**vitasa-org Setup Instructions and Developer notes**

**vitasa.org ftp and db names/password**

FTP:

Server: ftp.vitasa.org

Username: vitasa

Password: mSv5s03#uBs@903y

Explicit ftpes

**Database:**

Database Server: 50.62.209.76:3306

Database Name: CC\_VITA

Database Username: CC\_VITA12324

Database Password: ~~yxX07v^0$Cpog166~~; now= Kpq4a85#

**Muse overview**

Because the Muse files contain some javascript code for the google maps, because the forms function for the “Tell Us How We Did” doesn’t work, and because we have 2 separate files (English vs. Spanish) – some setup is required post upload within Muse.

**Spanish Language Support**

Turns out that Muse doesn’t support multi-language pages. So, the hack that is used is as follows:

* The primary index.html in the app root, looks at the language of the device (navigator.language) and redirects to either /es/ if Spanish (prefix is es, any variant), otherwise always to English in /en/.
* This is ‘index.html’ in the url root directory.

Only Spanish and English are supported. All non-Spanish get routed to English.

**Google Maps**

A Google Map is installed in 4 places, each requiring customization.

On in the Home page, and on the My Free Taxes page; then copies of this in the English and Spanish versions. The javascript code is added to the page to drop pins on Sites and support the flags with site details.

The code initializes the map, then pulls the list of sites from an S3 object, then installs the pins on the map with a handler for each pin.

The code is in ‘googlemapsscript.html’ but is added to the page in Object -> Insert HTML.

The trick is that the html id for the map object has to be set in the main file. The only known way to find this id is to bring up the page, use view source, and search for the map (an <iframe> with an href to maps.google.com then note the id of the enclosing <div>).

And, you get to do this 4 times, twice for English (1ea for Home and MFT) and twice for Spanish.

The top of the googlemapsscript file has the following lines that must be modified

**let *idMainDesktop*** = **"u2306"**;  
**let *idMFTDesktop*** = **"u1842"**;  
**let *idMainPhone*** = **"u1306"**;  
**let *idMFTPhone*** = **"u1371"**;

It appears that these ids don’t change once the page has been designed. If the map is removed and reinserted, then the id’s will change. Just republishing the site does not appear to cause the ids to change.

Google maps key for zsquared.net: AIzaSyAGzGpzxEbNLKC8pjEE7V88HOvDJhT8whI

Google maps key for vitasa.org:

**Sites.json**

The googlemapsscript depends on a file to know where the sites are located: sites.json. Presently, this file is in the vitasa directory. The backend will create and push to an S3 object but presently don’t know how to bypass the cross-site reference checks.

**Form post**

We want to allow customers to provide feedback. So, there is a page with a form to allow posting of feedback. The Form post is currently designed to send email (ughh). So, we replace the email send with a DB post. Note that we leave all of the other Adobe code in place (to check for excessive posts from the same IP address, form field validation, etc.).

We also add a separate page to allow view, acknowledge, and delete of feedback posts.

The general process for updating the website will be

* Do the upload from Muse
* Copy the following files scripts directory to replace selected Muse files in en/scripts and es/scripts.
  + form\_process.php
  + form\_throttle.php

Notes:

* Muse puts the form specific info in form-u916.php
* The other form-\* files are standard and don’t change from form to form
* The Adobe version of the files wants to use sqlite. That doesn’t work for some reason. It looks like it is never closing the db so when you come back around the db is locked.
* So, the changes integrated here are to use the SQL connected to the system.
* Therefore, you need to modify the scripts to put in the db name, user name, and password
* The throttling code limits to 25 the number of post coming from the same IP address in 2 hours. These are hard coded values and not magical.

Database schema assumed

* Table: feedback
  + id – INT, Primary key, autoincrement
  + formvalues – Text, not null
    - siteVisited: site name
    - workedWell: things that went well
    - needsToImprove: things to improve
    - contact: contact info provided, email or phone
  + timestamp – timestamp; updates whenever we touch the item
  + state – string, not null new, read
  + site – string, not null; name of site; also in the formvalues; is here so we can sort on it
  + datetime – string not null, when the item is created
* Table: throttle
  + id – INT, primary key, autoincrement
  + ip – varchar(39)
  + timestamp – timestamp

**To upload the Muse site to vitasa.org**

* In MUSE, export to HTML
* Copy to the contents to the en or es folder as appropriate
* Make sure you are using the correct Google Maps key (index and MFT)
* Copy sites.json as sitesjson.html into root (can’t apparently pull as json file from this site!)
* Copy forms-\* from scripts to en and es /scripts

>>>>> next: setup the db for user comments <<<<<<<<<<<<<<<

**Tests after upload and patches**

* verify pins on maps on both home and MFT
* post suggestion and verify arrival in DB
* if post suggestion doesn’t work, it is likely the names of the input and textarea’s in the forms

**Bugs:**

**ToDo**

* Figure a way around the cross-site reference so we can pull the sites.json from S3.
* Form post to DB; page to CRUD these posts