# MINOR PROJECT

## Interactive User Registration Form

Creating an interactive user registration form for our website. The form to be visually appealing, user-friendly, and responsive to different screen sizes.

### ALGORITHM

The code below represents an HTML document with embedded CSS and JavaScript. It creates a webpage with two sections: a welcome page and a user registration page. Below is an algorithmic breakdown of the code:

**HTML Structure:**

The document starts with the HTML5 doctype declaration.

The language is set to English.

The <head> section includes metadata and the title of the webpage.

The <style> tag contains CSS rules for styling the webpage.

The <body> section includes two main div elements: one for the welcome page and one for the registration page.

The welcome page contains a video background, a title, a description, and a registration button.

The registration page is initially hidden and includes a form for user registration with various input fields.

**CSS Styling:**

Styles are applied to the body, welcome page, and registration page.

Background colors, font styles, and layout properties are defined.

The form elements are styled for a clean and user-friendly interface.

**JavaScript Logic:**

Two JavaScript functions are defined within the <script> tag at the end of the body.

showRegistrationPage(): This function is called when the "Register Now" button is clicked on the welcome page. It hides the welcome page and displays the registration page.

registerUser(event): This function is called when the registration form is submitted. It prevents the default form submission behavior, performs registration logic (currently showing an alert), and optionally redirects the user or performs additional actions.

**Registration Form:**

The registration form includes fields for first name, last name, email, password, confirm password, date of birth, and gender.

A checkbox for agreeing to terms and conditions is included.

The form has a submit button triggering the registerUser function.

**Success Message:**

A success message div is present within the registration page but initially set to display:none.

This div becomes visible upon successful registration (currently displayed through an alert).

**Video Background:**

The welcome page has a video background (motionbg3.mp4), set to autoplay, loop, and muted.

If the browser does not support the video tag, a fallback message is displayed.

**Overall Flow:**

The initial state shows the welcome page.

Clicking the "Register Now" button triggers the showRegistrationPage function, hiding the welcome page and displaying the registration page.

Filling out the registration form and submitting it triggers the registerUser function, showing an alert for successful registration.

It's important to note that the actual registration logic (communicating with a server, storing user data, etc.) is not implemented in the provided code and would need to be added based on the specific requirements and server capabilities.

**CODE:**

Minor.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Vault of Codes - User Registration</title>
  <style>
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      margin: 0;
      padding: 0;
      height: 100vh;
      overflow: hidden;
      background-color: #f4f4f4;
    }

    .welcome-page {
      position: relative;
      height: 100vh;
  }

      display: flex;
      flex-direction: column;
      justify-content: center;
      align-items: center;
      color: #fff;
      text-align: center;
      overflow: hidden;
    }
    .video-background {
      position: fixed;
      top: 0;
      left: 0;
      width: 100%;
      height: 100%;
      object-fit: cover;
      z-index: -1;
    }
    .registration-page {
      background-image: url("C:\\Users\\rajit\\OneDrive\\Documents\\webdev-assignments\\bgimage5.png");
      background-size: cover;
      display: none;
      justify-content: center;
      align-items: center;
      height: 100vh;
      background-color: #f4f4f4;
      margin: 0;
      overflow: auto; /* Make the container scrollable */
    }
```

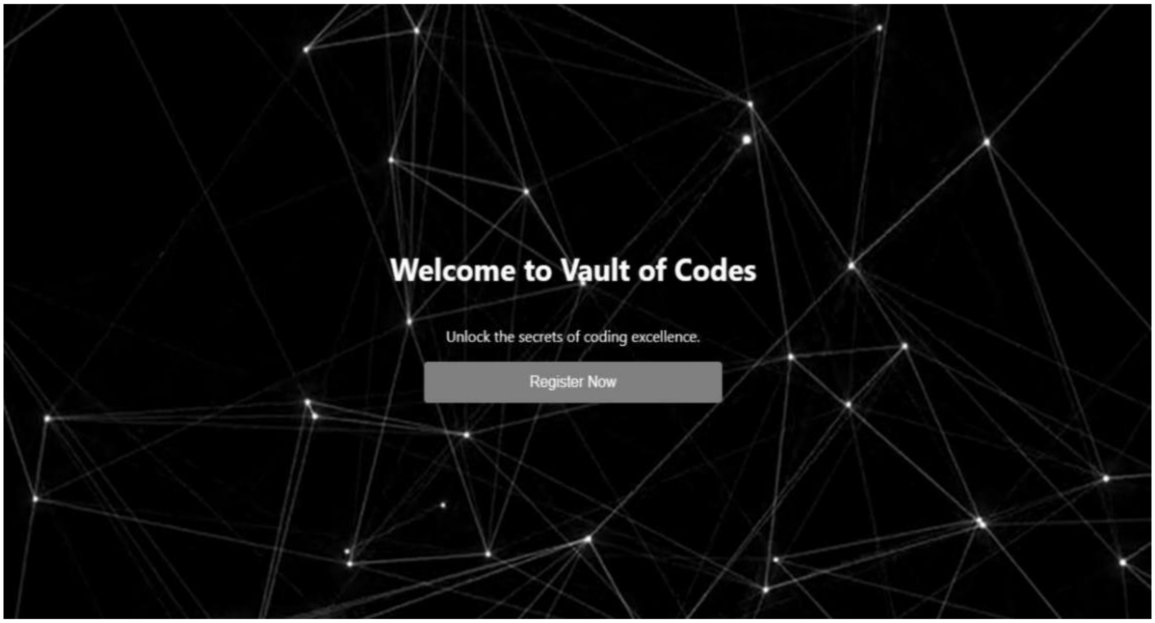
```
form {
   background-color: #fff;
   padding: 30px;
   border-radius: 8px;
   box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
   width: 100%;
   max-width: 400px;
   box-sizing: border-box;
   margin: auto;
}
h2 {
   text-align: center;
   color: #333;
}
label {
   display: block;
   margin-bottom: 8px;
   font-weight: bold;
   color: #333; }
input, select {
   width: 100%;
   padding: 10px;
   margin-bottom: 10px;
   box-sizing: border-box;
   border: 1px solid #ccc;
   border-radius: 4px;
   font-size: 16px;
}
button {
```

```
            background-color: gray;
            color: #fff;
            padding: 12px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
            font-size: 16px;
            width: 25%;
            box-sizing: border-box;
        }
        button:hover {
            background-color: gray;
        }
    </style>
</head>
<body>
<div class="welcome-page">
    <video class="video-background" autoplay loop muted>
        <source src="C:\Users\rajit\OneDrive\Desktop\motionbg3.mp4"
type="video/mp4">
        Your browser does not support the video tag.
    </video>
    <h1>Welcome to Vault of Codes</h1>
    <p>Unlock the secrets of coding excellence.</p>
    <button onclick="showRegistrationPage()">Register Now</button>
</div>
<div class="registration-page">
    <form id="registrationForm" onsubmit="registerUser(event)">
        <h2>User Registration Form</h2>
        <label for="firstlName">First Name:</label>
        <input type="text" id="firstName" name="firstName" required>
        <label for="lastName">Last Name:</label>
        <input type="text" id="lastName" name="lastName" required>
        <label for="email">Email Address:</label>
        <input type="email" id="email" name="email" required>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password" required>
        <label for="confirmPassword">Confirm Password:</label>
        <input type="password" id="confirmPassword" name="confirmPassword" required>
        <label for="dob">Date of Birth:</label>
        <input type="date" id="dob" name="dob" required>

        <label>Gender:</label>
        <div>
            <input type="radio" id="male" name="gender" value="male" required>
            <label for="male">Male</label>
            <input type="radio" id="female" name="gender" value="female" required>
            <label for="female">Female</label></div>
        <div>
            <input type="checkbox" id="termsAndConditions" required>
            <label for="termsAndConditions">I agree to the Terms and Conditions</label></div>
        <button type="submit">Register</button>
    </form>
    <div class="success-message" style="display: none;">
        <p>You have successfully registered!</p>
    </div></div>
<script>
```

```
  function showRegistrationPage() {
    document.querySelector('.welcome-page').style.display = 'none';
    document.querySelector('.registration-page').style.display = 'flex';
  }
  function registerUser(event) {
    event.preventDefault();
    alert('You have successfully registered!'); }
</script>
</body>
</html>
```

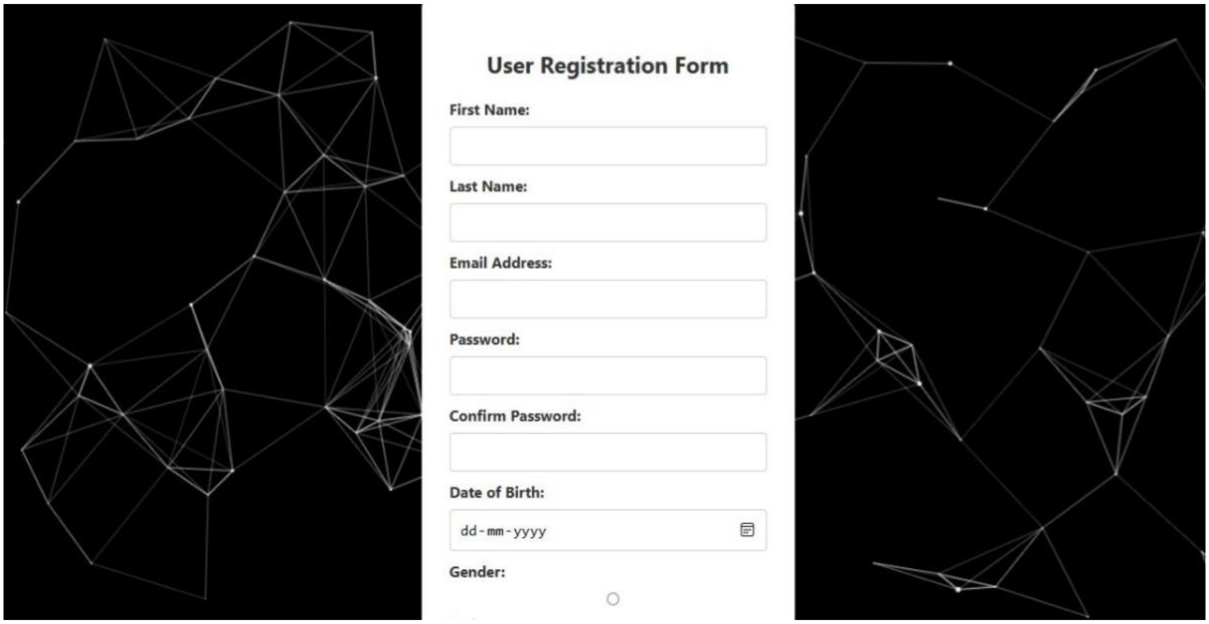
**OUTPUT**:



Welcome to Vault of Codes

Unlock the secrets of coding excellence.

This consists of a motion background such that it displays as animated GIFs



User Registration Form

First Name:

Last Name:

Email Address:

Password:

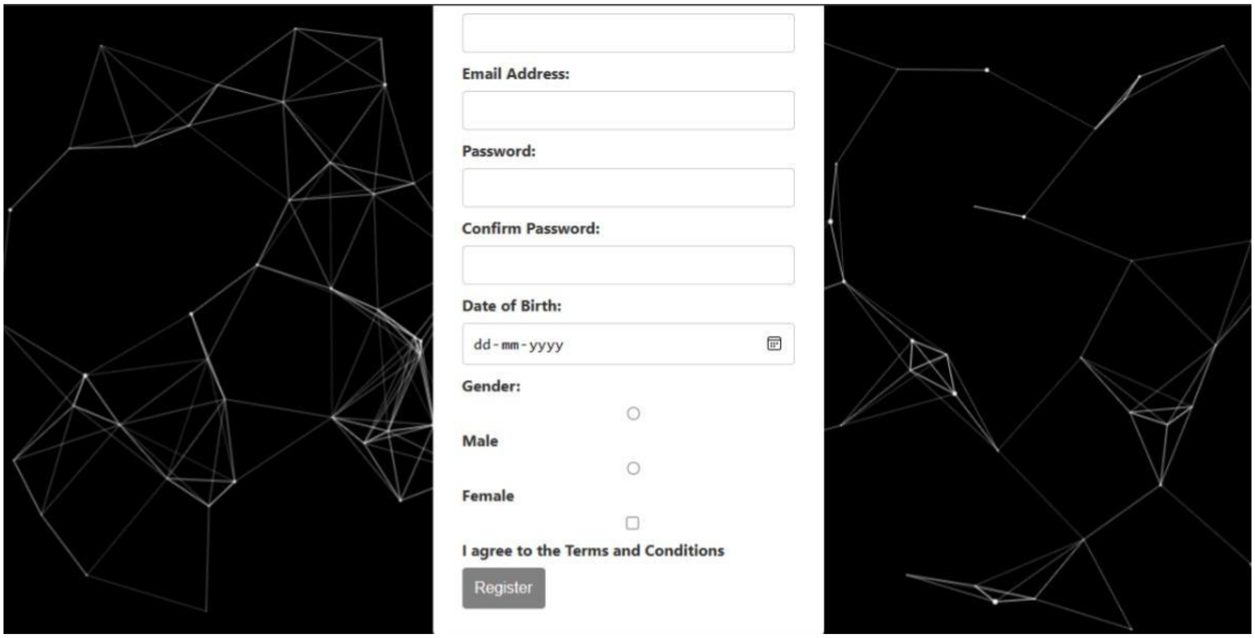
Confirm Password:

Date of Birth:

dd-mm-yyyy

Gender:

○



Email Address:

Password:

Confirm Password:

Date of Birth:

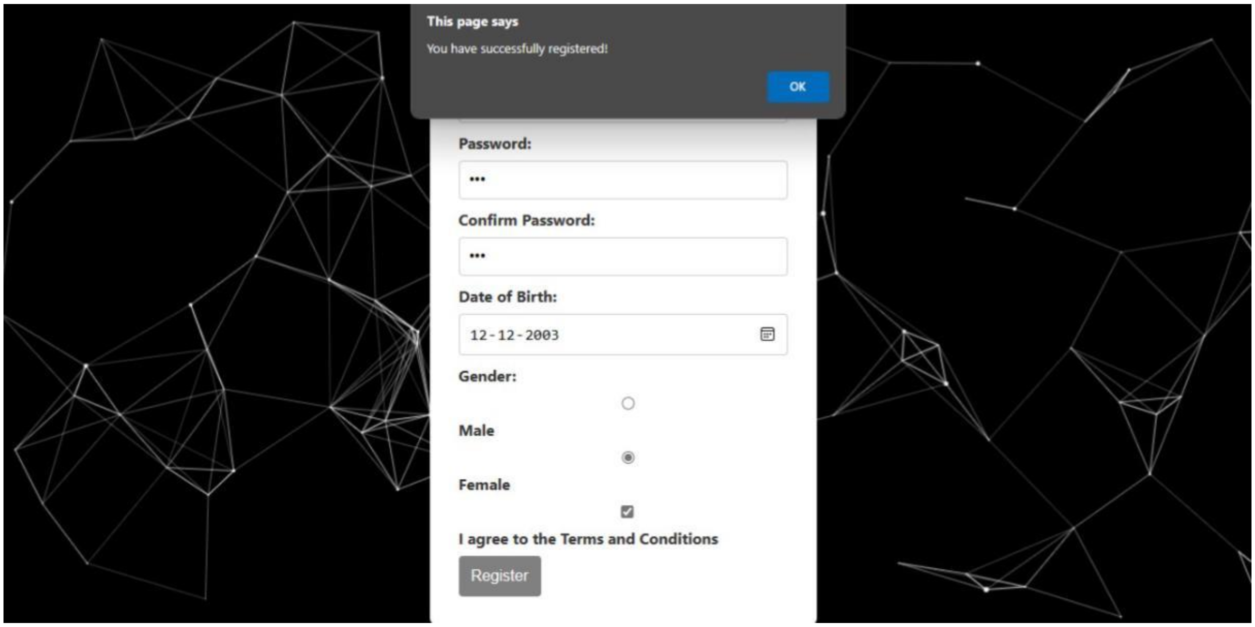
dd-mm-yyyy

Gender:

○ Male

○ Female

□ I agree to the Terms and Conditions



This page says

You have successfully registered!

Password:

•••

Confirm Password:

•••

Date of Birth:

12-12-2003

Gender:

○ Male

● Female

☑ I agree to the Terms and Conditions

Atlast the pop window will be displayed saying You have successfully registered

This is about the interactive user-friendly registration form.

# MAJOR PROJECT

## TEMPERATURE CONVERTOR

## About the project

### Introduction:

A temperature converter is a tool that allows users to convert temperatures between different units, such as Celsius, Fahrenheit, and Kelvin. It is a practical and commonly used application in various fields, including meteorology, cooking, and scientific research. This converter is typically implemented using a combination of HTML for structure, CSS for styling, and JavaScript for dynamic functionality.

### Working:

The temperature converter typically consists of input fields for the user to enter a temperature value and a dropdown menu to select the unit of the entered temperature. Users can then choose the desired output unit, and upon triggering a conversion action (e.g., clicking a button), the JavaScript code calculates and displays the converted temperature. The logic behind the conversion is based on the mathematical formulas that relate the different temperature scales.

### Uses:

- **Daily Life:** Individuals can use it for everyday tasks such as cooking, weather monitoring, and outdoor activities.
- **Science and Education:** Students and researchers can utilize the temperature converter for academic purposes in physics, chemistry, and meteorology.
- **Professional Applications:** Engineers, scientists, and professionals in various fields may use it for specific temperature conversions in their work.

**Advantages:**

- **User-Friendly:** The interface is typically simple and intuitive, making it easy for users to input data and obtain results.
- **Accessibility:** Being a web-based tool, it can be accessed from any device with a web browser, promoting widespread use.
- **Real-time Conversion:** JavaScript enables dynamic and real-time conversion without the need for page refresh.

**Disadvantages:**

- **Browser Dependency:** The functionality may depend on the user's browser and its compatibility with JavaScript.
- **Limited Features:** Some converters might lack advanced features or support for less common temperature units.
- **Privacy Concerns:** When used online, there may be concerns about data privacy, especially if sensitive temperature data is involved.

**Future Scope:**

The future scope for a temperature converter implemented using HTML, CSS, and JavaScript could involve:

- Enhanced Functionality: Adding support for more temperature units or advanced features.
- Responsive Design: Improving the user interface for better adaptability on different devices and screen sizes.
- Integration with APIs: Incorporating external APIs for real-time temperature data or additional functionalities.
- Localization: Offering support for multiple languages and region-specific temperature scales.
- Offline Functionality: Implementing a progressive web app (PWA) to allow users to use the converter even when offline.

**CODE:**

**Index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
  <link rel="stylesheet" href="style.css">
  <title>Temperature Converter</title>
</head>

<body>
  <div class="container">
    <div class="row justify-content-md-center">
      <div class="col-md-6 text-center">
        <h1>Temperature Converter</h1>
        <div id="celsius">
          <input type="number" class="form-control form-control-lg form-padding" id="input-celsius" placeholder="celsius">
        </div>

        <div id="fahrenheit">
          <input type="number" class="form-control form-control-lg form-padding" id="input-fahrenheit" placeholder="fahrenheit">
        </div>
        <div id="kelvin">
          <input type="number" class="form-control form-control-lg form-padding" id="input-kelvin" placeholder="kelvin">
        </div>
      </div>
    </div>
    <footer>
          <p>By Rajitha</p>
          <p><a href="https://github.com/itsellej" target="_blank"><img class="icons" src="https://res.cloudinary.com/itsellej/image/upload/v1533400136/rock-paper-scissors/github-icon.png" alt="Github icon"></a>
            <a href="https://codepen.io/itsellej/" target="_blank"><img class="icons" src="https://res.cloudinary.com/itsellej/image/upload/v1533400135/rock-paper-scissors/codepen-icon.png" alt="Codepen icon"></a></p>
          <p><em>Credit - icons from <a href="https://pngtree.com/">pngtree</a></em></p>
        </footer>
  </div>
  <script src="main.js" charset="utf-8"></script>
</body>
</html>
```

**Style.css**

```
body{
  margin-top: 60px;
  text-align:center;
  background-image: url("C:\\Users\\rajit\\OneDrive\\Desktop\\bgtemperatue.jpg");
  background-size: cover;
}
h1 {
  padding-bottom: 0.5rem;
}
input[type=number] {
  cursor: pointer;
}
#input-celsius{
  background-color: #52DDCF;
  font-size: 2.5rem;
}
#input-fahrenheit{
  background-color: #16BAC5;
  font-size: 2.5rem;
}
#input-kelvin{
  background-color: #42a0d9;
  font-size: 2.5rem;
}

.form-padding {
  padding: 2rem 1rem;
}
::-webkit-input-placeholder {
  color: #FFFFFF !important;
}
::-moz-placeholder {
  color: #FFFFFF !important;
}
:-ms-input-placeholder {
  color: #FFFFFF !important;
}
```

```
:-moz-placeholder {
  color: #FFFFFF !important;
}
footer{
  padding-top: 1.5rem;
  font-size: 0.7rem;
  line-height: 0.2rem;
  text-align:center;
}
.icons{
  max-width: 25px;
  height: auto;
}
```

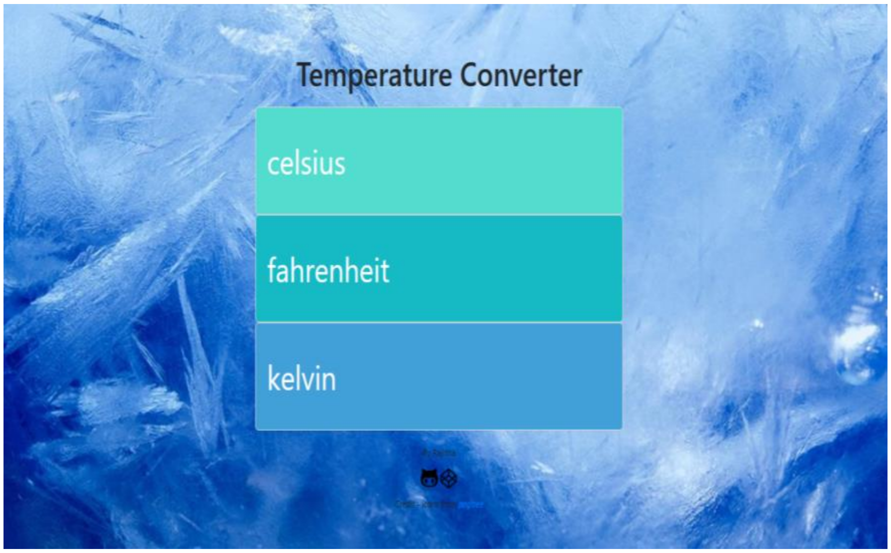
**Script.js**

```
const celsiusInput = document.querySelector('#celsius > input');
const fahrenheitInput = document.querySelector('#fahrenheit > input');
const kelvinInput = document.querySelector('#kelvin > input');
const roundToTwoDP = (num) => {
  return num.toFixed(2);
};
const celsiusToFaAndKe = () => {
  const celsiusTemp = parseFloat(celsiusInput.value);
  const fahrenheitTemp = (celsiusTemp * 1.8) + 32;
  const kelvinTemp = celsiusTemp + 273.15;
  fahrenheitInput.value = roundToTwoDP(fahrenheitTemp);
  kelvinInput.value = roundToTwoDP(kelvinTemp);
};
const fahrenheitToCeAndKe = () => {
  const fahrenheitTemp = parseFloat(fahrenheitInput.value);
  const celsiusTemp = (fahrenheitTemp - 32) * (5 / 9);
  const kelvinTemp = (fahrenheitTemp + 459.67) * (5 / 9);
  celsiusInput.value = roundToTwoDP(celsiusTemp);
  kelvinInput.value = roundToTwoDP(kelvinTemp);
};
const kelvinToCeAndFa = () => {
  const kelvinTemp = parseFloat(kelvinInput.value);
  const celsiusTemp = kelvinTemp - 273;
  const fahrenheitTemp =  1.8 * (kelvinTemp - 273) + 32;

  celsiusInput.value = roundToTwoDP(celsiusTemp);
  fahrenheitInput.value = roundToTwoDP(fahrenheitTemp);
};
const main = () => {
  celsiusInput.addEventListener('input', celsiusToFaAndKe);
  fahrenheitInput.addEventListener('input', fahrenheitToCeAndKe);
  kelvinInput.addEventListener('input', kelvinToCeAndFa);
};
main();
```

So on executing the above mentioned HTML, CSS, JAVASCRIPT pages the following output will be displayed.

**OUTPUT:**



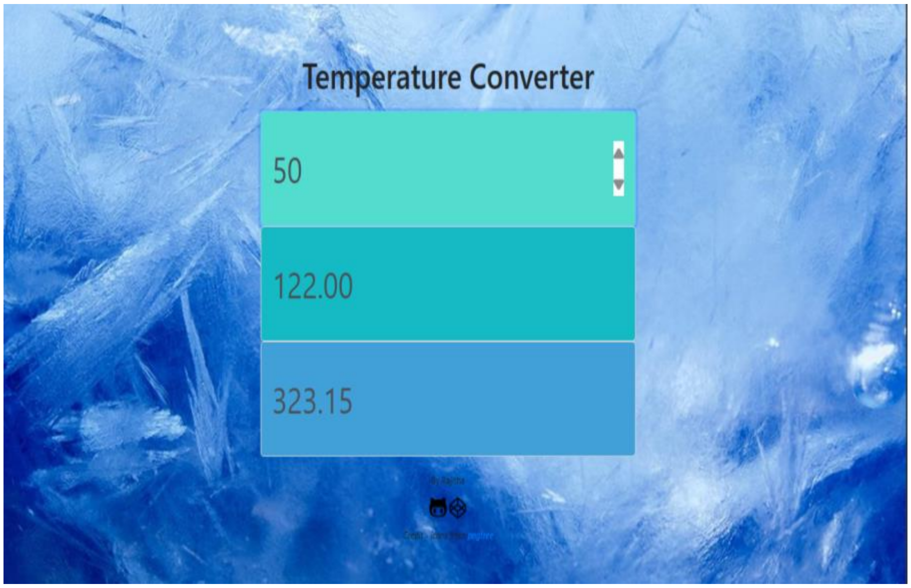
Temperature Converter

celsius

fahrenheit

kelvin

**IF CELSIUS IS GIVEN AS THE INPUT THE FOLLOWING OUTPUT RETURNS**



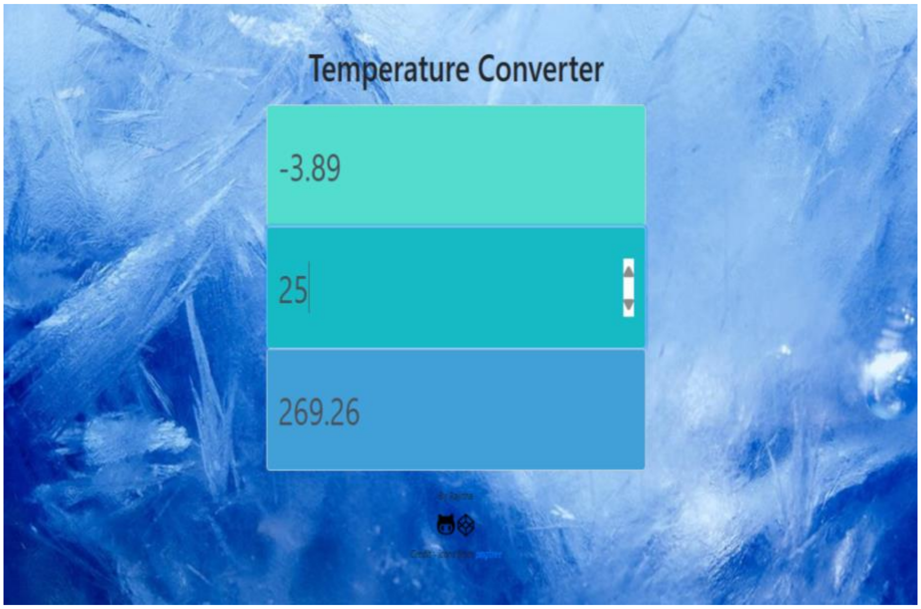
Temperature Converter

50

122.00

323.15

## IF FAHRENHEIT IS GIVEN AS INPUT THE FOLLOWING OUTPUT RETURNS



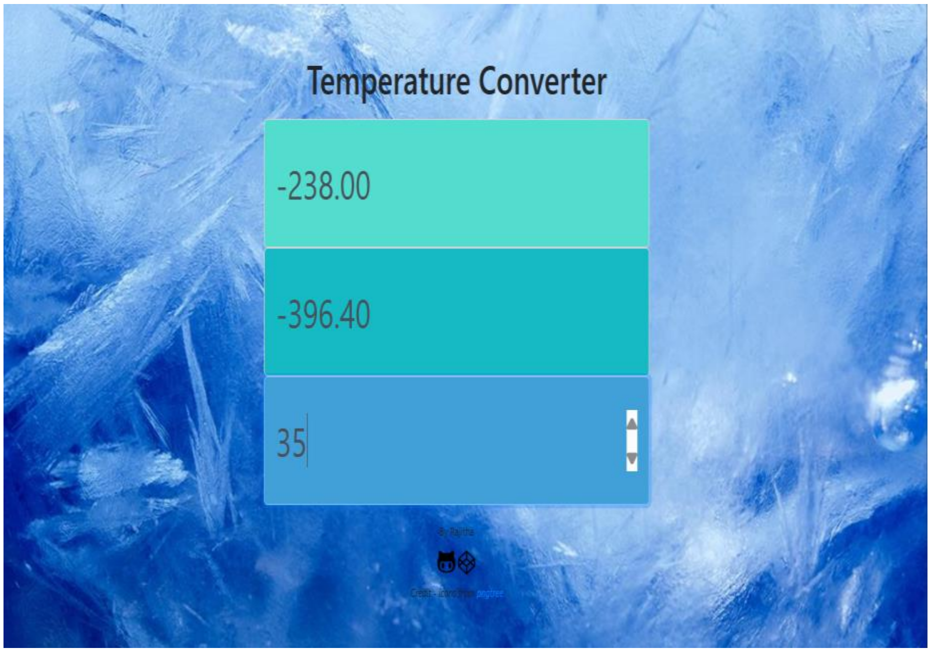
Temperature Converter

-3.89

25

269.26

## IF KELVIN IS GIVEN AS INPUT THE FOLLOWING OUTPUT RETURNS



Temperature Converter

-238.00

-396.40

35