บทความวิจัย (Research Article)

# ระบบเลือกตั้งสำเร็จรูปบนบล็อกเชน สำหรับกระทรวงการคลัง

## *Blockchain-based Election System for Ministry of Finance*

ปิยาณี สุขจำเริญ, สุขุมาล กิติสิน* และ ชวลิต ศรีสถาพรพัฒน์

ภาควิชาวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์
มหาวิทยาลัยเกษตรศาสตร์
* ผู้รับผิดชอบบทความ
sukumal.i@ku.th

Received: 8 Mar 2024
Revised: 26 Apr 2024
Accepted: 29 Apr 2024

**บทคัดย่อ**

งานวิจัยนี้นำเสนอการพัฒนาระบบเพื่อแก้ไขปัญหาการเลือกตั้งภายในของกระทรวงการคลังที่ปัจจุบันมีการเลือกตั้ง 2 วิธี คือ การเลือกตั้ง ณ สถานที่ที่ได้จัดเตรียมไว้ และการเลือกตั้งผ่านระบบที่เก็บข้อมูลแบบรวมศูนย์ ซึ่งอาจเกิดการทุจริตด้านความไม่โปร่งใสและความไม่เป็นธรรมได้ อีกทั้งการเลือกตั้งแต่ละครั้งมีเงื่อนไขที่แตกต่างกัน ทำให้ผู้พัฒนาระบบเลือกตั้งต้องพัฒนาระบบใหม่ทุกครั้งที่มีการเลือกตั้ง งานวิจัยนี้จึงนำเสนอระบบเลือกตั้งสำเร็จรูปบนบล็อกเชน สำหรับกระทรวงการคลัง โดยมีวัตถุประสงค์ 3 ข้อดังนี้ (1) พัฒนาระบบเลือกตั้งสำหรับใช้ในหน่วยงานกระทรวงการคลังที่ครอบคลุมความต้องการของผู้ใช้งาน (2) พัฒนาระบบเลือกตั้งที่สามารถปรับใช้งานได้กับทุกการเลือกตั้งตามเงื่อนไขที่ต้องการ (3) พัฒนาระบบเลือกตั้งที่โปร่งใส เป็นธรรมและตรวจสอบได้ โดยการวิจัยเริ่มต้นด้วยการศึกษาระบบเดิมและรวบรวมความต้องการของผู้ใช้งาน รวมถึงเอกสารและงานวิจัยที่เกี่ยวข้อง และวิเคราะห์ข้อมูลที่ได้จากการศึกษา จึงได้นำเทคโนโลยีบล็อกเชนเข้ามาช่วยให้ระบบมีความโปร่งใส และตรวจสอบได้ ซึ่งเป็นคุณสมบัติของบล็อกเชน และมีการออกแบบและพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ (GUI) และส่วนต่อประสานโปรแกรมประยุกต์ (API) ด้วย Next.js และพัฒนา smart contract โดยใช้ภาษา Solidity และ Hardhat สำหรับการพัฒนาบนแพลตฟอร์ม Ethereum และทำการทดสอบระบบ ดังนี้ (1) การทดสอบหน่วยย่อย (unit test) จำนวน 15 รายการ ซึ่งผลการทดสอบผ่านทั้ง 15 รายการ (2) การทดสอบประสิทธิภาพของระบบ แบ่งออกเป็น การทดสอบเวลาที่ใช้สำหรับการบันทึกและดึงข้อมูลจากบล็อกเชนและจากฐานข้อมูลกลางแบบรวมศูนย์ของระบบดั้งเดิม พบว่า การบันทึกข้อมูลลงบล็อกเชนใช้เวลาในการทำธุรกรรมมากกว่าการอ่านข้อมูลจากบล็อกเชน แต่การอ่านข้อมูลจากบล็อกเชนใช้เวลาน้อยกว่าเวลาที่ใช้ในการอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลาง และการทดสอบจำนวนธุรกรรมที่บล็อกเชนสามารถรองรับได้มากที่สุดใน 1 วินาที พบว่า ระบบรองรับการทำรายการเฉลี่ย 116.443 ธุรกรรมต่อวินาที และ (3) การประเมินคุณสมบัติของระบบ พบว่า ระบบมีความสอดคล้องของข้อมูล, ความสมบูรณ์ของข้อมูล และความโปร่งใสในการเข้าถึงข้อมูล ฯลฯ ที่สูง

**คำสำคัญ:** บล็อกเชน สมาร์ทคอนแทรค ส่วนต่อประสานกับผู้ใช้แบบกราฟิก ส่วนต่อประสานโปรแกรมประยุกต์

**Abstract**

This research presents the development of an Election System Packge Based on blockchain for Ministry of Finance. Currently, there are two methods: on-site elections and elections through a centralized data collection system. However, transparency and fairness issues, and each election has different conditions, requiring system developers to create a new system each time.

This study proposes a Election System Package Based on blockchain for Ministry of Finance with three objectives: (1) Develop an election system for use in the Ministry of Finance that meets the needs of users. (2) Develop an election system just once and can adapt it to every election. (3) Develop a transparent, fair, and auditable election system. The research begins by studying the existing system, gathering user requirements, reviewing relevant documents and research, and analyzing collected data. The proposed system introduces blockchain technology to enhance transparency and verification. The system includes a graphical user interface (GUI) and an application programming interface (API) developed using Next.js The smart contract is deployed using Solidity and Hardhat as the development environment for Ethereum. The system was tested as follows: (1) Unit test, totaling 15 items, with the test results passing all 15 items. (2) Performance testing, consisting of testing the time used for recording and reading data from blockchain and from the centralized database of traditional systems reveals that saving data to the blockchain takes more time to complete transactions than reading data from the blockchain. But reading data from a blockchain takes less time than it takes to read data from a centralized database, and testing the maximum number of transactions per second. The system supports an average of 116.443 transactions per second and (3)Evaluation of the system features. It was found that the system has high data consistency, high data completeness and high transparency in accessing information, etc.

**Keywords:** Blockchain, Smart contract, Graphical User Interface (GUI), Application Programming Interface (API)

## 1. บทนำ

สำนักงานปลัดกระทรวงการคลังเป็นหน่วยงานราชการหน่วยงานหนึ่งที่ต้องมีการเลือกตั้งภายในเป็นประจำทุกปี และในแต่ละปีมีการเลือกตั้งที่มีวัตถุประสงค์แตกต่างกัน โดยในปัจจุบันการเลือกตั้งของกระทรวงการคลังมี 2 วิธี คือ (1) การลงคะแนน ณ สถานที่ที่จัดเตรียมไว้ด้วยกระดาษ (2) การลงคะแนนผ่านระบบเลือกตั้งเดิมที่จัดเก็บข้อมูลในฐานข้อมูลกลาง ซึ่งต้องมีการพัฒนาระบบเลือกตั้งใหม่ทุกครั้งที่มีการเลือกตั้ง เนื่องจากการเลือกตั้งของกระทรวงการคลังในแต่ละครั้งมีจุดประสงค์ เงื่อนไขและความต้องการของผู้ใช้งานที่แตกต่างกันและไม่เหมือนการเลือกตั้งของหน่วยงานอื่นๆ เช่น การเลือกตั้งคณะกรรมการจริยธรรมฯ ผู้มีสิทธิ์เลือกตั้งจะต้องเป็นข้าราชการเท่านั้น ตำแหน่งอื่นไม่สามารถเลือกตั้งได้ และผู้มีสิทธิ์เลือกตั้งสามารถเลือกผู้สมัครได้ในแต่ละการเลือกตั้งไม่เท่ากัน เนื่องจากการเลือกตั้งในแต่ละครั้งมีคณะกรรมการที่หมดวาระไม่เท่ากัน ด้วยเหตุนี้จึงไม่สามารถนำระบบเลือกตั้งอื่นที่พัฒนาแล้วมาใช้กับการเลือกตั้งกระทรวงการคลังได้ อีกทั้ง งานดังกล่าวยังเป็นงานเดิมที่ต้องทำซ้ำๆ ทุกครั้งที่มีการเลือกตั้ง และการเลือกตั้งเป็นกระบวนการดำเนินงานที่ต้องการความโปร่งใส และตรวจสอบได้ แต่ในปัจจุบันวิธีการเลือกตั้งของสำนักงานปลัดกระทรวงการคลัง ทั้ง 2 วิธี เป็นวิธีการเลือกตั้งที่ผู้ไม่หวังดีสามารถแก้ไขข้อมูลได้โดยง่าย และอาจเกิดการทุจริต ความไม่เป็นธรรม ความไม่โปร่งใส ในการเลือกตั้งได้ อีกทั้งการเลือกตั้งที่ผู้มีสิทธิ์เลือกตั้งต้องเดินทางไปลงคะแนนเสียง ณ สถานที่ที่ได้จัดเตรียมไว้ หากผู้มีสิทธิ์เลือกตั้งไม่สามารถเดินทางมายังสถานที่ที่ได้จัดเตรียมไว้ได้ จะส่งผลให้ผู้มีสิทธิ์เลือกตั้งเสียสิทธิ์การลงคะแนนทันที

การพัฒนาระบบเลือกตั้งสำเร็จรูปบนบล็อกเชน สำหรับกระทรวงการคลัง (ระบบเลือกตั้งสำเร็จรูปฯ) เป็นวิธีการแก้ไขปัญหาที่กล่าวมาข้างต้น

## 2. งานวิจัยที่เกี่ยวข้อง

ผู้วิจัยศึกษาค้นคว้าแนวคิด ทฤษฎี ความรู้ จากเอกสารและงานวิจัยที่เกี่ยวข้อง เพื่อเป็นแนวทางในการดำเนินการวิจัย ดังนี้ smart contract เป็นสัญญาที่ไม่ได้ถูกบังคับใช้ด้วยกฎหมาย แต่จะขับเคลื่อนโดยอัตโนมัติด้วยฮาร์ดแวร์และซอฟต์แวร์ ที่ฝังอยู่ในสินค้าหรือบริการ ปัจจุบันแนวคิดของ smart contract ถูกนำมาใช้ใน บล็อกเชน เพื่อให้สามารถแลกเปลี่ยนสินทรัพย์ภายในบล็อกเชน ได้โดยอัตโนมัติ ซึ่ง smart contract และระเบียนที่เกี่ยวข้องจะถูกบันทึกลงในบล็อกเชน ทำให้ไม่สามารถเปลี่ยนแปลงได้ในภายหลัง [1]

บล็อกเชนเป็นหนึ่งในเทคโนโลยีจัดเก็บข้อมูลแบบกระจายศูนย์ ซึ่งเป็นเทคโนโลยีจัดเก็บข้อมูลแบบกระจายที่แชร์ข้อมูลกับทุก node ในเครือข่าย ซึ่งจะไม่ถูกควบคุมโดยบุคคลใด

บุคคลหนึ่งเพียงผู้เดียว โดยทุก node จะได้รับสำเนาของข้อมูล ซึ่งเมื่อมีการนำเข้าข้อมูลใหม่ ภายในแต่ละ node จะมีการอัพเดตข้อมูลโดยอัตโนมัติ โดยจะมีการเข้ารหัสข้อมูลก่อนอัพเดตข้อมูลลงในบล็อกเชน เพื่อความปลอดภัยของข้อมูล และทุก node ในเครือข่ายจะต้องยืนยันความถูกต้องของข้อมูลใหม่ที่เข้ามาก่อนอัพเดตข้อมูลลงในบล็อกเชน [2] ดังนั้น บล็อกเชนจึงเป็นเทคโนโลยีที่เหมาะสมในการพัฒนาระบบการเลือกตั้งที่ต้องการความถูกต้องของข้อมูล มีความโปร่งใส และสามารถตรวจสอบได้ โดยงานวิจัยของ Febriyanto et al. [3] นำเสนอวิธีการให้ผลลัพธ์ที่ถูกต้องและการตรวจสอบข้อมูลการลงคะแนนโดยใช้โทเคนในการลงคะแนน ซึ่งโทเคนสามารถใช้ได้เพียงครั้งเดียวและต้องใช้ภายในเวลาที่กำหนดเท่านั้น เมื่อทำการลงคะแนนเสียงโดยใช้โทเคนเรียบร้อยแล้ว ผลการเลือกตั้งแต่ละครั้งจะถูกเข้ารหัสโดยใช้เทคโนโลยีบล็อกเชนเพื่อความถูกต้องของข้อมูล จากความน่าเชื่อถือของผลการทดลองจากงานวิจัยชิ้นนี้พิสูจน์ว่าสามารถใช้เทคโนโลยีบล็อกเชน เพื่อสร้างระบบเลือกตั้งที่ช่วยปรับปรุงกระบวนการลงคะแนนและการคำนวณผลการเลือกตั้งได้อย่างรวดเร็ว ซึ่งคล้ายกับงานวิจัยของ Al-madani et al. [4] และ Hjálmarsson และ Hreiðarsson [5] ที่เสนอกระบวนการพัฒนาระบบเลือกตั้งที่จัดทำขึ้นบนพื้นฐานของเทคโนโลยี blockchain และงานวิจัยของ Malkawi , Yassein and Bataineh [6] ที่นำเสนอระบบเลือกตั้งที่สร้างขึ้นจากเทคโนโลยีบล็อกเชน ซึ่งผู้วิจัยได้นำเทคโนโลยีบล็อกเชนมาประยุกต์ใช้กับระบบการเลือกตั้งรัฐสภาจอร์แดน โดยใช้บล็อกเชนแบบปิด โดยบทความนี้ผู้เขียนได้แนะนำอัลกอริธึมใหม่สำหรับการปรับปรุงประสิทธิภาพของระบบการเลือกตั้งเมื่อมีการเพิ่มผู้มีสิทธิ์เลือกตั้งในบล็อกเชน, การตรวจสอบความถูกต้องของผู้มีสิทธิ์เลือกตั้ง และความถูกต้องของผลลัพธ์ โดยผู้วิจัยได้ทำการทดลองใช้ Quicksort Algorithm, insert into a sorted array algorithm และทดสอบโดยไม่ใช้อัลกอริธึมใดๆ ในการพัฒนาระบบ ซึ่งผลปรากฏว่าการใช้ insert into a sorted array algorithm ใช้เวลาในการดำเนินการน้อยที่สุด ส่งผลให้ผู้มีสิทธิ์เลือกตั้งใช้เวลาลงคะแนนเสียงน้อยลง ซึ่งเป็นการอำนวยความสะดวกรวดเร็วและเพิ่มประสิทธิภาพของระบบเลือกตั้ง

ทั้งนี้ การเรียงลำดับแบบเร็ว (Quicksort Algorithm) เป็นการเรียงลำดับข้อมูลโดยการแบ่งข้อมูลเป็นส่วนย่อยและเอาชนะ (Divide and Conquer) โดนอาศัยการเลือกข้อมูลมา 1 ตัว เรียกว่า pivot เพื่อใช้เป็นตัวแบ่งส่วนของข้อมูลออกเป็น 2 ส่วน คือ ส่วนที่น้อยกว่า pivot และส่วนที่มากกว่าหรือเท่ากับ pivot แล้วทำการเลือก pivot และแบ่งข้อมูลกับ 2 ส่วนย่อยซ้ำไปจนเหลือข้อมูลเพียงตัวเดียว เมื่อเทียบกับการเรียงลำดับแบบผสาน (Merge Sort) เป็นวิธีการเรียงข้อมูลขนาดใหญ่โดยใช้การแบ่งข้อมูลเป็นส่วนย่อยและเอาชนะ (Divide and Conquer) เช่นเดียวกัน ซึ่งการทำงานจะแบ่งข้อมูลขนาดใหญ่ให้มีขนาดเล็กที่มีความสมดุลกันของจำนวนข้อมูล แล้วนำข้อมูลที่มีขนาดเล็กมาเรียงข้อมูล จากนั้นนำข้อมูลมารวมกันเป็นข้อมูลเดียวที่ผ่านการเรียงข้อมูลแล้ว โดย Quicksort Algorithm ใช้การแบ่งข้อมูลเป็นส่วนย่อยและเอาชนะ (Divide and Conquer) เหมือนกับ Merge Sort แต่มีหลักการทำงานที่ต่างกัน ซึ่ง Quicksort Algorithm ทำงานได้ดีกับข้อมูลขนาดเล็ก ส่วน Merge Sort ทำงานได้ดีกับ ข้อมูลทุกขนาด [7,8,9]

การพัฒนาระบบเลือกตั้งโดยใช้บล็อกเชนมีงานวิจัยที่นำเทคนิคต่างๆ เข้ามาช่วยให้การเลือกตั้งมีความปลอดภัยมากยิ่งขึ้น เช่น Fernandes et al. [10] เสนอระบบการลงคะแนนอิเล็กทรอนิกส์แบบกระจายโดยใช้บล็อกเชน และใช้ secret contract เพื่อแก้ไขปัญหาการรักษาความเป็นส่วนตัว โดยการไม่เปิดเผยตัวตนของผู้มีสิทธิ์เลือกตั้งและการรักษาความลับของการลงคะแนน หรือการใช้เทคนิค Sidechain ในงานวิจัยของ Alvi et al. [11] เพื่อมาช่วยประหยัดค่า gas และลดเวลาการทำงานลง เนื่องจาก Sidechain คือ บล็อกเชนอีกเส้นหนึ่งที่ตั้งขึ้นมาเพื่อทำงานเพิ่มเติมจาก main chain และสามารถเชื่อมกลับไป main chain ได้

จากงานวิจัยที่กล่าวมาจะเห็นได้ว่า บล็อกเชนเป็นเทคโนโลยีที่ปลอดภัย มีความโปร่งใส เชื่อถือได้ ในการวิจัยนี้จึงได้นำบล็อกเชน เข้ามาใช้ในการพัฒนาระบบเลือกตั้งให้มีความโปร่งใส สามารถตรวจสอบได้ ซึ่งเป็นคุณสมบัติของบล็อกเชนโดยผู้พัฒนาระบบจะพัฒนาระบบเลือกตั้งสำเร็จรูปฯ ขึ้นมาแค่ครั้งเดียว ระบบเลือกตั้งดังกล่าวจะสามารถปรับใช้กับการเลือกตั้งได้ในหลายรูปแบบ โดยผู้ใช้งานสามารถกำหนดหรือจัดรูปแบบฟอร์มการเลือกตั้งได้ด้วยตนเองตามเงื่อนไขที่ต้องการ และอำนวยความสะดวกรวมเร็วในการเลือกตั้งผ่านระบบออนไลน์ไม่ต้องเดินทางไปสถานที่เลือกตั้ง

## 3. วิธีดำเนินการวิจัย

ผู้วิจัยได้วิเคราะห์ความต้องการของผู้ใช้งานและนำมาออกแบบกระบวนการทำงาน สถาปัตยกรรม โครงสร้างของข้อมูล และส่วนต่อประสานกับผู้ใช้งาน(GUI) สำหรับการพัฒนาระบบเลือกตั้งสำเร็จรูปฯ โดยมีรายละเอียดดังนี้

### 3.1 วิเคราะห์ระบบ

ผู้วิจัยได้ดำเนินการวิเคราะห์ปัญหาของระบบเดิมหรือ การดำเนินการแบบเดิม ซึ่งในปัจจุบันกระทรวงการคลัง มีการเลือกตั้ง 2 วิธี คือ (1) การลงคะแนนเสียงด้วยกระดาษ ณ สถานที่เลือกตั้ง และ (2) การเลือกตั้งผ่านระบบเดิมที่เก็บข้อมูลลงฐานข้อมูลแบบรวมศูนย์กลาง ซึ่งทั้ง 2 วิธีมีปัญหาในด้านของการทุจริต ความไม่โปร่งใส และไม่สามารถตรวจสอบได้ อีกทั้งการเลือกตั้งผ่านระบบเดิมจะต้องพัฒนาระบบเลือกตั้งใหม่ทุกครั้งที่มีการเลือกตั้งเนื่องจากการเลือกตั้งในแต่ละครั้ง มีเงื่อนไขที่แตกต่างกัน จึงมีแนวคิดในการแก้ไขปัญหา ดังนี้

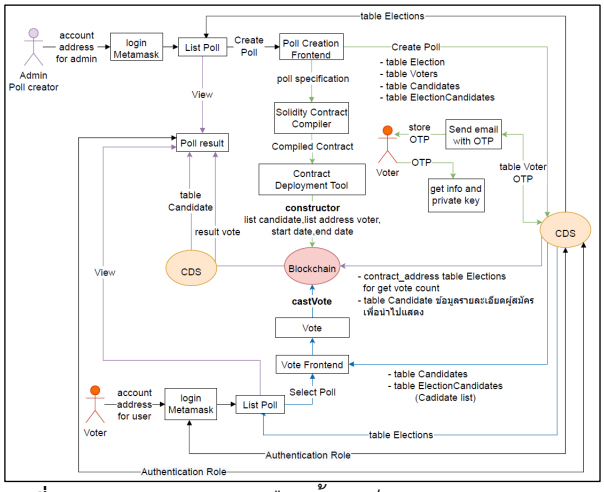
- พัฒนาระบบเลือกตั้งออนไลน์
- พัฒนาระบบเลือกตั้งสำเร็จรูปฯ ที่สามารถสร้างฟอร์มได้เองตามเงื่อนไขที่ผู้ใช้งานต้องการ
- พัฒนาระบบเลือกโดยใช้เทคโนโลยี บล็อกเชน

### 3.2 กระบวนการทำงานของระบบ

ระบบประกอบด้วยผู้ใช้งาน 2 กลุ่มคือ (1) กลุ่มผู้ใช้งานที่เป็นผู้ดูแลระบบ (2) กลุ่มผู้มีสิทธิ์เลือกตั้ง โดยผู้ใช้งานทั้ง 2 กลุ่มจะต้อง Login เข้าสู่ระบบ และระบบจะดำเนินการตรวจสอบสิทธิ์ที่แต่ละผู้ใช้งานได้รับ

กลุ่มผู้ใช้งานที่เป็น ผู้ดูแลระบบ เข้าสู่ระบบด้วย account address สำหรับผู้ดูแลระบบ โดยสามารถสร้างฟอร์มการเลือกตั้งตามเงื่อนไขที่ต้องการลงในระบบ จากนั้นระบบจะส่ง e-mail ที่ระบุ OTP ไปยังผู้มีสิทธิ์เลือกตั้งทั้งหมด เพื่อใช้ในการเข้าไปตรวจสอบข้อมูลที่ผู้มีสิทธิ์เลือกตั้งต้องใช้ในการเข้าสู่ระบบเลือกตั้งต่อไป ซึ่งช่วยให้ระบบมีความปลอดภัยมากยิ่งขึ้น อีกทั้งผู้ใช้งานที่เป็นผู้ดูแลระบบสามารถลบฟอร์มการเลือกตั้ง เรียกดูผลการเลือกตั้ง และส่ง e-mail ที่ระบุ OTP ให้ผู้มีสิทธิ์ได้

กลุ่มผู้ใช้งานที่เป็นผู้มีสิทธิ์เลือกตั้งสามารถเข้าสู่ระบบได้ โดยใช้ account address ของผู้มีสิทธิ์เลือกตั้ง ที่สามารถตรวจสอบได้จากหน้าตรวจสอบข้อมูลการเลือกตั้ง โดยผู้มีสิทธิ์เลือกตั้ง จะได้รับ e-mail ที่ระบุ OTP สำหรับใช้ในการเข้าสู่ระบบหน้าตรวจสอบข้อมูลการเลือกตั้ง หากผู้มีสิทธิ์เลือกตั้งที่ไม่เคยเข้าสู่ระบบเลือกตั้ง หรือ ลืม account address ที่ใช้สำหรับเข้าสู่ระบบเลือกตั้ง จะต้องนำ OTP ที่ได้รับ กรอกเข้าสู่หน้าตรวจสอบข้อมูลการเลือกตั้ง จากนั้นระบบจะแสดงข้อมูล การเลือกตั้งและ account address เพื่อนำ account address ที่ได้ไปใช้เข้าสู่ระบบเลือกตั้ง แต่หากทราบข้อมูลการเลือกตั้งและ account address แล้ว ไม่จำเป็นต้องเข้ามาหน้าตรวจสอบข้อมูลการเลือกตั้ง โดยสามารถเข้าสู่ระบบเลือกตั้ง และดูรายการเลือกตั้งที่ได้รับสิทธิ์ได้เลย ซึ่งสามารถลงคะแนนเสียงได้ 1 เสียง ต่อ 1 การเลือกตั้ง โดยไม่สามารถแก้ไขการลงคะแนนเสียงได้ และจะต้องเลือกตั้งได้ภายในระยะเวลากำหนดเท่านั้น อีกทั้งสามารถเรียกดูผลการเลือกตั้งได้ทันทีหลังจากลงคะแนนเสร็จตามสิทธิ์ที่ผู้ใช้งานได้รับ โดยมีภาพรวมการทำงานของระบบเลือกตั้งสำเร็จรูปฯ ดังรูปที่ 1

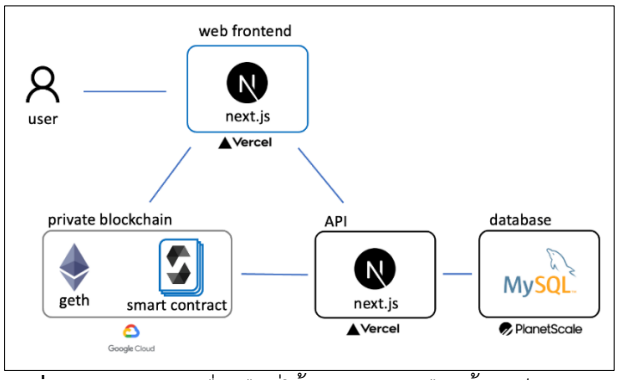



**รูปที่ 1** ภาพรวมของระบบเลือกตั้งสำเร็จรูปฯ

### 3.3 สถาปัตยกรรมของระบบ

ผู้วิจัยเลือกใช้คุณสมบัติเครื่องข่ายแบบปิดของ Ethereum ในการสร้างเครือข่ายบล็อกเชนเนื่องจาก Ethereum เป็นบล็อกเชนแพลตฟอร์มที่สามารถใช้งานได้โดยไม่มีค่าใช้จ่าย มีคู่มือและแหล่งเรียนรู้ที่เข้าถึงได้ง่าย อีกทั้งเป็นเครื่องมือที่มีความพร้อมใช้งานสูง และมีการใช้งานในวงกว้าง สามารถใช้ในการสร้างเครือข่ายบล็อกเชนใหม่และสามารถเลือกอัลกอริทึมที่ใช้ในกระบวนการ consensus ได้ตามความเหมาะสม โดยในการ

พัฒนาระบบเลือกตั้งสำเร็จรูปฯ ผู้วิจัยเลือกใช้อัลกอริทึม Clique ซึ่งเป็น consensus protocol ประเภทหนึ่งของ proof of authority (PoA) สำหรับการสร้างบล็อก เนื่องจากใช้พลังงานน้อย และใช้ชื่อเสียงของหน่วยงานเป็นเดิมพันในการสร้างบล็อก กล่าวคือ หน่วยงานจะต้องดูแลรักษาคอมพิวเตอร์ของตนเองให้มีความปลอดภัยเพื่อไม่ให้เสื่อมเสียชื่อเสียง จึงทำให้เป็นอัลกอริทึมที่เหมาะสมสำหรับการใช้งานในเครือข่ายที่ประกอบด้วยหน่วยงานภาครัฐ โดยผู้วิจัยพัฒนา frontend และ API ด้วย Next.js ซึ่งใช้บริการ Vercel เป็น static hosting website โดยจัดเก็บข้อมูลลงฐานข้อมูล MySQL ซึ่งใช้บริการ PlanetScale ในส่วนของบล็อกเชนแบบปิดที่ประกอบไปด้วย (1) smart contract ถูกพัฒนาโดยภาษา Solidity และใช้ Hardhat เป็น development environment สำหรับการพัฒนาบน Ethereum (2) Geth คือ node ของบล็อกเชน โดยนำขึ้นติดตั้งบน virtual machines ของ Google Cloud เพื่อใช้จำลองเครือข่ายเสมือนสำหรับการทดสอบ บล็อกเชน โดยมีภาพรวมของเครื่องมือที่ใช้ในการพัฒนาระบบเลือกตั้งสำเร็จรูปฯ ดังรูปที่ 2



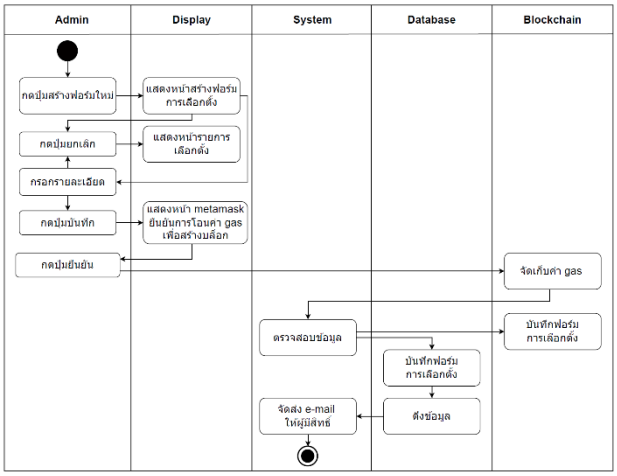

**รูปที่ 2** ภาพรวมของเครื่องมือที่ใช้พัฒนาระบบเลือกตั้งสำเร็จรูปฯ

### 3.3.1 ขั้นตอนการดำเนินงาน

การออกแบบขั้นตอนการดำเนินงาน โดยใช้แผนภาพกิจกรรม (activity diagram) เพื่อแสดงขั้นตอนการดำเนินงานในแต่ละขั้นตอน ดังนี้

#### 3.3.1.1 ระบบงานเลือกตั้งในส่วนของผู้ดูแลระบบ

ระบบเลือกตั้งในส่วนของผู้ดูแลระบบ เริ่มต้นด้วยการเข้าสู่ระบบด้วย account address จากนั้นระบบจะทำการตรวจสอบสิทธิ์ของผู้ใช้งาน ซึ่งหากตรวจสอบแล้วพบว่าผู้ใช้งานที่ได้รับสิทธิ์เป็นผู้ดูแลระบบจะสามารถเข้าสู่ระบบและระบบจะแสดงหน้าฟอร์มการเลือกตั้งทั้งหมดที่มีในระบบ โดยผู้ดูแลระบบสามารถดำเนินการลบฟอร์มการเลือกตั้ง, สร้างฟอร์มการเลือกตั้งใหม่, ส่ง e-mail ที่ระบุ OTP เพื่อใช้สำหรับเข้าไปตรวจสอบรายละเอียดการเลือกตั้งและ account address ให้ผู้ใช้งานที่มีสิทธิ์ในการเลือกตั้ง และดูผลการลงคะแนนได้ทั้งหมด โดยมีแผนภาพกิจกรรมของการสร้างฟอร์มการเลือกตั้ง ดังรูปที่ 3

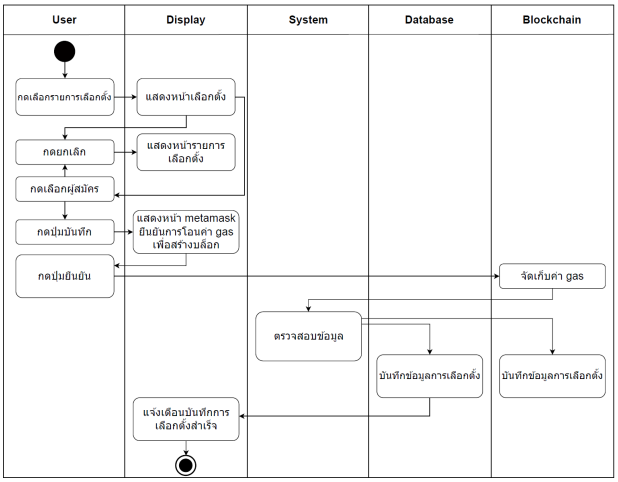



**รูปที่ 3** แผนภาพกิจกรรมของการสร้างฟอร์มการเลือกตั้ง

#### 3.3.1.2 ระบบงานเลือกตั้งในส่วนของผู้มีสิทธิ์เลือกตั้ง

ระบบเลือกตั้งในส่วนของผู้มีสิทธิ์เลือกตั้ง เริ่มต้นด้วยการที่ผู้มีสิทธิ์เลือกตั้งจะต้องได้รับ e-mail ที่ระบุ OTP เพื่อใช้สำหรับเข้าไปตรวจสอบรายละเอียดการเลือกตั้งที่ผู้ใช้งานได้รับสิทธิ์และตรวจสอบ account address เพื่อใช้สำหรับเข้าใช้งานระบบเลือกตั้ง ซึ่งหากผู้มีสิทธิ์เลือกตั้งที่ไม่เคยเข้าสู่ระบบเลือกตั้ง หรือลืม account address ที่ใช้สำหรับเข้าสู่ระบบเลือกตั้ง จะต้องนำ OTP ที่ได้รับ กรอกเข้าสู่หน้าตรวจสอบข้อมูลการเลือกตั้ง จากนั้นระบบจะแสดงข้อมูลการเลือกตั้งและ account address เพื่อนำ account address ที่ได้ไปใช้เข้าสู่ระบบเลือกตั้ง แต่หากทราบข้อมูลการเลือกตั้งและ account address แล้ว ไม่จำเป็นต้องเข้าหน้าตรวจสอบข้อมูลการเลือกตั้ง โดยสามารถเข้าสู่ระบบเลือกตั้งโดยใช้ account address ได้เลย โดยระบบจะแสดงหน้ารายการการเลือกตั้งที่ได้รับสิทธิ์ และสามารถ

ลงคะแนนเสียงหรือดูผลการเลือกตั้งได้ตามที่ได้รับสิทธิ์ โดยมีแผนภาพกิจกรรม ของการเลือกตั้ง ดังรูปที่ 4




**รูปที่ 4** แผนภาพกิจกรรมของการเลือกตั้ง

### 3.3.2 โครงสร้างข้อมูล

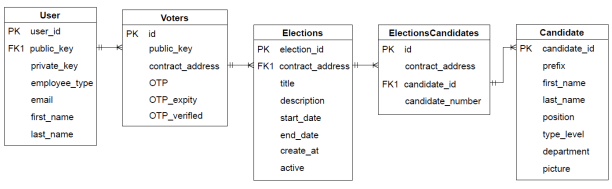
โครงสร้างข้อมูลของระบบเลือกตั้งสำเร็จรูปฯ ประกอบด้วย 2 ส่วน ดังนี้

#### 3.3.2.1 โครงสร้างข้อมูลของ smart contract

smart contract ที่พัฒนาขึ้นจะใช้ในการอ่าน-เขียนข้อมูลการเลือกตั้ง ซึ่งประกอบไปด้วย รายละเอียดเบื้องต้นของผู้สมัคร สถานะของผู้มีสิทธิ์เลือกตั้ง ผู้มีสิทธิ์เลือกตั้ง วัน เวลา เริ่มต้นและสิ้นสุดการเลือกตั้ง

#### 3.3.2.2 โครงสร้างฐานข้อมูล

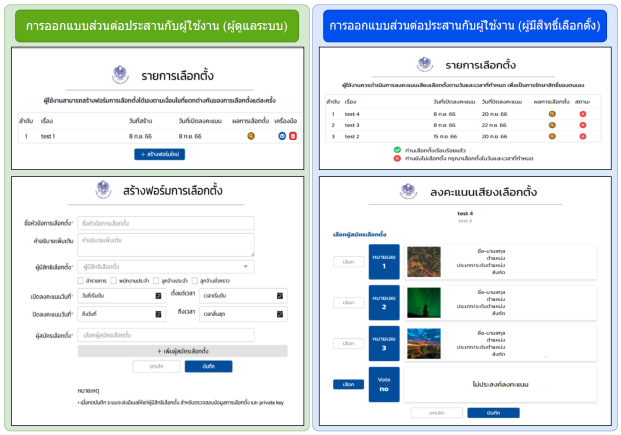
ฐานข้อมูลที่พัฒนาขึ้นใช้สำหรับอ่าน-เขียนข้อมูลผู้ใช้งาน ผู้มีสิทธิ์เลือกตั้ง รายละเอียดการเลือกตั้ง ผู้สมัครของแต่ละฟอร์ม และรายละเอียดผู้สมัครดังรูปที่ 5




**รูปที่ 5** โครงสร้างข้อมูลของฐานข้อมูล

### 3.3.3 การออกแบบส่วนต่อประสานกับผู้ใช้งาน (GUI)

การออกแบบส่วนต่อประสานกับผู้ใช้งานผ่านเครื่องมือ Figma เน้นการออกแบบที่ใช้งานง่ายสำหรับผู้ใช้งานที่มีหลายกลุ่มช่วงอายุ ซึ่งออกแบบส่วนต่อประสานกับผู้ใช้งานเป็น 2 ส่วนตามกลุ่มผู้ใช้งาน โดยมีตัวอย่างดังรูปที่ 6




**รูปที่ 6** ตัวอย่างการออกแบบส่วนต่อประสานกับผู้ใช้งาน

## 4. การพัฒนาระบบ

การพัฒนาระบบเลือกตั้งสำเร็จรูปฯ ประกอบไปด้วยการพัฒนา smart contract การพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้ (graphic user interface : GUI) การพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ (application programming interface : API) และการติดตั้งระบบ โดยมีรายละเอียดดังนี้

### 4.1 การพัฒนา smart contract

smart contract ที่พัฒนาขึ้นประกอบด้วยส่วนที่สำคัญ 4 ส่วน ได้แก่ การประกาศโครงสร้างข้อมูล การประกาศเหตุการณ์ การประกาศตัวแปร และการประกาศฟังก์ชัน โดยมีรายละเอียดดังนี้

#### 4.1.1 การประกาศโครงสร้างข้อมูล (structures)

เป็นการประกาศว่า smart contract นี้มีข้อมูลชนิดใดบ้าง และข้อมูลแต่ละชนิด มีโครงสร้างเป็นอย่างไร

#### 4.1.2 การประกาศเหตุการณ์ (events)

เป็นการประกาศเหตุการณ์ที่สามารถเกิดขึ้นได้ เช่น มีการลงคะแนนเสียงเลือกตั้ง เป็นต้น ซึ่งเป็นกลไกของ smart contract ที่ใช้ในการลงบันทึก (log) เหตุการณ์ลงบนบล็อกเชน และสามารถตอบสนองต่อเหตุการณ์ได้ตามความเหมาะสม เช่น

บันทึกข้อมูลที่ได้จากเหตุการณ์ ในระบบจัดการฐานข้อมูล หรือแจ้งเตือนให้ผู้ใช้ทราบ เป็นต้น รูปที่ 7 แสดงให้เห็นถึงการประกาศเหตุการณ์ VoteCast (มีการลงคะแนนเสียงเลือกตั้ง) ข้อมูลที่เป็นส่วนประกอบของเหตุการณ์ และการแพร่กระจาย (Emit) เหตุการณ์ในฟังก์ชัน castVote โดยใช้คำสั่ง Emit

```
pragma solidity ^0.8.9;
contract Election {
    // ...
    event VoteCast(uint256 indexed candidateId);
    // ...
    function castVote(uint256 _candidateId) public votingOpen {
      // ...
        emit VoteCast(_candidateId);
    }
}
```

**รูปที่ 7** ตัวอย่างการประกาศและแพร่กระจายเหตุการณ์

### 4.1.3 การประกาศตัวแปร (Variables)

เป็นการประกาศว่า smart contract นี้มีการจัดเก็บข้อมูลอะไรบ้าง โดยตัวแปรที่ประกาศภายใต้ contract โดยตรงจะใช้สำหรับการจัดเก็บข้อมูลลงใน บล็อกเชน ส่วนตัวแปรที่ประกาศในฟังก์ชันจะเป็นตัวแปรชั่วคราว

### 4.1.4 การประกาศฟังก์ชัน (Functions)

เป็นการประกาศว่า Smart contract นี้สามารถทำอะไรได้บ้าง จากรูปที่ 8 จะเห็นได้ว่ามีการประกาศฟังก์ชัน castVote ที่ใช้สำหรับการลงคะแนนเสียงเลือกตั้ง ซึ่งรับค่า ID ของผู้สมัคร โดยขั้นตอนการทำงาน ดังนี้

- ตรวจสอบว่าผู้ลงคะแนนเสียงจะต้องเป็นผู้มีสิทธิ์เลือกตั้งในฟอร์มนั้น ๆ จึงจะสามารถไปตรวจสอบเงื่อนไขต่อไปได้
- ตรวจสอบสถานะของผู้ลงคะแนนเสียงจะต้องไม่ลงคะแนนเสียงในฟอร์มการเลือกตั้งนั้น ๆ มาแล้ว จึงจะสามารถไปตรวจสอบเงื่อนไขต่อไปได้
- ตรวจสอบว่า ID ของผู้สมัครที่เลือกมามีอยู่ในรายการผู้สมัครของฟอร์มนั้น ๆ จึงจะสามารถไปตรวจสอบเงื่อนไขต่อไปได้
- เมื่อตรวจสอบผ่านทุกเงื่อนไขจะสามารถเพิ่มคะแนนให้กับผู้สมัครที่เลือกลงคะแนนและกำหนดสถานะของผู้มีสิทธิ์เลือกตั้งที่ลงคะแนนเสียงแล้วเป็น True และแพร่กระจายเหตุการณ์ VoteCast

```
pragma solidity ^0.8.9;
contract Election {
    // ...
    function castVote(uint256 _candidateId) public votingOpen {
        require(
            authorizedVoters[msg.sender],
            "You are not authorized to vote."
        );
        require(!voters[msg.sender], "You have already voted.");
        require(
            _candidateId > 0 && _candidateId <= candidates.length,
            "Invalid candidate ID."
        );
        candidates[_candidateId - 1].voteCount++;
        voters[msg.sender] = true;
        emit VoteCast(_candidateId);
    }
    // ...
}
```

**รูปที่ 8** ตัวอย่างการประกาศฟังก์ชันโดยใช้ภาษา Solidity

## 4.2 การพัฒนาส่วนต่อประสานกราฟิกกับผู้ใช้

ส่วนต่อประสานกราฟิกกับผู้ใช้ (graphic user interface : GUI) ในระบบเลือกตั้งสำเร็จรูปฯ เป็นเครื่องมือหลักที่ทำให้ผู้ใช้สามารถเข้าถึงบล็อกเชนและแหล่งข้อมูลที่เกี่ยวข้องได้ด้วยความสะดวก ผู้วิจัยได้ใช้ภาษา JavaScript เป็นภาษาหลักในการพัฒนา โดยใช้คลังโปรแกรม NextJS ซึ่งเป็นคลังโปรแกรม React ที่ช่วยให้ทำงานสะดวก และยืดหยุ่นมากยิ่งขึ้น โดยใช้เทคนิค server side Rendering ในการสร้าง hyperText markup language (HTML) ให้พร้อมที่ฝั่ง server แล้วส่งไปที่ client ให้พร้อมใช้งานได้ทันที ทำให้โหลดเว็บไซต์ได้เร็วขึ้น

นอกจากนี้ผู้วิจัยยังใช้คลังโปรแกรม web3 เป็นคลังโปรแกรมสำหรับใช้ในการประสานกับบล็อกเชน และ smart contract ที่ติดตั้งอยู่ในบล็อกเชน โดยคลังโปรแกรมนี้จะทำงานได้บนเว็บบราวเซอร์ที่เข้ากันได้กับ Ethereum หรือมีการเปิดใช้งานส่วนขยาย เช่น MetaMask หรือ Coinbase Wallet เป็นต้น โดยเมื่อมีการประสานกับบล็อกเชน บราวเซอร์จะให้ผู้ใช้ยืนยันตัวตนก่อนการทำธุรกรรม

## 4.3 การพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ การพัฒนาส่วนต่อประสานโปรแกรมประยุกต์ (application programming interface : API)

ผู้วิจัยใช้คลังโปรแกรม Next.js ในการพัฒนา ซึ่ง Next.js สามารถทำตัวเป็น backend APIs ได้เลย โดยที่ไม่ต้องมี server เพิ่ม เพียงแค่สร้างโฟลเดอร์ API ไว้ภายในโฟลเดอร์ pages ตัวอย่างเช่น

- /api/create-form สร้างฟอร์มการเลือกตั้ง

- /api/otp-validate ตรวจสอบเงื่อนไขของ OTP ที่ผู้ใช้งานระบุ และเปลี่ยนสถานะการใช้งาน
- /api/send-gas ส่งเหรียญ ether ให้กับผู้มีสิทธิ์เลือกตั้งเพื่อใช้เป็นค่า gas

## 5. การทดสอบและการประเมินผล

ในการพัฒนาระบบเลือกตั้งสำเร็จรูปฯ มีการทดสอบประเมินผล ซึ่งประกอบไปด้วย (1) การทดสอบประสิทธิภาพการทำงาน (performance testing) (2) การทดสอบหน่วยย่อย (unit testing) และ (3) การประเมินคุณสมบัติของระบบ โดยมีรายละเอียดดังนี้

### 5.1 การทดสอบประสิทธิภาพการทำงานของระบบ

#### 5.1.1 การทดสอบเวลาที่ใช้สำหรับการบันทึกและอ่านข้อมูลจากบล็อกเชนและจากฐานข้อมูลแบบรวมศูนย์กลางของระบบดั้งเดิม

ผู้วิจัยได้จัดเตรียมชุดคำสั่งสำหรับการทดสอบประสิทธิภาพการทำงานของระบบในด้านการทดสอบเวลาที่ใช้สำหรับการ deploy smart contract , บันทึก และดึงข้อมูลจากบล็อกเชน ซึ่งมีขั้นตอนการทำงานตามตัวอย่างชุดคำสั่งในรูปที่ 9

```
...
   it('castVote() time', async function () {
...
     await election.deployed();
     const startTime = Date.now();
     await election.connect(account1).castVote(1);
     const endTime = Date.now();
     const timeTaken = (endTime - startTime) / 1000;
     console.log(`castVote() time: ${timeTaken} seconds`);
   })
```

**รูปที่ 9** ตัวอย่างชุดคำสั่งที่ใช้ในการทดสอบเวลาที่ใช้ในการบันทึกข้อมูล

จากการทดสอบ พบว่าระบบเลือกตั้งสำเร็จรูปฯ ใช้เวลาในการ deploy smart contract เฉลี่ย 0.076 วินาที, เวลาที่ใช้ในการเขียนข้อมูลลงบล็อกเชน โดยทดสอบการเรียกใช้ฟังก์ชั่น castvote() ซึ่งเป็นฟังก์ชั่นสำหรับการบันทึกคะแนนการเลือกตั้งลงในบล็อกเชน เฉลี่ย 0.299 วินาที และเวลาที่ใช้ในการอ่านข้อมูลจากบล็อกเชน โดยทดสอบการเรียกใช้ฟังก์ชั่น getVoteCount() ซึ่งเป็นฟังก์ชั่นสำหรับการอ่านข้อมูลคะแนนการเลือกตั้งจาก บล็อกเชนเฉลี่ย 0.078 วินาที โดยเวลาที่ใช้ในการเขียนข้อมูลลงบล็อกเชนจะมากกว่าเวลาที่ใช้ในการอ่านข้อมูลจากบล็อกเชน เนื่องจากการเขียนข้อมูลลงบล็อกเชนได้จะต้องดำเนินการตามขั้นตอน ได้แก่ create , broadcast , validation และ add to chain โดยในส่วนของค่าส่วนเบี่ยงแบนมาตรฐานของเวลาที่ใช้ในการ deploy Smart contract , บันทึก และดึงข้อมูลจากบล็อกเชน มีค่าน้อยมากแสดงให้เห็นว่าผลการทดลองในแต่ละครั้งมีค่าใกล้เคียงกับค่าเฉลี่ย ซึ่งแสดงให้เห็นถึงความแม่นยำในการทดลอง โดยมีผลการทดสอบตามตารางที่ 1

**ตารางที่ 1** ผลการทดสอบเวลาที่ใช้สำหรับบันทึกและอ่านข้อมูลจากบล็อกเชน

| ครั้งที่ | เวลาที่ระบบใช้ deploy smart contract (วินาที) | เวลาที่ระบบใช้เขียนข้อมูล (วินาที) | เวลาที่ระบบใช้อ่านข้อมูล (วินาที) |
|---|---|---|---|
| 1 | 0.074 | 0.292 | 0.078 |
| 2 | 0.076 | 0.297 | 0.078 |
| 3 | 0.080 | 0.309 | 0.079 |
| 4 | 0.078 | 0.306 | 0.079 |
| 5 | 0.073 | 0.293 | 0.074 |
| **ค่าเฉลี่ย ($\bar{x}$)** | **0.076** | **0.299** | **0.078** |
| **ค่าส่วนเบี่ยงเบนมาตรฐาน ($\mu$)** | **0.00286** | **0.00770** | **0.00207** |

และผู้วิจัยได้ทดสอบประสิทธิภาพการทำงานของการลงคะแนนเสียงเลือกตั้งแบบดั้งเดิมที่มีการเก็บข้อมูลแบบรวมศูนย์ โดยในการพัฒนาระบบเลือกตั้งแบบดั้งเดิมแต่ละครั้งจะใช้เวลาในการพัฒนาระบบประมาณ 1 เดือน เพื่อให้ได้ระบบเลือกตั้งที่ตรงตามวัตถุประสงค์ในการเลือกตั้งแต่ละครั้งที่แตกต่างกันไป และจากการทดสอบ พบว่า ระบบเลือกตั้งแบบดั้งเดิมใช้เวลาในการเขียนข้อมูลลงฐานข้อมูลแบบรวมศูนย์กลางที่มีคอลัมน์จำนวน 20 คอลัมน์ โดยทดสอบการเรียกใช้ฟังก์ชั่นสำหรับการบันทึกคะแนนการเลือกตั้งลงในฐานข้อมูลแบบรวมศูนย์กลางเฉลี่ย 0.028 วินาที และเวลาที่ใช้ในการอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลางโดยทดสอบการเรียกใช้ฟังก์ชั่นสำหรับการอ่านข้อมูลคะแนนการเลือกตั้งจากฐานข้อมูลแบบรวมศูนย์กลาง เฉลี่ย 0.089 วินาที ซึ่งเวลาที่ใช้ในการเขียนข้อมูลลงฐานข้อมูลแบบรวมศูนย์กลางจะน้อยกว่าเวลาที่ใช้ในการอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลาง เนื่องจากการดึงข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลางในฐานข้อมูลขนาดใหญ่ระบบจัดการฐานข้อมูลจะต้องค้นหาข้อมูลจำนวนมากเพื่อดึงข้อมูลที่ถูกต้อง และต้องดึงข้อมูลประกอบกันหลายตารางเพื่อใช้ในการแสดงผล ซึ่งมีผลการทดสอบตามตารางที่ 2

จากการทดสอบเวลาที่ใช้สำหรับการบันทึกและอ่านข้อมูลจากบล็อกเชนและฐานข้อมูลแบบรวมศูนย์กลางของระบบดั้งเดิม พบว่า

เวลาที่ใช้สำหรับการบันทึกข้อมูลลงบล็อกเชนใช้เวลามากกว่าเวลาที่ใช้ในการบันทึกข้อมูลลงฐานข้อมูลแบบรวมศูนย์กลางของระบบดั้งเดิม เนื่องจากการบันทึกข้อมูลลงบล็อกเชนมีกระบวนการตรวจสอบความถูกต้องของข้อมูลก่อนบันทึกลงบล็อกเชน ซึ่งทำให้ระบบเลือกตั้งสำเร็จรูปฯ ที่ใช้บล็อกเชนมีคุณสมบัติตามตารางที่ 4 และการอ่านข้อมูลจากบล็อกเชนใช้เวลาน้อยกว่าเวลาที่ใช้ในการอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลาง เนื่องจากการดึงข้อมูลจากฐานข้อมูลที่มีข้อมูลจำนวนมากและต้องดึงข้อมูลจากหลายตารางมาประกอบกันเพื่อใช้ในการแสดงผล

**ตารางที่ 2** ผลการทดสอบเวลาที่ใช้สำหรับบันทึกและอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลาง

| ครั้งที่ | เวลาที่ระบบใช้เขียนข้อมูล (วินาที) | เวลาที่ระบบใช้อ่านข้อมูล (วินาที) |
|---|---|---|
| 1 | 0.030 | 0.090 |
| 2 | 0.025 | 0.088 |
| 3 | 0.029 | 0.089 |
| 4 | 0.030 | 0.091 |
| 5 | 0.027 | 0.088 |
| **ค่าเฉลี่ย ($\bar{x}$)** | **0.028** | **0.089** |
| **ค่าส่วนเบี่ยงเบนมาตรฐาน ($\mu$)** | **0.00217** | **0.00130** |

### 5.1.2 การทดสอบจำนวนธุรกรรมที่บล็อกเชนของระบบรับได้มากที่สุดใน 1 วินาที

ผู้วิจัยได้จัดเตรียมชุดคำสั่งสำหรับการทดสอบประสิทธิภาพการทำงานของระบบด้านการทดสอบจำนวนธุรกรรมที่ บล็อกเชนของระบบรองรับได้มากที่สุดในเวลา 1 วินาที ซึ่งมีขั้นตอนการทำงานตาม ตัวอย่างชุดคำสั่งในรูปที่ 10

จากการทดสอบ พบว่า ระบบเลือกตั้งสำเร็จรูปฯ รองรับการทำรายการธุรกรรมจำนวน 100 รายการ ได้โดยใช้เวลาเฉลี่ย 0.859 วินาที ซึ่งระบบสามารถรองรับการทำรายการเฉลี่ย 116.443 รายการใน 1 วินาที โดยในส่วนของค่าส่วนเบี่ยงแบนมาตรฐานของเวลาที่ใช้ระบบใช้ในการทำธุรกรรม 100 รายการและจำนวนธุรกรรมที่ระบบรองรับได้ใน 1 วินาที มีค่าน้อยมากแสดงให้เห็นว่าผลการทดลองในแต่ละครั้งมีค่าใกล้เคียงกับค่าเฉลี่ย ซึ่งแสดงให้เห็นถึงความแม่นยำในการทดลอง โดยมีผลการทดสอบตามตารางที่ 3

```
const NUM_TRANSACTIONS = 100;
...
 const promises = Array.from({ length: NUM_TRANSACTIONS }).map((_, index) =>
   sendTransaction(PRIVATE_KEY, currentNonce + index)
 );
 const startTime = Date.now();
 try {
   const transactions = await Promise.all(promises);
   const confirmationPromises = transactions.map((tx) =>
    waitForConfirmation(tx.hash)
   );
   await Promise.all(confirmationPromises);
   const endTime = Date.now();
...
```

**รูปที่ 10** ตัวอย่างชุดคำสั่งที่ใช้ในการทดสอบจำนวนธุรกรรมที่บล็อกเชนของระบบรับได้มากที่สุดใน 1 วินาที

**ตารางที่ 3** ผลการทดสอบจำนวนธุรกรรมที่ บล็อกเชน ของระบบรับได้มากที่สุดใน 1 วินาที

| ครั้งที่ | เวลาที่ระบบใช้ในการทำธุรกรรม จำนวน 100 รายการ (วินาที) | จำนวนธุรกรรมที่ระบบรองรับได้ใน 1 วินาที (รายการ) |
|---|---|---|
| 1 | 0.858 | 116.550 |
| 2 | 0.862 | 116.009 |
| 3 | 0.863 | 115.875 |
| 4 | 0.857 | 116.686 |
| 5 | 0.854 | 117.096 |
| **ค่าเฉลี่ย ($\bar{x}$)** | **0.859** | **116.443** |
| **ค่าส่วนเบี่ยงเบนมาตรฐาน ($\mu$)** | **0.004** | **0.502** |

จากการทดสอบ พบว่า ระบบเลือกตั้งสำเร็จรูปฯ รองรับการทำรายการธุรกรรมจำนวน 100 รายการ ได้โดยใช้เวลาเฉลี่ย 0.859 วินาที ซึ่งระบบสามารถรองรับการทำรายการเฉลี่ย 116.443 รายการ

ใน 1 วินาที โดยในส่วนของค่าส่วนเบี่ยงแบนมาตรฐานของเวลาที่ใช้ระบบใช้ในการทำธุรกรรม 100 รายการและจำนวนธุรกรรมที่ระบบรองรับได้ใน 1 วินาที มีค่าน้อยมากแสดงให้เห็นว่าผลการทดลองในแต่ละครั้งมีค่าใกล้เคียงกับค่าเฉลี่ย ซึ่งแสดงให้เห็นถึงความแม่นยำในการทดลอง โดยมีผลการทดสอบตามตารางที่ 3

### 5.2 การทดสอบหน่วยย่อย

smart contract ที่ติดตั้งลงใน บล็อกเชน แล้วจะไม่สามารถแก้ไขได้ ดังนั้น การทดสอบหน่วยย่อย (unit testing) จึงเป็นกระบวนการหนึ่งที่ช่วยให้มั่นใจว่า smart contract จะสามารถทำงานได้อย่างถูกต้องเมื่อนำไปติดตั้งลงใน บล็อกเชน โดยปัจจุบันมีภาษาและเครื่องมือสำหรับการทำสอบหน่วนย่อยสำหรับ smart contract ให้เลือกใช้ได้ตามความถนัด ผู้วิจัยเลือกใช้ภาษา Typescript และเครื่องมือ Hardhat ในการทดสอบหน่วยย่อย โดยเขียนชุดคำสั่งสำหรับการทดสอบหน่วยย่อยอัตโนมัติ (automated testing)

โดยผู้วิจัยได้จัดเตรียมกรณีทดสอบ (test case) สำหรับการทดสอบหน่วยย่อยสำหรับ smart contract ทั้งสิ้น 15 รายการ ดังนี้

1) สามารถสร้างฟอร์มการเลือกตั้งได้
2) สามารถแสดงรายการฟอร์มการเลือกตั้งได้ถูกต้องตามสิทธิ์ที่ได้รับ
3) สามารถเพิ่มผู้มีสิทธิ์เลือกตั้งในแต่ละฟอร์มได้
4) สามาถเพิ่มผู้สมัครในแต่ละฟอร์มได้
5) สามารถกำหนดวันและเวลาในการเลือกตั้งได้
6) ตรวจสอบสิทธิ์ของผู้มีสิทธิ์ก่อนทำการเลือกตั้งได้
7) สามารถตรวจสอบได้ว่าผู้มีสิทธิ์ใช้สิทธิ์ไปแล้วหรือยังได้
8) สามารถเลือกลงคะแนนเสียงได้
9) ผู้มีสิทธิ์เลือกตั้งสามารถเลือกตั้งได้ในช่วงระยะเวลาที่กำหนด
10) สามารถตรวจสอบได้ว่าหมายเลขที่ลงคะแนนเสียงเป็นหมายเลขผู้สมัครที่อยู่ในฟอร์มนั้น ๆ
11) สามารถเปลี่ยนสถานะของผู้มีสิทธิ์เลือกตั้งที่ได้เลือกตั้งไปแล้วได้
12) สามารถแสดงผลการเลือกตั้งได้ถูกต้อง
13) สามารถเรียกดูวันและเวลาที่เริ่มต้นของการเลือกตั้งในฟอร์มนั้นๆ ได้
14) สามารถเรียกดูวันและเวลาที่สิ้นสุดของการเลือกตั้งในฟอร์มนั้นๆ ได้
15) สามารถเรียกดูสถานะการเลือกตั้งของผู้มีสิทธิ์เลือกตั้งได้

โดยการทดสอบหน่วยย่อย (unit testing) ทั้ง 15 รายการ พบว่าผลการทดสอบ ผ่าน ทั้ง 15 รายการ โดยระบบเลือกตั้งสำเร็จรูปฯ สามารถดำเนินการได้ถูกต้องตรงตามผลการทดสอบที่คาดหวังทั้ง 15 รายการ

### 5.3 การประเมินคุณสมบัติของระบบ

ผู้วิจัยได้ประเมินคุณสมบัติของระบบเลือกตั้งสำเร็จรูปฯ โดยเปรียบเทียบระบบเลือกตั้งสำเร็จรูปฯ กับ และระบบเลือกตั้งดั้งเดิม ตามตารางที่ 4 ระบบเลือกตั้งสำเร็จรูปฯ หมายถึง ระบบเลือกตั้งที่มีการนำเทคโนโลยีบล็อกเชนมาใช้ในการเก็บข้อมูลการเลือกตั้ง ระบบเลือกตั้งดั้งเดิม หมายถึง ระบบเลือตั้งที่จัดเก็บข้อมูลในระบบจัดการฐานข้อมูลแบบรวมศูนย์ โดยมีผู้พัฒนาระบบเป็นผู้จัดการข้อมูลในฐานข้อมูลกลางด้วยตนเอง

**ตารางที่ 4** เปรียบเทียบคุณสมบัติของระบบเลือกตั้งสำเร็จรูปฯ และระบบดั้งเดิม

| คุณสมบัติ | ระบบเลือกตั้งสำเร็จรูปฯ | ระบบเลือกตั้งดังเดิม |
|---|---|---|
| ความสอดคล้องของข้อมูล | มี | ไม่มี |
| ความสมบูรณ์และความเที่ยงตรงของข้อมูล | มี | ไม่มี |
| ความโปร่งใสในการเข้าถึงข้อมูล | มี | ไม่มี |
| ความสามารถในการทำงานอย่างต่อเนื่องของระบบ | มากกว่า | น้อยกว่า |
| ความคงทนของข้อมูล | มากกว่า | น้อยกว่า |
| ค่าใช้จ่ายในการพัฒนา | มากกว่า | น้อยกว่า |
| ความท้าทายในการเริ่มต้นใช้งาน | มากกว่า | น้อยกว่า |
| ส่วนต่อประสานกับผู้ใช้ (user interface) | มี | มี |
| ประสบการณ์ของผู้ใช้ เช่น ความสะดวกสบายใช้งานง่าย (user experience) | มี | มี |

หมายเหตุ : “มากกว่า” และ “น้อยกว่า” คือ การเปรียบเทียบระหว่าง ระบบเลือกตั้งสำเร็จรูปฯ และระบบเลือกตั้งแบบดั้งเดิม

จากตารางที่ 4 เป็นการเปรียบเทียบคุณสมบัติของ 2 ระบบ ได้แก่ ระบบเลือกตั้งสำเร็จรูปฯ และระบบเลือกตั้งแบบดั้งเดิม ซึ่งจะเห็นได้ว่าระบบเลือกตั้งสำเร็จรูปฯ มีความสอดคล้องของข้อมูล (consistency) เนื่องจากบล็อกเชนเปรียบเสมือนเป็นฐานข้อมูลที่ผู้เข้าร่วมเครือข่ายทุกรายสามารถเข้าถึงได้ และมีกระบวนการตรวจสอบธุรกรรมก่อนที่จะบันทึกลงบล็อกเชน นอกจากนี้จะเห็นได้ว่าระบบเลือกตั้งสำเร็จรูปฯ ที่ใช้บล็อกเชน มีการจัดเก็บข้อมูลในรูปแบบของบล็อก โดยเชื่อมต่อแต่ละบล็อกด้วย hash function และกระจายให้ทุก ๆ node เก็บ ทำให้เกิดคุณสมบัติที่สำคัญของบล็อกเชน 3 ประการ คือ 1) ความถูกต้องเที่ยงตรงของข้อมูล

(data Integrity) เนื่องจากการเชื่อมโยงบล็อกปัจจุบันและ บล็อกก่อนหน้าด้วย hash function และทำการกระจายให้ทุก node เก็บ ทำให้ข้อมูลที่ถูกบันทึกลงในบล็อกเชนแล้วไม่สามารถแก้ไขหรือเปลี่ยนแปลงข้อมูลได้ (immutability) 2) ความโปร่งใสในการเข้าถึงข้อมูล (data transparency) เนื่องจากทุก node ในระบบบล็อกเชน จะเก็บข้อมูลเดียวกันทั้งหมดโดยไม่มี node ใด node หนึ่งเป็นตัวกลางที่มีอำนาจแต่เพียงผู้เดียวในการเก็บข้อมูล ดังนั้น การเข้าถึงข้อมูลใด ๆ จึงทำได้จาก node ตัวเองทันทีโดยไม่จำเป็นต้องร้องขอข้อมูลจากตัวกลาง และ 3) ความสามารถในการทำงานได้อย่างต่อเนื่องของระบบ (availability) เนื่องจากทุก node ในระบบบล็อกเชน จะเก็บข้อมูลเดียวกันทั้งหมดจึงสามารถทำงานทดแทนกันได้เมื่อมี node ที่ไม่สามารถให้บริการได้ในขณะนั้นโดยระบบจะทำการคัดลอกสำเนาข้อมูลให้เป็นข้อมูลชุดเดียวกันเมื่อ node กลับขึ้นมาให้บริการได้อีกครั้ง และโดยทั่วไปในบล็อกเชนจะสร้างสำเนาข้อมูลบล็อกเชนไว้ในคอมพิวเตอร์ที่เป็นส่วนหนึ่งของเครือข่าย ส่งผลให้มีสำเนาข้อมูลหลายแหล่ง เป็นปัจจัยหลักที่ทำให้ความคงทนของข้อมูล (resiliency) ในบล็อกเชนมีสูง

## 6. สรุปผลการวิจัย

ผู้วิจัยได้นำเสนอระบบเลือกตั้งสำเร็จรูปฯ ซึ่งจะช่วยให้ผู้พัฒนาระบบเลือกตั้งเพียงครั้งเดียวและสามารถปรับใช้ได้กับทุกการเลือกตั้ง ซึ่งช่วยลดระยะเวลาในการพัฒนาระบบและลดภาระงานเดิมๆ ที่ต้องพัฒนาระบบใหม่ทุกครั้งที่มีการเลือกตั้ง และได้ระบบเลือกตั้งที่ตรงต่อความต้องการของผู้ใช้งาน และได้ระบบเลือกตั้งที่มีความโปร่งใส ตรวจสอบได้ โดยผู้วิจัยได้วิเคราะห์ปัญหาและความต้องการของผู้ใช้งานเพื่อนำมาออกแบบสถาปัตยกรรม กระบวนการทำงาน โครงสร้างข้อมูล และส่วนต่อประสานกับผู้ใช้ของระบบ พร้อมทั้งอธิบายขั้นตอนการพัฒนา smart contract ส่วนต่อประสานกราฟิกกับผู้ใช้ (user interface) ส่วนตัวประสานโปรแกรมประยุกต์ (application programing interface) โดยผู้วิจัยเลือกใช้คุณสมบัติของเครือข่ายแบบปิดของ Ethereum ในการสร้างเครือข่ายบล็อกเชน และเลือกใช้อัลกอริทึม Clique ซึ่งเป็นอัลกอริทึมประเภท PoA สำหรับกระบวนการสร้างบล็อก นอกจากนี้ ผู้วิจัยยังได้ทดสอบระบบ ได้แก่ การทดสอบประสิทธิภาพของระบบ แบ่งออกเป็นการทดสอบเวลาที่ใช้สำหรับการบันทึกและดึงข้อมูลจากบล็อกเชนและจากฐานข้อมูลแบบรวมศูนย์กลางของระบบดั้งเดิม โดยพบว่า การบันทึกข้อมูลลงบล็อกเชนใช้เวลาในการทำธุรกรรมเฉลี่ย 0.299 วินาที ซึ่งมากกว่าการอ่านข้อมูลจาก บล็อกเชน ที่ใช้เวลาในการทำธุรกรรมเฉลี่ย 0.078 วินาที เนื่องจาก การบันทึกข้อมูลลงบล็อกเชนจะต้องมีขั้นตอนและกระบวนการในการตรวจสอบความถูกต้องก่อนบันทึกข้อมูลลงบล็อกเชน ในส่วนของเวลาในการเขียนข้อมูลลงฐานข้อมูลแบบรวมศูนย์กลางของระบบดั้งเดิม เฉลี่ย 0.028 วินาที ซึ่งน้อยกว่าเวลาที่ใช้ในการอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลางที่ใช้เวลาเฉลี่ย 0.089 วินาทีเนื่องจากการดึงข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลางในฐานข้อมูลขนาดใหญ่ระบบจัดการฐานข้อมูลจะต้องค้นหาข้อมูลจำนวนมากเพื่อดึงข้อมูลที่ถูกต้อง และต้องดึงข้อมูลประกอบกันหลายตารางเพื่อใช้ในการแสดงผล ดังนั้น เวลาที่ใช้สำหรับการบันทึกข้อมูลลงบล็อกเชนใช้เวลามากกว่าเวลาที่ใช้ในการบันทึกข้อมูลลงฐานข้อมูลแบบรวมศูนย์กลางของระบบดั้งเดิม เนื่องจากการบันทึกข้อมูลลงบล็อกเชนมีกระบวนการตรวจสอบความถูกต้องของข้อมูลก่อนบันทึก และการอ่านข้อมูลจากบล็อกเชนใช้เวลาน้อยกว่าเวลาที่ใช้ในการอ่านข้อมูลจากฐานข้อมูลแบบรวมศูนย์กลาง เนื่องจากการดึงข้อมูลจากฐานข้อมูลที่มีข้อมูลจำนวนมากและต้องดึงข้อมูลจากหลายตารางมาประกอบกันเพื่อใช้ในการแสดงผล และการทดสอบจำนวนธุรกรรมที่บล็อกเชนสามารถรองรับได้มากที่สุดใน 1 วินาที พบว่าระบบรองรับการทำรายการเฉลี่ย 116.443 ธุรกรรมต่อวินาที และการทดสอบหน่วยย่อย (unit test) โดยมีการทดสอบจำนวน 15 รายการ ซึ่งผลการทดสอบผ่าน ทั้ง 15 รายการ นอกจากนี้ ยังพบว่าระบบมีคุณสมบัติต่าง ๆ ได้แก่ ความสอดคล้องของข้อมูล consistency) ความสมบูรณ์ของข้อมูล (integrity) ความโปร่งใสในการเข้าถึงข้อมูล (data transparency) ความสามารถในการทำงานอย่างต่อเนื่องของระบบ (availability) และความคงทนของข้อมูล (resiliency) ที่สูง อย่างไรก็ตาม ระบบมีค่าใช้จ่ายในการพัฒนาสูงกว่าและมีความเป็นไปได้ในการติดตั้งระบบ (feasibility) ต่ำกว่าระบบเลือกตั้งที่จัดเก็บข้อมูลแบบรวมศูนย์

## 7. ข้อเสนอแนะ

ผู้วิจัยมีข้อเสนอแนะและแนวทางในการพัฒนา ดังนี้

- ในกรณีที่ต้องการจัดเก็บข้อมูลลับ หรือข้อมูลส่วนบุคคล ไว้นอกบล็อกเชน โดยจัดเก็บเฉพาะรหัสของระเบียบและแฮชของข้อมูลไว้ในบล็อกเชนเพื่อใช้อ้างอิงและยืนยันความถูกต้องของข้อมูลที่อยู่นอกบล็อกเชน
- ในกรณีที่ต้องการเพิ่มประสิทธิภาพให้เครือข่ายรองรับจำนวนธุรกรรมต่อวินาทีได้มากขึ้นในอนาคตที่อาจจะมีบล็อกเชนแพลตฟอร์มอื่นๆ ได้รับการปรับปรุงและพัฒนาประสิทธิภาพให้รองรับจำนวนธุรกรรมต่อวินาทีได้มากขึ้น โดยงานวิจัยนี้สามารถปรับไปใช้กับแพลตฟอร์มบล็อกเชน อื่นๆ ที่อาจจะมีขึ้นในอนาคตได้

## 8. เอกสารอ้างอิง


[1] Nick Szabo. (4 July 2022). *Smart Contracts*, [Online] Available : http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html/

[2] M. Chayakornvikrom, K. Wongprasert, S. Ketpong, C. Inthanok and S. Phatthanakhuha, *Blockchain for GOVERNMENT SERVICES*. BKK, Jan. 2021, pp.19-23.

[3] E. Febriyanto, Triyono ,N. Rahayu , K. Pangaribuan and P. A. Sunarya. “Using Blockchain Data Security Management for E-Voting Systems”. *International Conference on Cyber and IT Service Management. 23-24 Oct. Pangkal Indonesia* : pp. 1-4, 2020. doi: 10.1109/CITSM50537.2020.9268847.

[4] A. Al-madani, A. T. Gaikwad, V. Mahale and Z. A.T.Ahmed. “Decentralized E-voting system based on Smart Contract by using blockchain Technology”. *International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing. 30-31 Oct. Aurangabad India* : pp. 176-180, 2020. doi: 10.1109/ICSIDEMPC49020.2020.9299581.

[5] F. P. Hjálmarsson and G. K. Hreiðarsson, “Blockchain-Based E-Voting System” *International Conference on Cloud Computing. 2-7 Jul. San Francisco CA USA* : pp. 983-986, 2018. doi: 10.1109/CLOUD.2018.00151.

[6] M. Malkawi, M. Yassein and A. Bataineh. “Blockchain based voting system for Jordan parliament elections,” *International Journal of Electrical and Computer Engineerin*g., Vol. 11 (No. 5), pp. 4325-4335, 2021. doi: http://doi.org/10.11591/ijece.v11i5.pp4325-4335.

[7] เบญจมาศ ปัญญางาม, *เอกสารประกอบการสอน กระบวนวิชา 204251 โครงสร้างข้อมูล*. CMI, Thailand, 2018, pp.87-96.

[8] Geeksforgeeks. (26 April 2024). *Quick Sort vs Merge Sort*, [Online] Available : https://www.geeksforgeeks.org/quick-sort-vs-merge-sort/

[9] ปุญญพัฒน์ บวรเศรษฐพงศ์, กรกมล บุญส่งดี, ณัฐพล พัชรนันท์จินดา และ ภัทรทิชา ประเสริฐสังข์, *ASSIGMENT 2 QUICK SORT*. BKK, Thailand, 2015, pp.1-10.

[10] A. Fernandes, K. Garg, A. Agrawal and A. Bhatia, “Decentralized Online Voting using Blockchain and Secret Contracts”. *International Conference on Information Networking. 13-16 Jan. Jeju Island Korea* : pp. 582-587, 2021. doi: 10.1109/ICOIN50884.2021.9333966.

[11] S. T. Alvi, M. N. Uddin, L. Islam and S. Ahamed, “A blockchain based Cost effective Digital Voting System using SideChain and Smart Contracts”. *International Conference on Electrical and Computer Engineering. 17-19 Dec. Dhaka Bangladesh* : pp. 467-470, 2020. doi: 10.1109/ICECE51571.2020.9393081.