

# CSS

## LAB GUIDE

1. Create a navigation menu for Example Consulting like the one shown below. The navigation menu should remain fixed on top of the page even as the user scrolls. The page and footer too share same margins to left and right as in the navbar. Link text should appear with the red shade used in the navigation menu, when the link is hovered upon.

### Checklist

- \* A reset / normalize stylesheet is used.
- \* Styles for common utilities like margins are maintained in a separate stylesheet
- \* App-wide styles are maintained in a separate stylesheet
- \* Every widget's styles is maintained in its own stylesheet (eg. nav styles in a separate stylesheet)
- \* Stylesheets are included in logical order (more general stylesheets being above less general ones).
- \* The width of the content within the page (including navigation menu and footer) is 90% of the page, but should not exceed 1140px. The rest of horizontal space on the page is to be distributed to the left and right in equal measure.
- \* The navigation link for the current page is highlighted in red using a line below the link
- \* The navigation menu is fixed on top. Make sure the page has no horizontal scrollbars.
- \* The links are lightgray in normal state. They turn from lightgray to white when hovering.
- \* The logo appears slightly dulled in normal state. When hovered upon it should become slightly bright.
- \* Clicking the logo takes one to the home page
- \* Some space around the link should also be clickable. This allows enough space to click even when the link is not wide enough.

### Notes

- \* Variants of the logo are available in the logo folder. You can also create one yourself if you know a tool like Photoshop, or can create your own one using a service like <https://hatchful.shopify.com/>.
- \* Colors used are lightgray, navy, rgb( 241, 86, 90 )
- \* The height of the navigation menu is 64px. The size of font for navigation menu text is 24px. You can estimate the dimensions elsewhere accordingly.

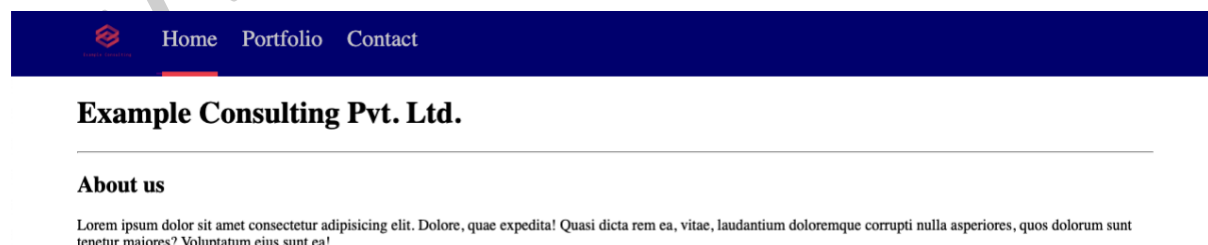


Figure 1 – Navigation menu of Example Consulting

2. **THEMING (PANEL):** Design a panel component according to the following specifications.
  - A heading with a background color

- A body with some content
- It should support 3 themes – primary (color: navy), secondary (color: olive) and error (color: crimson)

Define a base class for panel and modifier classes panel-primary, panel-secondary and panel-error in your design. Classes applicable to heading, body should be named appropriately.

The panels should set appropriate padding, borders etc.

A sample is shown below for the panels of different themes.



Figure 2a – Panel with primary theme color



Figure 2b – Panel with secondary theme color



Figure 2c – Panel with error theme color

**3. THEMING (BUTTON):** Design a button component according to the following specifications.

- It should have styles for states like hover / active (at least one should be different from the base style)
- Using your classes, one should be able to style links as buttons as well
- It should support 3 themes – primary (color: navy), secondary (color: olive) and error (color: crimson)

Use the base class - modifier class pattern in your design. Classes should be named appropriately.

A sample is shown below for the buttons of different themes.

The Google Search button is actually a link styled as a button and takes user to the Google search page.



Figure 3a – Buttons with various theme colors

Also try creating buttons of fixed width which ellipsize text when the width is not sufficient to contain the text. Sample is shown below. The following CSS properties can be helpful – overflow, text-overflow, white-space.



Figure 3b – Buttons with overflowing text being ellipsized

4. In a separate stylesheet, define custom CSS properties (CSS variables) for the colors in the theme. Use the styles defined in this to set the colors in the panel and the button widgets. Include all the stylesheets on a page, create some panels and buttons, and check it works.
5. Using flex box layout, design the following portion of a webpage on <https://www.telstra.com.au/>. Define classes for row and column of the grid layout and implement it.

## Latest updates

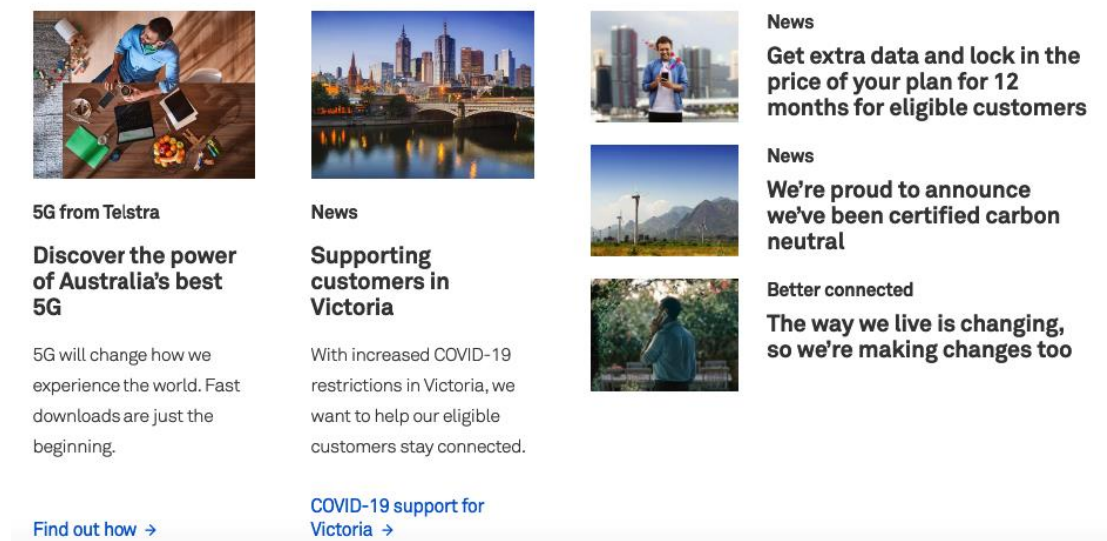


Figure 5 – A portion of Telstra website with column layout

**EXTRA CREDITS :** Make the page responsive using media queries. The column should stack up on small width devices. A mobile-first approach is recommended when designing the media queries.

6. Create a navigation menu like the one on the <https://www.apple.com> website. Make sure to form semantic markup using nav, ul/ol, li, a. Place them next to each other using display: inline-block for the list items containing the links. The navigation menu follows a stacked layout, with buttons to open and close the menu, on smaller width devices. Use a mobile-first approach.



Figure 6a - Navigation menu – mobile view (from <https://www.apple.com>)

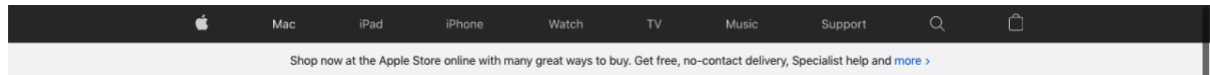


Figure 6b - Navigation menu - desktop view (from <https://www.apple.com>)

7. Recreate the menu above using float.
8. Recreate the menu above using float
9. Recreate sections of the <https://www.apple.com> website home page using the assets provided. Use a mobile-first approach.
10. Recreate the footer on the <https://www.apple.com> website. The footer follows a 5-column layout on desktop and stacked layout (with collapsed links) on smaller width devices. Use a mobile-first approach.

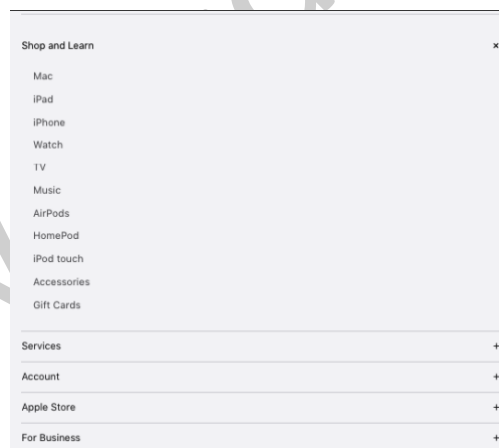


Figure 10a – Footer – mobile view (from <https://www.apple.com>)

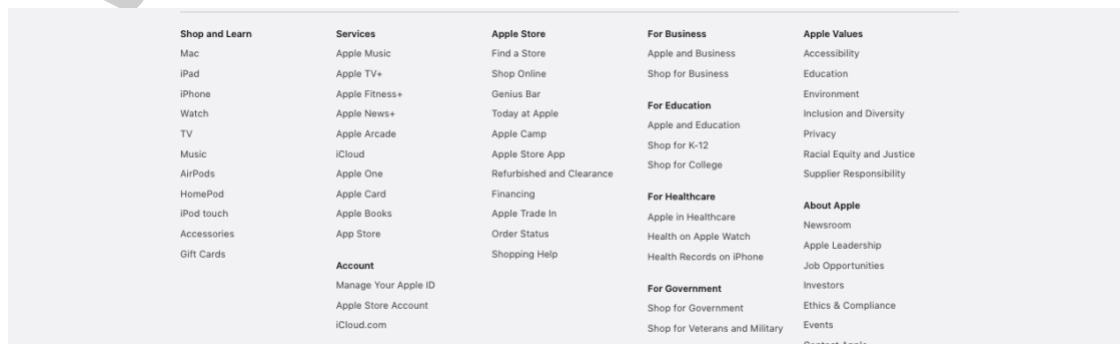


Figure 10b – Footer - desktop view (from <https://www.apple.com>)

11. Explore what CSS Sprites are and use them - <https://css-tricks.com/css-sprites/>
12. Use fixed position to build a login dialog like the one on Flipkart site. The dialog box should have an overlay in semi-transparent gray color behind. The close button should appear on the top right and just outside the dialog (use a combination of absolute and relative position). When the dialog is open, the page body should not be scrollable (**Hint**: Set overflow to hidden on the body).

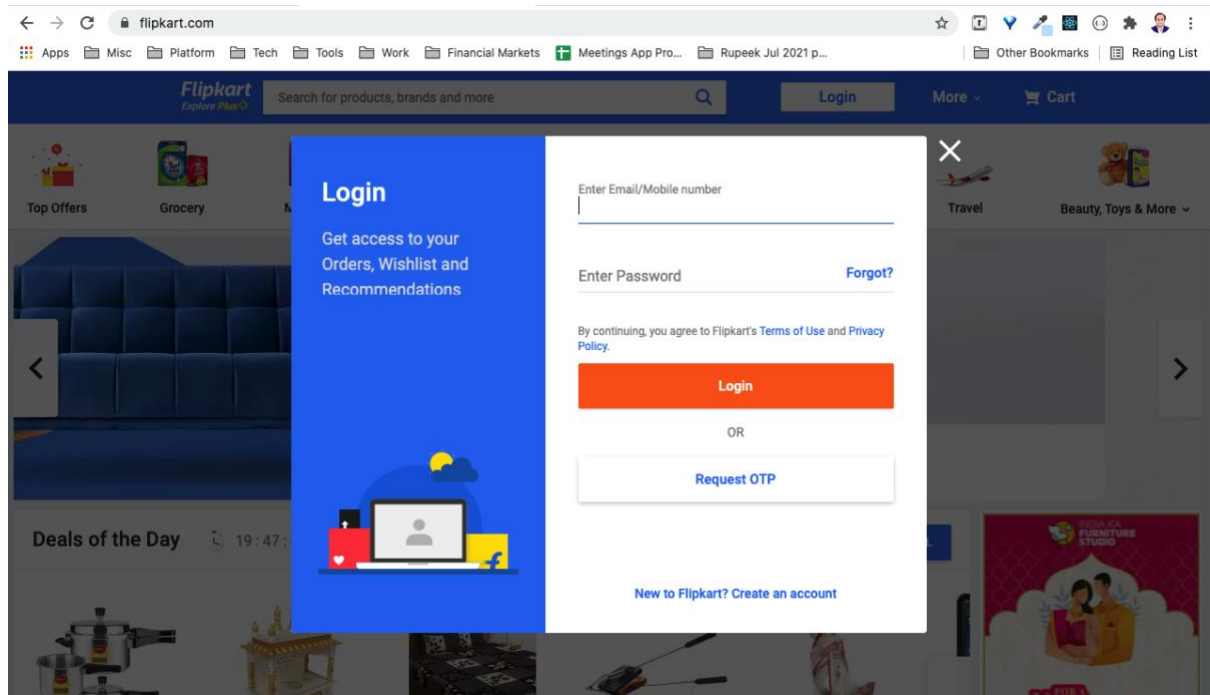


Figure 12 – Dialog box

13. Use absolute position to create a dropdown menu that opens up when user hovers over the “More” item. Use fixed /sticky positioning to have the navigation menu fixed on top of the page.

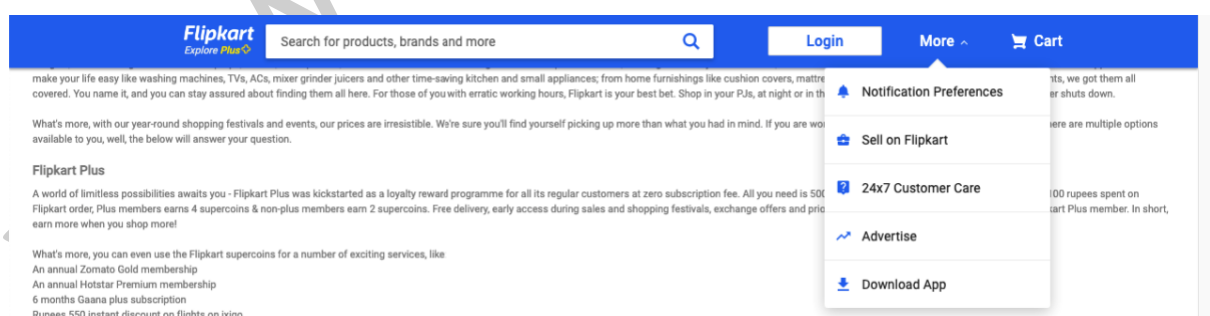


Figure 13 – Navigation menu with dropdown

14. Use flex box to build a responsive mobile-first approach to build a section that looks like so.

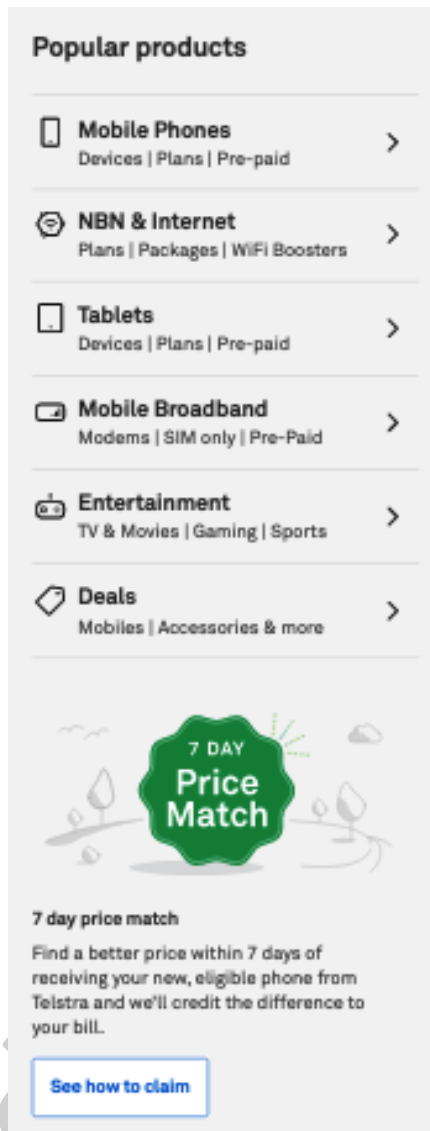


Figure 14a - Mobile View

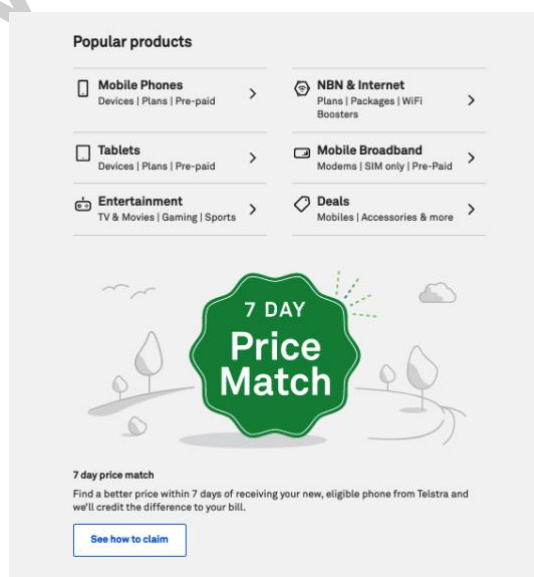


Figure 14b - Tablet view

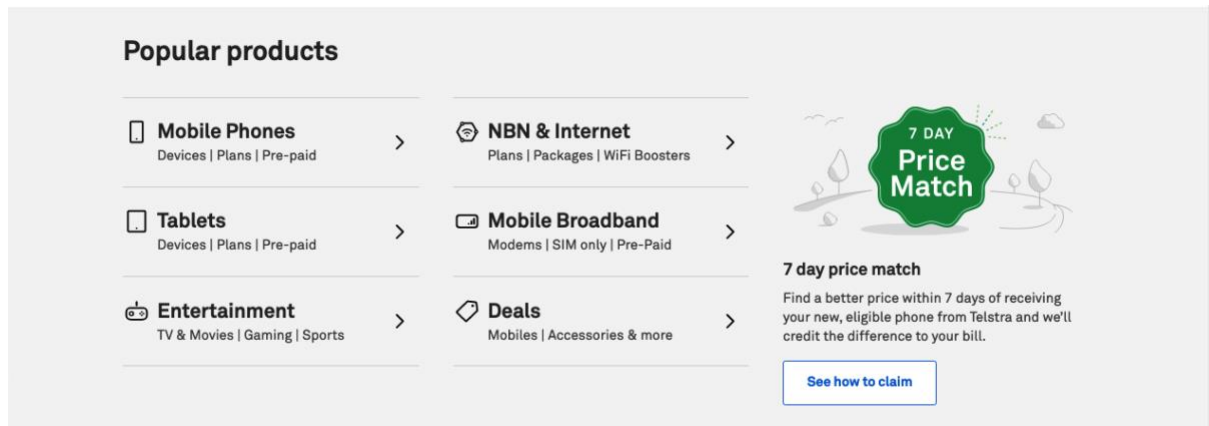


Figure 14c - Desktop view

## MORE EXERCISES:

1. Style the contact form in the Example Consulting website so that it looks good on both mobile and desktop. Use a mobile-first approach.
2. Build a form of your choice with various types of inputs. For example, a travel booking form with hotel reservation, airline search, user profile form etc. Use a mobile-first approach.
3. Design a 3-column layout with first column being twice the width of 2<sup>nd</sup> and 3<sup>rd</sup> columns
4. Design the 4-column layout found on <https://www.microsoft.com/en-in/> home page. Design using float. Redesign using flex box. Note that the layout goes from stacked on mobiles, to 2-column on tablets, to 4-column on desktops.

