

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем

Факультет автоматики и вычислительной техники

Кафедра систем автоматизации управления

Построение PWA приложений
Отчет по лабораторной работе №8
по дисциплине
«Глобальные сети»

Выполнил студент группы ИТ-4301-01-20 _____ /Лубягин А.К./

Проверил преподаватель _____ /Земцов М. А./

Киров 2020

1 Цель работы

Исследование технологий создания Progressive Web Application (PWA) приложения на стеке технологий Node.JS.

2 Ход работы

Установим (рисунок 1) и настроим сервер запуска приложений Node.js (рисунок 2).

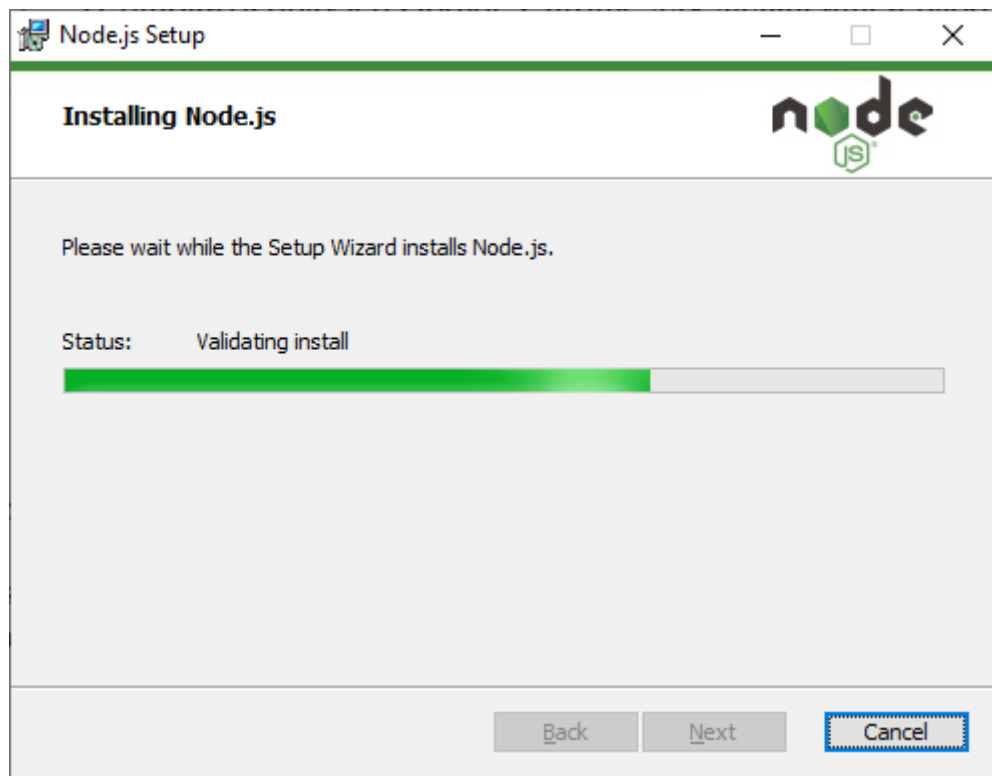


Рисунок 1 – Установка Node.js

```
C:\Users\Alex\ins-app>node -v  
v12.17.0  
  
C:\Users\Alex\ins-app>_
```

Рисунок 2 – Проверка версии Node.js

Для работы с Vue установим модуль Vue CLI (рисунок 3).

```
C:\Users\Alex\ins-app>vue -V
@vue/cli 4.4.1
C:\Users\Alex\ins-app>_
```

Рисунок 3 – Результат установки модуля Vue CLI

Для возможности создания PWA установим Workbox (рисунок 4).

```
C:\Users\Alex\ins-app>workbox --version
5.1.3
C:\Users\Alex\ins-app>_
```

Рисунок 4 – Результат установки Workbox

Для автоматизации задач разработки бэкенда установим Firebase CLI (рисунок 5).

```
C:\Users\Alex\ins-app>firebase --version
8.4.1
C:\Users\Alex\ins-app>_
```

Рисунок 5 – Результат установки Firebase CLI

Создадим git-репозиторий ISN_2020_8.

Далее перейдем непосредственно к разработке PWA приложения. Для этого создадим директорию с проектом. С помощью консоли терминала создадим приложения Vue командой «Vue create ins-app». Результат создания проекта представлен на рисунке 6.

images	02.06.2020 10:04
node_modules	02.06.2020 9:34
public	02.06.2020 12:27
src	02.06.2020 9:23
.gitignore	02.06.2020 9:23
babel.config.js	02.06.2020 9:23
package.json	02.06.2020 9:28
package-lock.json	02.06.2020 9:28
README.md	02.06.2020 9:24
workbox-config.js	02.06.2020 11:28

Рисунок 6 – Папка проекта приложения Vue

Запустим приложение для проверки с помощью команды «npm run serve» и откроем в браузере ссылку «localhost:8080» Результат представлен на рисунке 7.

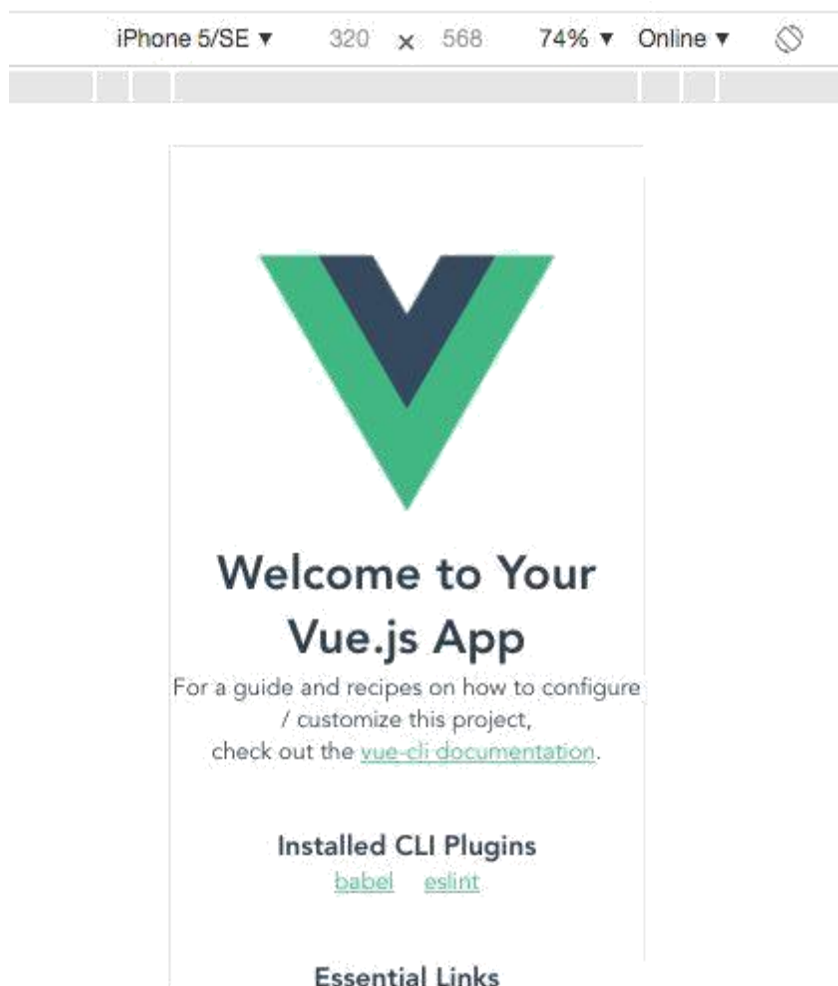


Рисунок 7 – Запуск приложения Vue

Для более аккуратной поддержки устройств с разными разрешениями установим поддержку декоратора Material Design, для этого установим компонент vuetify и подключим модуль в файле приложения main.js. Листинг на рисунке 7.

```
1 import Vue from 'vue'
2 import App from './App.vue'
3 import Vuetify from 'vuetify'
4
5 import 'vuetify/dist/vuetify.min.css'
6
7 Vue.config.productionTip = false
8 Vue.use(Vuetify)
9
10 new Vue({
11   render: h => h(App),
12 }).$mount('#app')
```

Рисунок 7 – Листинг main.js

Следующим шагом сгенерируем манифест приложения, используя сервис: <https://app-manifest.firebaseapp.com/> см. рисунок 8.

Web App Manifest Generator

The Web App Manifest is a JSON document that provides application metadata for Progressive Web Apps. Use the form below to generate the JSON file and optionally upload an app icon.

App Name INSAP	Short Name INSAP
Theme Color #2196f3	Background Color #2196f3
Display Mode Standalone	Orientation Portrait
Application Scope /	Start URL /

manifest.json

```
{
  "name": "INSAP",
  "short_name": "INSAP",
  "theme_color": "#2196f3",
  "background_color": "#2196f3",
  "display": "standalone",
  "orientation": "portrait",
  "scope": "/",
  "start_url": "/"
}
```

Generate Icons

The Web App Manifest allows for specifying icons of varying sizes. Upload a 512x512 sizes.

ICON image (5).png

GENERATE .ZIP

Рисунок 7 – Генерация манифеста

Подключим Manifest.js в приложении и проверим его отображение в Google dev, см. рисунок 8.

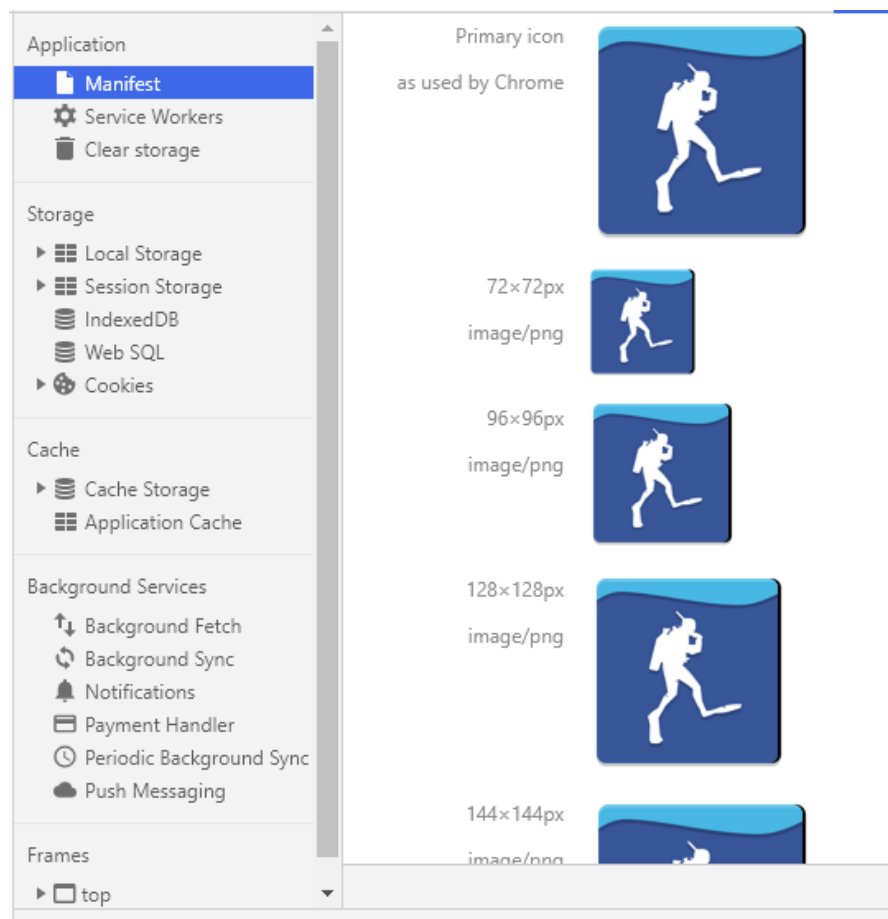


Рисунок 8 – Проверка Manifest.js

Для создания нашего сервисного работника мы используем библиотеку под названием Workbox, который является преемником sw-toolbox и sw-precache. Проверим работу Workbox на рисунке 9.

```
C:\Users\Alex\ins-app>workbox --version
5.1.3

C:\Users\Alex\ins-app>
```

Рисунок 9 – Проверка Workbox

Запустим мастер генерации ServiceWorker в корневой директории проекта и подключим к приложению. В результате работы проверьте работу ServiceWorker на рисунке 10.

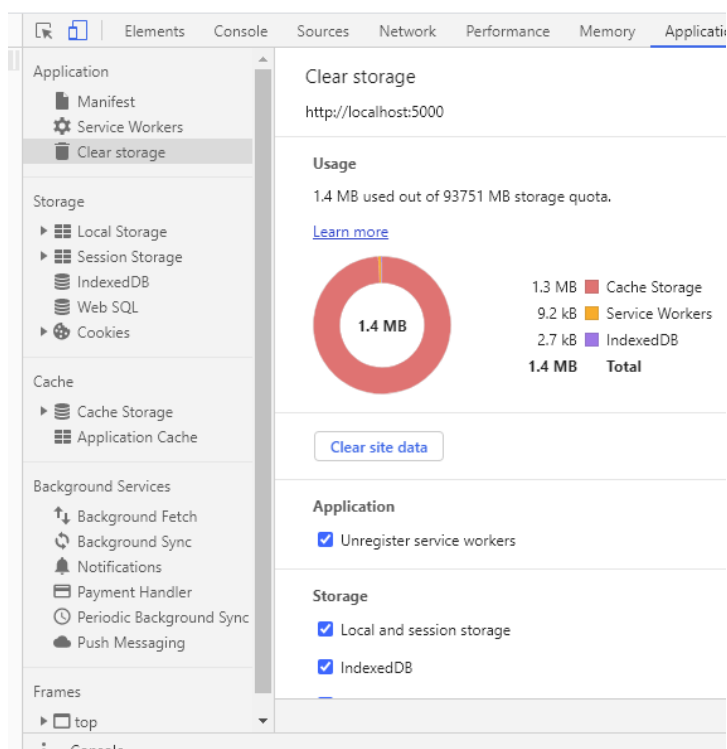


Рисунок 10 – Проверка ServiceWorker

Проведем аудит. В результате чего были выявлены недостатки с производительностью (рисунок 10).

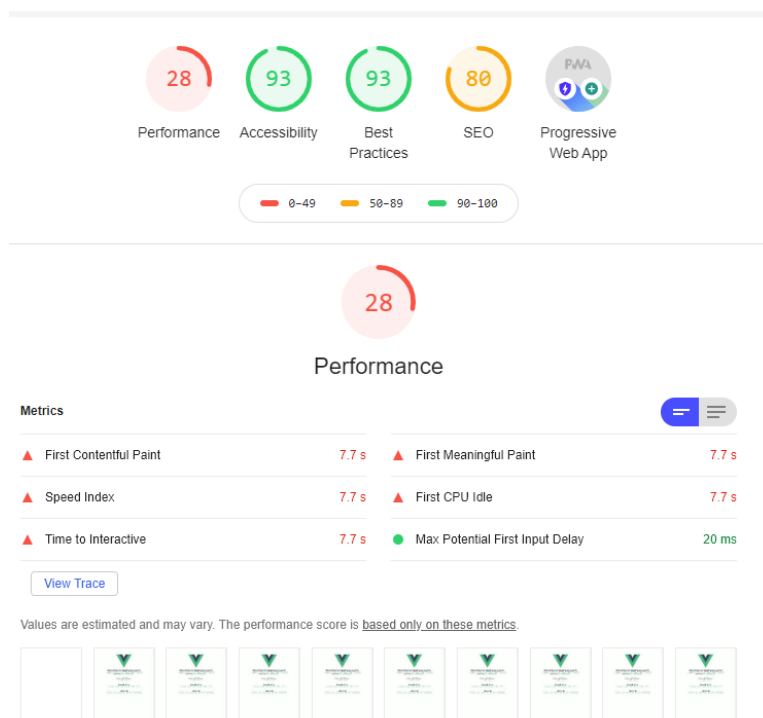


Рисунок 9 – Аудит

Репозиторий с приложением Vue находится по ссылке:

https://github.com/Amputator/ISN_2020_8

3 Вывод

В результате выполнения лабораторной работы были получены теоретические сведения о построении PWA приложений, также были изучены навыки работы с web-manifest, Workbox, Service Worker, Firebase и декоратором Material Design, Node.js и библиотекой Vue.js.