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 Third Increment:

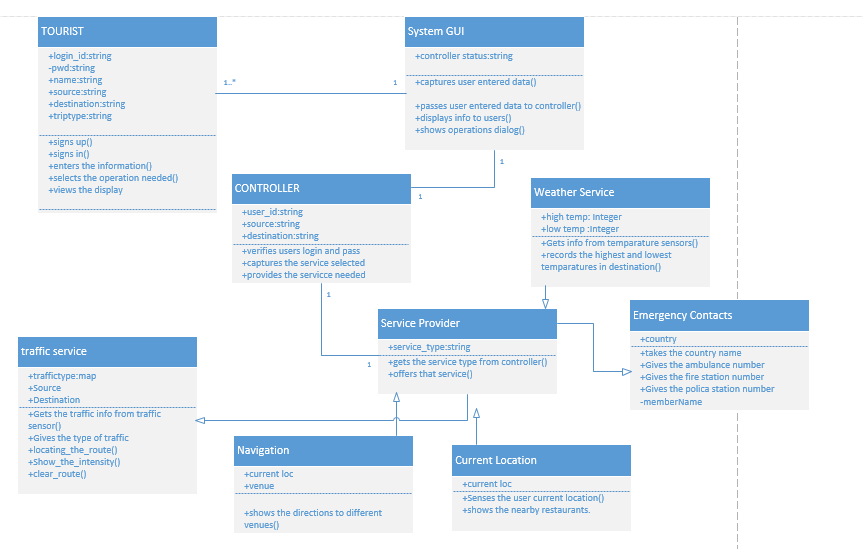
Emergency Contactes

Updated Traffic Service

Uploading photos into the application

Retrieving data from database

Class Diagram:



The class diagram describes the static structure of the application. The class diagram consists of classes, attributes and operations.

**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) weather service: This gives the weather conditions at the given destination.This gives the highest and lowest temperatures at the given position.

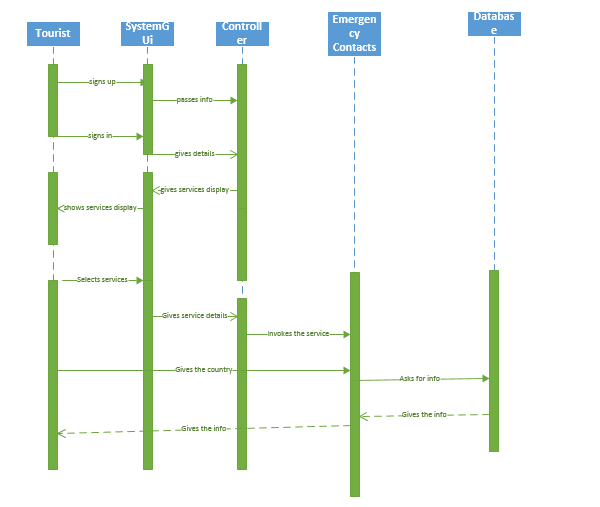
2)Traffic service: This gives the traffic updates at the place required by the user.

3)Current Location: The current location of the user is identified.

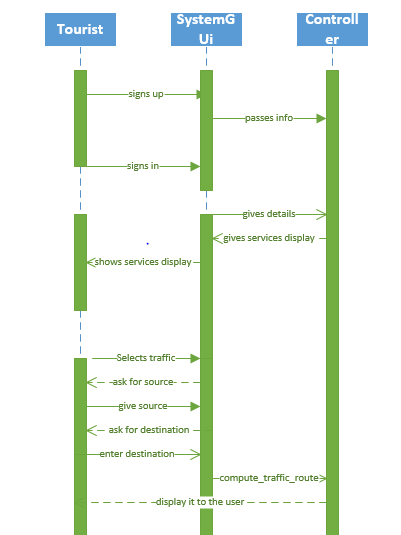
4)Navigation: This shows the navigation from to user’s current position to the place he wants to move.

5)Emergency Contacts: The user can get the emergency contacts like ambulance fire and police man.

Sequence Diagram for Emergency Contacts:

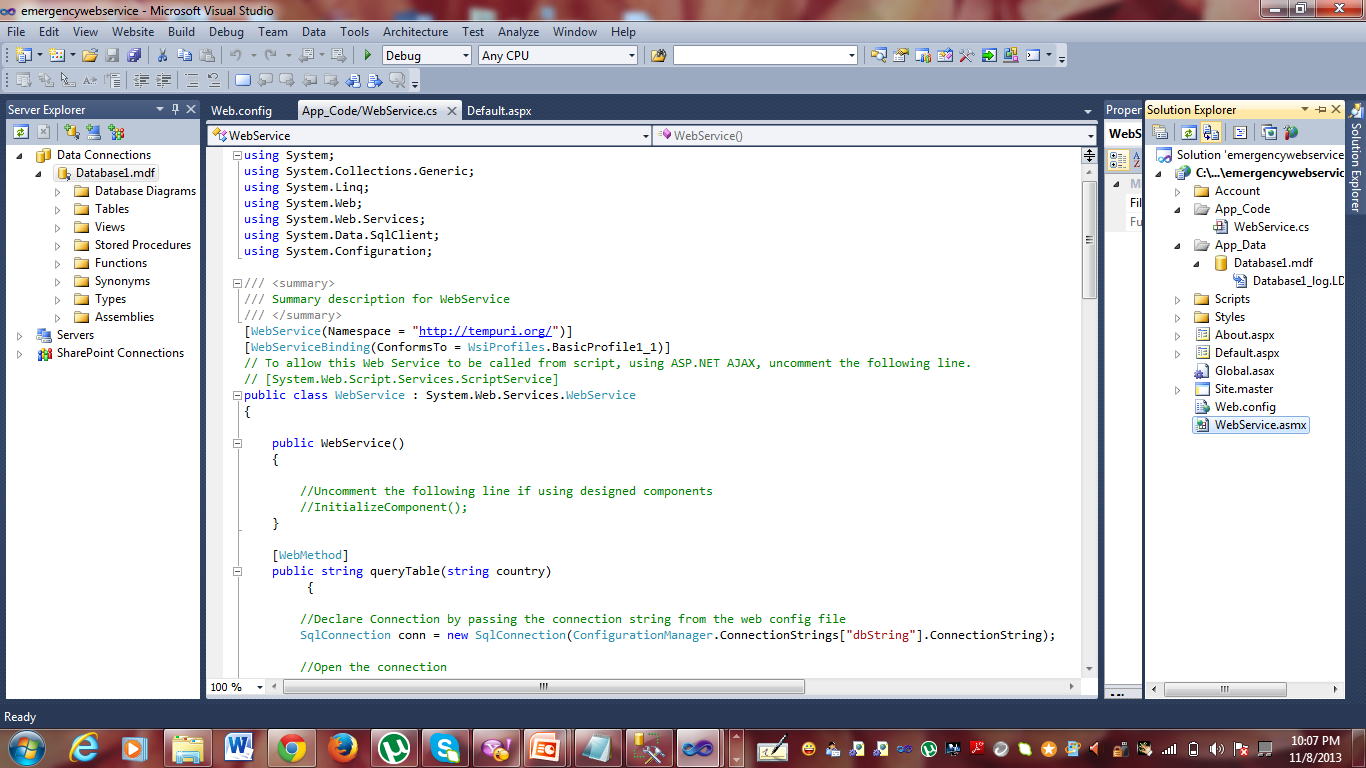


Sequence digram for traffic service:

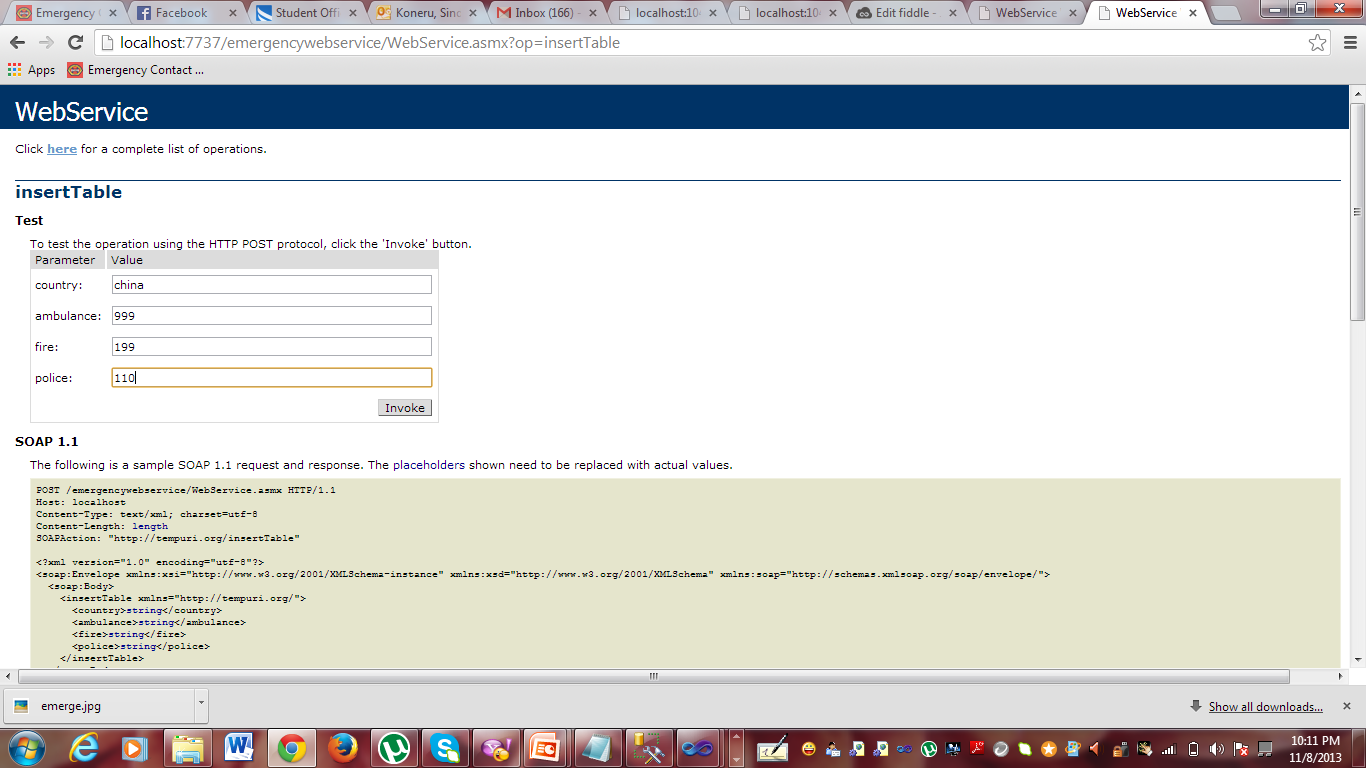


Emergency Contacts:

The emergency contact s like the local police station number, hospital number and any such emergency numbers as soon as the user selects the destination. So even in emergency all the user has to do is press one button. Additionally the app also allows some emergency contacts to be defined by the user.



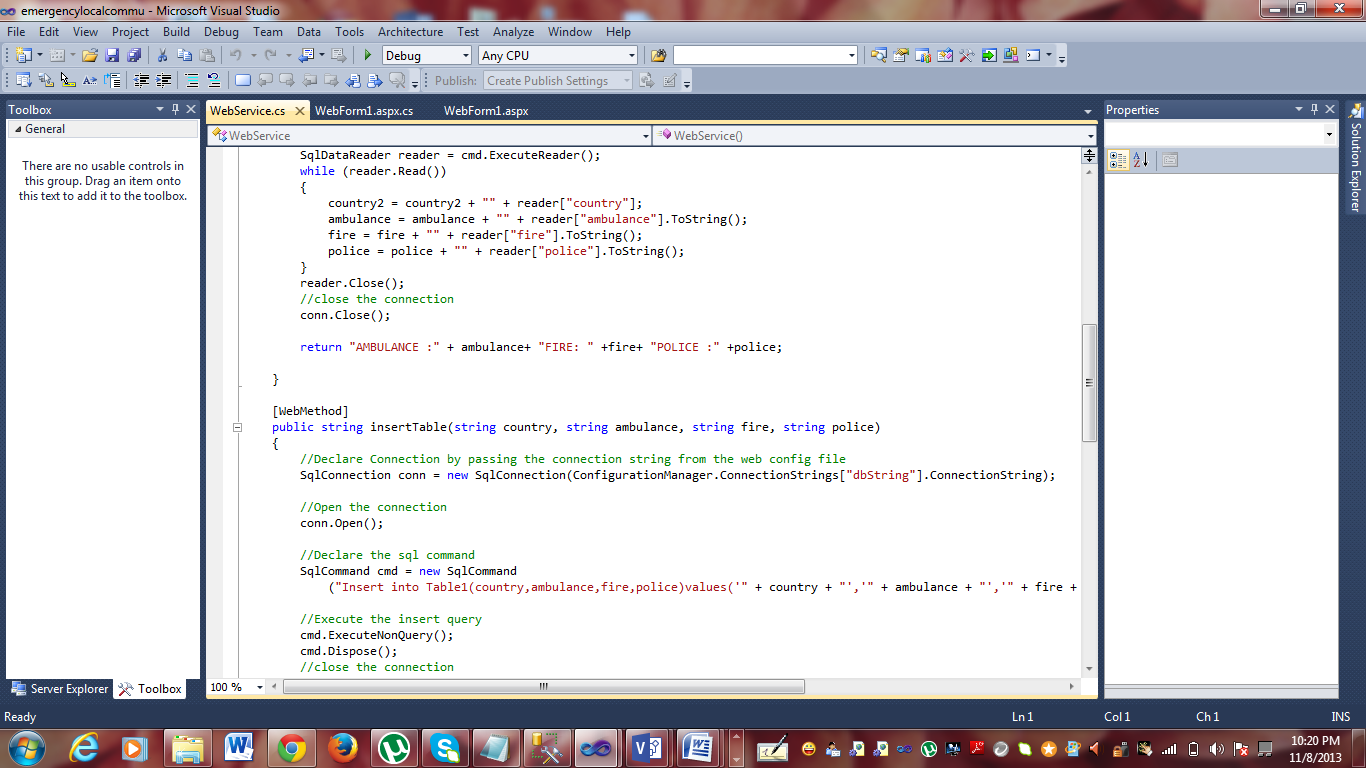
Code in the local machine to create web service for emergency contacts.



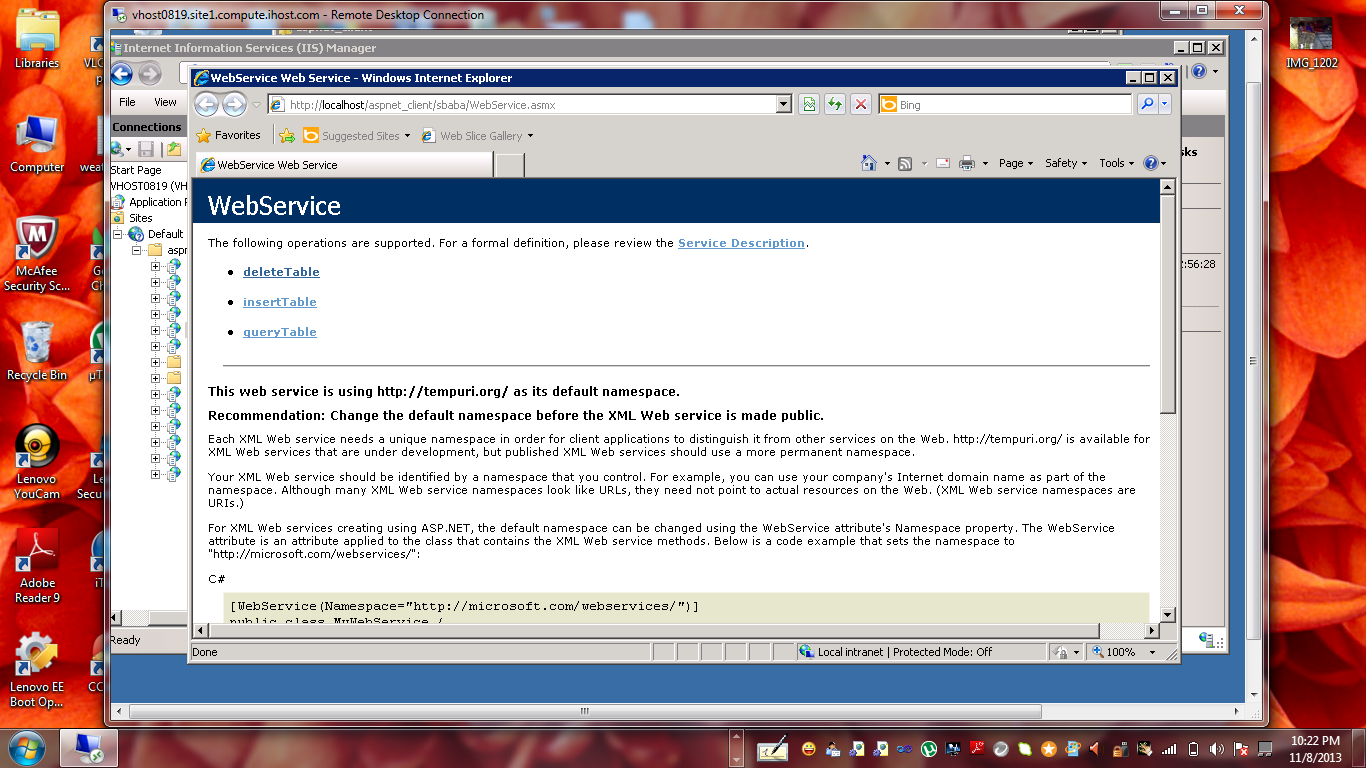
By executing that and after selecting insert table we get the above screen where we can enter our values to the database.

After the invoke we get success insert command.

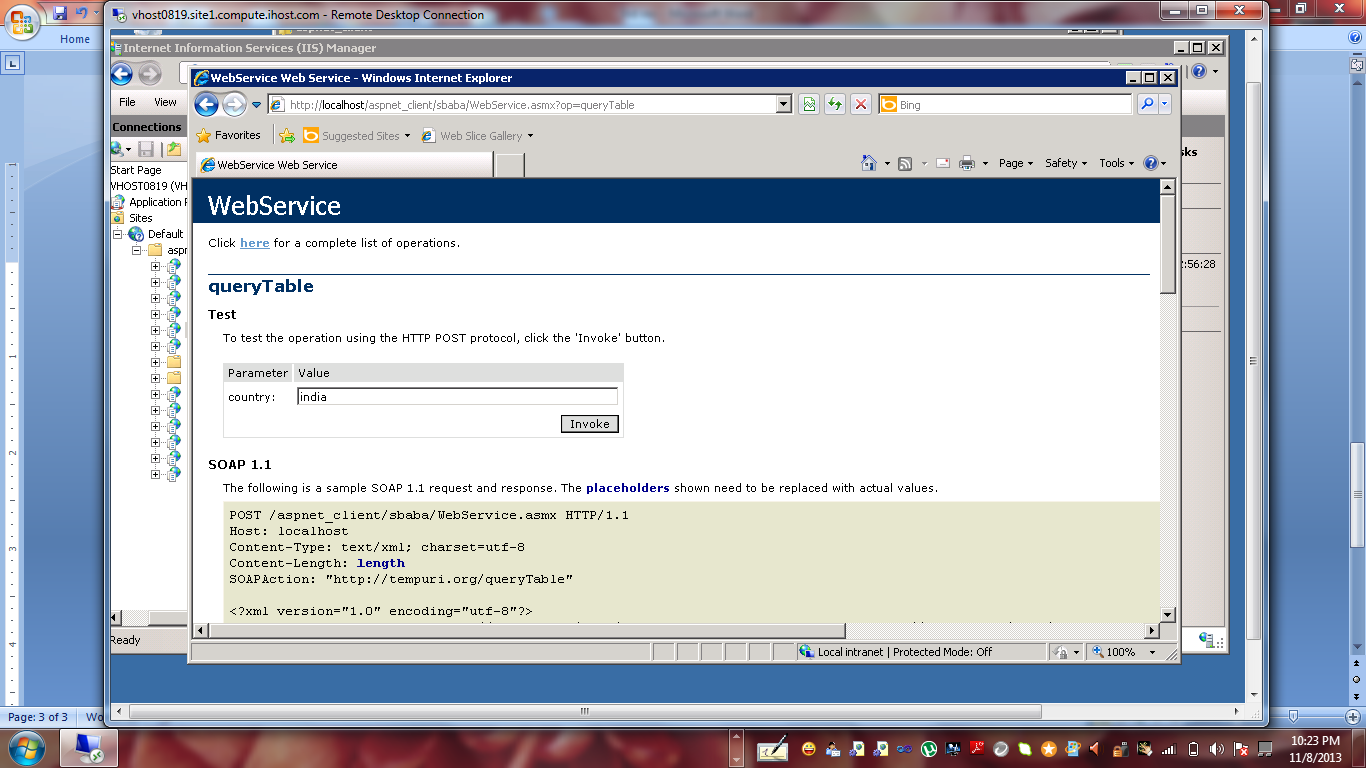
Then we can go to querytable and give the country name we get the details from the database.

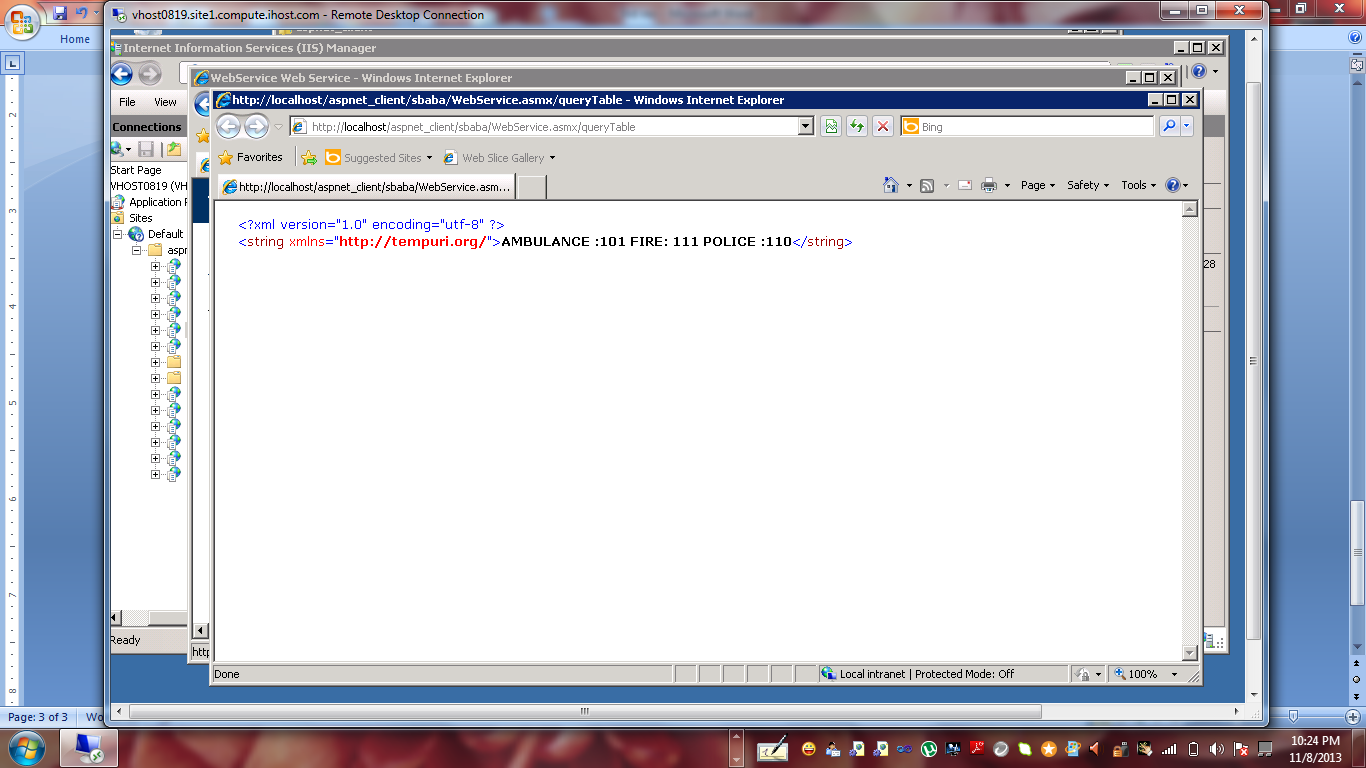


Code that is placed in the remote machine.



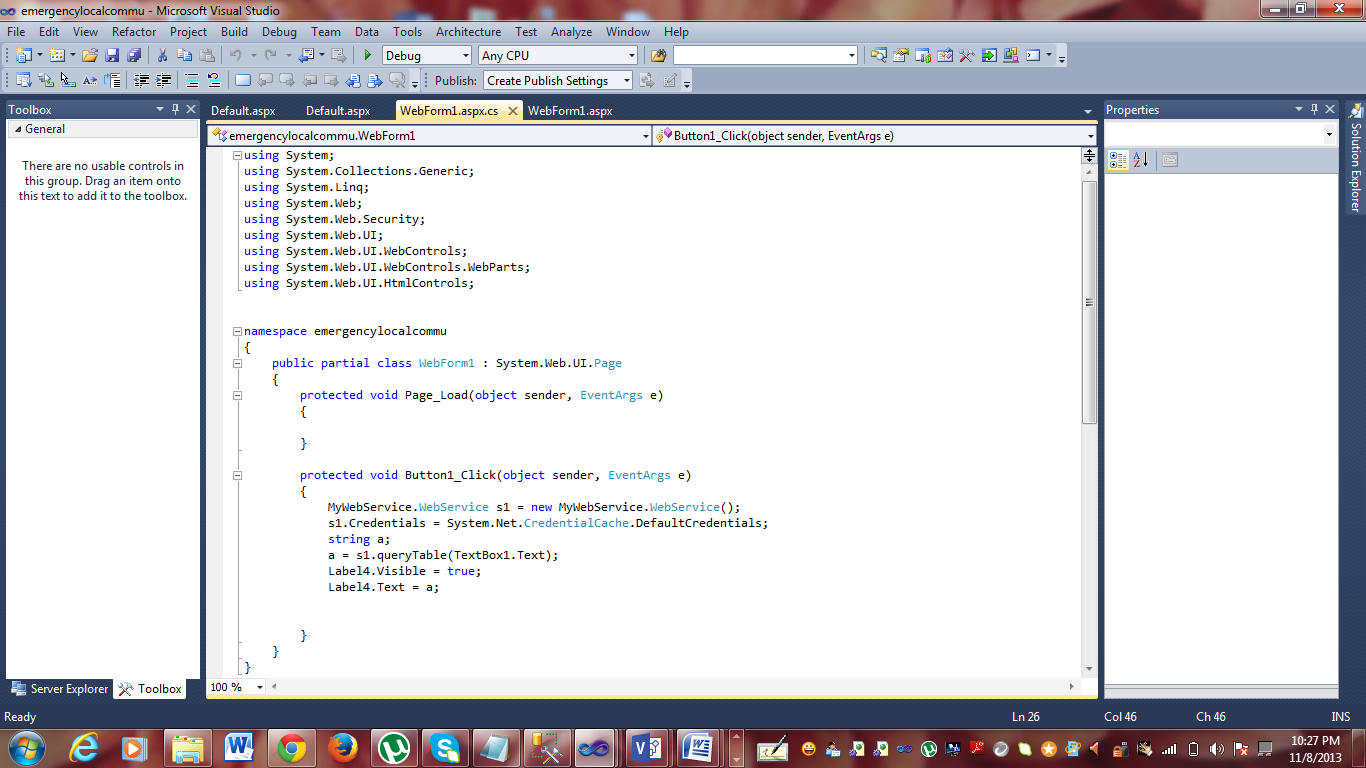
Implementing the web service in remote host.Deploying the service in remote host.





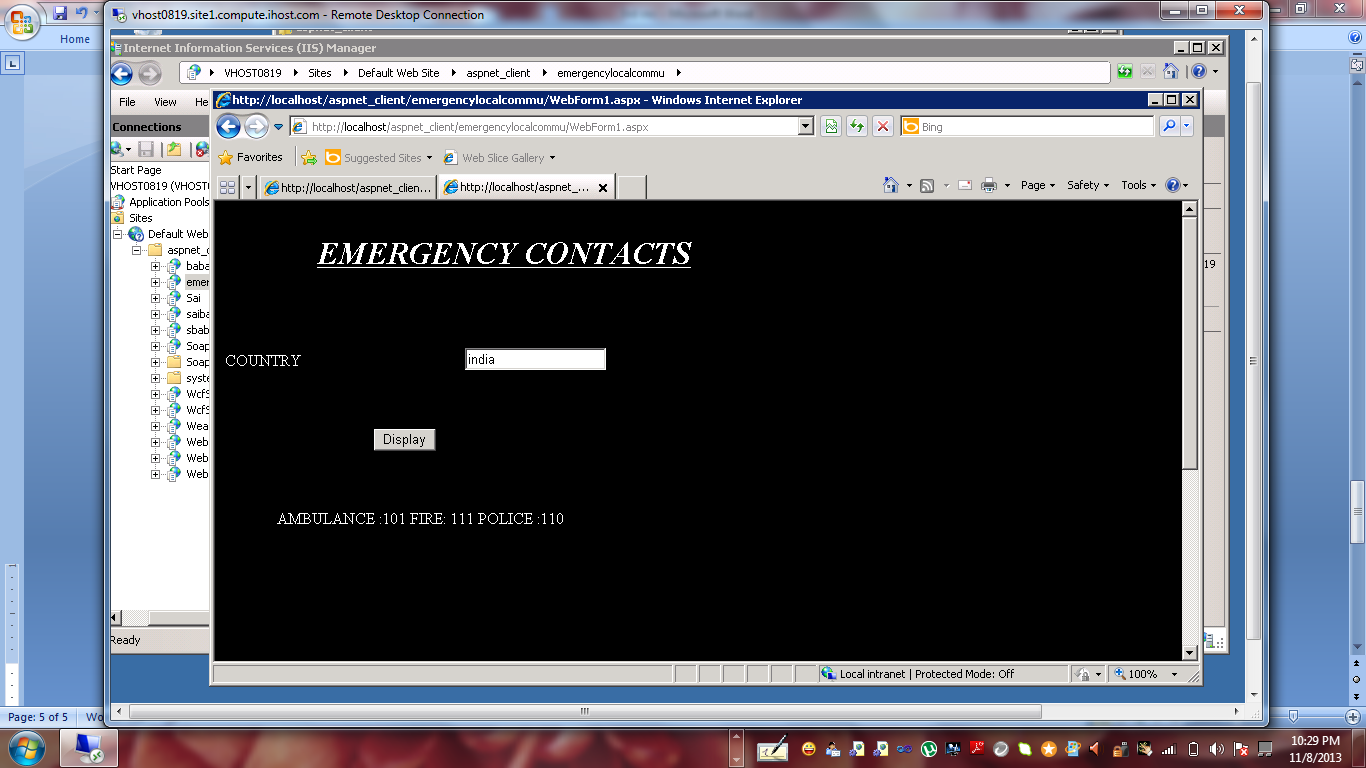
The output of the web service will be as the above one.

Now the above created web service is used our application.

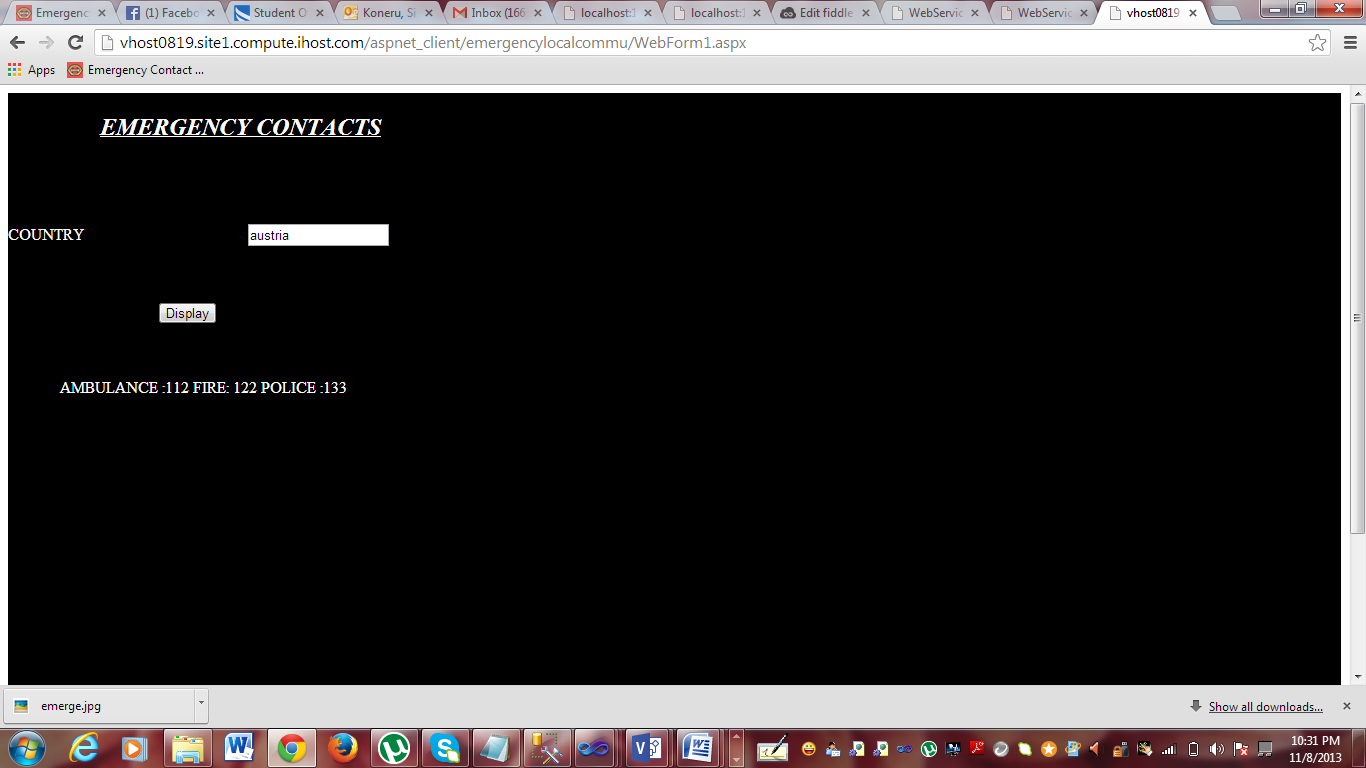


This program is used for the client to use web service created remotely.

Then the program is also deployed in the cloud.

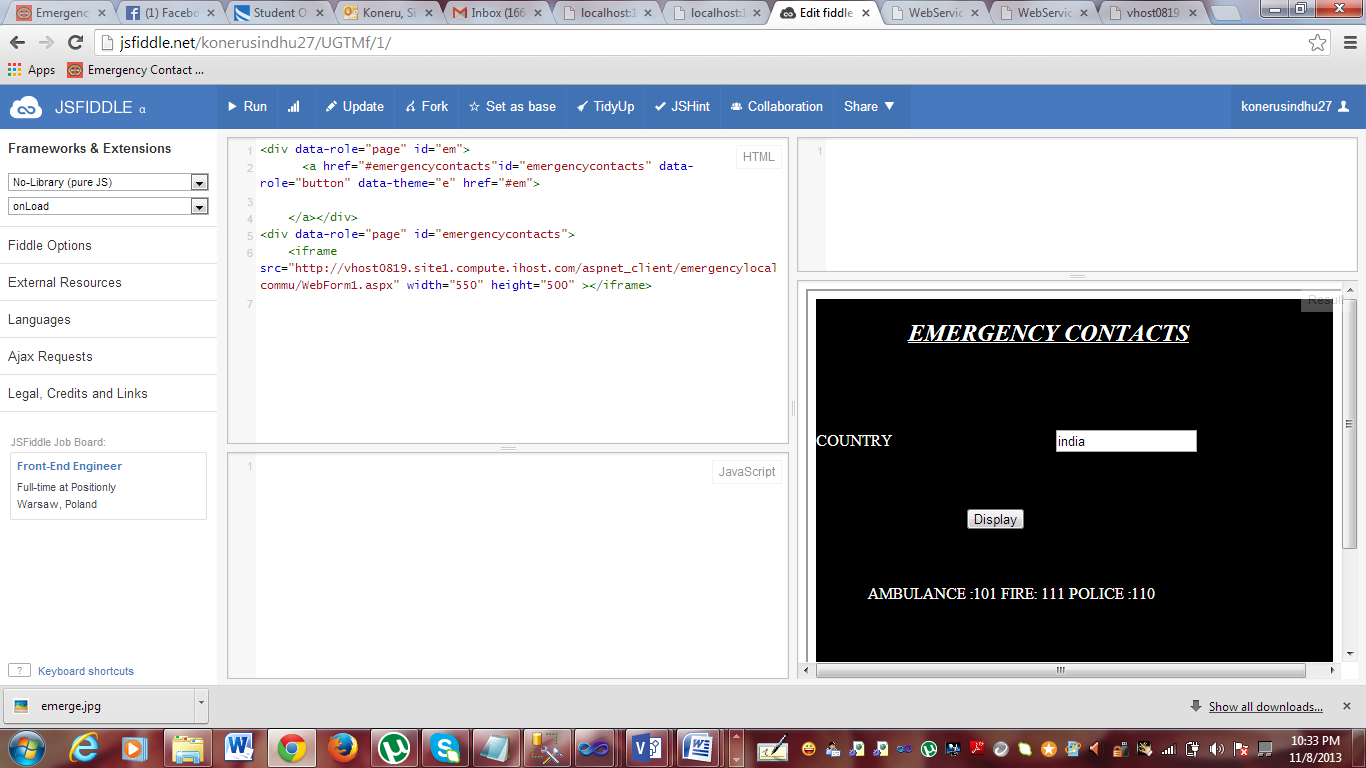


If the country name is given the emergency contacts of ambulance, fire and police is given.



In local host,it looks like this.

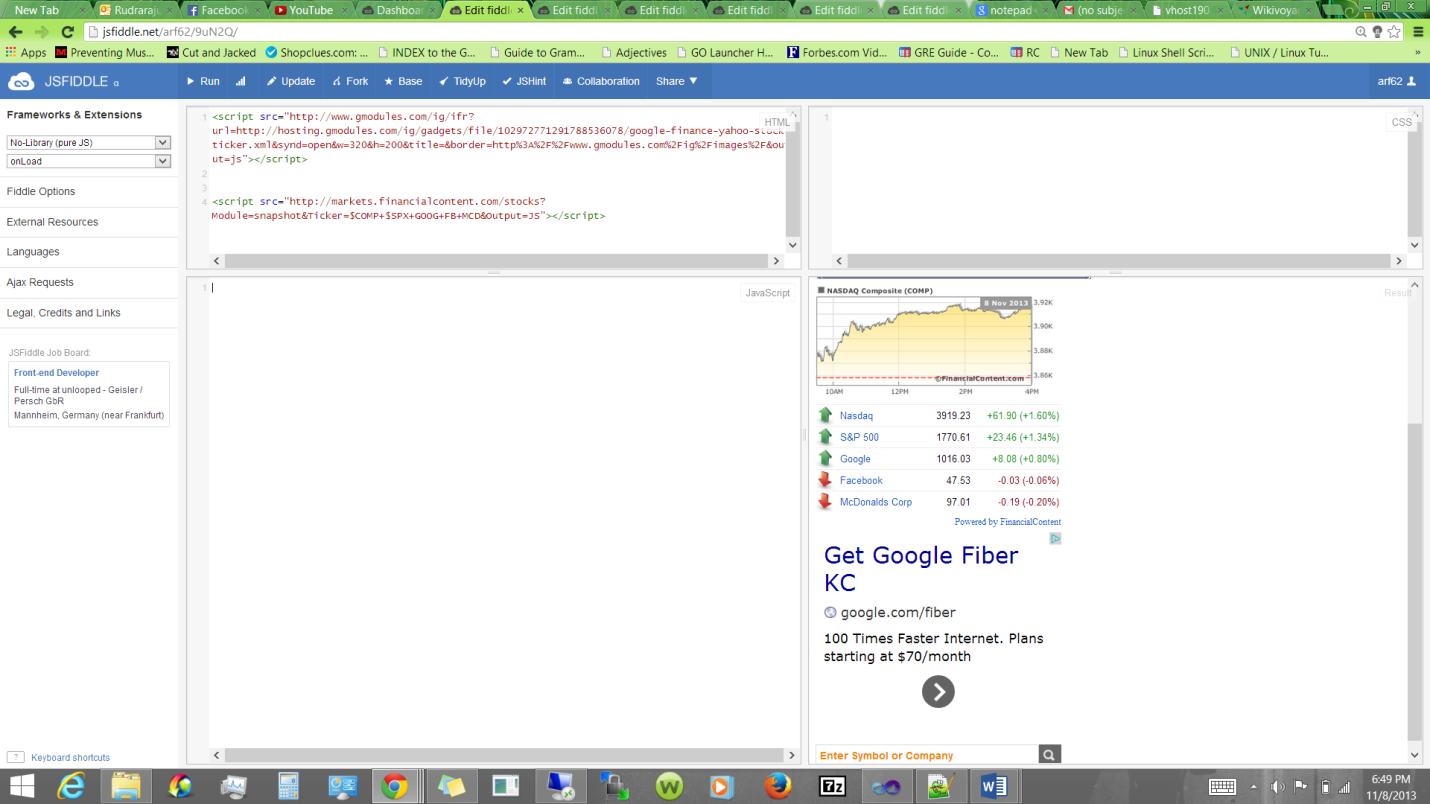
As we are using jsfiddle for this. This service is implemented in jsfiddle.



Finally, the emergency contacts service is implemented.

Stock market updates for business users.

As soon as the user clicks on the stock market button, the user is re directed to the below page with the stock updates of prominent companies.



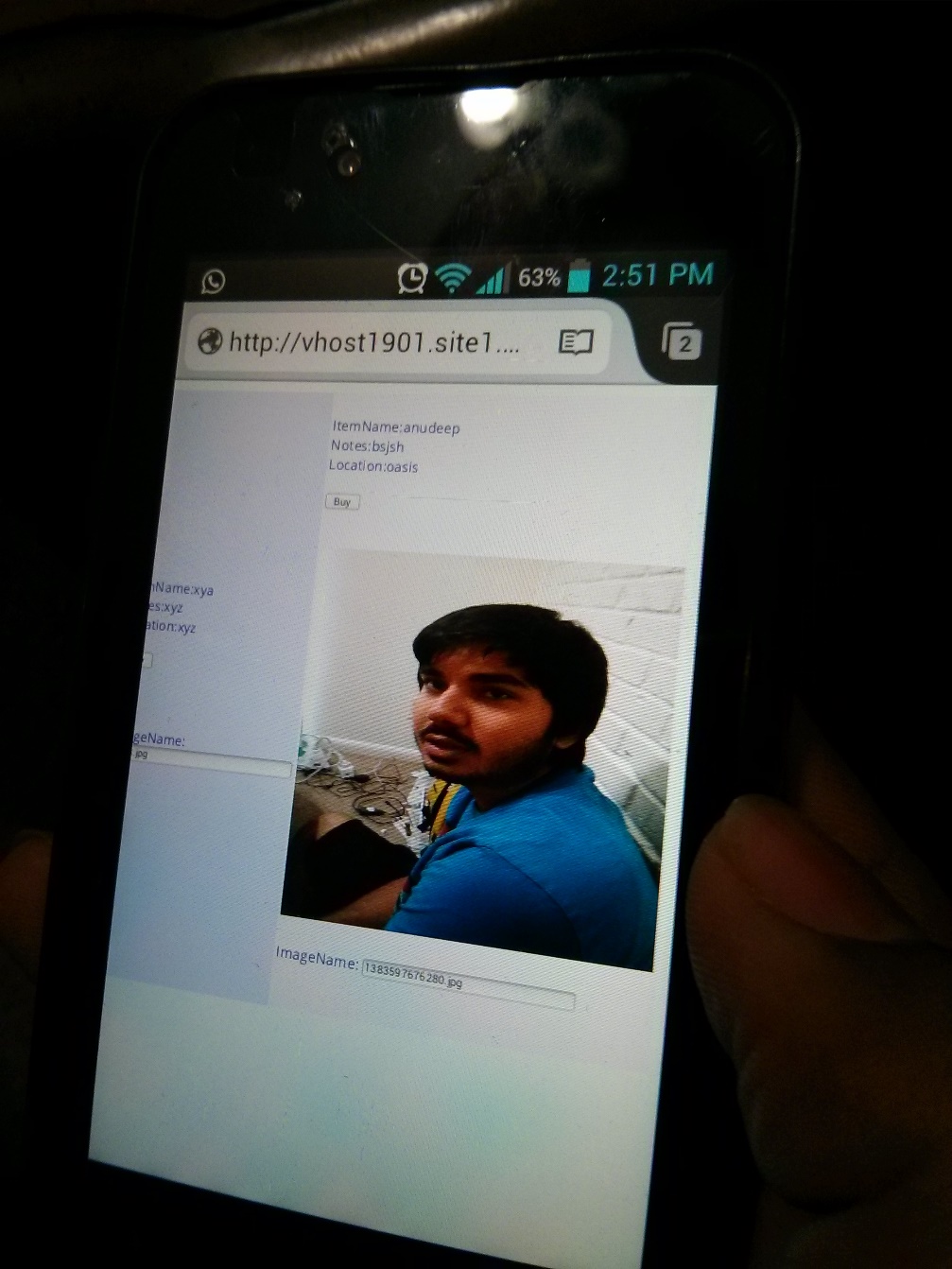
Wiki Voyager Travel guide information.

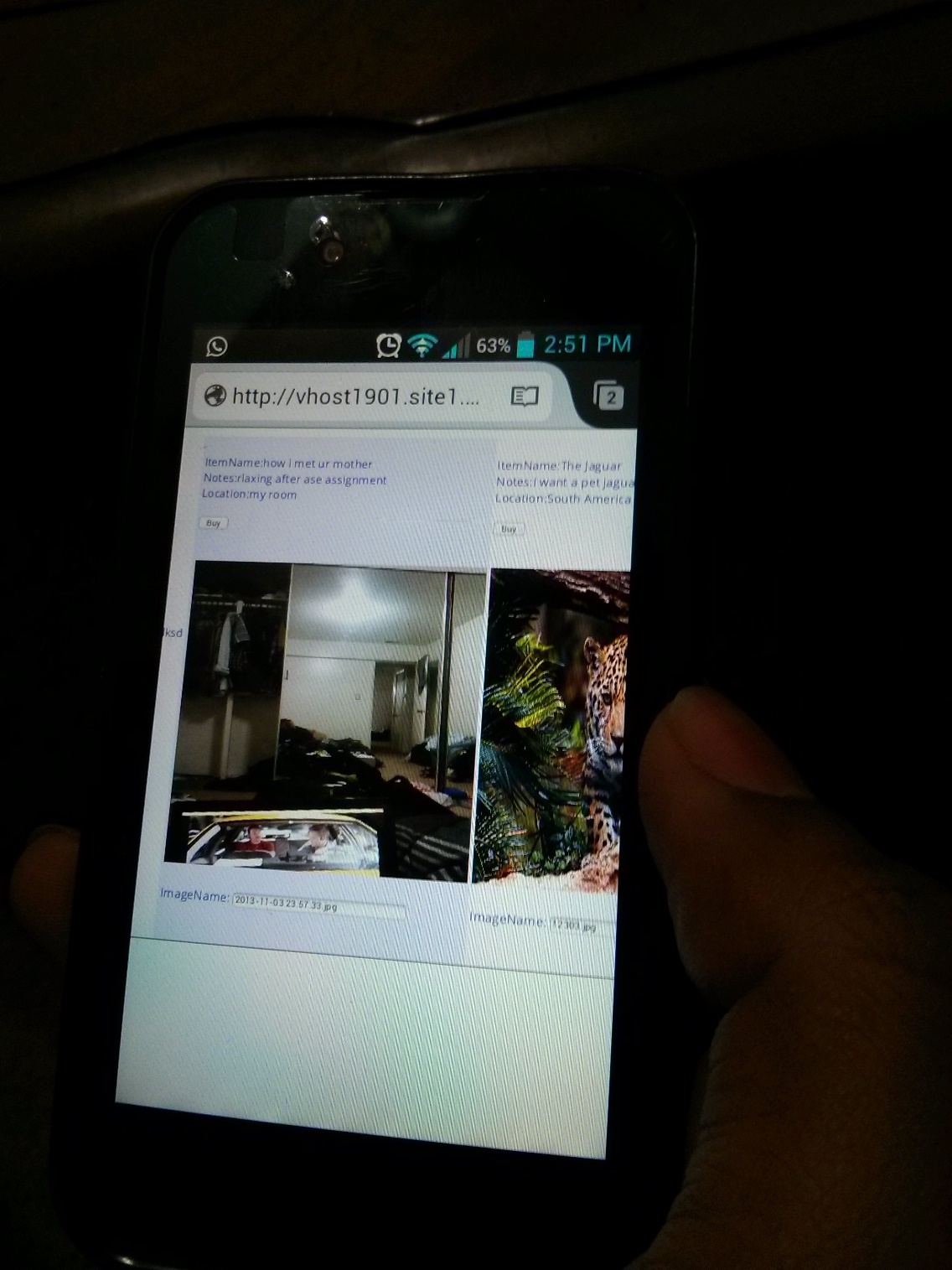
The user provides the name of the state and then clicks on the find a state button a new window opens which redirects the user to the wiki Voyager page of that state.



Take Photos from the application (in android phones)

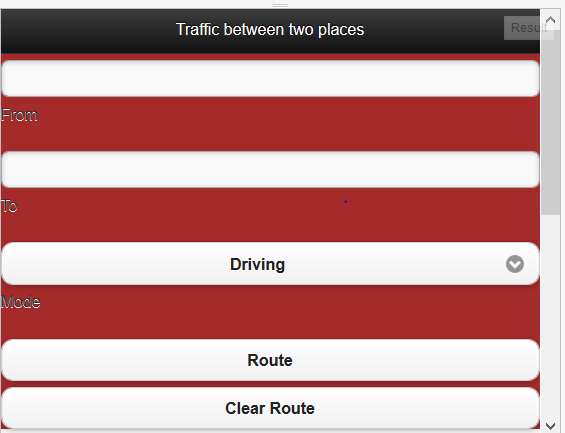
The users can use their android phones to take pictures and then view those pictures later whenever they want.



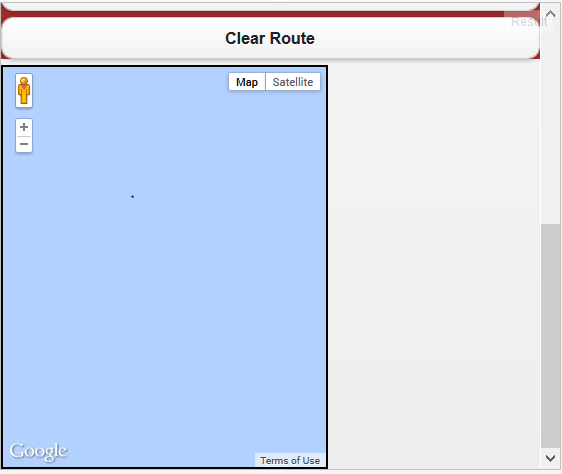


**Updated traffic service:**

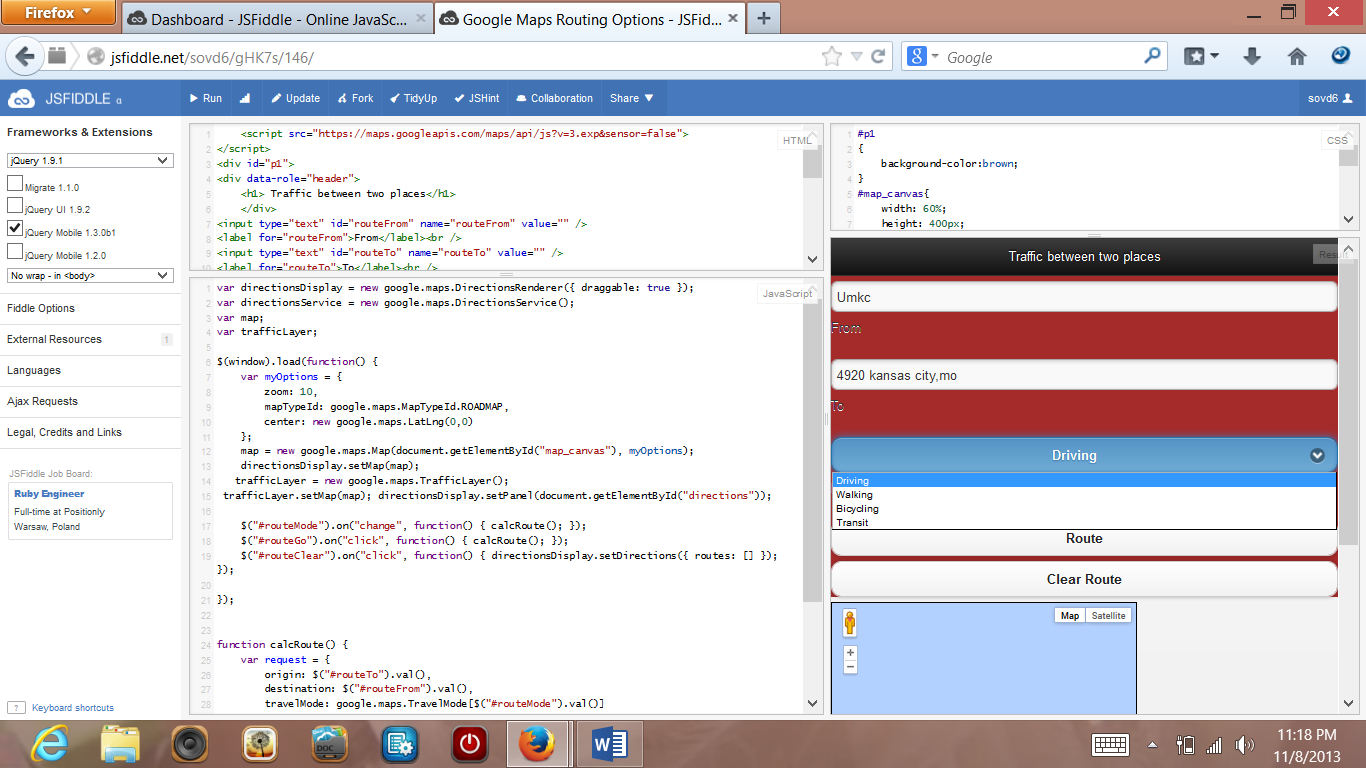
It is too vital for making decisions especially with respect to any rendezvous etc. Alike weather conditions traffic conditions also keep changing with respect to time. Predicting t will have its own benefits, it will enable human to make wise decision keeping traffic in concern.



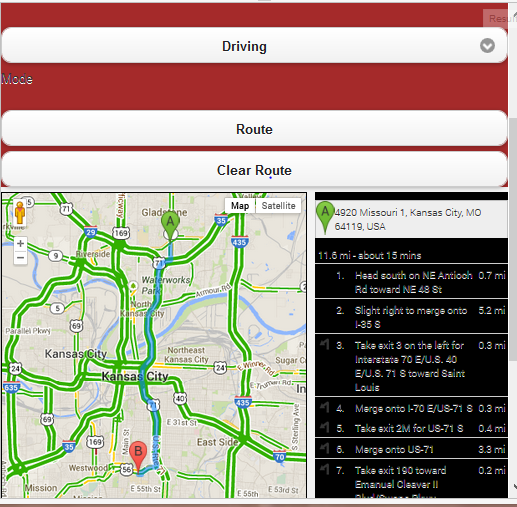
Initial screen appears as shown above. It depicts the intake of source, destination computing the route and displaying the traffic layer created upon the map.



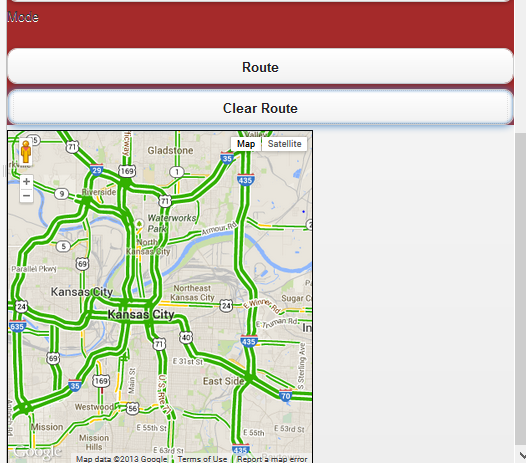
Lower part will have the screen as above. With the intial map initialized with (0,0) co-ordinates.



As an example source is given as umkc. Destination as 4920 grand avenue kc,mo. Choose the mode of transport say walking or driving etc. Here walking is choosen as the mode of transport.



Map along with the navigating route from source to destination. A shows source location and B shows destination. The colored lines from green to red shows intensity of traffic from heavy to thin.



On clicking the clear route the navigation route between the places gets disappeared.

On clicking the route button the route gets displayed again.

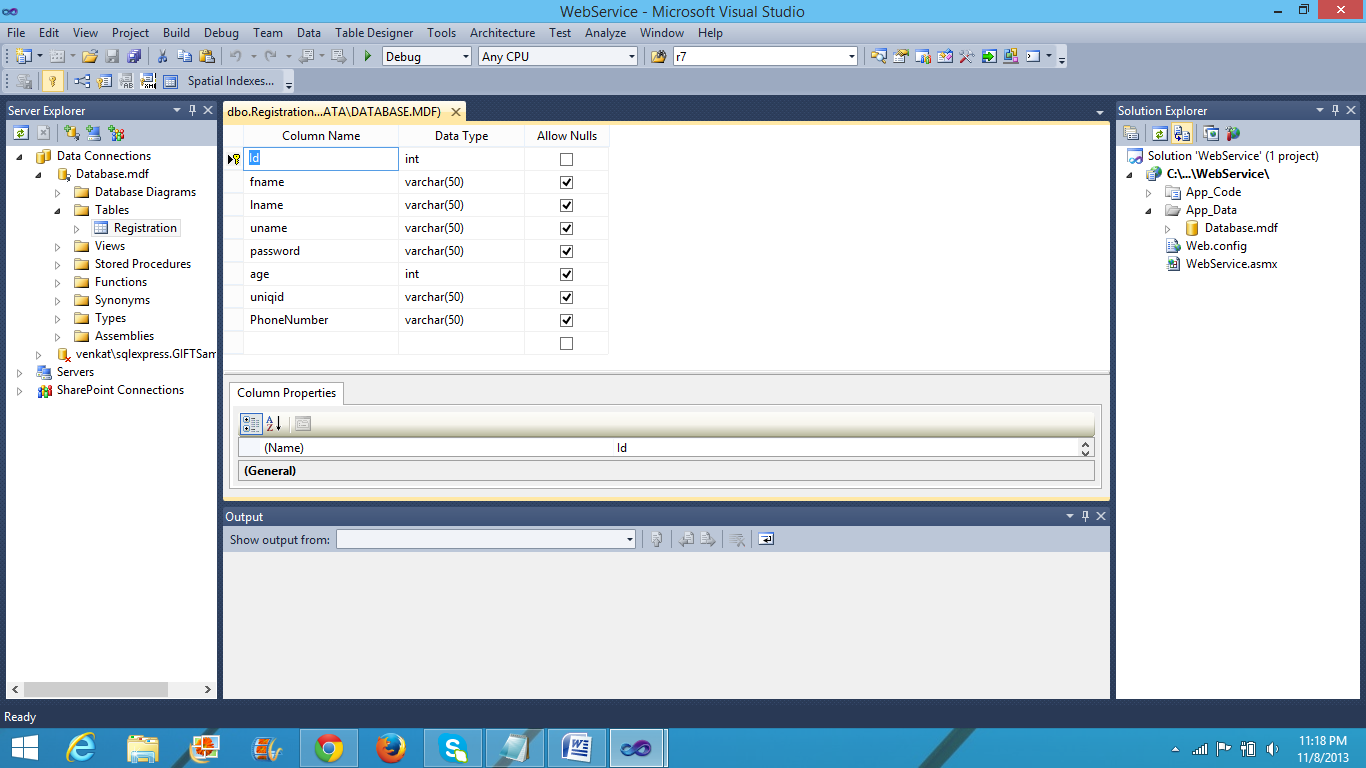
**Registrations**

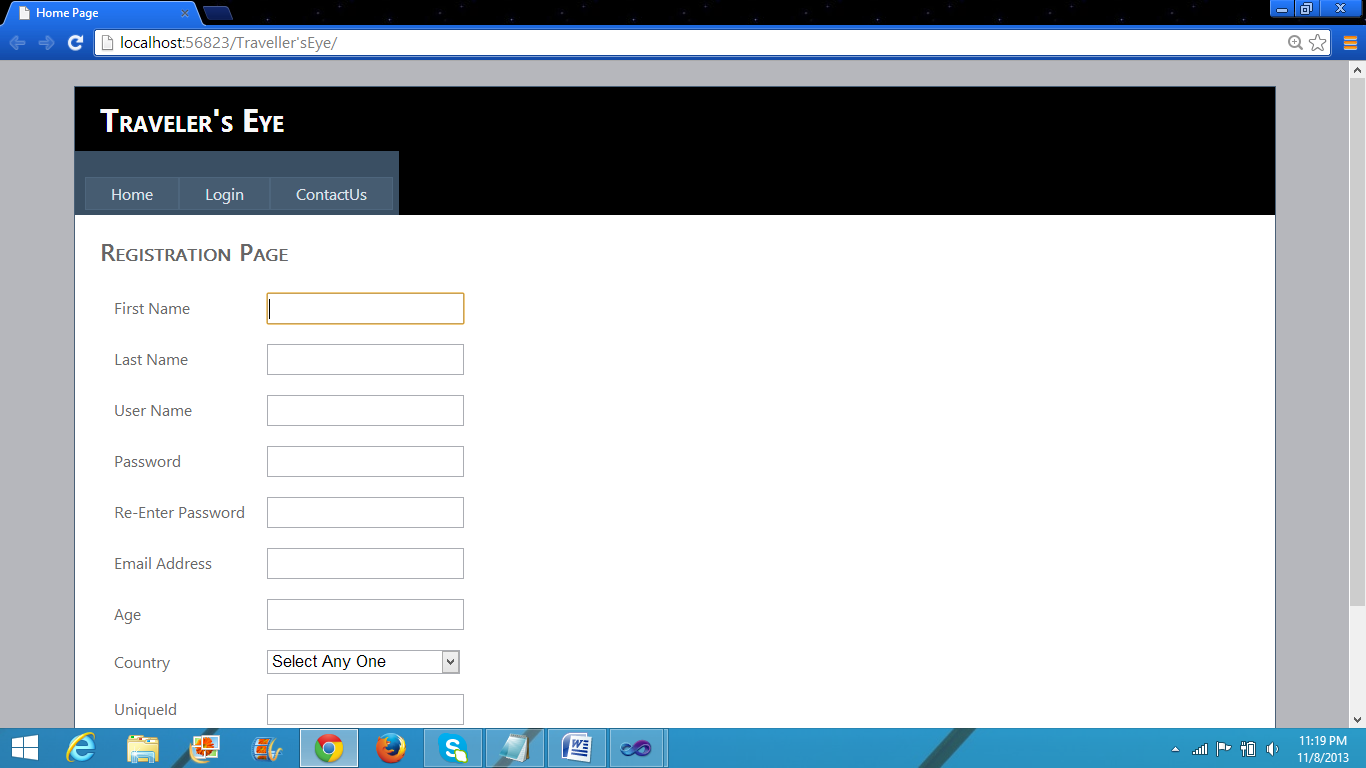
In the Increment -3 I have started working on the database and inserting data into it. I have completed the login and Registration pages client side validations and also implemented a SOAP based web service to register the user details and retrieve the data from that.

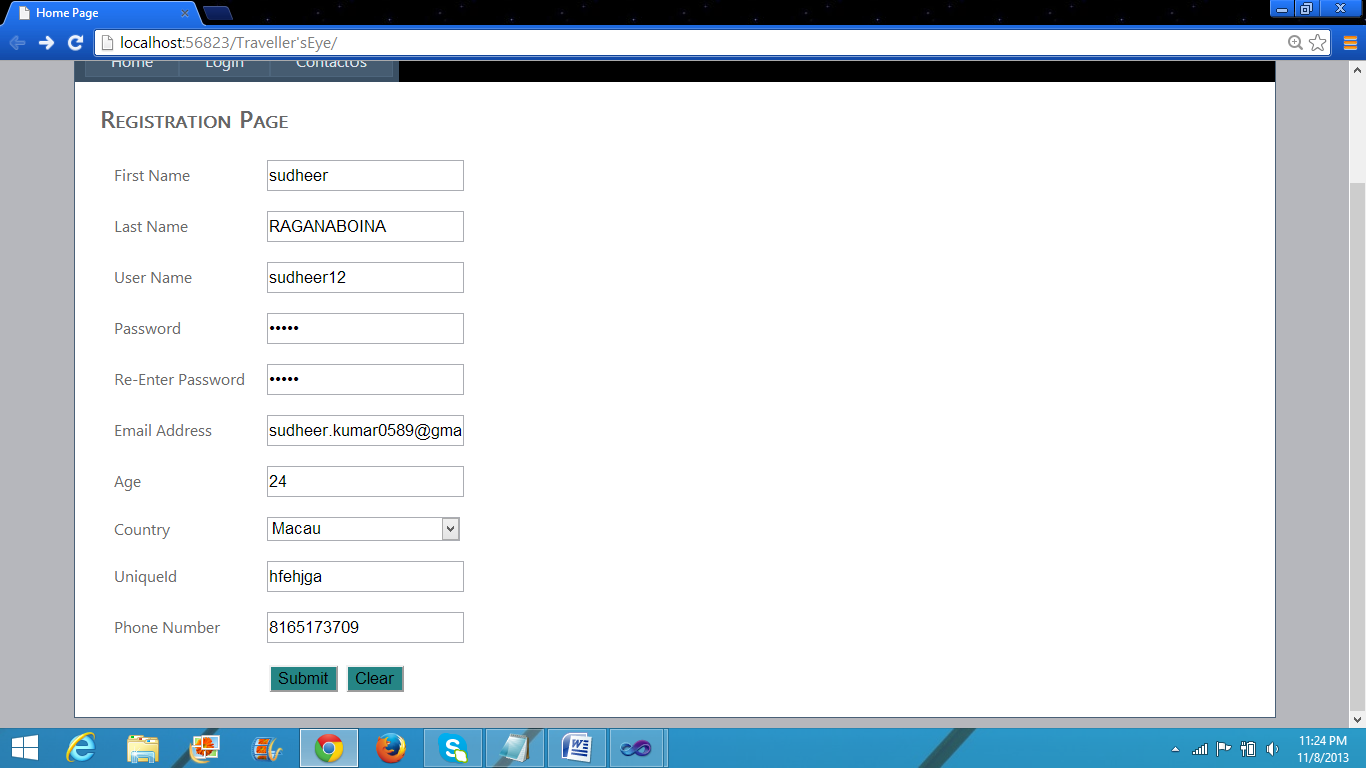
I have successfully able to create a database and push the data into that and was able to retrieve the information from the databases to login page, but using web service's.

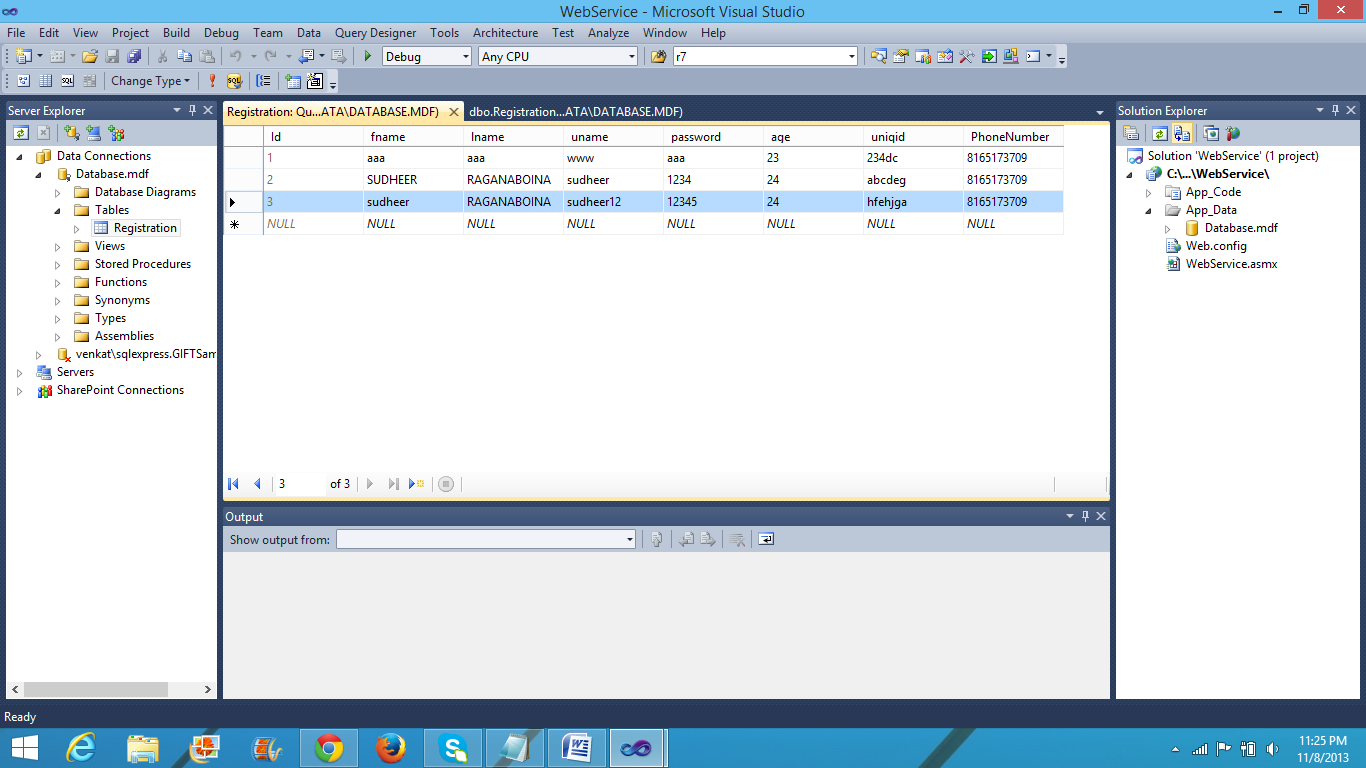
The user can register his details by providing the required fields and then after entering his details , he can use the login icon, enters the credentials and enters in to the site.

web service: http://localhost:37445/WebService/WebService.asmx









**Future work:** I am trying to deploy this code into the remote server, I have got some compatible issues regarding that so I am working on that.

Project Management:

Sindhu Koneru : worked on searching emergency API’s and implemented contact webservices using data base remotely. Client communicating with the service.

Sujitha Onteru: Worked on searching for traffic api’s (Myquest), make an extension to the traffic service by allowing user to give his source and destination, as well mode of travelling.

Ashok Rudraraju: Worked on Wikivoyage.org, uploading pictures from app and getting business details (stock exchange info) to the user.

Sudheer: Worked on gathering data for validations. Implemented the validation of the registration form.

Scrum do link:

<https://www.scrumdo.com/projects/project/project-150/iteration/78369>