Job Seeker

WAP Final Project Report

Project Plan

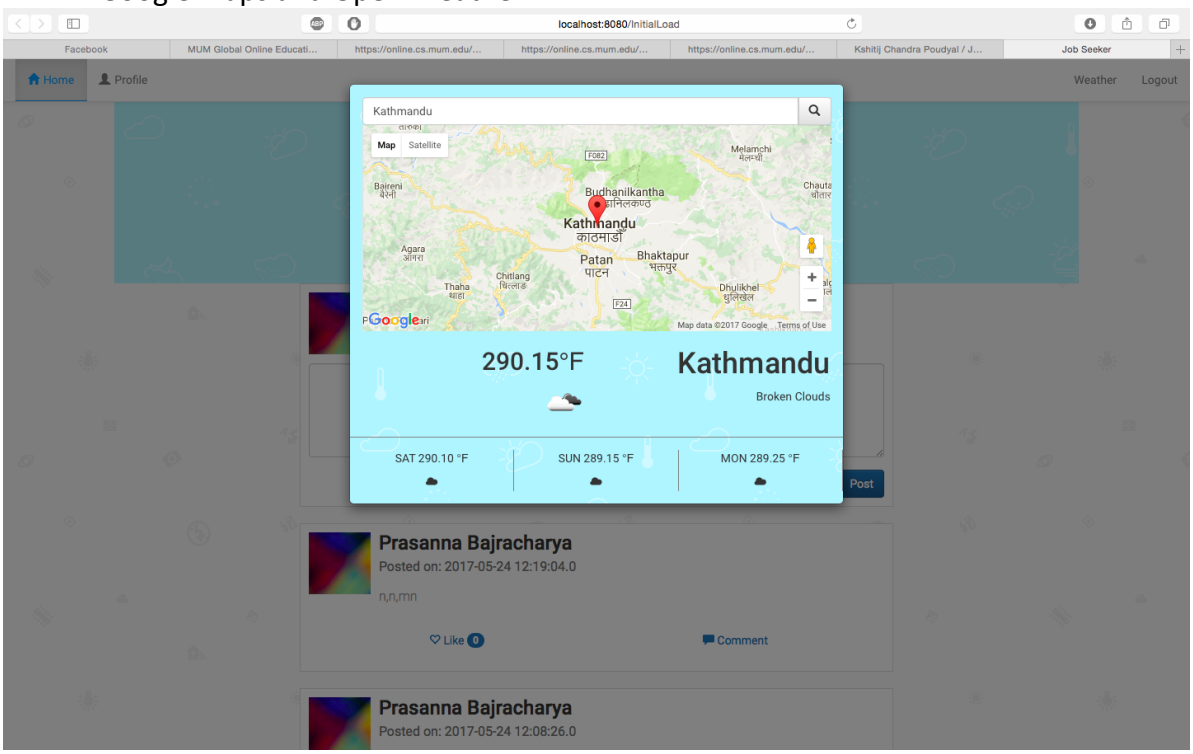
SN	TASK	PROGRESS
1	Layout Design (Whole Project)	Done
2	Database	Done
3	Registration Functionality	Done
4	Login Functionality	Done
5	Like Button Functionality	Done
6	Comment Button Functionality	Done
7	Home Page	Done
8	Update Post	Done
8	Profile Page	Done
9	Update User Profile	Done
10	Delete User Post	Done
11	Weather Information Implementation	Done
12	Google Map Implementation	Done

My Tasks and Status

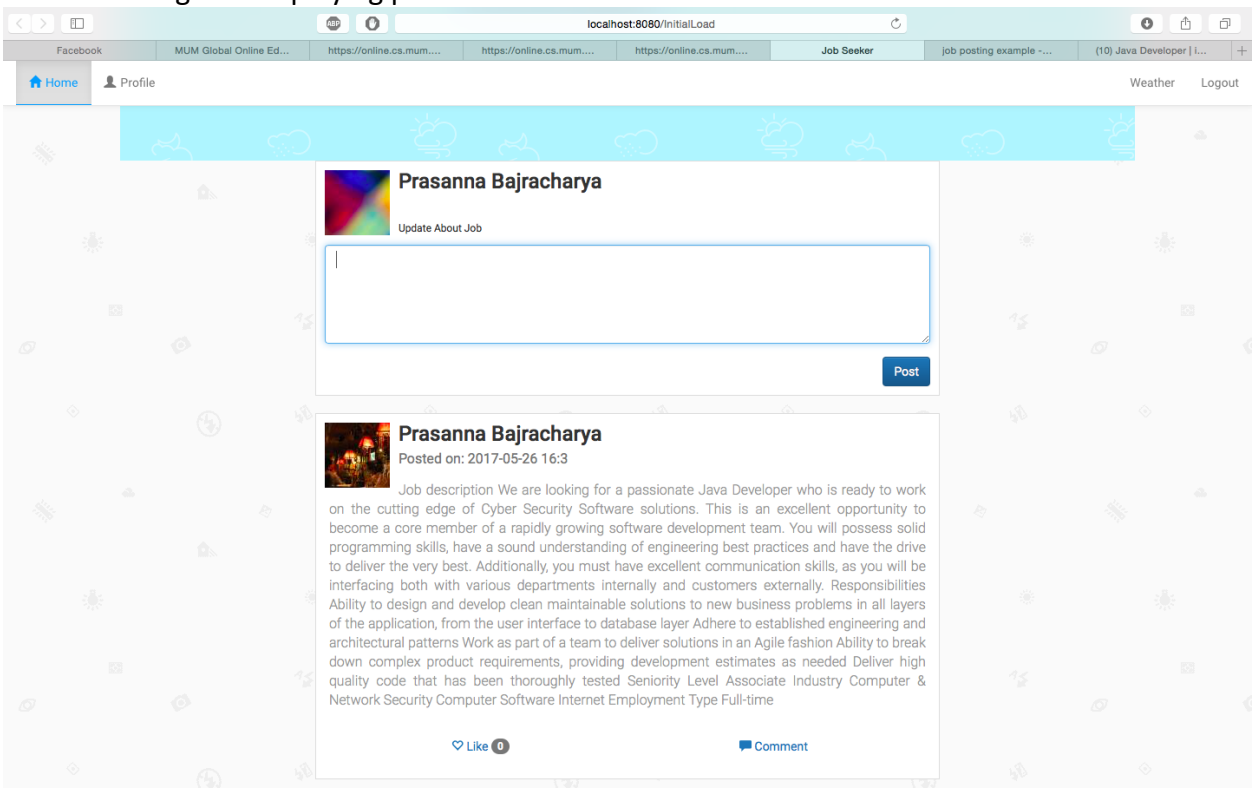
Tasks	Status	Remarks
Research on Google Maps API	Done	Configured and obtained API keys
Research on Open Weather API	Done	Configured and obtained API keys
Study method to fetch and interpret weather and forecast JSON information including icons	Done	Studied API methods and JSON parsing
Display current weather and display OpenWeatherMap icons	Done	Displaying icon was slight tricky than other JSON elements
Display 5 Day weather forecast	Done	Only 3 days forecast has been displayed
Display Google Maps with required coordinates and zoom	Done	Latitude and longitude info is fetched from weather api and fed into google maps
Overlay Weather information on top of Google Maps	Not Done	Faced issue in JSON conversion so could not be integrated in final project
Get coordinate information for the current logged on user from user profile	Not Done	Did not seem relevant so not implemented due to time restriction
Get coordinate information for user from current geolocation	Done	If user does not enter location info then it is fetched from browser location
Search and display weather information about their destination cities/zip code	Done	Search can be performed on the basis of both city name and zip code
End-to-end Use case: **Adding and Displaying Job posts**	Done	The main purpose of the application to add job posts and retain them permanently in database and display job postings from other users was successfully completed
Database Design	Done	Identified required info and created database
Create front end UI using Bootstrap CSS framework	Done	
Extra credit options – Responsive Design	Done	

Screenshots

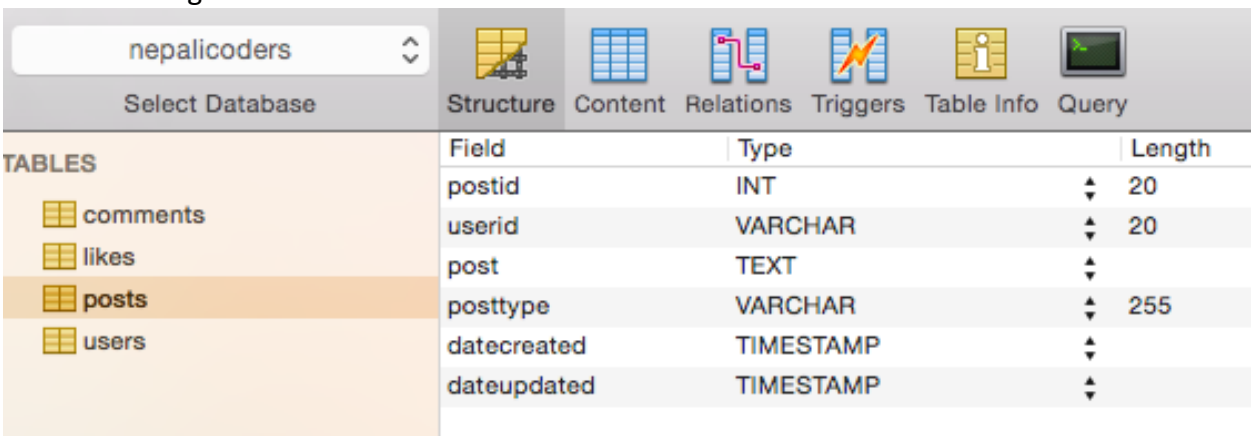
1. Google maps and Open Weather API



2. Adding and displaying post



3. Database Design



Sample Code

```
/**
 * Created by prasannabajracharya on 5/21/17.
 * Code to display weather information and google maps
 * If user does not input location information then it is fetched from current browser          location
 */
$(function () {

    $("#location-btn1").click(function () {

        var api = "http://api.openweathermap.org/data/2.5/weather?q=";

        var apiKey = "&appid=5039f6a0d872c67ff627191515c6304e";

        var zipCode = $("#location-info").val();

        var url = api + zipCode + apiKey;

        if (zipCode != "") {

            const initMap = data => {
                var lat;
                var lng;

                lat = data.coord.lat;
                lng = data.coord.lon;

                var weatherDesc = data.weather[0].description;
                var iconCode = data.weather[0].icon;
                var iconUrl = "http://openweathermap.org/img/w/" + iconCode + ".png";
                var temp = data.main.temp;
                var pressure = data.main.humidity;
                var city = data.name;

                $(".city").text(city);
                $(".type").text(weatherDesc);
                $(".degree").html(temp+"&#8457;");
                $(".weather-image").html("<img src='" + iconUrl + "'>");

                console.log("Weather : " + weatherDesc);
                console.log("Temp : " + temp);
                console.log("Pressure : " + pressure);
                console.log("City : " + city);

                var coordinates = {lat, lng};

                const map = new google.maps.Map(document.getElementById('map'), {
                    center: coordinates,
                    zoom: 10
```

```
                });

                var marker = new google.maps.Marker({
                    position: coordinates,
                    map: map
                });
            };

            $.getJSON(url, initMap);

        } else {

            var lat;
            var lng;

            getLocation();

            function getLocation() {
                if (navigator.geolocation) {
                    navigator.geolocation.getCurrentPosition(showPosition);

                } else {
                    $("#map").HTML = "Geolocation is not supported by this browser.";
                }
            }

            function showPosition(position) {
                lat = position.coords.latitude;
                lng = position.coords.longitude;

                var coordinates = {lat, lng};

                const map = new google.maps.Map(document.getElementById('map'), {
                    center: coordinates,
                    zoom: 10
                });

                var marker = new google.maps.Marker({
                    position: coordinates,
                    map: map
                });

            }
        }

    });

});



/**
    • Code snippet to insert Job post into database through servlet call and
    • Display information from database in the frontend application
**/

$("#comment").click(function (e) {
    e.preventDefault();

    $.post("/SubmitPost", {"post": $("#update-status-textarea").val()}).done(function (response) {
        if (response.status === "success") {
            let post = JSON.parse(response.post);

            $("#update-status-textarea").val("");

            const content = `<div class="row">
                <article class="col-md-12  col-centered article-container">
                    <header class="clearfix">
                        <img class="img-responsive" src="http://lorempixel.com/75/75" alt="profile-
picture"/>
                        <h1>${post.postedby}
                            <br>
                            <small>Posted on: ${post.datecreated}</small>
                        </h1>
                        <p>${post.postcontent}</p>
```

Essay

After completion of this project, we learnt a lot about dynamic web programming. I got chance to practically implement all the concepts of HTML, CSS, Javascript, JSP, Servlet and AJAX that I learnt in class. I have heard about MVC pattern a lot but got opportunity to actually implement in project. I came to know that using APIs helps us to implement complex features quickly and easily. I also learnt about making AJAX calls and JSON parsing.

Working in a team was really fun and interesting. Different people had different opinions and way of doing things. Having open discussion and meetings helped us resolve issues and also helped in improving good communication skills and leadership skills. I am amazed by the fact that different ideas of different people come together to create a brilliant useful application.

We face many technical difficulties in working in a team. Three of us were working in different parts of the application so consolidating them was a challenge. We used Gitlab to integrate our code and had to sit together whenever there was conflict in code. We soon realized that we need to update and pull our code very often to avoid conflicts. In addition to this, we also faced challenge in having integrated database. We installed centralized database in one of our team members and other team members used client application Sequel pro to just access that database.

I was learning dynamic web programming for the first time so I got stuck in coding many times but my team members were really helpful and helped me resolve issues. It was a great learning experience both technically and working in a team.