

STRUCTURE

Blockchain

- ที่มาและความสำคัญ
- หลักการพื้นฐาน + Protocol

Smart Contract

- ความหมายและความสำคัญ
- ภาษาที่ใช้ในการพัฒนา/tools
 - Solidity
 - Hardhat
 - Etherscan

Token

- อะไรคือมาตรฐาน ERC
- อะไรคือ Token และความสำคัญ
- ERC20(FT) กับ ERC721(NFT)

Cryptocurrency Wallet

- อะไรคือ Wallet
- ความสามารถพื้นฐาน
- หลักการทำงานพื้นฐาน
- Type of Wallets
- Hardware Wallet
 - Pros and Cons
 - Software ที่ใช้ในการทำงาน

Web application

- ความหมาย
- การพัฒนา web app ในปัจจุบัน + Library/Framework
- React JS
 - ลักษณะทั่วไปและจุดเด่น (ข้อดี)
 - Package ที่นิยมใช้เพิ่มเติมในการพัฒนา
 - React-Router

- Tailwindcss
- Redux/Redux-toolkit

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

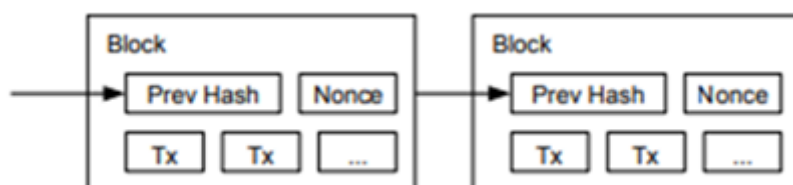
Blockchain

Blockchain เป็นเทคโนโลยีการจัดเก็บข้อมูลในลักษณะแบบกระจายศูนย์ ซึ่งอยู่ในระบบเครือข่ายคอมพิวเตอร์ โดยที่ตัวระบบมีลักษณะไม่มีศูนย์กลางในลักษณะแบบ Peer-to-peer ระบบเครือข่ายนั้นจะมีข้อกำหนดที่ทำให้ แต่ละจุดในเครือข่ายเห็นข้อมูลชุดเดียวกันทั้งหมด

แนวคิดของ Blockchain นั้นได้เริ่มกล่าวถึงในปี ค.ศ. 1991 โดย Stuart Haber และ W. Scott Stornetta โดยทั้งคู่ ได้เสนอแนวทางระบบสำหรับเอกสารที่มีการบันทึกเวลา (Timestamps) เพื่อไม่ให้เกิดการมาดัดแปลงแก้ไข จนกระทั่งปี ค.ศ. 2008 ได้มีเอกสาร Bitcoin ปรากฏตัวขึ้นและต่อมาเครือข่ายของมันก็กำเนิดขึ้นในเดือนมกราคม ปี ค.ศ. 2009 โดยบุคคลหรือกลุ่มคนผู้ใช้นามแฝงว่า Satoshi Nakamoto ทำให้แนวคิด Blockchain นั้นเป็นจริง Bitcoin คือสกุลเงินเหรียญดิจิทัลที่ไม่มีสถาบันการเงินเข้ามาควบคุม และใช้ระบบ Blockchain เป็นระบบในการทำธุรกรรม (Transaction) โดยไม่ต้องผ่านบุคคลที่สาม

หลักการทำงาน

ลักษณะการเก็บข้อมูลของ Blockchain นั้นจะเป็นการเก็บข้อมูลธุรกรรมสัญญาทั่วไป (Transaction) ลง Block หลายๆ Block และแต่ละ Block จะมีการเชื่อมโยงกันเป็นห่วงโซ่ (Chain) ยาวเป็นสายเดียว ภายใน Block นั้นจะมีค่า Hash Block ก่อนหน้าเพื่ออ้างอิงเป็นลูกโซ่และตรวจสอบความถูกต้อง



รูปที่ 2.1 การเชื่อมต่อระหว่าง block

หาก Block มีการเปลี่ยนแปลงข้อมูลจะทำให้ค่า Hash ของ Block นั้นเปลี่ยนตาม ส่งผลให้ Block ที่มีการเชื่อมต่อก่อนหน้านั้นไม่สามารถอ้างอิงถึง Block ที่มีการเปลี่ยนแปลงข้อมูลได้

และนำไปสู่กระบวนการตรวจสอบความถูกต้องในระบบเครือข่ายภายหลัง เนื่องจากการเรียง Block ไม่ตรงกับ จุดอื่นภายในเครือข่าย หรือหากมีความจำเป็นต้องแก้ไข Block นั้น Block ลำดับถัดไปที่มาต่อหลังจาก Block นี้ต้องแก้ไขค่า Hash ใหม่ตามทั้งหมด การแก้ไขข้อมูลจึงเป็นเรื่องยากและทำให้มีความปลอดภัยสูงหากมี Block ต่อหลังเป็นจำนวนมาก

ขั้นตอนการทำงาน

1. CREATE คือ การสร้าง Block ที่บรรจุคำสั่งขอทำรายการธุรกรรม
2. BROADCAST คือ ทำการกระจาย Block ใหม่นี้ให้กับทุก Node ในระบบ และบันทึกรายการธุรกรรมลง Ledger ให้กับทุก Node เพื่ออัปเดตว่ามี Block ใหม่เกิดขึ้นมา
3. VALIDATION คือ Node อื่น ๆ ในระบบทำการยืนยันและตรวจสอบข้อมูลของ Block นั้นว่าถูกต้องตามเงื่อนไข Validation โดยกระบวนการทำ Consensus ถือว่าเป็นส่วนหนึ่งของกระบวนการทำ Validation

องค์ประกอบ

Block คือ ชุดบรรจุข้อมูลซึ่งมี 2 ส่วนคือส่วนของสิ่งของต่าง ๆ ที่ใส่เข้าไปเรียกว่า Item และส่วนแพะหัวกล่องหรือ Header เพื่อใช้บอกให้คนอื่นทราบว่าบรรจุอะไรมา (แต่เปิดดู Item ภายในนั้นไม่ได้)

Chain คือ หลักการจดจำทุก ๆ ธุรกรรมของทุก ๆ คนในระบบและบันทึกข้อมูลพร้อมจัดทำเป็นสำเนาบัญชี Ledger แจกจ่ายให้กับทุกคนในระบบ

Consensus คือ ข้อตกลงร่วมกัน

Validation คือ การตรวจสอบความถูกต้องแบบทบทวนทั้งระบบและทุก Node

Smart Contract

Smart Contract หมายถึง กระบวนการทางดิจิทัล

ที่กำหนดขั้นตอนการทำธุรกรรมโดยอัตโนมัติไว้ล่วงหน้า โดยไม่ต้องอาศัยตัวกลาง อย่างเช่น ธนาคาร

ซึ่งการสร้าง Smart Contract ที่เป็นระบบอัตโนมัติอย่างเต็มรูปแบบ

โดยคู่สัญญาทั้งสองฝ่ายจะมีการตกลงกันก่อนหน้านี้ ถึงขั้นตอน กลไก ในการทำการธุรกรรมดังกล่าว

ซึ่งการพัฒนานี้ส่งผลกระทบต่อรูปแบบธุรกิจแบบดั้งเดิมของธนาคาร

Smart Contract เกิดมาจาก Nick Szabo ที่เป็นผู้เสนอไอเดียว่า Blockchain

สามารถใช้ในการบันทึกข้อตกลงของสัญญาที่สามารถดำเนินการได้ด้วยตัวเอง ไม่จำเป็นต้องมีคนกลาง

หรือใช้พนักงานในการมาตรวจสอบเอกสาร โดยทุกอย่างให้คอมพิวเตอร์และโปรแกรมจัดการ และการ

Hack ข้อมูลทำได้ยาก

ในที่นี้จะยกตัวอย่างการซื้อขายรถยนต์ โดยการทำสัญญาทั่วไปจะเป็นการทำสัญญาระหว่าง 2 ฝ่าย จากตัวอย่าง (ในรูปภาพที่ 3) หากอลิซต้องการซื้อรถยนต์จากบ๊อบ

ซึ่งต้องมีบุคคลที่สามที่เชื่อถือได้จะต้องยืนยันและรับรองความถูกต้องของข้อตกลงซึ่งอย่างน้อยหนึ่งคน

แต่โดยปกติแล้วบุคคลที่สามต้องเป็นผู้ที่เชื่อถือได้ เช่น หน่วยงานด้านทะเบียนยานยนต์ร่วมกับทนายความและ / หรือ บริษัท ประกันภัย

ซึ่งเป็นกระบวนการที่มีความซับซ้อนและใช้เวลานานและต้องเสียค่าธรรมเนียมเป็นจำนวนมากสำหรับคนกลางเหล่านี้

เครื่องมือที่ใช้พัฒนา

Solidity

เป็นภาษาโปรแกรมเชิงวัตถุที่มุ่งเน้นในการเขียนด้าน Smart Contract มีไวยากรณ์คล้ายกับ JavaScript, C ++ หรือ Java สามารถศึกษาข้อมูลเพิ่มเติมเกี่ยวกับ Solidity ที่นี่

Etherscan

Etherscan เป็น Blockchain Explorer และแพลตฟอร์มการวิเคราะห์ที่ให้รายละเอียดเกี่ยวกับธุรกรรมบล็อกเชน Ethereum ที่กำลังรอดำเนินการ หรือได้รับการยืนยัน

Etherscan เป็นเครื่องมือที่น่าเชื่อถือที่สุดสำหรับข้อมูลสาธารณะทั้งหมดบนบล็อกเชน Ethereum และบางครั้งเรียกว่า “Ethplorer” ข้อมูลนี้รวมถึงข้อมูลธุรกรรม ที่อยู่กระเป๋าเงิน Smart Contract และอื่นๆ อีกมากมาย

แอปพลิเคชันนี้เป็นอิสระ และไม่ได้รับการสนับสนุนหรือบริหารจัดการโดยมูลนิธิ Ethereum ซึ่งเป็นองค์กรไม่แสวงหาผลกำไร

ทีมงานที่อยู่เบื้องหลัง Etherscan

ประกอบด้วยนักพัฒนาที่มีประสบการณ์และผู้เชี่ยวชาญในอุตสาหกรรม ซึ่งพัฒนาแอป Etherscan เพื่อให้ผู้ใช้เข้าถึงบล็อกเชน Ethereum ได้มากขึ้น

แม้ว่า Etherscan จะเป็นแพลตฟอร์มแบบรวมศูนย์ แต่แอปนี้ช่วยให้ผู้คนค้นหาผ่านบล็อกเชน Ethereum ได้ง่ายขึ้น

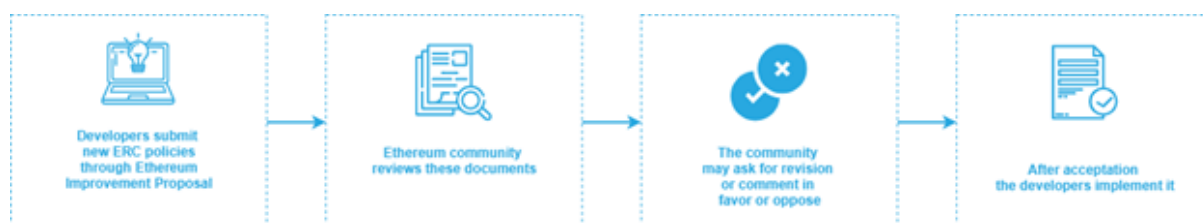
Tokens

คำว่า “Coin” และ “Token” มักถูกใช้ปะปนกัน จนอาจทำให้เกิดความเข้าใจผิดว่า ทั้งสองคำเป็นคำที่มีความหมายเดียวกัน แต่แท้จริงแล้ว ทั้ง Coin และ Token มีความหมายที่ต่างกันตามนิยามที่กำหนดไว้ในเรื่อง Blockchain ที่ขึ้นอยู่กับการทำงานของเหรียญนั้น ๆ และลักษณะการใช้งานของเหรียญ ซึ่งมีรายละเอียดดังต่อไปนี้

Token คือ เหรียญคริปโต ที่ไม่ได้มี Blockchain เป็นของตัวเอง แต่จะเป็นการสร้างขึ้นบน Blockchain ของแพลตฟอร์มอื่น มักสร้างขึ้นเพื่อวัตถุประสงค์ที่เฉพาะเจาะจงมากขึ้น อาจสร้างขึ้นมาเพื่อใช้ในระบบนิเวศหรือ Project บางอย่าง

Ethereum Request for Comment

Ethereum Request for Comment หรือที่เรียกว่า “ERC” เป็นขอความคิดเห็นจาก Ethereum โดย ERC ให้คำแนะนำทางเทคนิคและมาตรฐานในการพัฒนา Smart Contract บนเครือข่ายของ Ethereum



รูปที่ 2.2 ขั้นตอนการบัญญัติ ERC

ERC20

เป็นมาตรฐานในการพัฒนาที่นิยมใช้ใน Token รูปแบบ Fungible Token และอนุญาตให้ใช้รูปแบบ Application Programming Interface (API) ภายใน Smart Contract (ส่วนใหญ่ใช้ใน ICOs)

โดยที่ Fungible Token (FT) เป็นทรัพย์สินที่มีคุณสมบัติความสามารถทดแทนกันได้ ยกตัวอย่างเช่น ธนบัตรชนิดราคา 20 บาทที่ได้รับจากแม่ค้ากับธนบัตรชนิดราคา 20 บาทที่ได้รับจากธนาคาร สามารถใช้ทดแทนกันได้เนื่องจากคุณสมบัติความสามารถทดแทนกัน

ERC721

เป็นมาตรฐานในการพัฒนาที่นิยมใช้ใน Token รูปแบบ Non-Fungible Token และอนุญาตให้ใช้รูปแบบ Application Programming Interface (API) ภายใน Smart Contract นอกจากนี้ยังมีฟังก์ชันการถ่ายโอนและติดตาม Non-Fungible Tokens

โดยที่ Non-Fungible Token (NFT)

เป็นทรัพย์สินที่มีคุณสมบัติแตกต่างกันและไม่สามารถทดแทนกันได้ ยกตัวอย่างเช่น ที่ดินบ้านของครอบครัวกับที่ดินบ้านของเพื่อนไม่สามารถทดแทนกันได้ หรือ แมวของครอบครัวกับแมวของเพื่อนไม่สามารถทดแทนกันได้

Crypto Wallet

หลายคนมีโอกาสนักสะสมและใช้งาน Cryptocurrency ในโอกาสและวัตถุประสงค์ต่างๆ จนเข้าใจถึงกระบวนการใช้งานกันแล้ว ซึ่งในการใช้งานบางลักษณะ มีการขอให้ผู้ใช้กรอก Private Key เพื่อเข้าถึงเหรียญหรือยืนยันการทำธุรกรรม ซึ่ง Private Key จะเป็นชุดรหัสที่ผู้ใช้จะต้องเก็บรักษาไว้ หากสูญหายจะไม่สามารถเข้าถึงเหรียญที่เราได้ อีก หรือหากหลุดรั่วออกไปยังมิจฉาชีพก็อาจจะนำไปสู่การสูญเสียทรัพย์สินในบัญชีทั้งหมดได้

เพื่อป้องกันเหตุการณ์ดังกล่าว จึงมีการสร้าง “กระเป๋าสตางค์คริปโต” หรือ Crypto Wallet ขึ้น แม้จะใช้ชื่อเรียกว่ากระเป๋าสตางค์หรือ Wallet เหมือนกัน แต่ Crypto Wallet นั้นจะทำหน้าที่เก็บรักษา Private Key ของเรา ซึ่งสำหรับการใช้งานแล้ว มีความสำคัญไม่แพ้ตัวเหรียญด้วยเหตุผลที่อธิบายไปก่อนหน้านี้

ประเภทของ Wallet

Crypto Wallet เป็นกระเป๋าสตางค์ที่มีหลายลักษณะ ตั้งแต่เรียบง่ายสุดๆ ไปจนถึงมีนวัตกรรมเสริมมากมาย ซึ่งปัจจุบันในกลุ่มผู้ใช้งาน จะแบ่ง Crypto Wallet เป็น 2 ลักษณะ ได้แก่

1. Hot Wallet

หมายถึงกระเป๋าสตางค์ที่สร้างขึ้นบนระบบดิจิทัลเต็มรูปแบบและมีการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตตลอดเวลา ผู้ใช้จะเข้าถึงและใช้งานผ่าน Software ทั้งหมดเท่านั้น ตัวอย่างของ Hot Wallet ได้แก่ บัญชีที่เปิดใน Cryptocurrency Exchange Platform, Crypto Mobile Wallet Application ไปจนถึง Desktop Application ต่างๆ

จุดเด่นของ Hot Wallet คือการใช้งานง่ายเหมือน Financial Application ทั่วไป ตัวกระเป๋าสตางค์เชื่อมต่อกับเครือข่ายตลอดเวลา ผู้ใช้จึงสามารถแสดง Private Key เพื่อทำการแลกเปลี่ยนและโอนสินทรัพย์ได้สะดวก และด้วยการเชื่อมต่อตลอดเวลา จุดด้อยของ Hot Wallet จึงเป็นเรื่องการรักษาความปลอดภัยที่จะด้อยกว่ากระเป๋าสตางค์อีกประเภทหนึ่งนั่นเอง

2. Cold Wallet หมายถึงกระเป๋าสตางค์ที่สร้างในรูปแบบของ Hardware และไม่มีการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต

เมื่อผู้ต้องการทำธุรกรรมจะต้องดึงข้อมูลออกจาก Cold Wallet
ใส่เข้าไปในระบบเพื่อยืนยันการทำธุรกรรม

Cold Wallet ที่ได้รับความนิยมมี 2 รูปแบบ ได้แก่ Paper Wallet คือการบันทึก Wallet Address และ Private Key ในกระดาษ ทั้งในรูปของการจดหรือ Pattern ที่ทำการ Scan ผ่าน Software ได้ กับอีกรูปแบบหนึ่งคือ Hardware Wallet ซึ่งเป็นอุปกรณ์เก็บรหัส มักออกแบบด้วยรูปแบบบัตรหรือ Thumb Drive

จุดแข็งของ Cold Wallet คือด้านความปลอดภัย เนื่องจากการจะเข้าถึง Private Key นั้นต้องเข้าถึงตัว Wallet โดยตรง เรียกว่าลดโอกาสการโดนโจมตีทางดิจิทัลได้อย่างมาก ขณะเดียวกัน จุดด้อยของ Cold Wallet คือหากสูญหาย ก็ยากที่จะเข้าถึงการใช้งานใน Private Key ได้

จากคุณสมบัติข้างต้นของทั้ง 2 ประเภท สามารถระบุได้ว่า Hot Wallet เหมาะกับผู้ใช้งาน Cryptocurrency ทั่วไปหรือผู้ที่ต้องใช้งานรายวัน เก็บจำนวนเหรียญไว้ไม่สูง เน้นความคล่องตัวในการใช้งาน ขณะที่ Cold Wallet เหมาะกับผู้ที่ต้องการเก็บ Cryptocurrency จำนวนหรือมีมูลค่ามาก เน้นการเก็บรักษามากกว่าโอนถ่ายใช้งานเหรียญ

Hardware Wallet

เป็นอุปกรณ์ที่ทำหน้าที่เป็น crypto wallet ประเภทหนึ่งซึ่งมีหน้าที่ในการเก็บ private key ของผู้ใช้ไว้เพื่อใช้ในการ sign transaction ที่ต้องการใช้งานได้ ซึ่งสามารถให้ความปลอดภัยกับ key ได้มากกว่าการใช้ software wallet ทั่วไปซึ่งเก็บไว้บน exchange ต่าง ๆ ข้อดีของ hardware wallet คือ

1. พกพาได้สะดวก
2. มีความปลอดภัยสูง
3. เก็บ cryptocurrency ได้หลากหลาย
4. เหมาะกับการเก็บ cryptocurrency เพื่อเทรดตามราคา

ตัวอย่าง hardware wallet ที่วางขายในท้องตลาดเช่น Ledger, Trezor, SafePal เป็นต้น อุปกรณ์เหล่านี้มีความปลอดภัยสูงและมี function ต่าง ๆ ให้เลือกใช้ได้ตามที่ผู้ต้องการ หากต้องการลองใช้งานเป็นลักษณะ DIY สามารถทดลองใช้เป็น Pitrezor ซึ่งเป็น hardware wallet ที่สามารถใช้ได้บน Raspberry Pi และยังเป็น open-source อีกด้วย

Web Application

web application หรือ web app เป็นซอฟต์แวร์ชนิดหนึ่งที่รันบน web server ผู้ที่ต้องการใช้งานซอฟต์แวร์นี้สามารถเข้าใช้ได้ผ่าน web browser ที่มีการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต แอปพลิเคชันประเภทนี้ส่วนใหญ่จะทำงานโดยอาศัยโครงสร้างแบบ client-server โดยฝั่งผู้ใช้งาน (“client”) จะใช้บริการต่าง ๆ ผ่านทาง server

การพัฒนา Web Application ในปัจจุบัน

web application ส่วนใหญ่สามารถแบ่งส่วนการทำงานได้เป็น 2 ส่วนคือ front-end ซึ่งมีการพัฒนาโดยใช้งาน HTML, CSS, JavaScript และอีกส่วนคือ back-end ที่จะพัฒนาด้วยภาษา server-side เช่น Ruby, Python, JavaScript, Java เป็นต้น

การพัฒนาแอปพลิเคชันในฝั่ง front-end จะเป็นส่วนของการแสดงผล user interface และการพัฒนาในฝั่ง back-end จะเกี่ยวข้องกับการติดต่อกับ database, การสร้าง APIs, และการทำกิจกรรมเบื้องหลังของแอปพลิเคชัน

ในปัจจุบันการพัฒนาแอปพลิเคชันทั้ง 2 ฝั่งมีการนำ framework และ library มาใช้งานเพื่อให้ผู้พัฒนาสามารถทำงานได้ง่ายขึ้น ตัวอย่าง front-end framework/library เช่น ReactJS, Angular, VueJS เป็นต้น ตัวอย่าง back-end framework/library เช่น NodeJS, ASP.NET Core, Spring เป็นต้น นอกจากนี้ยังมี framework และ library ที่สามารถพัฒนาได้ทั้ง 2 ฝั่งเช่นกันเช่น ASP.NET Core, Django, Flask เป็นต้น

ข้อดีของการพัฒนาแอปพลิเคชันด้วย framework/library คือ

1. ซอฟต์แวร์มีเสถียรภาพ
2. พัฒนาได้เร็วยิ่งขึ้น
3. รูปแบบโค้ดมีความเป็นมาตรฐานและสม่ำเสมอ
4. ประสบการณ์การใช้งานที่ดีขึ้น

ข้อดีของการพัฒนาแอปพลิเคชันด้วย framework/library คือ

1. ซอฟต์แวร์ที่เรียบง่ายมีความซับซ้อนมากขึ้นโดยไม่จำเป็น
2. การดัดแปลงซอฟต์แวร์เดิมเข้ากับ framework/library ทำได้ยาก
3. ทางเลือกแนวทางพัฒนาถูกจำกัด
4. การ update อาจทำให้เกิดบัค
5. ต้องใช้เวลาในการศึกษาใช้งาน

React JS

React (React.JS หรือ ReactJS) เป็น front-end JavaScript library ที่ดูแลและพัฒนาโดย Meta ใช้ในการออกแบบ UI โดยใช้ component เป็นพื้นฐาน ซึ่งนักพัฒนาสามารถนำไปใช้งานได้ฟรี นอกจากนี้ยังเป็น open-source อีกด้วย

จุดเด่นของ React คือ

1. Declarative คือใช้แนวทางการเขียนโปรแกรมโดยเน้นที่ว่า “ทำอะไร” มากกว่า “ทำอย่างไร” ทำให้โค้ดสามารถคาดเดาได้และดีบั๊กได้ง่ายขึ้น
2. Component-Based โดยแต่ละ component จะจัดการ state ของตัวเอง และสามารถนำแต่ละ component มาประกอบเข้าด้วยกันได้ นอกจากนี้ component ต่าง ๆ ยังถูกสร้างขึ้นด้วย JavaScript ไม่ใช่ template จึงสามารถส่งผ่านข้อมูล JavaScript ไปได้ตรง ๆ โดยไม่ผ่าน DOM
3. Learn Once, Write Anywhere โดย React ไม่จำกัดเทคโนโลยีใด ๆ ที่ต้องนำมาใช้งานร่วมกัน นอกจากนี้ยังสามารถนำ React ไป render บน server โดยใช้ Node ก็ได้ และหากต้องการทำ mobile app ก็สามารถใช้ React Native พัฒนาได้เช่นกัน

การพัฒนา front-end ด้วย React ซึ่งในปัจจุบันมีความซับซ้อนมากขึ้น จึงมีการพัฒนาเป็น package ขึ้นมาเพื่อให้พัฒนาได้ง่ายขึ้น สำหรับ package ที่นิยมใช้งานกันเช่น

1. React Router เป็น package สำหรับทำการ routing ทั้งฝั่ง client-side และ server-side โดยสามารถใช้งานได้กับ React ทุกรูปแบบ
2. Redux เป็น state management package สำหรับแอปพลิเคชัน JavaScript โดยช่วยทำให้แอปพลิเคชันสามารถรันได้สม่ำเสมออยู่ในสภาพแวดล้อมที่แตกต่างกัน, ทำให้ทดสอบได้ง่ายขึ้น, และทำให้ผู้พัฒนาสามารถพัฒนาได้ดีขึ้น (ในปัจจุบันแนะนำให้ใช้งาน Redux Toolkit ซึ่งเป็น library เสริมที่พัฒนาโดยใช้ Redux เป็นหลัก) สำหรับการใช้งานกับ React แนะนำให้ใช้ package React Redux ควบคู่กันด้วย
3. tailwindcss เป็น utility CSS framework ที่ช่วยให้สามารถตกแต่ง UI ได้รวดเร็วยิ่งขึ้นโดยอาศัยการเรียกใช้งาน utility class ต่าง ๆ ที่ตัว package มีมาให้ นอกจากนี้ยังสามารถทำการปรับแต่งหรือเพิ่มเติม class ต่าง ๆ ได้ตามความต้องการอีกด้วย

Source

Blockchain

<https://www.simplilearn.com/tutorials/blockchain-tutorial/why-is-blockchain-important#:~:text=Blockchain%20helps%20in%20the%20verification,the%20origin%20of%20a%20product.>

<https://www.tfac.or.th/Article/Detail/135089#:~:text=%E0%B8%AB%E0%B8%A5%E0%B8%B1%E0%B8%81%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%97%E0%B8%B3%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B8%82%E0%B8%AD%E0%B8%87%E0%B9%80%E0%B8%97%E0%B8%84%E0%B9%82%E0%B8%99%E0%B9%82%E0%B8%A5%E0%B8%A2%E0%B8%B5%20Blockchain,%E0%B9%80%E0%B8%94%E0%B8%95%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B8%82%E0%B9%89%E0%B8%AD%E0%B8%A1%E0%B8%B9%E0%B8%A5%E0%B9%81%E0%B8%9A%E0%B8%9A>

Smart Contracts

<https://www.etda.or.th/th/Useful-Resource/knowledge-sharing/articles/index-article-other-p1/SMART-CONTRACT-%E0%B8%84%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3.aspx?feed=590fb9ad-c550-4bc5-9a56-459ad4891d74#:~:text=Smart%20Contract%20%E0%B8%AB%E0%B8%A1%E0%B8%B2%E0%B8%A2%E0%B8%96%E0%B8%B6%E0%B8%87%20%E0%B8%81%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%A7%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3,%E0%B8%99%E0%B8%B5%E0%B9%89%E0%B8%AA%E0%B9%88%E0%B8%87%E0%B8%9C%E0%B8%A5%E0%B8%81%E0%B8%A3%E0%B8%B0%E0%B8%97%E0%B8%9A%E0%B8%95%E0%B9%88%E0%B8%AD>

Tools ที่พัฒนา

<https://articles.devsight.me/smart-contract-%E0%B8%9A%E0%B8%99-ethereum-public-blockchain-permissionless-blockchain-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-solidity-path-3-95eb4a3c9cae>

<https://cryptosiam.com/what-is-etherscan/#:~:text=Etherscan%20%E0%B9%80%E0%B8%9B%E0%B9%87%E0%B8%99%E0%B8%AB%E0%B8%99%E0%B8%B6%E0%B9%88%E0%B8%87%E0%B9%83%E0%B8%99%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87%E0%B8%A1%E0%B8%B7%E0%B8%AD,%E0%B9%84%E0%B8%A1%E0%B9%88%E0%B8%88%E0%B8%B3%E0%B9%80%E0%B8%9B%E0%B9%87%E0%B8%99%E0%B8%95%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%A5%E0%B8%87%E0%B8%97%E0%B8%B0%E0%B9%80%E0%B8%9A%E0%B8%B5%E0%B8%A2%E0%B8%99>

Tokens

<https://zipmex.com/th/learn/difference-between-coin-token/>

ERC

<https://articles.devsight.me/smart-contract-%E0%B8%9A%E0%B8%99-ethereum-public-blockchain-permissionless-blockchain-%E0%B8%94%E0%B9%89%E0%B8%A7%E0%B8%A2-solidity-path-3-95eb4a3c9cae>

ประเภทของ wallet

<https://www.scb10x.com/blog/getto-know-cryptowallet>

Hardware wallet

<https://medium.com/bitkub/hardware-wallet-acf1868a9558>

<https://www.pitrezor.com/2018/02/pitrezor-homemade-trezor-bitcoin-wallet.html>

Web Application

https://en.wikipedia.org/wiki/Web_application

การพัฒนา web app ในปัจจุบัน

<https://trio.dev/blog/web-app-development>

<https://bluemodus.com/articles/pros-and-cons-of-using-a-front-end-library-or-framework>

React

[https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

<https://reactjs.org/>

<https://reactrouter.com/docs/en/v6/getting-started/tutorial>

<https://redux.js.org/introduction/getting-started>

<https://tailwindcss.com/>