4004-736 Project 1: Interactive Form Elements

**Project Title**: Searching Places – A place directory

**Name**: Hardik Shah

**Project Description**:

Searching Places directory uses nodes from my dataset to display available locations and zero down on a destination to display categories like restaurants, bars, beach parks, medical centers, etc. on Google Map with details like address of place, contact number, user rating and reviews. During selection of the available locations, the map zooms up/down dynamically with a zoom level corresponding to the location selected to that time. Eventually, when the last node is reached and the place category is selected, the final location is centered on the map along with the places which are pulled from Yelp API and mapped on the Google Map. I have customized the Yelp API and Google Maps API to a great extent so it can receive input through just one function (which is initMap() function) which includes the written Google Geocoding client from scratch which accepts an input address and gives me the Latitude and Longitude of that address. So that is my first step for displaying the result after which the Yelp API takes over. After you see the selected category mapped, you will see a ‘Tell a Friend about Searching Places?’ form. The form allows updating any friend of yours about the Searching Places project. There’s required validation built-in and if the form passes, it sends an E-Mail to the contact for the update.

I have also included animation on Dropdowns and created a jazzy Web 2.0 design (passes W3C XHTML validation) which does in line with the theme of the topic.   
  
The dataset is JSON and I’m using AJAX to retrieve the data asynchronously. This allows the data to be dynamic and allows it to change anytime. For JSON support in IE 7, I am using ‘JSON-sans-eval’ library. The dataset is scalable and you can include more nodes which will create more Dropdowns as per the project requirements.  
  
There’s one feature which I call it as 'I'm Feeling Lucky!' . It uses the HTML5 Geolocation support in browser to detect the coordinates of the location you are currently at and then maps the selected category in your location. For browsers, which do not support HTML5 Geolocation, it shows an alert message.

**Files used in the project**:

1. **Index.html** – The markup file. Calls the checkBrowser() function to test browser compatibility and then loads all CSS styles, HTML code and scripts.
2. **sendMail.php** – PHP file to receive the name and E-Mail of the friend as POST and sends the contact an E-Mail with the message.
3. **robots.txt** and **sitemap.xml** – Used for search engine optimization
4. /scripts/**js\_utility.js** –JavaScript utility file for project. Has short-hand function defined to get DOM object of an elements from its ID, trim() function for trimming strings and more. Also has the ‘checkBrowser()’ function which checks for cross-browser compatibility.
5. /scripts/**loader.js** – Has the places class, its constructor and subsequent functions and variables. It’s the core script file which loads the JSON data from dataset.js via AJAX asynchronously, populates all controls, handles all events and conditions, display and validates the form and calls the initMap() function (in yelp-map-script.js) which updates the Google map depending on the parameters passed.
6. /scripts/**yelp-map-script.js –** Has the initMap() function which is the entry point to using Google Maps API v2.0 and Yelp API v1.0. The initMap() function gets the coordinates of an address using Google Geocoding client (written by me from scratch) or gets it directly from the HTML5 Geolocation feature. The file also has the functions to load the map depending on the location coordinates passed and search term passed from initMap() function.
7. /scripts/**dataset.js** – Is the JSON dataset file for the project. Has the node for available countries, states, cities, towns. Each node has its label, node level and zoom level associated. The file also has the available Yelp categories like Restaurants, Bars, Beach Places, etc. and their search terms. The file also has the two form field details like field name, label, display size and max size allowed. The node data is in the form –

Label name, Node level, Zoom level, --Select--, Value1, Value2, Value3 ....

Note: The dataset is scalable such that you can add more nodes (i.e. more than 3 nodes) to increase the resulting Dropdowns displayed. Example – Try USA | New York | Rochester | Jefferson Road.

Note that while adding nodes, also increase the corresponding zoom levels and the zoom level of the category level to be displayed for expected results.

1. /scripts/**json-minified.js** – The ‘JSON-sans-eval’ library for loading JSON object from a JSON file for browsers which do not support JSON (i.e. IE 7).
2. /scripts/**cookies.js** – Script file for storing and retrieving cookies.
3. /styles/ - CSS styles for design and formatting.
4. /images/ - Images used in the project.

**Additional item covered in the project**:

1. Good reusable and scalable dataset and loader.js code. Can use more than 3 nodes.
2. Works across all types of browsers (including Internet Explorer 7) and directs to Firefox download page for non-modern browsers and IE 5 on Mac.
3. Applied customization on Yelp API and Google Maps API for plotting the categories on map and displaying the details.
4. Uses HTML5 Geolocation for the ‘I’m Feeling Lucky!’ feature for finding the current location of the visitor and displaying the results
5. The form includes required validation, data persistence using Local Storage/Cookies. Sends E-Mail to the recipient on successful form validation using PHP E-Mail capability.
6. Usage of utility file (for ease of use and code maintainability) and Object-Oriented JavaScript paradigm for writing code
7. Uses Web 2.0 design and Google Web Fonts for web typography. Passes XHTML 1.0 W3C Validation check.
8. Made use of robots.txt and sitemap.xml for SEO(Search Engine Optimization).