React Frontend Developer Test

Duration: 1 Hour

Focus: Pixel-Perfect Chrome-like Top Bar UI in React

Objective

Your task is to build a **mock Chrome browser window** using React. This is a frontend UI challenge focused on layout accuracy, component structure, and attention to visual detail. No backend or functional browsing logic is required.

What You Will Build

You need to create a **Chrome-like browser window** with the following:

1. Window Container

- Create a floating window (centered or fixed on screen) to represent a Chrome browser.
- Add three standard window controls on the top-right corner:
  - Minimize (–)
  - Maximize (□)
  - Close (×)

2. Chrome Top Bar

Inside the window, the top bar must include:

a. Tab Section

- At least **two tabs**.
- Clicking a tab should **visually switch** the active tab.
- Tab shape, spacing, and size should resemble Google Chrome.

b. Address Bar Area

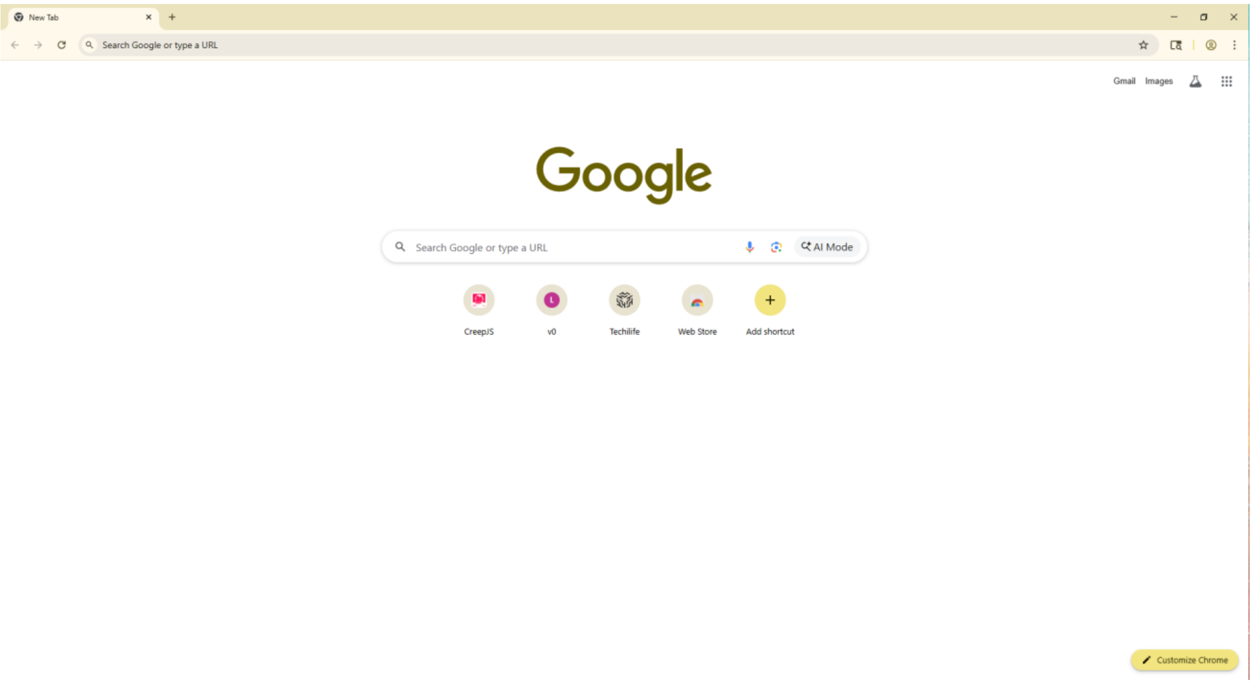
- A rounded address bar placed in the center of the top bar.
- Placeholder text: https://example.com

- Do not implement navigation functionality.

c. Icons

- Include these icons inside the top bar (positioned realistically):
    - Back (←)
    - Forward (→)
    - Refresh (⟳)
    - Lock Icon (🔒)
    - Star Icon (☆)
    - Vertical Menu (⋮)

You may use icons from libraries such as **Material Icons**, **Lucide**, or **Font Awesome**.



Technical Requirements

- Use **React with functional components** and **Hooks**.
- Use **plain CSS** or CSS Modules (do not use UI libraries like MUI or Tailwind).
- The UI must **closely match** the look and feel of a real Chrome browser (spacing, padding, alignment, borders, shadows, fonts).
- No backend or complex routing is required.
- The tab content area can be left empty or display a placeholder (e.g., "App goes here").

Bonus (Optional)

- Add a small transition or animation when switching tabs.
- Make the window responsive on different screen sizes.

Submission Instructions

1. Upload your code to **GitHub** or share a downloadable **ZIP file**.
2. Include a short **README** with:
    - Tools used
    - How to run the project locally
3. If possible, share a **live demo link** (optional).

Evaluation Criteria

Criterion	**Weight**
Visual Accuracy (Pixel-perfect UI)	High
Clean Component Structure	Medium
CSS Quality and Layout Handling	Medium
Tab Switching Logic	Low
Code Readability and Comments	Bonus

Estimated Time to Complete

60 minutes maximum.

You are encouraged to inspect the real Chrome browser interface for design reference during the task.