

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Институт математики и информационных систем
Факультет автоматики и вычислительной техники

Построение PWA приложений
Отчет по лабораторной работе №3 дисциплины
«Глобальные сети»

Выполнил студент группы ИТ-4301-01-20 _____ /Михеев А.О./
Проверил преподаватель _____ /Земцов М. А./

Киров 2020

1. Цель работы

Исследование технологий создания Progressive Web Application PWA приложения на стеке технологий NodeJS.

2. Задание

Разработать простое PWA приложение

3. Ход работы

3.1 Установка рабочего окружения

Для начала работы необходимо установить рабочее окружение.

С помощью node version manager был установлен Node.js и пакетный менеджер npm. Дополнительно был установлен пакетный менеджер yarn

При помощи пакетного менеджера yarn был установлен Vue CLI - командная оболочка которая позволяет быстро и удобно создавать базовые Vue-приложения

Средствами npm были установлены утилиты Workbox и Firebase

3.2 Разработка PWA приложения

С помощью Vue CLI командой `vue create isn-app` создаем минимальный проект рабочего приложения vue.

Командой `yarn serve` запускаем сервер разработки и переходим в браузере по адресу <http://localhost:8080>

На Рис 1 представлен скриншот страницы созданного vue приложения в режиме отображения на мобильном устройстве. Особых проблем не обнаружено.

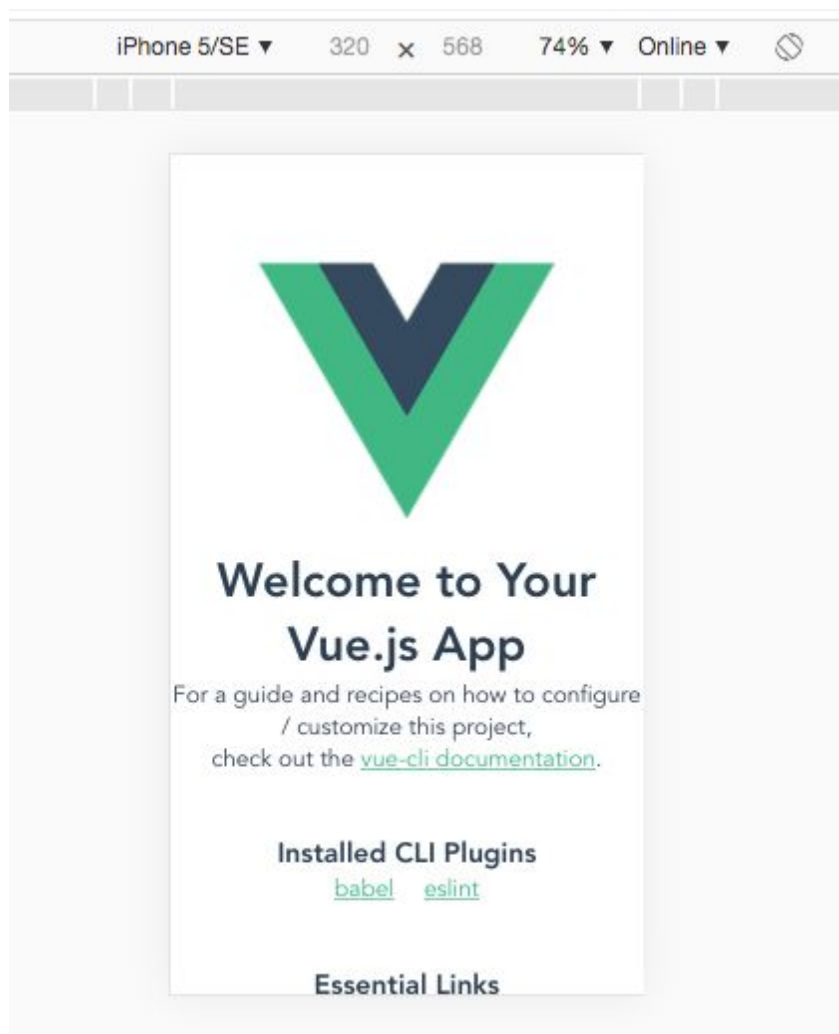


Рис 1 - Страница приложения

Для более аккуратной поддержки устройств с разными разрешениями установим поддержку декоратора Material Design, для этого установим компонент vuetify:

```
yarn add vuetify
```

И подключим модуль в файле приложения main.js:

```

import Vue from 'vue'
import App from './App.vue'
import Vuetify from 'vuetify'

import 'vuetify/dist/vuetify.css'

Vue.config.productionTip = false

Vue.use(Vuetify)

new Vue({
  render: h => h(App),
}).$mount('#app')

```

Рис 2 - содержимое файла main.js

Далее с помощью <https://app-manifest.firebaseapp.com/> сгенерируем файл-манифест приложения и подключим его в файле index.html

После подключения файла-манифеста удалось установить сайт как приложение на компьютер:

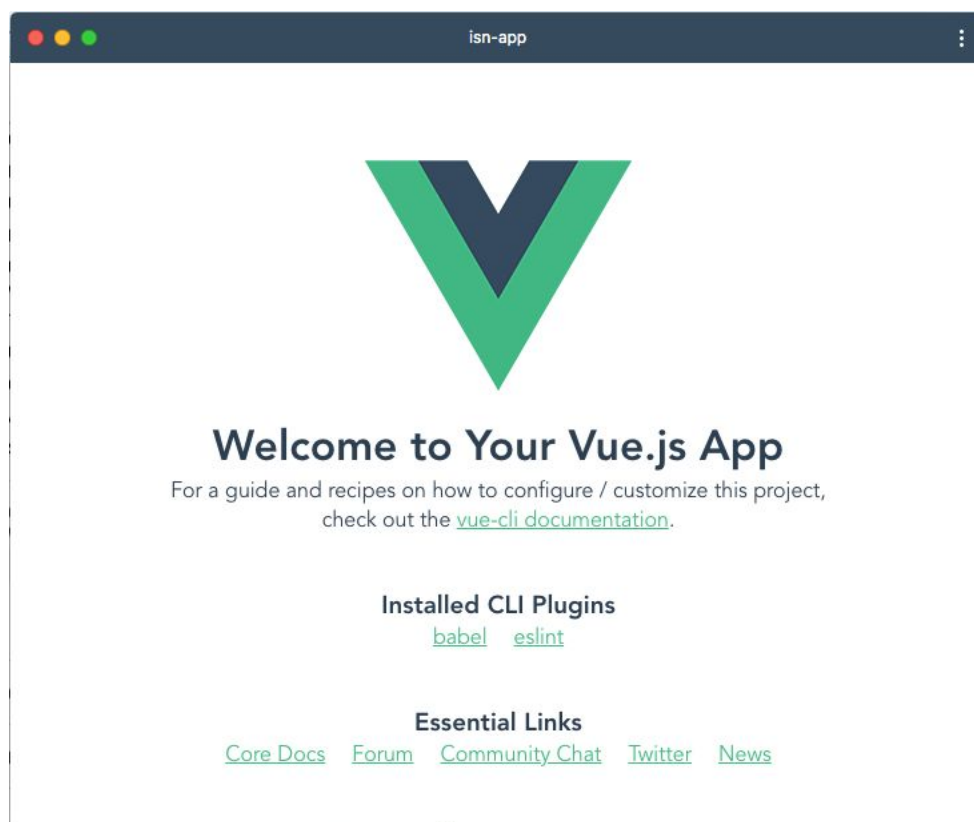


Рис 3 - окно приложения на компьютере

Далее запустим мастер генерации Service Worker в корневой директории проекта командой:

```
workbox wizard
```

Для подключения к приложению добавим код в файл public/index.html:

```
<script>
if ('serviceWorker' in navigator) {
  window.addEventListener('load', function () {
    navigator.serviceWorker.register('/sw.js')
  });
}
</script>
```

Затем скомпилируем статические файлы, перекомпилируем ServiceWorker и запустим сервер разработки следующими командами:

```
yarn build
```

```
workbox generateSW workbox-config.js
```

```
serve -s dist
```

На Рис 4 можно увидеть что ServiceWorker работает

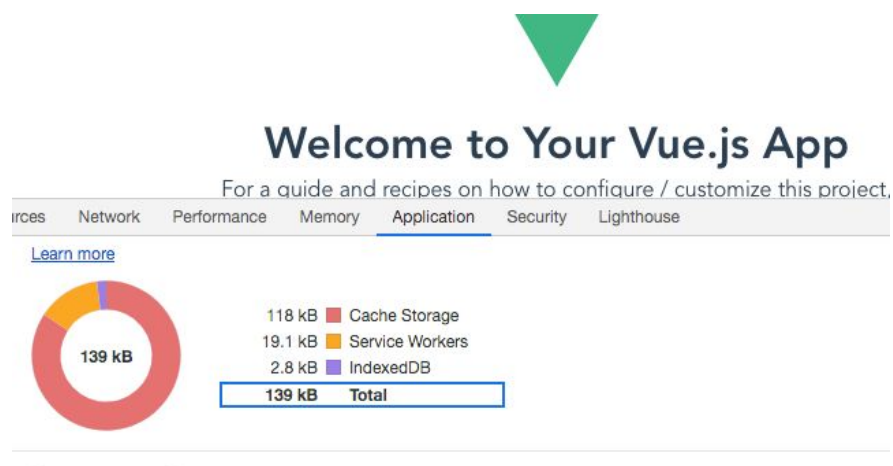


Рис 4 - результат работы ServiceWorker

4. Вывод

В результате выполнения лабораторной работы были получены теоретические сведения о построении PWA приложений, а так же навыки работы с Workbox, Service Worker декоратором Material Design, Node.js и библиотекой Vue.js.