Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Кафедра «Программное обеспечение автоматизированных систем»

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ:  и.о. зав. кафедрой ПОАС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_О.А. Сычев  «\_\_\_» \_\_20\_\_ г. |

Разработка веб-сервиса для сравнения интернет провайдеров по качеству и стоимости услуг

РУКОВОДСТВО СИСТЕМНОГО ПРОГРАММИСТА

ВКРБ–09.03.04–10.19–08–25–32

Листов 11

|  |  |
| --- | --- |
|  | Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_Розалиев В.Л.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |
| Нормоконтролер  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Кузнецова А.С. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г. | Исполнитель  студент группы ПрИн-467  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Керимов Т.И.  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г. |

Волгоград, 2025 г.

Аннотация

Документ представляет собой руководство системного программиста к выпускной работе бакалавра на тему «Разработка веб-сервиса для сравнения интернет провайдеров по качеству и стоимости услуг». В документе изложены общие сведения о программе, структура программы, настройка программы, проверка программы и сообщения системному программисту.

Документ включает в себя страниц — 11.

Ключевые слова: веб-сервис, интернет-провайдеры, сравнение тарифов, отзывы, Docker, FastAPI, React, PostgreSQL, Redis, JWT-авторизация.

Содержание

[1 Общие сведения о программе 4](#_Toc14557)

[2 Структура программы 5](#_Toc5799)

[3. Настройка программы 6](#_Toc7189)

[4 Проверка программы 8](#_Toc1689)

[5 Сообщения системному программисту 11](#_Toc4376)

# 1 Общие сведения о программе

Наименование программы — «ISP Compare» (далее веб-сервис).

Данная программа представляет собой веб-сервис для сравнения интернет-провайдеров по качеству и стоимости услуг. Веб-сервис предоставляет пользователям возможность просматривать и сравнивать тарифные планы различных интернет-провайдеров, фильтровать их по различным параметрам (стоимость, скорость, наличие дополнительных услуг), а также оставлять отзывы о провайдерах и их услугах.

Веб-сервис разработан с использованием следующих технологий:

- backend: Python 3.13, FastAPI;

- frontend: Typescript, React;

- база данных: PostgreSQL 17, Redis;

- контейнеризация: Docker, Docker Compose;

- прокси-сервер: Nginx (используется только в продакшен-среде. При тестировании не обязателен).

В состав технических средств для функционирования серверной части должен входить персональный компьютер или сервер, включающий в себя:

- процессор x86-64 с тактовой частотой не менее 2.0 ГГц;

- оперативную память объемом не менее 4 Гигабайт;

- свободное место на жестком диске не менее 500 Мб;

- сетевую карту с подключением к интернету.

Для клиентской части требуется персональный компьютер с установленным браузером (Google Chrome, Mozilla Firefox, Microsoft Edge, Яндекс.Браузер, Vivaldi) и доступом в интернет.

# 2 Структура программы

Программа состоит из нескольких основных компонентов, функционирующих в контейнерах Docker внутри одной docker-сети.

Backend – серверная часть, написанная на Python, FastAPI. Обрабатывает запросы от frontend-компонента, взаимодействует с базой данных PostgreSQL и Redis.

Frontend – веб-интерфейс, разработан на TypeScript, React 18. Отвечает за отображение интерфейса пользователю.

PostgreSQL – SQL база данных, используемая как основное хранилище данных в приложении. Хранит данные о пользователях, провайдерах, тарифах, отзывах, истории поиска и refresh-токенах.

Redis – NoSQL база данных. Выступает in-memory хранилищем данных для хранения blacklist-токенов, реализации rate-limiting, а также для кеширования часто запрашиваемых данных, например, списка провайдеров.

Nginx – веб-сервер и обратный прокси. Обеспечивает маршрутизацию между frontend и backend. Управляет SSL-сертификатами, а также обеспечивает кеширование статических файлов.

Функциональная структура веб-сервиса должна включать в себя перечисленные ниже функции:

- отображение детальной информации о тарифах провайдеров, включая скорость соединения, стоимость, дополнительные услуги, условия подключения и другие технические характеристики;

- сравнение тарифных планов по различным параметрам (скорость, стоимость, наличие дополнительных услуг) с возможностью выбора до 5 тарифов для одновременного сравнения. Сравнение включает расчет специальной метрики ценности (value score) для определения оптимального тарифа;

- фильтрация тарифов по различным параметрам (ценовой диапазон, скорость интернета, наличие дополнительных услуг);

- регистрация и авторизация пользователей с использованием JWT-токенов;

- возможность оставлять отзывы и оценки провайдерам;

- просмотр отзывов о провайдерах с отображением среднего рейтинга провайдера;

- сохранение истории поиска для авторизованных пользователей;

- административная панель для управления данными о провайдерах и тарифах.

# 3. Настройка программы

Для запуска программы необходима следующая программная среда:

- Docker и Docker Compose версии не ниже 1.29.0;

- Git для клонирования репозитория.

Программа может быть запущена в двух режимах.

Режим разработки (dev):

- предназначен для тестирования и отладки;

- включает в себя расширенное логирование;

Режим продакшн (prod):

- оптимизирован для производительности;

- требует настройки домена и SSL-сертификатов;

- включает в себя Nginx для обработки запросов;

Для тестирования и разработки необходимо использовать режим разработки (dev), так как он не требует дополнительной настройки домена и сертификатов.

# Для установки программы необходимо наличие следующего программного обеспечения и ресурсов:

# - операционная система Linux, macOS или Windows с установленным Docker Engine и Docker Compose;

# - программное обеспечение Git для получения исходных кодов;

# - доступность следующих сетевых портов: 8000 - для серверной части приложения (backend), 5173 - для клиентской части приложения (frontend), 5432 - для системы управления базами данных PostgreSQL, 6379 - для базы данных Redis.

Настройка программы выполняется в следующем порядке.

Получите исходный код приложения из удалённого репозитория:

git clone <https://github.com/bodaue/isp-compare.git>

Перейдите в директорию проекта:

cd isp-compare

Создайте файл конфигурации окружения на основе шаблона:

cp .env.example .env

Отредактируйте файл .env с установкой следующих параметров:

- APPLICATION\_TITLE - название приложения;

- APPLICATION\_DEBUG - режим отладки (True для режима разработки);

- JWT\_SECRET\_KEY - секретный ключ для системы аутентификации;

- POSTGRES\_USER - имя пользователя базы данных;

- POSTGRES\_PASSWORD - пароль для базы данных;

- REDIS\_PASSWORD - пароль для системы кеширования;

- INITIAL\_ADMIN\_PASSWORD - начальный пароль администратора системы.

Запустите приложение в режиме разработки:

docker compose -f docker-compose.dev.yml up --build

После успешного запуска компоненты будут доступны по следующим адресам:

- серверная часть (backend): <http://localhost:8000>;

- клиентская часть (frontend): <http://localhost:5173>;

- административная панель: <http://localhost:8000/admin>;

- документация API: <http://localhost:8000/docs>.

При эксплуатации программы следует учитывать, что доступ к административной панели осуществляется с использованием учетной записи:

- имя пользователя: admin;

- пароль: значение, указанное в параметре INITIAL\_ADMIN\_PASSWORD.

# Для того, чтобы остановить работу программы, выполните следующую команду:

docker compose -f docker-compose.dev.yml down

# 4 Проверка программы

Наиболее подходящий вариант проверки работоспособности данной программы – функциональное тестирование.

Тест-кейс №1: «Регистрация нового пользователя».

Действия тестировщика:

- открыть браузер и перейти на главную страницу сервиса;

- нажать на кнопку «Регистрация» в верхнем меню;

- заполнить поля формы регистрации валидными данными (полное имя: «Иван Иванов», имя пользователя: «ivan123», email: «ivan@example.com», пароль: «Password123»);

- нажать кнопку «Зарегистрироваться».

Ожидаемый результат: пользователь успешно регистрируется в системе, происходит автоматическая авторизация и перенаправление на главную страницу с отображением кнопок «Профиль» и «Выход» вместо кнопок «Вход» и «Регистрация».

Тест-кейс №2: «Регистрация с уже существующим именем пользователя».

Действия тестировщика:

- открыть браузер и перейти на страницу регистрации;

- заполнить форму регистрации, указав имя пользователя «admin» (уже существует в системе);

- нажать кнопку «Зарегистрироваться».

Ожидаемый результат: регистрация не выполняется, отображается сообщение об ошибке «Пользователь с таким именем уже зарегистрирован», пользователь остается на странице регистрации.

Тест-кейс №3: «Регистрация с некорректным паролем».

Действия тестировщика:

- перейти на страницу регистрации;

- заполнить все поля формы;

- в поле пароля ввести «12345678» (без заглавных букв);

- нажать кнопку «Зарегистрироваться».

Ожидаемый результат: регистрация не выполняется, отображается сообщение «Пароль должен содержать хотя бы одну заглавную букву», индикатор силы пароля показывает слабый уровень (weak).

Тест-кейс №4: «Регистрация с уже существующим email».

Действия тестировщика:

- открыть браузер и перейти на страницу регистрации;

- заполнить форму регистрации, указав email, который уже зарегистрирован в системе;

- нажать кнопку «Зарегистрироваться».

Ожидаемый результат: регистрация не выполняется, отображается сообщение «Пользователь с таким email уже зарегистрирован», пользователь остается на странице регистрации.

Тест-кейс №5: «Авторизация пользователя».

Действия тестировщика:

- открыть браузер и перейти на главную страницу сервиса;

- нажать на кнопку «Вход» в верхнем меню;

- ввести корректные данные (имя пользователя: «testuser», пароль: «Test123!»);

- нажать кнопку «Войти».

Ожидаемый результат: пользователь успешно авторизуется в системе, происходит перенаправление на главную страницу с отображением кнопок «Профиль» и «Выход» в верхнем меню.

Тест-кейс №6: «Авторизация с неверными учетными данными».

Действия тестировщика:

- открыть браузер и перейти на страницу авторизации;

- ввести несуществующее имя пользователя или неверный пароль;

- нажать кнопку «Войти».

Ожидаемый результат: авторизация не выполняется, отображается сообщение «Неверное имя пользователя или пароль» с количеством оставшихся попыток входа, после 10 неудачных попыток отображается сообщение «Попробуйте снова через 5 минут».

Тест-кейс №7: «Истечение сессии и обновление токена».

Действия тестировщика:

- авторизоваться в системе;

- оставить приложение открытым на 30 минут;

- попытаться выполнить любое действие, требующее авторизации (например, перейти в профиль).

Ожидаемый результат: система автоматически обновляет токен доступа через refresh token, действие выполняется успешно без необходимости повторной авторизации.

Тест-кейс №8: «Просмотр списка провайдеров».

Действия тестировщика:

- открыть браузер и перейти на главную страницу сервиса;

- нажать на пункт меню «Провайдеры».

Ожидаемый результат: отображается страница со списком доступных провайдеров с указанием названия, логотипа (или заглушки с первой буквой) и рейтинга для каждого провайдера.

Тест-кейс №9: «Просмотр детальной информации о провайдере».

Действия тестировщика:

- открыть браузер и перейти на страницу провайдеров;

- выбрать одного из провайдеров в списке и кликнуть на его карточку.

Ожидаемый результат: отображается страница с детальной информацией о выбранном провайдере с вкладками «Тарифы» и «Отзывы», по умолчанию активна вкладка «Тарифы» с отображением списка тарифных планов и ссылкой на официальный сайт провайдера.

Тест-кейс №10: «Просмотр детальной информации о тарифе».

Действия тестировщика:

- открыть браузер и перейти на страницу «Тарифы»;

- выбрать любой тариф и нажать на кнопку «Подробнее».

Ожидаемый результат: открывается страница с детальной информацией о тарифе, включающая название, провайдера, цену (обычную и акционную), скорость, дополнительные услуги (ТВ, телефония), информацию о стоимости подключения и кнопку «Подключить тариф».

Тест-кейс №11: «Фильтрация тарифов».

Действия тестировщика:

- открыть браузер и перейти на страницу «Тарифы»;

- нажать кнопку «Фильтры»;

- установить параметры: цена от 500 до 1000 руб., скорость от 100 Мбит/с, наличие ТВ;

- нажать кнопку «Применить».

Ожидаемый результат: отображается список тарифов, соответствующих заданным критериям (цена 500-1000 руб., скорость от 100 Мбит/с, включено ТВ), параметры поиска сохраняются в истории для авторизованных пользователей.

Тест-кейс №12: «Восстановление последних параметров поиска».

Действия тестировщика:

- авторизоваться в системе;

- перейти на страницу «Тарифы»;

- установить фильтры и применить их;

- обновить страницу браузера;

- нажать кнопку «Восстановить фильтры».

Ожидаемый результат: восстанавливаются последние использованные параметры фильтрации, автоматически применяются фильтры и отображаются результаты с учетом восстановленных параметров.

Тест-кейс №13: «Сравнение тарифов».

Действия тестировщика:

- открыть браузер и перейти на страницу «Тарифы»;

- нажать кнопку «Сравнить тарифы»;

- выбрать 3 тарифа, отметив их галочками;

- нажать кнопку «Сравнить выбранные (3)».

Ожидаемый результат: происходит переход на страницу сравнения с таблицей выбранных тарифов, где представлены цена, скорость, дополнительные услуги, годовая стоимость, выделены лучшие тарифы по категориям, отображается «Оценка ценности» и персонализированные рекомендации.

Тест-кейс №14: «Добавление отзыва о провайдере».

Действия тестировщика:

- авторизоваться в системе;

- перейти на страницу конкретного провайдера;

- перейти на вкладку «Отзывы»;

- нажать кнопку «Написать отзыв»;

- выставить рейтинг: 4 звезды;

- ввести текст отзыва: «Хороший провайдер, стабильное соединение»;

- нажать кнопку «Отправить отзыв».

Ожидаемый результат: отзыв успешно добавляется и отображается в списке отзывов, обновляется общий рейтинг провайдера, пользователь видит свой отзыв с кнопками «Редактировать» и «Удалить».

Тест-кейс №15: «Попытка добавить второй отзыв одному провайдеру».

Действия тестировщика:

- авторизоваться в системе;

- перейти на страницу провайдера, где уже есть отзыв от данного пользователя;

- перейти на вкладку «Отзывы».

Ожидаемый результат: кнопка «Написать отзыв» не отображается, пользователь может только редактировать или удалить существующий отзыв.

Тест-кейс №16: «Редактирование собственного отзыва».

Действия тестировщика:

- авторизоваться в системе;

- перейти на страницу провайдера с оставленным отзывом;

- на вкладке «Отзывы» нажать кнопку «Редактировать» у своего отзыва;

- изменить рейтинг на 5 звезд и текст на «Отличный сервис!»;

- нажать кнопку «Сохранить изменения».

Ожидаемый результат: отзыв успешно обновлен, обновляется общий рейтинг провайдера, изменения сразу отображаются в списке отзывов.

Тест-кейс №17: «Удаление собственного отзыва».

Действия тестировщика:

- авторизоваться в системе;

- перейти на страницу провайдера с оставленным отзывом;

- на вкладке «Отзывы» нажать кнопку «Удалить» у своего отзыва;

- подтвердить удаление в диалоговом окне.

Ожидаемый результат: отзыв успешно удален из списка, обновляется общий рейтинг провайдера, появляется возможность добавить новый отзыв с отображением кнопки «Написать отзыв».

Тест-кейс №18: «Редактирование профиля пользователя».

Действия тестировщика:

- авторизоваться в системе;

- перейти в раздел «Профиль»;

- нажать на вкладку «Редактировать»;

- изменить полное имя на «Петр Петров»;

- изменить имя пользователя на «petr\_new»;

- нажать кнопку «Сохранить изменения».

Ожидаемый результат: данные профиля успешно обновлены, отображается сообщение «Профиль успешно обновлен», новые данные отображаются во вкладке «Профиль».

Тест-кейс №19: «Смена пароля».

Действия тестировщика:

- авторизоваться в системе;

- перейти на страницу «Профиль»;

- перейти на вкладку «Сменить пароль»;

- ввести текущий пароль: «OldPass123»;

- ввести новый пароль: «NewPass456»;

- подтвердить новый пароль: «NewPass456»;

- нажать кнопку «Изменить пароль».

Ожидаемый результат: пароль успешно изменен, отображается сообщение «Пароль успешно изменен», при следующей авторизации необходимо использовать новый пароль.

Тест-кейс №20: «Подключение тарифа через модальное окно».

Действия тестировщика:

- перейти на страницу детальной информации о тарифе;

- нажать кнопку «Подключить тариф»;

- в открывшемся модальном окне просмотреть контактную информацию;

- нажать кнопку «Скопировать номер».

Ожидаемый результат: открывается модальное окно с информацией о подключении, отображается телефон провайдера и ссылка на сайт, при нажатии на «Скопировать номер» телефон копируется в буфер обмена и кнопка временно меняется на «✓ Скопировано», в системе аналитики фиксируется достижение цели.

# 5 Сообщения системному программисту

В случае, если программа завершила выполнение с ошибкой или во время работы произойдут ошибки выполнения, она выводит соответствующее уведомление с описанием ошибки.