

(Web Programming)

รหัสรายวิชาเดิม : ENGCE176 การเขียนโปรแกรมบนเว็บ

วิชาบังคับก่อน : ENGCE126 ระบบฐานข้อมูล

ศึกษาและฝึกปฏิบัติการเกี่ยวกับ เทคโนโลยีของอินเทอร์เน็ต โพรโทคอล HTTP กลไกจัดการการร้องขอในเว็บเซิร์ฟเวอร์ การเขียนโปรแกรมแบบ CGI และการสร้างหน้าเว็บแบบพลวัต การใช้งานคุกกี้ การติดต่อเชื่อมโยงกับฐานข้อมูล การปรับแต่งประสิทธิภาพและความปลอดภัยในการใช้งาน การเขียนโปรแกรมในฝั่งของ Client เช่น HTML, CSS, JavaScript และ ฝั่งของ Server เช่น PHP, ASP, JSP, AJAX เป็นต้น

Study and practice of an Internet technology and how the computers are communication between client and server take place using the HTTP. We will explore the basic mechanisms and concepts of process of web servers such as how to use cookies in JavaScript, database and security management for web developers. Students will gain hands-on experience with tools and techniques that simplify the creation and maintenance of dynamic web page, client program: HTML, CSS and JavaScript, and server program such as CGI, PHP, ASP, JSP, and AJAX.

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมบนเว็บ

สัปดาห์	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
1	บทนำและสถาปัตยกรรมเว็บ	<ul style="list-style-type: none"> • ปฏิบัติการ: ติดตั้งเครื่องมือพัฒนาและใช้งาน Git/GitHub
2	การสร้างโครงสร้างเอกสารเว็บด้วย HTML5	<ul style="list-style-type: none"> • ปฏิบัติการ: สร้างเว็บเพจด้วย Semantic HTML • การบ้าน: จัดทำเอกสารเว็บเพจ Resume
3	การจัดรูปแบบและการออกแบบเชิงตอบสนอง (CSS3)	<ul style="list-style-type: none"> • ปฏิบัติการ: สร้าง Responsive Layout ด้วย Flexbox และ Grid • การบ้าน: พัฒนาเว็บเพจ Portfolio ที่รองรับทุกอุปกรณ์
4	การเขียนโปรแกรมเชิงโต้ตอบด้วย JavaScript (ES6+)	<ul style="list-style-type: none"> • ปฏิบัติการ: การเข้าถึงและจัดการ DOM • การบ้าน: พัฒนาแบบฟอร์มพร้อมการตรวจสอบความถูกต้องของข้อมูล
5	การพัฒนาโปรแกรมด้วย TypeScript	<ul style="list-style-type: none"> • ปฏิบัติการ: การแปลงโปรแกรม JavaScript ให้เป็น TypeScript • การบ้าน: ออกแบบ Class และ Interface ด้วย TypeScript

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมบนเว็บ (ต่อ)

สัปดาห์	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
6	การวัดผลการเรียนรู้ที่ 1 (Frontend พื้นฐาน)	<ul style="list-style-type: none"> • การทดสอบย่อยครั้งที่ 1 (ประเมินความเข้าใจ HTML, CSS, JS, และ TS)
7	สถาปัตยกรรมคอมโพเนนต์ด้วย React	<ul style="list-style-type: none"> • ปฏิบัติการ: การสร้างคอมโพเนนต์และการส่งผ่านข้อมูล (Props) • การบ้าน: พัฒนาหน้าเว็บ Portfolio ด้วยสถาปัตยกรรมแบบคอมโพเนนต์
8	การจัดการสถานะและวงจรชีวิตใน React	<ul style="list-style-type: none"> • ปฏิบัติการ: การจัดการสถานะ (State) และผลข้างเคียง (Side Effects) ด้วย Hooks • การบ้าน: พัฒนาแอปพลิเคชัน To-Do List
9	สอบกลางภาค	<ul style="list-style-type: none"> • ดำเนินการสอบกลางภาค (ภาคทฤษฎีหรือภาคปฏิบัติตามความเหมาะสม)
10	การพัฒนาโปรแกรมฝั่งเซิร์ฟเวอร์ด้วย Node.js	<ul style="list-style-type: none"> • ปฏิบัติการ: การสร้างเว็บเซิร์ฟเวอร์ด้วย Node.js และ Express.js • การบ้าน: พัฒนาเว็บเซิร์ฟเวอร์ที่มีการจัดการเส้นทางเบื้องต้น
11	การพัฒนา RESTful API ด้วย Express.js	<ul style="list-style-type: none"> • ปฏิบัติการ: การพัฒนา RESTful API สำหรับการจัดการข้อมูล (CRUD) • การบ้าน: ออกแบบและพัฒนา CRUD API
12	การเชื่อมต่อฐานข้อมูล	<ul style="list-style-type: none"> • ปฏิบัติการ: การบูรณาการฐานข้อมูลเข้ากับ CRUD API
13	การยืนยันตัวตนและความปลอดภัย	<ul style="list-style-type: none"> • ปฏิบัติการ: การพัฒนากลไกการลงทะเบียนและเข้าสู่ระบบด้วย JWT • การบ้าน: ประยุกต์ใช้ระบบ Authentication กับ CRUD API
14	การวัดผลการเรียนรู้ที่ 2 (Backend)	<ul style="list-style-type: none"> • การทดสอบย่อยครั้งที่ 2 (ประเมินความเข้าใจ Backend ทั้งระบบ)
15	การบูรณาการระบบ Full-Stack	<ul style="list-style-type: none"> • ปฏิบัติการเชิงบูรณาการ: การเชื่อมต่อระบบ Frontend (React) และ Backend (Express API)
16	การควบคุมเวอร์ชันและการติดตั้งโปรแกรม	<ul style="list-style-type: none"> • ปฏิบัติการ: การติดตั้ง Full-Stack Application บนระบบออนไลน์ • การบ้าน: ส่งมอบลิงก์ของเว็บแอปพลิเคชันที่ติดตั้งบนระบบออนไลน์

ตารางที่ 1 แผนการสอนรายวิชา การเขียนโปรแกรมบนเว็บ (ต่อ)

สัปดาห์	หัวข้อการเรียนการสอน	กิจกรรม / การบ้าน / การประเมินผล
17	การทบทวนและสรุปผล	• กิจกรรม: การอภิปรายและตอบข้อซักถาม, การฝึกปฏิบัติแก้โจทย์ปัญหารวม
18	สอบปลายภาค	• ดำเนินการสอบปลายภาค (ภาคทฤษฎีหรือภาคปฏิบัติตามความเหมาะสม)

ตารางที่ 2 การประเมินผลรายวิชา การเขียนโปรแกรมบนเว็บ

องค์ประกอบการประเมินผล	รายละเอียดการประเมินผล	สัดส่วน (%)
งานเดี่ยวและแบบฝึกหัด	• การส่งงานที่ได้รับมอบหมายในแต่ละสัปดาห์ เช่น การพัฒนาเว็บเพจ Portfolio, แบบฟอร์ม, และ CRUD API • พิจารณาความสมบูรณ์ของโค้ดและความตรงต่อเวลา	20%
แบบทดสอบย่อย (2 ครั้ง)	• ครั้งที่ 1: ประเมินความเข้าใจเนื้อหา Frontend พื้นฐาน (HTML, CSS, JS, TypeScript) • ครั้งที่ 2: ประเมินความเข้าใจเนื้อหา Backend (Node.js, API, Database, Authentication)	20%
สอบกลางภาค	• การสอบภาคทฤษฎีหรือภาคปฏิบัติ เพื่อประเมินความเข้าใจเนื้อหาครึ่งแรกของรายวิชา	25%
สอบปลายภาค	• การสอบภาคทฤษฎีหรือภาคปฏิบัติ เพื่อประเมินความเข้าใจเนื้อหาครึ่งหลังของรายวิชา	25%
คุณลักษณะนิสัยและวินัยในการเรียนรู้	• การเข้าชั้นเรียนตรงเวลาและสม่ำเสมอ • ความรับผิดชอบในการส่งงาน • การมีส่วนร่วมในชั้นเรียนและการทำงานกลุ่ม • ความมีวินัยและความรับผิดชอบต่อตนเองและผู้อื่น	10%
รวม		100%