Milestone-1 All Module Notes

Module_1_HTML

1-1 Module Introduction And First-Ever Website

- To save a file as HTML, we need to add a .html extension with that file.

1-2 Use Vs Code And Introduction To Html

- Open first_website.html with VS Code.

1-3 Getting Introduced With Paragraph Bold And Italic

- For creating a paragraph use ** ** Tag.
- For making a text bold, use **** Tag.
- For making a text italic, use **** Tag.

1-4 Different Types Of Heading And Strong Tag

- To create a Heading, We can use various heading tags according to the priority and importance of the heading. Heading tags are from h1 to h6. They are below.
    - **<h1></h1>**
    - **<h2></h2>**
    - **… … … …**
    - **… … … …**
    - **<h6></h6>**
- H1 is most important and big. H6 is less important and the smallest among them.
- We can use **** tag instead **** tag for bolding a pieces of text. **** indicate more importance than **** for a particular text.

1-5: HTML Tag Attribute, Anchor, Href, Navigate

- **Tag Attribute:** attribute indicate some property of a HTML Tag, Ex. href="" of an anchor tag (<a>)
- **Anchor-** Anchor tag express a relationship with other web page. Ex. Text
- **Href** - Hypertext Reference

1-6 Display Image Online Image, Local Image, Folder Image

- To display an image in web page, we use **<img>** tag. It is a self-closing tag or **empty tag** (single tag).
- We need to declare the source of the image by using the src="" attribute into IMG tag. **Ex <img src="example.jpg">**
- There is another attribute for the IMG tag, which is the alt="" attribute. It includes a description of the image for the screen reader or in some cases if the image failed to load, this text will display on the screen.

1-7 List, Container Tag, Ordered List, Unordered List

- For Expressing the unordered list, we use **
** tag as a parent and **
** as the child tag.
- For the ordered list, we use **
** tag as a parent and **
** as the child tag.

1-8 Button, Input Text Password, Div, Small, Br

- Button - **<button></button>** tag is usually used to make a button like Submit, Cancel, Log In etc.
- Input - **<input></input>** tag is used to take various type of input from user like **text, password, radio, checkbox, select** etc (google 'html button type' to know more button type').
- Div - ** ** tag is used to make some division into a web page.
- Small - **<small></small>** tag is used to make some text realy tiny.
- Br - **
** tag is used to create a break between two lines in a web page.
- Hr - **<hr>** tag is used to create a horizontal line in a web page. It has also some attributes like dotted, dash, etc (Google for more info).

1-9 Html Structure, Head, Body, Title, Meta Tag

- Html Structure - Every HTML page has a default structure. Like **Doctype, Head, Body,** etc. To create an Html Structure (many people say it **HTML boilerplate**) simply enter **'!'** and hit Tab or Enter key in VS code.
- Head - Contain some necessary heading information about the HTML page like **meta info, title, favicon,** etc.
- Title - **<title></tilte>** tag contain the title of the page (show in browser tab). Generally used inside head tag.
- Meta Tag - **<meta>** tag contain some hidden information about the page. It generally doesn't show in the browser directly. It plays an important role for SEO and some other aspect like charset declaration.

Module_2_Basic_CSS

2-1 Introduction To CSS Module, Getting Started With CSS

- **CSS** - Cascading Style Sheets
- Create pretty-css.html file to explore basic CSS.

2-2 Style Tag, Embedded Style, Named Color, Hexcode, Rgb

- **Style Tag** - Create - <style></style> tag inside 'head' tag to apply CSS inside the page's elements. This type of style is called **Emedded Style** (Internal CSS).
- To apply CSS on a tag; syntax would be: selector {property : value;}. Ex: p {color : red;}
- Several method of apply value of color property are **Named Color** (ex: red, green, blue), **Hexcode** (ex: #14ce14), **RGB** (ex: rgb(20, 206, 20)).

2-3 Background-Color, Height, Width, Font, CSS Measuring Units

- **background-color** is a CSS property (value can be the various color name) that specifies the color of the background of an HTML element (e.g. p, h1, etc). E.g. h1{background-color: green;}
- **Height & width** denotes horizontal and vertical size of an HTML element mentioned in **CSS Measuring Unites** (px, em, rem, %, etc). E.g. p{ width: 50%;}

2-4 CSS Id, Class, Apply Styles To Multiple Elements, Id Vs Class

- **CSS Id** is used to apply an unique style on a specific HTML element as an attribute inside tag (e.g.) and '#id_name' as selector. E.g. #usa{color:green;}
- **CSS Class** is somewhat similar to **CSS Id** either **Class** can be repeated and the selector is '.' (dot) instead of '#' (hash symbol). E.g. .usa{color:green;}
-

2-5 Style A Group Of Elements, Style A Small Portion Of Text

- To **Style A Group of Elements**, we can merge everything into a special HTML tag called **division** tag ( ).
- To **Style A Small Portion of Text,** inject it into **span** tag ()

2-6 Border, Border Radius, Margin, Different Ways To Set Margin

- **Border** specifies the area or the boundary of an element or a group of elements with various styles (e.g. solid, dashed, dotted, etc), thickness (e.g. 5px, 10px), and color. E.g. border: 5px solid green;
- The roundness of the corner of the **Border** describes by **Border Radius** (e.g. border-radius: 10px).
- **Margin** denotes the relative tarnsparent gap between two-element. **One way** of set margin is margin: 5px 7px 10px 15px (applied in clockwise - top, right, bottom, left)

2-7 Padding, Different Ways To Set Padding, CSS Box Model

- The space between border and content is called **Padding.** The value of **Padding** is work like margin.
- **CSS Box Model** is a concept of a box combination of **Content, Padding, Border, and Margin** where Content is innermost and margin is the outermost thing.

2-8 Text Align, Display, Inline, Block, Inline-Block, External CSS

- **text-align** specifies the alignment (e.g. center, justify, etc) of a **Block** element (e.g. paragraph, etc).
- Some HTML elements stick in a place, do not interfere with other elements of the same line is called the **Inline element** (e.g. span, a, img, etc). A few elements that block the hole line where they have are called **Block elements.** (e.g. p, div, etc). **Inline-Block** elements are those which look like **inline elements** but acts like **Block elements.**
- CSS **Fonts** has various kinds of property and value. Google for ‘CSS font property.

2-9 External Inline CSS File Display Inline And Block

- When we keep all of our CSS code into a separate file (e.g. main.css) and link that file (using <link> tag) with the corresponding HTML file is called **External CSS. Inline CSS** is applied (in some specific cases) to the style attribute inside a tag.
- We can change the **default inline element** to a **Block element** by changing the display property and vice versa.

Module_3_Git_Github

3-1 GitHub Module Introduction

- **GitHub** is such a platform where developers can safely store, access, modify, share their code.

3-2 Install git, create GitHub repository

- Go to GitHub and create a new repository. It is better to use the same name in the repository and local directory.
- To go backward one directory use, **'cd ..**', to go forward use, '**cd directoryName**' in CMD.

3-3 Introduction to Git init, git add, git commit

- Open project folder with VS Code, add test files, open VS Code Terminal,
- *Run* ***echo "# my-first-repo" >> README.md (optional)*** - to create READ.ME file.
- Run ***git init -*** to initialize
- Run ***git add .***- to get ready for the push.

3-4 Set origin, Git push, git pull, and repo overview

- To commit use ***git commit -m "first commit"*** Replace "first commit" with a statement related to the project.
- Specify branch -> ***git branch -M main***
- Add origin -> ***git remote add origin repositoryURL***
- To Push -> ***git push -u origin main***

3-5 Send small incremental changes to GitHub

- After make a little change in code, we need to run three command to update the changed.
- Add - ***git add .***
- Coomit - ***git commit -m "commit text"***
- Push - ***git push***

3-6 Host simple website in GitHub using gh-pages

- To publish a website **(Host)** using **gh,** go the corresponding repository -> Settings -> Pages : Under Source Select branch name to publish. Wait a bit and Follow the given URL to see the site. Anyone can see it by visiting that given URL.

3-7 Common GitHub related issues faced by new developer

- Don't forget to save any changed
- Commit every time before push
- To see the website hosted in gh, we need to keep main code into index.html

3-8 [Advanced] Create git branch, merge branches

- To check current branch - ***git branch*** [* mark branch is current branch]
- To create new branch - ***git branch [branch name]***
- Switch to a branch - ***git checkout [branch name]***
- Merge a branch into the active branch - ***git merge [branch name]***
- Go here to know more branch command - Click Here

3-9 [Advanced] git pull, toggle branch, merge conflict

- Don't work on main branch, make side (feature) branch and merge it after finish.
- Update local repository to the newest commit - ***git pull***
- Check not added change - ***git status***
- Create a new branch and switch to it - ***git checkout -b [branch name]***

Module_4_Build_a_portfolio_site [Assignment]

4-1 Module Introduction, project overview, and git setup

- Go to the developer-portfolio repository & see the live website.
- Create a new repo in GitHub name 'developer-portfolio' as well as a folder in projects directory
- Create an index.html & style.css and push the started code to 'developer-portfolio' repo.

4-2 Simple image background remove, set fancy background

- Use RemoveBg to remove image background online for free, and Photopea (like photoshop) to add fancy background to it

4-3 Set google fonts and install live server for auto refresh

- We are going to build a website like this as a sample
- Add heading using h1 (50px) (use Poppins Google Font regular version in the entire body)
- Add 'Live Server' Extension in VS Code

4-4 Header partial style by using span

- Keep partial part of **h1** into span tag (add a class) to apply different styles on it.
- Use a relatively small **header** tag (like **h3**) to add subtitles & a **p** tag to add a description or paragraph.
- Use **img** tag to add an image, don't forget to carefully specify the image path into the **src** attribute.

4-5 Put container, side by side, image size, Link button style

- Wrap everything up till now (from h1 to img) into **<section></section>** tag and apply **display : flex** on it.
- Separate two-row, by adding the same class to each row, (left row will include **h1, h3, p** & **a** & right row will have the **img** only)
- Apply some style on each row in such a way, they will look pretty similar to the sample site. (hints: use **width, padding, color, bg-img [as linear-gradient], border-radius**)

4-6 Dream area, Background image, background-repeat

- Add a class at top section & use **bg-img** with **no-repeat** & reset **body** margin to 0
- Push everything up till now to GitHub.
- Markup dream-section with img, h1, h2, p, & a (as button).
- Style as before (like top-banner section) including bg-img at the right position.

4-7 Experience area, refactor flexible container, meaningful class name

- Push one more time to GitHub
- Markup Experience section using **h2, h1, h3, & p** in 2 row.
- Add **flexible-container** & **half-width** as the design requirement.

4-8 Box shadow, background image, experience title change

- Apply style in experience section to make it similar to sample site (**Hints**: use **box-shadow, width, padding, margin, bg-img** etc)
- To know more about box-shadow Read this article.

4-9 Border image, footer, GitHub pages hosting, live link

- Apply **border-image** & **border-image-slice** in **linear-gradient** form (at experience item) to make it similar to the sample site.
- Read this to know more about the **border-image-slice**
- Push the final work to GitHub and make it live host.

Module_05_Build_Personal_Website [Brand New]
[First Assignment]

5-1 Five Things You Should Pay Attention About Assignments

- Have to be 100% honest for preparing Assignment.
- Try to complete the full assignment before 4:00 PM
- Deeply Focus on detailed requirements
- Don't ask help, anybody, for assignment
- Submit Assignment **Repository link** and **Live Site link** very carefully

5-2 Personal Website Assignment To Finish Milestone 1

- Go to ProgrammingHero1 and clone the food-netw_ork repo.
- To open .fig use figma.com
- Text and images can be changed by keeping the style the same.
- Change the lorem-ipsum text by something meaningful.
- Design all the buttons with _target="blank" attribute with some real link.
- Use border-radius and box-shadow at the My recipe section.

5-3 How To Submit Assignment

- ***Never click on the skip button on the assignment page to avoid 50% mark cut.***
- Submit, repository link and live website link in the following format
  Live website: [GitHub hosting link]
  Code Link: [Github Repository link]
- Don't share assignment marks at Facebook group.

Module_5.5_[Bonus_Module]
Box Model, Pseudo Class, Position

5_5-1 Understand inline, block, inline-block elements

- HTML tags, block full area of a line [wheather content need or not], margin, padding, height, & width can be applied are called **Block elements**. E.g. div, address, h1-h6, header etc.
- Some HTML elements that have the opposite characteristics of block elements are called **Inline elements**. E.g. span, img, a, etc.

- When inline elements act like block element are called **inline-block** elements.
- We can change the default state of a tag using the power of CSS.
- Have a look Emmet Cheat-sheet to know useful emmet shortcuts.

5_5-2 CSS box model, div vs span, border image slice

- **Box Model** is a concept of displaying HTML elements on the web browser which is useful to apply CSS.
- Div is a block element used to categorize HTML tags for a purpose. Span is an inline element use to style a specific part of a page.
- Slicing an image for using as the border is called **border image slice.**

5_5-3 Pseudo class hover, class hover, visited, focus

- Pseudo class is used to change the style of an element in some specific case. E.g. **hover** (e.g. on a button), **visited** (e.g. link), and **focus** (e.g. on input).
- For applying hover class, use **selector:pseudoClass** syntax.

5_5-4 First child, nth child, pseudo element before after

- The pseudo-class **first-child** is only affected on the first element from a list of the same element & the **last-child** affects the last one.
- **Nth-child** is used to affect specific elements, like even, odd, or something else.
- **Before** and **after pseudo-element** is used to enter some text or style, before or after an HTML element.
- Pseudo element sytax is **selector::pseudoElement**

5_5-5 Position static relative absolute fixed sticky z-index

- **Position** indicates the position of an element on the screen where static is the default value.
- **Relative** means the object (element) is now movable top, bottom, left, or right.
- **Absolute** means it can set any area of the screen depending on its **position: relative;** element (default is the **body** element)
- **Fixed** means it will be fixed at the top of the screen while scrolling.
- **Sticky** means it will be stick at a position (described) until all of its siblings will be scrolled
- **Z-index** sets the priority of displaying content on the screen. A higher z-index value will be at the top compared with a lower one. Z index only works on positioned elements (position:absolute , position:relative , or position:fixed).

Milestone-2, All Module Notes

Module_06_HTML_5_Semantic_Tags

6-1 HTML5 Audio, video, youtube video, embed iframe

- **<audio></audio>** tag is used to show audio file in web page. **Src** attribute declares the path of the audio and the **controls** attribute shows the controls of audio.
- **<video></video>** tag is used to show video. Src & controls attribute have the same job as audio.
- We can show embedded video from other sources using **<iframe></frame>** like Youtube Video, etc.
- We can set the **height** & **width** attribute to controls the displaying size of audio, video or iframe.

6-2 HTML5 Semantic tags section article header footer main

- Difference between HTML4 and HTML5 important for interviews.
- Emmet Shortcurt: **".container"** - To create a div with a class **container**, **".blog*4>h2{item-$}+p>lorem50"** - to create 4 div with class **blog** inside a **h2** [with item 1-4] and a p [inside lorem50 word] each.
- HTML5 Semantic Tags: **header** (wrapped the top part of a page including navigation), **nav** (for navigation), **section** (a section of a page), **article** (a blog post or newspaper post), **aside** (side navigation), **footer** (footer area including copyright), **main** (wrapped everything except **header & footer**)

6-3 HTML table, table head, table body, table row, table data

- All the table information will have into the **table** tag. Every table has **tr** (table row), **td** (table data), **thead** & **th** (table head), **tbody (**table body). **thead>tr>th** and **tbody>tr>td.**
- For border-collapse, use **border-collapse: collapse;**

6-4 Html form input label fieldset legend textarea submit

- **Alt+Shift+DownArrow** to duplicate a line.
- **form>input** for collecting various types of input from users like **type: text, type: email, type: password,** etc.
- For setting **radio button** into **fieldset** - **fieldset>legend**{legend text}**+p**(optional)**>label>input:radio. for & id** should be same for each input and **name** should be same for hole radio option.

- For setting **checkbox** into **fieldset** - **fieldset>legend**{legend text}**+p**(optional)**>label>input:checkbox. for, id & name** should be same as **radio button**
- **textarea** tag is used to set a large text field.

6-5 Simple navbar and internal navigation among files

- Keep all the navigation markup into **header>nav**
- Use **ul>li>a** to create a navigation menu.
- Use text-decoration (a), list-style (li), display(ul), padding (li), hover(li), etc to style the navigation.

6-6 Add nested menu dropdown option on mouseover

- Make another **ul.dropdown>li*5>a{link-$}** inside a navigation item.
- Make the dropdown (ul) **display: absolute;** (to brings it out of the flow) and d**isplay: none;**
- Make it visible (**display: block**) while hovering over the main navigation **li.**
- Apply some CSS to style it appropriately.

6-7 (advanced) Explore SVG, create a svg smiley face

- No need to understand it fully right now. To make a smily face using **svg** simply copy & past the following code.

```
<svg viewBox="0 0 200 200" style="border: 1px solid red;" width="400" height="400">
        <circle cx="100" cy="100" r="82" fill="yellow"></circle>
        <circle cx="65" cy="82" r="18" fill="black"></circle>
        <circle cx="130" cy="82" r="18" fill="black"></circle>
        <path d="M65 125 A1, 0.8 0 0, 0 130 125" stroke="black" stroke-width="10"></path>
    </svg>
```

- To know more about SVG tags Read these two article. Article One, Article Two

6-8 All html tags, everything you need to know about html

- We have not need to know about all of the HTML5 tags at this position.
- Must have a look at All HTML Tags to get an idea about all HTML tags. But remember, it is not necessary to memorize all of them.

6-9 All html attributes Title attribute, comments, alt, data

- Must have a look at All HTML Attributes, But no need to remember all of this.
- A comment is used in HTML or CSS to make notes for future reference. It is not visible by the user, only developers can watch it.

Module_7_More_CSS

7-1 CSS3 Module Introduction visibility hidden vs display none

- **visibility: hidden** element will be at the position, but it can't be seen.
- **display: none** elements will kick out from the page (it feels that there was no element there) and other elements will be adjusted in their place regarding the flow of content.
- **Interview Important:** Different between visibility hidden and display none

7-2 CSS Overflow visible, hidden, scroll, text-overflow, ellipsis

- The **overflow** property is used to fit a large text into a small area or box [e.g. **hidden, scroll,** etc]
- **Text-overflow (ellipsis)** and **white-space (nowrap)** are used to fit a text into a small box followed by three dots (...). **Tooltip** (**title** attribute) is used to show the full text.

7-3 Optimize images, types of images, SVG, png, jpg, Sprite

- Don't use extra-large heavyweight images on the webpage. Compress image using **Tinypng**.
- **SVG** image is very small in size compared to other formats.
- **CSS sprite** is a technique of showing a specific part (an object) of an image on the webpage using CSS.

7-4 Add icons using font awesome and create social links

- Go to FontAwesome, After sign in, copy the kit and paste it into the header tag.
- Search for icon and past the icon tag along with accurate class name into the web page.

- Style the icon as like a text.

7-5 Transform element using rotate scale translate

- **Transform** has several values. **scale(0.4)** indicates the size of the object, **translate** and r**otate(75deg)** move and rotate the object (element) by direction.

7-6 Introduction to CSS animation Transition

- **Transition** property change an element style by given direction.
- **transition: <property> <duration> <timing-function> <delay>;**
- E.g. transition: width 4s ease 2s;

7-7 Cricket match animation using transition and transform

- Take an image of a bat, rotate it on **hover** using **transform** and **transform-origin** (looking like a shot).
- Make a ball using **div** and **background color.** Set it at the front of the bat using **position** (relative).
- **Translate** the ball using **transform** and **transition** when the bat touches the ball.

7-8 (advanced) CSS animation and bounce effect with balls

- Use **@keyframes Animation_name {}** to set an animation & **Animation: name duration timing-function delay iteration-count direction fill-mode** to call the animation.
- E.g. **animation: slider 2s ease-in-out 1s infinite alternate both;**

7-9 Float left, float right, min-height, max-height, cursor

- The **CSS float** property specifies how an element should float (**left** or **right**)
- **min-height** (or width) and **max-height** (or width) specifies the maximum height (or width) and minimum height (or width) of an element (like div).
- The **cursor** changes the mouse pointer style depending on its value. E.g. **help, wait, crosshair, not-allowed, zoom-in, grab**.

Module_08_Responsive_CSS_Layout

8-1 Module Introduction and website layout

- **Website layout** defines a structural pattern of a website. A designer (UI) designs a structure (layout) of a website and a developer (front end)) convert it into code.
- To get a solid idea about website layout Read this article.

8-2 Flexbox, flex direction justify content align items

- **Display: flex;** The flex layout allows responsive elements within a container to be automatically arranged depending upon screen size.
- **Justify-content** specifies the content (elements) to be horizontally **center, space-between**, etc.
- **Align-items** specifies the content to be vertically **center, end** (flex-end), etc.
- **Flex-grow->**This property specifies how much of the remaining space in the flex container should be assigned to the item (the flex grow factor).

8-3 Create login form and flexible nav using flexbox

- To show all input tag like the block element, use **display: flex** & **flex-direction: column** to the form (parent of all input tags)

8-4 CSS Grid layout template columns grid gap

- **Important for Interview**: Different between **Flexbox & Grid**
- **Display: grid;** enable the power of CSS gird. **Grid-template-columns** specifies the number of columns, size of the column, etc. E.g. for 4 equal column: **repeat(4, 1fr), grid-template-rows** is for rows.
- Red some article on CSS Grid - Article-1, Article-2, Article-3

8-5 Create calendar using css grid and flex box

- Take 35 div for date & 7 columns for the day, divide them into 7 columns using the **grid,** apply some style for different month-date and day.

8-6 Explore Responsiveness using Media query and flexbox

- Media queries is a feature of CSS 3 allowing content rendering to adapt to different conditions such as screen resolution. It became a W3C recommended standard in June 2012, and is a cornerstone technology of responsive web design.[Wikipedia]
- To make a responsive design using flexbox, set **display: flex & flex-wrap: wrap;** [It will auto adapt]
- What is 'Media query' / How do you make a website responsive without any framework?

  - **important for Interview**

8-7 Create a Photo Album using relative position and flexbox

- Use multiple photos to make a photo album.
- Use **display: flex** & **flex-wrap: wrap** to the album container for automatic responsiveness.

8-8 Apply transition transform in your photo album

- We can zoom in a little every image on the mouse over (hover) by setting **transform: scale(1.5)** & given a **transition** effect.

8-9 All CSS properties, everything you need to know about CSS

- Have a look at List of CSS3 Properties to know about all CSS3 properties.

Module_8.5_CSS_Recap
Debugging, Futur Strategy

8_5-1 CSS specificity, style priority and !important

- CSS Rules are interpreted by the browser from top to bottom, left to right. For this reason, the bottom line overrides the top. But there is some exception for this rules. The priority order is below,
  -- **!important > inline-css > id > class > elements**

8_5-2 (advanced) CSS Custom property and use var, calc

- For using **CSS Custom property** (Variable), at first define it (at the top) using **:root {--property_name: property_value; }** and call it using **var(--property_name);**
- **Calk** is not frequently used at the age of **flexbox.**

8_5-3 CSS debugging using chrome devtool Elements tab

- To open **chrome dev-tool** -> Right-click and select inspect (Press - **F12 or Ctrl+Shift+i**)
- We can see, live edit, add, modify, etc all the HTML & CSS of the page at the **Element** tab.

8_5-4 Advanced selectors, airbnb css style guides, BEM

- To know about **CSS Advanced Selector Read this article.**
- To know about **BEM,** read this article: Article-1, Article-2
- To know about CSS style guides - Read AirBNB and CSS Guidelines

8_5-5 Your Future strategy for HTML(HTML5) and CSS(CSS3)

- Review all the notes from the beginning to this point (about HTML5 & CSS3)
- Review **HTML 5 Tags/Elements** and **CSS3 Properties**

Module_9_HTML_CSS_Only_Landing_Page

9-1 Module Introduction and Project resources

- Clone **leader-board,** open **leaderboard.fig** using **Figma**
- Create a repo as well as a local directory named '**leader-board-css3'**
- Move images into the project folder and create starter **index.html** & **styles/style.css**

9-2 Use grid and flex layout to align the players

- Keep all the **.player** (every .player included **img** & **h3** as title) into **.players** parent div.
- Make 3 column using **display:flex & grid-template-columns: repeat(3, 1fir)** into **.players**
- Bring the title beside the image and vertically center align by applying **display: flex** & **align-items: center** on **.player**.

9-3 Extract style information from figma and apply to players

- Insert section heading & style it, import google fonts (in this case **'Poppins'**) & use them in the **body** if it is the main font.
- Set **display: flex** & **justify-content: center** to the **main** tag to make the container center align.
- Apply **background color** to the **body** & **container**, **set padding**, **width** to the **container**.

9-4 Style debug to fix style, add horizontal line break

- Add the horizontal line (**hr**) and use **margin**, **padding** precisely between each and every element by checking and inspecting.

9-5 Create one blog using flex layout and style adjustment

- Markup top blog part (at least one blog) using proper class name according to their hierarchical position.
- Use **display: flex** to align blog **thumbnail** and **blog-info** at the same row.
- Apply some style to make it similar to the sample file.

9-6 Add multiple blogs and use grid layout to get two columns

- Add more blogs and fix some style issues.
- Add course container and vertically center align with the top container using **flex-direction: column** & **align-items: center** into **main.**

9-7 Create Single course and use font awesome ratings

- Markup course content like course-banner, course-title, rating icon (using font-awesome), etc.

9-8 Add all courses and set grid layout for three columns

- Properly align using the **grid, flex,** etc, and apply styles to make it similar to the sample file.

9-9 Set Media query to display blog in mobile devices

- Change the style using **media query** for at least 3 breakpoints to comfortably view the page on small, medium, and large devices.
- **Hints:** use grid, grid-template-column, flex, justify-content, align-item, etc.

Module_10_Assignment_2

10-1 Create a Responsive football website as assignment

- Clon **responsive-football** & open football.fig file using **Figma**
- Must have the **header-banner** section with icons, but menu is **optional.**
- Must have to **add any hover effect** on **header-banner**, like transition, transformation, animation, etc (maybe on the button or any other element into the header)
- Must have the **player section** with the same grid layout, use meaningful text with image.
- Highlighted Copa America section is **optional**.
- **Footer-section** is mandatory with proper icons & copyright symbols and text.
- Website must have to be **mobile responsive** as given. Try to do table responsive as well.
- Markup & styles should be organized using proper comments & indentation.
- For bonus marks - Add two extra relevant sections with this kind of site before footer, use proper fonts as well as font awesome, class name should be meaningful, code should be organized.

Module_10.5_Bonus_Module

10_5-1 review flexbox and grid for one more time

- **Flex** is flexible. It is used in some cases where the elements need to align horizontally or vertically and items can be flexible with the width ratio.
- **Grid** is used to create fix layout, like a fixed number of columns, etc.
- A complete Guide to Flexobx and A complete Guide to Grid can be helpful.

10_5-2 Check out relative and absolute position one more time

- The **relative** position keeps an element in its place but makes it movable against its own default position.
- The **absolute** position makes an element out of the default HTML flow against its parent's relative position. If its parent hasn't any relative position it moves against the body.

10_5-3 Explore media query one more time

- **Media query** is used to change the style for different screen size devices.

10_5-4 Easier way to create CSS animation using animate css

- We can use various CSS Animation Libraries to create awesome animation easily.