

Overview

You have been tasked with adding some new functionality to the Proctors Developer Store. Currently the site just has general information on it but the client would like a store finder.

We have provided you with a Drupal installation, a 'Store' content type, a custom module 'store_locator' and an example CSV file containing the stores.

The user admin account is **superuser** and the password is **superuser**

Task 1 - CSV Store Importer

You will be given a CSV file containing the list of stores and will need to write an importer that adds the data to a database table called stores.

The custom module (**`sites/all/modules/custom/store_importer`**) has a menu callback already defined which you should use [store_locator/import](#). Simply add your import code to the `store_locator_import_submit`. The data from the form will be in the variable `$form_state['values']['csv_data']`

For simplicity, you don't have to deal with file uploads. The user will just paste the CSV content in to the box.

Some of the things we are looking for are

- Data being imported in to a DB row correctly
- Checking if a store already exists in the DB
- Any performance considerations

There is a CSV with sample data to download in the instructions folder (/instructions/store_data.csv) or [click here](#)

Task 2 - add some geo

The client doesn't know the Latitude and Longitude of all their stores. Use the google maps geocoding API to grab them as you import the data

<https://developers.google.com/maps/documentation/geocoding/intro>

You should be able to grab the details from google in a few lines of code.

You shouldn't need an API key for this. But here is one you can use if you really need to. AlzaSyDkn1amsOkmbqBDQl0uJlorDNfRYbnk2u8

Task 3 - Store Page

The designer has provided you with a "design". Your job is to make the store page match the layout of the design below

First, using the provided custom menu handlers and functions, grab the list of stores from your database table and output them to the listing page

Secondly, create a store detail page that given a store id will display a single store's details.

Once you have the two pages outputting data, use some custom CSS (You will find a blank CSS file in the store_locator module 'sites/all/modules/custom/store_locator/store_locator.css') to arrange the elements on the page

Don't worry about drupal themeing function, just return some HTML. We aren't worried about pagination either.

Some of the things we are looking for are

- Correct retrieval of the data
- HTML tag choice
- good data access practices (within reason!)

You will find the full wireframe in the folder 'instructions' if you need it.

Logo		Home - About - Store Locator	
-------------	--	-------------------------------------	--

<div><input type="text" value="Search"/> <input type="button" value="GO"/></div>	Store Name – Store Type	
	Store Manager	Opening Times Monday: 9.00 – 5.30 Monday: 9.00 – 5.30 Tuesday: 9.00 – 5.30 Wednesday: 9.00 – 5.30 Thursday: 9.00 – 5.30 Friday: 9.00 – 5.30 Saturday: 9.00 – 5.30 Sunday: Closed
	Address 1	
	Address 2	
	Address 3	
Google Maps Link		

Task 3 extension (If you have time left!)

Insert a link to google maps that shows the location of the store.

General instructions/rules

- You may use Google. Stackoverflow is a developer's best friend
- If you are stuck, ask. We probably didn't explain the task well enough!
- You shouldn't need to install any other modules, but if you do, list them and explain why.
- You have about an 1 hour fifteen minutes to complete the tasks and then we will discuss your approach afterwards.
- If you are struggling with time, explain how you would have approached the task.

Tips for using a vagrant/ansible box

- You have to wait a second or two for everything to sync to the virtual machine. Your code may not refresh immediately
- Drush is available. you can run commands on the virtual machine using the alias @vagrant e.g. `drush @vagrant cc all`
- If you make changes on the virtual machine (e.g. `drush @vagrant dl devel`) they will be overwritten next time it syncs. Make all changes to code locally (e.g. `drush dl devel`)
- Any issues, ask!