

App Design Packet

Topic 8: scope one user task, map web responsibilities, label three interface states, write a bounded prompt contract, record critique evidence, and prepare three Topic 9 tests.

Name	Date	Due
	mm/dd/yyyy	Before Topic 9; use the LMS calendar.

PREFERRED SUBMISSION One four-page PDF LastName-Topic08-AppDesign.pdf. LMS text, DOCX/export, scanned paper, or an instructor-approved accessible alternative is also accepted.	EQUAL CREDIT No code, video, or AI required Use the course seed, summary, source map, and paper/text wireframe. An approved live design probe earns no extra credit.
---	---

Local-save route: this page stores responses only in this browser; nothing is sent, and your name is not saved. On a shared device, use no private data, save only to personal storage, select **Clear local draft**, remove downloads/print artifacts, and close the browser. Use paper or LMS text if cleanup is uncertain.

Never submit: credentials, account/device screenshots, access proof, API keys, full chats, other people's names or identifiers, real student/peer/instructor records, grades, disability/conduct/advising facts, unpublished or protected material, or unnecessary identity evidence. Your required submission name and the supplied fictional course examples are permitted.

1 · Route, scope, and user task

Evidence route

Course materials + paper or typed wireframe

One-sentence scope: one user, one task, one screen, one primary action, one result

One feature removed—and why	Authorized input and what stays out

Workload guide: after drafting fields during both meetings, plan 35–50 focused minutes to finish. Accepted alternatives use the same concise evidence fields, not a longer paper; ask the instructor through the LMS before the deadline if a different format needs approval.

Map what the page means and does.

Layer	Responsibility in your concept	One visible example
HTML structure + meaning		
CSS presentation + layout		
JavaScript behavior + state		
Browser render + execute		

Three states and one recovery path

Initial state: what is visible before action?

Missing-input state: exact message, location, keyboard focus (active typing position), and preserved work

Result state: what appears after success?

Recovery: how can the user revise or start over?

Responsibility check: HTML gives controls meaning; CSS helps present hierarchy and state; JavaScript can respond to events and update state; the browser renders and executes. These responsibilities can interact—do not treat the map as four sealed boxes.

Make all three states inspectable.

Sketch inside the boxes after printing, or type a region-by-region outline. Visual polish is not assessed. Labels and state transitions are.

1 • Initial	2 • Missing input	3 • Result
Title, purpose, labels, inputs, primary action, result region...	Message, location, focus/attention, preserved values, retry...	Visible output, input summary, revise/start-over action...

AI-fit and human-judgment decision

Use a transparent rule, a model, or no automation? State the user benefit, added risk, and evidence needed.

Exact transformation: given these inputs, what rule produces what output? Include one concrete example.

Example: task + 25 minutes → 2 minutes prepare, 20 minutes work, 3 minutes reflect.

What must a person verify, revise, or approve before the result is used?

Topic 8 boundary: one-page client-only design specification. No login, backend, model API, private records, external action, or public deployment. An optional disposable layout/state probe does not replace Topic 9 implementation. A model feature needs a stated benefit plus a new verification plan.

Turn the design into testable instructions.

Nine-part first-build contract (maximum 650 characters): user/task, input, action, rule, result, error, accessibility, boundary, success

Build one self-contained client-only page for... Concise clauses are acceptable.

One exact peer, solo self-critique, or supplied-example finding

The field or sentence revised—and why

Three manual checks for Topic 9

1 · Normal input + exact expected output

2 · Missing input + expected state

3 · Revise/repeat + expected replacement

No-video course summary: specify the first version before building → inspect actual behavior against the contract → revise the smallest difference and retest. Use these same three moves for the transfer note.

Build-flow transfer note: what belongs before the first build, and what can only be learned after inspecting behavior?

Route and assistance disclosure: source/tool, affected decision, accepted/rejected/revised suggestion, checks, limitation, final human decision

Submission check

- ☐ I included one user/task scope, one removed feature, the four-part web map, and three interface states.
- ☐ My wireframe or text outline shows initial, missing-input, result, and recovery behavior.
- ☐ My prompt contract includes all nine parts and my three Topic 9 tests have exact, observable expected results.
- ☐ I submitted no private record, credential, account/device screen, access proof, API key, or full chat.

Evidence anchors: complete evidence addresses every named element with an observable expected result; developing evidence leaves elements missing or unclear; incomplete evidence is largely unsupported or not testable. The instructor evaluates scope, web responsibilities, states/wireframe, rule/contract, critique/revision, tests, and transfer—not drawing quality, prose polish, coding background, product access, live-tool use, or video access. Part 1 supplies Topic 9's rule and test oracle (the expected result, written before the run, that a test checks against). Official points or course weight come from the LMS; this packet sets no automatic pass threshold.