**Code**

**HTML Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <link rel="icon" type="image/svg+xml" href="logo.svg">

    <link rel="icon" href="favicon.ico">

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

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

    <title>WeatherWise - Your Weather Information</title>

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0/css/bootstrap.min.css" />

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css">

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

</head>

<body>

    <div class="jumbotron container-fluid mt-2 text-center p-2">

        <h1> Weather-Wise</h1>

    </div>

    <div class="container">

        <div class="row">

            <div class="col-sm-4 bg-dark">

                <br>

                <div class="input-group mb-3">

                    <input type="text" class="form-control" id="search-city" aria-label="City Search" aria-describedby="search-button" placeholder="Enter City">

                    <div class="input-group-append">

                        <button class="btn bg-primary text-light" id="search-button"><i class="fa fa-search"></i></button>

                    </div>

                </div>

                <button class="btn btn-primary" type="button" id="clear-history" style="color: black;">Clear history</button>

                <br><br>

                <ul class="list-group">

                    <!-- History items will be added here -->

                </ul>

            </div>

            <div class="col-sm-8">

                <div class="row ml-2">

                    <div class="col-sm-12" id="current-weather">

                        <h3 class="city-name mb-1 mt-2 bolder" id="current-city"></h3>

                        <p>Temperature:<span class="current" id="temperature"></span></p>

                        <p>Humidity:<span class="current" id="humidity"></span></p>

                        <p>Wind Speed:<span class="current" id="wind-speed"></span></p>

                        <p>UV index:<span class="current bg-danger rounded py-2 px-2 text-white" id="uv-index"></span></p>

                        <p><a href="Details.html">Click Here To Know More About this City</a></p>

                    </div>

                </div>

                <div class="col-sm-12" id="future-weather">

                    <br>

                    <h3 style="color: #111111;">5-Day Forecast:</h3>

                    <div class="row text-light">

                        <div class="col-sm-2 bg-primary forecast text-white ml-2 mb-3 p-2 mt-2 rounded" >

                            <p id="fDate0"></p>

                            <p id="fImg0"></p>

                            <p>Temp:<span id="fTemp0"></span></p>

                            <p>Humidity:<span id="fHumidity0"></span></p>

                        </div>

                        <div class="col-sm-2 bg-primary forecast text-white ml-2 mb-3 p-2 mt-2 rounded" >

                            <p id="fDate1"></p>

                            <p id="fImg1"></p>

                            <p>Temp:<span id="fTemp1"></span></p>

                            <p>Humidity:<span id="fHumidity1"></span></p>

                        </div>

                        <div class="col-sm-2 bg-primary forecast text-white ml-2 mb-3 p-2 mt-2 rounded">

                            <p id="fDate2"></p>

                            <p id="fImg2"></p>

                            <p>Temp:<span id="fTemp2"></span></p>

                            <p>Humidity:<span id="fHumidity2"></span></p>

                        </div>

                        <div class="col-sm-2 bg-primary forecast text-white ml-2 mb-3 p-2 mt-2 rounded">

                            <p id="fDate3"></p>

                            <p id="fImg3"></p>

                            <p>Temp:<span id="fTemp3"></span></p>

                            <p>Humidity:<span id="fHumidity3"></span></p>

                        </div>

                        <div class="col-sm-2 bg-primary forecast text-white ml-2 mb-3 p-2 mt-2 rounded" >

                            <p id="fDate4"></p>

                            <p id="fImg4"></p>

                            <p>Temp:<span id="fTemp4"></span></p>

                            <p>Humidity:<span id="fHumidity4"></span></p>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </div>

    <br>

    <br>

    <footer class="text-center py-3">

        <p>&copy; 2023 WeatherWise. All rights reserved. | Designed by Abhishek Singh</p>

        <div class="social">

            <a href="https://www.linkedin.com/in/itsabhishek98/" target="\_blank"><i class="fab fa-linkedin"></i></a>

            <a href="https://github.com/singhabhishek98" target="\_blank"><i class="fab fa-github"></i></a>

            <a href="https://www.facebook.com/profile.php?id=100011235600297" target="\_blank"><i class="fab fa-facebook"></i></a>

            <a href="https://www.instagram.com/\_itsabhi98/" target="\_blank"><i class="fab fa-instagram"></i></a>

        </div>

    </footer>

    <script src="https://code.jquery.com/jquery.js"></script>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.0.0/js/bootstrap.bundle.min.js"></script>

    <script src="script.js"></script>

    <script>

        document.getElementById("search-city").addEventListener("keyup", function(event) {

            if (event.key === "Enter") {

                document.getElementById("search-button").click();

            }

        });

    </script>

</body>

</html>

detail.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>City Details - WeatherWise</title>

    <link rel="stylesheet" href="style.css"> <!-- Include your custom CSS file if needed -->

    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"> <!-- Include Bootstrap CSS -->

    <style>

        /\* Custom CSS for enhancing the page's appearance \*/

        body {

            background-color: #f5f5f5;

            margin-bottom: 70px; /\* Adjust the margin to account for the footer's height \*/

        }

        .container {

            padding-top: 30px;

        }

        .card {

            border: 2px solid #3498db;

            border-radius: 10px;

            box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.2);

        }

        .card-title {

            color: #3498db;

        }

        .card-text {

            font-size: 18px;

            color: #333;

        }

        footer {

            position: fixed;

            bottom: 0;

            width: 100%;

            background-color: #3498db;

            color: #fff;

            padding: 10px 0;

        }

    </style>

</head>

<body>

    <main>

        <div class="container">

            <!-- Create a Bootstrap card to display city details -->

            <div class="card">

                <div class="card-header">

                    <h2 class="card-title">City Details</h2>

                </div>

                <div class="card-body">

                    <p class="card-text"><strong>City Name:</strong> <span id="cityName"></span></p>

                    <p class="card-text"><strong>Temperature:</strong> <span id="temperature"></span></p>

                    <p class="card-text"><strong>Humidity:</strong> <span id="humidity"></span></p>

                    <p class="card-text"><strong>Wind Speed:</strong> <span id="windSpeed"></span></p>

                    <p class="card-text"><strong>UV Index:</strong> <span id="uvIndex"></span></p>

                </div>

            </div>

        </div>

    </main>

    <footer>

        <p>&copy; 2023 WeatherWise. All rights reserved. | Designed by Abhishek Singh</p>

    </footer>

    <script>// In Details.html

document.addEventListener("DOMContentLoaded", function() {

    const urlSearchParams = new URLSearchParams(window.location.search);

    const params = Object.fromEntries(urlSearchParams.entries());

    // Display city details

    const cityNameElement = document.getElementById("cityName");

    const temperatureElement = document.getElementById("temperature");

    const humidityElement = document.getElementById("humidity");

    const windSpeedElement = document.getElementById("windSpeed");

    const uvIndexElement = document.getElementById("uvIndex");

    // Populate the city details in the card

    cityNameElement.textContent = params.cityName;

    temperatureElement.textContent = params.temperature;

    humidityElement.textContent = params.humidity;

    windSpeedElement.textContent = params.windSpeed;

    uvIndexElement.textContent = params.uvIndex;

});

    </script>

</body>

</html>

**Style.CSS:**

/\* Add custom styles to your page \*/

body {

    background-color: #f5f5f5;

    background-image: url('sky.webp'); /\* Add your background image URL here \*/

    background-size: 100% 100%;

    background-repeat: no-repeat;

    background-attachment: fixed;

}

.jumbotron {

    background-color: #007BFF;

    color: white;

}

#current-weather {

    background-color: #343a40;

    color: white;

    border-radius: 10px;

    padding: 10px;

    margin-top: 20px;

}

.forecast {

    background-color: #007BFF;

    color: white;

    border-radius: 10px;

    padding: 10px;

    margin-top: 20px;

}

footer {

    background-color: #343a40;

    color: white;

    padding: 10px;

    margin-top: 20px;

}

/\* Style the social icons \*/

.social a {

    color: white;

    font-size: 20px;

    margin: 0 10px;

    text-decoration: none;

}

Java Script:

//Declare a variable to store the searched city

var city = "";

// variable declaration

var searchCity = $("#search-city");

var searchButton = $("#search-button");

var clearButton = $("#clear-history");

var currentCity = $("#current-city");

var currentTemperature = $("#temperature");

var currentHumidity = $("#humidity");

var currentWSpeed = $("#wind-speed");

var currentUvindex = $("#uv-index");

var sCity = [];

// searches the city to see if it exists in the entries from the storage

function find(c) {

for (var i = 0; i < sCity.length; i++) {

if (c.toUpperCase() === sCity[i]) {

return -1;

}

}

return 1;

}

//Set up the API key

var APIKey = "a0aca8a89948154a4182dcecc780b513";

// Display the current and future weather to the user after grabbing the city from the input text box.

function displayWeather(event) {

event.preventDefault();

if (searchCity.val().trim() !== "") {

city = searchCity.val().trim();

currentWeather(city);

}

}

// Here we create the AJAX call

function currentWeather(city) {

// Here we build the URL so we can get data from the server side.

var queryURL = "https://api.openweathermap.org/data/2.5/weather?q=" + city + "&APPID=" + APIKey;

$.ajax({

url: queryURL,

method: "GET",

}).then(function (response) {

// parse the response to display the current weather including the City name. the Date and the weather icon.

console.log(response);

// Data object from server-side API for icon property.

var weathericon = response.weather[0].icon;

var iconurl = "https://openweathermap.org/img/wn/" + weathericon + "@2x.png";

// The date format method is taken from the https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Date

var date = new Date(response.dt \* 1000).toLocaleDateString();

// parse the response for the name of the city and concatenating the date and icon.

$(currentCity).html(response.name + " (" + date + ") " + "<img src=" + iconurl + ">");

// parse the response to display the current temperature in Celsius.

// Convert the temp to Celsius

var tempC = (response.main.temp - 273.15).toFixed(2);

$(currentTemperature).html(tempC + "&#8451");

// Display the Humidity

$(currentHumidity).html(response.main.humidity + "%");

// Display Wind speed and convert to meters per second (m/s)

var ws = response.wind.speed;

var windsmph = (ws).toFixed(1); // Wind speed is already in m/s

$(currentWSpeed).html(windsmph + " m/s");

// Display UV Index.

// By Geographic coordinates method and using appid and coordinates as a parameter, we are going to build our UV query URL inside the function below.

UVIndex(response.coord.lon, response.coord.lat);

forecast(response.id);

if (response.cod == 200) {

sCity = JSON.parse(localStorage.getItem("cityname"));

console.log(sCity);

if (sCity == null) {

sCity = [];

sCity.push(city.toUpperCase());

localStorage.setItem("cityname", JSON.stringify(sCity));

addToList(city);

} else {

if (find(city) > 0) {

sCity.push(city.toUpperCase());

localStorage.setItem("cityname", JSON.stringify(sCity));

addToList(city);

}

}

}

});

}

// This function returns the UVIindex response.

function UVIndex(ln, lt) {

//lets build the url for uvindex.

var uvqURL = "https://api.openweathermap.org/data/2.5/uvi?appid=" + APIKey + "&lat=" + lt + "&lon=" + ln;

$.ajax({

url: uvqURL,

method: "GET"

}).then(function (response) {

$(currentUvindex).html(response.value);

});

}

// Here we display the 5 days forecast for the current city.

function forecast(cityid) {

var dayover = false;

var queryforcastURL = "https://api.openweathermap.org/data/2.5/forecast?id=" + cityid + "&appid=" + APIKey;

$.ajax({

url: queryforcastURL,

method: "GET"

}).then(function (response) {

for (i = 0; i < 5; i++) {

var date = new Date((response.list[((i + 1) \* 8) - 1].dt) \* 1000).toLocaleDateString();

var iconcode = response.list[((i + 1) \* 8) - 1].weather[0].icon;

var iconurl = "https://openweathermap.org/img/wn/" + iconcode + ".png";

var tempK = response.list[((i + 1) \* 8) - 1].main.temp;

var tempC = (((tempK - 273.15)).toFixed(2)) + "&#8451";

var humidity = response.list[((i + 1) \* 8) - 1].main.humidity;

$("#fDate" + i).html(date);

$("#fImg" + i).html("<img src=" + iconurl + ">");

$("#fTemp" + i).html(tempC);

$("#fHumidity" + i).html(humidity + "%");

}

});

}

// Dynamically add the passed city to the search history

function addToList(c) {

var listEl = $("<li>" + c.toUpperCase() + "</li>");

$(listEl).attr("class", "list-group-item");

$(listEl).attr("data-value", c.toUpperCase());

$(".list-group").append(listEl);

}

// Display the past search again when the list group item is clicked in search history

function invokePastSearch(event) {

var liEl = event.target;

if (event.target.matches("li")) {

city = liEl.textContent.trim();

currentWeather(city);

}

}

// render function

function loadlastCity() {

$("ul").empty();

var sCity = JSON.parse(localStorage.getItem("cityname"));

if (sCity !== null) {

sCity = JSON.parse(localStorage.getItem("cityname"));

for (i = 0; i < sCity.length; i++) {

addToList(sCity[i]);

}

city = sCity[i - 1];

currentWeather(city);

}

}

// Clear the search history from the page

function clearHistory(event) {

event.preventDefault();

sCity = [];

localStorage.removeItem("cityname");

document.location.reload();

}

// In script.js

document.addEventListener("DOMContentLoaded", function() {

const cityDetailsLink = document.querySelector("#current-weather a");

cityDetailsLink.addEventListener("click", function(event) {

event.preventDefault();

// Get the city name and other city details from your current page

const cityName = document.querySelector("#current-city").textContent;

const temperature = document.querySelector("#temperature").textContent;

const humidity = document.querySelector("#humidity").textContent;

const windSpeed = document.querySelector("#wind-speed").textContent;

const uvIndex = document.querySelector("#uv-index").textContent;

// Construct the URL for the "Details.html" page with query parameters

const detailsURL = `Details.html?cityName=${cityName}&temperature=${temperature}&humidity=${humidity}&windSpeed=${windSpeed}&uvIndex=${uvIndex}`;

// Redirect to the "Details.html" page with the city details

window.location.href = detailsURL;

});

});

// Click Handlers

$("#search-button").on("click", displayWeather);

$(document).on("click", invokePastSearch);

$(window).on("load", loadlastCity);

$("#clear-history").on("click", clearHistory);

**weatherwiseapplication.java:**

package com.wise.weather.weatherwise;

import org.springframework.boot.SpringApplication;

import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class WeatherwiseApplication {

public static void main(String[] args) {

SpringApplication.run(WeatherwiseApplication.class, args);

}

}

**weathercontroller.java:**

ackage com.wise.weather.weatherwise.controllers;

import com.wise.weather.weatherwise.entity.City;

import com.wise.weather.weatherwise.entity.FamousPlace;

import com.wise.weather.weatherwise.services.WeatherService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.sql.SQLException;

import java.util.List;

@CrossOrigin(origins = "", allowedHeaders = "")

@RestController

public class WeatherController {

@Autowired

private WeatherService weatherService;

@PostMapping("/saveCity")

public int saveCity(@RequestBody City city) throws SQLException {

int id= weatherService.saveCity(city);

return id;

}

@GetMapping("/getCity/{name}")

public City getCity(@PathVariable("name") String name) throws SQLException {

City city = weatherService.getCityData(name);

return city;

}

@PostMapping("/updateCity")

public int updateCity(@RequestBody City city) throws SQLException {

int id = weatherService.updateCity(city);

return id;

}

@PostMapping("/saveFamousPlace")

public int saveFamousPlace(@RequestBody FamousPlace famousPlace) throws SQLException {

return weatherService.saveFamousPlace(famousPlace);

}

@GetMapping("/getFamousPlace/{cityName}")

public List<FamousPlace> getFamousPlace(@PathVariable("cityName") String cityName) throws SQLException {

return weatherService.getFamousPlaces(cityName);

    }

}

**WeatherDao.java:**

package com.wise.weather.weatherwise.dao;

import com.wise.weather.weatherwise.entity.City;

import com.wise.weather.weatherwise.entity.FamousPlace;

import org.springframework.stereotype.Repository;

import java.sql.\*;

import java.util.ArrayList;

import java.util.List;

import java.util.Properties;

@Repository

public class WeatherDao {

private String url = "jdbc:mysql://localhost:3306/weatherwise";

private String user = "root";

private String password = "abhishek@123";

public int saveCity(City city) throws SQLException {

Connection connection = getConnection();

Statement statement = connection.createStatement();

String query = "insert into city(id, name, temperature,humidity,windSpeed,uvIndex,description)" +

"values("+city.getId()+", '" +city.getName() + "', '" + city.getTemperature() + "', '" + city.getHumidity() + "'" +

", '" + city.getWindSpeed() + "', '" + city.getUvIndex() + "', '" + city.getDescription() + "' )";

System.out.println(">>> Query >> " + query);

statement.execute(query);

return city.getId();

}

public City getCityData(String name) throws SQLException {

Connection connection = getConnection();

Statement statement = connection.createStatement();

String query = "select \* from city where name = '" + name + "'";

System.out.println(">>> Query >> " + query);

ResultSet resultSet = statement.executeQuery(query);

City city = null;

while (resultSet.next()) {

city=new City();

city.setId(resultSet.getInt("id"));

city.setName(resultSet.getString("name"));

city.setHumidity(resultSet.getString("humidity"));

city.setTemperature(resultSet.getString("temperature"));

city.setDescription(resultSet.getString("description"));

city.setUvIndex(resultSet.getString("uvIndex"));

city.setWindSpeed(resultSet.getString("windSpeed"));

}

resultSet.close();

statement.close();

connection.close();

return city;

}

public int updateCity(City city) throws SQLException {

Connection connection = getConnection();

Statement statement = connection.createStatement();

StringBuilder query = new StringBuilder("update city set id = ").append(city.getId());

if (city.getName()!=null){

query.append(",name = '").append(city.getName()).append("'");

}

if (city.getDescription()!=null){

query.append(",description = '").append(city.getDescription()).append("'");

}

if (city.getHumidity()!=null){

query.append(",humidity = '").append(city.getHumidity()).append("'");

}

if (city.getTemperature()!=null){

query.append(",temperature = '").append(city.getTemperature()).append("'");

}

if (city.getUvIndex()!=null){

query.append(",uvIndex = '").append(city.getUvIndex()).append("'");

}

if (city.getWindSpeed()!=null){

query.append(",windSpeed = '").append(city.getWindSpeed()).append("'");

}

query.append(" where id = ").append(city.getId());

System.out.println(">>> Query >> " + query);

return statement.executeUpdate(query.toString());

}

private Connection getConnection() {

Properties info = new Properties();

info.put("user", user);

info.put("password", password);

Connection connection = null;

try {

connection = DriverManager.getConnection(url, info);

} catch (SQLException e) {

throw new RuntimeException(e);

}

return connection;

}

public List<FamousPlace> getFamousPlaces(int cityId) throws SQLException {

Connection connection = getConnection();

Statement statement = connection.createStatement();

String query = "select \* from famous\_places where city\_id =" + cityId;

System.out.println(">>> Query >> " + query);

ResultSet resultSet = statement.executeQuery(query);

List<FamousPlace> famousPlaces = new ArrayList<>();

while (resultSet.next()) {

FamousPlace place=new FamousPlace();

place.setCityId(resultSet.getInt("city\_id"));

place.setName(resultSet.getString("name"));

place.setUrl(resultSet.getString("url"));

place.setDescription(resultSet.getString("description"));

famousPlaces.add(place);

}

resultSet.close();

statement.close();

connection.close();

return famousPlaces;

}

public int saveFamousPlace(FamousPlace famousPlace) throws SQLException {

Connection connection = getConnection();

Statement statement = connection.createStatement();

String query = "insert into famous\_places(id, name, description, city\_id, url)" +

"values("+famousPlace.getId()+", '" +famousPlace.getName() + "', '" + famousPlace.getDescription() + "', '"

+famousPlace.getCityId()+"', '"+ famousPlace.getUrl() + "')";

System.out.println(">>> Query >> " + query);

statement.execute(query);

return famousPlace.getId();

    }

}

**city.java:**

import java.util.List;

public class City {

private int id;

private String name;

private String temperature;

private String humidity;

private String windSpeed;

private String uvIndex;

private String description;

private List<FamousPlace> famousPlaceList;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getTemperature() {

return temperature;

}

public void setTemperature(String temperature) {

this.temperature = temperature;

}

public String getHumidity() {

return humidity;

}

public void setHumidity(String humidity) {

this.humidity = humidity;

}

public String getWindSpeed() {

return windSpeed;

}

public void setWindSpeed(String windSpeed) {

this.windSpeed = windSpeed;

}

public String getUvIndex() {

return uvIndex;

}

public void setUvIndex(String uvIndex) {

this.uvIndex = uvIndex;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public List<FamousPlace> getFamousPlaceList() {

return famousPlaceList;

}

public void setFamousPlaceList(List<FamousPlace> famousPlaceList) {

this.famousPlaceList = famousPlaceList;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

    }

}

**FamousPlace.java:**

public int getCityId() {

return cityId;

}

public void setCityId(int cityId) {

this.cityId = cityId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getUrl() {

return url;

}

public void setUrl(String url) {

this.url = url;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

    }

}