

รหัสวิชา: ENGSE611

หน่วยกิต: 3 (1-4-4)

มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา



การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

MODERN WEB TECHNOLOGY DEVELOPMENT

บทนำสู่เทคโนโลยีเว็บ (Web Technology)

ประเภทวิชา: กลุ่มวิชาชีพเลือก

อาจารย์ผู้สอน

อาจารย์นริศ กำแพงแก้ว



E-mail : naris@rmutl.ac.th



Tel : 091-7915355



รหัสวิชา: ENGSE611

การพัฒนาเว็บด้วยเทคโนโลยีสมัยใหม่

หัวข้อ :

บทนำสู่เทคโนโลยีเว็บ

Web Technology

วัตถุประสงค์การเรียนรู้ (Learning Objectives)

1. อธิบายความหมายและองค์ประกอบพื้นฐานของเทคโนโลยีเว็บได้
2. เข้าใจโครงสร้างการทำงานของระบบเว็บแบบ Client–Server ได้
3. แยกความแตกต่างระหว่าง Frontend และ Backend ได้
4. ระบุเครื่องมือและเทคโนโลยีที่ใช้ในการพัฒนาเว็บได้อย่างถูกต้อง
5. แสดงความเข้าใจเกี่ยวกับกระบวนการพัฒนาเว็บไซต์ในภาพรวมได้

ในทุกๆ วัน เราต่างมีปฏิสัมพันธ์กับเว็บเทคโนโลยีอยู่ตลอดเวลา ไม่ว่าเราจะรู้ตัวหรือไม่

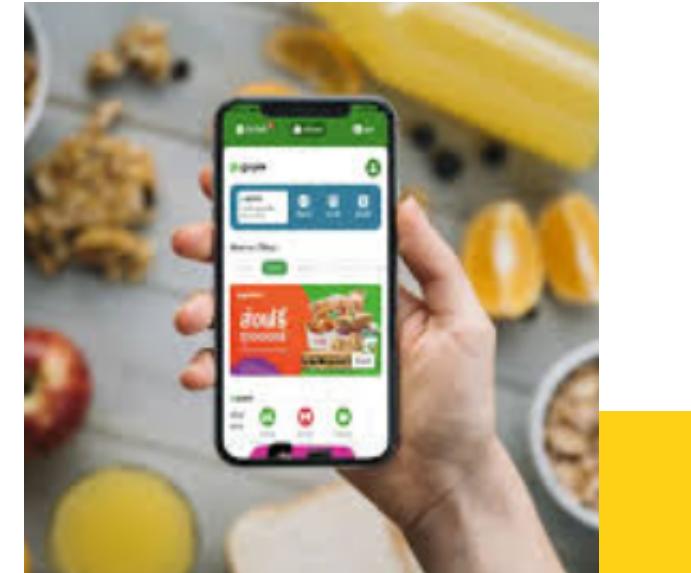
ชีวิตประจำวันกับ Website



เช็ค Social Media



ค้นหาข้อมูลใน Google



สั่งอาหารออนไลน์



ดูวิดีโอ YouTube

การไลฟ์ Facebook,
Instagram, หรือ TikTok
คือการถ่ายทอดสดจาก Server
มาแสดงผลผ่าน API

คุณกำลังส่งคำขอ
(Request) ไปยัง Server
ของ Google เพื่อให้ประมวล
ผลและส่งผลลัพธ์กลับมา

แอปพลิเคชันเหล่านี้ใช้เว็บ
เทคโนโลยีในการเชื่อมต่อระหว่าง
คุณ, ร้านอาหาร, และคนขับ

เป็นการสตรีมข้อมูลวิดีโอจาก
Server マイอังกฤษกอนของ
คุณแบบเรียลไทม์

ทั้งหมดนี้คือ "เว็บเทคโนโลยี"... แล้วมันทำงานอย่างไรกันนะ?

Internet vs. Website



01

Internet คือ
โครงสร้างพื้นฐาน

เปรียบเสมือน "ถนน" หรือเครือข่ายที่เชื่อมต่อคอมพิวเตอร์ กันโลกเข้าด้วยกัน เป็นระบบทางกายภาพของสายเคเบิล, เราเตอร์, และเซิร์ฟเวอร์ที่ทำให้ข้อมูลเดินทางได้



02

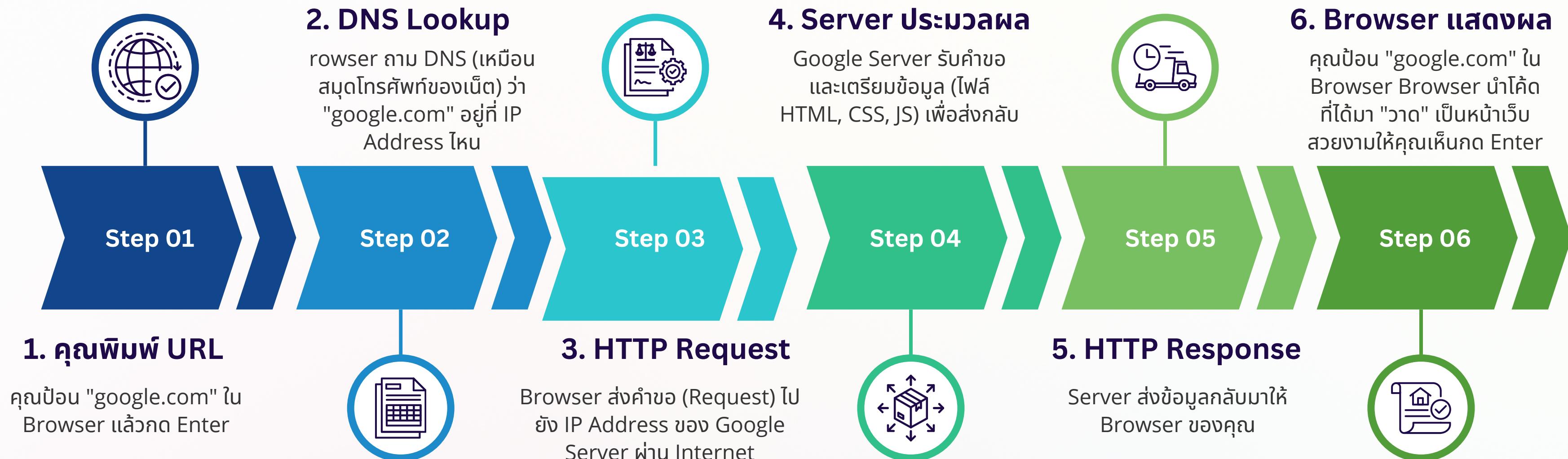
Website คือ
สิ่งที่อยู่บน Internet

เปรียบเสมือน "ร้านค้าหรือบ้าน" ที่ตั้งอยู่บนถนนเหล่านั้น เป็น ชุดของไฟล์ (HTML, CSS, รูปภาพ) ที่เก็บอยู่ในคอมพิวเตอร์ (Server) และเราสามารถเข้าถึงได้ผ่าน Internet

สรุป : Internet คือ ทางด่วน, Website คือ จุดหมายปลายทาง

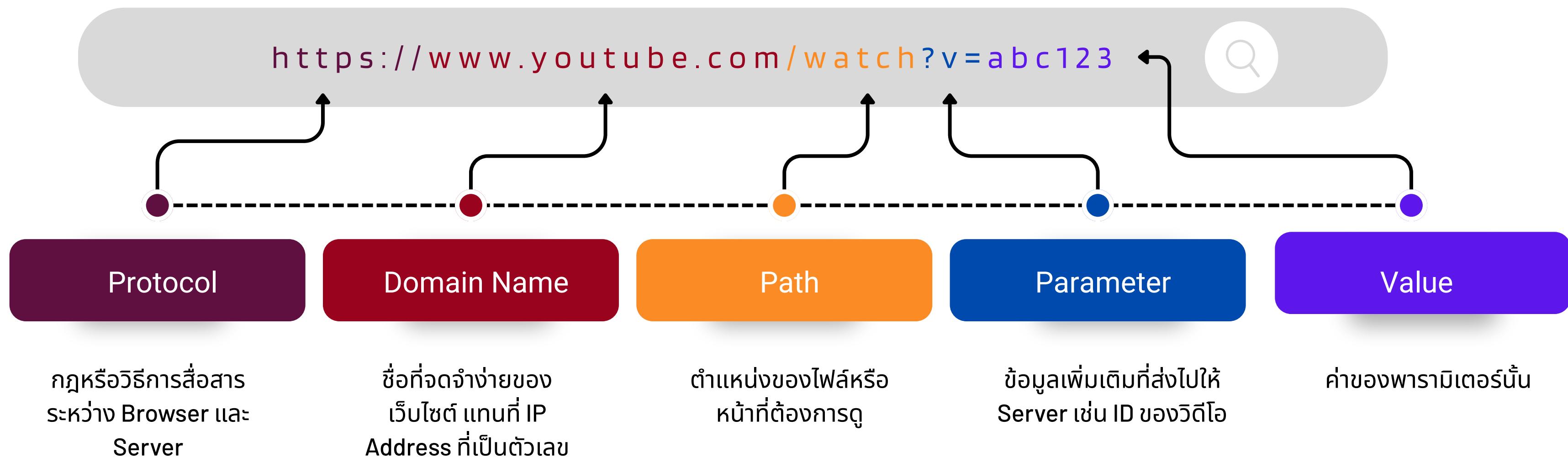
เมื่อพิมพ์ "google.com" เกิดอะไรขึ้น?

เบื้องหลังการแสดงผลเว็บเพจในไม้กีวีนาที มีขั้นตอนที่น่าทึ่งเกิดขึ้นมากมาย



URL คืออะไร? Address ของเว็บไซต์

URL (Uniform Resource Locator) คือที่อยู่ที่ไม่ซ้ำกันของทรัพยากรบนอินเทอร์เน็ต



HTTP vs HTTPS - ความปลอดภัยสำคัญมาก!



HTTP (เก่า) = โปสการ์ด

เปรียบเสมือนการส่ง "โปสการ์ด" ข้อมูลกั้งหนดที่ส่งเป็นตัวอักษรธรรมด้า (Plain Text) **ไม่ได้เข้ารหัส** หากมีผู้ไม่หวังดีดักจับข้อมูลระหว่างทาง ก็สามารถอ่านข้อมูลสำคัญของคุณได้ เช่น รหัสผ่าน, ข้อมูลส่วนตัว



เปรียบเสมือนการส่ง "จดหมายที่ปิดผนึกใบตู้เชฟ" ข้อมูลกั้งหนดจะถูก **เข้ารหัส (Encrypted)** ด้วย SSL/TLS certificate. แม้มีคนดักจับข้อมูลไป ก็จะไม่สามารถอ่านเนื้อหาข้างในได้ ทำให้ปลอดภัยสำหรับการทำธุกรรมทางการเงินหรือกรอกข้อมูลส่วนตัว

กฎทอง: เว็บไซต์ที่น่าเชื่อถือ ต้องใช้ HTTPS เสมอ!

Browser คือหน้าต่างสู่โลกออนไลน์

เว็บเบราว์เซอร์ไม่ใช่แค่โปรแกรมเปิดเน็ต แต่มันคือ นักแปลภาษา และ ศิลปิน ที่ชับช้อง

หน้าที่หลักของ Browser

1. ร้องขอและรับข้อมูล (Request & Response)

ทำหน้าที่เป็นตัวกลางส่งคำขอ (HTTP Request) ไปยัง Server และรับข้อมูล (HTTP Response) ซึ่งประกอบด้วยไฟล์โค้ดต่างๆ กลับมา

2. แปลภาษาและแสดงผล (Parsing & Rendering)

เมื่อได้รับโค้ด (HTML, CSS, JS) เบรราว์เซอร์จะทำการแปล (Parse) และ "วาด" (Render) ออกมาเป็นหน้าเว็บไซต์ที่มีปุ่ม, รูปภาพ, และข้อความสวยงามให้เราเห็นและโต้ตอบได้

เบราว์เซอร์ยอดนิยม:



chrome



Safari



Firefox



Microsoft Edge



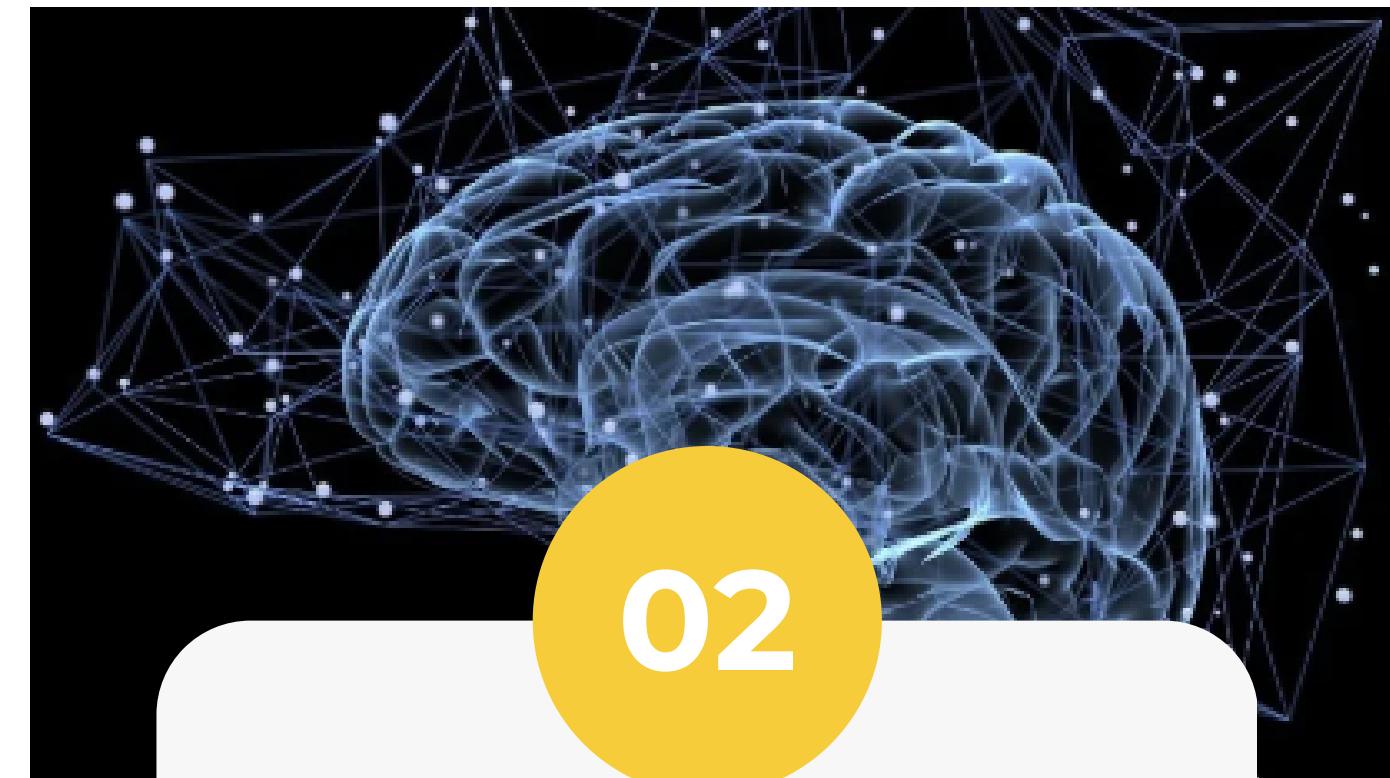
Server - ໂຮງໝານພລິຕເວີບໄຊຕ



01

ຖີ່ເກີບຂ້ອມູລ (Storage)

ເກີບໄຟລ໌ທີ່ໝາດຂອງເວີບໄຊຕ ໄນວ່າຈະເປັນໂຄດ
HTML, CSS, JavaScript, ຮູປກາພ, ວົດໂອ, ແລະ
ໜ້ານຂ້ອມູລຂອງຜູ້ໃຊ້ງານ



02

ຕັ້ງປະມາວລຸດ (Processing)

ຮອຮັບຄໍາຂອງຈາກເບຣາວເຊອຣ, ປະມາວລຸດຕາມຄໍາສັ່ງ
(ເຊັ່ນ ການດຶງຂ້ອມູລຈາກໜ້ານຂ້ອມູລ, ການຄຳນວນ),
ແລະເຕີຍແນວໃຈຂ້ອມູລເພື່ອສ່ວນກັບໄປ

ສຽງ : Server ຄື່ອ ມ້ອງສມຸດ ທີ່ Browser ຂອຍເມນັ້ງສື່ວ (ຂ້ອມູລ) ໄປໃຫ້ຄຸນວ່ານ

Port - ປະຕູສູ່ບົກາຕ່າງໆ

หาก IP Address ຄືອກື່ອຍູ້ຂອງອາຄາຣ, Port ກີ່ຄືອ ມາຍເລຂໍຫ້ອງຮັບ
ປະຕູ ກີ່ຮະບຸວ່າຕ້ອງກາຣຕິດຕ່ອບບົກາຕ່າງໆໃນອາຄາຣນີ້ນ

port
80

ບົກາ : **HTTP** (ໄມ່ເຂົາຮັສ)

ເປັນພອຣຕາມາຕຣ້ສ້ານທີ່ໃຊ້ໃນກາຣໃຫ້ບົກາຕີເວີບໃຫ້ຕົ້ວໄປ ໂດຍໄມ່ເຂົາຮັສ
ຂ້ອມູລ ເນື້ອຜູ້ໃຫ້ເປີດເວີບທີ່ຂຶ້ນຕົ້ນດ້ວຍ <http://> ຂ້ອມູລຈະຄຸກສ່າງຜ່ານ
ພອຣຕົ້ນໄດຍຕຽນ ທີ່ໜ່າຍກັບເວີບທີ່ໄປຕ້ອງໃຫ້ຂ້ອມູລສໍາຄັນ ເຊັ່ນ ເວີບ
ຂ່າວຮັບໂລກປະຈາກສັນພັນຮັນ

port
443

ບົກາ : **HTTPS** (ເຂົາຮັສດ້ວຍ TLS/SSL)

ເປັນພອຣຕາມາຕຣ້ສ້ານທີ່ໃຫ້ບົກາຕີເວີບໃຫ້ຕົ້ວທີ່ມີກາຣ ເຂົາຮັສຂ້ອມູລດ້ວຍ **TLS/SSL**
ໃຊ້ໃນເວີບທີ່ຕ້ອງປົກປ້ອງຂ້ອມູລຜູ້ໃຫ້ ເຊັ່ນ ກາຣເຂົາສູ່ຮະບບ ທໍາຮະເງີນ
ຮັບໂລກປະຈາກສັນພັນຮັນ ທີ່ **HTTPS** ຊ່ວຍໃຫ້ຂ້ອມູລຮະຫວ່າງຜູ້ໃຫ້ກັບ
ເຊີຣົ່ງເວີບປະຈາກກາຣດັກຝັງແລະປລອມແປລ່ງ

ໂດຍປົກຕົວແລ້ວ Browser ຈະເລືອກໃຫ້ Port 80 ອີ່ຈຳກັດໃຫ້ເວີບໃນ URL

port
21

ບົກາ : **FTP (File Transfer)**

ໃຊ້ສໍາຮັບ ກາຣຮັບ-ສ່າງໄຟຣ ຮະຫວ່າງເຄື່ອງຜູ້ໃຫ້ກັບເຊີຣົ່ງເວີບ ເຊັ່ນ ກາຣ
ອັປໂລດໄຟຣເວີບໃຫ້ຕົ້ວທີ່ໄປຈະປັບປຸງຜູ້ໃຫ້ແລະຮັສຜ່ານໃນກາຣ
ເຊື່ອມຕ່ວ

port
25/587

ບົກາ : **SMTP (Email)**

25 : ໃໃຊ້ສ່າງອື່ເມລະຫວ່າງເຊີຣົ່ງເວີບ ເຊັ່ນ **Gmail** ສ່າງລົງ **Outlook**
ຮັບໂລກປະຈາກສັນພັນຮັນ ປະຫວັດ
587 : ໃໃຊ້ສໍາຮັບຜູ້ໃຫ້ຕົ້ວໄປສ່າງອື່ເມລອອກຜ່ານແອປ ເຊັ່ນ **Outlook**
ຮັບໂລກປະຈາກສັນພັນຮັນ ໂດຍຮອງຮັບກາຣເຂົາຮັສ **STARTTLS** ເພື່ອຄວາມ
ປລອດກັບ

FRONTEND VS. BACKEND - ส่องด้านของเหรียญ



Frontend (Client-Side)

คือทุกสิ่งที่ผู้ใช้งานเห็นและต้องได้
บนหน้าจอ เปรียบเสมือน "หน้าร้าน"
ที่ตกแต่งสวยงามและมีพนักงานคอย
ต้อนรับ

- หน้าที่: แสดงผล UI, รับ Input จากผู้ใช้
- เทคโนโลยี: HTML, CSS, JavaScript (React)



Backend (Server-Side)

คือระบบที่ทำงานอยู่เบื้องหลัง ผู้ใช้
มองไม่เห็น เปรียบเสมือน "ห้องครัว
และคลังสินค้า" ที่จัดการวัตถุดิบและ
ปรุงอาหาร

- หน้าที่: จัดการฐานข้อมูล,
ประมวลผล Logic, ดูแล
ความปลอดภัย
- เทคโนโลยี: Node.js,
Python, Java, PHP

สิ่งที่เราจะเรียนในวิชา ENGSE611

เส้นทางการเรียนรู้ของเราจะครอบคลุมก้าวะที่จำเป็นสำหรับการเป็น Full-Stack Developer

Frontend Development

สร้างหน้าตาเว็บไซต์ที่สวยงามและตอบสนองต่อผู้ใช้ได้อย่างรวดเร็ว

- HTML: โครงสร้างของเว็บ
- CSS: การตกแต่งและจัดวาง Layout
- JavaScript: การสร้างปฏิสัมพันธ์กับผู้ใช้
- React.js: Library สำหรับสร้าง UI ที่ซับซ้อนและมีประสิทธิภาพ

Backend Development

พัฒนาระบบที่ทำงานหลังเพื่อจัดการข้อมูลและ Logic ของแอปพลิเคชัน

- Node.js: ทำให้ JavaScript สามารถทำงานบน Server ได้
- Express.js: Framework ที่ช่วยให้การสร้างเว็บเซิร์ฟเวอร์และ API เป็นเรื่องง่าย
- Database: การเชื่อมต่อและจัดการฐานข้อมูลเพื่อเก็บข้อมูลอย่างถาวร

การเลือกเครื่องมือที่เหมาะสมจะช่วยให้การพัฒนาซอฟต์แวร์ราบรื่นและมีประสิทธิภาพ

เครื่องมือที่ควรใช้



Visual Studio Code

Code Editor

โปรแกรมสำหรับเขียนโค้ด ควร
มีฟีเจอร์ช่วยอ่านวิเคราะห์ความลึก
เช่น Syntax Highlighting,
IntelliSense



Web Browser

ใช้สำหรับทดสอบและ Debug
โปรแกรมที่เราเขียน มีเครื่องมือ^{สำหรับนักพัฒนา (Developer Tools)} ที่ทรงพลัง

แนะนำ: Visual Studio Code
(VS Code)



Version Control

ระบบที่ช่วยติดตามการ
เปลี่ยนแปลงของโค้ด ทำให้
สามารถย้อนกลับ, ทำงานร่วม
กับผู้อื่นได้ง่าย

แนะนำ: Git และ GitHub



Runtime Environment

โปรแกรมที่จำเป็นสำหรับการ
รันโค้ด JavaScript ฝั่ง
Backend ซึ่งเราจะใช้ในช่วง
ครึ่งหลังของคอร์ส

แนะนำ: Node.js

ความปลอดภัยเว็บไซต์ - เรื่องสำคัญ!

ภัยคุกคามที่พบบ่อย

- SQL Injection:** แฮกเกอร์พยายามแทรกคำสั่ง SQL ที่เป็นอันตรายผ่านฟอร์มเพื่อบอกโมยหรือทำลายข้อมูลในฐานข้อมูล
- Cross-Site Scripting (XSS):** การฝังสคริปต์ที่เป็นอันตรายลงในหน้าเว็บ เพื่อบอกโมยข้อมูลของผู้ใช้คนอื่นที่เข้ามาดูหน้านั้น
- Password Attacks:** การพยายามเดารหัสผ่านด้วยวิธีต่างๆ เช่น Brute Force (สุ่มไปเรื่อยๆ) หรือใช้ Dictionary

วิธีป้องกันพื้นฐาน

- ใช้ HTTPS เสมอ:** เพื่อเข้ารหัสข้อมูลกึ่งหมดที่ส่งผ่านเครือข่าย
- Validate Input:** "อย่าไว้ใจข้อมูลจากผู้ใช้" ตรวจสอบและกรองข้อมูลทุกอย่างที่ผู้ใช้ส่งเข้ามา
- Hashing Passwords:** อย่าเก็บรหัสผ่านเป็นข้อความธรรมดา แต่ให้แปลง (Hash) เป็นข้อความที่อ่านไม่ออกรก่อนบันทึก



Developer Tools - เครื่องมือลับของโปรแกรมเมอร์

ทุกเบราว์เซอร์มีเครื่องมือสำหรับนักพัฒนา (DevTools) ติดตั้งมาให้ ซึ่งเป็นเหมือนมีดพกสวัสดาร์ Web Developer

วิธีเปิด: กด **Ctrl+Shift+I** หรือ **Cmd+Option+I** (Mac)

Elements Tab

ดูและแก้ไขโค้ด HTML และ CSS ของหน้าเว็บได้แบบส่วนๆ แนะนำสำหรับการทดลองปรับแก้ Layout หรือสี

Console Tab

เหมือนห้องทดลอง JavaScript สามารถดูข้อความ Error, Log ข้อมูล, และรันคำสั่ง JavaScript ได้โดยตรง

Network Tab

ติดตามดูการรับ-ส่งข้อมูล กั้งหนดระหว่าง Browser และ Server ดูว่าไฟล์ไหนโหลดช้า หรือ API ทำงานถูกต้องหรือไม่

Device Toolbar

จำลองการแสดงผลบนอุปกรณ์ต่างๆ เช่น มือถือ, แท็บเล็ต เพื่อทดสอบ Responsive Design ได้อย่างรวดเร็ว

การจัดการไฟล์โปรเจค

การจัดโครงสร้างโฟลเดอร์ที่ดีตั้งแต่เริ่มต้น จะช่วยให้โปรเจคสามารถเติบโตและบำรุงรักษาได้ง่ายในอนาคต

โครงสร้างโฟลเดอร์ที่แนะนำ

```
my-website/
  index.html -- หน้าหลักของเว็บไซต์
  css/ -- โฟลเดอร์สำหรับไฟล์ CSS ทั้งหมด
    style.css
  js/ -- โฟลเดอร์สำหรับไฟล์ JavaScript
    script.js
  images/ -- โฟลเดอร์สำหรับรูปภาพ
    logo.png
  pages/ -- สำหรับหน้าอื่นๆ (ถ้ามี)
    about.html
```

หลักการตั้งชื่อไฟล์และโฟลเดอร์

- ใช้ตัวพิมพ์เล็ก กึ่งหนด (lowercase)
- ห้ามเว้นวรค ให้ใช้ kebab-case (เช่น about-us.html) แทน
- ใช้ภาษาอังกฤษ เพื่อความเป็นสากล



ระบบงานของ WEB DEVELOPER

การพัฒนาเว็บไม่ใช่แค่การเขียนโค้ด แต่เป็นกระบวนการที่มีขั้นตอนชัดเจน

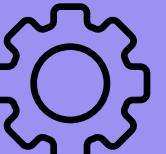
01



Planning (วางแผน)

กำหนดความเข้าใจความต้องการ, ออกแบบ Wireframe/UI, และเลือกเทคโนโลยีที่เหมาะสม

02



Development (พัฒนา)

ลงมือเขียนโค้ด Frontend และ Backend, ทดสอบการทำงานของแต่ละส่วน

03



Deployment (เผยแพร่)

นำโค้ดที่เสร็จแล้วขึ้นไปไว้บน Server จริง, ตั้งค่า Domain Name เพื่อให้คนทั่วไปเข้าใช้งานได้

04



Maintenance (บำรุงรักษา)

หลังจากเปิดใช้งาน, ต้องดูแล, อัปเดต, แก้ไขข้อผิดพลาด, และเพิ่มฟีเจอร์ใหม่ๆ ตามความต้องการ

เว็บไซต์ที่ดีต้องมีอะไรบ้าง?

การสร้างเว็บไซต์ที่ยอดเยี่ยมต้องอาศัยความสมดุลขององค์ประกอบ 4 ด้าน



ใช้งานง่าย (Usability)

ทำความเข้าใจความต้องการ, ออกรูปแบบ Wireframe/UI, และเลือกเทคโนโลยีที่เหมาะสม

01



ออกแบบสวย (Design)

ลงมือเขียนโค้ด Frontend และ Backend, ทดสอบการทำงานของแต่ละส่วน

02



ทำงานได้ (Functionality)

นำโค้ดที่เสร็จแล้วขึ้นไปไว้บน Server จริง, ตั้งค่า Domain Name เพื่อให้คนทั่วไปเข้าใช้งานได้

03



ปลอดภัย (Security)

หลังจากเปิดใช้งาน, ต้องดูแล, อัปเดต, แก้ไขข้อผิดพลาด, และเพิ่มฟีเจอร์ใหม่ๆ ตามความต้องการ

04

มาทำความรู้จักกับคำศัพท์เหล่านี้ให้คุ้นเคย เพราะเราจะได้เจอตตลอดเส้นทางการเป็นนักพัฒนาเว็บ

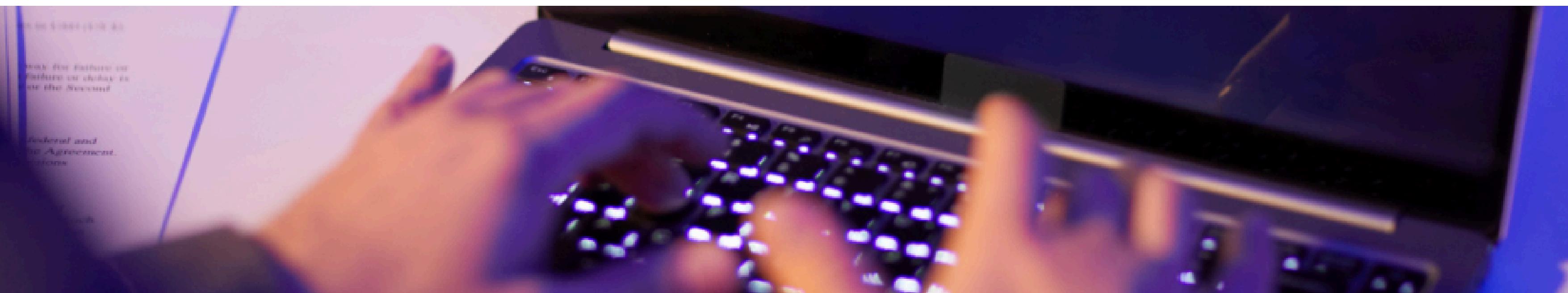
คำศัพท์สำคัญที่ต้องจำ

Technical Concepts

- Responsive:** การออกแบบที่ทำให้เว็บแสดงผลได้ดีบนทุกขนาดหน้าจอ (มือถือ, คอมพิวเตอร์)
- API (Application Programming Interface):** ช่องทางที่ซอฟต์แวร์ใช้คุยกันเพื่อแลกเปลี่ยนข้อมูล
- UX/UI:** ประสบการณ์ผู้ใช้ (UX) และหน้าตาส่วนติดต่อผู้ใช้ (UI)

File & Data Formats

- HTML (HyperText Markup Language):** ภาษาหลักสำหรับสร้างโครงสร้างเว็บ
- CSS (Cascading Style Sheets):** ภาษาสำหรับตกแต่งหน้าเว็บ
- JS (JavaScript):** ภาษาโปรแกรมสำหรับสร้างการโต้ตอบบนเว็บ
- JSON (JavaScript Object Notation):** รูปแบบมาตรฐานในการรับ-ส่งข้อมูลผ่าน API



เทรนด์ Web Development 2025+

โลกของเว็บเปลี่ยนแปลงเร็วเสมอ นี่คือเทคโนโลยีและแนวโน้มที่กำลังมาแรงและน่าจับตามอง

เทคโนโลยียอดนิยม

- React & Next.js:** ยังคงเป็นผู้นำในฝั่ง Frontend สำหรับการสร้างเว็บแอปที่ซับซ้อนและมีประสิทธิภาพสูง
- TypeScript:** ได้รับความนิยมมากขึ้นเรื่อยๆ เพราะช่วยลดข้อผิดพลาดในการเขียน JavaScript ขนาดใหญ่
- AI Integration:** การนำ AI มาใช้ในเว็บมีหลากหลายรูปแบบ ตั้งแต่ Chatbot, ระบบแนะนำ, ไปจนถึงการช่วยเขียนโค้ด (Copilot)

แนวโน้มการพัฒนา

- Mobile First Design:** เป็นมาตรฐานไปแล้ว การออกแบบโดยคิดถึงผู้ใช้มือถือก่อนเป็นอันดับแรก
- Progressive Web Apps (PWA):** การทำให้เว็บไซต์ทำงานได้เหมือนแอปพลิเคชันแท้ๆ เช่น การทำงานออฟไลน์, Push Notification
- Focus on Performance:** ความเร็วในการโหลดเว็บมีความสำคัญอย่างยิ่งต่อ UX และ SEO