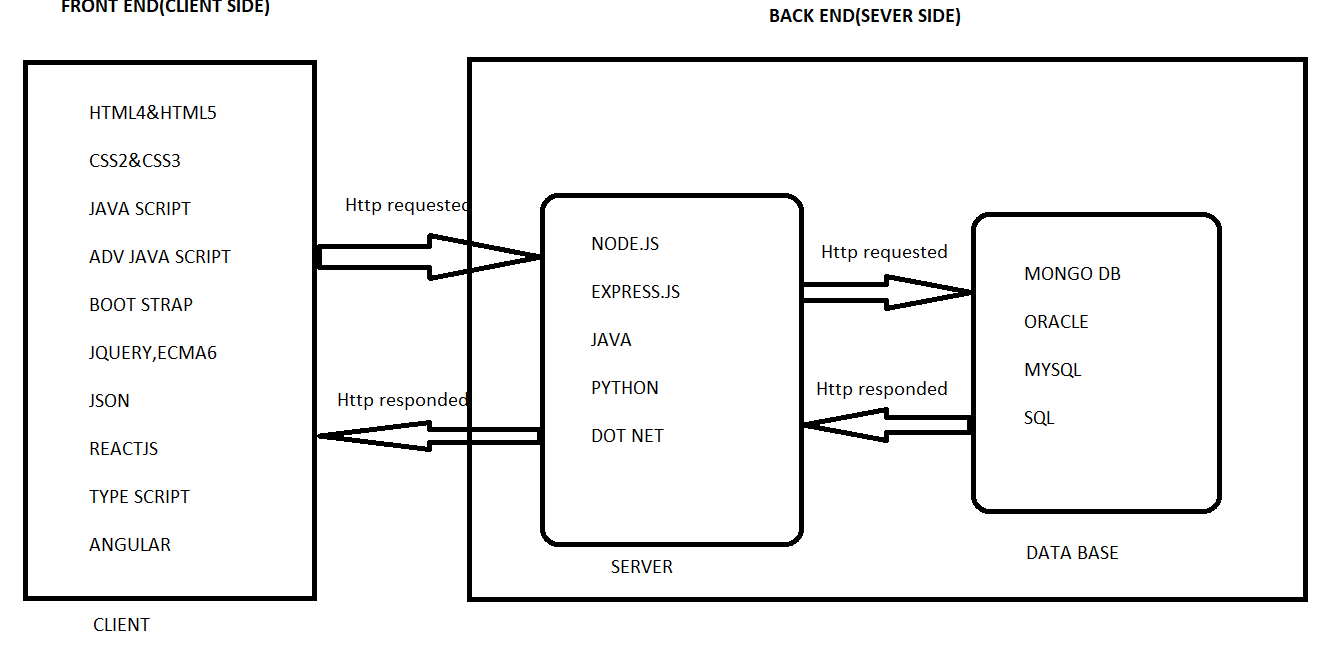
UI WEB DEVELOPEMENT

**FRONT END AND BACK END TECHNOLOGIES:**



**IMG:1-(UI DEVELOPMENT &TECHNOLOGIES)**

**MERN: Mongo DB, Express. JS, React JS, Node.JS.**

**MEAN: Mongo DB, Express. JS, Angular JS, Node.JS.**

**CLIENT SIDE (FRONT END):**

**It only defines UI which a user or client will unable to access the data in the data base.**

**1. HTML-Html stands for Hyper Text Markup Language it’s a using tags. Which is capable for display the data with in the page means the user or client were unable to access the data from the data from database.**

**2. CSS-**Css stands for **Cascading Style Sheet**. In order to add beautification to the html page **.**

**3. STATIC WEB PAGES-A static web page(sometimes called a Flat Page or a Stationary Page)is a web page that is delivered to the user’s web browser exactly as stored.**

**Ex**-College Web site (Html, Css).

**4. DYNAMIC WEB PAGES-**A dynamic web site(also referred to as a database-driven site)requires web programming and database design.

**Ex**- Flip card, Google, etc. (Html, Css, Js)

**5. JAVA SCRIPT-**Java script is a text-based programming language used both on the client side and server side that allows you to make web pages interactive.

**6. ECMA6-**European Computer Manufacturers Association .Ecma6 (ES6) was created to standardize java script.

**7. BOOTSTRAP-**Bootstrap is the most popular CSS framework for developing responsive and mobile first website.

**SERVER SIDE (BACK END):**

Code written by back end developers is what communicates the data base information to the browser.

**1. NODE.JS-** Node.js is a java script run time environment built on chrome’s V8 java script engine. The basic HTTP server, We will also be using file system, path and URL, All of which are native node.

**2. EXPRESS.JS-** Express.js is a free open source web application framework for node.js .It is used for designing and building web applications quickly and easily.

**DATA BASE (DB):**

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS).

**1. MONGO DB-** Mongo db is a document database with the scalability and flexibility that you want with the querying and indexing that you need.

**SOFTWARE USES:**

**1 .BROWSERS-** Edge, Internet Explorer, Google Chrome, etc…….

**2. EDITORS-** Sublime Text, Visual Studio Code.

**3. RESPOSITORY-** GitHub.

**4. TOOLS-** Node.JS.

**5. DATA BASE-** Mongo DB & Mongo DB Compass.

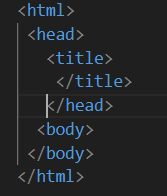
**HTML & CSS**

**HTML-** Hyper text markup language.

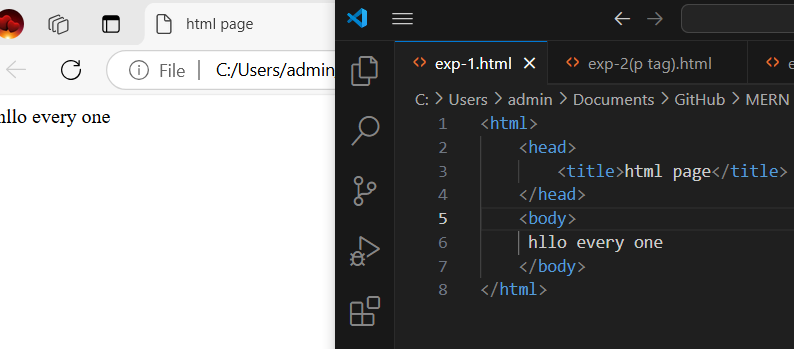
**Markup Language-** Language that uses tags to represent the content.

**EX-**<div>.....</div>

**Structure Of HTML:**

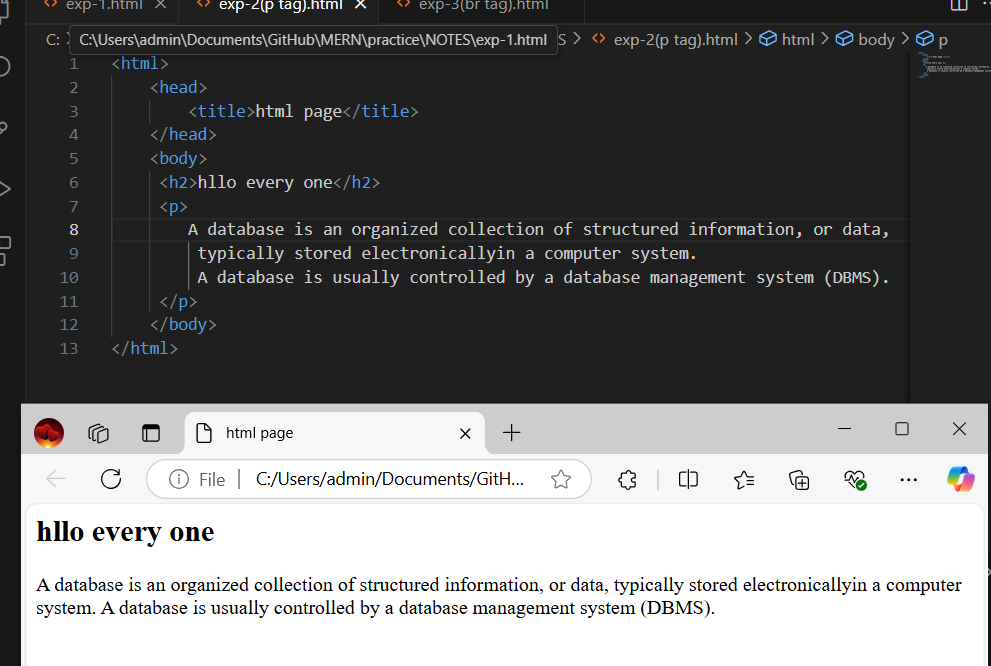
****

**IMG-2**

****

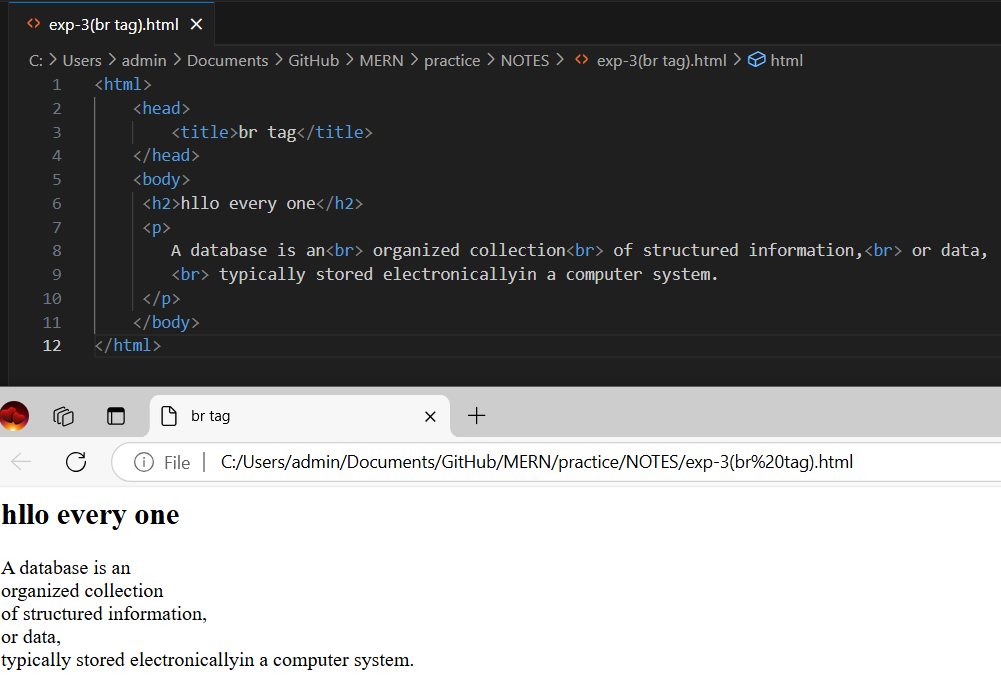
1. **<html> :** The hold complete content of the web page.
2. **<head> :** To indicate the head of the web page.
3. **<title> :** It holds the title of the web page.
4. **<body> :**To hold the actual content of the page.

**<P> Tag:** Paragraph tag used to hold the multiline text content.



**IMG-3:Paragraph tag**

**<br> Tag**: Break tag adds single line break.



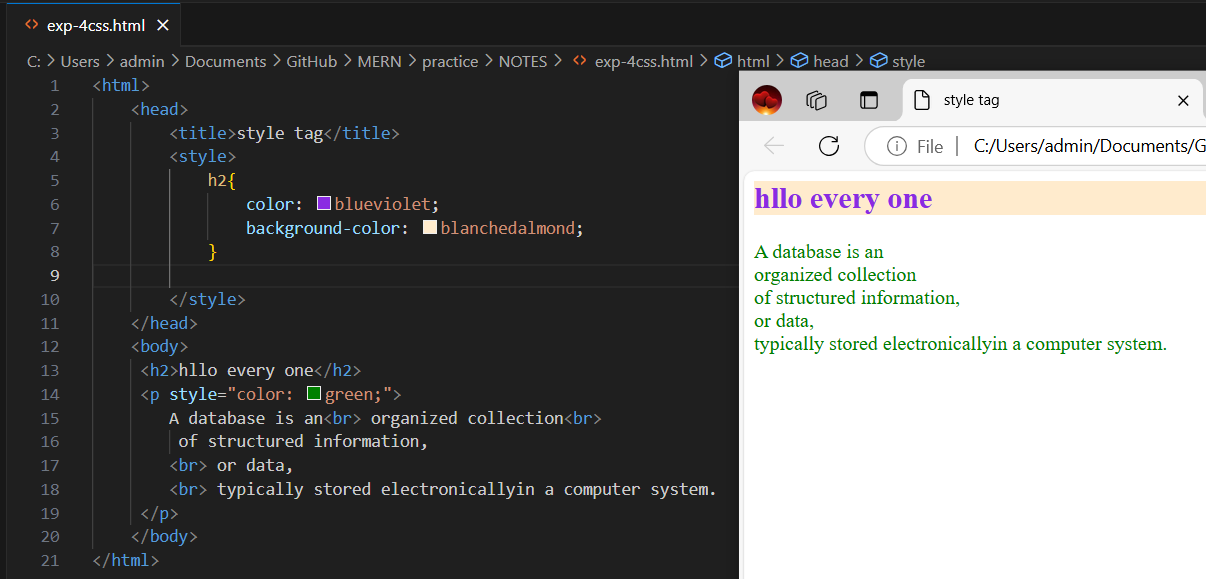
**IMG-4:BR TAG**

**CSS-**Cascading style sheet, An inline CSS is used to apply a unique style to a single element.

**STYLE ATTRIBUTE-** An inline CSS uses the style attribute of an html element. Style is a predefined tag capable of holding any no .of CSS classes.

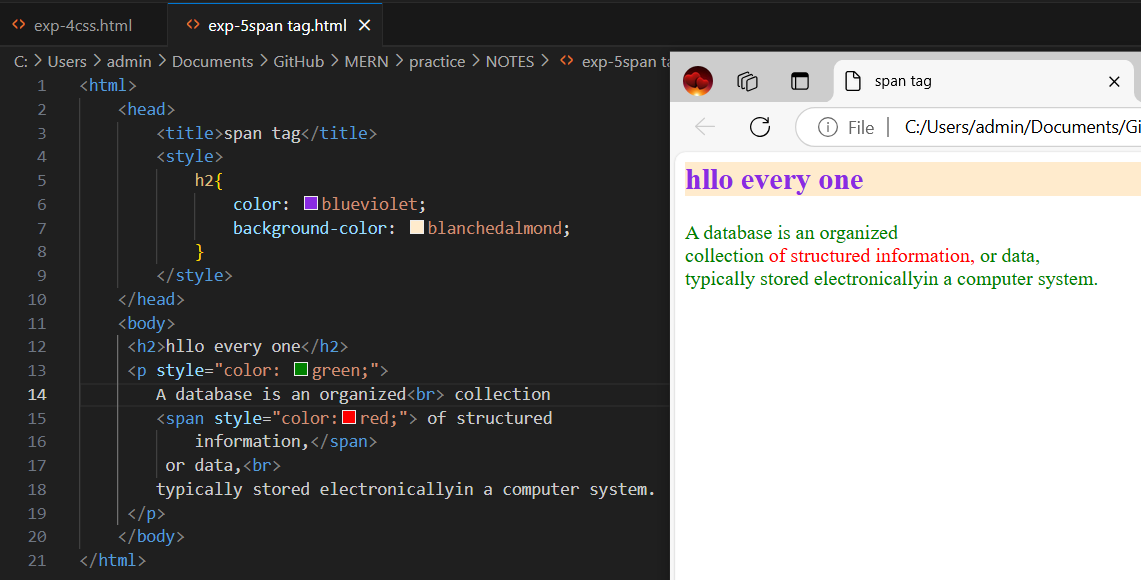
**1. COLOR:** The CSS color indicates the text color of a html page.

**2.BACKGROUND:** The CSS background indicates the background color of a html page.



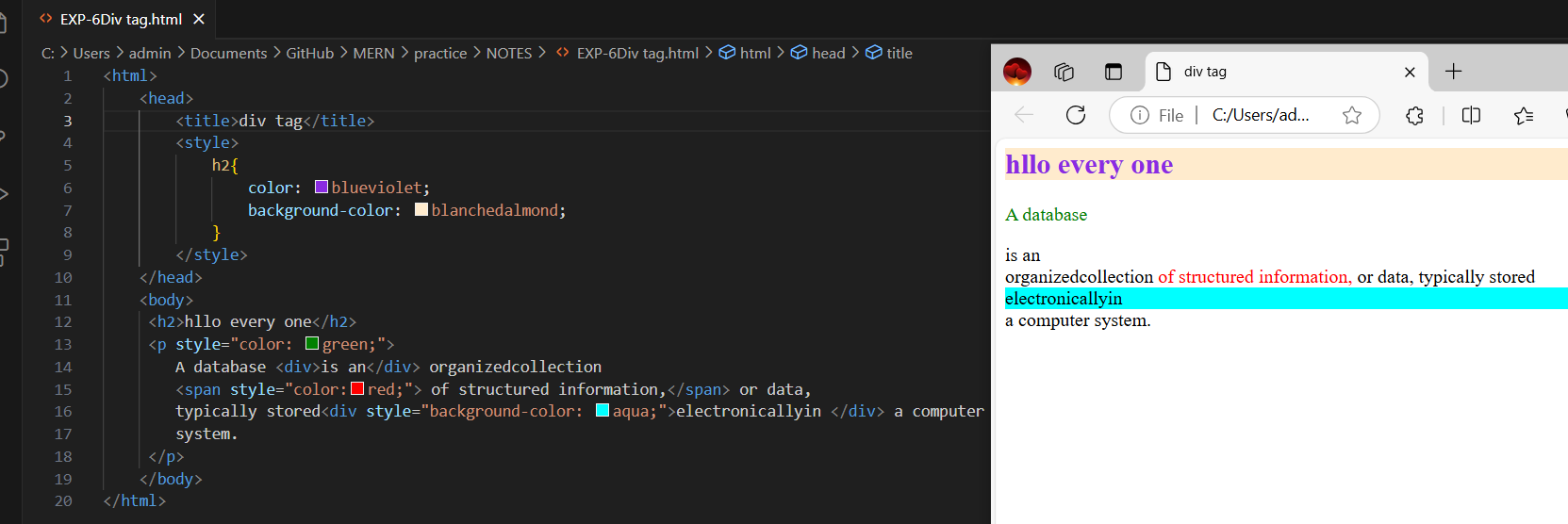
**IMG-5: CSS (STYLE TAG)**

**<span>Tag:** Inline elements to hold content in the same line. Best example for inline tag.



**IMG-6: SPAN TAG**

**<div>Tag:** Hold block content. Best example for block level tag.



**IMG-7: DIV TAG**

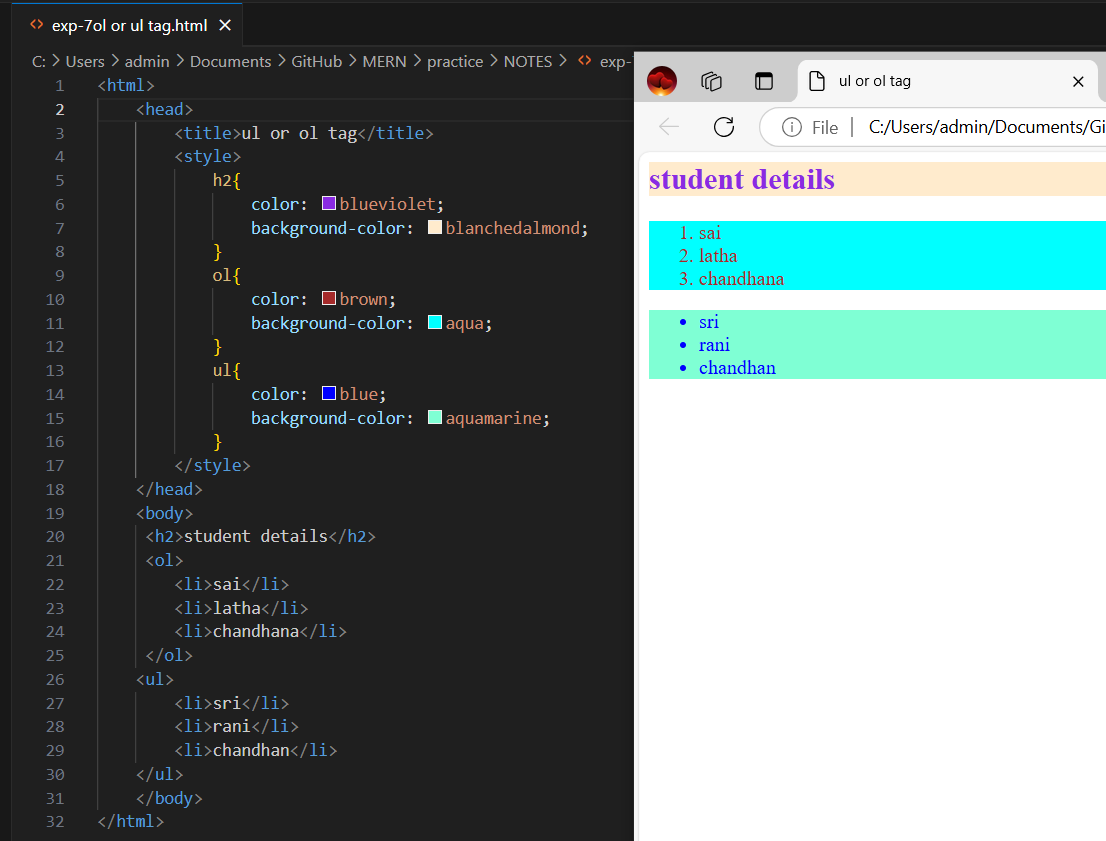
**<table>Tag:** To render the content in the row and column way.

**<ol> <ul>Tags:** To render the content in ordered or unordered way.

ol Tag by defult bullet symbol is coming

ul Tag by defult numeric number is coming

**<li>Tag:** To display the list of items.



**IMG-8: OL OR UL TAG**

**HTML ATTRIBUTES:** We can add extra information to the HTML elements through HTML attributes. The attributes can be both predefined and user defined.

**1.ID -** Using which we can unique reference to elements.

**2.NAME-** Name value can be added to elements.

**3.CLASS-** Using which we can add single/multiple css classes to element.

**4.ALT-** Using which we can add alternative text content.

**5.STYLE-** To add single to multiline CSS properties to the elements.

**6.TITLE-** Using which we can add title to any html elements.

**CSS PRIORITY:**As we have the four different ways to apply the css properties for the HTML elements.

**1. STYLE-**Among all the four different ways CSS been applied inline will always takes the higher priority in order.

**2. ID-**CSS been applied through ID, Which takes the second priority in order. Id name should be started with # operator.

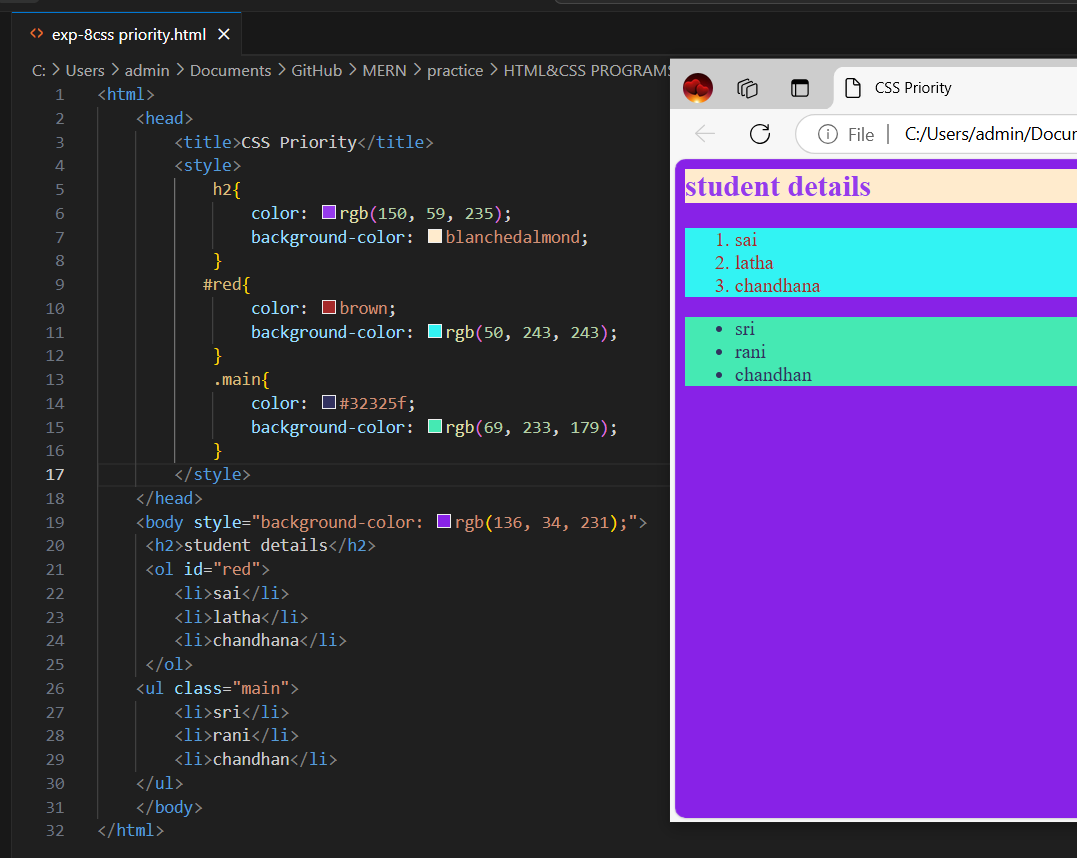
**3.CLASS-** CSS been applied through CLASS, Which takes third priority in order. Class name should be started with.(DOT) operator.

**4.TAG-** CSS been applied through TAG, Which takes least priority in order.

**CSS Color Values-**In CSS, colors can also be specified using RGB values.

**Ex-**background-color:rgb(255,99,17);

>RGB rang(-255 to 255)



**IMG-9: CSS PRIORITY &RGB**

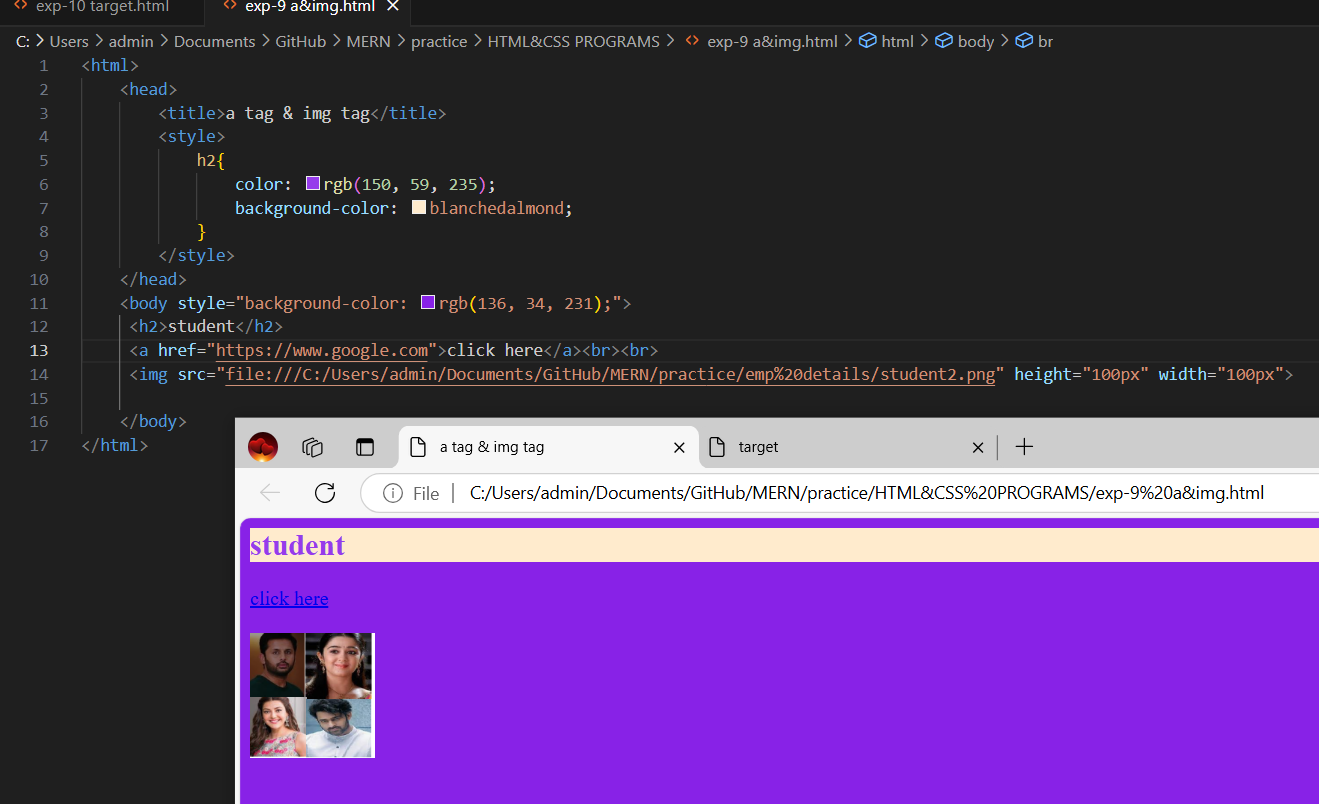
**ANCHOR TAG-**The <a>tag defines a hyperlink, Which is used to link from one page to another. The most important attribute of the <a>element is the **href** attribute, Which indicates the link’s destination.

**Ex-<**a href=”https://www.google.com”>press here</a>

**IMAGE TAG-**The <img>tag is used to embed an image in an HTML page. Images are not technically inserted into a web page; images are linked to web pages. The <img > tag creates a holding space for the referenced image.

**Ex-<**img src=”img\_213392.jpg”width=”100px”height=”500px”>

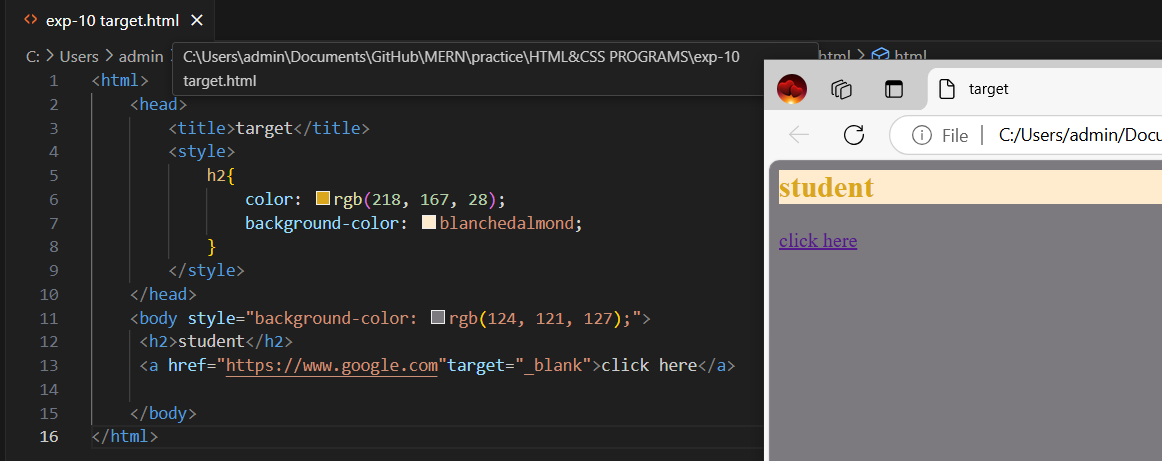
><img> tag is a self closing tag.



**IMG-10: ANCHOR TAG & IMG TAG**

**TARGET ATTRIBUTE:** The target attribute specifies a name or a keyword that indicates where to display the response that is received after submitting the form.

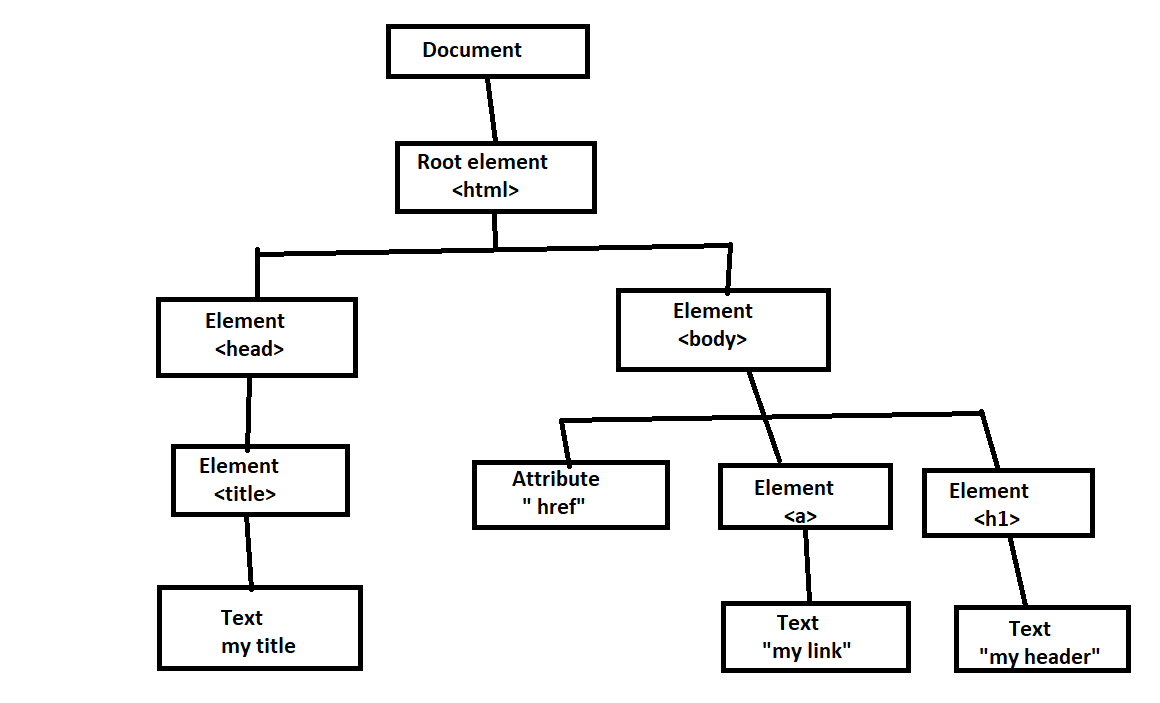
**Ex- <**a href=”<https://www.google.com>”target=”\_blank”>press here</a>



**IMG-11 TARGET**

**DOM(Document Object Model)Structure:**For every web page browser dynamically creates a DOM structure, Which indicates the tree structure of the current web pages with all the elements.

>The below diagram is example HTML tree structure.



**IMG-12 DOM STRUCTURE**

**CSS OVERFLOW:** The overflow property specifies whether to clip the content or to add scrollbars when the content of an element is too big to fit in the specified area.

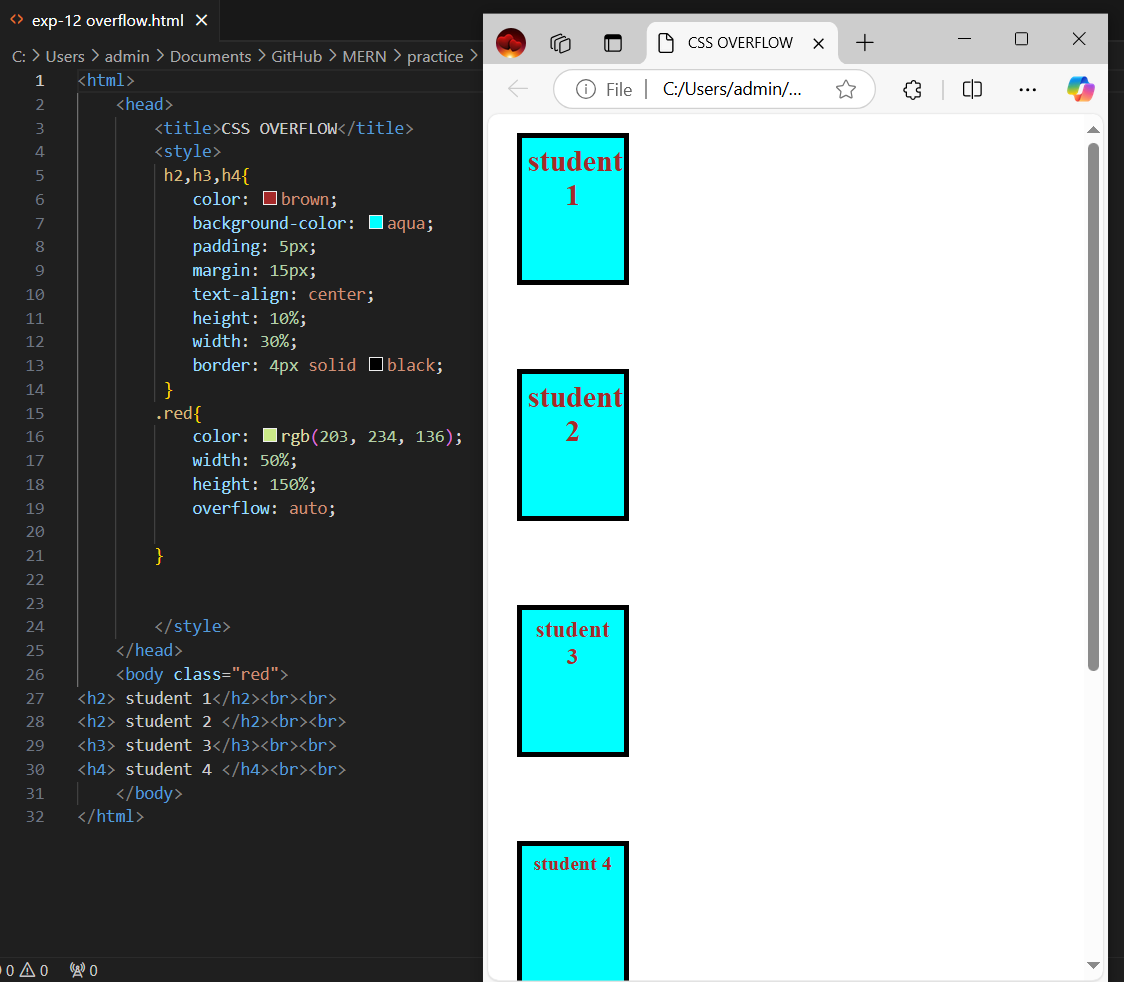
>The overflow property has the following values.

**1. VISIBLE-**Default . The overflow is not clipped. The content renders outside the element’s box.

**2. HIDDEN-** The overflow is clipped, and the rest of the content will be invisible.

**3. SCROLL-**The overflow is clipped, and a scrollbar is added to see the rest of the content.

**4.AUTO-**Similar to scroll, but it adds scrollbars only when necessary.



**IMG-13 OVERFLOW**

**CSS PADDING PROPERTY-**Through ‘padding’ property we could able to add space or gap between the border and the content of the container.

**Ex-**padding:5px;(adds padding of 5px to all the directions(top, left, right, bottom) )

Padding-left:5px; , padding-right:2px; ,padding-top:2px; , padding –bottem:1px;

**CSS MARGIN PROPERTY-** Through ‘padding’ property we could able to add space or gap above the border of the container.

**Ex-**margin:5px;(adds margin of 5px to all the directions(top, left, right, bottom) )

margin-left:5px; , margin-right:2px; ,margin-top:2px; , margin –bottem:1px;

**CSS BOX MODEL-** In CSS box model property while calculating the actual dimensions of an element ,it considers the following CSS properties.

>The total border space been occupied.

>The total margin space been occupied.

>Total padding space been occupied.

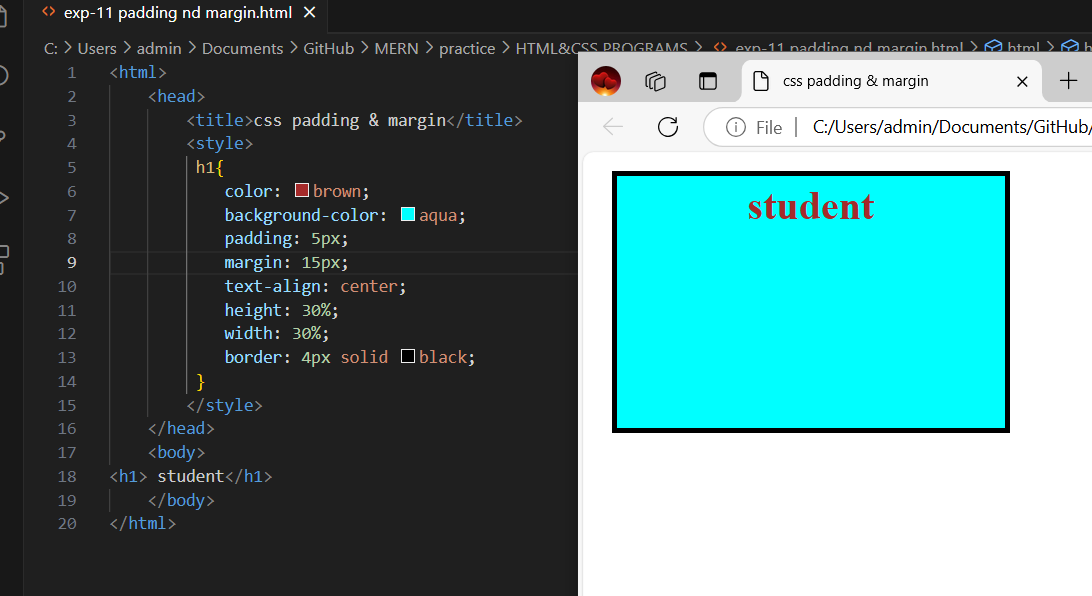
>Actual width & height of the DOM element.

**1. CONTENT-**The content of the box , Where text and images appear.

**2. PADDING-**Clears an area around the content. The padding is transparent.

**3. BORDER-**A border that goes around the padding and content.

**4. MARGIN-**Clears an area outside the border. The margin is transparent.

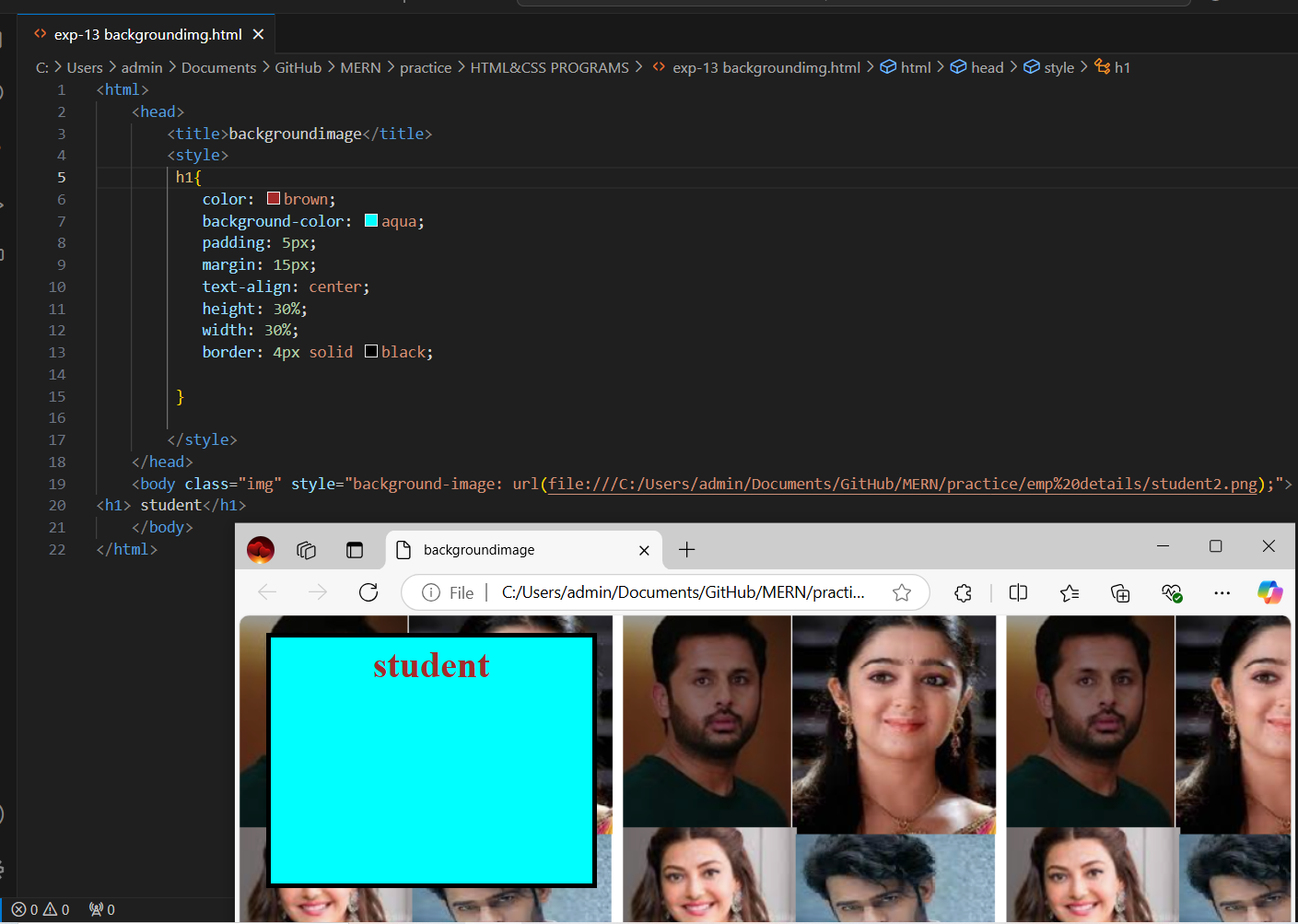


**IMG-14 PADDING &MARGIN**

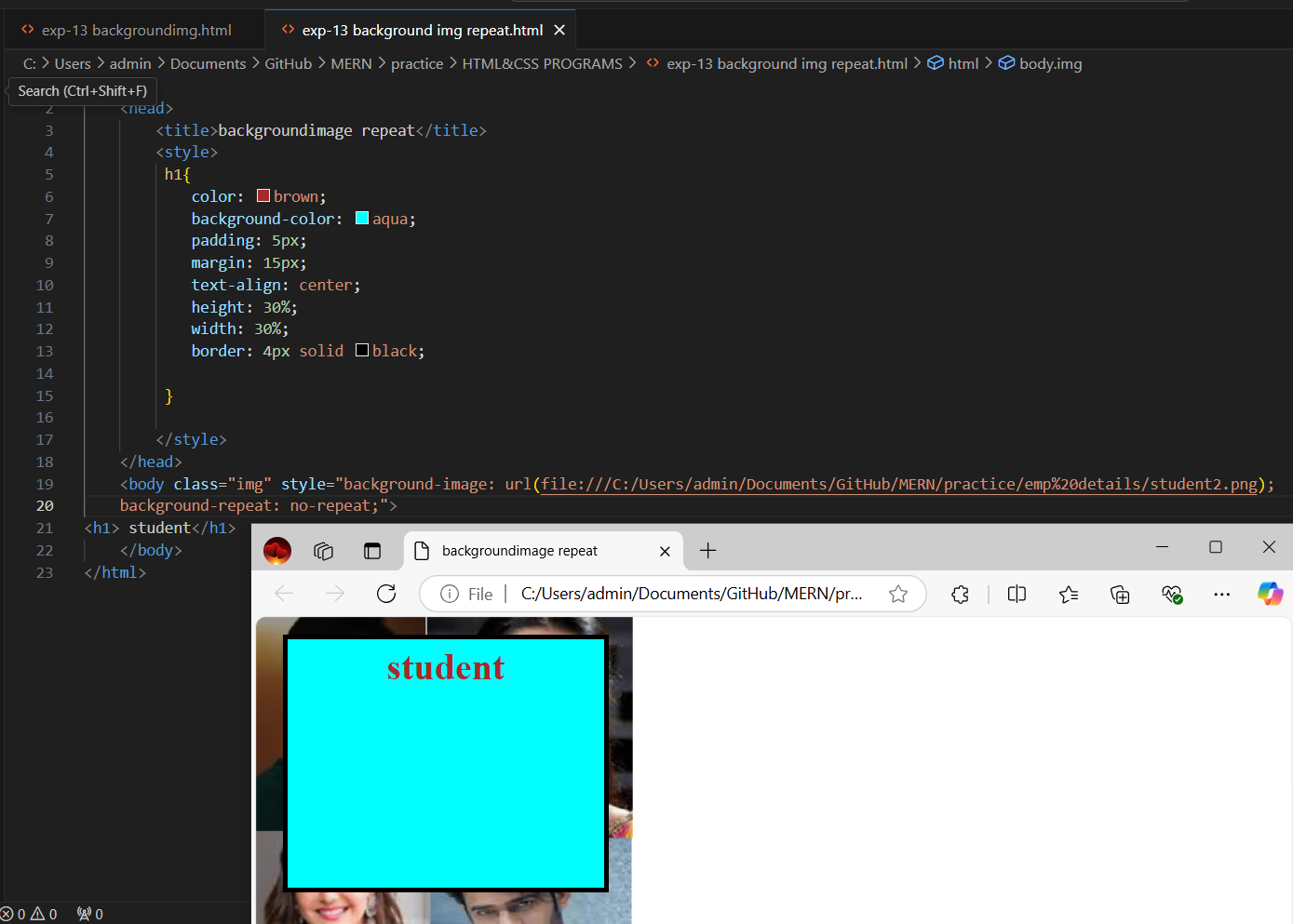
**BACKGROUND IMAGE-**A background image is specified for almost any HTML element. Use the html style attribute and the CSS background-image property.

**Ex-** background-image: URL(“ …. ”);

**>Background-repeat:** no-repeat; is used for the image not to repeat.



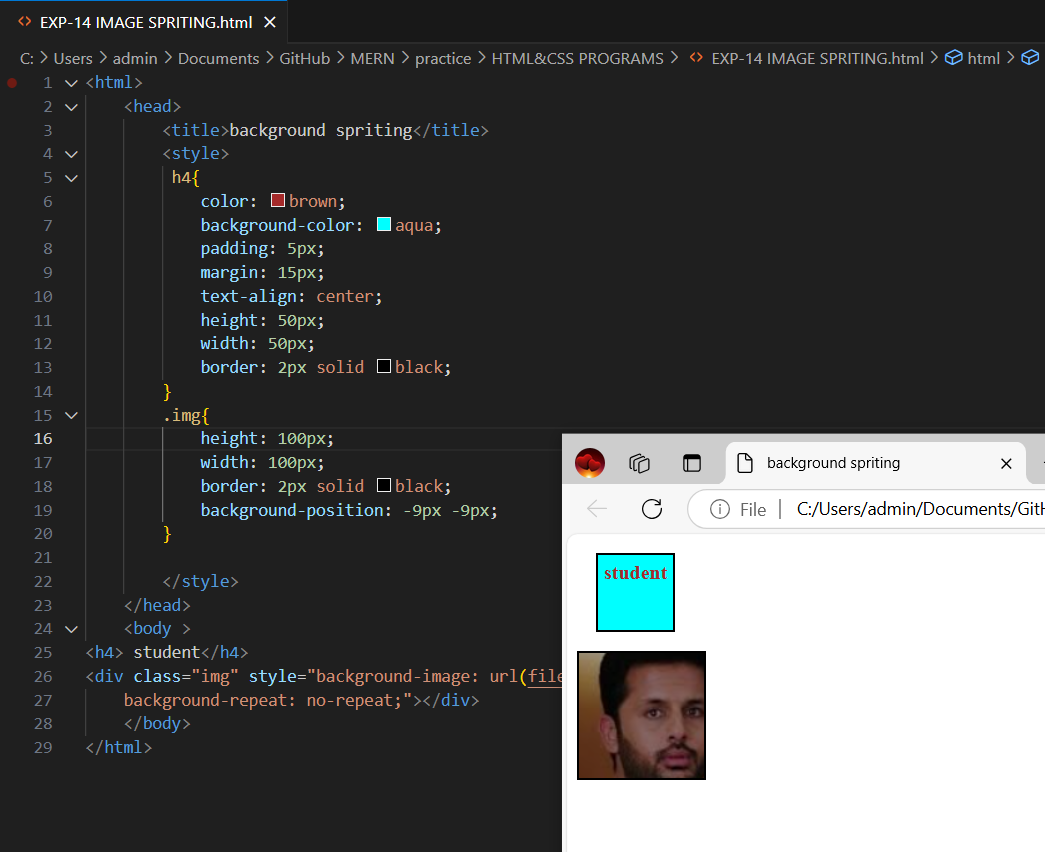
**IMG-BACKGROUND IMG**



**IMG-BACKGROUND IMG NO REPEAT**

**IMAGE SPRITING-** Through image spriting we increase the performance of the page by loading all the static images in a single column.

>Image spriting is only recommended for static web pages but not for dynamic web pages.



**IMG-IMAGE SPRITING**

**CSS POSITIONS:** In order to actually move the DOM elements to a required position without increasing its dimensions we use the following CSS properties(top, left, right, bottom).

>following the possible value a position attribute takes,

1. Static 2.Relative 3.Absolute 4.Fixed 5.Sticky

**1.STATIC-**The DOM element cannot be moved to any position from its default position.

>it will not consider the top , left, right, bottom properties.

**Ex-** position: static;

**2. RELATIVE:** It is capable of moving to any required position within the page. While moving to a new position it never loses space been occupied on load of the page. While moving to a new position it always moves relevant to its default position.

> it consider the top , left, right, bottom properties.

**Ex-** position: relative;

**3.ABSOLUTE-**It is capable of moving to any required position. While depending on the parent position it only depends on the parent whose position value is non-static.

>element with position absolute automatically jumps from default X-Y Axis to Z-Axis.

> it consider the top , left, right, bottom properties.

**Ex-** position: absolute;

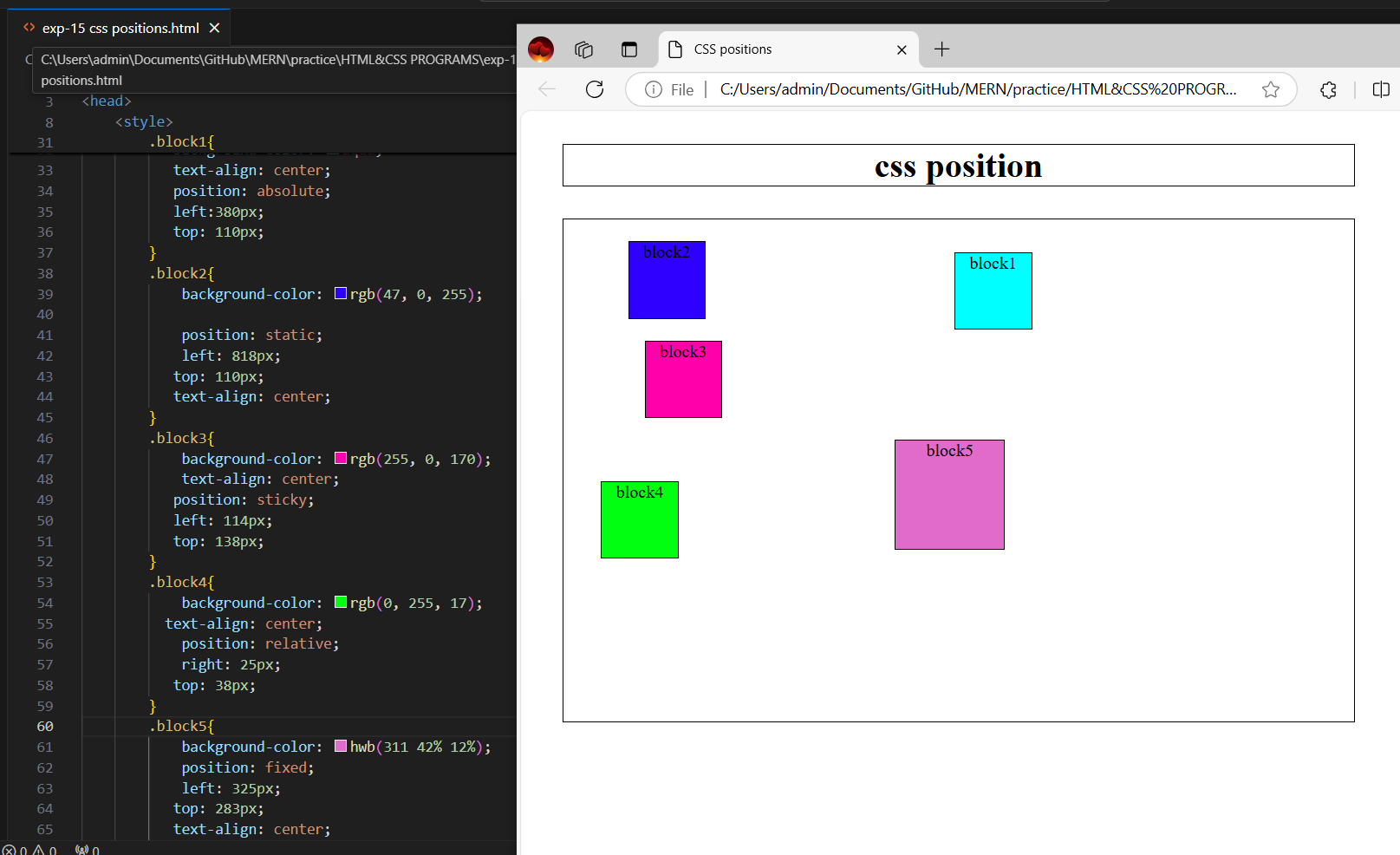
**4. FIXED:** Any DOM element with position fixed is almost like an element with position absolute. Where the only difference is once the element gets its fixed position it doesn’t move from its original position even when we scroll.

**Ex-** position: fixed;

**5.STICKY-**Any DOM element with position sticky is almost like an element with position relative, Where the only difference is once its position values been given ,if we try to scroll the element out of its view port, it automatically turns to fixed position and doesn’t get scrolled.

> it consider the top , left, right, bottom properties.

**Ex-** position: sticky;



**IMG-CSS POSITIONS**

**CSS Z-INDEX PROPERTY:** The element which are falling under Z-Axis there is a chance of multiple element override each other while rendering on the page.

>Z-Index property can only be applied to elements which fall under Z-Axis (element with position non-static).

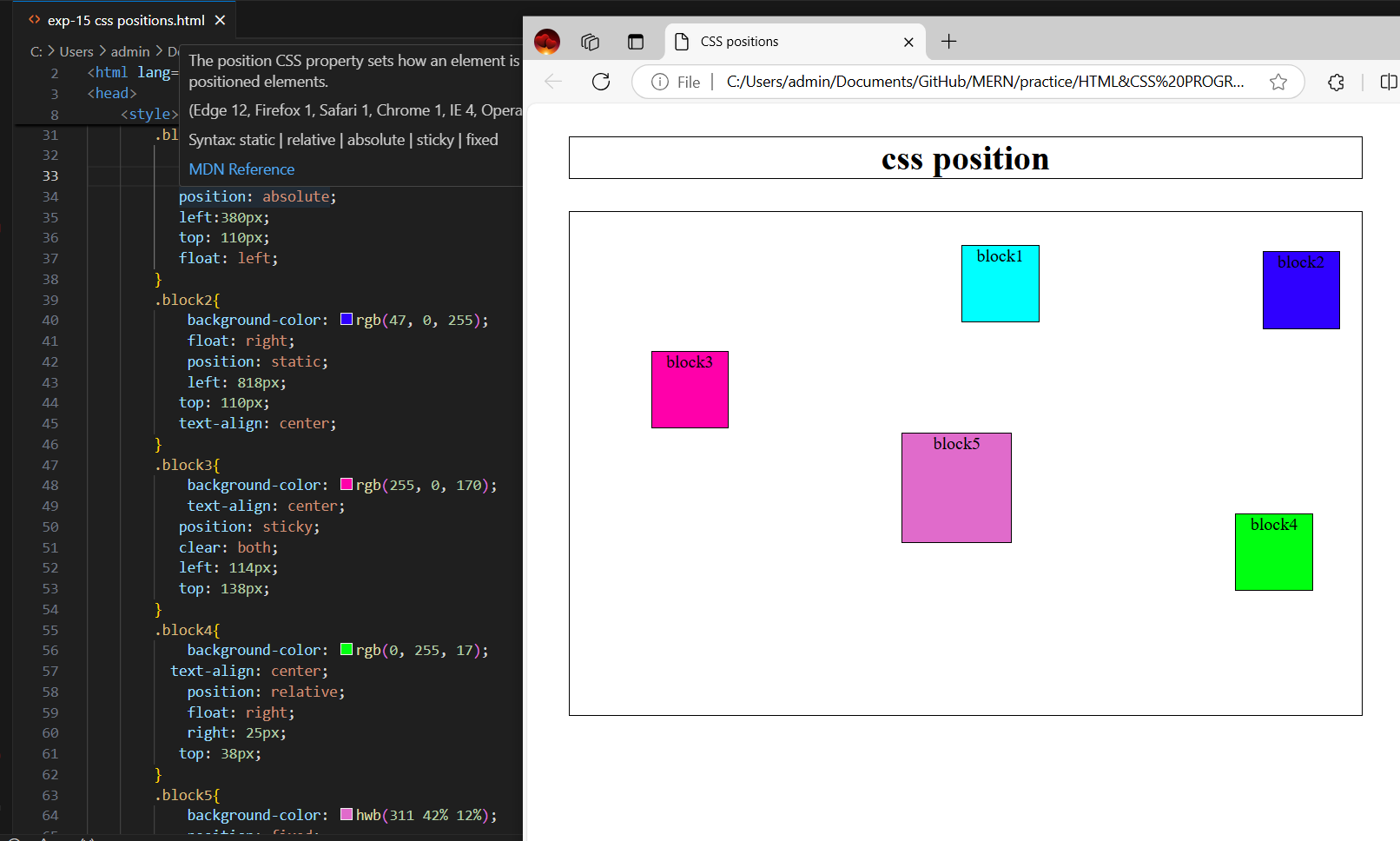
**Ex-** z-index : integer\_ value;

**CSS FLOAT PROPERTY:** Using CSS float property we can make the elements to the right side of the container or to the left side of the container .Following are the possible values of a float property takes,

**Ex-**float: left/right;

**>**To make the element not to follow previous element direction we make use of the CSS property clear.

**Ex-**clear: left/right/both;



**IMG-CSS FLOAT PROPERTY**

**HTML TABLES:** Following are predefined tags supported in html ,using which we could able to render any data in the form of row wise and column wise.

**1. <table>**holds the complete table content.

**2. < thead>**holds the table header.

**3. <tbody>**holds the body of the table.

**4. <th>**holds the table header cell.

**5. <tr>**holds the row of the table.

**6. <td>**holds the table data cell.

**7.** **<tfoot>**holds footer of the table.

**CSS DISPLAY PROPERTY:** Display is a CSS property through which we could able to change the default rendering type of any DOM element . following that the possible values it takes.

**>Display: Block;**

Makes the DOM element to render like a block level element .

**>Display: inline;**

Makes the DOM element to render like an inline element .

**>Display: inline-block;**

Makes the DOM element to render like a block element, but occupies in the same line as like in inline element

**>Display: none;**

Makes element to not to be shown on the page .It still exist in DOM structure.

**>Display: Flex;**

A CSS3 property to render flexible items on page.

>Both visibility hidden and display none properties makes element to be not visible on the page but the element still exists within the DOM structure.

**CSS PSEUDO CLASSES:** Following are the predefined pseudo classes been supported using which we could able to apply the CSS on elements not on load of the page but based on current state of the element.

**> :hover-** Applies set of CSS when there is a hover.

**> :empty-**Applied to any element which doesn’t have child element.

**>:disabled-**Applied to element with disabled state.

**>:enabled -** Applied to element with enabled state.

**>:active-**Selects only active link.

**>:checked**-Selects checkbox element with checked state.

**>:focus-**Select the element where it is in focus state.

**>:first-child –**Element which is in first child state.

**>:last –child-** Element which is in last child state.

>**Div: first-of-type –**To select every div element which is the first div of its parent.

>**a:link**-Selects unvisited links.

**>a:visited-**Selects visited links.

**>nth-child(2)-**Selects element in second position.

**>p:only-child-**Selects p tag which is only child of its elements.

**PSEUDO ELEMENTS-** Following are the predefined pseudo elements been supported using which we could able to apply CSS for not to the complete element but partially to the content of the element.

**>::after-**Add any element after the selected element.

**>::before-**Add any element before the selected element

**>::first-letter –**Only applies to the first letter of the container.

**>::first-line –**Only applies to the first line of the content.

**>::marker-**Only applies to set CSS for markers of the list items.

**>::selection-**Select the part/portion of the element content which is selected.

**CSS OPACITY PROPERTY:** Which elements get override each other we could able to control the transparency level of the elements through CSS opacity property.

>it takes a value between 0 to 1

**Ex**-opacity:0.2;

**HTML INPUT ELEMENTS-**Following are the pre-defined html element supported using which we could able to read different types of data from the user.

**<input type=”text”>(**to read text type of content)

**<input type=”password”>(**to read sensitive data)

**<input type=”date”>**

**<input type=”radio”>(**creates radio button)

**<input type=”checkbox”>(**creates check box)

**<input type=”button”>(**creates a button)

**<input type=”submit”>**

**Text area-(**creates a multiline text container )

**<select>**

**<option>option1</option>**

**<option>option2</option>**

**</select>(**creates a dropdown with multiple options)

**HTML Form Tag-** A predefined tag using which we could able to send user input data to the server.

>it makes the following mandatories i.e method and action.

**METHOD** attribute used to which we could specify the type of the communication while sending or receiving the data.

**ACTION** attribute used for through which we specify the path of the server to which communication should happen.

**Ex-<form action=”server url”metod=”get/post”>**

**……………………………………………………..**

**…………………………………………..**

**</form>**