WEB DEVELOPMENT

An Internship Report Submitted to the Department of

INFORMATION TECHNOLOGY

AND

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

By

PENNA KUNVESH (22KB1A1243)

Under the esteemed guidance of

Slash Mark

DEPARTMENT OF IT AND AI&DS

N.B.K.R INSTITUTE OF SCIENCE AND TECHNOLOGY

(Autonomous)

VIDYANAGAR-524413, TIRUPATI DISTRICT, AP

2024-25

(DURATION: 04 WEEKS)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of project would beincomplete without the people who made it posable because of their constant guidance and encouragement crowned our efforts with success.

We convey our special thanks to **Dr. Y. Venkata Rami Reddy**, Chairman of **N.B.K.R. Institute of Science and Technology**, for providing excellent infrastructure in our campus for the completion of the internship.

We convey our special thanks to **Sri N. Ram Kumar Reddy**, Correspondent of **N.B.K.R. Institute of Science and Technology**, for providing excellent infrastructure in our campus for the completion of the internship.

We are grateful to **Dr. V. Vijaya Kumar Reddy, Director of N.B.K.R Institute of Science and Technology** for allowing us to avail all the facilities in the college.

We express our sincere gratitude to **Dr. A. Narayana Rao, Professor, Head of Department, IT and AI&DS** for providing exceptional facilities for successful completion of our internship.

We express our sincere gratitude to **Mrs. S.Mamatha, Internship Assistant Professor, Department of IT and AI&DS**, for guiding us towards successful completion of internship.

We would like to convey our heartful thanks to **Staff members, Lab technicians and our friends**, who extended their cooperation in making this internship a successful one.

We would like to thank one and all who have helped us directly and indirectly to complete this internship successfully.

(INTERNSHIP ASSOCIATE)
Mrs. S. Mamatha

INDEX

List of the Concepts **Page No.**

List of the Concepts **Page No.**

1. INTRODUCTION

Web development is the process of creating and maintaining websites and web applications accessible via the internet or intranet. It combines design, development, and programming to build functional, user-friendly, and visually appealing websites using core technologies like HTML for content structure, CSS for styling, and JavaScript for interactivity. These technologies work together to create both static and dynamic websites.

Web development covers a broad scope, including designing user-friendly websites, ensuring responsiveness across devices, and implementing interactive elements. Client-side development involves structuring content with HTML, styling with CSS, and adding interactivity with JavaScript. Server-side development, which handles backend tasks like databases and user authentication, is often more complex. Responsive design ensures websites work on all devices, while SEO optimizes websites for better search engine visibility. Interactive content, powered by JavaScript, improves user engagement with dynamic features.

The history of web development tracks the evolution of the internet, starting in the 1990s with simple static pages using HTML. In the mid-1990s, CSS was introduced to separate content from design, followed by JavaScript in the late 1990s to add interactivity. The early 2000s saw the rise of Web 2.0 with user-generated content, and HTML5 and CSS3 introduced advanced features like multimedia embedding and responsive design. Modern development focuses on mobile-first design and uses frameworks like React and Angular for efficiency, while Progressive Web Apps (PWAs) offer app-like experiences in the browser.

Web development plays a crucial role in many sectors. In e-commerce, it enables online shopping platforms like Amazon. In education, platforms like Coursera provide e-learning experiences. Streaming services like Netflix and Spotify rely on web development for content delivery. Healthcare platforms offer online booking and consultations, while social networks like Facebook facilitate user interaction. Businesses use websites to promote their products, and individuals create portfolios to showcase their skills. Web development continues to shape how we interact with the digital world, making it an essential tool in various industries.

2.BASICS OF HTML, CSS, AND JAVASCRIPT

2.1 HTML (HyperText Markup Language)

HTML is the foundation of any web page. It defines the structure of the content that is displayed in the browser. HTML uses a system of tags to organize the content into a hierarchy of elements, which can represent everything from paragraphs to images and links. The beauty of HTML lies in its simplicity and versatility—it forms the backbone of all web content.

Key Concepts of HTML:

- **Tags and Elements:** HTML tags are used to define different elements on a web page. Every element typically consists of an opening and closing tag, though some tags are self-closing (e.g., image tags). Tags serve as markers that denote the start and end of an element.
- **Attributes:** Tags can include attributes that provide additional information about an element. Attributes are written in the opening tag and are usually expressed as a name-value pair, such as href for links or src for images. These attributes modify the behavior or presentation of the element.
- **Document Structure:** An HTML document is structured in a tree-like format. It begins with the <!DOCTYPE> declaration, followed by the <html> element. Inside the <html> tag, there is a <head> element containing metadata (such as the page title or linked stylesheets) and a <body> element that contains all the visible content.
- **Semantic HTML:** Beyond basic tags, HTML also supports semantic elements, such as <header>, <footer>, <article>, and <section>. These elements help organize content logically and improve accessibility, search engine optimization (SEO), and the overall structure of the web page.

Role of HTML in Web Development:

- **Web Page Structure:** HTML provides the basic structure that browsers use to render the page content. It allows web developers to organize and label sections of a page.
- **Accessibility:** HTML's semantic tags play a significant role in making websites more accessible to individuals with disabilities by improving screen reader compatibility.
- **SEO Benefits:** Proper use of HTML elements can positively influence SEO, making a website more discoverable by search engines.

2.2 CSS (Cascading Style Sheets)

CSS is the language used to describe the presentation and layout of a web page. While HTML is responsible for the structure, CSS allows developers to style and position elements on a page, adding colors, fonts, spacing, and other visual effects.

Key Concepts of CSS:

- **Selectors:** A CSS selector is used to target specific HTML elements in order to apply styles. Selectors can be simple, targeting tags like h1 for headers, or more complex, using classes or IDs to apply styles to specific elements on the page.
- **Properties and Values:** Each CSS rule consists of a property and a value. For example, the color property could be set to a value such as red or #FF0000, which changes the color of the text. There are hundreds of properties in CSS that control various aspects of an element's appearance, such as font-size, margin, padding, background-color, and more.
- **Box Model:** The CSS box model defines the layout structure for every HTML element. Each element is treated as a rectangular box with content, padding, borders, and margins. Understanding the box model is essential for controlling layout and spacing in web design.
- **Positioning and Layouts:** CSS allows for complex layouts, such as multi-column designs and responsive grids. The position property enables developers to control where elements are placed on the page, with options like static, relative, absolute, and fixed positioning. Flexbox and CSS Grid are modern tools for creating responsive, flexible layouts without the need for floats or complex calculations.
- **Responsive Design:** CSS plays a significant role in creating responsive web designs. Media queries allow developers to apply different styles based on the screen size, resolution, or orientation of the device, ensuring a seamless experience across desktops, tablets, and smartphones.

Role of CSS in Web Development:

- **Visual Styling:** CSS handles all the visual aspects of a webpage, including colors, typography, spacing, and positioning.

- **User Experience:** By controlling how content is displayed, CSS greatly enhances the user experience. It can make a website more intuitive and aesthetically pleasing, encouraging visitors to engage with the content.
- **Performance:** Well-organized CSS can improve page load times, making websites faster. By minimizing and optimizing CSS files, developers can significantly reduce file sizes and improve performance.

2.3 JavaScript

JavaScript is a programming language that enables interactive elements on a webpage. Unlike HTML and CSS, which are declarative, JavaScript is a procedural language used to add functionality and behavior to web pages.

Key Concepts of JavaScript:

- **Variables and Data Types:** JavaScript uses variables to store and manipulate data. Common data types in JavaScript include numbers, strings, booleans, arrays, and objects. Understanding data types is fundamental to working with JavaScript effectively.
- **Functions:** Functions in JavaScript are reusable blocks of code that can be called with specific arguments. They can return values and are key for managing logic in JavaScript programs. Functions allow developers to break down complex tasks into simpler, manageable components.
- **Control Structures:** JavaScript includes control structures like if-else statements, loops (such as for and while), and switch cases. These constructs allow developers to control the flow of execution based on certain conditions.
- **DOM Manipulation:** The Document Object Model (DOM) represents the structure of a webpage in memory. JavaScript can interact with the DOM to modify the content and structure of a page dynamically, such as changing the text of an element or adding new elements to the page.
- **Event Handling:** JavaScript enables websites to respond to user interactions like clicks, keypresses, and mouse movements. This is done using event listeners, which allow specific actions to be triggered when an event occurs, such as showing a hidden menu when a button is clicked.

Role of JavaScript in Web Development:

- **Interactivity:** JavaScript makes websites interactive by responding to user actions in real-time. This could include things like form validation, interactive maps, or dynamically changing content.
- **Client-Side and Server-Side:** Initially, JavaScript was only used for client-side scripting, but with the advent of Node.js, it is now also widely used for server-side programming, enabling full-stack development with a single language.
- **Web Application Development:** JavaScript is a key technology for building dynamic and rich web applications. Frameworks like React, Angular, and Vue.js make it easier to develop complex user interfaces, while Node.js enables server-side logic.

3.HTML STRUCTURE AND ELEMENTS

HTML (Hypertext Markup Language) is the backbone of any webpage, providing the structure and content layout that browsers render. Understanding the structure of an HTML document and its various elements is essential for web development. This section explores the key aspects of HTML structure, elements, and how they work together to form a webpage.

3.1 Basic Structure of an HTML Document

An HTML document is a text file that contains a structured arrangement of tags. These tags define the content and structure of a webpage. Every HTML document follows a basic structure that includes essential components to ensure the page is correctly interpreted by web browsers.

Key Components:

- **DOCTYPE Declaration:** The <!DOCTYPE> declaration is placed at the very beginning of an HTML document. It informs the browser about the type of document being rendered, which helps the browser to render the page correctly. For modern HTML documents, the declaration is written as <!DOCTYPE html>.

- **HTML Element (**<html>**):** The <html> element is the root of an HTML document. All other elements are nested inside the <html> tag. It tells the browser that everything inside is an HTML document.

- **Head Element (**<head>**):** The <head> section contains meta-information about the document that is not directly visible on the webpage. It typically includes:

    - **Metadata:** Tags like <meta> define the character set, language, and viewport settings.

    - **Title:** The <title> tag defines the title of the webpage, which appears in the browser tab.

    - **Link to Stylesheets:** The <link> tag is used to link external CSS stylesheets that define the visual appearance of the webpage.

    - **Scripts:** External JavaScript files can be included in the <head> or before the closing </body> tag.

- **Body Element (**<body>**):** The <body> element contains all the visible content of the webpage. Everything that a user sees—text, images, links, videos, forms, and interactive elements—resides within the <body> tag.

3.2 HTML Elements

HTML elements are the fundamental building blocks of a webpage. They consist of an opening tag, content, and a closing tag. Some elements, like images or line breaks, may be self-closing. Understanding the types of elements and their uses is critical in creating a meaningful and functional webpage.

Key Categories of HTML Elements:

- **Text Content Elements:** These elements are used to define and structure text content on the page.
    - **Headings:** The <h1> to <h6> elements define headings, with <h1> being the most important and <h6> the least important.
    - **Paragraph:** The <p> element is used to define paragraphs of text.
    - **Links:** The <a> (anchor) element is used to create hyperlinks to other web pages or resources.
    - **Lists:** Lists come in two main types: ordered lists (<ol>) and unordered lists (<ul>), with each list item defined by the <li> tag.
    - **Images:** The <img> tag is used to embed images in the document.
- **Form Elements:** Forms are essential for user interaction on web pages, such as collecting user input. Common form elements include:
    - **Input Fields:** The <input> element allows users to enter data, such as text, numbers, or passwords.
    - **Text Areas:** The <textarea> element creates a multi-line input field for longer text entries.
    - **Buttons:** The <button> element defines clickable buttons used to submit forms or trigger actions.
    - **Select Boxes:** The <select> element, combined with <option>, creates dropdown menus.
- **Multimedia Elements:** HTML includes elements for embedding multimedia content like audio and video.
    - **Audio:** The <audio> element allows for embedding sound files.

    - **Video:** The <video> element allows embedding video files.

- **Structural Elements:** These elements are used to organize and group content into logical sections.
    - **Division (Div):** The <div> element is a block-level container used to group content for styling or scripting purposes. It has no semantic meaning but is commonly used for layout and structure.
    - **Sectioning Elements:** HTML5 introduced new elements like <header>, <footer>, <article>, <section>, and <nav> that help define the document's structure semantically.

3.3 Nesting and Hierarchy of HTML Elements

HTML elements can be nested inside other elements to create complex layouts. Nesting refers to placing one element inside another, allowing for hierarchical structuring of content. Correct nesting helps maintain a logical structure for the page, ensuring content is displayed in an organized and accessible manner.

Best Practices for Nesting:

- **Proper Hierarchy:** Parent elements, like <div> or <section>, can contain child elements such as paragraphs, images, or lists. However, improper nesting can lead to rendering issues and difficulties in styling.
- **Avoid Overuse of** <div> **Tags:** While the <div> tag is useful for creating sections, it should not be overused. Semantic tags like <article>, <aside>, or <section> should be preferred as they provide meaningful structure and enhance the accessibility and SEO of the page.
- **Consistency:** Always ensure that every opening tag has a corresponding closing tag unless the element is self-closing (e.g., <img> or <br>). Unmatched tags can lead to errors and unexpected rendering.

3.4 HTML Attributes

HTML attributes provide additional information about an element. They modify the behavior or appearance of an element and are written in the opening tag.

Common HTML Attributes:

- **Class and ID:** The class attribute is used to assign a specific style or behavior to an element, often used in CSS or JavaScript. The id attribute assigns a unique identifier to an element, useful for targeting specific elements with styles or scripts.
- **Href (Hyperlink Reference):** The href attribute in the anchor (<a>) tag specifies the URL the link points to.
- **Src (Source):** The src attribute in elements like <img> and <script> specifies the source of the image or script file.
- **Alt (Alternative Text):** The alt attribute provides a textual description of an image. It is essential for accessibility purposes, particularly for users who rely on screen readers.
- **Placeholder:** The placeholder attribute is used in input fields to display a short hint or instructional text within the field.
- **Value:** The value attribute is commonly used in form elements to specify the value sent to the server when a form is submitted.

3.5 HTML 5 Semantic Elements

HTML5 introduced a set of semantic elements that give meaning to different parts of a webpage, improving accessibility, SEO, and code readability. These elements help define the structure of a document, making it easier to understand and navigate.

Important HTML5 Semantic Elements:

- <header>**:** Represents the header of a document or a section. It typically contains introductory content or navigation links.
- <footer>**:** Represents the footer of a document or section, usually containing metadata, copyright information, or links.

- <nav>**:** Represents a section of navigation links, helping organize the navigation structure of a webpage.
- <article>**:** Represents an independent piece of content, such as a blog post or news article, which can be distributed or syndicated.
- <section>**:** Represents a section of content, typically with a thematic grouping of content. It can be used to break a page into distinct sections.
- <aside>**:** Represents content that is tangentially related to the main content, such as sidebars or pull quotes.

3.6 Accessibility and Best Practices in HTML Structure

Accessibility is a key consideration in modern web development. Ensuring that HTML documents are structured properly helps users with disabilities, including those using screen readers, navigate websites more easily.

Accessibility Considerations:

- **Semantic Markup:** Use semantic HTML tags (<header>, <footer>, <article>, etc.) to help screen readers interpret the structure of a webpage.
- **Alt Text for Images:** Always provide descriptive alt text for images to help users with visual impairments.
- **Forms Accessibility:** Ensure that form elements are labeled properly with the label tag, and group related form elements using <fieldset> and <legend> tags.

4.CSS STYLING AND LAYOUTS

CSS enables developers to separate content from design, making it easier to manage and maintain a website's appearance. Through CSS, developers can apply styles to HTML elements, control their layout, and enhance user interactions.

4.1 Key Concepts in CSS Styling:

- **Selectors:** CSS selectors are used to target HTML elements that need to be styled. Selectors determine which HTML elements the CSS rules will apply to. There are several types of selectors:
  - **Element Selectors:** Target elements by their tag name, e.g., p for paragraphs, h1 for headings.
  - **Class Selectors:** Use a period (.) followed by a class name, e.g., .header. These can be applied to multiple elements.
  - **ID Selectors:** Use a hash (#) followed by an ID name, e.g., #footer. IDs should be unique within a page.
  - **Universal Selector:** The asterisk (*) targets all elements on the page.
- **Properties and Values:** Each CSS rule consists of a property and a value. The property defines the aspect of the element to be styled (e.g., color, font-size, margin), and the value specifies the style to be applied (e.g., red, 16px, 10px).
- **CSS Cascade and Specificity:** CSS rules cascade, meaning that styles are applied based on their order of appearance. Specificity determines which rule takes precedence when multiple rules conflict. For example, a more specific selector (e.g., ID selector) will override a more general one (e.g., element selector).
- **Box Model:** The CSS box model is the concept of how elements are rendered on the page. Each element is treated as a rectangular box that consists of:
  - **Content:** The actual content inside the element.
  - **Padding:** The space between the content and the border.
  - **Border:** A line around the padding (optional).
  - **Margin:** The space outside the border, separating the element from other elements.

Basic Styling Properties:

- **Text Styling:** CSS provides properties like font-family, font-size, font-weight, and text-align to control the appearance and alignment of text.
- **Color and Background:** The color property defines the text color, and the background-color property sets the background color of an element.
- **Borders and Shadows:** CSS allows you to add borders around elements using the border property and create shadows with box-shadow and text-shadow.
- **Spacing and Alignment:** The margin and padding properties control the spacing inside and outside elements, while the text-align property is used to align text within its container.

4.2 Advanced CSS Styling Techniques

CSS also allows for more advanced styling techniques, providing flexibility and control over the design and aesthetics of a website.

Typography:

Typography plays a critical role in the readability and appeal of a website. CSS provides several properties to control the typography of text:

- **Font Families:** The font-family property allows you to specify the typeface for your text. You can define web-safe fonts or use custom fonts through @font-face or font-loading services like Google Fonts.
- **Font Size and Weight:** The font-size property controls the size of the text, while font-weight sets the boldness of the font.
- **Line Height:** The line-height property adjusts the vertical spacing between lines of text, improving readability.
- **Text Transform and Decoration:** Properties like text-transform (to control capitalization) and text-decoration (for underlining, striking through, etc.) are often used for styling headings and links.]

CSS Pseudo-Classes and Pseudo-Elements:

- **Pseudo-Classes:** These are used to define the special state of an element, such as :hover (when an element is hovered over), :focus (when an element is in focus), and :active (when an element is being clicked).
- **Pseudo-Elements:** These allow styling of parts of an element, such as ::before and ::after, which add content before or after an element's content.

CSS Transitions and Animations:

- **Transitions:** CSS transitions allow you to animate changes in CSS properties over time. For example, you can smoothly change the background color or size of an element when it is hovered over.
- **Animations:** CSS animations provide more complex control over changes to elements over time, including keyframes to define the state of the animation at various points.

4.3 CSS Layout Techniques

CSS provides various techniques for creating layouts and positioning elements on a webpage. Understanding how to use CSS for layout management is essential for designing responsive and structured webpages.

Traditional Layout Techniques:

- **Float:** The float property was traditionally used to create layouts by placing elements side by side. However, it is now considered outdated for creating complex layouts and is mostly used for simple positioning tasks.
- **Positioning:** CSS provides several positioning schemes using the position property:
  - **Static:** The default position, where elements flow according to the normal document flow.
  - **Relative:** The element is positioned relative to its normal position.
  - **Absolute:** The element is positioned relative to the nearest positioned ancestor (non-static).
  - **Fixed:** The element is fixed relative to the viewport, meaning it stays in the same position even when the page is scrolled.

  - **Sticky:** The element toggles between relative and fixed positioning, depending on the user's scroll position.

Modern Layout Techniques:

- **Flexbox:** The CSS Flexbox layout is a powerful tool for creating flexible and responsive layouts. It allows for alignment and distribution of space within a container, even when the sizes of the items are unknown or dynamic. Flexbox makes it easy to create complex layouts with minimal code.
  - **Main Axis and Cross Axis:** Flexbox uses a main axis and cross axis to align items. The main axis runs in the direction of the container, while the cross axis runs perpendicular to it.
  - **Justify Content:** This property controls alignment along the main axis (e.g., center, space-between).
  - **Align Items and Align Self:** These properties control alignment along the cross axis for all items or individual items, respectively.
- **CSS Grid:** CSS Grid is another powerful layout system that allows you to create complex two-dimensional layouts. It works with both rows and columns, enabling the creation of intricate designs without the need for floats or positioning.
  - **Grid Lines:** The layout is defined by rows and columns, with elements being positioned on the grid using lines (e.g., grid-column and grid-row).
  - **Grid Template:** You can define the size and number of columns and rows with the grid-template-columns and grid-template-rows properties.
- **Responsive Layouts with Media Queries:** Media queries allow developers to create layouts that adjust to different screen sizes. By defining breakpoints, developers can specify different styles for various devices, from desktops to tablets and smartphones.
  - **Example Breakpoints:** Common breakpoints include @media (min-width: 768px) for tablets and @media (min-width: 1024px) for desktops.

4.4 Responsive Web Design with CSS

Responsive web design ensures that a webpage looks great and functions well across a wide range of devices, from desktops to smartphones. CSS plays a crucial role in building responsive layouts and adapting to different screen sizes.

Key Concepts in Responsive Design:

- **Fluid Layouts:** Fluid layouts use relative units like percentages or em instead of fixed units like pixels. This allows the layout to adjust based on the screen size.
- **Media Queries:** Media queries enable developers to apply specific styles depending on the device's characteristics, such as width, height, and orientation.
- **Viewport Units:** CSS provides viewport-relative units (vw, vh, vmin, vmax) that allow elements to scale relative to the viewport size.
- **Flexible Images:** The max-width: 100% property ensures that images scale properly within their containers, preventing overflow and maintaining responsiveness.

4.5 Best Practices for CSS Styling and Layouts

- **Separation of Concerns:** Keep the CSS code separate from HTML content. This improves maintainability and scalability, especially for larger websites.
- **Consistency in Naming Conventions:** Use consistent and descriptive class and ID names. BEM (Block Element Modifier) is a popular naming convention that helps maintain clarity in styling.
- **Performance Optimization:** Minimize and combine CSS files, use CSS shorthand properties, and avoid excessive use of complex selectors to improve page load times and performance.
- **Cross-Browser Compatibility:** Always test your designs in different browsers (Chrome, Firefox, Safari, etc.) to ensure that they render correctly across all platforms.

5. JAVASCRIPT INTERACTIVITY

JavaScript is a client-side scripting language that runs in the web browser, enabling developers to add dynamic behavior to web pages. By interacting with the page's Document Object Model (DOM), JavaScript enables actions like modifying text, changing styles, responding to user inputs, and updating the page without needing to reload it.

5.1 Key Concepts in JavaScript Interactivity:

- **Events:** In JavaScript, an event is an action that occurs due to user interaction with the webpage. Events include actions such as clicking buttons, submitting forms, hovering over elements, or pressing keys. JavaScript can respond to these events to trigger specific actions or changes on the webpage.

- **DOM Manipulation:** The DOM represents the structure of a webpage, where each element is treated as an object. JavaScript can interact with and manipulate these objects to change the content, appearance, or behavior of a webpage dynamically.

- **Functions:** Functions in JavaScript are blocks of reusable code that can be executed when an event occurs or when a specific condition is met. Functions are crucial for handling interactivity in a structured and organized way.

5.2 Event Handling in JavaScript

Event handling is at the core of interactivity in JavaScript. An event occurs when a user performs a specific action, such as clicking on a button or typing in a text field. JavaScript listens for these events and responds accordingly, triggering actions like showing or hiding content, changing colors, or submitting data.

Types of Events:

- **Mouse Events:** These events are triggered by actions performed with the mouse, such as clicking, double-clicking, hovering, or dragging. Common mouse events include click, mouseover, and mouseout.

- **Keyboard Events:** Keyboard events respond to actions involving the keyboard, such as pressing or releasing a key. These events include keydown, keyup, and keypress.

- **Form Events:** Form events are related to user interactions with form elements, such as submitting a form, entering text into an input field, or selecting a checkbox. Common form events include submit, input, and change.

- **Window Events:** These events are triggered by changes in the browser window, such as resizing, scrolling, or loading the page. Examples include resize, scroll, and load.

Event Listeners:

To respond to an event, JavaScript uses event listeners. An event listener is a function that waits for a specific event to occur and executes a set of instructions when that event is triggered. This is how JavaScript can respond to user interactions, such as clicks or form submissions.

5.3 DOM Manipulation with JavaScript

The DOM (Document Object Model) is a programming interface for web documents. It represents the structure of an HTML document as a tree of objects, where each object corresponds to a part of the page, such as elements, attributes, or text.

JavaScript can access and modify these objects to change the content or structure of the page dynamically. This allows for actions such as updating the text inside a paragraph, changing the background color of an element, or adding new elements to the page without reloading it.

Common DOM Manipulations:

- **Modifying Content:** JavaScript can change the text content of elements, such as paragraphs, headings, and links.
- **Changing Styles:** JavaScript can modify the CSS styles of elements, such as altering the background color, font size, or visibility.
- **Creating and Removing Elements:** JavaScript can create new HTML elements or remove existing ones, allowing developers to update the page's structure as needed.

5.4 Dynamic Content Updates

JavaScript enables dynamic updates to the content of a webpage without requiring the page to reload. This is particularly useful for creating responsive web applications that can update content in real-time, such as loading new data, submitting forms, or responding to user inputs.

AJAX (Asynchronous JavaScript and XML):

AJAX is a technique used to send and receive data asynchronously between the client (browser) and the server without refreshing the entire page. This allows web pages to load new content dynamically, making the user experience smoother and more interactive.

6.PROJECTS

6.1 Task -1: Music Player Application

Problem Statement: Design and Develop a Music Player Application.

Abstract:

The Music Player Application is a software tool designed to offer a streamlined and interactive platform for users to listen to their favorite music. It serves as a digital media player that can handle various audio formats and provides functionalities like play, pause, forward, backward, shuffle, repeat, and volume control. The application also enables users to create playlists, browse songs, and visualize metadata like song titles, album art, and artist names. The goal of this application is to deliver a rich, intuitive music experience while ensuring ease of use and compatibility across platforms.

By integrating modern UI/UX principles, the Music Player Application is designed to be responsive, lightweight, and efficient. Whether for casual listening or creating custom playlists, this music player provides an enjoyable user experience on any device, from desktops to mobile phones. Additionally, the application will allow users to control music playback through various interactive elements such as buttons, sliders, and menus.

Key Elements:

1. User Interface (UI)
2. Audio Management
3. Playlist Management
4. Playback Controls
5. Responsive Design

Source Code:

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Music Player</title>
    <link rel="stylesheet" href="style.css">
    <script src="script.js" defer></script>
    <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.13.0/css/all.min.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
   <div class="player">
        <div class="wrapper">
            <div class="details">
                <div class="now-playing">PLAYING x OF y</div>
                <div class="track-art"></div>
                <div class="track-name">Track Name</div>
                <div class="track-artist">Track Artist</div>
            </div>

            <div class="slider_container">
                <div class="current-time">00:00</div>
                 <input type="range" min="1" max="100" value="0" class="seek_slider"
onchange="seekTo()">
                 <div class="total-duration">00:00</div>
            </div>

            <div class="slider_container">
                <i class="fa fa-volume-down"></i>
                 <input type="range" min="1" max="100" value="99" class="volume_slider"
onchange="setVolume()">
                 <i class="fa fa-volume-up"></i>
            </div>

            <div class="buttons">
                 <div class="random-track" onclick="randomTrack()">
                     <i class="fas fa-random fa-2x" title="random"></i>
                </div>
                <div class="prev-track" onclick="prevTrack()">
                     <i class="fa fa-step-backward fa-2x"></i>
                 </div>
                 <div class="playpause-track" onclick="playpauseTrack()">
                     <i class="fa fa-play-circle fa-5x"></i>
                 </div>
                 <div class="next-track" onclick="nextTrack()">
                     <i class="fa fa-step-forward fa-2x"></i>
                 </div>
                 <div class="repeat-track" onclick="repeatTrack()">
                     <i class="fa fa-repeat fa-2x" title="repeat"></i>
                 </div>
            </div>

             <div id="wave">
                 <span class="stroke"></span>
                 <span class="stroke"></span>
                 <span class="stroke"></span>
                 <span class="stroke"></span>
                 <span class="stroke"></span>
                 <span class="stroke"></span>
                 <span class="stroke"></span>
             </div>
          </div>
   </div>
</body>
</html>
```

Style.css

```
body {
    font-family: Arial, Helvetica, sans-serif;
    background-color: white;
    font-weight: bold;
}

.player {
    height: 95vh;
    display: flex;
    align-items: center;
    flex-direction: column;
    justify-content: center;
}

.wrapper {
    border: 1px solid transparent;
    padding: 30px;
    border-radius: 20px;
    background-color: #ddd;
    box-shadow: rgba(0, 0, 0, 0.3) 0px 19px 38px, rgba(0, 0, 0, 0.22) 0px 15px 12px;
}

.details {
    display: flex;
    align-items: center;
    flex-direction: column;
    justify-content: center;
}
```

```
.track-art {
   margin: 25px;
   height: 250px;
   width: 250px;
   border: 2px solid #FFFAFA;
   background-size: cover;
   background-position: center;
   border-radius: 50%;
   -moz-box-shadow: 0px 6px 5px black;
   -webkit-box-shadow: 0px 6px 5px black;
   box-shadow: 0px 6px 5px black;
   -moz-border-radius: 190px;
   -webkit-border-radius: 190px;
   border-radius: 190px;
}

.now-playing {
   font-size: 1rem;
}

.track-name {
   font-size: 2.5rem;
}

.track-artist {
   margin-top: 5px;
   font-size: 1.5rem;
}

.buttons {
   display: flex;
   flex-direction: row;
   align-items: center;
   margin-bottom: 30px;
}

.active {
   color: black;
}

.repeat-track,
.random-track,
.playpause-track,
.prev-track,
.next-track {
   padding: 25px;
   opacity: 0.8;
   transition: opacity 0.2s;
}

.repeat-track:hover,
.random-track:hover,
.playpause-track:hover,
.prev-track:hover,
.next-track:hover {
    opacity: 1.0;
}

.slider_container {
    display: flex;
    justify-content: center;
    align-items: center;
}

 .seek_slider, .volume_slider {
     -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
    height: 5px;
    background: #83A9FF;
    -webkit-transition: 0.2s;
    transition: opacity 0.2s;
}

.seek_slider::-webkit-slider-thumb,
.volume_slider::-webkit-slider-thumb {
    -webkit-appearance: none;
    -moz-appearance: none;
    appearance: none;
    width: 15px;
    height: 15px;
    background: white;
    border: 3px solid #3774FF;
    cursor: grab;
    border-radius: 100%;
}

.seek_slider:hover,
.volume_slider:hover {
    opacity: 1.0;
}

.seek_slider {
    width: 60%;
}

.volume_slider {
    width: 30%;
}

.current-time,
.total-duration {
    padding: 10px;
```

```
}

i.fa-volume-down,
i.fa-volume-up {
   padding: 10px;
}

i,
i.fa-play-circle,
i.fa-pause-circle,
i.fa-step-forward,
i.fa-step-backward,
p {
   cursor: pointer;
}

.randomActive {
   color: black;
}

.rotate {
   animation: rotation 8s infinite linear;
}

@keyframes rotation {
   from {
      transform: rotate(0deg);
   }
   to {
      transform: rotate(359deg);
   }
}

.loader {
   height: 70px;
   display: flex;
   justify-content: center;
   align-items: center;
}

.loader .stroke {
   background: #f1f1f1;
   height: 150%;
   width: 10px;
   border-radius: 50px;
   margin: 0 5px;
   animation: animate 1.4s linear infinite;
}

@keyframes animate {
   50% {
      height: 20%;
```

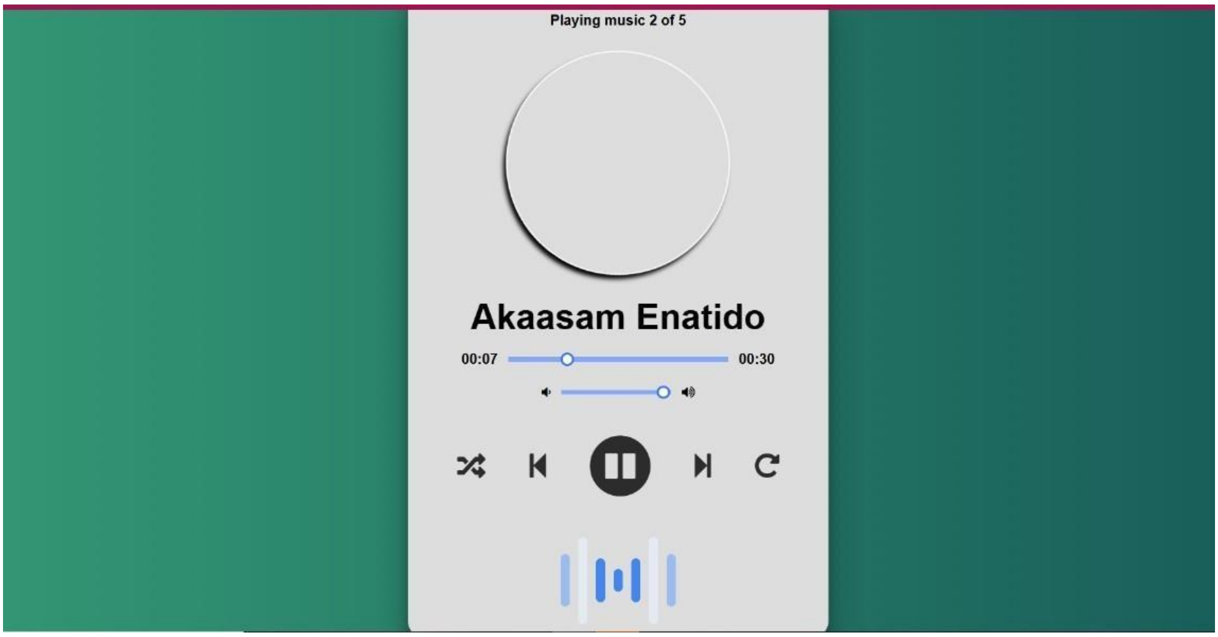
```
        background: #4286f4;
    }

    100% {
        background: #4286f4;
        height: 100%;
    }
}

.stroke:nth-child(1) {
    animation-delay: 0s;
}

.stroke:nth-child(2) {
    animation-delay: 0.3s;
}

.stroke:nth-child(3) {
    animation-delay: 0.6s;
}

.stroke:nth-child(4) {
    animation-delay: 0.9s;
}

.stroke:nth-child(5) {
    animation-delay: 0.6s;
}

.stroke:nth-child(6) {
    animation-delay: 0.3s;
}

.stroke:nth-child(7) {
    animation-delay: 0s;
}
```

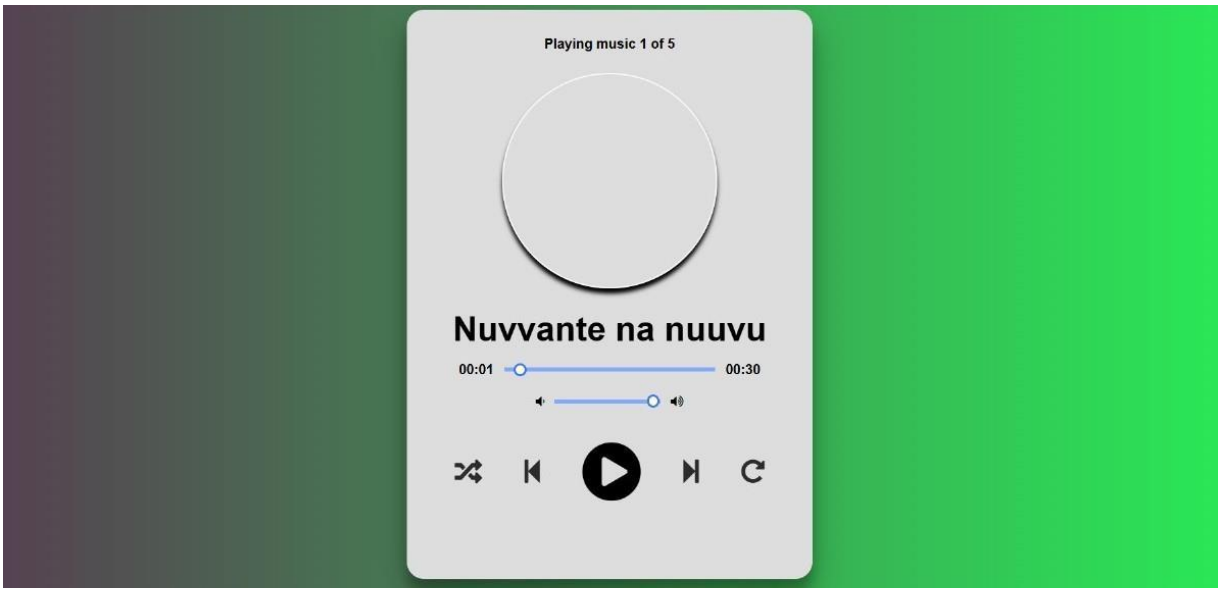
Script.js

```
let now_playing = document.querySelector('.now-playing');
let track_art = document.querySelector('.track-art');
let track_name = document.querySelector('.track-name');
let track_artist = document.querySelector('.track-artist');


let playpause_btn = document.querySelector('.playpause-track');
let next_btn = document.querySelector('.next-track');
let prev_btn = document.querySelector('.prev-track');
let seek_slider = document.querySelector('.seek_slider');
let volume_slider = document.querySelector('.volume_slider');
let curr_time = document.querySelector('.current-time');
let total_duration = document.querySelector('.total-duration');
let wave = document.getElementById('wave');
let randomIcon = document.querySelector('.fa-random');
let curr_track = document.createElement('audio');

let track_index = 0;
let isPlaying = false;
let isRandom = false;
let updateTimer;

const music_list = [
  {
      img : 'images/nani.png',
      name : 'Nuvvante na nuuvu',
      Movie : 'Krishna gadi veera prema katha,',
      music : 'music/nani.mp3'
  },
  {
      img : 'images/2.png',
      name : 'Akaasam Enatido',
      Movie : 'Nireekshana',
      music : 'music/2.mp3'
  },
  {
      img : 'images/kamini.png',
      name : 'KAMINI',
      Movie : 'Anugraheethan Antony',
      music : 'music/kamini.mp3'
  },
  {
      img : 'images/4.png',
      name : 'Manasulo Madhuve',
      Movie : 'Shakuni',
      music : 'music/4.mp3'
  },
  {
      img : 'images/5.png',
      name : 'Main Tera Boyfriend',
      Movie : 'Raabta',
      music : 'music/5.mp3'
  }
];

loadTrack(track_index);

function loadTrack(track_index){
  clearInterval(updateTimer);
  reset();

  curr_track.src = music_list[track_index].music;
  curr_track.load();

  track_art.style.backgroundImage = "url(" + music_list[track_index].img + ")";
  track_name.textContent = music_list[track_index].name;
  track_artist.textContent = music_list[track_index].artist;
  now_playing.textContent = "Playing music " + (track_index + 1) + " of " + music_list.length;

  updateTimer = setInterval(setUpdate, 1000);

  curr_track.addEventListener('ended', nextTrack);
  random_bg_color();
}

function random_bg_color(){
  let hex = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e'];
  let a;
  function populate(a){
      for(let i=0; i<6; i++){
          let x = Math.round(Math.random() * 14);
          let y = hex[x];
          a += y;
      }
      return a;
  }
  let Color1 = populate('#');
  let Color2 = populate('#');
  var angle = 'to right';

  let gradient = 'linear-gradient(' + angle + ',' + Color1 + ', ' + Color2 + ")";
  document.body.style.background = gradient;
}
function reset(){
  curr_time.textContent = "00:00";
  total_duration.textContent = "00:00";
  seek_slider.value = 0;
}
function randomTrack(){
  isRandom ? pauseRandom() : playRandom();
}
function playRandom(){
  isRandom = true;
  randomIcon.classList.add('randomActive');
}
function pauseRandom(){
  isRandom = false;
  randomIcon.classList.remove('randomActive');
}
function repeatTrack(){
  let current_index = track_index;
  loadTrack(current_index);
  playTrack();
```

```
}
function playpauseTrack(){
    isPlaying ? pauseTrack() : playTrack();
}
function playTrack(){
    curr_track.play();
    isPlaying = true;
    track_art.classList.add('rotate');
    wave.classList.add('loader');
    playpause_btn.innerHTML = '<i class="fa fa-pause-circle fa-5x"></i>';
}
function pauseTrack(){
    curr_track.pause();
    isPlaying = false;
    track_art.classList.remove('rotate');
    wave.classList.remove('loader');
    playpause_btn.innerHTML = '<i class="fa fa-play-circle fa-5x"></i>';
}
function nextTrack(){
    if(track_index < music_list.length - 1 && isRandom === false){
        track_index += 1;
    }else if(track_index < music_list.length - 1 && isRandom === true){
        let random_index = Number.parseInt(Math.random() * music_list.length);
        track_index = random_index;
    }else{
        track_index = 0;
    }
    loadTrack(track_index);
    playTrack();
}
function prevTrack(){
    if(track_index > 0){
        track_index -= 1;
    }else{
        track_index = music_list.length -1;
    }
    loadTrack(track_index);
    playTrack();
}
function seekTo(){
    let seekto = curr_track.duration * (seek_slider.value / 100);
    curr_track.currentTime = seekto;
}
function setVolume(){
    curr_track.volume = volume_slider.value / 100;
}
function setUpdate(){
    let seekPosition = 0;
    if(!isNaN(curr_track.duration)){
        seekPosition = curr_track.currentTime * (100 / curr_track.duration);
        seek_slider.value = seekPosition;

        let currentMinutes = Math.floor(curr_track.currentTime / 60);
        let currentSeconds = Math.floor(curr_track.currentTime - currentMinutes * 60);
        let durationMinutes = Math.floor(curr_track.duration / 60);
        let durationSeconds = Math.floor(curr_track.duration - durationMinutes * 60);

        if(currentSeconds < 10) {currentSeconds = "0" + currentSeconds; }
        if(durationSeconds < 10) { durationSeconds = "0" + durationSeconds; }
        if(currentMinutes < 10) {currentMinutes = "0" + currentMinutes; }
        if(durationMinutes < 10) { durationMinutes = "0" + durationMinutes; }

        curr_time.textContent = currentMinutes + ":" + currentSeconds;
        total_duration.textContent = durationMinutes + ":" + durationSeconds;
    }
}
```

Output:





Result: I have successfully implemented the Music Player Application.

6.2 Task -2: Web Application for Professional Resumes

Problem Statement: Design and Develop a Web Application for Professional Resumes.

Abstract:

The Web Application for Professional Resumes is a platform that enables users to easily create and customize their resumes online. This application provides a variety of pre-designed templates that users can choose from, allowing for flexibility in terms of layout and style. The user-friendly interface will guide individuals through the process of entering their personal details, educational background, work experience, skills, and certifications. Once the resume is created, users can download their resume in multiple formats (such as PDF or DOCX) or share it directly with employers via a link.

The goal of this application is to streamline the resume creation process by providing a convenient, efficient, and accessible tool that produces professional-quality resumes. The application will also allow users to update their resumes regularly, ensuring that they always have an up-to-date version ready for job applications. This will cater to both individuals who are creating their first resumes as well as professionals looking to keep their resumes current.

Key Elements:

1. User-Friendly Interface
2. Customizable Templates
3. Resume Sections
4. Download and Share Options
5. Cross-Platform Compatibility

Source Code:

Index.html

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8">

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <title>Home Page</title>

    <meta name="description" content="">

    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" href="main.css">

  </head>

  <body>


    <nav class = "navbar bg-white">

      <div class="container">

        <div class = "navbar-content">

          <div class = "brand-and-toggler">

            <a href = "index.html" class = "navbar-brand">

              <img src = "curriculum-vitae.png" alt = "" class = "navbar-brand-icon">

              <span class = "navbar-brand-text">Build Your <span>Resume.</span>

            </a>

            <button type = "button" class = "navbar-toggler-btn">

              <div class = "bars">
```

```
                        <div class = "bar"></div>
                        <div class = "bar"></div>
                        <div class = "bar"></div>
                    </div>
                </button>
            </div>
        </div>
    </div>
</nav>

<header class = "header bg-bright" id = "header">
    <div class = "container">
        <div class = "header-content text-center">
            <h6 class = "text-uppercase text-blue-dark fs-14 fw-6 ls-1"> Resume Builder</h6>

            <a href = "resume.html" class = "btn btn-primary text-uppercase">Create My Resume</a>
            <img src = "dublin-resume-templates.avif">
        </div>
    </div>
</header>

<div class="section-one">
    <div class="container">
        <div class = "section-one-content">
            <div class="section-one-l">
```

```
                        <img src = "visual-0c7080adf17f1f207276f613447c924f667dab34b7ac415cd7ef653172defd0b.svg">
                    </div>
                    <div class = "section-one-r text-center">
                        <h2 class = "lg-title">Use the best resume maker as your guide!</h2>
                        <p class = "text">Getting that dream job can seem like an impossible task. We're here to change that. Give yourself a real advantage with the best online resume maker: created by experts, imporved by data, trusted by millions of professionals.</p>
                        <div class = "btn-group">
                            <a href = "resume.html" class = "btn btn-primary text-uppercase">create my resume</a>
                            <a href = "#" class = "btn btn-secondary text-uppercase">watch video</a>
                        </div>
                    </div>
                </div>
            </div>
        </div>

        <div class = "section-two bg-bright">
            <div class="container">
                <div class="section-two-content">
                    <div class = "section-items">
                        <div class = "section-item">
                            <div class = "section-item-icon">
                                <img src = "feature-1-edf4481d69166ac81917d1e40e6597c8d61aa970ad44367ce78049bf830fbda5.svg" alt = "">
                            </div>
```

```
              <h5 class = "section-item-title">Make a resume that wins interviews!</h5>

              <p class = "text">Use our resume maker with its advanced creation tools to tell
professional story that engages recruiters, hiring managers and even CEOs.</p>

            </div>


            <div class = "section-item">

              <div class = "section-item-icon">

                <img src = "feature-2-
a7a471bd973c02a55d1b3f8aff578cd3c9a4c5ac4fc74423d94ecc04aef3492b.svg" alt = "">

              </div>

              <h5 class = "section-item-title">Resume writing made easy!</h5>

              <p class = "text">Resume writing has never been this effortless. Pre-generated
text, visual designs and more - all already integrated into the resume maker. Just fill in your
details.</p>

            </div>


            <div class = "section-item">

              <div class = "section-item-icon">

                <img src = "feature-3-
4e87a82f83e260488c36f8105e26f439fdc3ee5009372bb5e12d9421f32eabdd.svg" alt = "">

              </div>

              <h5 class = "section-item-title">A recruiter-tested CV maker tool</h5>

              <p class = "text">Our resume builder and its pre-generated content are tested by
recruiters and IT experts. We help your CV become truly competitive in the hiring process.</p>

            </div>

          </div>

        </div>
```

```
            </div>
        </div>


        <footer class = "footer bg-dark">
            <div class="container">
                <div class = "footer-content text-center">
                    <p class="fs-15">&copy;Copyright 2022. All Rights Reserved - 
<span>build.resume</span></p>
                </div>
            </div>
        </footer>


    </body>
</html>
```

Style.css

```
@import
url('https://fonts.googleapis.com/css2?family=Manrope:wght@200;300;400;500;600;700;800&family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

:root{
    --clr-blue: #1A91F0;
    --clr-blue-mid: #1170CD;
    --clr-blue-dark: #1A1C6A;
    --clr-white: #fff;
    --clr-bright: #EFF2F9;
    --clr-dark: #1e2532;
    --clr-black: #000;
    --clr-grey: #656e83;
    --clr-green: #084C41;
    --font-poppins: 'Poppins', sans-serif;
    --font-manrope: 'Manrope', sans-serif;;
    --transition: all 300ms ease-in-out;
}

*{
    padding: 0;
    margin: 0;
    box-sizing: border-box;
}

html{
    font-size: 10px;
}

body{
    font-size: 1.6rem;
    font-family: var(--font-poppins);
}

button{
    border: none;
    background-color: transparent;
    outline: 0;
    cursor: pointer;
    font-family: inherit;
}

img{
    width: 100%;
    display: block;
}
```

```
a{
    text-decoration: none;
}

/* fonts */
.font-poppins{font-family: var(--font-poppins);}
.font-manrope{font-family: var(--font-manrope);}

/* text colors */
.text-blue{color: var(--clr-blue);}
.text-blue-mid{color: var(--clr-blue-mid);}
.text-blue-dark{color: var(--clr-blue-dark);}
.text-bright{color: var(--clr-bright);}
.text-dark{color: var(--clr-dark);}
.text-grey{color: var(--clr-grey);}
.text-white{color: var(--clr-white);}

/* backgrounds */
.bg-blue{background-color: var(--clr-blue);}
.bg-blue-mid{background-color: var(--clr-blue-mid);}
.bg-blue-dark{background-color:  var(--clr-blue-dark);}
.bg-bright{background-color: var(--clr-bright);}
.bg-dark{background-color: var(--clr-dark);}
.bg-grey{background-color: var(--clr-grey);}
.bg-white{background-color: var(--clr-white);}
.bg-black{background-color: var(--clr-black);}
.bg-green{background-color: var(--clr-green);}

.text-center{ text-align: center;}
.text-left{text-align: left;}
.text-right{text-align: right;}
.text-uppercase{text-transform: uppercase;}
.text-lowercase{text-transform: lowercase;}
.text-capitalize{text-transform: capitalize;}
.text{
    color: var(--clr-dark);
    opacity: 0.9;
    margin: 2rem 0;
    line-height: 1.6;
}

.fw-2{font-weight: 200;}
.fw-3{font-weight: 300;}
.fw-4{font-weight: 400;}
.fw-5{font-weight: 500;}
.fw-6{font-weight: 600;}
.fw-7{font-weight: 700;}
.fw-8{font-weight: 800;}

.fs-13{font-size: 13px;}
.fs-14{font-size: 14px;}
.fs-15{font-size: 15px;}
.fs-16{font-size: 16px;}
.fs-17{font-size: 17px;}
.fs-18{font-size: 18px;}
.fs-19{font-size: 19px;}
.fs-20{font-size: 20px;}
.fs-21{font-size: 21px;}
.fs-22{font-size: 22px;}
.fs-23{font-size: 23px;}
.fs-24{font-size: 24px;}
.fs-25{font-size: 25px;}

.ls-1{letter-spacing: 1px;}
.ls-2{letter-spacing: 2px;}

.container{
    max-width: 1200px;
    margin: 0 auto;
    padding: 0 1.6rem;
```

```
}

/* bars button */
.bars{
    display: flex;
    flex-direction: column;
    justify-content: space-between;
    height: 16.5px;
    width: 25px;
}
.bars .bar{
    width: 100%;
    height: 2px;
    background-color: var(--clr-blue);
    transition: var(--transition);
}

.bars:hover .bar{
    background-color: var(--clr-dark);
}

/* buttons */
.btn{
    font-size: 14.5px;
    font-weight: 600;
    padding: 1.4rem 1.6rem;
    border-radius: 4px;
    display: inline-block;
}

.btn-primary{
    background-color: var(--clr-blue);
    color: var(--clr-white);
    border: 1px solid var(--clr-blue);
    transition: var(--transition);
```

```
}

.btn-primary:hover{
    background-color: transparent;
    color: var(--clr-dark);
    border-color:  var(--clr-grey);
}

.btn-secondary{
    background-color: transparent;
    color: var(--clr-dark);
    border: 1px solid var(--clr-grey);
    transition: var(--transition);
}

.btn-secondary:hover{
    background-color: var(--clr-blue);
    color: var(--clr-white);
    border-color:  var(--clr-blue);
}

.btn-group button:first-child, .btn-group a:first-child{
    margin-right: 1rem!important;
}

/* navbar part */
.navbar{
    height: 80px;
    display: flex;
    align-items: center;
    box-shadow: rgba(0, 0, 0, 0.08) 0px 3px 8px;
}

.navbar .container{
    width: 100%;
```

```
}

.navbar-brand{
    display: flex;
    align-items: center;
    justify-content: flex-start;
    font-size: 1.8rem;
}
.navbar-brand-text{
    color: var(--clr-dark);
    font-weight: 600;
}
.navbar-brand-text span{
    color: var(--clr-blue);
}
.navbar-brand-icon{
    width: 25px;
    margin-right: 6px;
    opacity: 0.8;
}
.brand-and-toggler{
    display: flex;
    align-items: center;
    justify-content: space-between;
}
.header{
    min-height: calc(100vh - 80px);
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
}
.header-content{
    max-width: 740px;
    margin-right: auto;
    margin-left: auto;
}
.header-content img{
    max-width: 760px;
    border-top-right-radius: 8px;
    border-top-left-radius: 8px;
    margin-top: 3.2rem;
}
.lg-title{
    margin: 1.4rem 0;
    font-size: 37px;
    line-height: 1.4;
    color: var(--clr-dark);
}
.header-content p{
    margin-bottom: 2.6rem;
    line-height: 1.6;
}


/* section one */
.section-one{
    padding: 64px 0;
    min-height: 80vh;
    display: flex;
    align-items: center;
}
.section-one-l img{
    max-width: 545px;
    margin-right: auto;
    margin-left: auto;
}
.section-one-r{
    margin-top: 4rem;
}
```

```
.section-one .btn-group{
    margin-top: 2rem;
}
.section-one-r{
    max-width: 545px;
    margin-right: auto;
    margin-left: auto;
}
.section-one-r .btn-group{
    margin-top: 3rem;
}

/* section two */
.section-two{
    padding: 64px 0;
}
.section-two .section-items{
    display: grid;
    gap: 2rem;
}

.section-two .section-item{
    max-width: 350px;
    text-align: center;
    margin-right: auto;
    margin-left: auto;
}
.section-two .section-item-icon{
    margin: 1rem 0;
}
.section-two .section-item-icon img{
    width: 80px;
    margin-right: auto;
    margin-left: auto;
```

```
}
.section-two .section-item-title{
    color: var(--clr-blue-dark);
    font-size: 1.8rem;
    font-weight: 600;
}
.section-two .text{
    margin: 0.9rem 0;
}

/* footer */
.footer{
    padding: 3rem 0;
}
.footer-content p{
    color: var(--clr-grey);
}
.footer-content p span{
    color: var(--clr-white);
}

/* media queries */
@media screen and (min-width: 768px){
    .section-two .section-items{
        grid-template-columns: repeat(2, 1fr);
    }
}

@media screen and (min-width: 992px){
    .section-one-content{
        display: grid;
        grid-template-columns: repeat(2, 1fr);
        column-gap: 3rem;
    }
    .section-one-r{
        text-align: left;
    }
    .section-two .section-items{
        grid-template-columns: repeat(3, 1fr);
    }
    .section-two .section-item{
        text-align: left;
    }
    .section-two .section-item-icon img{
        margin-left: 0;
    }
}



/* resume page */
#about-sc{
    padding: 64px 0;
}

.cv-form-row-title{
    background-color: var(--clr-dark);
    padding: 0.8rem 1.6rem;
    margin-bottom: 2rem;
}

.cv-form-row-title h3{
    color: var(--clr-white);
    font-weight: 500;
    text-transform: uppercase;
    letter-spacing: 1.5px;
    font-size: 1.7rem;
}
.cv-form-blk{
    margin: 3rem 0;
```

```
}
.cv-form-row{
    padding: 3rem 2rem 0 2rem;
    border: 1px solid rgba(0, 0, 0, 0.08);
    margin-bottom: 1rem;
    position: relative;
}
textarea{
    resize: none;
}
.form-elem{
    margin-bottom: 3rem;
    position: relative;
}
.form-label{
    display: block;
    font-weight: 600;
    font-size: 14px;
    color: var(--clr-dark);
    margin-bottom: 0.5rem;
}
.form-control{
    border-radius: none;
    border: 1px solid rgba(0, 0, 0, 0.1);
    font-size: 14px;
    padding: 0.8rem 1.6rem;
    font-family: inherit;
    width: 100%;
    outline: 0;
    transition: var(--transition);
}

.form-control:focus{
    border-color: rgba(0, 0, 0, 0.3);
}
```

```
.form-text{
    color: #ca0b00;
    font-size: 12px;
    position: absolute;
    letter-spacing: 0.5px;
    top: calc(100% + 2px);
    left: 0;
    width: 100%;
}
.cols-3, .cols-2{
    display: grid;
}
.repeater-add-btn{
    width: 25px;
    height: 25px;
    background-color: var(--clr-blue-mid);
    font-size: 1.6rem;
    color: var(--clr-white);
    margin: 1rem 0;
}
.repeater-remove-btn{
    position: absolute;
    top: 10px;
    right: 10px;
    z-index: 999;
    width: 25px;
    height: 25px;
    border-radius: 50%;
    background-color: #ca0b00;
    color: var(--clr-white);
    font-size: 1.6rem;
}

/* preview section */
.preview-cnt{
    border-radius: 5px;
    display: grid;
    grid-template-columns: 32% auto;
    box-shadow: rgba(149, 157, 165, 0.2) 0px 8px 24px;
    overflow: hidden;
}

.preview-cnt-l{
    padding: 3rem 3rem 2rem 3rem;
}
.preview-cnt-r{
    padding: 3rem 3rem 3rem 4rem;
}
.preview-cnt-l .preview-blk:nth-child(1){
    text-align: center;
}
.preview-image{
    width: 120px;
    height: 120px;
    border-radius: 50%;
    overflow: hidden;
    margin-right: auto;
    margin-left: auto;
}
.preview-image img{
    width: 100%;
    height: 100%;
    object-fit: cover;
}
.preview-item-name{
    font-size: 2.4rem;
    font-weight: 600;
    margin: 1.8rem 0;
    position: relative;
}
```

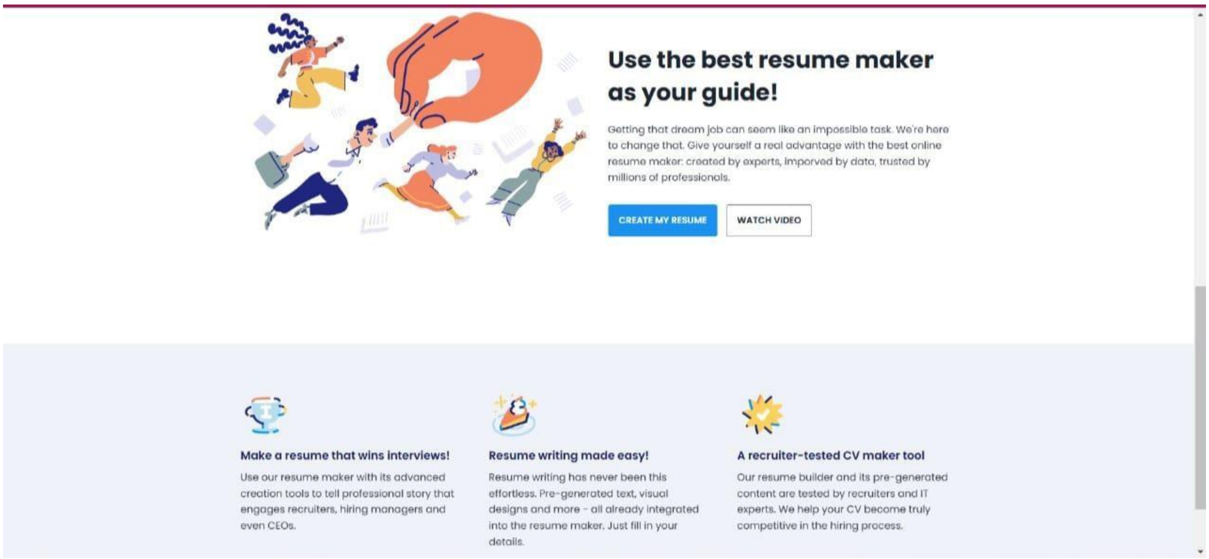
```
.preview-item-name::after{
    position: absolute;
    content: "";
    bottom: -10px;
    width: 50px;
    height: 1.5px;
    background-color: rgba(255, 255, 255, 0.5);
    left: 50%;
    transform: translateX(-50%);
}
.preview-blk{
    padding: 1rem 0;
    margin-bottom: 1rem;
}
.preview-blk-title h3{
    text-transform: uppercase;
    letter-spacing: 0.5px;
    border-bottom: 0.5px solid rgba(0, 0, 0, 0.08);
    padding-bottom: 0.5rem;
}
.preview-blk-title{
    margin-bottom: 1rem;
}
.preview-blk-list .preview-item{
    font-size: 1.5rem;
    margin-bottom: 0.2rem;
    opacity: 0.95;
}
.preview-cnt-r .preview-blk-title{
    color: var(--clr-dark);
}
.preview-cnt-r .preview-blk-list .preview-item{
    margin-top: 1.8rem;
}
```

```
.achievements-items.preview-blk-list .preview-item span:first-child,
.educations-items.preview-blk-list .preview-item span:first-child,
.experiences-items.preview-blk-list .preview-item span:first-child{
    display: block;
    font-weight: 600;
    margin-bottom: 1rem;
    background-color: rgba(0, 0, 0, 0.03);
}

.educations-items.preview-blk-list .preview-item span:nth-child(2),
.experiences-items.preview-blk-list .preview-item span:nth-child(2){
    font-weight: 600;
    margin-right: 1rem;
}

.educations-items.preview-blk-list .preview-item span:nth-child(3),
.experiences-items.preview-blk-list .preview-item span:nth-child(3){
    font-style: italic;
    margin-right: 1rem;
}

.educations-items.preview-blk-list .preview-item span:nth-child(4),
.educations-items.preview-blk-list .preview-item span:nth-child(5),
.experiences-items.preview-blk-list .preview-item span:nth-child(4),
.experiences-items.preview-blk-list .preview-item span:nth-child(5){
    margin-right: 1rem;
    background-color: var(--clr-green);
    color: var(--clr-white);
    padding: 0 1rem;
    border-radius: 0.6rem;
}

.educations-items.preview-blk-list .preview-item span:nth-child(6),
.experiences-items.preview-blk-list .preview-item span:nth-child(6){
    font-size: 13.5px;
    display: block;
    opacity: 0.8;
    margin-top: 1rem;
}
.projects-items.preview-blk-list .preview-item span{
    display: block;
}

@media screen and (min-width: 768px){
    .cols-3{
        grid-template-columns: repeat(3, 1fr);
        column-gap: 2rem;
    }
    .cols-2{
        grid-template-columns: repeat(2, 1fr);
        column-gap: 2rem;
    }
}

@media screen and (min-width: 992px){
    .cv-form-row{
        padding: 3rem 3rem 0rem 3rem;
    }
    .cols-3{
        grid-template-columns: repeat(3, 1fr);
    }
}

.print-btn-sc{
    margin: 2rem 0 6rem 0;
}

/* print section */
@media print{
    body *{
        visibility: hidden;
    }

    .non_print_area{
        display: none;
    }

    .print_area, .print_area *{
        visibility: visible;
    }

    .print_area{
        width: 100%;
        position: absolute;
        left: 0;
        top: 0;
        overflow: hidden;
    }
}
```

Script.js

```
// form repeater

$(document).ready(function(){

    $('.repeater').repeater({

        initEmpty: false,

        defaultValues: {

            'text-input': ''

        },

        show:function(){

            $(this).slideDown();
        },
        hide: function(deleteElement){
            $(this).slideUp(deleteElement);
            setTimeout(() => {
                generateCV();
            }, 500);
        },
        isFirstItemUndeletable: true
    })
})
```

<u>Output:</u>



Use the best resume maker as your guide!

Getting that dream job can seem like an impossible task. We're here to change that. Give yourself a real advantage with the best online resume maker: created by experts, imporved by data, trusted by millions of professionals.

Make a resume that wins interviews!

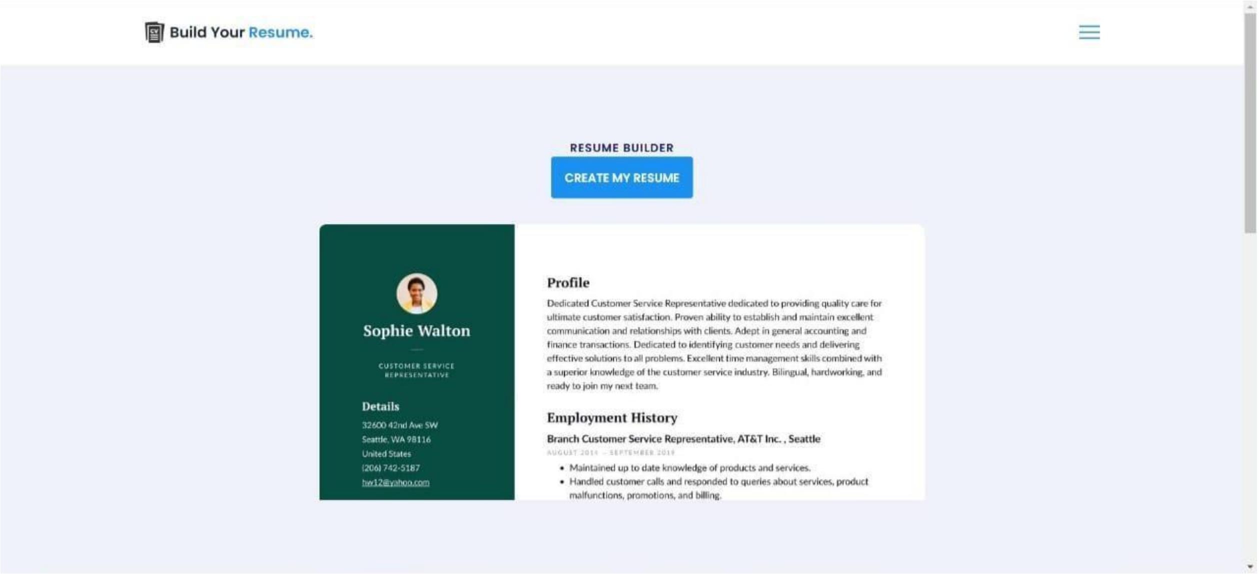
Use our resume maker with its advanced creation tools to tell professional story that engages recruiters, hiring managers and even CEOs.

Resume writing made easy!

Resume writing has never been this effortless. Pre-generated text, visual designs and more - all already integrated into the resume maker. Just fill in your details.

A recruiter-tested CV maker tool

Our resume builder and its pre-generated content are tested by recruiters and IT experts. We help your CV become truly competitive in the hiring process.


Build Your Resume.

RESUME BUILDER

CREATE MY RESUME

Sophie Walton

CUSTOMER SERVICE REPRESENTATIVE

Details

32600 42nd Ave SW
Seattle, WA 98116
United States
(206) 742-5187
hw12@yahoo.com

Profile

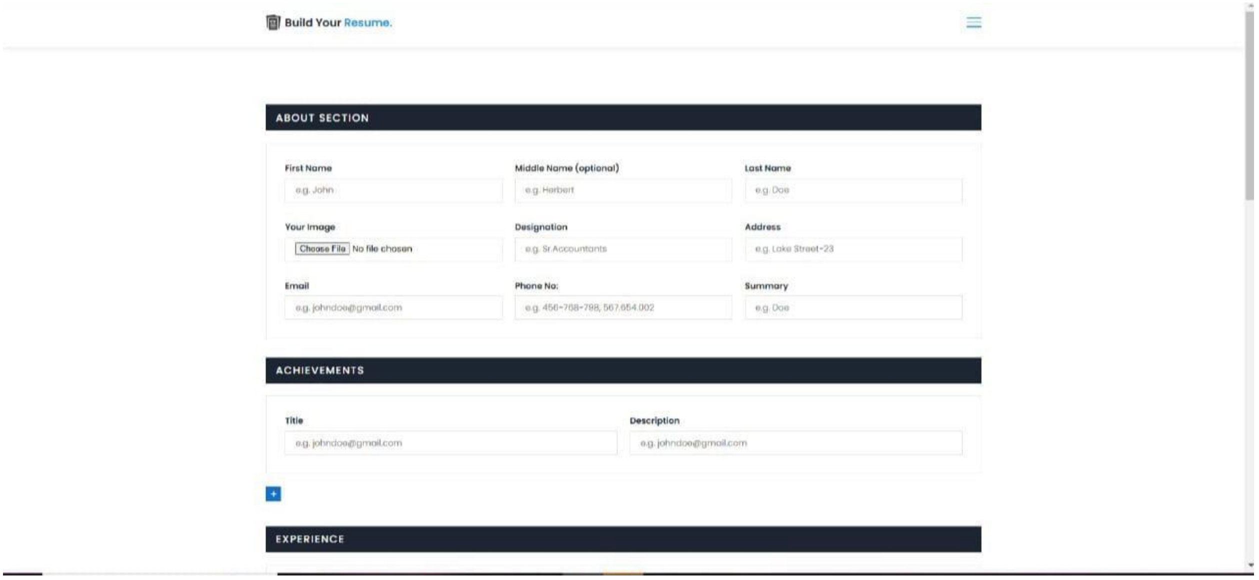
Dedicated Customer Service Representative dedicated to providing quality care for ultimate customer satisfaction. Proven ability to establish and maintain excellent communication and relationships with clients. Adept in general accounting and finance transactions. Dedicated to identifying customer needs and delivering effective solutions to all problems. Excellent time management skills combined with a superior knowledge of the customer service industry. Bilingual, hardworking, and ready to join my next team.

Employment History

Branch Customer Service Representative, AT&T Inc. , Seattle

AUGUST 2014 – SEPTEMBER 2019

- Maintained up to date knowledge of products and services.
- Handled customer calls and responded to queries about services, product malfunctions, promotions, and billing.


Build Your Resume.

ABOUT SECTION

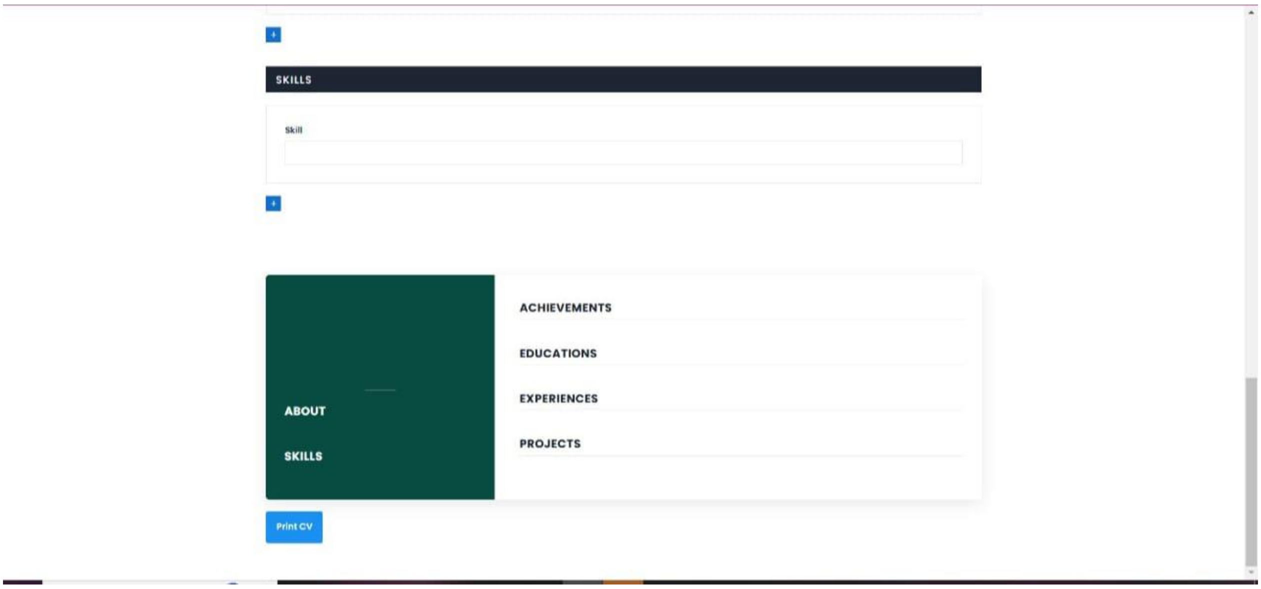
First Name: e.g. John
Middle Name (optional): e.g. Herbert
Last Name: e.g. Doe
Your Image: Choose File No file chosen
Designation: e.g. Sr.Accountants
Address: e.g. Lake Street-23
Email: e.g. johndoe@gmail.com
Phone No: e.g. 456-768-798, 567.654.002
Summary: e.g. Doe

ACHIEVEMENTS

Title: e.g. johndoe@gmail.com
Description: e.g. johndoe@gmail.com

+

EXPERIENCE


+

SKILLS

Skill

+

ABOUT

SKILLS

ACHIEVEMENTS

EDUCATIONS

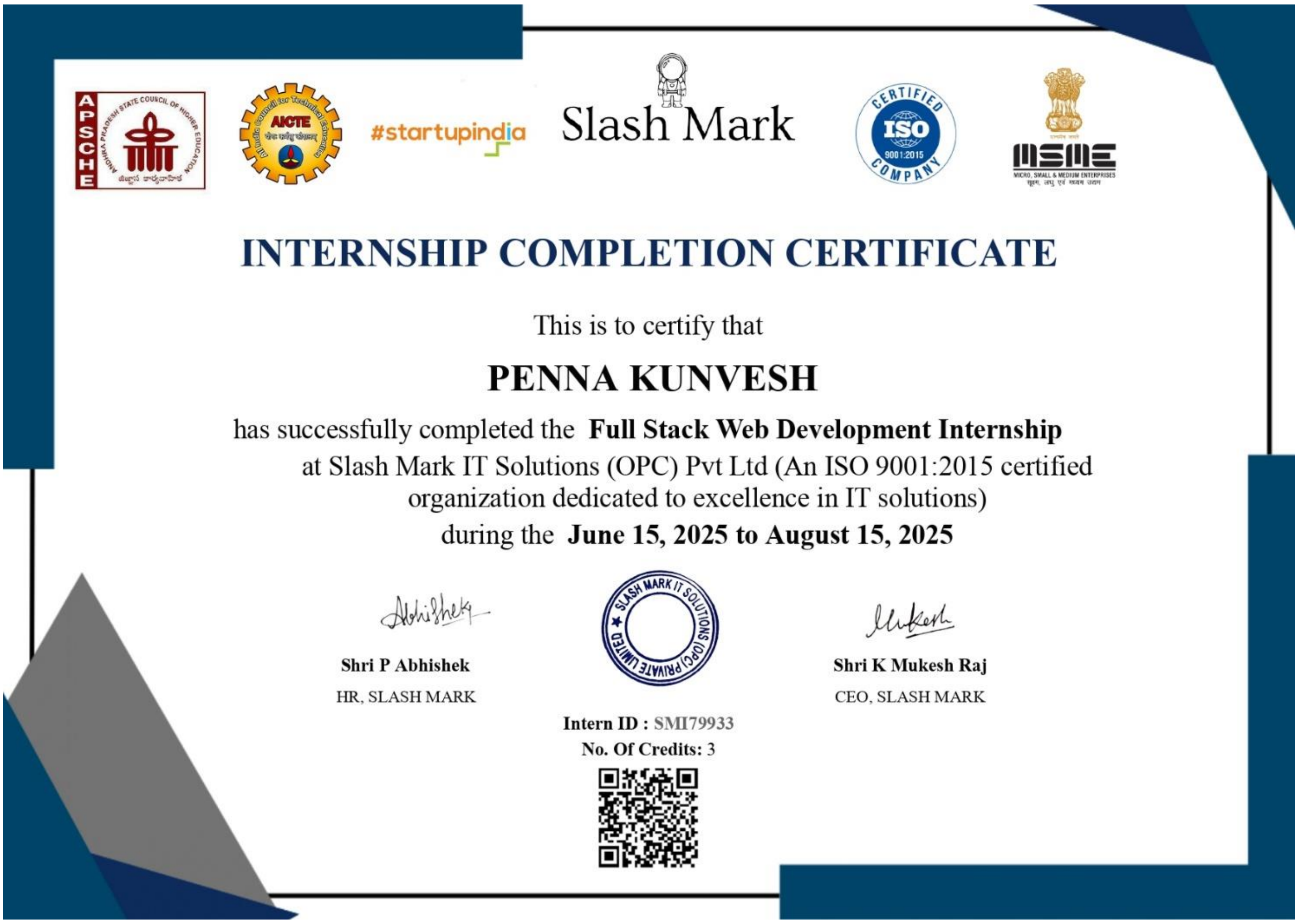
EXPERIENCES

PROJECTS

Print CV

Result: I have successfully implemented the Web Application for Professional Resumes.

7. CERTIFICATE



#startupindia

Slash Mark

CERTIFIED ISO 9001:2015 COMPANY

MSME

MICRO, SMALL & MEDIUM ENTERPRISES

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that

PENNA KUNVESH

has successfully completed the **Full Stack Web Development Internship** at Slash Mark IT Solutions (OPC) Pvt Ltd (An ISO 9001:2015 certified organization dedicated to excellence in IT solutions) during the **June 15, 2025 to August 15, 2025**

Abhishek

Shri P Abhishek
HR, SLASH MARK

SLASH MARK IT SOLUTIONS (OPC) PRIVATE LIMITED

Mukesh

Shri K Mukesh Raj
CEO, SLASH MARK

Intern ID : SMI79933
No. Of Credits: 3

8. CONCLUSION

In conclusion, web development plays a critical role in building dynamic, interactive, and user-friendly websites and applications. Through the use of key technologies like HTML, CSS, JavaScript, and various frameworks, developers can create websites that are visually appealing, functional, and responsive across different devices. The integration of front-end and back-end technologies enables seamless user experiences, while the continuous evolution of web standards ensures that web applications remain fast, secure, and scalable.

Web development not only facilitates effective communication and interaction online but also provides endless opportunities for innovation, creativity, and problem-solving. As the demand for web-based solutions continues to grow, mastering web development has become essential for anyone looking to make an impact in the digital world. Ultimately, the ability to build efficient, well-designed websites will continue to empower individuals and businesses to connect with audiences, streamline operations, and achieve their goals in the ever-evolving web ecosystem.