

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 _____/Еремин
М.С.

Проверил старший преподаватель _____/Земцов
М.А.

1 Цель работы

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

2 Задание

Разработать простое PWA приложение

3 Ход работы

Для начала работы необходимо установить рабочее окружение.

Был установлен Node.js (это JavaScript-окружение построенное на движке Chrome V8) и пакетный менеджер npm (рисунок 1).

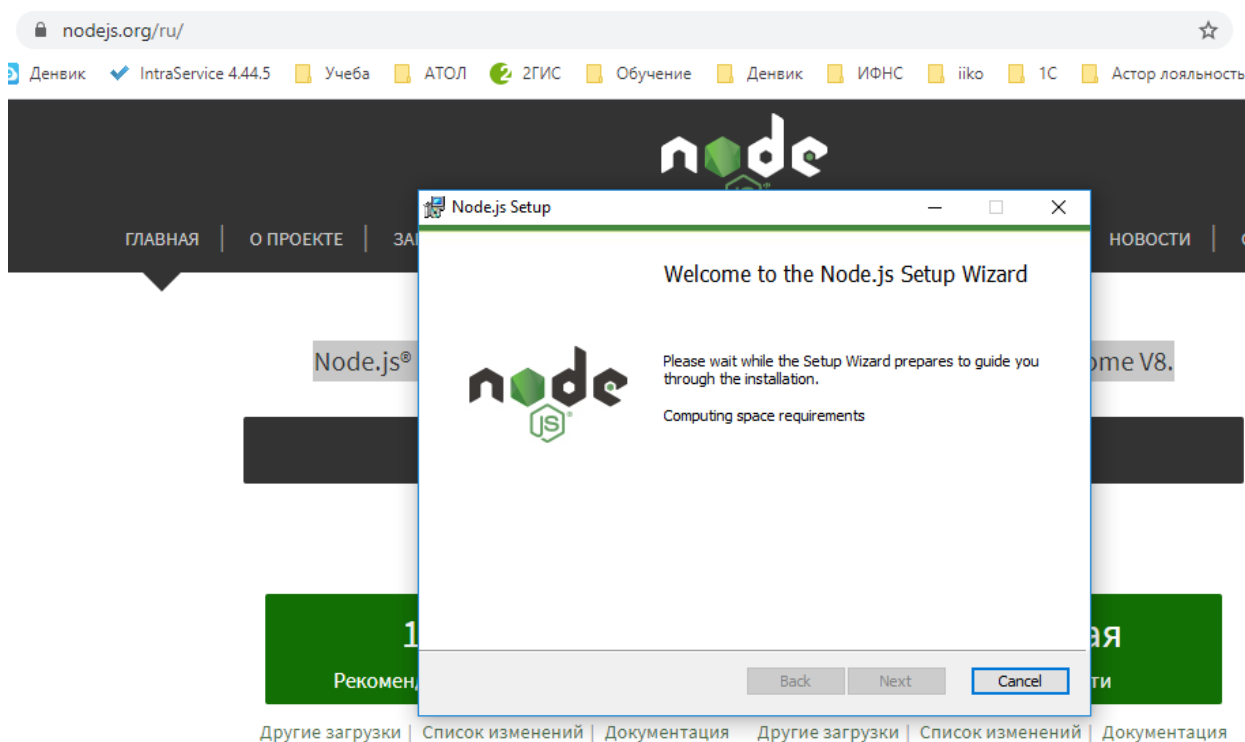


Рисунок 1 – установка Node js

Протестируем работу сервера (рисунок 2) и менеджера пакетов (рисунок 3)

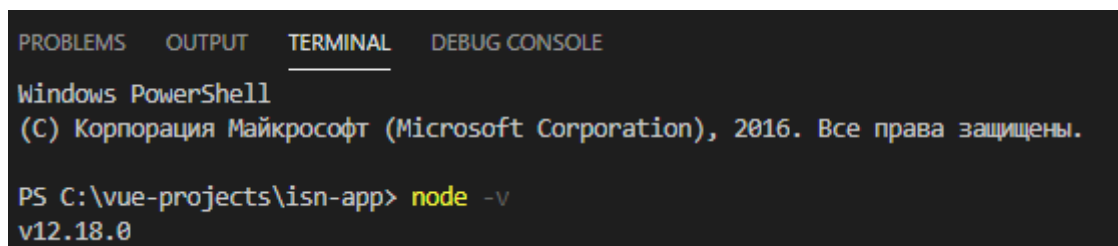


Рисунок 2 – тестирование Node js

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

npm -l      display full usage info
npm help <term> search for help on <term>
npm help npm involved overview

Specify configs in the ini-formatted file:
  C:\Users\admin\.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@6.14.4 C:\Program Files\nodejs\node_modules\npm
```

Рисунок 3 – тестирование пакетного менеджера npm

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

```
$npm install -g @vue/cli
```

Средствами npm были установлены утилиты Workbox

```
$npm install workbox-cli --global
```

и Firebase

```
$npm install -g firebase-tools
```

Создадим git-репозиторий ISN_2020_3

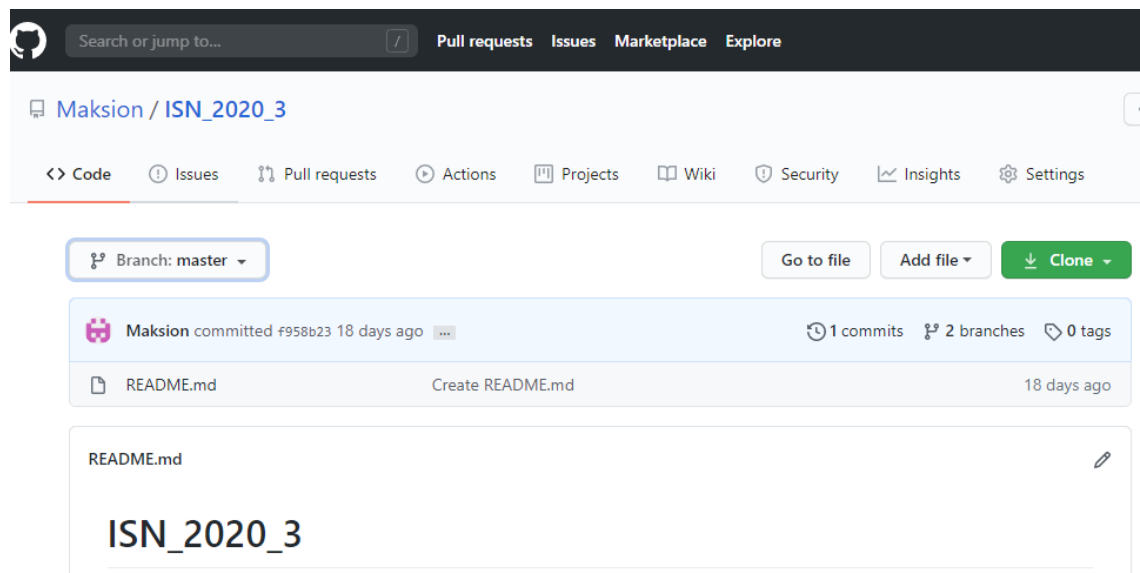


Рисунок 4 – git-репозиторий

Далее перейдем непосредственно к разработке PWA приложения. Для этого создадим директорию с проектами с правами gwrwgrw и откроем в ней консоль терминала.

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

Результат создания проекта представлен на рисунке 5

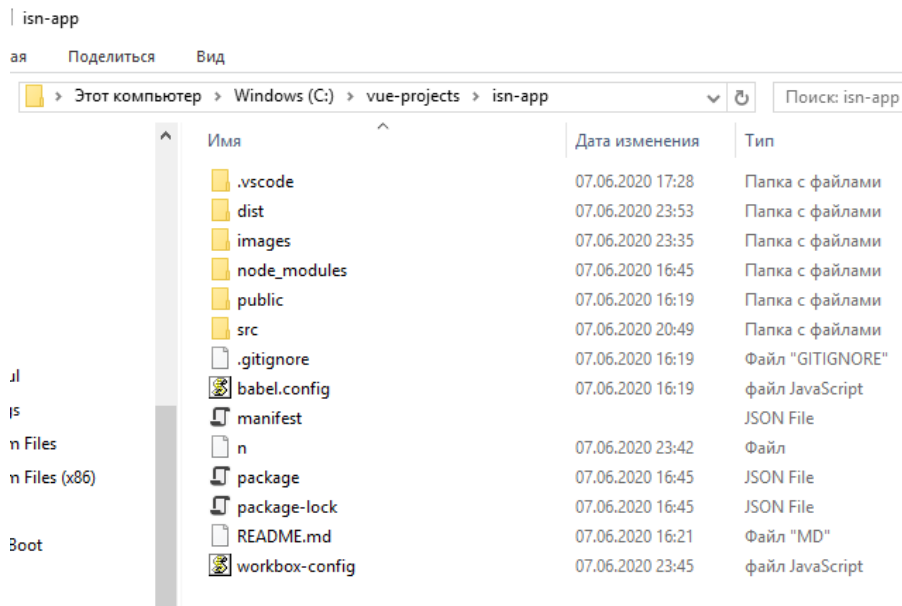


Рисунок 5 – Папка проекта приложения Vue

Командой `npm run serve` запускаем сервер разработки и переходим в браузере по адресу <http://localhost:8082> (рисунок 6)



Рисунок 6 - Проверка работоспособности приложения Vue

На рисунке 7 представлен скриншот страницы созданного vue приложения в режиме отображения на мобильном устройстве.



Рисунок 7 - приложение Vue

Для поддержки устройств с разными разрешениями установим поддержку декоратора Material Design, путем добавления компонента в vue (рисунок 8).

```
+ vuetify@2.3.2
updated 1 package and audited 1280 packages in 164.246s

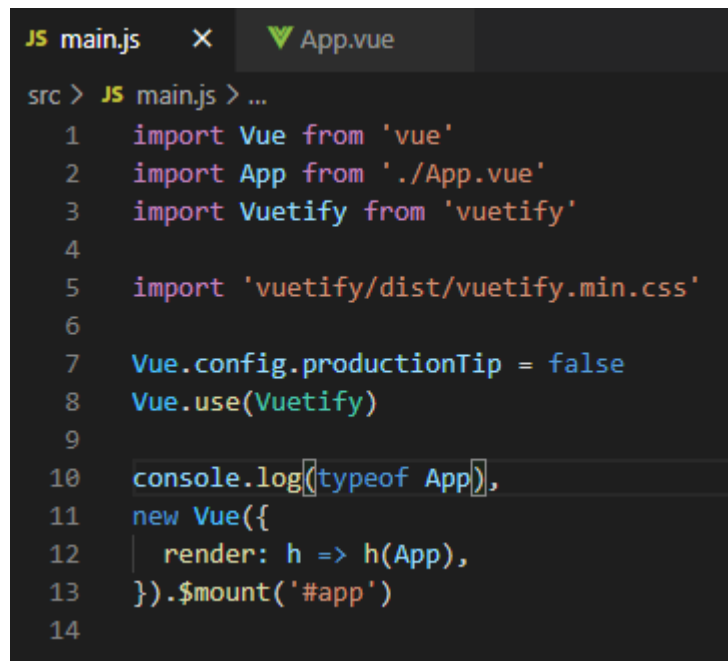
47 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities

PS C:\vue-projects\isn-app> |
```

Рисунок 8 - Установка декоратора Material Design

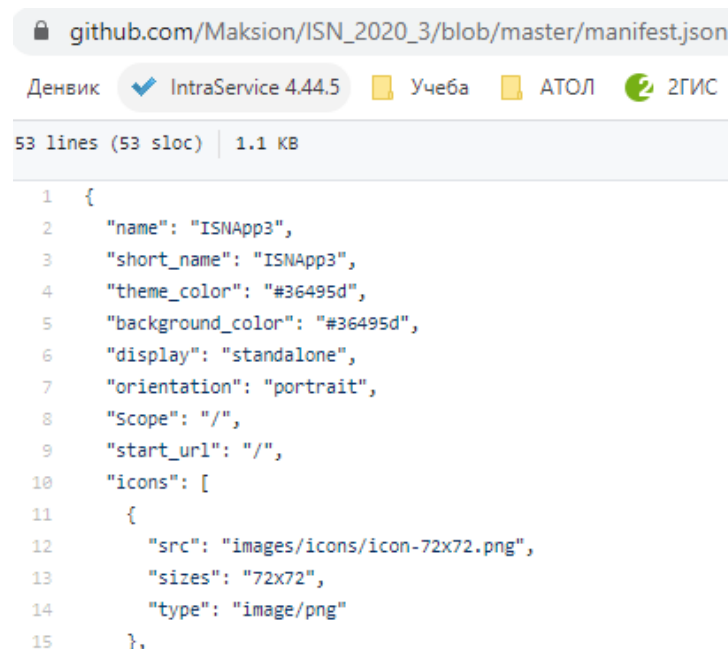
Далее подключим модуль в файле приложения main.js (рисунок 9).



```
JS main.js  X  App.vue
src > JS main.js > ...
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 console.log(typeof App),
11 new Vue({
12   render: h => h(App),
13 }).$mount('#app')
14
```

Рисунок 9 - подключение модуля Vuetify

Проверим работу полученного приложения в браузере Google Chrome при мобильном отображении контента. Сгенерируем манифест приложения с помощью сервиса Manifest. В результате получим конфиг, представленный на рисунке 10.



```
github.com/Maksion/ISN_2020_3/blob/master/manifest.json
Денвик  IntraService 4.44.5  Учеба  АТОЛ  2ГИС
53 lines (53 sloc) | 1.1 KB
1  {
2    "name": "ISNApp3",
3    "short_name": "ISNApp3",
4    "theme_color": "#36495d",
5    "background_color": "#36495d",
6    "display": "standalone",
7    "orientation": "portrait",
8    "Scope": "/",
9    "start_url": "/",
10   "icons": [
11     {
12       "src": "images/icons/icon-72x72.png",
13       "sizes": "72x72",
14       "type": "image/png"
15     },

```

Рисунок 10 – файл манифест

Полученные файлы, скачанные в архиве, распакуем и вставим в папку проекта. После этого добавим ссылку на манифест в index.html.

Далее создадим Service Worker, используя библиотеку Workbox. С помощью мастера генерации создадим файл sw.js в директории dist (рисунок 11).

Имя	Дата изменения
css	16.06.2020 8:30
img	16.06.2020 8:30
js	16.06.2020 8:30
favicon	16.06.2020 8:30
index	16.06.2020 8:30
manifest	16.06.2020 8:30
precache-manifest.1e97ada15d6e853f238...	16.06.2020 8:30
robots	16.06.2020 8:30
service-worker	16.06.2020 8:30

Рисунок 11 - Созданный файл sw.js

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:

```
npm build  
workbox generateSW workbox-config.js  
serve -s dist
```

На рисунке 12 можно увидеть что ServiceWorker работает

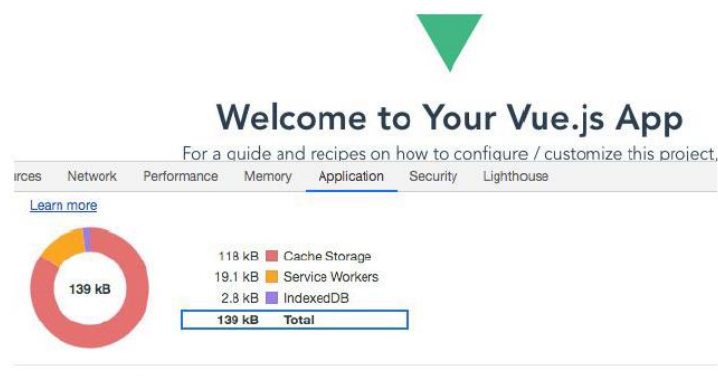


Рисунок 12 - результат работы ServiceWorker

Репозиторий с приложением Vue находится по ссылке:

https://github.com/Maksion/ISN_2020_3

4 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.