

ЛИТЕРАТУРА

1. Prismlibrarydocumentation.[Электронный ресурс]. – Электронные данные. – Режим доступа:
<http://prismlibrary.github.io/docs/index.html>
2. Руководство разработчика Prism. [Электронный ресурс]. – Электронные данные. – Режим доступа:
<https://habr.com/ru/post/176851/>

УДК004.042

Студ. Д. С. Кунец

Науч. рук. ст. преп. А. С. Наркевич
(кафедра программной инженерии, БГТУ)

ПРОГРЕССИВНЫЕ ВЕБ-ПРИЛОЖЕНИЯ

Цель работы – разработать веб-приложение для создания и прохождения обучающих курсов на языке программирования Scratch.

Решаемые задачи данного приложения:

- составление, изменение и удаление уроков;
- возможность создавать класс и приглашать в него студентов;
- возможность ведения статистики по урокам;
- возможность поиска урока;
- поддержка режима без доступа к сети интернет, который основан на технологии PWA.

Термин «*Progressive Web App*» введен Алексом Расселом и Франсесом Берриманом. По словам Алекса, прогрессивные веб-приложения — это всего лишь веб-сайты, которые принимали правильные витамины. То, что лежит в основе *PWA*, нельзя назвать новым фреймворком или новой технологией. Это, фактически, набор передовых методов разработки, которые позволяют сделать так, чтобы поведение веб-приложения оказалось бы очень похожим на поведение классических настольных или мобильных приложений.

PWA дают пользователю новые удобные возможности посредством прогрессивных улучшений приложений. В целом, это означает, что *PWA*, на мобильных устройствах разных поколений, будут работать примерно одинаково. Конечно, некоторые возможности обычных телефонных приложений могут быть им недоступны, но такие приложения обычно работают на разных устройствах так, как они и должны на них работать.

PWA — это веб приложение, созданное с использованием определенных технологий для достижения заданных целевых показателей, которые представлены на рисунке 1.

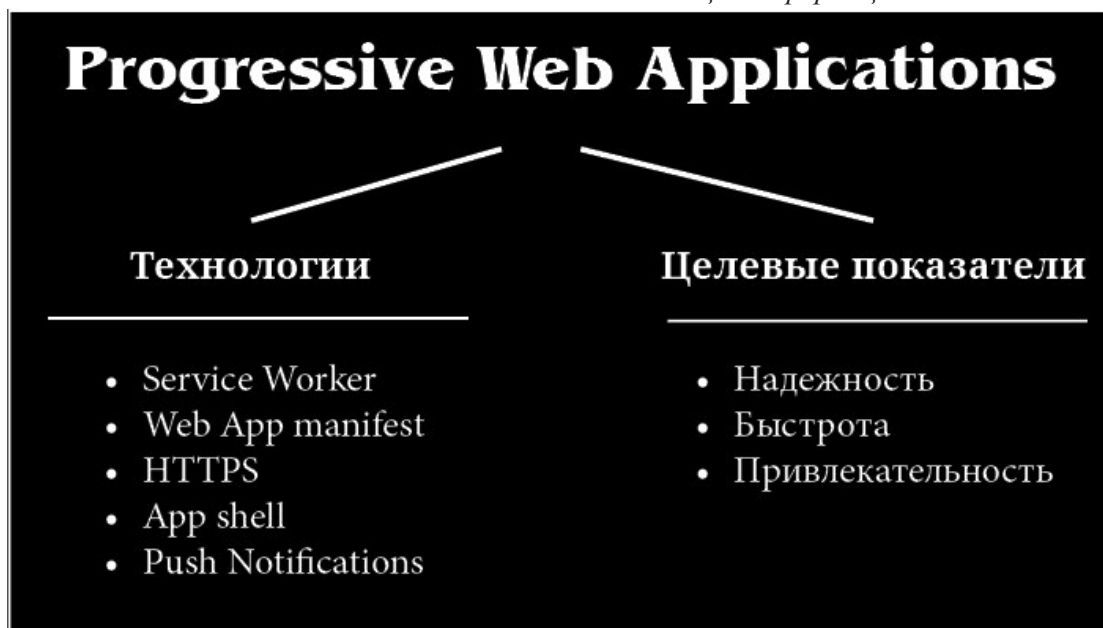


Рисунок 1 – Целевые показатели PWA

Надежность (*Reliable*) — приложение загружается и показывается сразу же, вне зависимости от статуса и качества сетевого соединения.

Быстрота (*Fast*) — взаимодействие данными по сети происходит быстро, *UI* плавный и отзывчивый.

Привлекательность (*Engaging*) — делает для пользователя опыт работы с приложением комфортным и приятным.

Service Worker – это воркеры, управляемые событиями, которые выполняются в фоне приложения и действуют в роли посредников между приложением и сетевыми ресурсами. Сервис-воркеры представляют собой *JavaScript*-программы, которые перехватывают обработку определённых событий и выполняют некие задачи. Они могут перехватывать сетевые запросы и кэшировать информацию для приложения. Их можно использовать для загрузки данных, которые планируется использовать для обеспечения работы приложения в оффлайн-режиме. Из хранилищ у *Service Worker*'а есть доступ к *Cache Storage* для *web* ресурсов, и *IndexedDB* для данных. Но, самое главное, полная свобода для реализации бизнес-логики. Можно, например, принять запрос от браузера, проверить состояние сети, взять данные из хранилища, произвести с ними операции и вернуть некий результат обратно в браузер — который будет думать, что ответ ему пришел от сервера. Схема работы PWA-приложения представлена на рисунке 2.

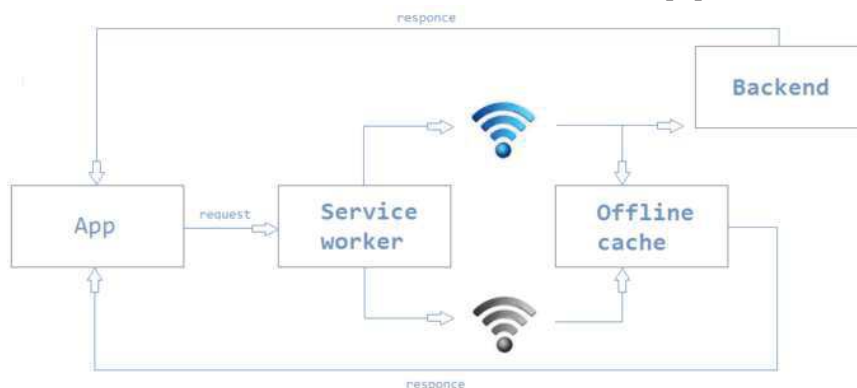


Рисунок 2—Схема работы PWA приложения

Следующие алгоритмы кэширования могут быть использованы:

- Cache with fallback to Network (кэш с резервом для сети) - используйте, если вы создаете автономное первое приложение. Если ответ уже находится в кэше, он будет передан пользователю, и онлайн-запрос никогда не будет выполнен. Если ответ еще не кэширован, приложение попытается загрузить его онлайн и затем поместить в кэш. Этот подход следует использовать для контента, который изменяется очень редко или не изменяется вообще.

- Network with fallback to Cache (сеть с откатом к кэшу) - это подход, при котором онлайн-пользователи всегда получают актуальную онлайн-версию, а автономные пользователи получают кэшированную версию. Используйте его для ресурсов, которые часто обновляются.

- Cache and Network race (кэш и сетевая гонка) - это когда вы ищете ответ в кеше, одновременно запрашивая онлайн-контент. Сначала вы показываете кэшированный ответ пользователя, а затем заменяете его новым контентом сразу после его появления или добавляете новый контент поверх кэшированных страниц, таких как Facebook и Twitter.

- HTTPS требуется, чтобы все передавалось по защищенному соединению. Критичным является тот факт, чтобы на сайте не было ссылок на незащищенные ресурсы.

Application shell— оболочка создаваемого приложения. При разработке PWA сайта «с нуля» для этого могут быть разные стратегии, но на готовом сайте удобно использовать деление на его статическую и динамическую части.

Web App manifest— JSON файл, декларативно определяющий для браузера название приложения, иконку, как будет выглядеть PWA (*fullscreen*, *standalone* и др.), цветовую тему, ориентацию экрана (книжная или альбомная) и некоторые другие параметры. Позволяет

«установить» *PWA* как отдельное приложение на домашний экран смартфона.

Push Notifications— самая популярная и самая злоупотребляющая технология *PWA*. Позволяет реализовать всплывающие уведомления на сайте.

Данное приложение позволяет преподавателю самостоятельно создавать уроки, при этом требуется только знание языка программирования *Scratch*. Перед студентами будет поставлена явная цель преподавателя, которую необходимо будет выполнить, используя уже имеющиеся знания по языку программирования *Scratch*. По результатам выполнения заданий преподаватель имеет возможность посмотреть статистику по урокам в группе и сделать некоторые выводы по тому, как студенты усваивают материал. Возможность использования приложения без доступа в интернет позволит активно использовать приложение в учреждениях образования.

ЛИТЕРАТУРА

1. *Progressive Web Apps*[Электронный ресурс] / PCmag. – Режим доступа: <https://developers.google.com/web/progressive-web-apps/>. – Дата доступа 10.04.2019.

УДК 004.738.1+004.85

Маг. Е. С. Помазай

Науч. рук. доц., канд. техн. наук А. И. Парамонов
(кафедра программной инженерии, БГТУ)

ПРОГНОЗИРОВАНИЕ РЕЗУЛЬТАТОВ ТЕННИСНЫХ МАТЧЕЙ С ПРИМЕНЕНИЕМ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ

В последние годы машинное обучение (ML – machine learning) превратилось, в большой бизнес – фирмы используют его, чтобы заработать денег, прикладные исследования бурно развиваются как в индустриальной, так и в академической среде. Прикладное машинное обучение совмещает в себе равные доли математических принципов и полученных эмпирическим путем приемов, – другими словами, это настоящее искусство [1]. С ростом популярности машинного обучения растут и его области применения. Помимо медицины, бизнеса, производства специалисты в области обработки данных, применяют алгоритмы машинного обучения при подборе рекомендаций, определении настроений, разработке искусственного интеллекта в компьютерных играх и т.д. Однако остаются сферы, в которых применение машинного обучения слабо изучено и целесообразность применения