

SCIENTIFIC CALCULATOR

A Summer Internship Report submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY In COMPUTER SCIENCE and ENGINEERING

Submitted by:
DESU RAMYA
22A91A05H3



Department Of Computer Science and Engineering

ADITYA ENGINEERING COLLEGE (A)

Approved by AICTE, Permanently affiliated to JNTUK & Accredited by NAAC with 'A++' Grade

Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956

Aditya Nagar, ADB Road –Surampalem 533437, E.G. Dist., A.P.,

2023-2024.

ADITYA ENGINEERING COLLEGE (A)

Approved by AICTE, Permanently Affiliated to JNTUK & Accredited by NAAC with 'A++' Grade
Recognized by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Aditya Nagar, ADB Road - Surampalem – 533437, E.G.Dist., A.P.,

Department Of Computer Science and Engineering



CERTIFICATE

This is to certify that the Internship report entitled “**SCIENTIFIC CALCULATOR**” is being submitted by

DESU RAMYA

(22A91A05H3)

In partial fulfillment of the requirements for award of the B.Tech degree in Computer Science and Engineering for the academic year 2022-2023.

Internship Coordinator

Guide Name, Qualification

Designation

Department of CSE

Head of the Department

Dr. A.Vanathi, M.E., Ph.D.,

Associate Professor

Department of CSE

DECLARATION

I hereby declare that the project entitled “**SCIENTIFIC CALCULATOR**” is a genuine project. This work has been submitted to the **ADITYA ENGINEERING COLLEGE**, Surampalem, permanently affiliated to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** in partial fulfillment of the **B.Tech** degree. We further declare that this project work has not been submitted in full or part of the award of any degree of this or any other educational institutions.

by

DESU RAMYA

(22A91A05H3)

Internship Completion Certificate

ACKNOWLEDGEMENT

First I would like to thank the Director of Organization Name, Hyderabad for giving me the opportunity to do an internship within the organization. I also would like all the people that worked along with me in Organization Name,, Hyderabad with their patience and openness they created an enjoyable working environment.

It is with immense pleasure that we would like to express our indebted gratitude to our Intenship coordinator ----- who has guided us a lot and encouraged us in every step of the intern project work, **his/her** valuable moral support and guidance throughout the Intern project helped us to a greater extent.

Our deepest thanks to **Dr. A.Vanathi, Associate Professor & Head of the Department** for inspiring us all the way and for arranging all the facilities and resources needed for our project.

We wish to thank **Dr. S. Rama Sree, Professor** in CSE and Dean(Academics) for her support and suggestions during our project work.

We owe our sincere gratitude to **Dr. M.Sreenivasa Reddy, Principal** for providing a great support and for giving us the opportunity of doing the project.

We are thankful to our **College Management** for providing all the facilities in time to us for completion of our project.

Not to forget, **Faculty, Lab Technicians, non-teaching staff and our friends** who have directly or indirectly helped and supported us in completing our project in time.

Abstract

Learning Objectives/Internship Objectives

- Internships are generally thought of to be reserved for college students looking to gain experience in a particular field. However, a wide array of people can benefit from Training Internships in order to receive real world experience and develop their skills.
- An objective for this position should emphasize the skills you already possess in the area and your interest in learning more
- Internships are utilized in a number of different career fields, including architecture, engineering, healthcare, economics, advertising and many more.
- Some internships are used to allow individuals to perform scientific research while others are specifically designed to allow people to gain first-hand experience working.
- Utilizing internships is a great way to build your resume and develop skills that can be emphasized in your resume for future jobs. When you are applying for a Training Internship, make sure to highlight any special skills or talents that can make you stand apart from the rest of the applicants so that you have an improved chance of landing the position.

WEEKLY OVERVIEW OF INTERNSHIP ACTIVITIES

1st WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	03/6/24	Monday	Made a plan for the project and finalized domain and the project
	04/6/24	Tuesday	Brief Introduction on what to learn regarding project and made a note of topics to be learnt
	05/6/24	Wednesday	Learnt about basic html tags and made a basic webpage
	06/6/24	Thursday	Learnt about classes and id attributes
	07/6/24	Friday	Learnt the creation of buttons, textfields using html
	08/6/24	Saturday	Learnt about linking of button, textfield

2nd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	10/6/24	Monday	Learnt about styling using css and linking html & css
	11/6/24	Tuesday	Working with some advanced topics of css like background, linear gradient., hover
	12/6/24	Wednesday	Working with all different components learnt
	13/6/24	Thursday	Learnt about javascript and implemented basic functions
	14/6/24	Friday	Learnt about Math functions
	15/6/24	Saturday	Implemented Math functions using Html

3rd WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	17/6/24	Monday	Holiday
	18/6/24	Tuesday	Started doing simple calculator project
	19/6/24	Wednesday	Incorprated styling to above webpage created
	20/6/24	Thursday	Implemented functionality for the calculator and made it work properly
	21/6/24	Friday	Started implementing scientific calculator and made all the keys of calculator
	22/6/24	Saturday	Added styling to webpage

4th WEEK	DATE	DAY	NAME OF THE TOPIC/MODULE COMPLETED
	24/6/24	Monday	Implemented functionality for the project
	25/6/24	Tuesday	Checking and verifying entire project
	26/6/24	Wednesday	Uploading it to github
	27/6/24	Thursday	Made abstract for documentation of the project
	28/6/24	Friday	Made entire documentation and ppt
	29/6/24	Saturday	Submission of project

INDEX

SNo.		Contents	Page No.
1.		History of html, css, javascript	01
2.		Introduction to scientific calculator	02
3.		Introduction to HTML	03
4.		Introduction to CSS	04
5.		Introduction to JAVASCRIPT	05
6.		Components Used in project	06-08
	6.1	HTML tags	06
	6.2	CSS properties	07
	6.3	Functionality	08
7		Source codes	09-12
	7.1.	HTML source code	09-10
	7.2.	CSS Source code	11
	7.3.	Javascript source code	12
8.		Output	13-14
9.		Github links of the project	14
10.		Conclusion	15

HISTORY OF HTML, CSS, JAVASCRIPT

HTML (HyperText Markup Language)

- **1989-1990:** Tim Berners-Lee, a physicist at CERN, proposes a project to create a system for sharing documents. He writes the first HTML and the first web browser, WorldWideWeb.
- **1991:** The first public description of HTML is made available on the internet.
- **1993:** The first widely used version of HTML, HTML 1.0, is released. It includes basic elements such as paragraphs, headings, lists, and links.
- **1995:** HTML 2.0 is published as an IETF (Internet Engineering Task Force) standard. This version includes features like forms and tables.
- **1997:** HTML 3.2 is published by the W3C (World Wide Web Consortium). This version adds support for applets, text flow around images, and more.
- **1999:** HTML 4.01 is released, introducing features like improved accessibility and internationalization.
- **2014:** HTML5 becomes a W3C recommendation, adding new elements for multimedia, better error handling, and enhanced APIs for web applications.

CSS (Cascading Style Sheets)

- **1994:** Håkon Wium Lie proposes the concept of CSS while working with Tim Berners-Lee at CERN.
- **1996:** The first CSS specification, CSS1, is published by the W3C. It introduces fundamental concepts like selectors, properties, and cascading.
- **1998:** CSS2 is released, adding support for positioning, z-index, media types, and more.
- **1999:** Work begins on CSS3, which is divided into modules to allow for more flexible development and implementation.
- **2011-Present:** Various CSS3 modules are published as W3C recommendations, introducing features like transitions, animations, and flexible box layouts.

JavaScript

- **1995:** Brendan Eich, working at Netscape, creates the first version of JavaScript in just ten days. It is initially called Mocha, then LiveScript, and finally JavaScript.
- **1996:** JavaScript is officially launched with Netscape Navigator 2.0. Microsoft introduces a similar language, JScript, with Internet Explorer 3.0.
- **1997:** The first version of ECMAScript (ES1), the standardized scripting language based on JavaScript, is published by Ecma International.
- **1999:** ES3 is released, adding regular expressions, better string handling, and other improvements.
- **2009:** ES5 introduces features like strict mode, JSON support, and improved array methods.
- **2015:** ES6 (also known as ECMAScript 2015 or ES2015) is released, bringing significant updates like classes, modules, arrow functions, and promises.
- **2016-Present:** Annual releases of ECMAScript introduce features such as async/await (ES2017), dynamic import (ES2020), and optional chaining (ES2020).

INTRODUCTION TO SCIENTIFIC CALCULATOR

A scientific calculator is an advanced type of calculator designed to handle mathematical functions beyond basic arithmetic operations. Unlike standard calculators, which are limited to addition, subtraction, multiplication, and division, scientific calculators can perform a wide range of functions including trigonometric, logarithmic, and exponential calculations. They are essential tools for students, engineers, scientists, and anyone who needs to perform complex mathematical computations.

Key Features of a Scientific Calculator

1. **Basic Arithmetic Operations:** Addition, subtraction, multiplication, and division.
2. **Trigonometric Functions:** Sine (sin), cosine (cos), tangent (tan), and their inverses.
3. **Logarithmic and Exponential Functions:** Logarithms (log, ln) and exponential functions (e^x).
4. **Roots and Powers:** Square roots, cube roots, and other nth roots; powers of numbers.
5. **Memory Functions:** Storing and recalling values for complex calculations.
6. **Parentheses:** Handling complex expressions by managing the order of operations.
7. **Constants:** Access to mathematical constants like π (pi) and e (Euler's number).

INTRODUCTION TO HTML

- HTML, or Hyper Text Markup Language, is the standard markup language used to create and design documents on the World Wide Web.
- HTML structures web pages and web applications by defining the content and layout of a webpage, allowing web browsers to render text, images, links, and other multimedia elements in a structured and visually appealing way.
- All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.
- The `<!DOCTYPE>` declaration represents the document type, and helps browsers to display web pages correctly.
- It must only appear once, at the top of the page (before any HTML tags).
- The `<!DOCTYPE>` declaration is not case sensitive.
- The HTML document itself begins with `<html>` and ends with `</html>`.
- The visible part of the HTML document is between `<body>` and `</body>`.
- The HTML **element** is everything from the start tag to the end tag:
- Syntax: `<tagname>Content goes here...</tagname>`

INTRODUCTION TO CSS

- CSS, or Cascading Style Sheets, is a style sheet language used for describing the presentation of a document written in HTML or XML.
- CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
- It enables the separation of content from presentation, improving content accessibility and flexibility.
- CSS is a powerful language that brings web pages to life, making them visually appealing and enhancing user experience.
- Developers can create responsive, accessible, and beautifully designed websites.
- There are three ways of inserting a style sheet:
 - External CSS :
 - Internal CSS
 - Inline CSS
- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.
- With an external style sheet, you can change the look of an entire website by changing just one file!
- Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.
- An inline style may be used to apply a unique style for a single element.
- To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

INTRODUCTION TO JAVASCRIPT

- JavaScript is a high-level, interpreted programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.
- It is primarily used to create interactive effects within web browsers, enhancing the user experience by allowing developers to create dynamic, responsive, and interactive web applications.
- **JavaScript** to program the behavior of web pages
- In HTML, JavaScript code is inserted between `<script>` and `</script>` tags.
- A JavaScript function is a block of JavaScript code, that can be executed when "called" for.
- Scripts can be placed in the `<body>`, or in the `<head>` section of an HTML page, or in both.
- External scripts are practical when the same code is used in many different web pages.
- JavaScript files have the file extension **.js**.
- To use an external script, put the name of the script file in the src (source) attribute of a `<script>` tag.

COMPONENTS USED IN PROJECT

HTMLTAGS

- The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.
- The <link> tag defines the relationship between the current document and an external resource.
- The <body> tag defines the document's body
- The <center> tag was used in HTML4 to center-align text.
- The <h1> to <h6> tags are used to define HTML headings.
- <h1> defines the most important heading. <h6> defines the least important heading.
- The <script> tag is used to embed a client-side script (JavaScript).
- The <script> element either contains scripting statements, or it points to an external script file through the src attribute.
- The <div> tag defines a division or a section in an HTML document.
- The <div> tag is used as a container for HTML elements - which is then styled with CSS or manipulated with JavaScript.
- The <div> tag is easily styled by using the class or id attribute.
- The <input> tag specifies an input field where the user can enter data.
- The <button> tag defines a clickable button.
- Inside a <button> element you can put text (and tags like <i>, , ,
, , etc.). That is not possible with a button created with the input element
- The onclick event occurs when the user clicks on an HTML element.
- The HTML class attribute is used to specify a class for an HTML element. Multiple HTML elements can share the same class.
- The HTML id attribute is used to specify a unique id for an HTML element. You cannot have more than one element with the same id in an HTML document.

CSS PROPERTIES

- If the style is to be added to a tag itself where ever the tag is used in html document we need to give style as

```
Tagname{  
Property: value  
....  
}
```

- If style is added to a particular class then we represent as

```
.classname{  
Property: value  
....  
}
```

- If the style is added to an id then it is as

```
#idname{  
Property: value  
...  
}
```

- **Background image** property assigns background image to the webpage
- **Linear gradient** is used when the color is to be a combination or mixture of colors.
- **Color** property assigns color to the text.
- **Width and height** defines width and height of divison respectively.
- **Padding** is used to create space around an element's content, inside of any defined borders.
- The **display** property specifies the display behavior (the type of rendering box) of an element.
- **Grid** displays an element as a block level grid container.
- The **grid-template-columns** property specifies the number (and the widths) of columns in a grid layout.
- **Button: hover** is used to animate button when hover on it.

FUNCTIONALITY

- The display function is used to display the value of the number when button is clicked.
- It takes the value as an argument and adds the value to the already existing value.
- The backspace function is used to remove recently added value to the text field.
- The clearsc function is to set the text field to original value i.e. empty value.
- The solve function is used to solve the expression and reflect the changes of result in the textfield.
- The eval() is used to evaluate the expression in the text field.
- Math.sin() is used to calculate sin values.
- Math.cos() is used to calculate cos values.
- Math.tan() is used to calculate tan values.
- Math.log() is used to find log to the base e value.
- Math.sqrt() is used to find squareroot of a number.
- Math.log10 is used to find log to the base 10 value.
- Math.PI displays constant pi value i.e.3.14.
- Math.E displays constant e value i.e.2.718.

HTML SOURCE CODE

```
<html>
<head>
  <title>Scientific Calculator</title>
  <link rel="stylesheet" type="text/css" href="ScientficCalculator.css">
</head>
<body>
  <center>
    <h1> SCIENTIFIC CALCULATOR</h1>
    <script type="text/javascript" src="ScientficCalculator.js"></script>
    <div class="ScientficCalculator">
      <input type="text" id="result">
      <div class="buttons">
        <button onclick="clearsc()">C</button>
        <button onclick="backspace()">DEL</button>
        <button onclick="display('/')">/</button>
        <button onclick="display('*')">X</button>
        <button onclick="display('7')">7</button>
        <button onclick="display('8')">8</button>
        <button onclick="display('9')">9</button>
        <button onclick="display('-')">-</button>
        <button onclick="display('4')">4</button>
        <button onclick="display('5')">5</button>
        <button onclick="display('6')">6</button>
        <button onclick="display('+)">+</button>
        <button onclick="display('1)">1</button>
        <button onclick="display('2)">2</button>
        <button onclick="display('3)">3</button>
        <button onclick="display('.')">.</button>
        <button onclick="display('0)">0</button>
        <button onclick="display('(')">(</button>
        <button onclick="display(')'">)</button>
        <button onclick="solve()">=</button>
        <button onclick="display('Math.sin()'">sin</button>
        <button onclick="display('Math.cos()'">cos</button>
        <button onclick="display('Math.tan()'">tan</button>
        <button onclick="display('Math.log()'">log</button>
        <button onclick="display('Math.sqrt()'">sqrt</button>
      </div>
    </div>
  </center>
</body>
</html>
```

`<button onclick="display('Math.log10()')>log10</button>`

`<button onclick="display('Math.PI')>pi</button>`

`<button onclick="display('Math.E')>e</button>`

`</div>`

`</div>`

`</center>`

`</body>`

`</html>`

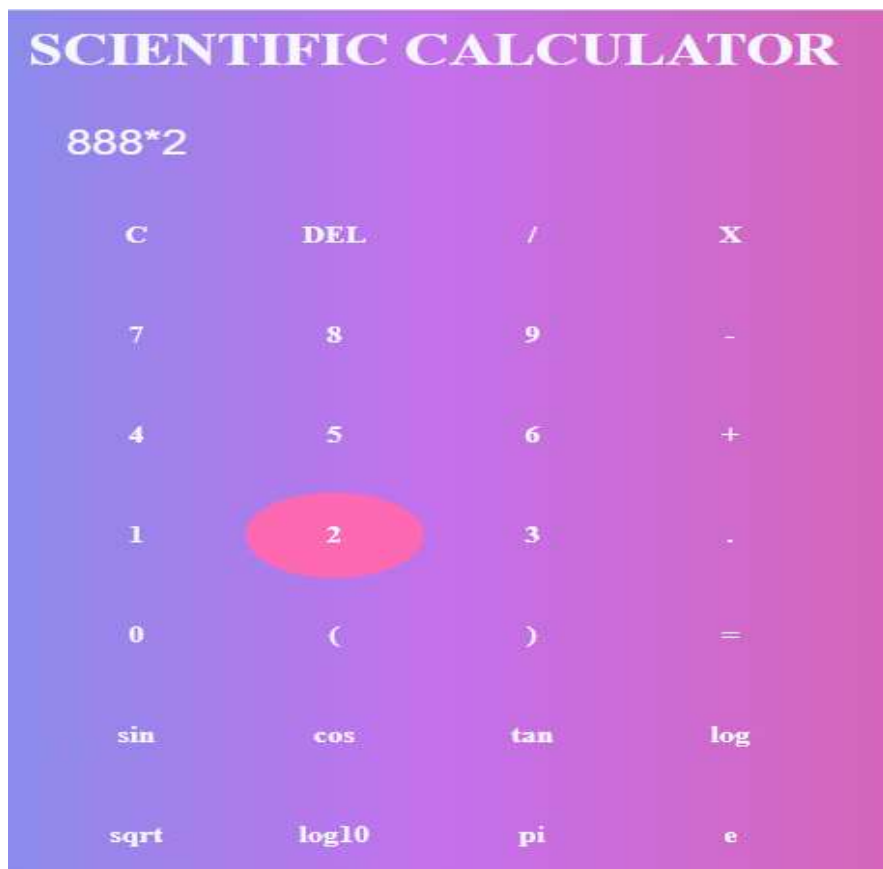
CSS SOURCE CODE

```
body{
    background-image:linear-gradient(to right,#12c2e9 , #c471ed , #f64f59) ;
    justify-content: center;
}
h1{
    color: ghostwhite;
}
#result{
    width: 400px;
    padding: 10px;
    font-size: 1.5em;
    border: none;
    margin-bottom: 10px;
    color: ghostwhite;
    background-color: transparent;
}
.buttons{
    display: grid;
    grid-template-columns: repeat(4,1fr);
    gap: 10px;
    width: 400px;
}
button{
    padding: 20px;
    border: none;
    background-color: transparent;
    color: ghostwhite;
    font-family: Times New Roman;
    font-size: 16px;
    font-weight: bold;
    border-radius: 50%;
}
button:hover{
    background-color: hotpink;
}
```

JAVASCRIPT SOURCE CODE

```
function display(val){
    document.getElementById('result').value += val;
}
function backspace() {
    let d=document.getElementById('result');
    d.value=d.value.slice(0,-1);
}
function clearsc(){
    document.getElementById('result').value="";
}
function solve(){
    let x = document.getElementById('result').value
    try{
        let y = eval(x);
        document.getElementById('result').value = y
    }
    catch(error){
        x='Error';
    }
}
```

OUTPUT





GITHUB LINKS

HTML CODE:

<https://github.com/DesuRamya/javaScript/blob/main/project1/ScientficCalculator.html>

CSS CODE:

<https://github.com/DesuRamya/javaScript/blob/main/project1/ScientficCalculator.css>

JAVASCRIPT CODE:

<https://github.com/DesuRamya/javaScript/blob/main/project1/ScientficCalculator.js>

OUTPUT VIDEO:

<https://github.com/DesuRamya/javaScript/blob/main/project1/Scientific%20Calculator%20-%20Google%20Chrome%202024-06-25%2019-54-32.mp4>

CONCLUSION

In conclusion, the development of a scientific calculator using HTML, CSS, and JavaScript has been a rewarding and educational experience. This project has provided valuable insights into the integration of front-end technologies to create a functional and user-friendly application.

Key Takeaways:

1. **HTML:** The structure of the calculator was crafted using HTML, providing a clear and organized framework for the interface. Each button, display screen, and functional component was meticulously placed to ensure an intuitive layout.
2. **CSS:** CSS was employed to style the calculator, enhancing its visual appeal and usability. Through the use of responsive design techniques, the calculator maintains its functionality and aesthetic across various devices and screen sizes.
3. **JavaScript:** The core logic and interactivity of the calculator were implemented using JavaScript. This involved writing functions to handle complex mathematical operations, managing user input, and dynamically updating the display. Error handling and edge case considerations were also integral to ensure robust performance.

Achievements:

- **Functionality:** The calculator supports a wide range of scientific operations, including basic arithmetic, trigonometric functions, logarithms, and power calculations. This makes it a versatile tool for users needing advanced mathematical capabilities.
- **User Experience:** Emphasis was placed on creating a seamless and intuitive user experience. The layout is straightforward, with clear labeling and responsive feedback to user inputs.
- **Learning and Growth:** This project has deepened our understanding of web development technologies and their practical applications. It has also honed our problem-solving skills and ability to debug and optimize code effectively.

Future Improvements:

While the current version of the scientific calculator is fully functional, there are always opportunities for further enhancement. Potential future improvements could include:

- **Additional Features:** Incorporating more advanced mathematical functions and constants.
- **User Customization:** Allowing users to customize the calculator's appearance and settings.
- **Performance Optimization:** Further refining the code to enhance performance and responsiveness.
- **Accessibility:** Improving accessibility features to ensure the calculator is usable by a wider audience, including those with disabilities.

In summary, this project has successfully demonstrated the capabilities of HTML, CSS, and JavaScript in building a complex, interactive application. It serves as a solid foundation for further exploration and development in the field of web-based tools and applications.

