Training report on Frontend Development

A Report submitted in partial fulfilment of the requirement

for the award of degree of

Bachelor of Technology

In

Electronics and Communication Engineering

Submitted by

VANSHIKA KABRA
(04896302820)

Submitted To

Dr. Neetu Sehrawat
(ECE Dept MSIT)

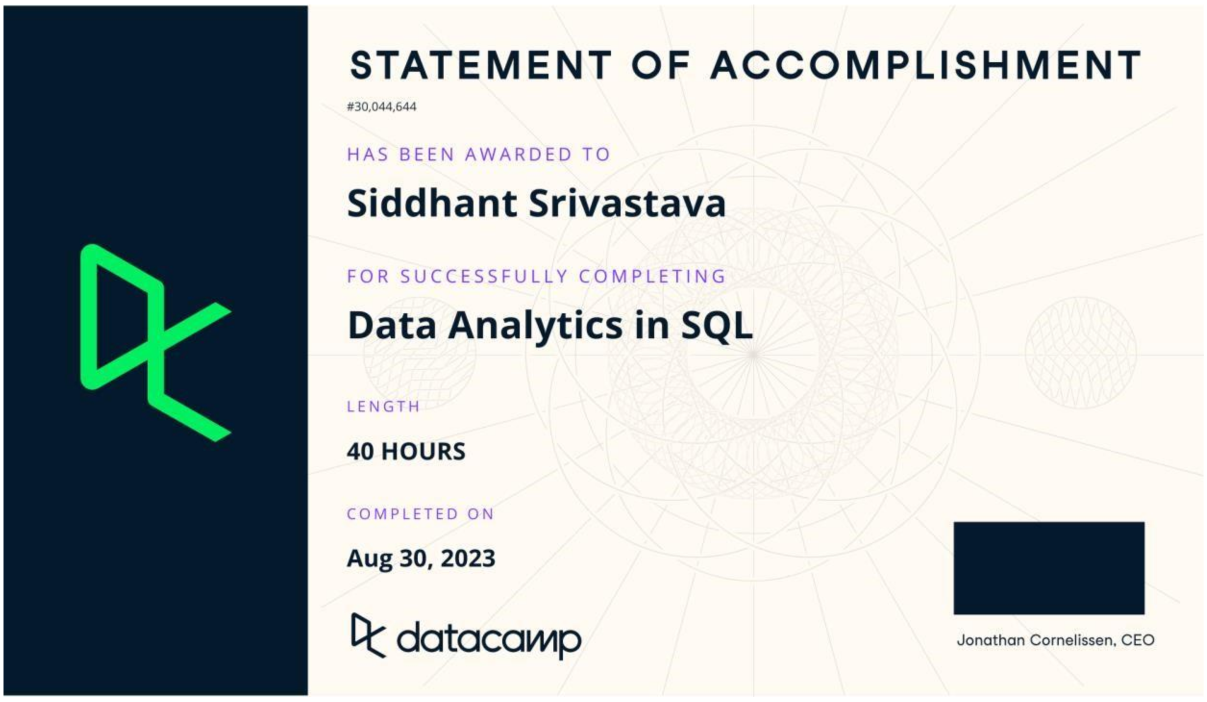
MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

C-4, Janakpuri, New Delhi-58

Affiliated to Guru Gobind Singh Indraprastha University, Delhi

Aug. 2023

CERTIFICATE



STATEMENT OF ACCOMPLISHMENT

#30,044,644

HAS BEEN AWARDED TO

Siddhant Srivastava

FOR SUCCESSFULLY COMPLETING

Data Analytics in SQL

LENGTH

40 HOURS

COMPLETED ON

Aug 30, 2023

datacamp

[redacted]

Jonathan Cornelissen, CEO

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

Summer/Industrial Training Evaluation Form F05 (MSIT-EXM-PA-02)

(Year 20...... -- 20.....)

Details of the Student **Details of the Organisation**

Name……………………………………. Name and address of organisation……………… Roll No………………………………….. ………………………………………………..

Branch and Semester……………………. Broader Area……………………………………

Mobile No………………………………. Name of Instructor………………………………

E-mail ID………………………………. Designation and Contact No……………………..

Student Performance Record

	No. of days Scheduled for the training	Number of days actually attended	Curriculum Scheduled for the student	Curriculum actually covered by the student
Week 1				
Week 2				
Week 3				
Week 4				
Week 5				
Week 6				

(Signature of the student)

Any comments or suggestions for the student performance during the training program (to be filled by instructor)……………………………………………………………………………………………………..
……
……

(Signature of the Instructor)

along with Seal

MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

Summer/Industrial Training Evaluation Form F05 (MSIT-EXM-PA-02)

(Year 20...... -- 20.....)

Student Evaluation Record (to be filled by Examiner)

S.No.	Evaluation Criteria	Performance
1.	Familiarization with the organisation and its working environment (out of 10)	
2.	Level of technical content in the project (out of 30)	
3.	Organisation and formation of Project Report (out of 20)	
4.	Power Point presentation and Viva-Voce examination (out of 40)	

Overall evaluation of the student (Excellent / V. Good / Good/ Satisfactory / Unsatisfactory)

………

………

(Signature and Name of the Examiner)

ACKNOWLEDGEMENT

I offer my sincere thanks and humble regards to Maharaja Surajmal Institute of Technology, Janakpuri Delhi for imparting us very valuable professional training in B.Tech.

I pay my gratitude and sincere regards to Acmegrade , the company for learning all web related skills and all the instructors of the company for giving me the cream of their knowledge. I am thankful to them as they have given me the knowledge of real world web industry related skills.

I am also thankful to my family and friends for constantly motivating me to complete the training and providing me with an environment, which enhanced my knowledge.

VANSHIKA KABRA

04896302820

ECE-3

TABLE OF CONTENTS

Introduction

In a nutshell, front-end web development, also called client-side development, refers to writing Hyper Text Markup Language (HTML), Cascading Style Sheets (CSS) and JavaScript code for a website or web application so that users can see and interact directly with them.

To delve further into this subject, it is important to know the basic differences between front-end development, back-end development and full stack development. Simply put, a front-end developer works on the parts / aspects of an app or website that users get to see and interact with. Whereas a back-end developer handles the behind-the-scenes aspects of the said app or website, say the infrastructure and databases. On the other hand, a full-stack developer does both, so it's someone who deals with the whole design process from start to finish.

The Importance of Front-End Development for Business Success

SQL is a specialized programming language designed for managing and querying relational databases. It has been a staple in the world of data for decades and is essential for anyone involved in data analysis, database management, or business intelligence. SQL allows us to interact with and manipulate data efficiently, making it an indispensable skill for data professionals, analysts, data scientists, and decision-makers.

Why FrontEnd Development?

The front-end team works with essentially every other department at SiteCrafting and GearLab. Collaboration and good communication is key throughout our process.

1. **User Experience (UX) :** Front-end development works with the UX team to help add new features to a website or improve the current usability of a design. Front-end also provides insight to the UX team on understanding possible designs, building out designs, and adjusting design elements.
2. **Visual Design:** In addition to UX, front-end most often collaborates with the visual design team collaborating on design effects and/or animations and working through design feasibility and usability through development
3. **Back-end Development:** The front-end and back-end developers continually communicate during all projects, focusing on aspects such as interactivity and usability and data manipulation.
4. **Content Strategy:** Front-end collaborates with content strategy when entering, creating, and strategizing website content. For example, our content team helps our developers determine the organization and placement of content, along with focusing on SEO.
5. **Quality Assurance (QA)**: All of our projects go through a QA process of thoroughly testing and looking for any potential misses or problems that were not caught during the earlier phases. After being on a project for a while it is essential to have fresh eyes look things over and make sure everything is running smooth and up to spec and QA helps us with that process.
6. **Project Management**: Throughout each project, our front-end developers communicate with the project management team, discussing budgets, timeframes, estimates and project planning.
7. **Front-End Development and the Future**: Front-end development is certainly not as simple as it once was viewed. There are many challenges that we must embrace and we always need to be willing to continually learn, grow, and listen. Teamwork helps keep us honest and from getting stuck in one place and saying good enough.

Fundamentals of HTML

2.1 What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

Each individual piece markup code (which would fall between "<" and ">" characters) is referred to as an element, though many people also refer to it as a tag. Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

1. Structure:

- **Tags:** HTML documents are made up of HTML tags. Tags are keywords enclosed in angle brackets `< >` which define the structure of web content.
- **Elements:** Tags can be combined to form elements, such as headings, paragraphs, links, images, lists, etc.

2. Basic Tags:

- `<html>`: Defines the root element of an HTML page.
- `<head>`: Contains meta-information about the HTML document.
- `<title>`: Sets the title of the HTML document (displayed on the title bar of the browser).
- `<body>`: Contains the content of the HTML document (text, images, links, etc.).

.

3. Text Formatting:

- `<h1> to <h6>`: Headings, with `<h1>` being the highest (most important) level and `<h6>` being the lowest.
- `<p>`: Defines a paragraph.
- `<strong>`: Makes text bold.
- `<em>`: Makes text italic.
- `<u>`: Underlines text.

4. Links and Navigation:

- `<a>`: Defines a hyperlink, linking to another document or resource.

- `<nav>`: Defines a section with navigation links.

- `<ul>`, `<ol>`, `<li>`: Create unordered and ordered lists with list items.

5. Images and Multimedia:

- `<img>`: Embeds images on a web page.
- `<audio>` and `<video>`: Embeds audio and video files respectively.

6. Forms and Input:

- `<form>`: Creates a form for user input.
- `<input>`: Defines an input field within a form.
- `<button>`: Defines a clickable button.
- `<select>`, `<option>`: Create dropdown menus.
- `<textarea>`: Defines a multiline text input control.

7. Attributes:

- HTML tags can have attributes like `id`, `class`, `src`, `href`, etc., providing additional information about an element.

8. Semantic Elements:

- `<header>`, `<footer>`, `<section>`, `<article>`, `<aside>`, `<main>`: Provide semantic meaning to the different parts of a web page, aiding in its structure and accessibility.

9. Version and Doctype:

- `<!DOCTYPE html>`: Declares the HTML version being used, ensuring proper rendering of the document in web browsers.

HTML provides the basic structure for web development, serving as the foundation upon which CSS (Cascading Style Sheets) and JavaScript can be added to enhance the visual presentation and interactivity of web pages.

Introduction To Tailwind CSS

3.1 What is Tailwind CSS?

Tailwind CSS can be used to style websites in the fastest and easiest way. Tailwind CSS is basically a utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override.

The beauty of this thing called tailwind is it doesn't impose design specifications or how your site should look, you simply bring tiny components together to construct a user interface that is unique. What Tailwind simply does is take a 'raw' CSS file, process this CSS file over a configuration file, and produce an output.

3.2 Why Tailwind CSS?

- The faster UI building process
- It is a utility-first CSS framework which means we can use utility classes to build custom designs without writing CSS as in the traditional approach.

3.3 Advantages of Tailwind CSS

- No more silly names for CSS classes and Id.
- Minimum lines of Code in CSS file.
- We can customize the designs to make the components.
- Makes the website responsive.
- Makes the changes in the desired manner.
  CSS is global in nature and if make changes in the file the property is changed in all the HTML files linked to it. But with the help of Tailwind CSS, we can use utility classes and make local changes.

Installation: There are two ways to use the CSS we can install them on our server or we can use the CDN link as well

Tailwind CSS streamlines the styling process, making it efficient for developers to create responsive and visually appealing web applications. Its utility-first approach, customization options, and active community support make it a popular choice for modern web development projects.

Tailwind CSS is a utility-first CSS framework that simplifies the process of styling web applications. Unlike traditional CSS frameworks, Tailwind doesn't provide pre-designed components but offers low-level utility classes that can be combined to create custom designs. Here's a summary of its key aspects:

1. Utility-First Approach:

- **Atomic CSS Classes:** Tailwind's classes are atomic, meaning each class represents a single utility (e.g., margin, padding, color). This allows developers to apply styles precisely without writing custom CSS rules.
- **Utility Class Composition:** Developers can compose utility classes to create complex designs. For instance, `class="bg-blue-500 text-white p-2 rounded"` sets a blue background, white text, padding, and rounded corners.

2. Easy Customization:

- **Configuration:** Tailwind's configuration file allows fine-tuning of the default styles. Developers can customize colors, spacing, typography, and more to match the project's design system.
- **Plugins:** Tailwind's plugin system enables the integration of additional utilities and features. Plugins can be used to add new classes, variants, or even entire components, enhancing Tailwind's functionality.

3. Rapid Prototyping:

- **Development Speed:** By eliminating the need to write custom CSS, developers can prototype and iterate designs quickly. Tailwind's intuitive class names make it easy to visualize styles.
- **Consistent Design:** The consistent use of utility classes promotes a cohesive design language throughout the application, ensuring a professional and polished look.

4. Responsive Design:

- **Mobile-First Approach:** Tailwind encourages starting designs for mobile devices and then progressively enhancing for larger screens. Responsive classes (e.g., `sm:`, `lg:`) enable easy adaptation to different viewports.
- **Responsive Utilities:** Tailwind provides responsive variants for classes, allowing developers to specify styles for different breakpoints. For example, `md:text-center` centers text on medium-sized screens.

5. Community and Ecosystem:

- **Active Community:** Tailwind has a vibrant community that actively contributes plugins, themes, and extensions. Community support forums and social media groups facilitate knowledge sharing and problem-solving.

- **Plugins and Extensions:** Tailwind's ecosystem includes a variety of plugins that add functionalities like dark mode, typography, forms, and more. These plugins save development time and effort.

6. Learning Curve:

- **Accessibility:** Tailwind's utility classes can improve accessibility by focusing on semantic HTML and making it easier to apply accessible styles. For example, classes like `sr-only` hide elements visually but keep them accessible for screen readers.
- **Advanced Usage:** While Tailwind is beginner-friendly, mastering complex layouts or unique designs might require a deeper understanding of class composition, responsive design strategies, and configuration options.

7. Integration and Build Process:

- **Build Tools Integration:** Tailwind integrates seamlessly into various build tools and frameworks. This integration allows for optimizations like purging unused styles, resulting in smaller CSS files and improved performance.
- **JIT Mode:** Just-in-Time (JIT) mode, introduced in Tailwind CSS 2.1 and later versions, compiles styles on-demand, reducing build times and making the development process more efficient.

3.4 How to Use Tailwind CSS?

1. Installation:

First, you need to install Tailwind CSS in your project. You can do this via npm or yarn. Open your terminal and run:

bash Copy code

```bash
# Using npm
npm install tailwindcss

# Using yarn
yarn add tailwindcss
```

2. Configuration:

Generate a Tailwind CSS configuration file (if you haven't already) by running:

```bash
npx tailwindcss init -p
```

This will create a `tailwind.config.js` file in your project.

3. Customization (Optional):

Edit the `tailwind.config.js` file to customize Tailwind's default settings. You can define custom colors, spacing, typography, and more according to your project's design requirements.

4. Including Tailwind CSS in Your Styles:

In your project's main CSS file (usually `styles.css` or `styles.scss`), import Tailwind's default styles:

```css
@import 'tailwindcss/base';
@import 'tailwindcss/components';
@import 'tailwindcss/utilities';
```

Or if you're using PostCSS, you can include it directly in your PostCSS configuration:

```javascript
module.exports = {
  plugins: [
    require('tailwindcss'),
    require('autoprefixer'),
    // ...other plugins
  ]
}
```

5. Using Tailwind CSS Classes:

Now, you can use Tailwind CSS classes directly in your HTML files. For example, to create a blue button, you can apply Tailwind's classes like this:

html Copy code

```html
<button class="bg-blue-500 hover:bg-blue-700 text-white font-bold py-2 px-4
  Click me
</button>
```

In this example:

- `bg-blue-500` sets the background color to a shade of blue.
- `hover:bg-blue-700` changes the background color on hover.
- `text-white` sets the text color to white.
- `font-bold` makes the text bold.
- `py-2` and `px-4` add padding to the top and bottom (py) and left and right (px) of the button.
- `rounded` adds rounded corners to the button.

You can mix and match these classes to create complex layouts and styles for various elements on your website or web application.

6. Responsive Design:

Tailwind makes it easy to create responsive designs. For example, if you want a different margin on small screens, you can do:

html Copy code

```html
<div class="m-4 sm:m-8 md:m-12 lg:m-16 xl:m-20">
  Content goes here
</div>
```

Tailwind CSS is an open source CSS framework. The main feature of this library is that, unlike other CSS frameworks like Bootstrap, it does not provide a series of predefined classes for elements such as buttons or tables.

Introduction To Javascript

4.1 What is Javascript?

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.

JavaScript is a high-level, versatile, and interpreted programming language primarily used for creating interactive and dynamic web content. It allows developers to add functionality, validate forms, manipulate data, and enhance user experience on websites. Unlike HTML and CSS, which are markup and styling languages respectively, JavaScript is a full-fledged programming language that enables complex computations and interactivity within web applications.

4.2 Key Features of Javascript

- **High-level Language:** JavaScript is a high-level language, meaning it abstracts away the complex details of the machine it runs on, allowing developers to focus on writing code without worrying about system-specific intricacies.
- **Interpreted Language:** JavaScript code is executed line by line, interpreted and executed by the browser's JavaScript engine without the need for compilation. This makes it easy to debug and test code quickly.
- **Dynamic Typing:** JavaScript is dynamically typed, meaning variables can hold values of any data type without explicit declaration. The type of a variable is determined during runtime.
- **Object-Oriented:** JavaScript supports object-oriented programming, allowing developers to create and use objects, encapsulation, and inheritance, providing a structured approach to writing code.
- **Event-Driven:** JavaScript is excellent for creating responsive and interactive user interfaces. It can respond to events like mouse clicks, keyboard inputs, and page loading, enabling developers to create dynamic web applications.
- **Cross-platform:** JavaScript is supported by all major web browsers, making it a cross-platform language. It can run on various operating systems, including Windows, macOS, and Linux.

4.3 Basic Syntax

JavaScript code is typically embedded within HTML documents using `<script>` tags, either in the `<head>` or `<body>` section of an HTML file.

html Copy code

```html
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Example</title>
    <script>
        // JavaScript code goes here
        console.log("Hello, World!");
    </script>
</head>
<body>
    <!-- HTML content -->
</body>
</html>
```

4.4 Fundamental Concepts of Javascript

1. Data Types:

a. **Primitive Data Types:**

- **Number:** Represents both integers and floating-point numbers.

javascript Copy code

```javascript
let age = 30;
let price = 19.99;
```

- **String:** Represents sequences of characters, enclosed in single (' '), double (" "), or backticks (` `) for template literals.

javascript Copy code

```javascript
let name = "Alice";
```

- **Boolean:** Represents true or false values.

```javascript
let isAdult = true;
```

- **Undefined:** Represents a variable that has been declared but not assigned a value.

```javascript
let variable;
```

- **Null:** Represents an intentional absence of any object value.

```javascript
let emptyValue = null;
```

- **Symbol (ES6):** Represents unique and immutable values, often used as object property keys.

```javascript
const uniqueSymbol = Symbol('description');
```

b. **Complex Data Types:**

- **Object:** Represents a collection of key-value pairs.

```javascript
let person = {
    name: "Bob",
    age: 25,
    isEmployed: true
};
```

- **Array:** Represents an ordered list of values.

```javascript
let numbers = [1, 2, 3, 4, 5];
```

2. Functions:

a. **Function Declaration:**

Functions in JavaScript are declared using the `function` keyword.

```javascript
function greet(name) {
    console.log("Hello, " + name + "!");
}

greet("Alice"); // Output: Hello, Alice!
```

b. **Function Expression:**

Functions can also be assigned to variables (function expressions).

```javascript
let greet = function(name) {
    console.log("Hello, " + name + "!");
};

greet("Bob"); // Output: Hello, Bob!
```

c. **Arrow Functions (ES6):**

Arrow functions provide a concise syntax for defining functions.

javascript Copy code

```javascript
let greet = (name) => {
    console.log("Hello, " + name + "!");
};

greet("Charlie"); // Output: Hello, Charlie!
```

d. **Higher-Order Functions:**

JavaScript functions can accept other functions as parameters and can return functions.

javascript Copy code

```javascript
function multiplier(factor) {
    return function(x) {
        return x * factor;
    };
}

let double = multiplier(2);
console.log(double(5)); // Output: 10
```

3. Objects and Prototypes:

a. **Object Creation:**

Objects can be created using object literals or constructors.

javascript Copy code

```javascript
// Object Literal
let person = {
    name: "David",
    age: 28
};

// Constructor Function
function Person(name, age) {
    this.name = name;
    this.age = age;
}

let newPerson = new Person("Eva", 30);
```

b. **Prototypes:**

JavaScript is prototype-based, meaning objects can inherit properties and methods from other objects.

javascript Copy code

```javascript
// Adding a method to the Person prototype
Person.prototype.greet = function() {
    console.log("Hello, my name is " + this.name + "!");
};

newPerson.greet(); // Output: Hello, my name is Eva!
```

.

4. Callbacks and Asynchronous Programming:

a. **Callbacks:**

Functions can be passed as arguments to other functions and executed later, allowing for asynchronous operations.

javascript Copy code

```javascript
function fetchData(callback) {
    // Simulating asynchronous operation
    setTimeout(function() {
        callback("Data fetched successfully!");
    }, 2000);
}

fetchData(function(data) {
    console.log(data); // Output: Data fetched successfully!
});
```

b. **Promises and Async/Await (ES6+):**

Promises provide a more structured way to handle asynchronous operations. Async/Await syntax simplifies working with asynchronous code even further.

```javascript
function fetchData() {
    return new Promise(function(resolve, reject) {
        setTimeout(function() {
            resolve("Data fetched successfully!");
        }, 2000);
    });
}

async function fetchDataAsync() {
    try {
        let data = await fetchData();
        console.log(data); // Output: Data fetched successfully!
    } catch (error) {
        console.error(error);
    }
}

fetchDataAsync();
```

JavaScript's rich set of features and its ability to handle various data types, functions, and asynchronous operations make it a powerful language for both beginner-friendly scripting and complex, scalable applications. These concepts form the basis for exploring more advanced topics in JavaScript and building sophisticated web applications.

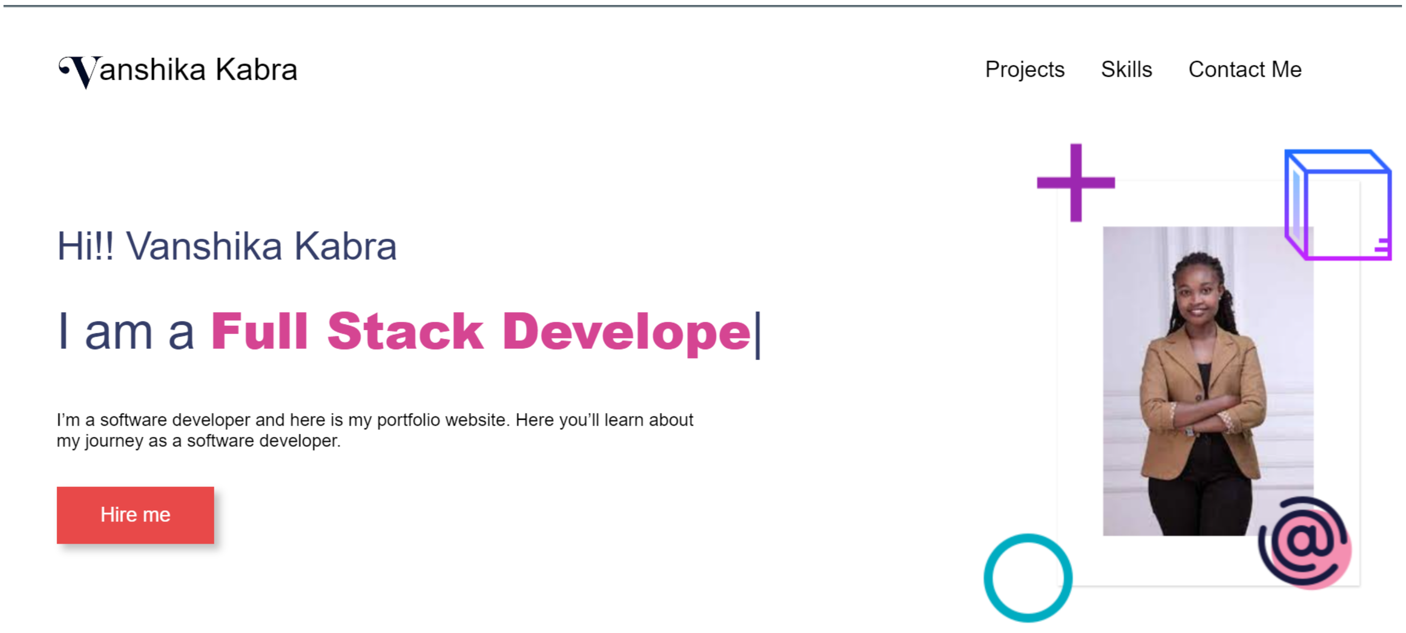
<u>Project</u>

<u>Name of the Project – A Portfolio Website</u>

<u>Aim of the Project:</u> A unique way to showcase your work and let others know about yourself.

<u>Name of the Technologies used:</u>

- HTML – Hypertext Markup Language
- CSS – Cascading Style Sheets
- JS – Javascript
- VS Code – Visual Studio Code


Vanshika Kabra

Projects Skills Contact Me

Hi!! Vanshika Kabra

I am a **Full Stack Develope**|

I'm a software developer and here is my portfolio website. Here you'll learn about my journey as a software developer.

Hire me

<u>Link of the Website</u> - [portfolio_website]

<u>Use of this Website -</u>

Anyone who wants to show about their ownself in a unique way can use this website. It can also be used as a resume. A person is required to change the name, photo, their project link in the code.

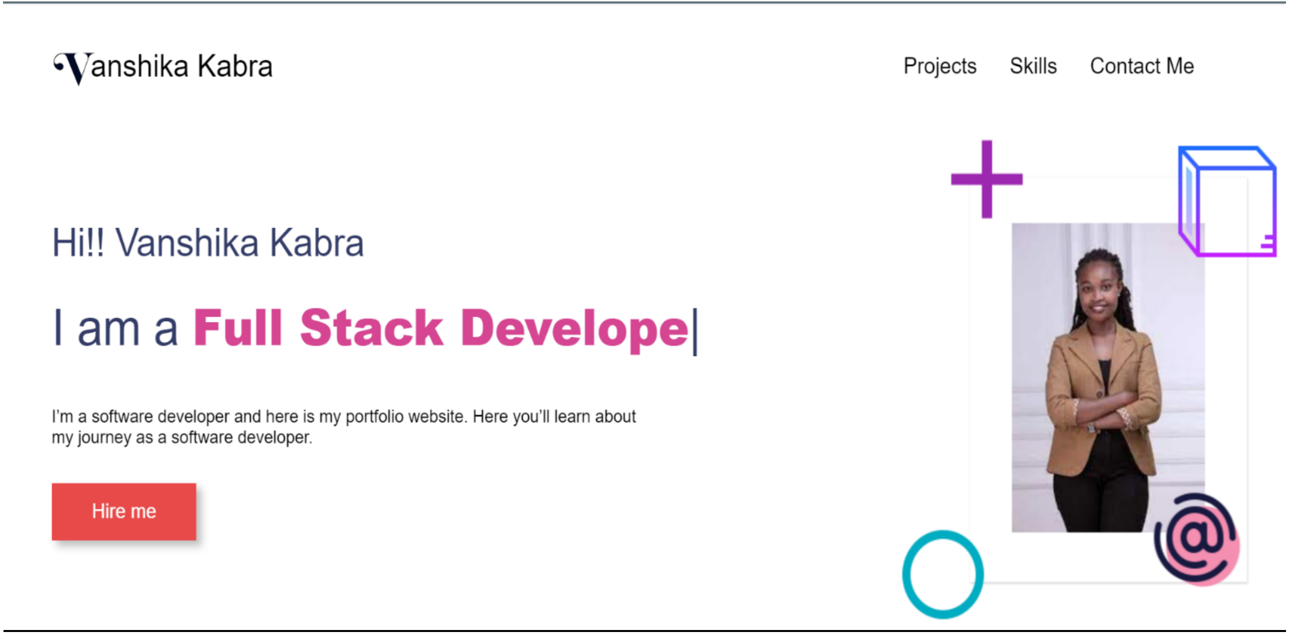
Introduction

Portfolio Websites provides brief information about the professional background of an individual. This project comprises of three files. First is an HTML file for creating markup of the application on the web. Then CSS is used to provide more styles to the website. Then Js file is used to give a proper functionality to the application through Javascript.

Different Section of the Portfolio website-

- *Home Section -*

  It displays my name, who I am, my photo, animations. On the top of it also has a navigation bar which displays other tabs of my website and my contact detail shortcuts on top right.

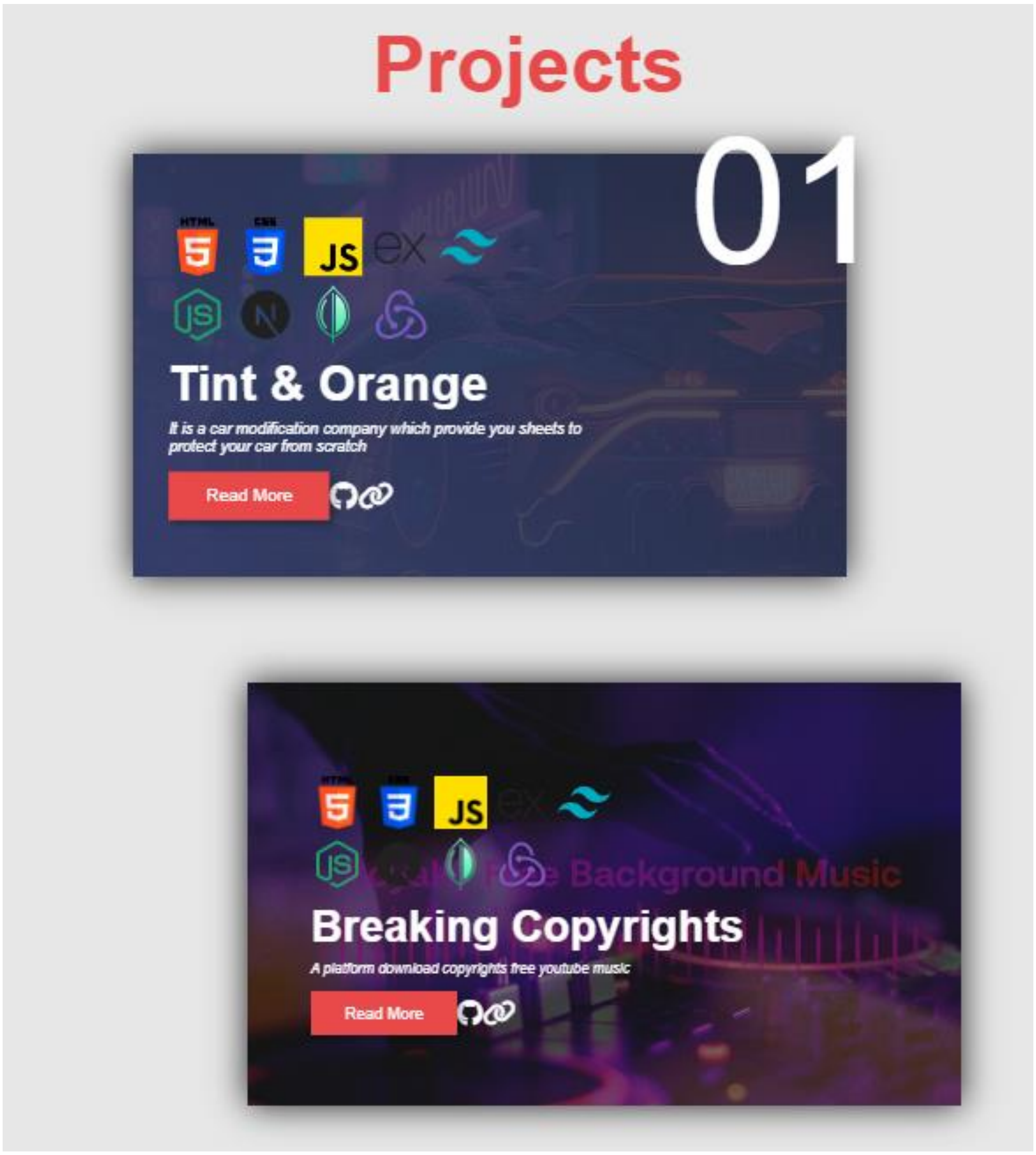


I'm a software developer and here is my portfolio website. Here you'll learn about my journey as a software developer.

Home Section

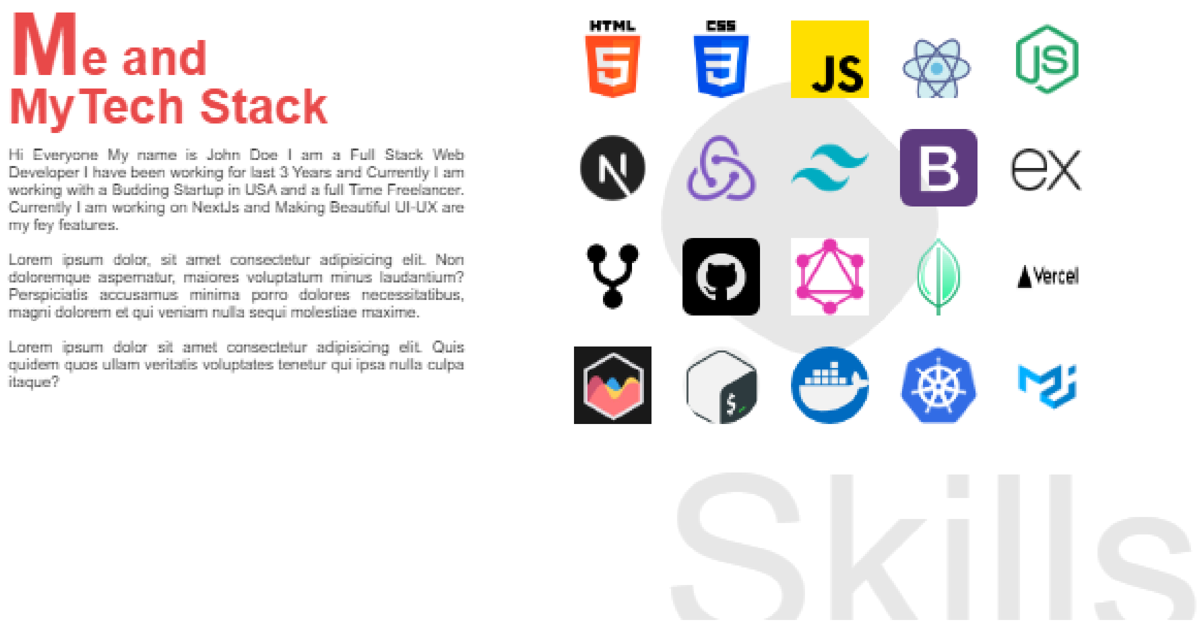
- *Project Section –*

  _This section shows the information about projects done by an individual, github links of their projects, technologies used and description about the project. It is made in the form of a card comprises of animations, gradient, icons etc.


Projects
01
Tint & Orange
It is a car modification company which provide you sheets to protect your car from scratch
Read More
Breaking Copyrights
A platform download copyrights free youtube music
Read More

- *<u>Me and My Tech Stack Section-</u>*

This section contains an information about an individual on the left side such as the name, their experience, where they are currently working etc.
The right side contains an animation of a blob and several tech icons which represents an individual's tech stack.



Me and
MyTech Stack

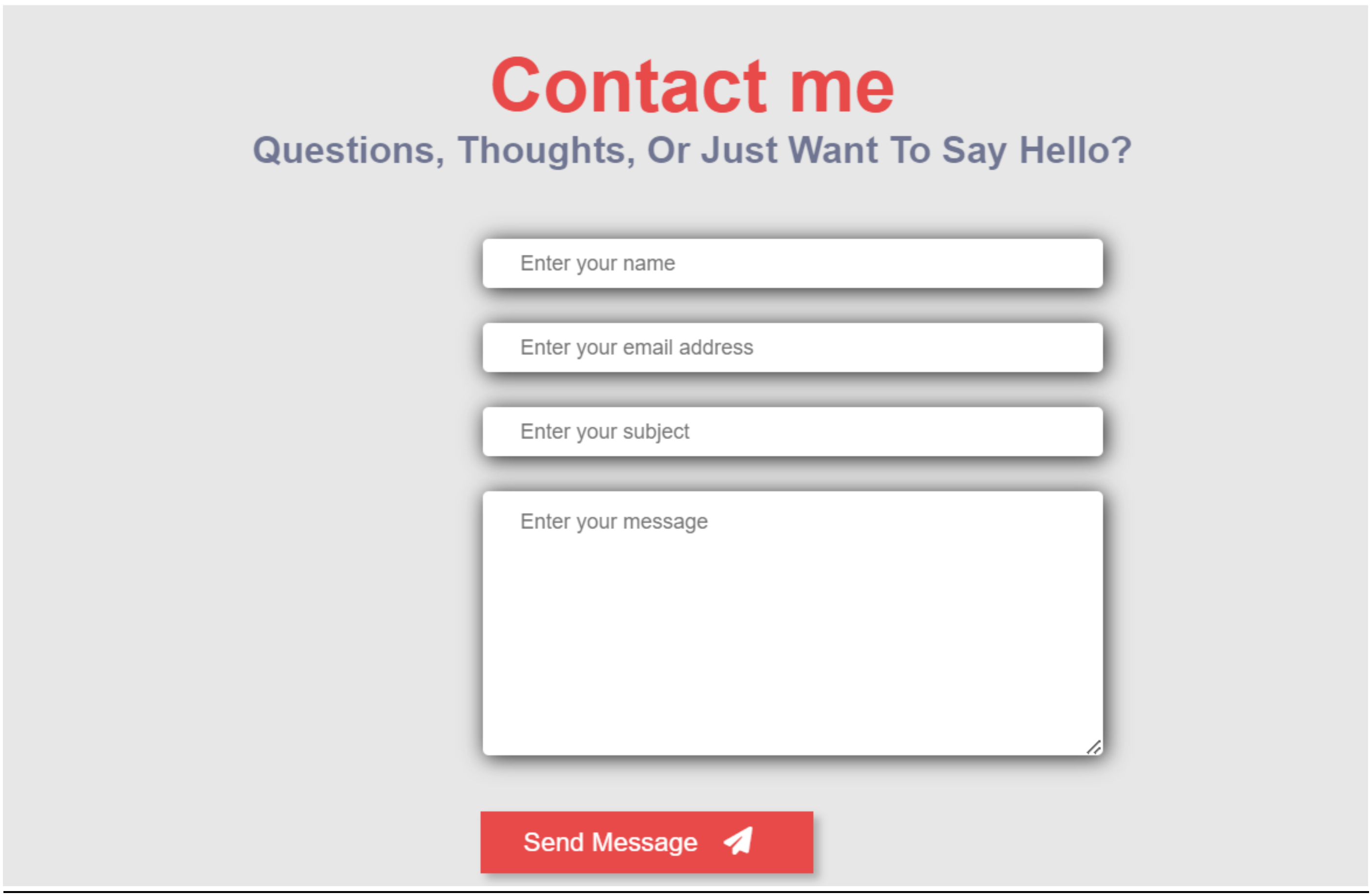
Hi Everyone My name is John Doe I am a Full Stack Web Developer I have been working for last 3 Years and Currently I am working with a Budding Startup in USA and a full Time Freelancer. Currently I am working on NextJs and Making Beautiful UI-UX are my fey features.

Lorem ipsum dolor, sit amet consectetur adipisicing elit. Non doloremque aspernatur, maiores voluptatum minus laudantium? Perspiciatis accusamus minima porro dolores necessitatibus, magni dolorem et qui veniam nulla sequi molestiae maxime.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis quidem quos ullam veritatis voluptates tenetur qui ipsa nulla culpa itaque?

- *<u>Contact Me Section-</u>*

This section contains a form that require name, email address, message of an individual who wants to contact the profile's owner.

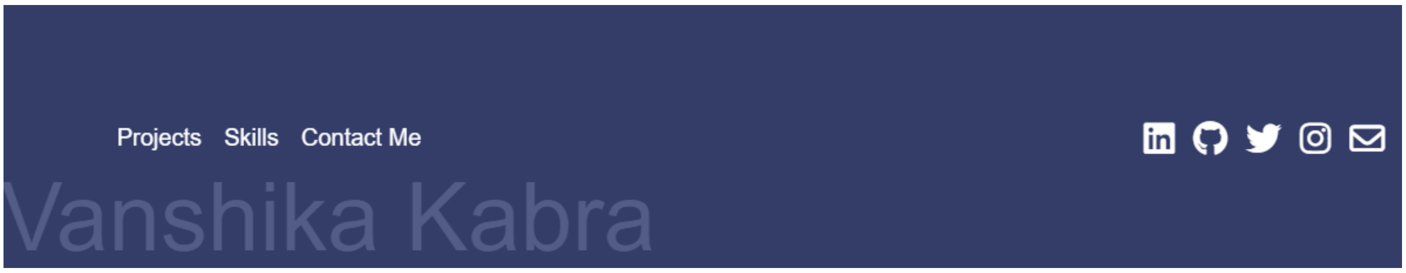


Contact me

Questions, Thoughts, Or Just Want To Say Hello?

- *Footer Section-*

This section contains various link such as LinkedIn, Github, Twitter, Instagram and message to contact the person for hiring.



Conclusion

Front End development primarily focuses on User Experience. Using the coding and design techniques, you as a front-end developer build the element of an application that are directly accessed by the end-users with the goal of easy to use, fast, and secure, user engagement and interactions. But thanks to the latest design and technologies which helps to build sophisticated design and interactive designs.

A major trend in front-end development in recent years has increased due to the use of mobile and smart devices. Therefore, front-end developers must ensure that their applications are consistent, fast and high-quality user experience for all the devices. That's very big challenge.

This is the guidance or learning path, not the rules. The objective is to accommodate the learning based on the experience of the people in the community.

References

1.	**W3Schools.** (n.d.). https://www.w3schools.com/
2.	**Tailwind CSS - https://tailwindcss.com/**
3.	**MDN Web Docs - https://developer.mozilla.org/en-US/docs/Web**
4.	**Js Docs - https://devdocs.io/javascript/**
5.	**Stack Overflow.** (n.d.). https://stackoverflow.com/