**CLIENT SIDE FORM DESIGN AND VALIDATION**

**AIM:**

To use CLIENT SIDE FORM DESIGN AND VALIDATION

**PROCEDURE:**

The basic formatting tags and their explanation and usage in html, are as follows,

|  |  |  |
| --- | --- | --- |
| **S.NO** | **HTML TAGS** | **DESCRIPTION** |
| 1 | <html> | The html tag in HTML is used to define the root of HTML and XHTML documents. |
| 2 | <a herf=” “> …</a> | create a hyperlink on the webpage. |
| **3** | **<button type = “button”>** | define the clickable button |
| **4** | **<code>** | define the piece of computer code. |
| **5** | **<div>** | make divisions of content in the web page |
| **6** | **<footer>** | define a footer of HTML document |
| **7** | **<head>** | define the head portion of the document which contains information related to the document. |
| **8** | **<img src=”” alt=”” width=”” height=””>** | add the image in HTML |
| **9** | **<label>** | provide a usability improvement for mouse users. |
| **10** | **<meta attribute-name=”value”>** | it provides important information about a document. |
| **11** | **<em>** | used to indicate the structural meaning of a block of text. |
| **12** | **<script>** | The script tag in HTML is used to define the client-side script. |
| **13** | **<section>** | defines the section of documents such as chapters, headers, footers, or any other sections. |
| **14** | **<span>** | The HTML span element is a generic inline container for inline elements and content. |
| **15** | **<table>… </table>** | HTML Table is an arrangement of data in rows and columns, or possibly in a more complex structure. |
| **16** | **<video src=”” controls> </video>** | HTML5 Video, along with knowing the different ways to add the videos to the HTML page. |
| **17** | **<link>** | Defines the relationship between a document and an external resource |
| **18** | **<meter>** | Defines a scalar measurement within a known range (a gauge) |
| **19** | **<p>** | Defines a paragraph |
| **20** | **<svg>** | Defines a container for SVG graphics |
| **21** | **<li>** | Defines a list item |
| **22** | **<ol>** | Defines an ordered list |
| **23** | **<ul>** | Defines and unordered list |
| **24** | **<input>** | Defines an input control |
| **25** | **<var>** | Defines a variable |

**SOURCE CODE :**

**LOGIN FORM**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>Experiment 0</title>

<link rel="stylesheet" href="styles.css" />

</head>

<body>

<div class="wrapper">

<div class="title">

<h2 class="fugaz-one-regular">Login To Overflow</h2>

<hr />

<h3 class="roboto">Fill the below form to login</h3>

<div class="Oauth">

<button class="inter signin">

<img src="./src/google.svg" class="logo" />

Sign in with Apple

</button>

<button class="inter signin">

<img src="./src/apple.svg" class="logo" />

Sign in with Apple

</button>

</div>

<hr />

<h3 class="roboto">Email</h3>

<div class="inputContainer">

<input type="email" name="Password" id="Email" class="inter"/>

</div>

<h3 class="roboto">Password</h3>

<input type="password" name="Password" id="Password" class="inter"/>

<a class="forgotPassword roboto" href="./forgetPassword"

>Forget Password?</a>

<br />

<div class="buttonWrapper">

<button class="roboto submit">Submit</button>

</div>

</div>

</div>

</body>

</html>

@import url('https://fonts.googleapis.com/css2?family=Fugaz+One&family=Inter:ital,opsz,wght@0,14..32,100..900;1,14..32,100..900&family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap');

body {

background: tomato;

}

h2 {

font-size: 1.7em;

}

.wrapper {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.title {

border: 3px solid black;

padding: 2em;

border-radius: 10px;

background-color: rgb(0, 0, 0);

color: aliceblue;

box-shadow: 0px 0px 1000px 2px #fbff00;

}

.Oauth {

display: flex;

gap: 5px

}

input {

border-radius: 5px;

height: 2em;

width: 78%;

border: none;

background: url("./src/email2.png") no-repeat 10px center;

background-size: 20px 20px;

padding: 10px 40px;

background-color: white;

}

.forgotPassword {

display: block;

padding: 10px;

color: aliceblue;

}

.submit {

height: 2.3em;

border-radius: 10px;

border: none;

width: 80%;

font-weight: 500;

font-size: medium;

}

.inter {

font-family: "Inter", serif;

font-optical-sizing: auto;

font-weight: 300;

font-style: normal;

}

.roboto {

font-family: "Roboto", serif;

font-optical-sizing: auto;

font-weight: 300;

font-style: normal;

}

.fugaz-one-regular {

font-family: "Fugaz One", serif;

font-weight: 400;

font-style: normal;

}

.signin {

font-size: 1em;

padding: 10px;

margin-bottom: 4px;

margin-right: 2px;

display: flex;

align-items: center;

gap: 10px;

border-radius: 6px;

border: none;

}

.buttonWrapper {

display: flex;

justify-content: center;

}

.logo {

height: 30px;

width: 30px;

}

#Password {

background: url("./src/password2.png") no-repeat 10px center;

background-size: 20px 20px;

padding: 10px 40px;

padding: 10px 40px;

background-color: white;

}

button {

transition: box-shadow 0.3s ease-out;

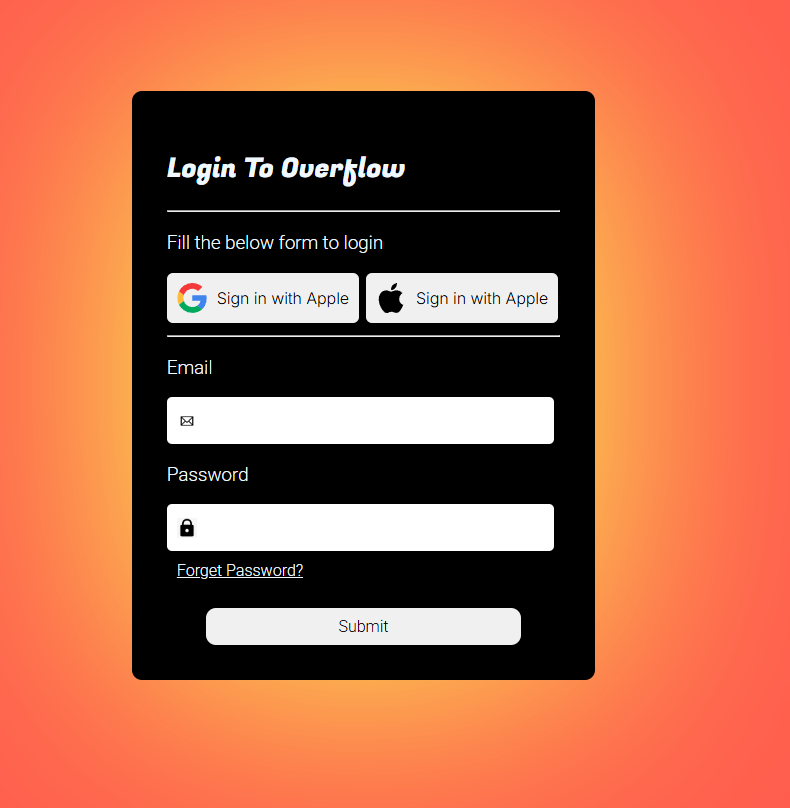
}

button:hover {

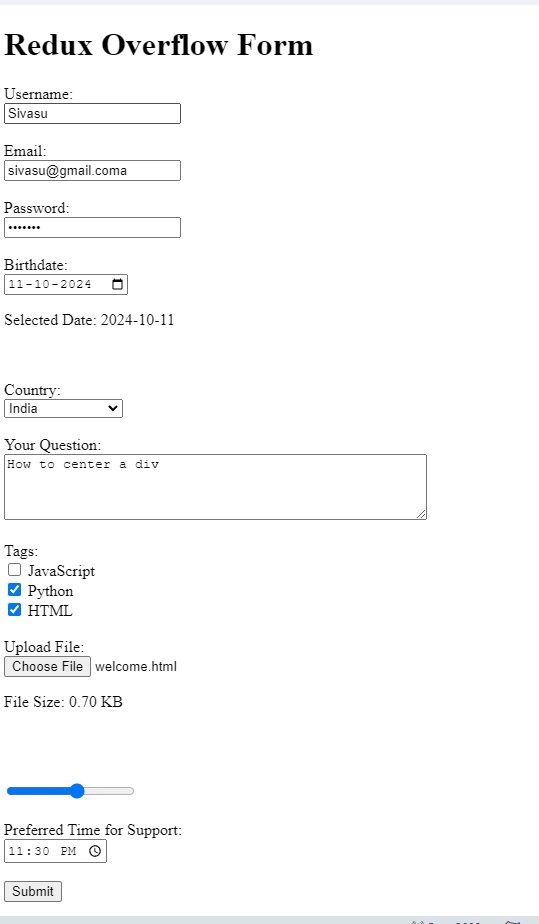
box-shadow: 0 0 15px #fbff00

}

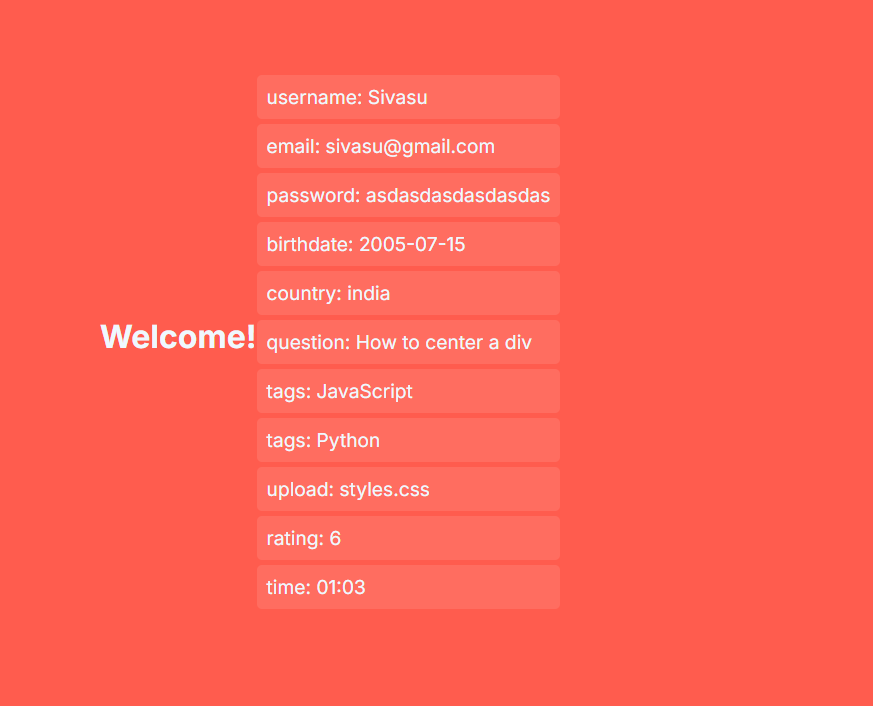
**OUTPUT 1 :**

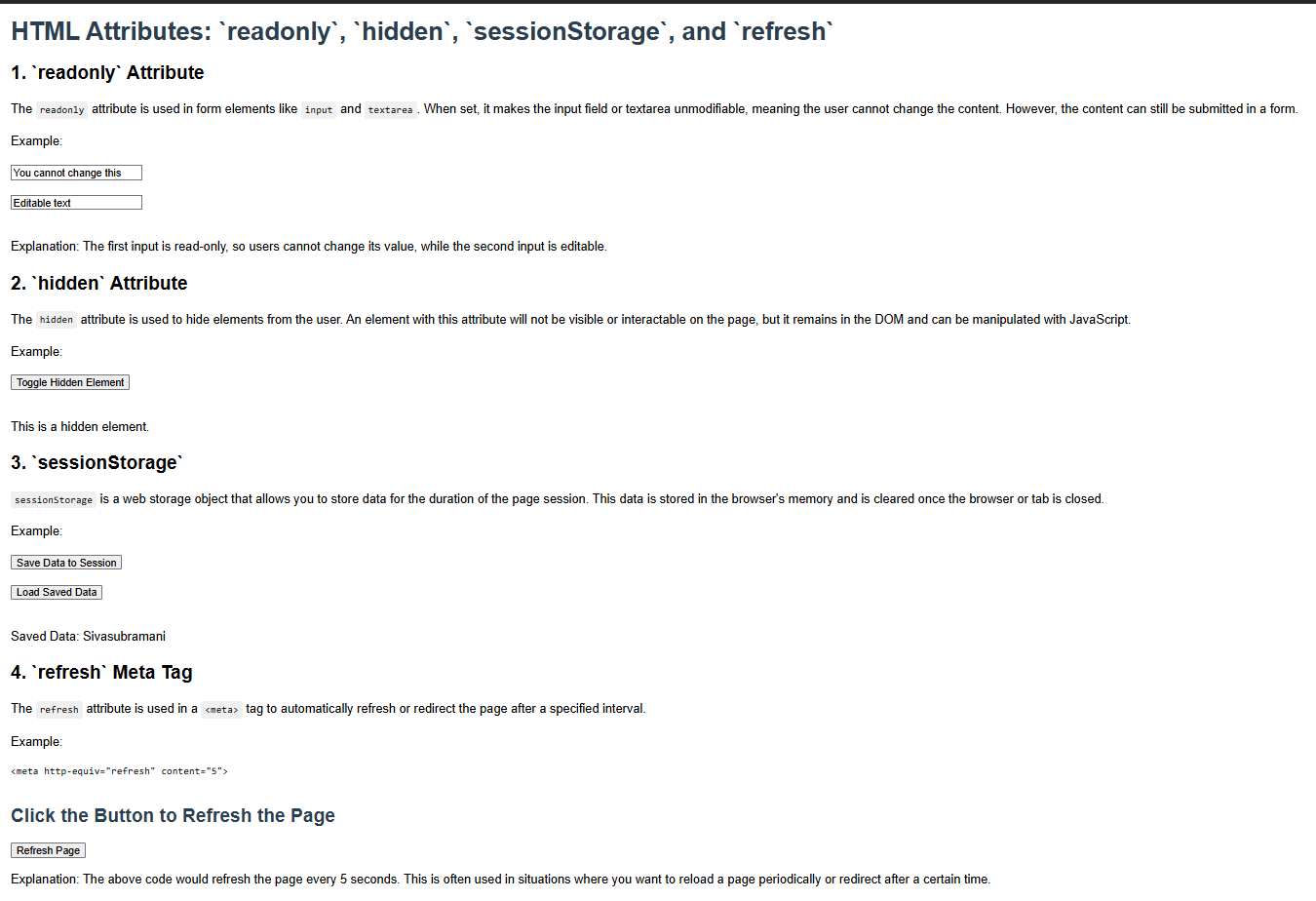


**OUTPUT 2 :**



**GET METHOD URL:**





**RESULT :**

**The basic formatting tags have been successfully implemented, the purposes and studied and the functionalities are observed.**