

Travel buddy
Second Increment Report

By

Team 19 - Sprinterz

Gattu Sampath
Pottabathula Prathyusha
Sadanand Kallakuri
Nalla Abhiram Reddy

Hotel Search



API :

We used google maps API to implement hotel search . Application has a text field with place auto complete function powered by google . we used google API to find near by search places to get details of Near by hotels .



Nearby Search Requests

```
service = new google.maps.places.PlacesService(map);
service.nearbySearch(request, callback);
```

```
    place: {
      placeId: results[0].place_id,
      location: results[0].geometry.location
    }
```

Auto complete places

```
var input = document.getElementById('searchTextField');
var options = {
  bounds: defaultBounds,
  types: ['establishment']
};

autocomplete = new google.maps.places.Autocomplete(input, options);
```

Language Translator



API:

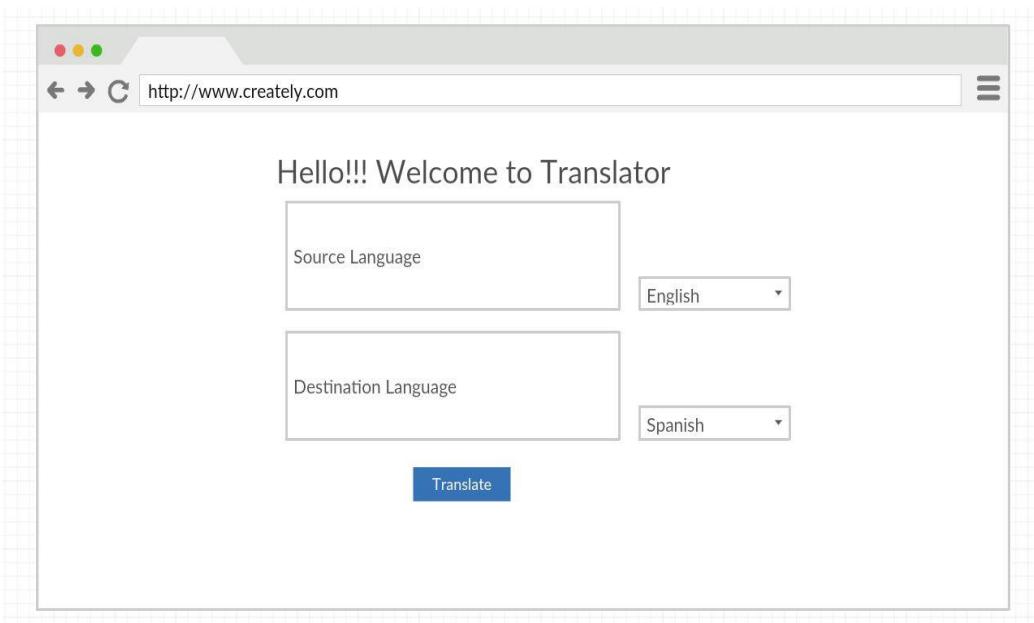
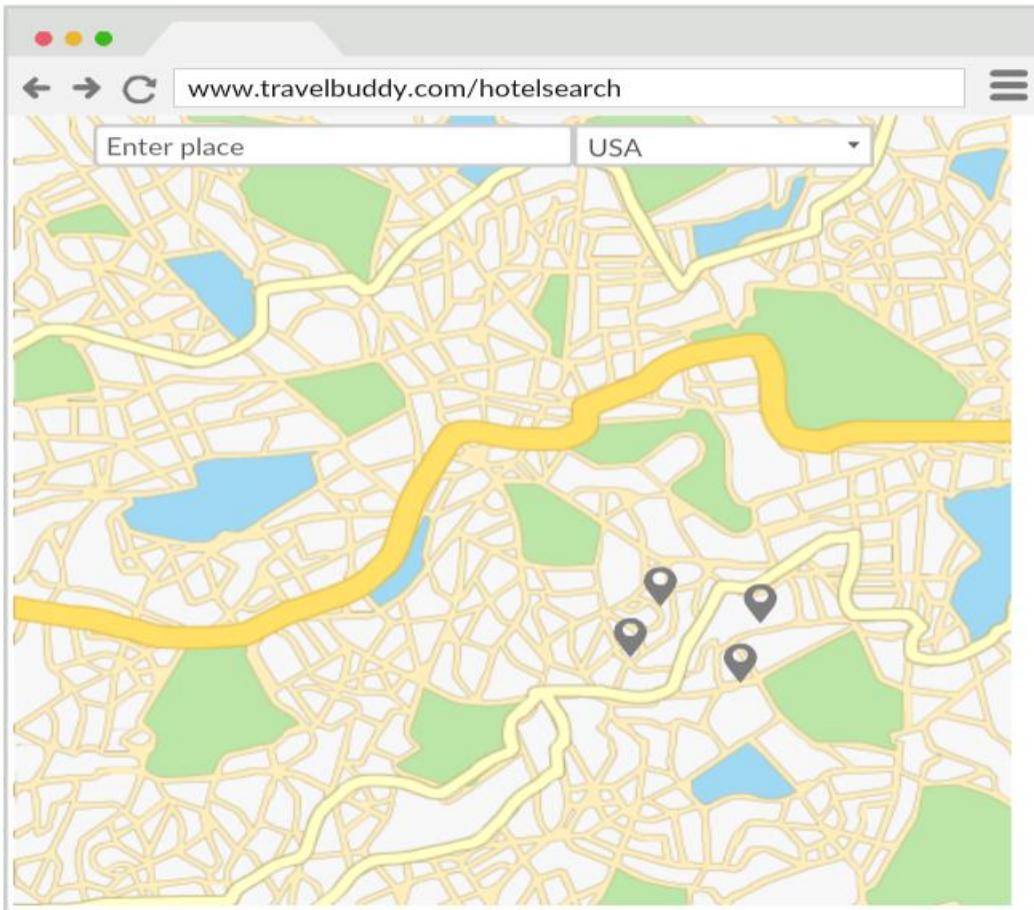
We have used Yandex API to language translation. The API gives access to the Yandex which is an online machine interpretation benefit. It underpins more than 70 dialects and can decipher isolate words or finish writings. The API makes it conceivable to insert Yandex. Interpret in a portable application or web benefit for end clients. Or, then again decipher expansive amounts of content, for example, specialized documentation

Yandex API usage:

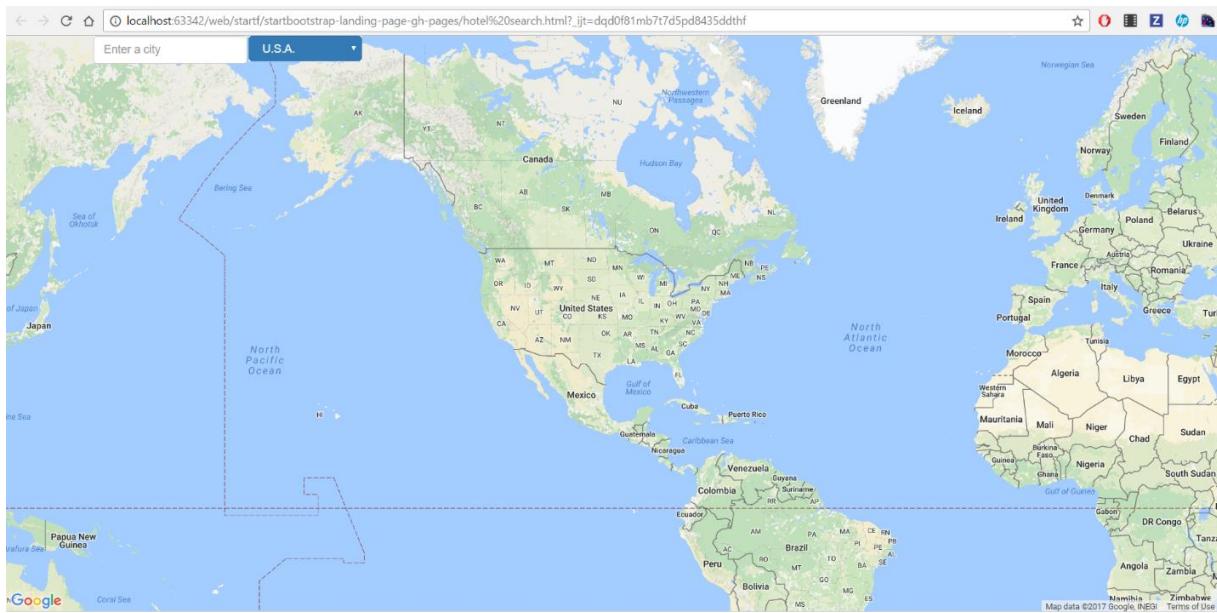
```
$http({
  method: 'GET',
  url : 'https://translate.yandex.net/api/v1.5/tr.json/translate?key=trns1.1.20170310T171435Z.71ee05394b2de15e.7458d4feed706ef41287d167b',
  contentType: "application/json"
}).success(function(response) {
  $scope.convertedText = response.text;
  document.getElementById("DestinationText").value=response.text;
});
});
```

Wireframe:

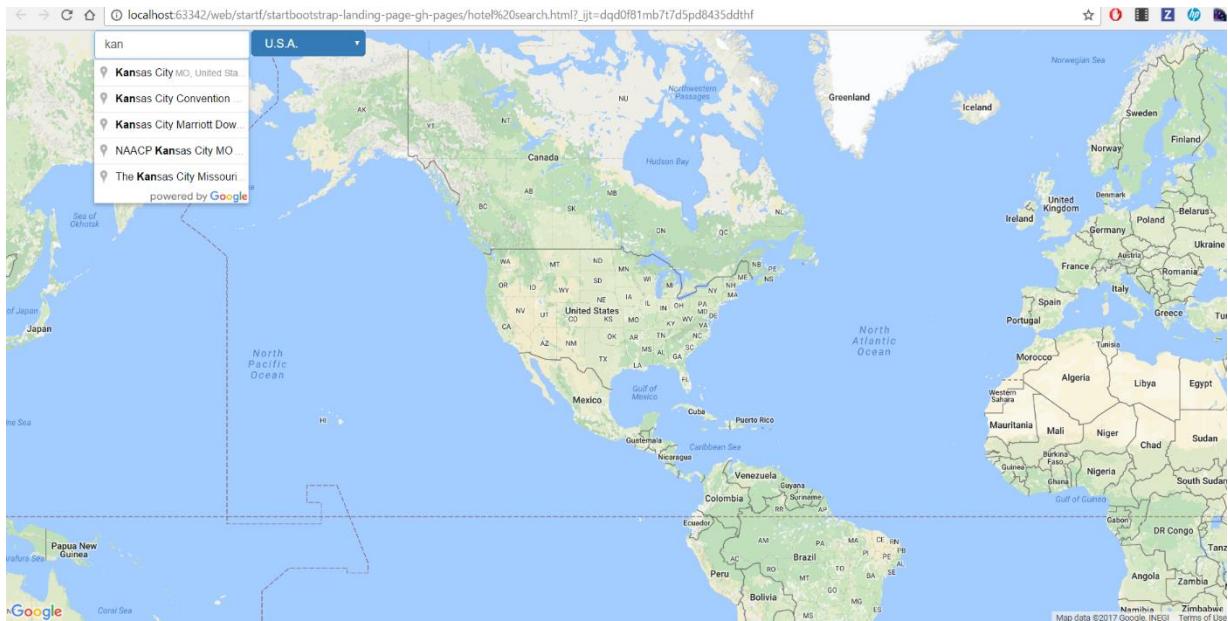
Screen shots :



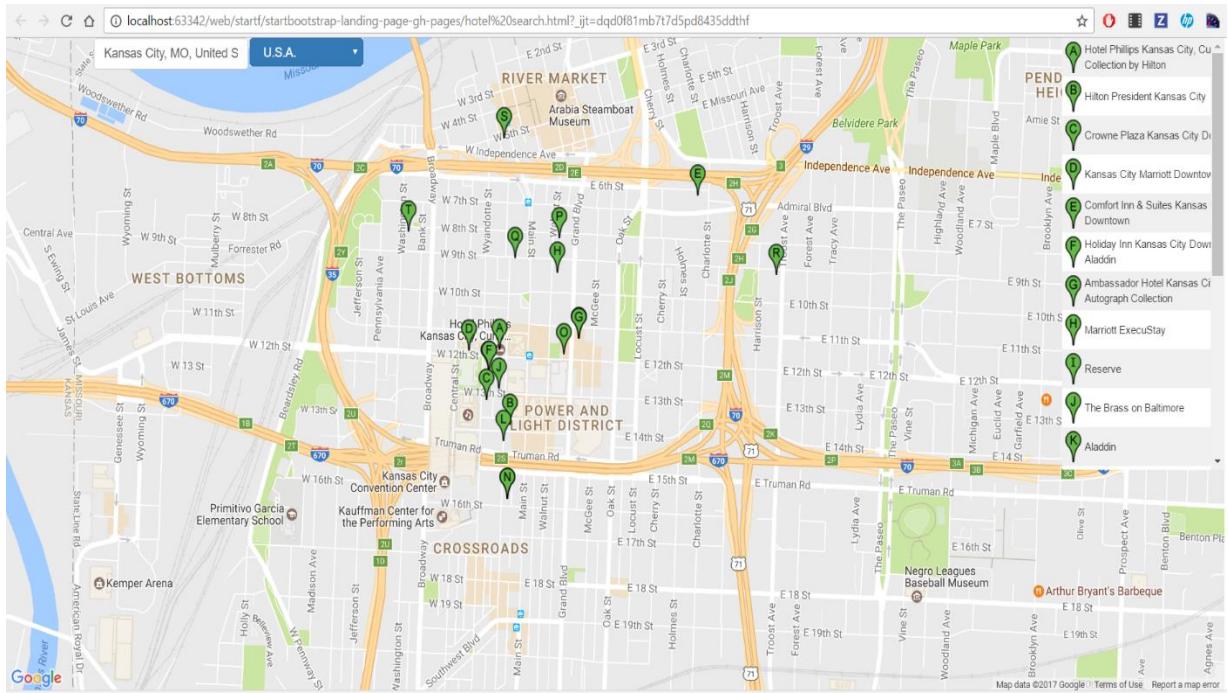
Hotel search page



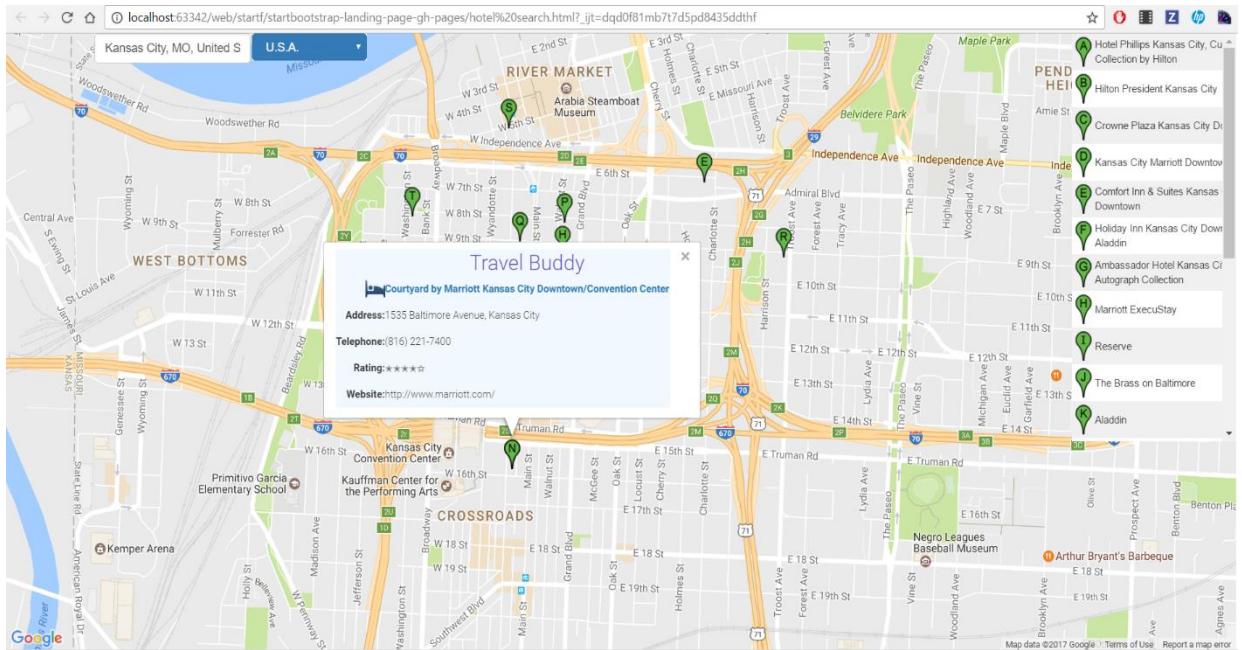
Auto complete

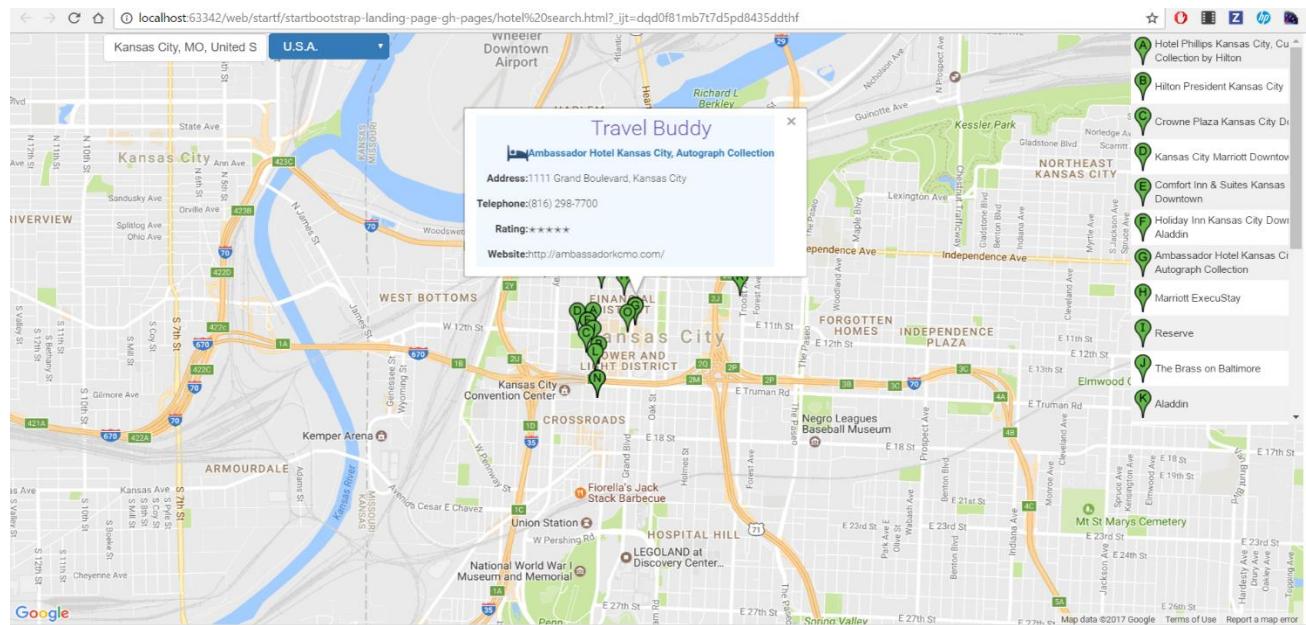


Hotel search

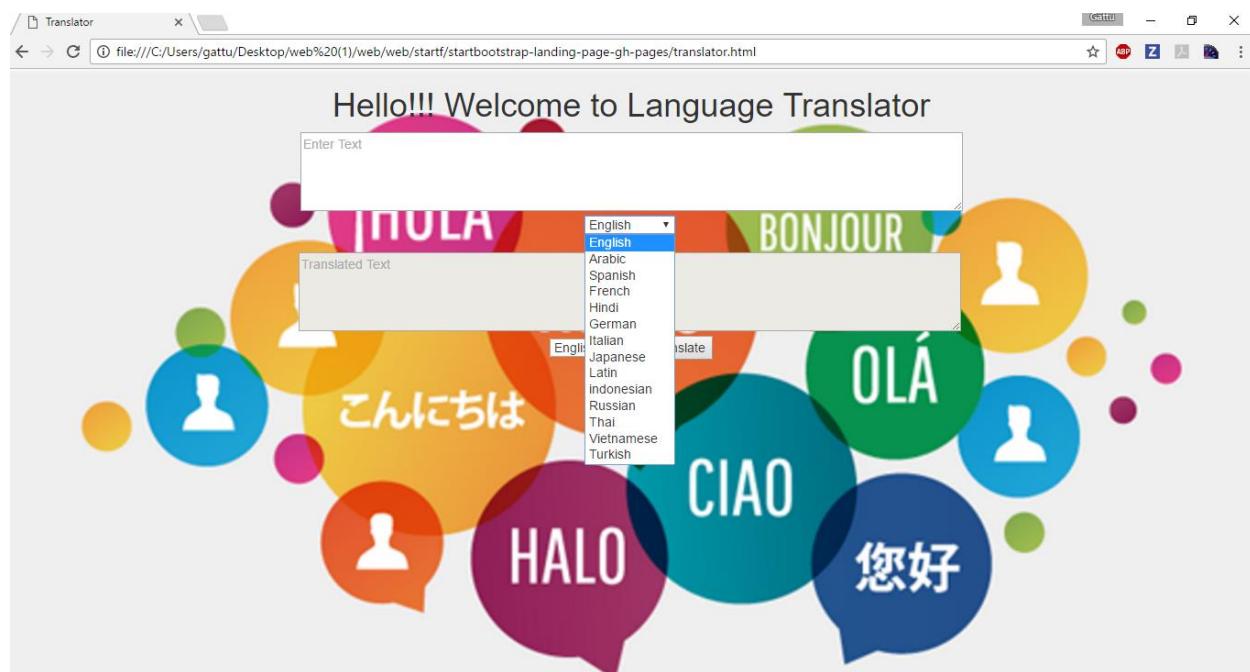


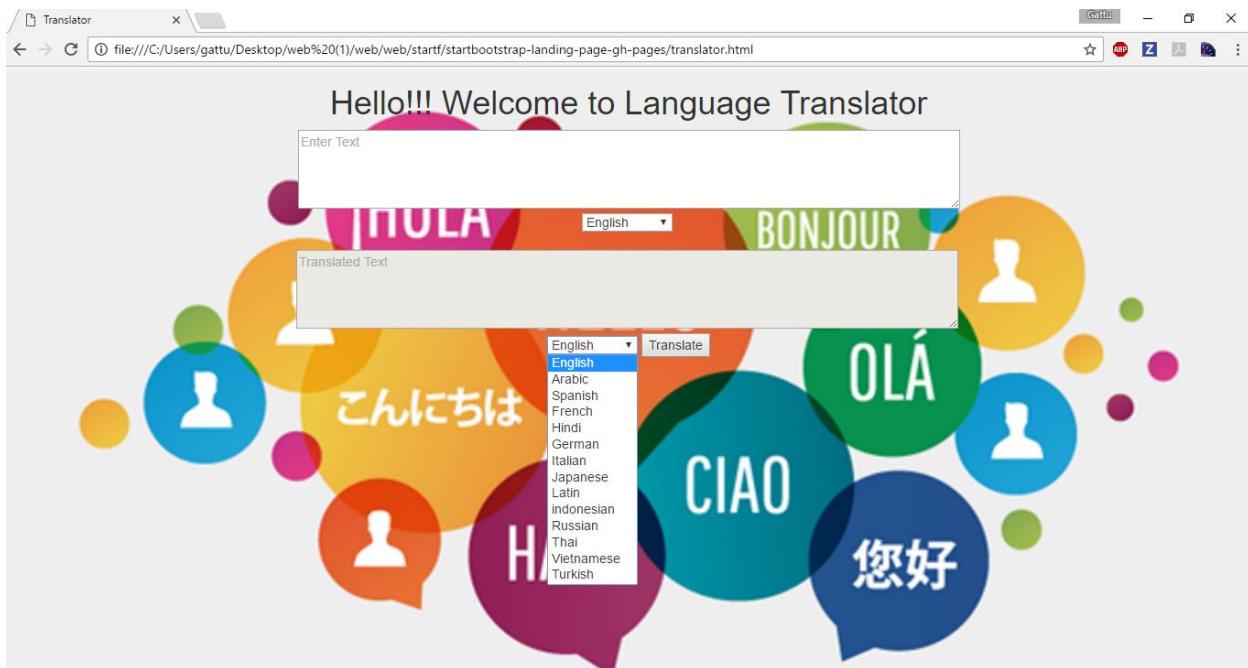
Hotel details





Translator

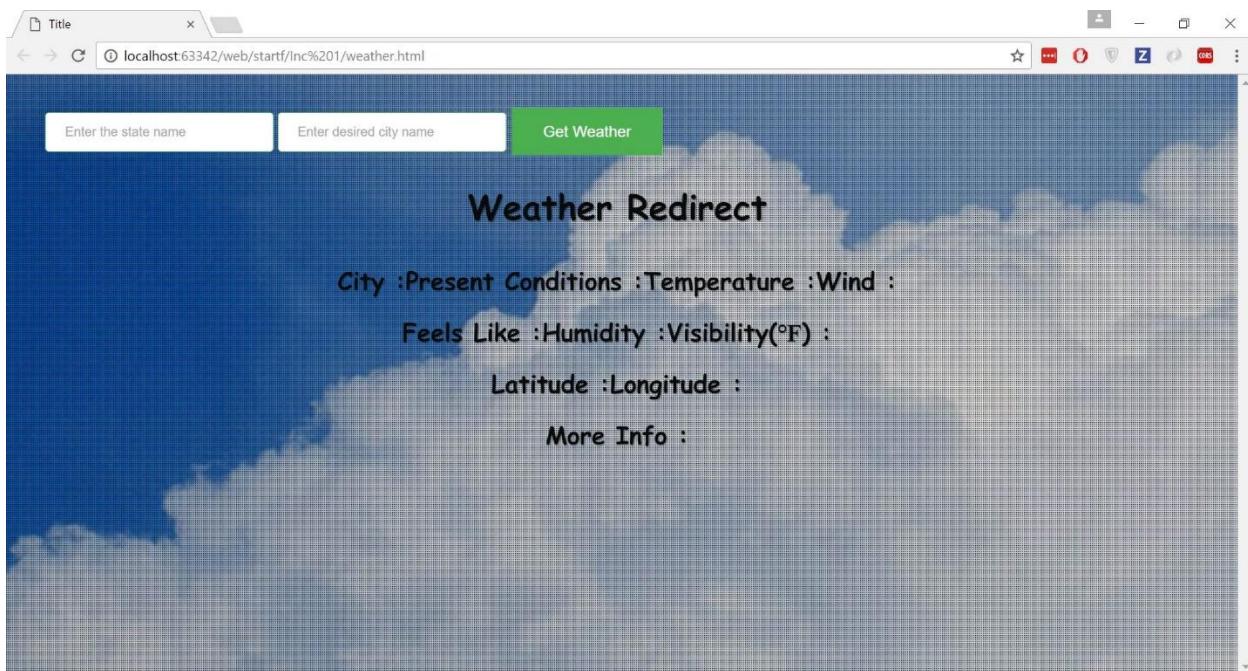




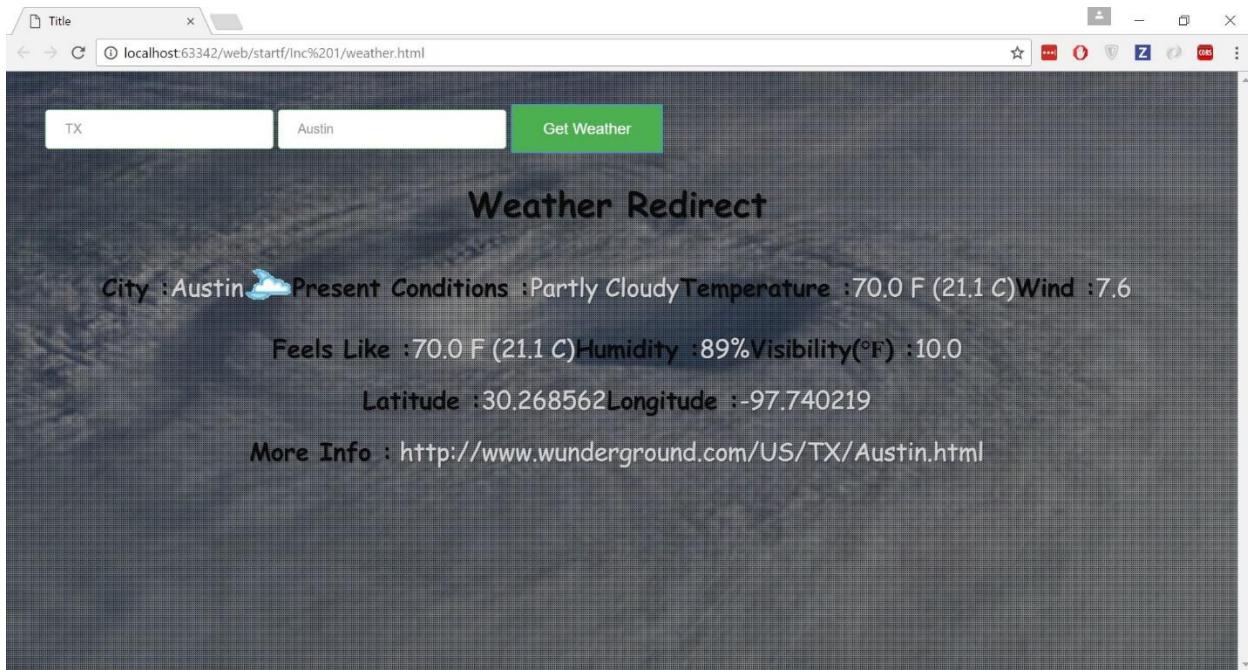


Weather API

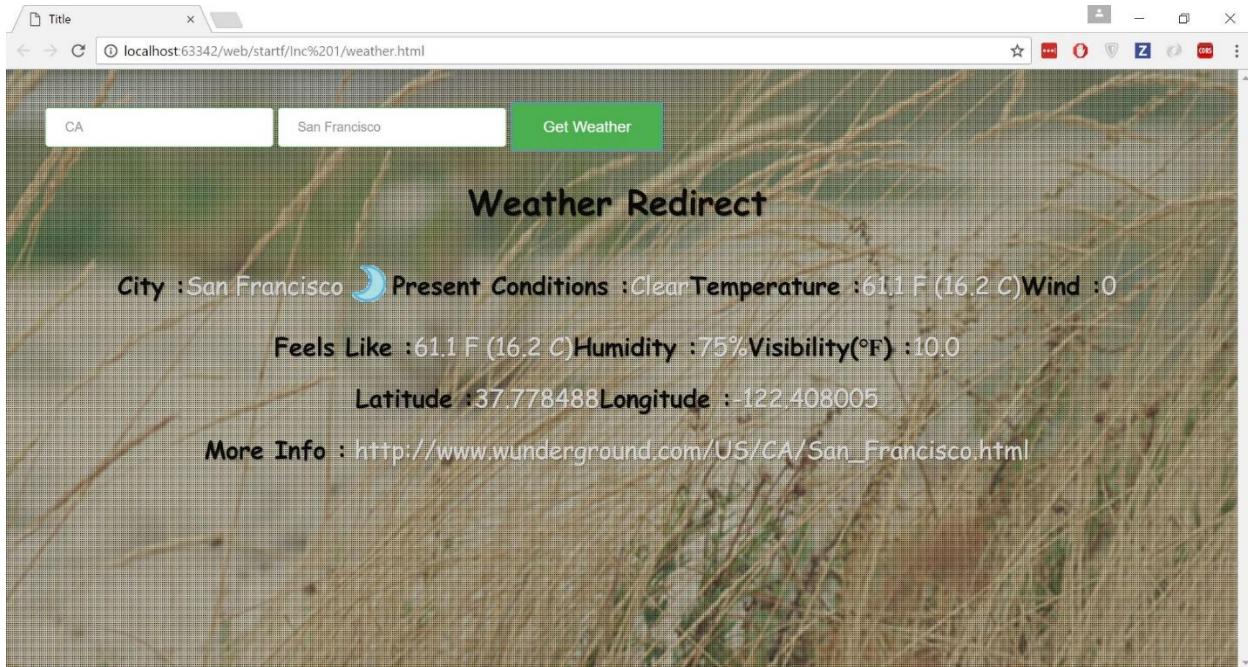
For this project we have used weatherunderground api for retrieving the weather conditions for a place given by the user. Our weather page provides information about the climate conditions, temperature, wind speed, humidity, visibility, also a link is provided for some additional information. The below image shows the weather page accessed by pressing on the weather image in the home page of the application.



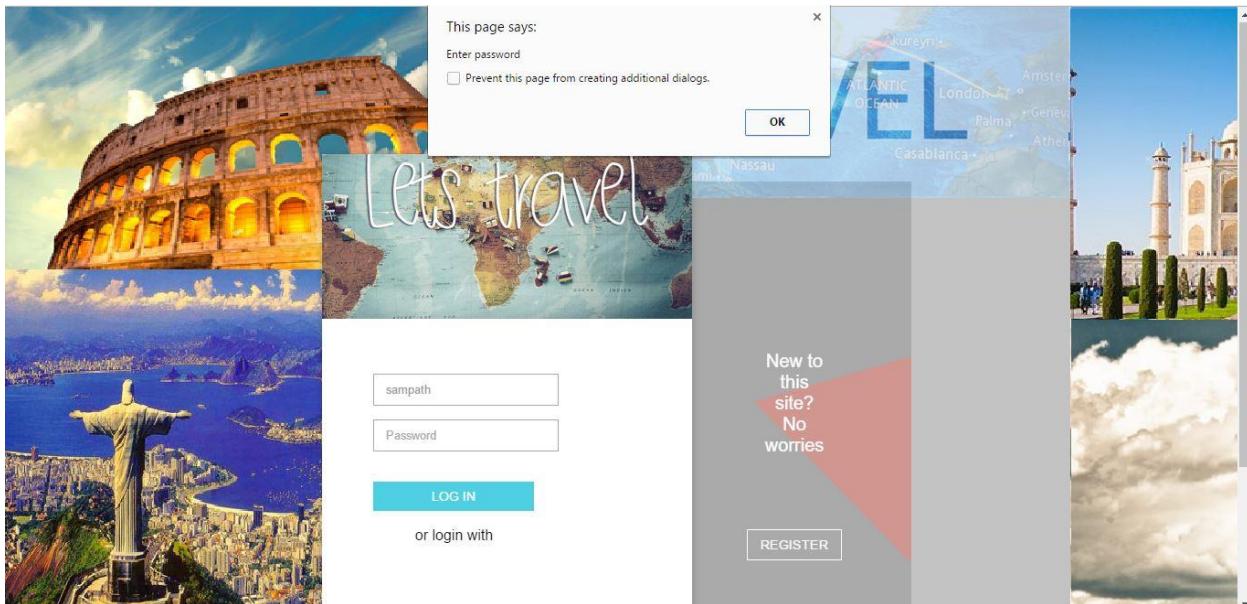
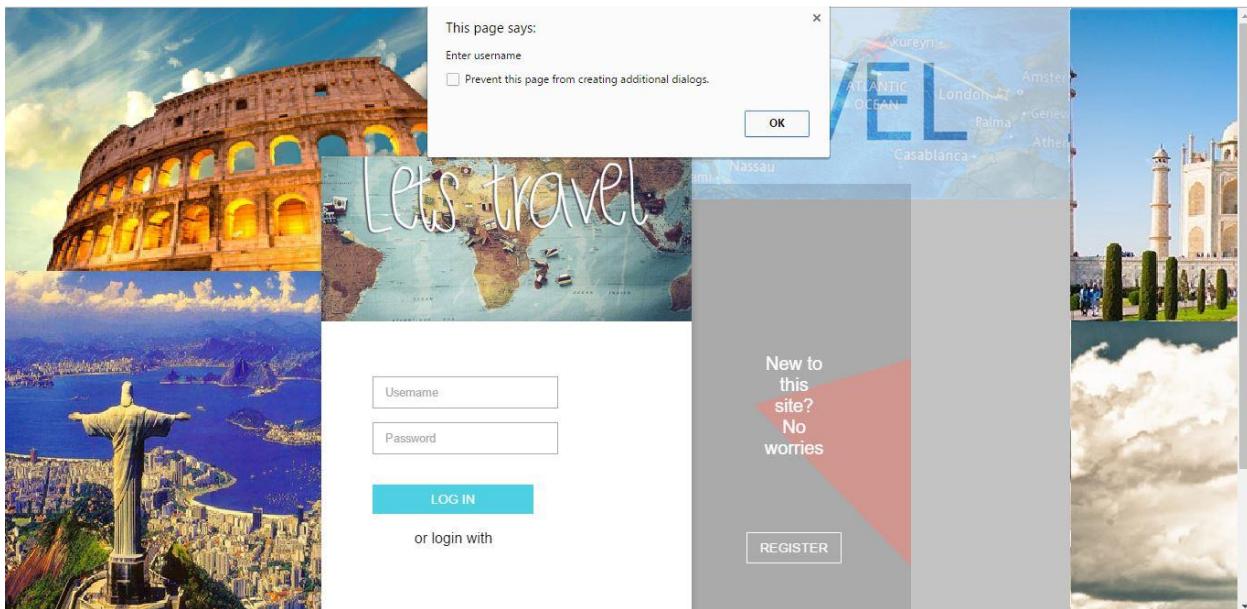
The below image shows the climate conditions in Austin.

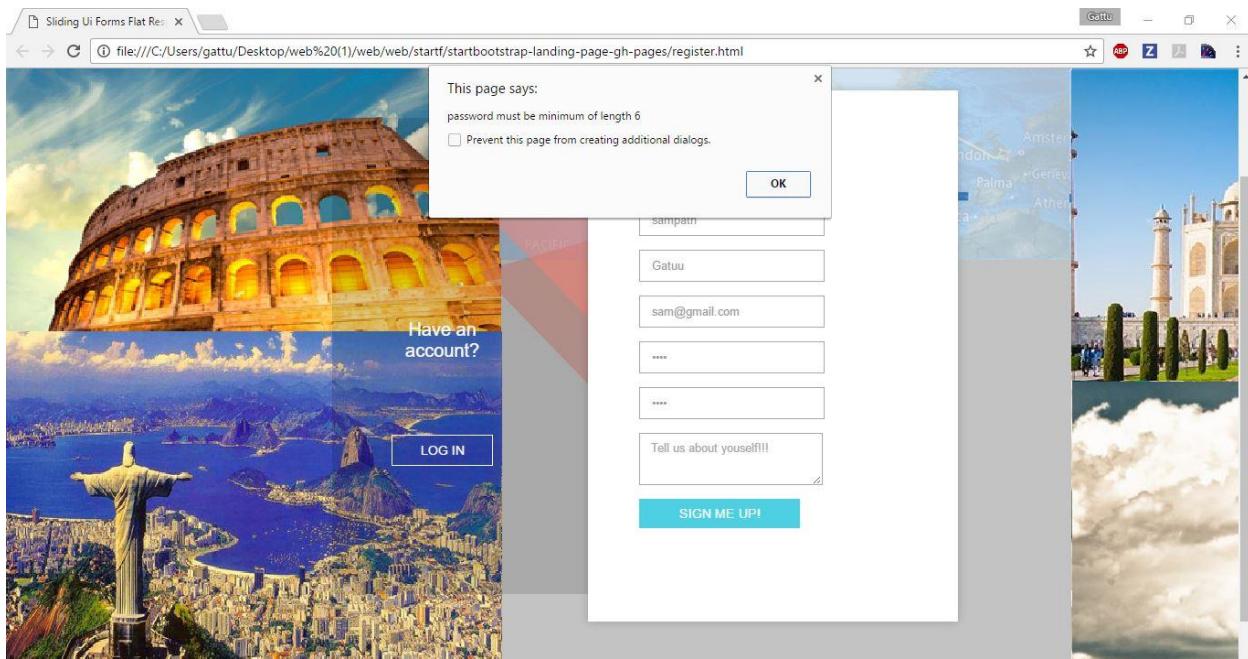
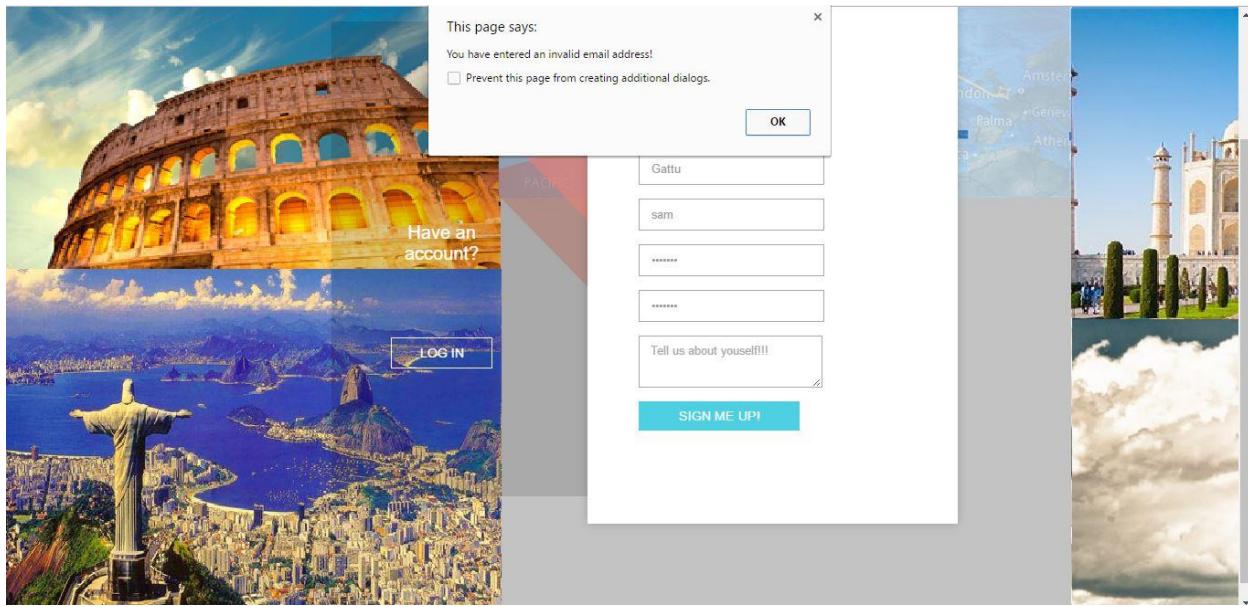


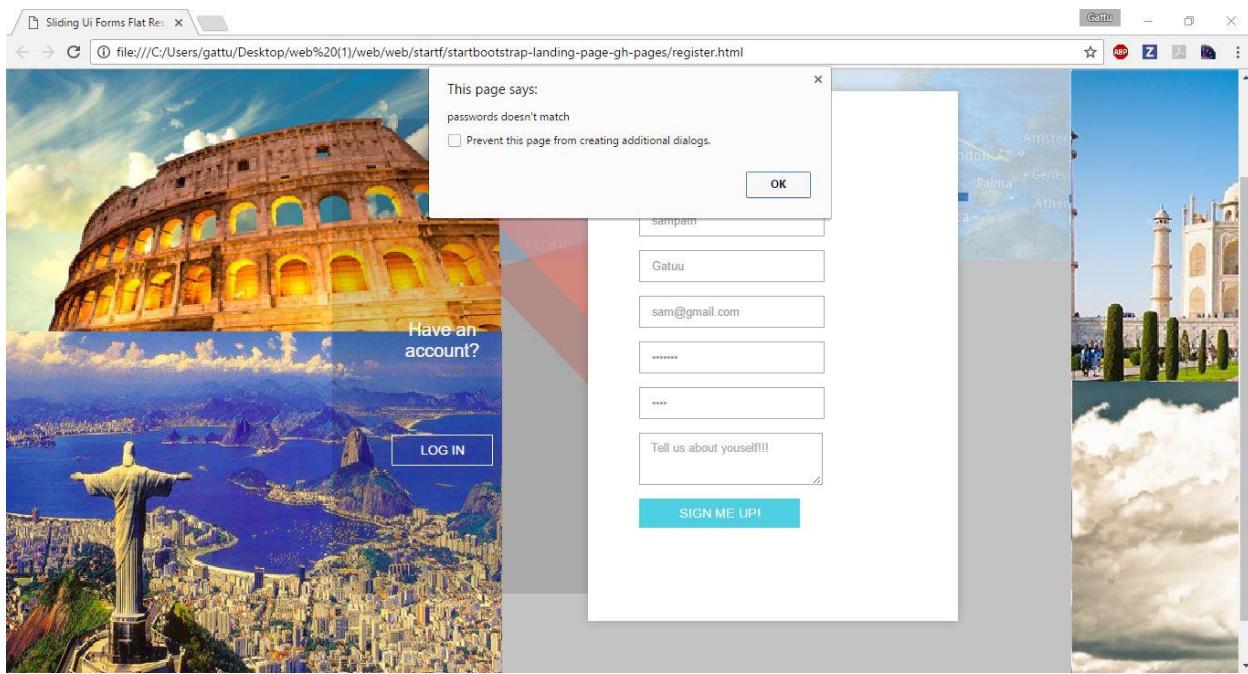
The below image shows the climate conditions of San Francisco.



Sign in /registration Page Validation



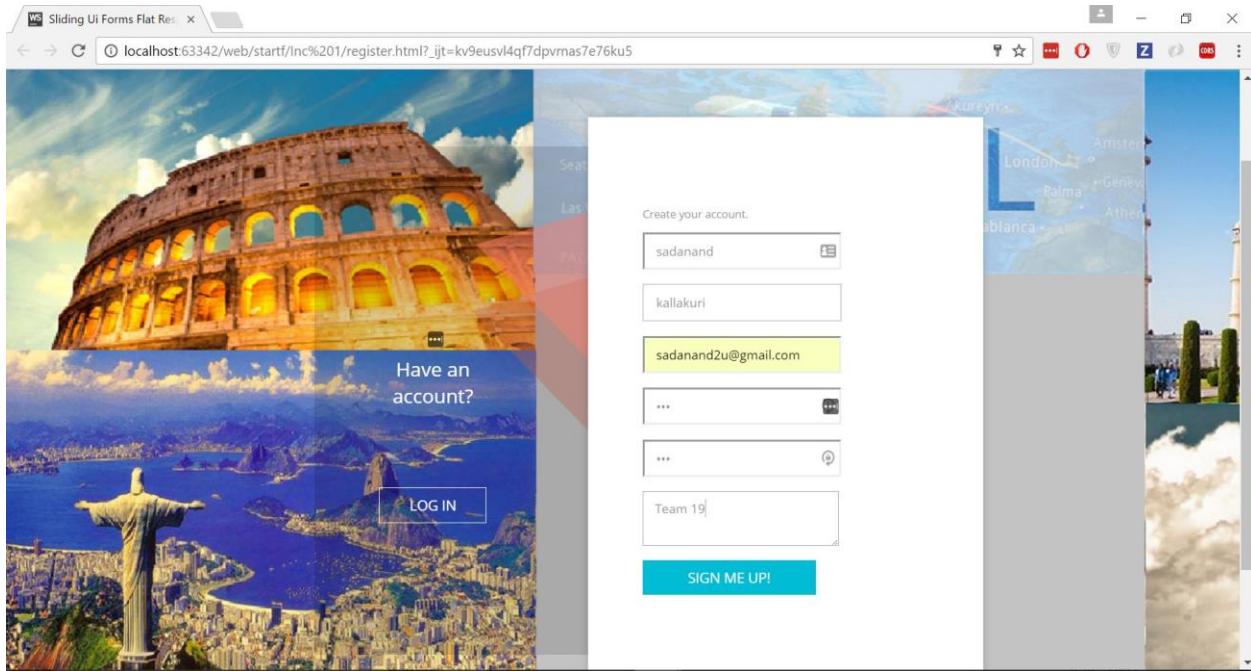




Mongo DB

Four our project MongoDB is used for storing the user's data.

In the below image, the data is being entered in the register page.



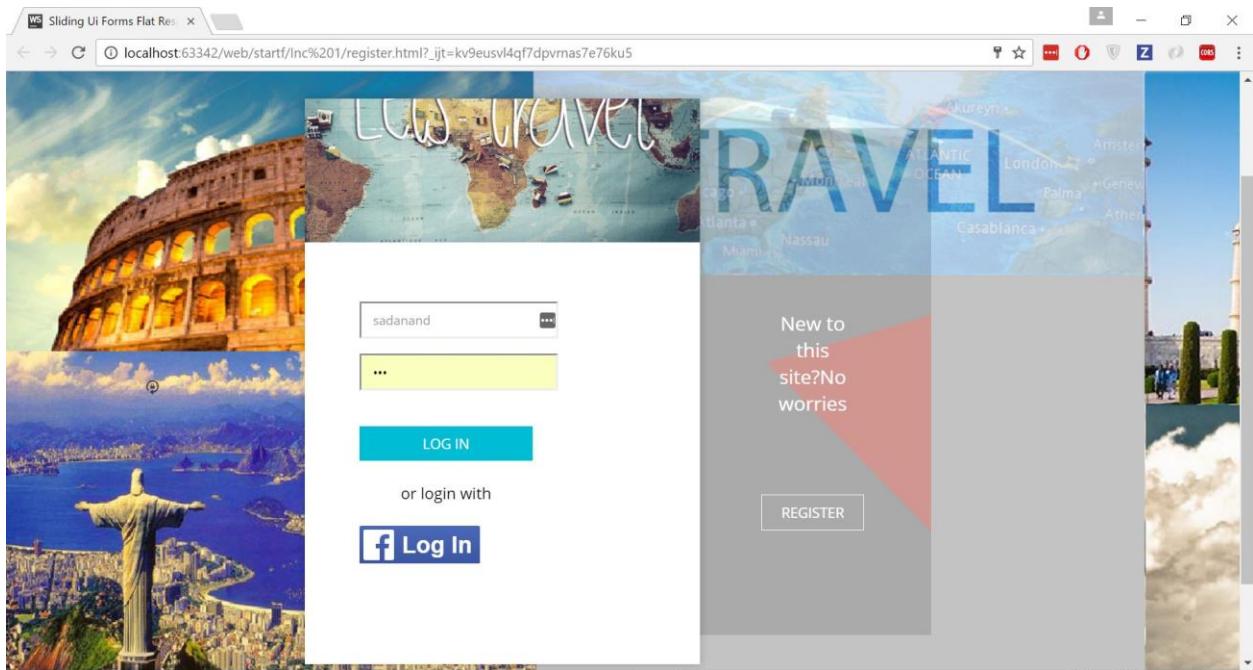
On successful entry, these details are stored in the mongo database, as shown in the below image.

A screenshot of the mLab MongoDB interface. It shows a list of user documents in a table format. One document is highlighted, showing the following data:

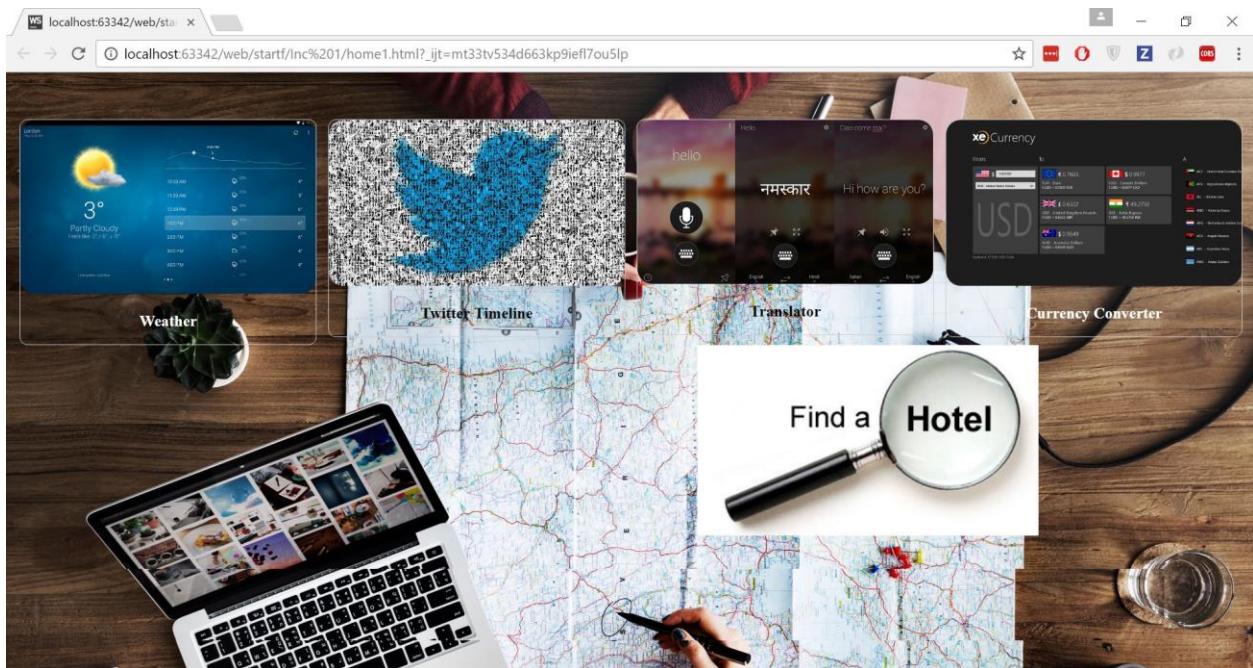
```
{  
  "firstname": "sadanand",  
  "lastname": "kallakuri",  
  "email": "sadanand2u@gmail.com",  
  "password": "sad",  
  "cpassword": "sad"  
}
```

The interface includes navigation buttons for records per page (10) and a total count of 6 of 6.

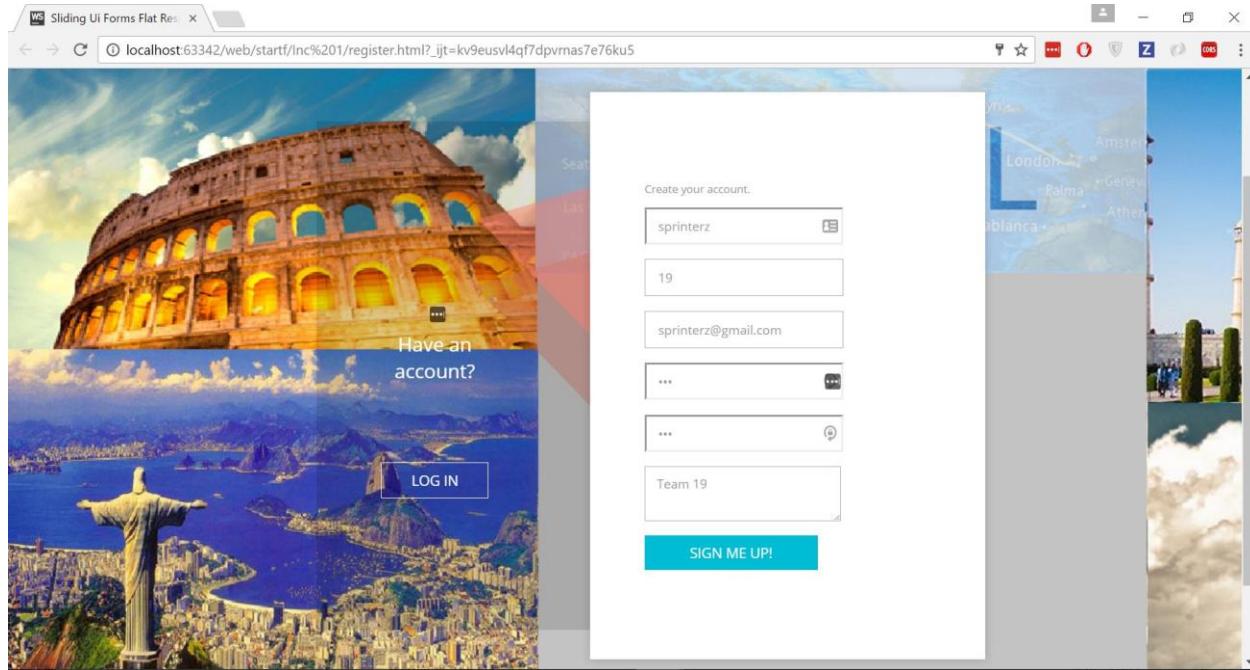
In the below image, the saved data is being entered in the login page.



On successful login, we are redirected to the home page.



Below images show other working examples.



A screenshot of the mLab MongoDB interface showing a single document in the "users" collection. The document contains the following data:

```
{  
  "email": "sudanand2u@gmail.com",  
  "firstname": "sprinterz",  
  "lastname": "19",  
  "email": "sprinterz@gmail.com",  
  "password": "ase",  
  "password": "ase"  
}
```

The interface includes a "records / page" dropdown set to 10, and a footer indicating "[1 - 7 of 7]".

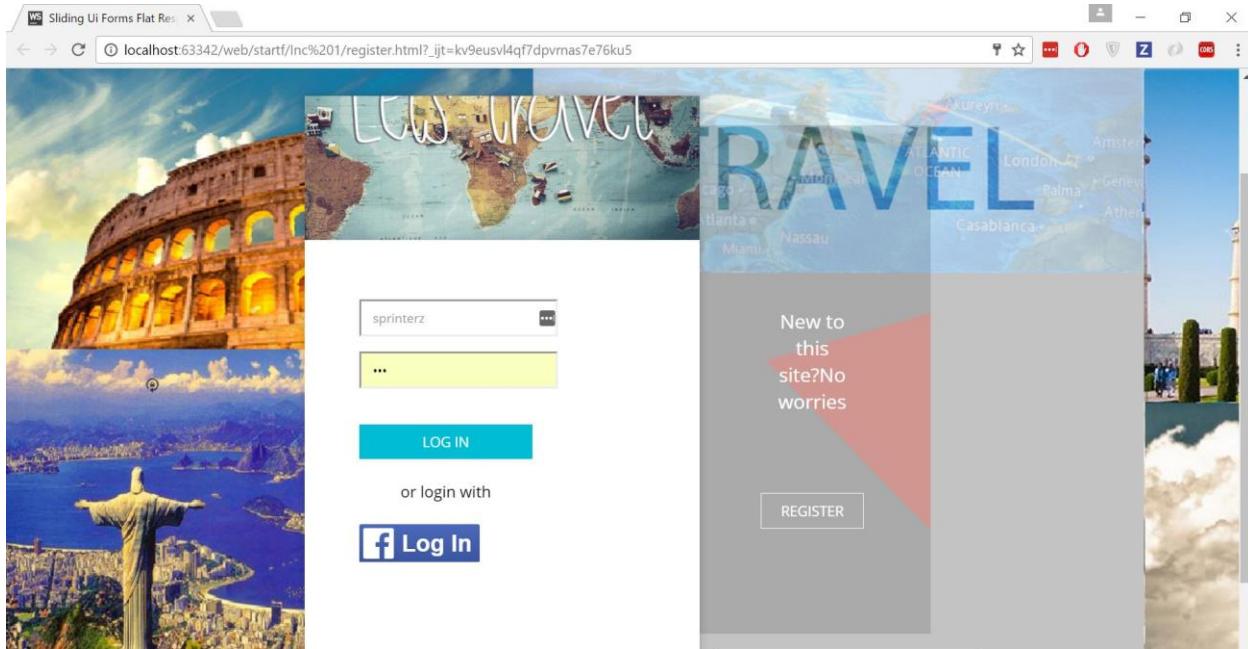
mLab logo and social media links (Twitter, Facebook, Google+).

COMPANY: About mLab, Blog.

PRODUCT: Plans & Pricing, Plan Comparison.

SUPPORT: mLab Status, Documentation, Email support@mlab.com.

CLOUDS: AWS, Azure, Google.



Project Management

Milestone	Status	Open	Closed
Increment 2	87% complete	2	14
Increment 1	75% complete	3	9
Increment 4	0% complete	0	0
Increment 3	0% complete	0	0

GitHub, Inc. [US] | https://github.com/ASE-Team19/Project#boards?milestones=Increment%202%232017-03-10&repos=82232445

This repository Search Pull requests Issues Gist ToDo Watch 0 Star 0 Fork 0

ASE-Team19 / Project Boards Reports Projects Wiki

View Repos (1/1) Labels Milestones Assignees Epics

Clear all filters

New Issues Icebox Backlog In Progress Review/QA Done Closed

Project #14 Ionic Home Page Increment 2

Project #15 Ionic Registration Increment 2

Project #20 Sequence Diagram Increment 2

Project #21 Class Diagram Increment 2

Project #24 Documentation Increment 2

Project #27 Weather Page Increment 2

Project #17 Google+ Login Increment 2

Project #22 Unit Testing Increment 2

Project #13 Web Logic Increment 2

Project #23 Deployment in Browser

GitHub, Inc. [US] | https://github.com/ASE-Team19/Project#reports?report=burndown&milestoneId=2328668

Completed Issues and Pull Requests Story points

Completed Issues and Pull Requests	Story points
Web Login Project #13	Not estimated
Facebook Login Project #16	Not estimated
Google+ Login Project #17	Not estimated
Wireframes Project #18	Not estimated
Architecture diagram Project #19	Not estimated
Sequence Diagram Project #20	Not estimated
Class Diagram Project #21	Not estimated
Unit Testing Project #22	Not estimated
Deployment in Browser Project #23	Not estimated
Documentation Project #24	Not estimated
Validations Project #25	0
Translator Project #26	0
Weather Page Project #27	0
Accommodation API Project #28	0



Unit testing

A	B	C	D	E	F
Test Case No.	Test Case Name	Pre Conditions	Input	Expected Output	Status
1	Login with null values	If user has not entered username and password	No values	Validation will be thrown to enter details	pass
2	Registration with null values	If user has not entered any details in registration page	No values	Validation will be thrown to enter details	pass
3	Registration with invalid email id	Email must contain "@" and "." with minimum 2 characters after "."	Email = gattu	Entered email must be shown wrong and should throw error	pass
4	Registration with invalid email id	Email must contain "@" and "." with minimum 2 characters after "."	Email = gattu@gmail	Entered email must be shown wrong and should throw error	pass
5	Registration with invalid email id	Email must contain "@" and "." with minimum 2 characters after "."	Email = gattu@gmail.c	Entered email must be shown wrong and should throw error	pass
6	Registration with invalid email id	Email must contain "@" and "." with minimum 2 characters after "."	Email = gattu@gmail.com	Validation is correct	pass
7	Registration with password	Password must be minimum of 6 characters	Password = samp	validation must be thrown to say password must be minimum of 6.	pass
8	Registration with wrong confirm password	Both the password and confirm password must match inorder to success	Password = sampath confirm password = sam	validation must be thrown saying password doesn't match	pass
9	Registration with all valid details	If user enters all the details correct	Name: Sampath Lname: Gattu Email: gattu@gmail.com password: sampath Confirm password: sampath	Validation is correct and page must be navigated to Home Page	pass

Implementation status report

Technology Used:

- HTML, CSS, Bootstrap, JS and Angular JS , MongoDB

Work Completed

- Design and Architecture of the Application
- Login, Registration and home page
- Login page register page validation
- User credentials stored in mongo DB
- Weather API , Translator
- Hotel search implementation

Contribution:

- Sampath Gattu – 25%
- Prathyusha Pottabathula – 25%
- Sadanand Kallakuri – 25%
- Nalla Abhiram Reddy – 25%

