

การแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้เป็นเดสก์ท็อปข้ามแพลตฟอร์มด้วยอิเล็กทรอนิกส์



บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2560

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

TRANSFORMING JAVASCRIPT-BASED WEB APPLICATION TO CROSS-
PLATFORM DESKTOP WITH ELECTRON

Mr. Kitti Kredpattanakul



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Software Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2017

Copyright of Chulalongkorn University

กิตติ เกร็ดพัฒนกุล : การแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้เป็นเดสก์ท็อปข้ามแพลตฟอร์มด้วยอิเล็กทรอนิกส์ (TRANSFORMING JAVASCRIPT-BASED WEB APPLICATION TO CROSS-PLATFORM DESKTOP WITH ELECTRON) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: รศ. ดร.ญาใจ ลิ้มปิยะกรณ, 55 หน้า.

ช่วงหลายปีที่ผ่านมา มีหลายแนวทางได้เกิดขึ้นเพื่อปรับเปลี่ยนแอปพลิเคชันที่มีอยู่ไปสู่โปรแกรมที่มีฐานร่วมกัน ในบรรดาแนวทางต่างๆ อิเล็กทรอนิกส์เป็นกรอบงานหนึ่งที่แพร่หลายรู้จักกันดีสำหรับนักพัฒนาเว็บเพื่อใช้สร้างเดสก์ท็อปแอปพลิเคชันข้ามแพลตฟอร์ม ด้วยเว็บเทคโนโลยีต่างๆ ที่คุ้นเคย ได้แก่ เอชทีเอ็มแอล ซีเอสเอส และจาวาสคริปต์ งานวิจัยนี้จึงได้นำเสนอแนวทางสำหรับการแปลงเว็บแอปพลิเคชันที่พัฒนาด้วยจาวาสคริปต์ให้เป็นเดสก์ท็อปแอปพลิเคชันที่สามารถประมวลผลได้บนระบบปฏิบัติการวินโดวส์ แมคโอเอส และลินุกซ์ เดสก์ท็อปแอปพลิเคชันที่เป็นผลลัพธ์จากการแปลงจะยังคงประกอบด้วยซอร์สโค้ดชุดเดิมที่สามารถพัฒนาได้ต่อไป



ภาควิชา วิศวกรรมคอมพิวเตอร์

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมซอฟต์แวร์

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ปีการศึกษา 2560

5970904021 : MAJOR SOFTWARE ENGINEERING

KEYWORDS: CROSS-PLATFORM DESKTOP / WEB APPLICATION / ELECTRON
FRAMEWORK / JAVASCRIPT

KITTI KREDPATTANAKUL: TRANSFORMING JAVASCRIPT-BASED WEB
APPLICATION TO CROSS-PLATFORM DESKTOP WITH ELECTRON. ADVISOR:
ASSOC. PROF. YACHAI LIMPIYAKORN, Ph.D., 55 pp.

Over the years, various ways emerge for evolving the existing applications towards a shared codebase. Among several, Electron is a well-known framework for web developers to build cross-platform desktop applications using familiar web technologies, such as HTML, CSS, and JavaScript. This paper thus presents an approach for transforming web applications created with JavaScript to desktop applications that can run on Windows, MacOS, and Linux. The output desktop application would remain the old set of source code for further development.



Department: Computer Engineering Student's Signature

Field of Study: Software Engineering Advisor's Signature

Academic Year: 2017

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์จากรองศาสตราจารย์ ดร.ญาใจ ลิ้มปิยะภรณ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ได้สละเวลาให้ความรู้ คำปรึกษา ตรวจสอบและแก้ไขข้อผิดพลาดต่างๆ ตลอดจนการกำกับดูแลและคอยติดตามความก้าวหน้า ทำให้การวิจัยนี้สำเร็จไปได้ด้วยดี ผู้วิจัยขอกราบขอบพระคุณเป็นอย่างสูงไว้ ณ โอกาสนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.สุกรี สินธุภิญโญ อาจารย์ ดร.ดวงดาว วิชาดา กุล และอาจารย์ ดร.ภาสกร อภิรักษ์วรพินิต กรรมการสอบวิทยานิพนธ์ ที่กรุณาเสียสละเวลา ให้คำแนะนำ ตรวจสอบ และแก้ไขวิทยานิพนธ์ฉบับนี้

ขอขอบพระคุณบิดา มารดา และญาติพี่น้องที่ให้การสนับสนุนและเป็นกำลังใจที่ดีให้เสมอมาและสนับสนุนด้านทุนทรัพย์ในการศึกษารวมไปถึงทุกท่านที่มีส่วนช่วยเหลือในการทำวิทยานิพนธ์ครั้งนี้ ซึ่งมีได้กล่าวนามในที่นี้

ท้ายที่สุด ผู้วิจัยขอขอบพระคุณเพื่อนๆ ทุกคน ที่คอยติดตามและให้กำลังใจ รวมถึงท่านอื่นๆ ที่มีได้กล่าวลงนามไว้ ณ ที่นี้ที่มีส่วนทำให้วิทยานิพนธ์สำเร็จลุล่วงไปได้ด้วยดีผู้วิจัยหวังเป็นอย่างยิ่งว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์บ้างไม่มากก็น้อยสำหรับผู้สนใจจะศึกษารายละเอียดต่อไป

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฌ
สารบัญรูป.....	ญ
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตงานวิจัย.....	2
1.4 ขั้นตอนการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์.....	3
1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์.....	3
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	4
2.1 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1.1 จาวาสคริปต์.....	4
2.1.2 กรอบงานอิเล็กทรอนิกส์ (Electron Framework).....	5
2.2 งานวิจัยที่เกี่ยวข้อง.....	7
2.2.1 WAIL: Collection-Based Personal Web Archiving.....	7
บทที่ 3 แนวคิดและวิธีวิจัย.....	8

3.1 เตรียมกระบวนการงานการแปลง (Prepare Transformation Procedure)	9
3.1.1 แปลงโครงการพัฒนาเว็บแอปพลิเคชันตัวอย่าง	9
3.1.2 ผลลัพธ์จากการเตรียมกระบวนการแปลง	18
3.2 จัดทำเครื่องมือการแปลง (Create Transformation Tool)	20
บทที่ 4 การพัฒนาเครื่องมือ	24
4.1 ความต้องการเชิงฟังก์ชัน	24
4.2 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสนับสนุน	24
4.3 ขั้นตอนการทำงานของเครื่องมือ	25
4.4 การสกริปต์ด้วยเครื่องมือที่พัฒนา	32
บทที่ 5 การทดสอบระบบ	37
5.1 จุดประสงค์ของการทดสอบ	37
5.2 การทดสอบระบบ	37
5.3 สรุปผลการทดสอบ	48
บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ	49
6.1 สรุปผลการวิจัย	49
6.2 ข้อจำกัดงานวิจัย	49
6.3 งานวิจัยในอนาคต	50
รายการอ้างอิง	51
ภาคผนวก	52
ประวัติผู้เขียนวิทยานิพนธ์	55

สารบัญตาราง

ตารางที่ 1 เวลาที่ใช้ในการแปลงเว็บแอปพลิเคชันให้เป็นเดสก์ท็อป	32
ตารางที่ 2 ผลการทดสอบการติดตั้งเครื่องมือ.....	39
ตารางที่ 3 ผลการทดสอบการรับคำสั่งและตัวแปร	39
ตารางที่ 4 ผลการทดสอบการสำรองข้อมูล	40
ตารางที่ 5 ผลการทดสอบการสร้างไฟล์เมม	41
ตารางที่ 6 ผลการทดสอบการแก้ไขไฟล์แพ็กเกจ.....	41
ตารางที่ 7 ผลการทดสอบการเรียกใช้คำสั่งบิลด์.....	44
ตารางที่ 8 ผลการทดสอบการแก้ไขไฟล์เอชทีเอ็มแอล.....	45
ตารางที่ 9 ผลการทดสอบการเรียกใช้คำสั่งเพื่อเปิดเดสก์ท็อปแอปพลิเคชัน	46
ตารางที่ 10 ผลการทดสอบการแสดงผลส่วนต่อประสานผู้ใช้งานของเดสก์ท็อปแอปพลิเคชัน	47
ตารางที่ 11 ผลการทดสอบการทำงานของเดสก์ท็อปแอปพลิเคชัน	47

สารบัญรูป

รูปที่ 1 ชนิดของกระบวนการทำงานในแอปพลิเคชันบนกรอบงานอิเล็กทรอนิกส์ [5].....	6
รูปที่ 2 ภาพรวมของงานวิจัย.....	8
รูปที่ 3 ตัวอย่างเว็บแอปพลิเคชันที่จะทำการทดลอง.....	9
รูปที่ 4 โครงสร้างโครงการรีแอกต์เจเอส.....	10
รูปที่ 5 เดสก์ท็อปแอปพลิเคชันตัวอย่าง.....	10
รูปที่ 6 โครงสร้างโครงการอิเล็กทรอนิกส์.....	11
รูปที่ 7 โครงสร้างโครงการหลังทำการทดลอง.....	11
รูปที่ 8 สคริปต์และความขึ้นต่อกันของโครงการพัฒนารีแอกต์เจเอส.....	12
รูปที่ 9 สคริปต์และความขึ้นต่อกันของโครงการพัฒนารีอิเล็กทรอนิกส์.....	13
รูปที่ 10 สคริปต์และความขึ้นต่อกันหลังจากทำการรวม.....	14
รูปที่ 11 ตัวอย่างซอร์สโค้ดในไฟล์เมน.....	15
รูปที่ 12 การติดตั้งโมดูลความขึ้นต่อกัน.....	16
รูปที่ 13 การป้อนคำสั่งบิลด์.....	17
รูปที่ 14 ไฟล์เอชทีเอ็มแอลของโครงใหม่.....	17
รูปที่ 15 การป้อนคำสั่งอิเล็กทรอนิกส์สตาร์ทสำหรับเปิดการทำงาน.....	18
รูปที่ 16 เดสก์ท็อปแอปพลิเคชันที่ได้จากการทดลอง.....	18
รูปที่ 17 โครงสร้างโครงการพัฒนาสำหรับกรอบงานอิเล็กทรอนิกส์.....	19
รูปที่ 18 กระบวนการงานการแปลง.....	20
รูปที่ 19 คำสั่งและตัวแปรสำหรับการแปลง.....	21
รูปที่ 20 ข้อมูลนำเข้าที่เป็นโครงการพัฒนารีแอกต์.....	21
รูปที่ 21 ส่วนต่อประสานผู้ใช้งานเว็บแอปพลิเคชันจากข้อมูลนำเข้า.....	22
รูปที่ 22 เดสก์ท็อปแอปพลิเคชันที่ถูกแปลงโดยเครื่องมือ.....	22
รูปที่ 23 ผังงานการสาธิตใช้เครื่องมือแปลงอัตโนมัติ.....	26

รูปที่ 24 การจัดเตรียมโครงการพัฒนาเว็บแอปพลิเคชันและไฟล์เครื่องมือ	26
รูปที่ 25 การติดตั้งเครื่องมือ	27
รูปที่ 26 การติดตั้งเครื่องมือสำเร็จ	27
รูปที่ 27 ทดสอบการรับคำสั่งของเครื่องมือ	27
รูปที่ 28 การใช้เครื่องมือแปลงเว็บแอปพลิเคชัน	28
รูปที่ 29 ตัวอย่างการทำงานของเครื่องมือ	28
รูปที่ 30 ข้อความสุดท้ายที่เครื่องมือแสดง	29
รูปที่ 31 รายการไฟล์และโฟลเดอร์หลังจากใช้เครื่องมือ	29
รูปที่ 32 รายการไฟล์และโฟลเดอร์ที่ถูกสำรองข้อมูล	30
รูปที่ 33 ไฟล์เอชทีเอ็มแอลที่ถูกแก้ไขโดยเครื่องมือ	30
รูปที่ 34 ไฟล์แพ็กเกจที่ถูกแก้ไขโดยเครื่องมือ	31
รูปที่ 35 สาธิตการใช้เครื่องมืออัตโนมัติ	33
รูปที่ 36 เครื่องมือทำงานตามกระบวนการแปลง	33
รูปที่ 37 เดสก์ท็อปแอปพลิเคชันที่ถูกเปิดใช้งานโดยเครื่องมืออัตโนมัติ	33
รูปที่ 38 เปรียบเทียบการแสดงผลแผนภูมิที่ 1	34
รูปที่ 39 เปรียบเทียบการแสดงผลแผนภูมิที่ 2	34
รูปที่ 40 เปรียบเทียบการแสดงผลแผนภูมิที่ 3	34
รูปที่ 41 เปรียบเทียบการแสดงผลแผนภูมิที่ 4	35
รูปที่ 42 เปรียบเทียบการแสดงผลแผนภูมิที่ 5	35
รูปที่ 43 เปรียบเทียบการแสดงผลแผนภูมิที่ 6	35
รูปที่ 44 เปรียบเทียบการแสดงผลแผนภูมิที่ 7	36
รูปที่ 45 เปรียบเทียบการแสดงผลแผนภูมิที่ 8	36
รูปที่ 46 เปรียบเทียบการแสดงผลแผนภูมิที่ 9	36
รูปที่ 47 หน้าส่วนต่อประสานการกำหนดค่าเริ่มต้นและการติดตั้งซอฟต์แวร์อินเทอร์เน็ตเจเอส	53

รูปที่ 48 ตรวจสอบการติดตั้งซอฟต์แวร์ผ่านบรรทัดคำสั่ง..... 54



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในช่วงหลายปีที่ผ่านมา กรอบงานจาวาสคริปต์ (JavaScript framework) มักนิยมนำมาใช้ ในการพัฒนาเว็บแอปพลิเคชัน เช่น รีแอคท์เจเอส (ReactJS) และแองกูลาร์เจเอส (AngularJS) ข้อดี ของเว็บแอปพลิเคชัน คือ สามารถใช้งานผ่านเว็บเบราว์เซอร์ได้โดยไม่ต้องติดตั้งบนแพลตฟอร์ม ในขณะที่ เดสก์ท็อปแอปพลิเคชันจะมีความยุ่งยากในการพัฒนาเพื่อให้รองรับกับแพลตฟอร์มที่ต่างกันออกไป

เมื่อมีความต้องการในการใช้งานแอปพลิเคชันบนเดสก์ท็อปจากผู้ใช้งานหรือมีความจำเป็น ต่อระบบ ยกตัวอย่างเช่น ระบบการทำงานเดิมมีการใช้งานบนเว็บเบราว์เซอร์ แต่ลูกค้าต้องการ ความสามารถที่เพิ่มมากขึ้นสำหรับการแจ้งเตือนผู้ใช้งานหรือมีความต้องการในการทำงานแบบไม่ ต้องการเชื่อมต่ออินเทอร์เน็ต ในปัจจุบันไม่จำเป็นต้องพัฒนาระบบใหม่เพื่อตอบสนองความต้องการ ต้องการเหล่านั้น ซึ่งหมายความว่าหากแอปพลิเคชันที่พัฒนาเสร็จสิ้นแล้วถูกพัฒนาบนกรอบงานจาวา สคริปต์ นักพัฒนาสามารถทำการแปลงให้เป็นเดสก์ท็อปแอปพลิเคชันได้โดยใช้กรอบงานที่เรียกว่า อิเล็กตรอน

การพัฒนาเดสก์ท็อปแอปพลิเคชันด้วยกรอบงานอิเล็กตรอน (Electron framework) [1] มี ความสะดวกสำหรับนักพัฒนา เนื่องจากนักพัฒนาไม่จำเป็นต้องเรียนรู้ภาษาใหม่ ในขณะที่เดียวกันยัง ได้แอปพลิเคชันซึ่งสามารถทำงานได้ทั้งบนระบบปฏิบัติการวินโดวส์ (Windows) แมคโอเอส (MacOS) และลินุกซ์ (Linux) หรือที่เรียกว่าเดสก์ท็อปแอปพลิเคชันข้ามแพลตฟอร์ม (Cross-platform desktop application) ซึ่งมีข้อดีเทียบเท่ากับแอปพลิเคชันบนเว็บ ด้วยกรอบงาน อิเล็กตรอน ทำให้สะดวกต่อการแปลงระบบเดิมบนเว็บเบราว์เซอร์มาเป็นเดสก์ท็อปแอปพลิเคชันโดย ที่นักพัฒนาไม่ต้องพัฒนาใหม่ทุกอย่างตั้งแต่เริ่ม เป็นการลดระยะเวลาในการพัฒนาและลดต้นทุน สำหรับการพัฒนาได้

เพื่อลดระยะเวลาในการแปลงแอปพลิเคชันและเป็นแนวทางในการพัฒนาเดสก์ท็อปแอป พลิเคชันเพิ่มเติม งานวิจัยนี้จึงได้นำเสนอแนวทางการพัฒนาเครื่องมือสำหรับแปลงโครงการพัฒนา เว็บแอปพลิเคชันโดยมีพื้นฐานบนกรอบงานรีแอคท์เจเอสให้เป็นโครงการพัฒนาเดสก์ท็อปแอปพลิเคชัน จากนั้นจึงทำการสร้างไฟล์ติดตั้งสำหรับระบบปฏิบัติการต่าง ๆ ซึ่งโครงการพัฒนาเดสก์ท็อปแอป พลิเคชันจะคงพื้นฐานบนกรอบงานเดิมไว้สำหรับการปรับปรุงแก้ไขแอปพลิเคชันในอนาคต และเพื่อ เป็นการประเมินคุณภาพของเครื่องมือ ผู้วิจัยจึงมีแนวทางการประเมินโดยใช้โครงการพัฒนาตัวอย่าง จากโครงการพัฒนาโอเพนซอร์ส มานำเข้าเครื่องมือและทวนสอบเดสก์ท็อปแอปพลิเคชันที่เป็น

ผลลัพธ์การแปลงจากเครื่องมือด้วยกรณีทดสอบเปรียบเทียบกับโครงการพัฒนาตัวอย่างซึ่งเป็นเว็บแอปพลิเคชัน

1.2 วัตถุประสงค์ของการวิจัย

เพื่อนำเสนอแนวทางและพัฒนาระบบสำหรับการอำนวยความสะดวกในการแปลงเว็บแอปพลิเคชันที่พัฒนาด้วยกรอบงานจาวาสคริปต์ให้เป็นเดสก์ท็อปแอปพลิเคชันข้ามแพลตฟอร์ม

1.3 ขอบเขตงานวิจัย

- 1) รองรับการแปลงในส่วนของการแสดงผลข้อมูลและการเรียกข้อมูลเท่านั้น
- 2) เว็บแอปพลิเคชันตั้งต้นต้องถูกพัฒนาบนกรอบงานรีแอคต์เจเอส
- 3) สามารถรับข้อมูลนำเข้า ประกอบด้วย
 - ไฟล์จาวาสคริปต์ในการพัฒนาเว็บแอปพลิเคชัน (app.js)
 - ไฟล์เอชทีเอ็มแอลในการพัฒนาเว็บแอปพลิเคชัน (index.html)
- 4) สามารถแสดงผลการทำงานในรูปแบบเดสก์ท็อปแอปพลิเคชันบนระบบปฏิบัติการวินโดวส์ แม็คโอเอส และลินุกซ์

1.4 ขั้นตอนการวิจัย

- 1) ค้นคว้าศึกษาและทำความเข้าใจทฤษฎีและงานวิจัยที่เกี่ยวข้อง
- 2) รวบรวมและศึกษาโครงสร้างของเว็บแอปพลิเคชันรีแอคต์เจเอส
- 3) ทดลองทำการแปลงเว็บแอปพลิเคชันตั้งต้นให้อยู่ในโครงสร้างของกรอบงานอิเล็กทรอนิกส์
- 4) ทดสอบผลลัพธ์การแปลงและกำหนดโครงสร้างสำหรับโครงการพัฒนาอิเล็กทรอนิกส์
- 5) ออกแบบความต้องการสำหรับเครื่องมืออัตโนมัติ
- 6) พัฒนาเครื่องมือ
- 7) ทดสอบและประเมินผลงานวิจัย
- 8) จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ได้วิธีการและเครื่องมือในการแปลงเว็บแอปพลิเคชันเป็นเดสก์ท็อปแอปพลิเคชันข้ามแพลตฟอร์ม ซึ่งช่วยลดทรัพยากรและค่าใช้จ่ายในการพัฒนา อีกทั้งทำให้นักพัฒนาไม่ต้องเริ่มกระบวนการในพัฒนาใหม่เพื่อตอบสนองความต้องการแอปพลิเคชันในรูปแบบเดสก์ท็อป

1.6 โครงสร้างของเนื้อหาในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกเป็น 6 บทด้วยกันคือ บทที่ 1 อธิบายถึงที่มาและความสำคัญของปัญหารวมถึงขอบเขตและประโยชน์ของงานวิจัย บทที่ 2 อธิบายถึงทฤษฎีที่เกี่ยวข้องและงานวิจัยที่เกี่ยวข้อง บทที่ 3 อธิบายถึงแนวคิดและวิธีการดำเนินการวิจัย บทที่ 4 อธิบายถึงวิธีการพัฒนาเครื่องมือสนับสนุนแนวคิดของงานวิจัย บทที่ 5 อธิบายวิธีการทดสอบและในบทสุดท้ายคือการสรุปงานวิจัยทั้งหมด รวมถึงงานวิจัยในอนาคต

1.7 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เรื่อง “Transforming JavaScript-Based Web Application to Cross-Platform Desktop with Electron”, Kitti Kredpattanakul and Yachai Limpiyakorn, เผยแพร่ใน Lecture Notes in Electrical Engineering, Volume 514, 2018, Pages 571-579. สืบเนื่องจากการประชุมวิชาการระดับนานาชาติ 9th iCatse Conference on Information Science and Applications (ICISA 2018), June 25-27, 2018, Hong Kong, China.

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ทฤษฎีที่เกี่ยวข้อง

2.1.1 จาวาสคริปต์

จาวาสคริปต์เป็นภาษาที่ได้รับความนิยมสำหรับการพัฒนาเว็บแอปพลิเคชันสมัยใหม่ ถูกสร้างขึ้นสำหรับเรียกใช้การทำงานหลักและส่วนต่อประสานผู้ใช้งานทางฝั่งผู้ขอใช้บริการ (Client-side) รวมไปถึงการพัฒนาแอปพลิเคชันแบบเรียลไทม์ที่ด้านของผู้ให้บริการ (Server-side) ผ่านโหนดเจเอส (Node.js) [2] โดยรูปแบบการทำงานด้านผู้ให้บริการแบบเก่ากำลังถูกปรับเปลี่ยนมาเป็นรูปแบบการทำงานของจาวาสคริปต์โดยใช้กรอบงานจาวาสคริปต์แบบใหม่ที่ขับเคลื่อนส่วนต่อประสานผู้ใช้งานให้สะดวกยิ่งขึ้น ในปัจจุบัน จาวาสคริปต์ใช้การรับข้อมูลผ่านเรสทูล์ฟูล (RESTful) หรือการเชื่อมต่อเว็บซ็อกเก็ต (WebSocket connection) ซึ่งจะถูกโยงไว้กับส่วนต่อประสานผู้ใช้งานในด้านของผู้ขอใช้บริการโดยอัตโนมัติ [3]

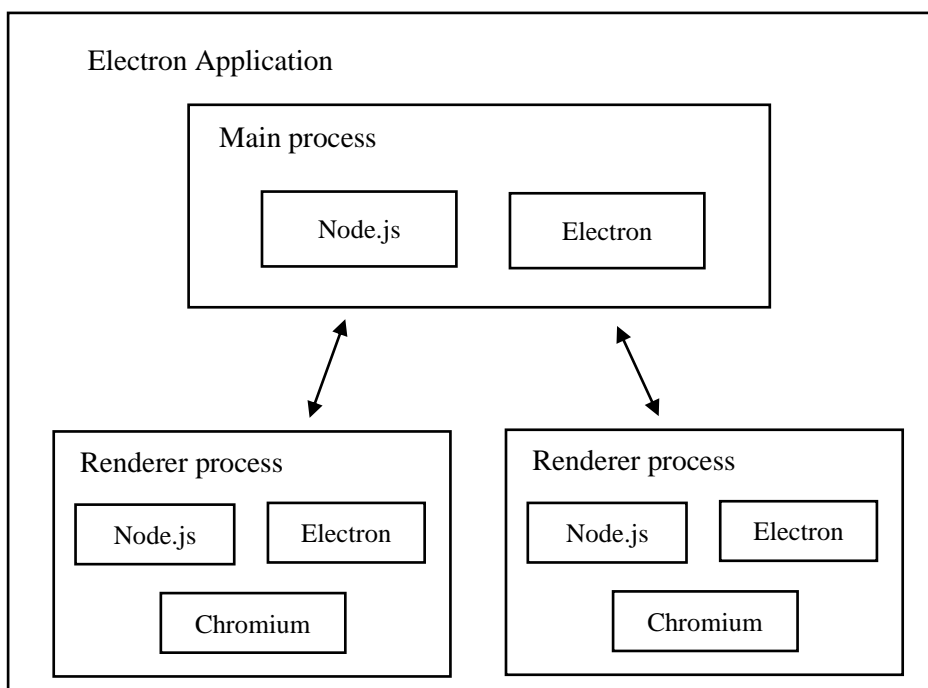
ปัจจุบันเว็บแอปพลิเคชันบนพื้นฐานของจาวาสคริปต์มีเทคโนโลยีในการช่วยพัฒนาที่เป็นที่นิยมยกตัวอย่างเช่น รีแอกต์เจเอส แองกูลาร์เจเอส และวูเจเอส (VueJS) โดยเทคโนโลยีแต่ละตัวถูกเรียกว่าเป็นกรอบงานจาวาสคริปต์หรือจาวาสคริปต์ไลบรารี (JavaScript Library) โดยรีแอกต์เจเอสและวูเจเอสเป็นไลบรารีเพื่อใช้จัดการส่วนแสดงผล ในขณะที่แองกูลาร์เจเอสเป็นกรอบงานที่มีลักษณะแบบเอ็มวีซี (MVC) ซึ่งไม่ได้จัดการส่วนแสดงผลเพียงอย่างเดียวแบบรีแอกต์เจเอสและวูเจเอส

เป้าหมายของเทคโนโลยีสำหรับช่วยพัฒนาเว็บแอปพลิเคชันเหล่านี้คือการทำให้งานมีความเร็วมากขึ้น โดยหน้าที่ของแต่ละเทคโนโลยีคือการจัดการกับดีโอเอ็ม (DOM: Document Object Model) หรือการจัดเก็บเอกสารในรูปแบบของวัตถุที่เป็นโครงสร้างต้นไม้ในเอชทีเอ็มแอล ซึ่งรีแอกต์เจเอสและวูเจเอสมีการใช้เทคโนโลยีดีโอเอ็มเสมือน (Virtual DOM) เพื่อหลีกเลี่ยงการปรับเปลี่ยนดีโอเอ็มให้มากที่สุดด้วยการจำลองดีโอเอ็มขึ้นมาหนึ่งชุด เมื่อมีการเรียกใช้งานเว็บและกำลังจะมีการปรับเปลี่ยนจึงทำการเปรียบเทียบและเปลี่ยนแปลงเฉพาะส่วนที่ต่างกับส่วนที่กำลังแสดงผลอยู่ก่อนหน้าเท่านั้น โดยแองกูลาร์เจเอสใช้เทคโนโลยีการตรวจจับการเปลี่ยนแปลง (Change Detection) ซึ่งทำการตรวจจับการเปลี่ยนแปลงที่เกิดขึ้นผ่านการทำเหตุการณ์ต่าง ๆ บนเว็บแอปพลิเคชันตัวอย่างเช่นเมื่อมีการกดปุ่มทำให้แองกูลาร์เจเอสรู้ว่าต้องทำการเปลี่ยนแปลงดีโอเอ็มเป็นต้น

2.1.2 กรอบงานอิเล็กทรอนิกส์ (Electron Framework)

กรอบงานอิเล็กทรอนิกส์เป็นกรอบงานโอเพนซอร์ส (Open Source Framework) ใช้เป็นเครื่องมือสำหรับการพัฒนาเดสก์ท็อปแอปพลิเคชันที่ถูกพัฒนาที่ GitHub) มีความสามารถทำให้การพัฒนาเดสก์ท็อปแอปพลิเคชันสามารถพัฒนาได้โดยเทคโนโลยีแบบเดียวกันกับที่ใช้ในการพัฒนาเว็บแอปพลิเคชันด้วย จาวาสคริปต์ (JavaScript) เอชทีเอ็มแอล (HTML) และซีเอสเอส (CSS) อิเล็กตรอนทำงานโดยการรวมกรอบงานโครเมียม (chromium content framework) และโหนดเจเอสเข้าด้วยกันเป็นกรอบงานเดียว [4] และมีความสามารถในการสร้างเดสก์ท็อปแอปพลิเคชันที่ทำงานได้บนระบบปฏิบัติการวินโดวส์ แมคโอเอส และลินุกซ์

รูปที่ 1 แสดงให้เห็นถึงกระบวนการทำงานสองชนิดภายในแอปพลิเคชันบนกรอบงานอิเล็กทรอนิกส์ นั่นคือกระบวนการหลัก (Main process) และกระบวนการนำเสนอ (Renderer process) โดยเมื่อแอปพลิเคชันถูกเรียกใช้งาน กรอบงานอิเล็กทรอนิกส์จะเริ่มกระบวนการหลักหนึ่งกระบวนการในทุก ๆ ครั้งที่แอปพลิเคชันบนกรอบงานอิเล็กทรอนิกส์ถูกเรียกใช้งาน หลังจากกระบวนการหลักเริ่มทำงานแล้วกรอบงานอิเล็กทรอนิกส์สามารถเปิดหน้าต่างการทำงานขึ้นมาได้หลายหน้าต่างการทำงานซึ่งหน้าต่างการทำงานของกรอบงานอิเล็กทรอนิกส์สามารถอ้างอิงได้เป็นหนึ่งหน้าเว็บไซต์บนเว็บแอปพลิเคชัน โดยแต่ละหน้าต่างการทำงานจะมีกระบวนการนำเสนอเป็นของตัวเอง โดยกระบวนการนำเสนอจะมีความคล้ายกันกับการแสดงผลบนเว็บไซต์ซึ่งแสดงผลวัตถุต่าง ๆ บนหน้าเอชทีเอ็มแอลของเว็บเพจ



รูปที่ 1 ชนิดของกระบวนการทำงานในแอปพลิเคชันบนกรอบงานอิเล็กทรอนิกส์ [5]

กระบวนการทำงานหลักจะเป็นตัวสื่อสารและควบคุมกระบวนการนำเสนอแต่ละกระบวนการ และความแตกต่างของกระบวนการสองชนิดนี้คือ หน้าที่ของกระบวนการหลักโดยเป็นตัวควบคุมวงจรชีวิตของแอปพลิเคชัน ซึ่งทำหน้าที่ติดต่อกับระบบปฏิบัติการแต่กระบวนการนำเสนอจะรับผิดชอบในส่วนของการจัดแสดงส่วนต่อประสานผู้ใช้งาน

ปกติแล้วการพัฒนาเดสก์ท็อปแอปพลิเคชันรูปแบบเก่าจะต้องใช้ทีมนักพัฒนาแบ่งออกเป็น 3 ทีมสำหรับแต่ละระบบปฏิบัติการ หมายความว่า การพัฒนาจะต้องมีซอร์สโค้ด (Source Code) 3 เวอร์ชัน ซึ่งนอกจากการพัฒนาเดสก์ท็อปแอปพลิเคชันด้วยกรอบงานอิเล็กทรอนิกส์จะพัฒนาด้วยซอร์สโค้ดเพียงชุดเดียวแล้ว การพัฒนาแบบนี้ยังมีความสามารถในการบำรุงรักษา (Maintainability) มากกว่าการพัฒนาแบบเดิมโดยลดระยะเวลาในการพัฒนาและค่าใช้จ่ายสำหรับการพัฒนาอีกด้วย

2.2 งานวิจัยที่เกี่ยวข้อง

2.2.1 WAIL: Collection-Based Personal Web Archiving

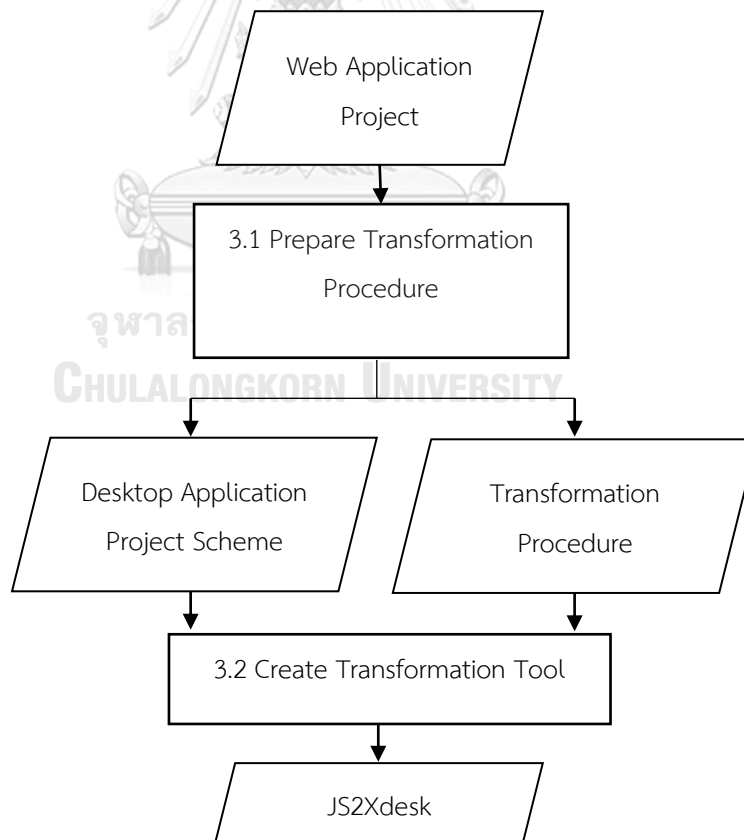
งานวิจัย [6] ได้นำเสนอ Web Archiving Integration Layer (WAIL) เป็นเดสก์ท็อปแอปพลิเคชันที่ถูกเขียนโดยเทคโนโลยีไพทอน (Python) สำหรับทำการจัดเก็บเว็บให้เป็นไฟล์เอกสารแบบบออปไฟล์ โดยผู้วิจัยได้ทำการสร้างแอปพลิเคชันใหม่อีกครั้งและทำการเพิ่มความสามารถให้กับเครื่องมือ ซึ่งผู้วิจัยได้นำเทคโนโลยีของกรอบงานอิเล็กทรอนิกส์เพื่อทำการพัฒนาเครื่องมือเวอร์ชันใหม่โดยมีวัตถุประสงค์ในการยกระดับความสามารถของของเครื่องมือโดยใช้เว็บเทคโนโลยี เช่น จาวาสคริปต์และโครเมียม และใช้กรอบงานอิเล็กทรอนิกส์ในการเพิ่มช่องทางสำหรับศักยภาพที่จะพัฒนาได้ในรูปแบบแอปพลิเคชันที่สามารถทำงานได้อย่างอิสระแบบข้ามแพลตฟอร์ม(stand-alone cross-platform native application) ซึ่งมีความสอดคล้องกันระหว่างระบบปฏิบัติการวินโดวส์ แมคโอเอส และลินุกซ์ นอกจากนี้ยังมีการใช้ประโยชน์จากกรอบงานอิเล็กทรอนิกส์โดยทำการสังเกตการณ์และตรวจจับการร้องขอและการตอบสนองเอชทีทีพี (HTTP Requests and HTTP Responses) เพื่อใช้เป็นข้อมูลสำหรับการวิจัยอีกด้วย

ผลลัพธ์ของงานวิจัยแสดงให้เห็นแอปพลิเคชันสำหรับจัดเก็บเว็บในรูปแบบเดสก์ท็อป โดยทำการใส่ข้อมูลนำเข้าเป็นที่อยู่ของเว็บไซต์(url) ผ่านทางส่วนต่อประสานผู้ใช้งานที่ถูกพัฒนาด้วยกรอบงานอิเล็กทรอนิกส์ โดยมีการตรวจสอบคุณภาพของเครื่องมือบนระบบปฏิบัติการอูบันตุ(Ubuntu) ซึ่งเป็นระบบปฏิบัติการลินุกซ์ชนิดหนึ่ง เพื่อแสดงผลการเปรียบเทียบกับแอปพลิเคชันเวอร์ชันก่อนหน้านี้ที่ทำงานโดยใช้เวลา 9 นาที 45 วินาที ในขณะที่การทำงานบนแอปพลิเคชันบนพื้นฐานของเบราวเซอร์พัฒนาโดยอิเล็กทรอนิกส์ใช้เวลาทำงาน 1 นาที 30 วินาที

บทที่ 3

แนวคิดและวิธีวิจัย

งานวิจัยนี้นำเสนอแนวทางการแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้กลายเป็นเดสก์ท็อปและเครื่องมือสำหรับการแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้กลายเป็นเดสก์ท็อปแบบอัตโนมัติ มีภาพรวมของแนวคิดและวิธีวิจัยแสดงดังรูปที่ 2 โดยจะแบ่งเป็นสองขั้นตอนหลักซึ่งในขั้นตอนแรกจะเป็นขั้นตอนการแปลงเว็บแอปพลิเคชันให้เป็นเดสก์ท็อปแอปพลิเคชันโดยมีนักพัฒนาเป็นผู้เกี่ยวข้องและจะอธิบายในหัวข้อ 3.1 เตรียมกระบวนการแปลง(Prepare Transformation Procedure) ซึ่งแนวคิดสำหรับงานวิจัยชิ้นนี้จะรับข้อมูลนำเข้าเป็นเว็บแอปพลิเคชันที่พัฒนาด้วยรีแอคต์เจเอส ส่วนในขั้นตอนที่สองจะนำความรู้ที่ได้จากขั้นตอนที่หนึ่งมาจัดทำเครื่องมือในการแปลงเว็บแอปพลิเคชันแบบอัตโนมัติให้ได้ผลลัพธ์เป็นเครื่องมือบรรทัดคำสั่ง(Command Line Tool) ที่เรียกว่า JS2Xdesk ซึ่งจะถูกอธิบายในหัวข้อ 3.2 จัดทำเครื่องมือการแปลง(Create Transformation Tool) ต่อไป



รูปที่ 2 ภาพรวมของงานวิจัย

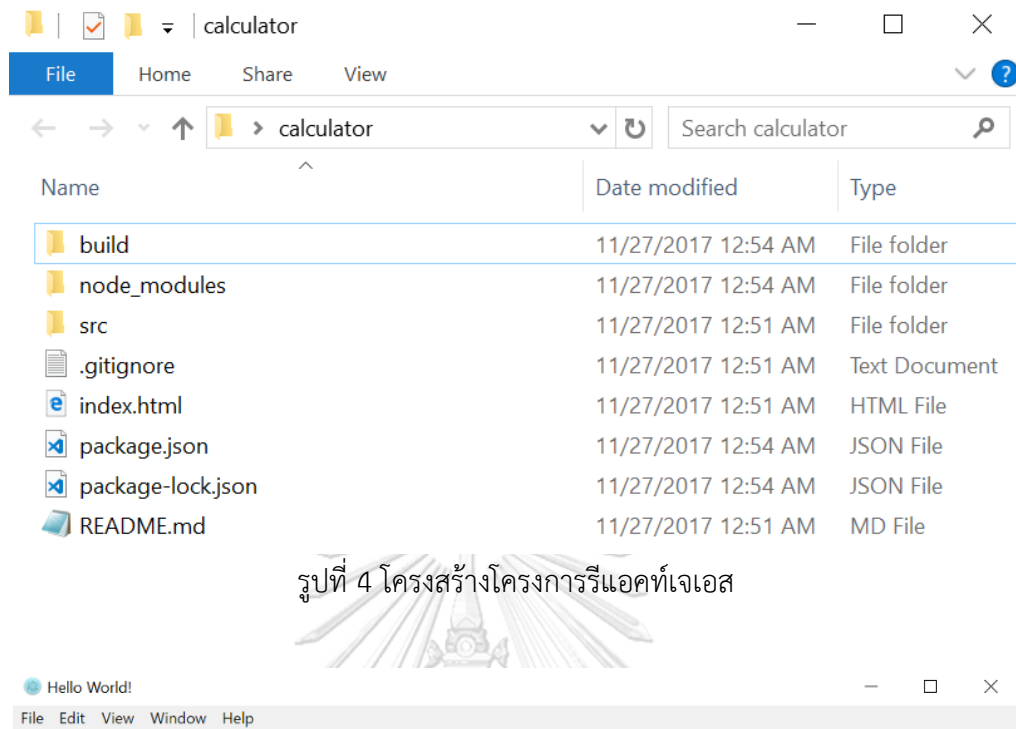
3.1 เตรียมกระบวนการงานการแปลง (Prepare Transformation Procedure)

3.1.1 แปลงโครงการพัฒนาเว็บแอปพลิเคชันตัวอย่าง

เนื่องด้วยเว็บไซต์ทางการของรีแอคท์เจเอสมีโครงการตัวอย่างพร้อมซอร์สโค้ด [7] เพื่อให้ นักพัฒนาที่สนใจนำมาศึกษาได้ในรูปแบบโอเพนซอร์ส จึงเป็นแหล่งข้อมูลที่น่าเชื่อถือสำหรับการ นำมาทดลองแปลงโครงการได้ ตัวอย่างเว็บแอปพลิเคชันที่จะทำการทดลองแสดงดังรูปที่ 3 และมี โครงสร้างโครงการรีแอคท์เจเอส ดังรูปที่ 4 จากนั้นนำโครงการตัวอย่างจากเว็บไซต์ของกรอบงาน อิเล็กตรอน [8] มาทดลองเปิดการทำงานซึ่งเป็นเดสก์ท็อปแอปพลิเคชันดังตัวอย่างดังรูปที่ 5 และมี โครงสร้างโครงการอิเล็กตรอนดังรูปที่ 6



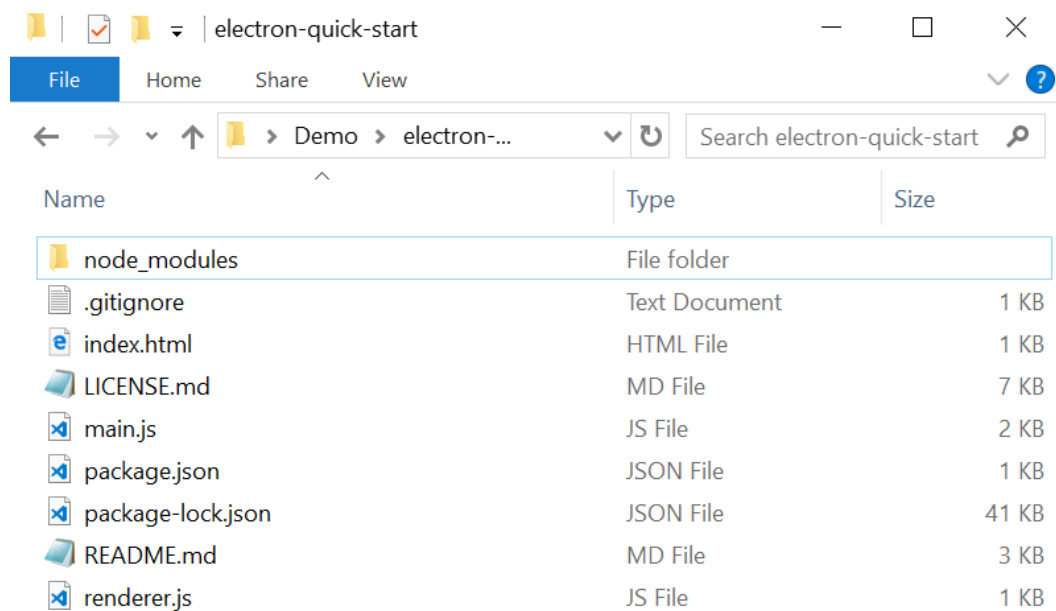
รูปที่ 3 ตัวอย่างเว็บแอปพลิเคชันที่จะทำการทดลอง



Hello World!

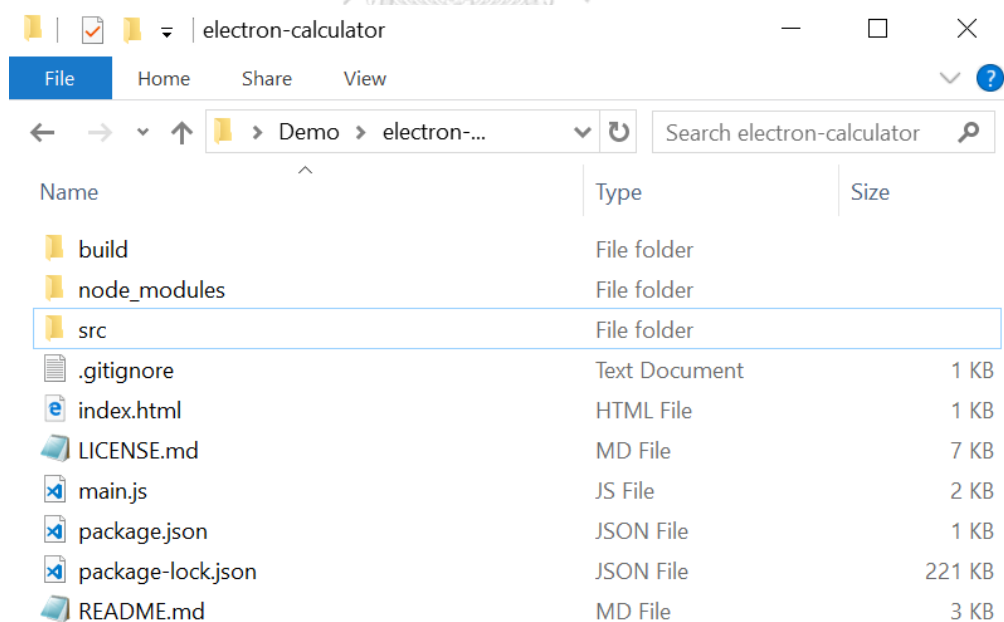
We are using Node.js 7.9.0, Chromium 58.0.3029.110, and Electron 1.7.9.

รูปที่ 5 เดสก์ท็อปแอปพลิเคชันตัวอย่าง



รูปที่ 6 โครงสร้างโครงการอิเล็กทรอนิกส์

ขั้นตอนต่อไปจึงทำการสร้างโครงการพัฒนาขึ้นโดยใช้ไฟล์ในโครงการอิเล็กทรอนิกส์รวมเข้าด้วยกันกับไฟล์ซอร์สโค้ดของโครงการรีแอกต์เจเอสกลายเป็นโครงการที่มีโครงสร้างดังรูปที่ 7



รูปที่ 7 โครงสร้างโครงการหลังทำการทดลอง

จากนั้นทำการรวมคำสั่งสำหรับโครงการพัฒนาหรือสคริปต์ (Script) และโหนดโมดูล (Node module) ที่จำเป็นในการพัฒนาโครงการหรือความขึ้นต่อกัน (Dependency) ของโครงการพัฒนารีแอคท์เจเอสดังรูปที่ 8 รวมเข้าด้วยกันกับสคริปต์และความขึ้นต่อกันของโครงการอิเล็กทรอนิกส์ดังรูปที่ 9 เป็นไฟล์แพ็คเกจใหม่ดังรูปที่ 10

```
{ } package.json x
1  {
2    "name": "calculator",
3    "version": "0.1.0",
4    "private": true,
5    "homepage": "http://ahfarmer.github.io/calculator",
6    "devDependencies": {
7      "gh-pages": "^0.11.0",
8      "react-scripts": "0.4.1"
9    },
10   "dependencies": {
11     "big.js": "^3.1.3",
12     "github-fork-ribbon-css": "^0.2.1",
13     "react": "^15.3.1",
14     "react-dom": "^15.3.1"
15   },
16   "scripts": {
17     "start": "react-scripts start",
18     "build": "react-scripts build",
19     "test": "react-scripts test --env=jsdom",
20     "eject": "react-scripts eject",
21     "deploy": "gh-pages -d build"
22   },
23   "eslintConfig": {
24     "extends": "./node_modules/react-scripts/config/eslint.js"
25   }
26 }
```

รูปที่ 8 สคริปต์และความขึ้นต่อกันของโครงการพัฒนารีแอคท์เจเอส


```

1  {
2    "name": "electron-quick-start",
3    "version": "1.0.0",
4    "description": "A minimal Electron application",
5    "main": "main.js",
6    "scripts": {
7      "start": "electron ."
8    },
9    "repository": "https://github.com/electron/electron-quick-start",
10   "keywords": [
11     "Electron",
12     "quick",
13     "start",
14     "tutorial",
15     "demo"
16   ],
17   "author": "GitHub",
18   "license": "CC0-1.0",
19   "devDependencies": {
20     "electron": "~1.8.2"
21   }
22 }

```

รูปที่ 9 สคริปต์และความขึ้นต่อกันของโครงการพัฒนาริอีเล็กตรอน

ในรูปที่ 10 มีข้อสังเกตอีกหนึ่งอย่างคือ มีการเพิ่มการระบุไฟล์เมน (“main”: “main.js”) เนื่องจากในการทำงานของกรอบงานอีเล็กตรอนจำเป็นต้องใช้ไฟล์เมนสำหรับการทำงาน จึงจำเป็นต้องระบุที่อยู่ของไฟล์เมนลงไปไฟล์แพ็คเกจ โดยเมื่อสั่งให้อีเล็กตรอนทำงานกรอบงานจะมาหาที่อยู่ของไฟล์เมนจากไฟล์แพ็คเกจ หากตรวจไม่พบจะไม่สามารถเปิดการทำงานได้ โดยจากในรูปจะระบุว่าไฟล์เมนอยู่ในโฟลเดอร์เดียวกับไฟล์แพ็คเกจดังรูปที่ 7 โดยหน้าที่เบื้องต้นของไฟล์เมนสามารถยกตัวอย่างได้เช่น ระบุขนาดของแอปพลิเคชันในหน่วยพิกเซล ระบุที่อยู่ของไฟล์เอชทีเอ็มแอลของแอปพลิเคชัน ตัวอย่างซอร์สโค้ดในไฟล์เมนในการแปลงแสดงได้ดังรูปที่ 11

```
{ } package.json x
1  {
2    "name": "electron-calculator",
3    "version": "1.0.0",
4    "main": "main.js",
5    "devDependencies": {
6      "gh-pages": "^0.11.0",
7      "react-scripts": "0.4.1",
8      "electron": "~1.8.2"
9    },
10   "dependencies": {
11     "big.js": "^3.1.3",
12     "github-fork-ribbon-css": "^0.2.1",
13     "react": "^15.3.1",
14     "react-dom": "^15.3.1"
15   },
16   "scripts": {
17     "electron-start": "electron .",
18     "start": "react-scripts start",
19     "build": "react-scripts build",
20     "test": "react-scripts test --env=jsdom",
21     "eject": "react-scripts eject",
22     "deploy": "gh-pages -d build"
23   }
24 }
25 |
```

รูปที่ 10 สคริปต์และความขึ้นต่อกันหลังจากทำการรวม

```

JS main.js  X
1  const electron = require('electron')
2  const app = electron.app
3  const BrowserWindow = electron.BrowserWindow
4  const path = require('path')
5  const url = require('url')
6
7  let mainWindow
8
9  function createWindow () {
10
11     mainWindow = new BrowserWindow({width: 800, height: 600})
12
13     mainWindow.loadURL(url.format({
14         pathname: path.join(__dirname, 'index.html'),
15         protocol: 'file:',
16         slashes: true
17     })))
18
19     // Open the DevTools.
20     mainWindow.webContents.openDevTools()
21
22     // Emitted when the window is closed.
23     mainWindow.on('closed', function () {
24         // Dereference the window object, usually you would store windows
25         // in an array if your app supports multi windows, this is the time
26         // when you should delete the corresponding element.
27         mainWindow = null
28     })
29 }
30

```

รูปที่ 11 ตัวอย่างซอร์สโค้ดในไฟล์เมน

จากรูปที่ 11 สังเกตได้ว่าการกำหนดให้ขนาดของแอปพลิเคชันมีขนาดกว้าง 800 พิกเซล และสูง 600 พิกเซลจากซอร์สโค้ด

```
mainWindow = new BrowserWindow({width: 800, height: 600})
```

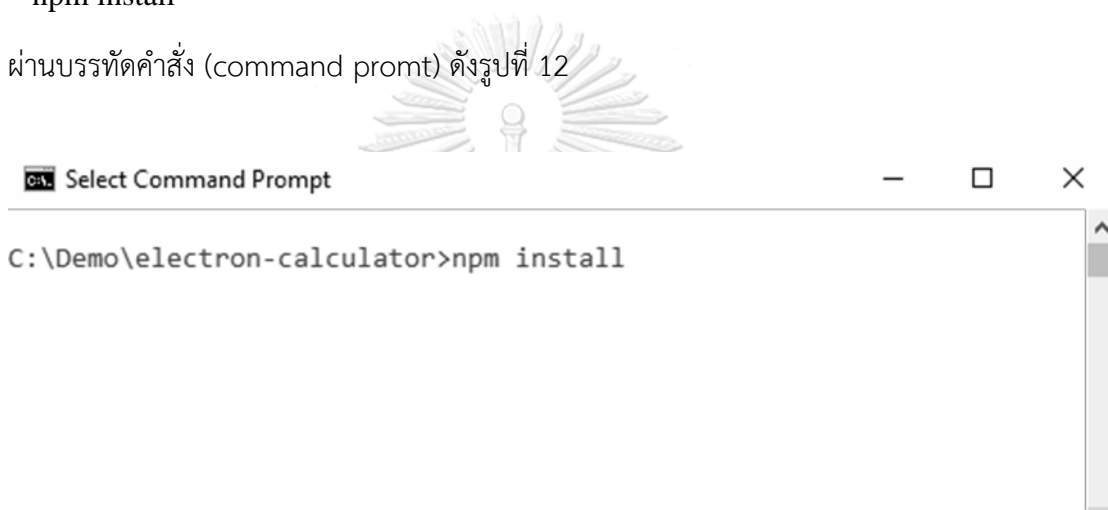
และระบุตำแหน่งของไฟล์เอชทีเอ็มแอลสำหรับแอปพลิเคชัน ซึ่งจากตัวอย่างหมายความว่าไฟล์เอชทีเอ็มแอลที่จะต้องใช้จะอยู่ในโฟลเดอร์เดียวกันกับโฟลเดอร์ของโครงการพัฒนาเสมอจากซอร์สโค้ด โดยนอกจากคำสั่งที่ยกตัวอย่างมาข้างต้นแล้วยังมีคำสั่งสำหรับช่วยนักพัฒนาในการพัฒนาเดสก์ท็อปแอปพลิเคชันอีกหลายคำสั่ง โดยทางผู้พัฒนารอบงานอิเล็กทรอนิกส์อนได้มีการอธิบายคำสั่งทั้งหมดที่สามารถใช้ได้ไว้บนเว็บไซต์ทางการของอิเล็กทรอนิกส์อน [9] อยู่ในรูปแบบของ API (Application Programming Interface) ของกรอบงาน

```
mainWindow.loadURL(url.format({
  pathname: path.join(__dirname, 'index.html'),
  protocol: 'file:',
  slashes: true
})))
```

หลังจากรวมไฟล์เสร็จสิ้นต้องทำการติดตั้งโมดูลความขึ้นต่อกันซึ่งจำเป็นต่อการทำงานของแอปพลิเคชันในขั้นต่อไปโดยป้อนคำสั่ง

npm install

ผ่านบรรทัดคำสั่ง (command prompt) ดังรูปที่ 12

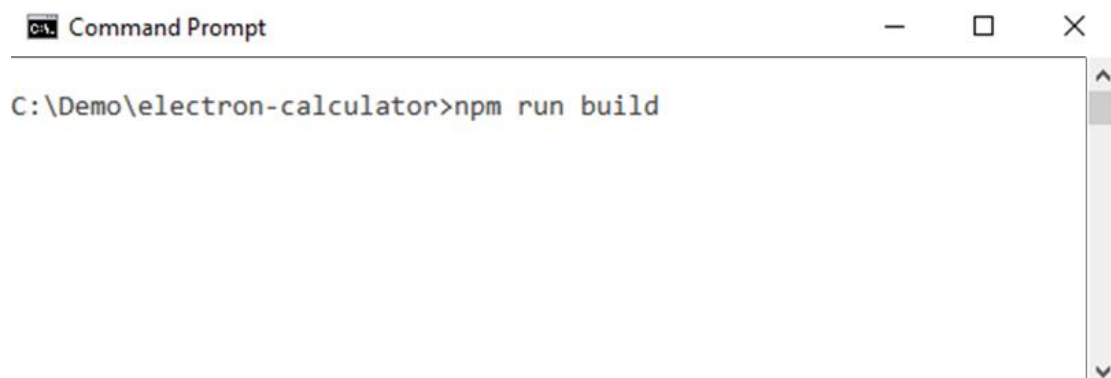


รูปที่ 12 การติดตั้งโมดูลความขึ้นต่อกัน

เนื่องจากโครงการพัฒนารีแอกต์เจเอสตัวอย่างมีการใช้เครื่องมือช่วยในการสร้างไฟล์เจเอสแบบคงที่ซึ่งรวมไปถึงการรวมซอร์สโค้ดต่าง ๆ ในโครงการพัฒนากตัวอย่างเช่น ในขั้นตอนการพัฒนา นักพัฒนาจะแบ่งตรรกะการทำงานไว้เป็นไฟล์ย่อย ๆ หรืออยู่ในแต่ละโฟลเดอร์ย่อย ๆ แล้วจึงเรียกใช้งานเพื่อความสะดวกในการอ่านและแก้ไขรวมไปถึงไฟล์ซีเอสเอสที่ใช้สำหรับการตกแต่งหน้าตาของเว็บไซต์ซึ่งถูกเรียกใช้งานในแต่ละหน้าโดยมีการเขียนแยกไฟล์กันตามแต่ละหน้าเว็บไซต์ โดยเครื่องมือนี้จะช่วยรวมซอร์สโค้ดในแต่ละไฟล์จาวาสคริปต์และซีเอสเอสที่มีการเรียกการใช้งานในโครงการให้มาอยู่ในไฟล์จาวาสคริปต์และไฟล์ซีเอสเอสแบบไฟล์เดียวซึ่งจะถูกสร้างไว้ในโฟลเดอร์บิลด์ (build) โดยผู้วิจัยเลือกใช้งานสำหรับการใช้งานร่วมกันกับการทำงานของอิเล็กทรอนิกส์ในงานวิจัยชิ้นนี้ โดยคำสั่งดังกล่าวใช้งานโดยป้อนคำสั่ง

npm run build

ผ่านบรรทัดคำสั่งดังรูปที่ 13 จากนั้นเอ็นพีเอ็มจะเรียกหาคำสั่งบิลด์ที่ถูกเขียนไว้ในไฟล์แพ็คเกจจากรูปที่ 10



```

Command Prompt
C:\Demo\electron-calculator>npm run build
  
```

รูปที่ 13 การป้อนคำสั่งบิลด์

หลังจากได้ไฟล์จาวาสคริปต์และไฟล์ซีเอสเอสแล้วจึงนำไปเพิ่มในไฟล์เอชทีเอ็มแอลของโครงการใหม่ดังรูปที่ 14



```

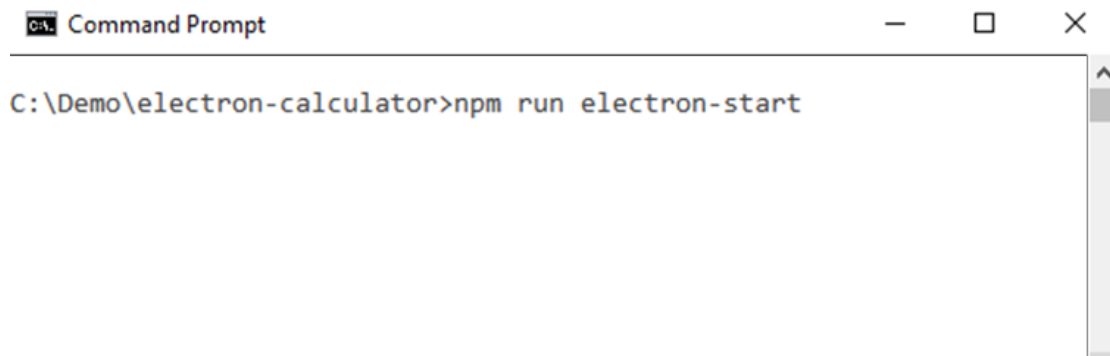
<> index.html x
1  <!doctype html>
2  <html lang="en">
3    <head>
4      <meta charset="utf-8">
5      <meta name="viewport" content="width=device-width, user-scalabl
6      <link rel="shortcut icon" href="./src/favicon.ico">
7      <title>Calculator</title>
8      <link href="build/static/css/main.2706593e.css" rel="stylesheet
9    </head>
10   <body>
11     <div id="root"></div>
12     <script defer src="build/static/js/main.7a8d59a3.js"></script>
13   </body>
14 </html>
  
```

รูปที่ 14 ไฟล์เอชทีเอ็มแอลของโครงการใหม่

เมื่อเสร็จสิ้นขั้นตอนการแปลงโครงสร้างโครงการพัฒนาและรวมไฟล์ซอร์สโค้ด จึงทำการเปิดการทำงานเพื่อทดสอบโดยป้อนคำสั่ง

```
npm run electron-start
```

ผ่านบรรทัดคำสั่งดังรูปที่ 15 จากนั้นเอ็นพีเอ็มจะเรียกหาคำสั่งอิเล็กทรอนิกส์ที่ถูกรวบรวมไว้ในไฟล์ แพ็กเกจจากรูปที่ 10



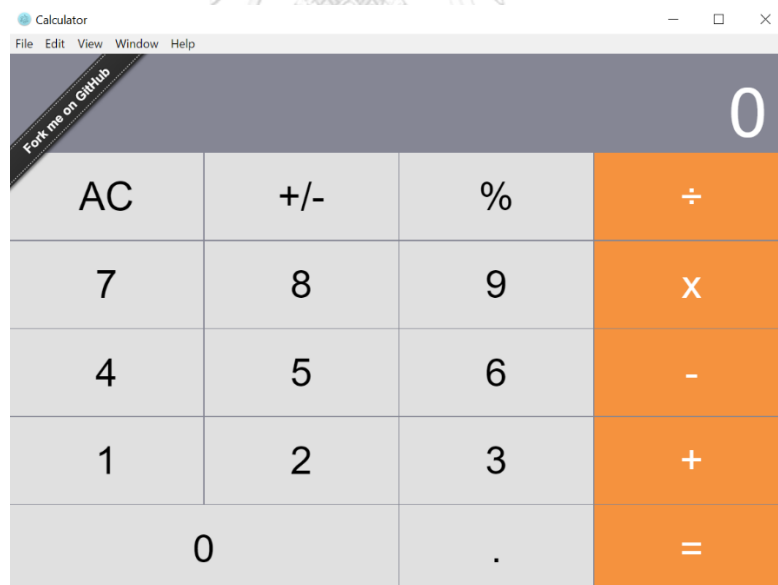
```

C:\Demo\electron-calculator>npm run electron-start

```

รูปที่ 15 การป้อนคำสั่งอิเล็กทรอนิกส์สำหรับเปิดการทำงาน

หน้าต่างแอปพลิเคชันผลลัพธ์จะปรากฏขึ้นมาแบบอัตโนมัติเป็นแอปพลิเคชันรีแอคต์เจเอสตัวอย่างซึ่งอยู่ในรูปแบบของเดสก์ท็อปแอปพลิเคชันดังรูปที่ 16 ซึ่งถูกเปิดใช้งานผ่านระบบปฏิบัติการวินโดวส์ รูปแบบของแถบคำสั่งด้านบนจะแตกต่างกันออกไปตามระบบปฏิบัติการที่ผู้ใช้ใช้งาน แต่ส่วนต่อประสานผู้ใช้งานและหลักการทำงานของแอปพลิเคชันซึ่งทำงานบนวินโดวส์จะทำงานในแบบเดียวกันสำหรับแมคโอเอสและลินุกซ์



รูปที่ 16 เดสก์ท็อปแอปพลิเคชันที่ได้จากการทดลอง

3.1.2 ผลลัพธ์จากการเตรียมกระบวนการแปลง

1) โครงสร้างโครงการพัฒนาสำหรับกรอบงานอิเล็กทรอนิกส์

หลังจากทำการทดลองแปลงโครงการพัฒนาด้วยมือสำเร็จทำให้ได้ต้นแบบโครงสร้างโครงการพัฒนาสำหรับกรอบงานอิเล็กทรอนิกส์ดังรูปที่ 17 ประกอบด้วยโฟลเดอร์บิลด์ (build) ซึ่งจัดเก็บไฟล์

app.css และ app.js จากโครงการเว็บแอปพลิเคชันไว้ โฟลเดอร์โหนดโมดูล (node_modules) ใช้จัดเก็บโมดูลความขึ้นต่อกันซึ่งมาจากโครงการทั้งสอง โฟลเดอร์เอซอร์ซี (src) ใช้เก็บไฟล์ซอร์สโค้ดต่าง ๆ ของโครงการเว็บแอปพลิเคชัน ไฟล์อินเดกซ์ (index.html) ใช้เก็บที่อยู่ของไฟล์ app.css และ app.js ไฟล์เมน (main.js) จากโครงการอิเล็กทรอนิกส์และไฟล์แพ็คเกจ (package.json) ซึ่งถูกรวมคำสั่งไว้จากทั้งสองโครงการ

```

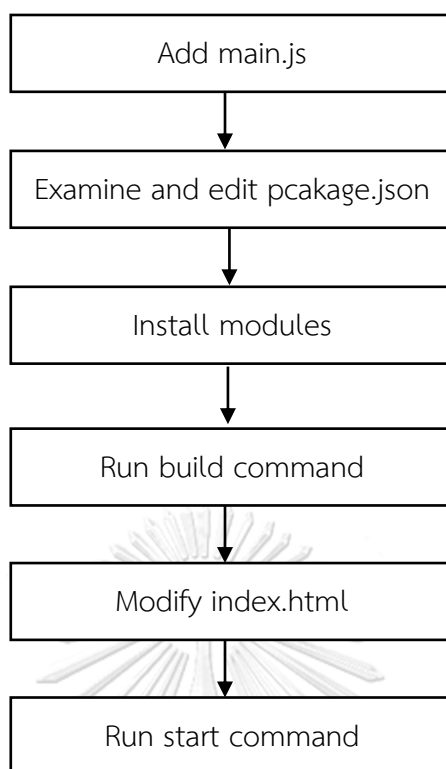
build
  static
    css
      app.css
    js
      app.js
node_modules
src
index.html
main.js
package.json

```

รูปที่ 17 โครงสร้างโครงการพัฒนาสำหรับกรอบงานอิเล็กทรอนิกส์

2) กระบวนการเปลี่ยนแปลง

หลังจากทำการทดลองแปลงโครงการพัฒนาด้วยมือสำเร็จทำให้ได้กระบวนการเปลี่ยนแปลง ซึ่งเป็นขั้นตอนต่าง ๆ ระหว่างที่ทำการแปลงตั้งแต่ขั้นตอนแรกจนถึงขั้นตอนสุดท้ายดังรูปที่ 18



รูปที่ 18 กระบวนการงานการแปลง

3.2 จัดทำเครื่องมือการแปลง (Create Transformation Tool)

สืบเนื่องจากจากขั้นตอนการเตรียมการแปลงในหัวข้อ 3.1 ทำให้ได้ผลลัพธ์เป็นโครงสร้างโครงการพัฒนาสำหรับกรอบงานอิเล็กทรอนิกส์และกระบวนการแปลงซึ่งมีความจำเป็นต่อการพัฒนาเครื่องมือแปลงอัตโนมัติ โดยมีสาเหตุเนื่องจากผลลัพธ์ที่ได้จากเครื่องมือการแปลงจะต้องมีลักษณะโครงสร้างของโครงการพัฒนาที่อยู่ในรูปแบบเดียวกันเพื่อทำให้เป็นมาตรฐานเดียวกัน (standardization) ให้กับผลลัพธ์ของเครื่องมือโดยใช้โครงสร้างโครงการพัฒนาสำหรับกรอบงานอิเล็กทรอนิกส์ที่ดังกล่าวเป็นต้นแบบ และแนวทางการปฏิบัติงานที่เครื่องมือจะต้องทำการแปลงเว็บแอปพลิเคชันโดยใช้ขั้นตอนแบบเดียวกันกับโครงการนำเข้าทุกโครงการ ซึ่งมีลักษณะเป็นสคริปต์การทำงานที่มีต้นแบบมาจากกระบวนการแปลง

ผลลัพธ์ที่ได้จากขั้นตอนนี้คือเครื่องมือแบบกระทำการได้ผ่านบรรทัดคำสั่ง (executable command line tool) ที่ใช้งานได้กับโครงการพัฒนาเว็บแอปพลิเคชันที่ใช้รีแอคต์เจเอสเป็นส่วนช่วยในการพัฒนา

รูปที่ 19 แสดงให้เห็นถึงตัวอย่างการใช้งานคอมมานด์ไลน์ (command line) เพื่อใช้เครื่องมือในการแปลงแอปพลิเคชันโดยอยู่ในรูปแบบคำสั่งซึ่งกำหนดดังต่อไปนี้

```
JS2Xdesk [main.js path] [html file path]
```


จากวิธีการแปลงด้วยมือทำให้ได้เงื่อนไขและโครงสร้างของโครงการ เครื่องมือถูกสร้างขึ้นแบบคอมมานด์ไลน์เพื่อใช้แปลงเว็บแอปพลิเคชันโดยการพัฒนาเครื่องมือใช้โหนดเจเอสและเอ็นพีเอ็ม (npm) [10]

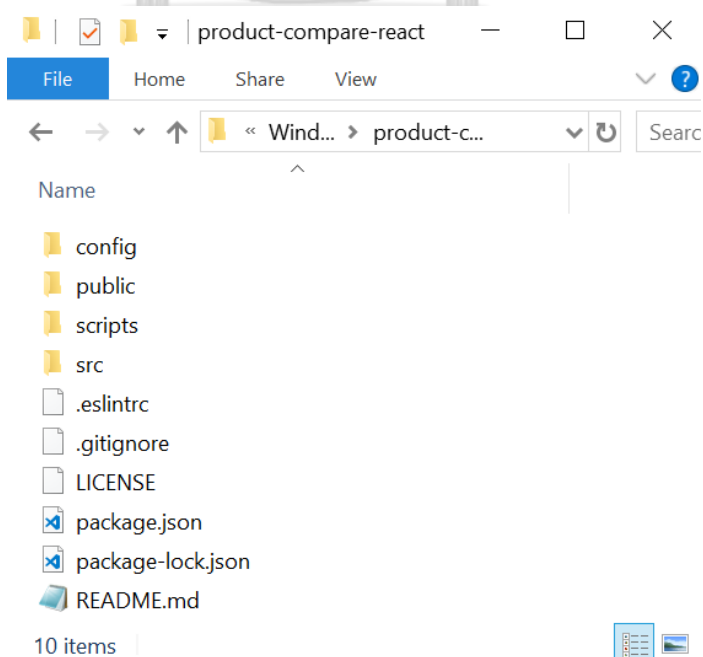


```

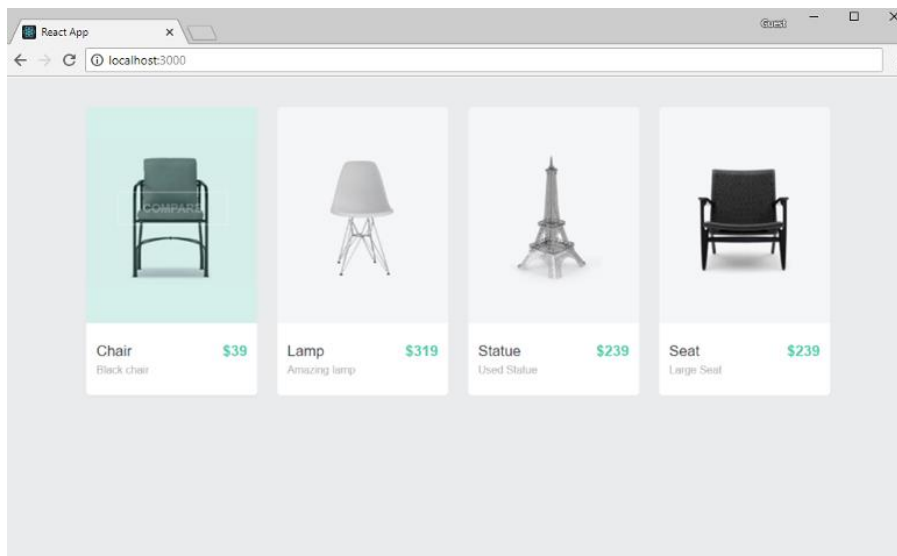
C:\Example web app>JS2Xdesk ./build/static/main.js ./index.html
  
```

รูปที่ 19 คำสั่งและตัวแปรสำหรับการแปลง

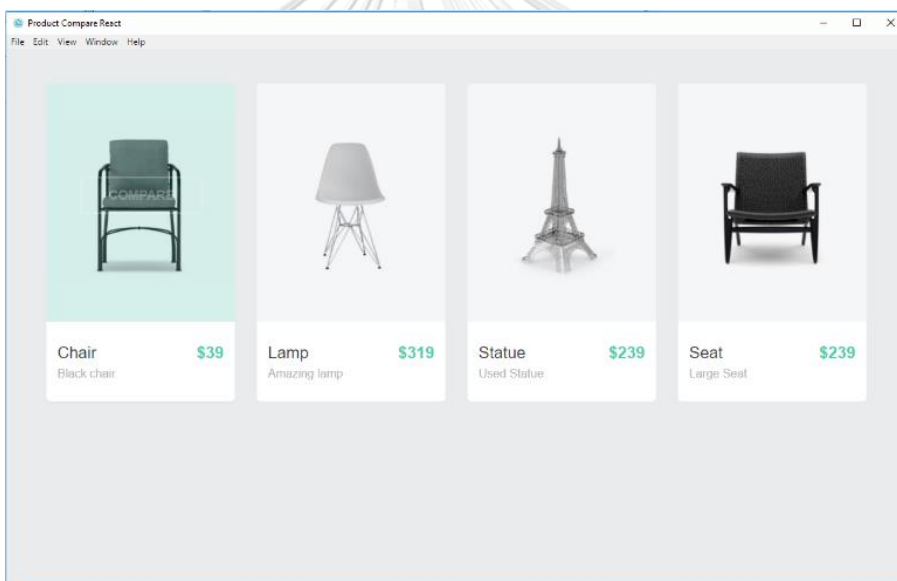
ด้วยตัวอย่างเว็บแอปพลิเคชันเปรียบเทียบข้อมูลสินค้าจากเว็บไซต์ของรีแอคท์เจเอส[7] โดยมีการแจกจ่ายซอร์สโค้ด[11]ไว้เพื่อทำการทดลอง จึงนำมาใช้ทดลองกับเครื่องมือเพื่อตรวจสอบความถูกต้องซึ่งมีข้อมูลนำเข้าเป็นโครงการพัฒนาเว็บแอปพลิเคชันแสดงดังรูปที่ 20 และมีส่วนต่อประสานผู้ใช้งานในรูปแบบเว็บแอปพลิเคชันดังรูปที่ 21 จากนั้นเครื่องมือทำงานจะแสดงให้เห็นถึงผลลัพธ์ของเครื่องมือดังแสดงในรูปที่ 22



รูปที่ 20 ข้อมูลนำเข้าที่เป็นโครงการพัฒนารีแอคท์



รูปที่ 21 ส่วนต่อประสานผู้ใช้งานเว็บแอปพลิเคชันจากข้อมูลนำเข้า



รูปที่ 22 เดสก์ท็อปแอปพลิเคชันที่ถูกแปลงโดยเครื่องมือ

จากการทดลองแปลงเว็บแอปพลิเคชันให้เป็นเดสก์ท็อป ทำให้เห็นว่าการแสดงผลของเดสก์ท็อปแอปพลิเคชันไม่มีความแตกต่างจากเว็บแอปพลิเคชัน รวมไปถึงการการทำงานต่าง ๆ ของแอปพลิเคชันทั้งสองรูปแบบก็ไม่มี ความแตกต่างเช่นเดียวกัน ซึ่งส่วนที่ไม่มีความแตกต่างจะเป็นหน้าจอที่ใช้แสดงผลและการทำงานบนหน้าจอส่วประสานของแอปพลิเคชันเท่านั้น จุดที่แตกต่างระหว่างแอปพลิเคชันสองแบบคือแถบด้านบนซึ่งใช้บอกชื่อของแอปพลิเคชัน (Title bar) และแถบการทำงานของแอปพลิเคชัน (Menu bar)

ซึ่งแนวทางในการแปลงนี้ยังมีข้อจำกัดที่มีสาเหตุมาจากเทคโนโลยีของอิเล็กทรอนิกส์ตัวอย่างเช่น ระบบการแจ้งเตือน(notification system) โดยทางผู้พัฒนาอิเล็กทรอนิกส์ได้จัดทำเป็นชุดคำสั่งที่แยกออกมาจากการแจ้งเตือนบนเว็บเบราว์เซอร์ที่ใช้เทคโนโลยีการแจ้งเตือนสำหรับเว็บ(web notification) ซึ่งทางผู้พัฒนาอิเล็กทรอนิกส์พัฒนาชุดคำสั่งสำหรับกรอบงานไว้โดยเฉพาะและได้แนะนำแนวทางในการเขียนที่จัดทำขึ้นเพื่อกรอบงานอิเล็กทรอนิกส์ในเอกสารของกรอบงานบนเว็บไซต์ทางการ [9] โดยเครื่องมือที่ถูกพัฒนานี้ยังไม่ครอบคลุมข้อจำกัดดังกล่าวในงานวิจัยนี้และสามารถจัดทำได้โดยจะต้องตรวจสอบโครงการเว็บแอปพลิเคชันและทำการเตรียมกระบวนการแปลงใหม่ในลักษณะเดียวกันกับหัวข้อ 3.1 ใหม่อีกครั้ง



บทที่ 4

การพัฒนาเครื่องมือ

รายละเอียดในบทนี้จะกล่าวถึงวิธีการออกแบบและพัฒนาเครื่องมือเพื่อใช้ในการช่วยแปลงเว็บแอปพลิเคชันโดยสร้างจากแนวคิดและเงื่อนไขที่ได้จากการทดลองแปลงด้วยมือที่ได้นำเสนอในบทที่ 3 โดยจะกล่าวถึงโครงสร้างของซอฟต์แวร์และสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ ซึ่งมีรายละเอียดดังต่อไปนี้

4.1 ความต้องการเชิงฟังก์ชัน

- 1) ระบบสามารถอ่านที่อยู่ของไฟล์ซึ่งเป็นตัวแปรนำเข้าผ่านคอมมานด์ไลน์ได้
- 2) ระบบสามารถสำรองไฟล์ทั้งหมดไว้เป็นสำเนาก่อนทำการแปลงได้
- 3) ระบบสามารถรวมสคริปต์และความขึ้นต่อกันของเว็บแอปพลิเคชันเข้าด้วยกันกับสคริปต์และความขึ้นต่อกันของระบบได้
- 4) ระบบสามารถติดตั้งโมดูลสำหรับกรอบงานอิเล็กทรอนิกส์ได้
- 5) ระบบสามารถแสดงผลการแปลงเว็บแอปพลิเคชันในรูปแบบของเดสก์ท็อปแอปพลิเคชันได้

4.2 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือสนับสนุน

สภาพแวดล้อมที่ใช้ในการพัฒนาระบบจะอ้างอิงมาจากคอมพิวเตอร์ที่ใช้ในการพัฒนา โดยประกอบไปด้วยฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ที่ใช้ในการพัฒนาระบบ ซึ่งมีรายละเอียดดังนี้

1) ระบบฮาร์ดแวร์

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบควรมีฮาร์ดแวร์ขั้นต่ำดังต่อไปนี้

- หน่วยการประมวลผล (CPU) ซีพียูอินเทลคอร์ไอ 7 ความเร็ว 2.4 กิกะเฮิรตซ์ (intel(R) Core(TM) i7 CPU 2.4 HZ)
- หน่วยความจำสำรอง (Memory) ความเร็ว 8 กิกะไบต์ (Ram 8 GB)
- จานบันทึกแบบแข็ง (Hardisk) ความจุ 500 กิกะไบต์ (Hardisk 500 GB)

2) ซอฟต์แวร์

เครื่องคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบมีซอฟต์แวร์ดังต่อไปนี้

- ระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 10 โฮม 64 บิต (Microsoft Windows 10 Home 64x bit)
- โหนดเจเอส (Node.js)
- เอ็นพีเอ็ม (npm)
- โปรแกรมวิซวลสตูดิโอโค้ด (Visual Studio Code)

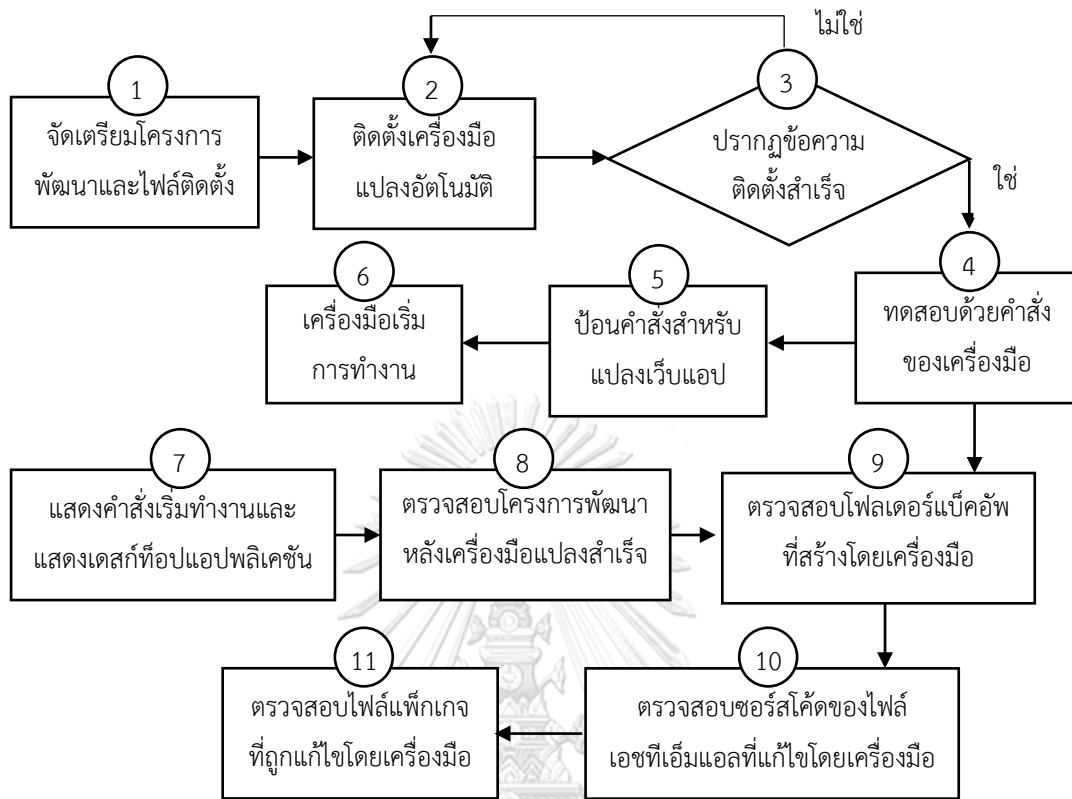
3) การติดตั้งซอฟต์แวร์

ทำการติดตั้งเครื่องมือในการพัฒนาระบบทั้งหมดลงในเครื่องคอมพิวเตอร์ที่ใช้พัฒนาระบบ โดยเริ่มลำดับการติดตั้งตามขั้นตอนดังต่อไปนี้

- 3.1) ติดตั้งระบบปฏิบัติการไมโครซอฟต์วินโดวส์ 10 โฮม 64
- 3.2) ติดตั้งโปรแกรมโนดเจเอส
- 3.3) ติดตั้งโปรแกรมเอ็นพีเอ็ม
- 3.4) ติดตั้งโปรแกรมวิซวลสตูดิโอโค้ด

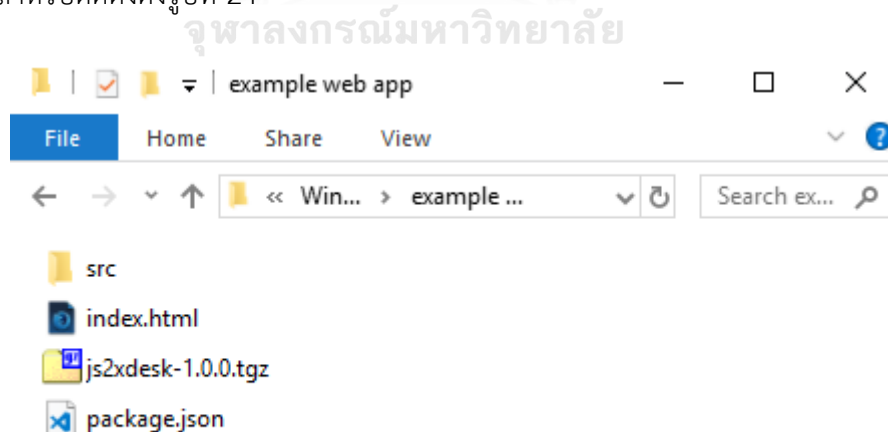
4.3 ขั้นตอนการทำงานของเครื่องมือ

ในส่วนนี้จะได้สาริตขั้นตอนการจัดเตรียม การใช้เครื่องมือด้วยเว็บแอปพลิเคชันเครื่องคิดเลขจากเว็บของรีแอคท์เจเอส[7] การตรวจสอบข้อมูลหลังทำการแปลงโดยใช้เครื่องมืออัตโนมัติ โดยมีขั้นตอนการทำงานและภาพส่วนต่อประสานดังผังงานรูปที่ 23 ซึ่งก่อนที่จะเริ่มการทำงานนักพัฒนาจะต้องติดตั้งโนดเจเอสไว้บนเครื่องที่จะใช้งานเครื่องมือไว้ก่อนดังรายละเอียดอธิบายในภาคผนวก



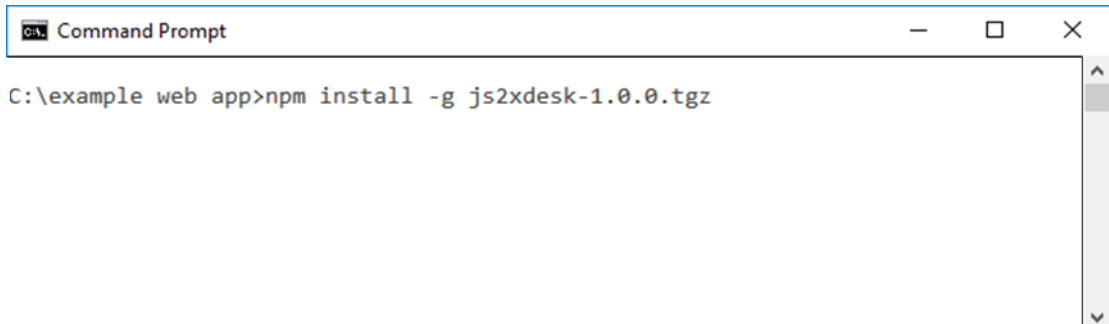
รูปที่ 23 ผังงานการสาธิตใช้เครื่องมือแปลงอัตโนมัติ

- 1) ผู้ใช้จัดเตรียมโครงการพัฒนาเว็บแอปพลิเคชันและไฟล์เครื่องมือ (js2xdesk-1.0.0.tgz) สำหรับติดตั้งดังรูปที่ 24



รูปที่ 24 การจัดเตรียมโครงการพัฒนาเว็บแอปพลิเคชันและไฟล์เครื่องมือ

- 2) ผู้ใช้เปิดบรรทัดคำสั่งในโพลเดอร์ที่มีไฟล์ติดตั้งอยู่และทำการติดตั้งเครื่องมือด้วยเอ็นพีเอ็มแบบทั่วไป (global) ด้วยคำสั่งดังรูปที่ 25



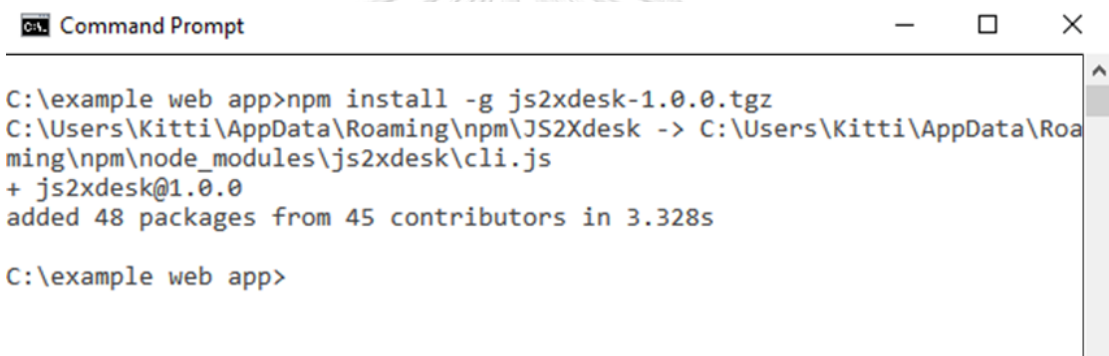
```

C:\example web app>npm install -g js2xdesk-1.0.0.tgz

```

รูปที่ 25 การติดตั้งเครื่องมือ

- 3) เมื่อติดตั้งสำเร็จจะปรากฏข้อความติดตั้งสำเร็จดังรูปที่ 26



```

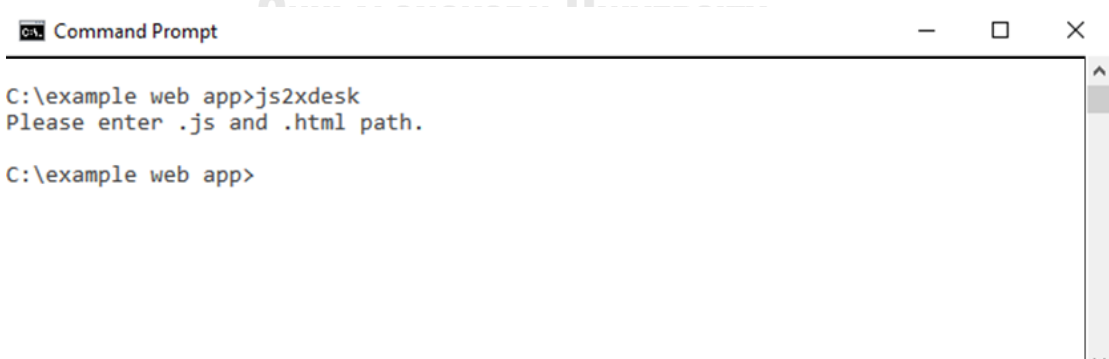
C:\example web app>npm install -g js2xdesk-1.0.0.tgz
C:\Users\Kitti\AppData\Roaming\npm\JS2Xdesk -> C:\Users\Kitti\AppData\Roaming\npm\node_modules\js2xdesk\cli.js
+ js2xdesk@1.0.0
added 48 packages from 45 contributors in 3.328s

C:\example web app>

```

รูปที่ 26 การติดตั้งเครื่องมือสำเร็จ

- 4) ทดสอบด้วยคำสั่งของเครื่องมือ ผู้จะใช้ได้การแจ้งเตือนให้ใส่ตัวแปรให้ครบดังรูปที่ 27



```

C:\example web app>js2xdesk
Please enter .js and .html path.

C:\example web app>

```

รูปที่ 27 ทดสอบการรับคำสั่งของเครื่องมือ

- 5) ป้อนคำสั่งสำหรับแปลงเว็บแอปพลิเคชันพร้อมตัวแปรด้วยคำสั่ง ดังรูปที่ 28 เพื่อเรียกใช้เครื่องมือ

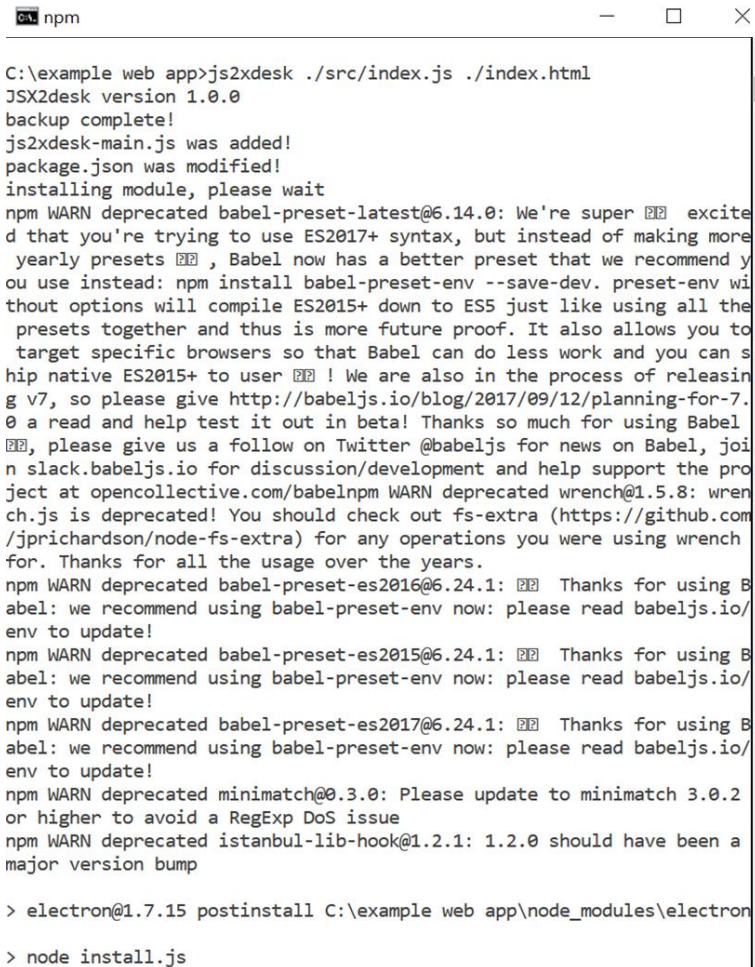
```
js2xdesk ./src/index.js ./index.html
```



```
Command Prompt
c:\example web app>js2xdesk ./src/index.js ./index.html
```

รูปที่ 28 การใช้เครื่องมือแปลงเว็บแอปพลิเคชัน

- 6) เมื่อเครื่องมือรับคำสั่งถูกต้อง เครื่องมือจะเริ่มการทำงานดังรูปที่ 29 โดยการทำงานของเครื่องมือจะเริ่มจากการสำรองข้อมูล เพิ่มไฟล์เมนไปทีโฟลเดอร์ รวมไฟล์แพ็คเกจของเว็บแอปพลิเคชันกับไฟล์แพ็คเกจของเครื่องมือ ติดตั้งโมดูลความขึ้นต่อกัน บิลด์ แก้ไขไฟล์เอชทีเอ็มแอล และเริ่มการทำงานเดสก์ท็อปแอปพลิเคชันด้วยคำสั่งของอิเล็กทรอนิกส์ตามลำดับ



```
npm
C:\example web app>js2xdesk ./src/index.js ./index.html
JSX2desk version 1.0.0
backup complete!
js2xdesk-main.js was added!
package.json was modified!
installing module, please wait
npm WARN deprecated babel-preset-latest@6.14.0: We're super excited that you're trying to use ES2017+ syntax, but instead of making more yearly presets, Babel now has a better preset that we recommend you use instead: npm install babel-preset-env --save-dev. preset-env without options will compile ES2015+ down to ES5 just like using all the presets together and thus is more future proof. It also allows you to target specific browsers so that Babel can do less work and you can ship native ES2015+ to users! We are also in the process of releasing v7, so please give http://babeljs.io/blog/2017/09/12/planning-for-7.0 a read and help test it out in beta! Thanks so much for using Babel. Please give us a follow on Twitter @babeljs for news on Babel, join slack.babeljs.io for discussion/development and help support the project at opencollective.com/babelnpm WARN deprecated wrench@1.5.8: wrench.js is deprecated! You should check out fs-extra (https://github.com/jprichardson/node-fs-extra) for any operations you were using wrench for. Thanks for all the usage over the years.
npm WARN deprecated babel-preset-es2016@6.24.1: Thanks for using Babel: we recommend using babel-preset-env now: please read babeljs.io/env to update!
npm WARN deprecated babel-preset-es2015@6.24.1: Thanks for using Babel: we recommend using babel-preset-env now: please read babeljs.io/env to update!
npm WARN deprecated babel-preset-es2017@6.24.1: Thanks for using Babel: we recommend using babel-preset-env now: please read babeljs.io/env to update!
npm WARN deprecated minimatch@0.3.0: Please update to minimatch 3.0.2 or higher to avoid a RegExp DoS issue
npm WARN deprecated istanbul-lib-hook@1.2.1: 1.2.0 should have been a major version bump
> electron@1.7.15 postinstall C:\example web app\node_modules\electron
> node install.js
```

รูปที่ 29 ตัวอย่างการทำงานของเครื่องมือ

- 7) แสดงคำสั่งเริ่มทำงานและแสดงเดสก์ท็อปปแอปพลิเคชันดังรูปที่ 30 ซึ่งเดสก์ท็อปปแอปพลิเคชันมีส่วนต่อประสานดังรูปที่ 16



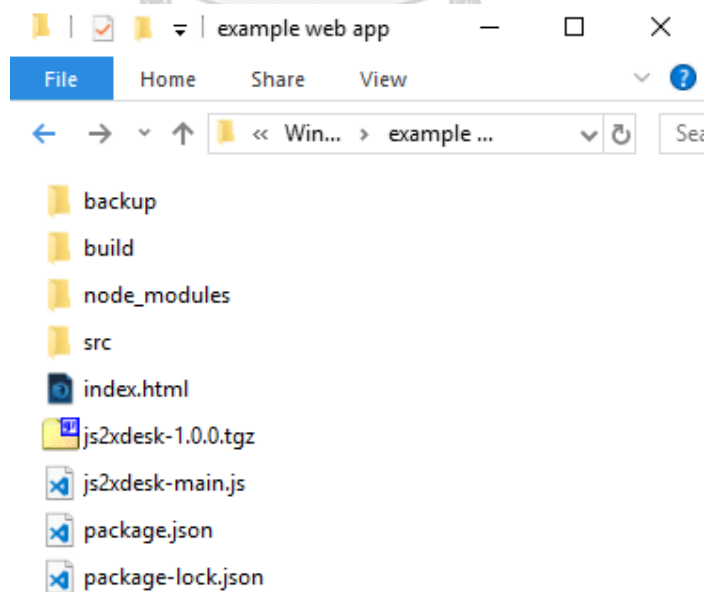
```

C:\ npm
.html was modified!
> calculator@0.1.0 js2xdesk-start C:\example web app
> electron .

```

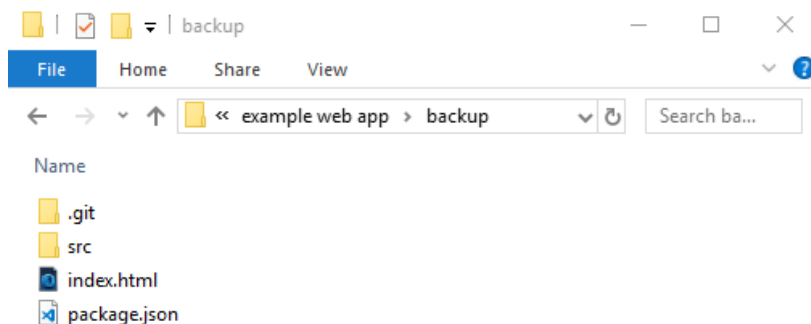
รูปที่ 30 ข้อความสุดท้ายที่เครื่องมือแสดง

- 8) ตรวจสอบโครงการพัฒนาหลังเครื่องมือแปลงสำเร็จดังรูปที่ 31 โดยเครื่องมือจะสร้าง
- 1) โฟลเดอร์แบ็คอัป (backup) เพื่อสำรองข้อมูลก่อนทำการแปลง
 - 2) โฟลเดอร์โหนดโมดูลจากคำสั่งการติดตั้งโดยเครื่องมือ
 - 3) ไฟล์เมน (js2xdesk-main.js) สำหรับการทำงานของกรอบงานอิเล็กทรอนิกส์



รูปที่ 31 รายการไฟล์และโฟลเดอร์หลังจากใช้เครื่องมือ

9) ตรวจสอบโฟลเดอร์แบ็คอัพที่สร้างโดยเครื่องมือดังรูปที่ 32



รูปที่ 32 รายการไฟล์และโฟลเดอร์ที่ถูกสำรองข้อมูล

10) ตรวจสอบซอร์สโค้ดของไฟล์เอชทีเอ็มแอลที่แก้ไขโดยเครื่องมือดังรูปที่ 33

```

<> index.html x
1 | <!DOCTYPE html><html lang="en"><head>
2 |   <meta charset="utf-8">
3 |   <meta name="viewport" content="width=device-width, user-scalable=no">
4 |   <link rel="shortcut icon" href="./src/favicon.ico">
5 |   <title>Calculators</title>
6 |   <link href="build/static/css/main.2706593e.css" rel="stylesheet"></head>
7 |   <body>
8 |     <div id="root"></div>
9 |     <a class="github-fork-ribbon left-top" href="https://github.com/ahfarmer/calculator-d"
10 |       Fork me on GitHub
11 |     </a>
12 |     <!--
13 |       This HTML file is a template.
14 |       If you open it directly in the browser, you will see an empty page.
15 |
16 |       You can add webfonts, meta tags, or analytics to this file.
17 |       The build step will place the bundled scripts into the <body> tag.
18 |
19 |       To begin the development, run `npm start` in this folder.
20 |       To create a production bundle, use `npm run build`.
21 |     -->
22 |
23 |
24 | <script defer src="build/static/js/main.8cdfd76c.js"></script></body></html>

```

รูปที่ 33 ไฟล์เอชทีเอ็มแอลที่ถูกแก้ไขโดยเครื่องมือ

11) ตรวจสอบไฟล์แพ็คเกจที่ถูกแก้ไขโดยเครื่องมือดังรูปที่ 34

```

{} package.json X
1  {
2    "name": "calculator",
3    "version": "0.1.0",
4    "private": true,
5    "devDependencies": {
6      "gh-pages": "^0.11.0",
7      "react-scripts": "0.4.1"
8    },
9    "dependencies": {
10     "big.js": "^3.1.3",
11     "github-fork-ribbon-css": "^0.2.1",
12     "react": "^15.3.1",
13     "react-dom": "^15.3.1",
14     "electron": "~1.7.8",
15     "shelljs": "^0.8.2"
16   },
17   "scripts": {
18     "start": "react-scripts start",
19     "build": "react-scripts build",
20     "test": "react-scripts test --env=jsdom",
21     "eject": "react-scripts eject",
22     "deploy": "gh-pages -d build",
23     "js2xdesk-start": "electron .",
24     "js2xdesk-build": "react-scripts build"
25   },
26   "eslintConfig": {
27     "extends": "./node_modules/react-scripts/config/eslint.js"
28   },
29   "main": "js2xdesk-main.js"
30 }

```

รูปที่ 34 ไฟล์แพ็คเกจที่ถูกแก้ไขโดยเครื่องมือ

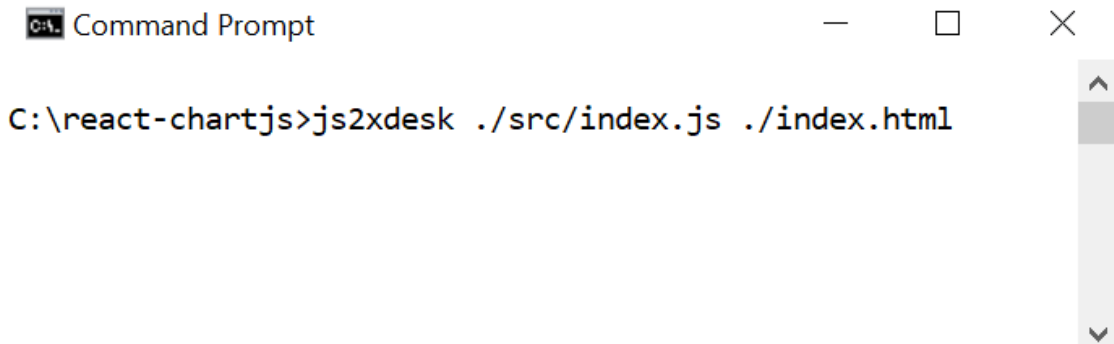
จากการทำการจับเวลานับตั้งแต่ป้อนคำสั่งในการใช้เครื่องมือดังรูปที่ 28 โดยใช้เครื่องมือเป็นผู้จับเวลา ในการทำงานทั้งหมดใช้เวลา 79 วินาทีซึ่งจากเวลาในการทำงานทั้งหมดใช้เวลาในการติดตั้งจากคำสั่ง `npm install` ดังรูปที่ 12 เป็นเวลา 65 วินาที เมื่อเปรียบเทียบกับการทำงานของผู้วิจัยซึ่งมีความรู้และประสบการณ์ในการทำการแปลงอยู่แล้วต้องใช้เวลาอย่างน้อย 30 นาที ผู้วิจัยคาดว่าหากนักพัฒนาที่มีประสบการณ์ทางด้านเว็บเทคโนโลยีได้ทำการแปลงเว็บแอปพลิเคชันให้เป็นเดสก์ท็อปโดยเพิ่งเริ่มศึกษาเป็นครั้งแรกอาจต้องศึกษาและทดลองแปลงโดยใช้เวลาทั้งหมดอย่างน้อย 16 ชั่วโมงซึ่งสามารถแสดงเปรียบเทียบได้ดังตารางที่ 1

ตารางที่ 1 เวลาที่ใช้ในการแปลงเว็บแอปพลิเคชันให้เป็นเดสก์ท็อป

ชนิดของงาน	เวลาที่ใช้		
	นักพัฒนาเว็บประสบการณ์ 2 ปี		เครื่องมืออัตโนมัติ
	มีความรู้และประสบการณ์ในการแปลง	ไม่เคยศึกษากรอบงานอิเล็กทรอนิกส์	
ศึกษารอบงานอิเล็กทรอนิกส์	-	8 ชั่วโมง	-
ทำการแปลงโครงการจากหัวข้อ 4.3	30 นาที	8 ชั่วโมง	79 วินาที
งานทั้งหมดในการแปลงโครงการจากหัวข้อ 4.3	30 นาที	16 ชั่วโมง	79 วินาที

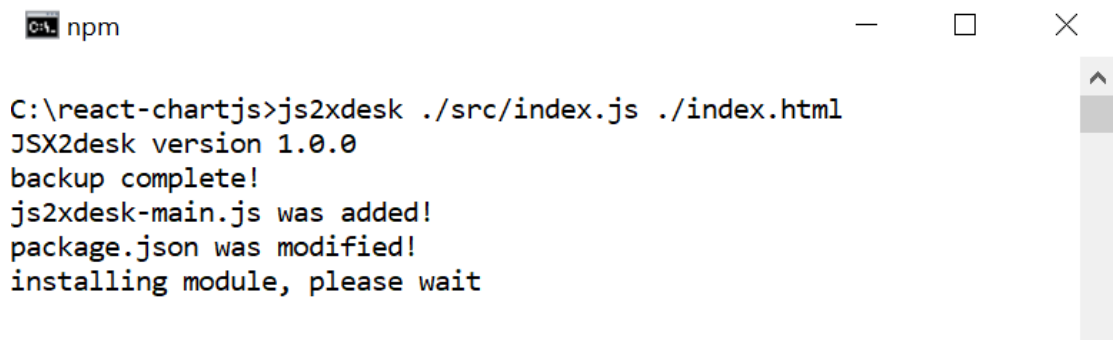
4.4 การสาธิตด้วยเครื่องมือที่พัฒนา

เพื่อเป็นการแสดงให้เห็นถึงผลการทำงานของเครื่องมืออัตโนมัติเพิ่มเติมจึงได้นำซอร์สโค้ดจากโครงการพัฒนาเว็บแอปพลิเคชันอื่น ๆ มาทำการสาธิต ซึ่งนำซอร์สโค้ดสำหรับแสดงผลแผนภูมิในรูปแบบต่าง ๆ มาจากโครงการพัฒนารีแอคต์ชาร์ตเจเอสทู(react-chartjs-2)[12] มาใช้เป็นข้อมูลนำเข้าให้กับเครื่องมือ โดยนักพัฒนาทำการใช้คำสั่งเรียกใช้งานเครื่องมือในที่อยู่ของโครงการพัฒนาเว็บแอปพลิเคชันดังรูปที่ 35 หลังจากนั้นเครื่องมือจะทำงานเป็นลำดับขั้นตามกระบวนการแปลงดังรูปที่ 36 และเครื่องมือทำการเปิดแอปพลิเคชันเป็นขั้นตอนสุดท้ายดังรูปที่ 37 จากนั้นทำการเปรียบเทียบการแสดงผลเดสก์ท็อปแอปพลิเคชันที่ได้กับเว็บแอปพลิเคชันบนเบราว์เซอร์ดังรูปที่ 38 ถึงรูปที่ 46 โดยหน้าต่างทางด้านซ้ายเป็นเว็บแอปพลิเคชันและหน้าต่างทางด้านขวาเป็นเดสก์ท็อปแอปพลิเคชันที่ได้จากการแปลงผ่านเครื่องมืออัตโนมัติ



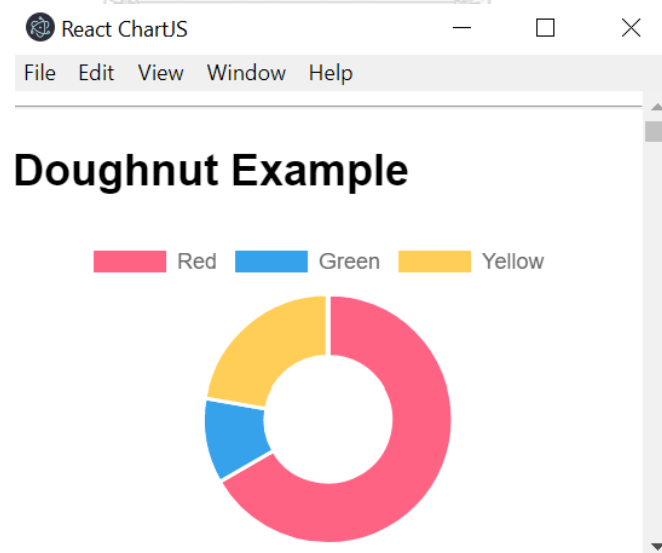
```
C:\react-chartjs>js2xdesk ./src/index.js ./index.html
```

รูปที่ 35 สาธิตการใช้เครื่องมืออัตโนมัติ

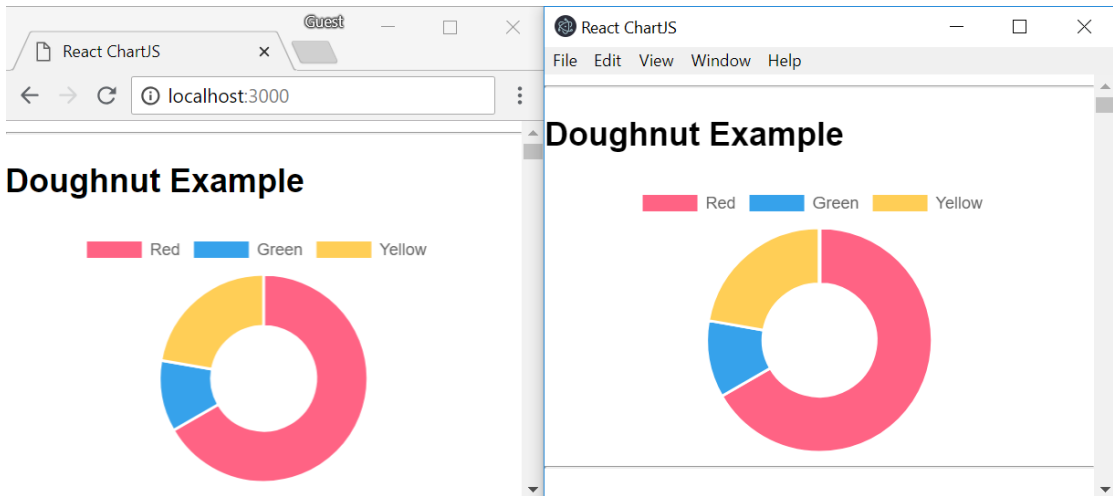


```
C:\react-chartjs>js2xdesk ./src/index.js ./index.html
JSX2desk version 1.0.0
backup complete!
js2xdesk-main.js was added!
package.json was modified!
installing module, please wait
```

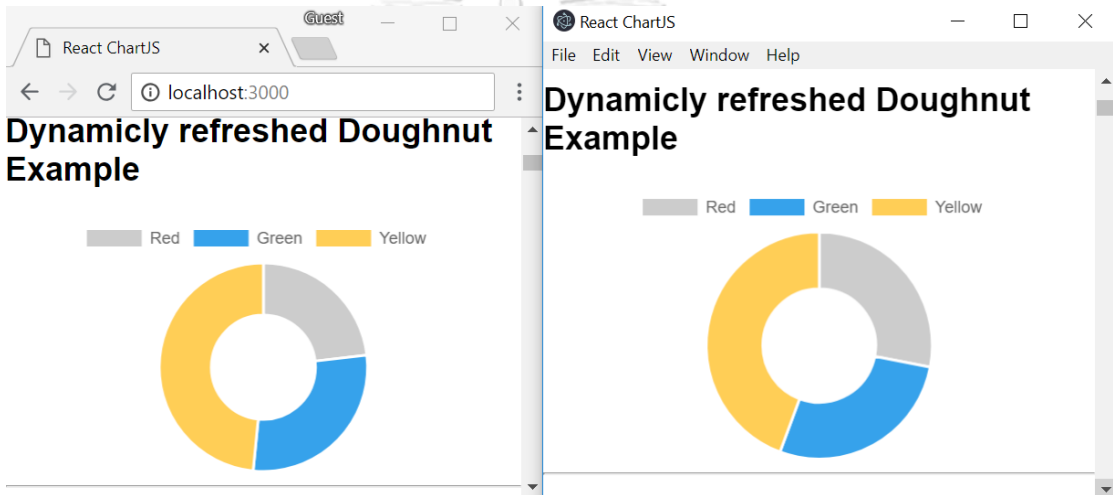
รูปที่ 36 เครื่องมือทำงานตามกระบวนการเปลี่ยนแปลง



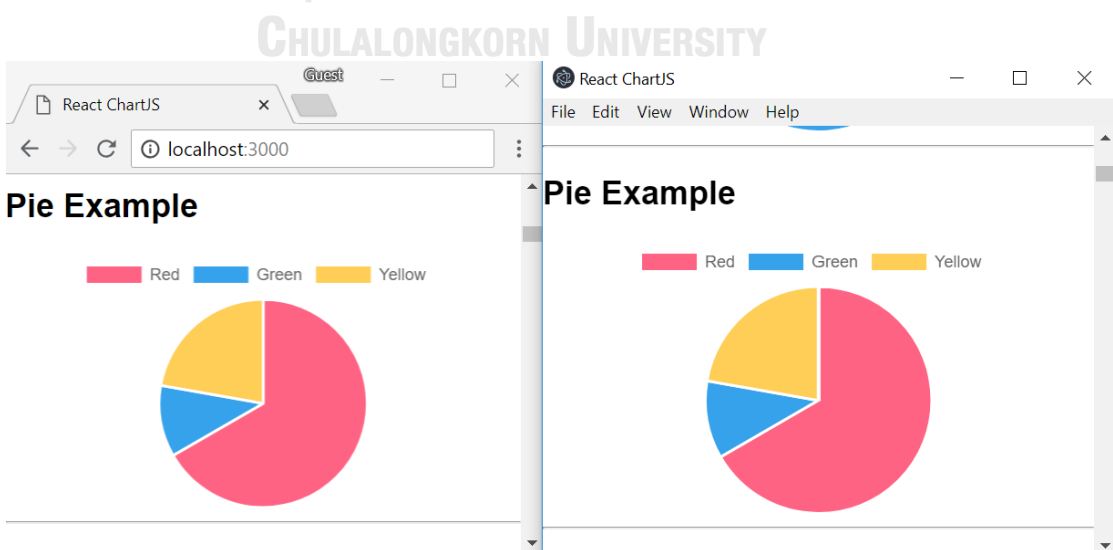
รูปที่ 37 เดสก์ท็อปแอปพลิเคชันที่ถูกเปิดใช้งานโดยเครื่องมืออัตโนมัติ



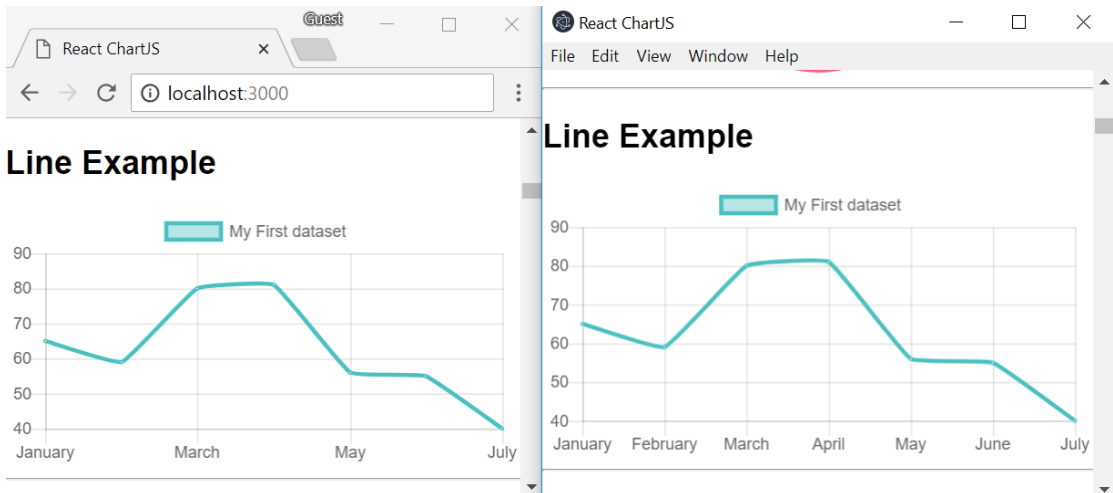
รูปที่ 38 เปรียบเทียบการแสดงผลแผนภูมิที่ 1



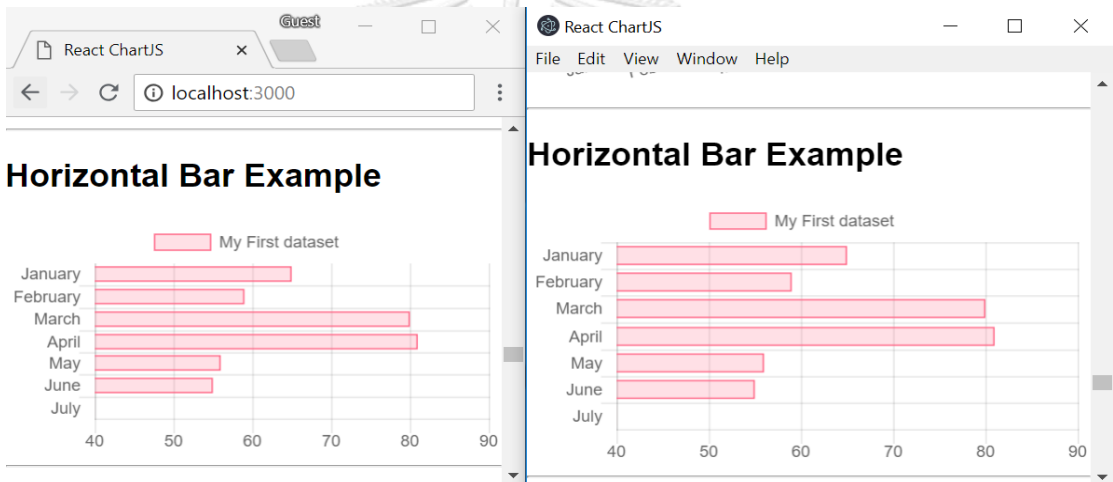
รูปที่ 39 เปรียบเทียบการแสดงผลแผนภูมิที่ 2



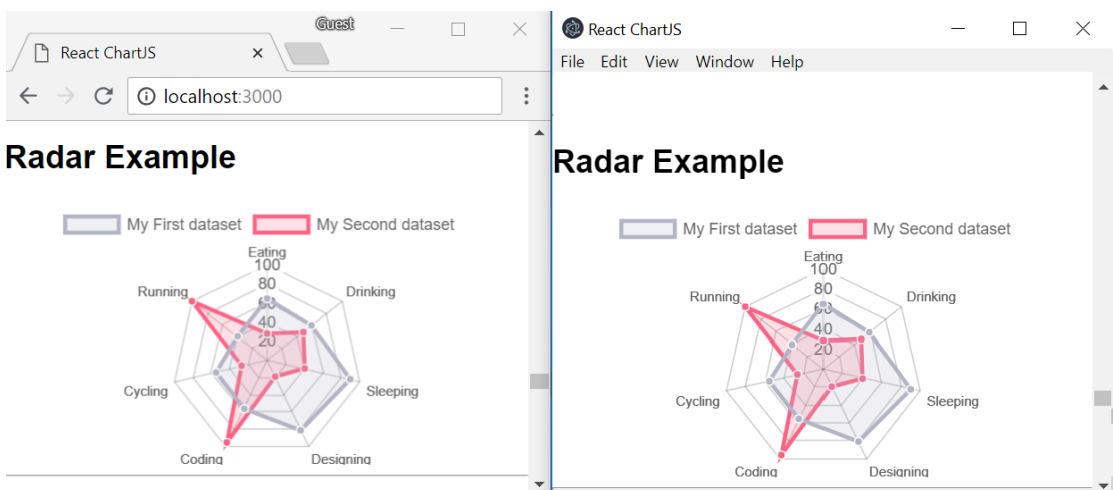
รูปที่ 40 เปรียบเทียบการแสดงผลแผนภูมิที่ 3



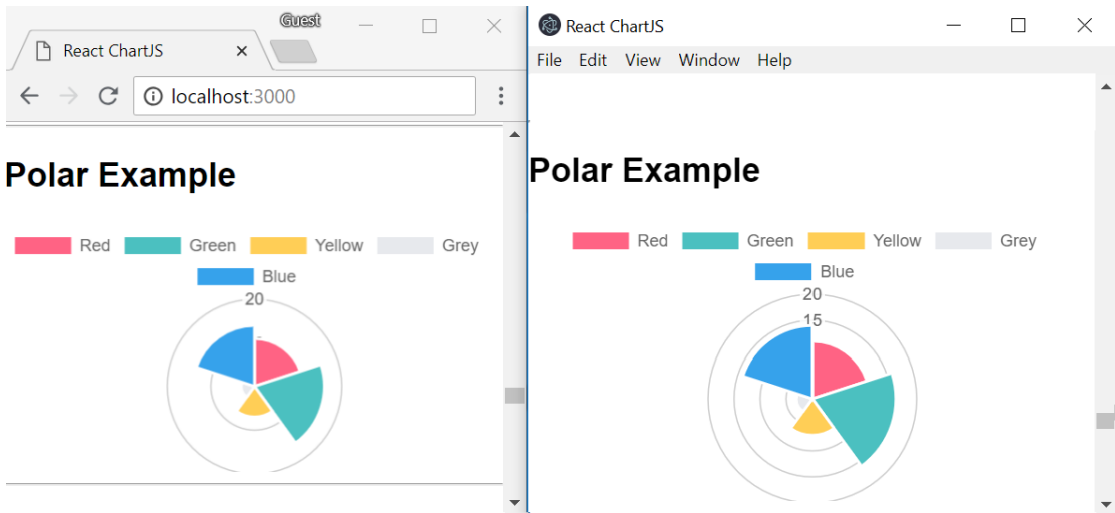
รูปที่ 41 เปรียบเทียบการแสดงผลแผนภูมิที่ 4



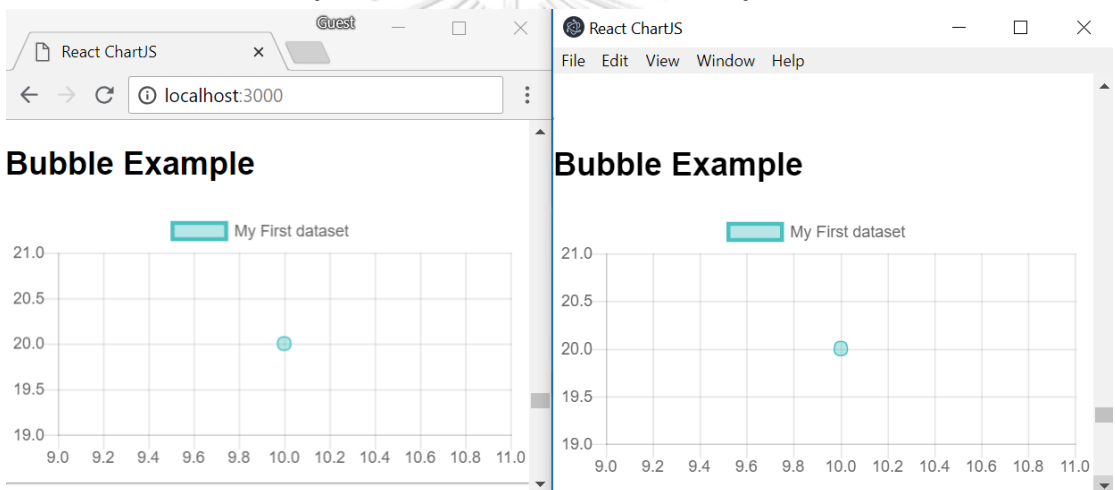
รูปที่ 42 เปรียบเทียบการแสดงผลแผนภูมิที่ 5



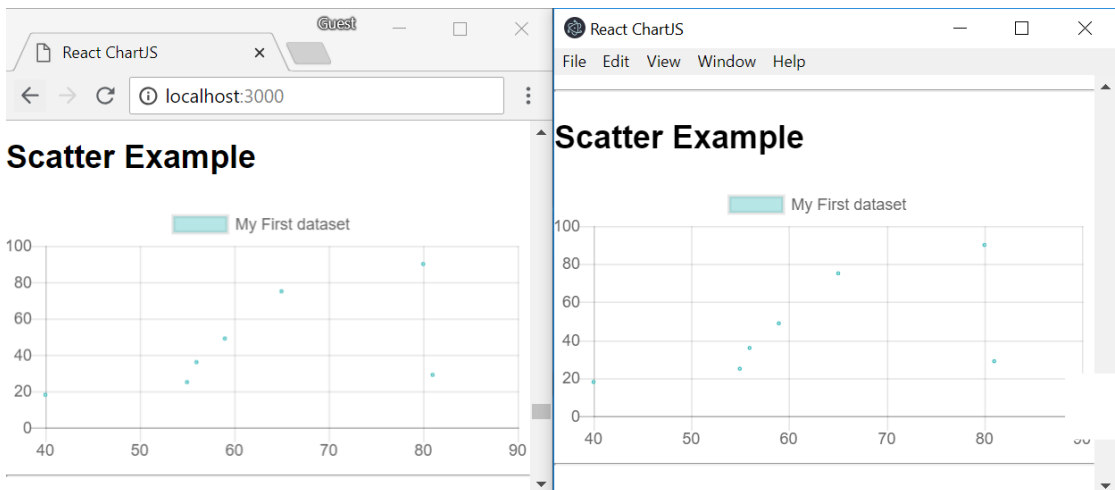
รูปที่ 43 เปรียบเทียบการแสดงผลแผนภูมิที่ 6



รูปที่ 44 เปรียบเทียบการแสดงผลแผนภูมิที่ 7



รูปที่ 45 เปรียบเทียบการแสดงผลแผนภูมิที่ 8



รูปที่ 46 เปรียบเทียบการแสดงผลแผนภูมิที่ 9

บทที่ 5

การทดสอบระบบ

5.1 จุดประสงค์ของการทดสอบ

จุดประสงค์ของการทดสอบระบบ เพื่อสนับสนุนแนวทางในการแปลงเว็บแอปพลิเคชันให้กลายเป็นเดสก์ท็อปแอปพลิเคชันแบบข้ามแพลตฟอร์มและการพัฒนาเครื่องมือที่สนับสนุนแนวทางในบทที่ 4 โดยเนื้อหาจะประกอบไปด้วยการทดสอบระบบ ตั้งแต่การติดตั้งระบบ การใช้งานคำสั่งของเครื่องมือ การแก้ไขไฟล์ซอร์สโค้ด ตลอดจนผลลัพธ์ของหน้าส่วนต่อประสานของเดสก์ท็อปแอปพลิเคชัน

5.2 การทดสอบระบบ

การทดสอบระบบที่ใช้เครื่องมือในการแปลงเว็บแอปพลิเคชันให้กลายเป็นเดสก์ท็อปแอปพลิเคชันได้ออกแบบวิธีทดสอบแบบกล่องดำ (Black Box Testing) ซึ่งจะทดสอบตามส่วนประกอบย่อยของความต้องการและมุ่งเน้นทดสอบฟังก์ชันการทำงาน รวมถึงผลลัพธ์ที่ได้จากการรวมทั้งหมดของส่วนประกอบย่อย ดังต่อไปนี้

- 1) การทดสอบการติดตั้งเครื่องมือ
 - 1.1 ทดสอบการติดตั้งเครื่องมือผ่านทางบรรทัดคำสั่ง
- 2) การทดสอบการรับคำสั่งและตัวแปร
 - 2.1 ทดสอบการรับคำสั่งและตัวแปรที่อยู่ไฟล์จาวาสคริปต์และไฟล์เอชทีเอ็มแอล
 - 2.2 ทดสอบการรับคำสั่งและตัวแปรที่อยู่ไฟล์จาวาสคริปต์
 - 2.3 ทดสอบการรับคำสั่งและตัวแปรที่อยู่ไฟล์เอชทีเอ็มแอล
 - 2.4 ทดสอบการป้อนคำสั่งและไม่ป้อนตัวแปร
- 3) การทดสอบการสำรองข้อมูล
 - 3.1 ทดสอบการสร้างไฟล์เดอร์แบคอัพและสำรองข้อมูลทั้งหมดในไฟล์เดอร์โครงการพัฒนาเว็บแอปพลิเคชัน

- 4) การทดสอบเครื่องมือในการสร้างไฟล์เมน
 - 4.1 ทดสอบการสร้างไฟล์เมนเพิ่มลงไปโนโพลเดอร์โครงการพัฒนาเว็บแอปพลิเคชัน
 - 4.2 ทดสอบการเพิ่มที่อยู่ไฟล์เอชทีเอ็มทีที่ได้จากการใส่ตัวแปรลงไปในบรรทัดคำสั่ง เพื่อเพิ่มที่ลงไปไฟล์เมน
- 5) การทดสอบเครื่องมือในการแก้ไขไฟล์แพ็กเกจ
 - 5.1 ทดสอบการแก้ไขไฟล์แพ็กเกจในกรณีที่ไฟล์แพ็กเกจต้องเพิ่มพรอปเพอร์ตี้ (property) เมน
 - 5.2 ทดสอบการแก้ไขไฟล์แพ็กเกจในกรณีที่ไฟล์แพ็กเกจต้องแก้ไขพรอปเพอร์ตี้สคริปต์
 - 5.3 ทดสอบการแก้ไขไฟล์แพ็กเกจในกรณีที่ไฟล์แพ็กเกจต้องแก้ไขพรอปเพอร์ตี้ดีเพนเดนซี
 - 5.4 ทดสอบการแก้ไขไฟล์แพ็กเกจในกรณีที่ไฟล์แพ็กเกจต้องแก้ไขพรอปเพอร์ตี้ดีเพนเดนซี (devDependencies)
- 6) การเครื่องมือในการเรียกใช้คำสั่งบิลด์
 - 6.1 ทดสอบการเรียกใช้คำสั่งบิลด์โดยเครื่องมือเป็นผู้เรียกใช้งาน
 - 6.2 ทดสอบการสร้างโพลเดอร์บิลด์โดยคำสั่งบิลด์ที่ถูกเรียกใช้โดยเครื่องมือ
 - 6.3 ทดสอบการสร้างไฟล์จากคำสั่งบิลด์โดยเครื่องมือจะสร้างไฟล์จาวาสคริปต์และไฟล์ซีเอสเอสแบบไฟล์เดียว
- 7) การทดสอบเครื่องมือในการแก้ไขไฟล์เอชทีเอ็มแอล
 - 7.1 ทดสอบการแก้ไขไฟล์เอชทีเอ็มแอลโดยเพิ่มแท็ก (Tag) สคริปต์ (<script>) และที่อยู่ของไฟล์จาวาสคริปต์แบบไฟล์เดียว
 - 7.2 ทดสอบการแก้ไขไฟล์เอชทีเอ็มแอลโดยเพิ่มแท็กสไตล์ (<style>)และที่อยู่ของไฟล์ซีเอสเอสแบบไฟล์เดียว
- 8) การทดสอบเครื่องมือในการเปิดแอปพลิเคชัน
 - 8.1 ทดสอบการเปิดการทำงานเดสก์ท็อปแอปพลิเคชันโดยเครื่องมือเรียกใช้คำสั่ง `npm run js2xdesk-start`

9) การทดสอบเดสก์ท็อปแอปพลิเคชันที่ได้จากการแปลงในส่วนของการแสดงผลส่วนต่อประสานผู้ใช้งาน

9.1 ทดสอบการแสดงผลส่วนต่อประสานผู้ใช้งานบนเดสก์ท็อปแอปพลิเคชันเปรียบเทียบกับ
กับการแสดงผลส่วนต่อประสานผู้ใช้งานบนเว็บแอปพลิเคชัน

10) การทดสอบเดสก์ท็อปแอปพลิเคชันที่ได้จากการแปลงในส่วนของการทำงาน

10.1 ทดสอบการทำงานเดสก์ท็อปแอปพลิเคชันที่ได้จากการแปลงในทุกหน้าที่
การทำงานหลักของแอปพลิเคชัน

ตารางที่ 2 ผลการทดสอบการติดตั้งเครื่องมือ

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC01	การทดสอบการติดตั้งเครื่องมือ	ทดสอบการติดตั้งเครื่องมือผ่านทางบรรทัดคำสั่ง	บรรทัดคำสั่งแสดงข้อความ "+js2xdesk@1.0.0"	ถูกต้อง

ตารางที่ 3 ผลการทดสอบการรับคำสั่งและตัวแปร

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC02	ทดสอบการป้อนคำสั่งและตัวแปรสองค่า	ทดสอบการรับคำสั่งและตัวแปรที่อยู่ไฟล์จาวาสคริปต์และไฟล์เอชทีเอ็มแอล	เครื่องมือรับคำสั่งและตัวแปรทั้งสองตัวได้	ถูกต้อง
TC03	ทดสอบการป้อนคำสั่งและตัวแปรหนึ่งค่า	ทดสอบการรับคำสั่งและตัวแปรที่อยู่ไฟล์จาวาสคริปต์	เครื่องมือรับคำสั่งและแสดงผลให้ป้อนตัวแปรให้ครบและหยุดการทำงาน	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC04	ทดสอบการรับคำสั่ง และตัวแปรที่อยู่ไฟล์ เอชทีเอ็มแอล	ทดสอบการป้อน คำสั่งและตัวแปร หนึ่งค่า	เครื่องมือรับคำสั่ง และแสดงผลให้ป้อน ตัวแปรให้ครบและ หยุดการทำงาน	ถูกต้อง
TC05	ทดสอบการรับคำสั่ง และไม่ป้อนตัวแปร	ทดสอบการป้อน คำสั่งและไม่ป้อนตัว แปร	เครื่องมือรับคำสั่ง และแสดงผลให้ป้อน ตัวแปรให้ครบและ หยุดการทำงาน	ถูกต้อง

ตารางที่ 4 ผลการทดสอบการสำรองข้อมูล

หมายเลข กรณีทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC06	ทดสอบการสำรอง ข้อมูล	ทดสอบการสร้าง ไฟล์เดอร์แบคอัพ และสำรองข้อมูล ทั้งหมดในไฟล์เดอร์ โครงการพัฒนาเว็บ แอปพลิเคชัน	เครื่องมือสำรอง ไฟล์และไฟล์เดอร์ ทั้งหมดไว้ใน ไฟล์เดอร์แบคอัพที่ ถูกสร้างโดย เครื่องมือ	ถูกต้อง

ตารางที่ 5 ผลการทดสอบการสร้างไฟล์เมน

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC07	ทดสอบการสร้าง ไฟล์เมน	ทดสอบการสร้างไฟล์ เมนเพิ่มลงไป ในโพลเดอร์โครงการ พัฒนาเว็บแอปพลิเคชัน	เครื่องมือเพิ่มไฟล์ เมนที่ได้จาก เครื่องมือลงไป ในโพลเดอร์โครงการ พัฒนาเว็บแอปพลิเคชัน	ถูกต้อง
TC08	ทดสอบการเพิ่มที่อยู่ ไฟล์เอชทีเอ็มแอลลง ไปในไฟล์เมน	ทดสอบการเพิ่มที่อยู่ ไฟล์เอชทีเอ็ม ซึ่งที่ ไฟล์ดังกล่าวได้จาก การรับตัวแปรมาจาก บรรทัดคำสั่ง เพื่อ เพิ่มที่ลงไปไฟล์ เมน	ที่อยู่ของไฟล์ เอชทีเอ็มแอลที่ รับมาจากบรรทัด คำสั่งถูกระบุลงไป ในไฟล์เมนที่โพลเดอร์ โครงการพัฒนาเว็บ แอปพลิเคชัน	ถูกต้อง

ตารางที่ 6 ผลการทดสอบการแก้ไขไฟล์แพ็กเกจ

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC09	ทดสอบการแก้ไข ไฟล์แพ็กเกจ	ทดสอบการแก้ไข ไฟล์แพ็กเกจในกรณี ที่ไฟล์แพ็กเกจต้อง เพิ่มพรอปเพอร์ตี้ (property) เมน	เครื่องมือแก้ไขไฟล์ แพ็กเกจโดยมี บรรทัดที่มีพรอป เพอร์ตี้เมนเพิ่มขึ้นมา คือ "main": "js2xdesk-main.js",	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC10	ทดสอบการแก้ไข ไฟล์แพ็กเกจ	ทดสอบการแก้ไข ไฟล์แพ็กเกจในกรณี ที่ไฟล์แพ็กเกจต้อง แก้ไขพรอปเพอร์ตี้ สคริปต์	เครื่องมือแก้ไขไฟล์ แพ็กเกจโดยรวม สคริปต์เดิมของ โครงการพัฒนาเว็บ แอปพลิเคชันเข้ากับ สคริปต์ของโครงการ พัฒนาเดสก์ท็อปที่ได้ จากเครื่องมือโดยมี สคริปต์ที่เพิ่มมาจาก เครื่องมือคือ "js2xdesk-start": "electron .", "js2xdesk- build": "react- scripts build"	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC11	ทดสอบการแก้ไข ไฟล์แพ็คเกจ	ทดสอบการแก้ไข ไฟล์แพ็คเกจในกรณี ที่ไฟล์แพ็คเกจต้อง แก้ไขพรอปเพอร์ตี้ เพนเดนซี	เครื่องมือแก้ไขไฟล์ แพ็คเกจโดยรวมดี เพนเดนซีเดิมของ โครงการพัฒนาเว็บ แอปพลิเคชันเข้ากับ ดีเพนเดนซีของ โครงการพัฒนา เดสก์ท็อปที่ได้จาก เครื่องมือโดยมีดีเพน เดนซีที่เพิ่มมาจาก เครื่องมือคือ "react": "^15.3.1", "electron": "~1.7.8", "shelljs": "^0.8.2"	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC12	ทดสอบการ แก้ไขไฟล์ แพ็คเกจ	ทดสอบการ แก้ไขไฟล์ แพ็คเกจในกรณี ที่ไฟล์แพ็คเกจ ต้องแก้ไขพรอป เพอร์ตี้เดฟตี เพนเดนซี (devDependencies)	เครื่องมือแก้ไขไฟล์แพ็คเกจโดยรวมเดฟตีเพน เดนซีเดิมของโครงการพัฒนาเว็บแอปพลิเคชัน ชั้นเข้ากับเดฟตีเพนเดนซีของโครงการพัฒนา เดสก์ท็อปที่ได้จากเครื่องมือโดยมีเดฟตีเพน เดนซีที่เพิ่มมาจากเครื่องมือคือ "react-scripts": "0.4.1"	ถูกต้อง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ตารางที่ 7 ผลการทดสอบการเรียกใช้คำสั่งบิลด์

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC13	ทดสอบการเรียกใช้ คำสั่งบิลด์	ทดสอบการเรียกใช้ คำสั่งบิลด์โดย เครื่องมือเป็นผู้ เรียกใช้งาน	เครื่องมือเรียกใช้ คำสั่งบิลด์	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC14	ทดสอบการสร้าง โพลเดอร์บิลด์	ทดสอบการสร้าง โพลเดอร์บิลด์โดย คำสั่งบิลด์ที่ถูก เรียกใช้โดยเครื่องมือ	เครื่องมือสร้าง โพลเดอร์บิลด์เพิ่มลง ไปในโพลเดอร์โครง กาพัฒนาเว็บแอป พลิเคชัน	ถูกต้อง
TC15	ทดสอบการสร้าง ไฟล์จากคำสั่งบิลด์	ทดสอบการสร้างไฟล์ จากคำสั่งบิลด์โดย เครื่องมือจะสร้าง ไฟล์จาวาสคริปต์และ ไฟล์ซีเอสเอสแบบ ไฟล์เดียว	เครื่องมือสร้างไฟล์ จาวาสคริปต์และซีเอส เอสแบบไฟล์เดียว	ถูกต้อง

ตารางที่ 8 ผลการทดสอบการแก้ไขไฟล์เอชทีเอ็มแอล

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC16	ทดสอบการแก้ไข ไฟล์เอชทีเอ็มแอล	ทดสอบการแก้ไข ไฟล์เอชทีเอ็มแอล โดยเพิ่มแท็ก (Tag) สคริปต์ (<script> และที่อยู่ของไฟล์จา วาสคริปต์แบบไฟล์ เดียว	เครื่องมือแก้ไขไฟล์ เอชทีเอ็มแอลโดย เพิ่มแท็กสคริปต์และ ระบุที่อยู่ของไฟล์จา วาสคริปต์แบบไฟล์ เดียวซึ่งถูกสร้างโดย คำสั่งบิลด์	ถูกต้อง

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC17	ทดสอบการแก้ไข ไฟล์เอชทีเอ็มแอล	ทดสอบการแก้ไข ไฟล์เอชทีเอ็มแอล โดยเพิ่มแท็กสไตร์ (<code><style></code>)และที่อยู่ ของไฟล์ซีเอสเอส แบบไฟล์เดียว	เครื่องมือแก้ไขไฟล์ เอชทีเอ็มแอลโดย เพิ่มแท็กสไตร์และ ระบุที่อยู่ของไฟล์ซี เอสเอสแบบไฟล์ เดียวซึ่งถูกสร้างโดย คำสั่งบิลด์	ถูกต้อง

ตารางที่ 9 ผลการทดสอบการเรียกใช้คำสั่งเพื่อเปิดเดสก์ท็อปแอปพลิเคชัน

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC18	ทดสอบการเปิดการ ทำงานเดสก์ท็อป แอปพลิเคชัน	ทดสอบการเปิดการ ทำงานเดสก์ท็อป แอปพลิเคชันโดย เครื่องมือเรียกใช้ คำสั่ง <code>npm run js2xdesk-start</code>	เครื่องมือเรียกใช้ คำสั่งเพื่อเปิดการ ทำงานของ เดสก์ท็อปแอปพลิเคชัน และมีหน้าต่าง แอปพลิเคชันเกิดขึ้น มาบนหน้าจอผู้ใช้	ถูกต้อง

ตารางที่ 10 ผลการทดสอบการแสดงผลส่วนต่อประสานผู้ใช้งานของเดสก์ท็อปแอปพลิเคชัน

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC19	ทดสอบการแสดงผล ส่วนต่อประสาน ผู้ใช้งาน	ทดสอบการแสดงผล ส่วนต่อประสาน ผู้ใช้งานบนเดสก์ท็อป แอปพลิเคชัน เปรียบเทียบกับ การแสดงผลส่วนต่อ ประสานผู้ใช้งานบน เว็บแอปพลิเคชัน	เดสก์ท็อปแอปพลิเคชัน ที่ได้จากการ แปลงแสดงผลส่วน ต่อประสานผู้ใช้งาน เหมือนกันกับส่วนต่อ ประสานผู้ใช้งานบน เว็บแอปพลิเคชัน	ถูกต้อง

ตารางที่ 11 ผลการทดสอบการทำงานของเดสก์ท็อปแอปพลิเคชัน

หมายเลข กรณี ทดสอบ	การทดสอบ	คำอธิบาย	ผลที่คาดหวัง	ผลการ ทดสอบ
TC20	ทดสอบการทำงานของ เดสก์ท็อปแอปพลิเคชัน ที่ได้จากการ แปลง	ทดสอบการทำงานของ เดสก์ท็อปแอปพลิเคชัน ที่ได้จากการแปลง ในทุกหน้าที่การ ทำงานหลักของแอป พลิเคชัน	เดสก์ท็อปแอปพลิเคชัน ที่ได้จากการ แปลงทำงานถูกต้อง ในทุกหน้าที่การ ทำงานหลักเหมือน กันกับเว็บแอปพลิเคชัน	ถูกต้อง

5.3 สรุปผลการทดสอบ

จากการทดสอบที่กล่าวมาสรุปได้ว่า แนวทางในการแปลงเว็บแอปพลิเคชันบนพื้นฐานของจาวาสคริปต์ให้เป็นเดสก์ท็อปแอปพลิเคชันสามารถทำงานได้ถูกต้อง โดยเดสก์ท็อปแอปพลิเคชันที่ได้จากการแปลงทำงานได้เหมือนกันกับการทำงานแบบเว็บแอปพลิเคชันบนเว็บเบราว์เซอร์ รวมไปถึงมีส่วนต่อประสานผู้ใช้งานของเดสก์ท็อปแอปพลิเคชันที่เหมือนกันกับของเว็บแอปพลิเคชัน อีกทั้งเครื่องมือยังทำงานตามแนวทางได้ถูกต้อง สามารถช่วยนักพัฒนาแปลงโครงการพัฒนาเว็บแอปพลิเคชันให้เป็นเดสก์ท็อปแอปพลิเคชันเพื่อเป็นแนวทางให้กับนักพัฒนาได้



บทที่ 6

สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้จะกล่าวถึงสรุปผลการวิจัย ข้อจำกัดของงานวิจัย และงานวิจัยในอนาคต โดยแต่ละส่วนที่กล่าวมานั้นมีรายละเอียดดังต่อไปนี้

6.1 สรุปผลการวิจัย

งานวิจัยนี้นำเสนอแนวทางในการแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้เป็นเดสก์ท็อปข้ามแพลตฟอร์ม โดยจะใช้โครงการพัฒนาเว็บแอปพลิเคชันที่ใช้รีแอคต์เจเอสในการช่วยพัฒนาส่วนต่อประสาน จากการช่วยเหลือของกรอบงานอิเล็กทรอนิกส์ที่มีความสามารถในการพัฒนาเดสก์ท็อปแอปพลิเคชันแบบข้ามแพลตฟอร์มและใช้เทคโนโลยีของการพัฒนาเว็บแอปพลิเคชันบนพื้นฐานของจาวาสคริปต์ ซึ่งเป็นที่รู้จักสำหรับการพัฒนาเดสก์ท็อปแอปพลิเคชันสมัยใหม่ซึ่งใช้ซอร์สโค้ดชุดเดียวสำหรับระบบปฏิบัติการทั้งสามแบบ คือ ระบบปฏิบัติการวินโดวส์ แมคโอเอส และลินุกซ์ โดยมีแอปพลิเคชันที่ถูกพัฒนาและแจกจ่ายออกมาแล้วจนมีผู้ใช้งานเป็นวงกว้าง ทำให้งานวิจัยนี้ได้ผลลัพธ์เป็นเดสก์ท็อปแอปพลิเคชันแบบข้ามแพลตฟอร์ม นอกจากนี้งานวิจัยนี้ยังได้พัฒนาเครื่องมือสนับสนุนในการแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้เป็นเดสก์ท็อปอย่างอัตโนมัติ โดยเครื่องมือดังกล่าวได้นำโทนดเจเอสและเอ็นพีเอ็มโมดูลมาช่วยในการสร้างเครื่องมือแบบคอมมานด์ไลน์ ซึ่งเครื่องมือที่ใช้แนวคิดในงานวิจัยสามารถช่วยนักพัฒนาในการแปลงเว็บแอปพลิเคชันบนพื้นฐานจาวาสคริปต์ให้เป็นเดสก์ท็อปแอปพลิเคชันแบบข้ามแพลตฟอร์มได้แบบอัตโนมัติเพื่อลดขั้นตอนการของการแปลงเว็บแอปพลิเคชัน

6.2 ข้อจำกัดงานวิจัย

- 1) ใช้เว็บแอปพลิเคชันที่มีรีแอคต์เจเอสเป็นเครื่องมือช่วยพัฒนาส่วนต่อประสานในงานวิจัยนี้เท่านั้น
- 2) ผู้ใช้จะต้องจัดเตรียมไฟล์แพ็คเกจไว้ที่โฟลเดอร์แรก (root directory) ของโครงการพัฒนาเว็บแอปพลิเคชัน
- 3) แนวคิดและเครื่องมือนี้ไม่รองรับโครงการพัฒนาที่ต้องมีขั้นตอนการคอมไพล์ไฟล์เข้าซ้อนกับของระบบ เช่น การคอมไพล์ไฟล์เอสซีเอสเอส (.scss) ให้เป็นไฟล์ซีเอสเอสในขั้นตอนก่อนการทำงานเดิมของเว็บแอปพลิเคชัน

6.3 งานวิจัยในอนาคต

- 1) นำแนวทางในการแปลงเว็บแอปพลิเคชันในงานวิจัยนี้ที่ใช้โครงการพัฒนาที่ใช้รีแอกต์เจเอสเป็นส่วนช่วยในการพัฒนาไปใช้พัฒนาต่อยอดกับการแปลงเว็บแอปพลิเคชันที่ใช้กรอบงานจาวาสคริปต์อื่น ๆ เป็นส่วนช่วยในการพัฒนา เช่น แองกูลาร์เจเอส หรืออูเจเอส
- 2) นำแนวทางในการแปลงเว็บแอปพลิเคชันไปประยุกต์ใช้กับอุตสาหกรรมซอฟต์แวร์เพื่อเป็นส่วนช่วยของการพัฒนาเดสก์ท็อปแอปพลิเคชันและส่งมอบผลงานให้กับลูกค้า



รายการอ้างอิง

- [1] Electron | Build cross platform desktop apps with JavaScript, HTML, and CSS. Available: <https://electronjs.org>
- [2] Node.js. Available: <https://nodejs.org/en>
- [3] Wesley Hales, *HTML5 and JavaScript Web Apps*: O'Reilly Media, 2012.
- [4] Muhammed Jasim, *Building Cross-Platform Desktop Applications with Electron*: Packt Publishing, 2017.
- [5] Felix Rieseberg, *Introducing Electron*: O'Reilly Media, 2017.
- [6] John A. Berlin, Mat Kelly, Michael L. Nelson, and Michele C. Weigle, "WAIL: Collection-Based Personal Web Archiving", ACM/IEEE Joint Conference on Digital Libraries, 2017.
- [7] Example Projects – React. Available: <https://reactjs.org/community/examples.html>
- [8] Electron Example Project. Available: <https://github.com/electron/electron-quick-start>
- [9] API | Electron. Available: <https://electronjs.org/docs/api>
- [10] npm. Available: <https://www.npmjs.com/>
- [11] GitHub - Rhymond/product-compare-react: React Example - Product Compare Page. Available: <https://github.com/Rhymond/product-compare-react>
- [12] react-chartjs-2 – npm. Available: <https://www.npmjs.com/package/react-chartjs-2>



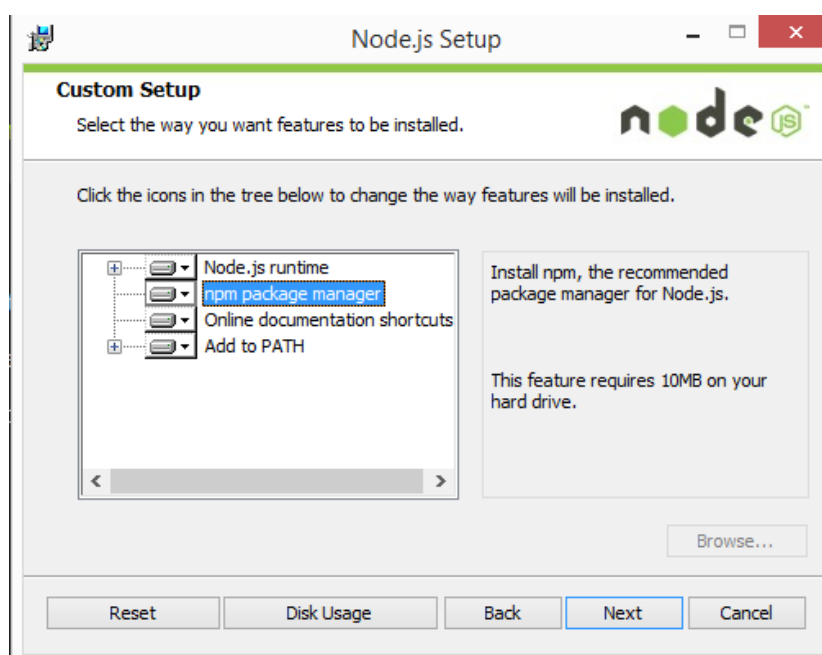
ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

การติดตั้งซอฟต์แวร์โนดเจส (NodeJs)


การทำงานของเครื่องมือในงานวิจัยนี้ มีการทำงานผ่านบรรทัดคำสั่งซึ่งเป็นเครื่องมือในการสนับสนุนการแปลงโครงการพัฒนารีแอกต์เจส โดยการทำงานของเครื่องมือต้องการซอฟต์แวร์โนดเจสเพื่อดำเนินการทำงาน โดยในการติดตั้งซอฟต์แวร์โนดเจสมิฉะนั้นขั้นตอนดังต่อไปนี้

- 1) ดาวน์โหลดตัวติดตั้งซอฟต์แวร์โนดเจสจากลิงค์ <https://nodejs.org/en/download>
- 2) เมื่อเปิดตัวติดตั้งขึ้นมาแล้วให้กำหนดค่าต่าง ๆ ตามที่ซอฟต์แวร์นั้นได้กำหนดค่าเริ่มต้น จากนั้นให้กดติดตั้งดังรูปที่ 47



รูปที่ 47 หน้าส่วนต่อประสานการกำหนดค่าเริ่มต้นและการติดตั้งซอฟต์แวร์โนดเจส

- 3) เมื่อติดตั้งซอฟต์แวร์สมบูรณ์ จำเป็นต้องเริ่มระบบปฏิบัติการของเครื่องคอมพิวเตอร์ใหม่
- 4) ทดสอบการติดตั้งสมบูรณ์โดยการเปิดบรรทัดคำสั่งและพิมพ์คำสั่งดังรูปที่ 48



```
Command Prompt
c:\>node -v
v9.11.1
c:\>npm -v
6.1.0
c:\>
```

รูปที่ 48 ตรวจสอบการติดตั้งซอฟต์แวร์ผ่านบรรทัดคำสั่ง

เมื่อป้อนคำสั่งดังรูปที่ 48 แล้วจะแสดงเวอร์ชันของซอฟต์แวร์ที่ติดตั้ง ถือว่าการติดตั้งครั้งนี้
สมบูรณ์



ประวัติผู้เขียนวิทยานิพนธ์

นายกิตติ เกร็ดพัฒนกุล เกิดเมื่อวันที่ 8 กุมภาพันธ์ พ.ศ. 2536 ที่จังหวัดกาญจนบุรี สำเร็จการศึกษาปริญญาตรีหลักสูตรวิศวกรรมศาสตรบัณฑิต (วศ.บ.) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ กำแพงแสน มหาวิทยาลัยเกษตรศาสตร์ ในปีการศึกษา 2557 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมซอฟต์แวร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2559

