

โมดูลการเรียนรู้

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

การพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End รหัสวิชา 31901-2004

ผู้จัดทำ

นายวิศวัต หนูมาศ

วิทยาลัยอาชีวศึกษานครศรีธรรมราช

สำนักงานคณะกรรมการการอาชีวศึกษา

กระทรวงศึกษาธิการ

หลักสูตรประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

ประเภทวิชา อุตสาหกรรมดิจิทัลและเทคโนโลยีสารสนเทศ กลุ่มอาชีพ ซอฟต์แวร์และการประยุกต์ สาขาวิชา เทคโนโลยีสารสนเทศ

รหัสวิชา 31901-2004

ชื่อวิชา

การพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 4 ชั่วโมง/สัปดาห์ รวม ...5.... ชั่วโมง จำนวน 3 หน่วยกิต

อ้างอิงมาตรฐาน

มาตรฐานอาชีพ สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) สาขาวิชาชีพเทคโนโลยีสารสนเทศและการสื่อสาร และดิจิทัลคอนเทนต์ อาชีพ นักพัฒนาระบบ ระดับ 4

ผลลัพธ์การเรียนรู้ระดับรายวิชา

พัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End เพื่อออกแบบส่วนติดต่อผู้ใช้ตามหลักการ ด้วยความละเอียดรอบคอบ รับผิดชอบ การสื่อสารและการทำงานเป็นทีม

จุดประสงค์รายวิชา เพื่อให้

1. เข้าใจการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End
2. มีทักษะในการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End
3. มีเจตคติและกิริยาที่ดีในการปฏิบัติงานด้วยความละเอียดรอบคอบ รับผิดชอบ การสื่อสาร การคิดเชิงนวัตกรรมและการทำงานเป็นทีม
4. มีความสามารถประยุกต์ใช้หลักการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End ในงานอาชีพ

สมรรถนะรายวิชา

1. ประมวลความรู้เกี่ยวกับหลักการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End ตามหลักการ
2. พัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End ตามหลักการและกระบวนการ
3. ทดสอบและแก้ไขข้อผิดพลาดของซอฟต์แวร์ ตามหลักการและกระบวนการ
4. จัดทำคู่มือการใช้งานตามหลักการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End
5. ประยุกต์ใช้หลักการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End ในงานอาชีพ

คำอธิบายรายวิชา

ศึกษาและปฏิบัติเกี่ยวกับหลักการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End ความหมาย ความสำคัญ ประโยชน์ของการพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End เลือกใช้ภาษาและเทคโนโลยีสมัยใหม่ในการพัฒนา (Vue/React/Angular/ฯลฯ) การทำงานร่วมกันเป็นทีม (Version Control System) ดำเนินการพัฒนาโปรแกรมแบบ Integration ตกแต่งเอกสารด้วย UI Component Library/CSS Framework เขียนโปรแกรมเชื่อมต่อ RESTful API/Web Services เขียนโปรแกรมติดต่อฐานข้อมูลแบบ SQL/NoSQL การทำ Authentication ในรูปแบบ JSON Web Token/Session หาจุดผิดพลาด ตามบันทึกข้อผิดพลาด แก้ไขข้อผิดพลาดของโปรแกรม ทดสอบการแก้ไขข้อผิดพลาดของโปรแกรม อ่าน Functional/Program Specification/UML เขียนโปรแกรมตาม Functional/Program Specification/UML ออกแบบการทดสอบ Integration Test ดำเนินการทดสอบโปรแกรมแบบ Integration Test จัดทำรายงาน ศึกษาการใช้งานโปรแกรมที่พัฒนาขึ้น จัดทำคู่มือการใช้งานโปรแกรม ตรวจสอบความถูกต้องของคู่มือการใช้งานโปรแกรม การส่งมอบซอฟต์แวร์เพื่อให้สามารถใช้งานได้

โมดูลการเรียนรู้

รหัสวิชา 31901-2004

ชื่อวิชา

การพัฒนาซอฟต์แวร์ด้วยเทคโนโลยี Front-End

ทฤษฎี 1 ชั่วโมง/สัปดาห์ ปฏิบัติ 4 ชั่วโมง/สัปดาห์ รวม ...5.... ชั่วโมง จำนวน 3 หน่วยกิต

ที่	ชื่อโมดูล	เวลาเรียน (ชม.)			หมายเหตุ
		ทฤษฎี	ปฏิบัติ	รวม	
1	สถาปัตยกรรม Front-End ยุคใหม่และระบบควบคุมเวอร์ชัน	2	8	10	
2	การพัฒนาส่วนติดต่อผู้ใช้ด้วย Component และ CSS Framework	3	12	15	
3	การพัฒนาซอฟต์แวร์แบบ Integration เชื่อมต่อ API และระบบพิสูจน์ตัวตน	4	16	20	UoC 10305
4	การทดสอบซอฟต์แวร์แบบ Integration Test และการจัดทำรายงาน	3	12	15	UoC 10401
5	การตรวจหาและแก้ไขข้อผิดพลาดของซอฟต์แวร์	3	12	15	UoC 10303
6	การจัดทำคู่มือการใช้งานและการส่งมอบซอฟต์แวร์	2	8	10	UoC 10703
	ประเมินผลสัมฤทธิ์การเรียนรู้ปลายภาค	1	4	5	Summative
รวม		18	72	90	18 สัปดาห์

โมดูลการเรียนรู้ที่ 1 สถาปัตยกรรม Front-End ยุคใหม่และระบบควบคุมเวอร์ชัน

เวลาเรียน 10 ชั่วโมง (ทฤษฎี 2 ชั่วโมง, ปฏิบัติ 8 ชั่วโมง)

1. ผลลัพธ์การเรียนรู้ (ความรู้+ทักษะ+จิตพิสัย=การประยุกต์ใช้)

ผู้เรียนมีความรู้ความเข้าใจเกี่ยวกับวิวัฒนาการและสถาปัตยกรรมของเว็บแอปพลิเคชันยุคใหม่ (SPA/SSR) เครื่องมือสนับสนุนการพัฒนา (Front-End Tooling) และระบบควบคุมเวอร์ชัน สามารถติดตั้ง จัดการสภาพแวดล้อมการพัฒนาด้วย Node.js, NPM, Vite และใช้คำสั่ง Git ในการจัดการซอร์สโค้ด กิ่งงาน (Branch) และการทำงานร่วมกันเป็นทีมบนแพลตฟอร์มคลาวด์ (GitHub/GitLab) ได้อย่างถูกต้อง มีระเบียบวินัย ละเอียดยรอบคอบ และสื่อสารในทีมได้อย่างมีประสิทธิภาพ

2. อ้างอิงมาตรฐาน (ถ้ามี)

มาตรฐานคุณวุฒิวิชาชีพ สถาบันคุณวุฒิวิชาชีพ (องค์การมหาชน) – สาขาวิชาชีพเทคโนโลยีสารสนเทศและการสื่อสาร และดิจิทัลคอนเทนต์ อาชีพนักพัฒนาระบบ ระดับ 4 (สมรรถนะการทำงานร่วมกันเป็นทีมด้วยระบบ Version Control System และการบริหารจัดการสภาพแวดล้อมการพัฒนาซอฟต์แวร์)

3. สมรรถนะประจำโมดูล

3.1 แสดงความรู้เกี่ยวกับวิวัฒนาการ สถาปัตยกรรม Front-End ยุคใหม่ เครื่องมือพัฒนา และหลักการทำงานของระบบ Version Control System (Git)

3.2 ปฏิบัติการติดตั้งเครื่องมือพัฒนาโปรแกรม กำหนดค่าสภาพแวดล้อม และใช้คำสั่ง Git (init, clone, add, commit, push, pull, branch, merge) ได้อย่างถูกต้องคล่องแคล่ว

3.3 บริหารจัดการพื้นที่จัดเก็บโค้ด (Repository) โครงการ Front-End บนระบบคลาวด์ร่วมกับทีม แก้ไขข้อขัดแย้งของโค้ด (Conflict Resolution) และสร้าง Pull Request ได้ถูกต้องตามหลักปฏิบัติที่ดีของอุตสาหกรรมซอฟต์แวร์

4. จุดประสงค์เชิงพฤติกรรม

4.1 พุทธิพิสัย

4.1.1 อธิบายความหมาย ความสำคัญ และข้อแตกต่างระหว่าง Traditional Web กับ Single Page Application (SPA) ได้ถูกต้อง

4.1.2 อธิบายวงจรการทำงานของ Git (Working Directory, Staging Area, Local Repository, Remote Repository) ได้แม่นยำ

4.1.3 ระบุกลยุทธ์การบริหารจัดการกิ่งงาน (Branching Strategy: Git Flow) สำหรับการทำงานร่วมกันเป็นทีมได้ถูกต้อง

4.2 ทักษะพิสัย

4.2.1 ติดตั้ง Node.js, NPM และสร้างโครงสร้างโครงการ Front-End ด้วย Vite ได้อย่างถูกต้อง

4.2.2 ปฏิบัติการกำหนดค่า Git Config, การสร้างและสลับกิ่งงาน (git branch, git checkout, git switch) ได้อย่างคล่องแคล่ว

4.2.3 ปฏิบัติการแก้ปัญหาการรวมกิ่งขัดแย้ง (Merge Conflicts) และการสร้าง Pull Request / Code Review ได้อย่างถูกต้อง

4.3 จิตพิสัย

4.3.1 ปฏิบัติงานด้วยความรับผิดชอบ มีวินัยในการเขียน Commit Message ที่มีความหมายชัดเจน ตรงต่อเวลา และเคารพกฎระเบียบของทีม

4.4 การประยุกต์ใช้

4.4.1 ริเริ่มโครงการ Front-End เชื่อมต่อกับระบบจัดเก็บโค้ดบน GitHub และบริหารจัดการการส่งงานร่วมกับสมาชิกในทีมผ่านกระบวนการ Git Workflow ได้เสร็จสมบูรณ์

5. สารการเรียนรู้

5.1 วิวัฒนาการและความสำคัญของการพัฒนาเว็บ Front-End (Web 1.0 สู่ Modern Web SPA, CSR vs SSR)

5.2 สภาพแวดล้อมและเครื่องมือพัฒนาสมัยใหม่ (Node.js Runtime, Package Managers: NPM/Yarn, Build Tool: Vite)

5.3 หลักการและสถาปัตยกรรมของระบบควบคุมเวอร์ชัน (Version Control Concepts: Centralized vs Distributed VCS)

5.4 การใช้งาน Git ระดับพื้นฐานถึงระดับกลาง (git init, status, add, commit, log, diff, restore)

5.5 การทำงานร่วมกันผ่าน Cloud Repository (GitHub/GitLab): git remote, push, pull, clone, fetch

5.6 กลยุทธ์การแตกกิ่งและการรวมกิ่ง (Branching & Merging): Feature Branches, Merge Conflicts Resolution, Pull Requests และ Code Review

6. รายละเอียดการเรียนรู้ประจำโมดูล

6.1 ทฤษฎี

เรียนรู้หลักการ สถาปัตยกรรมเว็บ เครื่องมือพัฒนา และคำสั่ง Git ผ่านสื่อการสอนอิเล็กทรอนิกส์ การบรรยาย การสาธิตขั้นตอนจริง และกรณีศึกษาการทำงานของทีมพัฒนาซอฟต์แวร์

6.2 ปฏิบัติ

ฝึกปฏิบัติจริงในห้องปฏิบัติการคอมพิวเตอร์ ทำการติดตั้งซอฟต์แวร์ เริ่มต้นโครงการ จำลองสถานการณ์การพัฒนาแบบรายบุคคลและเป็นกลุ่ม ฝึกการแตกกิ่งงาน การแก้ไขข้อขัดแย้งของโค้ด และการส่งงานผ่าน Pull Request บน GitHub

7. สื่อและแหล่งการเรียนรู้

ด้านความรู้ (ทฤษฎี) เวลาเรียน 2 ชั่วโมง

สื่อเรื่อง สื่อเรื่อง สถาปัตยกรรม Front-End และระบบเครื่องมือสมัยใหม่ (สไลด์มัลติมีเดียและแบบฝึกหัด)

สื่อเรื่อง สื่อเรื่อง ระบบควบคุมเวอร์ชัน Git และกลยุทธ์การทำงานเป็นทีม (วิดีโอสาธิตและแบบทดสอบ)

ด้านทักษะและการประยุกต์ใช้ (ปฏิบัติ) เวลาเรียน 8 ชั่วโมง

ฝึกปฏิบัติจริง

ใบปฏิบัติงานที่ 1.1 "การติดตั้งสภาพแวดล้อมการพัฒนาและการจัดการโครงการ Front-End ด้วย Git & GitHub"

หลักฐานการปฏิบัติงาน

ลิงก์ Remote Repository บน GitHub/GitLab, ประวัติ Commit History, กิ่งงาน Feature Branch และหลักฐานการ Merge Pull Request

เกณฑ์การปฏิบัติงาน

โครงสร้างโครงการถูกต้อง รันสำเร็จ, มีประวัติ Commit ไม่น้อยกว่า 5 รายการด้วยข้อความมาตรฐาน (Conventional Commits), กิ่งงานรวมสำเร็จโดยไม่มี Conflict ตกค้าง 100%

ขอบเขตการปฏิบัติงาน

สร้างโครงการ Front-End เปล่าด้วย Vite, เชื่อมโยง Git Local สู่ Remote Repository บน GitHub, แบ่งงานเป็นกิ่งย่อยอย่างน้อย 2 กิ่ง และ Merge เข้าสู่ Main Branch ให้เสร็จสิ้นภายในเวลา 240 นาที

วิธีการประเมิน

ประเมินการปฏิบัติงานและคุณภาพการจัดการโค้ดบน GitHub, การตรวจประวัติ Commit, และการสัมภาษณ์อธิบายขั้นตอน Git Workflow

เครื่องมือประเมิน

แบบประเมินผลการปฏิบัติงานด้านการจัดการซอร์สโค้ดและ Git Workflow (Rubric Assessment)

8. การวัดและประเมินผลประจำโมดูล

ภาคทฤษฎี (30 คะแนน)

คะแนนแบบทดสอบความรู้ระบบออนไลน์ (20 คะแนน)

คะแนนแบบฝึกหัดและการมีส่วนร่วมในการอภิปรายในชั้นเรียน (10 คะแนน)

ภาคปฏิบัติ (70 คะแนน)

คะแนนการใช้คำสั่ง Git CLI และการบริหารจัดการงาน (35 คะแนน)

คะแนนความถูกต้องสมบูรณ์ของโครงการบน GitHub และการทำ Pull Request (35 คะแนน)