CSS Explanation

css

body {

margin: 0;

font-family: Arial, sans-serif;

}

* margin: 0;: Removes all default margins from the body element, ensuring no unexpected spacing.
* font-family: Arial, sans-serif;: Sets the default font family for the entire document to Arial, with sans-serif as a fallback.

css

.hero-section {

background: url('hero-background.jpg');

background-size: cover;

background-position: center;

height: auto;/\*calc(100vh - 20px); \*/

margin: 10px;

padding-top: 60px; /\*Add paddings to prevent it from overlapping with the navbar\*/

display: flex;

justify-content: center;

align-items: flex-start;

}

* background: url('hero-background.jpg');: Sets a background image for the hero section.
* background-size: cover;: Ensures the background image covers the entire hero section.
* background-position: center;: Centers the background image.
* height: auto;/\*calc(100vh - 20px); \*/: Sets the height to auto. The commented-out part calc(100vh - 20px); would make the height of the hero section adjust to the viewport height minus 20 pixels.
* margin: 10px;: Adds a 10px margin on all sides of the hero section.
* padding-top: 60px;: Adds a 60px padding at the top to prevent overlapping with the navbar.
* display: flex;: Uses Flexbox for layout within the hero section.
* justify-content: center;: Centers the content horizontally.
* align-items: flex-start;: Aligns the content to the start (top) of the flex container.

css

.hero-content {

/\* background-color: rgba(93, 93, 93, 0.111); \*/

padding: 20px;

height: 170px;

border-radius: 10px;

color: rgb(77, 76, 74);

display: flex;

width: 100%;

}

* /\* background-color: rgba(93, 93, 93, 0.111); \*/: The background color is commented out. If enabled, it adds a semi-transparent background color to the content.
* padding: 20px;: Adds 20px padding on all sides of the content.
* height: 170px;: Sets a fixed height for the content.
* border-radius: 10px;: Rounds the corners of the content with a 10px radius.
* color: rgb(77, 76, 74);: Sets the text color to a specific shade of gray.
* display: flex;: Uses Flexbox for layout within the content.
* width: 100%;: Makes the content span the full width of its container.

css

.hero-left {

flex: 1;

display: flex;

align-items: center;

justify-content: flex-start;

padding: 10px;

font-size: 20px;

}

* flex: 1;: Allows the hero-left section to grow and fill available space proportionally.
* display: flex;: Uses Flexbox for layout within the left section.
* align-items: center;: Centers the items vertically within the left section.
* justify-content: flex-start;: Aligns the items to the start (left) of the flex container.
* padding: 10px;: Adds 10px padding on all sides of the left section.
* font-size: 20px;: Sets the font size of the text within the left section to 20px.

css

.hero-right {

flex: 1;

display: flex;

align-items: center;

justify-content: flex-end;

padding: 10px;

}

* flex: 1;: Allows the hero-right section to grow and fill available space proportionally.
* display: flex;: Uses Flexbox for layout within the right section.
* align-items: center;: Centers the items vertically within the right section.
* justify-content: flex-end;: Aligns the items to the end (right) of the flex container.
* padding: 10px;: Adds 10px padding on all sides of the right section.

css

.hero-title {

font-size: 3em;

margin: 0;

}

* font-size: 3em;: Sets the font size of the hero title to 3em.
* margin: 0;: Removes any default margins from the hero title.

css

.hero-description {

font-size: 1.2em;

margin: 0;

}

* font-size: 1.2em;: Sets the font size of the hero description to 1.2em.
* margin: 0;: Removes any default margins from the hero description.

css

.hero-content:hover {

transform: scale(1.03);

background: #58786288;

transition: 2s;

}

* transform: scale(1.03);: Scales the content slightly when hovered.
* background: #58786288;: Changes the background color to a semi-transparent shade when hovered.
* transition: 2s;: Smooths the transition effect over 2 seconds.

Comments for Non-Working Part

css

/\*

.hero\_Image\_container {

text-align: center;

}

.hero\_image {

display: inline-block;

width: calc(100% \* .5);

}

Does not WORK\*/

* The commented-out section indicates that centering and resizing the hero image this way did not work.

css

.main-image-container {

position: relative; /\* Set the container to relative position \*/

}

* position: relative;: Allows absolutely positioned elements inside this container to be positioned relative to it.

css

.main-image {

display: block; /\* Ensure it's treated as a block element \*/

width: 100%; /\* Full width of the container \*/

height: auto; /\* Maintain aspect ratio \*/

}

* display: block;: Makes the .main-image behave like a block-level element.
* width: 100%;: Ensures the image covers the full width of its container.
* height: auto;: Maintains the aspect ratio of the image.

css

.circle-image {

position: absolute; /\* Position absolutely within the container \*/

width: 20%; /\* Adjust size as needed \*/

height: auto; /\* Maintain aspect ratio \*/

bottom: 0; /\* Align the bottom of the circle image with the bottom of the main image \*/

left: 50%; /\* Center horizontally \*/

transform: translateX(-50%) translateY(50%); /\* Adjust positioning \*/

border-radius: 50%;

border: 4px solid black;

}

* position: absolute;: Positions the .circle-image relative to the .main-image-container.
* width: 20%;: Sets the width to 20% of the container's width.
* height: auto;: Maintains the aspect ratio of the circle image.
* bottom: 0;: Aligns the bottom of the circle image with the bottom of the main image.
* left: 50%;: Positions the circle image horizontally at the center.
* transform: translateX(-50%) translateY(50%);: Adjusts the position of the circle image to the center horizontally and a bit downwards.
* border-radius: 50%;: Makes the border circular.
* border: 4px solid black;: Adds a 4px solid black border around the circle image.