***How we connect database to Web Application(MVC):***

Step 1: Download Microsoft SQL server management.

Step 2: Open Microsoft SQL server management.

Step 3: Server Type: database engine.

Server Name: DESKTOP-1LKGFVL

Authentication: Window Authentication.

then connect.

Step 4: Right click on Database> New Databases..>Enter database name then OK.

Step 5: Create web APplication(mvc).

Step 6: Open appsettings.json file and copy the below code.

"ConnectionStrings": {

"DefaultConnection": "Server=DESKTOP-1LKGFVL;Database=TestingDatabase;user id=DESKTOP-1LKGFVL\\Dell;password=;Trusted\_Connection=true;Encrypt=false;;MultipleActiveResultSets=true;"

}

Set all parameter according to your needs.

Step 7: Open Startup.cs file and copy below code.

using DatabaseConnectionDemo.dbModels;

using Microsoft.AspNetCore.Builder;

using Microsoft.AspNetCore.Hosting;

using Microsoft.AspNetCore.HttpsPolicy;

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Configuration;

using Microsoft.Extensions.DependencyInjection;

using Microsoft.Extensions.Hosting;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace DatabaseConnectionDemo

{

public class Startup

{

public Startup(IConfiguration configuration)

{

Configuration = configuration;

}

public IConfiguration Configuration { get; }

// This method gets called by the runtime. Use this method to add services to the container.

public void ConfigureServices(IServiceCollection services)

{

@@@@@@@@code you want to add in your code@@@@@@@@@@@@@@@@@@@@@@@@

services.AddDbContext<dbContext>(options =>

options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

dbContext.ConnectionString = Configuration.GetConnectionString("DefaultConnection");

@@@@@@@code end@@@@@@@@@@@@@@@@@

services.AddControllersWithViews();

}

// This method gets called by the runtime. Use this method to configure the HTTP request pipeline.

public void Configure(IApplicationBuilder app, IWebHostEnvironment env)

{

if (env.IsDevelopment())

{

app.UseDeveloperExceptionPage();

}

else

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.UseEndpoints(endpoints =>

{

endpoints.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

});

}

}

}

If Startup.cs is not persent but program.cs file is there then you have to add below code.

using Microsoft.EntityFrameworkCore;

using Microsoft.Extensions.Configuration;

using zzootaAdminApp.dbModels;

var builder = WebApplication.CreateBuilder(args);

// Add services to the container.

builder.Services.AddControllersWithViews();

@@@@@@@@code you want to add in your code@@@@@@@@@@@@@@@@@@@@@@@@

builder.Services.AddDbContext<dbcontext>(options =>

{

options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"));

});

dbcontext.ConnectionString = builder.Configuration.GetConnectionString("DefaultConnection");

@@@@@@@code end@@@@@@@@@@@@@@@@@

var app = builder.Build();

// Configure the HTTP request pipeline.

if (!app.Environment.IsDevelopment())

{

app.UseExceptionHandler("/Home/Error");

// The default HSTS value is 30 days. You may want to change this for production scenarios, see https://aka.ms/aspnetcore-hsts.

app.UseHsts();

}

app.UseHttpsRedirection();

app.UseStaticFiles();

app.UseRouting();

app.UseAuthorization();

app.MapControllerRoute(

name: "default",

pattern: "{controller=Home}/{action=Index}/{id?}");

app.Run();

Step 8: Create new folder name it dbModels, inside it create two file named tables.cs and dbContext.cs.

Step 9: Code you want to add in your dbContext.cs:

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

namespace DatabaseConnectionDemo.dbModels

{

public class dbContext : DbContext

{

public dbContext(DbContextOptions<dbContext> options) : base(options)

{

}

public dbContext()

{

}

public static string ConnectionString { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

optionsBuilder.UseSqlServer(ConnectionString);

}

}

//Tables

public DbSet<Country> Country { get; set; }

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Country>().ToTable("Country");

}

}

}

Code you want to code in your Tables.cs:

using System;

using System.Collections.Generic;

using System.ComponentModel.DataAnnotations;

using System.Linq;

using System.Threading.Tasks;

namespace DatabaseConnectionDemo.dbModels

{

public class Country

{

[Key]

public int CountryId { set; get; }

public string CountryName { set; get; }

public string Timezone { set; get; }

public int PhoneCode { set; get; }

public DateTime Addedon { set; get; }

public string AddedBy { set; get; }

public DateTime UpdatedOn { set; get; }

public string UpdatedBy { set; get; }

public bool Deleted { set; get; }

public bool Archived { set; get; }

public int TestId { set; get; }

//Relation with User

}

}

Step 10: Add below package from Tools menu> nuget Package manager> Manage nuget packages for Solution...

If your applcation is .net 3.1 then install 3.1.3 version otherwise latest.

Microsoft.EntityFrameworkCore

Microsoft.EntityFrameworkCore.Tools

Microsoft.EntityFrameworkCore.SqlServer

Step 11: Perform the below command

Add-Migration Any\_Name\_To\_your\_migration

update-database -context dbContext

NOTE: SOME TIME PROJECT NOT WORKING DUE TO NOT HAVING PROPER .NET SDK INSTALL IN IT.  
LINK FOR SDK:  
[Download .NET 6.0 (Linux, macOS, and Windows) (microsoft.com)](https://dotnet.microsoft.com/en-us/download/dotnet/6.0)  
  
  
  
***COMMAND USE TO IMPORT DATABASE USING CMD.***sqlcmd -S .{SERVERNAME} -d Supagas{DATABASENAME} -i D:\Zzoota Admin\zzootaAdmin\zzootaAdmin.sql{FILE\_PATH}  
  
  
  
  
  
***Connectivity with mysql server using entity framework.***appsettings.json: add connection string{ "ConnectionStrings": { "DefaultConnection": "Server=localhost;Database=stm;Uid=root;Pwd=;" }, "Logging": { "LogLevel": { "Default": "Information", "Microsoft.AspNetCore": "Warning" } }, "AllowedHosts": "\*"}dbcontext.csusing Microsoft.EntityFrameworkCore;using System.Xml;namespace MysqlDatbaseConnectivity\_.dbModels{ public class dbContext : DbContext { public dbContext(DbContextOptions<dbContext> options) : base(options) { } public DbSet<Student> Student { get; set; } }}package to downloadMySql.EntityFrameworkCore 7.0.2Microsoft.EntityFrameworkCore.Tools 7.0.5Program.csbuilder.Services.AddDbContext<dbContext>(options =>options.UseMySQL(builder.Configuration.GetConnectionString("DefaultConnection")));tables.cs public class Student { public int Id { get; set; } public string Name { get; set; } public string Description { get; set; } }  
  
If version compatibilty issue occurs then install one by one every version and check which version is stable.If you want to create dbContext in Class Library.Add dbcontext:using Microsoft.EntityFrameworkCore;using System;using System.Collections.Generic;using System.Linq;using System.Text;using System.Threading.Tasks;namespace STMDbContext.DbModel{ public class DatabaseContext: DbContext { public DbSet<WebFleetDeviceTable> FeetTable { get; set; } //protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) //{ // optionsBuilder.UseMySQL("Server=localhost;Port=3306;database=webfeetdatabase;Uid=root;Pwd=root;"); //} protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder) { optionsBuilder.UseMySQL( "Server=localhost;Port=3306;database=webfeetdatabase;Uid=root;Pwd=root;", options => options.MigrationsAssembly("STMDbContext")//above line is for YourMigrationsAssembly which you get from write click on project in which you create dbContext and go to properties then click on assembly information button.// ); } protected override void OnModelCreating(ModelBuilder modelBuilder) { base.OnModelCreating(modelBuilder); } }} 2. make sure the migration you are doing in any project must be target project to can set it by right click on solution explorer then properties then single startup project.3. if you are performing any migration you must select particular project in package anger console and always use -context after add-migration and update-databse.  
  
***Connectivity with mysql server using entity framework.***

appsettings.json: add connection string

{

"ConnectionStrings": {

"DefaultConnection": "Server=localhost;Database=stm;Uid=root;Pwd=;"

},

"Logging": {

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"AllowedHosts": "\*"

}

dbcontext.cs

using Microsoft.EntityFrameworkCore;

using System.Xml;

namespace MysqlDatbaseConnectivity\_.dbModels

{

public class dbContext : DbContext

{

public dbContext(DbContextOptions<dbContext> options)

: base(options)

{

}

public DbSet<Student> Student { get; set; }

}

}

package to download

MySql.EntityFrameworkCore 7.0.2

Microsoft.EntityFrameworkCore.Tools 7.0.5

Program.cs

builder.Services.AddDbContext<dbContext>(options =>options.UseMySQL(builder.Configuration.GetConnectionString("DefaultConnection")));

tables.cs public class Student

{

public int Id { get; set; }

public string Name { get; set; }

public string Description { get; set; }

}

////////mysql connection

public class stmContext : DbContext

{

public DbSet<traffio\_users> traffio\_users { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseMySQL("server=stm-server.mysql.database.azure.com;port=3306;database=stm;user=zzootaadmin;password=p8ZiwoQ47trKoYspDiLWTov0Fb3rt56gtdferDFN78fb16xjaZ");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

base.OnModelCreating(modelBuilder);

}

}

public class traffio\_users

{

public int id { set; get; }

public string first\_name { set; get; }

public string last\_name { set; get; }

}

var context = new stmContext();

var userList=context.traffio\_users.Take(2).ToList();

var context = new stmContext();

var userList=context.traffio\_users.Take(2).ToList();

***ADD BEARER TOKEN IN REQUEST HEADER:***public async Task<List<Person>> GetPersons() { List<Person> GetPersonData = null; try { ***// Set the bearer token in the request headers client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", "98b5b4214736b33964df81dd6c8a9da6dc333c04f5c9539348541990da2fc8651079332f67ba55666a84cbad16af6283fe0c35e90f13e0201098b27a910703ee");*** HttpResponseMessage response = await client.GetAsync("api\_third\_party/v1\_person/person/format/json"); if (response.IsSuccessStatusCode) { var formatedResponse = await response.Content.ReadAsStringAsync(); GetPersonData = JsonConvert.DeserializeObject<List<Person>>(formatedResponse); return GetPersonData; } } catch (Exception Ex) { } return new List<Person>(); }  
  
 ***DECLARE ACCESS TOKEN GLOBALLY***  
*public STMHandler()*

*{*

*client.BaseAddress = new Uri("https://app.traff.io/api\_third\_party/");*

*client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", AccessToken);*

*}* **SOME TIME SQL THROW NO COLUMN EXIST ISSUE** *var GetJobData = "SELECT \* FROM job ORDER BY \"id\" DESC LIMIT 50";***POSTGRESSQL QUERY TO IMPORT DATABASE:** *psql -h stm-booking-portal-db.postgres.database.azure.com -U zzoota -d wafel\_stm -f wafel\_stm.sql  
  
  
or  
  
psql -h stm-booking-portal-db.postgres.database.azure.com -U zzoota -d ashz\_stm > ashz\_stm.sql* **WHEN REQUESTION FOR UNAUTHORISED:** *response.StatusCode == HttpStatusCode.Unauthorized  
  
  
password for postsql: swati@zzoota.com*

*port 5432****MYSQL RUN IN COMMAND PROMPT:*** *first configure environment variable*

*then add below command*

*mysql -u root -p database\_name < file\_name(for importing database tableS)  
  
The Startup class is defined with a constructor that accepts an instance of IConfiguration interface.IConfiguration is used to retrieve application settings from the appsettings.json file.The ConfigureServices method is called by the runtime to register services with the dependency injection container.In the ConfigureServices method, several services are added to the container, such as Razor Pages, DbContext, Identity, Authentication, Authorization, Swagger, and SignalR, among others.The Configure method is called by the runtime to configure the HTTP request pipeline.In the Configure method, the application is configured to use HTTPS redirection, static files, routing, authentication, authorization, cookie policy, session, and signalR, among others.The Configure method also maps the endpoints for the application.There are some additional methods defined in the class, such as OnTokenValidated, which is used to handle token validation in Xero OAuth2 authentication.Lastly, there are some static variables defined in the class for connection string settings.*

*To create a listener in C#, you can use the SignalR library to listen for incoming messages on a specified port. Here's an example of how you can create a SignalR listener in C#:First, install the Microsoft.AspNet.SignalR package via NuGet Package Manager. You can do this by right-clicking on your project in Visual Studio and selecting "Manage NuGet Packages...", then searching for "Microsoft.AspNet.SignalR" and installing it.Next, create a class that inherits from the Hub class provided by SignalR. This class will contain the logic to handle incoming messages. For example:csharpCopy codeusing Microsoft.AspNet.SignalR;public class MyHub : Hub{ public void ListenData(string data) { // Your implementation here }}This code defines a MyHub class that includes a ListenData method. You can implement your custom logic within the ListenData method to handle incoming data.In your application, create an instance of the HubConfiguration class, and use its MapSignalR method to map the MyHub class to a URL endpoint. For example:csharpCopy codeusing Microsoft.AspNet.SignalR;using Microsoft.Owin.Hosting;using Owin;using System;class Program{ static void Main(string[] args) { var url = "http://localhost:8080"; using (WebApp.Start(url)) { Console.WriteLine($"Server running on {url}"); var hubConfiguration = new HubConfiguration(); hubConfiguration.EnableDetailedErrors = true; hubConfiguration.EnableJavaScriptProxies = true; app.MapSignalR("/myhub", hubConfiguration); Console.ReadKey(); } }}This code starts a new SignalR server on port 8080 and maps the MyHub class to the URL /myhub.To start listening for incoming messages, create an instance of the HubConnection class and connect to the SignalR server. For example:csharpCopy codeusing Microsoft.AspNet.SignalR.Client;var url = "http://localhost:8080";var connection = new HubConnection(url);var hubProxy = connection.CreateHubProxy("MyHub");hubProxy.On<string>("ListenData", data =>{ // Handle incoming data here});connection.Start().Wait();This code creates a HubConnection instance and connects to the SignalR server at http://localhost:8080. It then creates a proxy for the MyHub class and registers a callback for the ListenData method.That's it! You now have a SignalR listener in C# that can receive incoming messages. Note that you can replace MyHub with the actual name of your hub class, and ListenData with the actual name of your method. Also, make sure that you have the appropriate SignalR packages installed in your project.***ACCESS MODEL IN JQUERY** *var TripStatusData =@Html.Raw(Json.Serialize(Model));***NEW CONCEPT** *document.querySelectorAll(".mvc-grid").forEach(element => new MvcGrid(element)); } function BlockPage() { mApp.blockPage({ overlayColor: "#000000", type: "loader", state: "primary", message: "Loading..." }); } function UnblockPage() { mApp.unblockPage(); }* **Copy text from textbox using jquery:**

*$("#{body}").delegate("#{buttonId}", "click", function () {*

*$("#{textboxID}").removeClass("d-none");*

*var copyText = document.getElementById('{TextboxId}');*

*copyText.select();*

*document.execCommand('copy');*

*$("#{TextBoxId}").addClass("d-none");*

*});***CONVERT STRING TO DATETIME in JQUERY** *moment(item.dateTime).format("DD-MMM-YYYY HH:mm:ss")***RELAOD PAGE AFTER 1minutes** *setInterval(function () { setTimeout(function () { window.location.reload(); }, 60000) }, 60000)* **GOOGLE DYNAMIC WEATHER FORCAST** *<style>*

*\* {*

*font-family: Arial, Helvetica, sans-serif;*

*padding: 0;*

*margin: 0;*

*text-align: center;*

*}*

*.app {*

*/\*background-image: url("https://mdbcdn.b-cdn.net/img/Photos/new-templates/bootstrap-weather/draw1.webp");\*/*

*background-color: #1f567c;*

*color: #fff;*

*width: 100%;*

*font-family: Tahoma,Arial,Verdana,Segoe,sans-serif;*

*font-size: 1rem;*

*}*

*.app h1 {*

*font-size: 5em;*

*padding: 10px;*

*margin-left: 25px;*

*font-weight: 300;*

*}*

*.app h2 {*

*font-weight: 300;*

*letter-spacing: 5px;*

*}*

*</style>*

*<div class="app row">*

*<div class="col-5">*

*<h5 class="text-left mt-2" id="date0">7 Friday</h5>*

*<h2 class="mt-4" id="location1">Brisbane</h2>*

*<img src="https://cdn-icons-png.flaticon.com/512/1163/1163661.png" id="temp-icon0" style="width:80%">*

*<h2 id="temp-main0" class="mt-5">1.28°C</h2>*

*<h6 id="condition0">Snowy</h6>*

*</div>*

*<div class="col-1">*

*<h5 class="text-left mt-2" id="date1">Date</h5>*

*<img src="https://cdn-icons-png.flaticon.com/512/1163/1163661.png" id="temp-icon1" style="width: 80%">*

*<h2 id="temp-main1" class="mt-5">1.28°C</h2>*

*<h6 id="condition1">Snowy</h6>*

*</div>*

*<div class="col-1">*

*<h5 class="text-left mt-2" id="date2">Date</h5>*

*<img src="https://cdn-icons-png.flaticon.com/512/1163/1163661.png" id="temp-icon2" style="width: 80%">*

*<h2 id="temp-main2" class="mt-5">1.28°C</h2>*

*<h6 id="condition2">Snowy</h6>*

*</div>*

*<div class="col-1">*

*<h5 class="text-left mt-2" id="date3">Date</h5>*

*<img src="https://cdn-icons-png.flaticon.com/512/1163/1163661.png" id="temp-icon3" style="width: 80%">*

*<h2 id="temp-main3" class="mt-5">1.28°C</h2>*

*<h6 id="condition3">Snowy</h6>*

*</div>*

*<div class="col-1">*

*<h5 class="text-left mt-2" id="date4">Date</h5>*

*<img src="https://cdn-icons-png.flaticon.com/512/1163/1163661.png" id="temp-icon4" style="width: 80%">*

*<h2 id="temp-main4" class="mt-5">1.28°C</h2>*

*<h6 id="condition4">Snowy</h6>*

*</div>*

*<div class="col-1">*

*<h5 class="text-left mt-2" id="date5">Date</h5>*

*<img src="https://cdn-icons-png.flaticon.com/512/1163/1163661.png" id="temp-icon5" style="width: 80%">*

*<h2 id="temp-main5" class="mt-5">1.28°C</h2>*

*<h6 id="condition5">Snowy</h6>*

*</div>*

*<div class="col-1">*

*<h5 class="text-left mt-2" id="date6">Date</h5>*

*<img src="https://cdn-icons-png.flaticon.com/512/1146/1146860.png" id="temp-icon6" style="width: 80%">*

*<h2 id="temp-main6" class="mt-5">1.28°C</h2>*

*<h6 id="condition6">Snowy</h6>*

*</div>*

*</div>*

*@section Scripts{*

*<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>*

*<script>*

*$(document).ready(function () {*

*console.log("enter");*

*if (navigator.geolocation) {*

*navigator.geolocation.getCurrentPosition(getWeather);*

*} else {*

*console.log("Geolocation is not supported by this browser.");*

*}*

*function getWeather(position) {*

*var lat = position.coords.latitude;*

*var long = position.coords.longitude;*

*var API\_KEY = '7cad50b4166c4aaa99ad8ed4e044f451';*

*var baseURL = `https://api.weatherbit.io/v2.0/forecast/daily?key=${API\_KEY}&lat=${lat}&lon=${long}`;*

*console.log(baseURL);*

*$.ajax({*

*url: baseURL,*

*type: 'Post',*

*success: function (sdata) {*

*var location = sdata.city\_name;*

*console.log(location);*

*for (var i = 0; i < sdata.data.length-9; i++) {*

*// var icon = sdata.current.condition.icon;*

*var description = sdata.data[i].weather.description;*

*var temp = sdata.data[i].temp;*

*var date = sdata.data[i].datetime;*

*console.log(description);*

*var parts = date.split(/[- :]/);*

*var months = ['JAN', 'FEB', 'MAr', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC'];*

*var MonthName = months[parts[1] - 1];*

*var DateInWord = `${parts[2]} ${MonthName} ${parts[0]}`;*

*console.log(DateInWord);*

*//var Time = `${parts[3]}:${parts[4]}`;*

*if (description == "Thunderstorm with light rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Thunderstorm with rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Thunderstorm with heavy rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Thunderstorm with light drizzle") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Thunderstorm with drizzle") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Thunderstorm with heavy drizzle") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Thunderstorm with Hail") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146860.png");*

*}*

*else if (description == "Light Drizzle") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/5370/premium/5370296.png?token=exp=1641799636~hmac=ace03848cd00706e5c6cff196cb8b48a");*

*}*

*else if (description == "Drizzle") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/5370/premium/5370296.png?token=exp=1641799636~hmac=ace03848cd00706e5c6cff196cb8b48a");*

*}*

*else if (description == "Heavy Drizzle") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/5370/premium/5370296.png?token=exp=1641799636~hmac=ace03848cd00706e5c6cff196cb8b48a");*

*}*

*else if (description == "Light rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/1850/premium/1850736.png?token=exp=1641817902~hmac=717343901dc5e00e8166e46aac045495");*

*}*

*else if (description == "Moderate Rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/1850/premium/1850736.png?token=exp=1641817902~hmac=717343901dc5e00e8166e46aac045495");*

*}*

*else if (description == "Heavy Rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/3157/premium/3157504.png?token=exp=1641799871~hmac=86d2af149c6c5327fdcf233590dd0b33");*

*}*

*else if (description == "Freezing rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/1850/premium/1850736.png?token=exp=1641800172~hmac=26fe3e0faef63fb23c7e14fbc471c904");*

*}*

*else if (description == "Light shower rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/1850/premium/1850736.png?token=exp=1641800172~hmac=26fe3e0faef63fb23c7e14fbc471c904");*

*}*

*else if (description == "Shower rain") {*

*console.log(description);*

*$('#temp-icon' + [i]).attr("src", "url(https://cdn-icons.flaticon.com/png/512/1850/premium/1850736.png?token=exp=1641800172~hmac=26fe3e0faef63fb23c7e14fbc471c904");*

*}*

*else if (description == "Heavy shower rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/1850/premium/1850736.png?token=exp=1641800172~hmac=26fe3e0faef63fb23c7e14fbc471c904");*

*}*

*else if (description == "Light snow") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Snow") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Heavy Snow") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Mix snow/rain") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Sleet") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/3262/premiu…=1641798559~hmac=af6468672df1088bc2a09e979e62237d");*

*}*

*else if (description == "Heavy sleet") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/3262/premiu…=1641798559~hmac=af6468672df1088bc2a09e979e62237d");*

*}*

*else if (description == "Snow shower") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Heavy snow shower") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Flurries") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2315/premium/2315309.png?token=exp=1641800069~hmac=a8a596ab992e547b06b0c2d5a32ac894");*

*}*

*else if (description == "Mist") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1779/1779807.png");*

*}*

*else if (description == "Smoke") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1779/1779807.png");*

*}*

*else if (description == "Haze") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1779/1779807.png");*

*}*

*else if (description == "Sand/dust") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1779/1779807.png");*

*}*

*else if (description == "Fog") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1779/1779807.png");*

*}*

*else if (description == "Freezing Fog") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1779/1779807.png");*

*}*

*else if (description == "Clear sky") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/890/890347.png");*

*}*

*else if (description == "Few clouds") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/2698/2698213.png");*

*}*

*else if (description == "Scattered clouds") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/2698/2698213.png");*

*}*

*else if (description == "Broken clouds") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons-png.flaticon.com/512/1146/1146869.png");*

*}*

*else if (description == "Overcast clouds") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2531/premium/2531612.png?token=exp=1641798675~hmac=a7a1044cd9a24c4528935d9dd83015b2");*

*}*

*else if (description == "Unknown Precipitation") {*

*$('#temp-icon' + [i]).attr("src", "https://cdn-icons.flaticon.com/png/512/2531/premium/2531612.png?token=exp=1641798675~hmac=a7a1044cd9a24c4528935d9dd83015b2");*

*}*

*$('#temp-main' + [i]).html(temp + "°C");*

*$('#condition' + [i]).html(description);*

*$('#location' + [i]).html(location);*

*$('#date' + [i]).html(DateInWord);*

*}*

*},*

*error: function (edata) {*

*alert("error");*

*}*

*});*

*}*

*});*

*</script>*

*}*