

CSS Portfolio Assignment – Show All These Features in One File

Create a **single HTML file** with internal `<style>` (no separate .css file) that represents your personal portfolio page.

Must include these sections (at minimum):

- Navigation bar (sticky)
- Hero section with heading + button(s)
- Projects section with **at least 4 project cards** (each with image + title + short description)
- About section with some text paragraphs
- Optional: Contact or Footer

Mandatory CSS Requirements – You must use ALL of the following concepts visibly and meaningfully:

★ **Every kind of selector**

- Element selector (body, h1, p, etc.)
- Class selector (.card, .btn, etc.)
- ID selector (#hero, #projects, etc.)
- All combinators:
 - Descendant (space)
 - Child (>)

★ **Pseudo-classes** (must be used and noticeable)

- :hover
- :focus
- :active
- :nth-child()

★ **Pseudo-elements**

- ::before
- ::after
- ::first-letter

★ **Show selector priority / specificity**

- Write rules with different specificity levels and make it visible which one wins (you can use colors or borders to prove it)
- Bonus: use !important once (only for teaching purpose) and comment why

★ **Box model in action**

- Create **cards** and **buttons** using:
 - padding, margin, border, border-radius
 - outline (especially visible on :focus or :hover)

★ **Overflow** – show **all four values** clearly

- Create 4 small boxes with long text inside and apply:
 - overflow: visible
 - overflow: hidden
 - overflow: scroll
 - overflow: auto

★ **Google Fonts**

- Import **at least two Google Fonts** (one for headings, one for body)
- Use them properly

★ **Typography controls**

- letter-spacing
- word-spacing
- line-height
- text-align
- text-decoration (different kinds)
- text-transform
- text-shadow

★ **Display types** – use all three meaningfully

- display: block
- display: inline
- display: inline-block

★ **Positioning** – use all four types

- position: static (default – mention it)
- position: relative (small offset or layering)
- position: fixed (e.g. back-to-top button)
- position: sticky (e.g. navigation bar)

★ **Flexbox** – must be used in two places

- One place: **4 project images/cards in a row** (use flex, justify-content, gap)
- Second place: **Navigation bar** (use flex, align-items, justify-content)
- Change flex-direction from row to column on mobile screens using media query

★ **Media query**

- At least one @media rule (max-width: 768px or similar)
- Change layout (especially flex-direction of nav and/or gallery)

Extra good-to-have (but not mandatory):

- Hover lift / scale effect on cards & buttons
- Smooth transition
- Nice color scheme
- Responsive images (object-fit, etc.)
- Comments in CSS explaining which concept is being shown where

Submission rules:

- One single .html file (everything inside <style>)
- Use placeholder images (via <https://via.placeholder.com> or unsplash links)
- Make it look clean and modern — don't just make ugly boxes

Good luck!

Show that you really understood CSS selectors, box model, positioning, flexbox, and responsive design.