ENGSE203 สอบกลางภาค — คู่มือนักศึกษา (ภาคปฏิบัติ · Sec 1 / ชุด A)

อ่านคู่มือนี้ก่อนวันสอบ · ภาคปฏิบัติ 40 คะแนน · 180 นาที · ใช้ AI ได้แต่ต้องเข้าใจ

---

0. ภาพรวม 3 ส่วนของการสอบ

ส่วน	คะแนน	รูปแบบ	AI
A · ทฤษฎี	40	กระดาษ ในห้อง	❌
B · ปฏิบัติ (คู่มือนี้)	40	เขียนโค้ดหน้าเครื่อง	✅
C · Oral	20	สัมภาษณ์รายคนจากงาน B	—

> **หัวใจ:** ส่วน B ใช้ AI ได้ แต่ **ทุกคนถูกสัมภาษณ์ (C) จากโค้ดที่ส่ง** ถ้าอธิบายไม่ได้ คะแนน B ถูกทบทวน — **ใช้ AI เพื่อเข้าใจ ไม่ใช่เพื่อลอก**

---

1. เตรียมตัวก่อนวันสอบ

ทำให้เสร็จ**ล่วงหน้า** (วันสอบจะไม่มีเวลาแก้ปัญหาติดตั้ง):

- ☐ ลงชื่อเข้าใช้ **GitHub** บนเครื่องที่จะสอบ (หรือรู้ user/password ของตัวเอง)
- ☐ มี **Node.js ≥ 22.12** (`node -v` เช็คได้) และ **Git** (`git --version`)
- ☐ ตั้งค่า git ครั้งเดียว:

  ```
  git config --global user.name "ชื่อ-รหัสของคุณ"
  git config --global user.email "อีเมล GitHub ของคุณ"
  ```

- ☐ ทบทวนเนื้อหา: React component/props/state, **useState/useEffect**, React Router (useParams), **การโหลดข้อมูล**, **localStorage/persist**, `map/filter`, การ debug ด้วย **DevTools**

---

2. วันสอบ — รับข้อสอบอย่างไร

ผู้สอนจะให้ **ลิงก์ repo ข้อสอบของคุณเอง** (สร้างจาก template ของวิชา) ตอนเริ่มสอบ · ทำตามนี้:

```
# 1) clone repo ข้อสอบของคุณ (URL ที่ผู้สอนให้)
git clone https://github.com/<คุณ>/engse203-midterm-<รหัส> engse203-midterm
```

```
cd engse203-midterm

# 2) ติดตั้งและเปิดแอป
npm ci
npm run dev              # เปิด http://localhost:5173
```

> ถ้า `npm ci` ผิดพลาด ลอง `npm install` · ถ้ายังไม่ได้ **ยกมือเรียกผู้สอนทันที** อย่านั่งแก้เงียบ ๆ จนเสียเวลา

เปิดค้างไว้ 3 ไฟล์: `README.md` · `B1_BUGS.md` · `AI_USAGE.md`

---

3. วิธีทำงาน — เดินทีละ checkpoint (สำคัญที่สุด)

วงจรทำงานต่อ 1 checkpoint — ทำซ้ำจนครบทุกงาน

1. อ่านโจทย์ → 2. ทำนายผล (ก่อนแก้) → 3. แก้ทีละจุด → 4. npm run dev แล้วสังเกต → 5. ตรงกับตัวอย่าง output? → ✅ 6. git commit

ถ้าข้อ 5 ไม่ตรง กลับไปข้อ 2 (ทำนายใหม่ว่าเพราะอะไร) — อย่าเดาสุ่มหรือก๊อป AI ทั้งก้อน

แผนที่ checkpoint ทั้งภาคปฏิบัติ (commit ทุกกล่อง)

B1 · Debug	B2 · Search	B3 · Persist	B4 · Component	เสร็จ
12 คะแนน	10 คะแนน	10 คะแนน	8 คะแนน	npm run build ✅
บั๊ก 6 จุด + B1_BUGS.md	ช่องค้นหา + ตัวกรอง	ปุ่มทำเสร็จ + รอด refresh	PriorityBadge	push + แจ้ง oral
commit: fix-bugs	commit: b2-search	commit: b3-persist	commit: b4-badge	

workflow

กฎเหล็ก: ก่อนแก้ให้ **ทำนายผลก่อน** → แก้ทีละจุด → `npm run dev` แล้ว **สังเกตจริง** → ตรงกับ "ตัวอย่าง output" ในข้อสอบไหม → ผ่านแล้ว **commit ทันที**

ทำไมต้อง commit ทุก checkpoint - ประวัติ commit = หลักฐานว่าคุณทำเอง ทีละขั้น - คนที่ก๊อป AI มาทั้งก้อน commit ทีเดียวจบ = สัญญาณให้ถาม oral หนัก - ถ้าโค้ดพังทีหลัง ย้อนกลับ checkpoint ที่ผ่านแล้วได้

commit อย่างไร

```
git add -A
git commit -m "B2.2: filter by name/details"   # ข้อความสื่อว่าทำอะไร
git push
```

---

4. งานทั้ง 4 (รายละเอียดเต็มอยู่ในกระดาษข้อสอบ)

งาน	คะแนน	สรุป	commit ตัวอย่าง
B1 Debug	12	หา/แก้บั๊ก 6 จุด + กรอก `B1_BUGS.md`	`B1: fix bugs 1-6`
B2 Search	10	เพิ่มช่องค้นหา (4 checkpoint ย่อย)	`B2.1…B2.4`
B3 Persist	10	ปุ่ม "ทำเสร็จ" เปลี่ยนสถานะ **รอด refresh**	`B3.1…B3.3`
B4 Component	8	สร้าง `PriorityBadge` + ใช้ใน `RequestCard`	`B4.1…B4.3`

เทียบผลกับ "ตัวอย่าง output" ในข้อสอบทุก checkpoint เช่น แผงสรุปที่ถูกต้องคือ ทั้งหมด 5 · รอดำเนินการ 3 · กำลังดำเนินการ 1 · เสร็จสิ้น 1

> ⚠️ ห้ามแก้ไฟล์ในโฟลเดอร์ `services/` (เขียนว่า "ให้มาแล้ว — ห้ามแก้")

---

5. บันทึกการใช้ AI — `AI_USAGE.md`

กรอก**ระหว่างทำ** (ไม่ใช่ตอนท้าย) ทุกครั้งที่ใช้ AI ช่วย เขียนสั้น ๆ:

```
## B2 · ช่องค้นหา
- ถาม AI ว่า: "จะกรอง array ด้วยหลายเงื่อนไขใน React ยังไง"
- ใช้คำตอบส่วน: แนวการ chain .filter() สองชั้น
- ผมแก้เอง: เปลี่ยนให้ค้นทั้ง requesterName และ details + toLowerCase
- เข้าใจว่า: summary ต้องนับจากข้อมูลทั้งหมด ไม่ใช่ที่กรองแล้ว
```

คนที่ใช้ AI แล้วเข้าใจจะกรอกได้ · คนที่ลอกจะกรอกไม่ตรงกับโค้ด

---

6. การส่งงาน — ต้องครบทั้ง 3 อย่าง

การส่ง = **โค้ด (commit)** + **ผลรัน (build)** + **Pull Request + tag** ขาดอย่างใดถือว่าไม่ครบ

> **branch:** ทำงานบน `main` ได้เลย หรือถ้าผู้สอนกำหนดให้ใช้ branch ให้ตั้งชื่อขึ้นต้นด้วย `exam/` หรือ `midterm/` (เช่น `exam/midterm`) — ระบบ deploy รองรับทั้งสองแบบ

```
# 1) ก่อนหมดเวลา ต้อง build ผ่าน
npm run build           # ต้องขึ้น "✓ built" ไม่มี error
```

```
# 2) commit + push ครั้งสุดท้าย
git add -A
git commit -m "final: all tasks + build passes"
git push

# 3) เปิด Pull Request บน GitHub
#    ถ้าทำงานบน branch: จาก branch ของคุณ -> main (กด "Compare & pull request")
#    ถ้าทำงานบน main โดยตรง: เปิด PR จาก tag/commit ตามที่ผู้สอนกำหนด

# 4) ติด tag เวอร์ชันที่ส่ง
git tag midterm-submission-v1
git push origin midterm-submission-v1
```

หน้า submission (GitHub Pages ของ repo คุณ) จะมีปุ่ม 3 อัน — ตรวจว่าเปิดได้ก่อนแจ้งเสร็จ:

- **View Result** → แอปที่ deploy แล้ว (มาจาก build อัตโนมัติ)
- **Source** → โค้ดใน repo
- **Pull Request** → PR ที่คุณเปิด

เมื่อ push ระบบจะ build + deploy ให้อัตโนมัติ (ผ่าน GitHub Actions) รอสัก 1–2 นาที แล้ว "View Result" จะใช้ได้

---

7. เตรียมตัว Oral (20 คะแนน)

หลังส่ง B ผู้สอนจะเรียกสัมภาษณ์รายคน ~5 นาที **สุ่มถามจากโค้ดที่คุณเพิ่งเขียน**

จะถูกถามแนวนี้ (เตรียมตอบด้วยภาษาตัวเอง): - "โค้ดจุดนี้ทำอะไร ทำไมเขียนแบบนี้" - "ทำไม React ต้องการ key" · "ทำไม summary ต้องนับจากข้อมูลทั้งหมด" · "persist ยังไงให้รอด refresh" - **micro-task สด** เช่น "ทำให้ค้นหาจากรหัส (id) ได้ด้วย" — ต้องแก้โค้ดตัวเองได้ใน 1 นาที

ถ้าอธิบายจุดที่สุ่มไม่ได้เลย คะแนน B ของจุดนั้นถูกทบทวน — นี่คือเหตุผลว่าทำไมต้องเข้าใจ ไม่ใช่แค่ลอก

---

8. Dos & Don'ts

✅ ทำ	❌ อย่าทำ
commit ทุก checkpoint	commit ทีเดียวตอนจบ
เทียบผลกับตัวอย่าง output	เดาว่าถูกโดยไม่รัน
ใช้ AI แล้วอ่าน/แก้ให้เข้าใจ	ก๊อป AI ทั้งก้อนโดยไม่อ่าน
กรอก AI_USAGE ระหว่างทำ	เว้นว่าง/กรอกมั่ว

✅ ทำ	❌ อย่าทำ
`npm run build` ก่อนส่ง	ส่งทั้งที่ build ไม่ผ่าน
ยกมือถามผู้สอนเมื่อติดตั้งพัง	นั่งแก้เงียบ ๆ จนหมดเวลา
—	คุยกับเพื่อน / ส่งไฟล์หากัน / แก้ `services/`

9. แก้ปัญหาที่พบบ่อย

อาการ	ทำอย่างไร
`npm ci` error	ลอง `npm install` · ถ้าไม่ได้เรียกผู้สอน
แก้บั๊กแล้วหน้าไม่เปลี่ยน	บันทึกไฟล์ยัง? · hard refresh (Ctrl/Cmd+Shift+R)
localStorage ค้างข้อมูลเก่า	กดปุ่ม "Reset Demo Data" ในแอป หรือ DevTools → Application → Local Storage → ลบคีย์
`npm run build` error	อ่านบรรทัด error — มักเป็น syntax/import ที่ผิด · แก้แล้วรันใหม่
push ไม่ได้	เช็คว่า login GitHub แล้ว · `git remote -v` ถูก repo ตัวเองไหม

10. เช็คลิสต์ก่อนแจ้งเสร็จ

- ☐ B1 แก้ครบ 6 จุด + `B1_BUGS.md` กรอกครบ
- ☐ B2 ค้นหาทำงานครบ 4 checkpoint
- ☐ B3 ปุ่มทำเสร็จ persist + รอด refresh (F5 แล้วยังอยู่)
- ☐ B4 `PriorityBadge` + ใช้ใน RequestCard
- ☐ `npm run build` **ผ่าน**
- ☐ `AI_USAGE.md` กรอกครบ
- ☐ push + เปิด Pull Request + tag `midterm-submission-v1`
- ☐ เปิดหน้า submission เช็ค View Result / Source / PR ได้
- ☐ **แจ้งผู้สอนว่าพร้อม oral**

ขอให้โชคดีครับ — เข้าใจจริงคือทางที่คุ้มที่สุด