

# การทดสอบประสิทธิภาพ FRONTEND FRAMWORK ในการ จัดการ DOM

*Presented by: Kittinun Roonsungneon*  
*65130695*



# ปัญหาและที่มา

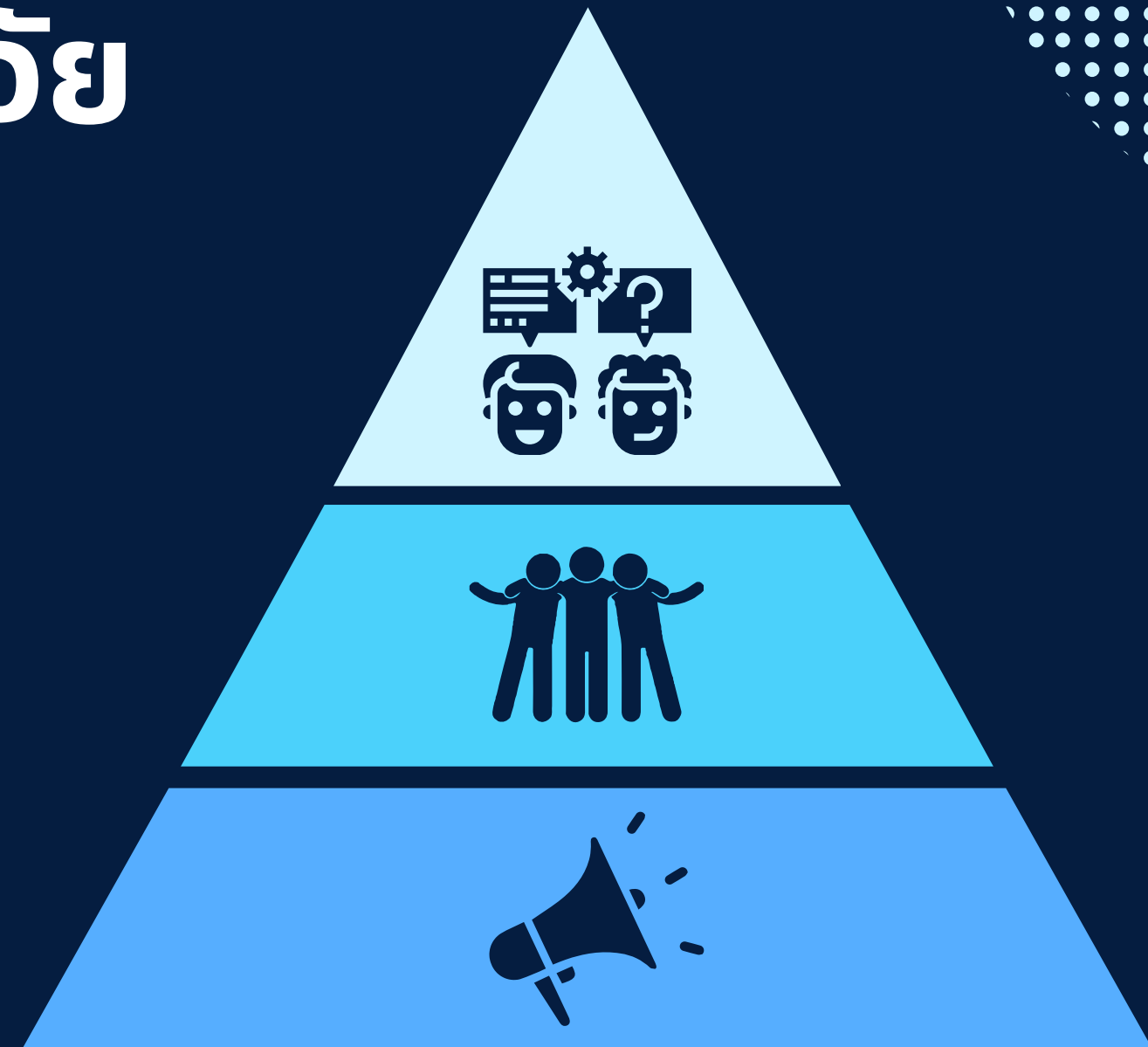
การพัฒนาแอปพลิเคชันเว็บในปัจจุบันมีการใช้เทคโนโลยีและเครื่องมือที่ทันสมัย เพื่อให้ผู้ใช้สามารถเข้าถึงและโต้ตอบกับเว็บไซต์ได้อย่างมีประสิทธิภาพและรวดเร็ว อย่างไรก็ตาม การพัฒนาและทดสอบประสิทธิภาพของ frontend พบว่าการจัดการกับ DOM มีความซับซ้อนและมีขนาดใหญ่ขึ้น ซึ่งอาจส่งผลต่อประสิทธิภาพของเว็บไซต์และประสิทธิภาพในการโหลดและโต้ตอบของผู้ใช้

ในการทดลองทำความเข้าใจกับปัญหาที่เกิดขึ้นในการจัดการ DOM และผลกระทบที่เกิดขึ้นต่อประสิทธิภาพของแอปพลิเคชันเว็บ การทำวิจัยนี้จะสร้างพื้นฐานในการพัฒนาเทคนิคและเครื่องมือที่ช่วยในการแก้ไขปัญหานี้ ซึ่งอาจสร้างผลการวิจัยที่สำคัญและมีประโยชน์ในการพัฒนาเว็บไซต์ให้มีประสิทธิภาพมากยิ่งขึ้นในอนาคต

---

# วัตถุประสงค์ของงานวิจัย

- 01 เปรียบเทียบประสิทธิภาพในการจัดการ DOM ของ Frontend Frameworks ต่างๆ เพื่อให้ผู้พัฒนารับถึง  
— ความแตกต่างระหว่าง Frameworks
- 02 ทราบปัจจัยที่ส่งผลต่อประสิทธิภาพของการจัดการ DOM  
— และการทำงานของแอปพลิเคชันเว็บ
- 03 ให้ข้อเสนอแนะเกี่ยวกับการเลือกใช้ Frontend Frameworks ในโปรเจกต์ต่างๆ



# ขอบเขตของงานวิจัย

## 1

ทดสอบประสิทธิภาพในการจัดการ DOM: การวัดความเร็วในการสร้าง, อัปเดต, และลบองค์ประกอบ DOM เพื่อทราบถึงประสิทธิภาพของ Frontend Frameworks ในด้านการจัดการ DOM

---

# งานวิจัยที่เกี่ยวข้อง

- บทความวิจัยชื่อ “A Comparative Analysis of Modern Frontend Frameworks สำหรับการสร้างแอปพลิเคชันเว็บขนาดใหญ่” มีจุดมุ่งหมายเพื่อวิเคราะห์ประสิทธิภาพของกรอบเวิร์กและไลบรารีที่ใช้ JavaScript ยอดนิยมสำหรับการพัฒนาเว็บ [1]
- บทความนี้เปรียบเทียบประสิทธิภาพของเฟรมเวิร์กที่แตกต่างกันตามเมตริกประสิทธิภาพของเว็บ เช่น ความพึงพอใจของผู้ใช้ เวลาถึงไบต์แรก เวลาในการวาดครั้งแรก และดัชนีความเร็ว [2]
- React และ Vue.js ถูกกล่าวถึงว่าเป็นเฟรมเวิร์ก JavaScript ยอดนิยมที่จะวิเคราะห์ในการวิจัย
- 
- 3.วิธีการทดลอง
- บทความนี้เปรียบเทียบเฟรมเวิร์กเว็บตามเมตริกประสิทธิภาพของเว็บ เช่น เวลาในการโหลดเนื้อหา DOM การมีส่วนร่วมของเรด เหตุการณ์การคลิก การสัมผัสหน้าจอ และเวลาตั้งแต่ผู้ใช้มีส่วนร่วมครั้งแรกกับเว็บไซต์ [2]
- 4.สรุปผลลัพธ์
- React และ Vue.js โหลดและนำเสนอเนื้อหาได้เร็วกว่า Angular โดยที่ React มีไซต์มากที่สุดที่มีการเปลี่ยนเลย์เอาต์สะสม 0.1อย่างใดก็ตาม กรอบเวิร์กทั้งหมดแสดงผลลัพธ์เชิงบวกในแง่ของเวลาตั้งแต่ผู้ใช้มีส่วนร่วมครั้งแรกกับเว็บไซต์ [2]

2023 International Conference on Disruptive Technologies (ICDT)

## A Comparative Analysis of Modern Frontend Frameworks for Building Large-Scale Web Applications

Pancham Singh  
Information Technology  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
singhpancham@akgec.ac.in

Mili Srivastava  
Information Technology  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
srivastavamili@akgec.ac.in

Mrignainy Kansal  
Information Technology  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
kansalmrignainy@akgec.ac.in

Aditya Pratap Singh  
Computer Science & Information  
Technology  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
pratapsinghaditya2001@gmail.com

Abhay Chauhan  
Computer Science & Information  
Technology  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
abhaychauhan232@gmail.com

Adarsh Gaur  
Computer Science & Information  
Technology  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
adarshgaur3@gmail.com

*Abstract*—Some tools have endured in the constantly shifting world of technology, whereas many others have slipped into oblivion over time. One thing we can all agree on is that JavaScript-based libraries and frameworks are here to stay. This incredible technology is adored by countless clients and developers because of its adaptability, it can be installed on both the server and client sides of the web applications. However, due to its unique syntax and quirks, JavaScript may be a difficult language to master. We can admit that we've often caught ourselves staring at a JavaScript code, wondering what was transpiring behind the scenes. This paper will analyze the performance of such popular JavaScript-based frameworks and libraries, and explain why they remain among the most significant JavaScript frameworks of the previous decade. This paper will compare the web frameworks to find a possible

hand, Vue.js is a newer framework which caught the limelight due to its fast performance. Vue is not backed by any tech giant yet and is maintained by an Open Source community led by Evan You, founder of Vue.js.

We want to emphasize that determining which one is quicker isn't the sole way to solve our performance issues. In reality, it is crucial to maintain a high level of development quality and to adhere to best practices [20].

All three frameworks have similar characteristics, including a component-based design and the ability to easily create UI elements. React and Vue.js are mostly declarative, and while Angular might be, it's more imperative. Nonetheless, they differ in terms of structure, architecture, and



# งานวิจัยที่เกี่ยวข้อง(ต่อ)

- หัวข้อ: การเปรียบเทียบและการวิเคราะห์กรอบหน้าและห้องสมุดยอดนิยม
- \*\*1.แนวคิด: \*\* การศึกษามุ่งเน้นไปที่การประเมินพารามิเตอร์สำหรับการพัฒนาเว็บ Frontend โดยเฉพาะการเปรียบเทียบและวิเคราะห์กรอบและไลบรารีด้านหน้าที่ใช้กันอย่างแพร่หลายมันเน้นถึงความสำคัญของการพัฒนาเว็บด้านหน้าในการสร้างประสบการณ์ออนไลน์ที่ยอดเยียมสำหรับผู้ใช้
- \*\*2.ทฤษฎีประยุกต์: \*\* การวิจัยตรวจสอบและเปรียบเทียบเฟรมเวิร์กด้านหน้าตามเกณฑ์ เช่น ความสามารถในการใช้งาน ประสิทธิภาพ ความสามารถในการปรับขนาด การมีส่วนร่วมของผู้ใช้ และความน่าเชื่อถือด้วยการตัดกันและวิเคราะห์ตัวเลือกที่มีอยู่ นักพัฒนาสามารถตัดสินใจอย่างมีข้อมูลเกี่ยวกับกรอบการทำงานที่ดีที่สุดสำหรับโครงการพัฒนาเว็บของพวกเขา [1]
- \*\*3.วิธีการทดลอง: \*\* แหล่งที่มาไม่ได้กล่าวถึงวิธีการทดลองเฉพาะที่ใช้ในการศึกษา
- \*\*4.สรุปผลลัพธ์: \*\* แหล่งที่มาไม่ได้ให้ผลลัพธ์ที่เฉพาะเจาะจงของการเปรียบเทียบและการวิเคราะห์กรอบด้านหน้าอย่างไรก็ตาม พวกเขาเน้นความแตกต่างที่สำคัญระหว่าง ReactJS และ AngularJS โดยกล่าวว่า ReactJS เป็นที่รู้จักกันดีในเรื่องโมดูลาร์ความเร็ว ความเรียบง่าย และความเป็นมิตรกับผู้ใช้ในขณะที่ AngularJS มีเส้นโค้งการเรียนรู้ที่ต่ำกว่า ฟังก์ชันการทำงานมากขึ้น และแนวทางที่ประกาศแหล่งที่มาซึ่งเน้นถึงความสำคัญของการพิจารณาข้อกำหนดของโครงการและดำเนินการศึกษาและวิเคราะห์อย่างละเอียดก่อนตัดสินใจเกี่ยวกับกรอบหรือห้องสมุดที่จะใช้ [2]

Proceedings of the Fourth International Conference on Electronics and Sustainable Communication Systems (ICESC-2023)  
IEEE Xplore Part Number: CFP23V66-ART; ISBN: 979-8-3503-0009-3

## Comparison and Analysis of Popular Frontend Frameworks and Libraries: An Evaluation of Parameters for Frontend Web Development

Gursheen Kaur  
Chitkara University Institute of Engineering and Technology,  
Chitkara University,  
Punjab, India  
gursheen1216.cse19@chitkara.edu.in

Raj Gaurang Tiwari  
Chitkara University Institute of Engineering and Technology,  
Chitkara University,  
Punjab, India  
rajgaurang@chitkara.edu.in

**Abstract**—Creating a fantastic online experience for users relies heavily on strong and innovative front-end web development. A quick introduction to the fundamentals and significance of frontend web development is provided in this study. This research examines and compares widely used front-end frameworks and libraries based on a number of criteria. Inclusive important criteria are usability, performance, scalability, user engagement, and reliability. By contrasting and analysing the available frontend frameworks, developers are better able to weigh the pros and cons of each option and choose the best one for their web development project environment.

**Keywords**— Frontend Web Development, Popular Frontend Frameworks and Libraries, Parameters, Evaluation, Comparison, Analysis.

### I. INTRODUCTION

A frontend framework is a set of tools and technologies that streamline the process of creating a website's user interface. These frameworks give developers with standardised building blocks and patterns that enable them to

Developers may choose the finest tools for the job by evaluating available frameworks and libraries. Furthermore, investigating front-end solutions aids in avoiding the widespread one-size-fits-all strategy. Using a single framework or library might slow down development since it ignores the fact that distinct web applications have varied needs. The variety that comes through research and analysis is essential for resolving difficult issues that arise during web development. Therefore, it helps foster originality and invention throughout product creation, which is essential for maintaining a competitive edge. The quality, speed, and efficiency of web application development are all improved by thorough investigation and analysis of frontend frameworks and libraries. In conclusion, developers need to give careful consideration to the choice of a frontend framework or library. It defines how sophisticated the system is, how fast it runs, and how easy it is to programme. We found that React was the most widely used and well-suited alternative for developing large-scale online apps because of

# ทฤษฎีที่เกี่ยวข้อง

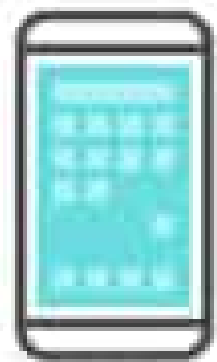
Frontend framework เป็นเครื่องมือหรือโครงสร้างที่ใช้ในการพัฒนาแอปพลิเคชันเว็บหรือเว็บไซต์ โดยมุ่งเน้นไปที่ส่วนของ user interface (UI) และ user experience (UX) ซึ่งสามารถช่วยให้การพัฒนาเว็บแอปพลิเคชันเป็นไปอย่างมีประสิทธิภาพและมีความสามารถที่ดีในการตอบสนองต่อผู้ใช้งาน

HTML (HYPERTEXT MARKUP LANGUAGE) เป็นภาษาที่ใช้ในการสร้างโครงสร้างของหน้าเว็บเพจ เพื่อกำหนดโครงสร้างของข้อมูลและเนื้อหาที่จะแสดงบนเว็บไซต์ เราใช้ HTML เพื่อกำหนดโครงสร้างของเอกสาร เช่น การกำหนดหัวข้อ, การจัดลำดับ, การระบุลิงค์, การแทรกภาพ และอื่นๆ

DOM (DOCUMENT OBJECT MODEL) คือโมเดลของเอกสารที่ใช้ในการแสดงโครงสร้างและสมาชิกของหน้าเว็บเพจหรือเว็บแอปพลิเคชัน โดย DOM จะแสดงโครงสร้างของหน้าเว็บในรูปแบบของอ็อบเจกต์ที่สามารถเข้าถึงและปรับเปลี่ยนได้ผ่าน JAVASCRIPT หรือโค้ดที่เกี่ยวข้อง ดังนั้น HTML และ DOM สัมพันธ์กันอย่างมีความสำคัญในการสร้างและแสดงเนื้อหาบนเว็บไซต์ โดย HTML จะใช้ในการกำหนดโครงสร้างของเอกสาร ในขณะที่ DOM จะใช้ในการแสดงโครงสร้างนั้นในรูปแบบของโมเดลที่ JAVASCRIPT สามารถเข้าถึงและจัดการได้



Web browser  
Desktop



Mobile app

# ทฤษฎีที่เกี่ยวข้อง

VIRTUAL DOM เป็นแนวคิดหนึ่งในการจัดการกับ DOM (DOCUMENT OBJECT MODEL) ในเว็บแอปพลิเคชันโดยมีจุดประสงค์เพื่อเพิ่มประสิทธิภาพในการอัปเดตและแสดงผลของ UI (USER INTERFACE) โดยส่วนใหญ่มักจะเป็นการใช้ใน FRONTEND FRAMEWORKS เช่น REACT.JS, VUE.JS, และ ANGULAR ซึ่งเป็นเครื่องมือที่ช่วยในการพัฒนาแอปพลิเคชันเว็บอย่างมีประสิทธิภาพมากขึ้น

หลักการทำงานของ VIRTUAL DOM คือการสร้างโครงสร้างของ DOM ในรูปแบบของโมเดลที่อยู่ระหว่างจริงกับที่อยู่บนหน้าเว็บ โดยการใช้โครงสร้างข้อมูลที่ถูกเรียกว่า VIRTUAL DOM จากนั้นเมื่อมีการเปลี่ยนแปลงในข้อมูลหรือสถานะของแอปพลิเคชัน โมเดล VIRTUAL DOM จะถูกอัปเดตตาม และทำการเปรียบเทียบกับ DOM จริงเพื่อหาความแตกต่าง และจากนั้นอัปเดต DOM จริงเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้น

การใช้ VIRTUAL DOM ช่วยลดความซับซ้อนและความหนาแน่นในการทำงานกับ DOM โดยอัปเดตเพียงแค่ส่วนที่มีการเปลี่ยนแปลง ทำให้การแสดงผลของ UI เป็นไปอย่างมีประสิทธิภาพและรวดเร็วขึ้น ซึ่งช่วยให้ผู้พัฒนาสามารถพัฒนาแอปพลิเคชันเว็บที่มีประสิทธิภาพและปรับปรุงได้อย่างสะดวกและมีประสิทธิภาพมากขึ้น



# ทฤษฎีที่เกี่ยวข้อง



Angular เป็นเฟรมเวิร์ก (framework) สำหรับพัฒนาเว็บแอปพลิเคชัน (web applications) และเว็บไซต์ ซึ่งถูกพัฒนาโดยทีม Angular ที่ Google และมีความนิยมและใช้งานอย่างแพร่หลายในวงการพัฒนาซอฟต์แวร์

1. Component-Based Architecture (โครงสร้างแบบคอมโพเนนต์): Angular มุ่งเน้นการพัฒนาแอปพลิเคชันในรูปแบบของคอมโพเนนต์ที่มีโครงสร้างแยกตามส่วนประกอบ ทำให้การพัฒนาและการบำรุงรักษาโค้ดเป็นไปอย่างมีประสิทธิภาพและเป็นระเบียบ
2. Dependency Injection (การฉีดสำเร็จรูป): Angular มีระบบ Dependency Injection ที่ช่วยในการจัดการความเชื่อมโยงของอ็อบเจกต์และการสร้างความสัมพันธ์ระหว่างโมดูลและคอมโพเนนต์ ซึ่งทำให้การพัฒนาและการทดสอบเป็นไปอย่างมีประสิทธิภาพ
3. Directives and Data Binding (คำสั่งและการผูกข้อมูล): Angular มีคำสั่งที่ช่วยในการจัดการ DOM และการทำงานของ UI components โดยใช้ Directives และมีระบบการผูกข้อมูลที่สามารถอัปเดตข้อมูลและสถานะของแอปพลิเคชันโดยอัตโนมัติ
4. Services and HTTP Client: Angular มีบริการ (services) ที่ช่วยในการจัดการโลจิกที่เกี่ยวข้องกับแอปพลิเคชัน เช่น การส่งข้อมูลผ่าน HTTP และการจัดการข้อมูลที่ได้รับจากเซิร์ฟเวอร์
5. Module System (ระบบโมดูล): Angular มีโมดูลที่ช่วยในการจัดการและแยกแยะโค้ดของแอปพลิเคชันอย่างมีประสิทธิภาพ เพื่อให้การพัฒนาเป็นไปอย่างมีระเบียบและประสิทธิภาพ
6. Routing and State Management (การจัดการเส้นทางและสถานะ): Angular มีไลบรารีที่ช่วยในการจัดการเส้นทางและสถานะของแอปพลิเคชัน เช่น Angular Router และ NgRx ซึ่งช่วยในการจัดการข้อมูลและสถานะของแอปพลิเคชันเป็นไปอย่างมีประสิทธิภาพและมีประสิทธิภาพสูง

# ทฤษฎีที่เกี่ยวข้อง

React เป็นเฟรมเวิร์ก (framework) สำหรับพัฒนาเว็บแอปพลิเคชัน (web applications) และเว็บไซต์ที่พัฒนาโดย Facebook ซึ่งมีความนิยมและใช้งานกันอย่างแพร่หลายในวงการพัฒนาเว็บไซต์ในปัจจุบัน

1. Component-Based Architecture (โครงสร้างแบบคอมโพเนนต์): React มุ่งเน้นไปที่การสร้างแอปพลิเคชันในรูปแบบของคอมโพเนนต์ที่มีโครงสร้างแยกตามส่วนประกอบ ทำให้การพัฒนาและบำรุงรักษาโค้ดเป็นไปอย่างมีประสิทธิภาพและสะดวก
2. Virtual DOM (เวอร์ชัน DOM เสมือน): React ใช้ Virtual DOM เพื่อประสิทธิภาพขณะการทำงานของเว็บแอปพลิเคชัน โดย Virtual DOM ช่วยลดการเรียกใช้งาน DOM จริงๆ ที่มีความสูงและซับซ้อน ทำให้การอัปเดตและแสดงผล UI เป็นไปอย่างมีประสิทธิภาพ
3. Reconciliation Algorithm (อัลกอริทึมการตรวจสอบและปรับปรุงสถานะ): React ใช้อัลกอริทึมในการตรวจสอบและปรับปรุงสถานะของ DOM components โดยเปรียบเทียบระหว่าง Virtual DOM กับ DOM จริง และทำการอัปเดตเฉพาะส่วนที่มีการเปลี่ยนแปลงเท่านั้น
4. One-Way Data Binding (การผูกข้อมูลแบบเข้ากันไปมาในทิศทางเดียว): React ใช้การผูกข้อมูลแบบเข้ากันไปมาในทิศทางเดียว (one-way data binding) ซึ่งช่วยให้การจัดการข้อมูลและสถานะของแอปพลิเคชันมีความเป็นระเบียบและชัดเจน
5. JSX (JavaScript XML): JSX เป็นส่วนของ React ที่ช่วยให้ผู้พัฒนาสร้าง UI components ในรูปแบบของ XML ซึ่งทำให้โค้ดดูสะดวกและอ่านง่ายขึ้น และมีประสิทธิภาพในการเขียนและแก้ไขโค้ด
6. State Management (การจัดการสถานะ): React มีเครื่องมือที่ช่วยในการจัดการสถานะของแอปพลิเคชันอย่างมีประสิทธิภาพ เช่น React Hooks หรือ Redux เพื่อช่วยในการจัดการข้อมูลและสถานะของแอปพลิเคชัน

# ทฤษฎีที่เกี่ยวข้อง



Vue.js เป็นเฟรมเวิร์ก (framework) สำหรับพัฒนาเว็บแอปพลิเคชัน (web applications) และเว็บไซต์ที่มีความยืดหยุ่นและสะดวกต่อการใช้งาน ซึ่งได้รับความนิยมอย่างแพร่หลายในวงการพัฒนาเว็บไซต์

1. Component-Based Architecture (โครงสร้างแบบคอมโพเนนต์): Vue.js มุ่งเน้นการพัฒนาแอปพลิเคชันในรูปแบบของคอมโพเนนต์ที่สามารถนำมาใช้ซ้ำได้ ซึ่งทำให้การพัฒนาและการบำรุงรักษาโค้ดเป็นไปอย่างมีประสิทธิภาพและเป็นระเบียบ
2. Reactive Data Binding (การผูกข้อมูลแบบโต้ตอบ): Vue.js มีระบบการผูกข้อมูลแบบโต้ตอบที่ช่วยให้สามารถอัปเดตข้อมูลและสถานะของแอปพลิเคชันโดยอัตโนมัติเมื่อข้อมูลเปลี่ยนแปลง ทำให้การจัดการข้อมูลเป็นไปอย่างมีประสิทธิภาพและเรียลไทม์
3. Virtual DOM (เวอร์ชัน DOM เสมือน): Vue.js ใช้ Virtual DOM เพื่อประสิทธิภาพในการแสดงผลและการอัปเดตข้อมูล โดย Virtual DOM ช่วยลดการทำงานกับ DOM จริงๆ ที่มีความซับซ้อน และช่วยในการอัปเดตและแสดงผลข้อมูลอย่างมีประสิทธิภาพ
4. Directives and Reactive Components (คำสั่งและคอมโพเนนต์ที่เป็นโต้ตอบ): Vue.js มีคำสั่งที่ช่วยในการจัดการ DOM และ UI components อย่างง่ายดายและเรียลไทม์ โดยมีคอนเซ็ปต์ที่เป็นโต้ตอบที่ช่วยให้สามารถจัดการสถานะและการแสดงผลข้อมูลได้อย่างสะดวก
5. Routing and State Management (การจัดการเส้นทางและสถานะ): Vue.js มีไลบรารีที่ช่วยในการจัดการเส้นทางและสถานะของแอปพลิเคชัน เช่น Vue Router และ Vuex ซึ่งช่วยให้การจัดการข้อมูลและสถานะของแอปพลิเคชันเป็นไปอย่างมีประสิทธิภาพและมีประสิทธิภาพสูง
6. Server-Side Rendering (การเรนเดอร์ที่เป็นด้านเซิร์ฟเวอร์): Vue.js มีความสามารถในการสร้างเว็บไซต์ที่สามารถทำงานได้ทั้งบนเว็บเซิร์ฟเวอร์และบนเบราว์เซอร์ ซึ่งช่วยให้สามารถให้ประสบการณ์การใช้งานที่ดีกับผู้ใช้ได้อย่างต่อเนื่อง

# ทฤษฎีที่เกี่ยวข้อง

Svelte เป็นเครื่องมือสำหรับพัฒนาเว็บแอปพลิเคชัน (web application) ที่ใช้ในการสร้าง User Interface (UI) โดยมุ่งเน้นไปที่การสร้างโค้ดที่มีประสิทธิภาพและง่ายต่อการบำรุงรักษา โดยที่เฟรมเวิร์กนี้จะทำการคอมไพล์ (compile) โค้ดให้กลายเป็นโค้ด JavaScript แบบเชื่อมต่อ (vanilla JavaScript) ที่ทำงานได้โดยตรงบนเบราว์เซอร์ ซึ่งช่วยลดปัญหาการจัดการ Virtual DOM และการทำงานของระบบที่ซับซ้อนในขณะรันเว็บแอปพลิเคชัน

1. Compiler Theory: การทฤษฎีเกี่ยวกับการคอมไพล์โค้ดและกระบวนการการทำงานของคอมไพเลอร์ เช่น การแปลงโค้ดจากภาษามนุษย์เป็นภาษาเครื่อง หรือวิธีการประมวลผลโค้ดเพื่อให้ได้เอาต์พุตที่ต้องการ
2. Virtual DOM vs. Reactive Programming: การศึกษาเกี่ยวกับความแตกต่างระหว่างการใช้ Virtual DOM ในเฟรมเวิร์กอื่น ๆ และวิธีการทำงานของการโต้ตอบแบบรีแอคทีฟใน Svelte ซึ่งช่วยลดการทำงานที่ไม่จำเป็นและเพิ่มประสิทธิภาพในการจัดการ DOM
3. Component-Based Architecture: การศึกษาเกี่ยวกับโครงสร้างและการออกแบบระบบโครงสร้างของเว็บแอปพลิเคชันที่ใช้งาน Svelte ซึ่งมีการแบ่งส่วน (component) และการสื่อสารระหว่างส่วนของโค้ด
4. Performance Optimization: การศึกษาเกี่ยวกับวิธีการเพิ่มประสิทธิภาพในการโหลดและแสดงผลของเว็บแอปพลิเคชันด้วย Svelte ซึ่งเน้นไปที่การลดขนาดของโค้ดและประสิทธิภาพการทำงานของ DOM
5. Reactivity and Data Binding: การศึกษาเกี่ยวกับวิธีการใช้งานระบบการโต้ตอบและการผูกข้อมูล (data binding) ใน Svelte ซึ่งช่วยให้สามารถอัปเดต UI โดยอัตโนมัติเมื่อข้อมูลเปลี่ยนแปลง
6. User Experience: การศึกษาผลกระทบต่อประสบการณ์ของผู้ใช้ที่ได้จากการใช้งานเว็บแอปพลิเคชันที่สร้างด้วย Svelte โดยการพิจารณาทิศทางประสิทธิผล



# ทฤษฎีที่เกี่ยวข้อง

Google Chrome DevTools ซึ่งสามารถใช้ในการ  
วัดประสิทธิภาพของการจัดการ DOM ได้โดยตรง  
โดยสามารถตรวจสอบเวลาที่ใช้ในการโหลดและ  
แสดงผลของ DOM ได้

# ทฤษฎีที่เกี่ยวข้อง

Google Chrome DevTools ซึ่งสามารถใช้ในการ  
วัดประสิทธิภาพของการจัดการ DOM ได้โดยตรง  
โดยสามารถตรวจสอบเวลาที่ใช้ในการโหลดและ  
แสดงผลของ DOM ได้