

บทที่ 1 บทนำ

ระบบติดตามการส่งขอตำแหน่งที่สูงขึ้นของมหาวิทยาลัยกาฬสินธุ์ถูกพัฒนาขึ้นเพื่อแก้ไขปัญหาและข้อจำกัดของกระบวนการเดิมขั้นตอนที่ซับซ้อนและใช้เวลานานกระบวนการขอตำแหน่งที่สูงขึ้นมีหลายขั้นตอนที่เกี่ยวข้องกับหน่วยงานและบุคคลหลายฝ่ายทำให้เกิดความล่าช้าและความไม่แน่นอนในการดำเนินการ ทางมหาวิทยาลัยกาฬสินธุ์เป็นส่วนหนึ่งของระบบสังคมและระบบการศึกษาระดับสูงของประเทศจึงมีภารกิจในการพัฒนาขีดความสามารถของทรัพยากรมนุษย์ให้มีคุณภาพและมีคุณสมบัติที่พึงประสงค์ตอบสนองความต้องการของการพัฒนาประเทศ รวมทั้งการวิจัยพัฒนา และการสร้าง ความความรู้เพื่อการพึ่งพาตนเองได้ระบบการกำหนดตำแหน่งทางวิชาการจากแผนพัฒนาให้ับผลงานในการขอตำแหน่งทางวิชาการมีความครอบคลุมและหลากหลายตามความถนัดมากขึ้น เช่น มีผลงานวิชาการที่หลากหลายมากขึ้นพิจารณาสิ่งประดิษฐ์และนวัตกรรม ผลกระทบต่อสังคม หรือชื่อเสียงที่ทำให้กับมหาวิทยาลัย เป็นต้น

การปรับโครงสร้างตำแหน่งวิชาการใหม่ซึ่งประกอบไปด้วยตำแหน่งทางวิชาการระดับความชำนาญ ความสามารถในการผลิตนวัตกรรม โดยมีตำแหน่งทางวิชาการต่างๆการพัฒนาขีดความสามารถของอาจารย์โดยให้มีการเข้าสู่ตำแหน่งทางวิชาการมีความสำคัญต่อมหาวิทยาลัยในแง่การจัดลำดับมหาวิทยาลัยโดยข้อมูลการจัดอันดับในด้านต่าง ๆ ของมหาวิทยาลัย ด้านที่มีการมีจำนวนผู้ดำรงตำแหน่งทางวิชาการเกี่ยวข้องและส่งผลทางตรงต่อการจัดลำดับเนื่องจากเป็นการแสดงถึงศักยภาพและความน่าเชื่อถือทางด้านวิชาการของมหาวิทยาลัยซึ่งส่งผลต่อการผลิตบัณฑิตที่มีคุณภาพซึ่งได้แก่ ด้านงานวิจัย การให้คะแนนด้านงานวิจัย จะดูจากคุณภาพงานวิจัย ปริมาณ รายได้ และชื่อเสียงของงานวิจัย ด้านการอ้างอิงการให้คะแนนด้านการอ้างอิง จะดูจากการอ้างอิงในงานวิจัยการนำงานวิจัยของสถาบันไปใช้อ้างอิงด้านการสอนการให้คะแนนด้านการสอน จะดูจากคุณภาพการสอนและสภาพแวดล้อมในการเรียนรู้ของผู้เรียน

1.1 ที่มาของปัญหา

ลดการใช้เอกสารจำนวนมาก : กระบวนการเดิมใช้เอกสารจำนวนมากในการยื่นขอและ

พิจารณา ซึ่งอาจเกิดปัญหาเอกสารสูญหายหรือจัดเก็บไม่เป็นระบบ ทำให้เกิดความยุ่ง

ยากในการติดตามและตรวจสอบ

ขาดความโปร่งใส : ผู้ยื่นขอตำแหน่งไม่สามารถติดตามสถานะการพิจารณาได้อย่าง

สะดวกและทันท่วงที ทำให้เกิดความไม่มั่นใจและความไม่พึงพอใจในกระบวนการ

ความยากลำบากในการติดตามผลการประเมิน: การติดตามผลการประเมินจากผู้ทรง

คุณวุฒิในระบบเดิมอาจทำได้ล่าช้าและไม่สะดวกเนื่องจากต้องติดต่อผ่านสำนักงานอธิการบดี

ความสำคัญของปัญหา

- ปัญหาและข้อจำกัดดังกล่าวส่งผลกระทบต่อหลายฝ่ายที่เกี่ยวข้อง ดังนี้

ผู้ยื่นขอตำแหน่ง : เกิดความไม่มั่นใจ ความกังวล และความไม่พึงพอใจ เนื่องจากไม่สามารถติดตามสถานะการพิจารณาได้ และอาจเกิดความล่าช้าในการได้รับตำแหน่งที่สูงขึ้น

ผู้มีอำนาจตัดสินใจ : ขาดข้อมูลที่จำเป็นและทันท่วงทีในการพิจารณาตำแหน่งทำให้การตัดสินใจอาจไม่เป็นไปอย่างมีประสิทธิภาพ

มหาวิทยาลัย : กระบวนการบริหารงานบุคคลไม่มีประสิทธิภาพเกิดความล่าช้าในการพัฒนาบุคลากรและอาจส่งผลกระทบต่อภาพลักษณ์ของมหาวิทยาลัยการพัฒนาระบบติดตามการส่งขอตำแหน่งที่สูงขึ้นจึงมีความสำคัญอย่างยิ่งในการแก้ไขปัญหาและข้อจำกัดดังกล่าวเพื่อเพิ่มประสิทธิภาพ ความโปร่งใส และความพึงพอใจของทุกฝ่ายที่เกี่ยวข้อง อีกทั้งยังเป็นการส่งเสริมการพัฒนาบุคลากรและยกระดับคุณภาพการบริหารงานบุคคลของมหาวิทยาลัยกาฬสินธุ์ให้ดียิ่งขึ้น

1.2 วัตถุประสงค์ของโครงการ

1.2.1. พัฒนาระบบติดตามสถานะการยื่นขอตำแหน่งที่สูงขึ้น: เพื่อให้ผู้ยื่นขอสามารถตรวจสอบสถานะการพิจารณาเอกสารของตนได้ตลอดเวลา

1.2.2. ลดระยะเวลาในการพิจารณา: โดยการปรับปรุงขั้นตอนการดำเนินงานและการแจ้งเตือนอัตโนมัติ

1.2.3. เพิ่มความโปร่งใส: โดยการแสดงข้อมูลที่จำเป็นและขั้นตอนการพิจารณาอย่างชัดเจน

1.2.4. ลดการใช้เอกสาร: โดยการนำระบบอิเล็กทรอนิกส์มาใช้ในการจัดเก็บและส่งเอกสาร

1.3 ขอบเขตของโครงการ

1.3.1. ผู้ใช้งาน: คณาจารย์และบุคลากรของมหาวิทยาลัยกาฬสินธุ์ที่ยื่นขอตำแหน่งที่สูงขึ้น

1.3.2. ตำแหน่งที่ครอบคลุม: ตำแหน่งทางวิชาการ (ผู้ช่วยศาสตราจารย์, รองศาสตราจารย์, ศาสตราจารย์) และตำแหน่งบริหาร (หัวหน้าภาควิชา, คณบดี, รองอธิการบดี, อธิการบดี)

1.3.3. ฟังก์ชันการทำงาน: การยื่นขอตำแหน่ง, การอัปโหลดเอกสาร, การตรวจสอบสถานะ, การแจ้งเตือน, การรายงาน

1.4 ประโยชน์ที่ได้รับ

1.4.1. ผู้ยื่นขอ: ทราบสถานะการพิจารณาอย่างรวดเร็ว, ลดความกังวล, วางแผนการทำงานได้ดีขึ้น

1.4.2. ผู้พิจารณา: กระบวนการพิจารณามีประสิทธิภาพมากขึ้น, ลดข้อผิดพลาด, ตรวจสอบย้อนหลังได้

1.4.3. มหาวิทยาลัย: ภาพลักษณ์ที่ดีขึ้น, พัฒนาศักยภาพได้รวดเร็ว, ลดต้นทุนการดำเนินงาน

1.5 เครื่องมือที่ใช้พัฒนา

1.5.1. ภาษาโปรแกรม: Python, JavaScript

1.5.2. เฟรมเวิร์ก: Django, React

1.5.3. ฐานข้อมูล: PostgreSQL

1.5.4. เครื่องมืออื่นๆ: Git (สำหรับการจัดการโค้ด), Docker (สำหรับการปรับใช้)

บทที่2

เอกสารและงานวิจัยที่เกี่ยวข้อง

ความเป็นมาของ React :

React เกิดขึ้นจากความต้องการในการสร้างส่วนต่อประสานผู้ใช้ (UI) ที่มีประสิทธิภาพและปรับขนาดได้ง่ายสำหรับแอปพลิเคชันขนาดใหญ่ โดย Jordan Walke วิศวกรซอฟต์แวร์ของ Facebook ได้เริ่มพัฒนา React ในปี 2011 และเปิดตัวเป็นโอเพนซอร์สในปี 2013 แรงบันดาลใจและปัญหาที่ React แก้ไข

- การจัดการ UI ที่ซับซ้อน: แอปพลิเคชันขนาดใหญ่มี UI ที่ซับซ้อนและเปลี่ยนแปลงบ่อย การอัปเดต UI แบบดั้งเดิมมักจะช้าและยากต่อการดูแลรักษา
- การนำโค้ดกลับมาใช้ใหม่: การสร้างส่วนประกอบ UI ที่สามารถนำกลับมาใช้ใหม่ได้เป็นสิ่งสำคัญในการพัฒนาแอปพลิเคชันขนาดใหญ่
- ประสิทธิภาพ: การแสดงผล UI ที่รวดเร็วและราบรื่นเป็นสิ่งสำคัญสำหรับประสบการณ์ผู้ใช้ที่ดี
- ประสิทธิภาพ: การแสดงผล UI ที่รวดเร็วและราบรื่นเป็นสิ่งสำคัญสำหรับประสบการณ์ผู้ใช้ที่ดี

แนวคิดหลักของ React :

- Component-Based Architecture: React สนับสนุนการสร้าง UI ด้วยคอมโพเนนต์ ซึ่งเป็นส่วนประกอบย่อย ๆ ที่สามารถนำกลับมาใช้ใหม่ได้ ทำให้โค้ดเป็นระเบียบและดูแลรักษาง่ายขึ้น
 - Virtual DOM: React ใช้ Virtual DOM ซึ่งเป็นสำเนาของ DOM จริง ทำให้สามารถอัปเดต UI ได้อย่างมีประสิทธิภาพเมื่อข้อมูลเปลี่ยนแปลง
 - Declarative Syntax: React ใช้ JSX ซึ่งเป็นส่วนขยายของ JavaScript ที่ช่วยให้เขียนโค้ด HTML-like ภายใน JavaScript ทำให้การสร้าง UI สะดวกและอ่านง่ายขึ้น
- การเติบโตและการยอมรับ React ได้รับความนิยมอย่างรวดเร็วในหมู่นักพัฒนาเนื่องจากความเรียบง่าย ประสิทธิภาพ และความยืดหยุ่น

พัฒนาการที่สำคัญของ React

- 2011: Jordan Walke เริ่มพัฒนา React
- 2013: React เปิดตัวเป็นโอเพ่นซอร์ส
- 2015: React Native เปิดตัว ทำให้สามารถใช้ React สร้างแอปพลิเคชันมือถือได้
- 2017: React Fiber เปิดตัว ปรับปรุงประสิทธิภาพและความสามารถในการตอบสนองของ

React

- 2020: React 17 เปิดตัว เน้นการปรับปรุงประสบการณ์การใช้งานและการทำงานร่วมกับเครื่องมืออื่นๆ

- 2022: React 18 เปิดตัว มีฟีเจอร์ใหม่ ๆ เช่น Concurrent Rendering และ Suspense ในอนาคต

ของ React

React ยังคงได้รับการพัฒนาและปรับปรุงอย่างต่อเนื่อง โดยมีเป้าหมายในการทำให้การสร้าง UI ที่ซับซ้อนเป็นเรื่องง่ายและมีประสิทธิภาพมากยิ่งขึ้น React ยังคงเป็นหนึ่งในไลบรารี JavaScript ที่ได้รับความนิยมมากที่สุด และมีแนวโน้มที่จะยังคงเป็นเช่นนั้นต่อไปในอนาคต

พื้นฐาน React.js

React.js เป็นไลบรารี JavaScript ยอดนิยมที่ใช้สำหรับสร้างส่วนต่อประสานผู้ใช้ (UI) ที่ทันสมัยและมีประสิทธิภาพสูง มาทำความเข้าใจพื้นฐานสำคัญๆ กัน:

1. คอมโพเนนต์ (Components)

- * หน่วยการสร้างพื้นฐาน: คอมโพเนนต์คือส่วนประกอบอิสระที่นำมาประกอบกันเป็น UI ทั้งหมด
- * นำกลับมาใช้ใหม่ได้: คอมโพเนนต์สามารถนำไปใช้ซ้ำในส่วนต่างๆ ของแอปพลิเคชัน ช่วยให้โค้ดเป็นระเบียบและดูแลรักษาง่าย
- * มีสองประเภท:
 - * ฟังก์ชันคอมโพเนนต์ (Functional Components): ง่ายต่อการเขียนและทดสอบ เหมาะสำหรับคอมโพเนนต์ที่ไม่มี state หรือ logic ซับซ้อน
 - * คลาสคอมโพเนนต์ (Class Components): มีความสามารถมากกว่า สามารถมี state และ lifecycle methods เหมาะสำหรับคอมโพเนนต์ที่มี logic ซับซ้อน

2. JSX (JavaScript XML)

- * ส่วนขยายของ JavaScript: JSX ช่วยให้เขียนโค้ด HTML-like ภายใน JavaScript ทำให้การสร้าง UI สะดวกและอ่านง่ายขึ้น
- * แปลงเป็น JavaScript: JSX จะถูกแปลงเป็น JavaScript ธรรมดาโดย Babel ก่อนนำไปรันบน browser

3. Props (Properties)

- * ส่งผ่านข้อมูล: Props ใช้สำหรับส่งผ่านข้อมูลจากคอมโพเนนต์ parent ไปยังคอมโพเนนต์ child
- * อ่านอย่างเดียว (Read-only): คอมโพเนนต์ child ไม่สามารถแก้ไข props ที่ได้รับมาได้โดยตรง
- * ทำให้คอมโพเนนต์มีความยืดหยุ่น: Props ช่วยให้คอมโพเนนต์สามารถปรับแต่งการทำงานได้ตามข้อมูลที่ได้รับ

4. State

- * ข้อมูลภายในคอมโพเนนต์: State ใช้สำหรับเก็บข้อมูลที่สามารถเปลี่ยนแปลงได้ภายในคอมโพเนนต์
- * เมื่อ state เปลี่ยน คอมโพเนนต์จะ render ใหม่: การอัปเดต state จะทำให้ React อัปเดต UI ให้ตรงกับ state ใหม่
- * useState hook: ใช้สำหรับสร้างและจัดการ state ใน functional components

5. Virtual DOM

* สำเนาของ DOM จริง: Virtual DOM เป็นโครงสร้างข้อมูลในหน่วยความจำที่แสดงถึง UI ของแอปพลิเคชัน

* เพิ่มประสิทธิภาพการอัปเดต UI: เมื่อ state เปลี่ยนแปลง React จะอัปเดต Virtual DOM ก่อน แล้วคำนวณความแตกต่างระหว่าง Virtual DOM เก่าและใหม่ จากนั้นอัปเดตเฉพาะส่วนที่จำเป็นบน DOM จริงเท่านั้น ทำให้การอัปเดต UI รวดเร็วและมีประสิทธิภาพ

6. เหตุการณ์ (Events)

* การโต้ตอบกับผู้ใช้: เหตุการณ์ (เช่น การคลิก การพิมพ์ การส่งแบบฟอร์ม) ช่วยให้แอปพลิเคชันตอบสนองต่อการกระทำของผู้ใช้

* การผูกเหตุการณ์: ผูก event handlers กับองค์ประกอบ JSX เพื่อจัดการเหตุการณ์ต่างๆ

* Synthetic events: React สร้าง abstraction layer เหนือ native events ของ browser ทำให้ event handling เป็นแบบ cross-browser

7. Lifecycle Methods (เฉพาะ Class Components)

* จัดการช่วงชีวิตของคอมโพเนนต์: Lifecycle methods อนุญาตให้รันโค้ดในช่วงเวลาต่างๆ ของชีวิตคอมโพเนนต์ เช่น เมื่อคอมโพเนนต์ถูกสร้างขึ้น อัปเดต หรือถูกทำลาย

* ตัวอย่าง: componentDidMount, componentDidUpdate, componentWillUnmount

การทำงานขั้นสูงของ React.js

เมื่อคุณเข้าใจพื้นฐานของ React.js แล้ว ลองมาสำรวจการทำงานขั้นสูงที่ช่วยให้คุณสร้างแอปพลิเคชันที่ซับซ้อนและมีประสิทธิภาพยิ่งขึ้น:

1. Context API

- * การจัดการ State ทั่วทั้งแอป: Context API ช่วยให้คุณสามารถแชร์ state ระหว่างคอมโพเนนต์ต่างๆ ได้โดยไม่ต้องส่ง props ผ่านหลายชั้น (prop drilling)
- * เหมาะสำหรับ: ข้อมูลที่ใช้ร่วมกันทั่วทั้งแอปพลิเคชัน เช่น ธีม ภาษา หรือข้อมูลผู้ใช้
- * ข้อควรระวัง: การใช้ Context มากเกินไปอาจทำให้โค้ดยากต่อการจัดการ

2. Higher-Order Components (HOCs)

- * คอมโพเนนต์ที่รับคอมโพเนนต์เป็น input: HOCs เป็นฟังก์ชันที่รับคอมโพเนนต์เป็น input และส่งกลับคอมโพเนนต์ใหม่ที่มีคุณสมบัติเพิ่มเติม
- * นำโค้ดกลับมาใช้ใหม่: HOCs ช่วยให้คุณแยก logic ที่ซับซ้อนออกจากคอมโพเนนต์ และนำ logic นั้นกลับมาใช้ใหม่ได้ในหลายๆ คอมโพเนนต์
- * ตัวอย่าง: การเพิ่ม authentication, การดึงข้อมูลจาก API, หรือการจัดการ state

3. Render Props

- * ส่งผ่านฟังก์ชันเป็น prop: Render props เป็นเทคนิคที่ช่วยให้คุณแชร์ logic ระหว่างคอมโพเนนต์ได้ โดยส่งผ่านฟังก์ชันเป็น prop ไปยังคอมโพเนนต์ child
- * ความยืดหยุ่น: Render props ให้ความยืดหยุ่นมากกว่า HOCs ในการกำหนดว่าจะ render อะไรและอย่างไร
- * ตัวอย่าง: การสร้างคอมโพเนนต์ที่สามารถปรับแต่งการแสดงผลได้

4. Error Boundaries

- * จัดการข้อผิดพลาด: Error boundaries เป็นคอมโพเนนต์พิเศษที่ช่วยให้คุณสามารถดักจับและจัดการข้อผิดพลาดที่เกิดขึ้นในคอมโพเนนต์ child ได้
- * ป้องกันแอปพลิเคชันล่ม: Error boundaries ช่วยป้องกันไม่ให้ข้อผิดพลาดในส่วนหนึ่งของแอปพลิเคชันส่งผลกระทบต่อส่วนอื่นๆ

5. Portals

- * Render คอมโพเนนต์นอก DOM hierarchy: Portals อนุญาตให้คุณ render คอมโพเนนต์ไปยังตำแหน่งอื่นใน DOM ที่อยู่นอกเหนือจาก hierarchy ของคอมโพเนนต์ parent
- * ตัวอย่าง: การสร้าง modals, tooltips, หรือ overlays

6. Refs

- * เข้าถึง DOM elements หรือคอมโพเนนต์โดยตรง: Refs ช่วยให้คุณสามารถเข้าถึง DOM elements หรือคอมโพเนนต์ React ได้โดยตรง
- * ใช้ด้วยความระมัดระวัง: การใช้ refs มากเกินไปอาจทำให้โค้ดยากต่อการดูแลรักษา ควรใช้ refs เฉพาะเมื่อจำเป็นจริงๆ

7. การเพิ่มประสิทธิภาพ (Performance Optimization)

- * Memoization: ใช้ `React.memo`, `useMemo`, และ `useCallback` เพื่อป้องกันการ render คอมโพเนนต์หรือการคำนวณซ้ำที่ไม่จำเป็น
- * Code splitting: แบ่ง code bundle ออกเป็น chunks เล็ก ๆ เพื่อโหลดเฉพาะส่วนที่จำเป็นเมื่อผู้ใช้เข้าถึงส่วนต่าง ๆ ของแอปพลิเคชัน
- * Lazy loading: โหลดคอมโพเนนต์หรือทรัพยากรอื่น ๆ เฉพาะเมื่อจำเป็น
- * Virtualization: แสดงเฉพาะรายการที่มองเห็นได้บนหน้าจอ เพื่อเพิ่มประสิทธิภาพการแสดงผลรายการขนาดใหญ่

8. การทดสอบ (Testing)

- * Jest: Test runner ยอดนิยมสำหรับ React applications
- * React Testing Library: ไลบรารีที่ช่วยในการทดสอบคอมโพเนนต์ React โดยเน้นที่การทดสอบจากมุมมองของผู้ใช้

9. React with TypeScript

- * TypeScript: Superset ของ JavaScript ที่เพิ่ม static typing
- * ข้อดี: ป้องกันข้อผิดพลาด, ปรับปรุง code readability, ช่วยให้ IDE สามารถให้ code completion และ refactoring ที่ดีขึ้น

10. State Management Libraries

- * Redux, MobX, Zustand: มีประโยชน์สำหรับการจัดการ state ที่ซับซ้อนและมีโครงสร้าง

การจัดการ State ใน React

React เป็นไลบรารีที่ใช้ในการสร้าง User Interface (UI) โดยที่หนึ่งในหลักการสำคัญของการพัฒนาแอปพลิเคชันใน React คือการจัดการกับ "State" ซึ่ง State เป็นข้อมูลหรือสถานะของคอมโพเนนต์ที่สามารถเปลี่ยนแปลงได้ โดยที่ UI จะทำการอัปเดตเมื่อ State มีการเปลี่ยนแปลง การจัดการ State จึงเป็นหัวใจสำคัญในการพัฒนาแอปพลิเคชันที่ตอบสนองต่อการกระทำของผู้ใช้ได้อย่างเหมาะสม

1. ความเข้าใจพื้นฐานเกี่ยวกับ State

ใน React, State เป็นข้อมูลที่ถูกเก็บอยู่ภายในคอมโพเนนต์หนึ่ง ๆ และสามารถส่งผ่านลงไปยังคอมโพเนนต์ลูกผ่านทาง props ได้ โดยที่ State จะถูกใช้ในกรณีที่ข้อมูลมีการเปลี่ยนแปลงอยู่เสมอ เช่น ข้อมูลที่ได้รับจากผู้ใช้ หรือผลลัพธ์ที่ได้จากการคำนวณบางอย่างภายในคอมโพเนนต์

State ถูกกำหนดและจัดการภายในคอมโพเนนต์ผ่านการใช้ `useState` hook ซึ่งเป็นฟังก์ชันที่ React ให้มาเพื่อใช้ในการจัดการ State ในฟังก์ชันคอมโพเนนต์ ตัวอย่างการใช้ `useState` มีดังนี้:

javascript

```
import React, { useState } from 'react';

function Counter() {
  // สร้าง State ชื่อ "count" พร้อมด้วยฟังก์ชัน "setCount" สำหรับเปลี่ยนค่า State
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>
        Click me
      </button>
    </div>
  );
}
```

ในตัวอย่างนี้ เราได้สร้าง State ที่ชื่อ `count` ซึ่งเก็บค่าจำนวนครั้งที่ปุ่มถูกคลิก และทุกครั้งที่ผู้ใช้กดปุ่ม ค่าของ `count` จะถูกอัปเดตด้วยการเรียกใช้ `setCount` ส่งผลให้ UI มีการอัปเดตตามไปด้วย

2. การจัดการ State ในคอมโพเนนต์หลายตัว

เมื่อแอปพลิเคชันของคุณมีความซับซ้อนมากขึ้น อาจจะมีการใช้ State ในหลาย ๆ คอมโพเนนต์ที่ต้องการสื่อสารกัน การจัดการ State ในระดับนี้สามารถทำได้โดยการย้าย State ขึ้นไปยังคอมโพเนนต์พARENT เพื่อให้สามารถส่งข้อมูลหรือฟังก์ชันลงไปคอมโพเนนต์ลูกที่เกี่ยวข้องได้

วิธีการนี้เรียกว่า "Lifting State Up" โดยการแชร์ State ระหว่างคอมโพเนนต์พARENT กับลูก ตัวอย่างเช่น หากมีคอมโพเนนต์สองตัวที่ต้องใช้ข้อมูลเดียวกันจาก State สามารถทำได้โดย:

javascript

```
import React, { useState } from 'react';

function ParentComponent() {
  const [sharedState, setSharedState] = useState("Hello");

  return (
    <div>
      <ChildComponent1 sharedState={sharedState} />
      <ChildComponent2 setSharedState={setSharedState} />
    </div>
  );
}

function ChildComponent1({ sharedState }) {
  return <div>{sharedState}</div>;
}

function ChildComponent2({ setSharedState }) {
  return <button onClick={() => setSharedState("Hello from Child 2!")}>
    Change State
  </button>;
}
```

ในตัวอย่างนี้ `sharedState` ถูกจัดการอยู่ใน `ParentComponent` และถูกส่งผ่านลงไปยัง `ChildComponent1` และ `ChildComponent2` ผ่านทาง props โดยที่ `ChildComponent2` สามารถเปลี่ยนค่า `sharedState` ได้ และ `ChildComponent1` จะแสดงผลลัพธ์ที่เปลี่ยนแปลงทันที

3. การใช้ Context สำหรับ State ที่ต้องแชร์ในหลายคอมโพเนนต์

ในกรณีที่แอปพลิเคชันมีคอมโพเนนต์หลายระดับที่ต้องใช้ State เดียวกัน การส่งผ่าน props ระหว่างหลายระดับอาจทำให้โค้ดดูซับซ้อนและยุ่งยาก ในกรณีนี้ React Context สามารถช่วยให้การแชร์ State ในหลาย ๆ คอมโพเนนต์ทำได้ง่ายขึ้น

React Context ให้คุณสามารถสร้างพื้นที่เก็บข้อมูล (Store) ที่สามารถเข้าถึงได้จากทุกคอมโพเนนต์ที่อยู่ใน Context นั้น ๆ ตัวอย่างการใช้งาน Context มีดังนี้:

javascript

```
import React, { createContext, useState, useContext } from 'react';

// สร้าง Context
const MyContext = createContext();

function ParentComponent() {
  const [state, setState] = useState("Hello Context");

  return (
    <MyContext.Provider value={{ state, setState }}>
      <ChildComponent />
    </MyContext.Provider>
  );
}

function ChildComponent() {
  const { state, setState } = useContext(MyContext);

  return (
    <div>
      <p>{state}</p>
      <button onClick={() => setState("Updated via Context")}>Update</button>
    </div>
  );
}
```

ในตัวอย่างนี้ **ParentComponent** จะสร้าง Context ที่มี State และฟังก์ชันการอัปเดต State ที่สามารถเข้าถึงได้ใน **ChildComponent** โดยใช้ **useContext** hook ซึ่งทำให้ไม่จำเป็นต้องส่ง props ผ่านทุกคอมโพเนนต์ที่อยู่ระหว่าง **ParentComponent** และ **ChildComponent**

4. การจัดการ State ขั้นสูงด้วย Redux

สำหรับแอปพลิเคชันที่มีการจัดการ State ที่ซับซ้อนขึ้น Redux สามารถเข้ามาช่วยในเรื่องนี้ Redux เป็นไลบรารีจัดการ State ที่ให้คุณจัดการ State ของแอปพลิเคชันในที่เดียว (centralized store) และสามารถควบคุมการเปลี่ยนแปลงของ State ผ่าน action และ reducer

การใช้ Redux จะช่วยให้การจัดการ State เป็นไปอย่างเป็นระบบ ทำให้สามารถควบคุมการเปลี่ยนแปลงได้ง่ายขึ้น โดยเฉพาะในแอปพลิเคชันที่มีการเปลี่ยนแปลง State จำนวนมาก

การทดสอบและ Debugging เป็นกระบวนการที่สำคัญในวงจรซอฟต์แวร์ เพื่อให้แน่ใจว่าซอฟต์แวร์ทำงานตามที่ออกแบบและไม่มีข้อผิดพลาด (bugs) ที่อาจก่อให้เกิดปัญหาในการใช้งาน กระบวนการนี้สามารถแบ่งออกเป็นสองส่วนหลักๆ คือ การทดสอบ (Testing) และการ Debugging

การทดสอบ (Testing)

การทดสอบเป็นขั้นตอนที่ใช้ตรวจสอบว่าโปรแกรมหรือซอฟต์แวร์ทำงานตามที่คาดหวัง โดยการทดสอบสามารถแบ่งออกเป็นหลายระดับและหลายประเภท ขึ้นอยู่กับวัตถุประสงค์และขอบเขตของการทดสอบ

ประเภทของการทดสอบ

1 : Unit Testing: เป็นการทดสอบที่มุ่งเน้นไปที่หน่วยหรือส่วนเล็กๆ ของซอฟต์แวร์ เช่น ฟังก์ชันหรือคลาสในโปรแกรม การทดสอบนี้มีวัตถุประสงค์เพื่อตรวจสอบว่าหน่วยเล็กๆ เหล่านี้ทำงานถูกต้องและสอดคล้องกับที่ออกแบบไว้

2 : Integration Testing: เป็นการทดสอบที่เน้นการตรวจสอบการทำงานร่วมกันของหน่วยต่างๆ ภายในโปรแกรม เพื่อให้แน่ใจว่าการเชื่อมต่อระหว่างหน่วยหรือโมดูลเหล่านี้ทำงานได้อย่างราบรื่น

3 : System Testing: เป็นการทดสอบที่มุ่งเน้นการทดสอบซอฟต์แวร์ทั้งหมดว่าเป็นไปตามความต้องการของระบบทั้งหมดหรือไม่ เป็นการทดสอบที่เกี่ยวข้องกับฟังก์ชันทั้งหมดของซอฟต์แวร์ในสภาพแวดล้อมที่ใกล้เคียงกับการใช้งานจริง

4 : Acceptance Testing: เป็นการทดสอบขั้นสุดท้ายที่ใช้ตรวจสอบว่าโปรแกรมหรือระบบตรงตามความต้องการของผู้ใช้งานหรือไม่ โดยทั่วไปแล้วจะดำเนินการโดยผู้ใช้งานหรือผู้ที่เกี่ยวข้องโดยตรงกับโครงการ

เทคนิคการทดสอบ

1 : Black-box Testing: เป็นการทดสอบโดยไม่ต้องคำนึงถึงโครงสร้างภายในของซอฟต์แวร์ แต่ทดสอบตามอินพุตและเอาต์พุตที่คาดหวังไว้

2 : White-box Testing: เป็นการทดสอบที่มุ่งเน้นการตรวจสอบโครงสร้างภายในของซอฟต์แวร์ ผู้ทดสอบต้องเข้าใจโค้ดและตรรกะการทำงานภายในของซอฟต์แวร์

3 : Regression Testing: เป็นการทดสอบซ้ำเพื่อให้แน่ใจว่าการปรับปรุงหรือแก้ไขบั๊กในซอฟต์แวร์ไม่ได้ก่อให้เกิดข้อผิดพลาดใหม่หรือส่งผลกระทบต่อส่วนอื่นๆ ของระบบ

การ Debugging

1 : Debugging เป็นกระบวนการที่ใช้ในการตรวจหาข้อผิดพลาดหรือบั๊กในซอฟต์แวร์ รวมถึงการหาวิธีแก้ไขข้อผิดพลาดเหล่านั้น ขั้นตอนนี้มักจะเกิดขึ้นหลังจากที่พบปัญหาในการทดสอบ

2 : กระบวนการ Debugging

การระบุปัญหา (Identify the Problem): ขั้นตอนแรกของการ Debugging คือการระบุและเข้าใจปัญหาที่เกิดขึ้น การรวบรวมข้อมูลเช่น ข้อความแสดงข้อผิดพลาด (error messages) หรือพฤติกรรมที่ไม่พึงประสงค์จะช่วยให้เราสามารถระบุปัญหาได้ง่ายขึ้น

3 : การแยกส่วนปัญหา (Isolate the Problem): หลังจากระบุปัญหาแล้ว ขั้นตอนต่อไปคือการแยกส่วนปัญหาเพื่อหาตำแหน่งที่ข้อผิดพลาดเกิดขึ้นในโค้ด กระบวนการนี้มักเกี่ยวข้องกับการตรวจสอบโค้ดทีละส่วนและการใช้เครื่องมือช่วยเช่น Debugger

4 : การวิเคราะห์ (Analyze the Problem): เมื่อระบุตำแหน่งที่ปัญหาเกิดขึ้นได้แล้ว เราต้องวิเคราะห์เพื่อหาสาเหตุที่แท้จริงของปัญหา อาจจะต้องพิจารณาเหตุการณ์หรือเงื่อนไขเฉพาะที่ทำให้เกิดข้อผิดพลาด

5 : การแก้ไข (Fix the Problem): หลังจากหาสาเหตุได้แล้ว ขั้นตอนสุดท้ายคือการแก้ไขข้อผิดพลาดในโค้ด ควรทำการแก้ไขอย่างระมัดระวังเพื่อไม่ให้เกิดปัญหาใหม่ตามมา

6 : การทดสอบซ้ำ (Test the Fix): หลังจากแก้ไขแล้ว ควรทำการทดสอบซ้ำเพื่อยืนยันว่าข้อผิดพลาดถูกแก้ไขจริงๆ และได้ส่งผลกระทบต่อส่วนอื่นของซอฟต์แวร์

เครื่องมือในการ Debugging

1 : Debugger: เป็นเครื่องมือหลักในการช่วยตรวจสอบและแก้ไขข้อผิดพลาดในโค้ด โดยมีความสามารถในการหยุดโปรแกรมชั่วคราว (breakpoints) ตรวจสอบค่าของตัวแปร และเดินโปรแกรมทีละบรรทัด (step through)

2 : Logging: เป็นวิธีการบันทึกข้อมูลขณะโปรแกรมทำงาน ซึ่งช่วยในการติดตามปัญหาที่เกิดขึ้น การเพิ่มการ log ในส่วนต่างๆ ของโค้ดสามารถช่วย

การประยุกต์ใช้งานจริงของซอฟต์แวร์หรือระบบที่พัฒนาขึ้นเป็นขั้นตอนที่มีความสำคัญอย่างยิ่งในกระบวนการพัฒนาซอฟต์แวร์ กระบวนการนี้ไม่เพียงแต่เกี่ยวข้องกับการทดสอบและการแก้ไขข้อผิดพลาดเท่านั้น แต่ยังรวมถึงการวางแผนการใช้งาน การติดตั้งและการดูแลรักษาในสภาพแวดล้อมจริง ซึ่งมีความซับซ้อนและต้องการการจัดการที่รอบคอบ

1. การเตรียมความพร้อมสำหรับการประยุกต์ใช้งานจริง

1.1 การวางแผนการใช้งาน (Deployment Planning)

การวางแผนการใช้งานเป็นขั้นตอนสำคัญก่อนที่จะนำซอฟต์แวร์ไปใช้ในสภาพแวดล้อมจริง ต้องพิจารณาหลายประเด็นเพื่อให้การประยุกต์ใช้งานเป็นไปอย่างราบรื่น ประเด็นเหล่านี้ได้แก่:

การเลือกสภาพแวดล้อม (Environment Selection): ต้องเลือกสภาพแวดล้อมที่เหมาะสมในการนำซอฟต์แวร์ไปใช้งาน เช่น เซิร์ฟเวอร์ที่เหมาะสม ระบบปฏิบัติการ และฐานข้อมูล

การทดสอบสภาพแวดล้อม (Environment Testing): ก่อนการใช้งานจริง ควรทดสอบซอฟต์แวร์ในสภาพ

แวดล้อมที่จะใช้งานจริง (Production Environment) เพื่อให้แน่ใจว่าทุกอย่างทำงานได้ตามปกติและไม่มีปัญหาที่ไม่คาดคิด

การจัดการข้อมูล (Data Management): ต้องวางแผนการย้ายข้อมูลจากระบบเก่ามายังระบบใหม่ หรือการตั้งค่าฐานข้อมูลเริ่มต้นอย่างถูกต้อง

1.2 การอบรมผู้ใช้งาน (User Training)

ผู้ใช้งานซอฟต์แวร์เป็นปัจจัยสำคัญที่มีผลต่อความสำเร็จของการประยุกต์ใช้งานจริง ดังนั้นการอบรมและให้ความรู้แก่ผู้ใช้งานในการใช้ซอฟต์แวร์ใหม่จึงเป็นสิ่งที่จำเป็น

การอบรมก่อนการใช้งาน (Pre-Deployment Training): ให้การอบรมแก่ผู้ใช้งานก่อนการใช้งานจริง เพื่อให้พวกเขาคุ้นเคยกับอินเทอร์เฟซและฟังก์ชันการทำงานของซอฟต์แวร์

การสนับสนุนหลังการใช้งาน (Post-Deployment Support): จัดเตรียมทีมสนับสนุนเพื่อช่วยแก้ไขปัญหาหรือคำถามที่เกิดขึ้นหลังการนำซอฟต์แวร์มาใช้ในสภาพแวดล้อมจริง

2. การติดตั้งและการใช้งาน (Deployment and Implementation)

2.1 การติดตั้งซอฟต์แวร์ (Software Installation)

ขั้นตอนการติดตั้งซอฟต์แวร์เป็นการนำซอฟต์แวร์ที่ผ่านการทดสอบแล้วมาติดตั้งในสภาพแวดล้อมจริง กระบวนการนี้อาจเกี่ยวข้องกับการตั้งค่าต่างๆ เช่น การติดตั้งเซิร์ฟเวอร์ การตั้งค่าเครือข่าย และการเชื่อมต่อกับฐานข้อมูล โดยทั่วไปจะต้องดำเนินการดังนี้:

การยืนยันความพร้อมของสภาพแวดล้อม (Environment Readiness): ตรวจสอบให้แน่ใจว่าสภาพแวดล้อมทุกอย่างพร้อมใช้งาน เช่น ฮาร์ดแวร์ ซอฟต์แวร์ และการเชื่อมต่อเครือข่าย

การติดตั้งและกำหนดค่า (Installation and Configuration): ดำเนินการติดตั้งซอฟต์แวร์ตามขั้นตอนที่กำหนด และกำหนดค่าให้สอดคล้องกับความต้องการขององค์กรหรือผู้ใช้งาน

การตรวจสอบการทำงานหลังการติดตั้ง (Post-Installation Testing): หลังจากติดตั้งแล้ว ควรตรวจสอบว่าซอฟต์แวร์ทำงานได้ถูกต้องตามที่คาดหวัง โดยเฉพาะอย่างยิ่งในการเชื่อมต่อกับส่วนประกอบอื่นๆ ของระบบ

2.2 การโยกย้ายข้อมูล (Data Migration)

การโยกย้ายข้อมูลเป็นกระบวนการที่สำคัญในกรณีที่ต้องเปลี่ยนจากระบบเก่ามาสู่ระบบใหม่ ต้องมีการวางแผนและดำเนินการอย่างรอบคอบเพื่อให้ข้อมูลที่สำคัญถูกย้ายมาอย่างสมบูรณ์และถูกต้อง

การวางแผนการโยกย้าย (Migration Planning): วางแผนกระบวนการโยกย้าย โดยพิจารณาประเด็นต่างๆ เช่น ขนาดของข้อมูล ความถูกต้องของข้อมูล และการตรวจสอบหลังการโยกย้าย

การทดสอบการโยกย้าย (Migration Testing): ทดสอบกระบวนการโยกย้ายในสภาพแวดล้อมจำลอง (Staging Environment) เพื่อหาปัญหาที่อาจเกิดขึ้นและทำการแก้ไขก่อนการโยกย้ายจริง

การดำเนินการโยกย้าย (Migration Execution): ดำเนินการโยกย้ายข้อมูลจริงตามแผนที่วางไว้ โดยมีการติดตามและตรวจสอบความถูกต้องของข้อมูลอย่างใกล้ชิด

3. การบำรุงรักษาและการสนับสนุน (Maintenance and Support)

3.1 การบำรุงรักษา (Maintenance)

หลังจากซอฟต์แวร์ถูกนำมาใช้งานในสภาพแวดล้อมจริงแล้ว จำเป็นต้องมีการบำรุงรักษาเพื่อให้ระบบทำงานได้อย่างต่อเนื่องและมีประสิทธิภาพ

การบำรุงรักษาเชิงป้องกัน (Preventive Maintenance): ดำเนินการบำรุงรักษาเพื่อป้องกันปัญหาที่อาจเกิดขึ้น เช่น การอัปเดตซอฟต์แวร์ การสำรองข้อมูล และการตรวจสอบความปลอดภัย

การบำรุงรักษาเชิงแก้ไข (Corrective Maintenance): แก้ไขปัญหาที่เกิดขึ้นในซอฟต์แวร์หรือระบบ เช่น การแก้ไขบั๊ก หรือการปรับปรุงฟังก์ชันการทำงานให้สอดคล้องกับความต้องการที่เปลี่ยนแปลงไป

3.2 การสนับสนุนหลังการใช้งาน (Post-Deployment Support)

การสนับสนุนหลังการใช้งานเป็นสิ่งจำเป็นในการช่วยให้ผู้ใช้งานสามารถใช้ซอฟต์แวร์ได้อย่างราบรื่น

การจัดทีมสนับสนุน (Support Team): จัดเตรียมทีมสนับสนุนที่มีความรู้และทักษะในการแก้ไขปัญหาที่เกิดขึ้น รวมถึงการตอบคำถามจากผู้ใช้งาน

การติดตามปัญหา (Issue Tracking): ใช้เครื่องมือในการติดตามปัญหาที่เกิดขึ้นในซอฟต์แวร์ เพื่อให้สามารถจัดการและแก้ไขได้อย่างรวดเร็ว

3.3 การปรับปรุงและอัปเดตซอฟต์แวร์ (Software Updates and Upgrades)

การปรับปรุงและอัปเดตซอฟต์แวร์เป็นกระบวนการที่ต่อเนื่องในการรักษาประสิทธิภาพและความปลอดภัยของระบบ

การอัปเดตเวอร์ชัน (Version Updates): อัปเดตซอฟต์แวร์เป็นเวอร์ชันใหม่เพื่อเพิ่มฟังก์ชันการทำงานหรือแก้ไขปัญหาด้านความปลอดภัย

การอัปเกรดระบบ (System Upgrades): อัปเกรดระบบเมื่อมีความต้องการเพิ่มประสิทธิภาพหรือเพิ่มความสามารถในการรองรับผู้ใช้งานที่เพิ่มขึ้น

แนวโน้มและอนาคตของ React.js

React.js ยังคงเป็นหนึ่งในไลบรารีที่ได้รับความนิยมมากที่สุดสำหรับการพัฒนาเว็บแอปพลิเคชัน และมีการพัฒนาอย่างต่อเนื่องเพื่อตอบสนองต่อความต้องการที่เปลี่ยนแปลงของวงการเทคโนโลยี โดยในอนาคต มีแนวโน้มที่น่าสนใจหลายประการที่จะส่งผลต่อทิศทางการพัฒนาและการใช้งาน React.js

ทิศทางการพัฒนา React

React.js ถูกออกแบบให้เน้นความเรียบง่ายและประสิทธิภาพในการพัฒนา UI ซึ่งทีมพัฒนา React ของ Facebook ยังคงมุ่งเน้นการปรับปรุงและเพิ่มฟีเจอร์ใหม่ๆ ที่ตอบสนองต่อความต้องการที่เปลี่ยนแปลงอย่างรวดเร็ว โดยเน้นการพัฒนาประสิทธิภาพของการทำงาน (performance) และประสบการณ์ของนักพัฒนา (developer experience)

หนึ่งในทิศทางสำคัญคือการพัฒนาฟีเจอร์ที่ช่วยให้การจัดการสถานะและการเรียกข้อมูลมีความยืดหยุ่นและง่ายยิ่งขึ้น โดยเฉพาะอย่างยิ่งในสภาพแวดล้อมที่มีความซับซ้อน เช่น เว็บแอปพลิเคชันที่ต้องรองรับการใช้งานที่มีจำนวนผู้ใช้สูงหรือแอปพลิเคชันแบบเรียลไทม์

เทรนด์ใหม่ ๆ ในการพัฒนาเว็บแอปพลิเคชัน

การพัฒนาเว็บแอปพลิเคชันมีเทรนด์ใหม่ๆ ที่กำลังเกิดขึ้นซึ่งมีผลกระทบต่อ React.js ได้แก่:

การเรนเดอร์แบบไฮบริด (Hybrid Rendering): การใช้เทคนิคการเรนเดอร์ทั้งฝั่งเซิร์ฟเวอร์และไคลเอนต์ร่วมกัน (Server-Side Rendering และ Client-Side Rendering) เช่น การใช้ `Next.js` ซึ่งช่วยเพิ่มประสิทธิภาพและ SEO ของเว็บแอปพลิเคชัน

ไมโครฟรอนต์เอนด์ (Micro Frontends): การแบ่งแอปพลิเคชันขนาดใหญ่เป็นโมดูลย่อยๆ ที่สามารถพัฒนาและปรับปรุงได้อย่างอิสระ โดย React.js สามารถถูกใช้เป็นส่วนหนึ่งของสถาปัตยกรรมนี้

ฟีเจอร์ใหม่ ๆ ใน React รุ่นล่าสุด

ในเวอร์ชันล่าสุดของ React มีการเพิ่มฟีเจอร์ที่น่าสนใจสองประการที่คาดว่าจะมีบทบาทสำคัญในการพัฒนาเว็บแอปพลิเคชันในอนาคต:

React Concurrent Mode

Concurrent Mode เป็นฟีเจอร์ที่ช่วยให้ React สามารถทำงานหลายอย่างพร้อมกันได้อย่างมีประสิทธิภาพยิ่งขึ้น ฟีเจอร์นี้ช่วยปรับปรุงประสบการณ์การใช้งานโดยการให้แอปพลิเคชันสามารถตอบสนองต่อผู้ใช้ได้ดียิ่งขึ้น แม้ในสถานการณ์ที่มีการประมวลผลข้อมูลขนาดใหญ่ Concurrent Mode ยังช่วยให้การเรนเดอร์ส่วนต่างๆ ของ UI เกิดขึ้นอย่างอิสระ ไม่ต้องรอให้การประมวลผลในส่วนอื่นเสร็จสิ้นก่อน

Suspense for Data Fetching

Suspense for Data Fetching ช่วยให้นักพัฒนาสามารถจัดการการดึงข้อมูล (data fetching) ได้ง่ายขึ้นโดยรวมกระบวนการนี้เข้ากับ UI โดยตรง เมื่อข้อมูลยังไม่พร้อมแสดงผล Suspense จะช่วยให้ UI สามารถแสดงสถานะ "กำลังโหลด" (loading) ได้อย่างราบรื่น ฟีเจอร์นี้ช่วยให้โค้ดที่เกี่ยวข้องกับการจัดการข้อมูลใน React ดูเป็นธรรมชาติมากขึ้นและลดความซับซ้อนในการจัดการสถานะของแอปพลิเคชัน

การปรับตัวของ React ใน Ecosystem ที่เปลี่ยนแปลง

React.js ยังคงรักษาความยืดหยุ่นในการใช้งานร่วมกับเทคโนโลยีและเครื่องมือต่างๆ ในระบบนิเวศ (Ecosystem) ที่เปลี่ยนแปลงอย่างรวดเร็ว

การผสมผสานรวมกับ Frameworks อื่น ๆ เช่น Next.js

Next.js เป็นเฟรมเวิร์กที่สร้างบน React.js และกำลังได้รับความนิยมอย่างมาก เนื่องจากสามารถใช้ในการพัฒนาเว็บแอปพลิเคชันที่เรนเดอร์ทั้งฝั่งเซิร์ฟเวอร์และไคลเอนต์ (SSR และ CSR) ได้อย่างมีประสิทธิภาพ Next.js ยังเพิ่มความสามารถในการจัดการเส้นทาง (routing) และการจัดการข้อมูลที่ซับซ้อนได้ง่ายขึ้น ทำให้เป็นตัวเลือกที่เหมาะสมสำหรับการพัฒนาเว็บแอปพลิเคชันที่ต้องการประสิทธิภาพสูง

การใช้งานร่วมกับ Tools ใหม่ ๆ เช่น Vite, SWC

Vite และ SWC เป็นเครื่องมือที่ได้รับความนิยมมากขึ้นเนื่องจากมีความรวดเร็วในการสร้างและปรับปรุงโค้ด (build and transpile code) โดย Vite เป็นเครื่องมือในการพัฒนาแอปพลิเคชันที่มีความรวดเร็วเนื่องจากใช้เทคนิคการทำ HMR (Hot Module Replacement) ที่ทันสมัย ส่วน SWC เป็นเครื่องมือในการแปลงโค้ด (transpiler) ที่ถูกเขียนขึ้นด้วย Rust ซึ่งมีประสิทธิภาพสูงกว่า Babel ในการแปลงโค้ด JavaScript และ TypeScript React.js สามารถใช้งานร่วมกับทั้งสองเครื่องมือนี้ได้เป็นอย่างดี ทำให้นักพัฒนาสามารถสร้างเว็บแอปพลิเคชันที่มีประสิทธิภาพและพัฒนารวดเร็วยิ่งขึ้น

ฐานข้อมูล (Database) คือระบบที่ใช้ในการจัดเก็บ, จัดการ, และดึงข้อมูลที่มีความสัมพันธ์กันอย่างเป็นระเบียบ สามารถรองรับการค้นหา, การปรับปรุง, และการจัดการข้อมูลจำนวนมากได้อย่างมีประสิทธิภาพ ฐานข้อมูลเป็นหัวใจสำคัญของระบบสารสนเทศในองค์กรต่างๆ ไม่ว่าจะเป็นการจัดการข้อมูลลูกค้า, ข้อมูลการขาย, หรือข้อมูลภายในองค์กร

ประเภทของฐานข้อมูล

ฐานข้อมูลเชิงสัมพันธ์ (Relational Database): ฐานข้อมูลชนิดนี้เก็บข้อมูลในรูปแบบตาราง (Table) ซึ่งแต่ละตารางมีความสัมพันธ์กันผ่านทางคีย์ ตัวอย่างเช่น MySQL, PostgreSQL, Oracle Database

ฐานข้อมูลไม่เชิงสัมพันธ์ (NoSQL Database): ฐานข้อมูลที่ออกแบบมาเพื่อจัดการข้อมูลที่ไม่ต้องการโครงสร้างตายตัว เช่น ข้อมูลที่เป็นเอกสาร, กราฟ, คีย์-แวลู เป็นต้น ตัวอย่างเช่น MongoDB, Cassandra, Redis

ฐานข้อมูลเชิงกราฟ (Graph Database): ฐานข้อมูลที่จัดเก็บข้อมูลในรูปแบบกราฟ เน้นการจัดการข้อมูลที่มีความสัมพันธ์ซับซ้อนระหว่างข้อมูล ตัวอย่างเช่น Neo4j

ฐานข้อมูลเชิงวัตถุ (Object-Oriented Database): ฐานข้อมูลที่เก็บข้อมูลในรูปแบบวัตถุ (Object) โดยวัตถุจะมีลักษณะคล้ายกับวัตถุในภาษาการเขียนโปรแกรมเชิงวัตถุ

องค์ประกอบของฐานข้อมูล

ข้อมูล (Data): ข้อมูลที่ถูกเก็บในฐานข้อมูล เช่น ข้อมูลลูกค้า, ข้อมูลสินค้า เป็นต้น

DBMS (Database Management System): ซอฟต์แวร์ที่ใช้ในการจัดการฐานข้อมูล ทำให้ผู้ใช้สามารถสร้าง, ดึงข้อมูล, อัปเดต, และลบข้อมูลได้

ตาราง (Table): โครงสร้างที่ใช้ในการจัดเก็บข้อมูลในฐานข้อมูลเชิงสัมพันธ์ ประกอบด้วยแถว (Rows) และคอลัมน์ (Columns)

คีย์ (Key): ใช้ในการระบุแถวในตารางหรือความสัมพันธ์ระหว่างตาราง เช่น Primary Key, Foreign Key

ข้อดีของการใช้ฐานข้อมูล

การจัดการข้อมูลได้อย่างมีประสิทธิภาพ: ทำให้การจัดการข้อมูลขนาดใหญ่เป็นไปอย่างมีระเบียบและมีประสิทธิภาพ

ความปลอดภัยของข้อมูล: ฐานข้อมูลสามารถกำหนดสิทธิ์การเข้าถึงข้อมูลได้ ทำให้ข้อมูลมีความปลอดภัย

การสำรองและกู้คืนข้อมูล: DBMS สามารถสำรองและกู้คืนข้อมูลได้ในกรณีที่เกิดปัญหาขึ้น

การสนับสนุนการทำงานร่วมกัน: หลายคนสามารถเข้าถึงและใช้งานฐานข้อมูลเดียวกันได้พร้อมกัน
การใช้งานในปัจจุบันฐานข้อมูลเป็นส่วนหนึ่งที่สำคัญในแอปพลิเคชันและระบบสารสนเทศต่างๆ ไม่ว่าจะเป็นเว็บไซต์, ระบบอีคอมเมิร์ซ, ระบบการเงิน และอื่นๆ อีกมากมาย ซึ่งจะช่วยให้ข้อมูลถูกจัดเก็บและจัดการอย่างเป็นระบบเพื่อการวิเคราะห์และการตัดสินใจที่ดีขึ้น

งานวิจัยที่เกี่ยวข้องของระบบขอตำแหน่งทางวิชาการ

การศึกษาเอกสารและงานวิจัยที่เกี่ยวข้อง ผู้วิจัยได้รวบรวมแนวคิด ทฤษฎีและงานวิจัยอื่น ๆ ที่เกี่ยวข้องเพื่อใช้เป็นแนวทางในการดำเนินการวิจัยดังนี้

1. แนวคิด ทฤษฎี การพัฒนาสายวิชาการ
2. สภาพปัญหาในการขอตำแหน่งทางวิชาการของมหาวิทยาลัยกาฬสินธุ์
3. หลักเกณฑ์และวิธีการพิจารณาแต่งตั้งบุคคลในการดำรงตำแหน่งทางวิชาการ
4. งานวิจัยที่เกี่ยวข้อง
5. กรอบแนวคิดในการวิจัย

2.1 แนวคิดทฤษฎีการพัฒนาสายวิชาการ

การพัฒนาสายความก้าวหน้าในงานหรือการพัฒนาเส้นทางสายวิชาการ เป็นแนวความคิดที่ ช่วยพัฒนาพนักงานในองค์กรให้ทำงานอย่างมีประสิทธิภาพ อีกทั้งยังมีโอกาสก้าวหน้าตามศักยภาพและความพร้อมของพนักงานแต่ละคน ดังนั้น การพัฒนาสายความก้าวหน้าในงานหรือการพัฒนาเส้นทางสาย วิชาการ จึงเป็นเครื่องมือทางการพัฒนาทรัพยากรมนุษย์ที่ก่อให้เกิดประโยชน์ทั้งกับพนักงานและองค์กร โดยผลของการพัฒนาอาชีพจะมี ประโยชน์คือ ทำให้องค์กรเดินไปสู่ความสำเร็จโดยมีวัตถุประสงค์ร่วมกันระหว่างตัวพนักงานและองค์กร

2.1.1 ประโยชน์ของการพัฒนาสายวิชาการ

การพัฒนาสายวิชาการ ของบุคลากรมีความสำคัญและก่อให้เกิดประโยชน์แก่องค์กร หลาย ประการ ดังต่อไปนี้คือ

1) ช่วยรับประกันว่าบุคลากรที่มีความสามารถจะอยู่กับองค์กรต่อไปบุคลากรที่มี ความสามารถ และประสบการณ์ย่อมเป็นที่ต้องการขององค์กรทุกแห่งและบุคคลเหล่านี้สามารถที่จะหา งาน ใหม่ได้อย่างรวดเร็วและง่ายดาย หากพวกเขาารู้สึกว่าองค์กรไม่ได้ให้ความสำคัญแก่พวกเขา และไม่ได้ รับนโยบายหรือมาตรการใด ๆ ที่จะส่งเสริมความก้าวหน้าในอาชีพแก่พวกเขาแล้ว โอกาสที่พวกเขาจะ ลาออกจากงานก็มีสูงยิ่งขึ้น ดังนั้น โครงการด้านการพัฒนาอาชีพจะเป็นสิ่ง เหนียวแน่นให้บุคลากรที่มี ความสามารถเหล่านี้ออกจากองค์กรไป

2) ช่วยให้องค์กรสามารถดึงบุคลากรที่มีความสามารถสูงเข้ามาทำงาน ได้มากขึ้น บุคคลที่มีความ สามารถสูงมักจะพิจารณาเลือกทำงานกับองค์กรซึ่งให้ความสำคัญต่ออนาคตและความ ก้าวหน้าของบุคลากร ดังนั้น หากองค์กรใดที่มีโครงการด้านการพัฒนาอาชีพ ก็ย่อมจะมีความ ได้เปรียบในการดึงดูดบุคลากรเหล่านั้นให้เข้ามาทำงานกับตนเอง

3) ช่วยสร้างภาพลักษณ์ที่ดีให้กับองค์กร องค์กรที่ให้ความสำคัญและความสนใจแก่ อนาคต และความก้าวหน้าของบุคลากร ย่อมจะทำให้ทั้งบุคลากรภายใน องค์กรและบุคลากรภายนอก องค์กรนั้นด้วยความรู้สึกที่ดี ซึ่งจะช่วยส่งเสริมภาพลักษณ์ และชื่อเสียงให้องค์กรเป็นอย่างดี

4) ช่วยให้บุคลากรมีการพัฒนาตนเองและลดความล้าสมัยในการทำงานตำแหน่งหนึ่ง ๆ อันอาจ จะทำให้บุคลากรล้าหลังต่อความเปลี่ยนแปลงต่าง ๆ ดังนั้น การพัฒนาอาชีพจะช่วยให้บุคลากร มีการพัฒนาตนเองอยู่ตลอดเวลา ทั้งนี้เพราะบุคลากรเหล่านั้นจำเป็นต้องเพิ่มพูนความรู้และ ความสามารถ ต่าง ๆ ที่จำเป็นก่อนที่จะสามารถก้าวขึ้นสู่ตำแหน่งอื่นๆ เป็นกระบวนการวางแผน อันเกี่ยวกับจุดมุ่งหมายในชีวิตการ ทำงานหรือเป้าหมายในอนาคตของตัวพนักงานเอง ซึ่งการที่ จะวางแผนเกี่ยวกับสายอาชีพของพนักงานนั้น ขึ้นอยู่กับปัจจัยดังต่อไปนี้คือ

- ความสนใจในด้านต่าง ๆ ของพนักงาน
- ขีดความสามารถทั้งเฉพาะตัวและการทำงานเป็นทีมของพนักงาน
- ศึกษาและสำรวจตำแหน่งงานต่าง ๆ ตลอดจนโอกาสที่จะเจริญก้าวหน้าในองค์กร
- กำหนดจุดมุ่งหมายที่พนักงานและองค์กรปรารถนาทั้งในระยะสั้น ระยะปานกลางและ ระยะยาว
- ฝึกอบรมและแนวทางการพัฒนาสำหรับพนักงานนั้นๆ
- ปฏิบัติตามแผนที่วางไว้อย่างเคร่งครัด

2.2 สภาพปัญหาในการขอกำหนดตำแหน่งทางวิชาการของมหาวิทยาลัย

1. จำนวนอาจารย์ที่มีคุณสมบัติยื่นขอตำแหน่งทางวิชาการ จำนวนผู้ที่มีคุณสมบัติฯ แต่ขอ ขำ

จำนวนผู้มีคุณสมบัติแล้วมีผลงานตีพิมพ์แล้ว แต่ไม่ได้ยื่นขอ

2. อัตราการเข้าสู่ตำแหน่งทางวิชาการของพนักงานสายวิชาการที่

3. แนวทางในการส่งเสริมและสนับสนุนบุคลากรสายวิชาการเพื่อเข้าสู่ตำแหน่งทางวิชาการ ใน

ระดับที่สูงขึ้น ของบุคลากรสายวิชาการ มหาวิทยาลัยเทศบาลลักษณะงานบริหารงาน บุคคล มี

บทบาทในการติดตาม ส่งเสริมดำเนินการในกระบวนการเสนอขอกำหนดตำแหน่งทางวิชาการ

ในขณะเดียวกันพบว่า การเข้าสู่ตำแหน่งทางวิชาการช้าและคงที่ในบางระดับ

การแก้ไขปัญหา

- กลยุทธ์ในการสนับสนุนให้อาจารย์มหาวิทยาลัยไทยมีการผลิตผลงานทางวิชาการมากขึ้น มี ตำแหน่งทางวิชาการที่สูงขึ้น ซึ่งเป็นส่วนหนึ่งในการพัฒนางานวิชาการของประเทศไทยนั้นสามารถกระทำ ได้ โดยการกำหนดกลยุทธ์ทั้งในระดับบุคคลและระดับองค์กร ดังนี้

1. กลยุทธ์ในระดับบุคคล

- 1.1 ในกรณีของประเทศไทย อาจารย์ควรผลิตผลงานวิจัยที่หลากหลาย เพราะเราไม่ทราบว่า ผู้อ่านงานชิ้นนั้น (Reader) มีรสนิยมต่องานวิชาการ (Preference) อย่างไร อาจารย์ควรมีงานวิจัยที่ดีพิมพ์ ทั้งในวารสารต่างประเทศ วารสารวิชาการในประเทศ
- 1.2 อาจารย์อาจเริ่มพยายามสร้างผลงานวิจัยของตัวเองคนเดียว (อาจารย์รุ่นใหม่มักเป็น CoAuthor) โดยขอจากแหล่งทุกที่ให้ในการผลิตผลงานสำหรับตีพิมพ์ในวารสารต่างประเทศ เช่น สกว. วช. หรือ สำนักวิจัยในมหาวิทยาลัยเอง
- 1.3 อาจารย์ต้องมีการวางแผนการขอตำแหน่งที่ชัดเจน (ตั้งแต่ระดับอาจารย์ถึงศาสตราจารย์) กำหนดผลงานที่ตัวเองจะใช้ขอให้ชัดเจน โดยวางแผนจะขอตำแหน่งทางวิชาการที่ตั้งเป้าถึงศาสตราจารย์ และกำหนดว่าจะเอาผลงานอะไรมาเพื่อขอตำแหน่งอะไรไม่จำเป็นต้องเอางานทุกชิ้นมาขอในตำแหน่งหนึ่ง ตำแหน่งใดทั้งหมด
- 1.4 อาจารย์ควรมีการรวบรวมผลงานที่ได้ตีพิมพ์ทุกชิ้นเป็นแฟ้มข้อมูลโดยมีการขอลายเซ็นการมีส่วนร่วมจากนักวิจัยท่านอื่น ๆ ทันทีเมื่องานเสร็จ
- 1.5 อาจารย์ควรเน้นทำงานวิจัยที่มีการพิมพ์เผยแพร่หรือสามารถนำมาส่งตีพิมพ์ในวารสารได้ เป็นความสำคัญโดยพยายามแบ่งสัดส่วนกับงานวิจัยประเภทที่ปรึกษา (ซึ่งเป็นงานวิจัยที่อาจเป็นประโยชน์ ต่อสังคมก็จริง แต่มักเป็นงานวิจัยที่อาจารย์ไม่มีอิสระทางความคิดมากเท่าที่ควร)

- 1.6 อาจารย์ควรเห็นความสำคัญของการขอตำแหน่งทางวิชาการ เนื่องจากตำแหน่งทางวิชาการ นำมาสู่เงินประจำตำแหน่งที่สูงขึ้น การได้รับการยอมรับในวงการวิชาการ (Reputation) โอกาสในการรับงาน ที่ปรึกษาในอัตราค่าจ้าง (Rate) ที่สูงขึ้น การเทียบระดับในข้าราชการที่สูงขึ้น และการได้รับโอกาสในการเข้า ไปสู่การเป็นที่ปรึกษาและการบริหารที่สูงขึ้น เป็นต้น
- 1.7 ในกรณีของอาจารย์ที่มีผลงานวิชาการมากพอ แต่ยังไม่ถึงเวลาในการขออาจารย์ท่านนั้น อาจพิจารณาที่จะขอตำแหน่งทางวิชาการในกรณีพิเศษ เพราะเป็นการลองที่มีโอกาสที่จะได้หรืออย่างน้อยก็ เสมอตัว

2. กลยุทธ์ในระดับคณะ

- 2.1 ภายในคณะเองอาจใช้ระบบของการสร้างแรงจูงใจหรือข้อบังคับเป็นเครื่องมือที่สำคัญ เช่น คณะอาจกำหนดการจ่ายค่าตอบแทนการสอน (ภาคพิเศษ) ที่แปรผันเป็นขั้นบันไดตามตำแหน่งทางวิชาการ (เช่น อาจารย์-1500, ผศ.- 2000, รศ-3000, ศ-4,000 ต่อชั่วโมง) เพื่อเป็นการสร้างแรงจูงใจทางการเงิน
- 2.2 คณะอาจชดเชยกับค่าเสียโอกาสและต้นทุนที่สูงขึ้นของอาจารย์ที่มีตำแหน่งทางวิชาการสูง โดยการลดชั่วโมงการสอนประจำของอาจารย์ท่านนั้นลง เพื่อให้อาจารย์จะได้มีเวลาในการผลิตผลงานทาง 11 วิชาการมากขึ้น แต่หลายที่กลับบังคับให้อาจารย์ที่มีตำแหน่งทางวิชาการต้องสอนหนังสือมากกว่าอาจารย์ที่ไม่มี ซึ่งตรงนี้จะยิ่งทำให้อาจารย์ไม่ยากที่จะขอตำแหน่งทางวิชาการ คณะควรจัดลำดับของการให้อาจารย์ ทำงานบริหารและการสอนที่แปรผันตามตำแหน่งโดยอาจารย์ที่ไม่มีตำแหน่งทางวิชาการควรให้น้ำหนักผลงาน วิชาการ ในขณะที่มีอาจารย์ที่มีตำแหน่งทางวิชาการสามารถเข้ามาทำงานบริหารของคณะได้ซึ่งกฎนี้ใช้ใน หลายมหาวิทยาลัยในยุโรป

- 2.3 คณะควรสนับสนุนให้อาจารย์ไปเสนอผลงานในต่างประเทศ โดยควรกำหนดให้อาจารย์ต้อง ไปต่างประเทศเพื่อนำเสนอผลงาน (Present) เท่านั้น โดยคณะต้องกำหนดให้ผลงานที่นำไปเสนอจะต้องมี การตีพิมพ์ในวารสารวิชาการต่างประเทศภายใน 12 ปี ไม่ใช่แค่ (Proceeding) และอาจกำหนดเป็นบัญชีดำ (Black List) สำหรับอาจารย์ที่ไปนำเสนอแต่กลับไม่นำงานนั้นไปตีพิมพ์

3.กลยุทธ์ในระดับสถาบัน/กระทรวง มหาวิทยาลัย/กระทรวงควรใช้ระบบของการสร้างแรงจูงใจดังนี้

- 3.1 กำหนดให้มหาวิทยาลัยต้องจ่ายค่าตอบแทนการสอน (ภาคปกติ) แปรผันตามตำแหน่งทาง วิชาการบางมหาวิทยาลัยเริ่มมีการให้เงินรางวัลผลงานที่มีการตีพิมพ์แต่เป็นนโยบายดังกล่าวอาจมีผลกระทบ ระยะสั้นเท่านั้น
- 3.2 มหาวิทยาลัย/กระทรวงอาจกำหนดระยะเวลาในการขอตำแหน่ง (เพื่อที่จะตัดสินใจในการ ต่ออายุการจ้างงาน) เช่น ขอตำแหน่ง ผศ. ใน 3 ปี (1+2 ปี) และขอตำแหน่ง รศ. ภายใน (5-10 ปี) ดังนั้น อาจารย์ที่เริ่มงานภายใน 108 ปีจะมีตำแหน่ง รศ. (โดยทั้งสองกรณีมีการนำมาปฏิบัติใช้แล้วในบาง มหาวิทยาลัย) หรืออาจใช้ระบบของ Tenure เหมือนกับต่างประเทศ
- 3.3 ศูนย์บริการวิชาการของมหาวิทยาลัยควรเพิ่มบทบาทในการเป็น Resource Center เช่น การตีพิมพ์งานวิจัยที่ผ่านศูนย์เป็นหนังสือ (และขอ ISBN number) พร้อมทั้งจัดจำหน่าย
- 3.4 มหาวิทยาลัย/ กระทรวงควรลดขั้นตอนและระยะเวลาในการพิจารณาบทความของผู้อ่าน (Reader) สำหรับการพิจารณาการขอตำแหน่งทางวิชาการ เช่นกำหนดมาตรการติดตาม/ลงโทษผู้อ่านที่ใช้ เวลามากกว่ากำหนดเช่น นอกจากติด Black-List แล้ว อาจไม่จ่ายเงินถ้าใช้เวลาเกิน ผู้ขอเสียค่าเสียโอกาส มากในการรอคอยโดยอาจใช้ระบบของคะแนนมากกว่าระบบของผู้อ่าน เพื่อความเป็นธรรมแก่อาจารย์ผู้ขอ และเพื่อเป็นลดขั้นตอนใน ต่างประเทศอาจใช้ระบบคะแนนโดยให้คะแนนสูงสำหรับงานวิชาการที่ตีพิมพ์ใน วารสารชั้นนำในต่างประเทศ หรืองานที่มีการตีพิมพ์หนังสือโดยไม่จำเป็นต้องมีผู้อ่านเพื่ออ่านงานดังกล่าว เพราะเป็นงานได้มีการผ่านกรรมการ (Referee) ในต่างประเทศแล้ว
- 3.5 มหาวิทยาลัย/ กระทรวงควรทำการสำรวจหรือวิจัยเกี่ยวกับเหตุผลของการไม่ขอตำแหน่ง ของอาจารย์เพื่อที่จะสามารถดำเนินกลยุทธ์ได้ตามลำดับที่เหมาะสม

2.3 หลักเกณฑ์และวิธีการพิจารณาแต่งตั้งบุคคลในการดำรงตำแหน่งทางวิชาการ มหาวิทยาลัย

ว่าด้วยหลักเกณฑ์และวิธีการแต่งตั้งผู้ช่วย ศาสตราจารย์ รองศาสตราจารย์ และศาสตราจารย์ ได้กำหนดหลักเกณฑ์เพื่อใช้พิจารณาการ เสนอขอกำหนดตำแหน่งทางวิชาการ โดยสรุป ดังนี้

1. ผู้ที่จะได้รับการพิจารณาแต่งตั้งให้ดำรงตำแหน่งผู้ช่วยศาสตราจารย์รองศาสตราจารย์หรือศาสตราจารย์ จะต้องมีการปฏิบัติงานตามลักษณะงานซึ่งมีปริมาณและคุณภาพ ตามที่มหาวิทยาลัย กำหนด และมีคุณสมบัติและผลงานทางวิชาการ ดังนี้

- (1) มีชั่วโมงสอนประจำรายวิชาตามที่ได้รับมอบหมาย โดยเฉลี่ยไม่น้อยกว่า 6 หน่วยกิต ต่อภาคการศึกษา หรือมีชั่วโมงสอนประจำรายวิชาและควบคุมการปฏิบัติงานของนักศึกษาซึ่งมีปริมาณงาน เทียบค่าได้ไม่น้อยกว่า 18 ชั่วโมงต่อสัปดาห์ต่อภาคการศึกษา ทั้งนี้ ต้องผ่านการประเมินคุณภาพและ ประสิทธิภาพการสอน โดยมีผลการประเมินไม่ต่ำกว่าระดับดีมาก
- (2) ควบคุมตรวจสอบการวิจัย ค้นคว้าวิจัย และหรือเป็นที่ปรึกษานักศึกษา ทั้งด้านวิชาการ และด้านกิจการนักศึกษาตามที่ได้รับมอบหมายไม่น้อยกว่า 5 ชั่วโมงต่อสัปดาห์ต่อภาคการศึกษา
- (3) มีผลงานการให้บริการทางวิชาการและหรือช่วยงานบริหารตามที่ได้รับมอบหมาย 2. ต้องเป็นผู้มีคุณสมบัติตรงตามคุณสมบัติเฉพาะสำหรับตำแหน่งตามมาตรฐานกำหนด ตำแหน่งที่มหาวิทยาลัยกำหนด คือ

2.1 ตำแหน่งผู้ช่วยศาสตราจารย์

- (1) อาจารย์ปริญญาโทหรือเทียบเท่าต้องดำรงตำแหน่งอาจารย์และปฏิบัติหน้าที่ในตำแหน่งดังกล่าวมาแล้วไม่น้อยกว่า 5 ปี หรือ
- (2) อาจารย์ปริญญาเอกหรือเทียบเท่าต้องดำรงตำแหน่งอาจารย์และปฏิบัติหน้าที่ในตำแหน่งดังกล่าวมาแล้วไม่น้อยกว่า 2 ปี

2.2 ตำแหน่งรองศาสตราจารย์

- (1) เป็นผู้ที่มีคุณวุฒิเช่นเดียวกับผู้ดำรงตำแหน่งอาจารย์
- (2) ได้ดำรงตำแหน่งผู้ช่วยศาสตราจารย์มาแล้วไม่น้อยกว่า 3 ปี

2.3 ตำแหน่งศาสตราจารย์ ต้องเป็นผู้มีคุณวุฒิและระยะเวลาการดำรงตำแหน่ง ดังนี้

- (1) เป็นผู้ที่มีคุณวุฒิเช่นเดียวกับผู้ดำรงตำแหน่งอาจารย์ และ
- (2) ได้ดำรงตำแหน่งรองศาสตราจารย์มาแล้วไม่น้อยกว่า 2 ปี

3. ผลงานทางวิชาการ

- 3.1 ตำแหน่งผู้ช่วยศาสตราจารย์ต้องเป็นผู้มีผลงานทางวิชาการตามเกณฑ์ชุดใดชุดหนึ่ง ดังต่อไปนี้ มี 2 เกณฑ์ ได้แก่ เกณฑ์ชุดที่1(เสนอเอกสารประมวลสาระรายวิชาและงานวิจัยหรือผลงานในลักษณะอื่น ๆ) เกณฑ์ชุดที่2 (เสนอผลงานวิจัยอย่างเดียว) ผลงานวิจัยไม่น้อยกว่า 2 เรื่องเต็ม คุณภาพดี โดยเรื่องที่เป็นผู้วิจัยหลักต้องดีมาก
- 3.2 ตำแหน่งรองศาสตราจารย์ต้องเป็นผู้มีผลงานทางวิชาการตามเกณฑ์ชุดใดชุดหนึ่งดังต่อไปนี้ มี 2 เกณฑ์ ได้แก่ เกณฑ์ชุดที่1(เสนอเอกสารประมวลสาระรายวิชาและงานวิจัยหรือผลงานในลักษณะอื่น ๆ) คุณภาพดีมาก เกณฑ์ชุดที่2 (เสนอผลงานวิจัยอย่างเดียว) มีผลงานวิจัยไม่น้อยกว่า 2 เรื่องเต็ม คุณภาพดี มากโดยเรื่องที่เป็นผู้วิจัยหลักต้องดีเด่น
- 3.3 ตำแหน่งศาสตราจารย์ต้องเป็นผู้มีผลงานทางวิชาการตามเกณฑ์ชุดใดชุดหนึ่งดังต่อไปนี้ มี 2 เกณฑ์ ได้แก่ เกณฑ์ชุดที่1เสนอตำราหรือหนังสือและมีผลงานวิจัยไม่น้อยกว่า 1 เรื่องเต็ม คุณภาพดีมาก เกณฑ์ชุดที่2 เสนอตำนหรือหนังสือหรือผลงานวิจัยไม่น้อยกว่า 1 เรื่องเต็ม คุณภาพดีเด่น ปริมาณเอกสารประมวลสาระรายวิชา ผลงานวิจัย ตำรา หนังสือ ให้เป็นไปตามที่ มหาวิทยาลัยกำหนดส่วนการจำแนกระดับคุณภาพผลงานทางวิชาการให้เป็นไปตามที่มหาวิทยาลัย กำหนด มีรายละเอียดในเอกสารคำจำกัดความของผลงานทางวิชาการและเกณฑ์คุณภาพตามเกณฑ์ใน ข้อบังคับซึ่งมหาวิทยาลัยมีการกำหนดปริมาณผลงานที่เกี่ยวข้องในการเสนอขอตำแหน่งทาง วิชาการไว้โดยสรุป ดังนี้ ผลงานวิจัย 1 เรื่องเต็ม ผู้ช่วยศาสตราจารย์ หมายถึง ผลงานวิจัย 1 เรื่อง ที่ผู้ขอเป็นผู้แต่งเพียงผู้เดียว (100%) หรือ ในกรณีที่ผลงานวิจัยมีผู้แต่งร่วม ผู้ขออาจนำเสนอมากกว่า 1 เรื่องก็ได้ โดยรวมสัดส่วนความเป็นเจ้าของ แล้วต้องเป็นเจ้าของผลงานรวมไม่น้อยกว่า 100% รองศาสตราจารย์ หมายถึง ผลงานวิจัย 1 เรื่อง ที่ผู้ขอเป็นผู้แต่งเพียงผู้เดียว (100%) หรือ ในกรณีที่ผลงานวิจัยมีผู้แต่งร่วม ผู้ขออาจนำเสนอมากกว่า 1 เรื่องก็ได้ โดยรวมสัดส่วนความเป็นเจ้าของ แล้วต้องเป็นเจ้าของผลงานรวมไม่น้อยกว่า 100% และต้องเป็นผู้วิจัยหลักอย่างน้อย 1 เรื่อง ศาสตราจารย์หมายถึง ผลงานวิจัย 1 เรื่อง ที่ผู้ขอเป็นผู้แต่งเพียงผู้เดียว (100%) หรือ ใน กรณีที่ผลงานวิจัยมีผู้แต่งร่วม ผู้ขออาจนำเสนอมากกว่า 1 เรื่องก็ได้ โดยรวมสัดส่วนความเป็นเจ้าของแล้ว ต้องเป็นเจ้าของผลงานรวมไม่น้อยกว่า 100% และต้องเป็นผู้วิจัยหลักทุกเรื่อง เอกสารประมวลสาระรายวิชา 1 รายวิชา หมายถึง ผลงานที่มีปริมาณเทียบเท่าการสอนไม่ น้อยกว่า 2 หน่วยกิต/ทวิภาค หรือ 3 หน่วยกิต/ไตรภาค ในระดับปริญญาตรี หรือเท่ากับ 1หน่วยกิต

ผู้วิจัยหลัก

หมายถึง ผู้วิจัยที่มีสัดส่วนร้อยละสูงสุดในการมีส่วนร่วมในผลงานวิจัยนั้น ๆ หรือ เป็นผู้วิจัยหรือผู้ประพันธ์ ซึ่งเป็นผู้ติดต่อกับสำนักพิมพ์ (Corresponding Author) สัดส่วนให้เป็นไป ตามที่มหาวิทยาลัยกำหนด กลไกและขั้นตอนการดำเนินการพิจารณา ดังนี้

1. อาจารย์ประจำผู้ประสงค์จะเสนอขอกำหนดตำแหน่งทางวิชาการหรือผู้บังคับบัญชาชั้นต้นระดับไม่ต่ำกว่าหัวหน้าสาขาวิชาการออกข้อมูลตามแบบที่มหาวิทยาลัยกำหนด แล้วนำเสนอพร้อมด้วย ผลงานทางวิชาการต่อคณบดีของสำนักวิชาที่สาขาวิชานั้นสังกัด เพื่อเสนอคณะกรรมการประจำ สำนักวิชาพิจารณาก่อน
2. คณะกรรมการประจำสำนักวิชาพิจารณาก่อนเบื้องต้นว่าอาจารย์ประจำผู้นั้นมี คุณสมบัติตามหลักเกณฑ์ที่มหาวิทยาลัยกำหนดหรือไม่ แล้วนำเสนอสภาวิชาการ
3. สภาวิชาการแต่งตั้งคณะกรรมการพิจารณาดำเนินทางวิชาการ ซึ่งประกอบด้วย ผู้ทรงคุณวุฒิในสาขาวิชาต่าง ๆ ที่มีการสอนในมหาวิทยาลัย จำนวนไม่เกิน 15 คน เป็นกรรมการ โดยประธานกรรมการให้แต่งตั้งจากกรรมการสภามหาวิทยาลัยประเภทผู้ทรงคุณวุฒิที่มีตำแหน่งทางวิชาการ ไม่ต่ำกว่าระดับศาสตราจารย์รองอธิการบดีหนึ่งคนที่กำกับดูแลงานตำแหน่งทางวิชาการ เป็นเลขานุการ และผู้ช่วยอธิการบดีหนึ่งคนที่อธิการบดีมอบหมายเป็นผู้ช่วยเลขานุการ และหัวหน้าส่วนการเจ้าหน้าที่ เป็นผู้ช่วยเลขานุการ ทั้งนี้ ให้คณะกรรมการพิจารณาดำเนินทางวิชาการมีหน้าที่พิจารณาผลงานทางวิชาการ แต่งตั้งผู้ทรงคุณวุฒิเป็นผู้พิจารณาผลงานทางวิชาการ ตลอดจนหน้าที่อื่น ๆ ที่เกี่ยวข้องและตามที่สภา วิชาการมอบหมาย โดยให้รายงานผลต่อสภาวิชาการ
4. คณะกรรมการพิจารณาดำเนินทางวิชาการพิจารณาแต่งตั้งผู้ทรงคุณวุฒิที่ดำรงตำแหน่งทางวิชาการไม่ต่ำกว่าตำแหน่งที่จะเสนอขอกำหนดและอยู่ในสาขาวิชาเดียวกันหรือสาขาวิชาที่เกี่ยวข้อง จำนวนไม่น้อยกว่า 3 คน โดยต้องเป็นผู้ทรงคุณวุฒิจากภายนอกมหาวิทยาลัยทั้งสิ้น เป็นผู้พิจารณาผลงาน ทางวิชาการ ทั้งนี้ ให้ใช้คะแนนเสียงไม่น้อยกว่ากึ่งหนึ่งในการตัดสิน
5. คณะกรรมการพิจารณาดำเนินทางวิชาการเห็นชอบผลการประเมินคุณภาพงานทาง วิชาการซึ่งผู้ทรงคุณวุฒิประเมินมาแล้วให้นำเสนอสภาวิชาการพิจารณาให้ความเห็นชอบเมื่อสภาวิชาการพิจารณาให้ความเห็นชอบแล้วให้นำเสนอคณะกรรมการบริหารงานบุคคลมหาวิทยาลัยเทคโนโลยี สุรนารีพิจารณาต่อไป 15 6. คณะกรรมการบริหารงานบุคคลมหาวิทยาลัยเทคโนโลยีสุรนารีพิจารณาอนุมัติการขอ กำหนด ตำแหน่งผู้ช่วยศาสตราจารย์และรองศาสตราจารย์ตามข้อเสนอแนะของสภาวิชาการแล้วให้อธิการบดีเป็น ผู้ออกคำสั่งแต่งตั้งให้ดำรงตำแหน่ง 7. สภามหาวิทยาลัยเป็นผู้พิจารณาอนุมัติการขอ กำหนดตำแหน่งศาสตราจารย์ตามข้อเสนอแนะ ของคณะกรรมการบริหารงานบุคคลมหาวิทยาลัยเทคโนโลยีสุรนารี แล้วให้นายกสภามหาวิทยาลัยเสนอ ขอให้รัฐมนตรีว่าการกระทรวงฯ นำความกราบบังคมทูลเพื่อทรงพระกรุณาโปรดเกล้าฯ แต่งตั้งต่อไป

2.4 งานวิจัยที่เกี่ยวข้อง

จากการทบทวนงานวิจัยที่เกี่ยวข้องกับปัจจัยที่ส่งผลกระทบต่อ การเข้าสู่ตำแหน่งทางวิชาการ ที่สูงขึ้น ของคณาจารย์พบการศึกษาที่ผ่านมา ศึกษาเรื่อง ปัจจัยองค์กรที่ส่งผลต่อเจตคติและความมุ่งมั่นในการ ผลิตผลงานทางวิชาการของอาจารย์สถาบันอุดมศึกษาเอกชน ผลการวิจัย พบว่า

1. ปัจจัยองค์กรที่ส่งผลต่อเจตคติและความมุ่งมั่นในการผลิตผลงานทางวิชาการของอาจารย์ใน สถาบัน อุดมศึกษาเอกชน โดย ภาพรวมอยู่ในระดับ มากโดยด้านที่มีค่าเฉลี่ยมากที่สุดคือด้านบรรยากาศ องค์กร รองลงมา คือ ความ เป็นผู้นำทางวิชาการของผู้บริหาร และด้านการส่งเสริมสนับสนุนของสถาบัน ตามลำดับ

2. ระดับเจตคติ และความมุ่งมั่นในการผลิตผลงานทางวิชาการของอาจารย์ในสถาบันอุดมศึกษาเอกชน โดยภาพรวมอยู่ใน ระดับมาก โดยข้อที่มีค่าเฉลี่ยมากที่สุดคืออาจารย์จะต้องมีความรู้ทางวิชาการ มีความกระตือรือร้นที่จะ แสวงหาความรู้ใหม่ ๆ อยู่เสมอ รองลงมาคืออาจารย์ต้อง เป็นแบบอย่างของผู้มีความรับผิดชอบด้านความ เจริญก้าวหน้าทางวิชาการให้แก่สถาบัน และข้อที่มี ค่าเฉลี่ยน้อยที่สุดคือ อาจารย์มีความทะเยอทะยาน ใน การพัฒนาความรู้ ความสามารถอยู่เสมอ และ อาจารย์รู้สึกเป็นผู้ที่มีความสุขในการผลิตผลงานทาง วิชาการ เพราะมีความพึงพอใจในตัวของตัวเอง โดยมีได้หวังรางวัล

3. ปัจจัยองค์กรที่ส่งผลต่อเจตคติและ ความมุ่งมั่นในการผลิตผลงานทางวิชาการของอาจารย์ในสถาบัน อุดมศึกษาเอกชน ที่ระดับ 0.01 คือด้าน บรรยากาศองค์กร นอกจากนี้ปัจจัยด้านความเป็นผู้นำทางวิชาการของผู้บริหาร สามารถร่วมพยากรณ์ เจตคติและความมุ่งมั่นในการผลิตผลงานทางวิชาการ ของอาจารย์ในสถาบันอุดมศึกษา

2.5 กรอบแนวคิดและการวิจัย

