

Прогрессивные веб- технологии

Ларионов Михаил Юрьевич

 Проектный офис
ИКТИБ

Институт
Компьютерных
Технологий и
Информационной
Безопасности



Техническое задание

Задача: Рассмотреть современные прогрессивные веб-технологии, которые способны улучшить пользовательский опыт и бизнес-эффективность.

Цель: Исследовать возможности внедрения и использования прогрессивных веб-приложений (**PWA**), **WebAssembly**, **WebRTC**, а также передовые подходы к кросс-платформенной разработке.

Проблема

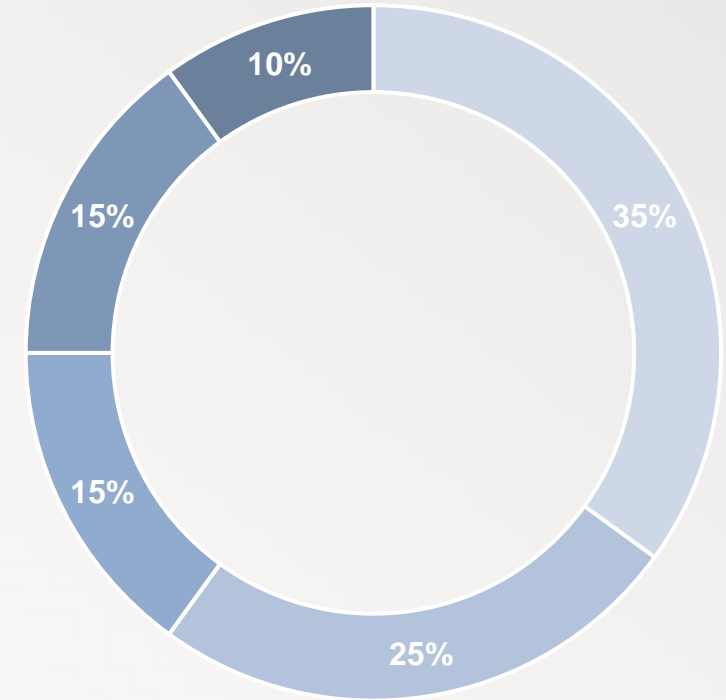
Наиболее распространённые проблемы:

- Долгая загрузка страниц и медленный отклик;
- ограниченные возможности для работы оффлайн;
- проблемы с безопасностью и конфиденциальностью данных;
- высокая стоимость разработки приложений для каждой платформы (**iOS, Android, web**);
- потребность в улучшении пользовательского взаимодействия.

Целевая аудитория

Потенциальные клиенты и заказчики:

1. Средний и малый бизнес (35%):
 - Ищут способы сократить расходы на разработку и улучшить клиентский опыт;
 - Примеры: онлайн-магазины, сервисы доставки, локальные компании.
2. Крупные корпорации и IT-компании (25%):
 - Стремятся к улучшению производительности своих веб-приложений и внедрению новых решений для клиентов;
 - Примеры: финансовые и медицинские сервисы, образовательные платформы, e-commerce.
3. Образовательные учреждения и EdTech (15%):
 - Интересуются внедрением WebRTC для организации дистанционного обучения, а также PWA для доступа к обучающим материалам;
 - Примеры: университеты, школы, онлайн-курсы.
4. Разработчики и IT-специалисты (15%):
 - Заинтересованы в новых технологиях для повышения квалификации и создания инновационных приложений;
 - Примеры: разработчики PWA, frontend и backend специалисты, компании-разработчики.
5. Конечные пользователи (активные интернет-пользователи) (10%):
 - Стремятся к удобству использования, доступу к приложениям оффлайн и повышенной безопасности;
 - Примеры: пользователи социальных сетей, интернет-магазинов, образовательных платформ.



Аналоги

Progressive Web Apps (PWA):

Характеристики: Возможность установки на рабочий стол или мобильное устройство. Поддержка работы оффлайн за счет кеширования данных. Push-уведомления для взаимодействия с пользователем. Обновление в реальном времени при подключении к интернету.

Структурные/программные особенности: Использование Service Workers для работы оффлайн и кэширования. Применение манифеста для настройки значков и внешнего вида приложения.

Сфера применения: Подходит для e-commerce, новостных порталов, соцсетей, а также любых веб-приложений с высоким трафиком и активным взаимодействием с пользователями.

Стоимость разработки: Как правило, ниже, чем у нативных мобильных приложений, так как PWA поддерживается на всех платформах без создания отдельных версий.

Примеры: Twitter Lite: Пример быстрого и легковесного приложения для социальной сети. Starbucks: Поддержка оффлайн-доступа к меню и оформлению заказов.

Примеры: **Twitter Lite, Starbucks.**

Аналоги

WebAssembly:

Характеристики: Байт-код, исполняемый в браузере, что значительно повышает производительность. Позволяет использовать C/C++ и Rust для создания высокопроизводительных веб-приложений.

Структурные/программные особенности: Работает внутри песочницы, обеспечивая безопасность. Поддержка многопоточности и работы с памятью.

Сфера применения: Подходит для приложений, требующих высокой производительности, таких как игры, редакторы графики, системы проектирования (например, САПР).

Стоимость разработки: Может быть выше из-за необходимости разработки на C++/Rust, но часто окупается за счёт улучшенной производительности и возможности переиспользования кода.

Примеры: **Figma, AutoCAD Web.**

Аналоги

WebRTC:

Характеристики: Поддержка потоковой передачи видео, аудио и данных в реальном времени. Обеспечивает шифрование для безопасной передачи данных.

Структурные/программные особенности: Использует протоколы STUN/TURN для обхода NAT. API для организации peer-to-peer соединений.

Сфера применения: Используется для видеозвонков, онлайн-конференций, совместного редактирования, игр и интерактивного обучения.

Стоимость разработки: Как правило, средняя, особенно если использовать готовые библиотеки для внедрения WebRTC.

Примеры: **Zoom, Google Meet.**

Технологии решения проблем

Задачи проекта и примененные технологии

Задачи:

- Обеспечить быструю загрузку и отзывчивость веб-приложения.
- Реализовать оффлайн-доступ и push-уведомления для повышения вовлеченности пользователей.
- Обеспечить поддержку высокой производительности для ресурсоемких задач.
- Создать возможности для безопасного и надежного обмена данными в реальном времени.

Технологии и инструменты:

- Языки программирования: JavaScript, C++ (для WebAssembly).
- Среда разработки: VS Code, WebStorm для фронтенда; используемая платформа — Node.js.
- Библиотеки и фреймворки: React для интерфейса, Workbox для управления кэшем, WebRTC API для обмена данными в реальном времени.
- Аппаратные средства: Серверы с поддержкой высоких нагрузок для тестирования производительности, тестовые устройства для кросс-платформенной проверки (iOS, Android).

Технологии решения проблем

Обоснование выбора технологий и суть решения

Обоснование выбора:

- JavaScript и React: Выбраны за их популярность и возможность создания интерактивных интерфейсов, поддерживающих обновления в реальном времени.
- WebAssembly (C++): Позволяет использовать мощные библиотеки для интенсивных вычислений и графики, что было невозможно на чистом JavaScript.
- WebRTC API: Обеспечивает легкое и безопасное взаимодействие между пользователями через видео и аудио.

Суть решения:

- Создание гибридного веб-приложения, совмещающего преимущества PWA и высокой производительности WebAssembly, с безопасными каналами связи на WebRTC.
- Использование Service Workers для работы оффлайн и push-уведомлений, что повышает лояльность и удобство для пользователей.

Технологии решения проблем

Текущие и целевые параметры, преимущества перед конкурентами

Текущие параметры:

- Время загрузки: 1–2 секунды при хорошей связи, до 5 секунд при слабом сигнале.
- Скорость обработки данных: 10–15 мс при интенсивных вычислениях с помощью WebAssembly.
- Безопасность: TLS-шифрование данных, передаваемых через WebRTC.

Целевые параметры:

- Время загрузки: Сократить до 1 секунды даже при слабой сети.
- Скорость обработки: Снизить задержку до 5 мс.
- Оффлайн-доступ: Обеспечить кэширование основных функций и данных.

Преимущества перед конкурентами:

- Быстрота и отзывчивость: Использование PWA и WebAssembly позволяет достичь более высокой скорости, чем у многих конкурентов.
- Гибкость и поддержка платформ: Отсутствие необходимости разработки отдельных мобильных приложений для разных ОС.
- Безопасность и надежность: WebRTC обеспечивает защиту данных, что делает приложение более привлекательным для компаний, ценящих конфиденциальность.

Продукт проекта

Обеспечить пользователям доступ к высокопроизводительному, удобному и безопасному веб-приложению, которое функционирует как нативное мобильное приложение, но не требует установки из App Store или Google Play.

Ключевые функции: Работа оффлайн, Push-уведомления, Поддержка видеозвонков и передачи данных в реальном времени, Высокая производительность.

Технические параметры:

- Скорость загрузки;
- Производительность (среднее время отклика);
- Оффлайн-доступ;
- Поддержка устройств;
- Видеосвязь и обмен данными.

Качественные характеристики: Кросс-платформенность, Безопасность, Доступность.

Стоимостные характеристики средняя, благодаря использованию единой кодовой базы для всех платформ (экономия на разработке отдельных мобильных приложений). Поддержка и обновление приложения обходится дешевле, так как все изменения вносятся в одно веб-приложение, доступное на всех устройствах.

Коммерциализуемость проекта

Продукт на основе прогрессивных веб-технологий имеет широкий спектр применений и может быть востребован в различных отраслях:

- Электронная коммерция (e-commerce);
- Образование (EdTech);
- Медиа и новостные порталы;
- Корпоративные порталы и бизнес-приложения.

Объем инвестиций:

- Разработка и тестирование: \$50,000–100,000;
- Маркетинг и продвижение: \$30,000–50,000;
- Поддержка и обновления: \$20,000;
- Итого стартовых инвестиций: \$100,000–150,000.

Формы получения инвестиций:

- Венчурное инвестирование;
- Гранты и государственные субсидии;
- Краудфандинг.

Кому можно продать проект:

- Средний и крупный бизнес в сфере e-commerce;
- Образовательные и социальные учреждения;
- Технологические компании.

Свободный раздел

По итогу работы проект показал высокий потенциал применения прогрессивных веб-технологий в различных сферах, подтвердив их эффективность для создания легковесных и производительных приложений. Технологии PWA, WebAssembly и WebRTC доказали свою способность обеспечить стабильность и гибкость, которые важны для современного бизнеса и пользователей.

Планы на дальнейшее развитие:

- Расширение функций и интеграция с AI-технологиями для адаптации и персонализации контента;
- Повышение уровня безопасности за счет использования биометрической аутентификации и расширенного шифрования данных.

Личный вклад и развитие навыков:

- Работа над проектом позволила усовершенствовать навыки работы с современными веб-технологиями и улучшить опыт в построении архитектуры высокопроизводительных приложений;
- Получен ценный опыт взаимодействия с процессами разработки.

Дополнительные возможности области расширения:

- Интеграция с облачными сервисами для дополнительного хранилища данных и улучшения масштабируемости;
- Внедрение функции аналитики для отслеживания активности пользователей и улучшения пользовательского опыта.

Заключение

Презентация спроектирована \ разработана студентом
1 курса, Бакалавр, Ларионовым Михаил Юрьевичем.

Контакты для связи:

Телефон: +7 (932) 113-19-16;

TG: @mk_larionov;

Почта: larionov_mu@mail.ru \ larionov404.mu@gmail.com \



 Проектный офис
ИКТИБ

 Институт
Компьютерных
Технологий и
Информационной
Безопасности

