

PWA ЯК ОДИН ІЗ СУЧАСНИХ ПІДХОДІВ ДЛЯ СТВОРЕННЯ МОБІЛЬНИХ ЗАСТОСУНКІВ

Сучасні мобільні застосунки, стикаються з проблемами, такими як обмеженість доступу до ринку через необхідність установки, знижена взаємодія з користувачем через обмежені ресурси і підтримка певних платформ. Багато бізнесів, можуть відчувати труднощі у створенні та підтримці власного мобільного застосунка через великі витрати і технічні труднощі [1]. Вирішенням є використання Progressive Web App (PWA) – технології, яка дає сайту функціональність застосунку[2].

PWA поєднує в собі переваги веб-сайту і мобільного застосунка, надаючи користувачам зручний доступ через браузер та збільшуючи взаємодію.

В основі PWA лежать 4 ключові технології, що революціонізують взаємодію веб-застосунків:

1. Service Worker. Service Worker – це JavaScript-файл, який дозволяє виконувати операції у браузері. Він може обробляти події, такі як push-повідомлення, працювати з кешем для збереження ресурсів на пристрої та виконувати завдання в офлайн-режимі. Це поліпшує швидкість, доступність та ефективність роботи застосунка.

2. App Shell. App Shell – це архітектурний підхід, при якому основна оболонка сторінок PWA завантажується в кеш під час першого відвідування. Після цього, при подальшому використанні застосунка, лише контент завантажується з сервера. Це забезпечує швидке завантаження.

3. HTTPS. PWA працює виключно через захищений протокол HTTPS. Це забезпечує зашифровану передачу даних, що підвищує безпеку користувачів. Для забезпечення роботи через HTTPS, необхідно підключити SSL-сертифікат до налаштувань хостингу.

4. Web App Manifest. Web App Manifest – це файл, що містить дані про застосунок, такі як назва, іконка, режим вікна тощо. Він дозволяє налаштувати вигляд і поведінку PWA на пристрої. Наприклад, режим вікна може бути мінімальним, автономним або повноекранним, що дає різноманітні варіанти відображення.

Дані технології, об'єднані в одне ціле, створюють потужну інфраструктуру для створення швидких і безпечних веб-застосунків, які можуть працювати як в онлайн, так і в офлайн-режимі, надаючи враження використання мобільного застосунка.

Розглянемо основні переваги використання даної технології:

- Легкий доступ без установки. PWA дозволяють користувачам отримувати доступ до застосунка через веб-браузер без необхідності встановлення. Це розширює аудиторію та спрощує взаємодію.

- Підтримка всіх платформ. PWA використовують стандартні веб-технології, тому підтримують всі основні платформи і пристрої, що робить їх універсальними для різних типів користувачів.

- Взаємодія в автономному режимі. PWA можуть працювати в автономному режимі, що дозволяє користувачам використовувати застосунок навіть при відсутності Інтернет - з'єднання.

- Збільшена швидкість. PWA відзначаються високою швидкістю завдяки кешуванню ресурсів.

- Висока безпека. Використання протоколу HTTPS та інших стандартів безпеки робить цю технологію безпечною для використання. Інформація користувача та дані зашифровані, що зменшує ризик витоку конфіденційної інформації.

- Проста розробка та підтримка. Стандартні веб-технології, такі як HTML, CSS і JavaScript використовуються для створення PWA, що полегшує розробку та підтримку. Це зменшує витрати на розробку і спрощує майбутнє супровідне обслуговування.

Незважаючи на переваги PWA має обмеження та недоліки:

- Обмежені можливості на iOS. На деяких пристроях з iOS PWA не можуть повноцінно використовувати функціональність, наприклад, таку як push-повідомлення та фонову синхронізація. Однак з часом все може змінитися, оскільки Apple постійно вдосконалює підтримку PWA.

- Обмежені функції пристроїв. Деякі функції пристроїв, наприклад, такі як доступ до деяких сенсорів або розташування, можуть бути менш доступними порівняно з нативними застосунками, що може обмежити можливості застосунка.

- Не усі функції працюють в офлайн-режимі: Хоча PWA розроблені для роботи в офлайн-режимі, не всі функції можуть бути доступні без Інтернет-з'єднання. Залежно від реалізації і потреб застосунка, може бути обмежена функціональність в офлайн-режимі.

Список використаних джерел

1. Progressive Web Apps. *Google Developers*. URL: <https://developers.google.com/web/progressive-web-apps> (date of access: 12.11.2023).

2. Безуглий В.О., Петросян Р.В. Сучасні сайти як прогресивні веб-додатки. Тези XI Міжнародної науково-технічної конференції «Інформаційно-комп'ютерні технології – 2020», м. Житомир, 9-11 квітня 2020 р. Житомир: Житомирська політехніка. 2020. С. 35 – 36.