

# **Elite events**

## **AN INTERNSHIP REPORT**

**Submitted by**

**Patel SheelKumar Kirankumar**

**210800107009**

**In partial fulfilment for the award of degree of**

**BACHELOR OF ENGINEERING**

**In**

**Computer Engineering**

**Vadodara Institute of Engineering, Kotambi**



**Gujarat Technological University, Ahmedabad**

**[JULY, 2024]**



**Vadodara Institute of engineering  
Kotambi, Vadodara.**

**CERTIFICATE**

This is to certify that the project report submitted along with the project entitled **Elite events** has been carried out by **Patel Sheelkumar Kirankumar** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in **Computer Engineering**, 7<sup>th</sup> semester of Gujarat Technological University, Ahmedabad during the academic year 2024-25.

Prof. Rikta Gohil  
**(Internal Guide)**

Prof. Jemisha Patel  
**(Head of the Department)**

## **CERTIFICATE GENERATED FROM GTU**

## JOINING LETTER OF COMPANY



To

The HOD,

Vadodara Institute Of Engineering ,

Vadodara.

**Subject :- Web Design Internship**

We are pleased to confirm you that **Mr. Sheelkumar Patel.** has been appointed for web design **Internship** Program in **Sparks To Ideas** Session starts from 27<sup>th</sup> June 2024. We are confident he will be able to make a significant contribution to the success of our company and we look forward to work with him.

Sincerely,

Ashish Meghani

**Sparks To Ideas**

*Ashish*

Managing Director

**Sparks To Ideas**

---

Address: - 406 Akshat Tower near Pakwan hotel opposite rajpath club SG highway Ahmadabad.

E-Mail:- [info@sparkstoideas.com](mailto:info@sparkstoideas.com)

Website: - [www.sparkstoideas.com](http://www.sparkstoideas.com)

---

## COMPANY CERTIFICATE



### INTERNSHIP COMPLETION CERTIFICATE

**Subject :- Web Design Internship**

This is to certify that **Mr. Sheelkumar Patel** has successfully completed our **Web Design Internship** Program in **Sparks To Ideas** session Starting from 29<sup>th</sup> June 2024 to 12<sup>th</sup> July 2024. He completed his internship in Web Design technology. We wish him all the best wishes for his bright future.

We trust that your experiences during this internship have provided you with valuable insights, skills, and practical knowledge that will serve as a solid foundation for your future endeavors. We are confident that you will continue to excel in your chosen field.

**Sincerely,**  
**Ashish Meghani**

**Sparks To Ideas**

A handwritten signature in blue ink that appears to read "DSMB".

**Managing Director**  
**Sparks To Ideas**

[www.sparkstoideas.com](http://www.sparkstoideas.com)  
[info@sparkstoideas.com](mailto:info@sparkstoideas.com)

406 Akshat Tower near Pakwan hotel  
opposite rajpath club SG highway  
Ahmadabad



## Vadodara Institute of engineering

**Kotambi, Vadodara.**

### **DECLARATION**

We hereby declare that the Internship report submitted along with the Internship entitled Elite Events submitted in partial fulfilment for the degree of Bachelor of Engineering in Computer Engineering to Gujarat Technological University, Ahmedabad, is a Bonafede record of original project work carried out by us at **Sparks To Ideas** under the supervision of **Astha Metha** and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the student

---

Sign of student

---

## **ACKNOWLEDGEMENT**

I am very thankful for giving me the opportunity to undertake my internship training at the prestigious "Sparks To Ideas". It was a very good learning experience for me to have at this site.

First, I would like to thank the Head of The Department **Prof . Jemisha Patel** of Vadodara Institute of engineering for giving permission to commence this Internship.

Further, I would like to express my sincere gratitude to my Industrial Mentor **Ashta Metha** for continuously guiding me at the company and dispel all my doubts with patience. Also, I am very obliged to my Internal Mentor **Prof. Rikita Gohil** for helping me throughout my internship and giving me necessary suggestions and advice along with their valuable coordination. Without their continuous continuous support it would not have been possible to complete my internship.

I own my wholehearted thanks and appreciation to the entire staff of the company for their cooperation & assistance during my internship.

I also thank my parents, friends, and all the members of the family for their precious support encouragement which they had provided in the completion of my work.

Sheel Patel  
Computer Engineering  
210800107009

## **ABSTRACT**

*This report is a detailed overview of my internship journey at Sparks To Ideas. During my Internship I have learned a lot about industry. I have learned to work in an industry space which not only enriched me professionally but also helped me grow personally as well. My contribution was appreciated by my supervisor and other members of the department. The career path I would be selecting for myself is quite influenced from my internship as I have had a great opportunity to practically learn about how the company works. I have tried my level best to make it meaningful by reflecting on my works at the Sparks To Ideas. Also, I have summarized my overall experience, with my learning and challenges faced as an intern.*

## **List of Figures**

## **List of Contents**

## Table of Contents

Chapter No.	Title	Page No.
	Acknowledgement	i
	Abstract	ii
	List of Figures	iii
	List of Tables	iv
	Table of Contents	v
1	Company overview	1
1.1	Company History	1
1.2	Company Services	1
1.3	Company Contact	1
2	Introduction to HTML	2
2.1	Introduction	
3	CSS and its properties	
4	Introduction to Javascript	
5	Bootstrap Framework	
6	Introduction to React-Js	
7	Project Creation	
	Summary and Conclusion	
	References	

# CHAPTER1: COMPANY OVERVIEW

## 1.1 COMPANY HISTORY

It was started in 2015 and is an IT Services Company with expert professionals that provides strategic business solutions and customized software. It is a team of dedicated and highly skilled software professionals focused on providing world class IT solutions.

## 1.2 COMPANY SERVICES

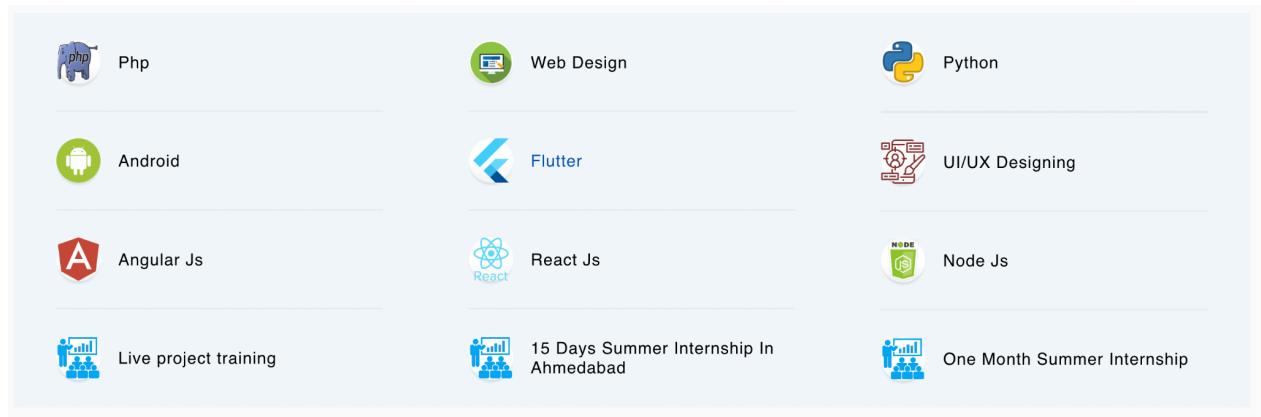


Fig 1.1 Company Services

## 1.3 COMPANY CONTACT

Address 1

406, Akshat Complex near pakwan hotel Opp rajpath Club, Sarkhej - Gandhinagar Highway,Ahmedabad,  
Gujarat 380054 (head office)

Address 2

55 Skillman Ave #4f, Jersey City, NJ 07306, USA

Phone

+91-9510203204

+91-8511777879

Email

info@sparkstoideas.com

# CHAPTER 2 : INTRODUCTION TO HTML

## 2.1 Introduction

HTML stands for **HyperText Markup Language**, is the standard markup language used to create web pages. It's a combination of Hypertext, which defines the link between web pages, and Markup language, which is used to define the text document within tags to structure web pages. This language is used to annotate text so that machines can understand and manipulate it accordingly. HTML is human-readable and uses tags to define what manipulation has to be done on the text. This guide will help you understand the workings of HTML and explain it with examples.

## 2.2 Structure of html

```
1  <!DOCTYPE html>
2
3  <head>
4  |  <title> INTRODUCTION TO HTML</title>
5  </head>
6  <body>
7  |  <p>My first paragraph</p>
8  </body>
9  </html>
```

Fig 2.1 Structure of html

The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document

The `<html>` element is the root element of an HTML page

The `<head>` element contains meta information about the HTML page

The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

The `<p>` element defines a paragraph

## 2.3 Heading tag

HTML headings are titles or subtitles that you want to display on a webpage. HTML headings are defined with the `<h1>` to `<h6>` tags. `<h1>` defines the most important heading. `<h6>` defines the least important heading

```
1  <!DOCTYPE html>
2
3  <head>
4      <title> INTRODUCTION TO HTML</title>
5  </head>
6  <body>
7      <h1>Heading 1</h1>
8      <h2>Heading 2</h2>
9      <h3>Heading 3</h3>
10     <h4>Heading 4</h4>
11     <h5>Heading 5</h5>
12     <h6>Heading 6</h6>
13 </body>
14 </html>
```

Fig2.2 Heading of HTML

## 2.4 HTML Table

In HTML, tables are used to organize and display data in rows and columns. They are structured using several tags and attributes to define the table structure, headings, rows, columns, and data cells.

**<table>**: This is the main element used to create a table in HTML. `border="1"`: Attribute to add a border to the table for visualization purposes. It's not recommended for styling; CSS should be used for that purpose.

**<tr>**: Stands for "table row" and is used to define a row in the table. Contains one or more `<td>` or `<th>` elements.

**<th>**: Table header cell. Used to define headings for columns or rows within the table. Typically used in the first row (`<tr>`) to define column headings.

**<td>**: Table data cell. Used to define regular data cells within the table. Used in subsequent rows (`<tr>`) to define data within each column.

**colspan and rowspan**: Attributes that allow cells to span multiple columns or rows, respectively.

## 2.5 HTML Form Tag

An HTML form is used to collect user input. The user input is most often sent to a server for processing.

Some of form tags are below:

**<form>**: It defines an HTML form to enter inputs by the user side.

**<input>**: It defines an input control.

**<textarea>**: It defines a multi-line input control.

**<label>**: It defines a label for an input element.

**<select>**: It defines a drop-down list.

**<option>**: It defines an option in a drop-down list.

**<button>**: It defines a clickable button.

## 2.6 Semantic tag and Non-Semantic tag

A semantic element clearly describes its meaning to both the browser and the developer. Examples of non-semantic elements: **<div>** and **<span>** - Tells nothing about its content. Examples of semantic elements: **<footer>**, **<heading>**, and **<article>** - Clearly defines its content.

### 2.6.1 Example of semantics tags:

**<header>**: Defines a header for a document or a section. Typically contains introductory content navigational links.

**<nav>**: Represents a section of navigation links for navigating within the document or related documents.

**<main>**: Specifies the main content of a document. Should not include content that is repeated across multiple pages, such as navigation links.

**<section>**: Defines a section in a document, such as chapters, headers, footers, or any other sections of the document.

**<article>**: Represents a self-contained composition that can independently exist and be reused in another context, such as a blog post or a news article.

**<aside>**: Defines content aside from the content it is placed in, often used for sidebars or tangentially related content.

**<footer>**: Represents a footer for a document or a section, typically containing information about the author, copyright data, links to related documents, etc.

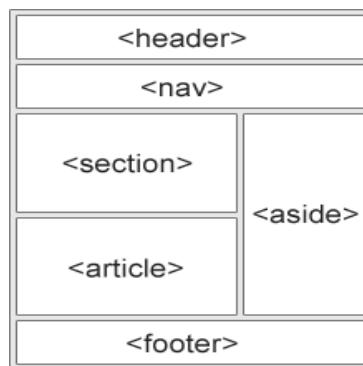


Fig2.3 Semantics tag

## 2.6.2 Example of non-semantics tags:

**<div>**: This tag is used as a generic container to group elements for styling purposes or to apply JavaScript behaviors. It does not provide any indication of the type of content it contains.

**<span>**: Similar to **<div>**, **<span>** is used for grouping inline elements and applying styles, but it does not convey any semantic meaning about the content within it.

## 1.7 Task of chapter 2

Problem: Make Application form for college admission using html

Solution:

**APPLICATION FORM**

---

**▼ Details**

**PERSONAL DETAIL:**

Ms./Mr./Miss./Mrs./Other:

Enter your full name:

Select your gender:  male  female

Date of birth:

Mother's name:

Father's name:

Mobile number:

Addhar.No:

Present Address :

---

**EDUCATIONAL DETAILS:**

class/degree	Name of school\Institute	Address and contact detail	Attended date from(mm/yy) to (mm/yy)	percentage	Board/University

---

**COURSE DETAIL:**  
course which have to study :

Mode of study  Online  Offline

---

**PAYMENT:**  
select your payment mode:

---

**SUPPORT REQUIREMENT:**

Please complete this section if you have any learning support/mobility requirements:

Do you have a disability, medical condition or require learning support?

If yes then give detail :

---

**DECLARATION:**

I certify that, to the best of my belief, the information I have provided is complete and true.

Signature of Applicant:

Date:

Time:

---

**INSTRUCTION:**

1. \* indicates mandatory  
2. Following document required (print):  

- Aadhar card
- 12th Marksheet
- 10th Marksheet
- ACPC merit page
- Subject result
- JEE result

Fig2.4 Application form

# CHAPTER 3 : CSS AND ITS PROPERTIES

## 3.1 CSS and its Types

CSS (Cascading Style Sheets) controls the visual presentation of web pages, including layout, colors, fonts, and animations.

### 3.1.1 Types of css

There are three types of css .

1) Inline css: Inline CSS involves applying styles directly to individual HTML elements using the style attribute. This method allows for specific styling of elements within the HTML document, overriding any external or internal styles.

```
1  <!DOCTYPE html>
2
3  <head>
4  |   <title> INTRODUCTION TO HTML</title>
5  </head>
6  <body>
7
8      <p style="color: #009900;
9          |   font-size:50px;
10         |   font-style:italic;
11         |   text-align:center;">
12          Inline Css
13      </p>
14
15  </body>
16  </html>
```

Fig3.1 Inline css

2) Internal css:Internal CSS is defined within the HTML document's <style> element. It applies styles to specific HTML elements, The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the <style> tag inside the head section of the HTML file.

```
<head>
    <title>Internal CSS</title>
    <style>
        .main {
            text-align: center;
        }

        .GFG {
            color: #009900;
            font-size: 50px;
            font-weight: bold;
        }
    </style>
</head>
```

Fig3.2 Internal css

3) External css: External CSS contains separate CSS files that contain only style properties with the help of tag attributes (For example class, id, heading, ... etc). CSS property is written in a separate file with a .css extension and should be linked to the HTML document using a **link** tag. It means that, for each element, style can be set only once and will be applied across web pages.

```
<!DOCTYPE html>
<html>
  <head>
    <title>External CSS</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div class="main">
      <div class="EFG">Types of css</div>
    </div>
  </body>
</html>
```

Fig3.3 External css

### 3.2 Types of Selector

CSS selectors are used to "find" (or select) the HTML elements you want to style. Mainly there are three types of selectors. They are as:

- 1) ID Selector: The id selector uses the id attribute of an HTML element to select a specific element.  
The id of an element is unique within a page, so the id selector is used to select one unique element.  
To select an element with a specific id, write a hash (#) character, followed by the id of the element.
- 2) Class Selectors: The class selector selects HTML elements with a specific class attribute. To select elements with a specific class, write a period (.) character, followed by the class name.
- 3) Element Selectors: Selects HTML elements based on their tag names.

### 3.3 Basic Properties of CSS

- 1) Background: Specifies the background color or image of an element.  
Background color: Sets the background color.  
Background-image: Sets the background image.
- 2) Color: Sets the text color.
- 3) Margin: Defines the space around an element.

- 4) Padding: Specifies the space between an element's content and border.
- 5) Typography: Controls the font properties. For e.g Font-family: Specifies the font family ,Font size: Sets the size of the font.Font-style: Applies italic or oblique style to the font.
- 6) Border: Sets the properties of an element's border . For e.gBorder-width: Specifies the width of the border ,Border-color: Sets the color of the border.
- 7) Width and Height: Sets the dimensions of an element.
- 8) Positioning: Positions an element relative to its containing element or the browser window.Position: Sets the positioning method (e.g., static, relative, absolute, fixed).
- 9) Float: Specifies whether an element should float to the left, right, or not.
- 10) Opacity: Specifies the transparency level of an element.Opacity: Sets the opacity of an element (0.0 to 1.0).

### 3.4 Task of chapter 3

Problem: Make Periodic table using html and css

Solution:

PERIODIC TABLE																		
1 H																		2 He
3 Li	4 Be																	
11 Na	12 Mg																	
19 K	20 Ca	21 Sc	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As	34 Se	35 Br	36 Kr	
37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe	
55 Cs	56 Ba	71 Lu	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Ti	82 Pb	83 Bi	84 Po	85 At	86 Rn	
87 Fr	88 Ra	103 Lr	104 Rf	105 Db	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Nh	114 Fi	115 Mc	116 Lv	117 Ts	118 Og	
57 La	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm	70 Yb					
89 Ac	90 Th	91 Pa	92 U	93 Np	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No					

Fig3.4 Periodic table using html and css

# CHAPTER 4 : INTRODUCTION TO JAVASCRIPT

## 4.1 What is Javascript ?

JavaScript is the Programming Language for the Web. JavaScript is both an imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

## 4.2 Types of Javascript:

There are two types of javascript. These are:

- 1) Internal Javascript :You can add JavaScript code in an HTML document by employing the dedicated HTML tag <script> that wraps around JavaScript code.The <script> tag can be placed in the <head> section of your HTML or in the <body> section, depending on when you want the JavaScript to load.

```
<!DOCTYPE html>
<html lang="en-US">
|<head>
| |<meta charset="UTF-8">
| |<meta name="viewport" content="width=device-width, initial-scale=1">
| |<title>Today's Date</title>
| |<script>
| | |let d = new Date();
| | |alert("Today's date is " + d);
| |</script>
|</head>
|<body>
|</body>
|</html>
```

Fig4.1 Internal Javascript

- 2) External Javascript:In order to accommodate larger scripts or scripts that will be used across several pages, JavaScript code generally lives in one or more js files that are referenced within HTML documents, similar to how external assets like CSS are referenced.

```
<!DOCTYPE html>
<html lang="en-US">

|<head>
| |<meta charset="UTF-8">
| |<meta name="viewport" content="width=device-width, initial-scale=1">
| |<title>Today's Date</title>
|</head>

|<body>
| |<script src="js/script.js"></script>
|</body>

|</html>
```

Fig4.2 External Javascript

## 4.3 Javascript Variables

In JavaScript Variables can be declared in 3 ways:

- 1) Using var :The scope of a var variable is functional or global scope. It can be updated and re-declared in the same scope
- 2) Using let :The scope of a let variable is block scope. It can be updated but cannot be redeclared in the same scope.
- 3) Using const: The scope of a const variable is block scope. It can neither be updated or re-declared in any scope.

## 4.4 Pop-up Boxes

JavaScript has three kinds of popup boxes: Alert box, Confirm box, and Prompt box.

- 1) Alert box: An alert box is often used if you want to make sure information comes through to the users.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax: `alert("I am an alert box!");`

- 2) Confirm box: A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed. If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

Syntax: `confirm("Press a button!");`

- 3) Prompt box :A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after enter an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null.

Syntax: `prompt("Please enter your name");`

## 4.5 Javascript DOM (Document object model)

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the document structure, style, and content. The important methods of document object are as follows:

Method	Description
<code>getElementById()</code>	returns the element having the given id value.
<code>getElementsByName()</code>	returns all the elements having the given name value.
<code>getElementsByTagName()</code>	returns all the elements having the given tag name.
<code>getElementsByClassName()</code>	returns all the elements having the given class name.

```

<!DOCTYPE html>
<html>

<body>

    <p id="intro">
        A Computer Science portal for geeks.
    </p>
    <p>
        This example illustrates the <b>getElementById</b> method.
    </p>
    <p id="demo"></p>
    <script>
        const element = document.getElementById("intro");
        document.getElementById("demo").innerHTML =
            "DOM introduction is: " + element.innerHTML;
    </script>
</body>

</html>

```

Fig4.3 Document Object Model

## 4.6 Event Handling

JavaScript Events are actions or occurrences that happen in the browser. They can be triggered by various user interactions or by the browser itself. Some common events are as follows

Event Attribute	Description
onclick	Triggered when an element is clicked.
onmouseover	Fired when the mouse pointer moves over an element.
onmouseout	Occurs when the mouse pointer leaves an element.
onkeydown	Fired when a key is pressed down.
onkeyup	Fired when a key is released.
onchange	Triggered when the value of an input element changes.
onload	Occurs when a page has finished loading.
onsubmit	Fired when a form is submitted.
onfocus	Occurs when an element gets focus.
onblur	Fired when an element loses focus.

```

1  <html>
2  <head>
3  |   <script>
4  |       function There() {
5  |           alert('Hi there!');
6  |       }
7  |   </script>
8  </head>
9  <body>
10 |     <button type="button"
11 |         onclick="There()"
12 |         style="margin-left: 10%;"
13 |             Click me
14 |     </button>
15 </body>
16
17 </html>

```

Fig4.4 Document Object Model

#### 4.7 Task of Chapter 4

Problem: Make BMI calculator using html,css, and javascript

Solution:

The screenshot shows a web-based BMI calculator. At the top, it says "Body Mass Index Calculator". Below that is a form with a blue border. Inside the form, there are two input fields: "Age" (set to 55) and "Gender" (set to "Female", indicated by a checked checkbox). Below the gender field are two more input fields: "Height (cm)" (set to 67) and "Weight (kg)" (set to 80). A blue "Submit" button is located below these fields. At the bottom of the page, the results are displayed: "Extremely obese" and "BMI: 248.72".

Fig3.5 BMI calculator

# CHAPTER 5 : BOOTSTRAP FRAMEWORK

## 5.1 Introduction to Bootstrap

Bootstrap is the most popular CSS Framework for developing responsive and mobile-first websites.

Bootstrap 5 is the newest version of Bootstrap .

### 5.1.1 Installation of Bootstrap

There are two ways in which we can install Bootstrap. These are:

Method 1(Compiled CSS and JS): This method of installing bootstrap is also easy but it can work offline ( doesn't require an internet connection ) but it might not work for some browsers.

Method 2 (BootstrapCDN): This method of installing Bootstrap is fairly easy but it requires Internet connection. It is highly recommended that you follow this method.

## 5.2 Bootstrap Grid System

Bootstrap Grid System allows up to 12 columns across the page. You can use each of them merge or individually or merge them together for wider columns. You can use all combinations of values summing up to 12. You can use 12 columns each of width 1, or use 4 columns each of width 3 or any other combination.

Grid Classes :The Bootstrap grid system has five classes that can be combined to make more layouts they areas follows :

xs (<576px): For Portrait Mobile Phones.

sm (>=576px): For Landscapes phones

md (>=768px): For Tablets/Phablets

lg (>=992px): For Small-sized Desktops/Laptops

xl (>=1200px): For Larger-sized Desktops/Laptops

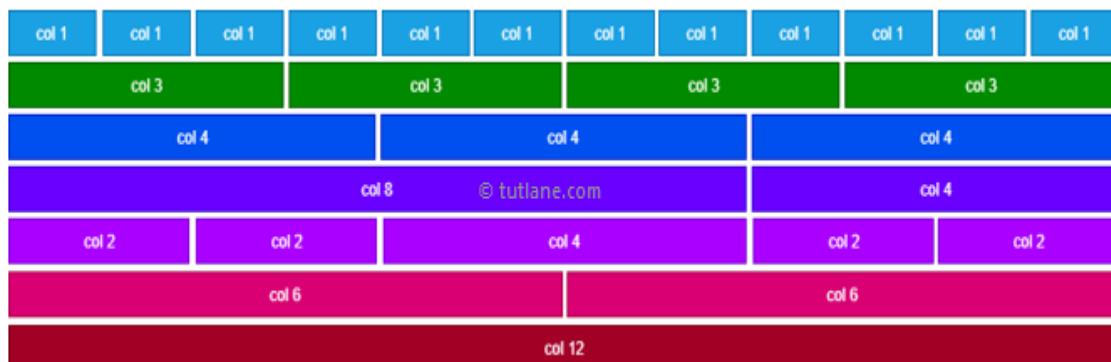


Fig5.1 Bootstrap Grid System

# CHAPTER 6 : INTRODUCTION TO REACT-JS

## 6.1 Introduction to React-js

It is an open-source JavaScript library, which is used to build user interfaces (UIs) and front-end web app or applications. It provides us with many features to build high performance web applications, whether it is small, big or complex. ReactJs is a highly optimized and component based JavaScript library, which is particularly popular for single-page applications (SPAs) and allows developers to create interactive and dynamic UI components.

### 6.1.1 Prerequisites to install React App

Before installation of React App you have to install Node.js latest version in your system. Because React is a JavaScript library, and node.js provides a runtime environment for executing JavaScript outside the browser. Also you can use npm and install and manage dependencies in React projects, once you have have installed Node.js in your system.

If you have not installed Node.js yet, then let's see how to install it and which version to install. First go to the official website of Node.js <https://nodejs.org>.

There on the home page, you will see two versions available for download, as you can see in the screenshot below.

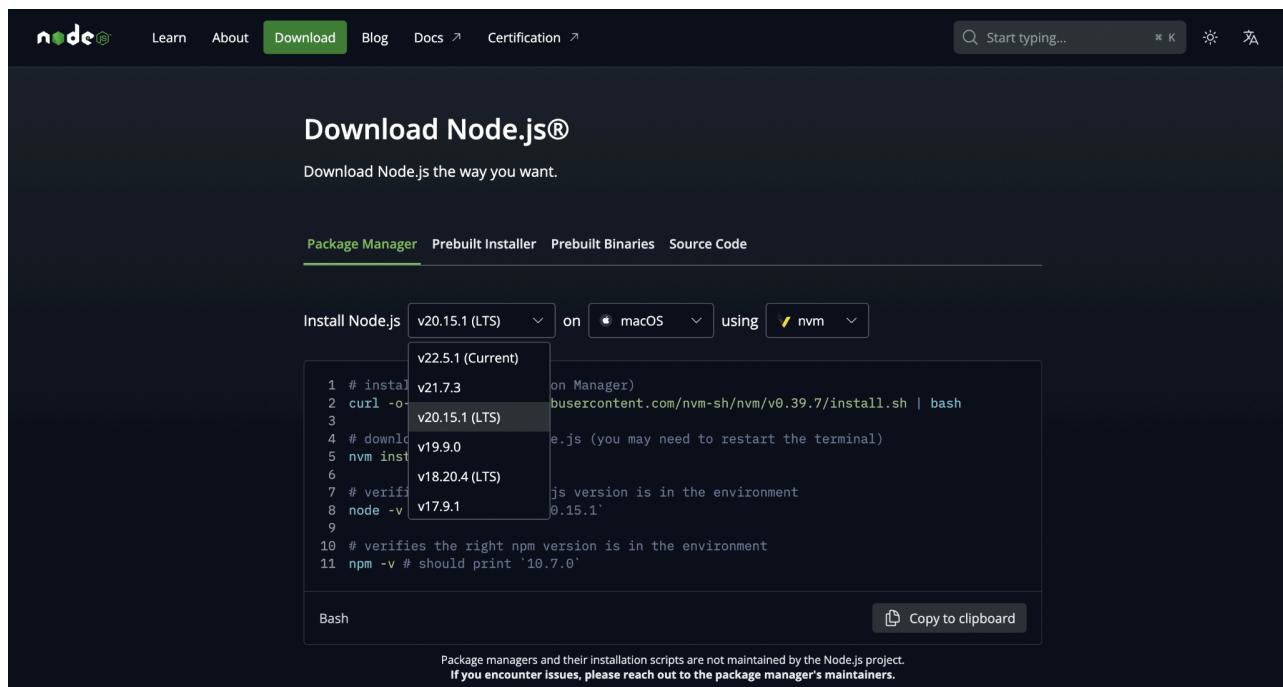


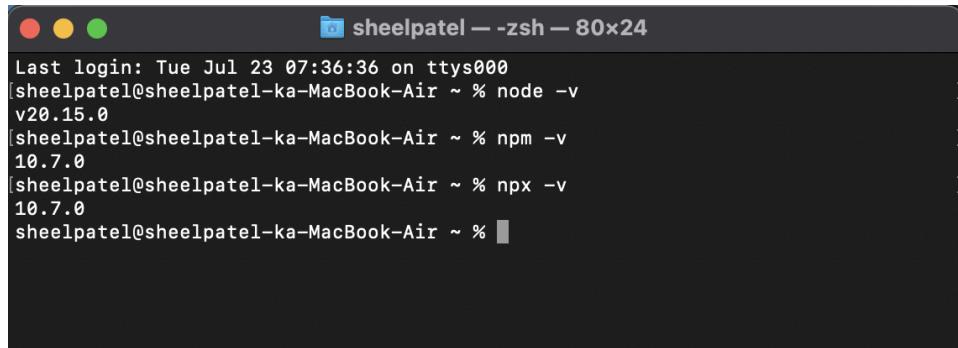
Fig 6.1 Installation of node.js

Download the **v20.15.1 LTS** version, as it is recommended for most users and it's widely used by developers for production environments. After the installer is downloaded. Install it by launching installation wizard and following all instructions of installation wizard, accepting license agreement and choosing the default installation settings.

After the installation, open the Command Prompt or PowerShell or Terminal (If you use macOS) then run the command `node -v` to check if Node.js is successfully installed.

Also run the command `npm -v` to check if the npm is successfully installed.

If you see the Node.js version displayed, then congratulations it's successfully installed.



```
Last login: Tue Jul 23 07:36:36 on ttys000
[sheelpatel@sheelpatel-ka-MacBook-Air ~ % node -v
v20.15.0
[sheelpatel@sheelpatel-ka-MacBook-Air ~ % npm -v
10.7.0
[sheelpatel@sheelpatel-ka-MacBook-Air ~ % npx -v
10.7.0
sheelpatel@sheelpatel-ka-MacBook-Air ~ % ]
```

Fig 6.2 To check Installation of node.js

## 6.2 Set up node js

ReactJs has a feature that you can install your react app by using `create-react-app`

### Installation method 1

Open your terminal in the directory you would like to create your application. Then run this command:

```
npx create-react-app my-react-app
```

In place of the “`my-react-app`” in this command line, you can replace and write the name of your app. Then you can move the `my-react-app` directory by using this command:

```
cd my-react-app .
```

### Installation method 2

There is an another way to create your react app directly without specifying its name in the command line, like this:

```
npx create-react-app .
```

In this case, first go to your directory and create a new folder with the name of your app. Then enter that folder and open your terminal and run the above command.

## 6.3 File Structure

After successfully completing the React app installation, we get the default folder structure of React.

It's something looks like this:

Let's understand the whole folder structure of our React App here.

```

my-react-app/
    ├── node_modules/
    ├── public/
    │   ├── index.html
    │   ├── favicon.ico
    │   └── ...other static assets like (Images, txt & json files, etc.)
    ├── src/
    │   ├── index.js
    │   ├── App.js
    │   ├── index.css
    │   ├── App.css
    │   ├── components/
    │   │   ├── Component1.js
    │   │   └── Component2.js
    │   ├── assets/
    │   │   └── ...other assets like images, fonts, etc.
    │   └── ...other application files
    ├── package.json
    ├── package-lock.json
    └── .gitignore
        ...other configuration files

```

Fig 6.3 File structure

- **node\_modules:** This folder is automatically generated. It contains project dependencies. You don't need to edit this folder, it is managed by npm.
- **public:** This folder contains static files/assets like images, logos, robots.txt, other json files, and the main *index.html* for the application. Here the *index.html* file loads our react app and renders it on the browser.
- **src:** The src folder is known as the source folder. It contains the main source code of the react app.
- **index.js:** Inside the src folder there is an index.js file. This is the entry point of the React app, where the root component (*App.js*) is rendered into the DOM.
- **App.js:** This is the root component of the React app.
- **Index.css:** This CSS file is used to set default style of overall layout like default font-styles, margins, paddings, etc. You can create CSS variables here.
- **App.css:** This CSS file is used for our root component *App.js*.
- **components:** You can create a components folder inside the src folder . Here we will create reusable react components, which we use throughout the React App. We will discuss this in detail in the React Components chapter.
- **package.json:** This is the configuration and dependencies file. This file contains important metadata about the project, including its name, version, dependencies, scripts, and more.
- **package-lock.json:** It's automatically generated by npm for package version consistency. You don't need to edit anything in this file.
- **.gitignore:** Specifies files and directories that should be ignored by Git version control.
- **README.md:** Project documentation, providing an overview of the project and instructions on how to run it.

### 6.3.1 Run the React App

It's time to run our React Application. So you can run the app by invoking the start script configured in the package.json file. Use this command:

```
npm start
```

The screenshot shows the VS Code interface with the 'TERMINAL' tab selected. The terminal window displays the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Compiled successfully!

You can now view demo in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.43.20:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

Fig 6.4 running of react-app

Then it will start the application in the local system and a new browser window will automatically pop up with <http://localhost:3000>

If the browser does not pop up automatically, then open your favorite browser and go to <http://localhost:3000/>

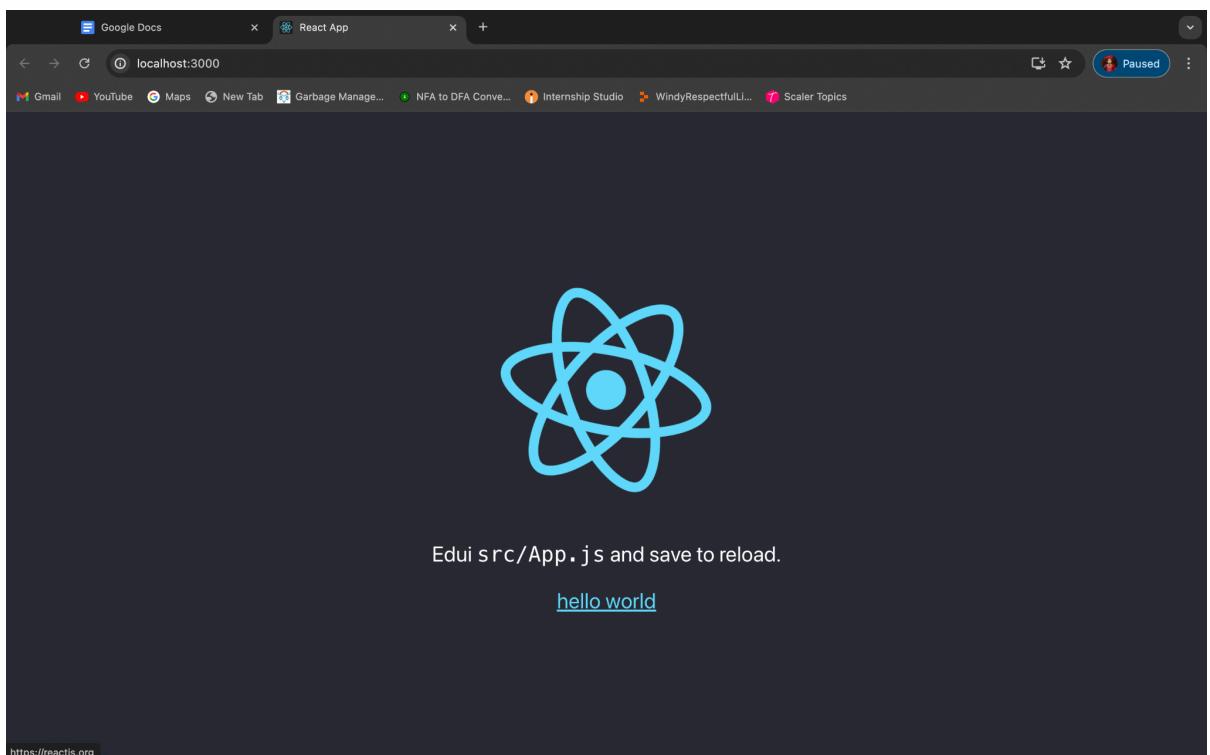


Fig 6.5 output in browser

## 6.4 React JSX

JSX stands for JavaScript XML. It is not a different language, it's a syntax extension used in React to define & describe the structure of React elements and makes it easier to write and add HTML in React. It allows you to write HTML elements directly in your JavaScript code.

Here is an example of JSX: Look, Using JSX we can write HTML syntax directly like this.

```
import React from 'react';

function App() {
  return (
    <h1>Hello World!</h1>
  );
}

export default App;
```

Fig 6.6 React JSX

### 6.4.1 JSX Embedding Expressions

You can write JavaScript expressions inside the curly braces {} in JSX. Here is some code examples of JSX expression given below for better understanding:

Variables:

```
function App() {
  const name = "Code With Random"
  return (
    <h1>Hello, {name}</h1>
  );
}

export default App;

// output
// Hello, Code With Random
```

Fig 6.7 JSX variables

Mathematical Operations:

```
function App() {
  return (
    <h1>{10 + 5}</h1>
  );
}

export default App;

// output
// 15
```

Fig 6.8 JSX Mathematical operation

Ternary Operators:

```
function App() {
  const age = 22;
  return (
    <h1>{age >= 18 ? "You are a Adult" : "You are not a adult!"}</h1>
  );
}

export default App;

// output
// "You are a Adult"
```

Fig 6.9 JSX ternary operators

## 6.5 React Components

What is a React Component?

As I said, React is a component based Javascript library. So the components are core and fundamental building blocks of React. If we say in a simple way, a component is a piece of code that can be reused anywhere throughout the react application.

Components make the task of building UIs much easier and leads to better code organization, reusability, and maintainability.

Let's understand the components more better way:

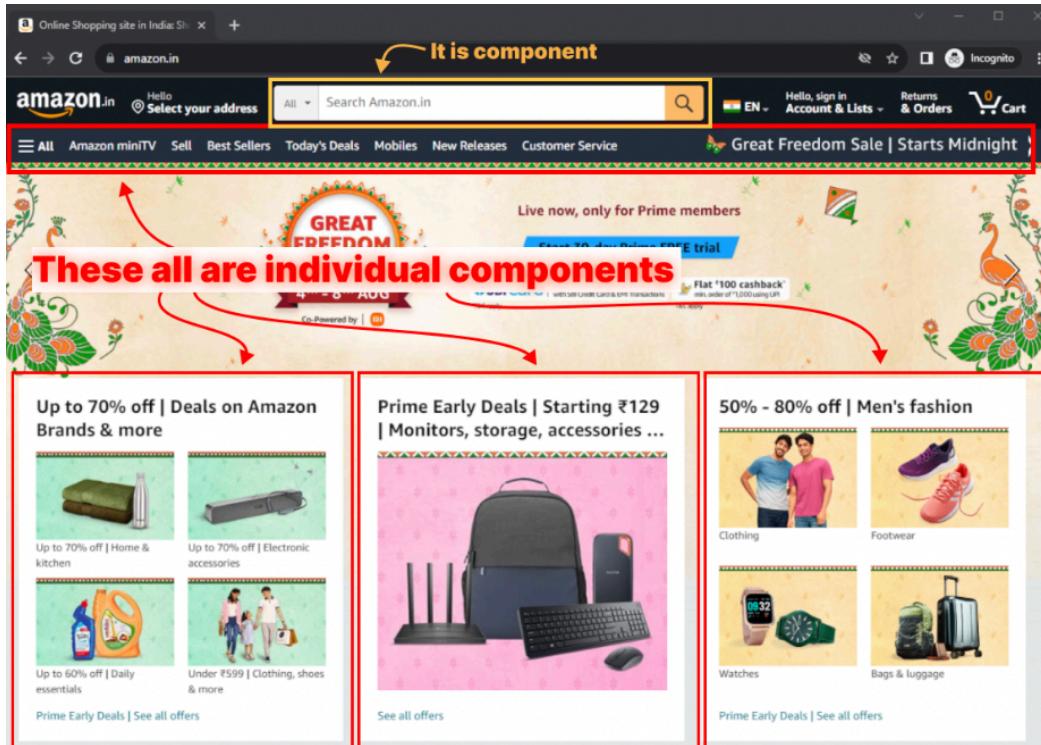


Fig 6.10 React components

You can see in the above image,I have taken a screenshot of Amazon's Home page UI. In that UI the items I have marked with red and yellow color, these are all the components.The search bar on the top is an individual component, the header is a component. Then you can see these three cards, these are not individual components, here a single component rendered 3 times with different data, and these data are dynamically rendered by using react props. This is the importance of components in React.We will discuss components with props further but for now Let's know the types of React Components.

### 6.5.1 Types of React Components

In React, there are mainly two types of components,Class components and Functional components.

#### Functional Components

Functional components are also known as stateless components. These are simply javascript functions and made with simple JSX code. These components are simple to use and easy to understand.

Syntax:

```
import React from 'react'

function User() {
  return (
    <h1>Functional components </h1>
  )
}

export default User;
```

#### Class Components

Class based Components are little different and complex than the Functional Components. Class components are stateful and when react needs to maintain its state,class components can be employed.We can pass data from one Class component to another. These are widely used in many complex projects.

Syntax:

```
import React, {Component} from 'react'
class Componentname extends Component
  render() {
    return (
      <h1>Welcome Text!</h1>;
    )
  }
}

export default Componentname;
```

## 6.6 React Fragment

In React, fragments allow you to return multiple elements in a component without wrapping them in a parent container.

Why use React Fragment?

In React when you render multiple elements it will require a 'div' tag around the content as this will only render a single root node inside it at a time.

Syntax :

```
import React from 'react'

function User() {
  return (
    <>
      <p>Hello</p>
      <p>User</p>
    </>
  )
}

export default User;
```

## 6.7 React Styling

In React, there are several ways to apply CSS styling to the React Components. Generally we can add css styles to the React components by using CSS classes through className attributes.

We will learn here the two most common ways of CSS styling in React:

- Inline styling
- CSS stylesheets

### 6.7.1 Inline styling

As you learned in the JSX chapter, JavaScript expressions are written inside curly braces in JSX, and since JavaScript objects also use curly braces, so when you use inline CSS styles in JSX, you have to use double curly braces (`{{}}`). For example:

```
import React from 'react'

function User() {
  return (
    <h1 style={{color: "red", backgroundColor: "blue"}}>Hello Style!</h1>
  )
}

export default User;
```

## 6.7.2 CSS stylesheets

Writing CSS in a stylesheet is probably the most common and basic approach to styling a React application. It's very easy to use and understand. You can use regular CSS stylesheets to style your components in React. Using CSS stylesheets in React is similar to using CSS stylesheets in traditional web development, but there are few differences between them.

Let's talk about How to use CSS stylesheets in React?

- First create a CSS file inside the 'src' folder, and give its name whatever is your choice but when you create a CSS file for a specific component, then i recommend to give its name the same as that component name. For example: if I have a component named User.js then I will create a CSS file for it named 'User.css' .
- Then import that CSS file into the Components. For example:

```
import React from 'react'  
import "../User.css"  
function User() {  
  return (  
    <h1 className='Heading'>Hello Style!</h1>  
  )  
}  
export default User;
```

## **CHAPTER 7 : Project Creation**











