

การพัฒนาบบบริหารจัดการเว็บไซต์คณะฯ พร้อมเพิ่มประสิทธิภาพ SEO

และการยืนยันตัวตนด้วยโทเคน

บทคัดย่อ

งานวิจัยนี้มีวัตถุประสงค์เพื่อออกแบบและพัฒนาระบบจัดการเว็บไซต์สำหรับคณะในมหาวิทยาลัย โดยเน้นการเพิ่มประสิทธิภาพในการเผยแพร่ข้อมูลและการเข้าถึงของผู้ใช้ผ่านเทคนิคการปรับปรุงประสิทธิภาพของเครื่องมือค้นหา (Search Engine Optimization: SEO) พร้อมทั้งเสริมสร้างความปลอดภัยในการเข้าถึงข้อมูลของผู้ใช้ด้วยระบบการยืนยันตัวตนแบบ Token Authentication โดยใช้ JSON Web Token (JWT) เพื่อรองรับการควบคุมสิทธิ์ในการเข้าถึงส่วนต่าง ๆ ของระบบอย่างปลอดภัย

ระบบที่พัฒนาขึ้นถูกออกแบบให้รองรับการจัดการเนื้อหาบนเว็บไซต์ (Content Management) อย่างเป็นระบบ มีส่วนของผู้ดูแลระบบสำหรับจัดการเนื้อหา ข่าวสาร บุคลากร และกิจกรรมของคณะ โดยสามารถใช้งานผ่านอินเทอร์เฟซที่เป็นมิตรกับผู้ใช้ อีกทั้งยังรองรับมาตรฐาน SEO เพื่อเพิ่มโอกาสในการค้นพบเว็บไซต์ผ่านเครื่องมือค้นหาชั้นนำ เช่น Google

ผลการทดสอบระบบแสดงให้เห็นว่าระบบสามารถทำงานได้อย่างมีประสิทธิภาพ ผู้ดูแลสามารถจัดการเนื้อหาได้ง่าย ผู้ใช้สามารถเข้าถึงข้อมูลได้สะดวก และระบบมีความปลอดภัยจากการเข้าถึงโดยไม่ได้รับอนุญาต ทั้งนี้ระบบสามารถนำไปประยุกต์ใช้งานจริงในหน่วยงานการศึกษาอื่น ได้อย่างเหมาะสม

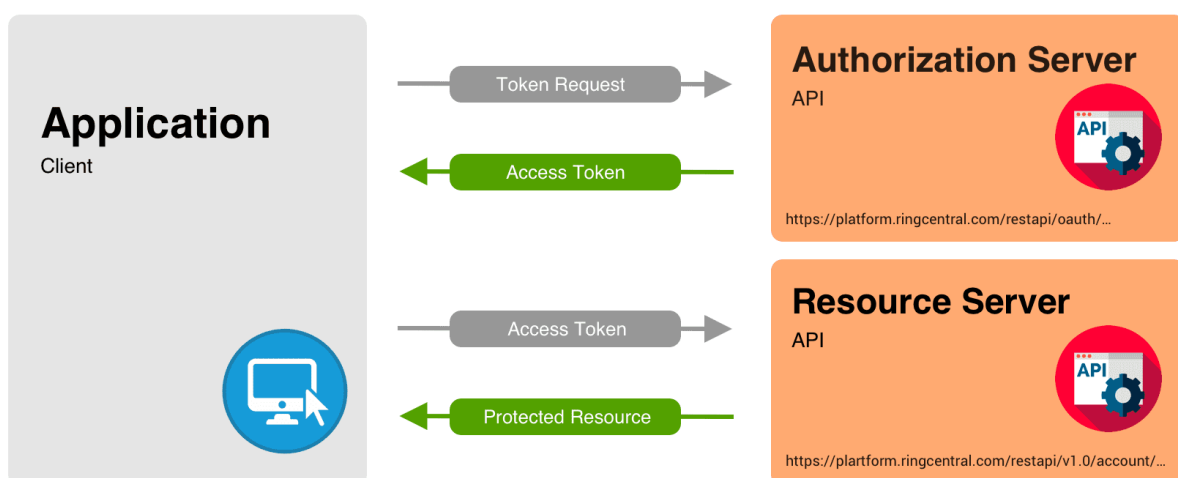
บทนำ

ในยุคดิจิทัล เว็บไซต์มีบทบาทสำคัญในการเป็นช่องทางสื่อสารและเผยแพร่ข้อมูลของหน่วยงานต่าง ๆ โดยเฉพาะในระดับคณะของมหาวิทยาลัย ซึ่งมีหน้าที่นำเสนอข่าวสารทางวิชาการ ข้อมูลบุคลากร กิจกรรม และบริการแก่นักศึกษาและบุคคลทั่วไป อย่างไรก็ตาม เว็บไซต์จำนวนมากยังขาดประสิทธิภาพในการเข้าถึงจากเครื่องมือค้นหา (Search Engines) และมีข้อจำกัดในการจัดการข้อมูล รวมถึงขาดระบบรักษาความปลอดภัยที่ทันสมัยสำหรับผู้ใช้งานที่มีสิทธิ์เข้าถึงข้อมูลเฉพาะกลุ่ม

จากปัญหาดังกล่าว งานวิจัยนี้มีวัตถุประสงค์เพื่อพัฒนา ระบบจัดการเว็บไซต์ระดับคณะในมหาวิทยาลัย ที่สามารถรองรับการปรับปรุงโครงสร้างเว็บไซต์เพื่อให้เหมาะกับการค้นหาจากเครื่องมือค้นหา (Search Engine Optimization: SEO) และเพิ่มความปลอดภัยในการใช้งานด้วย ระบบยืนยันตัวตนแบบ Token (JSON Web Token: JWT) เพื่อควบคุมสิทธิ์ในการเข้าถึงเนื้อหาภายในระบบได้อย่างปลอดภัยและเหมาะสมตามบทบาทของผู้ใช้งาน

ระบบพัฒนาขึ้นโดยใช้เทคโนโลยีเว็บสมัยใหม่ โดยแยกส่วนการทำงานของระบบออกเป็นฝั่งผู้ใช้งาน (Frontend) ที่พัฒนาด้วย Nuxt.js และฝั่งบริการ API (Backend) ที่พัฒนาด้วย Spring Boot ซึ่งช่วยให้ระบบมีความยืดหยุ่น ปรับขยายง่าย และรองรับการใช้งานในอนาคต นอกจากนี้ ระบบยังรองรับการจัดการเนื้อหาสำคัญ เช่น ข่าวกิจกรรม ข้อมูลบุคลากร และโครงสร้างเมนูเว็บไซต์ ผ่านส่วนของผู้ดูแลระบบที่ใช้งานง่าย

บทความนี้นำเสนอโครงสร้างระบบ รายละเอียดการพัฒนา และผลการประเมินในด้านต่าง ๆ ได้แก่ ความสามารถในการค้นหา (SEO Score), ความปลอดภัยในการเข้าถึงระบบ และความพึงพอใจของผู้ใช้งาน ซึ่งผลการทดลองแสดงให้เห็นว่าระบบที่พัฒนาขึ้นสามารถตอบโจทย์การใช้งานได้อย่างมีประสิทธิภาพ และสามารถประยุกต์ใช้กับคณะหรือหน่วยงานอื่นในสถาบันอุดมศึกษาได้



งานวิจัยที่เกี่ยวข้อง

ในปัจจุบัน หลายมหาวิทยาลัยได้พัฒนาเว็บไซต์เพื่อเป็นช่องทางหลักในการประชาสัมพันธ์ข้อมูลข่าวสารของคณะ ซึ่งมีงานวิจัยหลายฉบับที่กล่าวถึงการออกแบบเว็บไซต์เชิงโครงสร้าง การเพิ่มประสิทธิภาพ SEO และการรักษาความปลอดภัยของระบบ เช่น งานวิจัยของ ศุภลักษณ์ และคณะ (2564) ได้ศึกษาแนวทางการพัฒนาเว็บไซต์หน่วยงานราชการที่เน้นความเป็นมิตรต่อผู้ใช้ (Usability) และรองรับอุปกรณ์พกพา (Mobile responsive)

อีกทั้ง งานของ นิธิกร (2565) ได้นำเสนอการใช้เทคนิค SEO เพื่อเพิ่มการเข้าถึงของเว็บไซต์ในระดับท้องถิ่น โดยใช้เครื่องมือ Google Search Console และการจัดโครงสร้าง URL อย่างเป็นระบบ

ในด้านความปลอดภัยของระบบ มีงานวิจัยของ สุรเดช (2563) ซึ่งได้เสนอการนำ JSON Web Token (JWT) มาประยุกต์ใช้กับระบบหลังบ้านเพื่อควบคุมสิทธิ์ของผู้ใช้งานอย่างปลอดภัย

งานวิจัยฉบับนี้จึงได้นำแนวทางจากงานก่อนหน้า มาพัฒนาเว็บไซต์คณะให้ทันสมัย ปลอดภัย และเหมาะสมต่อการค้นหาผ่านเครื่องมือค้นหา (Search Engine) พร้อมทั้งมีระบบจัดการเนื้อหาภายในที่ใช้งานง่ายสำหรับเจ้าหน้าที่

ระเบียบวิธีวิจัย

การดำเนินงานวิจัยในครั้งนี้แบ่งออกเป็น 5 ขั้นตอนหลัก ดังนี้

1. การวิเคราะห์ความต้องการของระบบ

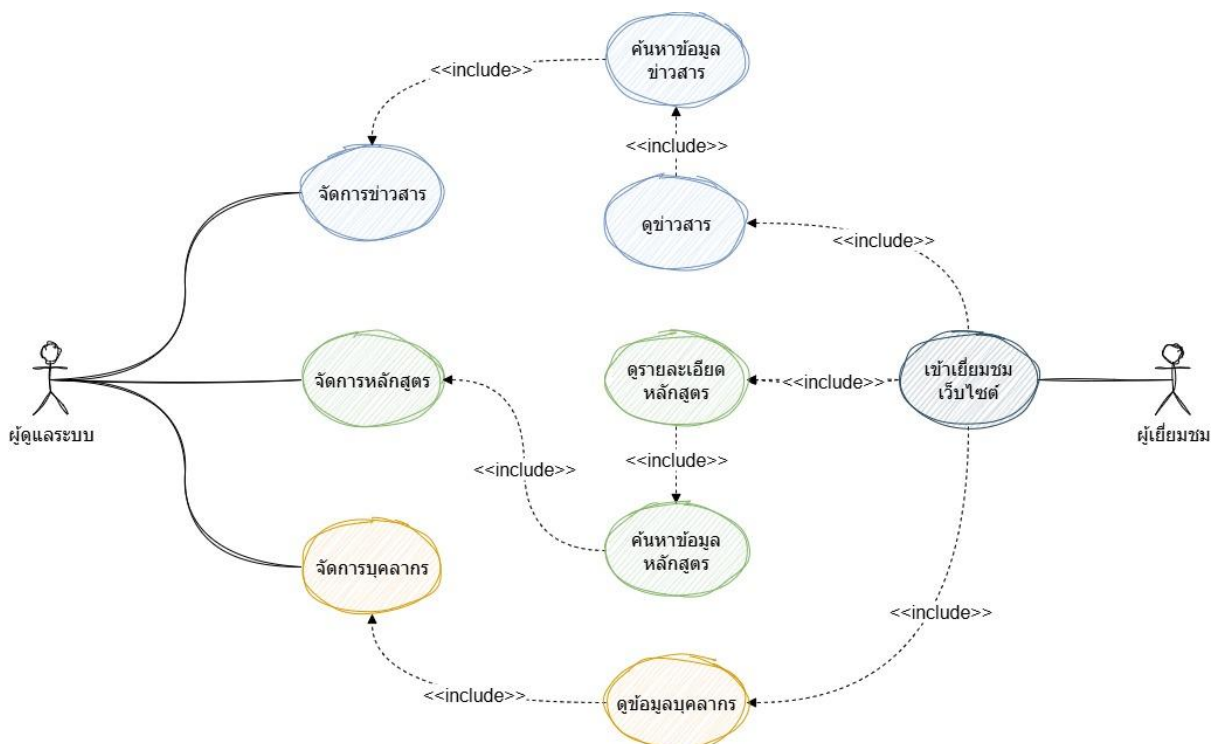
ทำการศึกษาปัญหา และรวบรวมความต้องการจากกลุ่มผู้ใช้งานจริง ได้แก่ บุคลากรของคณะ นักศึกษา และเจ้าหน้าที่ฝ่ายสารสนเทศ เพื่อกำหนดคุณลักษณะและฟังก์ชันที่ระบบควรมี เช่น ระบบจัดการบทความ ข่าวสาร ระบบแสดงข้อมูลหลักสูตร เป็นต้น

ความต้องการระบบ

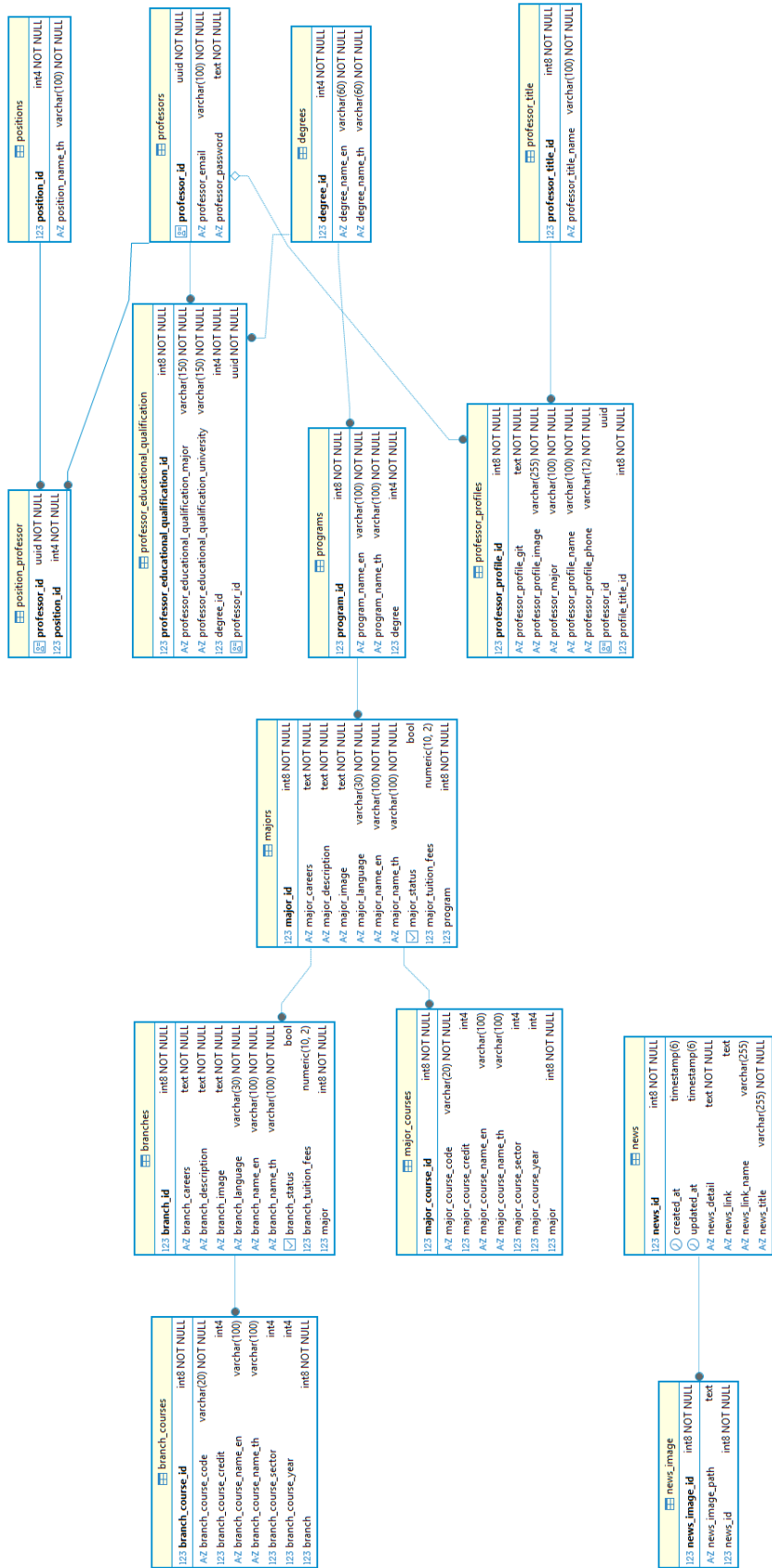
1. ต้องการเว็บไซต์จัดการเนื้อหาโดยไม่ต้องเขียนโค้ดเองในภายหลัง
2. ต้องการระบบความปลอดภัยที่พึงพาได้
3. ต้องการเว็บไซต์ติดอันดับการค้นหา SEO เพื่อให้ผู้คนเข้าถึงสถาบันและคณะมากขึ้น

2. การออกแบบระบบ

ทำการออกแบบโครงสร้างฐานข้อมูล (Database Schema), ส่วนติดต่อผู้ใช้งาน (User Interface), และสถาปัตยกรรมระบบโดยรวม (System Architecture) โดยพิจารณาถึงความปลอดภัย (Token-Based Authentication), ความสามารถในการค้นหา (SEO Structure) และความง่ายต่อการบำรุงรักษา



แผนภาพ Use Case Diagram ของระบบบริหารจัดการเว็บไซต์คณะฯ



แผนภาพการออกแบบระบบฐานข้อมูล PostgreSQL ของระบบบริหารจัดการเว็บไซต์คณะฯ

3. การพัฒนาระบบ

ใช้เทคโนโลยี เช่น Java Spring Boot, PostgreSQL และ Nuxt 3 ในการพัฒนา API และเว็บไซต์ พร้อมทั้งบูรณาการ SEO ด้วยการปรับ Meta Tag, URL และ Sitemap รวมถึงการใช้ JWT Token เพื่อควบคุมการเข้าถึงระบบ

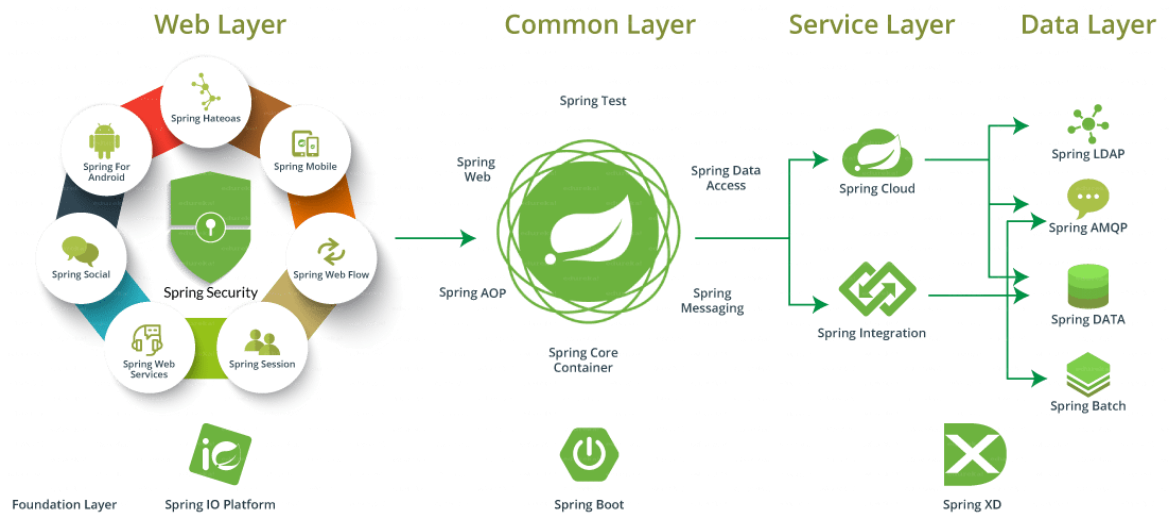
เทคนิคการทำ SEO Meta Tage ด้วย Nuxt.js

```
useHead({
  title: major.value?.nameTh || 'ข้อมูลสาขาวิชา',
  htmlAttrs: {
    lang: 'th'
  },
  meta: [
    {
      name: 'description',
      content: major.value?.description?.slice(0, 160) ||
      'รายละเอียดหลักสูตร สาขาวิชา คณะ และอาชีพในอนาคต',
    },
    {
      name: 'keywords',
      content: `${major.value?.nameTh}, ${major.value?.nameEn}
      , มหาวิทยาลัย, สาขาวิชา, คณะ, เรียนต่อ, ค่าเทอม, อาชีพ`,
    },
    {
      property: 'og:title',
      content: major.value?.nameTh || 'ข้อมูลสาขาวิชา',
    },
    {
      property: 'og:description',
      content: major.value?.description?.slice(0, 160) ||
      'รายละเอียดหลักสูตร สาขาวิชา คณะ และอาชีพในอนาคต',
    },
    {
      property: 'og:image',
      content: major.value?.image ? `${apiBase}/images/${major.value.image}` :
      '/default-og-image.jpg',
    },
    {
      property: 'og:url',
      content: `${apiBase}/major/${route.params.id}`,
    },
    {
      name: 'twitter:card',
      content: 'summary_large_image',
    },
  ],
  link: [
    {
      rel: 'canonical',
      href: `${apiBase}/major/${route.params.id}`,
    },
  ],
})
```

ภาพวิธีการทำ SEO ด้วย Meta Tage ของ Nuxt.js พร้อมการทำ SEO ร่วมกับ SSR

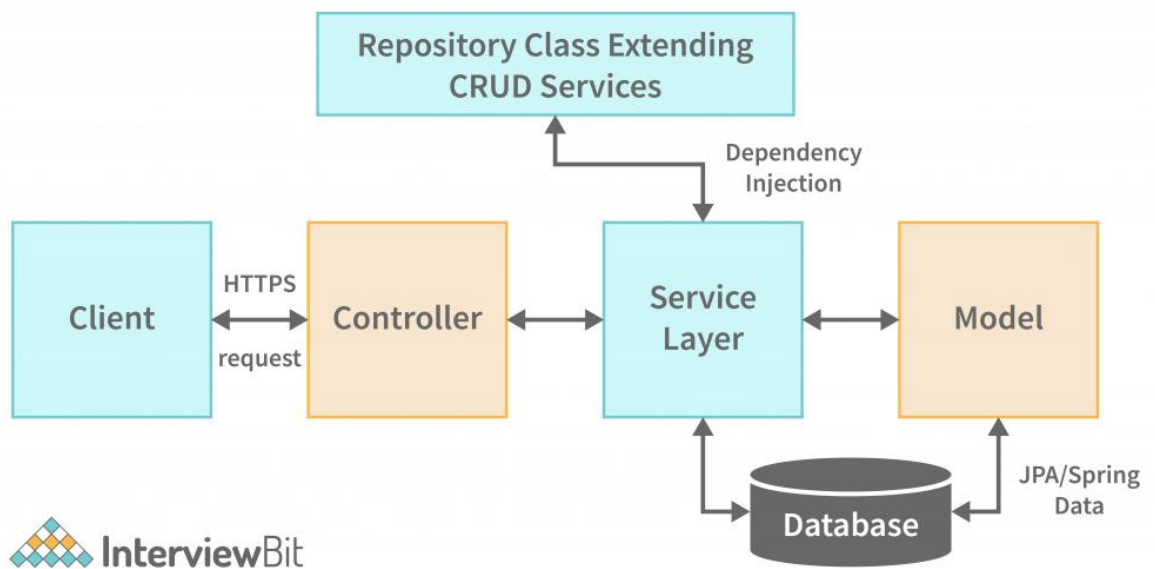
การพัฒนาระบบ Backend ด้วย ภาษา Java ใช้งานกับ Spring Boot

edureka!

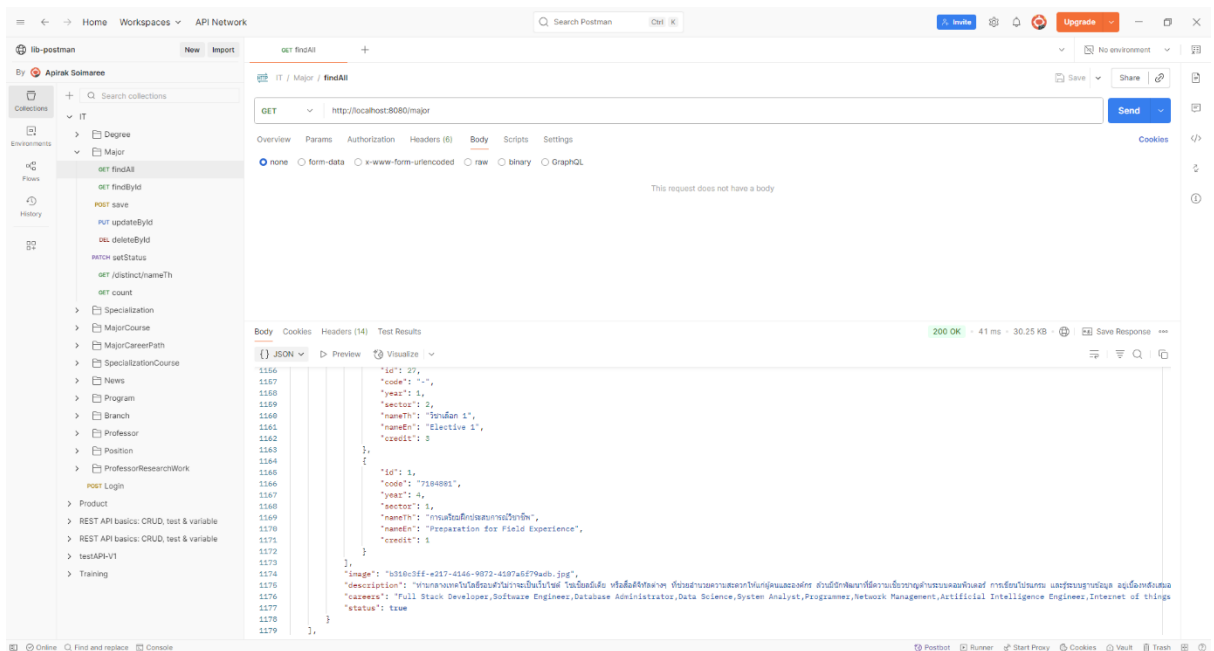


ภาพ Spring Boot Ecosystem

Spring Boot Flow Architecture

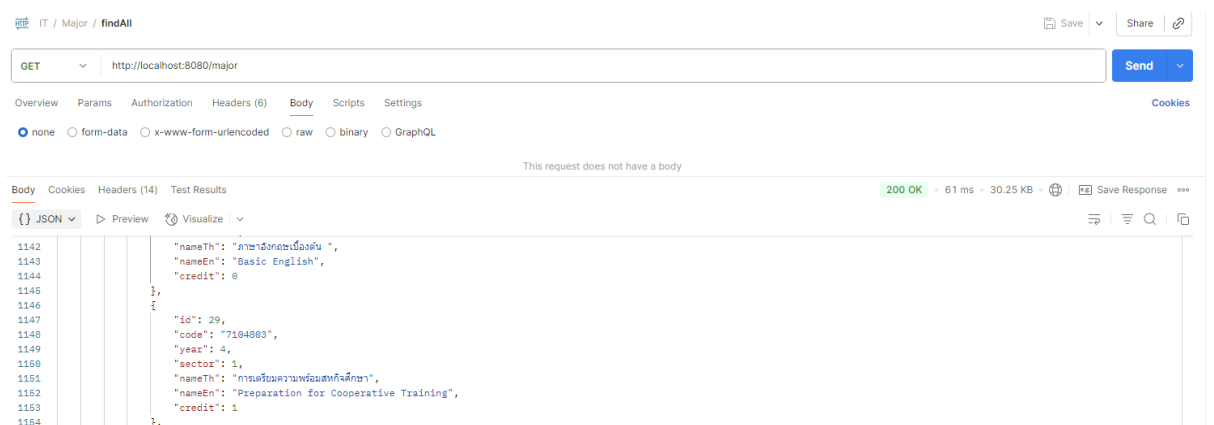


Spring Boot Structure



ภาพนี้มีการแสดง latency ในการดึงข้อมูล object ขนาดใหญ่ และมีข้อมูลจำนวนมาก แต่ยังประสิทธิภาพการทำงานที่ดีมาก เนื่องจากการใช้งานเทคโนโลยีใหม่ของภาษา Java คือ Virtual Threads

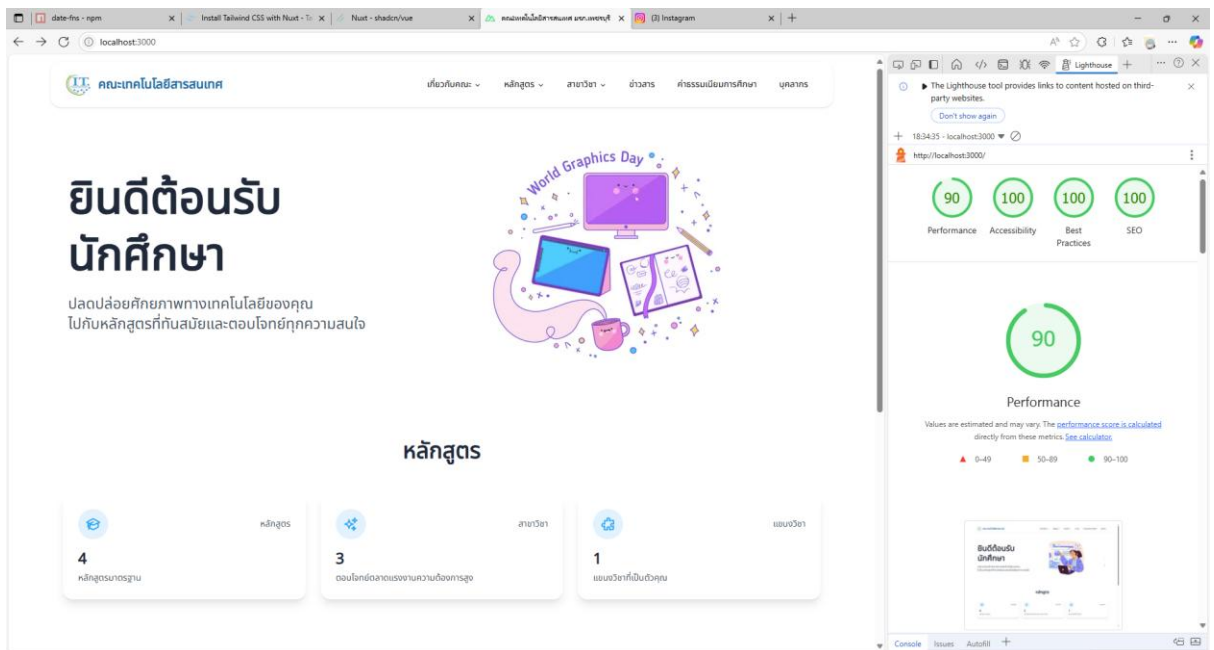
มีการทดสอบเมื่อปิด Virtual Threads ได้ผลสรุปได้ว่า เทคโนโลยี Virtual Threads มีส่วนช่วยให้ประสิทธิภาพที่ดีขึ้นมาก โดยที่ไม่จำเป็นต้องเขียนแบบ Asynchronous เขียนแบบ Synchronous ก็ได้ประสิทธิภาพเทียบเคียง Asynchronous Non-Blocking I/O เมื่อปิด Virtual Threads มี latency เพิ่มขึ้นประมาณ 20-100ms ในทุกๆ Request อย่างไรก็ตาม ขึ้นอยู่กับขนาดของข้อมูลที่ถูกดึง ความเร็วในการอ่าน Disk และ Network ของแต่ละ Environment



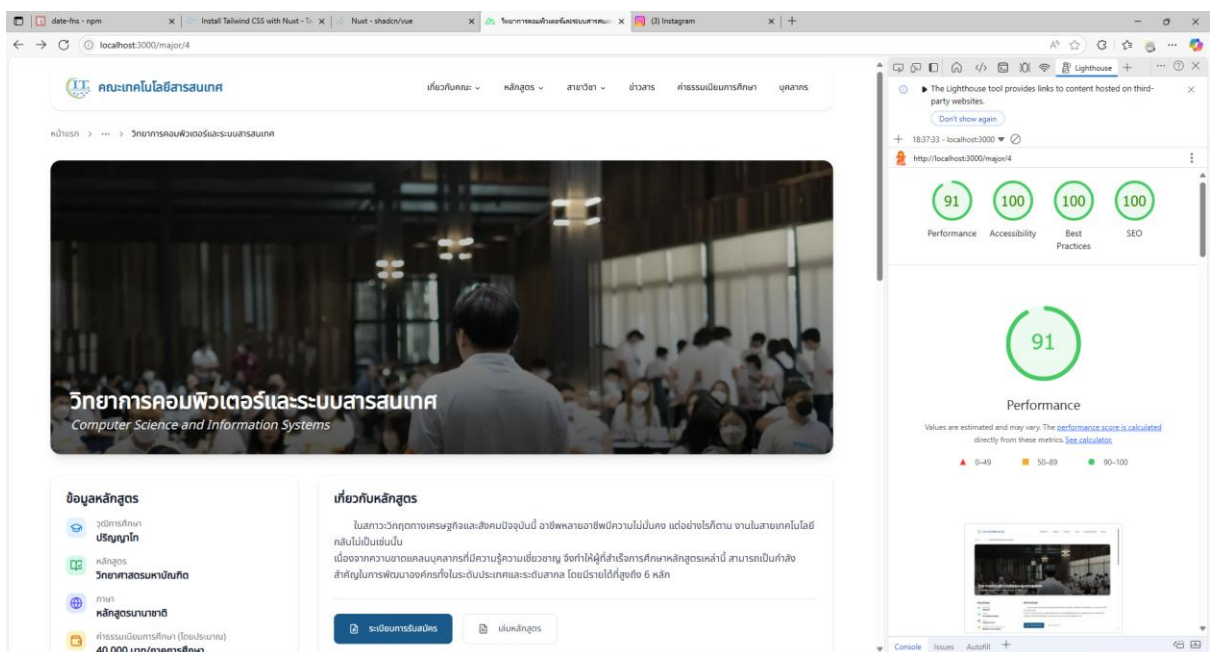
4. การทดสอบระบบ

ทำการทดสอบระบบด้วยเทคนิค Unit Testing และ Integration Testing เพื่อให้แน่ใจว่าทุกฟังก์ชันทำงานถูกต้อง รวมถึงทำการทดสอบ SEO ด้วยเครื่องมือ เช่น Google Lighthouse และทดสอบความปลอดภัยของ Token Authentication

ทดสอบประสิทธิภาพ



ภาพผลการทดสอบคะแนน lighthouse หน้าหลัก

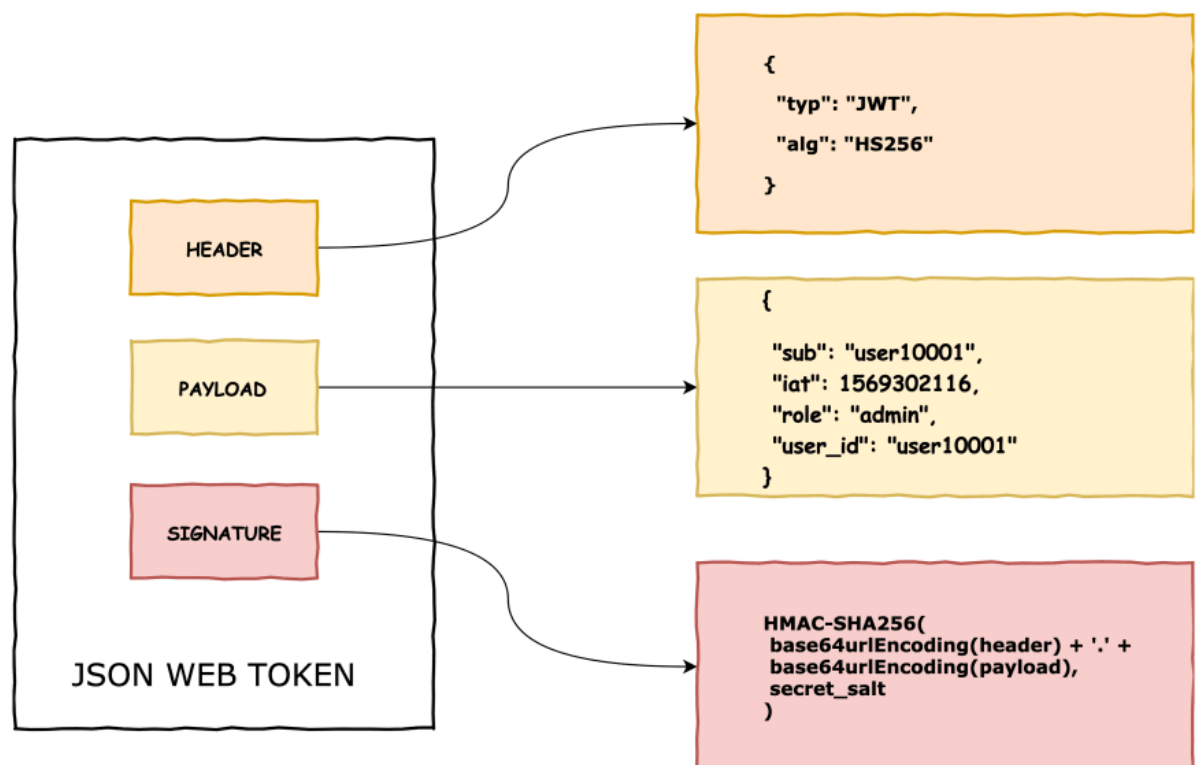


ภาพผลการทดสอบคะแนน lighthouse หน้าสาขาวิชา

ทดสอบระบบ Authorization ด้วย JWT(Json Web Token)

The screenshot shows the JWT Debugger interface. The 'ENCODED VALUE' field contains a long base64-encoded string. The 'DECODED HEADER' field shows a JSON object: `{ "alg": "RS384" }`. The 'DECODED PAYLOAD' field shows a JSON object: `{ "sub": "apirak.soi@mail.pbru.ac.th", "roles": [{ "authority": "ผู้ดูแล" }, { "authority": "อาจารย์ประจำสาขาวิชา" }], "iat": 1753352180, "exp": 1753359380 }`.

ภาพการทดสอบ Json Web Token



JSON WEB TOKEN

ทดสอบฟังก์ชันการทำงานต่างๆ ด้วย Junit , Spring Test

```
@BeforeEach // Method ที่จะรันก่อนทุกๆ @Test
void setUp() {
    // เตรียมข้อมูลตัวอย่าง
    Program program = new Program();
    program.setId(1L);
    program.setNameTh("หลักสูตรวิศวกรรมศาสตรบัณฑิต");

    major = new Major();
    major.setId(1L);
    major.setNameTh("วิศวกรรมคอมพิวเตอร์");
    major.setNameEn("Computer Engineering");
    major.setProgram(program);
    major.setStatus(true);

    majorRequest = new MajorRequest();
    majorRequest.setProgramId(1L);
    majorRequest.setNameTh("วิศวกรรมคอมพิวเตอร์");
    majorRequest.setNameEn("Computer Engineering");
    majorRequest.setStatus(true);
}

@Test
void findById_whenMajorExists_shouldReturnMajor() {
    // Given: กำหนดว่าเมื่อ majorRepository.findById(1L) ถูกเรียก จะต้องคืนค่า
    // Optional ที่มี major object ของเรา
    when(majorRepository.findById(1L)).thenReturn(Optional.of(major));

    // When: เรียก service method ที่ต้องการทดสอบ
    Optional<Major> foundMajor = majorService.findById(1L);

    // Then: ตรวจสอบผลลัพธ์
    assertTrue(foundMajor.isPresent(), "ควรจะมี Major เจอ");
    assertEquals("วิศวกรรมคอมพิวเตอร์", foundMajor.get().getNameTh(),
        "ชื่อภาษาไทยของ Major ควรจะตรงกัน");

    // Verify: ตรวจสอบว่า majorRepository.findById(1L) ถูกเรียกใช้งาน 1 ครั้ง
    verify(majorRepository, times(1)).findById(1L);
}
```

ภาพการทดสอบฟังก์ชันด้วย Test Annotations ของ Spring Boot

5. การประเมินผล

ดำเนินการประเมินผลจากกลุ่มผู้ใช้งาน (กลุ่มตัวอย่าง) โดยใช้แบบสอบถามด้านความพึงพอใจ และการสัมภาษณ์เชิงลึก เพื่อปรับปรุงให้ระบบมีประสิทธิภาพและตรงตามความต้องการ

การประเมิน	ค่าเฉลี่ย	ส่วนเบี่ยงเบนมาตรฐาน
การใช้งานส่วนต่อประสานผู้ใช้ สำหรับผู้ใช้งานทั่วไป	3.53	0.51
การใช้งานส่วนต่อประสานผู้ใช้ สำหรับผู้ดูแล	3.41	0.51
การใช้สี รูปร่าง ชิ้นส่วน และองค์ประกอบของเว็บ ในบริบทต่างๆ	3.53	0.51
การจัดองค์ประกอบของเว็บไซต์	3.38	0.50
การออกแบบระบบครบสมบูรณ์ มีความชัดเจน และถูกต้องตามความต้องการ	4.00	0.00
ระบบมีความยืดหยุ่น สามารถรองรับการ Maintenance และการ Scale ได้ในอนาคต	4.00	0.00
Diagram มีความถูกต้อง มีความละเอียด และเข้าใจง่าย	4.00	0.00
การออกแบบระบบถูกต้องตามหลักการ ทั้งส่วน System Analyst & Software Architecture สามารถใช้สื่อสารร่วมกับคนอื่นได้	4.00	0.00
ประสิทธิภาพการโหลดข้อมูลเว็บไซต์	4.00	0.00
การจัดการ การโหลดข้อมูลส่วน Backend Web API	4.00	0.00
การจัดการ Search Engine Optimization (SEO) กระบวนการที่จะทำให้เว็บไซต์ได้รับ Organic Traffic จำนวนมาก จากการติดอันดับบนหน้าการค้นหา	4.00	0.00
โค้ดมี Algorithm ที่ถูกต้อง มีประสิทธิภาพ เข้าใจได้ง่าย	3.47	0.62
การนำเสนอข้อมูลไม่ซับซ้อนและเข้าใจง่าย	3.47	0.62
การนำเสนอข้อมูลมีความหลากหลาย	3.35	0.61
การค้นหาข้อมูล การกรองข้อมูล	3.41	0.71
การจัดการข้อมูลที่มีความ Sensitive	3.35	0.70
เว็บไซต์มีการป้องกันการโจมตีอย่างครอบคลุม	3.41	0.71
เว็บไซต์มีความปลอดภัย มั่นคง น่าเชื่อถือ	3.35	0.61
เว็บไซต์มี log สำหรับการทำงานต่างๆ เพื่อดูการทำงานของระบบ สามารถติดตามการทำงานได้ เมื่อมีภัยอันตราย	3.18	0.73
เว็บไซต์มีการจัดการสิทธิการใช้งาน	3.29	0.59