

1 Введение

Продукт **PulseNews** – это веб-приложение, представляющее собой онлайн новостной портал с широкими возможностями для пользователей. Система позволяет пользователям искать, читать, добавлять, изменять и удалять книги, а также взаимодействовать с другими пользователями через комментарии, лайки и статусы чтения.

Границы проекта: это веб-приложение, представляющее собой онлайн-новостной портал с широким функционалом для пользователей. Система позволяет читать актуальные материалы, искать публикации по тегам, делиться новостями, а также взаимодействовать с другими пользователями через комментарии, лайки и дизлайки.

2 Требования пользователя

2.1 Программные интерфейсы

Продукт будет взаимодействовать со следующими внешними и внутренними системами:

- **PostgreSQL:** Основная реляционная база данных для хранения всей структурированной информации: статей, пользователей, категорий, тегов.

- **Redis:** Сервер кэширования для ускорения обработки частых запросов (например, главной страницы). Также выступает в качестве брокера сообщений для обработки фоновых задач.

- **Внешние новостные источники:** Веб-сайты, данные с которых будут собираться персерами для наполнения базы данных.

- **Nginx:** Веб-сервер, который будет обслуживать статические файлы фронтенда.

Основной стек внутренних сервисов:

- **Для серверной части:** Python, Django, Django REST Framework, Celery, Gunicorn.

- **Для клиентской части:** TypeScript, React.

- **Для парсинга новостей:** Newspaper3k, BeautifulSoup4.

2.2 Интерфейс пользователя

Система будет взаимодействовать с пользователем через веб-сайт. Основные элементы интерфейса и их поведение:

- **Лента новостей:** Центральный хаб портала. Отображает ленту последних новостей, отсортированных по дате публикации (сначала самые свежие).

- **Страница статьи:** Детальное представление выбранной новости.

- **Страница категорий и тэгов:** Страница, которая отображает все статьи, принадлежащие к определенной или помеченные определенным тегом.

- **Поиск и фильтрация:** Система для поиска и уточнения запросов пользователя.

- **Административный интерфейс (CRUD):** Защищенный раздел для редакторов и администраторов.

2.3 Характеристики пользователей

Рядовые пользователи: Основная масса аудитории. Люди с разным уровнем технической грамотности, которые приходят на сайт, чтобы узнать новости, прочитать статьи на интересующие темы, возможно, обсудить их.

Авторы/Администраторы: Команда, отвечающая за наполнение и модерацию контента на портале. Это могут быть журналисты, редакторы, владельцы сайта. Как правило, они имеют более высокий уровень технической грамотности.

2.4 Предположения и зависимости

Предполагается, что пользователи имеют стабильное интернет-соединение для доступа к приложению.

Сервисы, такие как PostgreSQL, должны быть доступны и работать без сбоев для корректного функционирования системы.

Обеспечение безопасности пользовательских данных и контента является критически важным.

3 Системные требования

3.1 Функциональные требования

- Система должна предоставлять аутентифицированным пользователям с правами автора/редактора возможность создавать, редактировать, публиковать и удалять новостные статьи.

- Система должна предоставлять возможность назначать статьи к категориям и добавлять к ним теги.

- Система должна предоставлять администраторам возможность настраивать и запускать парсинг новостей с заданных внешних источников в ручном и автоматическом режиме (по расписанию).

- Система должна предоставлять пользователю возможность просматривать ленту новостей, отсортированную по дате публикации.

- Система должна предоставлять пользователю возможность искать новости по ключевым словам в заголовке и тексте статьи, а также фильтровать их по категориям, тегам и дате.

- Система должна позволять пользователям регистрироваться и аутентифицироваться на портале.

- Аутентифицированные пользователи должны иметь возможность оставлять комментарии к новостным статьям.

- Аутентифицированные пользователи должны иметь возможность ставить реакции (лайки, дизлайки) статьям и/или комментариям.

- Аутентифицированные пользователи должны иметь возможность сохранять статьи в "Закладки" для последующего чтения

3.2 Нефункциональные требования

3.2.1 АТРИБУТЫ КАЧЕСТВА

- **Надёжность:** Система должна быть устойчива к сбоям. Так как архитектура распределенная, сбой одного микросервиса не должен приводить к полному отказу всей системы. Для этого важна правильная настройка отказоустойчивости и мониторинга, измеряемая через показатель **MTBF (Mean Time Between Failures)**.

- **Производительность:** Приложение должно обрабатывать запросы пользователей быстро, особенно при поиске и загрузке книг.

Это измеряется через **время отклика** для ключевых операций, таких как поиск и чтение.

- **Безопасность:** Пользовательские данные (пароли, личная информация) и права доступа должны быть надёжно защищены. Все данные, передаваемые между фронтендом и бэкендом, должны быть зашифрованы. Это измеряется через **количество выявленных уязвимостей** при тестировании на проникновение.

- **Масштабируемость:** Архитектура микросервисов должна позволять масштабировать каждый сервис независимо, чтобы справляться с ростом нагрузки и количества пользователей. Это можно измерять по **максимальному количеству одновременных пользователей**, которое система может поддерживать без деградации производительности.

- **Удобство использования (Usability):** Интерфейс должен быть интуитивно понятным для пользователей с разным уровнем технической грамотности. Это можно оценить через **тестирование пользовательского опыта** и сбор обратной связи.

