Assignment: Coffee Shop GUI (Decorator Pattern)

1. Purpose

Design a desktop web GUI that lets a user build a custom coffee order by composing a base beverage with optional condiments. The interface should make the Decorator idea visible (i.e., "wrapping" features) through clear interactions, labels, and real-time pricing.

2. Learning Outcomes

By completing this, you will:

- Translate object-oriented composition (Decorator) into intuitive UI flows.
- Design stateful interfaces (size, base, add-ons) with real-time feedback.
- Apply accessibility, usability, and error-handling principles.
- Plan testing and validation for a transactional UI.

3. Scenario

A coffee shop offers bases (Espresso, House Blend, Dark Roast, Decaf) and condiments (Milk, Mocha, Soy, Whip, Caramel, Vanilla). Sizes: Small, Medium, Large. Price formula:

$$\text{Total Price} = \text{Base} \times \text{Size Multiplier} + \sum(\text{Condiment} \times \text{Size Multiplier})$$

Multipliers: Small = 1.00, Medium = 1.20, Large = 1.40. Reasonable base and condiment prices may be chosen, but they should be visible in the UI.

4. Functional Requirements

The GUI must:

- Allow selection of one base beverage.
- Allow selection of exactly one size.
- Enable adding/removing condiments in any combination.
- Display a live order summary updating on every change:

  - Full description (e.g., "Espresso (large), milk, mocha")
  - Line-item subtotal math (base, condiments, size multiplier)
  - Final price in currency format
- Provide order controls:
  - Clear/Reset
  - Add to Cart and Checkout
- Include validation and errors:
  - Warn if no base is selected
  - Prevent checkout if the cart is empty
- Offer persistence (e.g., remember last selection during session)
- Contain a Help/Info panel explaining how "add-ons wrap the drink" (Decorator concept)

5. Non-Functional Requirements

- **Responsive:** Works well on laptop and tablet (desktop design 1280×800).
- **Accessible:**
  - Keyboard navigable, visible focus states.
  - Labels for all inputs, fieldset/legend for grouped controls.
- **Performance:** Price updates should feel instantaneous (<100ms).
- **Consistency:** Use a clear design system (spacing, typography, hierarchy).

6. Required UI Deliverables

A. Wireframes (Low-Fidelity)

Provide at least three screens:

1. Initial builder (no base selected)
2. Configured drink (base+size+condiments, showing total)
3. Error/empty state or checkout modal

Annotate how condiments toggle, where price/description change, and how size impacts cost.

B. Hi-Fi Mockups

Include color, type, spacing, component states (default/hover/disabled), and a price breakdown module.

C. Interaction Flow

Show a flow diagram: Choose base → Choose size → Add condiments → Review total → Checkout. Note all state transitions.

D. Microcopy & Labels

Provide button text, helper text, validation messages, and tooltips explaining the Decorator concept.

7. Information Architecture & Components

- **Left panel:** Base & Size selectors.
- **Middle panel:** Condiments (checkboxes or tiles).
- **Right panel:** "Your Drink" card showing name, items, multiplier, taxes, total, and buttons.

Alternative layouts are allowed with justification.

8. Pricing & Math Presentation

The UI must show how the total is calculated, for example:

Dark Roast	$\$1.50 \times 1.40 = \$2.10$
Milk	$\$0.20 \times 1.40 = \$0.28$
Mocha	$\$0.35 \times 1.40 = \$0.49$
Total	\$2.87

Provide a toggle to expand/collapse the breakdown and visually highlight updates when the size changes.

9. Accessibility Checklist

- Input groups have labels and legends.
- All controls reachable via Tab order; Escape closes modals.
- Live price updates are announced or visually animated.
- Error messages tied to the related field.
- Minimum target size 44×44 px; adequate spacing.
- Color contrast checked and ratio documented.

10. Usability Plan

Design three scenario-based tasks:

1. Build a Large House Blend with Soy and Mocha; verify total within 30s.
2. Remove Mocha, switch to Medium, confirm new total.
3. Build a Decaf with Whip and proceed to checkout.

Record observations on time, errors, and hesitation points.

11. Edge Cases to Cover

- Changing size after adding condiments.
- Clearing all selections.
- Attempting checkout without a base.
- (Optional) Handling duplicate condiments.

12. What to Submit

- A PDF report (6–10 pages) containing wireframes, mockups, diagrams, code and analysis.
- Design assets (PNGs/SVGs of screens).
- (Optional) Clickable prototype link (Figma, Adobe XD).

13. Rubric / Evaluation Criteria

Criterion	Points	Description
Functional Implementation	25	Meets all base and condiment requirements; price updates correctly and instantly.
User Interface Design	20	Clear, consistent layout and visual hierarchy; communicates the Decorator concept.
Accessibility and Responsiveness	15	Follows WCAG basics; keyboard-friendly; adapts to various screen sizes.
Usability Testing Plan	10	Demonstrates thoughtful test cases and user-centered evaluation.
Documentation & Deliverables	10	Well-organized report, labeled figures, complete submission package.
Creativity & Visual Appeal	10	Engaging, polished interface with thoughtful details.
Code Quality / Prototype Logic	10	Clean structure, readable naming, and functional prototype behavior.
Total	**100**	