Weather Forecasting Application using Web-based Model-View-Whatever Framework

[1]Kathiravan Srinivasan, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India

[2]Anant Nema, Department of Computer Science and Information Engineering, National Ilan University, Yilan City, Taiwan, (R.O.C) Department of Computer Science and Engineering, The LNMIIT, Jaipur, India

[3]Chao-Hsi Huang, Department of Computer Science and Information Engineering, National Ilan University, Yilan City, Taiwan, (R.O.C)

[4]Tung Yang Ho, Department of Computer Science and Information Engineering, National Ilan University, Yilan City, Taiwan, (R.O.C)

***Abstract*--Weather is the state of the atmosphere at a given place and time in regards to heat, cloudiness, dryness, sunshine, wind, and rain. Of all the geophysical phenomena weather is the most significant one that influences us. Weather can vary greatly and largely depends on climate, seasons and various other factors. The chief goal of this work is to get the weather forecast of any city throughout the world through an application. This paper aims at creating a web application using javascript framework AngularJS.**

I. INTRODUCTION

In this work, a Single Page Weather forecasting Application in AngularJS 1.x using Model View Whatever (MV*) framework[1][2]. Model-View-Whatever paradigm is a term used to indicate the capability of the framework being used to give the option to choose from Model-View-Controller (MVC) or Model-View-ViewModel (MVVM) or any other approaches. Angular developers aides us to use "whatever" of the following systems. There's MVC, MVVM, Model-View-Presenter (MVP), and probably some more. The * is just a wildcard, not something specific. All the primary fundamentals of AngularJS are used in this application like ngRoute and ngResource dependencies, directives, routes, controllers, and so on. It has two pages, one is the homepage that takes the input as a city from the user, and the other is the Forecast page that will show the weather forecast of two, five and seven days. It is a single page application, so these two pages are to be set up using the routing within the AngularJS.

Model is data; the view is the HTML template for the component. The controller controls the data of Angular JS application[3]. It can be metaphorically related to Television(TV), there are various channels, with different information on them supplied by cable provider(the Model). The TV screen displays these channels to us (the View). Further, pressing the buttons on the remote controls affects what we see and how we see the Controller. After the change of data in the model, this will get reflected in the view. Besides, the change of data in the view, which in turn causes an update in the model. Also, this process takes place instantaneously and in an automatic manner that ensures the regular update of the model and the view. Additionally, the controllers control the applications in AngularJS. Further, due to the model and the view's instantaneous synchronization, this makes the controller to part away from the view, and it only focuses on the model data. Moreover, this assignment is performed using data binding in AngularJS; the view will reflect any changes made in the controller.

In AngularJS, dependency injections offer a function with an object. Instead of creating an object within a function, the task is transferred to the function, similar to scope services. Angular parse out the parameters and makes an array of it, and if the $scope is passed to it, it sees and passes or injects the object named $scope to the function. The scope is a service that's a part of the angular model [4]. So in this application $scope, $resource and custom service, i.e., cityService are injected.

Directives are most essential components of any angular js application [4]. Although AngularJS ships with a wide range of directives. It is something that introduces new syntax.

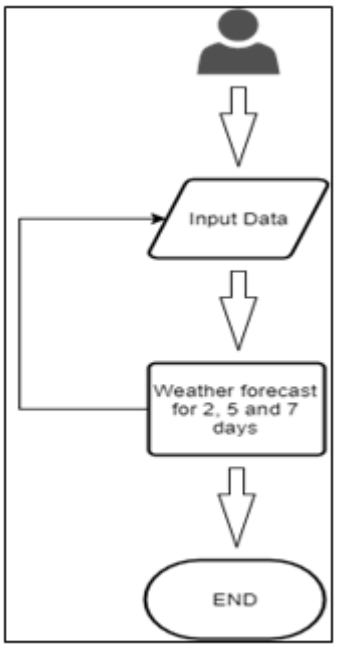

Figure 1. Flow Chart of Application

In this application, custom directives are used to show the temperatures of two, five and seven days. It is not necessary to use custom directives, but by not using them there may occur repetition of elements, which may cause a problem, thereby its preferable to use it for a broad application. Therefore, it is a good practice to wrap up the reusable code, so that it can be reused more efficiently. Also, it becomes easy for the developer to use it, it also makes the code cleaner and more manageable. Custom service is used to share data between these two pages [4]. For example, to send the name of the city from homepage to forecast page services are used. Watchers and digest cycle will keep track of the name input[5-9].

II. PROPOSED MODEL

The user will give the name of the city on the homepage and in the next page by default two days forecast can be seen of that city. The user can select from 2, 5, 7 and get the corresponding days forecast. It uses MV* model, so inside the communication flow what happens is, a client calls/initiates the HTML(View) with some input. Then the controller gets the inputs and knows what the required task is. Then the controller calls the Factories, Services, so on, inside it (Model) to prepare the desired business specific output bound to the given input. The Model then processes the input and gives the desired output to the controller. Then the controller makes some display specific adjustments; then the HTML will display. In this application desired output is the temperature of the city. In this application, dependencies like ngRoute and ngResource are used. ngRoute is to navigate to different pages reloading the page so that it remains single page application[7]-[9] whereas ngResource is used to bind the data from API to the application page. ngResource is a factory which creates a resource object that lets us interact with RESTful server-side data sources[8], this service also wraps up the HTTP service that comes with angular js, that is why resource service makes it easier to get the data.

$resource("API-Link",{callback : "JSON_CALLBACK"},{ get: { method : JSONP} });

As the data is coming from outside the web application, i.e., from an API, this may cause a security problem, the browser may think it as a hack attempt, so that can be resolved by giving JSON callback using JSONP. Writing this will ensure that application can interact with the API and browser will not a complaint about that. Temperature that comes from the API is in Kelvin to convert it to Fahrenheit using the formula,

$$F = 9/5(K * 273) + 32 \quad (1)$$

where F is the temperature in Fahrenheit and K is in Kelvin. Using the service called routeParams, i.e., route parameters we can navigate application pages to show forecast for two days, five days and seven days.

Just to get the better user experience, if the forecast is of two days, two will be highlighted, if it is for five days, five will be highlighted, so that user can quickly figure out that forecast is for how many days.

III. RESULTS

In figure 2, it is the Home page of the web application, which requires the name of the city as an input.

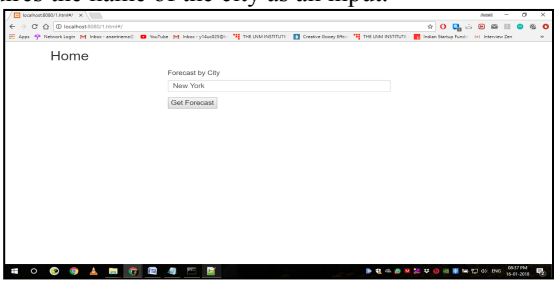

Figure 2. Home Screen of Application

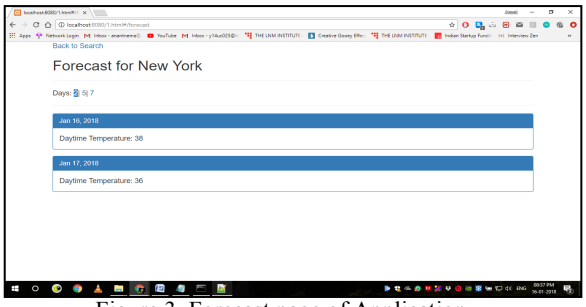

Figure 3. Forecast page of Application

In figure 3, it is the web page, which shows a two-day forecast of the inputted city.

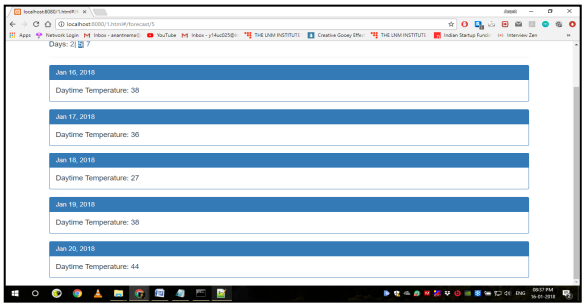

Figure 4. Forecast page of Application

In figure 4, it is the web page, which shows five days forecast of the inputted city. Forecast of seven days is same as above only, but with a forecast of 7 days.

IV. CONCLUSION AND FUTURE WORK

This application based on web technology AngularJS 1.x (Javascript framework) can show the weather forecast of two, five and seven days of any city in the world. As the data is coming from an API, so other parameters can also be considered in the application like wind speed, humidity, pressure, latitude, longitude, and so on. Along with that data can also be visualized which can perhaps give the more detailed understanding or information about the weather condition of a given place.

REFERENCES

[1] https://www.infragistics.com/community/blogs/b/nanil/posts/exploring-javascript-mv-frameworks-part-1-hello-backbonejs

[2] https://www.w3schools.com/angular/angular_controllers.asp

[3] https://angularjs.org/

[4] E. E. C. Osorio, SeokYoon Kang, Bum-Su Kim, JoHo Lim, Kyong Hoon Kim and Ki-Il Kim, "Development of data collecting system for forecasting with meteorological sensors," 2017 International Conference on Information Networking (ICOIN), Da Nang, 2017, pp. 453-456.

[5] M. M. u. Hussain, B. Avci, G. Trajcevski and P. Scheuermann, "Incorporating Weather Updates for Public Transportation Users of Recommendation Systems," 2016 17th IEEE International Conference on Mobile Data Management (MDM), Porto, 2016, pp. 333-336.

[6] K. Ramar and T. T. Mirnalinee, "A semantic web for weather forecasting systems," 2014 International Conference on Recent Trends in Information Technology, Chennai, 2014, pp. 1-6.

[7] K. Kojima et al., "Risk Management of Heatstroke Based on Fast Computation of Temperature and Water Loss using Weather Data for Exposure to Ambient Heat and Solar Radiation," in IEEE Access, vol. PP, no. 99, pp. 1-1.

[8] Y. Kolokolov, A. Monovskaya, V. Volkov and A. Frolov, "Intelligent integration of open-access weather-climate data on local urban areas," 2017 9th IEEE IDAACS, Bucharest, 2017, pp. 465-470