Homework 8: Property search using Zillow & Bootstrap, JQuery, and Facebook Mashup

Objectives

- Become familiar with the AJAX, JSON& XML technologies.
- Use a combination of HTML, CSS, DOM, XMLHttpRequest, and XML.
- Get hands-on experience in Amazon cloud computing (AWS)
- Get hands-on experience on how to use Bootstrap and jQuery UI to enhance the user experience
- Provide an interface to perform property search from Zillow! and post details to Facebook.

Background

2.1 AJAX & JSON

AJAX (Asynchronous JavaScript + XML) incorporates several technologies:

- Standards-based presentation using XHTML and CSS;
- Dynamic display and interaction using the Document Object Model (DOM);
- Data interchange and manipulation using XML and XSLT;
- Asynchronous data retrieval using XMLHttpRequest;
- JavaScript binding everything together.

See the class slides at http://cs-server.usc.edu:45678/slides/ajax.pdf

JSON, short for JavaScript Object Notation, is a lightweight data interchange format. Its main application is in AJAX web application programming, where it serves as an alternative to the use of the XML format for data exchange between client and server. See the class slides at http://cs-server.usc.edu:45678/slides/JSON1.pdf

2.2 Zillow API

The new Zillow API Network turns member sites into mini real estate portals by offering fresh real estate content to keep people coming back.

http://www.zillow.com/howto/api/APIOverview.htm

2.3 Bootstrap Library

Bootstrap is a free collection of tools for creating responsive websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. To learn more details about Bootstrap please refer to the lecture material on Responsive Design and
http://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)

2.4 Facebook

Facebook provides developers with an API called the Facebook Platform. Facebook Connect is the next iteration of Platform, which provides a set of API's that enable Facebook members to log onto third-party websites, applications and mobile devices with their Facebook identity. While logged in, users can connect with friends via these media and post information and updates to their Facebook profile.
Below are a few links for Facebook Connect:
https://developers.facebook.com/
https://developers.facebook.com/docs/javascript

2.5 Amazon Web Services (AWS)

AWS is Amazon's implementation of cloud computing. Included in AWS is Amazon Elastic Compute Cloud (EC2), which delivers scalable, pay-as-you-go compute capacity in the cloud, and AWS Elastic Beanstalk, an even easier way to quickly deploy and manage applications in the AWS cloud. You simply upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server, PHP, and Python, Passenger for Ruby, IIS 7.5 for .NET, and Apache Tomcat for Java.
The Amazon Web Services homepage is available at:
http://aws.amazon.com/

2.6 Implementation Hints

- Get started with Bootstrap Library
  To know how to get started with Bootstrap, please refer to the link http://getbootstrap.com/getting-started/. You need to import the necessary CSS file and JS file provided by Bootstrap.

- jQuery Validation

You are required to use jQuery validation together with Bootstrap to validate user input. Please refer to the links below for more details about the implementation of this functionality.
http://jqueryvalidation.org/
http://getbootstrap.com/css/#forms-control-validation

- Bootstrap UI Components
  In this assignment you will need to use Bootstrap Form, Table, Tab and Carousel to implement all the functionality mentioned in the Description below.
  Bootstrap Form http://getbootstrap.com/css/#forms
  Bootstrap Table http://getbootstrap.com/css/#tables
  Bootstrap Tabs http://getbootstrap.com/javascript/#tabs
  Bootstrap Carousel http://getbootstrap.com/javascript/#carousel

- Using Google Map
  You are required to embed Google Map into your page. Google Map provides a JavaScript-based service. Before you import Google Map, you will have to create a Google developer account with your Google e-mail account, and set up an application for using the Google Map JavaScript service. Please refer to the link below to get started with Google Map.
  https://developers.google.com/maps/documentation/javascript/tutorial

Description

Similar to HW6, in this exercise you are asked to create a webpage that allows users to search for real estate listings using the Zillow API (www.zillow.com) and the results will be displayed in a tabular format.
The difference being, in this homework you will create a PHP file to return a JSON formatted data to the front-end and the front-end (making asynchronous AJAX calls) will parse the JSON data and show it in a nicer-looking UI (using Bootstrap JS).

A user will first open a page as shown below in Figure 1, where he/she can enter a property address in the format (street address, city, state) for looking up real estate listings in Zillow if available. The state text-field should be a drop-down list of all US states abbreviated in two letters (e.g., CA or IN). An example is shown in Figure 1.

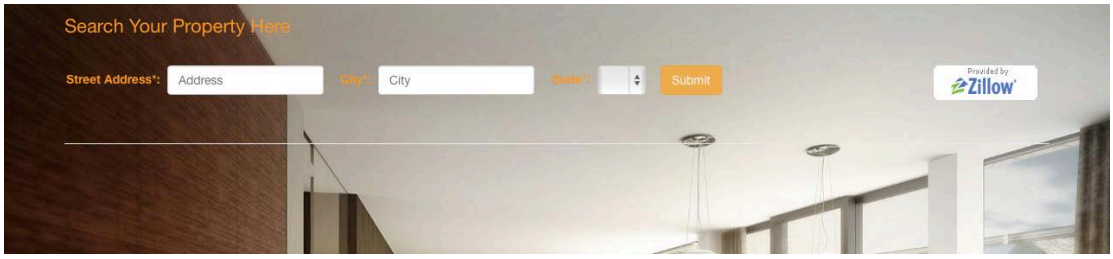


Figure 1: Initial Screen to Enter Property Address

The user should enter values for street address, city and state before clicking on the Search button. Once the user clicks on the Search button, a JavaScript program will check that the data is valid. If the user did not enter one of the data items, then a message should be shown with appropriate text requesting the user to provide the missing information. Popups are not acceptable. An example of the messages is shown in Figure 2, and an example of valid input is shown in Figure 3. The validation texts and CSS classes are all provided using the Bootstrap API.

The form uses jQuery validation to validate all required fields. 1) When a user clicks the "Search" button, if any of the required fields is empty then the corresponding field should give alert information like shown below and block the execution of the search. 2) If a user is deleting the content of a required field, when the required field is totally empty, then an appropriate message should appear denoting that this field cannot be empty. One example of this case is shown in Figure 4. For details of implementation please refer to jQuery documentation.

Figure 2: Invalid Inputs and error message

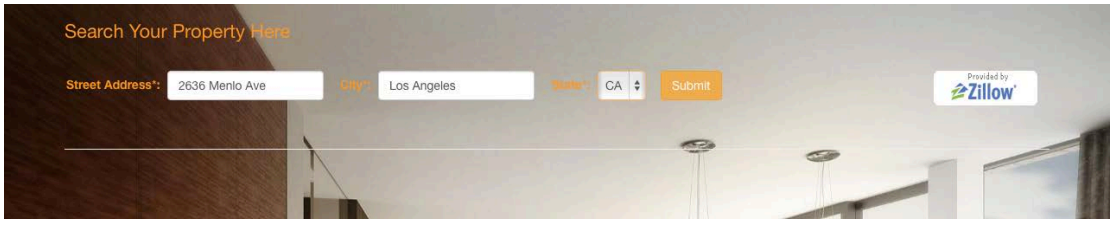


Figure 3: Valid Inputs

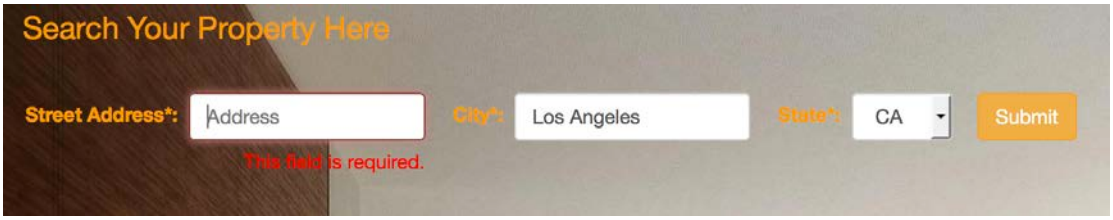


Figure 4: Clear a required input

Once the validation is successful, the JavaScript function .ajax() is executed to start an asynchronous transaction with a PHP file (script) running on your AWS server, and passing the street name, city name and state name as parameters of the transaction.

The PHP file you request is based on your HW#6, the difference is that this time the file does not need to display the data as HTML but instead will convert the XML data received from Zillow to JSON and return it to your browser and use JavaScript to extract data from it and display the results.

You should use the domain name of the AWS server you created in HW#7 to make the request. For example, if your AWS server domain is called default-environment-randomstring.elasticbeanstalk.com, and the user enters '2636 Menlo Ave', 'Los Angeles', 'CA'' as parameters, then a query of the following type will be generated:

http://default-environment-randomstring.elasticbeanstalk.com/?streetInput=2636%20Menlo%20Ave&cityInput=Los%20Angeles&stateInput=CA

Then you can send the request to the PHP file by passing the URL to $.ajax(). If you are using the URL above, you can use a GET method to request the resource. (Notice that you are required to provide this link to your homework list to let graders check whether the PHP code is running on Amazon AWS. Please refer to the grading guideline for details). Alternatively you can also use a POST method similar to the example below.

The AJAX call:

```
$.ajax({
        url: 'URL you created in HW#7',
        // this is the parameter list
        data: { streetInput: streetInput,
                cityInput: cityInput,
                stateInput:stateInput},
        type: 'POST',
        success: function(output) {
                        // parse the data here
                },
        error: function(){
                }
});
```

Similar to HW6, if the property address is incorrect or the Zillow portal does not have information for the property address, your page should show the message "No exact match found -- Verify that the given address is correct". For example when entering the address "2636 Men Ave, Los Angeles, CA" a sample output is shown in Figure 5.

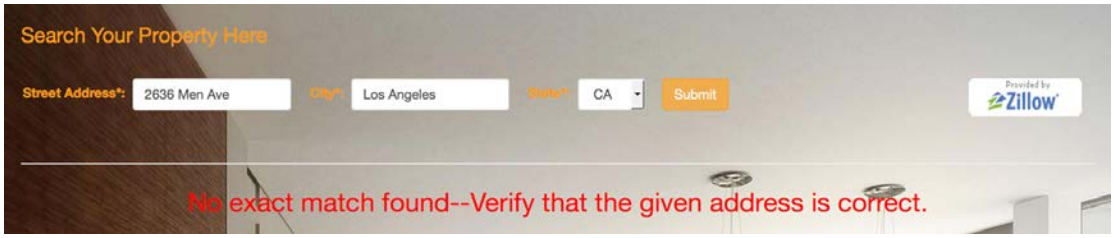


Figure 5: Example Output for a Property Address for Which Zillow Has No Information

The link below is the background image we used, but you can also go on Google Images and find appropriate images related to real estate to use as background.
http://www.neovisual.cz/files/V0779-d9.jpg

Basic Info Tab

This should show the basic real estate information table you used in HW6.
All the monetary values must be in comma format and sizes and areas followed by sq. ft. If the 30 day overall change or the All time rent change has decreased or increased, it should be clearly shown with an up or down array image as in Figure 6. The see more details, a link should redirect to the Zillow page for the same property.

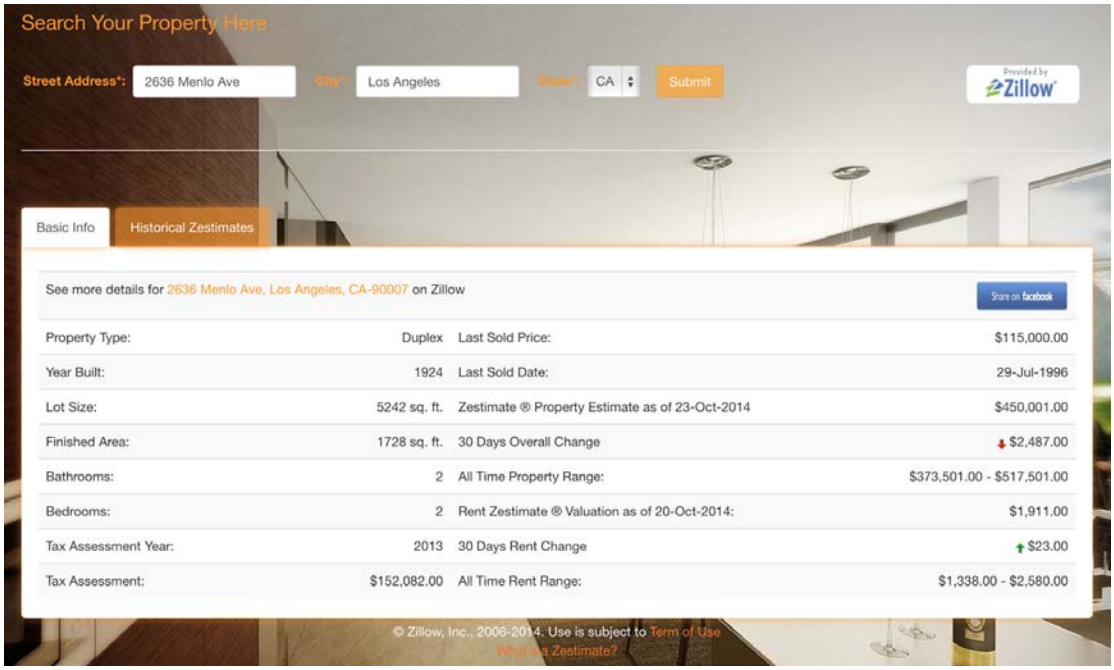


Figure 6: Basic Info Tab

The GetDeepSearchResults API would get you all the needed results.
See the API reference:
http://www.zillow.com/howto/api/GetDeepSearchResults.htm
If any of the field is unavailable, you should show "N/A" instead or an empty field. Note that if the lastSoldPrice is 0.00 and lastSoldDate is 01-Jan-1970, it

indicates that Zillow doesn't have that information and it should be shown as "N/A".

This tab also has a "Share on Facebook" button that would allow the user to share the basic information as Facebook post. When you share a Post to Facebook, it should have following format:

Figure 7: Post to Facebook

The link on the post should directly link to the Zillow page for the property details.
When the button is pressed, the web application does the following:

- Authorizes the user to Facebook (i.e. logs him/her in) using the application and user credentials if the user is not already logged in to Facebook;

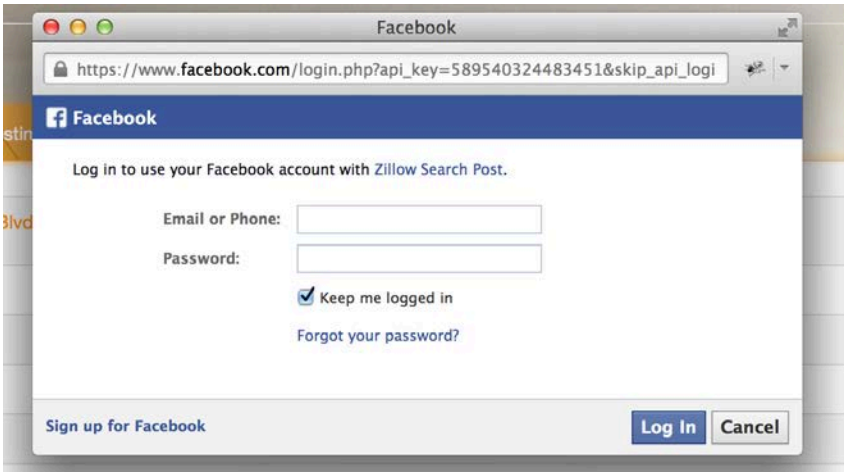


Figure 8: Facebook Login

- Posts an Update Status message to the feed.
- The above two steps can be performed using the Facebook Connect API, using the JavaScript SDK, which provides a rich set of client-side functionality for accessing Facebook's server-side API calls. It is documented at: https://developers.facebook.com/docs/reference/javascript/
- The format of the feed to be posted is as follows:

Property address (street address, city, state and zip code)

A string ("Property information from Zillow.com")
Last Sold Price: $xxx.xx 30 Days Overall Change: +/-$xx.xx

Display the appropriate values for Property address, Last Sold Price and 30 Days Overall Change. After posting, clicking on the post in your Facebook page should direct the page to the details of this property on Zillow. The Historical Zestimate chart (only one of the three) image must be displayed as a part of the post (see Figure 7 and details in Zestimate Tab section about Historical Zestimate Charts). Once the post has been published, you should show an alert box informing the user of whether the post has been published or not. For example:

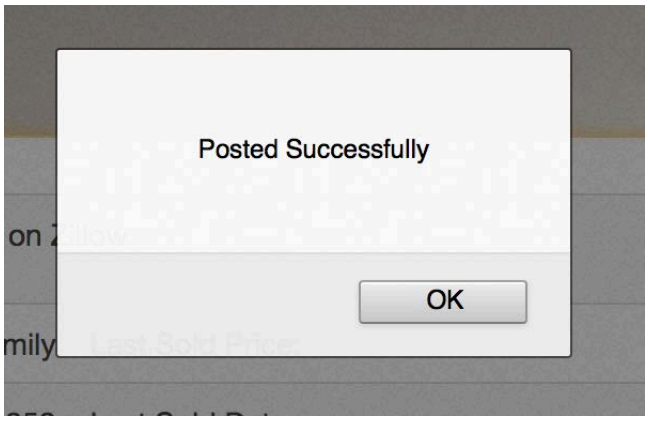


Figure 9: Successful post Alert box

Historical Zestimates Tab

The second tab will display the Historical Zestimate charts of the searched property. It needs to show three charts in a photo gallery.

Bootstrap components used:

Carousal: the charts are shown in a Bootstrap carousal. Look in the bootstrap documentation to find how carousal works.

Image switch: the images should switch automatically

Click on left: when clicking on the left button the current image would shift to the left to show the next one.

Click on right: when clicking on the right button current image would shift to the right to show the previous image.

Explanation for each image: there should be an explanation for each image at the bottom of the tab. The explanation should be:

Historical Zestimate for the past 1 year (5 years / 10 years)
Property Address

Page indicator: there should be a page indicator in the lower right corner of the tab, when the image switches, the indicator should also change value correspondingly.

Figure 10: Zestimate Chart 1 year

Figure 11: Zestimate Chart 5 years

Figure 12: Zestimate Chart 10 years

The Zestimate Charts:

GetChart API
The GetChart API generates a URL for an image file that displays historical Zestimates for a specific property. The API accepts as input the Zillow Property ID as well as a chart type: either percentage or dollar value change. Optionally, the API accepts width and height parameters that constrain the size of the image. The historical data can be for the past 1 year, 5 years or 10 years.

You should use the image of height 300 and width 600 for the 3 charts of each of 1, 5 and 10 years. The charts should be of unit-type: percent.

See the API Reference: http://www.zillow.com/howto/api/GetChart.htm

Responsive to Mobile Devices

Bootstrap provides a complete mechanism to make Web page responsive to different mobile devices. In this exercise you will get hands-on experience with responsive design using the Bootstrap Grid System.

Make the Form responsive to Mobile Devices
You are supposed to make the search form at the top of the page responsive to mobile devices. If the page is loading on a smart phone or a tablet, the form should display according to the width of the devices. One example is shown in the Figure 13 below.

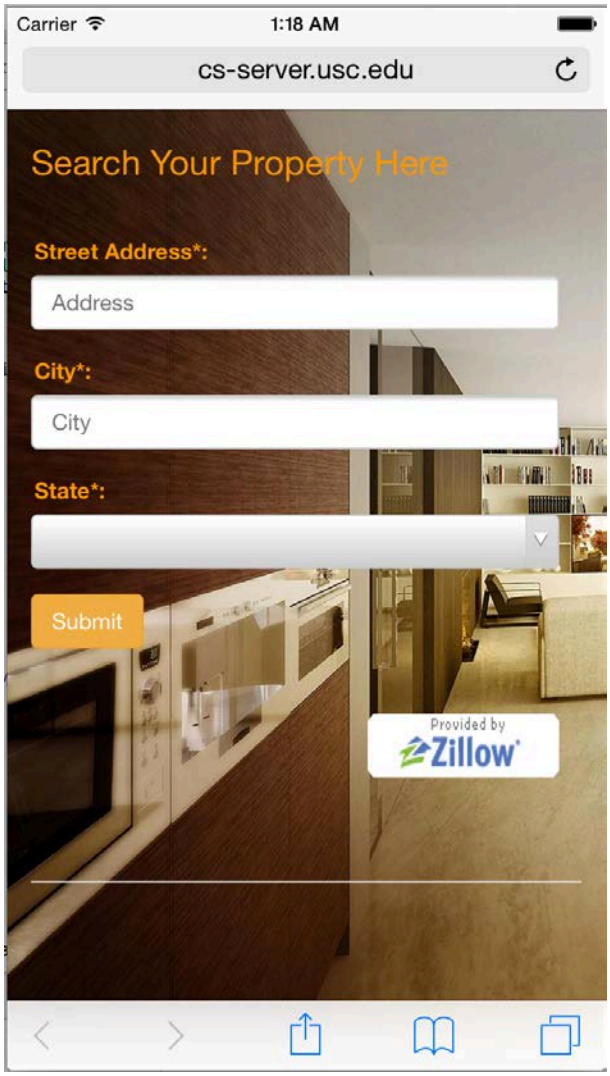
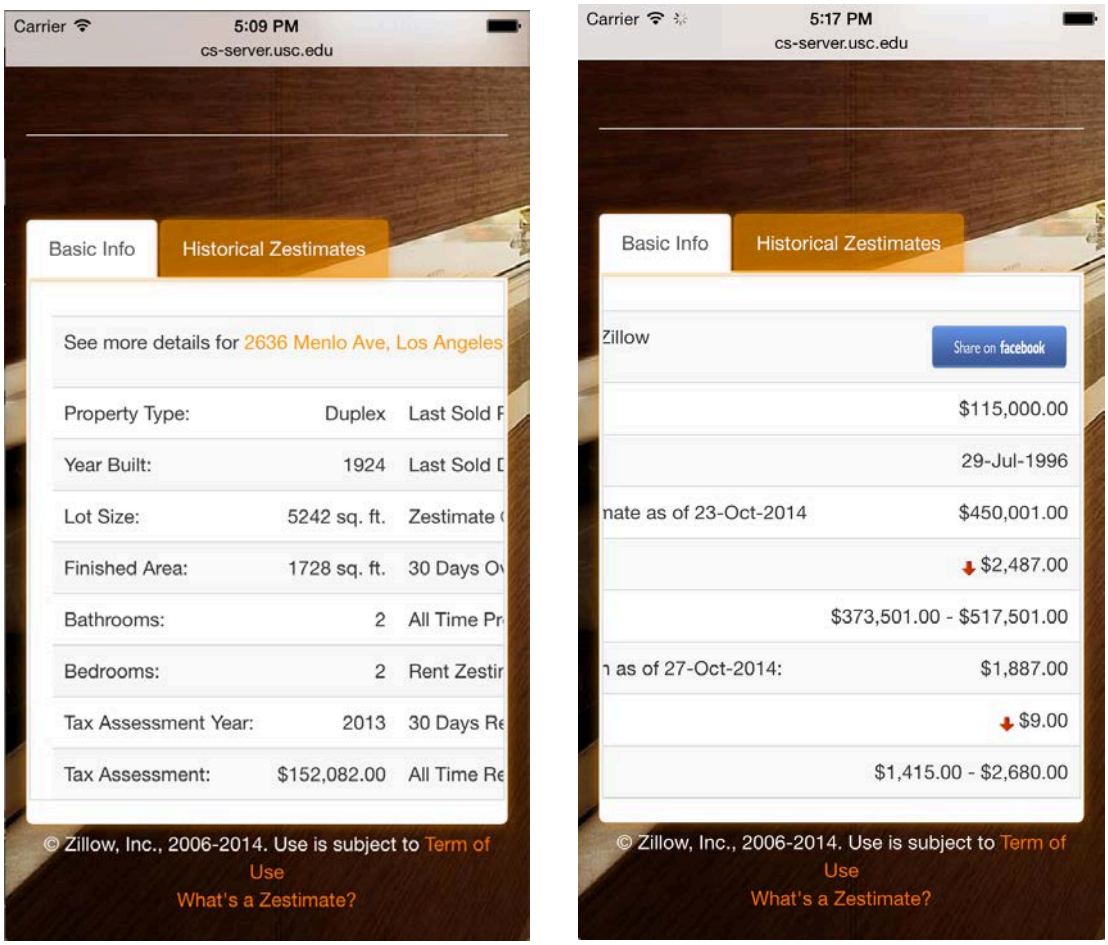


Figure 13: Search form on iPhone

Make the Table responsive to Mobile Devices

You are supposed to make the basic information table in the first tab to be responsive to mobile devices. If the page is loading on a smart phone or a tablet, the table should become scrollable; one of the examples is shown in the figures below.



Figures 14-15: scrollable basic info table on mobile devices

Make the Zestimate Charts responsive to Mobile Devices

You are supposed to make the charts displayed in the Bootstrap carousel responsive to mobile devices. If the page is loading on a smart phone or a tablet, the chart image should fill the whole carousel, rather than display the image at the center of the carousel container. An example is shown in Figure 16 below.

Figure 16: Responsive charts

Zillow Branding Requirements

To use Zillow API, you have to include Zillow branding requirements in your page based on the instructions in
http://www.zillow.com/howto/api/BrandingRequirements.htm

Zillow logo
There should be Zillow logo to the right of the search form, see Figure 1. The image is the same as the one in HW6, and it should also be a link which will redirect to zillow.com.

Zillow Disclaimer
Below the result tabs you must mention Zillow disclaimer, check Figure 6 for details.

The JSON output

Your JSON output from the PHP file should have all the necessary data and links to generate the front-end tabs and information. One way to do so is shown below. Note you may create your own JSON format data as you will be responsible for parsing it.

Sample JSON output:


```
{
  "result":{
    "homedetails":"http:\/\/www.zillow.com\/homedetails\/1248-W-Adams-Blvd-APT-101-Los-Angeles-CA-90007\/2108691793_zpid\/",
    "street":"1248 W Adams Blvd APT 101",
    "city":"Los Angeles",
    "state":"CA",
    "zipcode":"90007",
    "latitude":"34.032382",
    "longitude":"-118.287811",
    "useCode":"MultiFamily2To4",
    "lastSoldPrice":"0.00",
    "yearBuilt":"1922",
    "lastSoldDate":"01-Jan-1970",
    "lotSizeSqFt":"",
    "estimateLastUpdate":"16-Oct-2014",
    "estimateAmount":"460,460.00",
    "finishedSqFt":"",
    "estimateValueChangeSign":"+",
```

```
    "imgn":"http:\/\/www-
scf.usc.edu\/~csci571\/2014Spring\/hw6\/down_r.gif",
    "imgp":"http:\/\/www-
scf.usc.edu\/~csci571\/2014Spring\/hw6\/up_g.gif",
    "estimateValueChange":"712.00",
    "bathrooms":"2",
    "estimateValuationRangeLow":"423,623.00",
    "estimateValuationRangeHigh":"506,506.00",
    "bedrooms":"2",
    "restimateLastUpdate":"13-Oct-2014",
    "restimateAmount":"2,201.00",
    "taxAssessmentYear":"",
    "restimateValueChangeSign":"-",
    "restimateValueChange":"67.00",
    "taxAssessment":"0.00",
    "restimateValuationRangeLow":"1,651.00",
    "restimateValuationRangeHigh":"2,685.00"
  },
  "images":{
    "photoGallery":"http:\/\/www.zillow.com\/homedetails\/1248-W-Adams-
Blvd-APT-101-Los-Angeles-CA-
90007\/2108691793_zpid\/#image=lightbox%3Dtrue",
    "image1":"http:\/\/photos3.zillowstatic.com\/p_d\/ISx3pdhz6rfvls0000000
000.jpg"
  },
  "chart":{
    "1year":{
      "url":"http:\/\/www.zillow.com\/app?chartDuration=1year&chartType=pa
rtner&height=300&page=webservice%2FGetChart&service=chart&showPercent
=true&width=600&zpid=2108691793"
    },
    "5years":{
      "url":"http:\/\/www.zillow.com\/app?chartDuration=5years&chartType=p
artner&height=300&page=webservice%2FGetChart&service=chart&showPerce
nt=true&width=600&zpid=2108691793"
    },
    "10years":{
      "url":"http:\/\/www.zillow.com\/app?chartDuration=10years&chartType=
partner&height=300&page=webservice%2FGetChart&service=chart&showPerc
ent=true&width=600&zpid=2108691793"
    }
  }
}
```

Files to Submit

On your course homework page, you should update the HW8 link to refer your new initial web page. Also, Submit your files electronically to the csci571 account so that they can be graded and compared to all other students' code via the MOSS code comparison tool.

IMPORTANT:

All discussion and explanation in Piazza related to this homework become part of the homework description. So please review all Piazza threads before finishing the assignment, especially the Summary, which will be posted about a week before the exercise deadline.

In your Facebook application settings, you should go to the "Status & Review" section and choose "Yes" for the question "Do you want to make this app and all its live features available to the general public?" as shown in the Figure 17. If you answer "YES", anyone will be able to log in through her/his Facebook account and post via your web interface. Otherwise, the developer will be the only person who is able to use the Facebook functionality in the web page. If graders try to test the functionality of Facebook button and they are not able to log in you will lose **all** points related to the Facebook component in your homework grade.

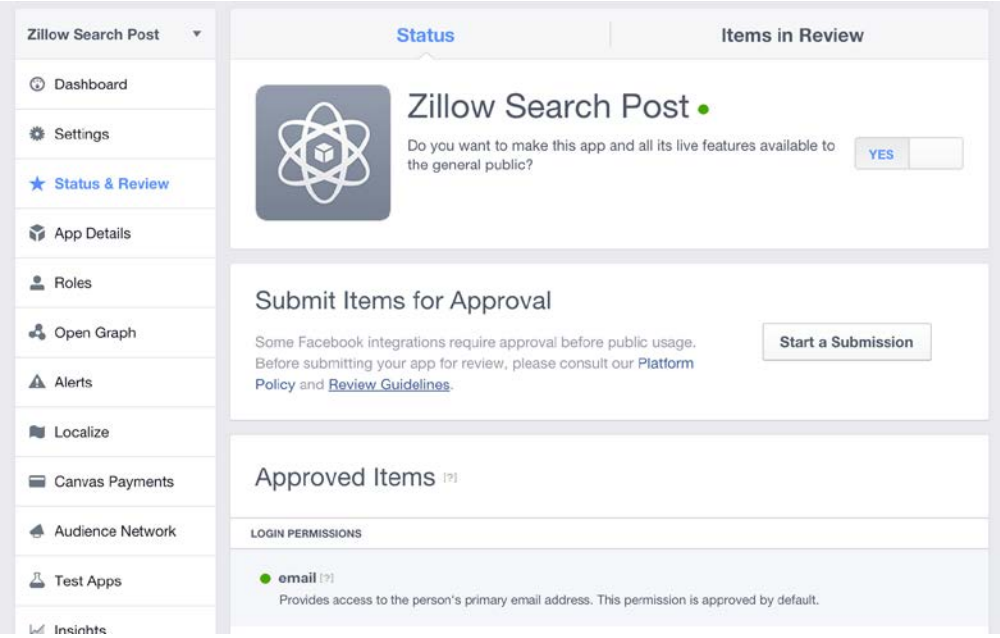


Figure 17 Status & Review of Facebook Application