|  |  |
| --- | --- |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  Институт математики и информационных систем  Факультет автоматики и вычислительной техники  Кафедра систем автоматизации управления | |
| Отчет по практической работе №2  по дисциплине  Инфокоммуникационные системы и сети  Вариант 19 | |
|  | Выполнили: |
|  | студенты гр. ИТб-3301 |
|  | Денисова А.А.  Прокопьев М.И. |
|  | Проверил: |
|  | Земцов М.А. |
| Киров  2020 | |

**Цель работы**: разработка первого PWA приложения на стеке технологий NodeJS.

Ход выполнения разработки:

Сначала были установлены и настроены компоненты, необходимые для дальнейших действий:

- Node.js;

- Бразуер Google Chrome;

- Библиотека Vue.js;

- Workbox;

- Firebase;

- Репозиторий на Github.

Для разработки PWA-приложения была создана директория, в которой был создан проект минимально рабочего приложения isn-app (рисунок 1).

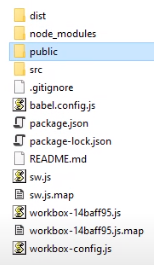


Рисунок 1 – Структуры рабочей директории

Была протестирована работа базового приложения путем выполнения команды: npm run serve. Далее была изучена работа клиентской части данного приложения (рисунок 2). Также был установлен декоратор Material Design для более аккуратной поддержки устройств с разными разрешениями (тема Material Degign используется Google в мобильных приложениях, поэтому такой веб-сайт будет стилизован под мобильные приложения).

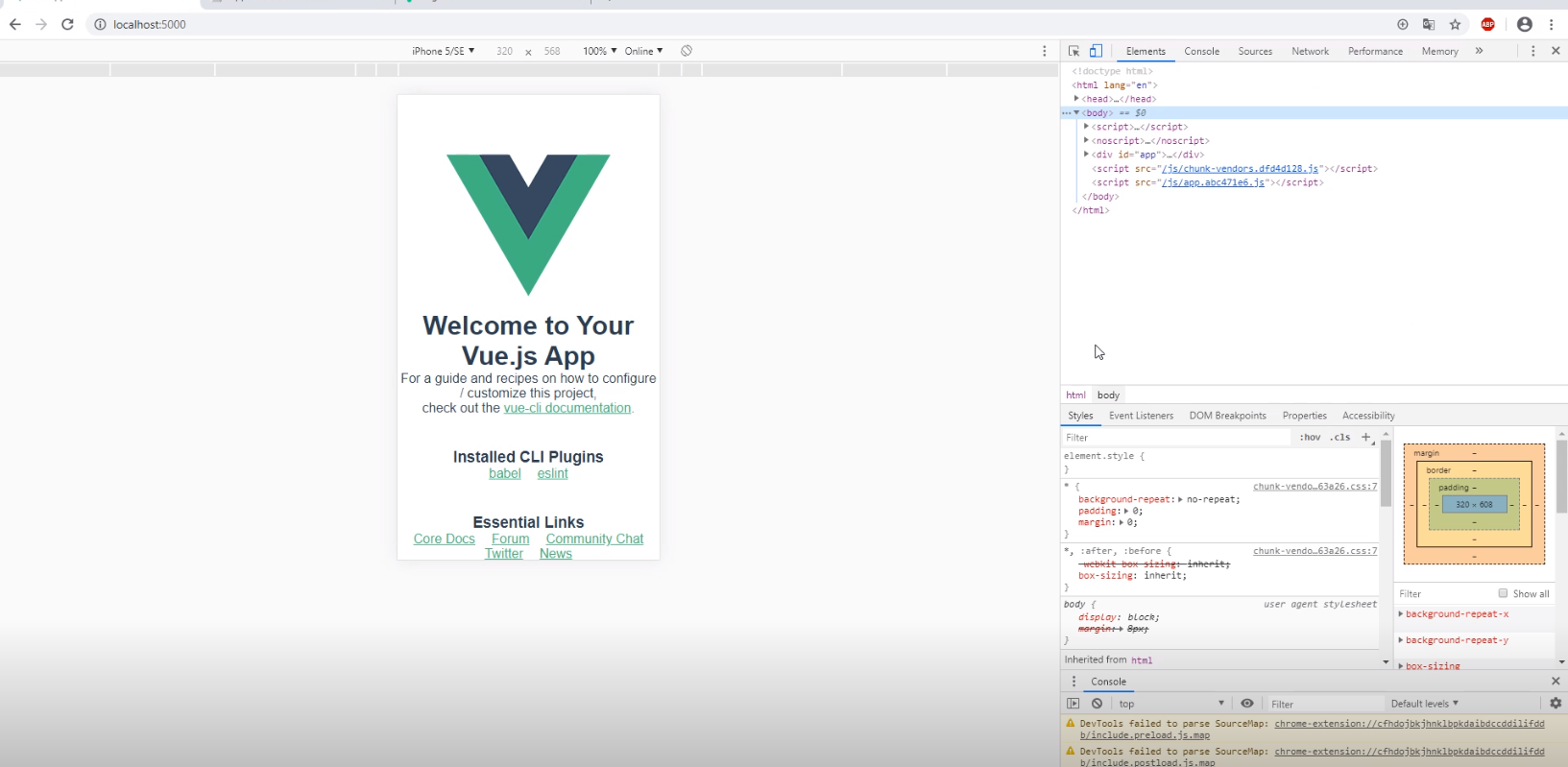


Рисунок 2 – Клиентская часть приложения

Был сгенерирован манифест приложения с помощью сервиса <https://app-manifest.firebaseapp.com/> и помещен в директорию приложения. Содержимое сгенерированного манифеста приложения изображено на рисунке 3. Также в index.html была добавлена строка: <link rel="manifest" href="/manifest.json">.



Рисунок 3 – Манифест приложения

Затем был создан ServiceWorker - сценарий, который выполняется в фоновом режиме, независимо от того, открыта ли веб-страница. Для этого была использована библиотека Workbox и код, представленный на рисунке 4.

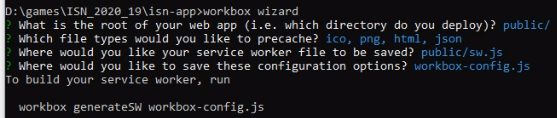


Рисунок 4 – Создание ServiceWorker

Для подключения сервисного работника к приложению, в index.html был добавлен скрипт:

<script> if ('serviceWorker' in navigator) {

window.addEventListener('load',function() {

navigator.serviceWorker.register('/sw.js'); }); }

</script>

Затем проект был заново собран и запущен. DevTools показывает, что Service Worker успешно активирован и запущен. Результат отображен на рисунке 5.

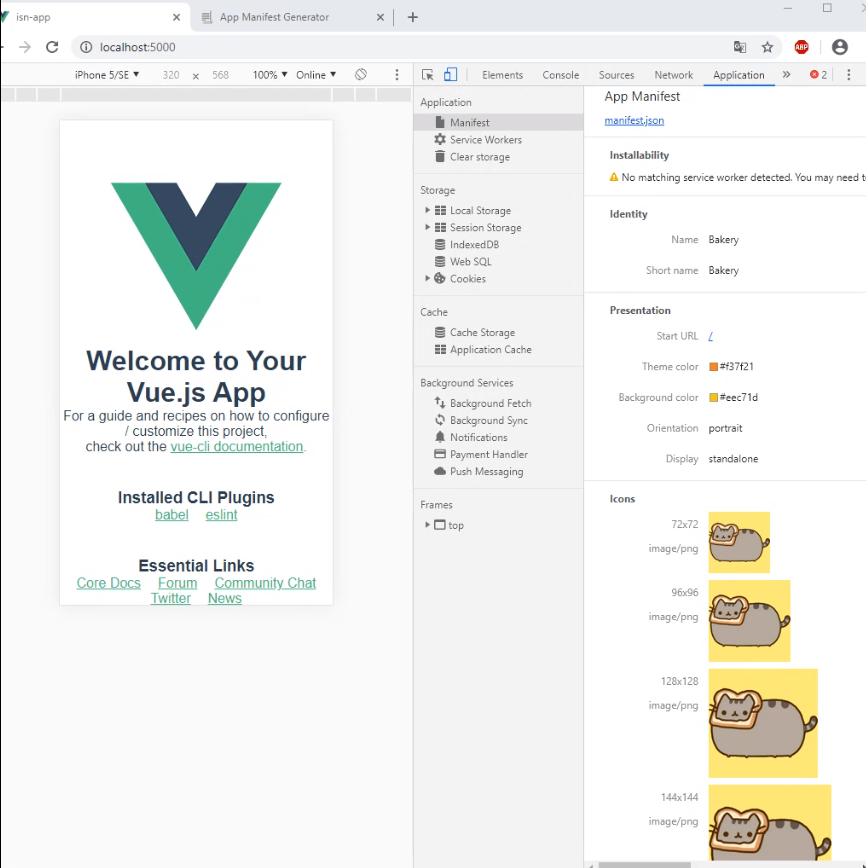


Рисунок 5 – Проверка работы ServiceWorker

Затем были спроектированы экраны мобильного приложения по заданной тематике согласно варианту (19 – Хлебопекарня). Проектирование экранов было реализовано в приложении Figma. Результат спроектированных экранов продемонстрирован на рисунке 6.

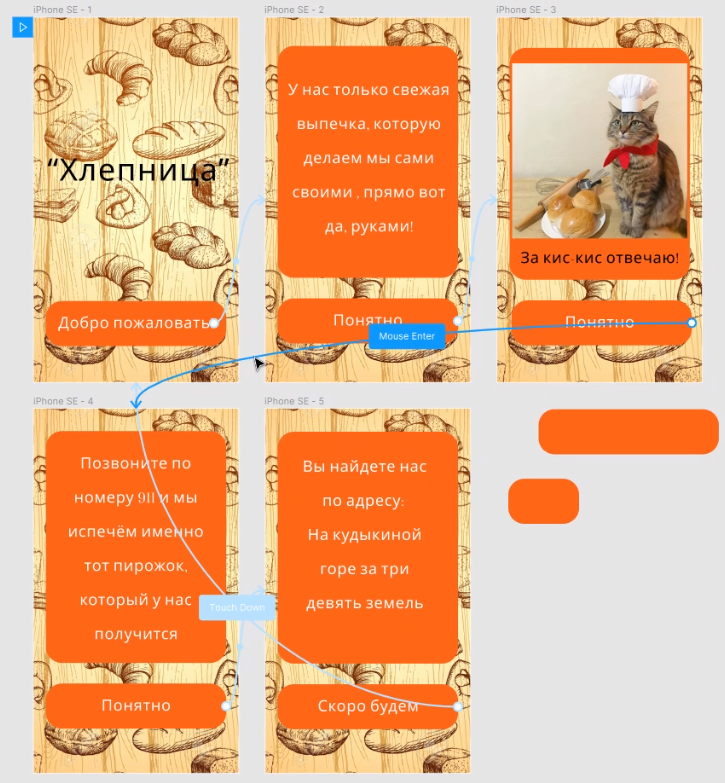


Рисунок 6 – Экраны приложения

**Вывод**: в ходе лабораторной работы были получены базовые навыки создания простейшего приложения с помощью фреймвора Vue на стеке технологий Node.js, произведено ознакомление с такими компонентами, как Workbox и Firebase. Запуск самого приложения производится в браузере Google Chrome, тестирование – средствами DevTools. Также были получены навыки работы в среде макетирования Figma, с помощью которой был построен макет мобильного приложения для хлебопекарни, состоящий из 5 экранов.