

Lab — สร้างเว็บไซต์หลักสุดด้วย Codex แบบ step by step

ภาพและ transcript ใน walkthrough นี้เก็บ snapshot เดิม 11 บทเรียนไว้ตามการทดลองจริง ปัจจุบันคลังคอร์สเพิ่ม un 11 — บริการ Cloudflare เป็นบทที่ 12 และมี Lab RW Web Skills สำหรับฝึก workflow เพิ่ม

Lab นี้พาผู้เรียนสร้างเว็บไซต์ตัวอย่างสำหรับ **AI Web Studio — ออกแบบและพัฒนาเว็บไซต์ด้วย AI** ซึ่งเป็นหน่วย workshop ส่วนหนึ่งของ **977-121 Module: Website Design and Development**, มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต (Prince of Songkla University, Phuket Campus) ตั้งแต่ข้อมูลตั้งต้นจนเป็นหน้าเว็บที่ตรวจได้ ชื่อ module และสถาบันตรวจเทียบได้จาก หน้าหลักสูตร Digital Engineering ของวิทยาลัยการคอมพิวเตอร์ ม.อ. ภูเก็ต

ตัวอย่างนี้เป็น **สื่อประกอบการเรียน** ไม่ใช่เว็บไซต์หรือบริการทางการของมหาวิทยาลัย ไม่สร้างตรามหาวิทยาลัย ชื่อผู้สอน จำนวนหน่วยกิต เกณฑ์คะแนน ตารางภาคเรียน ช่องทางสมัคร หรือระบบ login ขึ้นเอง เวลา **2 วัน × 4 ชั่วโมง รวม 8 ชั่วโมงและพักวันละ 10 นาที** หมายถึง workshop ใน repository นี้ ไม่ใช่ชั่วโมงทั้งหมดหรือคำอธิบายรายวิชาฉบับมหาวิทยาลัย

เวลาทำ Lab เดิมประมาณ **160–180 นาทีสำหรับเรียนต่อด้วยตัวเอง** ผู้สอนสามารถเลือกสาริตบางช่วงใน workshop โดยไม่เพิ่มเวลารวม 8 ชั่วโมง เป้าหมายไม่ใช่ให้ Codex สร้างทุกอย่างในคำสั่งเดียว แต่ให้เห็นว่าแต่ละ prompt เปลี่ยน artifact ใด และผู้เรียนตรวจสอบผลอย่างไรก่อนสั่งขั้นถัดไป

ผลงานเมื่อจบ Lab

ผู้เรียนจะมี:

- source inventory ที่แยก FACT, PROPOSAL และ GAP
- website brief ที่อ้างกลับไปยังข้อมูลตั้งต้น
- semantic HTML ก่อนแต่งภาพ
- responsive Light/Dark design ตามข้อกำหนด Rawinnipa สำหรับสื่อหลักสูตร
- JavaScript สำหรับค้นหาและกรองบทเรียน Day 1 / Day 2 / ทั้งหมด
- QA record พร้อม prompt แก่ไขรอบสุดท้าย
- release packet ที่มี source, prompt, transcript และภาพแต่ละช่วง

เปิด เว็บไซต์ตัวอย่างฉบับสุดท้าย และใช้หน้านี้อ่านลำดับการสร้าง อย่างดูเฉพาะภาพสุดท้าย เพราะหลักฐานสำคัญอยู่ที่ prompt, diff และจุดตรวจของแต่ละรอบ

แผนเวลา

ช่วง	เวลาแนะนำ	ผลที่ต้องได้ก่อนผ่าน
0. เตรียม workspace และอ่านข้อมูล	15 นาที	source inventory และข้อห้ามแต่งข้อมูล
1. สร้าง brief	20 นาที	brief ที่มี audience, goal, sections, acceptance criteria และ GAP
2. สร้าง semantic HTML	25 นาที	เนื้อหาและลิงก์ใช้ได้โดยยังไม่พึ่ง CSS/JS
3. ออกแบบ Light/Dark และ responsive	35 นาที	layout 390px/desktop, Light default, Dark toggle ที่อ่านชัด
4. เพิ่มการค้นหาและตัวกรอง	25 นาที	keyboard ใช้ได้และไม่มีผลลวงเมื่อไม่พบรายการ
5. ตรวจ QA และแก้แบบเป็นรอบ	25-40 นาที	evidence ของปัญหา การแก้ และการตรวจซ้ำ
6. เตรียมเผยแพร่และส่งมอบ	15-20 นาที	ไฟล์ครบ ลิงก์ถูก และแยก local/dry-run/public evidence

Artifact ที่ใช้ติดตามการทำงาน

Artifact	หน้าที่
inputs.md	ข้อมูลที่ Codex ได้รับจริง พร้อม source ID
scope.txt	ขอบเขตไฟล์ที่แก้ได้ สิ่งห้ามแต่ง และ stop condition ของการรันแยก
prompts/01-brief.txt ถึง 05-review.txt	prompt ที่ส่งจริงในแต่ละรอบ
prompts/06-handoff.txt	ปิด GAP ที่ได้รับคำตอบและสรุปผลตรวจเพื่อส่งต่อ
prompts/07-fix-controls.txt	แก้ control boundary จากการตรวจและอัปเดต regression check
transcripts/	บันทึกข้อความสั่งและเหตุการณ์ CLI ที่คิดมาเพื่อการสอน
stages/02/ , stages/03/	snapshot ของเว็บระหว่างทาง

Artifact	หน้าที่
/examples/course-website/	source และผลสุดท้ายของตัวอย่าง

ภาพ Terminal ใน Lab แสดง **Codex CLI ผ่าน terminal viewer แบบอ่านอย่างเดียว** ไม่ใช่หน้าต่าง Codex Desktop ส่วนภาพ browser เป็นผลของหน้าที่ preview จริง Transcript และภาพมีไว้ช่วยตรวจสอบย้อนหลัง แต่ไม่แทนการอ่านไฟล์หรือรัน test

ตัวอย่างใช้ `codex exec --json` เพื่อเก็บ event stream เป็น JSON Lines (JSONL) แล้วเลือกเฉพาะเหตุการณ์ที่ช่วยอธิบายการทำงานมาแสดง เอกสารทางการระบุว่า stream นี้อาจมี event ของข้อความ agent, command, file change และ tool call; ผู้เรียนจึงต้อง redact path หรือข้อมูลลับก่อนเผยแพร่ ดู Codex non-interactive mode การมี JSONL ยืนยันว่า event ถูกบันทึก แต่ไม่ยืนยันว่าโค้ดหรือหน้าเว็บถูกต้อง

เริ่มจากศูนย์: ดาวน์โหลด เปิด Terminal และเตรียมพื้นที่ฝึก

- เปิดหน้า Resources ของหลักสูตร แล้วดาวน์โหลด **Source ZIP**
- แตก ZIP จะได้โฟลเดอร์ `ai-web-studio/` จากนั้นเปิด Terminal ที่โฟลเดอร์นี้ ไม่ใช่ที่ `Downloads` หรือโฟลเดอร์อื่นนอก
- ตรวจสอบว่าเห็น `package.json` , `README.md` , `public/` และ `docs/` แล้วติดตั้ง dependency ตาม lockfile

ตัวอย่าง Bash บน Linux/macOS/WSL หลังแตก ZIP แล้ว:

```
cd /path/to/ai-web-studio
pwd
ls
npm ci
npm run setup
npm run build
```

`npm run setup` สร้าง local secret ใน `.dev.vars` ตาม README อย่าเปิด คัดลอก หรือถ่ายค่า secret ลงภาพ หากต้องการตรวจ baseline ให้เปิด server ด้วย:

```
npm run preview
```

เปิด `http://127.0.0.1:3320/` แล้วหยุดด้วย `Ctrl+C` เมื่อดูเสร็จ พอร์ต 3320 และ inspector 3322 เป็นพอร์ตที่ project ของเราใช้ตาม shared-infra; อย่าเปลี่ยนไปใช้พอร์ตของ project อื่น คำสั่งทางเลือก `npm run dev` จะรัน build แล้วเปิด preview เดียวกัน จึงใช้แทน `npm run build` ตามด้วย `npm run preview` ได้

สร้าง workspace แยกเพื่อไม่กับตัวอย่างสำเร็จ

กลับมาที่ root `ai-web-studio/` แล้วสร้าง `practice/psu-course/` คัดลอกเฉพาะ input snapshot ก่อน clarification และ scope เข้าไป:

```
mkdir -p practice/psu-course
cp public/labs/course-website-build/stages/01/inputs.md practice/psu-course/inputs.md
cp public/labs/course-website-build/stages/01/scope.txt practice/psu-course/scope.txt
ls practice/psu-course
```

ถ้าไม่ได้ใช้ Bash/WSL ให้สร้างไฟล์เตอร์และ Copy/Paste สองไฟล์นี้ด้วย file manager หรือ editor โดยรักษาชื่อ `inputs.md` และ `scope.txt` จุดสำคัญคือ Codex แก่ได้เฉพาะ workspace ฝึก และ source ตัวอย่างสุดท้ายใน `/examples/course-website/` ยังไม่ถูกกับ

เลือกวิธีส่ง prompt

วิธีโต้ตอบ เหมาะกับผู้เริ่มต้น: เปิด session ที่ workspace ฝึก แล้วคัดลอก prompt ของแต่ละขั้น จากลิงก์ในแต่ละขั้นวางที่ละชุด รอให้งานรอบหนึ่งจบและตรวจไฟล์ก่อนส่งรอบถัดไป

```
codex -C practice/psu-course
```

วิธี CLI/JSONL สำหรับผู้ที่ต้องการ transcript: ตัวอย่างนี้ส่ง prompt 01 ผ่าน stdin และบันทึก event stream โดยไม่ให้ Codex แตะ source ตัวอย่างสุดท้าย:

```
codex exec -C practice/psu-course \
  --sandbox workspace-write \
  --json - \
  < public/labs/course-website-build/prompts/01-brief.txt \
  > practice/psu-course/01-events.jsonl
```

คำสั่งนี้เป็นตัวอย่างรูปแบบหนึ่ง ไม่บังคับให้ผู้เรียนใช้ JSONL ทุกขั้น ผู้ใช้ interactive สามารถเก็บ prompt files กับภาพ Terminal แทนได้ ตรวจไฟล์ใน `practice/psu-course/` ทุกครั้งก่อนเดินต่อ และอย่าส่ง credential หรือเนื้อหา JSONL ที่ก่อกวนขึ้นเว็บโดยไม่ตรวจข้อมูล

0. เตรียมข้อมูลและขอบเขต — 15 นาที

อ่านตามลำดับ:

1. บทเรียน 00 — เตรียมเครื่องและพื้นฐานเว็บ เพื่อแยก source, build output และ URL ที่ browser เปิด

- 2. บทที่ 1 — พื้นฐาน Vibe Coding เพื่อกำหนดขอบเขต, evidence และ stop condition
- 3. Lab Research → Website brief → Brandbook เพื่อแยก FACT, INFERENCE, PROPOSAL และ GAP
- 4. บทที่ 3 — Specs และ Prompts เพื่อเขียน acceptance criteria ก่อนแก้ code

ข้อมูลตั้งต้นของตัวอย่างอยู่ใน `inputs.md` และขอบเขตอยู่ใน `scope.txt` ของ workspace แยก โดยมี source ID สำคัญ ดังนี้:

- **S1:** ชื่อ module และมหาวิทยาลัยที่ใช้ให้
- **S2:** ขอบเขต workshop 2 วัน × 4 ชั่วโมงจาก course data และคู่มือผู้สอน
- **S3:** excerpt ของกฎ Rawinnipa และ `DESIGN.md` ที่จำเป็นต่อสื่อหลักสูตร ไม่ได้แบบ `DESIGN.md` ทั้งไฟล์ให้การรับแยก
- **S4:** รายการบทเรียน 00–10 และ URL ที่มีจริง
- **S5:** ปลายทางหลัก เช่น resources, handbook และ labs
- **S6:** font assets ที่ repository มีอยู่

GAP ต้องคงเป็น GAP จนได้ข้อมูลจากเจ้าของ เช่น ชื่อผู้สอน หน่วยกิต เกณฑ์คะแนน ตารางภาคเรียน LMS และช่องทางสมัคร การมีพื้นที่ว่างบนหน้าเว็บไม่ใช่เหตุผลให้ AI แต่งข้อมูลเติม

การแบ่งบทเรียนเป็น **Day 1: 00–05** และ **Day 2: 06–10** ยังไม่มีใน brief รอบแรก จึงถูกบันทึกเป็น GAP แล้วผู้ควบคุมงานให้คำชี้แจงเพิ่มก่อนรอบ interaction นี้เป็นตัวอย่างสำคัญของการทำงานทีละขั้น: เมื่อข้อมูลไม่พอให้หยุดถามหรือเพิ่ม source ที่ตรวจได้ แล้วค่อยเปลี่ยน dataset อย่าให้ AI แบ่งวันเองและเขียนย้อนหลังว่าเป็นข้อมูลเดิม

ใช้ worksheet นี้ก่อนเปิด Codex:

คำถาม	คำตอบของผู้เรียน
ผู้ใช้หลักของ prototype คือใคร	
ผู้ใช้ควรทำอะไรได้หนึ่งอย่างหลังเข้าเว็บ	
ข้อความใดมี source รองรับ	
ข้อความใดเป็นเพียง PROPOSAL	
ข้อมูลใดห้ามสร้างเพิ่ม	
หน้าที่จะสร้างต้องเชื่อมไปที่ URL ใด	
จะพิสูจน์ว่า “เสร็จ” ด้วยอะไร	

ก่อนเริ่ม session ตรวจ working directory และไฟล์ที่อนุญาตให้แก้ไข แล้วบอก Codex ชัดเจนว่าไม่ให้ deploy, เพิ่ม dependency หรือแก้ส่วนอื่นของ repository ในรอบ brief

1. ให้ Codex สร้าง website brief — 20 นาที

รอบแรกยังไม่สร้าง HTML ให้ Codex อ่าน inputs.md และ scope.txt ใน workspace แยก แล้วเขียน brief ที่แยกข้อมูลยืนยันแล้วออกจากข้อเสนอ กฎการออกแบบที่ต้องใช้มีอยู่ใน excerpt S3; agent ไม่ได้อ่าน DESIGN.md ทั้งไฟล์ในรอบนี้ จุดนี้ตรงกับ บทที่ 3 และต่อจาก evidence workflow ใน Research-to-Brand Lab

Prompt 01 ที่ส่งจริง — brief

อ่าน inputs.md และ scope.txt แล้วสร้าง brief.md กับ evidence.csv เท่านั้น ยังไม่สร้างหน้าเว็บ
brief: เป้าหมาย audience ที่ติดป้ายข้อเสนอ CTA หลัก/รองและปลายทาง ลำดับsection content inventory states light/dark responsive accessibility acceptance checks และ GAP ที่ต้องถามเจ้าของ
ยึดชื่อรายวิชาและ PSU Phuket ตามข้อมูล; workshop 2วันx4ชั่วโมงเป็นเพียงส่วนหนึ่งของmodule. แยก
FACT/INFERENCE/PROPOSAL/GAP และ source IDs S1-S6 ใน evidence.csv
อธิบายทางเลือกภาพ2แบบและเลือกหนึ่งเป็น draft โดยยึดแบรนด์เดิม ไม่สร้างตรามหาวิทยาลัยหรือข้อมูลผู้สอน/หน่วยกิต
ไม่รันweb researchในรอบนี้ รายงานว่าเป็นการสังเคราะห์ข้อมูลที่แนบ

```
CODEX CLI 0.153.4 / STEP 01 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 01-final.txt - < prompts/01-brief.txt

PROMPT SENT
อ่าน inputs.md และ scope.txt แล้วสร้าง brief.md กับ evidence.csv เท่านั้น ยังไม่สร้างหน้าเว็บ
brief: เป้าหมาย audience ที่ติดป้ายข้อเสนอ CTA หลัก/รองและปลายทาง ลำดับsection content inventory states light/dark responsive accessibility acceptance checks และ GAP ที่ต้องถาม
เจ้าของ
ยึดชื่อรายวิชาและ PSU Phuket ตามข้อมูล; workshop 2วันx4ชั่วโมงเป็นเพียงส่วนหนึ่งของmodule. แยก FACT/INFERENCE/PROPOSAL/GAP และ source IDs S1-S6 ใน evidence.csv
อธิบายทางเลือกภาพ2แบบและเลือกหนึ่งเป็น draft โดยยึดแบรนด์เดิม ไม่สร้างตรามหาวิทยาลัยหรือข้อมูลผู้สอน/หน่วยกิต
ไม่รันweb researchในรอบนี้ รายงานว่าเป็นการสังเคราะห์ข้อมูลที่แนบ

CAPTURE READY 01 prompt
```

ขั้น 01 ภาพ Terminal replay ของ prompt ที่ส่งจริงเพื่อสร้าง brief

เปิดภาพขนาดเต็ม

```
CODEX CLI 0.153.4 / STEP 01 / RESULT
Read-only replay of a real CLI run. Not Codex Desktop.

Recorded commands: 3 | File-change events: 2

CODEX FINAL RESPONSE (verbatim; local paths shortened)
สร้าง [brief.md](<workspace>/brief.md) และ [evidence.csv](<workspace>/evidence.csv) แล้วทำนั้น
เป็นการสังเคราะห์จาก 'inputs.md' และ 'scope.txt' ที่แนบมาไว้; ไม่ทำ web research และไม่ได้สร้างหน้าเว็บ
ตรวจแล้วว่า evidence ใช้ FACT / INFERENCE / PROPOSAL / GAP ครบ พร้อม S1-S6. GAP หลักคือ mapping Day 1/Day 2, ปลายทาง CTA รายการทพเรียน, สถานะบทความที่กำลังสร้าง, theme toggle และ
ข้อมูลทางวิชาการที่ยังไม่ยืนยัน.

Full prompts / commands / outputs: transcripts/01.txt

CAPTURE READY 01 result
```

ขั้น 01 คำตอบจริงจาก Codex ระบุไฟล์ brief และข้อมูลที่ยังขาด

เปิดภาพขนาดเต็ม

เปิด brief ที่ Codex สร้างในรอบแรก คู่กับภาพผลตอบแทน

ภาพขึ้นนี้ควรให้เห็น prompt file ที่ใช้, event สำคัญของ Codex CLI และ brief ที่เกิดขึ้น สังเกตว่า agent เก็บ Day mapping เป็น GAP หรือไม่ ภาพเป็นหลักฐานลำดับเหตุการณ์ ไม่ใช่หลักฐานว่า brief ถูกต้องครบถ้วน

ไฟล์ prompt ฉบับคัดลอกอยู่ที่ 01-brief.txt และ transcript ของรอบนี้อยู่ในชุด `transcripts/` ของ Lab

ตรวจ brief ก่อนผ่าน

- ชื่อหลักสูตรและสถาบันตรงกับ S1
- อธิบาย 8 ชั่วโมงว่าเป็น workshop subset
- มี audience เป็น PROPOSAL ไม่อ้างว่าเป็นผลวิจัยผู้เรียน
- มีเป้าหมายหลัก, information hierarchy, CTA, section list และ acceptance criteria
- รายการบทเรียนอ้าง S4 และไม่มีบท/URL ที่แต่งขึ้น
- เก็บข้อมูลผู้สอน หน่วยกิต คะแนน สัมผัสและ LMS เป็น GAP
- ไม่อ้างว่า prototype เป็นระบบทางการหรือได้รับการรับรองเพิ่ม

ถ้า brief เติมข้อมูลที่ไม่มี source ให้แก้ brief ก่อนเดินต่อ อย่ารอแก้ด้วยการซ่อนข้อความใน CSS

หลังตรวจสอบ 01 ผู้ควบคุมงานยืนยันการแบ่งวันแล้ว ให้เพิ่ม clarification ลง `inputs.md` ของ workspace ก่อนส่ง prompt 02 วิธีที่ทำซ้ำตัวอย่างได้คือคัดลอก updated snapshot ซึ่งมี source เติมพร้อม Day mapping ที่เพิ่มภายหลัง:

```
cp public/labs/course-website-build/inputs.md practice/psu-course/inputs.md
```

อ่าน diff หรือเปิดไฟล์เทียบก่อน เพื่อยืนยันว่าการเปลี่ยนแปลงคือ Day 1: 00–05 และ Day 2: 06–10 ไม่ใช่การเพิ่มข้อมูลมหาวิทยาลัยส่วนอื่น หากทำงานจากเว็บแทน source tree ให้ดาวน์โหลด `/labs/course-website-build/inputs.md` แล้วแทนเฉพาะ `practice/psu-course/inputs.md`

2. สร้าง semantic HTML ก่อนแต่งภาพ — 25 นาที

ให้ Codex อ่าน brief ที่ผ่านการตรวจ แล้วสร้างหน้า HTML ซึ่งยังทำงานได้เมื่อปิด CSS และ JavaScript ลำดับเนื้อหาควรเป็น header/navigation → hero → จุดเริ่มต้นของผู้เรียน → ภาพรวม 2 วัน → รายการ 11 บท → resources → footer

ขั้นนี้ใช้หลักจาก บทเรียน 04 — Design, assets และคุณภาพ เฉพาะเรื่อง content hierarchy, semantic landmarks, headings และ links ยังไม่ต้องเพิ่ม form controls หรือพยายามทำหน้าให้เหมือนภาพสำเร็จ

Prompt 02 ที่ส่งจริง — HTML

อ่าน inputs.md, scope.txt และ brief.md สร้าง index.html เท่านั้น ใช้ semantic HTML ภาษาไทยที่ทำงานได้โดยยังไม่มี CSS/JS

หน้าที่ของเว็บ: อธิบายบริบท 977-121 Module: Website Design and Development ที่ PSU Phuket และพาผู้เรียนเข้าสู่สื่อ AI Web Studio

ต้องมี skip link ไป main tabindex=-1, header/nav, hero พร้อมชื่อรายวิชาและป้ายเว็บไซต์ตัวอย่างเพื่อการเรียนรู้, เป้าหมายworkshop, แผน2วัน (4ชั่วโมงต่อวันรวมพัก), รายการบทเรียนครบ11บท, แหล่งเรียนต่อ และ footer ที่ลิงก์ rawinsoft.com ในฐานะผู้จัดทำสื่อ

CTAหลักไป /learn/lessons/00-course-map/ (ตรวจสอบชื่อจริงใน inputs.md ก่อนใช้; หากต่างให้ใช้ URL ใน inputs)

CTAรองไป #lessons; linkคู่มือไป /handbook/ และ /resources/

แบ่งบท00-05 เป็น day1, 06-10 เป็น day2 ตามคลังบทเรียน ไม่สัญญาว่าจะสอนครบทุกบทใน8ชั่วโมง

วางแต่ละบทเป็น article[data-lesson][data-day="day1" หรือ "day2"] โดยมีชื่อ ลิงก์อ่าน คำอธิบายสั้นจากข้อบท และลำดับที่จริง

ยังไม่สร้าง search หรือ theme button ที่กดแล้วไม่มีผล ปุ่มเหล่านั้นเพิ่มในขั้นJS

ไม่สร้างฟอร์มสมัคร login ข้อมูลผู้สอน หน่วยกิต คะแนน review หรือโลโก้มหาวิทยาลัย

ไม่ใช้ภาพหรือรันserver ไม่แก้ brief/evidence. ส่งท้ายสรุปสิ่งที่ทำและสิ่งที่ยังไม่ทำอย่างสั้น

เปิด prompt ที่ 02-html.txt

ใน interactive session เดิม ให้วาง prompt 02 แล้วตรวจไฟล์ที่เปลี่ยน หากปิด session ไปแล้วให้เปิด `codex -C practice/psu-course` ไม่ก่อนวาง prompt นี้ การใช้ workspace เดิมทำให้รอบถัดไปอ่านผลที่สร้างไว้ได้ แม้จะเป็น session ใหม่

จุดตรวจ HTML

1. ปิด stylesheet ชั่วคราวแล้วอ่านหน้าโล่จากบนลงล่างได้
2. มี header , nav , main , section headings และ footer เหมาะกับความหมาย
3. ลิงก์บทเรียนทั้ง 11 รายการชี้ไปยัง path จาก S4
4. รายการบทเรียนมีข้อมูลที่รอบ interaction จะนำไปสร้าง search/filter ได้ โดยยังไม่ต้องมี control
5. ไม่มี search, day filter หรือ theme toggle ใน stage นี้ เพราะจะเพิ่มหลัง content และ visual structure ผ่านการตรวจ
6. ไม่มีข้อมูลสมมติที่ทำให้เข้าใจว่าเป็นกำหนดการหรือบริการจริง

ภาพระหว่างทาง — ชั้น 02 HTML

```
CODEX CLI 0.153.4 / STEP 02 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 02-final.txt - < prompts/02-html.txt

PROMPT SENT
อ่าน inputs.md, scope.txt และ brief.md สร้าง index.html เท่านั้น ใช้ semantic HTML ภาษาไทยที่ทำงานได้โดยยังไม่มี CSS/JS
หน้าที่ของเว็บ: อธิบายบริบท 977-121 Module: Website Design and Development ที่ PSU Phuket และพาผู้เรียน เข้าสู่ AI Web Studio
ต้องมี skip link ใน main tabindex=-1, header/nav, hero พร้อมชื่อรายวิชาและป้าย เว็บไซต์ตัวอย่างเพื่อการเรียน, เป้าหมายworkshop, แผน2วัน (4ชั่วโมงต่อวันรวมพัก), รายการบทเรียนครบ11บท, แหล่งเร
ย่นต่อ และ footer ที่ลิงก์ rawinsoft.com ในฐานะผู้จัดทำสื่อ
CTAหลักไป /learn/lessons/00-course-map/ (ตรวจชื่อจริงใน inputs.md ก่อนใช้; หากต่างให้ใช้ URL ใน inputs)
CTAรองไป #lessons; linkคู่มือไป /handbook/ และ /resources/
แบ่งบท00-05 เป็น day1, 06-10 เป็น day2 ตามคําสั่งบทเรียน ไม่สั้กฏว่าจะสอนครบทุกบทใน๒ชั่วโมง
วางแต่ละบทเป็น article[data-lesson][data-day="day1" หรือ "day2"] โดยมีชื่อ ลิงก์อ่าน คำอธิบายสั้นจากข้อบท และลำดับที่จริง
ยังไม่สร้าง search หรือ theme button ที่คมแล้วไม่มีผล เป็น หล้าโน้น พิมพ์ในVS
ไม่สร้างฟอร์มสมัคร login ข้อมูลผู้สอน หน่วยกิต คะแนน review หรือโลโก้มหาวิทยาลัย
ไม่ใช้ภาพหรือserver ไม่เก็บ brief/evidence. ส่งท้ายสรุปสิ่งที่ทำและสิ่งที่ยังไม่ทำอย่างสั้น

CAPTURE READY 02 prompt
█
```

ชั้น 02 prompt สร้าง semantic HTML ที่ส่งจริง

เปิดภาพขนาดเต็ม

[ข้ามไปยังเนื้อหาหลัก](#)

เว็บไซต์ตัวอย่างประกอบการเรียน

977-121 Module: Website Design and Development

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

[แผน workshop บทเรียน แหล่งเรียนต่อ คู่มือการเรียนรู้ ทรัพยากร](#)

AI Web Studio

977-121 Module: Website Design and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

[เริ่มจากบทเรียน 00 คู่มือเรียนทั้งหมด](#)

เป้าหมายของ workshop

AI Web Studio เป็นส่วนหนึ่งของรายวิชา เพื่อช่วยผู้เรียนเลือกจุดเริ่มต้น เข้าถึงบทเรียนจริง และฝึกสร้างเว็บไซต์ด้วย AI อย่างมีขั้นตอน

- เข้าใจเครื่องมือและพื้นฐานของเว็บก่อนเริ่มลงมือ
- ใช้ specs, prompts และเครื่องมือ AI เพื่อพัฒนาเว็บไซต์
- ตรวจสอบภาพและส่งมอบงานที่ตรวจสอบได้

workshop นี้ไม่ใช่รายละเอียดรายวิชาฉบับมหาวิทยาลัยหรือระบบมหาวิทยาลัยอย่างเป็นทางการ

แผน AI Web Studio 2 วัน

รวม 8 ชั่วโมง: 2 วัน วันละ 4 ชั่วโมง และพัก 10 นาทีต่อวัน

แผนนี้เป็นเส้นทางเลือกใช้คําสั่งบทเรียนตามเวลาจริงในห้องเรียน ไม่ได้สัญญาว่าจะสอนครบทุกบทภายใน 8 ชั่วโมง

Day 1 — เริ่มต้น ออกแบบ และสร้างฐานงาน

4 ชั่วโมง รวมพัก 10 นาที: เลือกกิจกรรมจากบท 00-05 เพื่อเตรียมเครื่องมือ วางบริบท และเริ่มทำเว็บไซต์

Day 2 — ขยายงาน ตรวจสอบ และส่งมอบ

4 ชั่วโมง รวมพัก 10 นาที: เลือกกิจกรรมจากบท 06-10 เพื่อทำงานกับข้อมูล ประสพการณ์เว็บ การ deploy และการส่งมอบ

ชั้น 02 หน้า HTML ก่อนแต่ง CSS ที่ viewport 1440 × 1000

เปิดภาพขนาดเต็ม

เปิด snapshot HTML ชั้น 02 — อ่านเนื้อหาและลองกดลิงก์ก่อนดูความสวยงาม

เก็บ snapshot ขึ้นนี้ไว้ใน `stages/02/` ภาพควรช่วยตอบว่าโครงเนื้อหาครบและอ่านตามลำดับได้หรือไม่ การยังไม่เห็น `search/theme control` เป็นผลที่ตึงใจไว้ ภาพไม่ใช่หลักฐานว่า `interaction` หรือ `accessibility` ผ่านแล้ว

3. เพิ่ม Rawinnipa-consistent responsive design — 35 นาที

เมื่อ HTML ถูกต้องแล้วจึงให้ Codex เพิ่ม CSS ตาม บทเรียน 04 และ excerpt S3 ที่คัดจาก `visual source of truth` สำหรับ **course teaching media** เท่านั้น ไม่ใช่กฎนี้ไปเปลี่ยนแบบสดตัวอย่าง HUSH, บ้านสบาย หรือเว็บธุรกิจอื่น Stage นี้เขียน style สำหรับ Light และ Dark ให้ครบ แต่ยังไม่เพิ่ม theme toggle; control และ persistence จะอยู่ในรอบ interaction

ข้อกำหนดหลัก:

- Light เป็นค่าเริ่มต้น: canvas `#F7F7F7`, text `#0A0A0A`, surface สีขาว
- ใช้ Rawin Green `#3EDC81` เป็น accent อย่างจำกัด; ปุ่มพื้นเขียวใช้ข้อความเข้ม
- ข้อความเขียวบนพื้นอ่อนใช้ `#17653F` เมื่อเหมาะสม
- Dark ใช้ canvas `#0A0A0A`, surface `#171717`, text `#F7F7F7`
- Inter สำหรับ Latin และ Noto Sans Thai สำหรับภาษาไทยจาก `/fonts/`
- mobile 390px ต้องไม่เกิด horizontal overflow และ controls ต้องแตะ/กดได้สะดวก
- focus ต้องมองเห็น, reduced motion ต้องยังใช้เนื้อหาได้ และ print กลับเป็นกระดาษขาว
- ไม่สร้างหรือดัดแปลงตรามหาวิทยาลัย

Prompt 03 ที่ส่งจริง — responsive design

อ่าน `inputs.md`, `scope.txt`, `brief.md` และ `index.html` ปรับ `index.html` เค้าที่ต้องใช้ class และสร้าง `style.css` กับ `design-notes.md`

ห้ามเปลี่ยนข้อมูลชื่อรายวิชา สถาบัน เส้นทางบทเรียน หรือแต่งข้อมูลวิชาการเพิ่ม

ออกแบบหน้า course website ให้ชวนเรียน มีเนื้อหาอ่านง่าย ใช้หลักfrontend-design: herobอกงานชัด เส้นทางเรียนเป็นจุดเด่นหนึ่งแห่ง ลำดับจริงช่วยนำทาง ไม่ใช่heroภาพstock

ยึดแบรนด์Rawinnipaของสื่อการสอนตามS3 ไม่อ้างเป็นPSUcorporatebrand. ใช้Inter + Noto Sans Thai

localfontsจากS6; defaultlight แม้ระบบ0Sdark

สร้าง CSS semantic tokens สำหรับsurface/text/muted/line/action/focus และ dark override `html[data-theme="dark"]` พร้อมprintlight

ภาพรวม lightเป็นกระดาษoffwhite/textดำ มีgreenaccentที่CTAเดียว ส่วนcourseheroมีการจัดวางชื่อและpreviewเส้นทาง Brief→Build→Verify→Publish ที่เป็นHTML/CSS

รองรับ320/390/1440px: ไม่ใช่fixedwidthจนล้น, หัวข้อไทยไม่ตัด, navwrap, cardsเรียง1columnบนมือถือ, focusvisible, ลิงก์ชัด, hitareaเหมาะสม, ไม่มีanimationที่จำเป็นต่อเนื้อหา

stylelight/darkทั้งสองผ่านคู่มือข้อความพื้นหลัง; form/controls stylesเตรียมไว้ตามroleแต่ยังไม่เพิ่มJSในขั้นนี้

เก็บสิ่งที่เลือก/เหตุผล/evidenceID/ข้อจำกัดในdesign-notes.md ล้วนๆ. ห้ามใช้remoteCDNใหม่

```
CODEX CLI 0.153.4 / STEP 03 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 03-final.txt - < prompts/03-design.txt

PROMPT SENT
อ่าน inputs.md, scope.txt, brief.md และ index.html ปรับ index.html เฝ้าที่ต้องใช้ class และสร้าง style.css กับ design-notes.md
ห้าม เปลี่ยนข้อมูลชื่อรายวิชา สถาบัน เส้นทางบทเรียน หรือแต่งข้อมูลวิชาการเพิ่ม
ออกแบบหน้า course website ให้ชวนเรียน มีเนื้อหาอ่านง่าย ใช้หลักfrontend-design: heroungงานชัด เส้นทางการเรียนรู้ เป็นจุดเด่นที่แบ่ง ลำดับจริงช่วยนำทาง ไม่ใช่heroภาพstock
ยึดแบรนด์สีawinnkipaของสื่อการสอนตามS3 ไม่อ้างเป็นPSUcorporatebrand. ใช้Inter + Noto Sans Thai localfontsจากS6; defaultlight เฝาระบบOSdark
สร้าง CSS semantic tokens สำหรับsurface/text/muted/line/action/focus และ dark override html[data-theme="dark"] พร้อมprintlight
ภาพรวม light เป็นกระดานwhite/textสีgreenaccentใช้CTA สีขาว ส่วนcourseheroมีการจัดวางสื่อและpreviewเส้นทาง Brief-Build-Verify-Publish ที่เน้นHTML/CSS
รองรับ220/390/1440px: ไม่ใช่fixedwidthจอเส้น, หัวข้อใหม่ไม่ติด, navwrap, cardsเรียงnicolumnบนมือถือ, focusvisible, ลิงกชัด, hitareaเหมาะสม, ไม่มีanimationที่จำเป็นต่อเนื้อหา
stylelight/darkให้สองผ่านดูสื่อความที่หลัง; form/controls stylesเตรียมไว้ตามroleแต่ยังไม่พิมพ์JSในขั้นนี้
เก็บลิงที่ ลีน/เพคผล/evidenceID/ข้อจำกัดในdesign-notes.md ลื่นๆ ห้ามใช้remoteCDNใหม่

CAPTURE READY 03 prompt
█
```

ขั้น 03 prompt ออกแบบ responsive CSS ตามแบรนด์สื่อหลักสูตร

เปิดภาพขนาดเต็ม

เว็บไซต์ตัวอย่างประกอบการเรียน

977-121 Module: Website Design and Development

[แผน workshop](#) [บทเรียน](#) [แหล่งเรียนต่อ](#) [คู่มือการเรียนรู้](#) [ทรัพยากร](#)

มหาวิทยาลัยของสถาบันครุ วิทยาเขตภูเก็ต

AI WEB STUDIO

977-121 Module:
Website Design
and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

เริ่มจากบทเรียน 00

[ดูบทเรียนทั้งหมด](#)

ภาพรวมเส้นทางการทำงาน

01 Brief

02 Build

03 Verify

04 Publish

เป้าหมายของ workshop

ขั้น 03 หลังเพิ่ม CSS โหมด Light บนคอมพิวเตอร์ 1440 × 1000

เปิดภาพขนาดเต็ม

เว็บไซต์ตัวอย่างประกอบการเรียน

977-121 Module: Website Design and Development

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

[แผน workshop](#) [บทเรียน](#) [แหล่งเรียนต่อ](#)

[คู่มือการเรียน](#) [ทรัพยากร](#)

AI WEB STUDIO

977-121 Module: Website Design and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

เริ่มจากบทเรียน 00

[ดูบทเรียนทั้งหมด](#)

ภาพรวมเส้นทางการทำงาน

- 01 Brief
- 02 Build
- 03 Verify

ขั้น 03 การจัดหน้า Light บนมือถือ 390 × 844

[เปิดภาพขนาดเต็ม](#)

[เปิด snapshot CSS ขั้น 03](#) — เนื้อหาเดิมถูกจัดลำดับด้วย typography พื้นที่ว่าง และสี ยังไม่มีปุ่มสลับโหมดในช่วงนี้

ภาพขั้นนี้ควรแสดงหน้าเดียวกันที่ desktop/mobile และ Light/Dark ตามวิธีตั้งสถานะเพื่อ preview สังเกต hierarchy, overflow และ readability อย่าตีความภาพว่า theme toggle หรือ persistence ผ่าน เพราะสองอย่างนั้นยังไม่ถูกสร้าง

เปิด prompt ที่ [03-design.txt](#)

ส่ง prompt 03 ใน `practice/psu-course/` เดิม Codex ควรแก้ style ของงานที่มีอยู่ ไม่สร้าง project ชุดที่สอง ตรวจสอบรายชื่อไฟล์หลังจบและเก็บภาพ stage 03 ก่อนส่ง prompt 04

ตรวจ design ด้วยสถานะจริง

สถานะ	สิ่งที่ดู
390px Light	หัวเรื่องไม่ล้น, cards เรียงอ่านง่าย, ปุ่มไม่ชิดขอบ
Desktop Light	hierarchy, line length และ green accent ไม่แย่งทุกจุด
390px Dark	เมื่อกำหนด dark state เพื่อ preview ข้อความ, border และพื้นผิวอ่านได้
Keyboard	tab order ตามหน้าและ focus ไม่หาย
Reduced motion	ไม่มีข้อมูลที่ต้องรอ animation จึงจะเห็น
Print	พื้นขาว ตัวอักษรเข้ม และลิงก์ยังอ่านได้

ภาพระหว่างทาง — ชั้น 03 responsive design

เก็บ desktop และ mobile อย่างน้อย mode ละหนึ่งภาพใน `stages/03/` พร้อมชื่อ viewport/theme ภาพสวยไม่ถือว่าผ่าน ถ้ายังไม่ได้ตรวจ overflow, keyboard และข้อความจริง ภาพจะถูกรวมกับ terminal capture ในหัวข้อหลังจบ QA

4. เพิ่ม search และ Day filter — 25 นาที

เพิ่ม controls และ JavaScript หลังจากหน้า static ใช้ได้แล้ว ฟังก์ชันมีเพียงค้นหาบทเรียน, กรอง **ทั้งหมด / Day 1 / Day 2** จาก dataset 11 บทเรียน, และสลับ Light/Dark พร้อมจำค่าที่ผู้ใช้เลือก ไม่ต้องใช้ API หรือฐานข้อมูลสำหรับข้อมูลสาธารณะนี้ รอบนี้ใช้คำชี้แจงใหม่ว่า Day 1 คือ 00–05 และ Day 2 คือ 06–10 พร้อมบันทึกว่าเป็น source clarification หลัง brief แรก

ชั้นนี้เชื่อมกับ บทเรียน 05 — เว็บบริษัท ในแง่ progressive enhancement และ UI states และกับ บทเรียน 06 — Data authority ในแง่ไม่เปลี่ยนข้อมูลต้นทางเป็นค่าที่เเต่ใน client แต่ตัวอย่างนี้เป็น static content จึงไม่ควรเพิ่ม Worker/D1 โดยไม่มีเหตุผล

Prompt 04 ที่ส่งจริง — interaction

```
อ่าน inputs.md, scope.txt, index.html และ style.css เพิ่ม progressive enhancement โดยสร้าง main.js และ
แก้ HTML/CSS เฉพาะที่จำเป็น
1) มีปุ่ม #theme-toggle: defaultLight, toggleDark/Light, data-themeบนhtml, aria-pressed=trueเมื่อDark,
ข้อความบอกโหมด/การกระทำชัดเจน. จำเฉพาะค่าที่ผู้ใช้เลือกใน localStorage key psu-course-website-theme-v1.
storageถูกปิดต้องยังใช้ได้ด้วยtry/catch. ไม่มีค่าที่บันทึกแล้วให้Lightแม้OSDark
2) ค้นหาบทเรียน #lesson-search มีlabelจริง: trimช่องว่าง ใช้ชื่อ+คำอธิบายของ11บท matchอย่างปลอดภัย (ไม่
ใช้innerHTMLจากผู้ใช้)
```

3) filter buttons data-filter=all/day1/day2 มีaria-pressedที่ตรง, ค้นหากับfilterทำงานร่วมกันบน article[data-lesson][data-day]. วัน1บท00-05, วัน2บท06-10

4) #lesson-status role=status aria-live=politeแจ้งจำนวนที่พบ, #lesson-empty แสดงเมื่อ0ผล, #clear-search ค้นหาว่างและfilterall แล้วfocussearch ไม่ทำให้focusกระโดดขณะพิมพ์

5) controlsค้นหา/filter/themeเริ่มมีhiddenในHTMLและแสดงเฉพาะเมื่อJSพร้อม ถ้าJSปิดยังเห็น1บทและเข้าเรียนด้วยanchorได้. ให้CSSรักษาความหมาย [hidden] { display:none !important }

ไม่ต้องเพิ่มprogress tracking analytics forms API framework หรือdependencies. ไม่มีcredentials/login. reducedmotionและprintยังทำงาน. เก็บ acceptance-notes.md ระบุการตรวจที่ทำได้จริงและสิ่งที่ยังrobrowser ไม่อ่านรับbrowserถ้ายังไม่ได้รับ

```
CODEX CLI 0.153.4 / STEP 04 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 04-final.txt - < prompts/04-interaction.txt

PROMPT SENT
อ่าน inputs.md, scope.txt, index.html และ style.css เพิ่ม progressive enhancement โดยสร้าง main.js และแก้ HTML/CSS เฉพาะที่จำเป็น
1) มีปุ่ม #theme-toggle: defaultLight, toggleDark/Light, data-themeในhtml, aria-pressed=trueเมื่อDark, ข้อความบอกโหมด/การกระทำชัดเจน. ถ้าเฉพาะค่าที่ใช้เลือกใน localStorage key p
su-course-website-theme-v1. storageถูกปิดต้องยังใช้ได้ด้วยtry/catch. ไม่มีค่าที่บันทึกแล้วให้Lightเมื่อSDark
2) ค้นหาพร้อม #lesson-search มีlabelจริง: trimช่องว่าง ใช้ชื่อ+คำอธิบายของ1บท matchอย่างปลอดภัย (ไม่ใช่innerHTMLจากผู้ใช้)
3) filter-buttons data-filter=all/day1/day2 มีaria-pressedที่ตรง, ค้นหากับfilterทำงานร่วมกันบน article[data-lesson][data-day]. วัน1บท00-05, วัน2บท06-10
4) #lesson-status role=status aria-live=politeแจ้งจำนวนที่พบ, #lesson-empty แสดงเมื่อ0ผล, #clear-search ค้นหาว่างและfilterall แล้วfocussearch ไม่ทำให้focusกระโดดขณะพิมพ์
5) controlsค้นหา/filter/themeเริ่มมีhiddenในHTMLและแสดงเฉพาะเมื่อJSพร้อม ถ้าJSปิดยังเห็น1บทและเข้าเรียนด้วยanchorได้. ให้CSSรักษาความหมาย [hidden] { display:none !important }
ไม่ต้องเพิ่มprogress tracking analytics forms API framework หรือdependencies. ไม่มีcredentials/login. reducedmotionและprintยังทำงาน. เก็บ acceptance-notes.md ระบุการตรวจที่ทำได้
จริงและสิ่งที่ยังrobrowser ไม่อ่านรับbrowserถ้ายังไม่ได้รับ

CAPTURE READY 04 prompt
█
```

ขั้น 04 prompt เพิ่ม search day filters และ theme toggle

เปิดภาพขนาดเต็ม

เส้นทางบทเรียน

บท 00-05 อยู่ในเส้นทาง Day 1 และบท 06-10 อยู่ในเส้นทาง Day 2 เพื่อใช้เลือกเนื้อหาใน workshop

ค้นหาบทเรียน

GSAP

ล้างการค้นหาและตัวกรอง

เลือกเส้นทาง workshop

ทั้งหมด

Day 1

Day 2

พบบทเรียน 1 จาก 11 บท

Day 2: บท 06-10

07 — GSAP product story, storefront และ quote demo

สร้าง product story, storefront และ quote demo ด้วย GSAP

[อ่านบทเรียน 07](#)

ชั้น 04 ตัวอย่างค้น GSAP ร่วมกับ Day 2 แล้วแสดงจำนวนผล

เปิดภาพขนาดเต็ม

เส้นทางบทเรียน

บท 00-05 อยู่ในเส้นทาง Day 1 และบท 06-10 อยู่ในเส้นทาง Day 2 เพื่อใช้เลือกเนื้อหาใน workshop

ค้นหาบทเรียน

ไม่มีบทเรียนชื่อนี้

×

ล้างการค้นหาและตัวกรอง

เลือกเส้นทาง workshop

ทั้งหมด

Day 1

Day 2

พบบทเรียน 0 จาก 11 บท

ไม่พบบทเรียนที่ตรงกับการค้นหาและตัวกรองนี้

ชั้น 04 คำค้นที่ไม่มีผลและปุ่มล้างการค้นหาพร้อมตัวกรอง

เปิดภาพขนาดเต็ม

เปิด snapshot interaction ชั้น 04 — ลองเปลี่ยนเงื่อนไขด้วยตนเอง

ภาพชั้นนี้ควรให้เห็น prompt clarification เรื่อง Day mapping และหน้าเว็บหลังมี search, filters และ theme toggle แล้วสังเกต active state, result count และ empty state จาก browser ภาพหนึ่งสถานะยังไม่พิสูจน์ keyboard behavior, persistence หรือทุกชุดคำค้น

เปิด prompt ที่ `04-interaction.txt`

ส่ง prompt 04 ใน workspace เดิม ชั้นนี้จึงเห็น updated `inputs.md` , HTML และ CSS จากรอบก่อนพร้อมกัน หลังจบให้ลอง controls ใน browser preview ก่อนส่งรอบ review

กรณีทดสอบขั้นต่ำ

การกระทำ	ผลที่คาดหวัง
เปิดหน้าโดยไม่พิมพ์	เห็นบทเรียนทั้งหมด 11 รายการ
ค้นด้วยคำในชื่อบท	เหลือเฉพาะรายการที่ตรงและแจ้งจำนวนอย่างสุภาพ
เลือก Day 1/Day 2	แสดงเฉพาะกลุ่มที่กำหนดไว้ในข้อมูลของหน้า

การกระทำ	ผลที่คาดหวัง
ค้นแล้วเลือก filter	เงื่อนไขสองอย่างทำงานร่วมกัน
ค้นคำที่ไม่มี	แสดง empty state และวิธีกลับมาดูทั้งหมด
ลบข้อความในช่องค้นหาเอง	รายการกลับมาตาม filter ปัจจุบัน
กด "ล้างการค้นหาและตัวกรอง"	คืนคำค้นว่างและ filter ทั้งหมด เห็น 11 un แล้ว focus กลับช่องค้นหา
ใช้ keyboard	focus, Enter/Space และลำดับ tab ทำงาน
ปิด JavaScript	ยังอ่านและเปิดบทเรียนทั้งหมดได้
เลือก Dark แล้ว reload	ยังคงค่าที่ผู้ใช้เลือก; first visit ที่ไม่มีค่าต้องเป็น Light

อย่าใช้สีเพียงอย่างเดียวบอก filter ที่ active ใส่ข้อความ/สถานะเชิงโปรแกรม และอย่าซ่อนจำนวนผลจาก screen reader หากจำนวนนั้นเปลี่ยนจากการค้นหา

5. ให้ Codex review แล้วแก้จากหลักฐาน — 25–40 นาที

การ review ต้องเกิดหลังมีหน้าให้ตรวจ ส่งขอบเขตไฟล์และ acceptance criteria ให้ Codex ตรวจ source, links, responsive states, keyboard, theme persistence และข้อมูลนี้อาจกล่าวเกินหลักฐาน ขอให้รายงาน finding พร้อมตำแหน่งและวิธีพิสูจน์ รอบนี้อนุญาตให้แก้ข้อผิดพลาดที่ชี้จาก source ได้ และต้องบันทึกเหตุผลกับขอบเขตที่ยังไม่ได้ทดสอบ

Prompt 05 review ที่ส่งจริง

```
อ่าน inputs.md, brief.md, index.html, style.css, main.js และ acceptance-notes.md แล้วทบทวน
เฉพาะsourceในโฟลเดอร์นี้
ตรวจข้อมูลจริงเทียบsource, mapping11บท/Day1Day2, defaultlight+dark storageerror, การค้นรวม
กับfilter/empty/reset, noJSfallback, semanticHTML keyboardfocus alt printและreducedmotion
รัน node --check main.js ได้ แต่อย่ารันserverหรืออ้างbrowser/contrastผ่านจากการอ่านโค้ด
หากพบข้อผิดพลาดที่พิสูจน์ได้ให้แก้ไฟล์ที่เกี่ยวข้องและเขียน review.md ระบุสาเหตุ สิ่งที่เปลี่ยน ผลคำสั่ง และbrowserchecksที่ยังรอ
หากไม่พบให้บันทึกตามจริง ไม่สร้างปัญหาหรือผลทดสอบขึ้นเอง. เว็บไซต์นี้เป็นตัวอย่างสื่อรายวิชา ไม่ใช่LMSหรือเว็บมหาวิทยาลัยที่รับรอง
แล้ว
```

```
CODEX CLI 0.153.4 / STEP 05 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 05-final.txt - < prompts/05-review.txt

PROMPT SENT
อ่าน inputs.md, brief.md, index.html, style.css, main.js และ acceptance-notes.md แล้วทบทวน เฉพาะsourceในไฟล์ ดอร์นี้
ตรวจสอบสิ่งที่เขียน source, mappinginput/Day1Day2, defaultlight+dark storageerror, การค้นร่วมกับfilter/empty/reset, noJSfallback, semanticHTML keyboardfocus alt printและreduc
edmotion
รัน node --check main.js ได้ แต่อย่ารันserverหรืออ้างbrowser/contrastผ่านจากการอ่านได้
หากพบข้อผิดพลาดที่พิสูจน์ได้ให้แจ้งสิ่งที่เกี่ยวข้องและเขียน review.md ระบุสาเหตุ สิ่งที่ไม่เปลี่ยน ผลลัพธ์ และbrowserchecksที่ยังรอ
หากไม่พบให้บันทึกตามจริง ไม่สร้างปัญหาหรือผลทดสอบขึ้นเอง. เว็บไซต์นี้เป็นตัวอย่างสื่อรายวิชา ไม่ใช่LMSหรือ เว็บบทเรียนที่รับรองแล้ว

CAPTURE READY 05 prompt
█
```

ขั้น 05 prompt ให้ Codex ตรวจสอบ source และแก้ไขสิ่งที่ชี้ได้

เปิดภาพขนาดเต็ม

```
CODEX CLI 0.153.4 / STEP 05 / RESULT
Read-only replay of a real CLI run. Not Codex Desktop.

Recorded commands: 6 | File-change events: 2

CODEX FINAL RESPONSE (verbatim; local paths shortened)
ตรวจสอบและแก้ไขข้อบกพร่อง reduced-motion แล้ว: เพิ่ม fallback ให้ skip link ไม่ใช่ `transform` เมื่อผู้ใช้ตั้งค่า reduced motion

- แก้ [style.css](<workspace>/style.css:88)
- อัปเดต [acceptance-notes.md](<workspace>/acceptance-notes.md:10)
- บันทึกการเปลี่ยนแปลงใน [review.md](<workspace>/review.md:1)

เป็นบันทึกย่อ: เมื่อวัน 11 เม, Day 1 = 00-05 (6), Day 2 = 06-10 (5); light default, dark/localStorage error handling, search+filter+empty+reset, no-JS fallback, semantic
/focus/print ตรวจสอบที่อ่านได้จากโค้ด

`node --check main.js` ผ่าน (exit 0)

ไม่ได้รัน server หรือ browser และไม่อ้างผล browser/contrast pass; รายการ runtime checks ที่ยังรอระบุไว้ใน `review.md` แล้ว.

Full prompts / commands / outputs: transcripts/05.txt

CAPTURE READY 05 result
█
```

ขั้น 05 คำตอบจริงรายงานการปรับ CSS และแยก browser checks ที่ยังไม่ได้รับ

เปิดภาพขนาดเต็ม

รอบ 05 Codex เพิ่ม reduced-motion fallback ให้ skip link จากการอ่าน source ผลนี้เป็นการทบทวนของผู้สร้างเอง จึงยังต้องตรวจใน browser โดยอีกขั้นหนึ่ง อ่าน review.md ของรอบ 05

ภาพขั้น review ควรแสดง prompt กับ finding ที่อ้างตำแหน่งไฟล์หรือ evidence ได้ สังเกตว่า agent แยก Verified, Finding และ Not tested หรือไม่ ภาพรายงานไม่พิสูจน์ว่า finding ถูกแก้แล้ว

เปิด prompt ที่ 05-review.txt

ส่ง prompt 05 ใน workspace เต็ม โดยให้ตรวจสอบ source ก่อนแก้ไขและบันทึกสิ่งที่แก้ตามจริง หากใช้ non-interactive mode ให้เปลี่ยนเฉพาะชื่อ prompt/input และชื่อไฟล์ JSONL ในตัวอย่างคำสั่ง อย่า redirect กับ 01-events.jsonl

จัดผล review เป็นสามกลุ่ม:

- **Verified:** มี command/browser evidence รองรับ
- **Finding:** พบปัญหาและชี้ตำแหน่ง/ผลกระทบได้
- **Not tested:** ยังไม่มีเครื่องมือหรือ evidence พอ

คำว่า “looks good” ไม่ใช่หลักฐาน และการไม่มี error ใน console ไม่ได้ยืนยัน keyboard, contrast, link target หรือ mobile overflow

รอบ 06 — ปิด GAP ที่ยืนยันแล้วและส่งต่องาน

หลังทดสอบหน้าเว็บ พบว่า brief/evidence ฉบับแรกยังเก็บบางเรื่องเป็น GAP แม้เราให้ข้อมูลเพิ่มเติมใน prompt 02 และ 04 แล้ว จึงมีรอบ 06 เพื่ออัปเดตเอกสารให้ตรงกับ implementation โดยไม่แก้ไขหน้าเว็บ

เรื่อง	รอบแรก	ข้อมูลที่ได้รับเพิ่ม
แบ่งบทเรียนตามวัน	ยังรอ mapping	Day 1 un 00-05 / Day 2 un 06-10 จากตารางของโครงการ
สลับ Light/Dark	ยังเป็นคำถามใน brief	เป็นข้อกำหนดชัดเจนใน prompt 04
CTA ดูบทเรียนทั้งหมด	ยังรอยืนยันปลายทาง	#lessons ตาม prompt 02
ผู้สอน/หน่วยกิต/คะแนน/LMS	GAP	ยังไม่ยืนยันสำหรับเว็บไซด์นี้ จึงยังไม่เติม

แบบ implementation-decisions.md และผล browser ที่ตรวจจริงก่อนส่ง prompt นี้ ในตัวอย่างใช้ผลจาก parent runner ส่วน Codex รอบ 06 อ่านรายงานและสรุปส่งต่อ ไม่ได้เป็นผู้รับ browser เอง

อ่าน inputs.md, implementation-decisions.md, browser-verification.json, brief.md, evidence.csv และ source ที่มี

รอบนี้ห้ามแก้ index.html/style.css/main.js และห้ามรัน server/deploy/modelอื่น

แก้ brief.md และ evidence.csv ให้สะท้อนข้อกำหนดที่ได้รับเพิ่มจริง:

- Day1 บท00-05 และDay2บท06-10 ยืนยันแล้วจาก dayData ของโครงการ
- มี theme toggle Light/Dark ยืนยันแล้วใน prompt04; Lightเป็นdefaultและเก็บเฉพาะตัวเลือกของผู้ใช้
- CTAรองดูบทเรียนทั้งหมดไป #lessons ยืนยันในprompt02 ไม่ใช่ปลายทาง / ที่ยังสงสัยในbriefแรก

บันทึกแหล่งการยืนยันและการเปลี่ยนสถานะ GAP โดยไม่ลบประวัติรอบแรก (snapshotเดิมเก็บแยกแล้ว)

อย่าอ้างว่าข้อมูลผู้สอน หน่วยกิต เกณฑ์คะแนน หรือLMSได้รับการยืนยันแล้ว

สร้าง handoff.md ที่สรุปไฟล์/CTA/interaction/token/หลักฐานตรวจ/สิ่งที่ยังไม่ทำ

อ่านผล browser ที่แนบแล้วรายงานว่า parent runner เป็นผู้ทดสอบ ไม่อ้างว่าคุณรัน browser เอง

เกณฑ์ที่ผ่านคือเฉพาะในรายงาน; ยังไม่อ้างเผยแพร่ public จนมีหลักฐานจากขั้นdeploy

อธิบายสั้นๆ ว่าเมื่อข้อกำหนดได้รับคำตอบแล้ว ต้องแก้ทั้ง brief/evidence และimplementationให้ตรงกัน

```
CODEX CLI 0.153.4 / STEP 06 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 06-final.txt - < prompts/06-handoff.txt

PROMPT SENT
อ่าน inputs.md, implementation-decisions.md, browser-verification.json, brief.md, evidence.csv และ source ที่มี
รอบนี้ห้ามแก้ index.html/style.css/main.js และห้ามรัน server/deploy/modelอื่น
แก้ brief.md และ evidence.csv ให้สะท้อนข้อกำหนดที่ได้รับเพิ่มจริง:
- Day1 บท00-05 และDay2บท06-10 ยืนยันแล้วจาก dayData ของโครงการ
- มี theme toggle Light/Dark ยืนยันแล้วใน prompt04; Lightเป็นdefaultและเก็บเฉพาะตัวเลือกของผู้ใช้
- CTAรองดูบทเรียนทั้งหมดไป #lessons ยืนยันในprompt02 ไม่ใช่ปลายทาง / ที่ยังสงสัยในbriefแรก
บันทึกแหล่งการยืนยันและการเปลี่ยนสถานะ GAP โดยไม่ลบประวัติรอบแรก (snapshotเดิมเก็บแยกแล้ว)
อย่าอ้างว่าข้อมูลผู้สอน หน่วยกิต เกณฑ์คะแนน หรือLMSได้รับการยืนยันแล้ว
สร้าง handoff.md ที่สรุปไฟล์/CTA/interaction/token/หลักฐานตรวจ/สิ่งที่ยังไม่ทำ
อ่านผล browser ที่แนบแล้วรายงานว่า parent runner เป็นผู้ทดสอบ ไม่อ้างว่าคุณรัน browser เอง
เกณฑ์ที่ผ่านคือเฉพาะในรายงาน; ยังไม่อ้างเผยแพร่ public จนมีหลักฐานจากขั้นdeploy
อธิบายสั้นๆ ว่าเมื่อข้อกำหนดได้รับคำตอบแล้ว ต้องแก้ทั้ง brief/evidence และimplementationให้ตรงกัน

CAPTURE READY 06 prompt
```

ขั้น 06 prompt ปิดช่องว่างใน brief ด้วยข้อมูลที่ยืนยันแล้วและผลทดสอบที่แบบ

เปิดภาพขนาดเต็ม

```
CODEX CLI 0.153.4 / STEP 06 / RESULT
Read-only replay of a real CLI run. Not Codex Desktop.

Recorded commands: 7 | File-change events: 2

CODEX FINAL RESPONSE (verbatim; local paths shortened)
ขั้น ตอนเอกสารครบแล้ว โดยไม่แตะ 'index.html', 'style.css', 'main.js' และไม่ได้รัน server, browser, deploy หรือโมเดลอื่น

- [brief.md](<workspace>/brief.md:10): ยืนยัน Day 1 ณ 00-05 / Day 2 ณ 06-10, CTA ระบุไป '#lessons', และ theme toggle light-default ที่เน้นเฉพาะ explicit user choice หรือ
นตารางปิด GAP เดิม
- [evidence.csv](<workspace>/evidence.csv:14): เพิ่มหลักฐาน D1-D3, ผลตรวจ parent-run 5/5, GAP เรื่อง public deployment และ 'STATUS_CHANGE' ที่คงร่องรอย GAP รอบแรก
- [handoff.md](<workspace>/handoff.md:1): สรุปไฟล์, CTA, interaction/token, หลักฐานตรวจ และสิ่งที่ยังไม่ยืนยัน

ผา browser ที่แนะนำระบุว่า parent runner เป็นผู้รัน Playwright-not this Codex run-ผ่าน 5/5 และยัง 'not yet tested in public'; จึงไม่ได้อ้างว่าเผยแพร่แล้ว

ยังไม่อ้างยืนยันผู้สอน, หน่วยกิต, เกณฑ์คะแนน หรือ LMS. ตรวจ CSV แล้วมี 20 records และ 5 คอลัมน์ครบทุกแถว. เมื่อได้ข้อกำหนดใหม่ ต้องปรับ brief, evidence และ implementation ให้สอดคล้องกันเสมอ.

Full prompts / commands / outputs: transcripts/06.txt

CAPTURE READY 06 result
```

ขั้น 06 คำตอบจริงเกี่ยวกับ brief evidence และ handoff ที่อัปเดต

เปิดภาพขนาดเต็ม

เปิด handoff รอบ 06 — ระบุผลตรวจ local ตามหลักฐาน ณ รอบนั้น ยังไม่ใช้หลักฐาน public deployment

เปิด prompt 06-handoff.txt, การตัดสินใจที่**สิ่งเพิ่ม** และ ผล browser ที่ใช้ แล้วเทียบ brief รอบแรก กับ brief ที่อัปเดตแล้ว เพื่อดูการปิดช่องว่างโดยไม่ต่างข้อมูลธุรกิจเพิ่ม

รอบ 07 — แก่จากการตรวจหน้าตาจริง แล้วเพิ่ม regression check

แม้ browser suite รอบแรกผ่าน การตรวจเพิ่มพบว่าขอบช่องคั่นหากสลับกับพื้น ทั้ง Light และ Dark เราจึงเพิ่ม `--control-line` แยกจากเส้นแบ่งตกแต่ง และใช้กับ controls เพื่อให้มองเห็นขอบชัดเจนขึ้น

กรณีช่องกรอกที่ต้องใช้ขอบเพื่อบอกตำแหน่ง control ให้ตรวจ contrast อย่างน้อย 3:1 กับสีที่ติดกัน ไม่ได้หมายความว่าเส้นตกแต่งทุกเส้นต้องผ่านเกณฑ์เดียวกัน W3C: Non-text Contrast

การตรวจ browser โดย parent/reviewer พบปัญหาที่ต้องแก้ตามหลักฐานนี้:

- ขอบ input/button ที่ไม่ได้เลือกกลืนกับพื้น .lesson-controls: light #D0D0D0 บน #FFF =1.54:1; dark #555 บน #171717 =2.40:1

เพราะfillของcontrolเหมือนcontainer ขอบนี้จึงเป็น visual boundary ของcontrolและต้องมีcontrastอย่างน้อย3:1

- ลิงก์ walkthrough ใน footer ยังมีข้อความ กำลังสร้าง ซึ่งจะไม่ตรงกับฉบับเผยแพร่

แก้เฉพาะ index.html, style.css และบันทึก review-fix.md/อัปเดต handoff.md:

เพิ่ม semantic token `--control-line` สำหรับขอบcontrol โดยใช้ light #767676 และ dark #858585

นำไปใช้กับ input, filter buttons และ clear/theme buttons ตามที่ต้องมีขอบ ไม่จำเป็นต้องเปลี่ยนdividerตกแต่งทั้งหมด รักษาfocus, layout, lightdefault/dark, storagefailure, search/filter/reset/noJS behaviorเดิม ไม่แก้main.js

เปลี่ยนข้อความท้ายลิงก์walkthroughเป็นข้อความที่อ่านได้ตามปกติและยังชี้URLเดิม

อย่าแก้historical snapshotsนอกรอบนี้ อย่ารันserverหรือdeploy

บันทึกว่าbrowser/contrastrecheckรอบparentrunner ไม่อ้างผ่านเองจากการอ่านโค้ด

```
CODEX CLI 0.153.4 / STEP 07 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <workspace> --sandbox workspace-write --json --output-last-message 07-final.txt - < prompts/07-fix-controls.txt

PROMPT SENT
การตรวจ browser โดย parent/reviewer พบปัญหาที่ต้องแก้ตามหลักฐานนี้:
- ขอบ input/button ที่ไม่ได้เลือกสีพื้น .lesson-controls: light #D8D8D8 บน #FFF =1.54:1; dark #555 บน #171717 =2.40:1
  เพราะfillของcontrol พื้นcontainer ขอบนี้จึงเป็น visual boundary ของcontrolและต้องมีcontrastอย่างน้อย3:1
- สิ่งกีดขวาง walkthrough ใน footer ยังมีข้อความ กำลังสร้าง ซึ่งจะไม่ตรงกับฉบับเผยแพร่
แก้ไข index.html, style.css และบันทึก review-fix.md/บันทึก handoff.md:
เพิ่ม semantic token --control-line สำหรับขอบcontrol โดยใช้ light #767676 และ dark #958585
นำไปใช้กับ input, filter buttons และ clear/theme buttons ตามที่คิดจะมีขอบ ไม่จำเป็นต้องเปลี่ยนdividerตกแต่งทั้งหมด
รักษาfocus, layout, lightdefault/dark, storagefailure, search/filter/reset/noJS behaviori เดิม ไม่แก้main.js
เปลี่ยนเพื่อความท้ายสิ่ง walkthrough เป็นข้อความที่อ่านได้ตามปกติและยังสืบURL เดิม
อย่าแก้historical snapshotsเมกรุ่นนี้ อย่ารันserverหรือdeploy
บันทึกว่าbrowser/contrastchecksofparentrunner ไม่อ้างผ่าน อกจากการอ่านโค้ด

CAPTURE READY 07 prompt
```

รอบ 07 prompt แก้ control boundary จาก finding ที่ตรวจได้

เปิดภาพขนาดเต็ม

ค้นหาบทเรียน

ค้นหาจากชื่อหรือคำอธิบายบทเรียน

เลือกเส้นทาง workshop

ทั้งหมด

Day 1

Day 2

ล้างการค้นหาและตัวกรอง

พบบทเรียน 11 จาก 11 บท

ก่อนแก้ไข ขอบช่องค้นหาและตัวกรอง Light ของ snapshot 04

เปิดภาพขนาดเต็ม

ค้นหาบทเรียน

ค้นหาจากชื่อหรือคำอธิบายบทเรียน

เลือกเส้นทาง workshop

ทั้งหมด

Day 1

Day 2

ล้างการค้นหาและตัวกรอง

พบบทเรียน 11 จาก 11 บท

หลังแก้ไข Light ใช้ semantic control-line และตรวจสอบสีจาก browser

เปิดภาพขนาดเต็ม

ค้นหาบทเรียน

ค้นหาจากชื่อหรือคำอธิบายบทเรียน

เลือกเส้นทาง workshop

ทั้งหมด

Day 1

Day 2

ล้างการค้นหาและตัวกรอง

พบบทเรียน 11 จาก 11 บท

หลังแก้ไข Dark ทำให้ขอบ controls ชัดกับพื้นเข้ม

เปิดภาพขนาดเต็ม

เพิ่ม regression assertion จากสี่ที่ browser คำนวณจริงสำหรับช่องค้นหา ปุ่มกรองที่ยังไม่เลือก และปุ่มล้าง โดยตรวจทั้งสอง โหมดที่ 320/390/1440px เก็บ snapshot สอบ 07 และ prompt แก้อธิบาย แยกจากภาพก่อนแก้ไข เพื่อไม่เขียนทับประวัติการเรียนรู้

ตารางบันทึก QA

ID	Finding / claim	Evidence ก่อนแก้ไข	ไฟล์ที่แก้ไข	Evidence หลังแก้ไข	สถานะ
Q1					
Q2					
Q3					

ใช้ targeted checks ก่อน แล้วค่อยรัน lint/build/document checks ที่ repository กำหนด บันทึก command, exit code และขอบเขตของสิ่งที่ตรวจจริง ถ้า test รันไม่ได้ให้บอกเหตุผลและใช้ next-best check โดยไม่เปลี่ยน not tested เป็น passed

6. ตรวจ final page และเตรียมเผยแพร่ — 15–20 นาที

เปิด บทเรียน 08 — Cloudflare deployment และ บทเรียน 09 — Production Operations ก่อนเผยแพร่ เว็บไซต์ที่เผยแพร่พร้อมเอกสารชุดนี้ใช้ deployment path ของ **เว็บไซต์หลักสูตรที่ผู้สอนมีอยู่** และเพิ่มตัวอย่างที่ `/examples/course-website/` ; นี้ไม่ใช่ปลายทางที่ผู้เรียนควร deploy กับ และไม่ใช้บริการ production ในนามมหาวิทยาลัย

Preview งานของผู้เรียนโดยไม่กับ worked example

เมื่อ `practice/psu-course/` มี `index.html` , `style.css` และ `main.js` แล้ว ให้คัดลอกเฉพาะสามไฟล์ไปยัง static practice path:

```
mkdir -p public/practice/course-website
cp practice/psu-course/index.html public/practice/course-website/index.html
cp practice/psu-course/style.css public/practice/course-website/style.css
cp practice/psu-course/main.js public/practice/course-website/main.js
npm run build
npm run preview
```

เปิด `http://127.0.0.1:3320/practice/course-website/` และหยุด server ด้วย `Ctrl+C` เมื่อเสร็จ ถ้า agent ใช้ชื่อ CSS/JS ต่างจากนี้ให้อ่าน `index.html` แล้วคัดลอกไฟล์ local ที่หน้านั้นอ้างอิง ห้ามคัดลอก dependency จาก URL ภายนอกมาโดยไม่ตรวจสอบสิทธิ์และที่มา

ไม่ควรคัดลอกงานฝึกไปกับ `examples/course-website/` เพราะ path นั้นเป็น worked example ของ source bundle หากจำเป็นต้องทดลอง integration ที่ path เดียวกัน ให้เก็บสำเนาไฟล์เดิมก่อนและทำใน source copy ของตนเองเท่านั้น

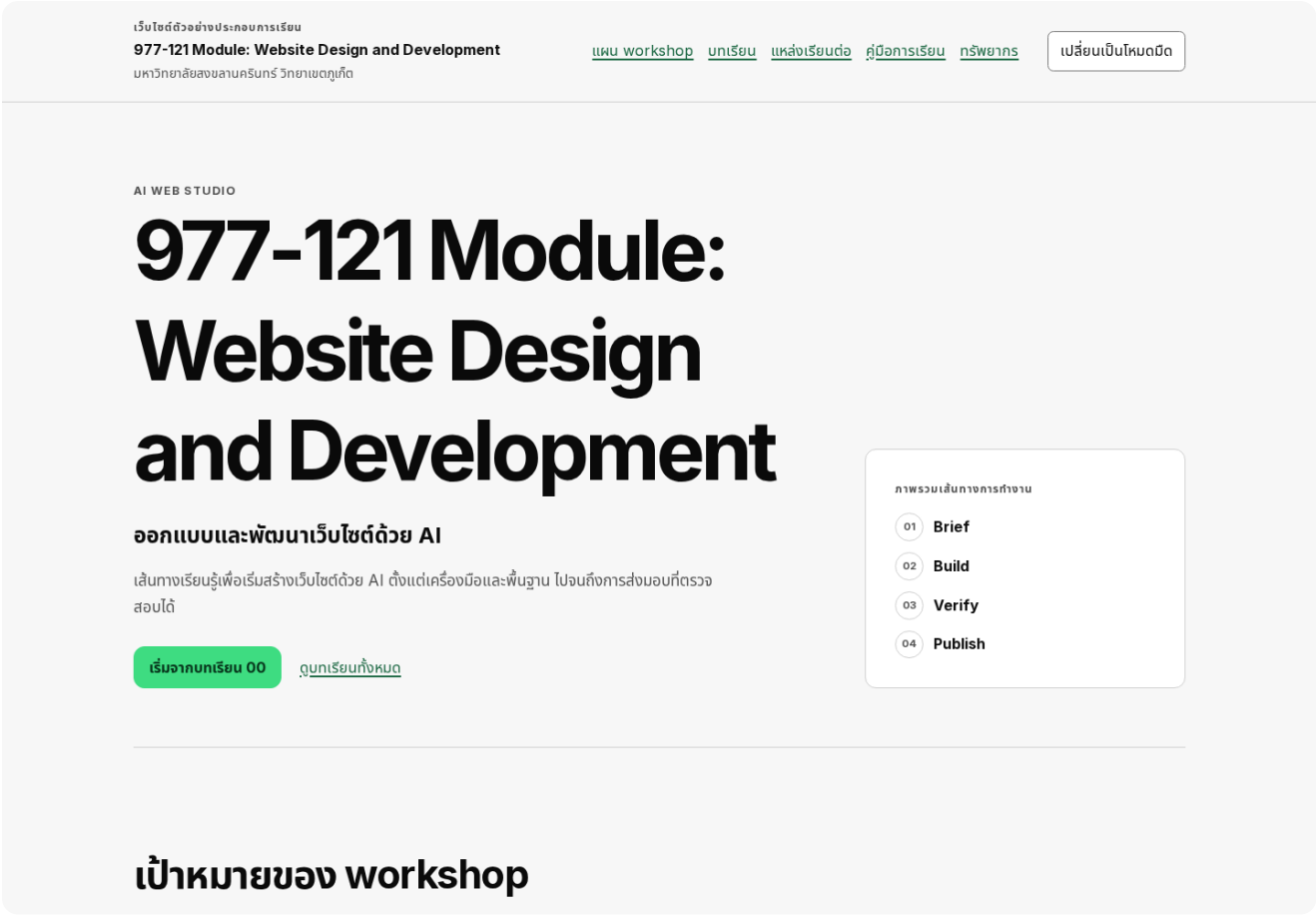
ตรวจสอบลำดับ:

1. เปิด final page ที่ desktop/mobile ทั้ง Light และ Dark
2. ตรวจสอบ 11 lesson links, resources, handbook, labs และ `https://rawinsoft.com/`
3. ทดลอง search, filter, empty state, reset และ theme persistence
4. ตรวจสอบ keyboard, visible focus, reduced motion และ print
5. รับ targeted tests, documentation checks, lint และ build ตาม repository
6. ทำ deployment dry-run และแยกผล dry-run ออกจาก public deployment
7. เมื่อเผยแพร่ในระบบที่มีสิทธิ์แล้ว จึงเก็บ public URL, version และ HTTP/browser smoke evidence
8. อัปเดต PDF/source bundle ผ่านหน้า Resources และตรวจสอบว่า download ตรงกับ source รุ่นเดียวกัน

อย่าเขียนว่า deploy สำเร็จจาก build หรือ dry-run เพียงอย่างเดียว และอย่าใส่ token, cookies, account ID ที่ไม่จำเป็น หรือ secret ลง transcript/screenshot

ทำต่อที่ Lab Deploy เว็บตัวอย่างขึ้น Cloudflare ก็ละชั้น: export เฉพาะ HTML/CSS/JS/fonts ของเว็บนี้เป็น Static Assets และ deploy ด้วยชื่อ Worker ของผู้เรียน ไม่ต้องสร้าง D1 ส่วน บทเรียน 08 ใช้เมื่อ deploy เว็บไซต์หลักสูตรทั้งชุดที่มี API และฐานข้อมูล ทั้งสองเส้นทางใช้ config คณະไฟล์

ภาพผลสุดท้ายและลำดับคำสั่ง



เว็บไซต์ฉบับสุดท้าย Light ที่ desktop 1440 × 1000

เปิดภาพขนาดเต็ม

AI WEB STUDIO

977-121 Module: Website Design and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

เริ่มจากบทเรียน 00

[ดูบทเรียนทั้งหมด](#)

ภาพรวมเส้นทางการทำงาน

- 01 Brief
- 02 Build
- 03 Verify
- 04 Publish

เป้าหมายของ workshop

เว็บไซต์ฉบับสุดท้าย Dark ที่ desktop 1440 × 1000

เปิดภาพขนาดเต็ม

เว็บไซต์ตัวอย่างประกอบการเรียน

977-121 Module: Website Design and Development

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

[แผน workshop](#) [บทเรียน](#) [แหล่งเรียนต่อ](#)

[คู่มือการเรียน](#) [ทรัพยากร](#)

เปลี่ยนเป็นโหมดมืด

AI WEB STUDIO

977-121 Module: Website Design and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

เริ่มจากบทเรียน 00

[ดูบทเรียนทั้งหมด](#)

ภาพรวมเส้นทางการทำงาน

01 Brief

เว็บไซต์ฉบับสุดท้าย Light ที่ mobile 390 × 844

[เปิดภาพขนาดเต็ม](#)



เว็บไซต์ฉบับสุดท้าย Dark ที่ mobile 390 × 844

เปิดภาพขนาดเต็ม

ดูทั้งหน้า desktop · ดูทั้งหน้า mobile · เปิดชุดไฟล์/ภาพทุกชั้น

อ่านเอกสารนี้เป็น PDF แยกเล่มหรือดาวน์โหลด source ได้ที่ Resources

ภาพในชุดนี้มี caption ระบุ **ชั้น**, **viewport**, **theme** และ**สิ่งที่กำลังพิสูจน์** หากภาพ Terminal มี path, username หรือ credential ที่ไม่จำเป็นให้ redact ก่อนเผยแพร่ แต่เก็บ prompt สำคัญครบในไฟล์ข้อความเพื่อให้ทำซ้ำได้

เชื่อม Lab นี้กับ 11 บทเรียน

บท	ใช้ตรงไหนใน Lab
<u>00</u>	เตรียมเครื่อง, source/build และ local preview
<u>01</u>	prompt contract, ขอบเขตไฟล์ และ evidence
<u>02</u>	เลือกใช้ skill/MCP เท่าที่แก้ปัญหาวจริง; Lab นี้ไม่ต้องเพิ่ม dependency
<u>03</u>	brief, acceptance criteria และ prompt เป็นรอบ
<u>04</u>	hierarchy, tokens, responsive และ accessibility
<u>05</u>	semantic form/control states และ progressive enhancement
<u>06</u>	แยก data authority ออกจาก UI; ไม่แต่ง dataset
<u>07</u>	animation เป็น optional enhancement ไม่กั้น content
<u>08</u>	build, dry-run, deploy และ public smoke evidence
<u>09</u>	secrets, logs, rollback และ release record
<u>10</u>	ส่งมอบ source, prompts, tests และข้อจำกัดให้คนถัดไป

หากต้องการออกแบบผ่านเครื่องมือภาพก่อนเขียน code ให้ใช้ Claude Design, Google Stitch และ Figma MCP ระหว่าง brief กับ HTML หากต้องการสร้าง brand system จากข้อมูลดิบ ให้ทำ Research-to-Brand Lab ก่อน Lab นี้ ทั้งสองทางเลือกเป็นเนื้อหาเรียนต่อ ไม่เพิ่ม workshop 8 ชั่วโมง

Worksheet ส่งงาน

กรอกและแนบไฟล์นี้กับงาน:

ชื่อโครงการ:

วันที่ทำ:

Codex client/version ที่เห็นจริง:

Source IDs ที่ใช้:

FACT สำคัญ 3 ข้อ:

PROPOSAL ที่เลือก 2 ข้อและเหตุผล:

GAP ที่ยังไม่เติม:

Primary user task:

Primary CTA และผลหลังคลิก:

Sections ที่สร้าง:

Prompt files ที่ส่งจริง:

ไฟล์ที่ Codex แก่ในแต่ละรอบ:

Finding สำคัญจาก review:

การแก้และ evidence หลังแก้:

Targeted test:

Lint/build/document check:

Mobile/keyboard/theme evidence:

สถานะ deploy: not run / dry-run / public verified

Public URL และ version (เมื่อมีจริง):

สิ่งที่ยัง not tested:

ข้อควรรู้สำหรับผู้ดูแลคนถัดไป:

Rubric 20 คะแนน

หัวข้อ	5 คะแนน	3 คะแนน	1 คะแนน
ข้อมูลและ brief	ทุก claim สำคัญย้อน source ได้, แยก GAP/PROPOSAL และ acceptance criteria ชัด	มี source แต่บาง claim/ เกณฑ์ยังคลุมเครือ	แต่งข้อมูลหรือไม่มี trace
HTML และการออกแบบ	semantic, responsive, Light/Dark, focus/print และ brand constraints ครบ	หน้าหลักใช้ได้แต่ขาดบาง state/evidence	เน้นภาพอย่างเดียวหรือใช้ไม่ได้บน mobile/keyboard
Interaction	search/filter/empty/reset ทำงานร่วมกันและยังอ่านได้เมื่อปิด JS	happy path ใช้ได้แต่ state หรือ accessibility ไม่ครบ	ผลลวง, link เสีย หรือ content ฟัง JS ทั้งหมด
QA และ handoff	prompt/transcript/diff/test/release evidence ทำซ้ำได้และระบุ not tested	มีหลักฐานบางส่วนแต่ยังแยก build/dry-run/public ไม่ชัด	อ้างว่าผ่านหรือ deploy สำเร็จโดยไม่มีหลักฐาน

คะแนนเต็มไม่ได้หมายความว่า เป็นระบบมหาวิทยาลัยพร้อมใช้งานจริง งานนี้ประเมินกระบวนการเปลี่ยนข้อมูลเป็นเว็บที่ตรวจย้อนกลับได้ตามขอบเขตของ workshop

Definition of Done

- final page เปิดได้ที่ `/examples/course-website/`
- เนื้อหาตรง source inventory และไม่มีข้อมูล GAP ถูกแต่งเติม
- ลิงก์บทเรียน 00-10 และ resources สำคัญเปิดได้
- Light เป็นค่าเริ่มต้นและผู้ใช้เลือก Dark ได้
- search/filter/empty/reset ใช้ได้ด้วย mouse และ keyboard
- หน้า 390px ไม่มี horizontal overflow ที่เกิดจาก component
- prompt ที่ส่งจริงและ transcript ที่เผยแพร่ไม่ปะปนกับคำอธิบายย้อนหลัง
- QA findings มี evidence ก่อนและหลังแก้ หรือระบุว่า not tested
- PDF/source bundle รุ่นเผยแพร่เข้าถึงผ่าน Resources
- public/deploy claim มีหลักฐานจากปลายทางจริง; หากยังไม่ deploy ให้สถานะคงเป็น local หรือ dry-run

เมื่อครบรายการนี้ ผู้เรียนจะไม่ได้เพียงหน้าเว็บหนึ่งหน้า แต่มีร่องรอยการตัดสินใจที่อธิบายได้ว่า Codex ได้ข้อมูลอะไร ถูกส่งอย่างไร เปลี่ยนไฟล์ใด และผู้เรียนตรวจสอบผลด้วยหลักฐานใด

เอกสาร brief/evidence/handoff ที่ระดับรากได้รับการปรับให้ตรงกับ build โดยผู้ร่วมงานหลังรอบ 07 ๑ บันทึกการรวมงาน และ ผลทดสอบสุดท้าย ส่วน snapshot แต่ละรอบยังเก็บข้อมูลเดิมตามลำดับจริง

Lab เสริม — Export และ Deploy เว็บไซต์หลักสูตรไป Cloudflare

Lab นี้ต่อจาก [สร้างเว็บไซต์หลักสูตรด้วย Codex แบบ step by step](#) เมื่อผู้เรียนมีหน้า `index.html` , `style.css` และ `main.js` ที่ตรวจแล้ว เป้าหมายคือแยกหน้าเว็บออกเป็นแพ็คเกจ **Cloudflare Static Assets** ที่ตรวจง่าย แล้ว deploy ไปยัง Cloudflare account และ Worker name ของผู้เรียนเอง

ใช้เวลาประมาณ **45–60 นาทีสำหรับเรียนต่อด้วยตัวเอง** ไม่เพิ่มเวลา workshop ซึ่งยังคง **2 วัน × 4 ชั่วโมง รวม 8 ชั่วโมงและพักวันละ 10 นาที** เว็บไซต์นี้เป็นสื่อประกอบ **AI Web Studio — ออกแบบและพัฒนาเว็บไซต์ด้วย AI** ใน **977-121 Module: Website Design and Development**, มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต ไม่ใช่ระบบมหาวิทยาลัยอย่างเป็นทางการ

สิ่งที่ Lab นี้ deploy

แพ็คเกจที่ export มีโครงสร้างดังนี้:

```
artifacts/course-website-cloudflare/
├─ wrangler.jsonc
├─ README.md
├─ public/
│   ├─ index.html
│   ├─ style.css
│   ├─ main.js
│   ├─ 404.html
│   ├─ _headers
│   └─ fonts/
```

`wrangler.jsonc` ใช้ `assets.directory: "./public"` โดยไม่มี `main` , D1 binding หรือ API route แพ็คเกจนี้จึงเป็น static site หน้าเดียว: theme, search และ filters ทำงานใน browser แต่ไม่มีฐานข้อมูล, login, LMS, form backend หรือ API ของหลักสูตรเดิม Cloudflare รองรับ static assets โดยไม่ต้องมี Worker script และคืน 404 เมื่อไม่มี asset/script จัดการ request ตาม config ที่เลือก [Workers Static Assets](#) และ [Workers best practices](#)

ไฟล์ต้นฉบับใน `examples/course-website/` หรือ `practice/psu-course/` จะไม่ถูกแก้ไขโดย exporter สิ่งแบบ root-relative ที่ชี้ไป lesson, handbook และ resources ของ course shell จะถูกเปลี่ยนเป็น absolute URL ของเว็บไซต์หลักสูตรที่มีอยู่ เพื่อให้ลิงก์ยังเปิดแหล่งเดิมได้หลังแยก deploy ส่วน font files และ license ที่เกี่ยวข้องถูกคัดลอกมาอยู่ในแพ็คเกจและ CSS ยังโหลดจาก local deployment

ก่อนเริ่ม

- ทำ Step-by-step Lab จนได้ source ที่ต้องการเผยแพร่
- เปิด Terminal ที่ root ai-web-studio/ ซึ่งมี package.json และ lockfile
- ติดตั้ง dependency แล้วด้วย npm ci
- เลือก Worker name ของตน เช่น psu-course-nok-2026 ใช้ตัวพิมพ์เล็ก ตัวเลขและขีดกลาง โดยอักขระแรกและสุดท้าย ต้องเป็นตัวอักษรหรือตัวเลข และต้องไม่ใช่ชื่อ psu-course-website-demo ของ worked example
- อ่าน Setup Wrangler, login Cloudflare และ API token แล้วเลือก OAuth หรือ token ของบัญชีตนเอง
- ตรวจสอบ shared-infra ก่อนเปิด local service; Lab นี้ไม่เปิดพอร์ตใหม่และใช้ preview 3320/inspector 3322 ที่ project จองไว้เท่านั้น

Exporter ปฏิเสธชื่อ Worker ของ course shell และชื่อ vibe-to-production-academy แต่ผู้เรียนยังต้องไม่แก้ config หรือใช้ --name ไปชี้ target นั้นเอง การใช้ชื่อเฉพาะช่วยลดโอกาส deploy กับ Worker ของเพื่อน แต่ผู้เรียนยังต้องตรวจสอบ account และชื่อ target ก่อนคำสั่งจริงทุกครั้ง

ก่อนนำ prompt จริงไปใช้กับงานของผู้เรียน

Prompt 08-10 ด้านล่างเป็นข้อความที่ใช้จริงกับ worked example ให้ปรับบริบทก่อนส่ง:

- รอบ 08: เปลี่ยน source เป็น practice/psu-course และใช้ชื่อ Worker ของตน
- รอบ 09: แทนชื่อ psu-course-website-demo ทุกตำแหน่งด้วย target ของตน และยืนยันบัญชีจากผล whoami จริง ลบข้อความที่บอกว่า parent ตรวจสอบ auth/target หรือได้รับ code10007 แล้ว หากผู้เรียนยังไม่ได้ตรวจสอบ อย่าส่งข้อมูลยืนยันที่ไม่จริงให้ AI
- รอบ 10: ใช้ URL จากผล deploy ของตนและรายงานที่ตนรับเสร็จแล้ว ถ้ายังไม่มี deployment.json / 09-handoff.md ให้บันทึก URL/version จาก CLI ก่อน หรือปรับ prompt ให้อ่าน log จริงที่มีแทน

ชื่อบัญชีและชื่อ Worker เป็นบริบทให้ AI ตรวจสอบเป้าหมายได้ ส่วน token/password ไม่ต้องใส่ใน prompt ทุกครั้งที่แก้ไขชื่อหรือ source ให้ใช้คำเดียวกันตั้งแต่ export จนถึง deploy และ verify

ภาพรวมสิทธิ์แต่ละช่วง

ช่วง	สิ่งที่ AI/ผู้เรียนทำได้	สิ่งที่ยังไม่อนุญาต
Prepare	อ่าน source, export av artifacts/ , ตรวจสอบไฟล์, preview และ dry-run	login, deploy จริง, สร้าง/แก้ cloud resource
Deploy	deploy static package ไปยัง account + Worker name ที่ระบุ	D1, KV, R2, routes, custom domain, secrets, deletion

ช่วง	สิ่งที่ AI/ผู้เรียนทำได้	สิ่งที่ยังไม่อนุญาต
Verify	เปิด public URL แบบ read-only และตรวจ UI/links/404	ส่ง form, เปลี่ยน deployment หรือสร้างข้อมูล

Dry-run เป็น local packaging/config check ไม่ใช่การอนุญาต deploy การ login สำเร็จก็ไม่ใช่การอนุญาตให้ AI เลือก target แทนผู้เรียน

1. ตรวจ source และ export — 10 นาที

ตรวจ source ก่อน ถ้าเป็นงานที่ทำตามตัวอย่าง ให้ใช้ `practice/psu-course` ; ถ้าต้องการทำซ้ำ worked example ของผู้สอนจึงใช้ `examples/course-website`

```
pwd
find practice/psu-course -maxdepth 1 -type f -print
npm run export:course-site -- \
  --source practice/psu-course \
  --name psu-course-yourname
```

ผู้สอนที่ต้องการสร้างแพ็คเกจจาก worked example ใช้ source อีก path แต่ยังคงส่งชื่อ target ที่ตั้งใจ:

```
npm run export:course-site -- \
  --source examples/course-website \
  --name psu-course-yourname
```

แทน `psu-course-yourname` ด้วยชื่อเฉพาะของตน ห้ามคัดลอกชื่อ deploy ของผู้สอน เมื่อ command จบ ให้ตรวจ output โดยยังไม่ login/deploy:

```
rg --files artifacts/course-website-cloudflare
sed -n '1,160p' artifacts/course-website-cloudflare/wrangler.jsonc
sed -n '1,200p' artifacts/course-website-cloudflare/README.md
```

จุดตรวจ:

- `wrangler.jsonc` มีชื่อที่ผู้เรียนระบุและ `assets.directory` เป็น `./public`
- ไม่มี `main` , `d1_databases` , API binding, route หรือ custom domain
- `public/` มี HTML/CSS/JS, `404.html` , `_headers` , fonts และ license ตาม manifest/README
- HTML ไม่อ้างไฟล์ใน `practice/` หรือ `examples/` ซึ่งจะไม่มีบน deployment

- ลิงค์ lesson/handbook/resources เป็น absolute URL ของ course origin ที่ระบุใน README
- source directory เดิมยังมีเนื้อหาเหมือนก่อน export

หาก config ใช้ `not_found_handling: "404-page"` ให้ยืนยันว่ามี `public/404.html` ; Cloudflare ระบุว่าโหมดนี้ส่ง custom page พร้อมสถานะ 404 ส่วน `single-page-application` จะตอบ `index.html` สำหรับ path ที่ไม่ตรง asset จึงไม่เหมาะกับเป้าหมาย 404 ของ Lab นี้ Static Assets routing

Prompt 08 — เตรียมและตรวจ export โดยยังไม่ deploy

เตรียม deploy เว็บไซต์ที่สร้างใน Lab ก่อนหน้าเป็น Cloudflare Workers Static Assets แบบแยก
อ่าน `scripts/export-course-site.mjs`, `examples/course-website/index.html`, `style.css`, `main.js` และ `artifacts/course-deploy/research.md`
ทำงานจาก repository นี้ รัน `npm run export:course-site -- --source examples/course-website --name psu-course-yourname`
ตรวจ `artifacts/course-website-cloudflare/wrangler.jsonc` ว่ามี `assets.directory ./public`, `404-page` พร้อม `404.html` และไม่มี `main`, `binding`, `D1`, `routes`, `secrets` หรือ `cron`
ตรวจว่าลิงก์กับเพี้ยนไปยัง origin ของเว็บไซต์หลักสุด ส่วน `CSS/JS/fonts` อยู่ในชุด export และเก็บ license ฟอนต์ครบ
รัน `npx wrangler deploy --config artifacts/course-website-cloudflare/wrangler.jsonc --dry-run` โดยบันทึกผลใน `artifacts/course-deploy/site-dry-run.txt`
บันทึกผลตรวจแบบสั้นใน `artifacts/course-deploy/08-check.md` รวมไฟล์ที่ได้และสิ่งที่ `dry-run` พิสูจน์/ยังไม่พิสูจน์
อนุญาตสร้างเฉพาะชุด export และหลักฐานตาม path ข้างต้น ไม่แก้ source ต้นฉบับหรือ config หลัก ไม่อ่าน/แสดง token หรือ `auth-private.txt` ไม่ `login/logout` และยังไม่ deploy สาธารณะ

Prompt นี้ต้องอนุญาตเฉพาะ `local read/write` ในแพ็คเกจและ `--dry-run` พร้อมสั่งให้หยุดก่อน `login` หรือ `deploy` จริง
รายงาน expected output แยกจากสิ่งที่ command พิสูจน์ได้

ภาพหลักฐานการ export

```
CODEX CLI 0.154.0 / STEP 08 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <course-repository> --sandbox workspace-write --json --output-last-message artifacts/course-deploy/08-final.txt - < public/labs/course-website-deploy/prompts/08-prepare.txt

PROMPT SENT
เตรียม deploy เว็บไซต์ที่สร้างใน Lab ก่อนหน้าเป็น Cloudflare Workers Static Assets แบบแยก
อ่าน scripts/export-course-site.mjs, examples/course-website/index.html, style.css, main.js และ artifacts/course-deploy/research.md
ทำงานจาก repository นี้ รัน npm run export:course-site -- --source examples/course-website --name psu-course-yourname
ตรวจ artifacts/course-website-cloudflare/wrangler.jsonc ว่ามี assets.directory ./public, 404-page พร้อม 404.html และไม่มี main, binding, D1, routes, secrets หรือ cron
ตรวจว่าลิงก์กับเพี้ยนไปยัง origin ของเว็บไซต์หลักสุด ส่วน CSS/JS/fonts อยู่ในชุด export และเก็บ license ฟอนต์ครบ
รัน npx wrangler deploy --config artifacts/course-website-cloudflare/wrangler.jsonc --dry-run โดยบันทึกผลใน artifacts/course-deploy/site-dry-run.txt
บันทึกผลตรวจแบบสั้นใน artifacts/course-deploy/08-check.md รวมไฟล์ที่ได้และสิ่งที่ dry-run พิสูจน์/ยังไม่พิสูจน์
อนุญาตสร้างเฉพาะชุด export และหลักฐานตาม path ข้างต้น ไม่แก้ source ต้นฉบับหรือ config หลัก ไม่อ่าน/แสดง token หรือ auth-private.txt ไม่ login/logout และยังไม่ deploy สาธารณะ

CAPTURE READY 08 prompt
```

สอ 08: prompt จริงที่ให้ Codex เตรียมแพ็คเกจและตรวจขอบเขต

เปิดภาพขนาดใหญ่

```
CODEX CLI 0.154.0 / STEP 08 / RESULT
Read-only replay of a real CLI run. Not Codex Desktop.

Recorded commands: 9 | File-change events: 1

CODEX FINAL RESPONSE (verbatim; local paths shortened)
เตรียมชุด deploy แบบแยก เรียนรู้แล้ว โดยยังไม่ได้ login หรือ deploy สาธารณะ

- [wrangler.jsonc](<course-repository>/artifacts/course-website-cloudflare/wrangler.jsonc) ใช้ './public', '404-page' และไม่มี 'main', binding, D1, routes, secrets หรือ cron
- ลิงก์เนื้อหา 20 จุดขึ้นกับ origin หลักสุด
- CSS, JS, ฟอนต์ WOFF2 และ OFL license อยู่ใน export ครบ
- Dry-run ผ่านด้วย exit code 0 และรายงาน 'No bindings found'
- Source และ config หลักมี hash ไม่เปลี่ยน

หลักฐาน: [site-dry-run.txt](<course-repository>/artifacts/course-deploy/site-dry-run.txt) และ [08-check.md](<course-repository>/artifacts/course-deploy/08-check.md)

Full prompts / commands / outputs: transcripts/08.txt

CAPTURE READY 08 result
```

สอบ 08: คำตอบจริงหลัง export และ dry-run ผ่าน

เปิดภาพขนาดเต็ม

ภาพควรแสดง command, รายชื่อไฟล์และส่วนสำคัญของ config จาก terminal replay ที่มาจาก log จริง ปิดบัง username/path ที่ไม่จำเป็น และห้ามสร้างภาพ OAuth, Dashboard หรือ token ปลอม ภาพหนึ่งภาพไม่พิสูจน์ว่า assets ทุกไฟล์เปิดได้

2. Preview แเพ็กเกจเดียวกับที่จะ deploy — 5–10 นาที

หยุด preview เดิมของตนเองด้วย `Ctrl+C` ก่อน เพราะแพ็กเกจนี้ใช้พอร์ตคู่เดียวกัน ถ้าพอร์ตมีเจ้าของอื่นให้ตรวจ shared-infra และแก้การชนก่อน ไม่เลือกพอร์ตอื่นโดยเดา รับจาก root ของ source เดิม:

```
npx wrangler dev \
  --config artifacts/course-website-cloudflare/wrangler.jsonc \
  --ip 127.0.0.1 --port 3320 \
  --inspector-ip 127.0.0.1 --inspector-port 3322
```

เปิด `http://127.0.0.1:3320/` นี่คือ `public/` ในแพ็กเกจที่จะ deploy โดยตรง ไม่ใช่หน้า `/examples/course-website/` ของ course shell หยุดด้วย `Ctrl+C` เมื่อเสร็จ เพื่อให้พอร์ตกลับมาใช้กับ preview ของหลักสุดได้

ตรวจสอบอย่างน้อย:

- desktop และ 390px เปิดหน้าแรกได้
- first visit เป็น Light; toggle ไป Dark แล้ว reload ยังจำค่าได้
- search, Day 1/Day 2, empty state และ reset ทำงาน
- lesson links ออกไปยัง course origin ที่ถูกต้อง
- ไม่มีปุ่มหรือข้อความที่อ้างว่ามี API/login/backend
- เปิด path ที่ไม่มีจริงแล้วได้ `404.html` ที่ช่วยกลับหน้าแรก

บันทึกสิ่งที่เห็นเป็น local preview evidence อย่างเรียกว่า public deployment

3. Login และยืนยัน account — 5 นาที

ใช้วิธีใน คู่มือ Cloudflare authentication ไม่บันทึกภาพหน้าจอที่มี token, authorization code, cookies หรือข้อมูล account ที่ไม่จำเป็น

```
npx wrangler --version
npx wrangler login
npx wrangler whoami
```

บนเครื่องที่ browser callback ไม่เหมาะ ให้ใช้ device flow หรือ API token ตามคู่มือเดิม ตรวจสอบ account ที่ `whoami` แสดงกับบัญชีสำหรับแบบฝึกหัด หากไม่ตรงให้หยุดก่อน `dry-run/deploy`

หลักฐานที่เก็บได้คือ Wrangler version, วิธี auth ที่เลือกโดยไม่เปิดค่า secret และ account label/ID ที่ redact ตามความจำเป็น `whoami` ไม่พิสูจน์ว่าสามารถ deploy Worker ชื่อนั้นได้

4. Dry-run ด้วย config ที่ export — 5 นาที

รันจาก root `ai-web-studio/` ด้วย path เดิมทุกครั้ง:

```
npx wrangler deploy \
  --config artifacts/course-website-cloudflare/wrangler.jsonc \
  --dry-run
```

ตรวจสอบ output ว่า Wrangler อ่าน config และ asset directory ที่ตั้งใจ ไม่มี Worker code หรือ binding ที่ไม่ได้ขอ Dry-run อาจตรวจพบ config/ไฟล์ผิด แต่ไม่เขียน Worker และไม่พิสูจน์ credential, Worker name availability หรือ public URL ตัวเลือก `--config` และ `--dry-run` เป็นส่วนของ Wrangler deploy workflow; ตรวจสอบ syntax กับ CLI รุ่นที่ติดตั้ง และ Wrangler commands ก่อนใช้ในรุ่นอบรมใหม่

ภาพหลักฐาน dry-run

```
WRANGLER 4.130.0 / DRY RUN
Read-only replay of a real CLI run. Not Codex Desktop.

WRANGLER 4.130.0 / ACTUAL COMMAND OUTPUT (paths shortened)
$ npx wrangler deploy --config artifacts/course-website-cloudflare/wrangler.jsonc --dry-run

🔥 wrangler 4.130.0 (update available 4.131.0)

🌟 Read 12 files from the assets directory <course-repository>/artifacts/course-website-cloudflare/public
Total Upload: 0.31 KiB / gzip: 0.22 KiB
No bindings found.
--dry-run: exiting now.

Cloudflare collects anonymous telemetry about your usage of Wrangler. Learn more at https://github.com/cloudflare/workers-sdk/tree/main/packages/wrangler/telemetry.md

CAPTURE READY 08 dry-run
```

Wrangler dry-run จาก log จริง: อ่าน assets และไม่มี bindings โดยยังไม่ deploy

เปิดภาพขนาดเต็ม

ภาพควรเห็นคำสั่ง `--config` และ `--dry-run` ควบ พร้อม summary ที่ไม่เปิด secret ภาพนี้ใช้ยืนยันว่ารัน dry-run กับ artifact ใด ไม่ใช้ยืนยันว่า deployment จริงสำเร็จ

5. อนุญาตและ deploy ไป target ของผู้เรียน — 10 นาที

ก่อนคำสั่งจริง เขียน target ลง worksheet ให้ครู:

Cloudflare account ที่ whoami แสดง: _____
Worker name ใน wrangler.jsonc: _____
Config: artifacts/course-website-cloudflare/wrangler.jsonc
Resource ที่อนุญาต: static Worker/Assets target ใหม่เท่านั้น
Resource ที่ไม่อนุญาต: D1, KV, R2, routes, domain, secrets, deletion

เมื่อ account และชื่อถูกต้องแล้ว จึงรันคำสั่งจริงด้วย config เดียวกับ dry-run:

```
npx wrangler deploy \  
  --config artifacts/course-website-cloudflare/wrangler.jsonc
```

คัดลอก public URL จาก output ที่ Wrangler แสดง อย่าคำนวณหรือเดา URL จาก Worker name เพราะ account อาจใช้ subdomain/config ต่างกัน Worked example ของผู้สอนอาจ override CLI ด้วย `--name psu-course-website-demo` ; ชื่อนั้นสงวนสำหรับหลักฐานของผู้สอน ผู้เรียนต้องใช้ชื่อของตนที่อยู่ใน exported config

Prompt 09 — อนุญาต deploy target ใหม่ที่ระบุเท่านั้น

Deploy เว็บไซต์ Static Assets ที่เตรียมไว้ไป Cloudflare ด้วย target ที่ระบุเท่านั้น

ผู้ใช้อนุญาตให้ deploy เว็บไซต์อย่างนี้แล้ว parent ตรวจสอบ existing CLI authentication พร้อมและ Worker psu-course-website-demo ยังไม่มีในบัญชี (API code10007)

อ่าน artifacts/course-website-cloudflare/wrangler.jsonc, artifacts/course-deploy/08-check.md และ artifacts/course-deploy/site-dry-run-final.txt

Parent เพิ่มการป้องกันชื่อ course หลักและชื่อผลิตภัณฑ์ใน exporter หลังรอบ 08 และทดสอบผ่านแล้ว;

ชุด public และ config default ยังเหมือนเดิม

อนุญาตให้สร้าง/อัปเดต Worker ชื่อ psu-course-website-demo ในบัญชีที่ CLI ใช้อยู่ สำหรับเว็บไซต์นี้เท่านั้น

รัน npx wrangler deploy --config artifacts/course-website-cloudflare/wrangler.jsonc --name psu-course-website-demo และบันทึก output จริงไว้ artifacts/course-deploy/site-deploy.txt

ห้ามใช้ config หลักที่ root ห้าม deploy ชื่อ vibe-to-production-academy ห้ามเปลี่ยน D1, KV, R2, routes, cron, secrets, custom domain หรือทรัพยากรอื่น

ไม่ login/logout และไม่อ่าน/แสดง auth-private.txt, target-check-private.txt หรือ token

บันทึก artifacts/course-deploy/09-handoff.md โดยนำ public URL และ Current Version ID จากผล CLI จริง

แยก deploy สำเร็จจากการตรวจสอบเว็บซึ่งยังต้องทำรอบ 10

ถ้าคำสั่งล้มเหลว ให้รายงาน error และหยุดโดยไม่เปลี่ยน target เอง

Prompt deploy ต้องระบุ account label/ID ที่ตรวจแล้ว, Worker name, config path และอนุญาต command เดียวอย่างชัดเจน พร้อมห้ามฐานข้อมูล/bindings/routes/domains/secrets/deletion หาก target ไม่ตรงหรือ CLI ขอการเปลี่ยนแปลงเกินขอบเขต ให้หยุดและรายงานแทนการเลือกเอง

กรณีจริงที่พบ: คำสั่งใน sandbox ไม่สำเร็จ

รอบ 09 ของ Codex ส่งคำสั่ง deploy จริง แต่จบด้วย fetch failed และมี error เขียน Wrangler debug log เพราะ filesystem ใน sandbox เป็น read-only จึงไม่มี URL/version ให้ยืนยันจากรอบนั้น ผู้ร่วมงานรับคำสั่งเดิมด้วย config และชื่อ Worker เดิมจาก terminal ที่มีสิทธิ์ แล้วเก็บ log retry แยกไว้ การ retry นี้เป็นคำสั่งของผู้ร่วมงาน ไม่ใช่ผลสำเร็จของ Codex รอบ 09

ถ้าผู้เรียนพบเช่นเดียวกัน ให้แยกตรวจสอบ network และสิทธิ์ของ terminal/sandbox อ่าน error ก่อนแก้ แล้วใช้ terminal ที่ได้รับสิทธิ์กับ target เดิม ไม่แก้ด้วยการเลือกบัญชี/Worker อื่นหรือใส่ token ลง prompt และไม่ถือว่าเปลี่ยน timeout แล้วจะสำเร็จเสมอ

```
CODEX CLI 0.154.0 / STEP 09 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <course-repository> --sandbox workspace-write --json --output-last-message artifacts/course-deploy/09-final.txt - < public/labs/course-website-deploy /prompts/09-deploy.txt

PROMPT SENT
Deploy เว็บไซต์ Static Assets ที่เตรียมไว้ไป Cloudflare ด้วย target ที่ระบุเท่านั้น
ถ้าเขียนผิดให้ deploy เสร็จแล้วยังได้ parent ตรวจสอบ existing CLI authentication พร้อมและ Worker psu-course-website-demo ยังไม่มีในบัญชี (API code10007)
อ่าน artifacts/course-website-cloudflare/wrangler.jsonc, artifacts/course-deploy/08-check.md และ artifacts/course-deploy/site-dry-run-final.txt
parent ให้คำแนะนำคือ course หลักและชื่อคือดูใน exporter หลังรอบ 08 และทดสอบผ่านแล้ว; จาก public และ config default ยังเหมือนเดิม
อนุญาตให้สร้าง/รัน task Worker ชื่อ psu-course-website-demo ในบัญชีที่ CLI ให้อยู่ สำหรับ รันสคริปต์นี้ เท่านั้น
รัน npx wrangler deploy --config artifacts/course-website-cloudflare/wrangler.jsonc --name psu-course-website-demo และบันทึก output จดไว้ artifacts/course-deploy/site-d
eploy.txt
ห้ามใช้ config หลักที่ root ห้าม deploy ชื่อ vibe-to-production-academy ห้ามเปลี่ยน D1, KV, R2, routes, cron, secrets, custom domain หรือทรัพยากรอื่น
ไม่ login/logout และไม่อ่าน/แสดง auth-private.txt, target-check-private.txt หรือ token
บันทึก artifacts/course-deploy/09-handoff.md โดยนำ public URL และ Current Version ID จากผล CLI จดไว้ แยก deploy สำหรับออกจากตรวจสอบ วัณซึ่งยังต้องทำรอบ 10
ถ้าคำสั่งล้มเหลว ให้รายงาน error และหยุดโดยไม่เปลี่ยน target เอง

CAPTURE READY 09 prompt
█
```

รอบ 09: prompt ระบุ target และอนุญาต deploy เฉพาะเว็บไซต์อย่าง

เปิดภาพขนาดเต็ม

```
CODEX CLI 0.154.0 / STEP 09 / RESULT
Read-only replay of a real CLI run. Not Codex Desktop.

Recorded commands: 4 | File-change events: 1

CODEX FINAL RESPONSE (verbatim; local paths shortened)
Deploy ไม่สำเร็จและหยุดแล้วตามเงื่อนไข

- คำสั่งจบด้วย exit code 1: 'fetch failed' จากปัญหา connectivity
- มี error รองจาก filesystem แบบ read-only ขณะ Wrangler พยายามเขียน debug log
- บันทึก CLI output จดไว้ที่ [site-deploy.txt](<course-repository>/artifacts/course-deploy/site-deploy.txt)
- ไม่สร้าง '09-handoff.md' เพราะไม่มี Public URL หรือ Current Version ID
- ไม่ retry, ไม่เปลี่ยน target, ไม่ login/logout และไม่แตะทรัพยากร Cloudflare อื่น
- รูปแบบคำสั่งตรงกับ [Wrangler deploy documentation](https://developers.cloudflare.com/workers/wrangler/commands/workers/)

Full prompts / commands / outputs: transcripts/09.txt

CAPTURE READY 09 result
█
```

รอบ 09: Codex รายงาน deploy ล้มเหลว จึงยังไม่มี URL/version จากรอบนี้

เปิดภาพขนาดเต็ม

```
WRANGLER 4.130.0 / PARENT TERMINAL RETRY
Read-only replay of a real CLI run. Not Codex Desktop.

WRANGLER 4.130.0 / ACTUAL COMMAND OUTPUT (paths shortened)
Parent terminal retry after the Codex sandbox attempt failed.
$ npx wrangler deploy --config artifacts/course-website-cloudflare/wrangler.jsonc --name psu-course-website-demo

🌀 wrangler 4.130.0 (update available 4.131.0)

🌀 Building list of assets...
🌟 Read 12 files from the assets directory <course-repository>/artifacts/course-website-cloudflare/public
🌀 Starting asset upload...
🌀 Found 10 new or modified static assets to upload. Proceeding with upload...
+ /404.html
+ /fonts/NotoSansThai-OFL.txt
+ /fonts/NotoSansThai-Bold.woff2
+ /fonts/Inter-Bold.woff2
+ /main.js
+ /style.css
+ /fonts/NotoSansThai-Regular.woff2
+ /fonts/Inter-OFL.txt
+ /index.html
+ /fonts/Inter-Regular.woff2
Uploaded 4 of 10 assets
Uploaded 7 of 10 assets
Uploaded 10 of 10 assets
🌟 Success! Uploaded 10 files (2.22 sec)

Total Upload: 0.31 KiB / gzip: 0.22 KiB
Uploaded psu-course-website-demo (9.04 sec)
Deployed psu-course-website-demo triggers (5.13 sec)
https://psu-course-website-demo.rawinnipa-soft.workers.dev
Current Version ID: 21ef7990-e844-4e57-93ab-53224712b04c

CAPTURE READY 09 deploy
█
```

รอบ retry จาก parent terminal: คำสั่งเดิมสำเร็จและได้ URL/version จริง

เปิดภาพขนาดเต็ม

ภาพควรเป็น terminal replay จาก deploy log จริง เห็น target, upload/deploy status และ URL ที่ Wrangler คั่น โดย redact account ID เมื่อไม่จำเป็น ไม่ใช้ภาพ Dashboard ที่จัดฉากหรือ OAuth flow จำลอง และอย่าใส่ URL หาก command ยังไม่คั่น URL จริง

6. Verify public URL — 10 นาที

เก็บ URL ที่ Wrangler คั่นไว้ในตัวแปร Bash เพื่อหลีกเลี่ยงการพิมพ์หลายครั้ง:

```
COURSE_SITE_URL='https://URL-FROM-WRANGLER'
curl -fsSI "$COURSE_SITE_URL/"
curl -sS -o /dev/null -w '%{http_code}\n' "$COURSE_SITE_URL/path-that-does-not-exist"
```

แทนค่า placeholder ด้วย URL จริงก่อนรัน จากนั้นเปิด URL ใน browser และตรวจ:

สถานะ	สิ่งที่ต้องสังเกต	หลักฐานที่เก็บ
Desktop Light	หน้าแรก, hierarchy, fonts และ first-visit theme	URL + viewport + screenshot
Desktop Dark	toggle, readable text และ reload persistence	screenshot ก่อน/หลัง reload
Mobile 390px	ไม่มี horizontal overflow; controls ใช้ได้	viewport + screenshot
Search/filter	ผล, จำนวน, empty state และ reset	คำค้น/ตัวกรองที่ใช้
Lesson links	เปิด course origin ที่ export s-ๆ	destination URL อย่างน้อย 2 ลิงก์
Unknown path	ตอบด้วย 404 behavior ของแพ็คเกจ	status + หน้า 404
Backend/API	ไม่มีฟังก์ชันปลอมและไม่อ้างว่ามี API	source/UI observation

ถ้า origin ส่ง fallback HTML พร้อม HTTP status ที่ต่างจากที่คาด ให้บันทึก status จริงและเทียบ _headers /platform behavior ก่อนสรุป อย่าเปลี่ยนผลให้เป็น 404 เพียงเพราะเห็นหน้า 404

รับตัวตรวจจาก terminal ที่มี browser tooling

เมื่อใช้ source เติม ให้รับคำสั่งนี้จาก root หลังใส่ URL จริงในตัวแปรข้างต้น:

```
node scripts/check-course-site.mjs --url "$COURSE_SITE_URL"
```

ตัวตรวจเขียน artifacts/course-deploy/site-checks.json และภาพใน public/images/course-deploy/ ตรวจสอบ asset hashes รวม license, 404, controls, 1440/390px ทั้งสองอัน, accessibility ระดับ serious/critical และตัวอย่างการเปิดบทเรียนจริง ถ้าเปลี่ยน course origin ตอน export ให้เพิ่ม --course-origin https://YOUR-COURSE-ORIGIN ด้วย ข้อมูลนี้เป็นผลของ browser runner; Codex สอบ 10 อ่านผลที่รันเสร็จแล้วเพื่อสรุปเท่านั้น ผู้ใช้ ZIP อย่างเดียว ตรวจสอบด้วย browser และ worksheet แทน ไม่จำเป็นต้องติดตั้ง test dependencies

Prompt 10 — ให้ AI ตรวจรายงาน public site แบบ read-only

ตรวจหลักฐานหลัง deploy เว็บไซต์ตัวอย่างขึ้น Cloudflare โดยไม่เปลี่ยน deployment
อ่าน artifacts/course-deploy/09-handoff.md, deployment.json, site-checks.json และ scripts/check-course-site.mjs
ผล browser มาจาก parent ที่รัน node scripts/check-course-site.mjs --url https://psu-course-website-demo.rawinnipa-soft.workers.dev แล้ว พร้อมภาพใน public/images/course-deploy/
ตรวจว่า URL/version มาจาก CLI retry จริง; Codex รอบ 09 ล้มเหลวและ parent เป็นผู้ retry ไม่อ้างว่า Codex 09 deploy สำเร็จ
ตรวจหลักฐาน asset hashes รวม font licenses, HTTP 404 สำหรับ path หายและ /api/health, Light/default, search/filter/reset, Light/Dark ที่ 1440/390, theme persistence และลิงก์บทเรียนไป origin ที่ตั้งใจ พร้อม HTTP สำเร็จ
เขียน artifacts/course-deploy/10-verification.md แยกสิ่งที่ report ยืนยันและข้อจำกัด เช่น ไม่ได้ทดสอบ custom domain, API/LMS หรือทุก browser
ไม่รัน browser/deploy/login เพิ่มเติมในรอบนี้ ไม่แก้ source หรือ config ไม่อ่าน/แสดง token หรือไฟล์ private
สรุปชื่อ Worker, public URL, version, จำนวน checks และขั้นตอนที่ผู้เรียนควรทำเมื่อแก้ source แล้วจะ deploy อีกครั้ง

Prompts รอบนี้ อ่านรายงานและ source ของตัวตรวจ ไม่รัน browser หรือ deploy เพิ่ม รายงานต้องแยกสิ่งที่ตรวจผ่าน กับสิ่งที่ไม่ได้ทดสอบ และไม่อ้างว่าทุกปลายทางผ่านจากการเปิดเพียงลิงก์ตัวแทน

ภาพหลักฐาน public site

```
CODEX CLI 0.154.0 / STEP 10 / PROMPT
Read-only replay of a real CLI run. Not Codex Desktop.

$ codex exec -C <course-repository> --sandbox workspace-write --json --output-last-message artifacts/course-deploy/10-final.txt - < public/labs/course-website-deploy/prompts/10-verify.txt

PROMPT SENT
ตรวจหลักฐานหลัง deploy เว็บไซต์ตัวอย่างขึ้น Cloudflare โดยไม่เปลี่ยน deployment
อ่าน artifacts/course-deploy/09-handoff.md, deployment.json, site-checks.json และ scripts/check-course-site.mjs
ผล browser มาจาก parent ที่รัน node scripts/check-course-site.mjs --url https://psu-course-website-demo.rawinnipa-soft.workers.dev แล้ว พร้อมภาพใน public/images/course-deploy/
ตรวจว่า URL/version มาจาก CLI retry จริง; Codex รอบ 09 ล้มเหลวและ parent เป็นผู้ retry ไม่อ้างว่า Codex 09 deploy สำเร็จ
ตรวจหลักฐาน asset hashes รวม font licenses, HTTP 404 สำหรับ path หายและ /api/health, Light/default, search/filter/reset, Light/Dark ที่ 1440/390, theme persistence และลิงก์บทเรียนไป origin ที่ตั้งใจ พร้อม HTTP สำเร็จ
เขียน artifacts/course-deploy/10-verification.md แยกสิ่งที่ report ยืนยันและข้อจำกัด เช่น ไม่ได้ทดสอบ custom domain, API/LMS หรือทุก browser
ไม่รัน browser/deploy/login เพิ่มเติมในรอบนี้ ไม่แก้ source หรือ config ไม่อ่าน/แสดง token หรือไฟล์ private
สรุปชื่อ Worker, public URL, version, จำนวน checks และขั้นตอนที่ผู้เรียนควรทำเมื่อแก้ source แล้วจะ deploy อีกครั้ง

CAPTURE READY 10 prompt
```

สอ 10: ให้ Codex อ่านหลักฐานการตรวจจาก parent browser runner

เปิดภาพขนาดเต็ม

```
CODEX CLI 0.154.0 / STEP 10 / RESULT
Read-only replay of a real CLI run. Not Codex Desktop.

Recorded commands: 5 | File-change events: 1

CODEX FINAL RESPONSE (verbatim; local paths shortened)
ตรวจ สร้างและสร้างรายงานที่ [10-verification.md](<course-repository>/artifacts/course-deploy/10-verification.md)

สรุปหลักฐาน:

- Worker: `psu-course-website-demo`
- Public URL: `https://psu-course-website-demo.rawinnipa-soft.workers.dev`
- Version: `21ef7990-e844-4e57-93ab-53224712b04c`
- Deployment สำเร็จจาก `parent terminal retry`; Codex รอบ 09 ล้มเหลวและไม่ได้ deploy สำเร็จ
- ผ่าน 26 check gates, `failures: []`
- Asset 9 รายการตอบ 200 และ SHA-256 ตรงกัน รวม font licenses ทั้งสองไฟล์
- ผ่าน 404, Light default, search/filter/reset, theme persistence และ 4 viewport/theme combinations
- ลิงก์พร้อมขึ้น 11 รายการเข้าไป origin ที่ตั้งใจ; ลิงก์แรกผ่าน navigation จริงด้วย HTTP 200
- ระบุดึงจากชื่อ custom domain, API/LMS, browser ขึ้น และ HTTP ของอีก 10 ลิงก์ไว้แล้ว

รอบนี้ไม่ได้รัน browser/deploy/login เพิ่ม ไม่แก้ source/config และสร้างเฉพาะรายงาน verification ดังกล่าว.

Full prompts / commands / outputs: transcripts/10.txt

CAPTURE READY 10 result
```

รอบ 10: คำตอบตรวจหลักฐาน โดยไม่อ้างว่ารัน browser เอง

เปิดภาพขนาดเต็ม

ภาพควรมี URL จริง, viewport/theme และสิ่งที่กำลังพิสูจน์ ภาพหน้าจอ browser ไม่พิสูจน์ HTTP status, persistence ทุกกรณี หรือ source integrity จึงต้องอยู่คู่กับ command log และ worksheet

Day 1 — เริ่มต้น ออกแบบ และสร้างฐานงาน

4 ชั่วโมง รวมพัก 10 นาที: เลือกกิจกรรมจากบท 00-05 เพื่อเตรียมเครื่องมือ วางบริบท และเริ่มทำเว็บไซต์

Day 2 — ขยายงาน ตรวจสอบ และส่งมอบ

4 ชั่วโมง รวมพัก 10 นาที: เลือกกิจกรรมจากบท 06-10 เพื่อทำงานกับข้อมูล ประสานการนับเงิน การ deploy และการส่งมอบ

เส้นทางบทเรียน

บท 00-05 อยู่ในเส้นทาง Day 1 และบท 06-10 อยู่ในเส้นทาง Day 2 เพื่อใช้เลือกเนื้อหาใน workshop

ค้นหาบทเรียน

เลือกเส้นทาง workshop

ทั้งหมด

Day 1

Day 2

ล้างการค้นหาและตัวกรอง

พบบทเรียน 11 จาก 11 บท

Day 1: บท 00-05

00 — เตรียมเครื่องและเข้าใจว่าเว็บทำงานอย่างไร

เตรียมเครื่องมือและทำความเข้าใจพื้นฐานการทำงานของเว็บก่อนเริ่มสร้างงาน

[อ่านบทเรียน 00](#)

01 — พื้นฐาน Vibe Coding สำหรับเว็บที่ส่งมอบได้จริง

เรียนรู้พื้นฐาน Vibe Coding เพื่อสร้างเว็บที่พร้อมส่งมอบ

[อ่านบทเรียน 01](#)

02 — Agent Skills, MCP และ Plugins โดยไม่ทำให้บริบทพัง

ทำความเข้าใจ Agent Skills, MCP และ Plugins พร้อมดูแบบรับการทำงาน

[อ่านบทเรียน 02](#)

03 — Specs และ Prompts สำหรับเว็บบริษัท, Catalog และเว็บ animation

เขียน specs และ prompts สำหรับเว็บบริษัท

04 — ออกแบบ, ภาพ และคุณภาพก่อนเขียนหน้าเว็บ

วางแผนออกแบบ ภาพ และเกณฑ์คุณภาพก่อนเริ่ม

05 — เว็บบริษัท: contact form ที่รับกับ demo นี้ได้จริง

สร้างเว็บบริษัทและ contact form ที่ทำงานกับ

เว็บไซต์ Cloudflare ที่เปิดจริง: 1440px Light

เปิดภาพขนาดเต็ม

AI WEB STUDIO

977-121 Module: Website Design and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

เริ่มจากบทเรียน 00

[ดูบทเรียนทั้งหมด](#)

ภาพรวมเส้นทางการทำงาน

- 01 Brief
- 02 Build
- 03 Verify
- 04 Publish

เป้าหมายของ workshop

เว็บไซต์ Cloudflare ที่เปิดจริง: 1440px Dark

เปิดภาพขนาดเต็ม

เว็บไซต์ตัวอย่างประกอบการเรียน

977-121 Module: Website Design and Development

มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตภูเก็ต

[แผน workshop](#) [บทเรียน](#) [แหล่งเรียนต่อ](#)

[คู่มือการเรียน](#) [ทรัพยากร](#)

เปลี่ยนเป็นโหมดมืด

AI WEB STUDIO

977-121 Module: Website Design and Development

ออกแบบและพัฒนาเว็บไซต์ด้วย AI

เส้นทางเรียนรู้เพื่อเริ่มสร้างเว็บไซต์ด้วย AI ตั้งแต่เครื่องมือและพื้นฐาน ไปจนถึงการส่งมอบที่ตรวจสอบได้

เริ่มจากบทเรียน 00

[ดูบทเรียนทั้งหมด](#)

ภาพรวมเส้นทางการทำงาน

01 Brief

เว็บไซต์ Cloudflare ที่เปิดจริง: 390px Light

[เปิดภาพขนาดเต็ม](#)



เว็บไซต์ Cloudflare ที่เปิดจริง: 390px Dark

เปิดภาพขนาดเต็ม

ทางเลือกเมื่อมีเฉพาะแพ็คเกจ ZIP

หากผู้เรียนดาวน์โหลดเฉพาะแพ็คเกจ Cloudflare จาก Resources และไม่ได้ใช้ source repository ให้แตก ZIP แล้วเปิด Terminal ในโฟลเดอร์ที่มี `wrangler.jsonc` กับ `public/` ต้องมี Node.js/npm เพื่อใช้ `npx` แต่ไม่ต้องมี Python หรือ Node project เพิ่ม

```
cd /path/to/psu-course-website
npx --yes wrangler@4.130.0 --version
npx --yes wrangler@4.130.0 login
npx --yes wrangler@4.130.0 whoami
npx --yes wrangler@4.130.0 deploy --config wrangler.jsonc --dry-run
```

หลังตรวจ account, Worker name ใน config และ dry-run แล้ว คำสั่ง deploy จริงคือ:

```
npx --yes wrangler@4.130.0 deploy --config wrangler.jsonc
```

ใช้ Wrangler รุ่นเดียวกันใน login, dry-run และ deploy เพื่อให้หลักฐานเทียบกันได้ `npx --yes` อนุญาตให้ npm ตาหน้าโหลด package ที่ pin ไว้ จึงต้องใช้อินเทอร์เน็ตและตรวจ package/version ก่อนรัน ในห้องเรียนที่มี source เต็มให้ใช้ dependency จาก lockfile เป็นทางเลือก

ปลายทาง `workers.dev` เป็น URL สาธารณะของ account ให้เผยแพร่เฉพาะข้อมูลสมมติหรือข้อมูลหลักสูตรที่อนุญาตแล้ว และไม่เพิ่ม custom domain/routes ใน Lab นี้ [workers.dev routing](#)

แก้ไขและ redeploy อย่างปลอดภัย

หลังพบปัญหา อย่าแก้ไขไฟล์ใน `artifacts/course-website-cloudflare/public/` แล้วลืมนำ source เพราะ export สอน ถัดไปจะกับการแก้ นั้น ให้แก้ `practice/psu-course/` หรือ source copy ที่ตรวจแล้ว จากนั้น:

1. ตรวจ diff และรับ local QA ของ Step-by-step Lab
2. export ใหม่ด้วย `--source` และ `--name` เดิม
3. ตรวจ `wrangler.jsonc` และไฟล์ output อีกครั้ง
4. รัน dry-run ด้วย config เดิม
5. deploy ไป Worker name เดิมหลังยืนยัน account/target
6. verify public URL ใหม่และบันทึก version/time ที่ CLI แสดงจริง

หาก release ใหม่เสีย ให้คืน source จากสำเนาหรือ revision ที่เคย verify แล้ว ทำ export ใหม่ด้วยชื่อเดิม และ deploy config เดิมไป target เดิม จากนั้นตรวจ public URL ซ้ำ การกู้คืน source ไม่ได้ย้อน deployment จนกว่าจะ deploy artifact ที่คืนแล้วสำเร็จ และ static rollback นี้ไม่เกี่ยวกับ D1 เพราะแพ็คเกจไม่มี database

Troubleshooting

อาการ	ตรวจ	วิธีแก้ที่ขอบเขตเล็กที่สุด
exporter หา source ไม่เจอ	อยู่ project root หรือไม่; path มีสามไฟล์หรือไม่	ใช้ <code>--source practice/psu-course</code> หรือ path จริง
font/JS/CSS 404	เปิด HTML/CSS ดู relative path และตรวจไฟล์ใน <code>public/</code>	export ใหม่; อย่าแก้ URL ให้ชี้กลับ local disk
lesson links เปิด 404 ใน Worker ใหม่	link ถูก rewrite เป็น course origin หรือยัง	ตรวจ exporter input/origin แล้ว export ใหม่
<code>whoami</code> แสดง account ผิด	OAuth/token environment และ account selection	หยุด deploy; กลับไป auth lab และเลือก account ใหม่
dry-run wu <code>main /D1</code>	ใช้ config ผิดหรือ exporter output ผิด	หยุด; ตรวจ exact <code>--config</code> และ export ใหม่
Worker name ขน	ชื่อใน config เป็นชื่อส่วนตัวหรือไม่	export ใหม่ด้วยชื่อเฉพาะ; อย่า deploy กับชื่อคนอื่น
deploy ผ่านแต่ไม่รู้ URL	อ่าน output/log ของ deploy เดิม	ใช้ URL ที่ CLI ค้น; อย่าเดาจากชื่อ
path ที่ไม่จริงไม่เป็น 404 ตามคาด	ตรวจ status จริง, <code>404.html</code> , <code>_headers</code> และ platform behavior	บันทึก finding แล้วแก้ package/sourceก่อน redeploy
หน้าเปิดได้แต่ search ไม่ทำงาน	console, <code>main.js</code> , selectors และ CSP/header	แก้ source, QA, export และ deploy ใหม่ตามลำดับ

Worksheet ส่งงาน

วันที่/เวลา:

Source path:

Source revision หรือ hash ที่ตรวจได้:

Export command:

Worker name:

Cloudflare account ที่ยืนยัน (redact ได้):

ไฟล์ใน export package:

ข้อยืนยันว่าไม่มี main/D1/API:

ผล local preview:
ผล dry-run และเวลาที่รัน:

คำนวณขนาด deploy ระบุ target ว่า:
Deploy command:
Public URL จาก Wrangler:
Version/deployment identifier จาก CLI (ถ้ามี):

Desktop Light/Dark:
Mobile 390px:
Search/filter/reset:
Lesson links ที่เปิดจริง:
Unknown-path status/behavior:
สิ่งที่ยัง Not tested:

ถ้าต้องกู้คืน จะใช้ source รุ่นใด:
ผู้ตรวจและเวลาตรวจซ้ำ:

Definition of Done

- source ต้นฉบับยังอยู่ครบและแก้ที่ source ก่อน export เสร็จ
- exported config ชื่อ `./public` และไม่มี Worker code, D1 หรือ API binding
- Worker name เป็นชื่อเฉพาะของผู้เรียนและ account ตรงกับ `whoami`
- dry-run กับ exact config ถูกบันทึกแยกจาก deploy จริง
- public URL มาจาก Wrangler output ไม่ได้คำนวณขึ้นเอง
- ตรวจ desktop/mobile, Light/Dark, search/filter/reset, lesson links และ unknown path uu public URL
- ไม่อ้างว่า static site เป็น LMS, backend หรือเว็บไซต์มหาวิทยาลัยอย่างเป็นทางการ
- screenshots มาจาก terminal/browser จริงและไม่เปิด credential
- worksheet แยก Verified, Finding และ Not tested พร้อม recovery source

เมื่อครบรายการนี้ ผู้เรียนมีทั้ง static deployment และหลักฐานว่ารู้ว่าจะ deploy อะไร ไป account/target ใด ตรวจอะไรแล้ว และจะคืนรุ่นเดิมอย่างไร โดยไม่ต้อง D1/API ของ course repository เข้ามาในงานที่ไม่ต้องใช้

ผลที่รับจริงและไฟล์หลักฐาน

[เปิดเว็บที่deployจริง](#) · [prompt/transcript/log/ภาพแต่ละรอบ](#) · [ดาวน์โหลดPDFรวมขึ้นสร้างเว็บและdeploy พร้อมsource/ZIPstatic](#)

รอบที่บันทึกใช้ Codex CLI 0.154.0 และ Wrangler 4.130.0 วันที่ 11 กันยายน 2026 ชื่อ Worker คือ `psu-course-website-demo` และversionที่CLI retrayงานคือ `21ef7990-e844-4e57-93ab-53224712b04c` ตรวจpublic

assets9 รายการ (รวมlicense 2 ไฟล์), 4 ขนาดหน้าจอ/รีม และไม่พบfailures การตรวจนี้ไม่ใช้การรับรองทุกbrowserหรือระบบมหาวิทยาลัย ภาพTerminalเป็นread-only replayจากเหตุการณ์จริง ส่วนภาพbrowserเปิดจากpublicURLที่CLIรายงาน

หลังรอบ 08 ผู้รวมงานเพิ่มguard ป้องกันชื่อ Worker หลักระยะที่ลงท้ายด้วยซีดจากข้อพบของreviewer แล้วรับregression checks 3 รายการและdry-runซ้ำ fastaticpackageของชื่อที่ถูกต้องไม่เปลี่ยน ส่วนCodex 09 ที่ล้มเหลวกับparent retryเก็บคนละlogอย่างชัดเจน