



Web Engineering and Applications

Assignment 2

Progressive Web App

จัดทำโดย

นางสาวจิรายุทธ	ภู่นพกิจ	รหัสนักศึกษา 5910110054
นายชานน	พรศิริวงษ์	รหัสนักศึกษา 5910110071
นายพลิชฐ์	ธนดำเกิง	รหัสนักศึกษา 5910110216
นางสาวสุดาทิพย์	ทวีสุข	รหัสนักศึกษา 5910110354
นางสาวสุนิสา	ธรรมสุน	รหัสนักศึกษา 5910110358

เสนอ

อาจารย์สุนทร วิฑูรพจน์

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

## Assignment 2: progressive web app

### Firestore Hosting

[https://web-app-3a6d4.firebaseio.com/?fbclid=IwAR0sFDKbjuG3TFHPHIGY\\_PoY7VhGCxnLtLHSRte5sP0M\\_UnUo5ZoKLeGHXg](https://web-app-3a6d4.firebaseio.com/?fbclid=IwAR0sFDKbjuG3TFHPHIGY_PoY7VhGCxnLtLHSRte5sP0M_UnUo5ZoKLeGHXg)

### ความหมายของ Progressive Web Apps

Progressive Web Apps (PWA) คือ มาตรฐานการทำเว็บจาก Google ที่นำจุดเด่นของ Website และ Application มารวมกัน ผู้ใช้สามารถเข้าเว็บไซต์และใช้งานประหนึ่งว่าเป็น Application เลย ทั้งหน้าตาและฟีเจอร์ แต่โดยเบื้องหลังแล้วยังเป็นเทคโนโลยีของเว็บไซต์อยู่ (ใช้ HTML, รันด้วย Web Browser)

### ที่มาของ Progressive Web Apps

มาตรฐานนี้ผู้เริ่มต้นคือ Google โดย Google มองว่าทุกวันนี้ผู้ใช้โทรศัพท์ติดตั้งแอปใหม่น้อยมากถึงมากที่สุด รวมไปถึงถ้าเว็บคอนเทนต์สักเว็บหนึ่งต้องการที่จะทำ Application เพื่อให้ผู้ติดตามอ่าน ก็ดูเหมือนจะเป็นการลงทุนที่มากเกินไป

จึงมีแนวคิดทำแอปที่รันด้วย Web Browser ซึ่งเบื้องหลังคือเว็บไซต์เดิมที่ทำไว้อยู่แล้ว โดยนำ Logo ของเว็บนั้นมาอยู่ที่หน้า Homescreen เมื่อกดเข้าไปจะพบกับเว็บไซต์ที่มีหน้าตาและฟีเจอร์แบบแอปเลย

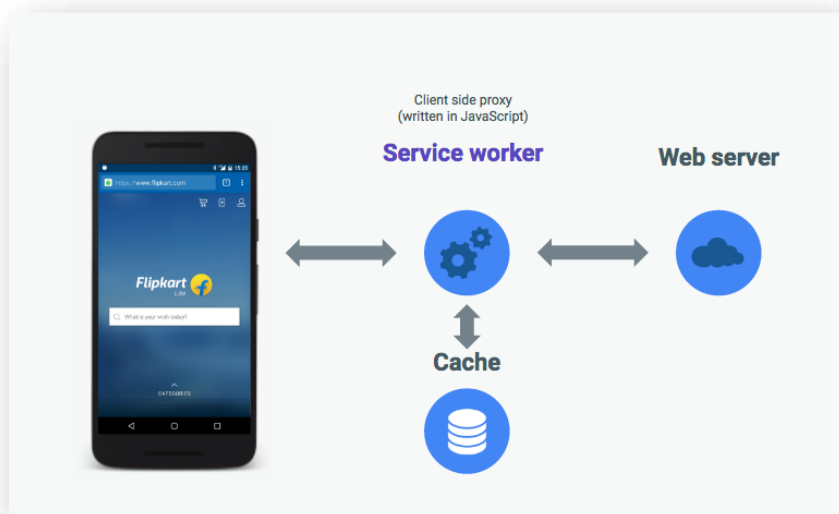
### ฟีเจอร์หลักโดยทั่วไปของ Progressive Web Apps

- ใช้งานได้ทุก Browser และทุก Mobile Platform
- ทำ Cache เก็บไว้ในตัว นักพัฒนาจะเป็นคนกำหนดว่าจะ Cache สิ่งใดไว้บ้าง ทำให้สามารถใช้งานได้ทั้ง Online และ Offline
- มีการอัปเดตข้อมูลทันทีเมื่อ Online
- Layout เหมือนกับ Application ผู้ใช้ไม่ต้องเรียนรู้ใหม่
- สามารถ Push Notification ได้ โดยที่ไม่ต้องเปิดเว็บหรือแอปทิ้งไว้

## การทำงานของ Progressive Web Apps

การทำงานของ PWA แบ่งออกเป็น 2 ส่วนหลักๆ คือ ServiceWorker กับ Manifest.json

### 1. ServiceWorker



ServiceWorker หรือ SW คือไฟล์ Client-side proxy ที่เราเขียนขึ้นมาด้วย JavaScript เมื่อ Users เข้ามาเว็บเรา เราก็จะทำการ Install SW ลงไปในเครื่องของ User คนนั้นๆ หน้าที่ของ SW ก็คือ กำหนดให้ Cache สิ่งต่างๆที่เราจำเป็นในเว็บของเราไว้ ซึ่งเราก็กำหนดได้ว่าจะให้ Cache ส่วนไหน ไม่ Cache ส่วนไหน

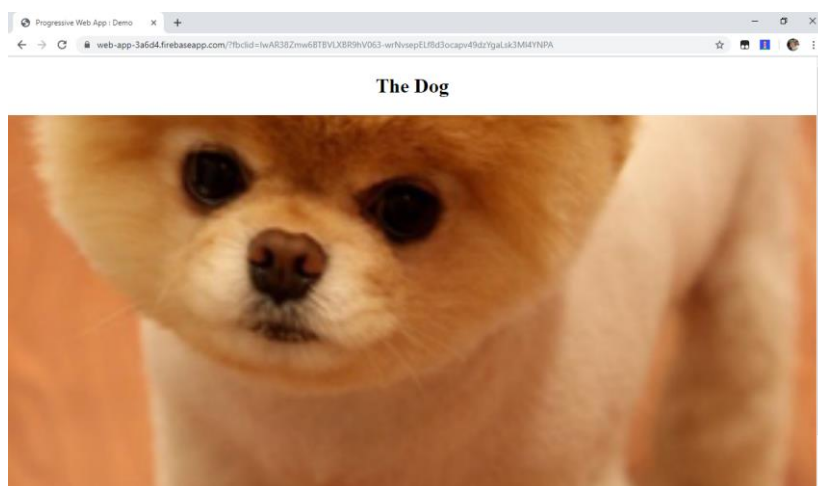
### 2. Manifest.json

คือไฟล์ JSON เล็กๆที่เราใส่เข้าไปใน head ของ html หน้าที่ดังนี้

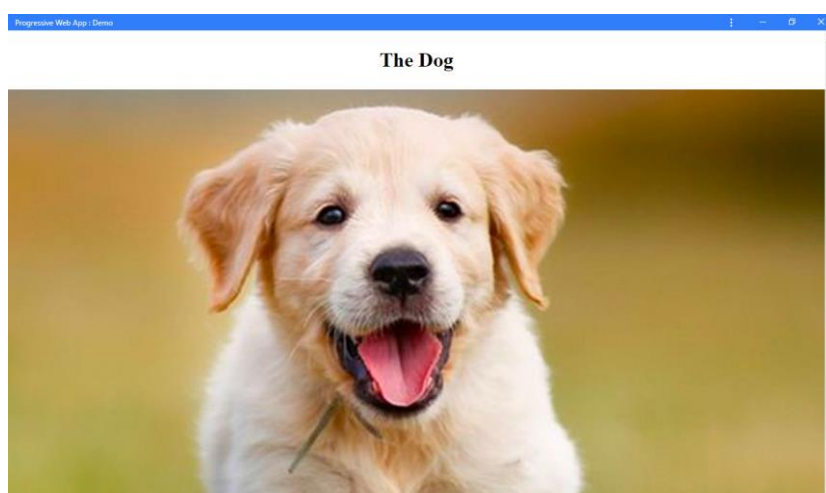
- ทำให้เว็บของเรามี Icon สวยๆบน หน้า Home screen เมื่อ Users กด Add to homescreen เว็บของเรา
- สามารถเปิดเว็บแบบ Full screen mode ไม่มี Address bar เมื่อ Users กดเข้ามาจากหน้า Homescreen
- ควบคุมมุมมองแนวตั้ง แนวนอน ของ Users ได้
- ระบุ สี และ Icon ที่จะใช้มาประกอบเป็น Splash screen (หน้าจอ ตอนกดแอปขึ้นมา ลองดูรูปข้างบนครับ อันขวาสุด)

## เริ่มต้นการสร้างการใช้งาน Progressive Web Apps

- ทำการสร้างในส่วนของ Template ที่มีส่วนประกอบดังต่อไปนี้
  - ❖ Html , Css , Js : ทำหน้าที่ในส่วนของการแสดงผล
  - ❖ ServiceWorker(sw.js) / Manifest.json : เข้ามาช่วยในเรื่องของการทำให้เว็บแอปพลิเคชันเป็นรูปแบบของ Progressive Web App
- ทำการเรียกใช้บริการในส่วนของ Firebase ฟังก์ชัน Hosting ที่เข้ามาช่วยในการทำให้เว็บของเรานั้นสามารถเข้าใช้งานได้จากลิงก์ Https / Http
- ผลลัพธ์ที่ได้จากการสร้างการใช้งานเว็บที่มีรูปแบบเป็น Progressive Web App



รูปที่ 1 ทำงานผ่านลิงก์ URL



รูปที่ 2 ทำงานผ่านการ install Desktop Google chrome

## เริ่มต้นการทดสอบประสิทธิภาพผ่านเครื่องมือ Lighthouse

- เริ่มต้น จากการติดตั้งในส่วนของปลั๊กอินที่มีชื่อว่า “Lighthouse”



- Lighthouse คืออะไร

Lighthouse คือ Tool ตัวหนึ่งของทาง Google ที่ใช้สำหรับวัดประสิทธิภาพของ Progressive Web App(PWA) ว่ามีการทำงานเต็มประสิทธิภาพหรือมีความเป็นมาตรฐานมากน้อยเพียงใดโดยจะแบ่งออกเป็น 4 ส่วนหลักในการพิจารณา

- ❖ **Progressive Web App:** ผ่านมาตรฐานของ PWA หรือไม่
- ❖ **Performance:** ความเร็วในการดาวน์โหลดของ Web หรือไม่
- ❖ **Accessibility:** สามารถเข้าได้ทุกกลุ่มบุคคลหรือไม่
- ❖ **Best Practices:** มีการทำตาม Best Practice ของ modern web development หรือไม่

- ผลลัพธ์หลังจากการทดสอบประสิทธิภาพ

