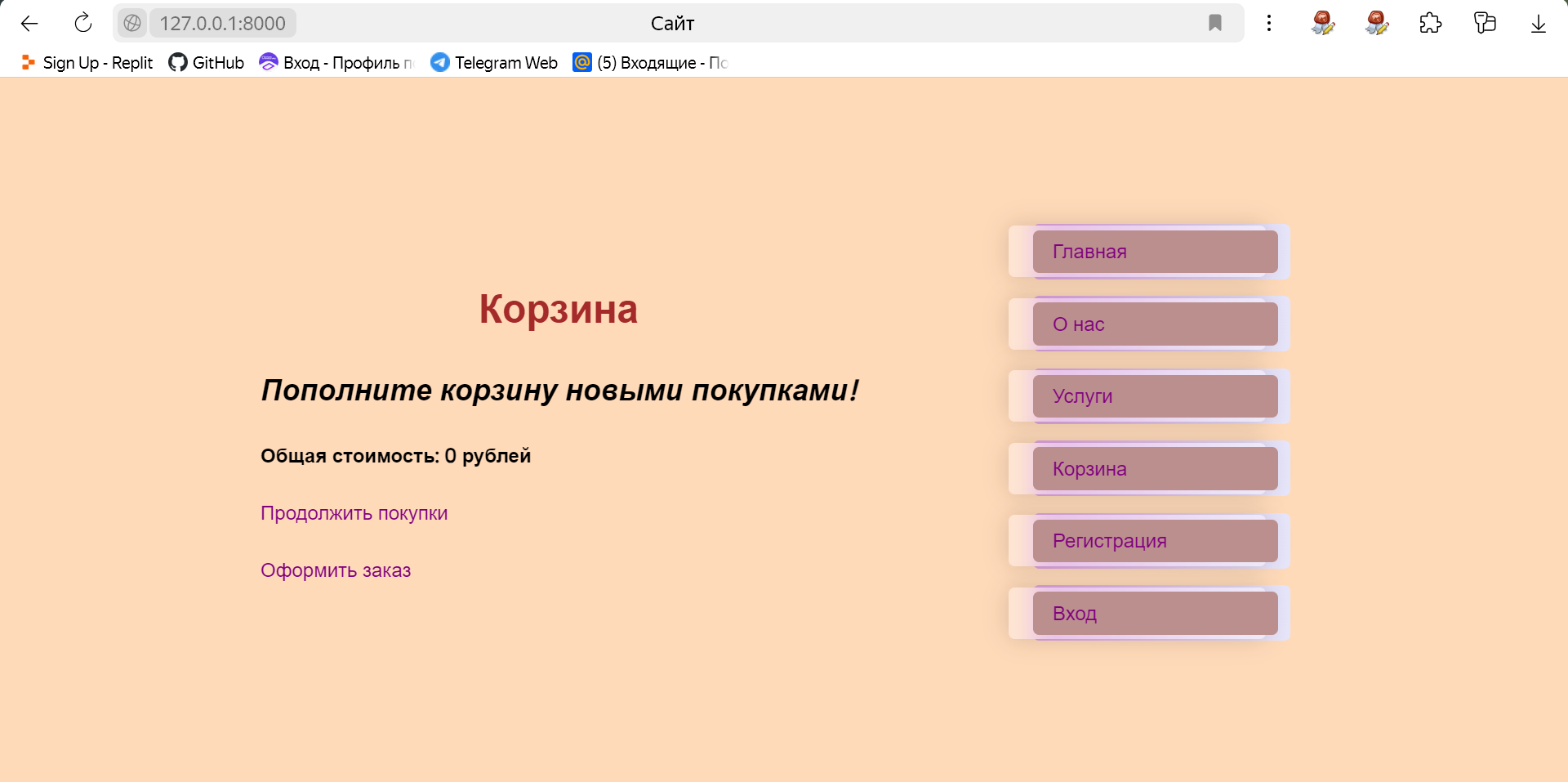
4. Корзина.

Корзина доступна только авторизованным пользователям. При попытке перехода в Корзину неавторизованный пользователь будет перекинут на страницу «Вход».

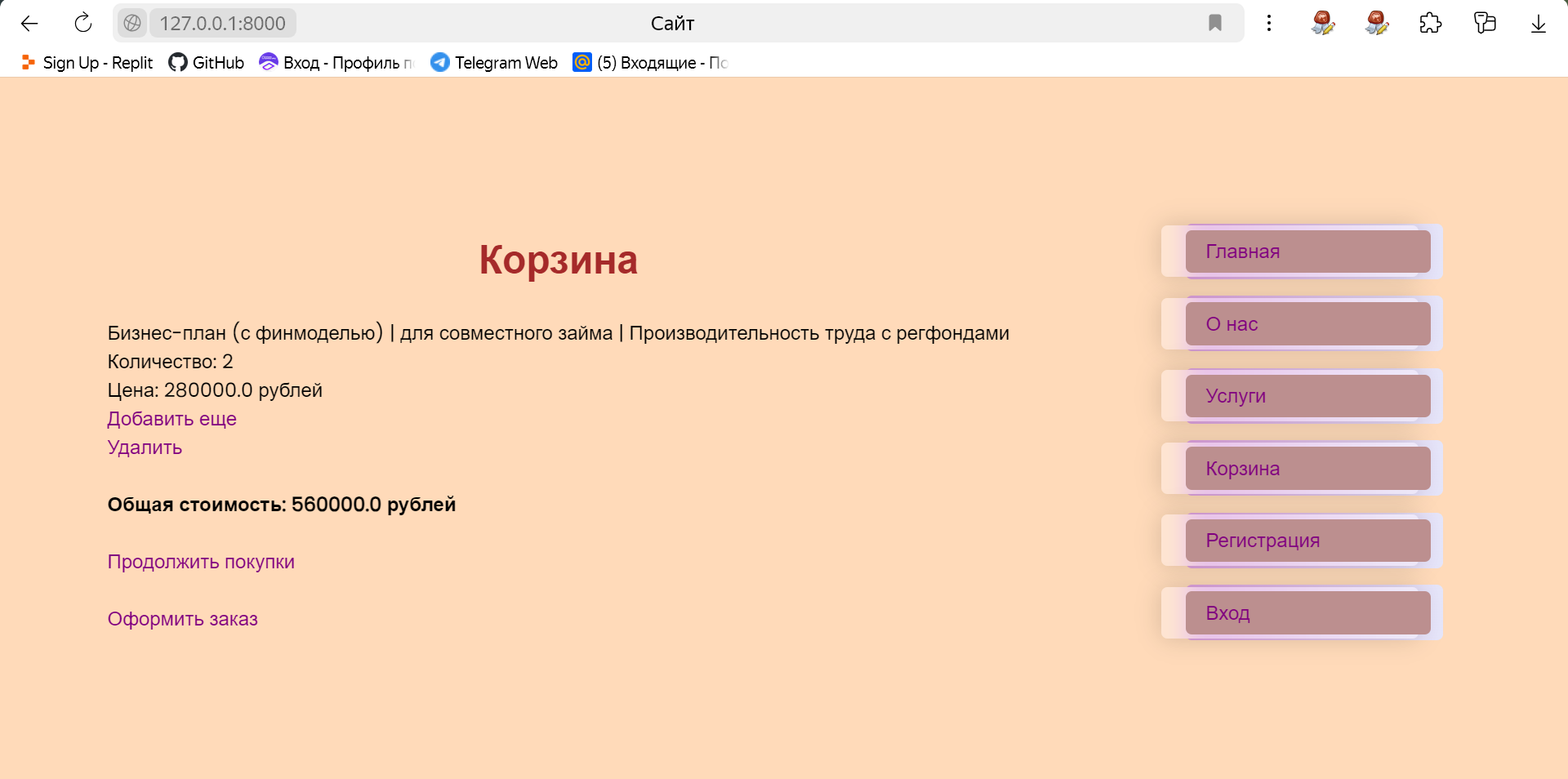
После входа пользователь может:

* увеличивать/уменьшать количество товаров в Корзине,
* продолжать покупки – реализована ссылка на страницу Услуги,
* оформлять заказ – реализована ссылка на страницу разработчика в телеграме (в перспективе возможно доработать и отправлять заказ на почту или в телеграм-бот).

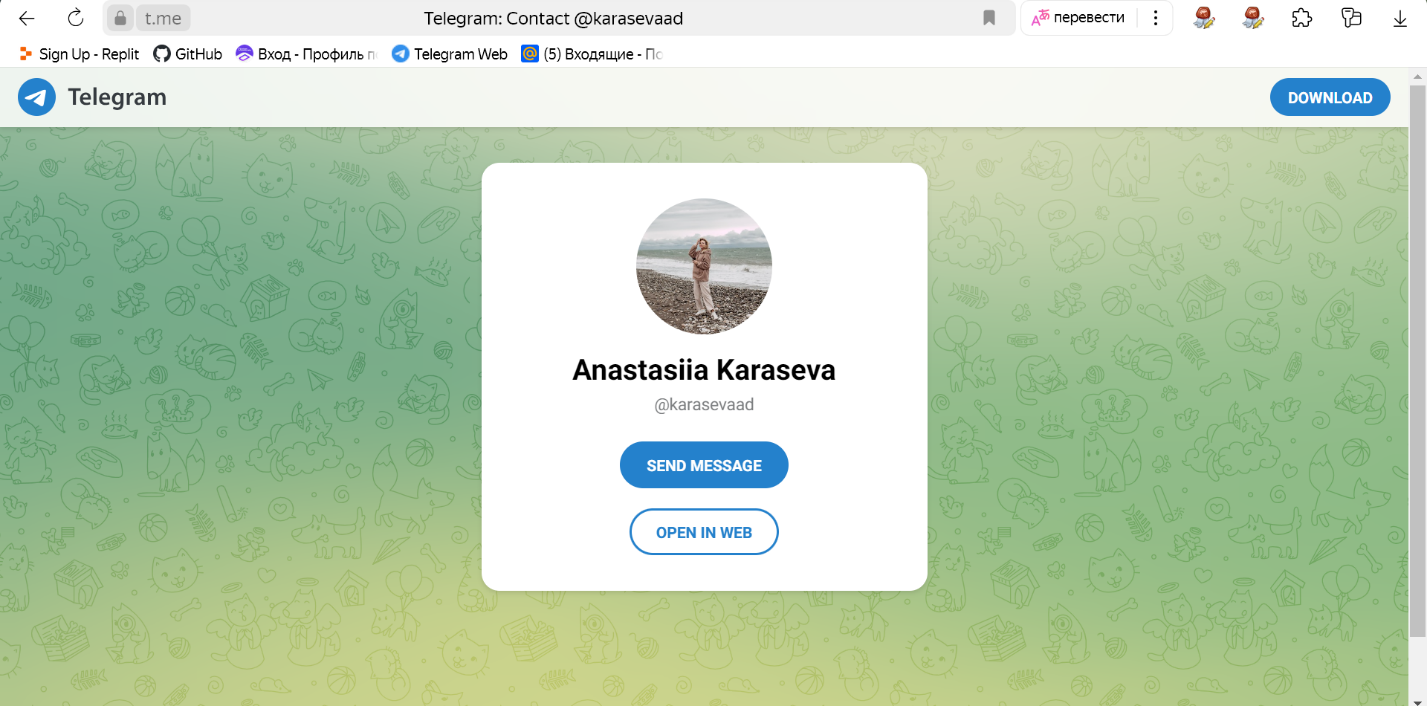
Есть общее меню (такое же, как на главной странице, но справа).



*Рис. 7. Корзина (пустая)*



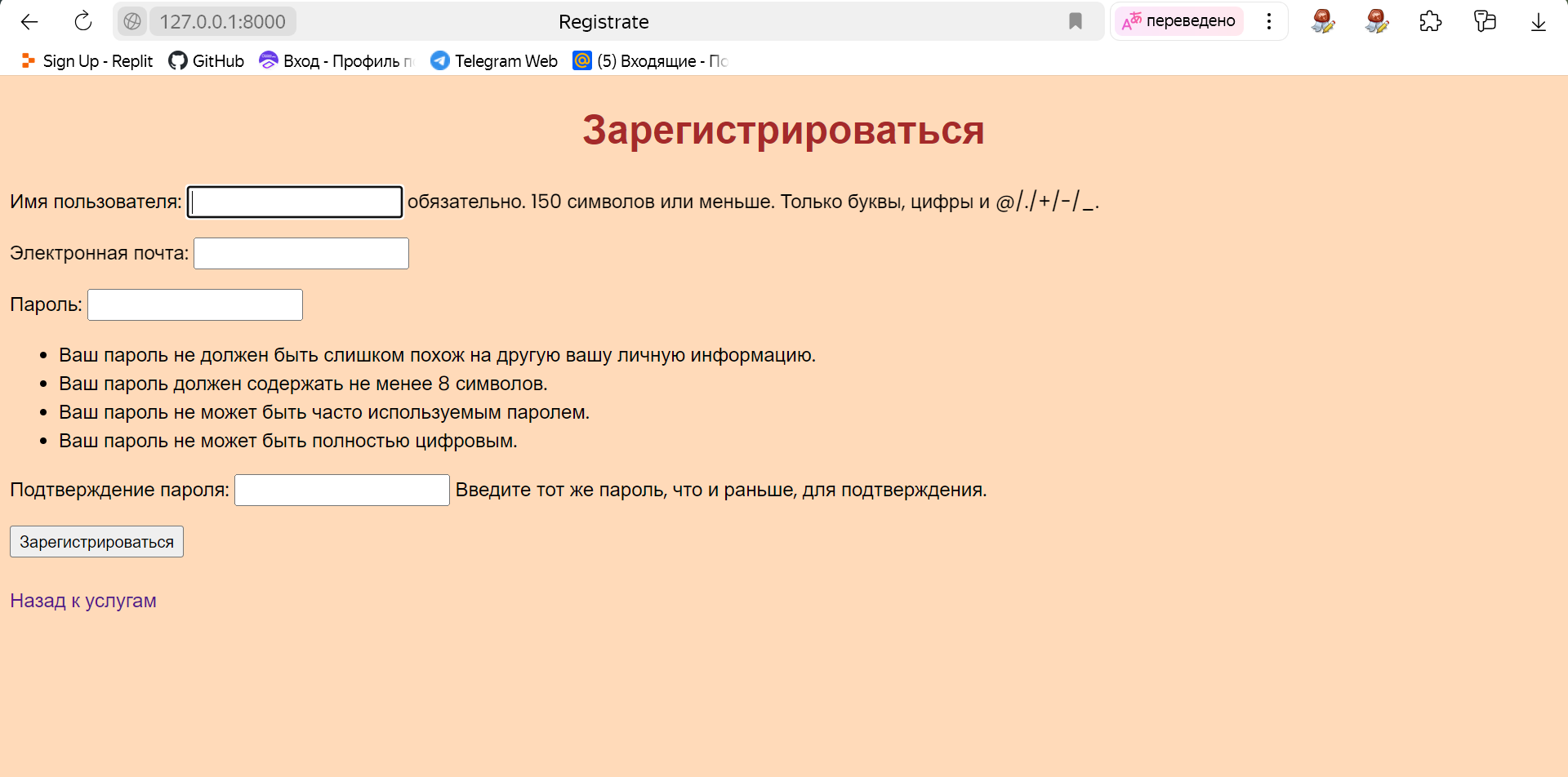
*Рис. 8. Корзина (с услугами)*



*Рис. 9. Оформление заказа (переход из корзины)*

5. Регистрация.

На странице представлена Django-форма для регистрации пользователя. Также в случае, если пользователь передумал, предусмотрен возврат к услугам.

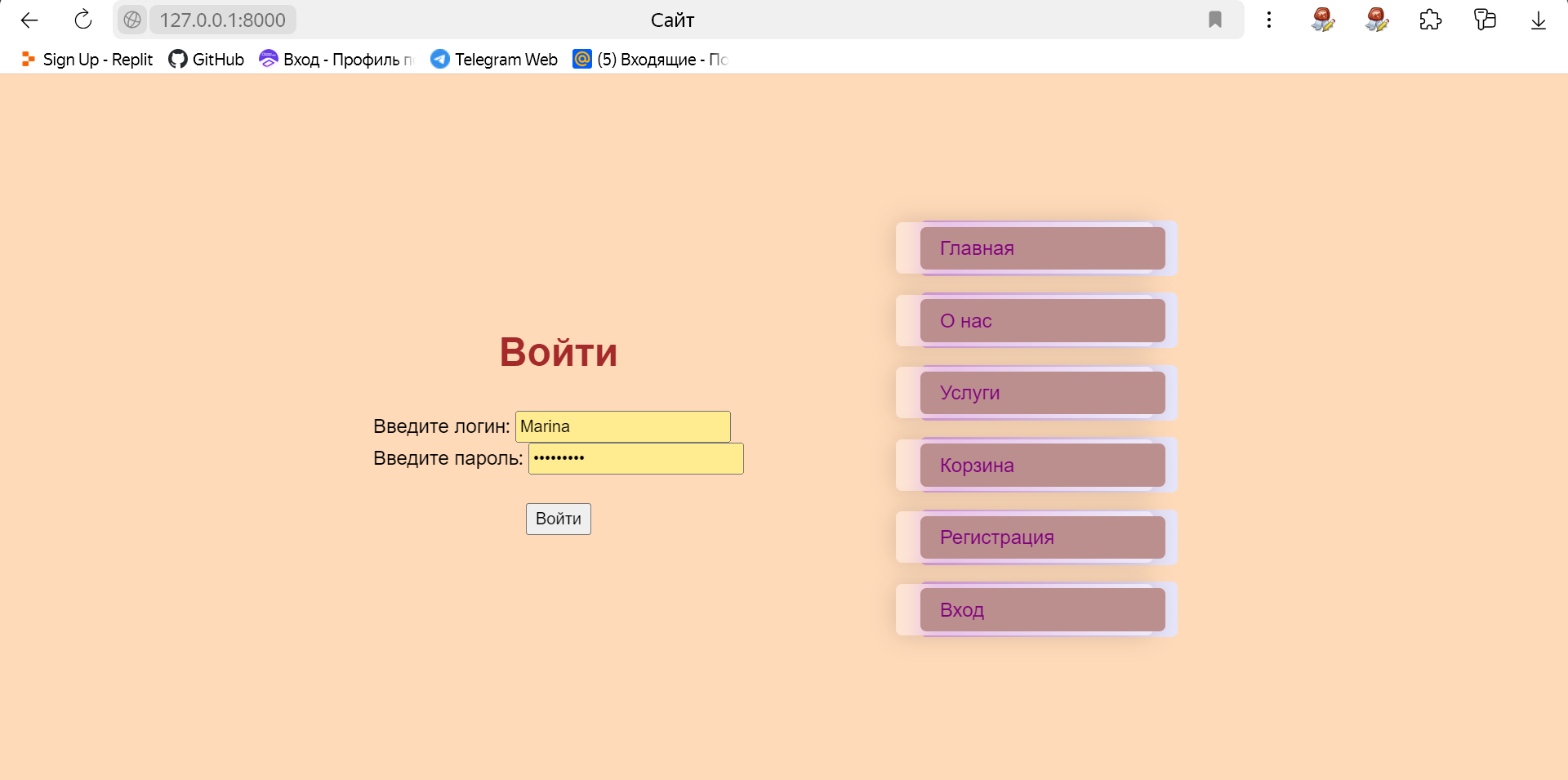


*Рис. 10. Регистрация пользователя*

6. Вход.

На странице представлена Django-форма для входа пользователя.

Есть общее меню (такое же, как на главной странице, но справа).



*Рис. 11. Вход пользователя*

***В ходе анализа получившегося сайта выявлен ряд возможностей, которые еще можно реализовать в будущем:***

1. Дополнить функционал оформления заказа – передача данных через телеграм-бот или на электронную почту;

2. При необходимости добавить пагинацию в корзину. В связи с особенностью деятельности финансового аналитика в настоящий момент этот функционал не требуется. Однако в рамках проекта пагинация успешно реализована на странице Услуги.

# ЗАКЛЮЧЕНИЕ

Проект реализован в соответствии с запланированными этапами разработки, с соблюдением основных требований и технических требований к проекту.

При выполнении этапа разработки использованы следующие *источники информации:*

[Аутентификация в приложениях](https://tproger.ru/articles/kak-nastroit-autentifikaciyu-v-veb-prilozheniyah-na-django)

[Ограничение доступа в приложениях](https://proglib.io/p/django-s-nulya-chast-2-registraciya-avtorizaciya-ogranichenie-dostupa-2022-06-08)

[Функционал корзины](https://www.geeksforgeeks.org/how-to-add-cart-in-a-web-page-using-django/)

[Цвета html](https://colorscheme.ru/html-colors.html)

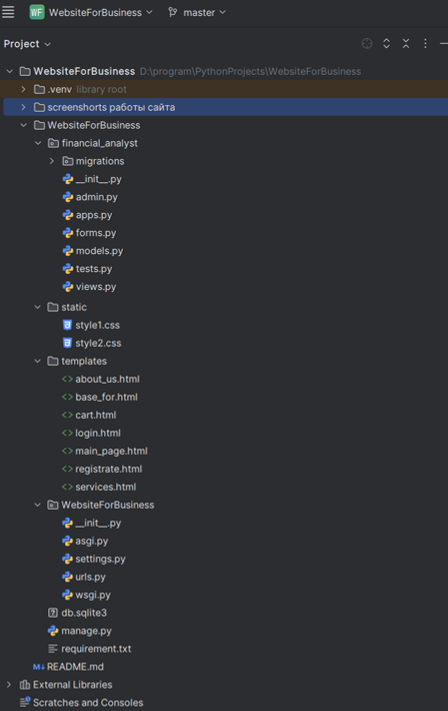
[Таблица основных тегов html](https://uguide.ru/tablica-osnovnykh-tegov-html-s-primerami)

В будущем возможна доработка проекта с учетом требований заказчика. Кроме того, проект может быть масштабирован для других видов бизнеса.

# ПРИЛОЖЕНИЯ

# Приложение 1

Файловая структура проекта



# Приложение 2

Перечень необходимых библиотек

asgiref==3.8.1  
Django==5.1.4  
sqlparse==0.5.3  
tzdata==2024.2