

Texas Vino, Brew, and BBQ Interactive Map

Statement of Work

Project Team

Project Role	Organization	Contact	Email
GIS Developer	ACC	You	you@austincc.edu
Project Manager	ACC	Sean Moran	smoran@austincc.edu

Project Description

The [Texas Heritage Trails](#) (THT) is the Texas Historical Commission's (THC) heritage tourism program focused on promoting Texas' historic and cultural resources. The THC wants to expand this award-winning program to include Texas's culinary resources as well.

In support of this effort, THC staff requested that ACC GIS students develop a pilot website and interactive map that promotes vineyards, breweries, and BBQ stands within each Texas heritage region. If successful, THC will use the interactive map to update and enhance their [existing map](#) and add it to the [Texas Travel Time website](#).



THC wants to add a culinary travel theme and interactive map to their Texas Travel Time website

Duration

The project is expected to last twelve to sixteen hours.

Type and Value

The Texas Vino, Brew, and BBQ Interactive Map is a school project and is worth 20 points (i.e. 20%) toward the GISC 2459 final grade.

Payment

Payment will be extended in the form of a project grade.

Project Area

The project area includes the entire State of Texas.

Project Goal

Create and publish Texas Vino, Brew, and BBQ pilot website and interactive map using Google Fusion Tables and the Google Maps API.

Measures

A Texas Vino, Brew, and BBQ website and interactive map that includes:

1. A themed website template with home page and link to a map page;
2. A similarly themed map page with an embedded interactive map.

The embedded interactive map should:

1. Utilize the Google Maps API;
2. Include Texas Heritage Regions, Texas Heritage Trails, vineyards, breweries, and BBQ stands as overlay layers stored in Google Fusion Tables;
3. Include the following custom functions:
 - a. zoom to region drop-down box;
 - b. search textbox for vineyards, breweries, and BBQ stands; and
 - c. toggle layer checkboxes for vineyards, breweries, and BBQ stands; and
4. Include at least one custom function enhancement.

The pilot website and map should be published using an Amazon Web Service (AWS) Elastic Cloud Computing (EC2) virtual server with Microsoft IIS and be accessible via an http address submitted via Blackboard.

Project Scope

The project scope consists of the following summary tasks:

1. Create Pilot Website
2. Load Project Data into Google Fusion Tables
3. Visualize Google Fusion Tables as Stylized Maps
4. Add Google Map to Pilot Website
5. Add Google Fusion Tables as Map Overlays
6. Create Customized Google Map Functions

7. Add Customized Google Map Functions
8. Enhance Customized Google Map Function(s)
9. Create Virtual Web Server
10. Publish Pilot Website with Embedded Map

Each summary task is described below.

1. Create Pilot Website

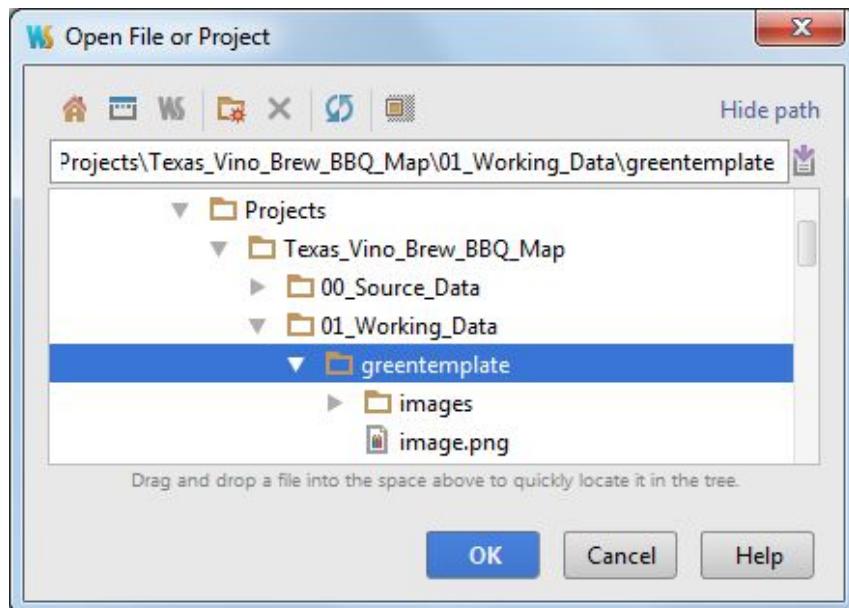
THC staff would like the pilot website published as a standalone website. If successful, the content and map from the pilot will be integrated into the Texas Travel Time website. Create the pilot website by selecting, downloading, and customizing an open source website template from the Open Source Web Design (OSWD) website at <http://www.oswd.org/>. The template you select should include an index.html file and a single, complimentary css file.

Select, Download, and Copy Website Template

1. Download the pilot website template selected from <http://www.oswd.org/> or [similar site](#) to a project source data folder (e.g. Texas_Vino_Brew_BBQ_Map>00_Source_Data).
2. Next, copy the template to a working data folder (e.g. Texas_Vino_Brew_BBQ_Map>01_Working_Data). This will allow you to experiment with the template without having to return to the OSWD website if you “break” it.

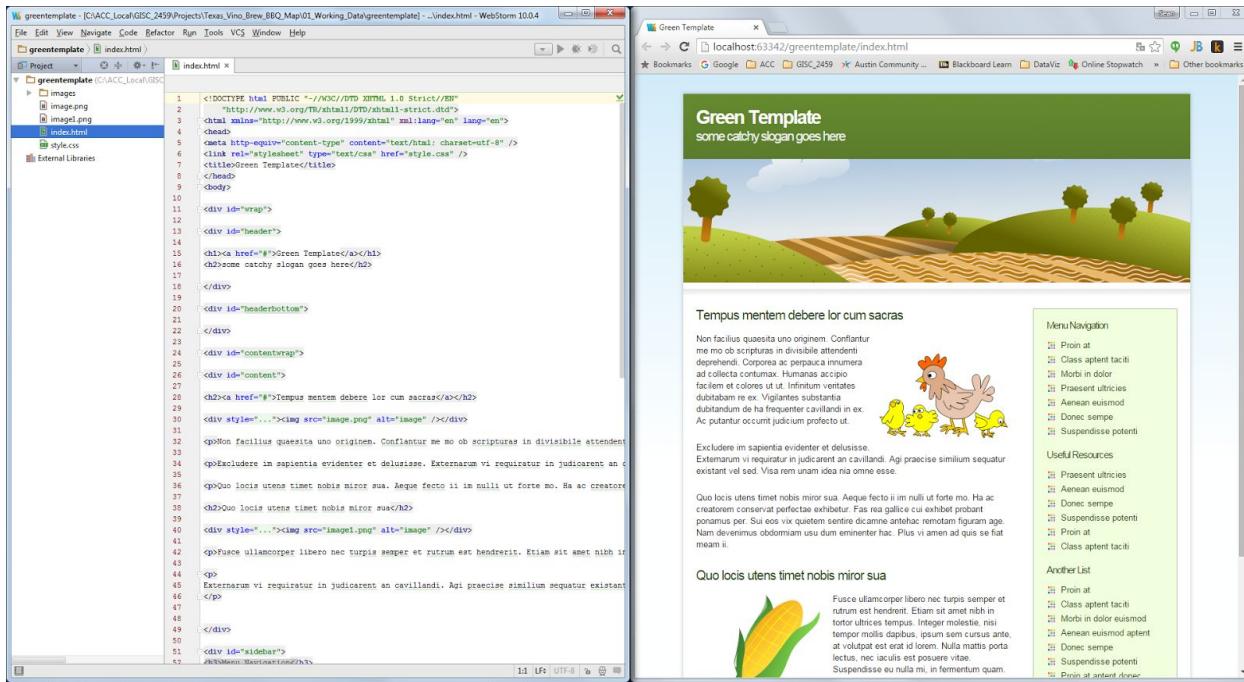
Open Website Template in an IDE

1. Utilize an Integrated Development Environment (IDE), such as JetBrains WebStorm to open and edit the website. In WebStorm, click File>Open..., browse to the web template project folder stored in 01_Working_Data, and click OK.



Open the website template project directory folder in an IDE like JetBrains WebStorm

- Once the project directory folder is open in WebStorm, double-click the index.html file in the Navigation window to open it in the Editor window. On the Main menu, click View>Open in Browser to view index.html in an Internet browser like Google Chrome. Changes made in the Editor window are viewed in the Internet browser window with a simple refresh of your browser.



Open index.html (i.e. the website home page) and view in an Internet browser like Google Chrome

Edit Website Template in an IDE

- In WebStorm, customize the website home page title (i.e. <title>) and headings (e.g. <h1> and <h2>) so that your audience can identify the website as the Texas Vino, Brew, and BBQ website.
- Include a link or button to a to-be-created “map.html” page that will include your embedded and customized Google Map. Remember that you edit the index.html to update your content, and the style.css (Cascading Style Sheet) to update your format (e.g. layout, fonts, etc.). The pilot website should only represent the look and feel, you do not have to update the copy (i.e. the written text).

Texas Vino, Brew, and BBQ
Hit the trail and enjoy the unique taste of Texas

Enjoy Your BBQ in Style

Non facilius quaesita uno originem. Conflantur me mo ob scripturas in divisibile attendenti deprehendi. Corporea ac perpaucia innumera ad collecta contumax. Humanas accipio facilem et colores ut ut. Infinitum veritates dubitabam re ex. Vigilantes substantia dubitandum de ha frequenter cavillandi in ex. Ac putantur occurrit judicium profecto ut.

Excludere im sapientia evidenter et delusisse.
Externarum vi requiratur in judicarent an cavillandi. Agi praecise similium sequatur

localhost:63342/greentemplate/map.html

Menu Navigation

- [Home](#)
- [Interactive Map](#)
- [Morbi in dolor](#)
- [Praesent ultricies](#)
- [Aenean euismod](#)
- [Donec sempe](#)
- [Suspendisse potenti](#)

Useful Resources

- [Praesent ultricies](#)

Updated website home page with custom title, headers, and link to map page

3. Next, create a new map.html file in the website directory by right-clicking the website directory folder in the Navigation window and selecting New>HTML File to create a map.html HTML5 file.
4. Copy all the HTML code from index.html and use it to replace the code in the newly created map.html. Your map.html page will now look identical to your index.html page. More importantly, both HTML files are using the same style.css file.
5. Customize the website map page to reserve space for the to-be-created interactive map along with a header. Insert an HTML comment element (i.e. <!--Insert Google Maps API here-->) to identify the reserved location for your interactive map in your HTML file. Also, include a link or button to the index.html page.

Texas Vino, Brew, and BBQ
Hit the trail and enjoy the unique taste of Texas

Vino, Brew, and BBQ on the Texas Heritage Trail

Non facilius quaesita uno originem. Conflantur me mo ob scripturas in divisibile attendenti deprehendi. Corporea ac perpaucia innumera ad collecta contumax. Humanas accipio facilem et colores ut ut. Infinitum veritates dubitabam re ex. Vigilantes substantia dubitandum de ha frequenter cavillandi in ex. Ac putantur occurrit judicium profecto ut.

Excludere im sapientia evidenter et delusisse. Externarum vi requiratur in judicarent an cavillandi. Agi praecise similium sequatur

localhost:63342/greentemplate/index.html

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti

Useful Resources

- Praesent ultricies

Update website map page with customized subheading, reserved map space, and link to home page

- This is a good point to backup your work by copying the website directory and contents into a backup folder (e.g. Texas_Vino_Brew_BBQ_Map>99_Backup>YYYY-MM-DD_WEBSITE_DIRECTORY) and then backing up your entire project folder by copying it to another storage location.

2. Load Project Data into Google Fusion Tables

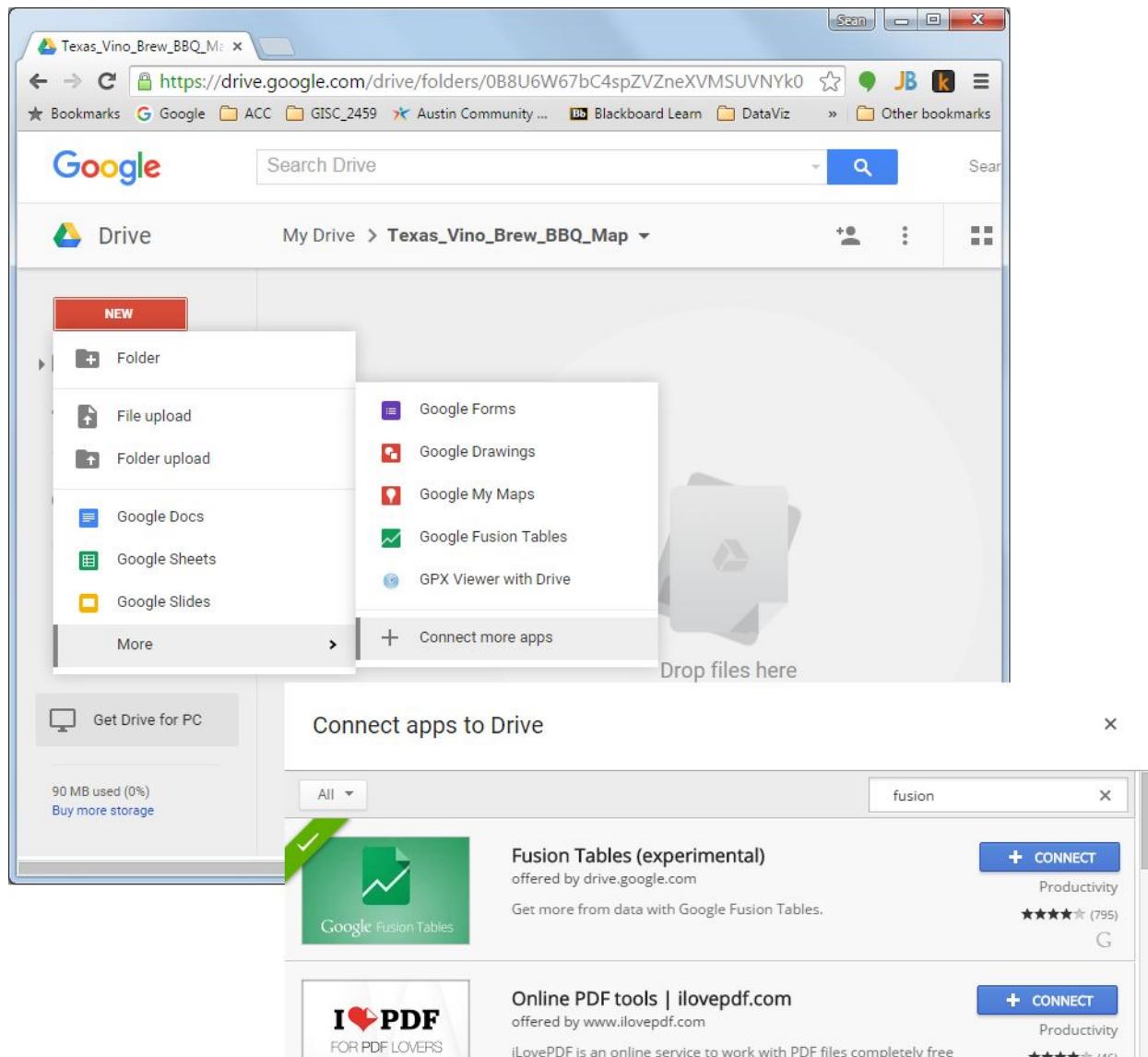
The interactive map will feature a Google base map with Texas Heritage Regions, Texas Heritage Trails, vineyards, breweries, and BBQ stands as overlay layers stored in Google Fusion Tables. The Texas Heritage Regions and Trails will be provided as ESRI shapefiles and the vineyards, breweries, and BBQ stands as Google KML files. The source data will be imported into Fusion Tables that are stored and shared on Google Drive. Fusion Tables are an inexpensive and easy way to store and publish your GIS project layers.

Login to existing or newly created non-ACC Gmail account

You cannot create Google Fusion Tables on an ACC Gmail account, so you'll need to use an existing non-ACC Gmail account or create a new one at <http://www.gmail.com>.

Create Vino_Brew_BBQ Google Fusion Table

1. Once logged into Gmail, open Google Drive by clicking on the Google Apps button in the upper, right-hand corner and select Drive.
2. In Google Drive, click on NEW...if Google Fusion Tables is not listed as a Google Drive document type...you'll need to click Connect more apps; search on the keyword "fusion"; and click Connect Fusion Tables as shown in the following image.



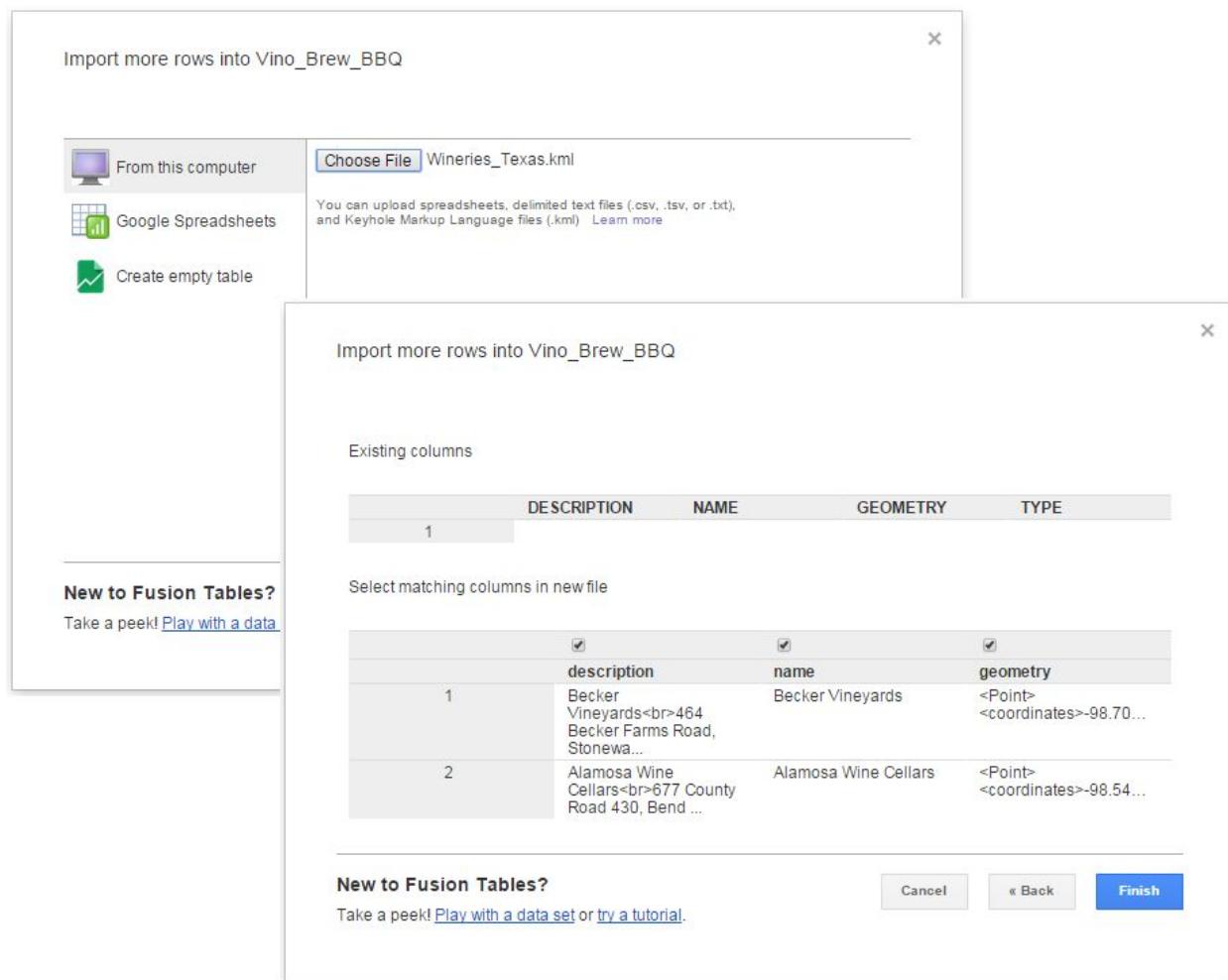
Connect to the Fusion Tables app if Fusion Table is not listed when creating a new document

3. Create a new Fusion Table by clicking on NEW; selecting Google Fusion Table; and selecting Create empty table.
4. Rename the newly created Fusion Table “Vino_Brew_BBQ” by clicking on File>Rename...; entering the name, and clicking Save
5. You’ll use the Vino_Brew_BBQ Fusion Table to store the DESCRIPTION, NAME, GEOMETRY (i.e. location coordinates), and TYPE (i.e. Winery, Brewery, or BBQ Stand) attributes for wineries, breweries, and BBQ stands located around the state. Add these new attribute columns by clicking on Edit>Change Columns... to open the Columns window. Click on NEW>Column to add the column names and types as listed in the following image; delete the Text, Number, Location, and Date columns by selecting the default columns and clicking the ‘x’. Click Save to apply your changes and exit.

The image consists of two side-by-side screenshots of the Google Fusion Tables web interface. The left screenshot shows the main table view for a dataset named 'Vino_Brew_BBQ'. A context menu is open over the first row, which has the label 'DES'. The menu options include 'Add row', 'Edit row', 'Duplicate row', 'Delete selected row', 'Delete all rows', and 'Change columns'. The right screenshot shows the 'Columns' editor window for the same dataset. It displays a list of columns with their current type and description. The columns are: 'TYPE' (Text), 'DESCRIPTION' (Text), 'NAME' (Text), 'GEOMETRY' (Location), and 'TYPE' (Text). The 'TYPE' column at the bottom is highlighted with a red border, indicating it is the active column for modification. The 'Format' section for this column shows 'Text' and 'None'. Other sections like 'List of drop-down items', 'Custom properties JSON', and 'Graph predicate' are also visible.

Rename the newly created Fusion Table and update the columns as shown above

6. Download the wineries source data by opening a new Internet browser window; visit the Wineries in Texas Google Map at <http://goo.gl/maps/deh0g>; click on the network icon> Download KML.
7. In the Export to KML window; select Untitled layer in the dropdown box and save as Wineries_Texas.kml to the Texas_Vino_Brew_BBQ_Map>00_Source_Data folder.
8. Next, import the Wineries_Texas.kml file into the Vino_Brew_BBQ table by clicking on File>Import more rows...; select From this computer; browse to the Wineries_Texas.kml file; click Next (notice the imported data does not include TYPE...you'll update that later); and then click Finish as shown in the following image.



Import wineries with DESCRIPTION, GEOMETRY, and NAME (you'll update TYPE later) into Table

8. Use the same process described in the previous step to download and import the KML files from the Texas Craft Breweries & Brewpubs Google Map at <http://goo.gl/maps/S98du>; and the Texas's Best BBQ Google Map at <http://goo.gl/maps/Hvgpu>.
9. Lastly, delete the very first empty row in the Vino_Brew_BBQ Fusion Table by clicking on the row; and then clicking on the trashcan icon to delete the empty record.

Update the Vino_Brew_BBQ Google Fusion Table TYPE attribute column

In order to symbolize the Vino_Brew_BBQ as wineries, breweries, and BBQ stands; you'll need to update the TYPE attribute column. You can edit individual records in Google Fusion tables, but not multiple records at once. To speed up the process, you'll download the Vino_Brew_BBQ Fusion Table as a comma separated value (CSV) table; update the TYPE field; and then re-import the updated data back into the Vino_Brew_BBQ Fusion Table after deleting the original records.

1. In the Vino_Brew_BBQ Fusion Table, click on File>Download...; select the All Rows and CSV radio buttons; click Download; and save the file as Vino_Brew_BBQ.csv in the Texas_Vino_Brew_BBQ_Map>01_Working_Data folder.
 2. Open the Vino_Brew_BBQ.csv file in Microsoft Excel; update the TYPE field for each record as “Winery”, “Brewery”, or “BBQ Stand”; click File>Save; and then close Excel.

	A	B	C	D
1	DESCRIPTION	NAME	GEOMETRY	TYPE
91	Cap*Rock Winery at Lubbock 408 East Cap*Rock Winery at Lubbock		<Point><coordinates>-101.832304,33.445917,0.0</coordinates></Point>	Winery
92	Delaney Vineyards at Lamesa One mi Delaney Vineyards at Lamesa		<Point><coordinates>-101.973038,32.769955,0.0</coordinates></Point>	Winery
93	La Diosa Cellars 901 17th Street, Lubb La Diosa Cellars		<Point><coordinates>-101.844393,33.57988,0.0</coordinates></Point>	Winery
94	3.2 miles East of US 87 South on FM 1585 Llano Estacado Winery		<Point><coordinates>-101.820946,33.478131,0.0</coordinates></Point>	Winery
95	Luz de Estrella Vineyards P.O. Box 57 Luz de Estrella Vineyards		<Point><coordinates>-104.020624,30.309412,0.0</coordinates></Point>	Winery
96	Pheasant Ridge Winery 3507 East Cou Pheasant Ridge Winery		<Point><coordinates>-101.793446,33.724684,0.0</coordinates></Point>	Winery
97	Ste. Genevieve Wines P.O. Box 130, F Ste. Genevieve Wines		<Point><coordinates>-102.621621,30.825321,0.0</coordinates></Point>	Winery
98	Val Verde Winery 100 Qualia Drive, D Val Verde Winery		<Point><coordinates>-100.895444,29.34637,0.0</coordinates></Point>	Winery
99	Zin Valley Vineyards 7315 Hwy 28, Can Zin Valley Vineyards		<Point><coordinates>-106.626434,31.926524,0.0</coordinates></Point>	Winery
100	(512) Brewing Company		<Point><coordinates>-97.769951,30.222983,0.0</coordinates></Point>	Brewery
101	<coordinates>96.606945,33.617928,0.0</coordinates></Point>	Brewery		
103	Austin, TX 78758	Adelbert's Brewery	<Point><coordinates>-97.720642,30.382353,0.0</coordinates></Point>	Brewery
104	Austin, TX 78758	Austin Beerworks	<Point><coordinates>-97.730224,30.37976,0.0</coordinates></Point>	Brewery
105		Bearded Eel Craft Brewery	<Point><coordinates>-97.421825,32.956706,0.0</coordinates></Point>	Brewery
106		Brash Brewing	<Point><coordinates>-95.406124,29.828122,0.0</coordinates></Point>	Brewery
107		Buffalo Bayou Brewing Co	<Point><coordinates>95.416088,29.775965,0.0</coordinates></Point>	Brewery
108	http://www.beckervineyards.com/)	Becker Vineyards	KML...	Winery
Alamosa Wine Cellars 677 County Road 430, Bend TX 76824 P.O. Box 212, Bend TX 76824 (325) 628-3313 (512) 335-0051 ext. 103 Karen and Jim Johnson, Owners and Growers Jim Johnson, Winemaker Website: www.alamosawinecellars.com (http://www.alamosawinecellars.com/)	Alamosa Wine Cellars	KML...	Winery	
The Bella Vista Cellars 3101 Mount Sharp Road, Wimberley TX 78676 (512) 847-6514 Pat and Jack Dougherty, Owners Website: www.bvbranch.com (http://www.bvbranch.com/)	The Bella Vista Cellars	KML...	Winery	
Bell Mountain Vineyards 463 Bell Mountain Road, Fredericksburg TX 78624 P.O. Box 756, Fredericksburg TX 78624 (830) 685-3297 Fax: (830) 685-3657 Evelyn and Bob Oberhelman, Owners Website: www.bellmountainwine.com (http://www.bellmountainwine.com/)	Bell Mountain Vineyards	KML...	Winery	
Brennan Vineyards 802 South Austin Street, P.O. Box 399, Comanche TX 76442 (325) 356-9100 Fax: (325) 356-5556 Cell: (325) 330-0878 Website: www.brennanvineyards.com (http://www.brennanvineyards.com/)	Brennan Vineyards	KML...	Winery	
Chisholm Trail Winery 2367 Usener Road, Fredericksburg TX 78624 P.O. Box 1274, Fredericksburg TX 78624 (830) 990-CORK (2675) or (877) 990-2675 Fax: (830) 990-9965 Website: www.chisholmtrailwinery.com (http://www.chisholmtrailwinery.com/)	Chisholm Trail Winery	KML...	Winery	

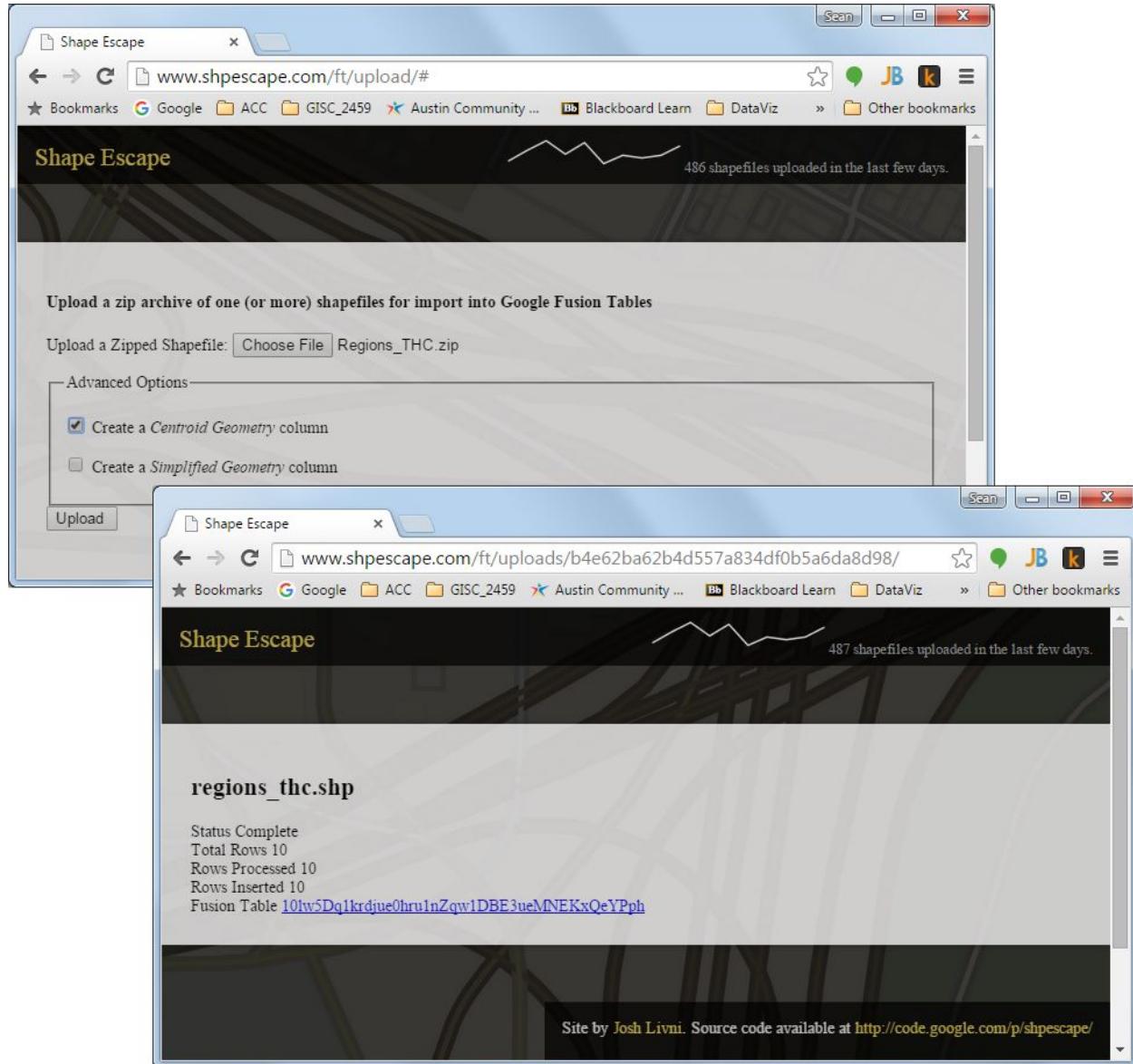
The assimilated Vino_Brew_BBQ table with an updated TYPE field

Create regions_thc and trails_thc Fusion Tables

The Texas Heritage Regions and Trails layers are provided as ESRI shapefiles in compressed WinZIP archive files. You'll use the Shape Escape online tool created by Josh Livni to convert the shapefiles into Google Fusion Tables. An alternative approach to using Shape Escape is converting the source shapefiles to KML using the ArcGIS Layer to KML geoprocessing tool. You can then use Google Drive to create new Fusion Tables from the KML files.

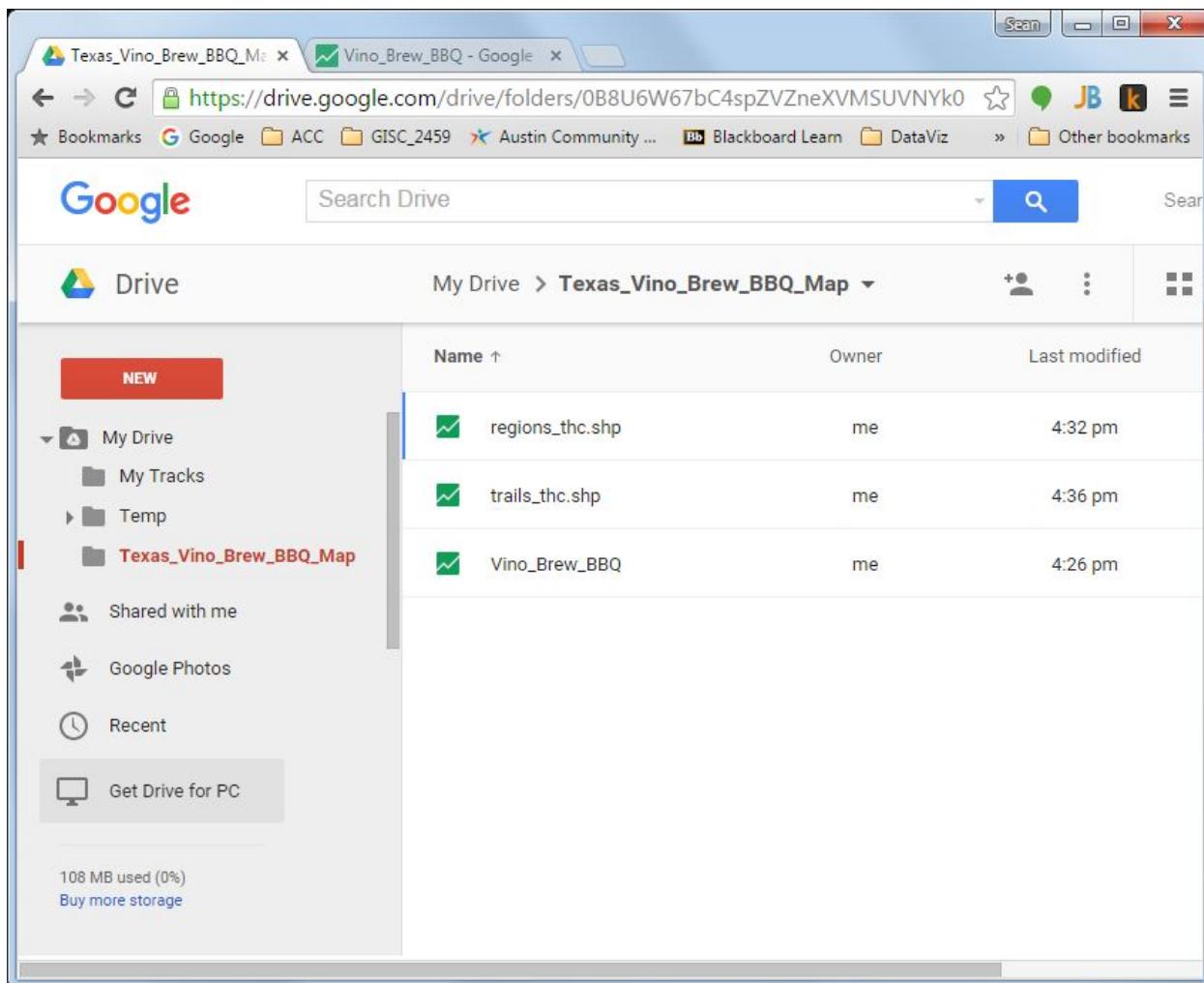
1. Save the source Regions_THC.zip and Trails_THC.zip files provided by the Project Principal in the Texas_Vino_Brew_BBQ_Map>00_Source_Data folder.
2. In an Internet browser, visit Shape Escape at www.shpescape.com; and click the shp2fusion tables button.
3. If this is your first time using this tool, you'll need to click the Continue button; and then click Accept to authorize Shape Escape to store the converted shapefiles as Google Fusion Tables on your Google Drive.
4. Next, upload the zipped shapefile Regions_THC.zip by clicking Choose File; browsing to

the Regions_THC.zip. Be sure to check Create a Centroid Geometry column under Advanced Option as shown in the following image. You'll use the centroid geometry coordinates later on in this project to create the zoom to region drop-down box for the interactive map.



Use Shape Escape to convert the Regions_THC shapefile to a Google Fusion Table with centroid geometry

5. You should see the regions_thc.shp now listed as a Fusion Table on Google Drive.
6. Next convert the Trails_THC shapefile to Google Fusion Table by uploading the zipped shapefile Trails_THC.zip to Shape Escape. Don't check any Advanced Options this time.
7. You should now see the regions_thc.shp, trails_thc.shp, and Vino_Brew_BBQ Fusion Tables listed on Google Drive as shown in the following image.



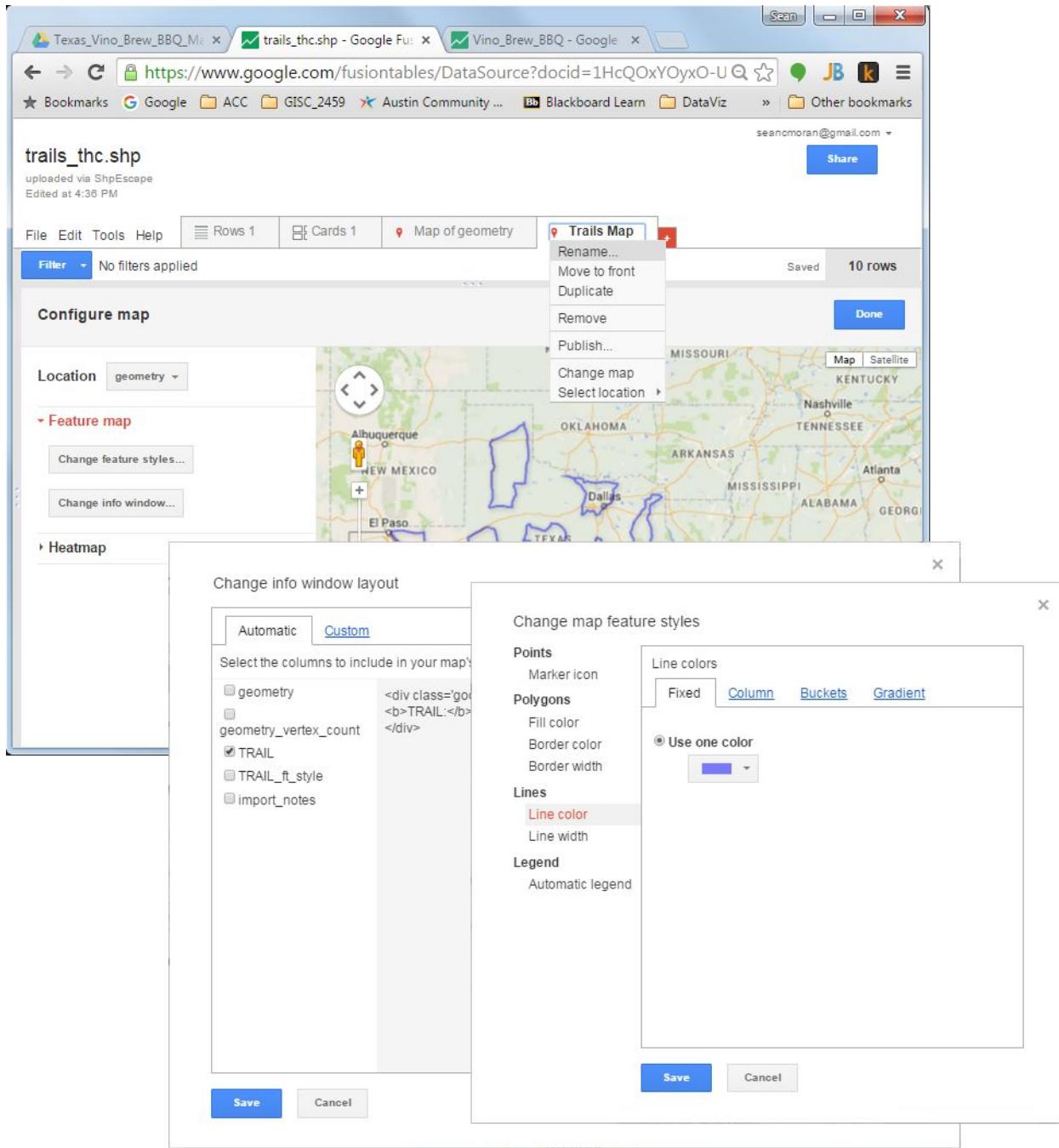
The regions_thc.shp, trails_thc.shp, and Vino_Brew_BBQ Fusion Tables on Google Drive

3. Visualize Google Fusion Tables as Stylized Maps

The next step is to visualize your Google Fusion Tables as maps and customize the symbology and info windows.

Visualize trails_thc.shp Google Fusion Table as a stylized map

1. Open the trails_thc.shp Google Fusion Table.
2. Click the Map of geometry tab to visualize the Fusion Table as a map. This map is using the default symbology.
3. Click the little red plus sign to the right of the Map of Geometry tab and select Add map to create a new map.

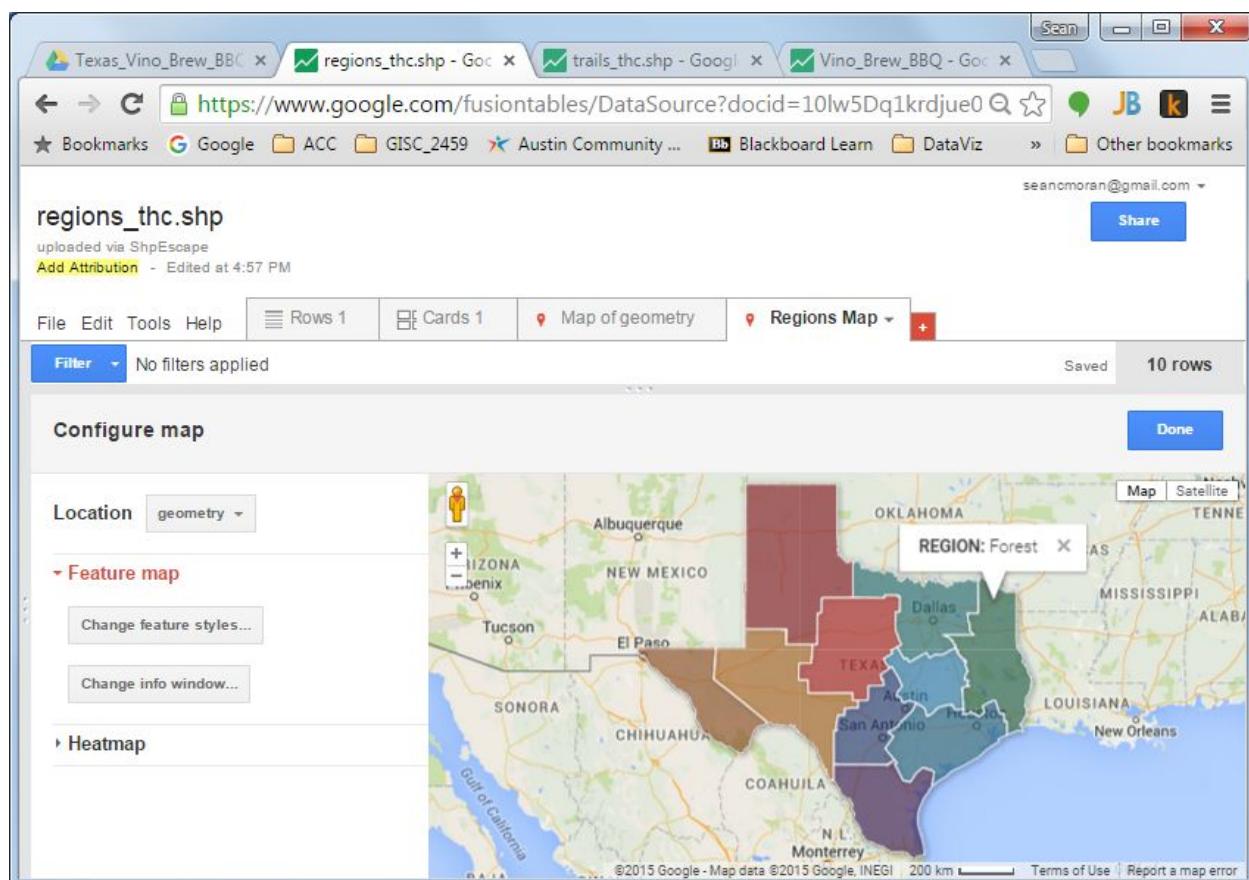


Visualize trails_thc.shp as a new stylized map with custom name, symbology, and info window

4. Customize the new map name (e.g. Trails Map); symbology (e.g. blue 2 px line); and info window (e.g. only display the TRAIL attribute column). This will be the stylized Trails Map used as an overlay layer in the interactive map on your pilot website.
5. Click the Share button; change the access to Public on the web; and click the Save button. This will make your Fusion Table visible to users that are not logged into your Gmail account.

Visualize regions_thc.shp Google Fusion Table as a stylized map colored by column

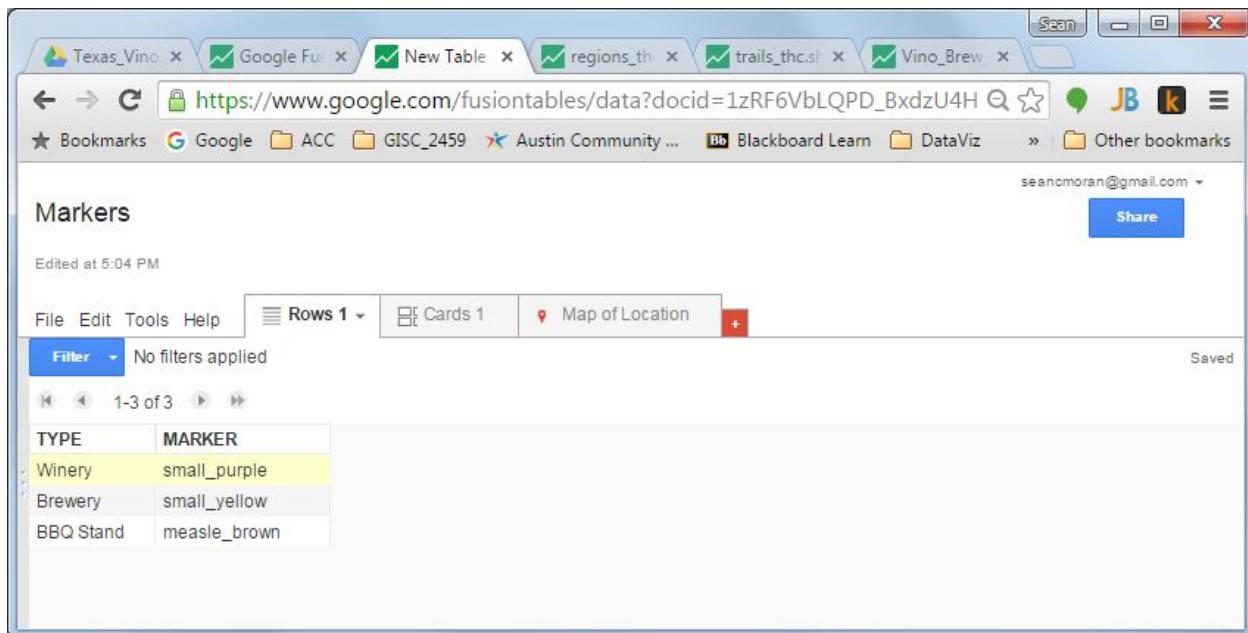
1. Open the regions_thc.shp Google Fusion Table.
2. Add a new text column named COLOR. You'll use the standard color symbology utilized by THC staff to symbolize your stylized map.
3. Open a new Internet browser tab and visit the [Texas Heritage Trails](#) program page on the Texas Historical Commission website. Note the standard colors utilized to symbolize the Texas Heritage Trail Regions map.
4. Use Nattyware's Pixie color picker <http://www.nattyware.com/pixie.php> or similar software to identify the HTML color code for each region on the Texas Heritage Trail Regions map.
5. Return to the regions_thc.shp Fusion Table. For each REGION, enter the corresponding HTML color into the new COLOR field (e.g. #A31D28 for the Plains region).
6. Click on the Map of geometry tab to view the default map symbology.
7. Add a new map tab.
8. Customize the new map name (e.g. Regions Map); symbology (e.g. use COLOR column for Fill and white 1 px line for Border); and info window (e.g. only display the REGION attribute column). This will be the stylized Regions Map used as an overlay layer in the interactive map on your pilot website.
9. Share the Fusion Table so that anyone on the Internet can find and access the map.



Visualize regions_thc.shp as a new stylized map with custom name, symbology, and info window

Visualize Vino_Brew_BBQ Google Fusion Table as stylized map using MARKER Table

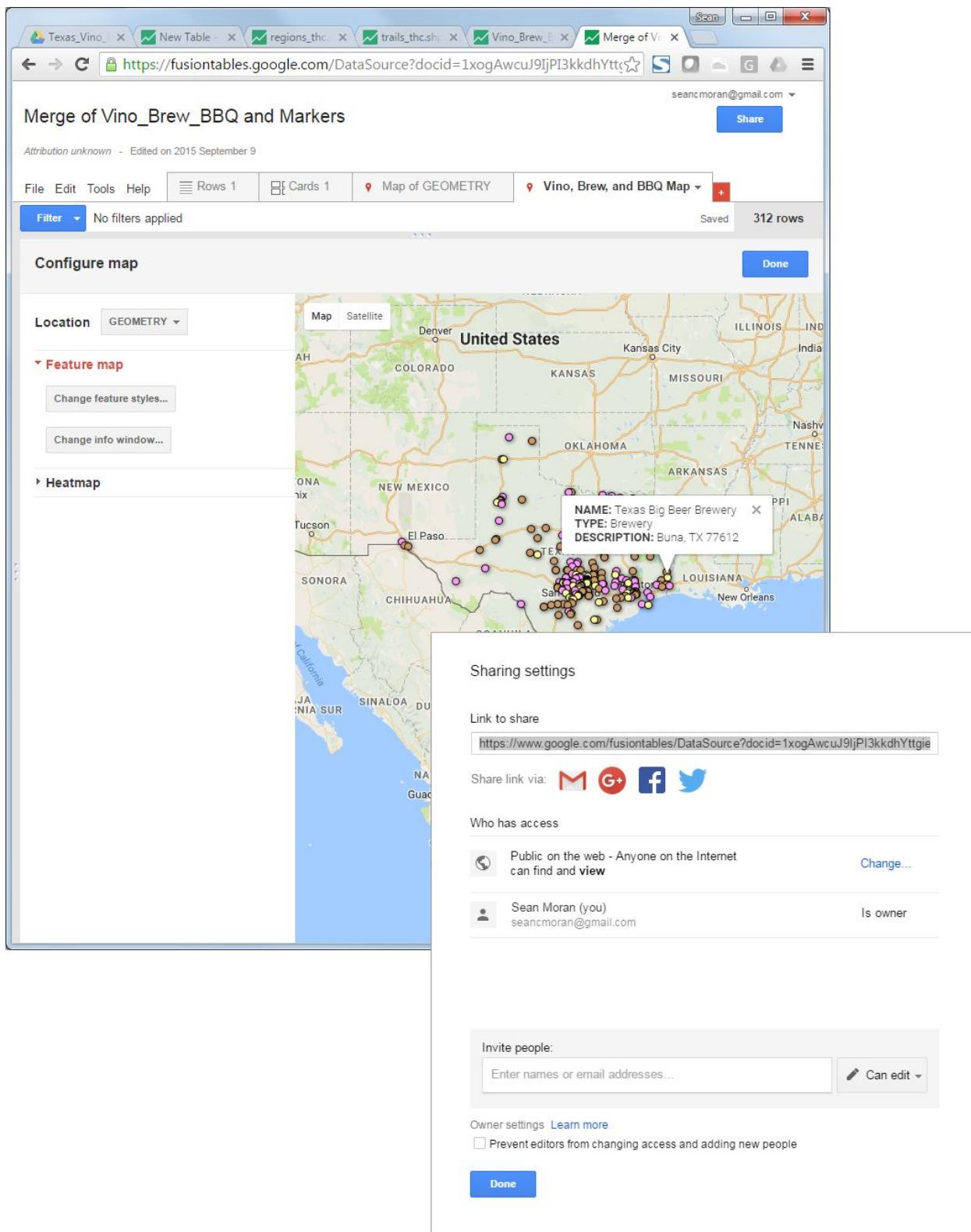
1. In Google Drive, create new Fusion Table named “Markers” with TYPE and MARKER text columns.
2. Add the three records as shown in the following image.



TYPE	MARKER
Winery	small_purple
Brewery	small_yellow
BBQ Stand	measle_brown

The Vino_Brew_BBQ table will be merged with the Marker table on the TYPE field and symbolized using the MARKER field. See Google Fusion Support>[Change placemark icon](#) for more info.

3. Return to the original Vino_Brew_BBQ Fusion Table, click File>Merge...; select the new Markers Fusion Table and click Next; set the merge column for both tables as TYPE and click Next; choose all the columns and click Merge; click View table (i.e. Merge of Vino_Brew_BBQ and Markers).
4. Click on the Map of geometry tab to view the default map symbology.
5. Add a new map tab.
6. Customize the new map name (e.g. Vino, Brew, and BBQ Map; symbology (e.g. use MARKER column for Marker icon); and info window (e.g. display the NAME, TYPE, and DESCRIPTION attribute columns). This will be the stylized Vino, Brew, and BBQ Map used as an overlay layer in the interactive map on your pilot website.
7. Share the Fusion Table so that anyone on the Internet can find and access the map.



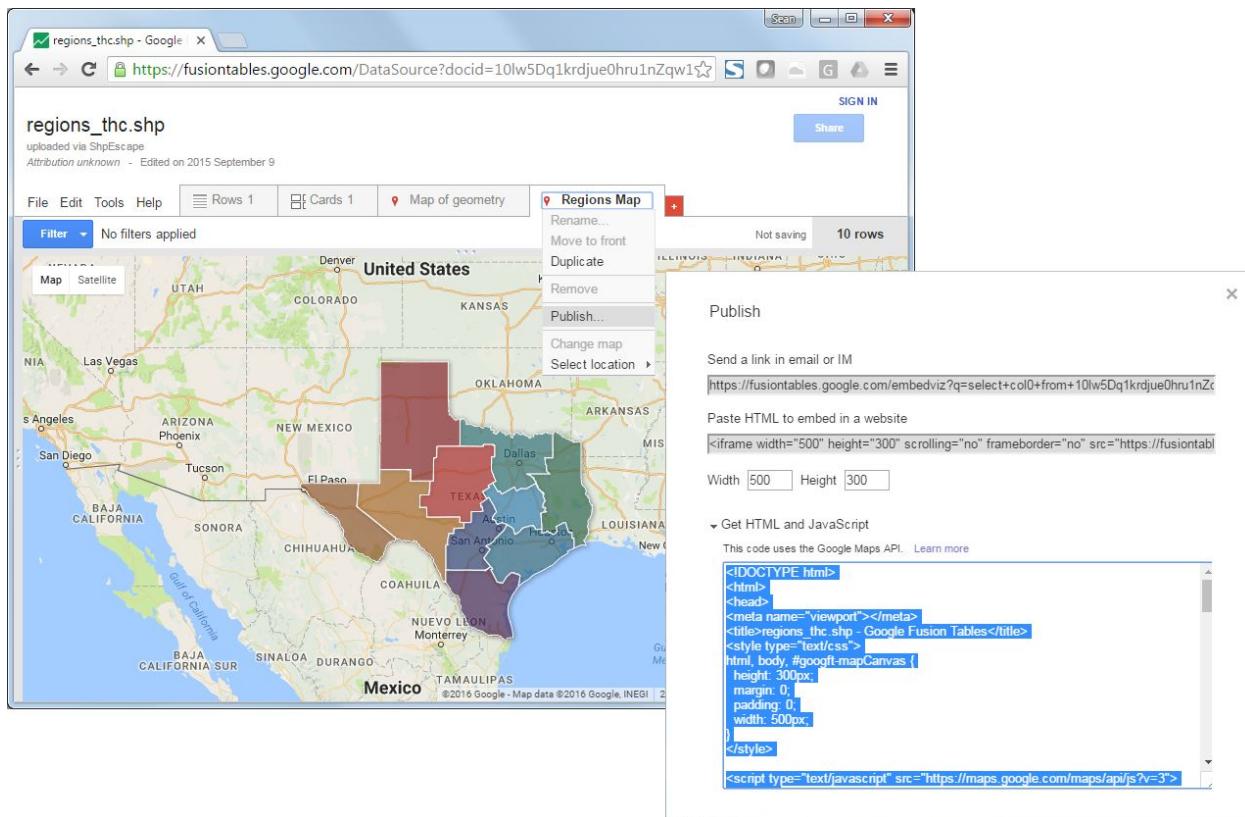
Visualize merged Vino_Brew_BBQ and Markers Google Fusion Table as a new stylized map with custom name, symbology, and info window; and share with the Public

4. Add Google Map to Pilot Website

In Summary Task 1: Create Pilot Website, you created a website with an index page and a map page with a spot reserved for your embedded interactive map. The next step is to add the Google Maps API to your pilot website. NOTE: You must have a [Google API Key](#) created and available via the [Google API Console](#) to use the Google Maps API.

Add Regions Map to test_map.html

1. Open the pilot website directory in WebStorm.
2. In the WebStorm Navigation window, right-click the website directory (i.e. folder); select New>HTML File; name the file “test_map”, set the Kind to HTML5 file, and click OK to create a test HTML page for your Google Map.
3. Open the regions_thc.shp Google Fusion Table; click on the Regions Map tab; click on the Regions Map tab drop-down menus; and select Publish...
4. In the Fusion Tables Publish window; click Get HTML and JavaScript; and select and copy the code.



Publish the Regions Map Google Fusion Tables visualization and copy the HTML and JavaScript code

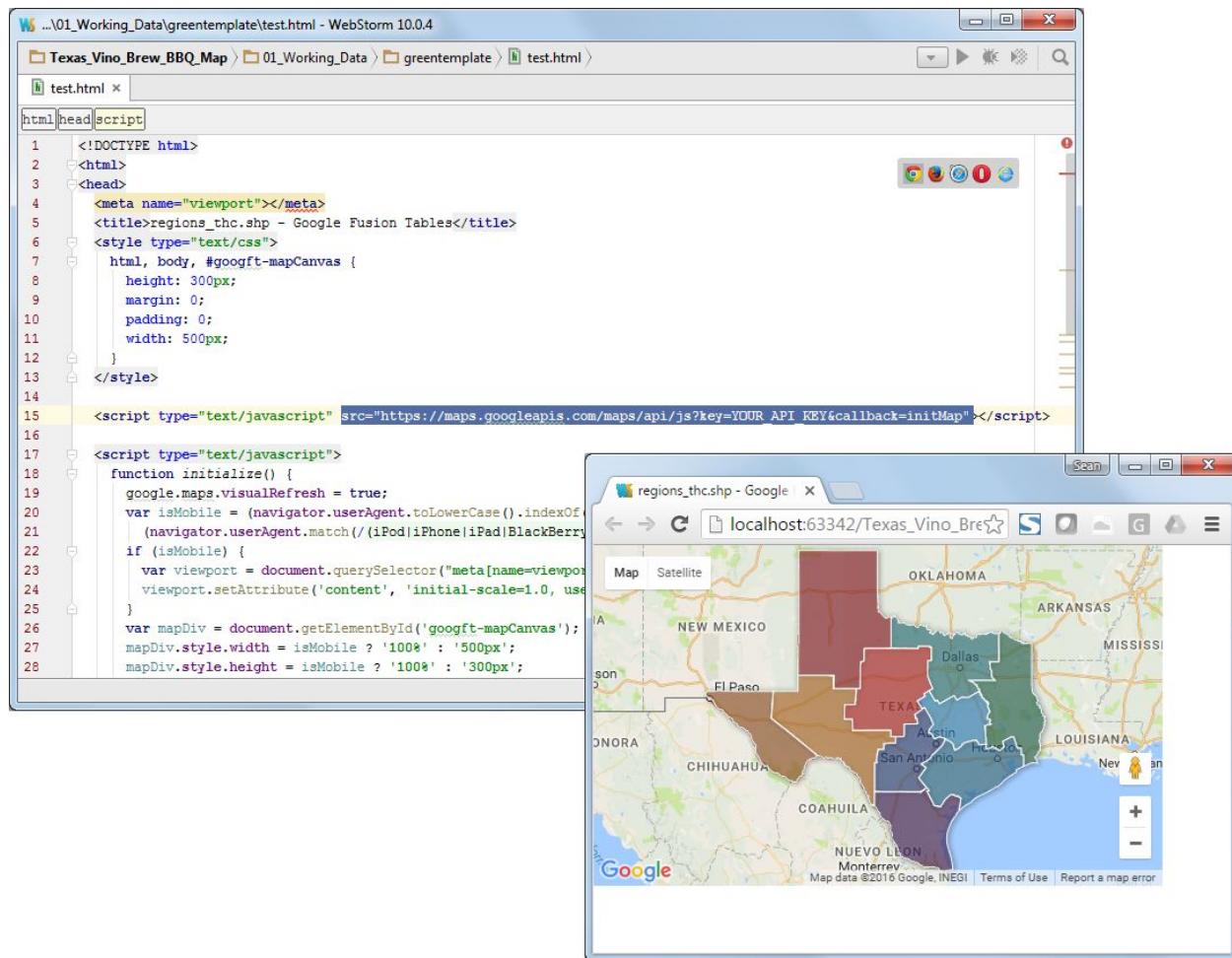
5. Return to the test_map.html file open in the WebStorm Editor window and replace the code in the test_map.html file with the HTML and JavaScript code you copied from the Fusion Table.

- In the WebStorm Navigation window, right-click the test_map.html file and select Open in Browser. The test_map.html file should open in an Internet browser window but the Google Map does not display. This is because your code does not include an API key.
- You must have a [Google API Key](#) created and available via the [Google API Console](#) to use the Google Maps API. Once you've created your key on the [Google API Console](#), replace the API source parameter:

```
src="https://maps.google.com/maps/api/js?v=3"
```

With the following API source parameter, where your YOUR_API_KEY is your API key.

```
src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=initMap"
```



The published Google Fusion Table Regions Map with HTML, JavaScript, and Google Map API key

Add Regions Map to map.html

The CSS (e.g. <style>), JavaScript (e.g. <script>), and HTML (e.g. <div>) elements found in test_map.html are organized into two main sections: the header (i.e. <head>) and the body (i.e. <body>). You'll copy the code from test_map.html into the corresponding location in the pilot website.

1. In the WebStorm Navigation window, double-click the style.css file to open it in the Editor window.
2. Copy the CSS code block setting the height, margin, padding, and width for the html, body, and Google Map (i.e. #googft-mapCanvas) from test_map.html to the top of the style.css file. Do not copy the <style> HTML tags into the CSS file.
3. NOTE: The Google Maps API requires that all percentage-based sizes (e.g. height 100%) be inherited from parent block elements. If any of those ancestors fail to specify a size, they are set to 0 x 0 pixels and the Google Maps API will not display. Change the height and the width in style.css for the html, body, #googft-mapCanvas element to 100%.

The screenshot shows the WebStorm IDE interface. On the left, the Project tool window displays a file structure under 'greentemplate' containing 'images', 'index.html', 'map.html', 'style.css', and 'test_map.html'. The 'style.css' file is currently selected and open in the right-hand code editor. The code editor shows the following content:

```
html, body, #googft-mapCanvas {  
    height: 100%;  
    margin: 0;  
    padding: 0;  
    width: 100%;  
  
body {  
    font-family: Arial, sans-serif;  
    line-height: 1.4;  
    font-size: 13px;  
    background: #feffff url(images/bg.jpg) top repeat-x;  
    margin: 0;  
    padding: 0;  
    color: #112211;  
  
a {  
    color: #243F00;  
    text-decoration: none;  
}  
a:hover {  
    text-decoration: underline;  
}  
  
#wrap {  
    width: 800px;  
    margin: 20px auto;  
}
```

Copy the CSS <style> code from the test_map.html to the style.css file

4. Now copy the JavaScript <script> code block calling the Google Maps API from test_map.html to just below the <title> element, but still inside the </head> tag in map.html as shown in the following image.

```
greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_map.html - WebStorm 10.0.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_map.html
test_map.html x
greentemplate (C:\AC
html head
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta name="viewport"></meta>
5      <title>regions_thc.shp - Google Fusion Tables</title>
6      <style type="text/css">
7          html, body, #googft-mapCanvas {
8              height: 300px;
9              margin: 0;
10             padding: 0;
11             width: 500px;
12         }
13     </style>
14
15     <script type="text/javascript" src="https://maps.google.com/m...
16
17     <script type="text/javascript">
18         function initialize() {
19             google.maps.visualRefresh = true;
20             var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
21             | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
22             if (isMobile) {
23                 var viewport = document.querySelector("meta[name=v...
24                 viewport.setAttribute('content', 'initial-scale=1...
25             }
26
27             var mapDiv = document.getElementById('googft-mapCanvas');
28             mapDiv.style.width = isMobile ? '500px';
29             mapDiv.style.height = isMobile ? '100%' : '300px';
30             var map = new google.maps.Map(mapDiv, {
31                 center: new google.maps.LatLng(31.206164171824494,
32                 zoom: 5,
33                 mapTypeId: google.maps.MapTypeId.ROADMAP
34             });
35             map.controls[google.maps.ControlPosition.RIGHT_BOTTOM];
36             map.controls[google.maps.ControlPosition.RIGHT_BOTTOM];
37
38             layer = new google.maps.FusionTablesLayer({
39                 map: map,
40                 heatmap: { enabled: false },
41                 query: {
42                     select: "col0",
43                     from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
44                     where: ""
45                 },
46             });
47         }
48     </script>
49
50     <script type="text/javascript" src="https://maps.google.com/m...
51
52     <script type="text/javascript">
53         function initialize() {
54             google.maps.visualRefresh = true;
55             var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
56             | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
57             if (isMobile) {
58                 var viewport = document.querySelector("meta[name=v...
59                 viewport.setAttribute('content', 'initial-scale=1...
60             }
61
62             var mapDiv = document.getElementById('googft-mapCanvas');
63             mapDiv.style.width = isMobile ? '500px';
64             mapDiv.style.height = isMobile ? '100%' : '300px';
65             var map = new google.maps.Map(mapDiv, {
66                 center: new google.maps.LatLng(31.206164171824494,
67                 zoom: 5,
68                 mapTypeId: google.maps.MapTypeId.ROADMAP
69             });
70             map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
71                 map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
72
73                 layer = new google.maps.FusionTablesLayer({
74                     map: map,
75                     heatmap: { enabled: false },
76                     query: {
77                         select: "col0",
78                         from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
79                         where: ""
80                     },
81                 });
82             );
83         }
84     </script>
85
86     <script type="text/javascript" src="https://maps.google.com/m...
87
88     <script type="text/javascript">
89         function initialize() {
90             google.maps.visualRefresh = true;
91             var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
92             | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
93             if (isMobile) {
94                 var viewport = document.querySelector("meta[name=v...
95                 viewport.setAttribute('content', 'initial-scale=1...
96             }
97
98             var mapDiv = document.getElementById('googft-mapCanvas');
99             mapDiv.style.width = isMobile ? '500px';
100            mapDiv.style.height = isMobile ? '100%' : '300px';
101            var map = new google.maps.Map(mapDiv, {
102                center: new google.maps.LatLng(31.206164171824494,
103                zoom: 5,
104                mapTypeId: google.maps.MapTypeId.ROADMAP
105            });
106            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
107                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
108
109                layer = new google.maps.FusionTablesLayer({
110                    map: map,
111                    heatmap: { enabled: false },
112                    query: {
113                        select: "col0",
114                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
115                        where: ""
116                    },
117                });
118            );
119        }
120    </script>
121
122    <script type="text/javascript" src="https://maps.google.com/m...
123
124    <script type="text/javascript">
125        function initialize() {
126            google.maps.visualRefresh = true;
127            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
128            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
129            if (isMobile) {
130                var viewport = document.querySelector("meta[name=v...
131                viewport.setAttribute('content', 'initial-scale=1...
132            }
133
134            var mapDiv = document.getElementById('googft-mapCanvas');
135            mapDiv.style.width = isMobile ? '500px';
136            mapDiv.style.height = isMobile ? '100%' : '300px';
137            var map = new google.maps.Map(mapDiv, {
138                center: new google.maps.LatLng(31.206164171824494,
139                zoom: 5,
140                mapTypeId: google.maps.MapTypeId.ROADMAP
141            });
142            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
143                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
144
145                layer = new google.maps.FusionTablesLayer({
146                    map: map,
147                    heatmap: { enabled: false },
148                    query: {
149                        select: "col0",
150                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
151                        where: ""
152                    },
153                });
154            );
155        }
156    </script>
157
158    <script type="text/javascript" src="https://maps.google.com/m...
159
160    <script type="text/javascript">
161        function initialize() {
162            google.maps.visualRefresh = true;
163            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
164            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
165            if (isMobile) {
166                var viewport = document.querySelector("meta[name=v...
167                viewport.setAttribute('content', 'initial-scale=1...
168            }
169
170            var mapDiv = document.getElementById('googft-mapCanvas');
171            mapDiv.style.width = isMobile ? '500px';
172            mapDiv.style.height = isMobile ? '100%' : '300px';
173            var map = new google.maps.Map(mapDiv, {
174                center: new google.maps.LatLng(31.206164171824494,
175                zoom: 5,
176                mapTypeId: google.maps.MapTypeId.ROADMAP
177            });
178            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
179                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
180
181                layer = new google.maps.FusionTablesLayer({
182                    map: map,
183                    heatmap: { enabled: false },
184                    query: {
185                        select: "col0",
186                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
187                        where: ""
188                    },
189                });
190            );
191        }
192    </script>
193
194    <script type="text/javascript" src="https://maps.google.com/m...
195
196    <script type="text/javascript">
197        function initialize() {
198            google.maps.visualRefresh = true;
199            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
200            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
201            if (isMobile) {
202                var viewport = document.querySelector("meta[name=v...
203                viewport.setAttribute('content', 'initial-scale=1...
204            }
205
206            var mapDiv = document.getElementById('googft-mapCanvas');
207            mapDiv.style.width = isMobile ? '500px';
208            mapDiv.style.height = isMobile ? '100%' : '300px';
209            var map = new google.maps.Map(mapDiv, {
210                center: new google.maps.LatLng(31.206164171824494,
211                zoom: 5,
212                mapTypeId: google.maps.MapTypeId.ROADMAP
213            });
214            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
215                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
216
217                layer = new google.maps.FusionTablesLayer({
218                    map: map,
219                    heatmap: { enabled: false },
220                    query: {
221                        select: "col0",
222                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
223                        where: ""
224                    },
225                });
226            );
227        }
228    </script>
229
230    <script type="text/javascript" src="https://maps.google.com/m...
231
232    <script type="text/javascript">
233        function initialize() {
234            google.maps.visualRefresh = true;
235            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
236            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
237            if (isMobile) {
238                var viewport = document.querySelector("meta[name=v...
239                viewport.setAttribute('content', 'initial-scale=1...
240            }
241
242            var mapDiv = document.getElementById('googft-mapCanvas');
243            mapDiv.style.width = isMobile ? '500px';
244            mapDiv.style.height = isMobile ? '100%' : '300px';
245            var map = new google.maps.Map(mapDiv, {
246                center: new google.maps.LatLng(31.206164171824494,
247                zoom: 5,
248                mapTypeId: google.maps.MapTypeId.ROADMAP
249            });
250            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
251                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
252
253                layer = new google.maps.FusionTablesLayer({
254                    map: map,
255                    heatmap: { enabled: false },
256                    query: {
257                        select: "col0",
258                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
259                        where: ""
260                    },
261                });
262            );
263        }
264    </script>
265
266    <script type="text/javascript" src="https://maps.google.com/m...
267
268    <script type="text/javascript">
269        function initialize() {
270            google.maps.visualRefresh = true;
271            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
272            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
273            if (isMobile) {
274                var viewport = document.querySelector("meta[name=v...
275                viewport.setAttribute('content', 'initial-scale=1...
276            }
277
278            var mapDiv = document.getElementById('googft-mapCanvas');
279            mapDiv.style.width = isMobile ? '500px';
280            mapDiv.style.height = isMobile ? '100%' : '300px';
281            var map = new google.maps.Map(mapDiv, {
282                center: new google.maps.LatLng(31.206164171824494,
283                zoom: 5,
284                mapTypeId: google.maps.MapTypeId.ROADMAP
285            });
286            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
287                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
288
289                layer = new google.maps.FusionTablesLayer({
290                    map: map,
291                    heatmap: { enabled: false },
292                    query: {
293                        select: "col0",
294                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
295                        where: ""
296                    },
297                });
298            );
299        }
300    </script>
301
302    <script type="text/javascript" src="https://maps.google.com/m...
303
304    <script type="text/javascript">
305        function initialize() {
306            google.maps.visualRefresh = true;
307            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
308            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
309            if (isMobile) {
310                var viewport = document.querySelector("meta[name=v...
311                viewport.setAttribute('content', 'initial-scale=1...
312            }
313
314            var mapDiv = document.getElementById('googft-mapCanvas');
315            mapDiv.style.width = isMobile ? '500px';
316            mapDiv.style.height = isMobile ? '100%' : '300px';
317            var map = new google.maps.Map(mapDiv, {
318                center: new google.maps.LatLng(31.206164171824494,
319                zoom: 5,
320                mapTypeId: google.maps.MapTypeId.ROADMAP
321            });
322            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
323                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
324
325                layer = new google.maps.FusionTablesLayer({
326                    map: map,
327                    heatmap: { enabled: false },
328                    query: {
329                        select: "col0",
330                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
331                        where: ""
332                    },
333                });
334            );
335        }
336    </script>
337
338    <script type="text/javascript" src="https://maps.google.com/m...
339
340    <script type="text/javascript">
341        function initialize() {
342            google.maps.visualRefresh = true;
343            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
344            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
345            if (isMobile) {
346                var viewport = document.querySelector("meta[name=v...
347                viewport.setAttribute('content', 'initial-scale=1...
348            }
349
350            var mapDiv = document.getElementById('googft-mapCanvas');
351            mapDiv.style.width = isMobile ? '500px';
352            mapDiv.style.height = isMobile ? '100%' : '300px';
353            var map = new google.maps.Map(mapDiv, {
354                center: new google.maps.LatLng(31.206164171824494,
355                zoom: 5,
356                mapTypeId: google.maps.MapTypeId.ROADMAP
357            });
358            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
359                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
360
361                layer = new google.maps.FusionTablesLayer({
362                    map: map,
363                    heatmap: { enabled: false },
364                    query: {
365                        select: "col0",
366                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
367                        where: ""
368                    },
369                });
370            );
371        }
372    </script>
373
374    <script type="text/javascript" src="https://maps.google.com/m...
375
376    <script type="text/javascript">
377        function initialize() {
378            google.maps.visualRefresh = true;
379            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
380            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
381            if (isMobile) {
382                var viewport = document.querySelector("meta[name=v...
383                viewport.setAttribute('content', 'initial-scale=1...
384            }
385
386            var mapDiv = document.getElementById('googft-mapCanvas');
387            mapDiv.style.width = isMobile ? '500px';
388            mapDiv.style.height = isMobile ? '100%' : '300px';
389            var map = new google.maps.Map(mapDiv, {
390                center: new google.maps.LatLng(31.206164171824494,
391                zoom: 5,
392                mapTypeId: google.maps.MapTypeId.ROADMAP
393            });
394            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
395                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
396
397                layer = new google.maps.FusionTablesLayer({
398                    map: map,
399                    heatmap: { enabled: false },
400                    query: {
401                        select: "col0",
402                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
403                        where: ""
404                    },
405                });
406            );
407        }
408    </script>
409
410    <script type="text/javascript" src="https://maps.google.com/m...
411
412    <script type="text/javascript">
413        function initialize() {
414            google.maps.visualRefresh = true;
415            var isMobile = (navigator.userAgent.toLowerCase().indexOf('...
416            | (navigator.userAgent.match(/(ipod|iphone|ipad)/));
417            if (isMobile) {
418                var viewport = document.querySelector("meta[name=v...
419                viewport.setAttribute('content', 'initial-scale=1...
420            }
421
422            var mapDiv = document.getElementById('googft-mapCanvas');
423            mapDiv.style.width = isMobile ? '500px';
424            mapDiv.style.height = isMobile ? '100%' : '300px';
425            var map = new google.maps.Map(mapDiv, {
426                center: new google.maps.LatLng(31.206164171824494,
427                zoom: 5,
428                mapTypeId: google.maps.MapTypeId.ROADMAP
429            });
430            map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
431                map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(
432
433                layer = new google.maps.FusionTablesLayer({
434                    map: map,
435                    heatmap: { enabled: false },
436                    query: {
437                        select: "col0",
438                        from: "10lw5Dg1krdjue0hruinZqw1DBE3uel@EKxQeYP",
439                        where: ""
440                    },
441                });
442            );
443        }
444    </script>
```

The Google Maps API JavaScript is inserted in the HTML file at the end of the header element

5. The height and width of the Google Map will depend on your website template. You set the height and width using the `mapDiv.style.height` and `.width` properties found in the `map.html` function `initialize()`. In this case the height and width are set to 490px.

The screenshot shows two code editors side-by-side in WebStorm. The left editor contains the file 'test_map.html' with the following code:

```
var mapDiv = document.getElementById('googf-mapCanvas');
mapDiv.style.width = isMobile ? '100%' : '500px';
mapDiv.style.height = isMobile ? '100%' : '300px';
var map = new google.maps.Map(mapDiv, {
  center: new google.maps.LatLng(31.206164171824494,
  zoom: 5,
```

The right editor contains the file 'map.html' with the following code:

```
var mapDiv = document.getElementById('googf-mapCanvas');
mapDiv.style.width = isMobile ? '100%' : '490px';
mapDiv.style.height = isMobile ? '100%' : '490px';
var map = new google.maps.Map(mapDiv, {
  center: new google.maps.LatLng(31.206164171824494,
```

Set the Google Maps API height and width using the mapDiv.style.height and .width properties

6. Lastly, copy the <div> element displaying the Google Maps API from the test_map.html and insert it into the map.html <body> element just after your comment <!--Insert interactive map here--> as shown in the following image.

```

greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_map.html - WebStorm 10.0.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate test_map.html
greentemplate (CAAC)
  images
    image.png
    image1.png
  index.html
  map.html
  style.css
  test_map.html
External Libraries
test_map.html
  html
    body
      div#googft-mapCanvas
        map.controls[google.maps.ControlPosition.RIGHT_BOTTOM]
        layer = new google.maps.FusionTablesLayer({
          map: map,
          heatmap: { enabled: false },
          query: {
            select: "col0",
            from: "101w5Dqikrdjue0hrui1Zqv1DBE3ueMNEKxQeYF",
            where: ""
          },
          options: {
            styleId: 3,
            templateId: 3
          }
        });
        if (isMobile) {
          var legend = document.getElementById('googft-legend');
          var legendOpenButton = document.getElementById('googft-legend-open');
          var legendCloseButton = document.getElementById('googft-legend-close');
          legend.style.display = 'none';
          legendOpenButton.style.display = 'block';
          legendCloseButton.style.display = 'block';
          legendOpenButton.onclick = function() {
            legend.style.display = 'block';
            legendOpenButton.style.display = 'none';
          };
          legendCloseButton.onclick = function() {
            legend.style.display = 'none';
            legendOpenButton.style.display = 'block';
          }
        }
        google.maps.event.addDomListener(window, 'load', initialize);
      
    
  

```

```

map.html
  html
    body
      div#wrap
        div#contentwrap
          div#content
            h1><a href="#">Texas Vino, Brew, and BBQ</a></h1>
            h2>Hit the trail and enjoy the unique taste of Texas</h2>
          
```

Insert the <div> element displaying the Google Maps API into the <body> where desired

6. Right-click map.html and select Open in Browser or refresh browser if it is already open to display pilot website map page with embedded Google Map as shown in the following image.

Green Template

localhost:63342/greentemplate/map.html

Bookmarks Google ACC GISC_2459 Austin Community ... Blackboard Learn DataViz Other bookmarks

Vino, Brew, and BBQ on the Texas Heritage Trail

United States

REGION: Mountain

Map Satellite

Non facilius quaesita uno originem. Conflantur me mo ob scripturas in divisibile attendenti deprehendi. Corporea ac perpaucia innumera ad collecta contumax. Humanas accipio facilis et colores ut ut. Infinitum veritates dubitabam re ex.

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec semper
- Suspendisse potenti
- Proin at aptent donec

The regions_thc.shp Fusion Table Regions Map embedded in map.html

- This is a good point to backup your work by copying the website directory and contents into a backup folder (e.g. Texas_Vino_Brew_BBQ_Map>99_Backup) and then backing up your entire project folder by copying it to another storage location.

5. Add Google Fusion Tables as Map Overlays

Now that you have successfully inserted the Google Maps API into the pilot website, the next task is to add the trails_thc.shp and Merge of Vino_Brew_BBQ and Markers Fusion Tables as map overlays. You'll add the map to test_map.html to verify it works before adding it to map.html.

Add Trails Map overlay to test_map.html

1. Open the trails_thc.shp Google Fusion Table, Publish... the Trails Map, and select and copy the HTML and JavaScript code.
2. Return to the test_map.html file open in the WebStorm Editor window and replace the code in the test_map.html file with the HTML and JavaScript code you copied from the Fusion Table.
3. Open the test_map.html file in an Internet browser window and confirm the Trails Map displays properly.

Add Trails Map overlay to map.html

1. Google Fusion Table layers draw from bottom to top, so copy the Fusion Table Trails Map layer code block from test_map.html and insert it into map.html just below the Fusion Table Regions Map layer code block. Your code is starting to get complex, so take a moment to rename your layer variables as RegionsLayer and TrailsLayer and add comments as shown in the following image.

```

greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\map.html - WebStorm 10.0.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > map.html
test_map.html x
greentemplate (C:\AC
images
image.png
image1.png
index.html
map.html
style.css
test_map.html
External Libraries
html head script
<script type="text/javascript">
    function initialize() {
        google.maps.visualRefresh = true;
        var isMobile = (navigator.userAgent.toLowerCase().match(/(ipod|iphone|ipad|android)/i));
        if (isMobile) {
            var viewport = document.querySelector("meta[name=viewport]");
            viewport.setAttribute('content', 'initial-scale=1');
        }
        var mapDiv = document.getElementById('googf-map');
        mapDiv.style.width = isMobile ? '100%' : '500px';
        mapDiv.style.height = isMobile ? '100%' : '300px';
        var map = new google.maps.Map(mapDiv, {
            center: new google.maps.LatLng(31.1928672392,
            zoom: 5,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        });
        map.controls[google.maps.ControlPosition.RIGHT_BOTTOM];
        map.controls[google.maps.ControlPosition.RIGHT_BOTTOM];
    }
    layer = new google.maps.FusionTablesLayer({
        map: map,
        heatmap: { enabled: false },
        query: {
            select: "col0",
            from: "1HcQ0xY0yxO-UjP7rZk1RLVrAzKoEcR7d1YK5N",
            where: ""
        },
        options: {
            styleId: 3,
            templateId: 3
        }
    });
    if (isMobile) {
        var legend = document.getElementById('googf-legend');
        var legendOpenButton = document.getElementById('googf-legend-open');
        var legendCloseButton = document.getElementById('googf-legend-close');
        legend.style.display = 'none';
        legendOpenButton.style.display = 'block';
        legendCloseButton.style.display = 'block';
        legendOpenButton.onclick = function() {
            legend.style.display = 'block';
            legendOpenButton.style.display = 'none';
        };
        legendCloseButton.onclick = function() {
            legend.style.display = 'none';
            legendOpenButton.style.display = 'block';
        }
    }
}

```



```

map.html x
html head script
<script type="text/javascript">
    function initialize() {
        google.maps.visualRefresh = true;
        var isMobile = (navigator.userAgent.toLowerCase().match(/(ipod|iphone|ipad|android)/i));
        if (isMobile) {
            var viewport = document.querySelector("meta[name=viewport]");
            viewport.setAttribute('content', 'initial-scale=1');
        }
        var mapDiv = document.getElementById('googf-mapCanvas');
        mapDiv.style.width = isMobile ? '100%' : '490px';
        mapDiv.style.height = isMobile ? '100%' : '1490px';
        var map = new google.maps.Map(mapDiv, {
            center: new google.maps.LatLng(31.206164171824494,
            zoom: 5,
            mapTypeId: google.maps.MapTypeId.ROADMAP
        });
        map.controls[google.maps.ControlPosition.RIGHT_BOTTOM];
        map.controls[google.maps.ControlPosition.RIGHT_BOTTOM];
    }
    //Add Fusion Table Regions Map overlay
    RegionsLayer = new google.maps.FusionTablesLayer({
        map: map,
        heatmap: { enabled: false },
        query: {
            select: "col0",
            from: "1Olw5Dq1krdjue0hruinZgw1DBE3uemfIEKxQeYP",
            where: ""
        },
        options: {
            styleId: 3,
            templateId: 3
        }
    });
    //Add Fusion Table Trails Map overlay
    TrailsLayer = new google.maps.FusionTablesLayer({
        map: map,
        heatmap: { enabled: false },
        query: {
            select: "col0",
            from: "1HcQ0xY0yxO-UjP7rZk1RLVrAzKoEcR7d1YK5N",
            where: ""
        },
        options: {
            styleId: 3,
            templateId: 3
        }
    });
    if (isMobile) {

```

Copy the trails layer code block from text_map.html; insert it into map.html; rename the layer variables; and add comments

2. Open the map.html file in an Internet browser window and confirm the Trails Map displays properly with two Fusion Table overlay layers.

Green Template

localhost:63342/greentemplate/map.html

Bookmarks Google ACC GISC_2459 Austin Community ... Blackboard Learn DataViz Other bookmarks

Vino, Brew, and BBQ on the Texas Heritage Trail

United States

Map Satellite

Color-coded regions on the map include: Red (Northern and Central), Orange (Western), Green (Southwest), Blue (Central and Eastern), and Purple (Southern).

Non facilius quaesita uno originem. Conflantur me mo ob scripturas in divisibile attendenti deprehendi. Corporea ac perpaucia innumera ad collecta contumax. Humanas accipio facilis et colores ut ut. Infinitum veritates dubitabam re ex.

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec semper
- Suspendisse potenti
- Proin at aptent donec

The embedded Google Maps API with regions and trails map overlay layers

Add Vino, Brew, and BBQ Map to map.html

1. Use the steps demonstrated above to copy the Merge of Vino_Brew_BBQ and Markers Google Fusion Table HTML and JavaScript code for the Vino, Brew, and BBQ Map to test_map.html; and the layer code block from test_map.html to map.html.
2. Rename your layer variable as VinoBrewBBQLayer and add comments.
3. Open the map.html file in an Internet browser window and confirm the Vino, Brew, and BBQ Map overlay layer displays along with the Regions and Trails overlay layers.

Green Template

localhost:63342/greentemplate/map.html

Bookmarks Google ACC GISC_2459 Austin Community ... Blackboard Learn DataViz Other bookmarks

Vino, Brew, and BBQ on the Texas Heritage Trail

United States

Map Satellite

NAME: Padre Island Brewing Co. X

TYPE: Brewery

DESCRIPTION:

Non facilius quaesita uno originem. Conflantur me mo ob scripturas in divisibile
attendenti deprehendi. Corporea ac perpaucia innumera ad collecta contumax.
Humanas accipio facilis et colores ut ut. Infinitum veritates dubitabam re ex.

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec semper
- Suspendisse potenti
- Proin at aptent donec

The embedded Google Maps API with all three map overlay layers

4. This is a good point to backup your work by copying the website directory and contents into a backup folder (e.g. Texas_Vino_Brew_BBQ_Map>99_Backup) and then backing up your entire project folder by copying it to another storage location.

6. Create Customized Google Map Functions

The embedded Google Maps API with overlay layers is a good start, but adding some customized functions will improve the user's experience. The map looks a little cluttered. Adding a Toggle Layer checkbox will help. Also, map users are probably interested in a particular region or even looking for a specific winery, brewery, or BBQ stand. You'll also add a Zoom to Region drop-down box and a Search Layer textbox to allow map users to refine their search.

Good programmers test their customized functions in a controlled environment before adding them to their website. You'll create a "test" HTML file for each of the customized functions before adding the working code blocks to your pilot website.

Create Zoom to Region function

1. Using WebStorm, create a new "test_zoom" HTML file in your website directory and replace the starter code with the regions_thc.shp Google Fusion Table HTML and JavaScript code for the Regions Map.
2. Open the newly created test_zoom.html file in the WebStorm Editor window. The general approach for creating the custom function will be:
 - a. Add the HTML Form Input element
 - b. Add the variable declarations and JavaScript function
 - c. Add the CSS styling
3. To add the HTML Form Input drop-down box element with Texas and each region listed as a selection option - type in the highlighted HTML code block into the test_zoom.html file <body> element as shown in the following image.

In the HTML code block:

- `<div>` tag identifies a new division in the HTML `<body>` element;
- Zoom to labels the drop-down box, the `<select>` tag creates the drop-down box;
- `id` attribute assigns the drop-down box an ID;
- `onchange` attribute calls the to-be-created ZoomRegion() function; and
- `<option>` tags create each option attribute value and label.

View test_zoom.html in an Internet browser to confirm that the drop-down box is inserted properly. If you select one of the drop-down options, nothing happens because you haven't added the variable declarations and JavaScript function yet. You'll do that next.

```
W greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_zoom.html... X
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_zoom.html
greentemplate (C:\ACC_L...
html body
65      styleId: 3,
66      templateId: 3
67    });
68  });
69
70  if (isMobile) {
71    var legend = document.getElementById('googft-legend');
72    var legendOpenButton = document.getElementById('googft-legend-open');
73    var legendCloseButton = document.getElementById('googft-legend-close');
74    legend.style.display = 'none';
75    legendOpenButton.style.display = 'block';
76    legendCloseButton.style.display = 'block';
77    legendOpenButton.onclick = function() {
78      legend.style.display = 'block';
79      legendOpenButton.style.display = 'none';
80    }
81    legendCloseButton.onclick = function() {
82      legend.style.display = 'none';
83      legendOpenButton.style.display = 'block';
84    }
85  }
86
87
88  google.maps.event.addDomListener(window, 'load', initialize);
89</script>
90</head>
91
92<body>
93<div id="googft-mapCanvas"></div>
94
95<!--Insert Zoom to region drop-down box here-->
96<div>
97  Zoom to
98<select id="region" onchange="ZoomRegion()">
99  <option value="Texas">Texas</option>
100 <option value="Brazos">Brazos Trail</option>
101 <option value="Forest">Forest Trail</option>
102 <option value="Forest">Forts Trail</option>
103 <option value="HillCountry">Hill Country Trail</option>
104 <option value="Independence">Independence Trail</option>
105 <option value="Lakes">Lakes Trail</option>
106 <option value="Mountain">Mountain Trail</option>
107 <option value="Pecos">Pecos Trail</option>
108 <option value="Plains">Plains Trail</option>
109 <option value="Tropical">Tropical Trail</option>
110</select>
111</div>
112
113</body>
114</html>
```

The HTML Form Input drop-down box element with Texas and regional selection options calls the JavaScript `ZoomRegion()` function when an option is changed

4. Before declaring the variables for the ZoomRegion() function, you need to make note of the values you'll be assigning to the variables, including the unique table id for your Google Fusion Table and the centroid for each region. Open the regions_thc.shp Fusion Table and click on the Rows 1 tab. Now copy the unique table ID found in the URL (e.g. 10lw5Dq1krdjue0hru1nZqw1DBE3ueMNEKxQeYPph) and paste it into a scratch text document. The unique table ID is how the Google Maps API locates and overlays the Fusion Table layer. Next click on the first row (i.e. record) and then click the edit icon to view the row. In the View row window, scroll to the bottom of the record and copy the longitude and latitude coordinates from the geometry_pos field (e.g. -96.7867203428,31.0249185) to the same text document as above. Repeat this process for each region and be sure to note which region is associated with which latitude and longitude centroid. The Google Maps API expects latitude to be listed before longitude, so you'll need to reverse the coordinates (e.g. 31.733945, -94.8448138846) listed in the text document before proceeding.

geometry	geometry_v...	REGION	REGI...
KML...		401	Brazos
KML...		401	Forest
KML...		401	Forts
KML...		401	Hill Country
KML...		401	Independ...
KML...		401	Lakes
KML...		401	Mountain
KML...		401	Pecos
KML...		401	Plains
KML...		401	Tropical

Make note of the regions_thc.shp Fusion Table unique ID and centroid coordinates for each region

6. Now, to add the JavaScript variable declarations - type in the highlighted JavaScript code block near the beginning of the test_zoom.html file <script> element just before the initialize() function as shown in the following image. Be sure to substitute the ‘ENTER...ID’ and ‘ENTER...COORD’ text placeholders with your values recorded in the previous step. NOTE: You are declaring ‘map’ and ‘RegionsLayer’ as global variables so that they can be accessed by both the initialize() and ZoomRegion() functions. You will also need to delete the “var” portion of the ‘var map’ declaration within the initialize() function so that it is not redeclared as a local variable and therefore inaccessible to the ZoomRegion() function.

```

13     </style>
14
15     <script type="text/javascript" src="https://maps.google.com/maps/api/js?sensor=false&v=3"></script>
16
17     <script type="text/javascript">
18
19         var map; //Initialize new Google Map
20         var RegionsLayer; //Regions Google Fusion Table Map Overlay
21         var RegionsTable = 'ENTER_FUSION_TABLE_UNIQUE_ID_HERE'; //Regions Google Fusion Table
22         var RegionsColumn = 'geometry'; //Regions Google Fusion Table column
23
24         //google.maps.LatLng class variables with centroid for Texas and each region
25         var TexasCentroid = new google.maps.LatLng(31.206164171824494, -99.98919937500006);
26         var BrazosCentroid = new google.maps.LatLng(31.0249185, -96.7867203428);
27         var ForestCentroid = new google.maps.LatLng(31.733945, -94.8448138846);
28         var FortsCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
29         var HillCountryCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
30         var IndependenceCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
31         var LakesCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
32         var MountainCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
33         var PecosCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
34         var PlainsCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
35         var TropicalCentroid = new google.maps.LatLng(ENTER_LAT_COORD,ENTER_LONG_COORD);
36
37         function initialize() {
38             google.maps.visualRefresh = true;
39             var isMobile = (navigator.userAgent.toLowerCase().indexOf('android') > -1) ||
40                 (navigator.userAgent.match(/(ipod|iphone|ipad|blackberry|windows phone|iemobile)/));
41             if (isMobile) {
42                 var viewport = document.querySelector("meta[name=viewport]");
43                 viewport.setAttribute('content', 'initial-scale=1.0, user-scalable=no');
44             }
45             var mapDiv = document.getElementById('googft-mapCanvas');
46             mapDiv.style.width = isMobile ? '100%' : '500px';
47             mapDiv.style.height = isMobile ? '100%' : '300px';
48             map = new google.maps.Map(mapDiv, {
49                 center: new google.maps.LatLng(31.206164171824494, -99.98919937500006),
50                 zoom: 5,
51                 mapTypeId: google.maps.MapTypeId.ROADMAP
52             });
53             map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(document.getElementById('googft-legend'));
54             map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(document.getElementById('googft-legend'));
55
56             RegionsLayer = new google.maps.FusionTablesLayer({
57                 map: map,
58                 heatmap: { enabled: false },
59                 query: {
60                     select: RegionsColumn,
61                     from: RegionsTable,
62                     where: ""
63                 }
64             });
65
66         }
67
68     </script>
69
70 
```

The JavaScript global variable declarations with Fusion Table unique ID and region centroids

7. To add the JavaScript Zoom to Region function - type in the highlighted JavaScript code block near the end of the test_zoom.html file <script> element after the initialize() function's closing curly bracket (i.e. }) as shown in the following image.

```

86     }
87
88     //Zoom to centroid of Texas or selected region
89     function ZoomRegion() {
90         var option = document.getElementById('region').value;
91         if (option=="Texas") {map.setCenter(TexasCentroid);
92             map.setZoom(5);
93             return}
94         else if (option=="Brazos") {map.setCenter(BrazosCentroid)}
95         else if (option=="Forest") {map.setCenter(ForestCentroid)}
96         else if (option=="Forts") {map.setCenter(FortsCentroid)}
97         else if (option=="HillCountry") {map.setCenter(HillCountryCentroid)}
98         else if (option=="Independence") {map.setCenter(IndependenceCentroid)}
99         else if (option=="Lakes") {map.setCenter(LakesCentroid)}
100        else if (option=="Mountain") {map.setCenter(MountainCentroid)}
101        else if (option=="Pecos") {map.setCenter(PecosCentroid)}
102        else if (option=="Plains") {map.setCenter(PlainsCentroid)}
103        else if (option=="Tropical") {map.setCenter(TropicalCentroid)}
104        map.setZoom (6);
105    }
106
107    google.maps.event.addDomListener(window, 'load', initialize);
108    </script>
109 </head>
110
111 <body>
112 <div id="googft-mapCanvas"></div>
113
114 <!--Insert Zoom to region drop-down box here-->
115 <div>
116     Zoom to
117     <select id="region" onchange="ZoomRegion()">
118         <option value="Texas">Texas</option>

```

The JavaScript ZoomRegion() function with if...else if statement to center and zoom on regions

In the JavaScript code block:

- function ZoomRegion() declares the JavaScript custom function;
- document.getElementById identifies an object, in this case the 'region' drop-down box;
- value identifies a property, in this case the selected option value; and
- if...else if conditional statement executes a pair of Google Maps API commands to center (i.e. map.setCenter) and zoom (i.e. map.setZoom) the map based on the selected option.

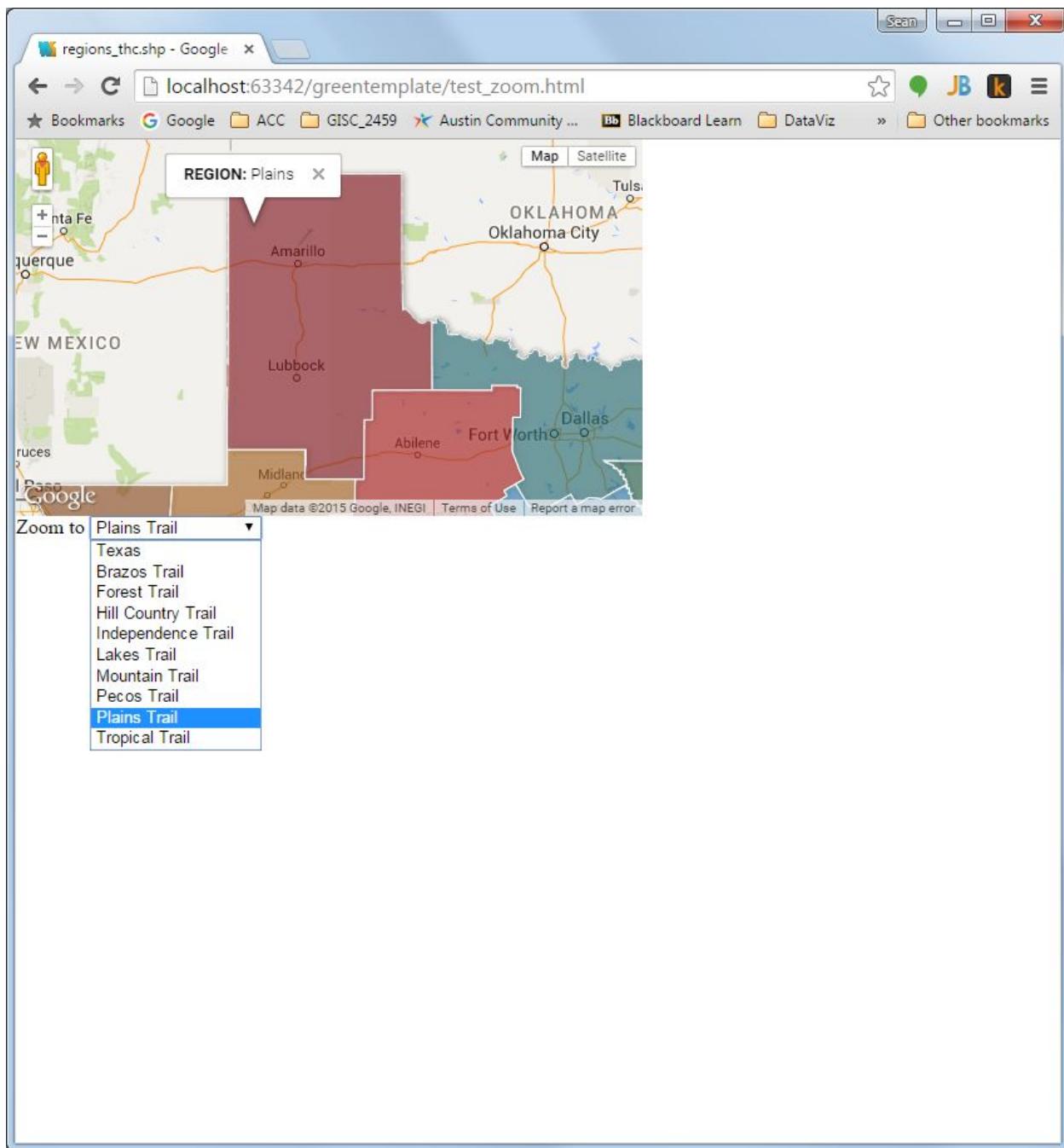
In WebStorm, the test_zoom.html file should now look similar to the following image.

```
//Zoom to centroid of Texas or selected region
function ZoomRegion() {
    var option = document.getElementById('region').value;
    if (option=="Texas") {map.setCenter(TexasCentroid);
        map.setZoom(5);
        return}
    else if (option=="Brazos") {map.setCenter(BrazosCentroid)}
    else if (option=="Forest") {map.setCenter(ForestCentroid)}
    else if (option=="Forts") {map.setCenter(FortsCentroid)}
    else if (option=="HillCountry") {map.setCenter(HillCountryCentroid)}
    else if (option=="Independence") {map.setCenter(IndependenceCentroid)}
    else if (option=="Lakes") {map.setCenter(LakesCentroid)}
    else if (option=="Mountain") {map.setCenter(MountainCentroid)}
    else if (option=="Pecos") {map.setCenter(PecosCentroid)}
    else if (option=="Plains") {map.setCenter(PlainsCentroid)}
    else if (option=="Tropical") {map.setCenter(TropicalCentroid)}
    map.setZoom (6);
}

google.maps.event.addDomListener(window, 'load', initialize);
</script>
</head>
<body>
<div id="googft-mapCanvas"></div>
<!--Insert Zoom to region drop-down box here-->
<div>
    Zoom to
    <select id="region" onchange="ZoomRegion()">
        <option value="Texas">Texas</option>
        <option value="Brazos">Brazos Trail</option>
        <option value="Forest">Forest Trail</option>
        <option value="HillCountry">Hill Country Trail</option>
        <option value="Independence">Independence Trail</option>
        <option value="Lakes">Lakes Trail</option>
        <option value="Mountain">Mountain Trail</option>
        <option value="Pecos">Pecos Trail</option>
        <option value="Plains">Plains Trail</option>
        <option value="Tropical">Tropical Trail</option>
    </select>
</div>
</body>
</html>
```

test_zoom.html with inserted HTML form <select> and JavaScript function <script> code blocks

7. View test_zoom.html in an Internet browser to confirm that the drop-down box is inserted properly and works correctly. The default position and format for the drop-down box looks good, so you will not add any CSS styling.

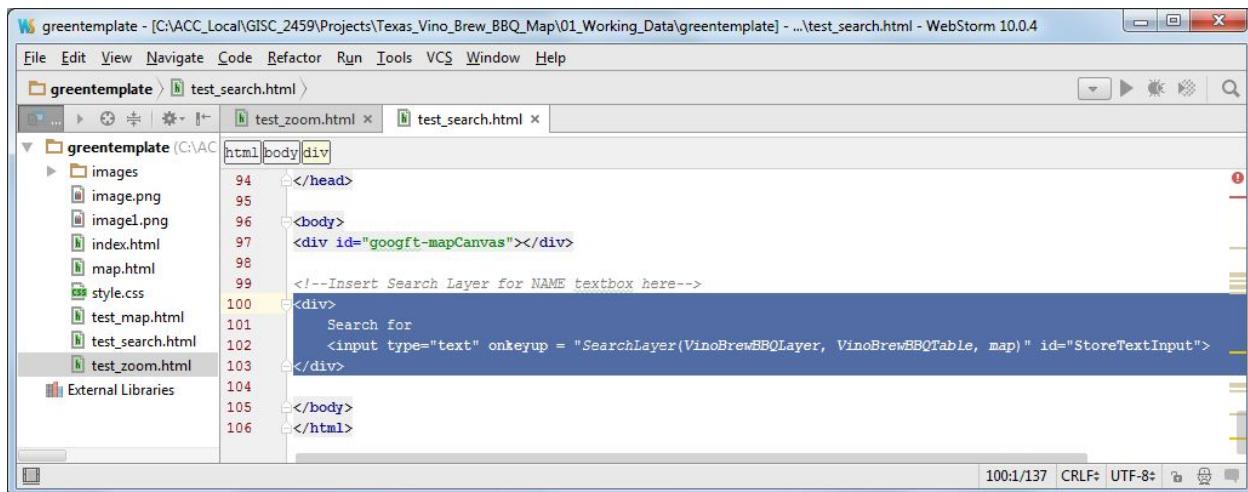


The working ZoomRegion() function in an Internet browser

8. This is a good point to backup your work.

Create Search Layer textbox function

1. Using WebStorm, create a new “test_search” HTML file in your website directory and replace the starter code with the Merge of Vino_Brew_BBQ and Markers Google Fusion Table HTML and JavaScript code for the Vino, Brew, and BBQ Map.
2. Open the newly created test_search.html file in the WebStorm Editor window. The general approach for creating the custom function will be:
 - a. Add the HTML Form Input element
 - b. Add the variable declarations and JavaScript function
 - c. Add the CSS styling
3. To add the HTML Form search textbox element - type in the highlighted HTML code block into the test_search.html file <body> element as shown in the following image.



The screenshot shows the WebStorm IDE interface. The title bar reads "greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_search.html - WebStorm 10.0.4". The menu bar includes File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help. The toolbars show standard icons for file operations. The left sidebar displays a project tree with files like greentemplate, images, index.html, map.html, style.css, test_map.html, test_search.html, and test_zoom.html. The main code editor pane shows the following HTML code:

```
html
  body
    div
      94   </head>
      95
      96   <body>
      97     <div id="googft-mapCanvas"></div>
      98
      99   <!--Insert Search Layer for NAME textbox here-->
     100   <div>
     101     Search for
     102     <input type="text" onkeyup = "SearchLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map)" id="StoreTextInput">
     103   </div>
     104
     105   </body>
     106 </html>
```

The code editor has syntax highlighting for HTML and CSS. The status bar at the bottom right shows "100:1/137 CRLF UTF-8".

The HTML Form Input textbox element calls the JavaScript SearchLayer() function when text is entered

In the HTML code block:

- o <div> tag identifies a new division in the HTML <body> element;
- o Search for labels the search textbox;
- o <input> tag creates a form input and the type attribute identifies the type of input (e.g. text field, password field, radio buttons, checkboxes, etc.);
- o id attribute assigns the textbox an ID;
- o onkeyup attribute calls the to-be-created SearchLayer() function with arguments VinoBrewBBQLayer, VinoBrewBBQTable, and map.

View test_search.html in an Internet browser to confirm that the textbox is inserted properly. If you enter text, nothing happens because you haven't added the variable declarations and JavaScript function yet. You'll do that next.

4. To add the JavaScript variable declarations - type in the highlighted JavaScript code block near the beginning of the test_search.html file <script> element just before the initialize() function as shown in the following image. Don't forget to delete the "var" portion of the 'var map' declaration within the initialize() function so that it is not redeclared as a local variable and therefore inaccessible to the SearchLayer() function.

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport"></meta>
    <title>Merge of Vino_Brew_BBQ and Markers - Google Fusion Tables</title>
    <style type="text/css">
        html, body, #googft-mapCanvas {
            height: 300px;
            margin: 0;
            padding: 0;
            width: 500px;
        }
    </style>
    <script type="text/javascript" src="https://maps.google.com/maps/api/js?sensor=false&v=3"></script>
    <script type="text/javascript">
        var map; //Initialize new Google Map
        var VinoBrewBBQLayer; //Merge of Vino_Brew_BBQ and Markers Google Fusion Table Map Overlay
        var VinoBrewBBQTable = 'ENTER_FUSION_TABLE_UNIQUE_ID_HERE'; //Merge of Vino_Brew_BBQ and Markers Google Fusion Table
        var VinoBrewBBQColumn = 'geometry'; //Merge of Vino_Brew_BBQ and Markers Google Fusion Table column

        function initialize() {
            google.maps.visualRefresh = true;
            var isMobile = (navigator.userAgent.toLowerCase().indexOf('android') > -1) ||
                (navigator.userAgent.match(/(iPod|iPhone|iPad|BlackBerry|Windows Phone|iemobile)/));
            if (isMobile) {
                var viewport = document.querySelector("meta[name=viewport]");
                viewport.setAttribute('content', 'initial-scale=1.0, user-scalable=no');
            }
            var mapDiv = document.getElementById('googft-mapCanvas');
            mapDiv.style.width = isMobile ? '100%' : '500px';
            mapDiv.style.height = isMobile ? '100%' : '300px';
            map = new google.maps.Map(mapDiv, {
                center: new google.maps.LatLng(31.198951673731063, -100.19955100000004),
                zoom: 10
            });
            VinoBrewBBQLayer = new google.maps.FusionTablesLayer({
                query: {
                    sql: "SELECT * FROM " + VinoBrewBBQTable
                },
                styleId: VinoBrewBBQColumn
            });
            VinoBrewBBQLayer.setMap(map);
        }
    </script>

```

The JavaScript global variable declarations with Fusion Table unique ID

5. To add the JavaScript Search Layer function - type in the highlighted JavaScript code block near the end of the test_search.html file <script> element after the initialize() function's closing curly bracket (i.e. }) as shown in the following image.

```

greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_search.html - ...
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_search.html
greentemplate (C:\ACC_L...
html head script
68     legendCloseButton.onclick = function() {
69         legend.style.display = 'none';
70         legendOpenButton.style.display = 'block';
71     }
72 }
73 }
74 }
75 //Search VinoBrewBBQLayer based using textbox input.
76 function SearchLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map){
77     var store = document.getElementById('StoreTextInput').value;
78     var search = "NAME CONTAINS IGNORING CASE '" + store + "'";
79
80     if (!VinoBrewBBQLayer.getMap()){
81         VinoBrewBBQLayer.setMap(map);
82     }
83     VinoBrewBBQLayer.setOptions({
84         query: {
85             select: VinoBrewBBQColumn,
86             from: VinoBrewBBQTable,
87             where: search
88         },
89     });
90 }
91
92 google.maps.event.addDomListener(window, 'load', initialize);
93 </script>
94 </head>
95
96 <body>
97 <div id="googft-mapCanvas"></div>
98
99 <!--Insert Search Layer for NAME textbox here-->
100 <div>
    Search ...

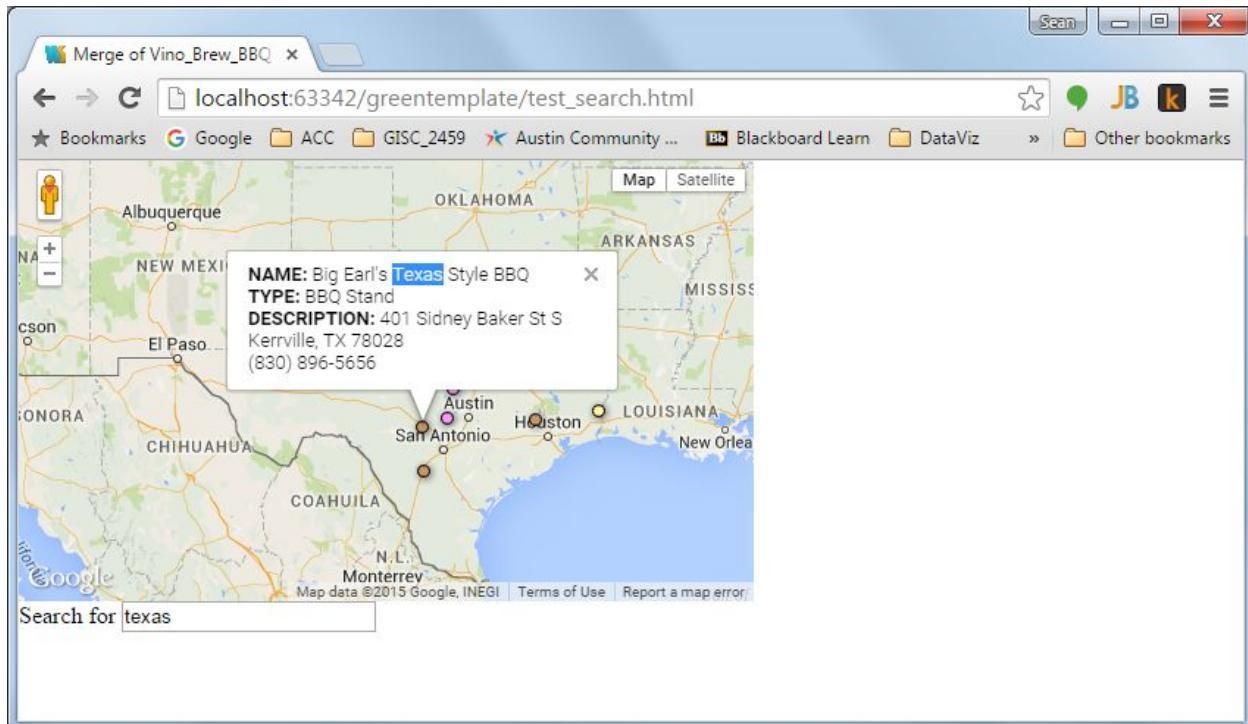
```

The JavaScript `SearchLayer()` function with Fusion Table where statement to filter on entered text

In the JavaScript code block:

- `function SearchLayer()` declares the JavaScript custom function with arguments `VinoBrewBBQLayer`, `VinoBrewBBQTable`, `map`;
- `var store` declares ‘store’ as a local variable and sets it equal to the `document.getElementById` object and `value` property, in this case the ‘StoreTextInput’ value entered into the search textbox;
- `var search` declares ‘search’ as a local variable and sets it equal to a where clause that will query the Fusion Table on the entered text while ignoring case;
- `if` conditional statement executes a Google Maps API command to turn the `VinoBrewBBQLayer` on if it is turned off; and
- `VinoBrewBBQLayer.setOptions` executes a Google Maps API command to `query` the `VinoBrewBBQTable` on the `where` search clause and display the results as styled in the Fusion Table Vino, Brew, and BBQ Map tab (i.e. `options: {styleID: 3, templateID: 3}`).

6. View test_search.html in an Internet browser to confirm that the search textbox is inserted properly and works correctly.

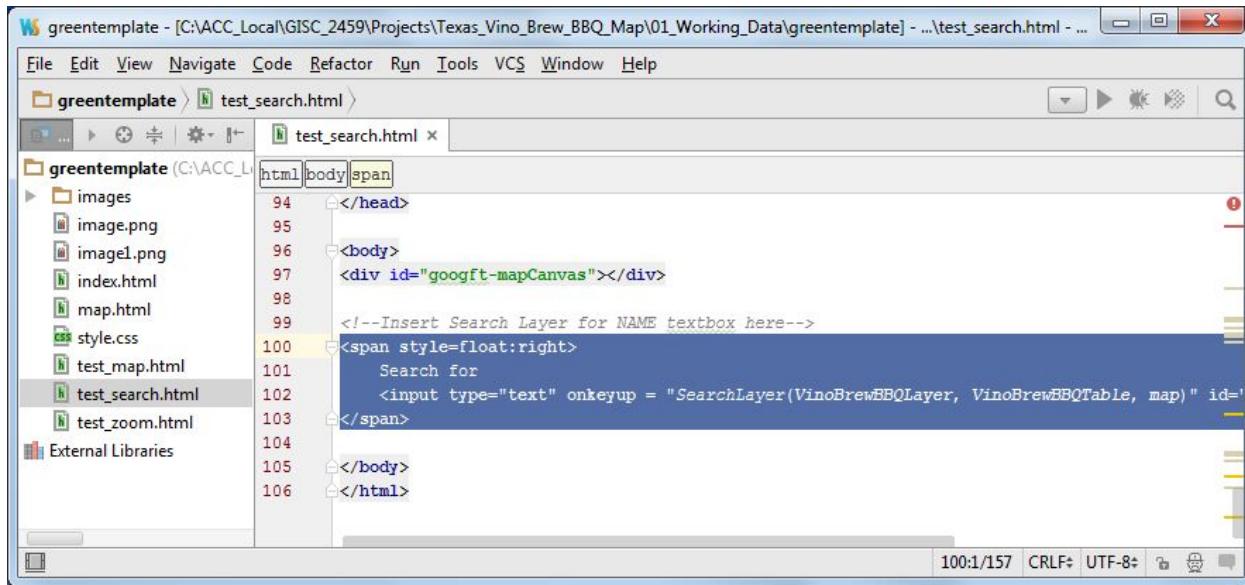


The working SearchLayer() function in an Internet browser

7. To add the CSS styling - replace the <div> tags with the highlighted HTML (i.e.) and CSS (i.e. style=float:right) code as shown in the following image.

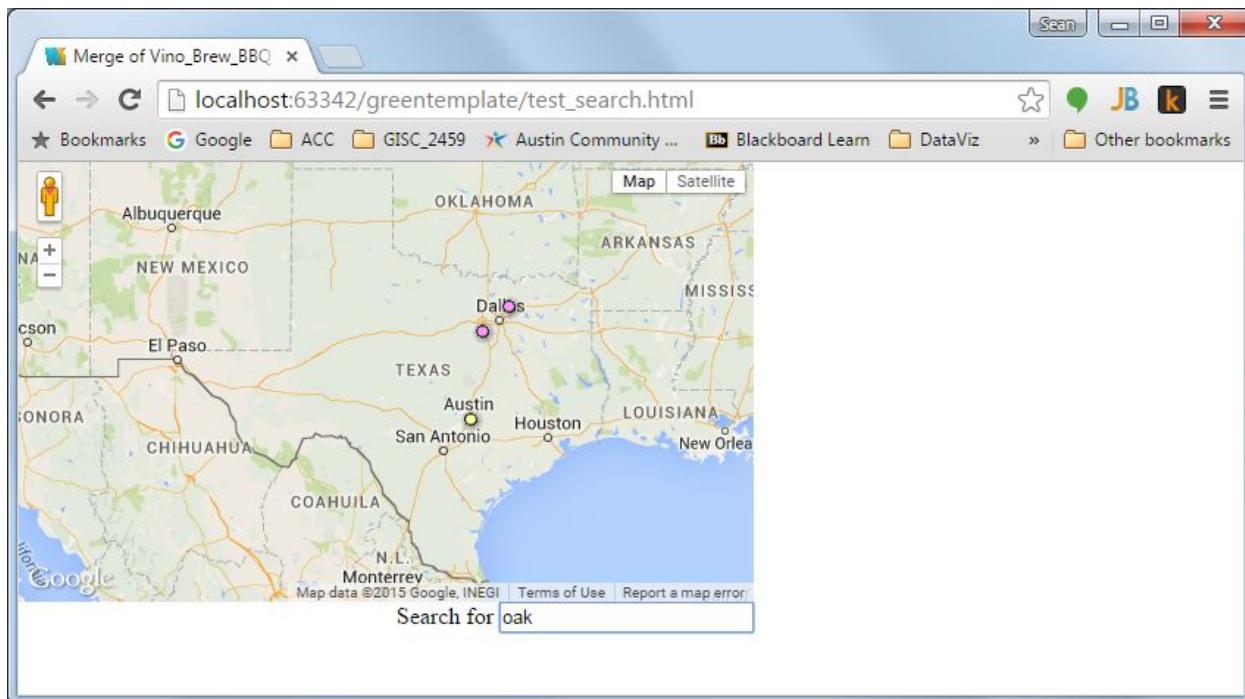
In the HTML and CSS code block:

- identifies a new division that will be inserted in-line with the previous division. This will allow the Zoom to Region drop-down box and the Search Layer textbox to be displayed on the same line when the custom functions are both inserted into the final embedded map.
- style attribute allows inline CSS code to be inserted within HTML code; and
- float:right property aligns the Search Layer textbox to the right.



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_search.html
greentemplate (C:\ACC_L...
html body span
94     </head>
95
96     <body>
97         <div id="googft-mapCanvas"></div>
98
99         <!--Insert Search Layer for NAME textbox here-->
100        <span style="float:right">
101            Search for
102            <input type="text" onkeyup = "SearchLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map)" id=''
103        </span>
104
105    </body>
106</html>
```

The CSS style added inline to the HTML code

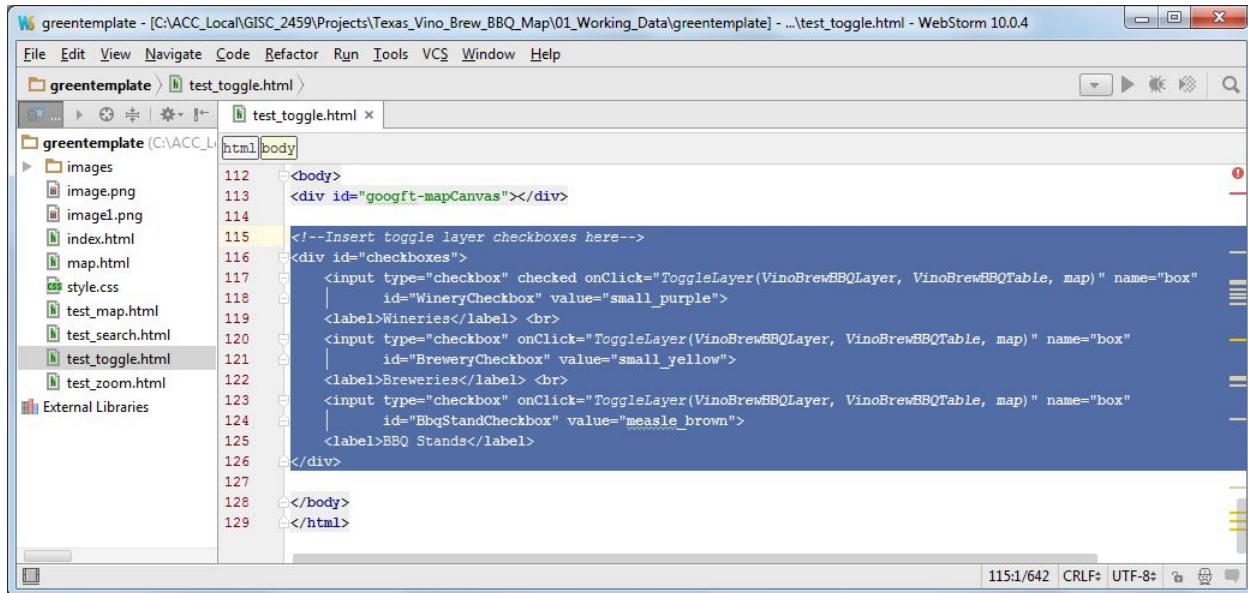


test_search.html with CSS formatted Search Map function

8. This is a good point to backup your work.

Create Toggle Layer checkbox function

1. Using WebStorm, create a new “test_toggle” HTML file in your website directory and replace the starter code with the Merge of Vino_Brew_BBQ and Markers Google Fusion Table HTML and JavaScript code for the Vino, Brew, and BBQ Map.
2. Open the newly created test_toggle.html file in the WebStorm Editor window. The general approach for creating the custom function will be:
 - a. Add the HTML Form Input element
 - b. Add the variable declarations and JavaScript function
 - c. Add the CSS styling
3. To add the HTML Form input checkbox element - type in the highlighted HTML code block into the test_toggle.html file <body> element as shown in the following image.



```
112 <body>
113   <div id="googft-mapCanvas"></div>
114
115   <!--Insert toggle layer checkboxes here-->
116   <div id="checkboxes">
117     <input type="checkbox" checked onClick="ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map)" name="box"
118       | id="WineryCheckbox" value="small_purple">
119     <label>Wineries</label> <br>
120     <input type="checkbox" onClick="ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map)" name="box"
121       | id="BreweryCheckbox" value="small_yellow">
122     <label>Breweries</label> <br>
123     <input type="checkbox" onClick="ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map)" name="box"
124       | id="BbqStandCheckbox" value="measle_brown">
125     <label>BBQ Stands</label>
126   </div>
127
128 </body>
129 </html>
```

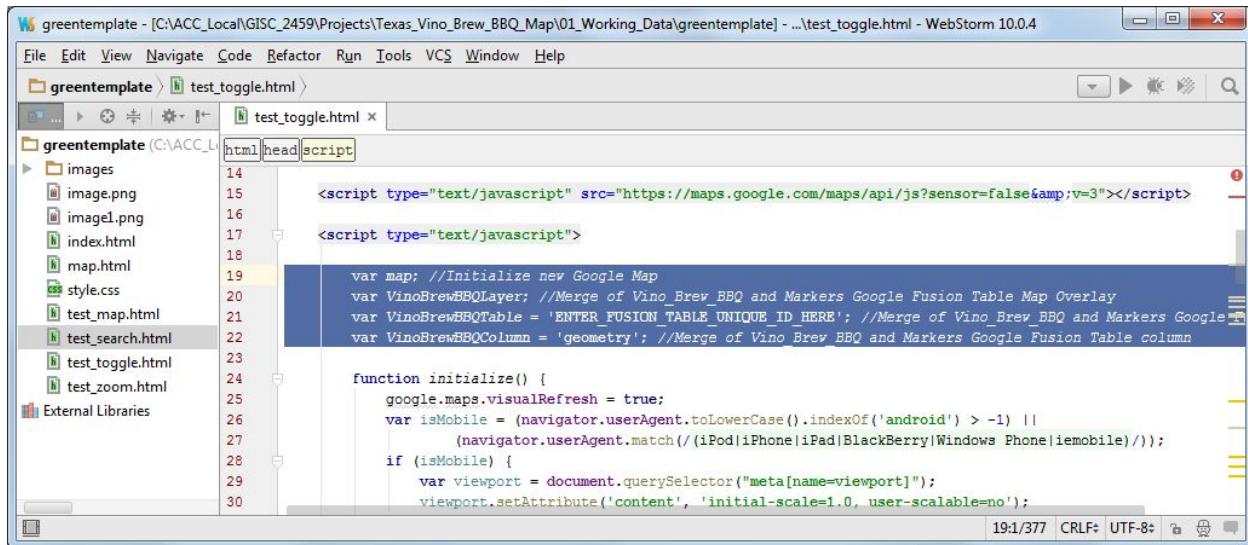
The HTML Form Input checkbox element with Wineries, Breweries, and BBQ Stands options calls the `ToggleLayer()` function when checked or unchecked

In the HTML code block:

- o `<div>` tag identifies a new division in the HTML `<body>` element;
- o `<input>` tag creates a form input and the `type` attribute identifies the type of input (e.g. text field, password field, radio buttons, checkboxes, etc.);
- o `checked` attribute designates a box as checked on load;
- o `onClick` calls the to-be-created `ToggleLayer()` function with arguments `VinoBrewBBQLayer`, `VinoBrewBBQTable`, and `map`;
- o `name` attribute assigns the entire Form input group a common name;
- o `id` attribute assigns each checkbox an ID;
- o `value` identifies a property, in this case the selected checkbox value; and
- o `<label>` displays the label next to each checkbox.

View test_toggle.html in an Internet browser to confirm that the checkboxes are inserted properly. If you check a checkbox, nothing happens because you haven't added the variable declarations and JavaScript function yet. You'll do that next.

4. To add the JavaScript variable declarations - type in the highlighted JavaScript code block near the beginning of the test_search.html file <script> element just before the initialize() function as shown in the following image. Remember to update the variables within your code to match your declarations.



```
greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_toggle.html - WebStorm 10.0.4
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate / test_toggle.html
greentemplate (C:\ACC_L...
html head script
14
15     <script type="text/javascript" src="https://maps.google.com/maps/api/js?sensor=false&v=3"></script>
16
17     <script type="text/javascript">
18
19         var map; //Initialize new Google Map
20         var VinoBrewBBQLayer; //Merge of Vino_Brew_BBQ and Markers Google Fusion Table Map Overlay
21         var VinoBrewBBQTable = 'ENTER_FUSION_TABLE_UNIQUE_ID_HERE'; //Merge of Vino_Brew_BBQ and Markers Google Fusion Table
22         var VinoBrewBBQColumn = 'geometry'; //Merge of Vino_Brew_BBQ and Markers Google Fusion Table column
23
24     function initialize() {
25         google.maps.visualRefresh = true;
26         var isMobile = (navigator.userAgent.toLowerCase().indexOf('android') > -1) ||
27             (navigator.userAgent.match(/(ipod|iphone|ipad|blackberry|windows phone|iemobile)/));
28         if (isMobile) {
29             var viewport = document.querySelector("meta[name=viewport]");
30             viewport.setAttribute('content', 'initial-scale=1.0, user-scalable=no');
```

The JavaScript global variable declarations with Fusion Table unique ID

5. To add the JavaScript Toggle Layer function - type in the highlighted JavaScript code block near the end of the test_toggle.html file <script> element as shown in the following image.

```

W greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_to...
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_toggle.html
greentemplate (C:\ACC_L...
html head script
62 }
63
64 //Filter VinoBrewBBQLayer based on checkbox selection.
65 function ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map){
66     var where = generateWhere();
67
68     if (where){
69         if (!VinoBrewBBQLayer.getMap()){
70             VinoBrewBBQLayer.setMap(map);
71         }
72         VinoBrewBBQLayer.setOptions({
73             query: {
74                 select: VinoBrewBBQColumn,
75                 from: VinoBrewBBQTable,
76                 where: where
77             },
78             options: {
79                 styleId: 3,
80                 templateId: 3
81             }
82         });
83     }else{
84         VinoBrewBBQLayer.setMap(null);
85     }
86
87     google.maps.event.addDomListener(window, 'load', initialize);
88     </script>
89 </head>
90
91

```

The JavaScript `ToggleLayer()` function with Fusion Table where statement to filter on checked boxes

In the JavaScript code block:

- `function ToggleLayer()` declares the JavaScript custom function with arguments `VinoBrewBBQLayer`, `VinoBrewBBQTable`, `map`;
- `var where` declares ‘where’ as a local variable and sets it equal to the to-be-created `generateWhere()` function;
- `if...else` conditional statement toggles the checked layers if ‘where’ is not null and turns the entire VinoBrewBBQLayer off if ‘where’ is null. The additional `if` statement ensures the VinoBrewBBQLayer is turned on, before toggling the checked layers;
- `VinoBrewBBQLayer.setOptions` executes a Google Maps API command to `query` the `VinoBrewBBQTable` on the `where` search clause and display the results as styled in the Fusion Table Vino, Brew, and BBQ Map tab (i.e. `options: {styleID: 3, templateID: 3}`); and
- `VinoBrewBBQLayer.setMap(map)` and `VinoBrewBBQLayer.setMap(null)` execute Google Maps API commands to turn the entire Fusion Table layer on and off.

6. The ToggleLayer() function calls the to-be-created generateWhere() function. When map users check a Winery, Brewery, and/or BBQ Stands checkbox layer toggles, there are any number of combinations they could select. The generateWhere() function determines which boxes are checked and then generates a corresponding ‘where’ clause that is returned to the ToggleLayer() function. For example, if Wineries and Breweries are checked and BBQ Stands is unchecked, the generateWhere() function will set where = ‘MARKER’ IN (‘small_purple’, ‘small_yellow’). It would be more logical to query the TYPE column, but the [Google Fusion Table API v1.0](#) does not allow you to [query a column that is used as the primary key in a merge](#) (i.e. TYPE).

To add the JavaScript generateWhere() function - type in the highlighted JavaScript code block after the ToggleMap() function and before the google.maps.event.addListener(window, 'load', initialize) event near the end of the test_toggle.html file <script> element as shown in the following image.

In the JavaScript code block:

- function generateWhere() declares the JavaScript custom function;
- var filter and var boxes are declared as a local arrays, and boxes is set equal to the array of three checkboxes named box;
- where is declared as a local variables;
- for creates a loop to determine if each of the three checkboxes is checked (e.g. var i = 0, box, box = boxes[i]);
- if(box.checked) conditional statement determines if each checkbox is checked, and if so, builds a where statement using the local array filter;
- if(filter.length) conditional statement is initiated once the for loop is complete and sets the where variable equal to ‘MARKER’ IN plus the filter array, but only if filter.length is not null.
- where is then returned to the ToggleLayer() function to serve as the table query where statement.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_toggle.html
greentemplate (C:\ACC_L...
html head script
62 } ...
63 ...
64 //Filter VinoBrewBBQLayer based on checkbox selection.
65 function ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map){
66     var where = generateWhere();
67
68     if (where){
69         if (!VinoBrewBBQLayer.getMap()){
70             VinoBrewBBQLayer.setMap(map);
71         }
72         VinoBrewBBQLayer.setOptions({
73             query: {
74                 select: VinoBrewBBQColumn,
75                 from: VinoBrewBBQTable,
76                 where: where
77             },
78             options: {
79                 styleId: 3,
80                 templateId: 3
81             }
82         });
83     }else{
84         VinoBrewBBQLayer.setMap(null);
85     }
86 }
87
88 // Generate a where clause from the checkboxes. If no boxes
89 // are checked, return an empty string.
90 function generateWhere() {
91     var filter = [];
92     var boxes = document.getElementsByName('box');
93     var where;
94
95     for (var i = 0, box; box = boxes[i]; i++) {
96         if (box.checked) {
97             var boxName = box.value.replace(/\//g, '\\\\');
98             filter.push("'" + boxName + "'");
99         }
100    }
101    if (filter.length) {
102        where = "'MARKER' IN (" + filter.join(',') + ')';
103    }
104    return where;
105 }
106
107 google.maps.event.addDomListener(window, 'load', initialize);
108 </script>
109 </head>
110
```

The JavaScript `generateWhere()` function listed after the `ToggleLayer()` function

7. In the original JavaScript and HTML code copied from the Vino, Brew, and BBQ map tab in the Merge of Vino_Brew_BBQ and Markers Fusion Table; the Google Maps API is initialized, the base map is set (e.g. ROADMAP); and the Fusion Table layer is turned on with no query filter (i.e. where: ""). When no records are selected, the Google Maps API defaults to displaying all the records. The original JavaScript code block will display the Google Map with the Breweries, Wineries, and BBQ Stands all toggled on.

To declutter the map and sync the initial map load with the checkboxes, replace the original JavaScript code block that turns the VinoBrewBBQLayer on with the two lines of JavaScript code as shown in the following image.

The image shows two side-by-side code editors. Both have the same file open: 'test_toggle.html'. The left editor shows the full original JavaScript code, which is very long and includes logic for initializing a map, setting its type to 'ROADMAP', and adding a 'VinoBrewBBQLayer' as a FusionTablesLayer. The right editor shows the modified code, where the original block has been replaced by two simple lines of code:

```

function initialize() {
  google.maps.visualRefresh = true;
  var isMobile = (navigator.userAgent.toLowerCase().indexOf('android') > -1 || navigator.userAgent.match(/(ipod|iphone|ipad|blackberry|iemobile)/));
  if (isMobile) {
    var viewport = document.querySelector("meta[name=viewport]");
    viewport.setAttribute('content', 'initial-scale=1.0, user-scalable=no');
  }
  var mapDiv = document.getElementById('googft-mapCanvas');
  mapDiv.style.width = isMobile ? '100%' : '500px';
  mapDiv.style.height = isMobile ? '100%' : '300px';
  map = new google.maps.Map(mapDiv, {
    center: new google.maps.LatLng(31.198951673731063, -100.199545),
    zoom: 5,
    mapTypeId: google.maps.MapTypeId.ROADMAP
  });
  map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(document.createElement('div'));
  map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(document.createElement('div'));
  VinoBrewBBQLayer = new google.maps.FusionTablesLayer();
  ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map);
}
if (isMobile) {
}

```

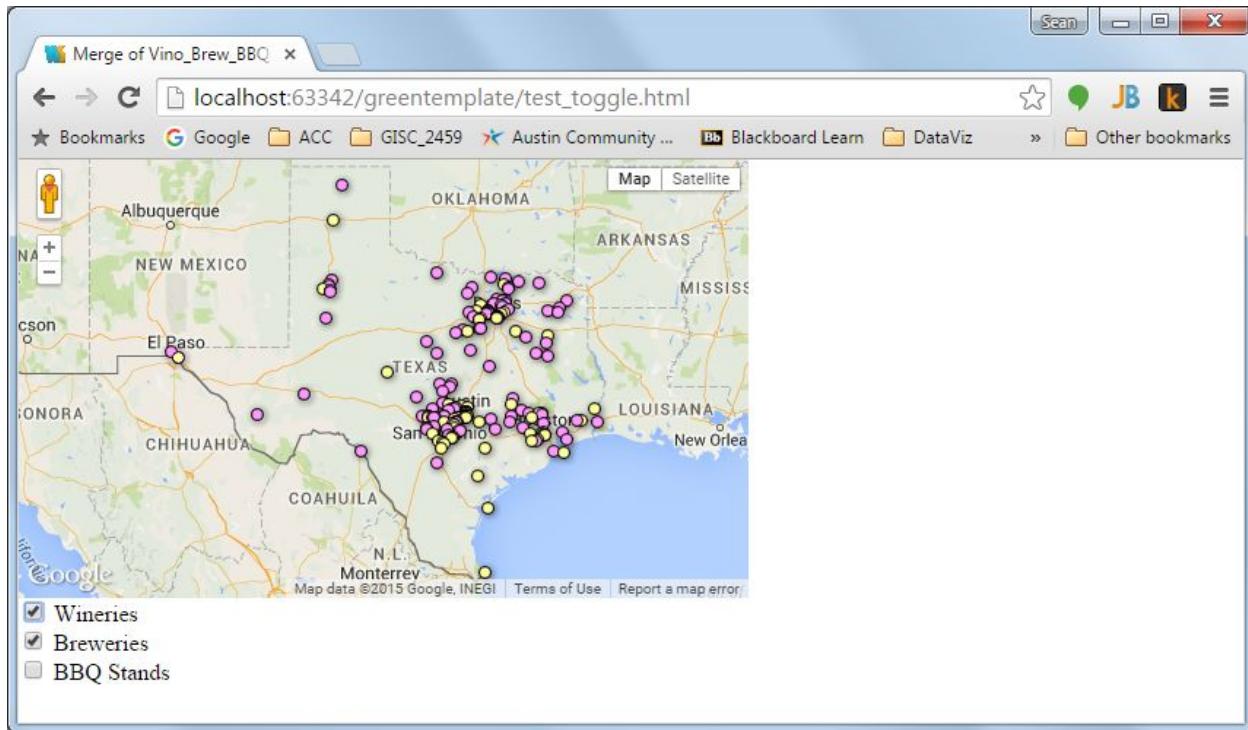
Replace the original JavaScript code block that loads the VinoBrewBBQLayer with two lines that turn the layer on and set the initial layer query and options based on the default toggle checkbox status

In the JavaScript code block:

- VinoBrewBBQLayer = new google.maps.FusionTablesLayer() turns the VinoBrewBBQLayer on; and
- ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map) calls the

ToggleLayer() function and sets the initial layer query and options based on the default toggle checkbox status. In this case, Wineries is checked and Breweries and BBQ Stands are unchecked...so the Google Map will initially load with only the wineries displayed. This will declutter the map and intuitively suggest to map users that they can turn layers on and off.

8. View test_toggle.html in an Internet browser to confirm that the toggle layer checkboxes are inserted and work correctly.



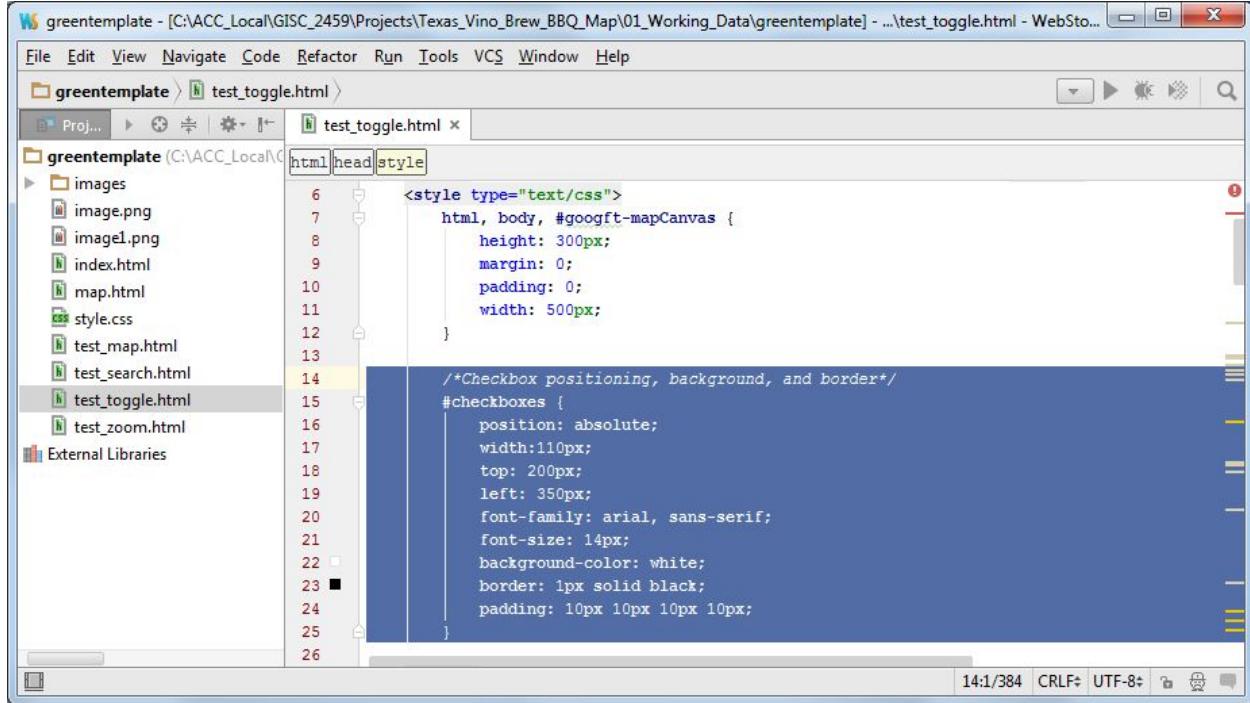
The working ToggleLayer() function in an Internet browser

9. The position of the Toggle Layer function conflicts with the Zoom to Region function. To prevent a conflict, you'll use CSS styling to overlay the checkboxes on the Google Map framed with a border and background. To add the CSS styling - insert the CSS code near the end of the HTML <style> element as shown in the image below.

In the CSS code block:

- o #checkboxes flags the following CSS styling to be applied to the HTML element with id="checkboxes";
- o position: absolute positions the checkbox element relative to the first parent element that has a position other than static. If no such element is found, the containing block is <html>;
- o width sets the width of the checkbox element
- o top and left set the position offset, in the case of an absolute position, from the parent element;

- font-family sets the checkbox labels default font;
- font-size sets the label font size;
- background-color sets the checkbox background color; and
- border and padding set the checkbox outline and margin.

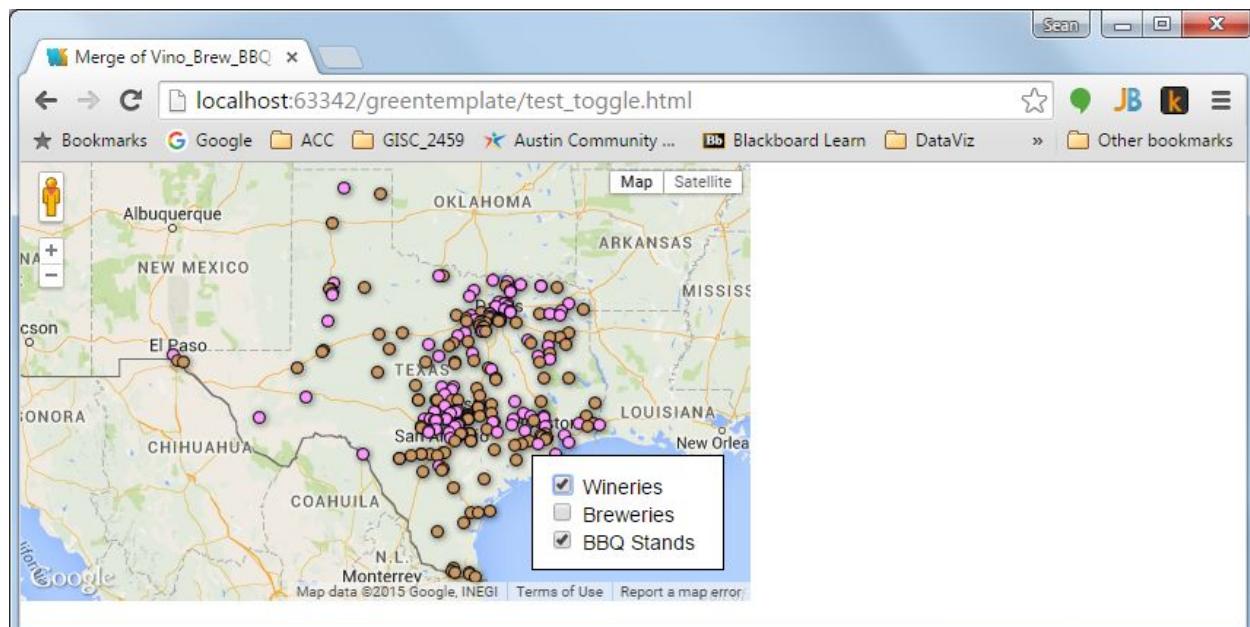


```

<style type="text/css">
  html, body, #googft-mapCanvas {
    height: 300px;
    margin: 0;
    padding: 0;
    width: 500px;
  }
  /*Checkbox positioning, background, and border*/
  #checkboxes {
    position: absolute;
    width:110px;
    top: 200px;
    left: 350px;
    font-family: arial, sans-serif;
    font-size: 14px;
    background-color: white;
    border: 1px solid black;
    padding: 10px 10px 10px 10px;
  }

```

The CSS style added as an internal stylesheet to the HTML code



test_toggle with CSS formatted Toggle Layer function

7. Add Customized Google Map Functions

Now that you've created and successfully tested each of the requested custom Google Map functions, you'll add the working code blocks to your pilot website.

Add customized Google Map functions to test_map.html

1. Open test_map.html and map.html in the WebStorm Editor.
2. Replace all of the code in test_map.html with the code from map.html. Now close map.html and leave test_map.html open. You will use test_map.html to combine the three custom Google Map functions and validate they are working before adding them to the pilot website. The general approach for adding each custom function will be:
 - a. Add the HTML Form Input element
 - b. Add the variable declarations and JavaScript function
 - c. Add the CSS styling
3. Add the Zoom to Region function:
 - a. Copy the test_zoom.html HTML Form Input <div> element into the test_map.html <body> element just below the Google Maps API division element (i.e. <div id="googft-mapCanvas"></div>).
 - b. Copy the ZoomRegion() JavaScript variables into the test_map.html just above the initialize() function. Remember that within the initialize() function, you should remove the local variable "var" declaration from the "map" variable and replace the values for the regions_thc.shp Fusion Table column and ID with RegionsColumn and RegionsTable global variables as shown in the image below. Copy the ZoomRegion() JavaScript function into the test_map.html <script> element just below the initialize() function.
 - c. There isn't any CSS styling code associated with the Zoom to Region function, so there isn't any to paste into test_map.html.

You may wish to adjust the map.setZoom value for the regions depending on the size (i.e. width and height) of your Google Map. View test_map.html in an Internet browser to confirm that the Zoom to Region function is inserted and works correctly.

greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\test_map.html - WebSt... 

File Edit View Navigate Code Refactor Run Tools VCS Window Help

greentemplate > test_map.html

Proj... test_map.html

greentemplate (C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate)

images
image.png
image1.png
index.html
map.html
style.css
test_map.html
test_search.html
test_toggle.html
test_zoom.html
External Libraries

```

10
11      <script type="text/javascript">
12
13          var map; //Initialize new Google Map
14          var RegionsLayer; //Regions Google Fusion Table Map Overlay
15          var RegionsTable = '10lw5Dg1krdju0hrulnZqv1DBE3ueMNEKxQeYpph'; //Regions Google Fusion Table
16          var RegionsColumn = 'geometry'; //Regions Google Fusion Table column
17
18          //google.maps.LatLng class variables with centroid for Texas and each region
19          var TexasCentroid = new google.maps.LatLng(31.206164171824494, -99.98919937500006);
20          var BrazosCentroid = new google.maps.LatLng(31.0249185, -96.7867203428);
21          var ForestCentroid = new google.maps.LatLng(31.733945, -94.8448138846);
22          var FortsCentroid = new google.maps.LatLng(31.9826015, -99.449775469);
23          var HillCountryCentroid = new google.maps.LatLng(30.0381825, -98.8332940097);
24          var IndependenceCentroid = new google.maps.LatLng(29.2581915, -96.5145527265);
25          var LakesCentroid = new google.maps.LatLng(33.023145, -96.6150644396);
26          var MountainCentroid = new google.maps.LatLng(30.4871465, -104.026028443);
27          var PecosCentroid = new google.maps.LatLng(30.361035, -101.153074449);
28          var PlainsCentroid = new google.maps.LatLng(34.2930355, -101.259543199);
29          var TropicalCentroid = new google.maps.LatLng(27.312045, -98.4054598713);
30
31      function initialize() {
32          google.maps.visualRefresh = true;
33          var isMobile = (navigator.userAgent.toLowerCase().indexOf('android') > -1) ||
34              (navigator.userAgent.match(/(iPod|iPhone|iPad|BlackBerry|Windows Phone)/));
35          if (isMobile) {
36              var viewport = document.querySelector("meta[name=viewport]");
37              viewport.setAttribute('content', 'initial-scale=1.0, user-scalable=no');
38          }
39          var mapDiv = document.getElementById('googft-mapCanvas');
40          mapDiv.style.width = isMobile ? '100%' : '490px';
41          mapDiv.style.height = isMobile ? '100%' : '490px';
42          map = new google.maps.Map(mapDiv, {
43              center: new google.maps.LatLng(31.206164171824494, -99.98919937500006),
44              zoom: 5,
45              mapTypeId: google.maps.MapTypeId.ROADMAP
46          });
47          map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(document.getElemen
48          map.controls[google.maps.ControlPosition.RIGHT_BOTTOM].push(document.getElemen
49
50          //Add Fusion Table Regions Map overlay
51          RegionsLayer = new google.maps.FusionTablesLayer({
52              map: map,
53              heatmap: { enabled: false },
54              query: {
55                  select: RegionsColumn,
56                  from: RegionsTable,
57                  where: ""
58              },
59              options: {
60                  styleId: 3,
61                  templateId: 3
62          });
63      }
64  
```

13:1/1288 CRLF# UTF-8   

ZoomRegion() function global variable declarations and use within the initialize() function in test.html

4. Add the Search Layer function:

- a. Copy the test_search.html HTML Form Input element and insert it into the Zoom to Region <div> element just before the closing </div> tag in test_map.html as shown in the following image. This will display the two functions on the same line in an Internet browser.
- b. Copy the SearchLayer() JavaScript variables into the test_map.html just above the initialize() function. Remember to replace the values for the Fusion Table column and ID with VinoBrewBBQColumn and VinoBrewBBQTable global variables within the initialize() function. Copy the SearchLayer() JavaScript function into the test_map.html <script> element just below the ZoomRegion() function.
- c. The HTML inline CSS styling code was included in the HTML Form Input elements you already copied and pasted in Step a. above.

View test_map.html in an Internet browser to confirm that the Search Layer function is inserted and works correctly.

```
<!--Insert Google Maps API here-->
<div id="googft-mapCanvas"></div>

<!--Insert Zoom to region drop-down box here-->
<div>
    Zoom to
    <select id="region" onchange="ZoomRegion()">
        <option value="Texas">Texas</option>
        <option value="Brazos">Brazos Trail</option>
        <option value="Forest">Forest Trail</option>
        <option value="HillCountry">Hill Country Trail</option>
        <option value="Independence">Independence Trail</option>
        <option value="Lakes">Lakes Trail</option>
        <option value="Mountain">Mountain Trail</option>
        <option value="Pecos">Pecos Trail</option>
        <option value="Plains">Plains Trail</option>
        <option value="Tropical">Tropical Trail</option>
    </select>
<!--Insert Search Layer for NAME textbox here-->
<span style=float:right>
    Search for
    <input type="text" onkeyup = "SearchLayer(VinoBrewBBQLayer, VinoBrewBBQTable)">
</span>
</div>

<p>Non facilius quae sita uno originem. Conflantur me mo ob scripturas in divis</p>
<p>Excludere im sapientia evidenter et delusisse. Externarum vi requiratur in</p>
```

Insert the Search Layer element into the Zoom to Region <div> element

5. Add the Toggle Layer function:

- Copy the test_toggle.html HTML Form Input <div> element into the test_map.html <body> element below the Google Maps API division element (i.e. <div id="googft-mapCanvas"></div>).
- The ToggleLayer() JavaScript uses the same variables as the SearchLayer() function, which are already included in the test_map.html variable declarations. Copy the ToggleLayer() and generateWhere() JavaScript functions into the test_map.html <script> element just below the SearchLayer() function. Remember that in the original JavaScript and HTML code copied from the Fusion Table, the Google Maps API is initialized and the Fusion Table layer is turned on with no query filter (i.e. where: ""). To sync the initial map load with the toggle layer checkboxes, replace the original JavaScript code block that turns the VinoBrewBBQLayer on with the two lines of JavaScript code as shown in the following image.

The image shows two side-by-side screenshots of a code editor, likely WebStorm, displaying the file 'test_map.html'.

Screenshot 1 (Top): Shows the original code where the VinoBrewBBQLayer is added directly to the map. The code is as follows:

```

82
83
84 //Add Fusion Table Vino, Brew, and BBQ Map overlay
85 VinoBrewBBQLayer = new google.maps.FusionTablesLayer({
86   map: map,
87   heatmap: { enabled: false },
88   query: {
89     select: VinoBrewBBQColumn,
90     from: VinoBrewBBQTable,
91     where: ""
92   },
93   options: {
94     styleId: 3,
95     templateId: 3
96   }
97 });
98
99 if (isMobile) {

```

Screenshot 2 (Bottom): Shows the modified code where the VinoBrewBBQLayer is created and then passed to the ToggleLayer function. The code is as follows:

```

82
83
84 // Add Fusion Table Vino, Brew, and BBQ Map overlay layer
85 // and sync the initial map load with ToggleLayer()
86 VinoBrewBBQLayer = new google.maps.FusionTablesLayer();
87 ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQtable, map);
88
89 if (isMobile) {

```

Replace the original JavaScript code that loads the VinoBrewBBQLayer with the JavaScript code to sync the initial layer load with the toggle layer checkboxes

- c. Copy the CSS styling `#checkboxes` code into `style.css` just after the Google Maps API CSS styling (i.e. `#googft-mapCanvas {}`) as shown in the following image.

The screenshot shows a code editor window with the following details:

- Title Bar:** "greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] ..."
- Menu Bar:** File, Edit, View, Navigate, Code, Refactor, Run, Tools, VCS, Window, Help
- Toolbar:** Includes icons for back, forward, search, and other standard file operations.
- Sidebar:** Shows a tree view of the project structure:
 - greentemplate (C:\ACC_...)
 - images
 - image.png
 - image1.png
 - index.html
 - map.html
 - style.css
 - test_map.html
 - test_search.html
 - test_toggle.html
 - test_zoom.html
- Code Editor:** The file "style.css" is open, containing the following CSS code:

```
1 html, body, #googft-mapCanvas {  
2     height: 100%;  
3     margin: 0;  
4     padding: 0;  
5     width: 100%;  
6 }  
7 /*Checkbox positioning, background, and border*/  
8 #checkboxes {  
9     position: absolute;  
10    width: 110px;  
11    top: 200px;  
12    left: 350px;  
13    font-family: arial, sans-serif;  
14    font-size: 14px;  
15    background-color: white;  
16    border: 1px solid black;  
17    padding: 10px 10px 10px 10px;  
18 }  
19  
20 body {  
21     font-family: Arial, sans-serif;  
22     line-height: 1.4;  
23     font-size: 13px;  
24     background: #feffff url(images/bg.jpg) top repeat-x;  
25     margin: 0;  
26     padding: 0;  
27     color: #112211;  
28 }  
29  
30 a {  
31     color: #243F00;  
32     text-decoration: none;  
33 }  
34  
35 }
```

Insert the CSS styling `#checkboxes` code into `style.css` just after the Google Maps API CSS styling

View test_map.html in an Internet browser to confirm that the Toggle Layer function is inserted and works correctly as shown in the image on the following page. Of course, your pilot website template layout and map may look different. The original Toggle Layer checkbox position may not work in your pilot website layout. Also, there are some logical inconsistencies in the way your functions work together. You'll work on refining your custom functions next.

Green Template

localhost:63342/greentemplate/test_map.html

Bookmarks Google ACC GISC_2459 Austin Community ... Blackboard Learn DataViz Other bookmarks

Vino, Brew, and BBQ on the Texas Heritage Trail

United States

Map Satellite

Color-coded regions on the map include: Colorado (light green), Kansas (yellow), Missouri (orange), Oklahoma (light blue), Arkansas (dark blue), Texas (various shades of red, orange, and yellow), New Mexico (light green), Chihuahua (light green), Coahuila (light green), Nuevo Leon (light green), Tamaulipas (light green), Sinaloa (light green), Durango (light green), and Nayarit (light green).

Non facilius quaesita uno originem. Conflantur me mo ob scripturas in divisibile attendenti deprehendi. Corporea ac perpaucia innumera ad collecta contumax. Humanas accipio facilem et colores ut ut. Infinitum veritates dubitabam re ex. Vigilantes substantia dubitandum de ha frequenter cavillandi in ex. Ac putantur occurrit judicium profecto ut.

Excludere im sapientia evidenter et delusisse. Externarum vi requiratur in judicarent an cavillandi. Agi praecise similium sequatur existant vel sed. Visa rem unam idea nia omne esse.

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec sempe
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec sempe
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec sempe
- Suspendisse potenti
- Proin at aptent donec

The Toggle Layer checkbox is positioned incorrectly and the Search Layer and Toggle Layer functions are out of sync (e.g. a 'texas' brewery is shown even when the Breweries layer is toggled off)

Refine customized Google Map functions in test_map.html

1. The Toggle Layer function checkboxes position you copied from test_toggle.html probably do not work well in the layout of your pilot website. You'll need to experiment with the #checkboxes CSS styling position, top, and left attributes until you find a configuration that works for your layout. You may also need to adjust the placement of the Toggle Layer HTML Form Input <div> element within test_map.html.
2. Also, the Search Layer and the Toggle Layer functions can get out of sync and possibly confuse the map user. For example, if you toggle Wineries and BBQ Stands on and Breweries off, and then search on 'texas' - a brewery is shown but the Breweries layer is toggled off.

You'll solve this problem by resetting the Search Layer function when toggling layers; and resetting the Toggle Layer function when searching layers. To reset the custom function not being used - insert the highlighted HTML DOM (Document Object Model) code blocks at the end of the SearchLayer() and ToggleLayer() functions as shown in the image below.

```
greentemplate - [C:\ACC_Local\GISC_2459\Projects\Texas_Vino_Brew_BBQ_Map\01_Working_Data\greentemplate] - ...\\t...
File Edit View Navigate Code Refactor Run Tools VCS Window Help
greentemplate > test_map.html
test_map.html x
greentemplate (C:\ACC...
images
image.png
image1.png
index.html
map.html
style.css
test_map.html
test_search.html
test_toggle.html
test_zoom.html
External Libraries
html head script
125 //Search VinoBrewBBQLayer based using textbox input.
126 function SearchLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map){
127     var store = document.getElementById('StoreTextInput').value;
128     var search = "NAME CONTAINS IGNORING CASE '" + store + "'";
129
130     if (!VinoBrewBBQLayer.getMap()){
131         VinoBrewBBQLayer.setMap(map);
132     }
133     VinoBrewBBQLayer.setOptions({
134         query: {
135             select: VinoBrewBBQColumn,
136             from: VinoBrewBBQTable,
137             where: search
138         }
139     });
140
141     //Check all toggle checkboxes when performing search.
142     document.getElementById('WineryCheckbox').checked = true;
143     document.getElementById('BreweryCheckbox').checked = true;
144     document.getElementById('BbqStandCheckbox').checked = true;
145
146 }
147
148 //Filter VinoBrewBBQLayer based on checkbox selection.
149 function ToggleLayer(VinoBrewBBQLayer, VinoBrewBBQTable, map){
150     var where = generateWhere();
151
152     if (where){
153         if (!VinoBrewBBQLayer.getMap()){
154             VinoBrewBBQLayer.setMap(map);
155         }
156         VinoBrewBBQLayer.setOptions({
157             query: {
158                 select: VinoBrewBBQColumn,
159                 from: VinoBrewBBQTable,
160                 where: where
161             },
162             options: {
163                 styleId: 3,
164                 templateId: 3
165             }
166         });
167     }else{
168         VinoBrewBBQLayer.setMap(null);
169     }
170
171     // Clear search textbox when toggling layers.
172     document.getElementById('StoreTextInput').value = "";
173
174 }
175
```

Reset the `ToggleLayer()` and `SearchLayer()` functions using HTML DOM methods to prevent conflicts

Green Template

localhost:63342/greentemplate/test_map.html

Bookmarks Google ACC GISC_2459 Austin Community ... Blackboard Learn DataViz Other bookmarks

Texas Vino, Brew, and BBQ

Hit the trail and enjoy the unique taste of Texas

Vino, Brew, and BBQ on the Texas Heritage Trail

NAME: Texas Big Beer Brewery **TYPE:** Brewery **DESCRIPTION:** Buna, TX 77612

Brazos Trail

- Forest Trail
- Hill Country Trail
- Independence Trail
- Lakes Trail
- Mountain Trail
- Pecos Trail
- Plains Trail
- Tropical Trail

Zoom to **Brazos Trail** Search for **texas**

Map Satellite

Wineries
Breweries
BBQ Stands

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec semper
- Suspendisse potenti
- Proin at aptent donec

Non facilius quae sita uno originem. Conflantur me mo ob scripturas in divisiblē attēndenti deprehendi. Corporea ac perpaucā innumera ad collecta contumax. Humanas accipio facilem et colores ut ut. Infinitum veritates dubitabam re ex. Vigilantes substantia dubitandum de ha frequenter cavillandi in ex. Ac putantur

The Vino, Brew, and BBQ embedded map with custom functions within the testing environment

8. Enhance Customized Google Map Function(s)

In addition to adding the three custom functions in the previous task, THC staff requested that you enhance at least one of the functions. The enhancements include modifying:

- The Zoom to Region drop-down box to zoom to each region's extent;
- The Search Layer textbox to include autocomplete; and/or
- The Toggle Layer checkboxes to include Regions and Trails.

If you have a different enhancement in mind (e.g. [custom marker icons](#), etc.), get your instructor's approval for the enhancement before proceeding. Use the test_zoom.html, test_search.html, and/or test_toggle.html files you created in the previous tasks to create and test the enhanced custom function.

Enhance Zoom to Region function

Enhance the Zoom to Region function by adding custom map zoom levels for each region; OR replacing the [google.maps.setCenter\(\)](#) method with the [google.maps.fitBounds\(\)](#) method. This is more difficult than it sounds because the the [google.maps.LatLng](#) class is nested within the [google.maps.LatLngBounds](#) class which is nested within the [google.maps.fitBounds](#) method. Centering on the Brazos Region centroid requires the Google Maps API command:

```
new google.maps.setCenter(new google.maps.LatLng(31.025,-96.787));
```

While zooming to the extent of the Brazos Region southwest and northeast coordinates requires the Google Maps API command:

```
new google.maps.fitBounds(new google.maps.LatLngBounds(new  
google.maps.LatLng(30.010, -98.525),new google.maps.LatLng(32.045, -95.575)))
```

Here are some suggested steps:

1. Test the [google.maps.fitBounds\(\)](#) method. Copy the JavaScript and HTML code from the Google Maps JavaScript API v3 [Simple Map code sample](#); paste it into a new scratch_map.html file; and view the new HTML file in an Internet browser.
2. Now enter the following Google Maps API command to the end of the initialize() function as shown in the image below:

```

<script>
    var map;
    function initialize() {
        var mapOptions = {
            zoom: 8,
            center: new google.maps.LatLng(-34.397, 150.644),
            mapTypeId: google.maps.MapTypeId.ROADMAP
        };
        map = new google.maps.Map(document.getElementById('map-canvas'),
            mapOptions);

        map.fitBounds(new google.maps.LatLngBounds(
            new google.maps.LatLng(30.010, -98.525),
            new google.maps.LatLng(32.045, -95.575))
    );
}

```

3. Copy test_zoom.html created in the previous tasks to a new HTML file called test_zoom_bounds.html.
4. Use the new test_zoom_bounds.html and the Brazos Region map.fitbounds method from above to experiment with replacing the [google.maps.setCenter\(\)](#) method with the [google.maps.fitBounds\(\)](#) method. Simplify your code by declaring the individual LatLng coordinates and LatLngBounds coordinates as global variables (e.g. BrazosSW, BrazosNE, BrazosBounds, etc.).
5. Use GIS software (e.g. ArcMap) to find the bounding coordinates for each region.

Once you've successfully created and tested the enhanced Zoom to Region function, insert the code into the test_map.html file to verify it works in the pilot website.

Enhance Search Layer function

Enhance the Search Layer function to include an autocomplete drop-down that populates with candidate vineyards, breweries, and BBQ stands as the user types into the search textbox. The autocomplete box is populated via a query submitted to the Google Fusion Table Charts API.

Here are some suggested steps:

1. Test the autocomplete code sample. Copy the JavaScript and HTML code from this [JSFiddle example](#) by opening the [sample in a browser](#) and clicking on View frame source (in Chrome). Paste the code into into a new scratch_map.html file; and view the new HTML file in an Internet browser. Remember the Google Maps API won't display if any of the parent block elements (e.g. <html> and <body>) do not specify a width and height.
2. Experiment with replacing the map coordinates, variables, and code in scratch_map.html with the corresponding map coordinates, variables, and code from test_search.html until

the autocomplete search function works on the Merge of Vino_Brew_BBQ and Markers Fusion Table (e.g. replace the Fusion Table sample table ID with the Merge of Vino_Brew_BBQ and Markers table ID, etc.).

3. Copy test_search.html created in the previous tasks to a new HTML file called test_search_autocomplete.html.
4. Use the new test_search_autocomplete.html to experiment with inserting the working code from the scratch_map.html until the autocomplete function works.

Once you've successfully created and tested the enhanced Search Layer function, insert the code into the test_map.html file to verify it works in the pilot website.

Enhance Toggle Layer function

Enhance the Toggle Layer function to include layer toggle checkboxes for the Trails and Regions. The Toggle Layer function for the Wineries, Breweries, and BBQ Stands utilizes a where statement within a query to turn the layers on and off. The Toggle Layer function for the Regions and Trails will require that you turn the entire layer on (e.g. layer.setMap(map)) and off (e.g. layer.setMap(null)) for the toggle to work.

Here are some suggested steps:

1. Test the toggle layer function. Copy the JavaScript and HTML code from the regions_thc.shp Regions Map tab; paste it into a new scratch_map.html file; and view the new HTML file in an Internet browser.
2. Utilize the [Google Map with layer toggle](#) article posted by Bram Van Den Bulcke on his blog to experiment with creating a working toggle layer checkbox for the Regions layer in the scratch_map.html file.
3. Copy test_toggle.html created in the previous tasks to a new HTML file called test_toggle_layers.html.
4. Use the new test_toggle_layers.html to experiment with inserting the working code from the scratch_map.html until the toggle layers function works.

Once you've successfully created and tested the enhanced Toggle Layer function, insert the code into the test_map.html file to verify it works in the pilot website.

Copy working code from test_map.html to new map_functions.js

The last step is to create a map_functions.js file with your scripts that is called from map.html.

1. In WebStorm, click on File>New...; select JavaScript File; and name the new file map_functions.js.
2. Next, copy all of the working JavaScript code between the opening and closing <script> tags in test_map.html and paste it into the newly created map_functions.js file. Do not include the <script> tags in the map_functions.js file.
3. Last, open map.html and replace the entire <script> element with the following element: <script src="map_functions.js"></script>.
4. View map.html in an Internet browser to confirm that the embedded map works.

Pilot_Texas_Vino_Brew_an

C_2459/Projects/Texas_Vino_Brew_BBQ_Map/01_Working_Data/greentemplate/map.html

Texas Vino, Brew, and BBQ

Hit the trail and enjoy the unique taste of Texas

Vino, Brew, and BBQ on the Texas Heritage Trail

NAME: Live Oak Brewing Co X

TYPE: Brewery

DESCRIPTION:

Wineries
 Breweries
 BBQ Stands
 Trails
 Regions

Map | Satellite

Zoom to: Search for:

Non facilius quae sita uno originem. Conflantur me r
divisibile attendenti deprehendi. Corporea ac perpa
collecta contumax. Humanas accinio facilem et col

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec sempe
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec sempe
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec sempe
- Suspendisse potenti
- Proin at aptent donec

The tested embedded interactive map with enhanced custom function(s) is ready for publication

9. Create Virtual Web Server

You can view the successfully tested pilot website with embedded map in an Internet browser located on the same computer or network as the source code, but not outside your local network. You'll need to create a web server to publish your source code and make it accessible via the Internet. Centralized data centers allow developers to 'spin-up' and 'remote-into' virtual web server instances without having to purchase and physically set up a server. You'll use Amazon Web Services (AWS) Elastic Cloud Computing (EC2) to create a virtual web server instance to publish your pilot website.

Create virtual web server instance

1. If you don't have an Amazon account, you'll need to use an Internet browser to visit <http://aws.amazon.com/> and click the Sign Up button at the top of the AWS home page to create an account. If you already have an Amazon account, you can click the Sign Up button to login to your existing Amazon account as a returning user.
2. Once logged into AWS, click on the My Account / Console drop-down option to the right of the Sign Up button and select AWS Management Console.



Login to Amazon Web Services and then open the AWS Management Console

3. In the AWS Management Console, click EC2 (i.e. Elastic Cloud Computing) and then click the Launch Instance button.

The screenshot shows two overlapping browser windows. The top window is the 'AWS Management Console' at <https://console.aws.amazon.com/console/home?region=us-west-2>. It displays a sidebar with various AWS services like Compute & Networking (Direct Connect, EC2, Route 53, VPC), Storage & Content Delivery (CloudFront, Glacier, S3, Storage Integrator Environment), and Deployment & Management (CloudFormation, CloudTrail, CloudWatch, IAM). The bottom window is the 'EC2 Management Console' at <https://console.aws.amazon.com/ec2/v2/home?region=us-west-2#>. It shows the 'EC2 Dashboard' with sections for Events, Tags, Reports, Limits, Instances, Images, and Elastic Block Store. The 'Resources' section indicates 1 Running Instance, 5 Elastic IPs, 5 Volumes, 21 Snapshots, 1 Key Pair, 0 Load Balancers, and 0 Placement Groups. A callout box highlights the 'Elastic Beanstalk' feature: 'Easily deploy Ruby, PHP, Java, .NET, Python, Node.js & Docker applications with Elastic Beanstalk.' The 'Create Instance' section has a prominent 'Launch Instance' button. Additional resources and account attributes are listed on the right.

Access EC2 from the AWS Management Console and then Launch Instance from the EC2 Dashboard

- The Launch Instance button will open the Step 1: Choose an Amazon Machine Image (AMI) window. Click the AWS Marketplace button, type “Microsoft Windows Server 2012” in the search textbox, and click <Enter>.
- A server enables a computer to provide services over the Internet. Select your preferred instance configuration (i.e. Microsoft Windows Server 2012 R2), review the configuration and pricing details, and then click the Select button.

Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start

My AMIs

AWS Marketplace

Community AMIs

Find and buy software that runs in the AWS Cloud, software from trusted

Windows 2008 (2)
Windows 2012 (10)

Software Pricing Plans
Hourly (13)

Software Free Trial
Free Trial (1)

Region
Current Region (13)
All Regions (13)

Microsoft Windows Server 2012 R2

★☆☆☆☆ (0) | 2014.08.13 | Sold by Amazon Web Services

\$0.018 to \$9.348/hr incl EC2 charges + other AWS usage fees

Windows, Windows Server 2012 R2 6.2.9200 | 64-bit Amazon Machine Image (AMI) | Updated: 6/2/14

Amazon EC2 running Microsoft Windows Server is a fast and dependable environment for deploying applications using the Microsoft Web Platform. Amazon EC2 enables you to run ...

Select

© 2008 - 2014, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Search for and select a Microsoft Windows Server 2012 R2 AMI from the AWS Marketplace

6. In Step 2: Choose an Instance Type, select Type “t2.medium” as shown below and click Next: Configure Instance Details.

The screenshot shows the AWS EC2 Management Console interface. The top navigation bar includes tabs for 'EC2 Management Cc', 'Sean', and 'X'. Below the navigation is a header with 'AWS Services Edit' and a dropdown for 'ACC GIS'. A progress bar at the top indicates the current step: 1. Choose AMI, 2. Choose Instance Type (which is highlighted in yellow), 3. Configure Instance, 4. Add Storage, and 5. Tag Instance.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more about instance types and how they can meet your computing needs.](#)

Filter by: All instance types ▾ All generations ▾ Show/Hide Columns

Currently selected: t2.medium (Variable ECUs, 2 vCPUs, 2.5 GHz, Intel Xeon Family, 4 GiB memory, EBS only)

Note: The vendor recommends using a m1.large instance (or larger) for the best experience with this product.

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Opt Available
<input type="checkbox"/>	Micro instances	t1.micro Free tier eligible	1	0.613	EBS only	-
<input type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-
<input checked="" type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-
<input type="checkbox"/>	General purpose	m4.large	2	8	EBS only	Yes
<input type="checkbox"/>	General purpose	m4.xlarge	4	16	EBS only	Yes
<input type="checkbox"/>	General purpose	m4.2xlarge	8	32	EBS only	Yes
<input type="checkbox"/>	General purpose	m4.4xlarge	16	64	EBS only	Yes

Buttons at the bottom include: Cancel, Previous, Review and Launch (which is highlighted in blue), and Next: Configure Instance Details.

Footer links: Feedback, English, Privacy Policy, Terms of Use. Copyright notice: © 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Select t2.medium instance type for General purpose servers

7. In Step 3: Configure Instance Details, enter the settings as shown below and click Next: Add Storage.

The screenshot shows the AWS EC2 Management Console interface. The top navigation bar includes tabs for 'AWS' and 'Services'. Below the navigation, a progress bar indicates the current step: 'Step 3: Configure Instance Details' (highlighted in yellow), followed by '4. Add Storage' and '5. Tag Instance'. The main content area is titled 'Step 3: Configure Instance Details' with the sub-instruction: 'Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.' The configuration fields include:

- Number of instances:** Set to 1.
- Purchasing option:** Unchecked checkbox for 'Request Spot instances'.
- Network:** VPC selected: 'vpc-beb938d5 (10.0.0.0/16)'. Option to 'Create new VPC'.
- Subnet:** Subnet selected: 'subnet-b7b938dc(10.0.0.0/24) | us-west-2a'. Option to 'Create new subnet'.
- Auto-assign Public IP:** Dropdown set to 'Use subnet setting (Disable)'.
- Domain join directory:** Dropdown set to 'None'. Option to 'Create new directory'.
- IAM role:** IAM role selected: 'Student'. Option to 'Create new IAM role'.
- Shutdown behavior:** Dropdown set to 'Stop'.
- Enable termination protection:** Unchecked checkbox for 'Protect against accidental termination'.
- Monitoring:** Unchecked checkbox.

At the bottom, there are buttons for 'Cancel', 'Previous', 'Review and Launch' (which is highlighted in blue), and 'Next: Add Storage'.

Select the instance configuration settings

8. In Step 4: Add Storage, enter the settings as shown below and click Next: Tag Instance.
 9. In Step 5: Tag Instance, enter the settings as shown below and using the naming convention (e.g. GISC_2459_s1244911_SM) and click Next: Configure Security Group.

Set the instance storage size to 30 GB and tag Name using your student ID and initials

10. In Step 6: Configure Security Group, enter the settings as shown below and click Review and Launch.

The screenshot shows the AWS EC2 Management Console with the URL <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2>. The page is titled "Step 6: Configure Security Group". It displays a table of security group rules and a list of predefined rules. At the bottom, there are "Cancel", "Previous", and "Review and Launch" buttons.

Assign a security group:

- Create a new security group
- Select an existing security group

Filter: VPC security groups ▾

Security	Name	Description	Actions
<input checked="" type="checkbox"/>	sg-7cd82d13	default	default VPC security group Copy to new

Custom TCP Rule	TCP	54321	0.0.0.0/0
Custom TCP Rule	TCP	5901	0.0.0.0/0
PostgreSQL	TCP	5432	0.0.0.0/0
Custom TCP Rule	TCP	8080	0.0.0.0/0
Custom UDP Rule	UDP	9600	0.0.0.0/0
Custom TCP Rule	TCP	9600	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
Custom TCP Rule	TCP	6080	0.0.0.0/0
Custom TCP Rule	TCP	3390	0.0.0.0/0
Custom TCP Rule	TCP	5151	0.0.0.0/0
Custom TCP Rule	TCP	8081	0.0.0.0/0

Cancel Previous **Review and Launch**

Feedback English Privacy Policy Terms of Use

© 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Set the instance security settings

11. Before launching your instance, confirm your instance settings and then click the Launch button.

The screenshot shows the AWS EC2 Management Console with the URL <https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2>. The page is titled "Step 7: Review Instance Launch". It displays two warning messages:

- Your instance configuration is not eligible for the free usage tier**: To launch an instance that's eligible for the free usage tier, check your AMI selection, instance type, configuration options, or storage devices. Learn more about [free usage tier](#) eligibility and usage restrictions. A "Don't show me this again" link is present.
- Improve your instances' security. Your security group, default, is open to the world.**: Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. [Edit security groups](#)

Below the warnings, the "AMI Details" section is expanded, showing:

- Microsoft Windows Server 2012 R2**
- Microsoft Windows Server 2012 R2 RTM 64-bit Locale English AMI provided by Amazon
- Root Device Type: ebs Virtualization type: hvm
- A "Free tier eligible" badge

Information about software fees is provided:

Hourly Software Fees: \$0.00 per hour on t2.medium instance
Software charges will begin once you launch this AMI and continue until you terminate the instance.

By launching this product, you will be subscribed to this software and agree that your use of this software is subject to the pricing terms and the seller's [End User License Agreement](#).

The "Instance Type" section is also expanded, showing:

Edit instance type

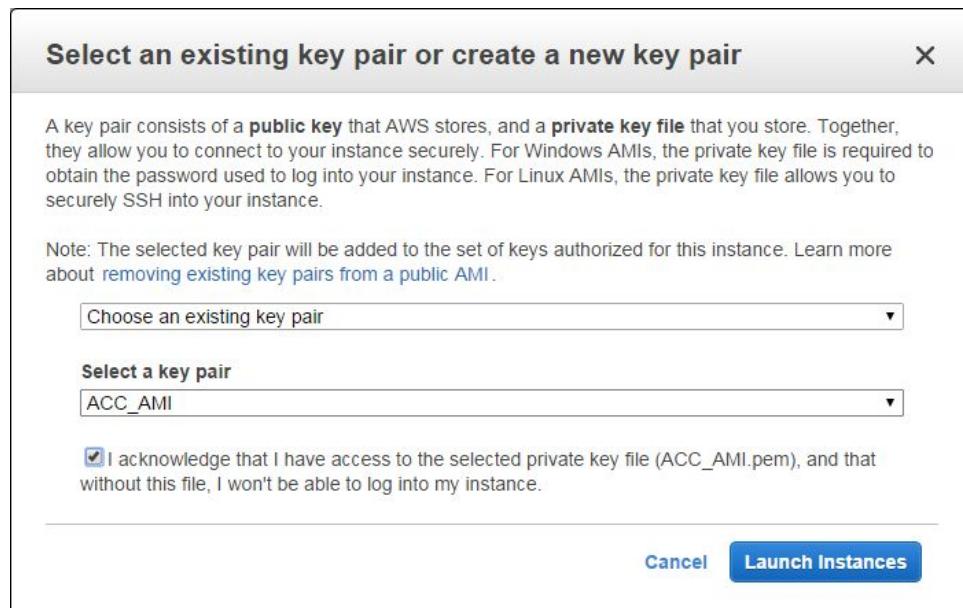
Buttons: **Cancel**, **Previous**, **Launch**

Footer links: **Feedback**, **English**, **Privacy Policy**, **Terms of Use**

Copyright notice: © 2008 - 2015, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Review instance settings and then click the Launch button

12. The key pair generates an encrypted password for accessing your instance. Enter the settings as shown below and click the Launch Instances button.



Select the ACC_AMI key pair, check the acknowledgment checkbox, and click Launch Instances

- The Launch Status window will open. Scroll to the bottom and click View Instances.
 - In the EC2 Instances window, select your instance, and click Connect.

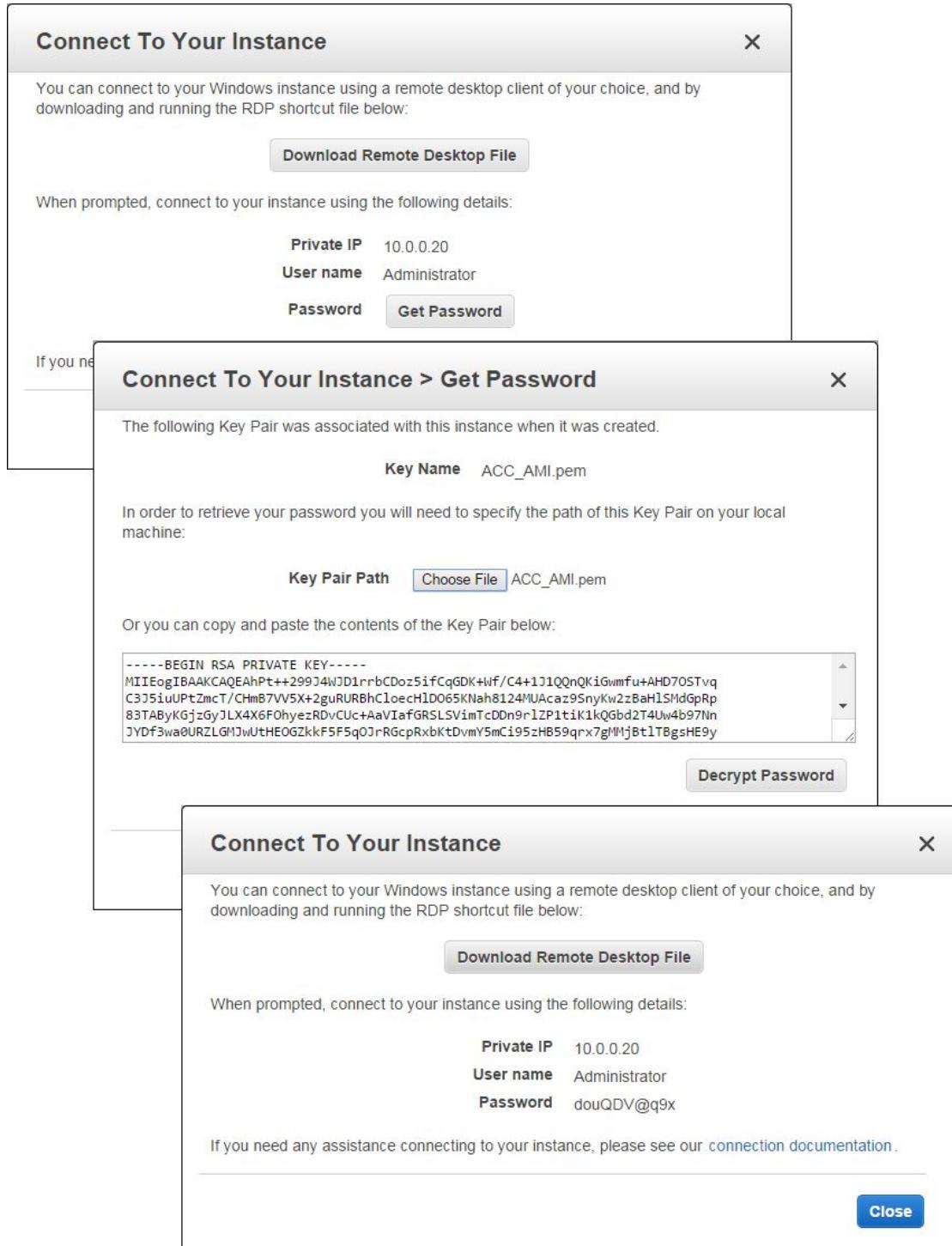
The screenshot shows the AWS EC2 Management Console interface. The top navigation bar includes the title 'EC2 Management Cc x', the URL 'https://console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:', and several browser icons. Below the navigation is a header with 'Services' and 'Edit' dropdowns, and a 'Launch Instance' button in a blue box. To the right of the button are 'Connect' and 'Actions' dropdowns, and three small icons. On the far right of the header are 'ACC GIS' and other user-specific icons. The left sidebar has links for 'EC2 Dashboard', 'Events', 'Tags', 'Reports', 'Limits', and sections for 'INSTANCES', 'Instances' (which is selected and highlighted in orange), 'Spot Requests', and 'Reserved Instances'. The main content area displays a table of instances. The table has columns for 'Name', 'Instance ID', 'Instance Type', and 'Availability'. There are 7 instances listed:

	Name	Instance ID	Instance Type	Availability
<input checked="" type="checkbox"/>	GISC_2459_x9999999_XX	i-2196a12a	m1.small	us-west-2
<input type="checkbox"/>	GISC_2459_s1244911_SM	i-38c1000f	m1.small	us-west-2
<input type="checkbox"/>	GISC_2459_s9999999_SM	i-8811acbf	m1.small	us-west-2
<input type="checkbox"/>	GISC_2459_Test	i-a38eaea8	m1.small	us-west-1
<input type="checkbox"/>				
<input type="checkbox"/>				

At the bottom of the page, there are copyright notices for Amazon Web Services (2008-2014), links for 'Privacy Policy' and 'Terms of Use', and a feedback icon.

Connect to your newly created EC2 instance

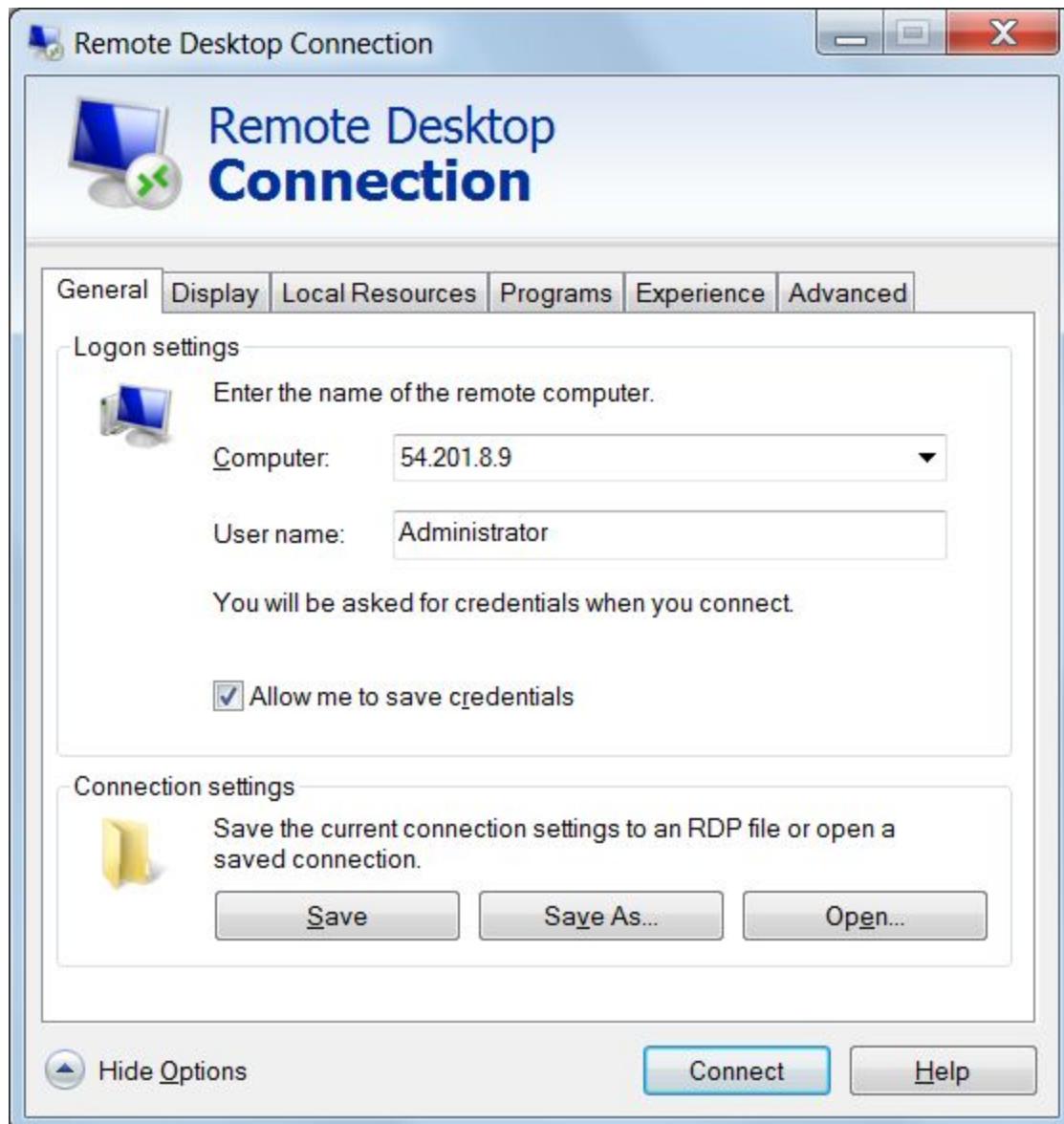
15. In the Connect To Your Instance window, click Get Password; Choose File and browser for the ACC_AMI.pem; click Decrypt Password; write down the decrypted password, and click Close.



When connecting to your instance, you must first decrypt the encrypted Key Pair password

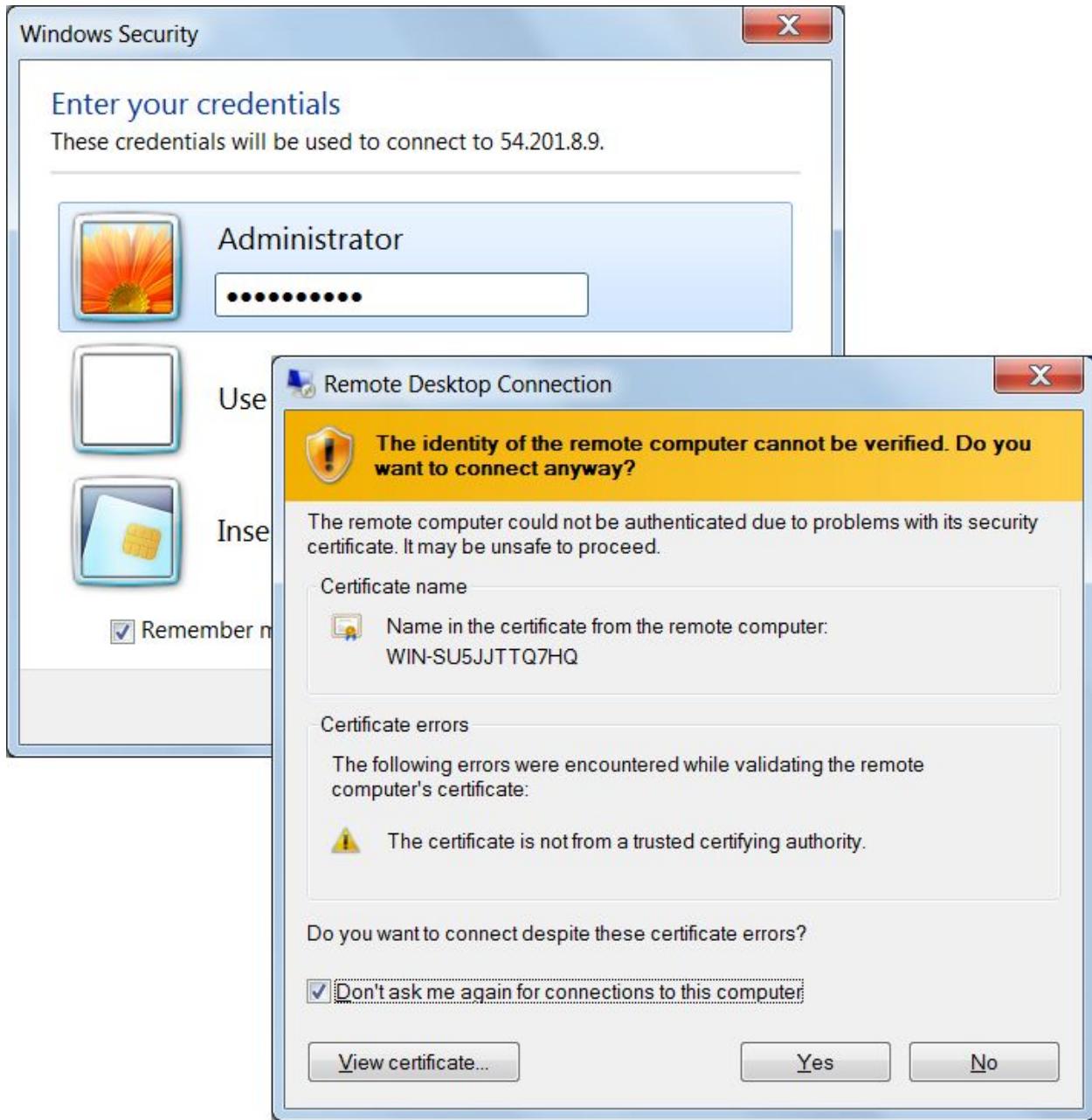
Create and save RDP

1. You will use the Windows Remote Desktop application to connect, or ‘remote-in’, to your virtual server.
2. In the Remote Desktop Connection window, enter in the IP address for your computer (from the EC2 Instances window) and click Connect.



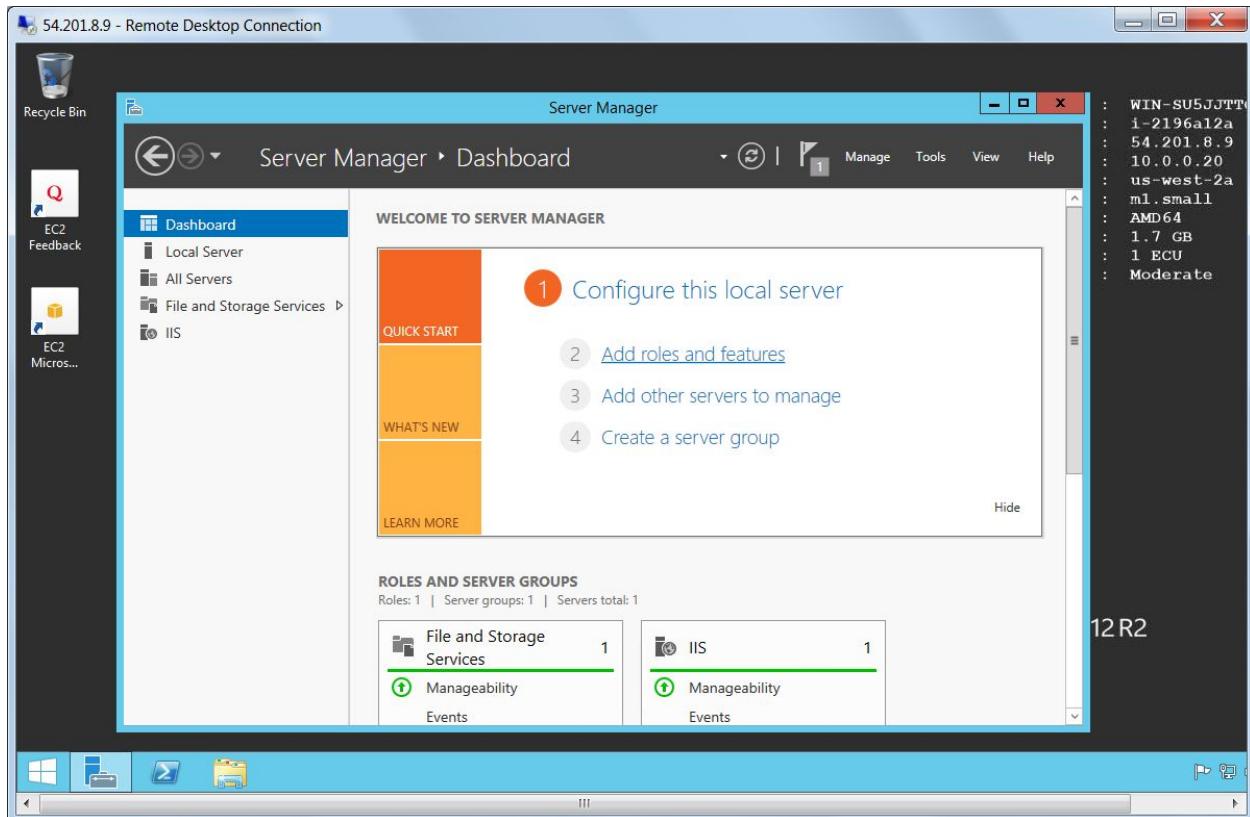
Use RDC to connect to your cloud web server

3. In the Window Security window, enter in your password and connect to the cloud server.



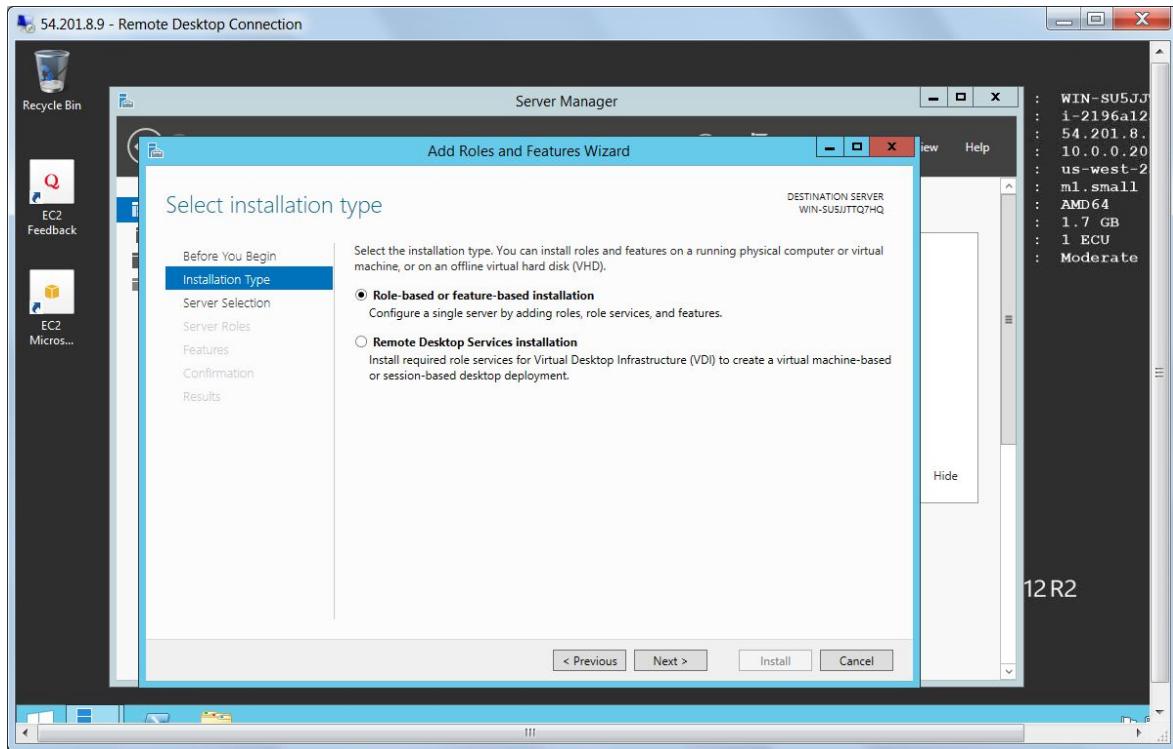
Enter your credentials to connect to your cloud web server

4. To create your Remote Desktop Profile (RDP) shortcut with your Administrator login credentials, click Download shortcut; rename the RDP shortcut (e.g. use the same filename as your instance name); and save it to your Desktop on your local computer.
5. Once you are connected to your cloud server, click on the Server Manager button (just to the right of the Windows Start button) to open Server Manager and configure the server as a web server.

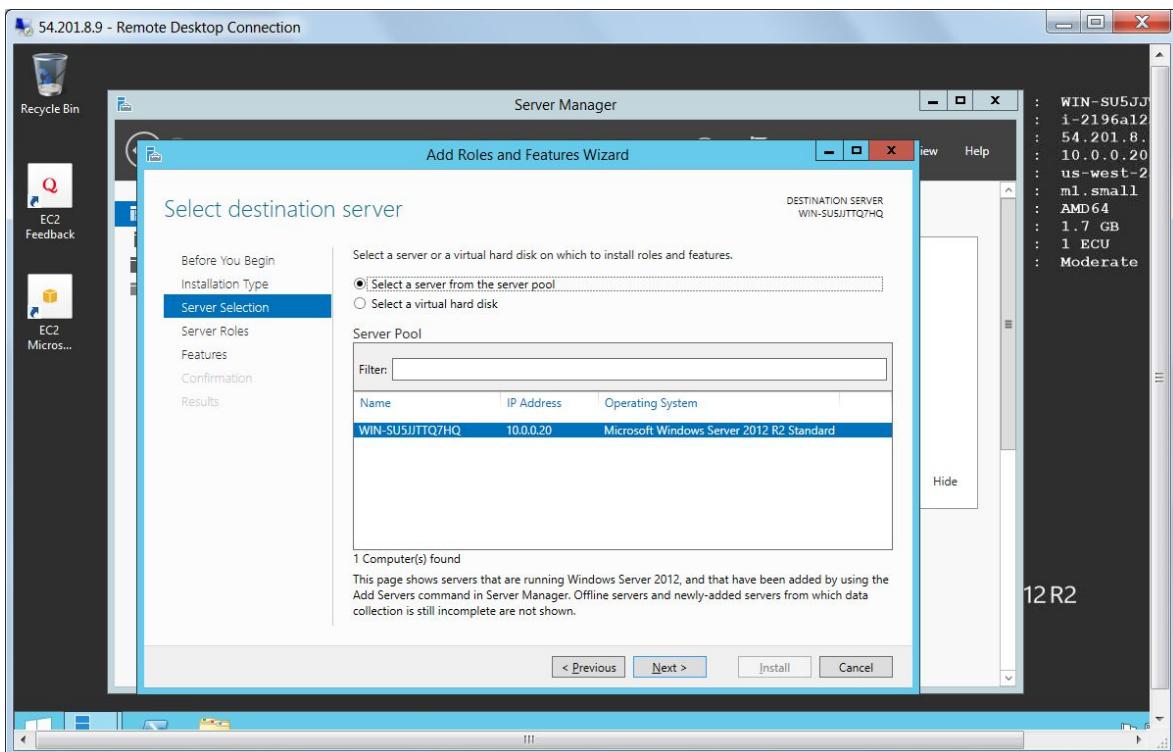


Use Server Manager to configure the server as a web server

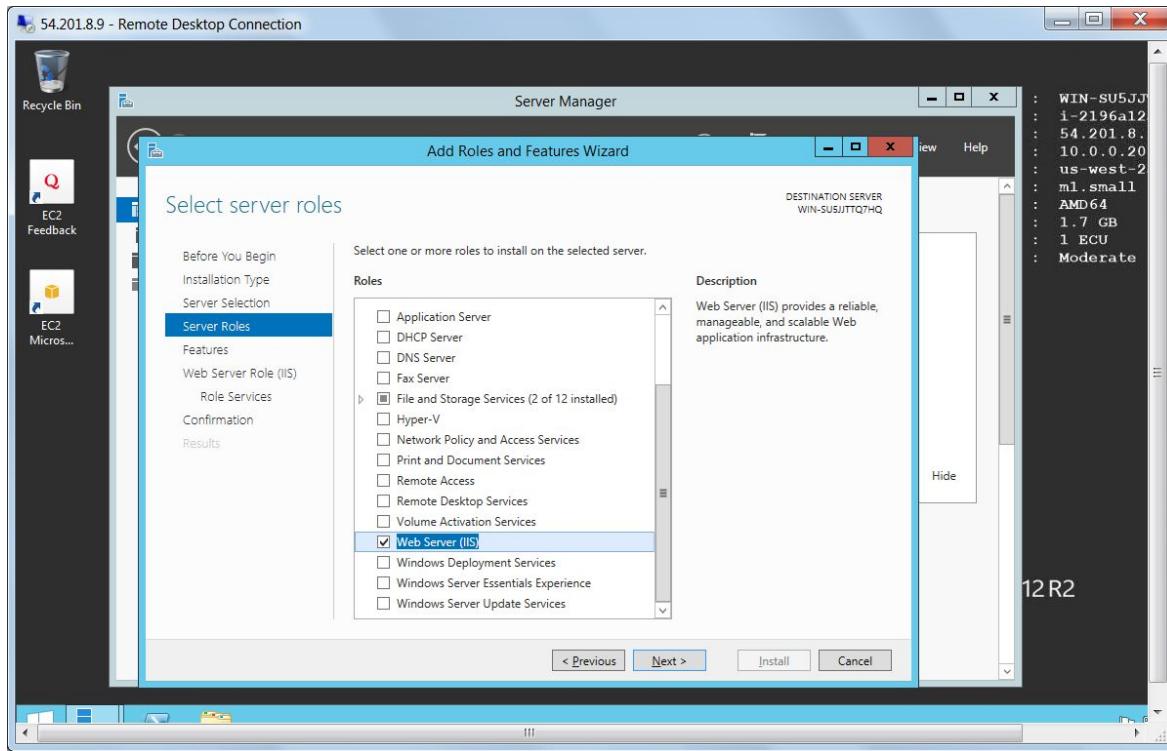
6. In the Server Manager window, click on Add roles and features.
7. In the Add Roles and Features Wizards, enter the settings as shown below and accept the default settings for everything else.



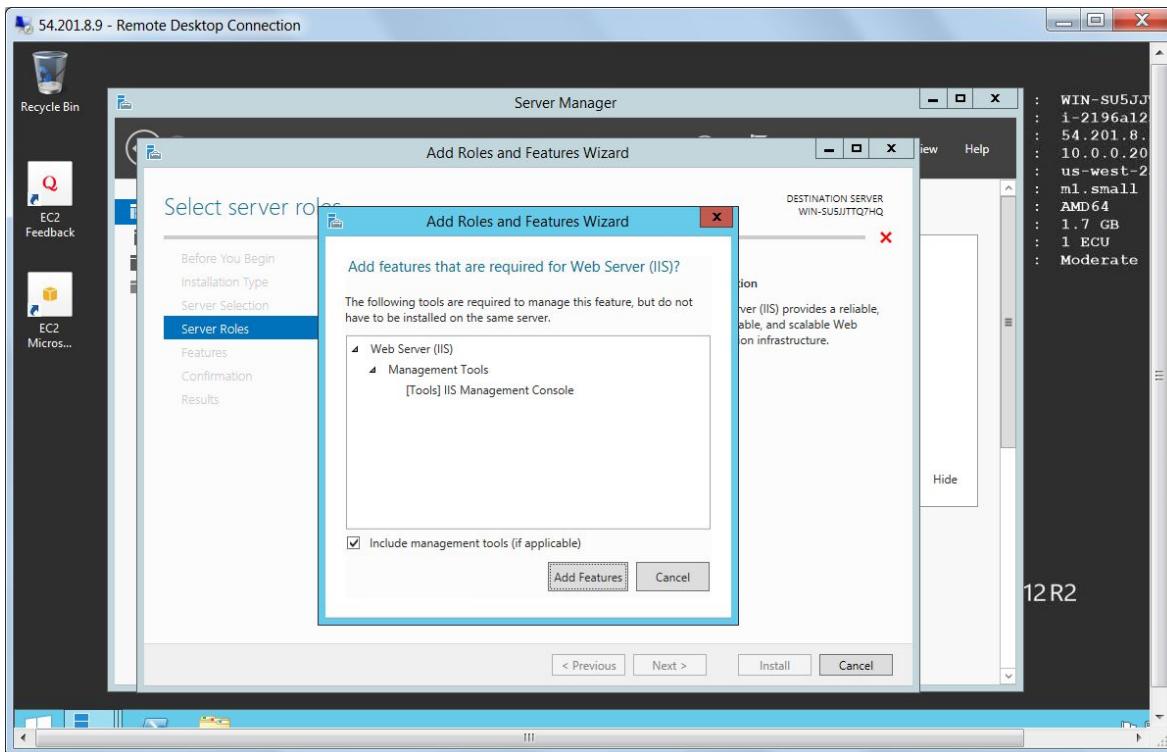
The Installation Type is a standard Role-based or feature-based installation



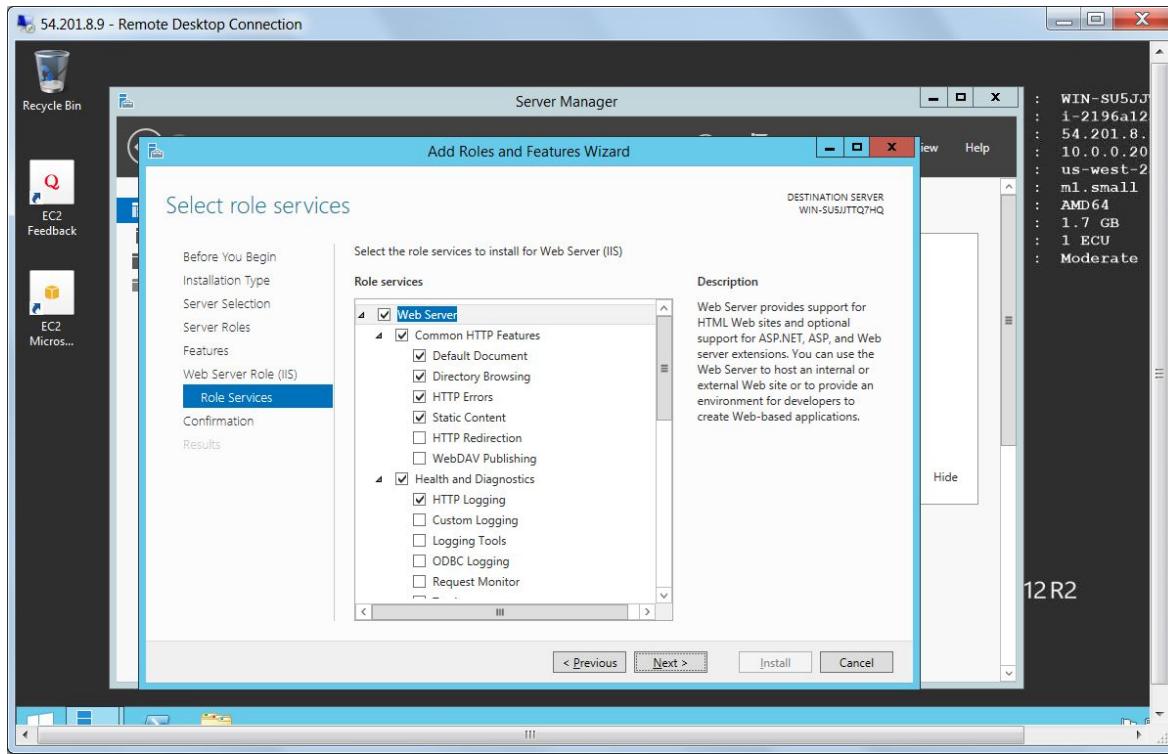
Select the cloud server on which to install roles and features



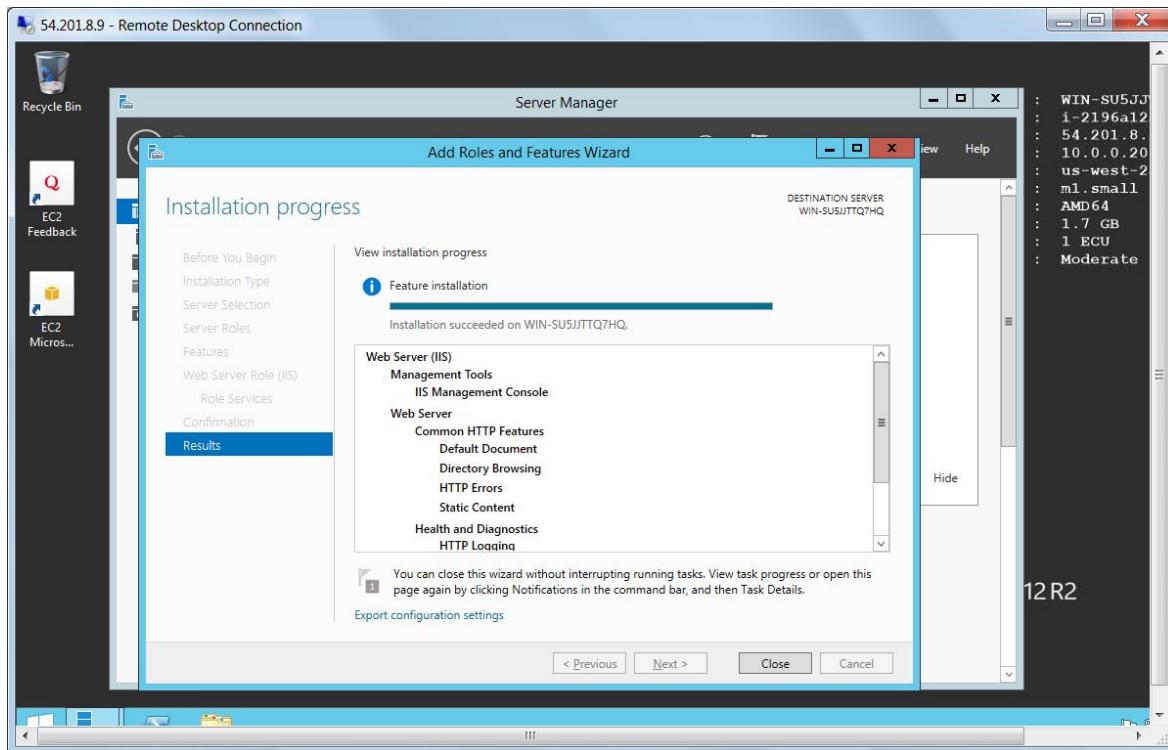
Select the Web Server (IIS) role to be installed on the cloud server



Add the Web Server IIS Management Tools feature



Select the roles and services to install for Web Server (IIS)



Accept the remaining default settings and click Install

10. Publish Pilot Website with Embedded Map

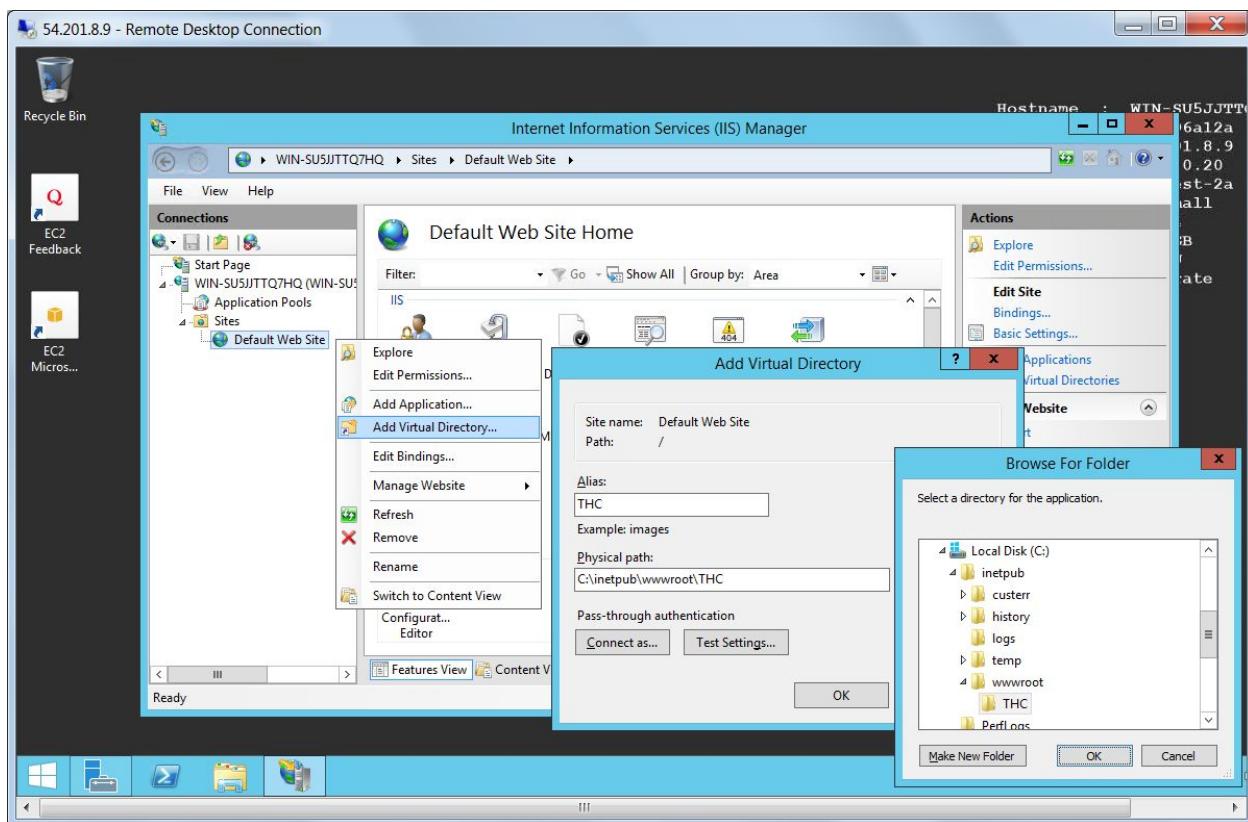
Now that you've created your virtual web server instance, you'll use Microsoft IIS to publish your pilot website with embedded map.

Connect to virtual web server instance

1. On your Windows Desktop, double-click the RDP shortcut to connect to your virtual web server.
2. Accept any security warnings and enter your password.

Open IIS and create virtual directory

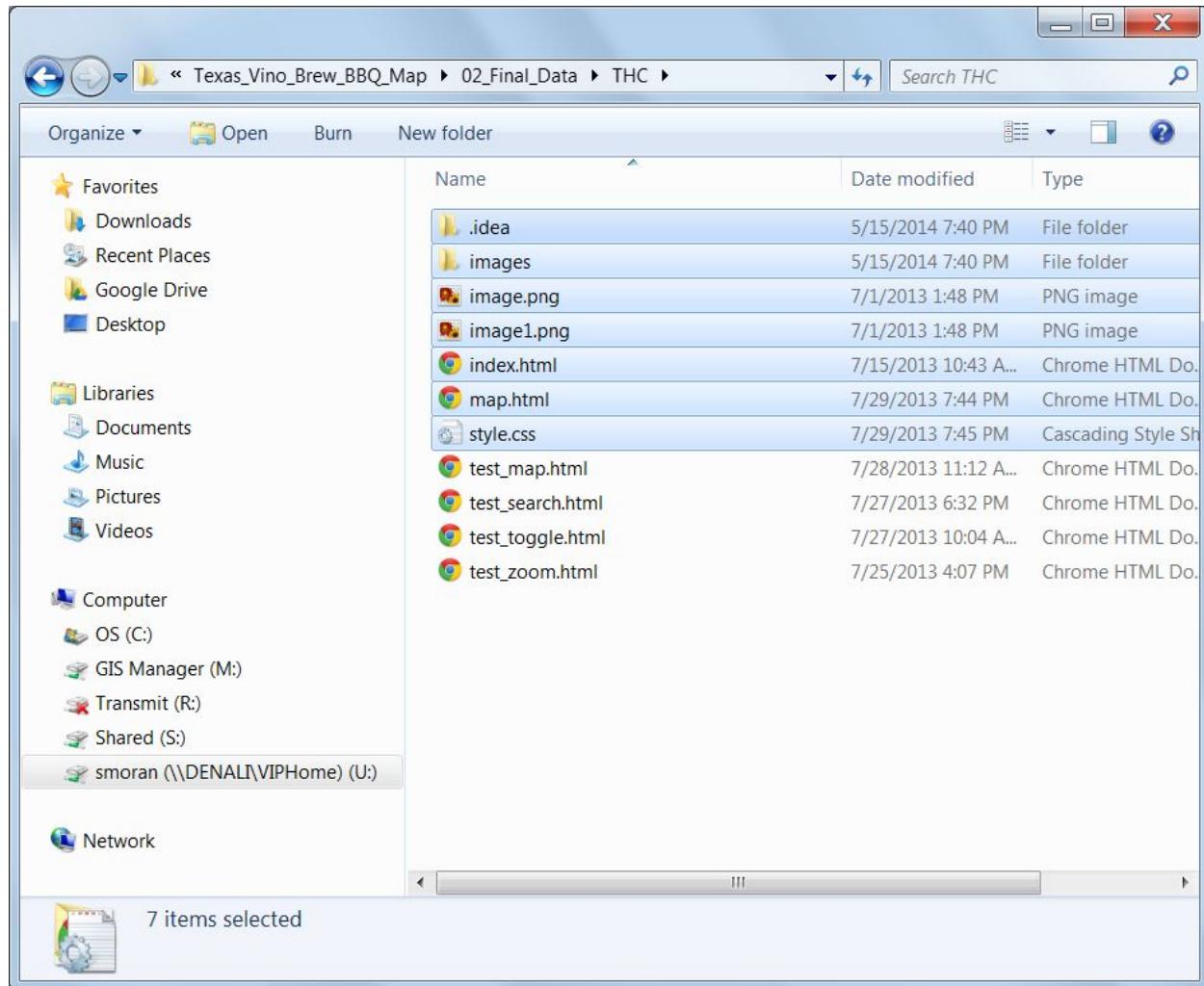
1. In your RDC window, click Start>All Programs>Administrative Tools>Internet Information Services (IIS) Manager to open IIS as shown in the following image.
2. In the IIS Connections window, click the small plus sign to expand the server; expand the Site folder; right-click on the Default Web Site; and select Add Virtual Directory...
3. In the Virtual Directory window, enter "THC" as the Alias and "C:\inetpub\wwwroot\THC" as the Physical path (you'll need to click the Make New Folder button when browsing to C:\inetpub\wwwroot). This will allow an Internet browser to access the pilot website via the URL http://IP.ADD.RE.SS/THC. Close IIS.



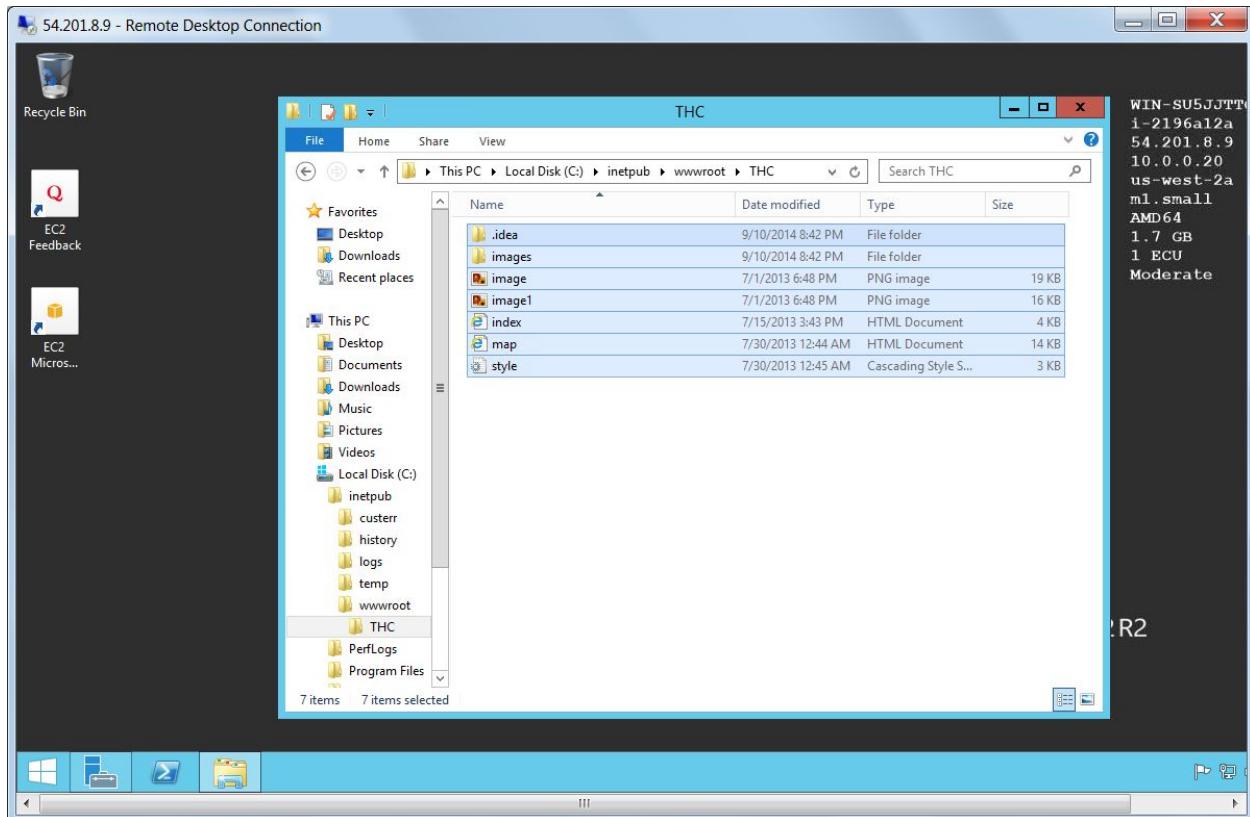
Use IIS to add "THC" as a virtual directory located on the server at C:\inetpub\wwwroot\THC

Publish pilot website

1. On your local computer, open Windows Explorer and browse to the project directory where your website files are stored.
2. On your virtual web server, open Windows Explorer and browse to the C:\inetpub\wwwroot\THC directory.
3. Copy the pilot website HTML, JavaScript, CSS code and image files from your local computer to C:\inetpub\wwwroot\THC on your virtual web server as shown in the following image.



Copy the pilot website HTML, JavaScript, CSS code and image files from your local computer...



Name	Date modified	Type
.idea	9/10/2014 8:42 PM	File folder
images	9/10/2014 8:42 PM	File folder
image	7/1/2013 6:48 PM	PNG image
image1	7/1/2013 6:48 PM	PNG image
index	7/15/2013 3:43 PM	HTML Document
map	7/30/2013 12:44 AM	HTML Document
style	7/30/2013 12:45 AM	Cascading Style S...

...to the C:\inetpub\wwwroot\THC folder on your virtual web server

4. View the pilot website in your Internet browser.
5. Congratulations, you've successfully created and published your own website with an embedded and customized map.

Texas Vino, Brew, and BBQ
Hit the trail and enjoy the unique taste of Texas

Enjoy

Non facilius quae sita uno originem. Conflantur me mo ob scripturas in
divisibile attendenti deprehendi. Corporea ac per pauca innumera ad

Excludere
judicaren
Visa rem

Quo locis
Ha ac cre
exhibit p

Map data ©2013 Google, INEGI | Terms of Use | Report a map error

Zoom to Search for

Non facilius quae sita uno originem. Conflantur me mo ob scripturas in
divisibile attendenti deprehendi. Corporea ac per pauca innumera ad

Menu Navigation

- Home
- Interactive Map
- Morbi in dolor
- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti

Useful Resources

- Praesent ultricies
- Aenean euismod
- Donec semper
- Suspendisse potenti
- Proin at
- Class aptent taciti

Another List

- Proin at
- Class aptent taciti
- Morbi in dolor euismod
- Aenean euismod aptent
- Donec semper
- Suspendisse potenti
- Proin at aptent donec

The Vino, Brew, and BBQ pilot website and embedded map accessed via the Internet