**Abstract**

This document details how to set up the website architecture as of May 14th, 2016.

This document assumes that a webserver is already up and running and that a public\_html folder exists and that your site points directly into it.

**Quick set-up**

1. Unzip the zip file into the public\_html folder
   1. mv *zipfile.zip* public\_html/.
   2. cd public\_html/
   3. Unzip *zipfile.zip*
   4. rm *zipfile.zip*
2. Create a database with table *events*, columns *id*, *name*, *comments*, *assetDirName*, *isActive* (int primary key ai, varchar, longer varchar, varchar, tinyint)
   1. In this project, we used MySQL.
   2. Update public\_html/include/dbConnect.php to reflect the *hostname*, *username*, *password*, and *database* to connect to the database.

**Things to install**

* The webserver is assumed to be up and running.
* PHP used was 5.3.3
* MySQL used was 5.1.73 with version 14.14.

**Frameworks**

The website is powered by AngularJS, Bootstrap, and Google charts. All three dependencies are hosted by CDNs.

* AngularJS relies on googleapis
* Bootstrap relies on bootstrapcdn
  + JQuery is needed for bootstrap, JQuery is on googleapis
* Google charts relies on gstatic (/charts/loader.js)

These should not need changing. If something is not loading, it may be that the host is down or that the hosted content is no longer hosted.

The links can also be changed to upgrade to another version or to support another version.

* AngularJS is version 1.4.9
* Bootstrap is version 3.3.6
  + JQuery is version 1.12.0
* Google charts uses the “current” version

**public\_html**

Inside public\_html should have several folders and an index.php file;

* API/ folder:
  + Hosts php files for interactivity
* app/ folder:
  + Hosts files used by AngularJS; this file is used most often for modification
* assets/ folder:
  + Contains data and images. There is a specific structure to follow with this. See below
* css/ folder:
  + Used to host custom css files
* include/ folder:
  + Files which get called often by API folders are often put in here; often times they are *required* by the API files such as database logins.
* Index.php file:
  + The file that gets loaded when pointed to a root directory. Could change in the future with routing

**Database structure**

There is 1 table called *events* with four columns: *id*, *name*, *comments*, *assetDirName, isActive*.

* Id is an autoincrementing primary key column.
* Name will be a string. An example of a possible value would be “May 2016 Structural test”
* Comments will be comments made for anyone viewing a particular page. It may be used in the future for instructions for a professor to write quiz questions with.
* assetDirName is the name of the folder that data and images are filed under with. Since this field is going to be implemented, the file structure mandates only filesystem legal characters/syntax.
* isActive is 1 if the test should be shown to the public, 0 if hidden to everyone.

**Interacting with the database**

In order for our webapp to communicate with the database, the login credentials need to be changed after the database has been made. In *public\_html/include/dbConnect.php*, update the respective *hostname*, *username*, *password*, and *database* fields with what you set up an account with.

**Assets folder structure**

Inside assets/ will be a set of folders that will relate to entries matching the database column *assetDirName*. In this example, there will be a test with an assetDirName of “test1”.

In assets/test1/ -- there will be three additional folders

* data/
  + In this folder will be a csv file. The CSV file will have no empty lines, and will have a first line dedicated to column headers.
  + It is currently a static “beamtestdata.csv”, it is possible in the future to just load the only csv file it finds in the specific folder.
* imgs/
  + These contain all the native photos for the test. They are uncompressed and often rather large.
  + They MUST be in the format of “IMG\_NNNNN.JPG”, where N’s are digits and there are as many N’s as needed, as long as number of N’s > 0
* resized/
  + Basically the same as the imgs/ folder except compressed for faster loading times. These images are referenced by the carousel so that they can load in a reasonable time.