

WEB DESIGN

FINAL EXAM

Course	Web Design
Exam Type	FINAL EXAM
Duration	2 hours
Generation	GEN-11
Date	Dec 30 th 2025
Exam Rules	<ul style="list-style-type: none"> • All documents allowed • The following actions are forbidden and will conduct to a 0 score: <ul style="list-style-type: none"> ○ Talking or exchanging information with another student ○ Using Internet or any Network

Evaluation

PART A – CONCEPTUAL UNDERSTANDING		
Exercise 1	MCQ	10 points
Exercise 2	Open Questions	20 points
PART B – CODING PROBLEMS		
PROBLEM 1	Layouting	20 points
PROBLEM 2	Responsiveness	20 points
PROBLEM 3	JS	30 points

To submit

- Submit this **document** AND your **code** (ZIP) on MS TEAM

PART A – CONCEPTUAL UNDERSTANDING

EXERCISE 1 (10 pts) MULTIPLE CHOICE

Q1 - What's wrong with the following code?

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
</ul>
<ol>
  <li>Step 1</li>
  <li>Step 2</li>
</ol>
```

- (A) `` and `` tags should not be mixed on the same page.
- (B) There is nothing wrong with this code.**
- (C) The list items inside `` should start at `<li value="1">`.
- (D) Both lists should be `` to keep the same style.

Q2 - Which unit is relative to the font-size of the root element (`<html>`)?

- (A) %
- (B) rem**
- (C) em
- (D) vh

Q3 - You want to align all items inside a container to the center both vertically and horizontally. Which option describes the correct approach?

- (A) `display: flex; justify-content: flex-start; align-items: flex-start;`
- (B) `display: flex; justify-content: center; align-items: center;`**
- (C) `display: grid; justify-items: center; align-items: center;`
- (D) `display: flex; justify-content: space-between; align-items: stretch;`

Q4 - How many columns and rows will the following grid have?

```
.grid-container {  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    grid-template-rows: auto 100px;  
    gap: 20px;  
}
```

(A) 3 columns and 2 rows

(B) 3 rows and 2 columns

(C) 2 columns and 3 rows

(D) 2 columns and 3 rows with gaps

Q5 - What background color will the .card element have when the screen width is **900px**?

```
.card {  
    background-color: white;  
}  
  
@media (max-width: 800px) {  
    .card {  
        background-color: lightblue;  
    }  
}
```

(A) White

(B) Lightblue

(C) Gray

(D) Transparent

Q6 - You want the `.container` element to change its background color to `lightblue` when the screen width is 768px or smaller.

Which of the following CSS code snippets correctly achieves this behavior?

(A)

```
.container {
    background-color: white;
}

@media (min-width: 768px) {
    .container {
        background-color: lightblue;
    }
}
```

(C)

```
.container {
    background-color: lightblue;
}

@media screen 768px {
    .container {
        background-color: white;
    }
}
```

(B)

```
@media (max-width: 768px) {
    .container {
        background-color: lightblue;
    }
}
```

(D)

```
@media (width <=768px) {
    .container {
        color: lightblue;
    }
}
```

Q7 - Which combination of design choices best supports a **CALM** and **MINIMALIST** website personality?

(A) Bright contrasting colors, heavy shadows, many icons, and tight spacing

(B) Neutral color palette, generous whitespace, simple typography, and soft shadows

(C) Multiple font styles, playful illustrations, bold colors, and animated icons

(D) Strong gradients, sharp border-radius, dense content, and high visual complexity

Q8 - Which Tailwind CSS class combination correctly creates a centered button with padding, blue background, and white text?

(A)

```
<button class="center padding blue text-white">  
    Click me  
</button>
```

(B)

```
<button class="flex justify-center items-center p-4 bg-blue-500 text-white">  
    Click me  
</button>
```

(C)

```
<button class="align-center pad-4 background-blue color-white">  
    Click me  
</button>
```

(D)

```
<button class="btn btn-primary">  
    Click me  
</button>
```

Q9 - What will be the output of the following JavaScript code?

```
let x = 5;  
if (x > 3) {  
    x = x + 2;  
} else {  
    x = x - 2;  
}  
console.log(x);
```

(A) 3

(B) 5

(C) 7

(D) 10

Q10 - Which of the following JavaScript statements correctly selects an element with the id "myButton" and changes its text to "Clicked"?

- (A) `document.getElement("myButton").text = "Clicked";`
- (B) `document.getElementById("myButton").innerText = "Clicked";`
- (C) `document.querySelector("#myButton").textContent = "Click";`
- (D) `document.getElementsByName("myButton").value = "Clicked";`

EXERCISE 2 (20 pts) OPEN QUESTIONS

Q1 - You need to apply a style to **five elements** on a page. Would you use a **class**, an **ID**, or a **type selector**? Explain your choice and discuss the potential advantages and risks of each method.

I would use class selector because it is more flexible and reuseable.

Class selector change can multiple element at the same time while ID can only change 1 element at a time.

ID is for unique and not a good idea for repeat styling

Type Selector : style cause to all mentioned element and something the element that we dont want to style still style it (cause unwanted styling tag).

Q2 - You are tasked with designing a button with an **exact width of 100px**, including padding and borders. Which **box-sizing** mode should you use, and why?

I use border-box . This makes it easy to create a button with an exact total width of 100px without needing to manually subtract padding or border sizes.

Q3 – You are creating a photo gallery. Would you use Flexbox or Grid for the layout? Explain your decision, considering **alignment**, and **responsiveness**.

I would use grid for creating photo gallery because grid is more easier to use and consider how many rows or columns we want it to set to , while flex we cannot do that.

So , grid is more useable and fit for this scenario.

Q4 – You need to add space between a button and the text above it. Should you use margin or padding on the button? Justify your choice.

I would use margin to add space between a button and text above it because margin is more fit it gives a gap(outside space) between element , while padding is giving space in the element it self.



Q4 – Here is a website folder structure.

```
website/
├── index.html
├── about.html
└── assets/
    ├── css/
    │   ├── style.css
    │   └── theme.css
    ├── js/
    │   ├── script.js
    │   └── app.js
    └── images/
        ├── logo.jpg
        └── banner.jpg
    └── videos/
        └── intro.mp4
└── docs/
    └── manuals/
        └── manual.pdf
```

Fill up the table with the **hyperlink**, so that we **can create an anchor link** from the **source folder** to the **destination file**.

PATH TYPE	SOURCE FOLDER	DESTINATION FILE	HYPERLINK LINK
ABSOLUTE		script.js	href= "website/assets/js/script.js"
RELATIVE	website/	style.css	href= "./assets/css/style.css"
RELATIVE	website/docs/	manual.pdf	href= "./manuals/manual.pdf"
RELATIVE	website/assets/videos/	intro.mp4	href= "./videos/intro.mp4"
ABSOLUTE		intro.mp4	href= "website/assets/videos/intro.mp4"

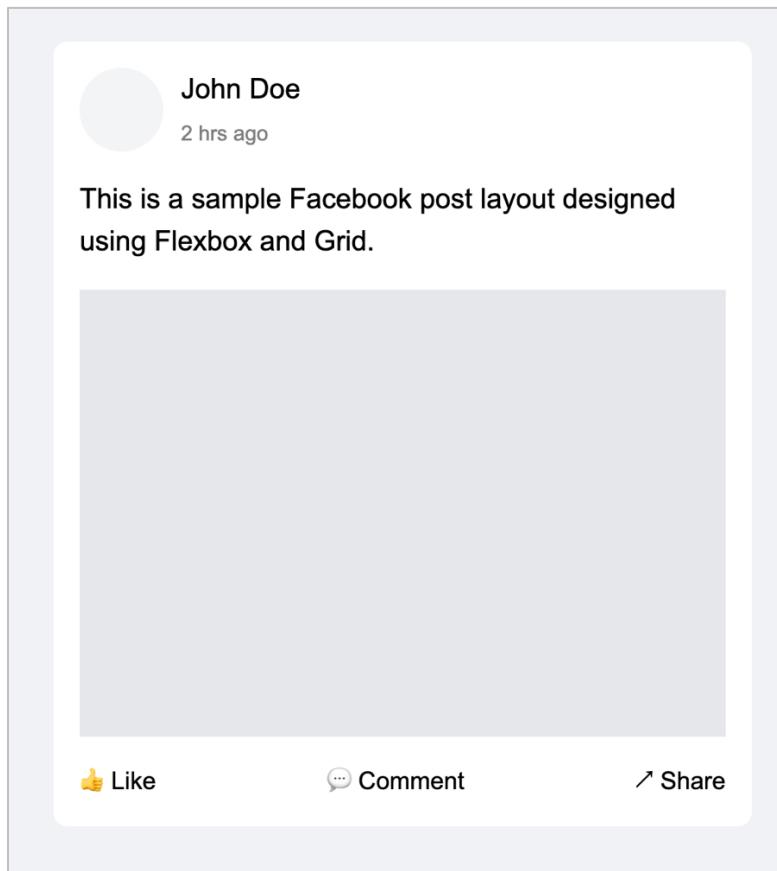
PART B – CODING PROBLEMS

PROBLEM-1 – LAYOUTS – 20 POINTS

Design a simple Facebook-style post layout using HTML and CSS (Flexbox and/or Grid).

NOTES:

- ✓ You can use **regular CSS** or **Tailwind CSS**
- ✓ Use a fixed post container size of **400px** width (height auto or around **450–500px**)
- ✓ Use Emojis as button icons.

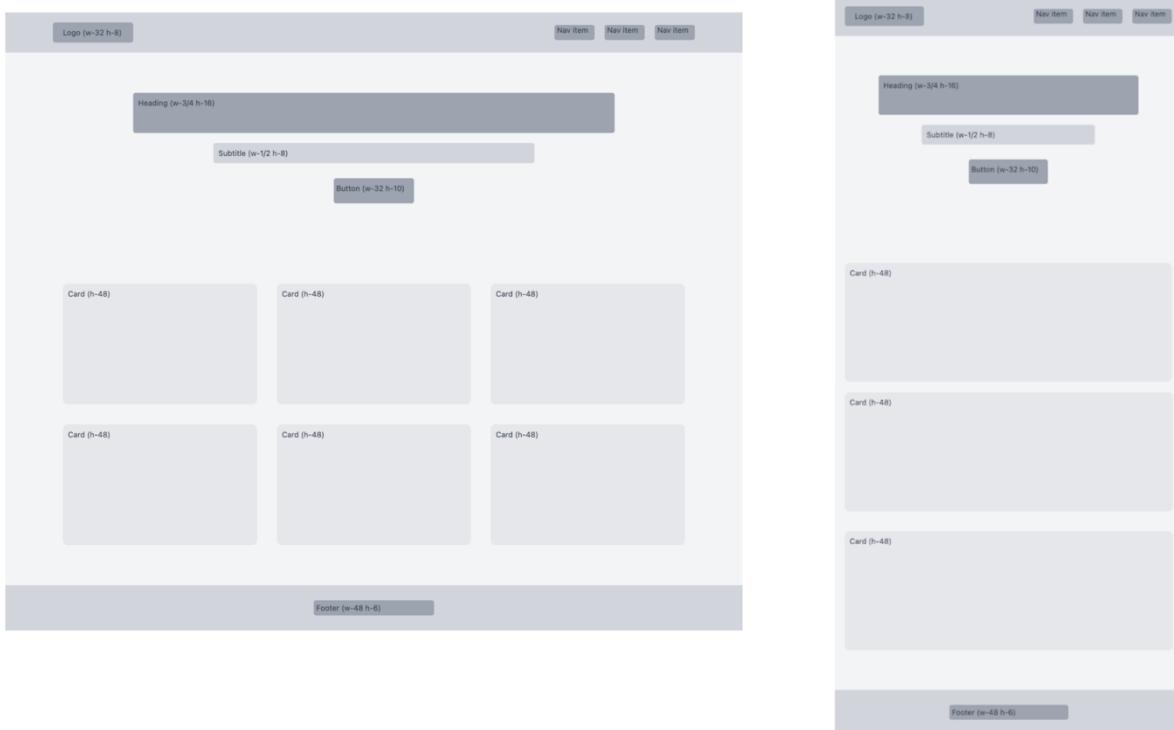


PROBLEM-2 – RESPONSIVENESS – 20 POINTS

Create a single HTML file that implements a responsive web page layout matching the provided wireframe.

NOTES:

- ✓ You can use regular CSS or Tailwind CSS



PROBLEM-3 – JS – 30 POINTS

- ✓ Create a simple web page using HTML, CSS with **3 tab buttons**, and **a paragraph description**.
- ✓ When **user clicks on a tab button**, use JS to **update the paragraph description** accordingly

NOTES:

- ✓ You can use regular CSS or Tailwind CSS
- ✓ Use Emojis as button icons.

