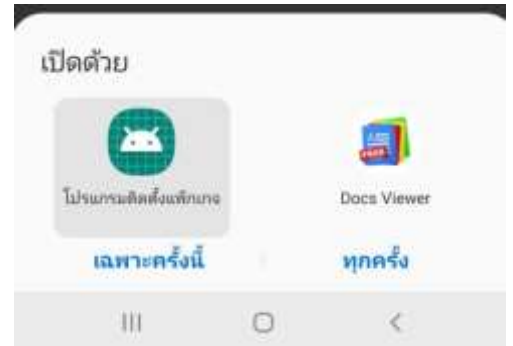


สรุปขั้นตอนการจัดทำแอปพลิเคชัน lpmuseum.net

1. ศึกษาความต้องการของแอปพลิเคชัน

พบว่า ต้องการทำ webview เชื่อมโยงกับเว็บไซต์ที่เป็น responsive web design สามารถแสดงผลบนอุปกรณ์ต่าง ๆ ได้ และผู้ดูแลโครงการได้จดโดเมนเป็น .net เรียบร้อยแล้ว สามารถเข้าบริหารจัดการผ่าน Direct Admin Control Panel ซึ่งสามารถใช้บริการ upload หรือ download ผ่าน ftp protocol และเปิดโฮมเพจได้ผ่านพอร์ต 80 ส่วนการพัฒนาด้วย thinkable.com นั้น การใช้ webview จะต้องใช้บริการ ssl ซึ่งระบบของ web hosting มีให้บริการแบบฟรี และมีข้อจำกัด ขั้นตอนการใช้ ssl คือ download เพิ่มทั้งระบบจาก public_html และ upload เข้าไปยัง private_html แล้วแก้ไขรหัสต้นฉบับเพื่อให้สามารถทำงานกับ https ได้ แต่ผลงานที่ได้จะมี logo ของ thinkable จึงพัฒนาด้วยเครื่องมือนี้เพื่อเปรียบเทียบผลระหว่าง android กับ ios เท่านั้น ซึ่งผู้พัฒนาเลือกพัฒนา application เพื่อส่งผลงานเผยแพร่ใน play store เนื่องจากมีความยืดหยุ่นในการพัฒนา และติดตั้งได้ง่ายโดยผู้ใช้งานมากกว่า



2. เตรียมเครื่องมือพัฒนาโปรแกรมหลัก

จากการศึกษา พบว่า เครื่องมือที่นิยมพัฒนาแอปพลิเคชันในปัจจุบัน คือ nodejs และ react native เชื่อมโยงกับ expo.io เผยแพร่บน google play store แล้วได้ทดสอบคู่ขนานไปกับการใช้บริการพัฒนาแอปพลิเคชันบน thinkable.com พบว่า การใช้ react native จะมีประสิทธิภาพ และไม่ติดปัญหา license หลังพัฒนาแล้วเสร็จ เนื่องจากเครื่องมือต่าง ๆ เป็น open source จึงได้ติดตั้งโปรแกรมต่าง ๆ เพื่อใช้พัฒนา ดังนี้



- ติดตั้ง nodejs จาก <https://nodejs.org/en/download>
- เข้าคอสแล้วสั่งเริ่มต้น react native project ด้วย npm install -g create-react-native-app
- ติดตั้ง yarn ด้วย <https://classic.yarnpkg.com/en/docs/install/#windows-stable>

- ติดตั้ง expo ด้วย npm install -g expo-cli
- สร้าง project ด้วย create-react-native-app myapp
- ทดสอบ project ด้วย npm start

3. เตรียมเพิ่ม app.json ซึ่งเป็นข้อกำหนดแอปแต่ละ release

```
{ "expo": { "name": "LP Museum", "slug": "lpmuseum", "privacy": "public", "sdkVersion": "39.0.0", "platforms": ["android", "ios", "web"], "version": "1.0.1", "orientation": "portrait", "icon": "./assets/icon.png", "splash": { "image": "./assets/splash.png", "resizeMode": "contain", "backgroundColor": "#ffffdd" }, "updates": { "fallbackToCacheTimeout": 0 }, "assetBundlePatterns": [ "**/*" ], "android": { "package": "com.thaiall.lpmuseum", "versionCode": 2, "permissions": [] }, "ios": { "bundleIdentifier": "com.thaiall.lpmuseum", "supportsTablet": true } } }
```

4. เตรียมเพิ่ม App.js

โดยเลือกเขียนคำสั่ง html เพื่อแสดงหน้าโฮมเพจแรก แทนการใช้ webview แบบกำหนด url โดยตรง ซึ่งจะไม่มีหน้าแรกในแอปพลิเคชัน แต่เรียกหน้าแรกจาก url มาแสดง ซึ่งพบว่าวิธีการเรียก url โดยตรง อาจไม่ผ่านเงื่อนไขการเผยแพร่แอปพลิเคชันของ google play store จึงพัฒนาแบบนั้นในรุ่นที่ 1 เมื่อต้องพัฒนาเพื่อส่งเข้า google play store จึงใช้เทคนิค html ทั้งหมด ก่อนพัฒนาเป็นรุ่นต่อ ๆ ไป



```
import React, { Component } from 'react'; import { WebView } from 'react-native-webview'; export default class App extends Component { render() { return ( <WebView originWhitelist={['*']} source={{ html: "<center><h1><div style='font-size:72;'>LP MUSEUM</div> <br>การจัดการพิพิธภัณฑ์ชุมชน<br>ในระบบอิเล็กทรอนิกส์ของจังหวัดลำปาง<br>โดย พระครูสิริธรรมบัณฑิต, ผศ. <br>ผู้อำนวยการวิทยาลัยสงฆ์นครลำปาง<br>หัวหน้าแผนวิจัยและคณะ<br><br><a
```

```
href='http://www.lpmuseum.net/banlook.html'>วัดบ้านหลุก</a> <br/><a  
href='http://www.lpmuseum.net/pongsanook.html'>วัดปงสนุกเหนือ</a>  
<br/><a href='http://www.lpmuseum.net/raihen.html'>วัดไหล่หินหลวง</a>  
<br/><a href=http://www.lpmuseum.net/research.html> <img  
src=http://www.lpmuseum.net/images/research%20index.jpg></a></h1></  
center>" }} style={{marginTop: 50}} /> ); }
```

5. ติดตั้งและสั่งประมวลผล

ซึ่งการ build แต่ละครั้งอาจพบปัญหาแตกต่างกัน จากความไม่เข้ากันของ application จึงต้องมีการติดตั้ง ทดสอบ และเลือกรุ่นที่เข้ากันได้มาใช้งาน ดังตัวอย่างการติดตั้ง react-native-webview แบบกำหนดรุ่น ซึ่งตัวอย่างนี้ใช้รุ่น 10.7.0 และพบว่าสามารถประมวลผลได้ตามปกติ บน web

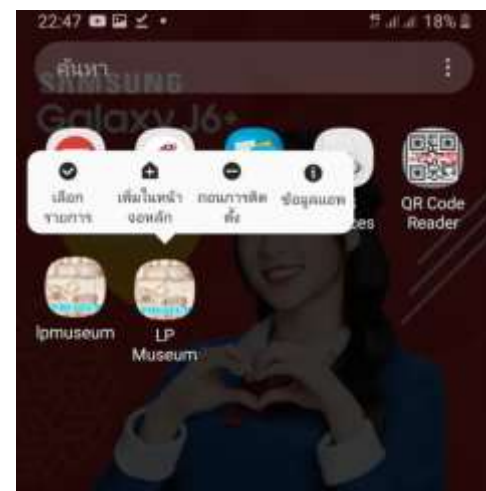
- npm install --save [react-native-webview@10.7.0](#)
- npm audit fix
- npm install
- npm audit fix
- npm run web

6. ส่งผลงานขึ้นไปยัง expo.io

เริ่มต้นจากการสมัครใช้งาน expo.io ในฐานะผู้พัฒนา ที่จะส่ง application ขึ้นไปเผยแพร่ และมี application ของ expo ในฝั่งผู้พัฒนา ที่สามารถติดตั้งบน android mobile เพื่อทดสอบ applicaiton เหล่านั้นได้โดยตรง และเชื่อมโยงบัญชีผลงานของผู้พัฒนา ซึ่งการ build ผลงานหลังการปรับปรุงในแต่ละ release ตามข้อเสนอแนะของ tester นั้น มีขั้นตอนดังนี้

```
expo fetch:android:keystore
```

```
expo build:android -t app-bundle
```



expo build:android

expo build:ios

7. เข้าไปในเว็บไซต์ expo.io

โดยใช้บัญชีที่ผูกกับระบบ ที่ใช้ในการ build แล้ว download เพิ่ม myapp.apk , myapp.aab และ myapp.tar.gz เพื่อนำไปใช้เผยแพร่ หรือเตรียมสำหรับนำเข้าไป play store ซึ่งการส่งเข้า play store ต้องแสดงถึงความก้าวหน้าโดยกำหนดรุ่นตามลำดับ ทำให้ต้องมีการ build หลายครั้ง เพื่อให้ได้ release หลายรุ่น แต่การใช้เพิ่ม myapp.apk สามารถนำไปใช้บน android mobile ได้ทันที หรือแชร์ใน website ผ่าน qr code ก็ยังสามารถทำได้ และใช้งานได้จริง ซึ่งการติดตั้ง myapp.apk แบบไม่ผ่าน play store ก็จะมีขั้นตอนที่ผู้ใช้ต้องเข้าใจ



8. เตรียมภาพสำหรับประกอบการพัฒนาแอป และเผยแพร่ใน store

การเตรียมภาพ เริ่มต้นจากเข้าไปเลือกภาพจาก lpmuseum.net และตัดต่อภาพ ตามขนาดที่ต้องการ โดยใช้ Adobe Photoshop, Microsoft Paint และ Irfan view เพื่อจัดเตรียมภาพ icon , splash , cover และภาพตัวอย่างหน้าจอบนอุปกรณ์

9. เข้าไปสมัครใช้งาน google developer account

เข้าไปที่ <https://play.google.com/console> และชำระค่าธรรมเนียม \$25 เพื่อการได้รับสิทธิ์ในการ upload application ขึ้นไปยัง play store ดำเนินการส่งเพิ่ม myapp.apk พร้อมรายละเอียดต่าง ๆ ของแอปพลิเคชัน โดยส่งเข้าไป เพื่อรับการทดสอบในแต่ละ release จนถึงการส่งขึ้น production release แล้วรอการทดสอบ และตรวจสอบเงื่อนไขต่าง ๆ อีกระยะหนึ่ง จึงจะได้รับการอนุมัติเผยแพร่ใน play store

