

Build the Newsletter Sign-up Page

A step-by-step guide using only HTML and CSS Flexbox. It tells you *what* to do and *where to look*, but you write the code.

Before you start

- Open the **design/** folder and look at **desktop-design.jpg** and **desktop-success.jpg**.
- Open **style-guide.md** for the colors and the font.
- Look inside **assets/** to see which images and fonts you have.
- Work in two files: **index.html** (structure) and **style.css** (looks). Later you add **success.html**.
- Save often and refresh the browser after every small change.

GOLDEN RULE: Build one small thing, check the browser, then move on. Never write a whole page before testing.

PART 1

Plan the page

Goal: Know what you are building before typing.

- Look at the desktop design. Draw boxes on paper: what is the biggest box? What is inside it?
- How many columns does the card have?
- Write down the list of elements: heading, paragraph, list, label, input, button, images.
- Write down the colors and the font from the style guide.

THINK: Ask yourself: which elements sit next to each other, and which sit on top of each other? That decides where you will use **flex-direction** later.

PART 2

Write the HTML structure

Goal: A plain, unstyled page with all the content.

- Link your **style.css** file inside the **<head>** (use a **<link>** tag).
- Create one wrapper element for the whole white card. Which semantic tag fits? (try **main** or **section**)
- Inside it, make two children: one for the text side, one for the image side.
- In the text side add: a **h1**, a **p**, a **ul** with three **li**, then a **form**.
- In the form add a **label**, an **input** and a **button**.
- Add the check icon **icon-list.svg** in each list item with an **img** tag.
- Add the illustration images to the image side.

HINT: Connect a label to its input: the label's **for** attribute must match the input's **id**. Give the input **type="email"** and a **placeholder**.

HINT: Always give images an **alt** attribute. Decorative images can have an empty one.

Checkpoint: refresh. You should see an ugly but complete page with every piece of text and every image.

PART 3

Reset and set up the page

Goal: Start from a clean base.

- Remove the default spacing from every element with the ***** selector (**margin**, **padding**).
- Add **box-sizing: border-box** in the same rule. Search what it does and why everybody uses it.
- Load the Roboto font from **assets/fonts** using **@font-face** (you need one rule per weight: 400 and 700).
- On **body** set the font-family, font-size and the dark blue background color.

HINT: Colors in the style guide are written as **hsl(...)**. You can use them as they are.

PART 4

Center the card with Flexbox

Goal: Place the white card in the middle of the screen.

- Make **body** a flex container with **display: flex**.
- Use **justify-content** to center horizontally and **align-items** to center vertically.
- Give the body a **min-height** so there is room to center in (think of the full screen height: **100vh**).
- Give the card a white background.

REMEMBER: **justify-content** works along the main axis, **align-items** works along the cross axis. Try changing the values and watch what moves.

PART 5

Build the card layout

Goal: Text on the left, image on the right, like the design.

- Make the card a flex container with **display: flex**. Children sit side by side by default.
- Give the card a **max-width**, **padding**, rounded corners (**border-radius**) and a **box-shadow**.
- Use **gap** to put space between the text side and the image side.
- Make the text side take the remaining space with **flex: 1**.
- Give the text side some padding, and make it a flex column with a **gap** between heading, paragraph, list and form.
- Give the image a fixed **width** (look at the design, it is about 400px).
- Use **align-items** so the text is vertically centered next to the image.

HINT: This guide is for desktop only, so check your page in a wide browser window (around 1440px).

PART 6

Style the text and the list

Goal: Match the typography and the check list.

- Set the **h1** size and weight. Compare it with the design, it is large and bold.

- Set a comfortable **line-height** on paragraphs.
- Remove the bullet dots of the list: **list-style**.
- Make each **li** a flex container so the icon and the text sit on one line.
- Use **align-items** to vertically center the icon with the text and **gap** to space them.
- Stack the list items vertically with spacing between them (the **ul** can be flex with a column direction).

PART 7

Style the form

Goal: A clean input and a dark button.

- Make the **form** a flex column with a **gap**.
- Make the label small and bold.
- Style the input: **padding**, **border**, **border-radius**, **font-size**.
- Inputs do not inherit the font by default. Search **font-family: inherit**.
- Style the button: background color, white text, bold, **padding**, **border-radius**, no border.
- Add **cursor: pointer** to the button.
- Add a hover effect with **:hover**. In the design the button turns into a red/orange gradient.

HINT: A gradient is a background image. Search for **linear-gradient** on MDN.

PART 8

Build the success page

Goal: A second page shown after subscribing.

- Create **success.html** and link the same **style.css**.
- Add a card with: the success icon, an **h1**, a paragraph and a dismiss button.
- Reuse the card styles you already wrote.
- Make it narrower than the sign-up card (**width** or **max-width**).
- On the form in **index.html**, set the **action** attribute to the new page.
- Make the dismiss button a link (**a** tag) that goes back to **index.html**, then style it like a button.

HINT: Links are inline by default and have underlines. You will need **text-decoration** and **text-align** to make a link look like your button.

BONUS: Add the word **required** to the email input and the browser validates the email for you.

Final check

Goal: Compare with the design and clean up.

- Compare your page with the design side by side. Check spacing, font sizes and colors.
- Check it at a desktop width (around 1440px).
- Is there any horizontal scrollbar? If so, find the element that is too wide.
- Are all images showing? Check file paths in the browser console (F12).
- Is the HTML indented and the CSS grouped by sections with comments?
- Can you reach the input and the button using only the keyboard (Tab key)?

When you get stuck

- Right-click the page and choose **Inspect** to see which CSS rules apply to an element.
- Give an element a temporary **border: 1px solid red** to see its size and position.
- Read about any property on **developer.mozilla.org** (search: MDN + property name).
- Practice flexbox online with **Flexbox Froggy**.
- Change one thing at a time. If it breaks, undo the last change.

Flexbox cheat sheet

Property	Put it on	What it does
display: flex	parent	Turns children into flex items
flex-direction	parent	row (side by side) or column (stacked)
justify-content	parent	Aligns children along the main axis
align-items	parent	Aligns children along the cross axis
gap	parent	Space between children
flex: 1	child	Child grows to fill free space
flex-shrink	child	Controls whether a child may get smaller