

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

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 №3
по дисциплине
«Глобальные сети»

Выполнил:

студент группы ИТб-4301-01-20

Максимов А. А.

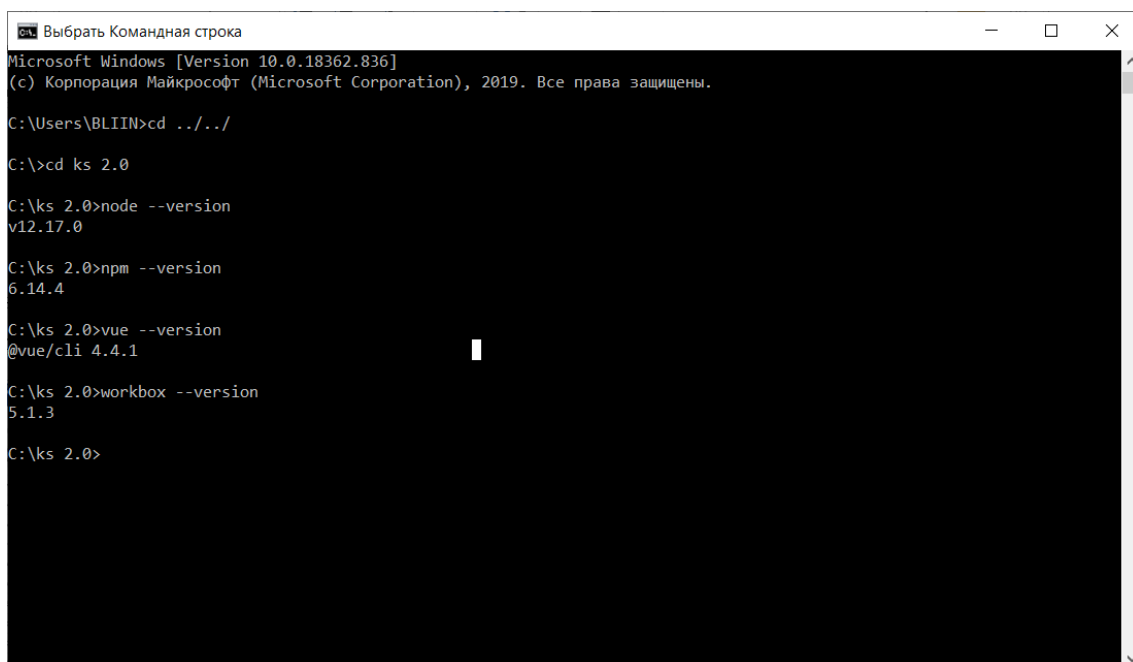
Киров 2020

1 Цель работы

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

2 Ход работы

Установим и настроим сервер запуска приложений Node.js, а также необходимый набор библиотек и фреймворков. Проверим их работоспособность (рисунок 1).



```
Выбрать Командная строка
Microsoft Windows [Version 10.0.18362.836]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\BLIIN>cd ../../

C:\>cd ks 2.0

C:\ks 2.0>node --version
v12.17.0

C:\ks 2.0>npm --version
6.14.4

C:\ks 2.0>vue --version
@vue/cli 4.4.1

C:\ks 2.0>workbox --version
5.1.3

C:\ks 2.0>
```

Рисунок 1 – Проверка установленных библиотек

Сформируем готовый минимальный проект рабочего приложения vue, для ознакомления Результат создания проекта представлен на рисунке 2.

dist	02.06.2020 12:43	Папка с файлами	
node_modules	02.06.2020 12:13	Папка с файлами	
public	02.06.2020 12:41	Папка с файлами	
src	02.06.2020 12:08	Папка с файлами	
.gitignore	02.06.2020 12:08	Текстовый докум...	1 КБ
babel.config.js	02.06.2020 12:08	файл JavaScript	1 КБ
package.json	02.06.2020 12:13	Файл "JSON"	1 КБ
package-lock.json	02.06.2020 12:13	Файл "JSON"	443 КБ
README.md	02.06.2020 12:08	Файл "MD"	1 КБ
workbox-config.js	02.06.2020 12:39	файл JavaScript	1 КБ

Рисунок 2 – Сформированный проект приложения Vue

Проверим работоспособность приложения. Результат проверки представлен на рисунке 3.

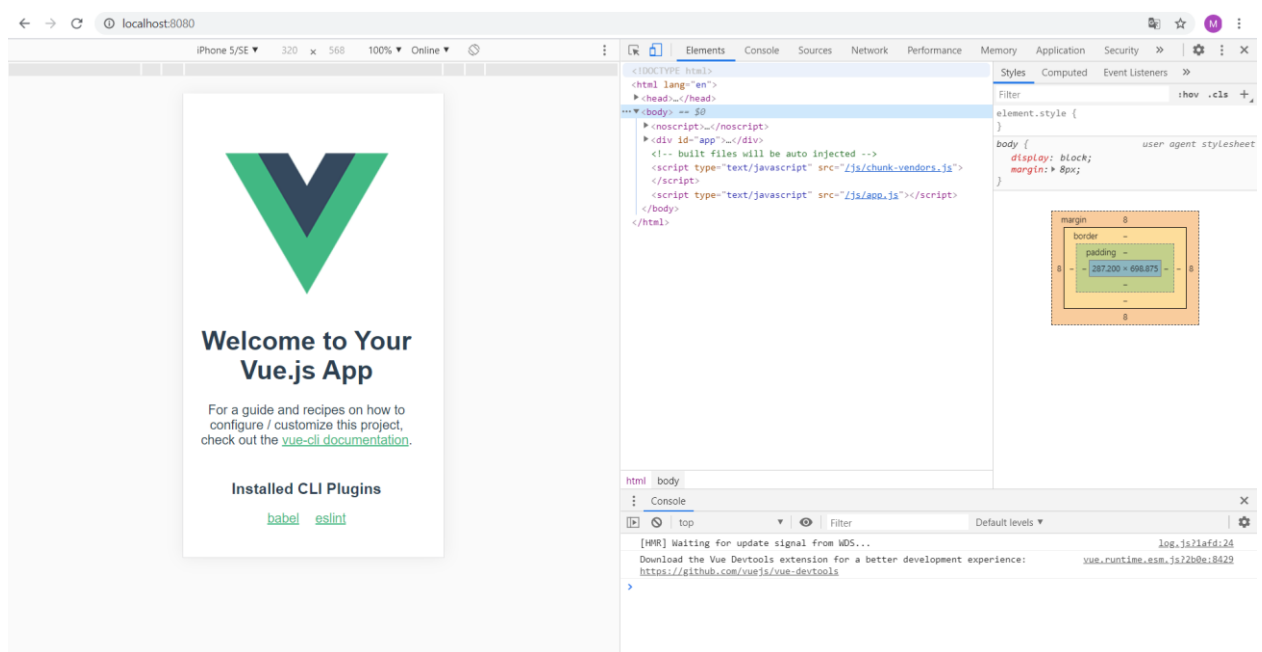


Рисунок 3 – Проверка работоспособности приложения Vue

Для поддержки устройств с разными разрешениями, установим поддержку декоратора Material Design, путем добавления компонента в vue. Далее подключим модуль в файле приложения main.js (рисунок 4).

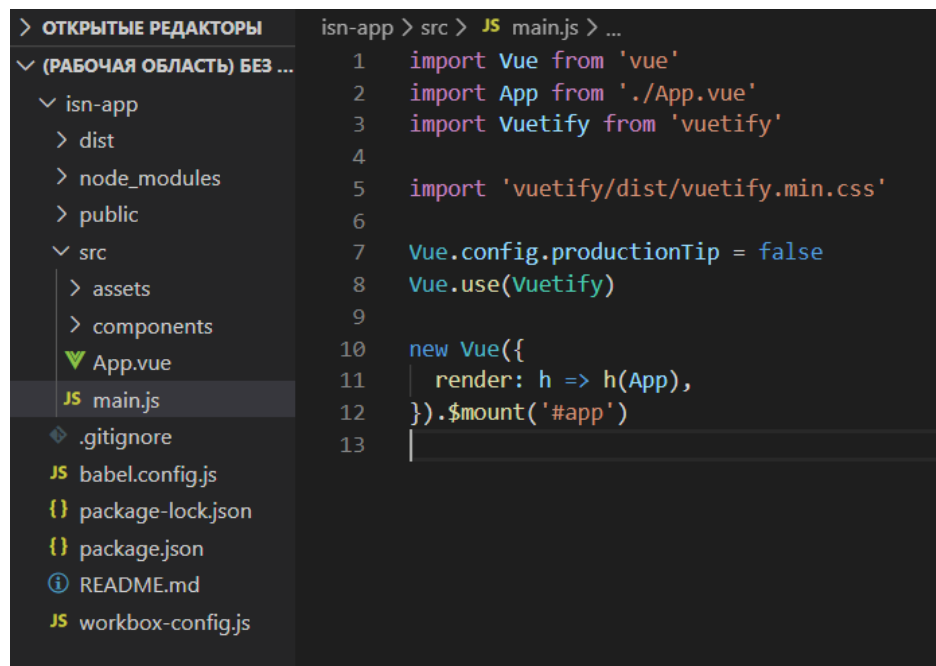


Рисунок 4 – подключение модуля Vuetify

Сгенерируем манифест приложения с помощью сервиса Manifest. Полученные ресурсы, скачанные в сжатом файле, распакуем и вставим в папку проекта. После этого добавим ссылку на манифест в index.html (рисунок 5, 6).

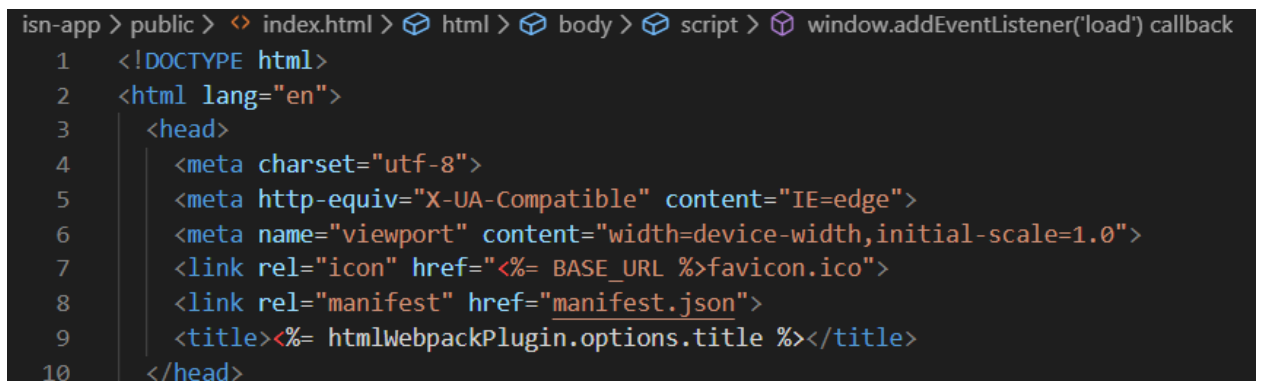


Рисунок 5 – Добавление ссылки в index.html

[manifest.json](#)

Installability

⚠ No matching service worker detected. You may need to reload the page, or check that the scope of the service worker for the current pag

Identity

Name ISNApp

Short name ISNApp

Presentation

Start URL [/](#)

Theme color ■ #36495d

Background color ■ #36495d

Orientation portrait

Display standalone


Icons

☐ Show only the minimum safe area for maskable icons


Need help? Read our [documentation on maskable icons](#).

Primary icon

as used by Chrome



72×72px
image/png



96×96px
image/png




Рисунок 6 – Манифест в google developer tools

Далее создадим Service Worker, используя библиотеку Workbox. Сгенерируем файл sw.js в директории public/ (рисунок 7).

images	02.06.2020 12:21	Папка с файлами	
favicon.ico	02.06.2020 12:08	Значок	5 КБ
index.html	02.06.2020 12:40	Chrome HTML Do...	1 КБ
manifest.json	02.06.2020 12:20	Файл "JSON"	2 КБ
sw.js	02.06.2020 12:44	файл JavaScript	2 КБ
sw.js.map	02.06.2020 12:44	Файл "MAP"	3 КБ
workbox-21eb2654.js	02.06.2020 12:44	файл JavaScript	8 КБ
workbox-21eb2654.js.map	02.06.2020 12:44	Файл "MAP"	72 КБ

Рисунок 7 – Созданный файл sw.js в директории public

Проведем повторный аудит полученного приложения (рисунок 8, 9).

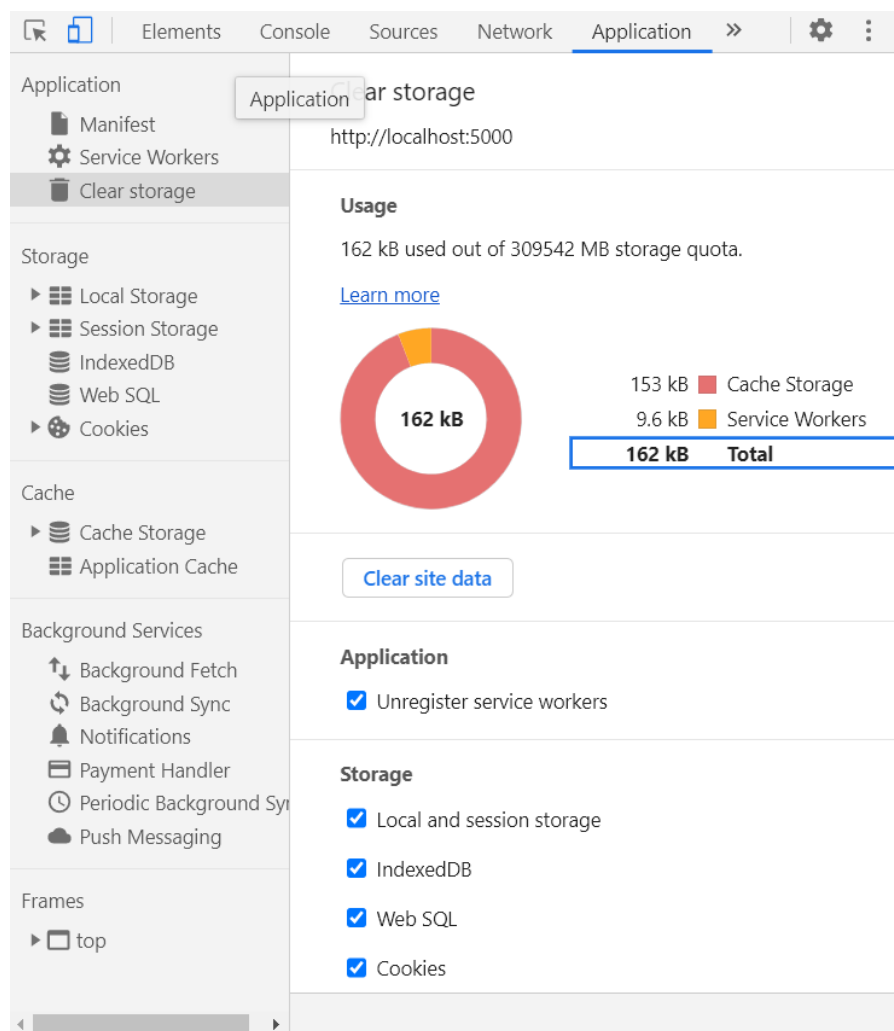


Рисунок 8 –Аудит полученного приложения Vue

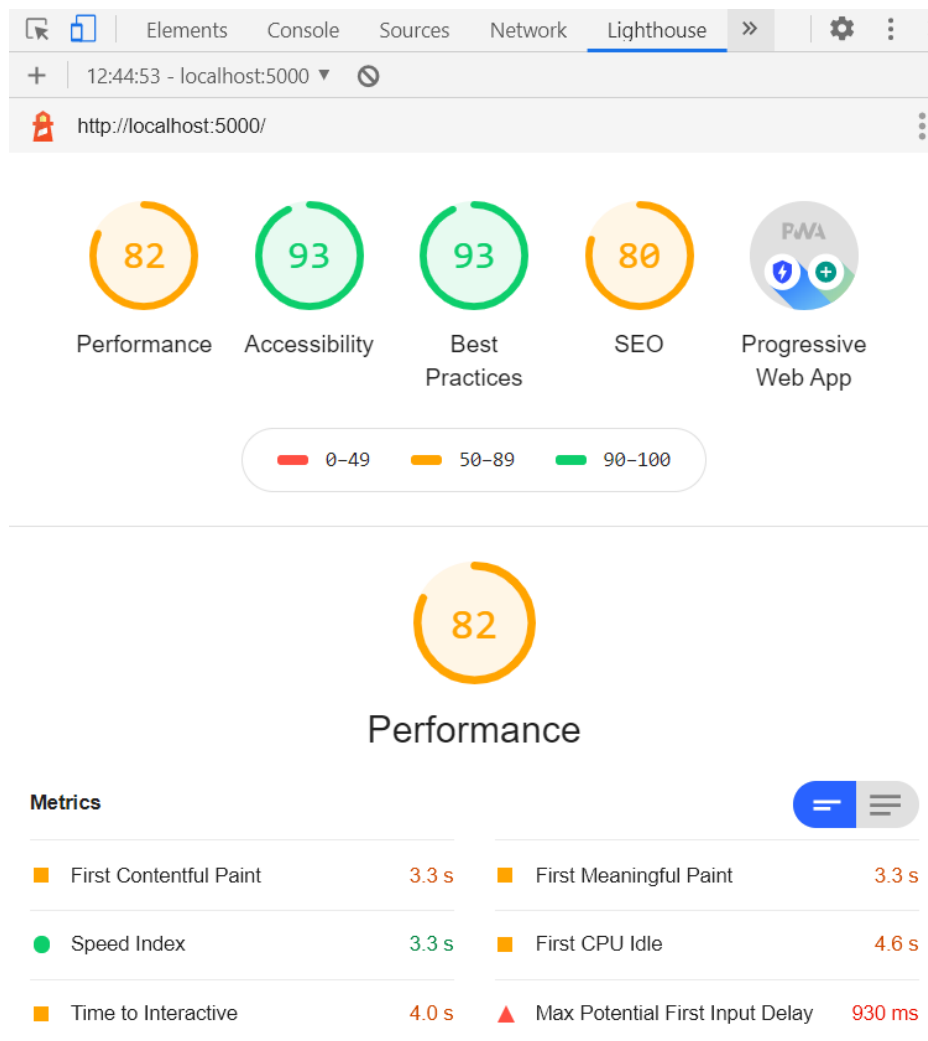


Рисунок 9 – Результат Lighthouse

Репозиторий с приложением Vue находится по ссылке:

https://github.com/BLIIN/ISN_2020_9

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, декоратором Material Design, Node.js и библиотекой Vue.js.