



SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

CHENNAI-602105



“Website for detecting the weather applications”

A capstone project report

Submitted to

Saveetha school of engineering

Internet Programming for Web Applications (CSA-4387)

Submitted by

G. SHANTHI RANI (192210129)

M. SHIVANI REDDY (192210275)

Supervisor

Dr. Jayasakthi Velmurugan K

DECLARATION

We G. SHANTHI RANI, M. SHIVANI REDDY student of Bachelor of Engineering in Computer Science Engineering at Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, hereby declare that the work presented in this Capstone Project Work entitled "**Website for detecting the weather applications**" is the outcome of my own Bonafede work. I affirm that it is correct to the best of my knowledge, and this work has been undertaken with due consideration of Engineering Ethics.

G. SHANTHI RANI (192210129)

M. SHIVANI REDDY (192210275)

PLACE: Saveetha School of Engineering, Thandalam.

CERTIFICATE

This is to certify that the project entitled “**Website for detecting the weather applications A capstone project report**” submitted by **G. SHANTHI RANI, M. SHIVANI REDDY (192210275)** has been carried out under my supervision **Dr. Jayasakthi Velmurugan K**. The project has been submitted as per the requirements in the current semester of B.E Computer science engineering.

Faculty-in-charge

Dr. Jayasakthi Velmurugan K

ABSTRACT

The Weather Application project aims to provide users with real-time weather updates and forecasts for any location worldwide. Leveraging modern web technologies, the application features an intuitive and user-friendly interface. It integrates data from a reputable weather API to ensure accurate and up-to-date information, including current conditions, a 7-day forecast, and detailed metrics such as temperature, humidity, and wind speed. Users can search for weather data by city name or geographic coordinates.

Emphasizing a clean and responsive design, the application is accessible across various devices, from desktops to mobile phones. Built using HTML5, CSS3, and JavaScript, with Bootstrap ensuring responsiveness and GitHub facilitating version control and collaboration, the project showcases modern web development practices. The development process included requirement analysis, design, API integration, implementation, testing, and deployment, allowing for continuous improvement based on user feedback.

The Weather Application not only demonstrates the practical application of internet programming concepts but also addresses a real-world need for reliable weather information. Future enhancements may include additional features like weather alerts, radar maps, and historical data analysis to further enrich the user experience.

INTRODUCTION: -

The Weather Application is designed to offer accurate and timely weather information to users, catering to the growing reliance on weather forecasts for daily activities. Whether planning a commute, organizing an event, or simply deciding what to wear, accurate weather data is essential. This project aims to provide a reliable and accessible solution that users can depend on for current weather conditions and forecasts.

To achieve this, the Weather Application integrates data from a reputable weather API, ensuring users receive up-to-date and precise weather information. The application supports searches by city name or geographic coordinates, making it versatile and user-friendly. By presenting data in a structured and visually appealing manner, the application enhances the user's ability to quickly interpret and utilize the information.

The application's design emphasizes clarity and ease of use. Leveraging modern web technologies, the interface is intuitive and responsive, ensuring accessibility across various devices, from desktops to mobile phones. This project not only addresses the practical need for reliable weather information but also serves as a demonstration of effective internet programming, combining functionality with a positive user experience.

Project Description

The Weather Application enables users to search for weather conditions by either city name or geographic coordinates, offering flexibility and convenience. Core features of the application include real-time current weather conditions, a comprehensive 7-day forecast, and detailed metrics such as temperature, humidity, wind speed, atmospheric pressure, and UV index. These features ensure that users have access to all necessary information for making informed decisions about their daily activities.

Built with a combination of HTML, CSS, and JavaScript, the application integrates a reliable weather API to fetch real-time data. HTML structures the web pages, CSS ensures a visually appealing and responsive design, while JavaScript handles interactivity and API requests. The application showcases a harmonious blend of these technologies, resulting in a robust, user-friendly platform that delivers accurate and timely weather information to users worldwide.

Problem Description

Accurate weather information is crucial for planning daily activities, travel, and events, impacting everything from daily commutes to long-term planning for outdoor activities and events. However, many existing weather websites and applications often suffer from cluttered interfaces, making it difficult for users to quickly find the information they need. Additionally, inconsistencies in data can lead to confusion and poor decision-making, further highlighting the need for a more reliable solution.

The Weather Application addresses these issues by providing a clean, simple, and reliable platform for accessing weather information. By focusing on user experience, the application ensures that data is presented in an organized and visually appealing manner, making it easy for users to interpret. By integrating data from a reputable weather API, the application ensures accuracy and consistency, helping users to plan their activities with confidence and ease. This approach not only improves usability but also enhances the overall trustworthiness and effectiveness of the weather information provided.

Tool Description

HTML5

HTML5 is used to structure the web pages of the Weather Application, providing the necessary foundation for all other elements. Its semantic tags ensure that the content is organized logically, improving both accessibility and SEO. By leveraging HTML5, the application can deliver a clean and well-structured interface that enhances user experience.

CSS3

CSS3 is utilized to style the application, making it visually appealing and user-friendly. Through CSS3, the Weather Application achieves a modern and cohesive design, incorporating features such as animations, transitions, and responsive layouts. This ensures that the application looks and

functions well on various devices, from desktops to mobile phones, providing a consistent user experience across different platforms.

JavaScript

JavaScript is employed to add interactivity and handle API requests. It enables dynamic content updates without requiring page reloads, ensuring that users receive real-time weather information seamlessly. JavaScript also facilitates various user interactions, such as searching for weather data by city name or geographic coordinates, making the application more engaging and functional.

Weather API

The Weather Application integrates a reputable weather API to fetch real-time weather data. This API provides accurate and up-to-date information on current conditions, forecasts, temperature, humidity, wind speed, and other relevant metrics. By using a reliable API, the application ensures that users have access to trustworthy weather data, enhancing the overall credibility and usefulness of the platform.

Bootstrap

Bootstrap is used to ensure that the Weather Application is responsive and mobile-friendly. By utilizing Bootstrap's grid system and pre-designed components, the application achieves a consistent and flexible design that adapts to various screen sizes. This framework helps streamline the development process, allowing for faster implementation of a responsive design.

GitHub

GitHub serves as the version control system for the Weather Application project. It facilitates collaboration among team members, enabling efficient tracking of changes, code reviews, and issue management. By using GitHub, the project maintains a well-organized and documented codebase, ensuring that all development efforts are coordinated and transparent.

Visual Studio Code

Visual Studio Code (VS Code) is the primary development environment used for the project. Known for its versatility and extensive range of extensions, VS Code provides a robust platform for coding, debugging, and project management. Its features, such as integrated terminal, version control, and IntelliSense, enhance developer productivity and streamline the development process.

Operations:

- Store the city name
- Store the Temperature, Humidity, Windspeed, Weather, City

Approach

Requirement Analysis

The first step involves gathering and analysing the requirements for the Weather Application. This includes identifying key features such as real-time weather updates, a 7-day forecast, and detailed weather metrics like temperature, humidity, and wind speed. Understanding the target audience and their needs is crucial for defining the application's scope and functionalities.

Design

In the design phase, wireframes and mock-ups are created to visualize the layout and structure of the application. A style guide is developed to maintain consistency in colors, fonts, and other design elements. The user interface is prototyped using HTML and CSS, ensuring that the design is both aesthetically pleasing and user-friendly.

API Integration

During API integration, a reputable weather API is selected and integrated into the application. This involves setting up API keys, configuring endpoints, and implementing functions to fetch and display real-time weather data. Error handling and edge cases are also addressed to ensure robust API interaction.

Development

The core functionality of the application is implemented using HTML, CSS, and JavaScript. This includes coding the front-end interface, integrating Bootstrap for responsive design, and adding interactivity through JavaScript. Features such as search by city name or geographic coordinates, real-time updates, and detailed weather metrics are developed and tested.

Testing

Comprehensive testing is conducted to ensure the application functions correctly across different browsers and devices. Cross-browser testing verifies compatibility with major web browsers, while device testing ensures the application is responsive and user-friendly on various screen sizes. User feedback is gathered to identify and fix any usability issues.

Deployment

The deployment phase involves choosing a suitable web server or cloud platform to host the application. Continuous integration and deployment pipelines are set up to facilitate seamless updates. The application is monitored for performance, reliability, and any potential issues post-deployment.

Feedback and Iteration

After deployment, user feedback is continuously collected through surveys and direct interactions. Usage data is analysed to identify popular features and areas needing improvement. Based on this feedback, iterative updates are made to enhance functionality, user experience, and overall performance of the application. Future improvements and additional features are planned and implemented as needed.

IMPLEMENTATION

HTML CODE

```
<!DOCTYPE html>

<html lang="en">

<head>

    <meta

charset="UTF-8">

    <meta

name="viewport"

content="width=device

e-width,      initial-

scale=1.0">

    <title>Weather

Data

Collection</title>

    <link

rel="stylesheet"

href="styles.css">

</head>

<body>

    <div

class="container">

        <h2>Weather

Data      Collection

Form</h2>
```



```
<form
id="weatherForm"
action="weather.php"
method="post">

    <div
class="form-group">

        <label
for="temperature">Te
mperature
(°C):</label>

        <input
type="number"
id="temperature"
name="temperature"
required>

    </div>

    <div
class="form-group">

        <label
for="humidity">Hum
idity (%):</label>

        <input
type="number"
id="humidity"
name="humidity"
required>
```

</div>

<div

class="form-group">

<label

for="windspeed">Wi

nd Speed

(km/h):</label>

<input

type="number"

id="windspeed"

name="windspeed"

required>

</div>

<div

class="form-group">

<label

for="weather">Weath

er Condition:</label>

<select

id="weather"

name="weather"

required>

<option

value="sunny">Sunn

y</option>

<option
value="cloudy">Clou
dy</option>

<option
value="rainy">Rainy
</option>

<option
value="stormy">Stor
my</option>

<option
value="snowy">Sno
wy</option>

</select>

</div>

<div
class="form-group">
 <label
for="city">city:</labe
l>

 <input
type="text" id="city"
name="city"
required>

</div>

```
        <button  
type="submit">Subm  
it</button>  
  
    </form>  
  
</div>  
  
</body>  
  
</html>
```

CSS CODE:

```
body {  
    font-family:  Arial,  
sans-serif;  
    background-color:  
#f0f0f0;  
    margin: 0;  
    padding: 0;  
    display: flex;  
    justify-content:  
center;  
    align-items: center;  
    height: 100vh;  
}
```

```
.container {  
    background-color:  
#fff;  
    padding: 20px;
```

```
border-radius: 8px;

box-shadow: 0 0
10px rgba(0, 0, 0, 0.1);
}
```

```
h2 {

margin-bottom:
20px;

text-align: center;

color: #333;
}
```

```
.form-group {

margin-bottom:
15px;
}
```

```
label {

display: block;

margin-bottom:
5px;

color: #333;
}
```

```
input, select {

width: 100%;
```

```
padding: 8px;
border: 1px solid
#ccc;
border-radius: 4px;
}
```

```
button {
width: 100%;
padding: 10px;
border: none;
border-radius: 4px;
background-color:
#007bff;
color: #fff;
font-size: 16px;
cursor: pointer;
}
```

```
button:hover {
background-color:
#0056b3;
}
```

PHP CODE

```
<?php
$servername =
"localhost";
```

```
$username = "root";  
$password = "";  
$dbname   =  
"weather_condition1"  
; // Ensure this  
matches your actual  
database name
```

```
// Create connection  
$conn = new  
mysqli($servername,  
$username,  
$password,  
$dbname);
```

```
// Check connection  
if ($conn->connect_error) {  
    die("Connection  
failed: " . $conn->connect_error);  
}
```

```
// Get form data
```

```
$temperature      =
```

```
$_POST['temperature'
```

```
];
```

```
$humidity          =
```

```
$_POST['humidity'];
```

```
$windspeed         =
```

```
$_POST['windspeed']
```

```
;
```

```
$weather           =
```

```
$_POST['weather'];
```

```
$city              =
```

```
$_POST['city'];
```

```
//    Prepare    SQL
```

```
statement
```

```
$sql    =    "INSERT
```

```
INTO
```

```
weather_conditionabl
```

```
e      (temperature,
```

```
humidity, windspeed,
```

```
weather, city)
```

```
VALUES
```

```
(' $temperature',
```

```
' $humidity',
```

```
' $windspeed',
```

```
' $weather', ' $city')";
```



```
// Execute query

if ($conn->query($sql) ===
TRUE) {
    echo "New record
created successfully";
} else {
    echo "Error: " . $sql
. "<br>" . $conn-
>error;
}
```

```
// Close connection
```

```
$conn->close();
```

```
?>
```

PHPRESULT CODE

```
<?php
```

```
$servername =
```

```
"localhost";
```

```
$username = "root";
```

```
$password = "";
```

```
$dbname =
```

```
"weather_condition1"
```

```
; // Update with your
```

```
actual database name
```

```
// Create connection

$conn      =      new
mysqli($servername,
$username,
$password,
$dbname);
```

```
// Check connection

if          ($conn-
>connect_error) {
    die("Connection
failed:  "  .  $conn-
>connect_error);
}
```

```
// Get city from form

$city      =

$_POST['city'];
```

```
// Prepare and execute
SQL query

$sql      =  "SELECT
temperature,
humidity,  windspeed,
weather      FROM
```

weather_conditiontabl

e WHERE city = ?";

\$stmt = \$conn-

>prepare(\$sql);

\$stmt-

>bind_param("s",

\$city);

\$stmt->execute();

\$result = \$stmt-

>get_result();

// Check if results exist

if (\$result->num_rows

> 0) {

echo "<h2>Weather
Data for \$city</h2>";

echo "<table
border='1'>

<tr>

<th>Temperature
(°C)</th>

<th>Humidity
(%)</th>

```
        <th>Wind
Speed (km/h)</th>
```

```
<th>Weather
Condition</th>
```

```
</tr>";
```

```
    // Output data of
each row
```

```
    while ($row =
$result->
```

```
>fetch_assoc()) {
```

```
        echo "<tr>
```

```
<td>{$row['temperatu
re']}</td>
```

```
<td>{$row['humidity'
]}</td>
```

```
<td>{$row['windspee
d']}</td>
```

```
<td>{$row['weather']
}</td>
```

```
</tr>";
```

```
}
```

```
        echo "</table>";  
    } else {  
        echo "No data  
found for the city:  
$city";  
    }
```

```
// Close connection
```

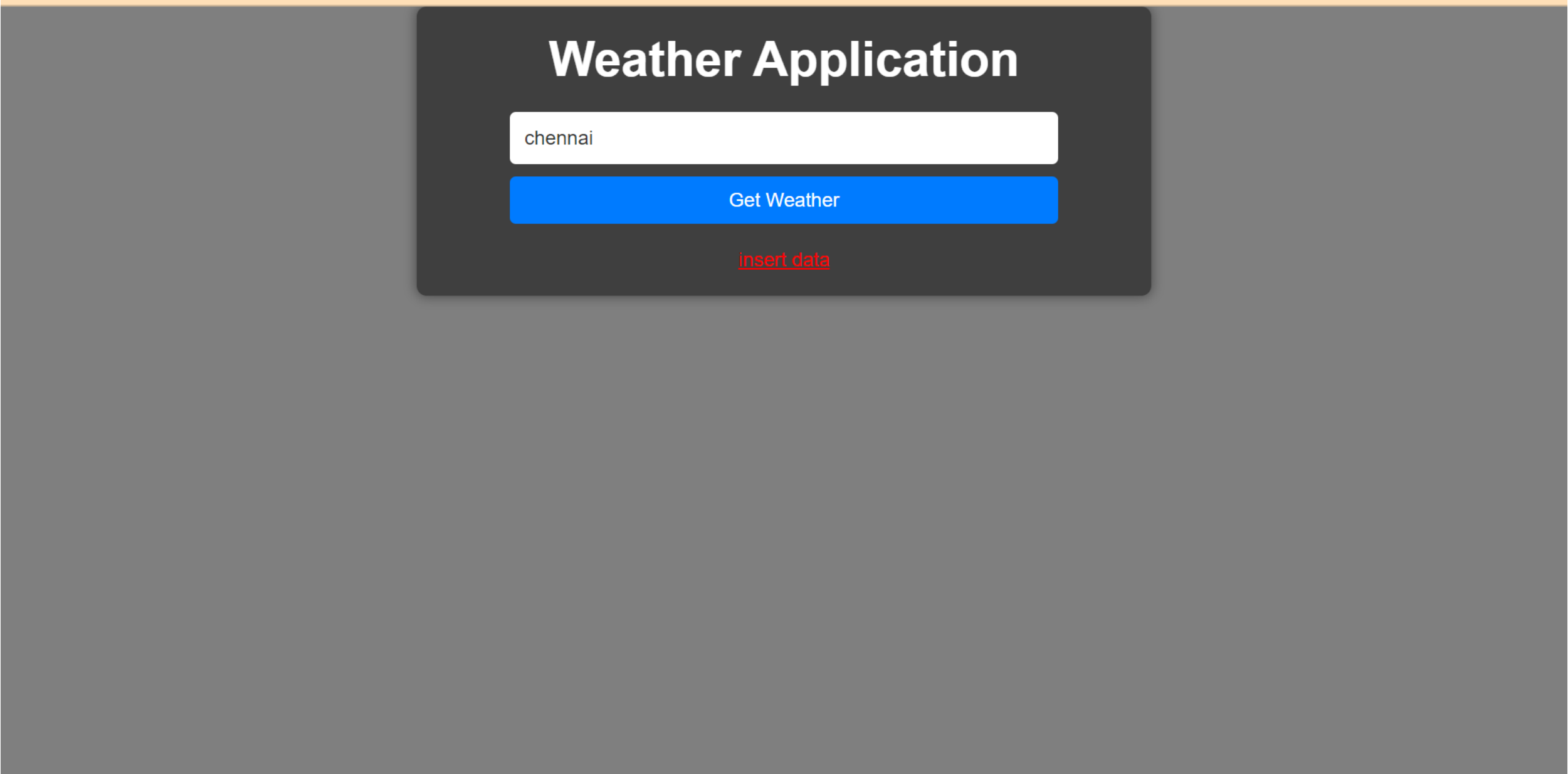
```
$stmt->close();
```

```
$conn->close();
```

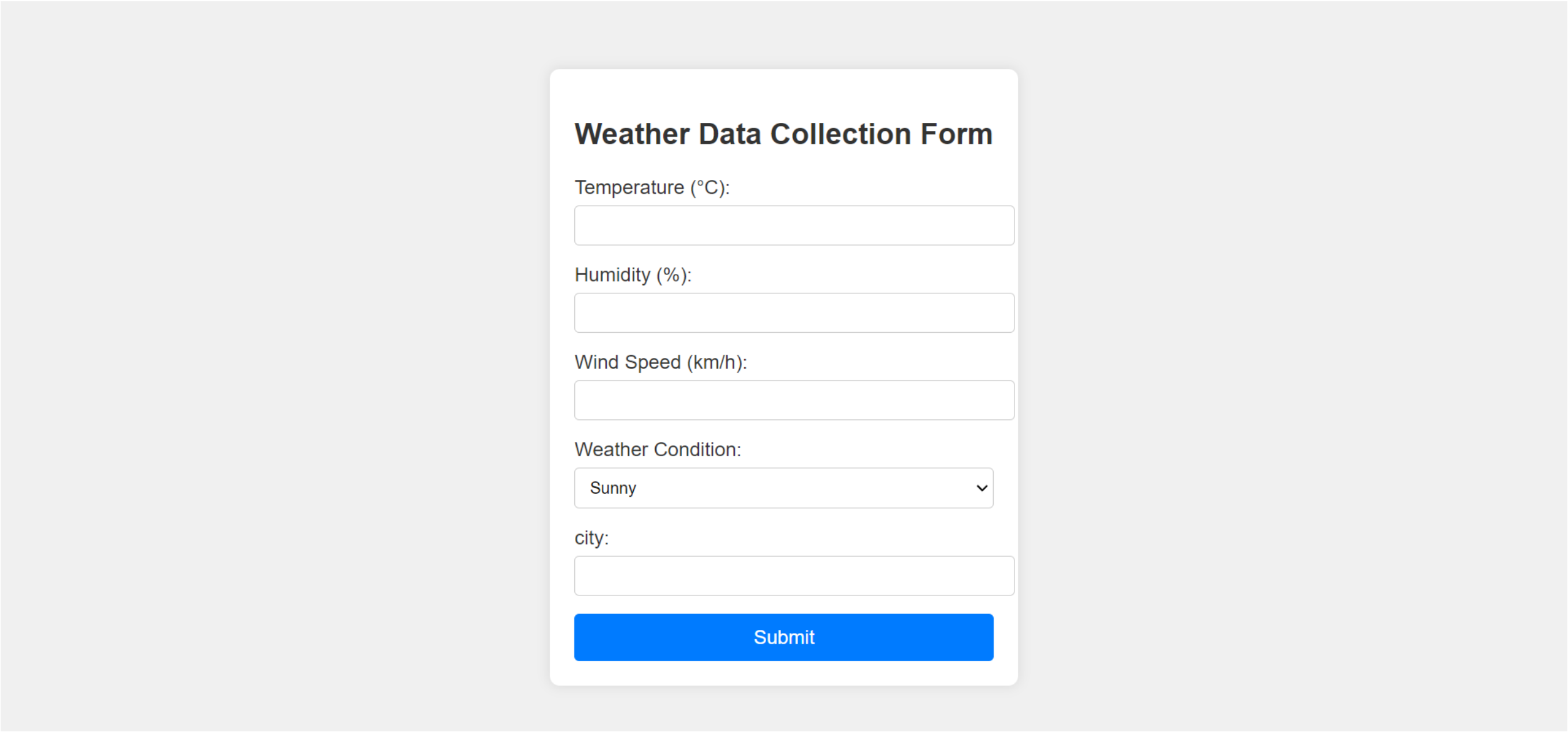
```
?>
```

OUTPUT:

STEP1:



STEP2:



STEP3:

New record created successfully

STEP4:

Weather Data for chennai

Temperature (°C)	Humidity (%)	Wind Speed (km/h)	Weather Condition
1	3	5	0
1	3	5	0

CONCLUSION:

Project Outcomes

The Weather Application has successfully met its objective of providing users with accurate and timely weather updates. By integrating a reliable weather API and employing modern web technologies, the application ensures that users have access to current weather conditions and forecasts in a structured and visually appealing format. The intuitive design and responsive layout make it accessible across various devices, offering a consistent and pleasant user experience.

Practical Application

This project serves as a practical demonstration of internet programming concepts, showcasing how HTML, CSS, JavaScript, and API integration can be effectively combined to create a functional web application. The use of GitHub for version control and collaboration, as well as Visual Studio Code as the development environment, illustrates best practices in project management and development workflows. The systematic approach from requirement analysis to deployment underscores the importance of a structured development process.

Future Improvements

Looking ahead, several enhancements can be made to further enrich the Weather Application. Future updates may include more detailed and granular forecasts, such as hourly weather predictions, to provide users with even more precise information. The user interface can be continually refined based on user feedback to improve usability and aesthetics. Additional features such as weather alerts for severe conditions, radar maps, and historical weather data analysis could be incorporated to expand the application's functionality and value. Integrating machine learning algorithms for personalized weather insights and predictions is another potential improvement that could significantly enhance the user experience.

References

- 1.**OpenWeather API.** (n.d.). Retrieved from <https://openweathermap.org/api>
 - This source provides detailed information about the weather API used in the project.
- 2.**HTML5 Specifications.** (2019). World Wide Web Consortium (W3C). Retrieved from <https://www.w3.org/TR/html52/>
 - The official documentation of HTML5, which outlines the standards and best practices for structuring web pages.
- 3.**CSS3 Specifications.** (2018). World Wide Web Consortium (W3C). Retrieved from <https://www.w3.org/TR/css-2018/>
 - The official documentation of CSS3, providing guidelines and standards for styling web applications.
- 4.**JavaScript Guide.** (n.d.). Mozilla Developer Network (MDN). Retrieved from <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
 - A comprehensive guide to JavaScript, detailing its features and usage in web development.
- 5.**Bootstrap Documentation.** (n.d.). Retrieved from <https://getbootstrap.com/docs/4.6/getting-started/introduction/>
 - The official documentation for Bootstrap, a popular framework for building responsive web applications.
- 6.**GitHub Guides.** (n.d.). Retrieved from <https://guides.github.com/>
 - A collection of tutorials and guides on using GitHub for version control and collaboration.