Date transfer method change(no pop)

Set up websocket connection between backend and frontend, test it.

Create update tempvalues func for backend

Bind temp values array between frontend and backend

Ajax polling method for Data push(Temperature) from backend to webpage,with timer for rerun(10 sec) (backend+fronted), replacing normal get function.

Receive and bind new data to frontend html display variables(backend)

Ajax polling method for Data push(Temperature) from backend to webpage,with timer for rerun(1 sec) (backend+fronted)

--stash

\*Research WebApp/Website difference

C# -> WCF integration -> REST interface usage -> WebInvoke POST method for Data transfer between backend and frontend

.CSV language file is made by me, within which I write every line of text in a different language, and link the frontend display to the backend.csv section through a language key that displays all text in said language

Dictionary is an angular addon that helps me sort through the csv file using keys and pairs(I believe)

\*Understand the task

\*Research POST method meaning

\*Research last 2 lines of instruction 1, language file .CSV and Dictionary for language messages(they want me to use some sort of dictionary, build an algorithm that will automatically translate all text to a different language by going through dictionary, or do so using an already built dictionary element.)

\*Create Angular environment

\*Launch basic angular environment page

\*Route from default page to web building page

\*Rotem Logo component

<!-- <button class="btn-submit" (click)="nothing()" value="Submit">

        {{ ilan }}

       </button> -->

ng-show="myform.email.$invalid && myform.psw.$touched"

ng-show="myform.psw.$invalid && myform.psw.$touched"

\*Complete the task

\*Frontend building

\*Login page

\*Form component

\*Email and password validity verification

\*Add “if” function to verify above task

\*Sign up page

\*Sign up text component

\*Form and sign up button component

\*Back to login button component

\*Logged in page

\*Information component

\*Variable receiving and storing

\*Mandatory list fulfillment upgrade

\*Login page

\*Sign up page

\*Variable receiving and storing

\*Backend integration

\*Dictionary integration

Learn how to create views in .net and redirect to view from controller

Integration into xl csv

Language csv

how to serve angular app from .net

The server should serve the angular app while injecting the translation object (for the corresponding language) using a global js variable in the “window”, the angular app will use the translation object from the “window”

Final page view

Log in count/time update

Language file

Angular update to 1.8?, maybe, if required.

UpdateLanguage func backend and frontend

Frontend post values

Backend post creation

Create xl file

Configure xl file

Return post from backend into frontend and add to service page

Build AngularJS table

Send User info on login and signup to angular Js table page

Create currentUser database

On Angular signUp and Login backend post functions, update currentUser database

Create backend get function for currentUser

Create AngularJS Frontend get function for currentUser

Update AngularJS html to display currentUser’s required properties

Extract Json user and backend

Display them above TableProperties

Display User info above AngularJS table

// var interval = 1; // 5 seconds

        // interval = interval \* 1000;

        // var obj = {}; // specify the parameters

        // window.setInterval(poll, interval);

        // function poll() {

        //     $.ajax({

        //         url: `${window.location.origin}/WeatherForecast`,

        //         type: "GET",

        //         data: obj,

        //         dataType: "json",

        //         traditional: true,

        //         contentType: "application/json; charset=utf-8",

        //         success: function (data) {

        //             return data

        //         },

        //         error: function () {

        //             alert("An error has occured!!!");

        //         }

        //     });

        // }

 // for(weather in $scope.weatherList){

        //   weather.date = d

        //   $scope.weatherList[weather].date = $filter('date')

        //     (weather.date, 'dd/MM/yy');

        // }

  //   $scope.$watch('$viewContentLoaded', function(){

  //     $scope.gfg = "GeeksForGeeks"

  //     let userLocal=localStorage.getItem('userDetails');

  //     if(userLocal!==null){

  //     return JSON.parse(userLocal);

  //     }

  //     return null;

  //  });

  //  $scope.$watch('$viewContentLoaded', function(){

  //   let currentLanguage = JSON.parse(localStorage.getItem('currentLanguage')||'');

  //   return currentLanguage;

  //  });

 //   $scope.GetUser = function(value) {

  //     let userLocal=localStorage.getItem('userDetails');

  //     if(userLocal!==null){

  //       return JSON.parse(userLocal);

  //     }

  //     return null;

  //   };

      <!-- <p>{{LoggedInLanguage.Hello}} {{user.username}}</p>

      <p>{{LoggedInLanguage.Last\_Log}} {{user.lastLogin}}</p>

      <p>{{LoggedInLanguage.Log\_Count}} {{user.loginCount}}</p> -->

    // $scope.nameFilter = null;

    // $scope.user = {};

//public async Task StartTimer(CancellationToken cancellationToken)

//{

// //await Task.Run(async () =>

// //{

// // while (true)

// // {

// // UpdateWeather();

// // await Task.Delay(1000, cancellationToken);

// // if (cancellationToken.IsCancellationRequested)

// // break;

// // }

// //});

//}