TRAVELER’S EYE

**Introduction:**

Being international students we all understand the intricacies involved in travel. The stress of keeping track of all the information about the destination and the travel, especially when it comes to travelling alone, takes away from the fun one can have while travelling. Traveler’s eye is aimed to make a traveler’s life easy by providing relevant useful information whenever the user needs it, at a simple button click or the touch of a finger.

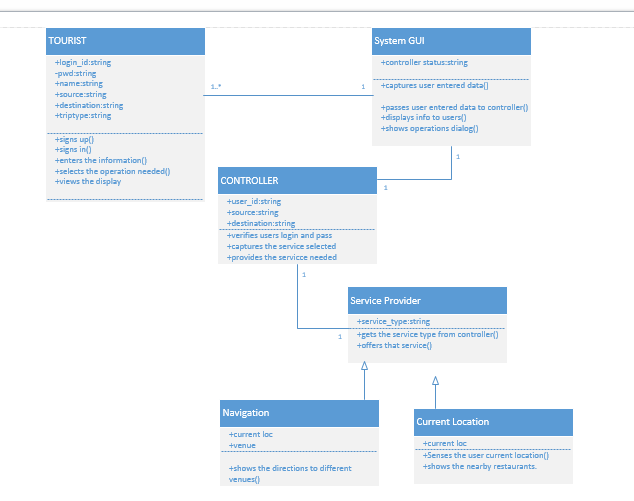
Services Designed in First Increment:

In the first increment we created the user interface.

Here along with user interface two of the services are implemented.

1. In the first service we obtain the current location of the user and show the restaurants nearby his location.
2. In the second service we show the direction from source to destination.

CLASS DIAGRAM FOR FIRST INCREMENT:



**Classes:**

Tourist: This class refers to the user of the application. The user should first sign up with application and them login into it and give his details. And then can use the services of the application.

SystemGUI: This is the interface between the user and the application. It captures the users data and passes it on to the controller .And displays the useful information to user as requested by the controller.

Controller: This is the heart of the whole application. It analyses everything. It updates and verifies database. Contacts the service provider and the brings the service needed into play.

Service Provider: This is used to provide services to the application. Each service is asub class to this super class.

1)Current Location: The current location of the user is identified and the restaurants nearby are located.

2)Navigation: This shows the navigation from one place to other.

**Relationship-Multiplicity:**

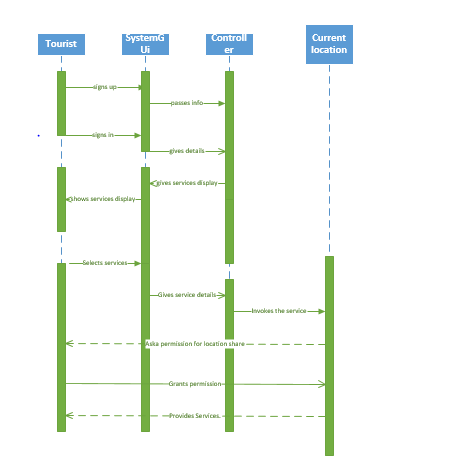
Tourist-SystemGUI: The SystemGUI is same and it can be used by multiple user’s .Hence the multiplicity is one or many-one. The relationship is association relationship.

SystemGUI-Controller: There is a purely one to one relationship between these two. As only single piece of them exists.They have an aasociation relationship.

Controller-Service Provider: These both also have association relaitonsip.And they have one to one multiplicity.

Service provider-navigation, service provider-current location: These have parent child relationship. The Service provider is the parent class and the individual services are the child classes.

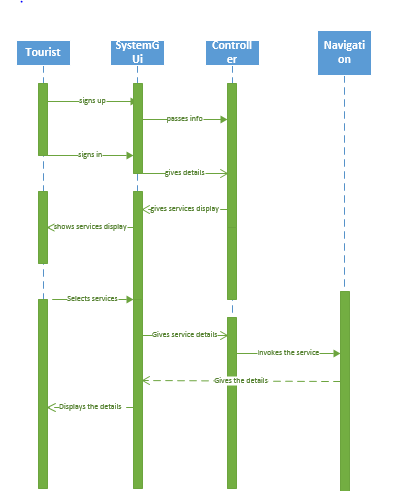
**SEQUENCE DIAGRAM FOR CURRETN LOCATION:**



**Steps:**

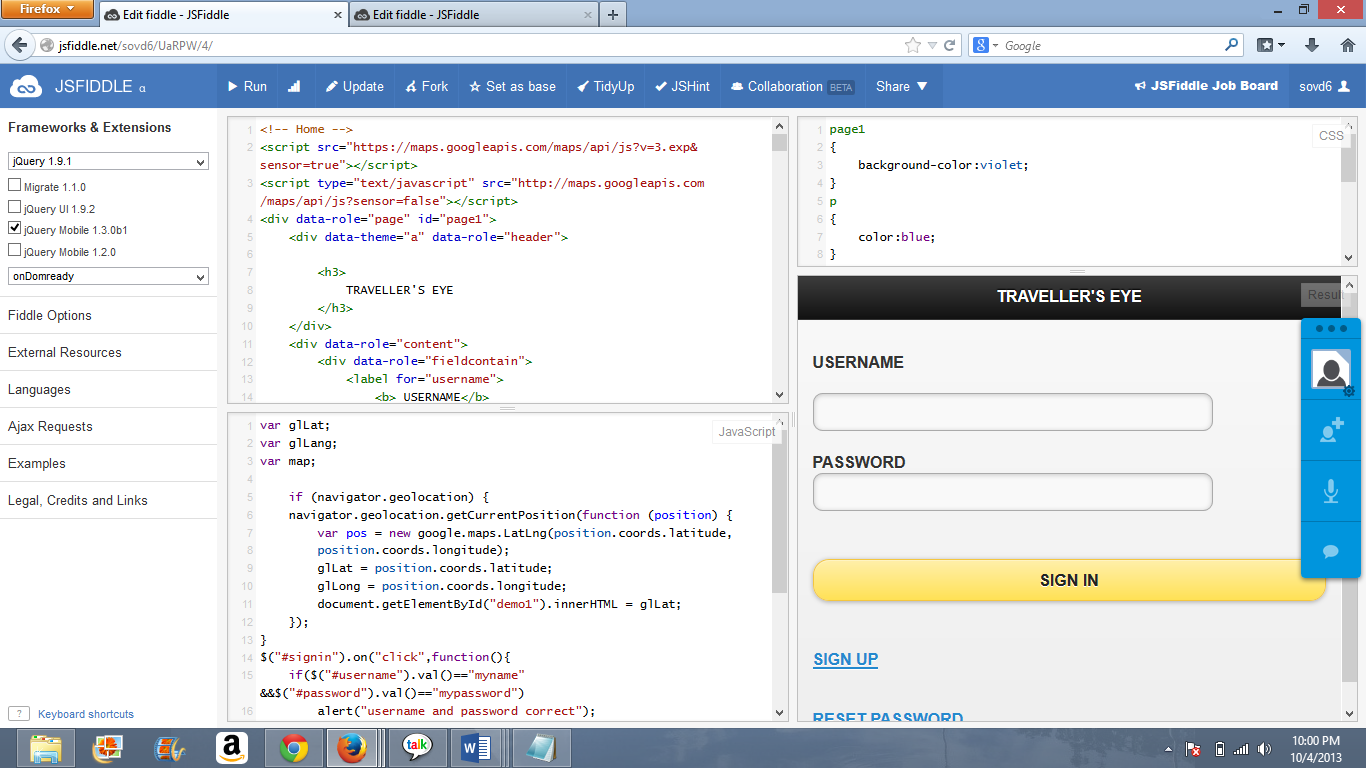
* The user first signs up using SystemGUI.
* The GUI informs the controller.
* The controller stores the information in the database.
* The user now logs in again by taking the help of SystemGUI.
* The SystemGUI informs the controller.
* The controller asks the SytemGUI to display the services.
* The SystemGUI displays the services to the user.
* The user selects one of the services.
* This is informed to the controller with the help of the SystemGUI.
* That service is provided by the service provider by contacting its sub classes.
* The service provider provides the service to the user using sub class current location.

**SEQUENCE DIAGRAM FOR NAVIGATION:**

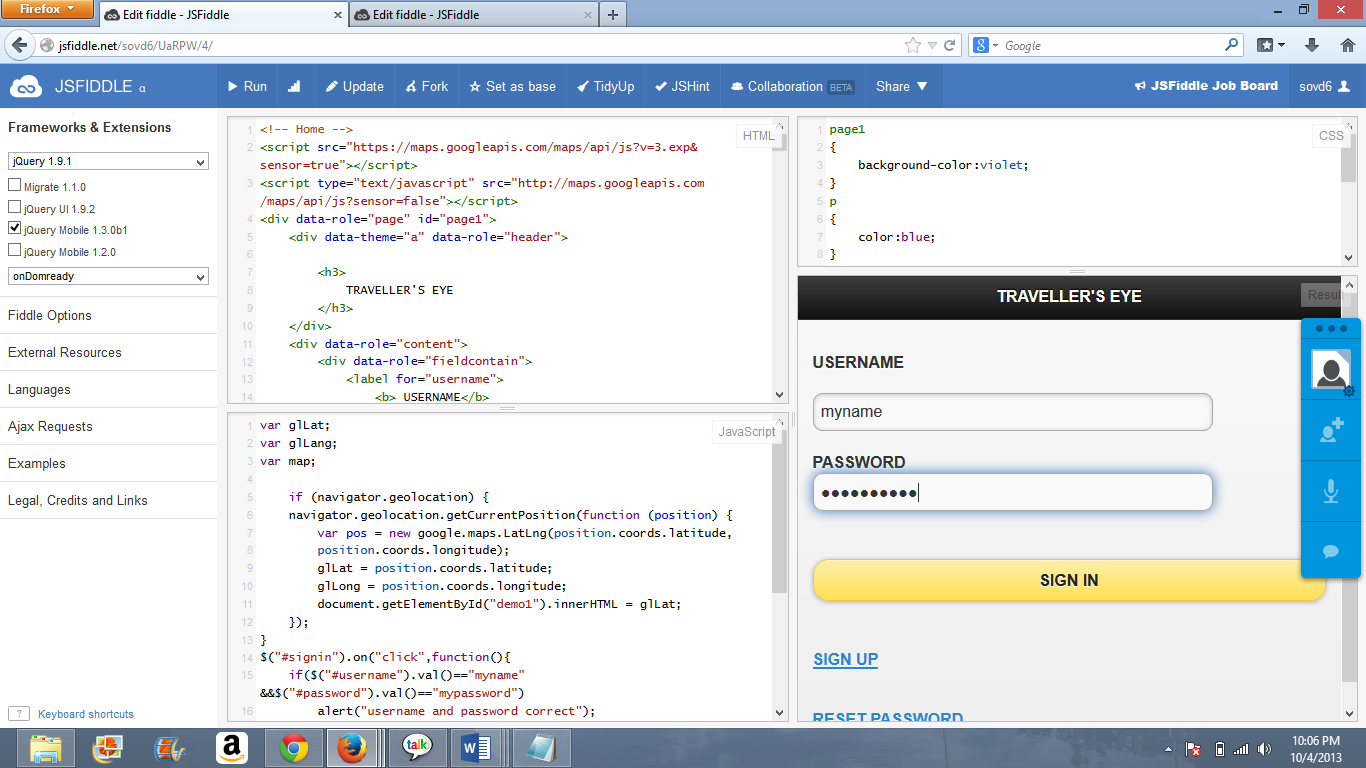
****

**MOBILE INTERFACE: (GUI)**

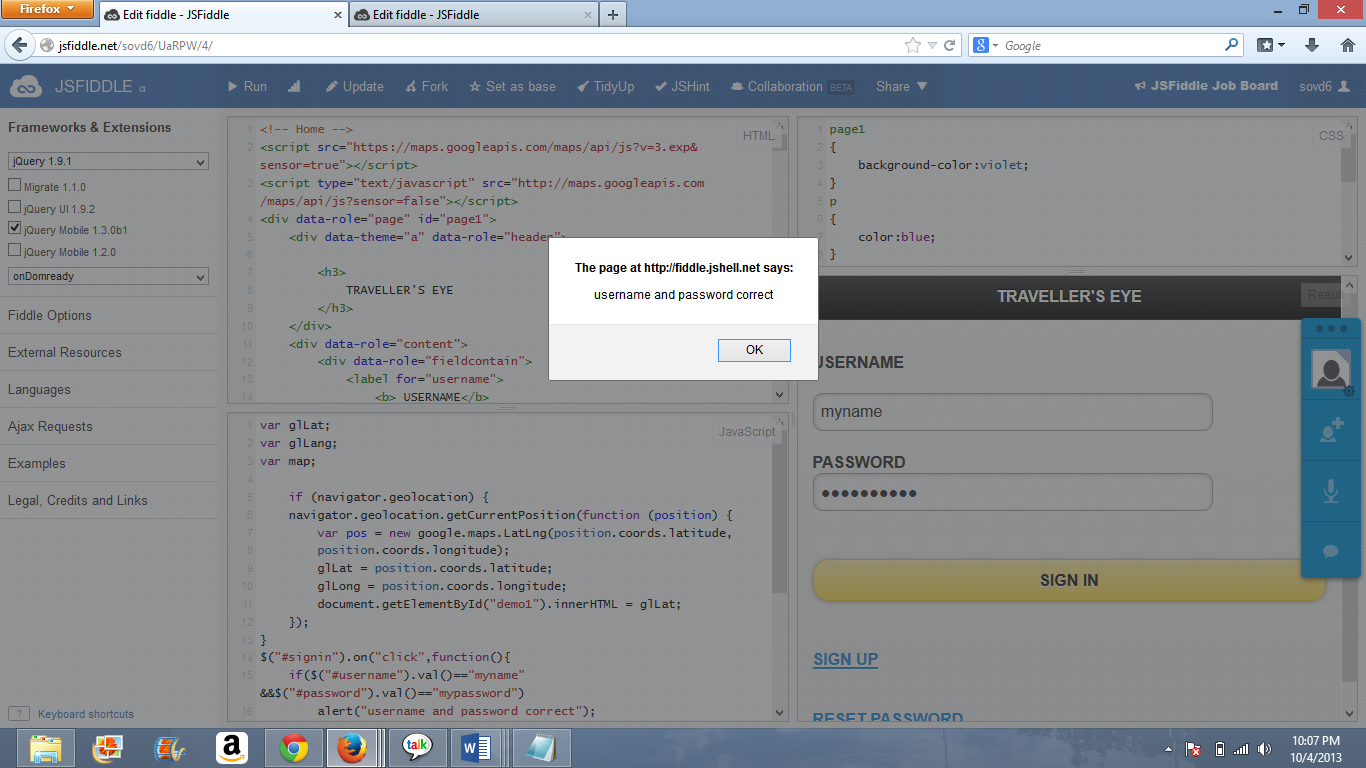
The User Interface is created using HTML5 and CSS3 is used for styling purpose. The elements are given movement by using Jquery. First we have an login page from where we can perform sign in or sign up or reset password actions. Once we give our username along with password and signs in we can access the we can access the services provided by the application.



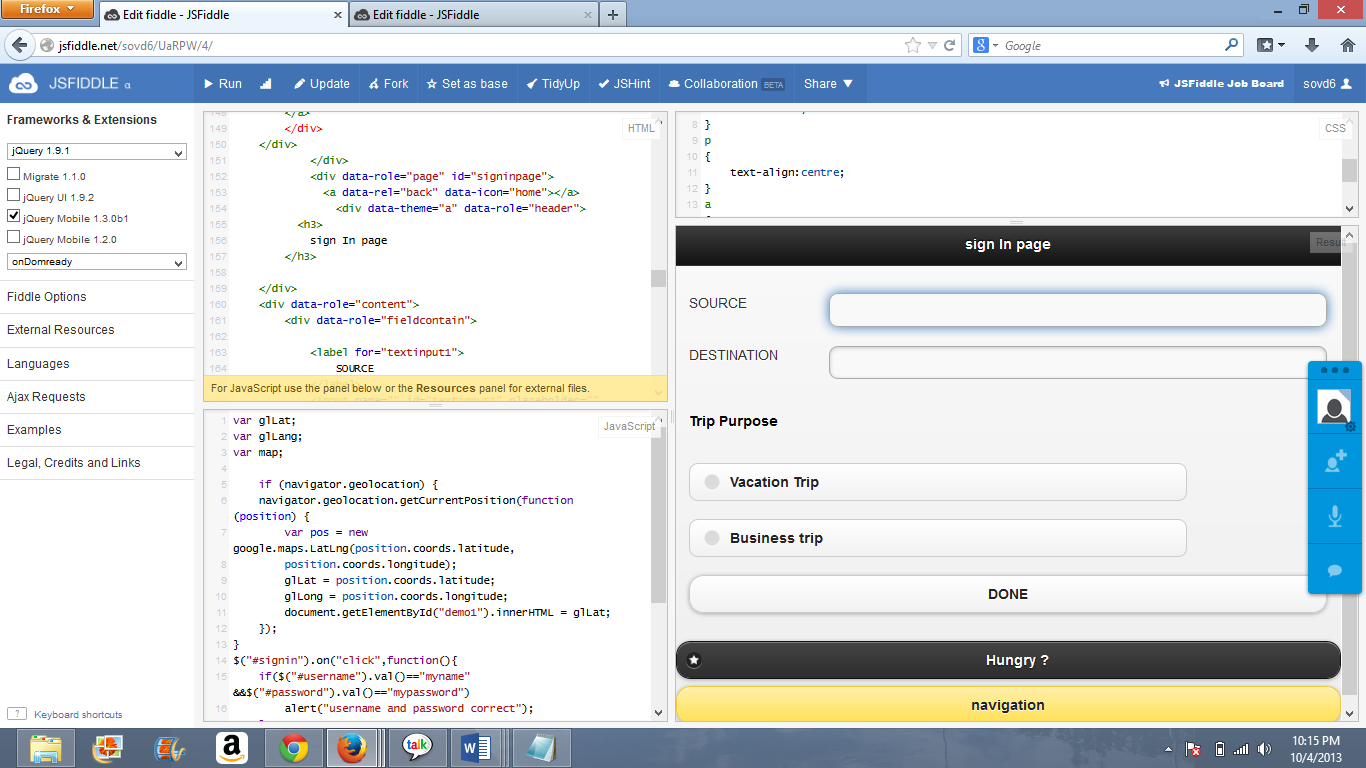
1. **This is the initial start up of our project. Above screen shot depicts the display of the user interface on a mobile device. It shows the login page for an user. Waits for user to click on sign after entering the credentials which are setup prior to the login, and checks his authenticity and redirects him to a new sign in page.**



1. **He now enters his login details. System now checks if the given details equate to the setup details.**

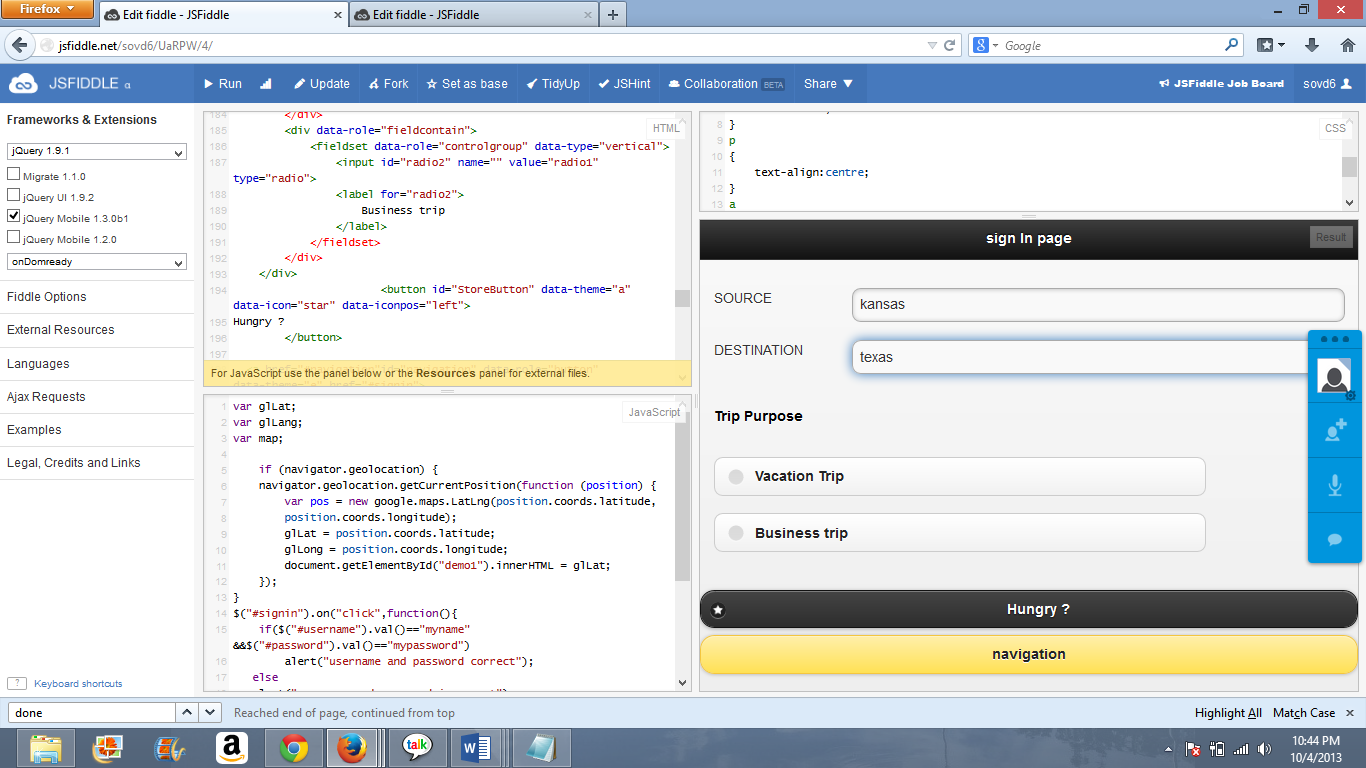


1. **As user has entered the correct login details it now alerts saying that the credentials are correct besides directing him to the destined page.**

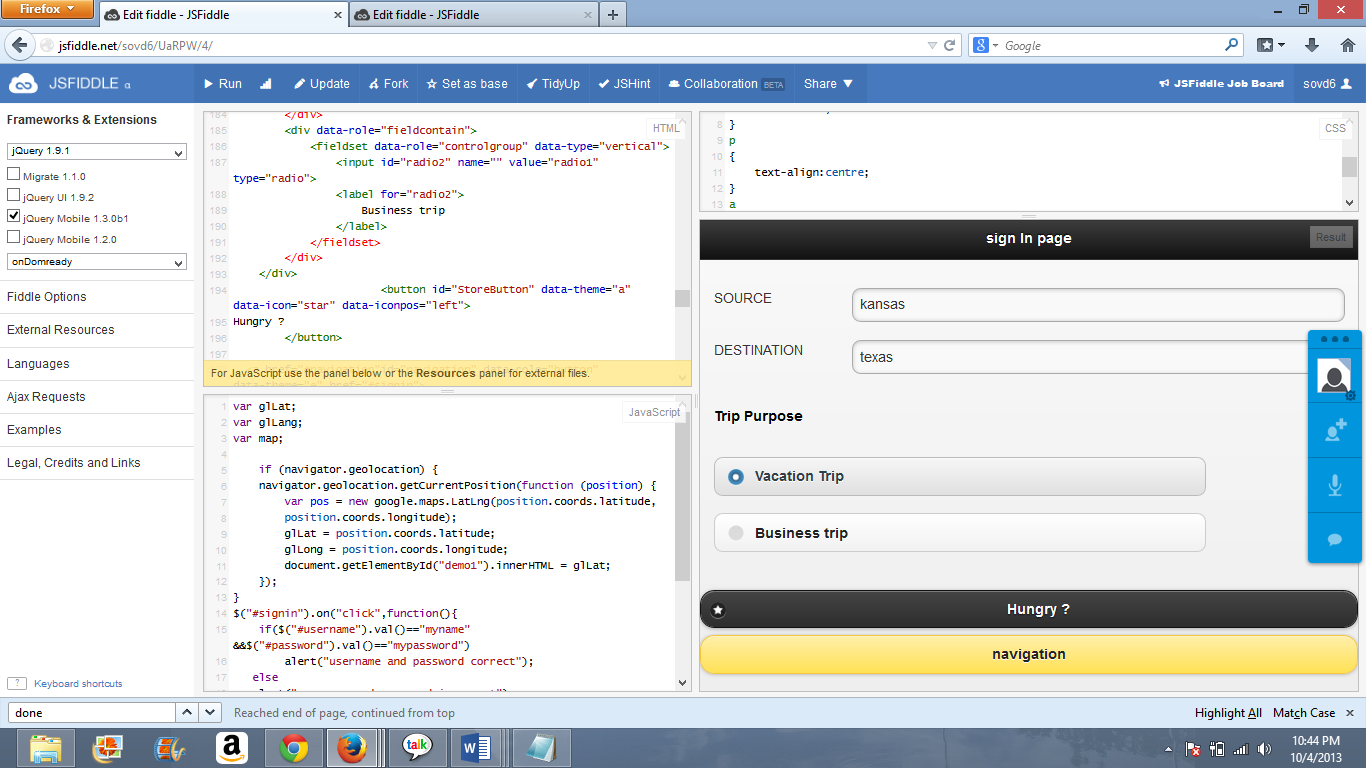


1. **In the sign page the detailed view of necessary details are labelled. It asks user to enter his intended source and destination, next is for his kind of trip purpose like is he on a vocational or business trip and finally the “done” button. Later on the page there is even more discrete functional options available for the user.**

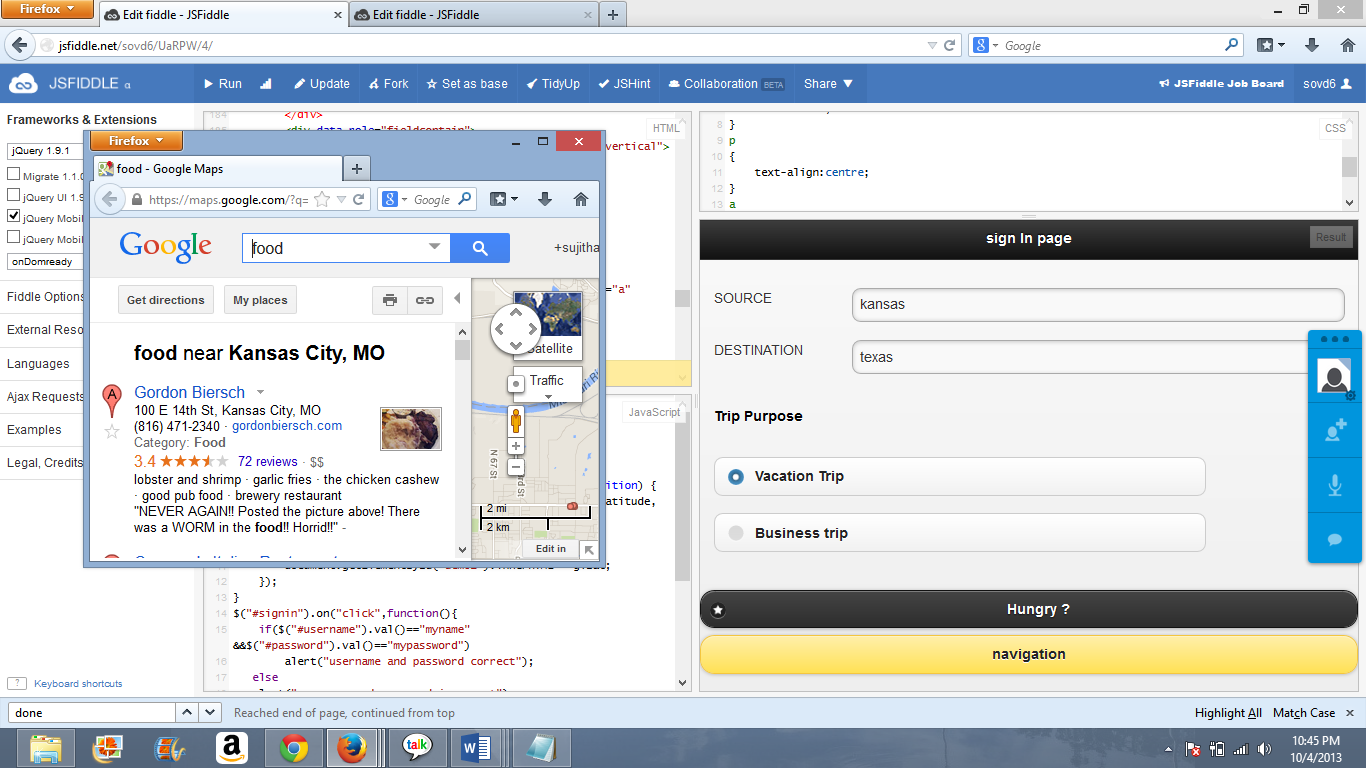
* **“Hungry” which collects the locations of the food sources like restaurants, food stalls, pizza huts etc. in his surroundings.**
* **“Navigation” which loads the navigation between his source and destination.**



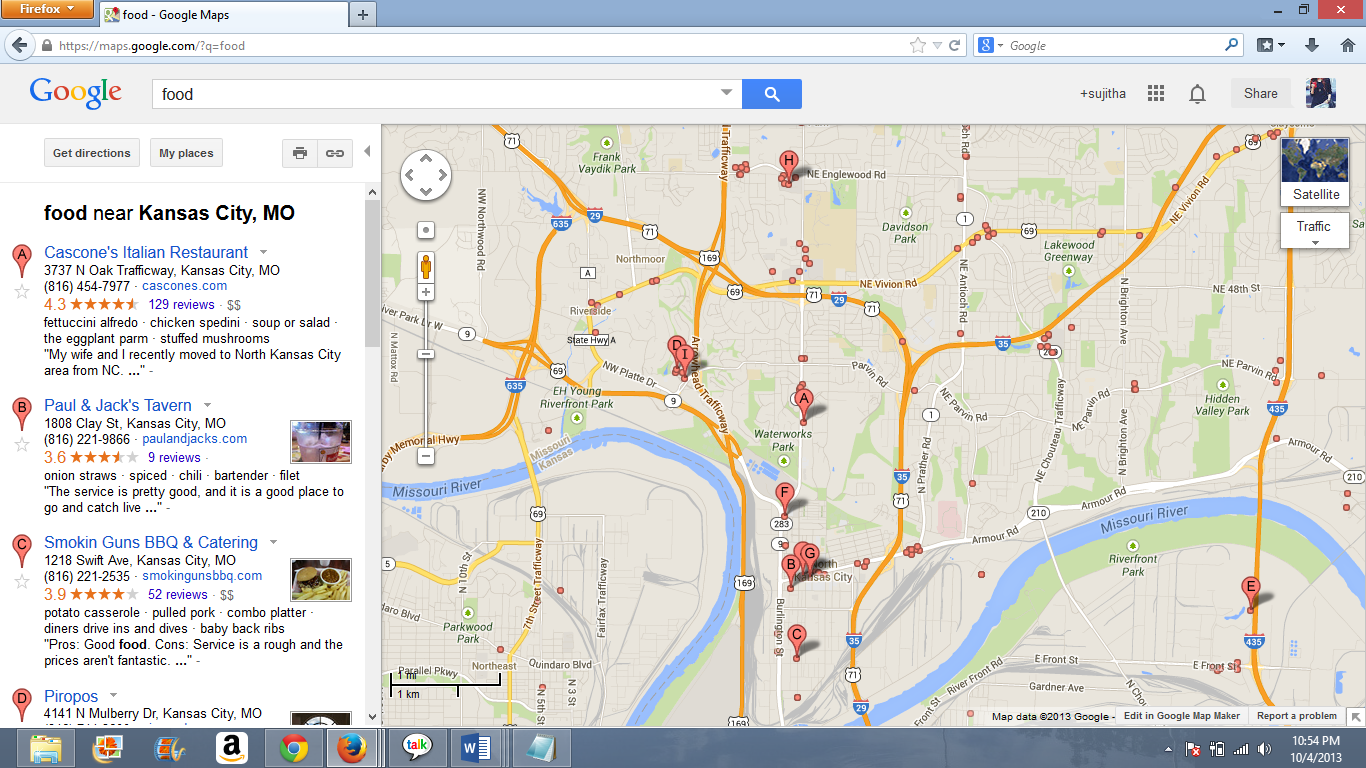
1. **Say the user in our project is now intending to travel from Kansas City to Texas, so he enters the locations in the corresponding input labels.**



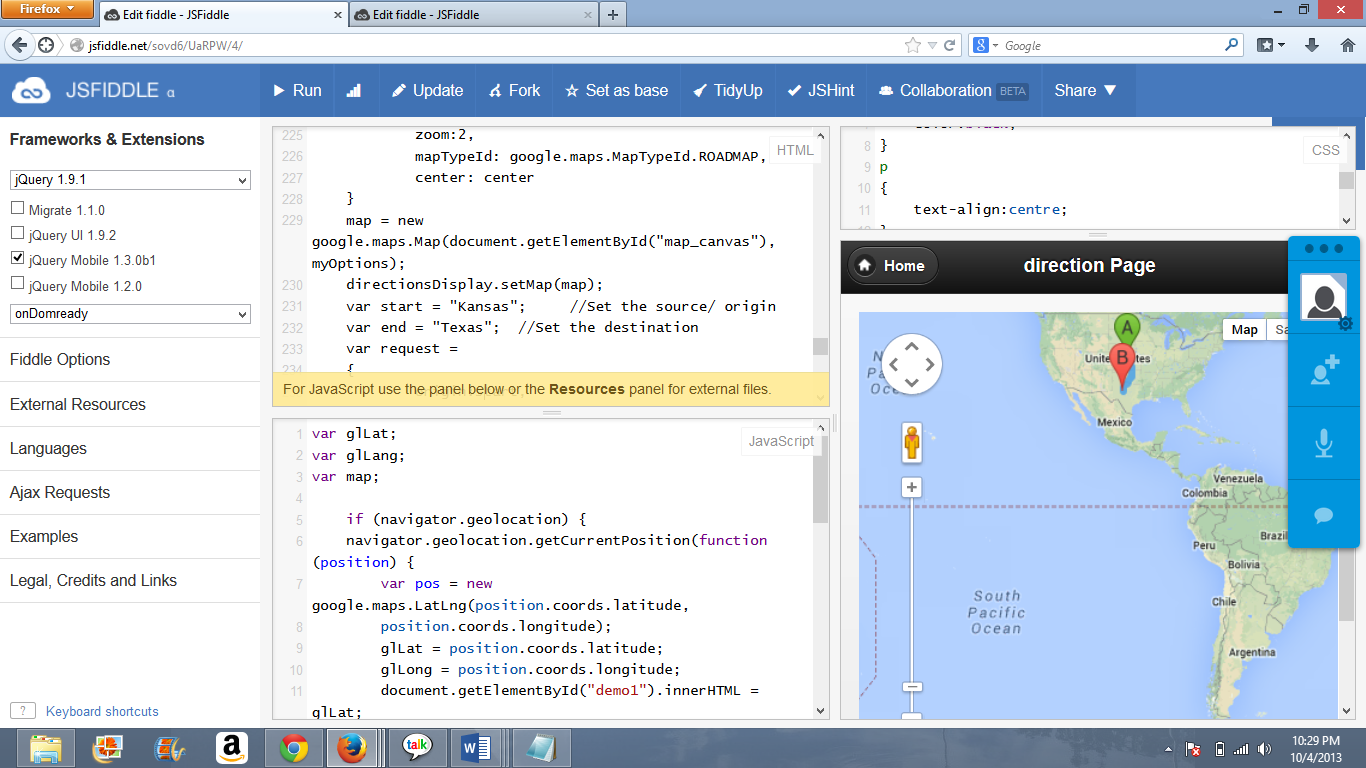
1. **User enters his purpose of trip say Vacation trip.**



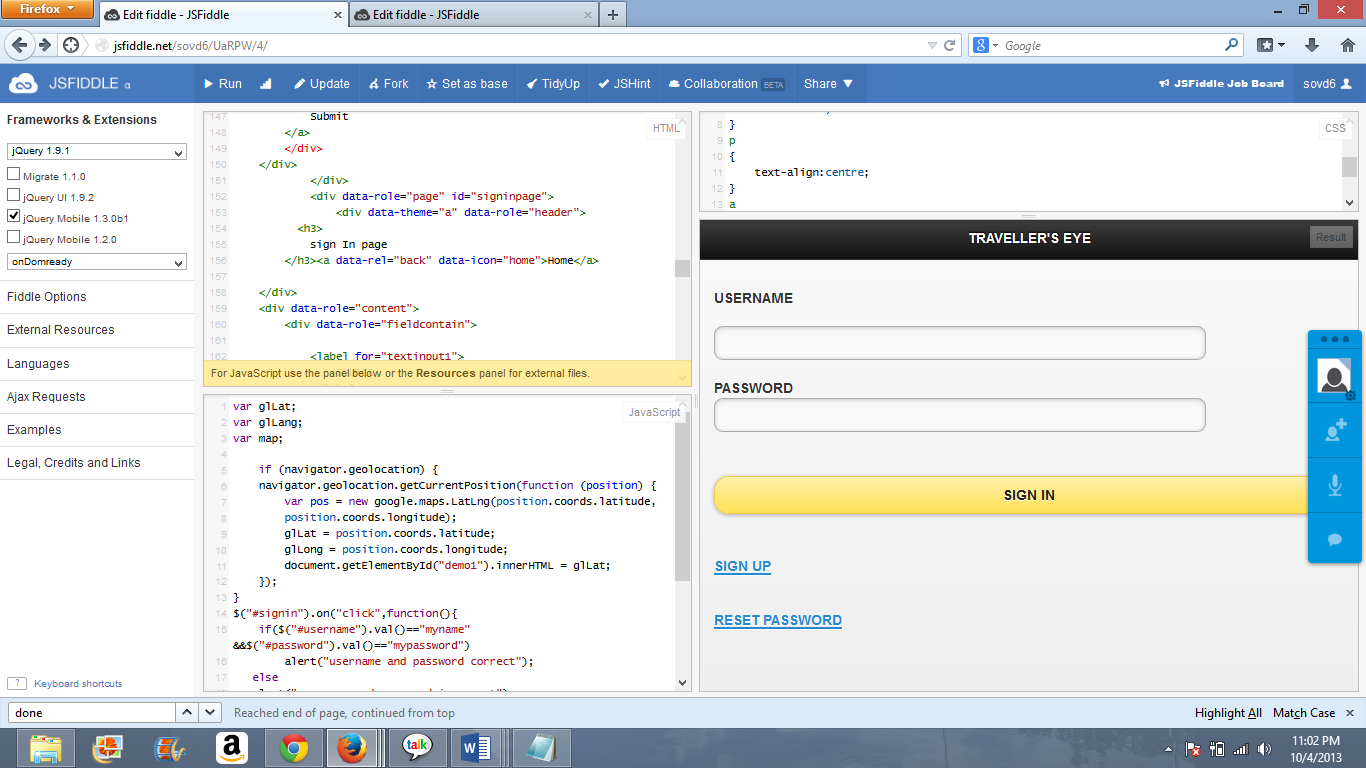
1. **Next function is an alluring feature of the project. Suppose user feels hungry and say he is new to the place don’t where to go and where the food stalls are?**



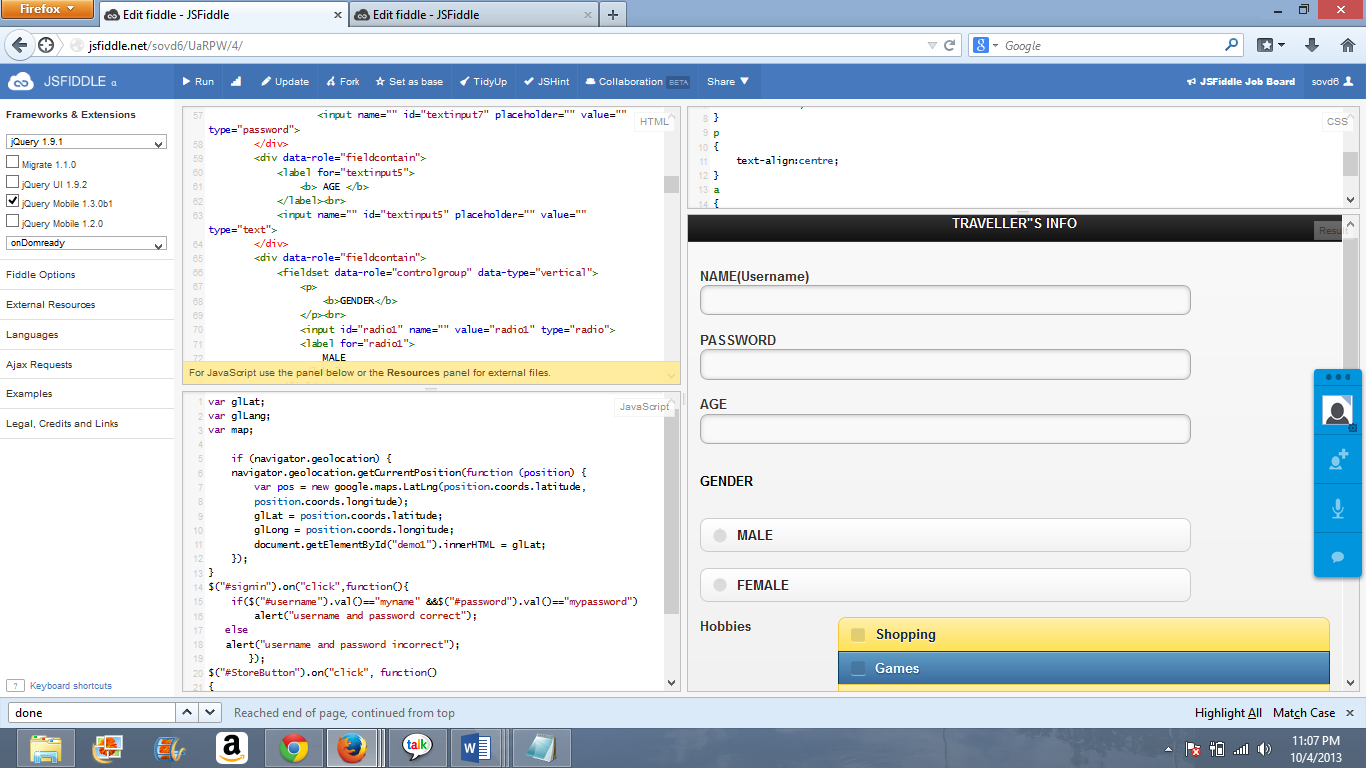
1. **Above screen shows the google map plotting the locations of the food sources in the nearby regions.**

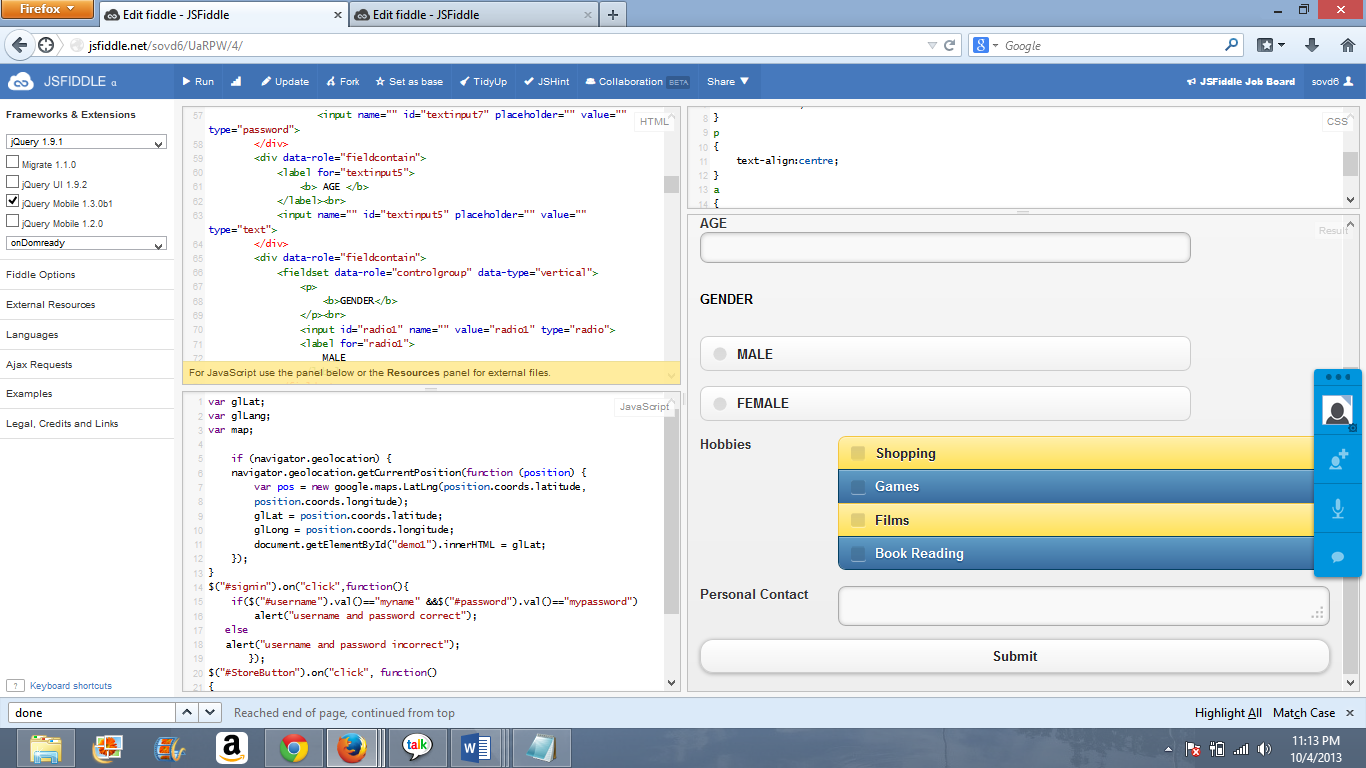


1. **Next feature is the display of the navigating route between Kansas and Texas. A is the source (Kansas) and b is the destination (Texas).**



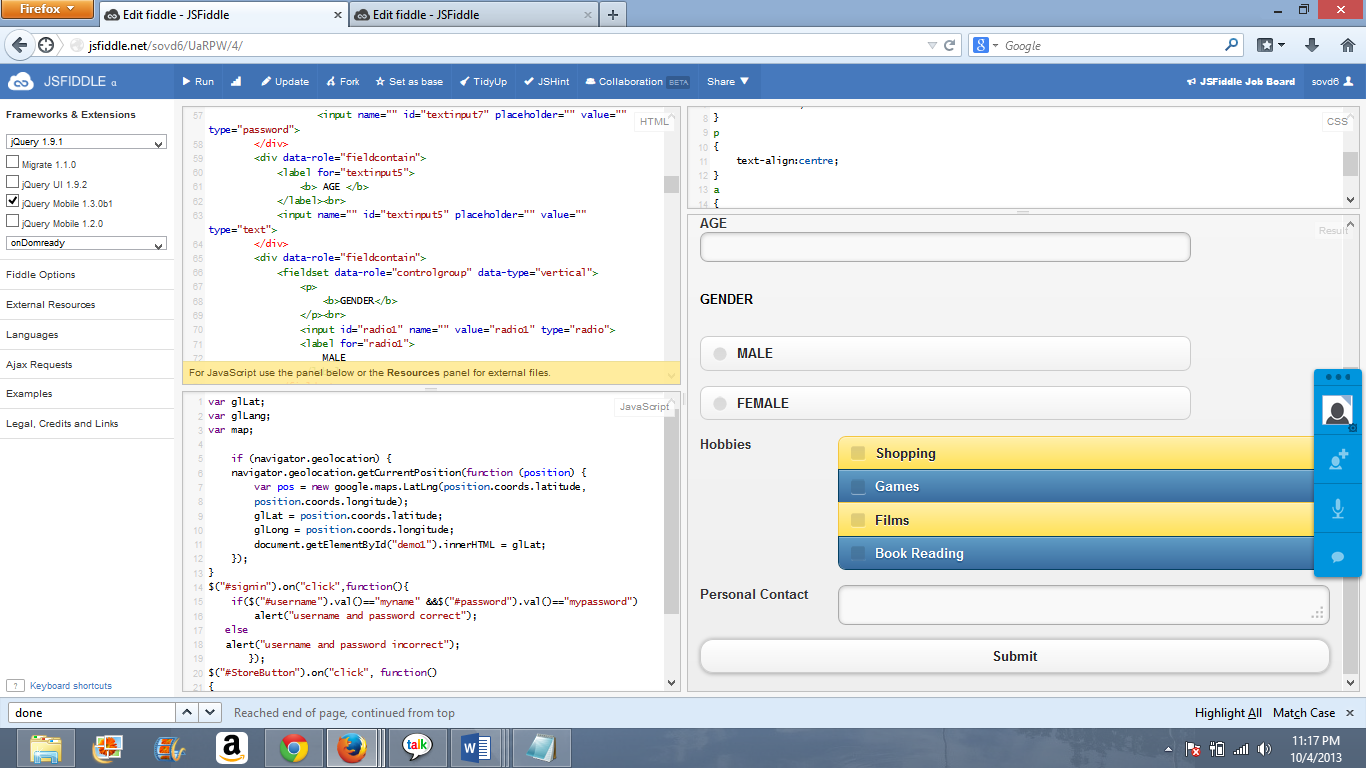
1. **On the other hand the initial display page of the interface shows a way for an user to signup into the traveler’s eye.**

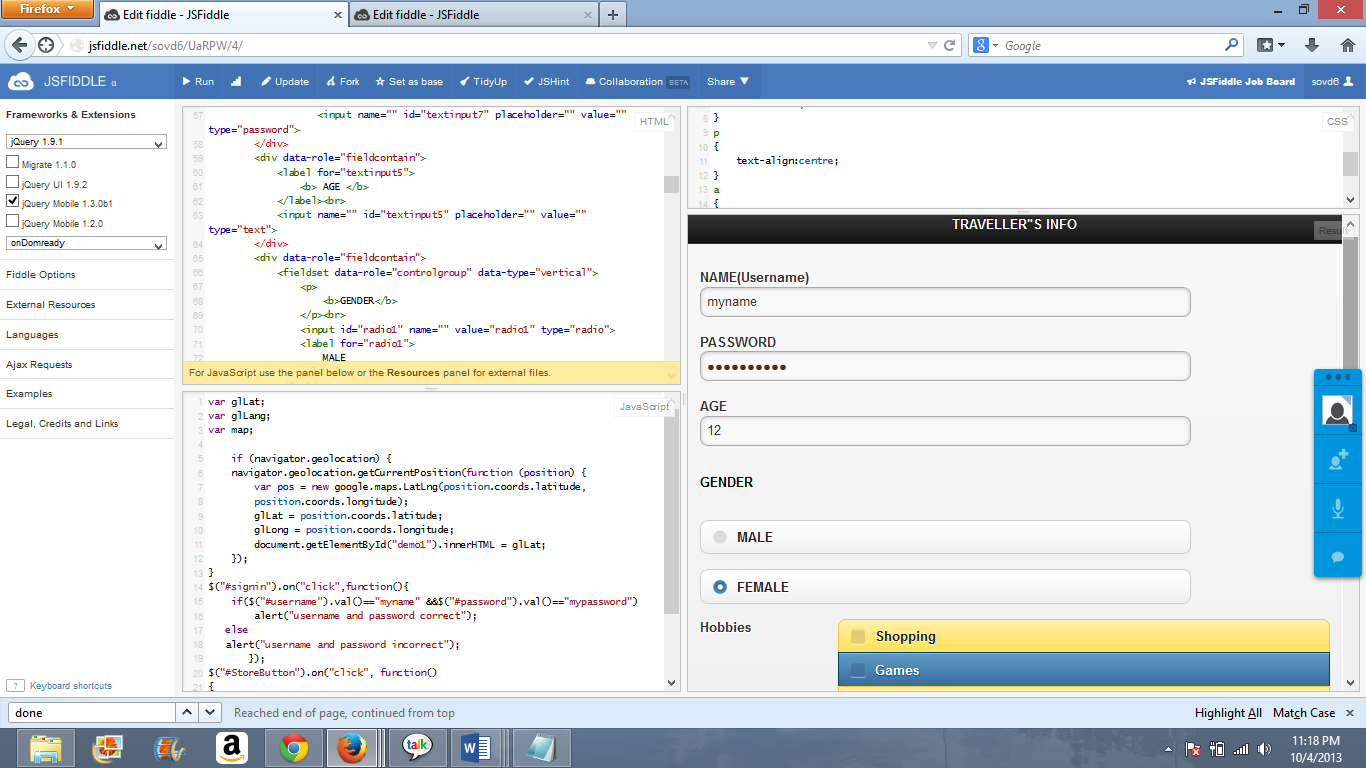


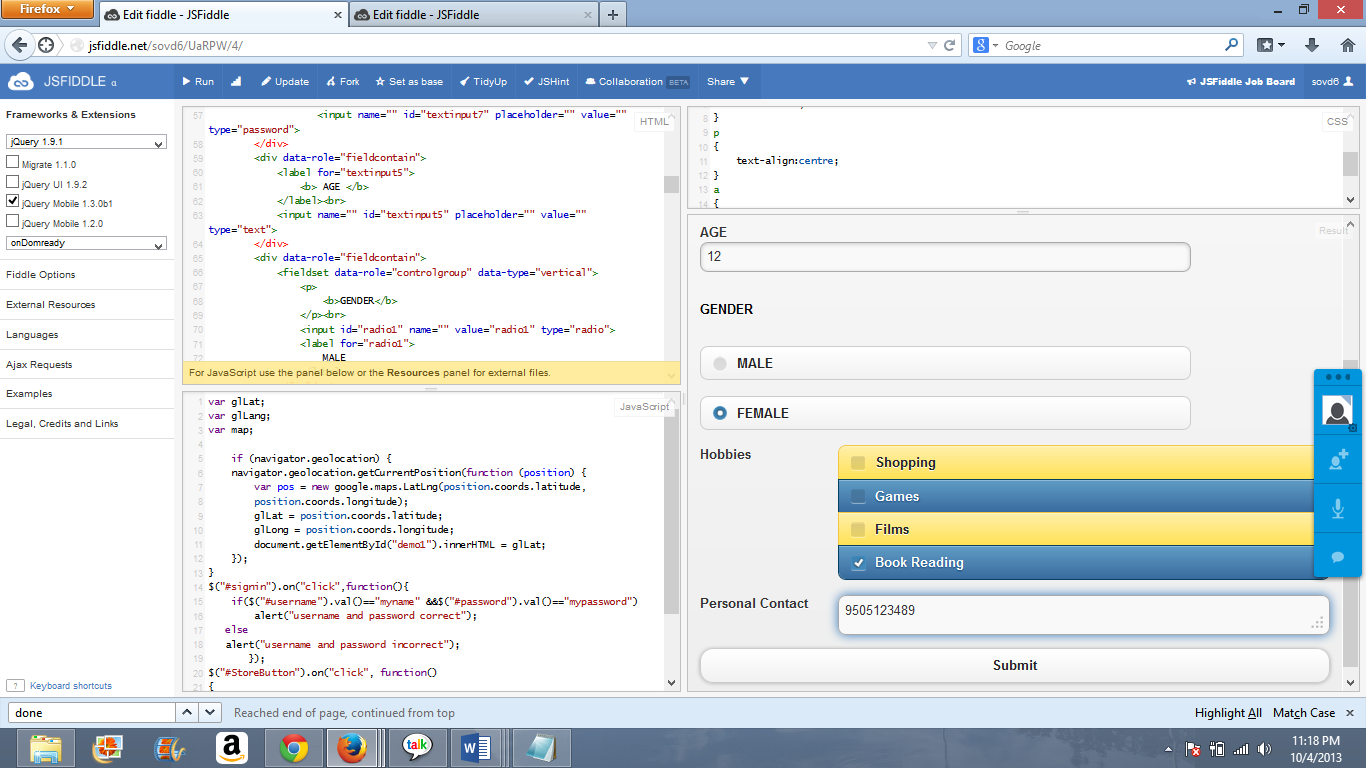


1. **This is how the sign up page looks. It has things that are intended to be considered by a user for setup up his login in the traveler’s eye.**

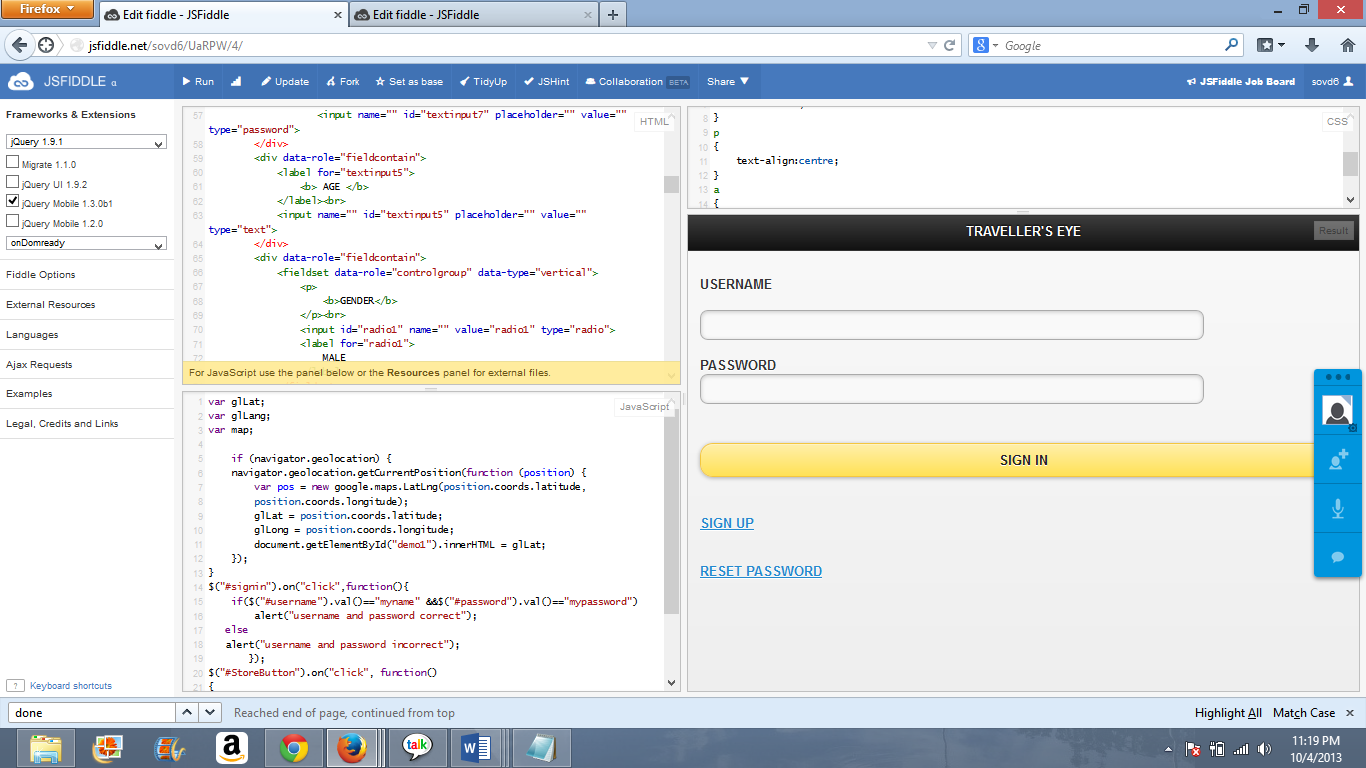
* **“Name” is the label that is created to take in the user’s name.**
* **“Password” is another input that is taken for the setup of a password as intended by the user. Password will not be displayed on the screen due to security purposes the way user enters his password it will be shown as dots.**
* **“Age” is taken so as to display the further things as the way they are categorized.**
* **“Gender” interface allows him to choose between male and female with the usage of the radio buttons.**
* **“Hobbies” includes the typical hobbies of people like shopping, games, films, reading books. “Personal Contact” his personal contact which can then be used for correspondence.**
* **At last the “submit” which is used to post the information to the system, he can login to the traveler’s with his newly created details.**



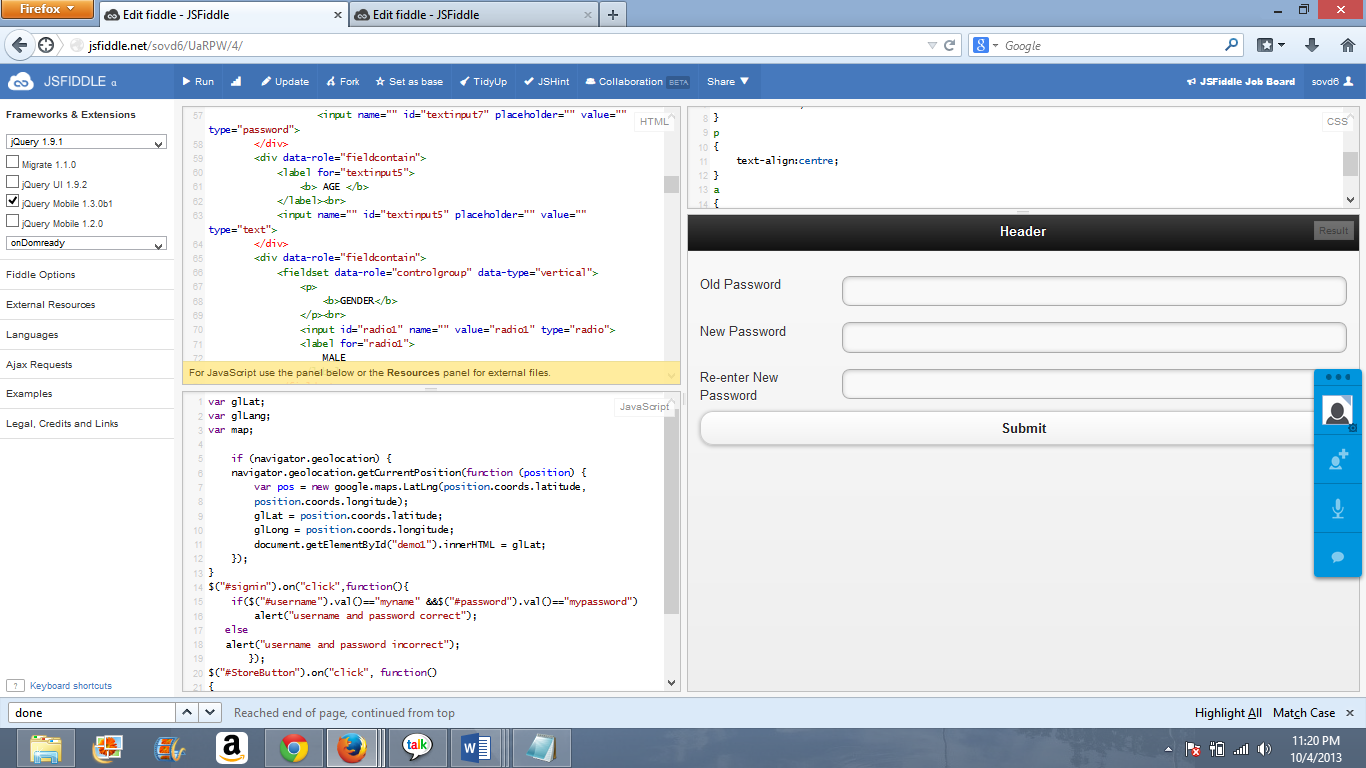




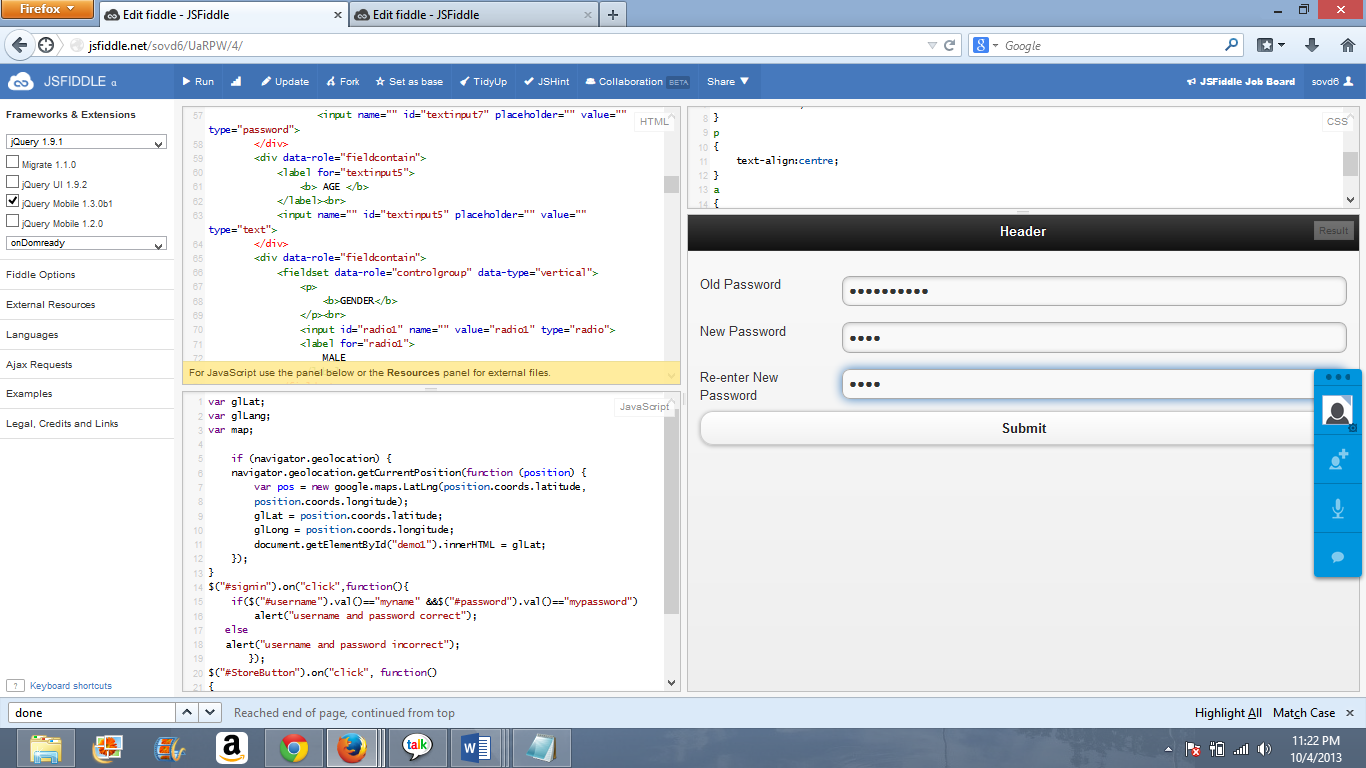
1. **After entering the details and submitting he now returns to the intial page which will allow him to login directly.**



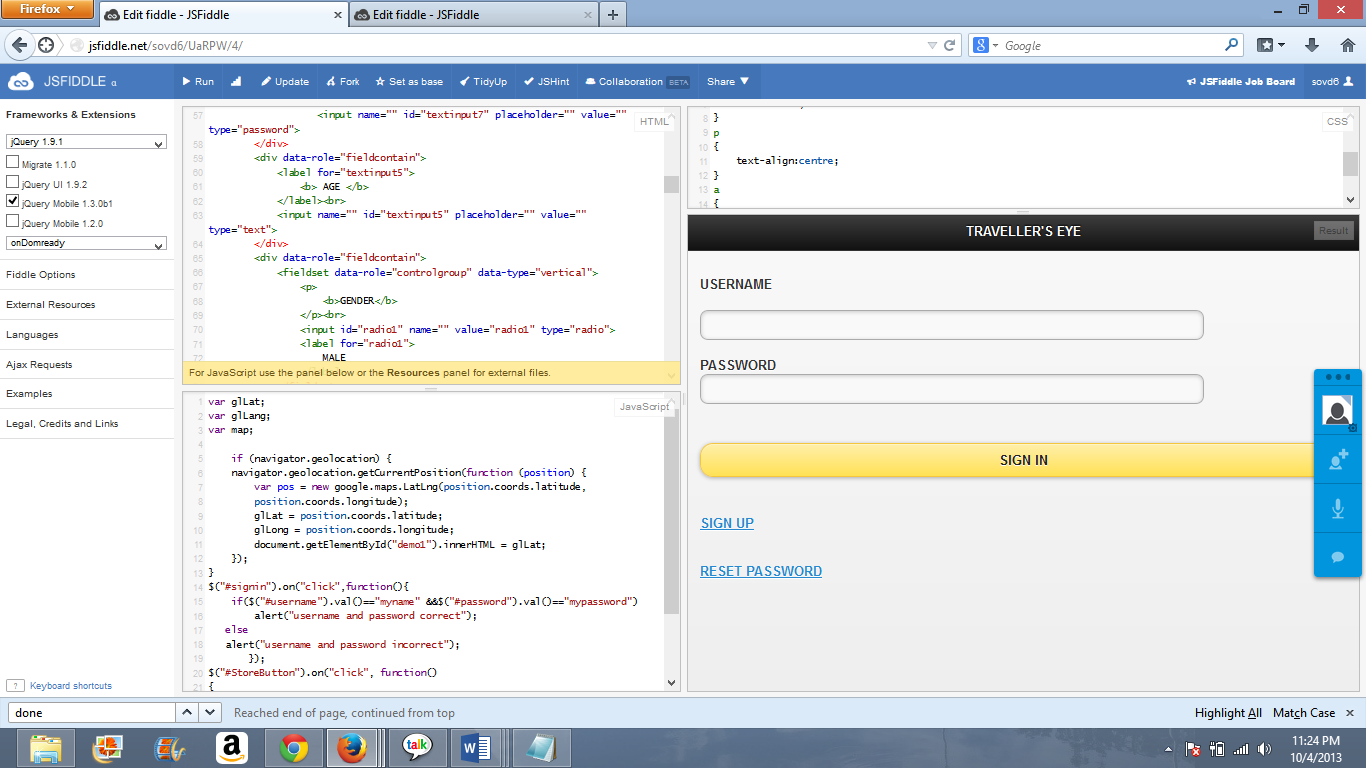
1. Besides creating and signing in he also has an option for resetting his password.



1. “Old Password” entered by the user. “New Password” new password as wished by the user. Re-entering the password, finally submit for updating the details.



1. Enters the necessary details on clicking submit he returns to the initial page, where he can login using his new password.



**Testing the GUI (Graphical User Interface):**

1. When the application is executed on the first click it is displaying the proper screen fit output.
2. Each and every field in the page is aligned properly i.e; text boxes, radio buttons.
3. By clicking on each input text field the pointer of the mouse is changing as the Cursor.
4. When the “Tab” button is clicked, the cursor is shifting its control to the next input text box.
5. The application is allowing the user to select a single radio button at a single instance.
6. It is allowing multiple selection of check boxes.
7. When the user enters the password the password should not be displayed and the password is not displayed as characters.

**Scrum do link:**

[**https://www.scrumdo.com/projects/project/project-150/iteration/78367**](https://www.scrumdo.com/projects/project/project-150/iteration/78367)