UDC 004.774

PROGRESSIVE WEB APPLICATIONS AS MEANS OF INCREASING WEB SERVICES FUNCTIONALITY

G.A. Piskun
Associate Professor, Department of Information Computer Systems Design, PhD of Technical sciences, Associate Professor
piskunbsuir@gmail.com

V.F. Alekseev
Associate Professor, Department of Information Computer Systems Design, PhD of Technical sciences, Associate Professor
alexvikt.minsk@gmail.com

T.M. Voronko
Software Engineer of the Center of Information Technologies of National Statistical Committee of the Republic of Belarus, master student of BSUIR
voronko232001@gmail.com

G.A. Piskun
Graduated from the Belarusian State University of Informatics and Radioelectronics. The area of scientific interests is related to modeling and optimal design of information and computer systems, organization of educational and research processes at a technical university.

V.F. Alekseev
Graduated from the Minsk Radio Engineering Institute. The area of scientific interests is related to the development of methods and algorithms for constructing information and computer systems, the organization of educational and research processes at a technical university.

T.M. Voronko
2nd year master's student in the specialty «Electronic Systems and Technologies» of the Department of Information and Computer Systems Design. Software engineer in the department of maintaining and development of information systems in the Center of Information Technologies of National Statistical Committee of the Republic of Belarus.

Abstract. An analysis of the relevance and advantages of progressive web applications in comparison with regular and native ones is carried out, the necessary requirements for creating a progressive web application are described, one of the ways to create such application is demonstrated using the example of a program that displays the current date and time according to the UTC standard without an Internet connection.

It has been established that progressive web applications can expand the functionality of related services, improving the user experience for their customers.

Keywords: progressive web application, service worker, web application manifest.

Introduction. Online services are becoming more and more essential in people's day-to-day lives. With ever-growing digitalization of all kinds of data surrounding us and constantly increasing accessibility and capabilities of Internet-connected devices, usage of web and native applications corresponding to these services is noticeably increasing, always demanding web developers to provide the best user experience possible.

One of the crucial factors making an application easy and pleasant for the client to use is it's accessibility. It means that the process of using the application should be responsible and fast, without being disorienting. It also demands that the app can be reached while having poor

Internet connection or even providing some of it's functionality while not being connected to the network at all. To say more, making an application distinct and easy to run through operating system's interface hugely increases it's accessibility [1].

Progressive web applications (PWA) can be helpful in achieving these goals.

Definition of PWA. A progressive web app is a type of application developed using web technologies such as JavaScript, CSS, and HTML. It closely resembles a regular web page in appearance and behavior. PWAs are easily discoverable in search engine page results and can be shared via links. What sets PWAs apart is their ability to provide functionalities similar to native mobile apps. They can operate offline, deliver push notifications, and leverage device hardware in a manner consistent with native apps [2].

PWAs merge accessibility and lightness of regular web applications together with functionality and built-in features of native applications resulting in increasing their capabilities and reachability. The graph corresponding to this statement is shown in figure 1 [3].

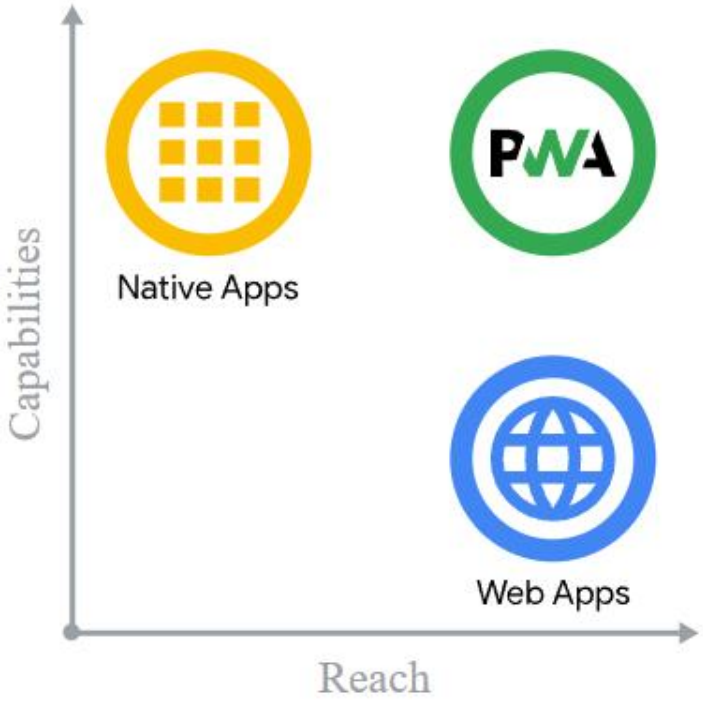

Figure 1. Comparison between types of applications by their capabilities and reachability

Progressive Web Apps are web applications that have been designed to be capable, reliable, and installable. These three pillars transform them into an experience that feels like a platform-specific application [3]:

1 ***Capability.*** Progressive Web Applications are designed to run on multiple operating systems and device classes from a single codebase, providing an identical user experience and being adaptive. It also combines functions of regular web and native applications, so all modern Web APIs, as well as a big variety of native APIs are available to use.

2 ***Reliability***. A reliable Progressive Web App feels fast and dependable regardless of the network conditions.

PWAs offer very fast loading speed and an ability to work under poor Internet connection or even offline with the help of service workers that store application's metadata and loaded resources on client's device via caching and overall lightness of such applications achieved by minification and compressing of files.

3 ***Installability***. Installed Progressive Web Apps run in a standalone window instead of a browser tab. They're launchable from on the user's home screen, dock, taskbar, or shelf. It's

possible to search for them on a device and jump between them with the app switcher, making them feel like part of the device they're installed on.

PWA requirements. In order to make an application a PWA the following criteria should be met [4]:

1 ***Web application manifest, with the correct members filled in***. It is the key element, which lists all the information about the website in a JSON format.

It usually resides in the root folder of a web app. It contains useful information, such as the app's title, paths to different-sized icons that can be used to represent the app on an OS (such as an icon on the home screen, an entry in the Start menu, or an icon on the desktop), and a background color to use in loading or splash screens. This information is needed for the browser to present the web app properly during the installation process, as well as within the device's app-launching interface, such as the home screen of a mobile device.

2 ***The website to be served from a secure (HTTPS) domain***. It makes PWA safer to use, because all transmitted data is encrypted. HTTPS prevents intruders from tampering with or passively listening in on the communications between the application and it's users.

PWA functionality is also available while developing an application if it's served over local development environment via localhost or 127.0.0.1 with or without a port number.

3 ***An icon to represent the app on the device***. It is recommended to provide multiple versions of an icon ranging in their sizes in order to make it appear correctly on any device.

4 ***A service worker registered, to allow the app to work offline***. It essentially acts as proxy server that sits between web applications, the browser, and the network (when available). It is required, among other things, to enable the creation of effective offline experiences, intercept network requests and take appropriate action based on whether the network is available, and update assets residing on the server. It also allows access to push notifications and background sync APIs.

The service worker is immediately downloaded when a user first accesses a service worker–controlled site or page. After that, it is updated when:

– a navigation to an in-scope page occurs;

– an event is fired on the service worker and it hasn't been downloaded in the last 24 hours.

Installation is attempted when the downloaded file is found to be new — either different to an existing service worker (byte-wise compared), or the first service worker encountered for this page or site.

If this is the first time a service worker has been made available, installation is attempted, then after a successful installation, it is activated. If there is an existing service worker available, the new version is installed in the background, but not yet activated — at this point it is called the worker in waiting. It is only activated when there are no longer any pages loaded that are still using the old service worker. As soon as there are no more pages to be loaded, the new service worker activates (becoming the active worker).

Creating React Application with the Progressive Web App Template. As an example, a simple React PWA that shows current date and time in Coordinated Universal Time (UTC)-format while not being connected to the Internet was created.

The first step is opening a command line interface from the directory that will contain PWA and running the following command (node.js should be installed on developer's machine) [4]:

```
npx create-react-app name-of-our-pwa-app --template cra-template-pwa
```

The resolving of this command resulted in building the application with the following structure as shown in figure 2 [5].

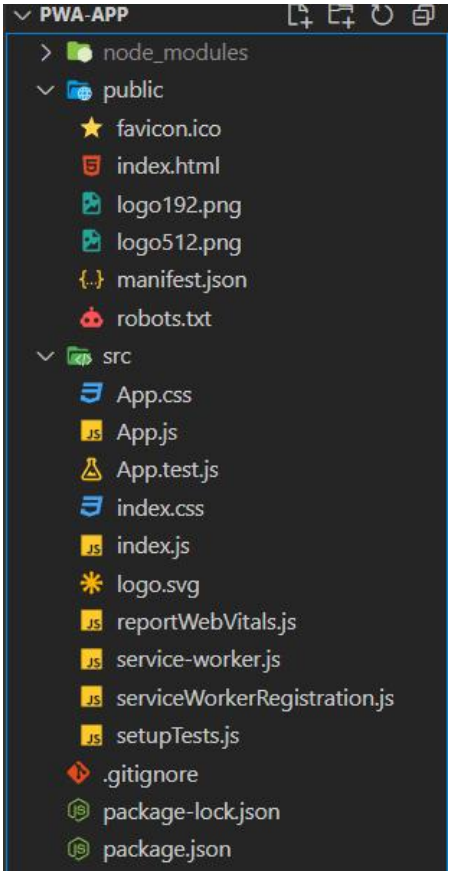

Figure 2. The initial structure of PWA built with React

This initial project already meets all the demands of making PWA installable, because service worker and web application manifest are built into React Progressive Web App Template by default.

The modified «manifest.json» file code is listed below [5].

```
{
  "short_name": "PWA example",
  "name": "PWA built for Big Data Conference",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
```

```
}
```

The code for «icons» array was left intact only changing the source «.png» and «.ico» files to custom ones.

The modified «App.js» file code that outputs date and time data that is updated every second is listed below.

```
import React, { useState } from "react";
import "./App.css";

function App() {
  const [date, setDate] = useState(new Date());

  setInterval(() => setDate(new Date()), 1000);

  return (
    <div className="App">
      <header className="App-header">
        <div>{date.toUTCString()}</div>
      </header>
    </div>
  );
}
export default App;
```

Inside «index.js» file the method of «serviceWorkerRegistration» instance should be changed from «unregister» to «register» in order to allow offline functionality with the help of service worker, that caches assets and scripts into client's device local storage [6]. It is recommended to be done on the final stages of development, because in some cases recently added local changes are becoming available with some delay in the development environment because of cached assets.

The service worker is only enabled in the production environment, e.g. the output of «npm run build».

After deploying the production build and running it inside browser, the PWA installation icon can be seen in URL bar as shown in figure 3.

Figure 3. PWA installation icon

After clicking on it the following prompt to install the app is shown as figure 4 illustrates.

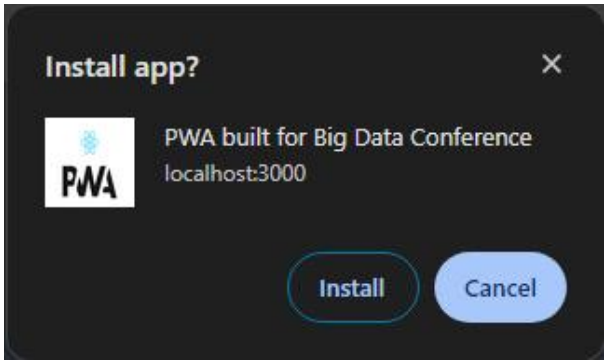

Figure 4. App installation prompt

Figure 5 shows installed PWA running as a standalone windows application with no Internet connection while still providing its functionality to show current date and time in UTC-format.

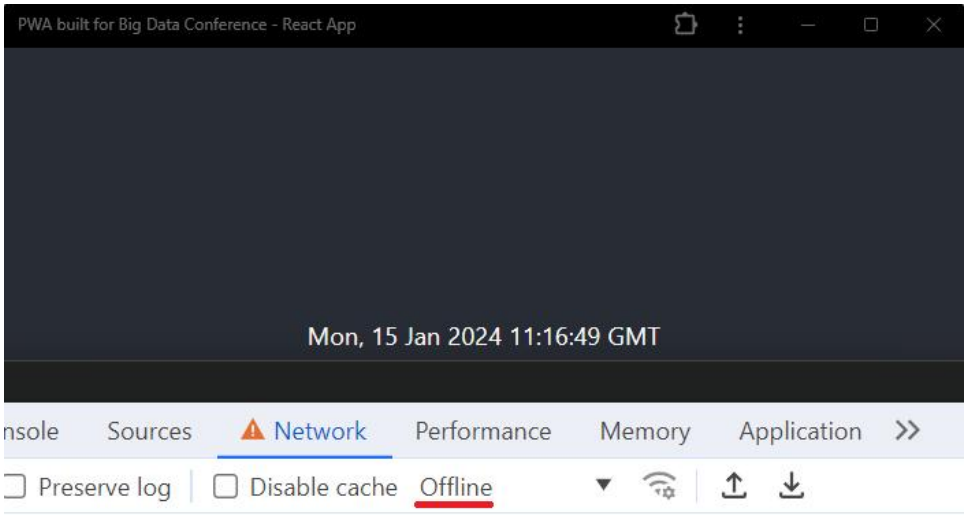

Figure 5. PWA working offline

Conclusion. At their heart, Progressive Web Apps are just web applications. Using progressive enhancement, new capabilities are enabled in modern browsers. Using service workers and a web app manifest, web application becomes reliable and installable. If the new capabilities aren't available, users still get the core experience. Companies that have launched Progressive Web Apps have seen impressive results. For example, Twitter saw a 65% increase in pages per session, 75% more Tweets, and a 20% decrease in bounce rate, all while reducing the size of their app by over 97%.

Progressive Web Apps provide developers with a unique opportunity to deliver more accessible and easy to reach web experience to their user base. Using the latest web features to bring enhanced capabilities and reliability, Progressive Web Apps can be installed by anyone, anywhere, on any device with a single codebase.

References

[1] Introduction to Progressive Web Apps [Electronic resource]. Mode of access: https://www.developer.com/web-services/intro-progressive-web-apps/. Date of access: 15.01.2024.

[2] Introduction to progressive web apps [Electronic resource]. Mode of access: https://www.divante.com/reports/pwabook/what-are-progressive-web-apps. Date of access: 15.01.2024.

[3] What are progressive web apps? [Electronic resource]. Mode of access: https://web.dev/articles/what-are-pwas. Date of access: 15.01.2024.

[4] How to make PWAs installable [Electronic resource]. Mode of access: https://developer.mozilla.org/enUS/docs/Web/Progressive_web_apps/Tutorials/js13kGames/Installable_PWAs. Date of access: 15.01.2024.

[5] How to Build a Progressive Web App (PWA) with React [Electronic resource]. Mode of access: https://www.loginradius.com/blog/engineering/guest-post/how-to-build-a-progressive-web-app-with-react/. Date of access: 15.01.2024.

[6] Using Service Workers [Electronic resource]. Mode of access: https://developer.mozilla.org/en-US/docs/Web/API/Service_Worker_API/Using_Service_Workers. Date of access: 15.01.2024.

author's contribution

Piskun Gennadiy Adamovich – determining the research problem, describing the working principle of Progressive Web Applications, analyzing results of research.

Alekseev Viktor Fedorovich – leading a study to assess the relevance and compatibility of Progressive Web Applications.

Voronko Timofei Maksimovich – describing key principles of Progressive Web Applications and it's requirements, developing the example project, forming the structure of the article.

ПРОГРЕССИВНЫЕ ВЕБ-ПРИЛОЖЕНИЯ КАК ИНСТРУМЕНТ ДЛЯ УВЕЛИЧЕНИЯ ФУНКЦИОНАЛЬНОСТИ ОНЛАЙН-СЕРВИСОВ

Г.А. Пискун
Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент

В.Ф. Алексеев
Доцент кафедры проектирования информационно-компьютерных систем БГУИР, кандидат технических наук, доцент

Т.М. Воронко
Инженер-программист Центра информационных технологий Белстата, магистрант БГУИР

Аннотация. Выполнен анализ актуальности и преимуществ прогрессивных веб-приложений в сравнении с обычными и нативными приложениями, описаны необходимые требования для создания прогрессивного веб-приложения, продемонстрирован один из способов создания такого приложения на примере программы, отображающей текущую дату и время по стандарту всемирного координированного времени без подключения к сети Интернет.

Установлено, что прогрессивные веб-приложения способны расширить функционал связанных с ними сервисов, улучшая опыт использования для их клиентов.

Ключевые слова: прогрессивное веб-приложение, сервис-воркер, манифест веб-приложения.