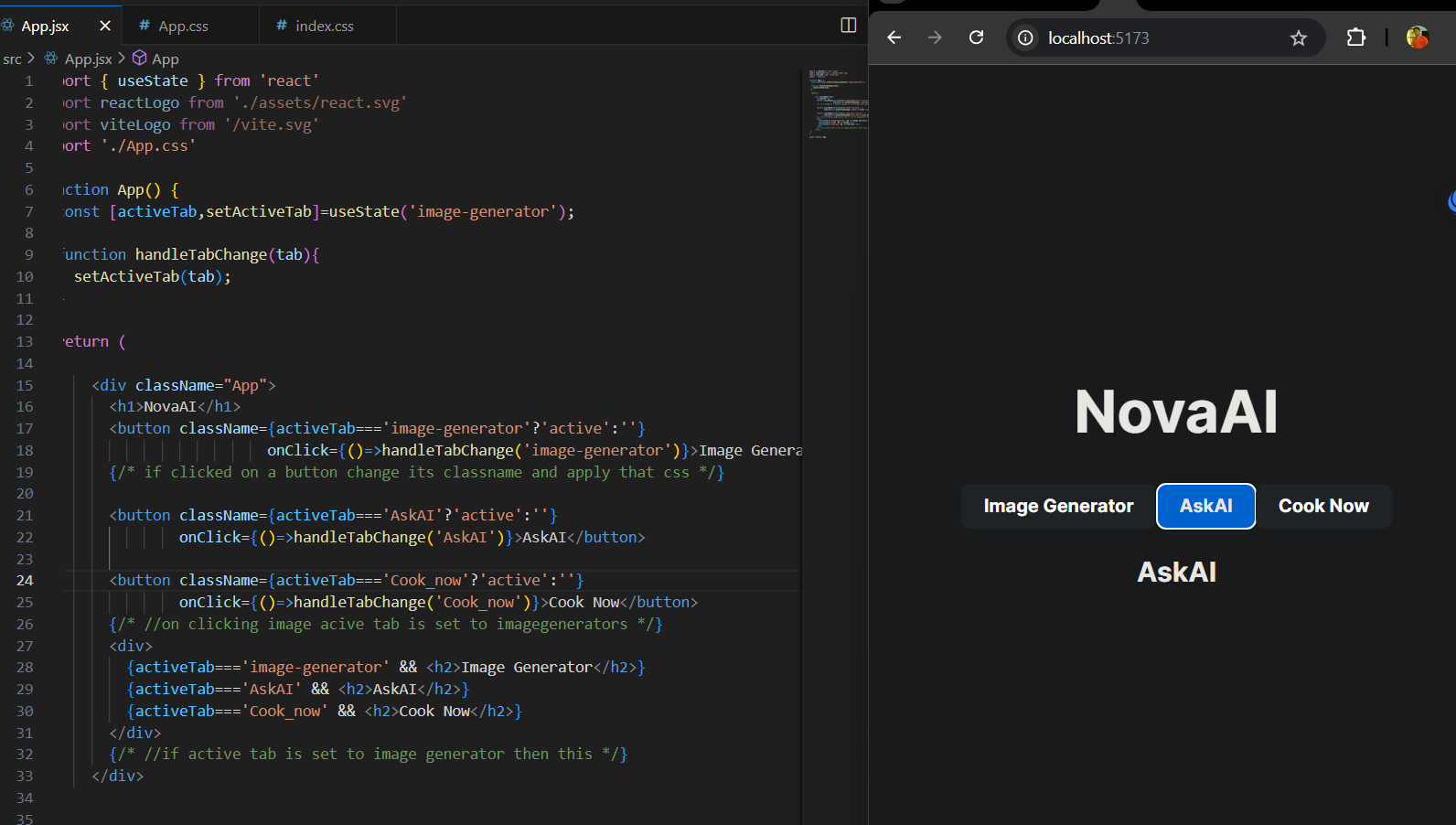
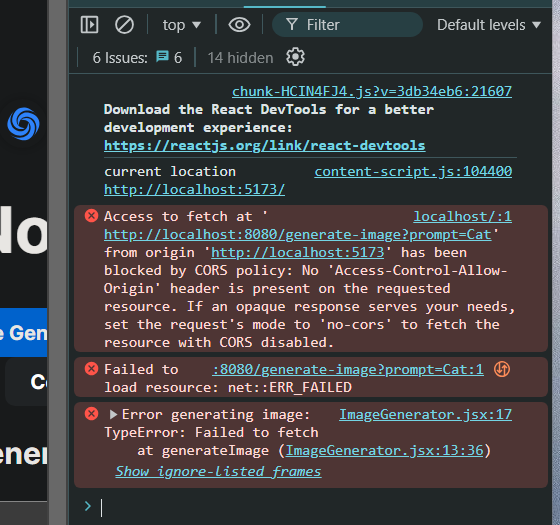
FrontEnd notes

## Conditional rendering

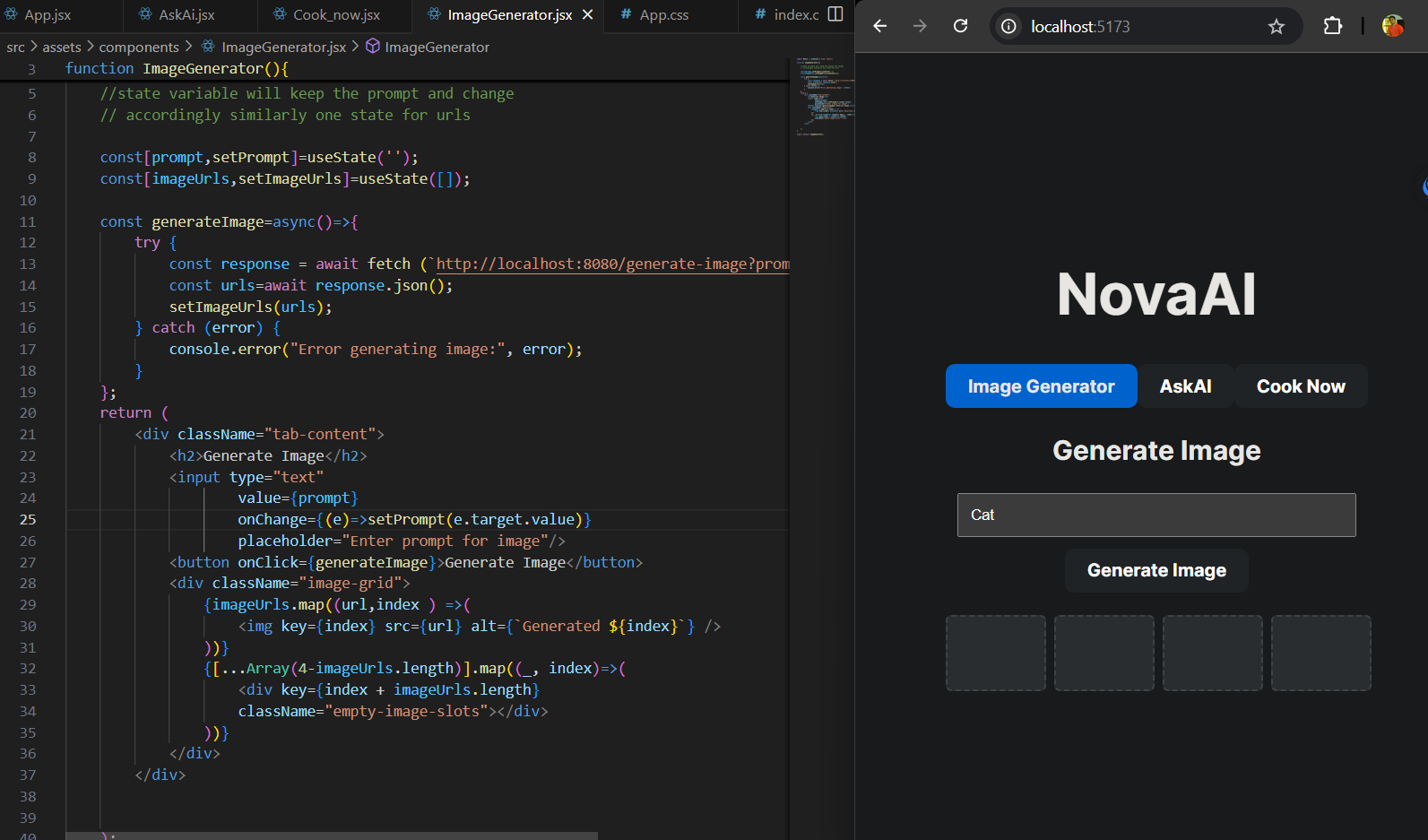


# Cors Error

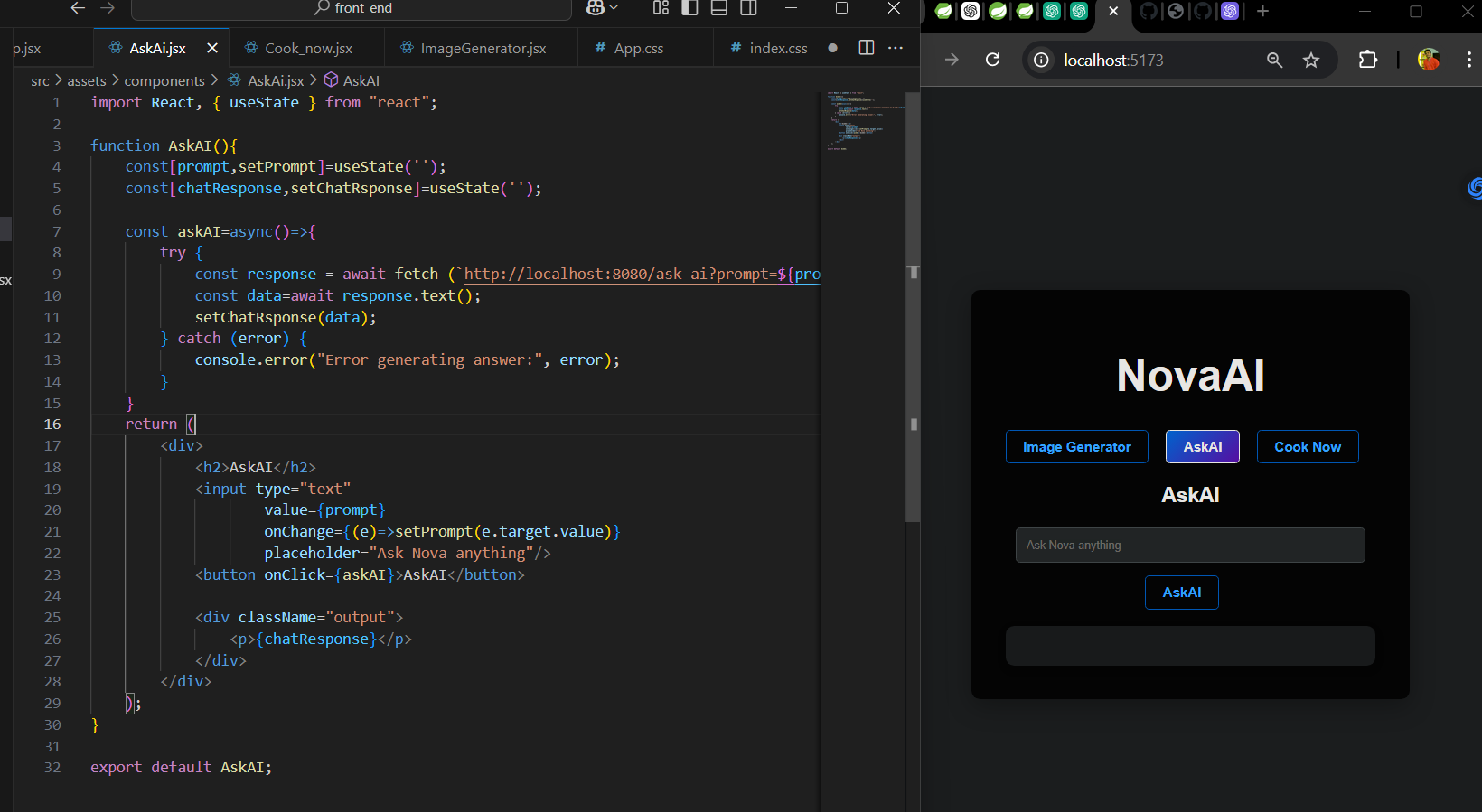


When u try to access a port from different port in this case 8080 and 3000

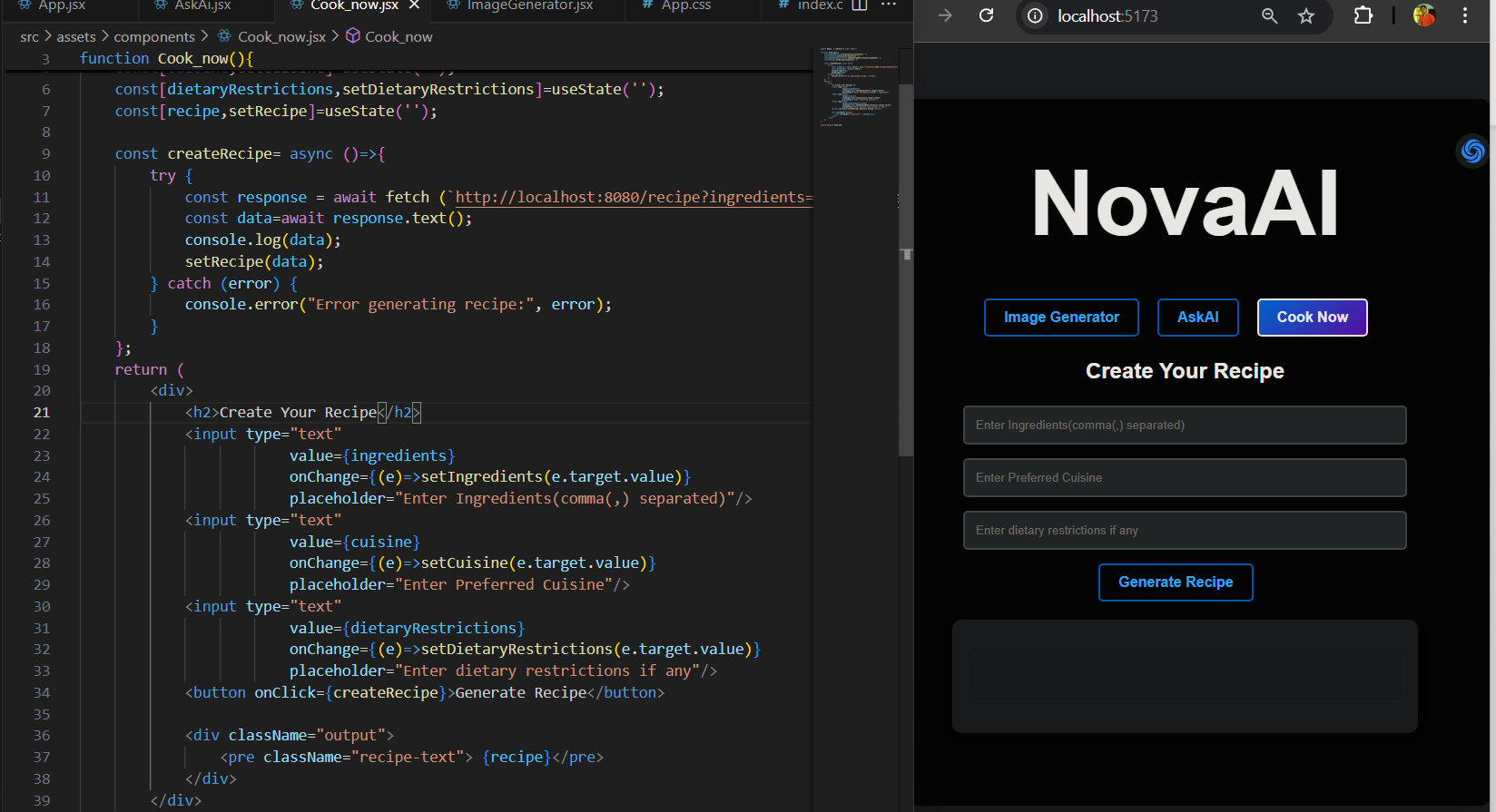
# Image Generator frontend



# AskAI Frontend



# Cook\_now frontend

0

# CSS

:root {

  --primary-color: #007bff;

  --secondary-color: #f0f0f0;

  --text-color: #333;

  --bg-light: #ffffff;

  --bg-dark: #242424;

  --gradient: linear-gradient(135deg, #007bff, #6a11cb);

}

body {

  font-family: 'Poppins', sans-serif;

  margin: 0;

  background: var(--bg-dark);

  color: #fff;

  display: flex;

  justify-content: center;

  align-items: center;

  min-height: 100vh;

}

h1{

  font-size: 100px;

  margin-top: -20px;

  margin-bottom: 50px;

}

.App {

  text-align: center;

  max-width: 800px;

  margin: 0 auto;

  padding: 80px;

  background: rgba(0, 0, 0, 0.85);

  box-shadow: 0 8px 30px rgba(0, 0, 0, 0.3);

  border-radius: 10px;

  animation: fadeIn 1s ease-in-out;

}

.tabs {

  display: flex;

  justify-content: space-around;

  margin-bottom: 20px;

}

button {

  padding: 10px 20px;

  cursor: pointer;

  border: 2px solid var(--primary-color);

  background: transparent;

  color: var(--primary-color);

  font-weight: bold;

  border-radius: 5px;

  transition: all 0.3s ease;

  margin-right: 20px;

}

button.active,

button:hover {

  background: var(--gradient);

  color: #fff;

  border-color: var(--gradient);

}

input {

  padding: 12px;

  width: calc(100% - 24px);

  margin-bottom: 15px;

  box-sizing: border-box;

  border: 2px solid #ccc;

  border-radius: 5px;

  background-color: #2c2c2c;

  color: #fff;

  outline: none;

  transition: border-color 0.3s ease;

}

input:focus {

  border-color: var(--primary-color);

}

.tab-content {

  margin-top: 20px;

}

.image-grid {

  display: grid;

  grid-template-columns: repeat(4, 1fr);

  gap: 15px;

  margin-top: 20px;

}

img {

  width: 100%;

  height: auto;

  border: 3px solid rgba(255, 255, 255, 0.1);

  border-radius: 10px;

  transition: transform 0.3s ease, box-shadow 0.3s ease;

}

img:hover {

  transform: scale(1.05);

  box-shadow: 0 4px 15px rgba(0, 0, 0, 0.5);

}

.empty-image-slots {

  width: 100%;

  height: 0;

  padding-bottom: 75%; /\* Maintain aspect ratio for empty slots \*/

  border: 2px dashed rgba(255, 255, 255, 0.5);

  border-radius: 5px;

  background: rgba(255, 255, 255, 0.1);

}

.output {

  margin-top: 20px;

  text-align: left;

  background: var(--bg-light);

  color: var(--text-color);

  padding: 15px;

  border-radius: 10px;

  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);

}

.recipe-text {

  max-width: 100%;

  padding: 15px;

  box-sizing: border-box;

  word-wrap: break-word;

  overflow-y: auto;

  font-size: calc(12px + 0.5vw);

  line-height: 1.8;

  border: 1px solid rgba(255, 255, 255, 0.2);

  border-radius: 10px;

  margin-bottom: 20px;

  background: rgba(255, 255, 255, 0.1);

  color: #fff;

}

/\* Add animations \*/

@keyframes fadeIn {

  from {

    opacity: 0;

    transform: translateY(20px);

  }

  to {

    opacity: 1;

    transform: translateY(0);

  }

}

Here are the important CSS properties used in the provided code, along with brief notes on their function:

### 1. **CSS Variables (**--variable-name**)**

* **Purpose**: Used to define reusable values for colors, gradients, and other properties.
* **Example**:
* --primary-color: #007bff;
* --secondary-color: #f0f0f0;
* --text-color: #333;

### 2. **Flexbox Layout**

* **Purpose**: Aligns and distributes content within a container in a flexible way.
* **Example**:
* body {
* display: flex;
* justify-content: center; /\* Centers content horizontally \*/
* align-items: center; /\* Centers content vertically \*/
* }

### 3. **Grid Layout (**grid-template-columns**,** gap**)**

* **Purpose**: Defines a grid layout for elements, specifying the number of columns and the gaps between them.
* **Example**:
* .image-grid {
* display: grid;
* grid-template-columns: repeat(4, 1fr); /\* 4 equal-width columns \*/
* gap: 15px; /\* Space between grid items \*/
* }

### 4. **Transitions (**transition**)**

* **Purpose**: Smoothly animates changes to CSS properties when they are triggered (e.g., on hover).
* **Example**:
* button {
* transition: all 0.3s ease; /\* Smooth transition for all properties \*/
* }

### 5. **Box Shadow (**box-shadow**)**

* **Purpose**: Adds shadow effects around elements to create depth.
* **Example**:
* .App {
* box-shadow: 0 8px 30px rgba(0, 0, 0, 0.3);
* }
* img:hover {
* box-shadow: 0 4px 15px rgba(0, 0, 0, 0.5);
* }

### 6. **Border Radius (**border-radius**)**

* **Purpose**: Rounds the corners of elements.
* **Example**:
* button {
* border-radius: 5px;
* }
* .App {
* border-radius: 10px;
* }

### 7. **Box Sizing (**box-sizing**)**

* **Purpose**: Controls how the width and height of an element are calculated (including padding and border).
* **Example**:
* input {
* box-sizing: border-box;
* }

### 8. **Font and Text Styling**

* **Purpose**: Defines the font-family, font-size, line-height, and text alignment.
* **Example**:
* body {
* font-family: 'Poppins', sans-serif;
* }
* .recipe-text {
* font-size: calc(12px + 0.5vw); /\* Dynamic font size based on viewport width \*/
* }

### 9. **Background Styling (**background**,** background-color**)**

* **Purpose**: Sets the background color or image of an element.
* **Example**:
* body {
* background: var(--bg-dark); /\* Using a CSS variable \*/
* }
* input {
* background-color: #2c2c2c; /\* Solid background for input \*/
* }

### 10. **Hover Effects (**:hover**)**

* **Purpose**: Applies styles when an element is hovered over by the user.
* **Example**:
* button:hover {
* background: var(--gradient);
* color: #fff;
* border-color: var(--gradient);
* }
* img:hover {
* transform: scale(1.05); /\* Slightly enlarges image \*/
* }

### 11. **Focus State (**:focus**)**

* **Purpose**: Defines styles for an element when it is focused (e.g., on form inputs).
* **Example**:
* input:focus {
* border-color: var(--primary-color); /\* Highlights border when focused \*/
* }

### 12. **Keyframe Animations (**@keyframes**)**

* **Purpose**: Defines animations with multiple steps (e.g., fade-in).
* **Example**:
* @keyframes fadeIn {
* from {
* opacity: 0;
* transform: translateY(20px);
* }
* to {
* opacity: 1;
* transform: translateY(0);
* }
* }

These properties play a crucial role in creating a visually appealing, interactive, and responsive layout.