Lab 2: In-Class Task

HTML5 & CSS3 — Flexbox, Grid, Forms, Responsive Design

Duration: 30 minutes

Task: Create a Spotify Login & Pricing Page

You must recreate a simplified version of a **Spotify-style** web page that includes a login section and a pricing plans section. The page must use **Flexbox and/or CSS Grid**, **semantic HTML5 tags**, and be **responsive** at one breakpoint.

Page Requirements

1. Document Setup

- Create `index.html` and `styles.css` files
- Use proper HTML5 structure with `<!DOCTYPE html>`
- Link the external stylesheet
- Set the page title to "Spotify - Login & Plans"
- Add a CSS reset at the top of your stylesheet:

```
* { box-sizing: border-box; margin: 0; padding: 0; }
```

---

2. Header & Navigation

Use a `<header>` element containing a `<nav>` with:

- A logo text "Spotify" in a `<span>` with `class="logo"` (use green color `🟩 #1DB954`)
- A `<ul>` navigation with links: Premium, Support, Download
- On the right side: "Sign Up" and "Log In" links with `class="nav-auth"`

Styling requirements:

- Use **Flexbox** to align logo on the left and navigation + auth links on the right
- `justify-content: space-between` for the main layout
- Black or dark background (`⬛ #191414`)
- White text, green hover effect on links

- Sticky or fixed position at the top

---

3. Login Section

Use a `<main>` element. Inside, create a `<section>` with `id="login"` containing a `<form>`:

- Heading: "Log in to Spotify"
- A horizontal divider with text "OR" in the middle (line — OR — line pattern)
- Email/username input:
  - `<label>` with text "Email or username"
  - `<input>` with `type="email"`, `placeholder`, `id`, and `name` attributes
- Password input:
  - `<label>` with text "Password"
  - `<input>` with `type="password"`, `placeholder`, `id`, and `name` attributes
- A "Remember me" checkbox:
  - `<input type="checkbox">` with a corresponding `<label>`
- A submit button: "Log In" with `class="btn-login"`
- A link below: "Forgot your password?"
- A separator line
- Text: "Don't have an account?" with a link "Sign up for Spotify"

Styling requirements:

- Center the form on the page (max-width: 450px, margin auto)
- Dark background for the page, white/light form card with padding
- Labels above inputs, inputs at full width
- Inputs with border, padding, border-radius, focus state (highlight border color on `:focus`)
- Green login button (🟩 `#1DB954`) with white text, full width, rounded corners, hover darkens
- Checkbox and label on the same line using **Flexbox**
- Links in green color

---

4. Pricing Plans Section

Create a `<section>` with `id="pricing"` containing:

- A heading: "Choose Your Plan"
- A subtitle paragraph: "Listen without limits. Try 1 month free."

- Three plan cards using **CSS Grid** (`grid-template-columns`) or **Flexbox**, each in a `<div>` with `class="plan-card"`:

Free Plan

- Plan name in `<h3>`
- Price: "$0/month"
- A `<ul>` list of features:
  - Ad-supported listening
  - Basic audio quality
  - 1 account
  - No offline mode
- A button: "Get Started" with `class="btn-outline"`

Premium Individual

- Add additional `class="featured"` to this card
- Add a `<span class="badge">` with text "MOST POPULAR"
- Price: "$9.99/month"
- Features:
  - Ad-free music
  - High audio quality
  - 1 account
  - Offline mode
  - On-demand playback
- A button: "Try Free for 1 Month" with `class="btn-primary"`

Premium Duo

- Price: "$14.99/month"
- Features:
  - All Individual features
  - 2 accounts
  - Duo Mix: shared playlist
  - Ad-free podcasts
- A button: "Try Free for 1 Month" with `class="btn-outline"`

Styling requirements:

- Use **CSS Grid** with 3 equal columns (or Flexbox with equal-width items)
- Cards with border, padding, border-radius
- The `.featured` card should stand out:
  - Green border or green top accent bar
  - Slight shadow or different background
  - Badge positioned at the top of the card
- Price in large bold font
- Feature list without bullets, with spacing between items
- Use checkmarks (✓) before each feature via CSS `::before` or inline text
- `.btn-primary`: green background, white text, full width
- `.btn-outline`: transparent background, green border and text, full width
- Hover effects on buttons

---

5. Footer

Use a `<footer>` element containing:

- Three columns using **Flexbox** or **Grid**:
  - **Company**: Links — About, Jobs, Press
  - **Links**: Links — Help, Privacy, Terms
  - **Social**: Links — Instagram, Twitter, Facebook
- Copyright text below: "© 2025 Spotify AB"

Styling requirements:

- Dark background (● `#191414`), light gray text
- Columns displayed in a row
- Links stacked vertically within each column
- Column headings in white, bold
- Links in gray, turning white on hover
- Copyright centered, smaller font, with a top border or separator

---

6. Responsive Design

Add a media query for screens **768px and below**:

- Navigation links should be hidden (simplified for mobile)

- Login form takes full width with side padding
- Pricing cards stack vertically (single column)
- Footer columns stack vertically
- Adjust font sizes and padding as needed

```css
@media (max-width: 768px) {
  /* Your responsive styles here */
}
```

---

7. Global Styling Requirements

- Use a consistent font: `'Helvetica Neue', Arial, sans-serif`
- Smooth scroll: `html { scroll-behavior: smooth; }`
- All interactive elements (buttons, links, inputs) must have `transition` for smooth effects
- All `<input>` elements must have associated `<label>` elements (accessibility)
- Use CSS custom properties (variables) for colors:

```css
 :root {
   --spotify-green: #1DB954;
   --spotify-black: #191414;
   --spotify-dark: #121212;
   --text-white: #FFFFFF;
   --text-gray: #B3B3B3;
 }
```

- Use semantic tags: `<header>`, `<nav>`, `<main>`, `<section>`, `<footer>`, `<form>`

Files to Submit

```
inclass/
  index.html
  styles.css
```

Concepts Being Tested

- Semantic HTML5 elements (`header`, `nav`, `main`, `section`, `footer`, `form`)
- Form elements with proper `<label>` associations
- Flexbox layout (navigation, checkbox row, footer)
- CSS Grid layout (pricing cards)
- CSS custom properties (variables)
- Pseudo-elements (`::before` for checkmarks)
- Responsive design with `@media` queries
- Interactive states (`:hover`, `:focus`)
- CSS transitions
- Button variants (filled vs outlined)
- Badge/label positioning
- Centering techniques (Flexbox centering, margin auto)

Good luck!