PROGRESSIVE E WEB APPS (PWAS)

THE FUTURE OF WEB DEVELOPMENT

TABLE OF CONTENTS

INTRODUCTION

- PWAs Merge Web & Native App Capabilities: They offer a seamless experience using service workers, web app manifests, and HTTPS.
- Bridging the Gap: PWAs enable features previously exclusive to native apps, reducing the need for cross-platform development.
- Research Contribution: Developed two cross-platform mobile apps and one PWA for comparison, shared in an open-source repository.
- Academic Interest: Aims to address the lack of research in this area and encourage academic exploration.
- Key Features: Responsive design, offline functionality, app-like performance, and reliability.

IMPORTANCE

- Cross-Device Compatibility – PWAs work seamlessly on any modern web browser, making them accessible across desktops, tablets, and smartphones without requiring platform-specific development.
- Cost-Effective Development – A single codebase runs on multiple platforms, reducing the time, effort, and cost needed to develop and maintain separate native apps for iOS and Android.
- App-Like Experience – PWAs provide smooth performance, fast loading times, and responsive designs, mimicking the look and feel of native applications without requiring installation from an app store.
- Offline Functionality – With the help of service workers, PWAs can function with limited or no internet connectivity, allowing users to access cached content and core features even when offline.

IMPORTANCE

- SEO & Discoverability – Unlike native apps that rely on app stores, PWAs are indexed by search engines, making them easier to find and driving organic traffic to businesses.
- Seamless Updates – Users always have the latest version of the app as PWAs update automatically in the background without requiring manual downloads or approvals from app stores.
- Home Screen Installation – Users can add PWAs to their home screen with a single tap, making them easily accessible like native apps without taking up significant storage space.
- Lower Storage Usage – PWAs consume minimal device storage compared to native apps, as they don’t require full installations and run directly from the browser.
- Higher User Engagement – Features like push notifications enable businesses to send real-time updates, promotions, and alerts, encouraging repeat visits and increasing conversions.

LITERATURE REVIEW

Building Progressive Web Apps: Bringing the Power of Native to the Browser :TAL ATER

- Practical, Hands-On Guide – Focuses on learning through coding rather than just theory, providing real-world examples and step-by-step instructions.

- Service Workers – Teaches how to implement service workers, which enable offline functionality, background syncing, and push notifications, making PWAs behave like native apps.

- Push Notifications – Covers how to set up push notifications to keep users engaged by displaying updates even when they aren't actively using the app.

- Performance Optimization – Provides insights on making web-based assets fast, efficient, and responsive, ensuring PWAs run as smoothly as native apps.

- Cross-Platform Compatibility – Focuses on ensuring PWAs function consistently across different devices, browsers, and operating systems.

LITERATURE REVIEW

Progressive Web Apps: JASON GRISBY

- Grigsby emphasizes that PWAs should not just function well but should be engaging, intuitive, and delightful to use.

- Focuses on user experience (UX) by covering aspects like snappy navigation, fast loading times, and seamless offline experiences.

- Rather than diving deep into coding, the book focuses on how PWAs can enhance UX and business outcomes.

- Makes PWA concepts accessible to a broad audience, including product managers, designers, and business executives.

- Provides real-world use cases for PWAs, including applications in e-commerce, content publishing, and business solutions.

LITERATURE REVIEW

Mastering Progressive Web Apps: Denis Rothman

- Service Workers – Enables offline functionality, background sync, and push notifications.
- Web App Manifest – Defines app metadata like icons, display mode, and home screen behaviour.
- Cache API – Stores assets locally to improve load times and offline access.
- Performance Optimization Techniques – Ensuring PWAs run efficiently and provide a smooth user experience.
- Payment Requests & Emerging Web Features – Discusses modern browser-based payment solutions and new APIs beneficial to PWAs.

LITERATURE REVIEW

Progressive Web Apps: John M. Wargo

- Web App Manifest – Defines the app's metadata (name, icon, theme, display mode) for a native-like installation experience.
- Service Workers – Explains how they enable background tasks, offline capabilities, caching, and network request interception.
- Offline access is a major focus, teaching how to store and retrieve data when no internet connection is available.
- Breaks down complex concepts into easy-to-follow instructions, making it accessible to newcomers.
- Ideal for those who prefer learning by doing rather than theoretical discussions.

LIMITATIONS

- Limited Hardware Access: PWAs run within a browser, which restricts their ability to use advanced hardware features like Bluetooth, proximity sensors, gyroscope, and some camera functionalities—features that native apps can fully utilize.

- iOS Restrictions: Apple has been historically slower in adopting full PWA capabilities compared to Android. Even though push notification support has improved in recent updates, iOS still has limitations on offline caching and background operations, affecting PWA performance.

- Absence from App Stores: Unlike native apps, PWAs are not readily available in major app stores like the Apple App Store and Google Play, making them harder for users to find and install. Some workarounds exist for Android, but it’s not as seamless.

- Browser Security Sandbox: PWAs operate within the browser’s security sandbox, which helps protect users but also restricts access to certain system-level features that native apps can use.

SCOPE

- Expanding Capabilities – PWAs are gaining deeper device integration, allowing access to Bluetooth, sensors, and advanced camera features, making them more powerful.
- Performance Boost with Web Assembly – Near-native speed enables PWAs to handle heavy computations, graphics-intensive tasks, and machine learning directly in the browser.
- Enhanced User Experience – Smarter offline functionality ensures seamless access even with poor connectivity, while interactive push notifications improve engagement.
- Improved Discoverability – Efforts are being made to integrate PWAs into app stores, making them easier to find and install.
- Better Developer Experience – Enhanced tools, standardized APIs, and simplified workflows make PWA development faster and more efficient.

CONCLUSION

- Expanding Capabilities – PWAs are gaining deeper device integration, allowing access to Bluetooth, sensors, and advanced camera features, making them more powerful.
- Performance Boost with Web Assembly – Near-native speed enables PWAs to handle heavy computations, graphics-intensive tasks, and machine learning directly in the browser.
- Enhanced User Experience – Smarter offline functionality ensures seamless access even with poor connectivity, while interactive push notifications improve engagement.
- Improved Discoverability – Efforts are being made to integrate PWAs into app stores, making them easier to find and install.
- Better Developer Experience – Enhanced tools, standardized APIs, and simplified workflows make PWA development faster and more efficient.

Thynk Unlimited

THANK YOU

FOR YOUR NICE ATTENTION

Telephone
+123-456-7890

Address
123 Anywhere St., Any City, ST 12345

Website
www.reallygreatsite.com

December 2023