Fraunhofer Database & Website

This file is a guide for anyone who means to make changes to the Fraunhofer Software, big or small.

This software includes three programs: - Tooling: for product tracking. - Purchasing: processing purchase orders. - Data Analysis: gathering and searching sample information.

The following programming languages were used in the making of this system:

- PHP
- HTML
- JavaScript
  - Good resource for JS
- CSS
  - Online tutorials for all of the above
- MySQL
  - SQL Free introduction class

Table of contents

Install

Below is a list of the programs that needs to be installed on a new computer to start working on the system. More thorough guide on how to get the programs working can be found at the end of this documentation. Note that the server computer should have all of this already installed and you can make changes to the live database without having to download anything.

MySQL workbench

MySQL workbench is a IDE used to write MySQL code. Free download from this link.

MySQL community server

MySQL community server is a free download that ables you to use MySQL on your computer. Download it from this link.

A guide to get started can be found here.

MAMP

To have a running website connected to a MySQL database use MAMP. You can download it here for free. MAMP setup guide can be found here.

CRON JOB

In the purchasing database we have an event that checks if there is any purchase order worth over $1000 that is expected to be delivered in 5 days and send an email to the employee who is expecting that order. In order to do that we set up a cron job which runs the "dailyScript.php" file once a day. Write: `crontab -e` in terminal and in there should be `30 11 * * * php /path/dailyScript.php` the 'path' part is the full path to that script so if you have to move the server to a different computer then you also have to set up this cron job. the `30 11 * * *` means that this happens 11:30 AM every day. To change the date to lets say 2:00 pm then you would write `0 14 * * *`.

Jshint and Uglify

JSHint was used to detect errors and potential problems in the JavaScript code. Uglify was used in this project to minify all JavaScript files. This is done so that the website is as quick as it can possibly be, since the computer in the lab is outdated and the internet connection might get slow this is very important. While developing it is better to include the JavaScript file you are making changes to in the HTML/PHP views, since compiling with uglify takes a few seconds and if you are making a lot of small changes and testing them those seconds add up quickly. A more detailed guide on how to make changes on JavaScript code can be found here. To run those tools we used Grunt.

Getting started

Accessing the code

The source code is stored on the shared folder under Perla Osk Hjartardottir and is called Fraunhofer. A source code management system called GitHub was used during development of this project. It is not necessary that you continue using it but we found it very helpful to look through my changes and it also serves as a backup for the code. At the time writing this file the code is stored here. If you want to continue using GitHub just make a new project on your account and push the code from the shared folder on to it.

Database architecture

The entity relation diagrams for this system were designed using draw.io and can be found on the shared folder under Perla Osk Hjartardottir. To change the ER-diagram just go to draw.io and open the `.xml` files that are stored in Perla's shared folder. If you are not sure how to read a a ER-diagram this documentation is a pretty good source on relationship patterns in SQL. There are many ways to show relations in a ER-diagram, we decided to use the following:

- ---------- is a zero-to-many relationship.
- --------> is a zero-to-one relationship.
- ====== is a one-to-many relationship.
- =====> is a one-to-one relationship.

Changing the code

Fixing minor errors

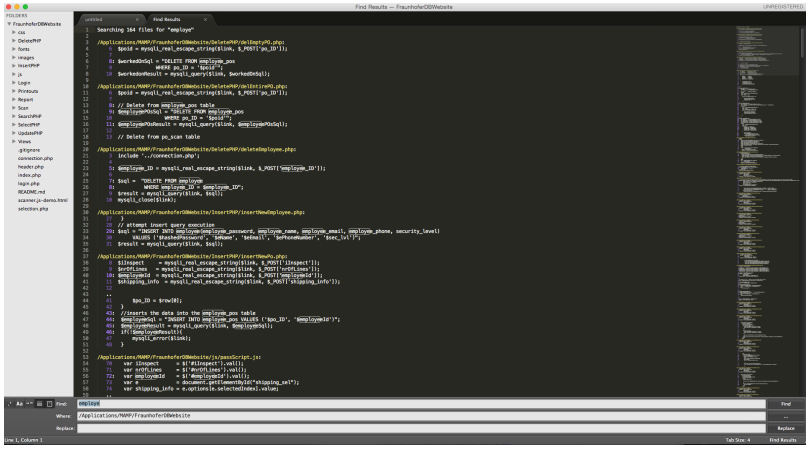
To fix minor things like spelling errors, incorrect grammar etc. Follow these steps:

1. Go to the server computer
2. Open Sublime Text 2
3. Find the source code. At the time writing the code is stored on `CCL/Shared/Perla Osk Hjartardottir`
4. Drag the folder called `Fraunhofer` in to Sublime Text 2
    - **Now you should have access to every single line of code in the project so make sure to not change anything else!**
5. Press `⌘-Shift-f` to search all the files for the text you want to change
    - Type in the search string you need to change
    - After pressing `Enter` a new file will open up showing **every single instance** of the text searched for in the project
    - To search only this file press `⌘-f`. If you can make your search more specific you will have less files to look through
6. When you find the text you want to change double click the yellow string above it, this is the file name containing the text.

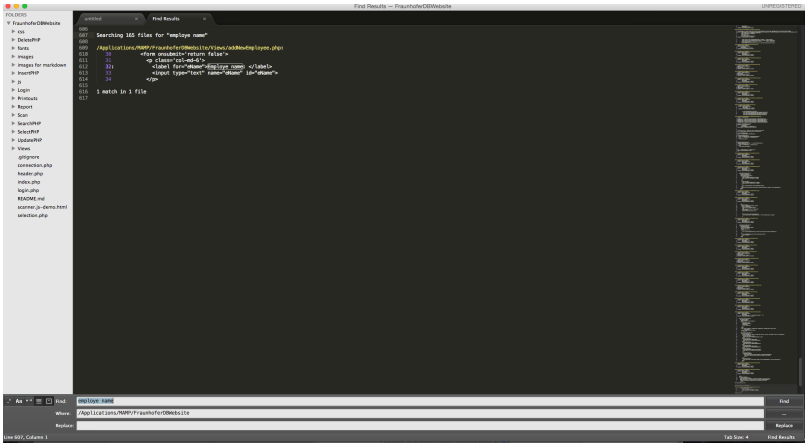
7. Now the correct file should be open. Search this file for the error that needs to be fixed, you can do this either manually or by pressing `⌘-f` and typing the text again.
8. Fix the error and save the file, `File->Save` or `⌘-s`
9. Check the website and make sure that the changes are correct

The pictures display how to fix a spelling error of the word Employee

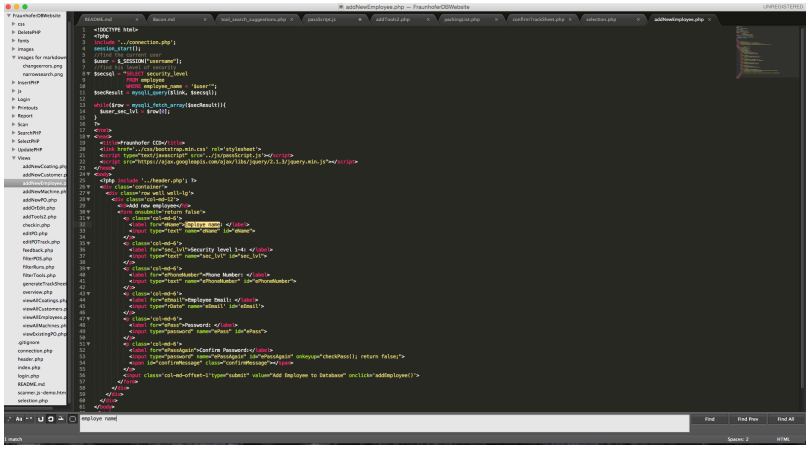
Step 5



Step 5.3



Step 8



Changing database tables/procedures etc.

Interacting directly with MySQL should only be needed if you need to do any of the following

1. Add a new table to the database.
2. Add a new field to a table.
3. Adding/changing procedures, triggers or functions.
4. Testing queries before adding them to the website.

To make changes to the database tables, procedures, triggers or functions it is required to have some background knowledge on databases so this guide will focus on explaining how and where the code is stored. To make changes to the live database you have to go to the server computer, open MySQL workbench and connect to the server called `Live Fraunhofer Database`. When a page called `Query 1` opens type in `Use Fraunhofer;` in the editor and you are ready to start making changes.

We highly recommend trying all changes first on a mock database! A guide to set up a mock database can be found here.

To change SQL code for other things like inserting, deleting, updateing or selecting this code is stored in the PHP files in the project folder and can be changed from there, but again, you should test the changes on a test website before making changes to the source code.

Changing Javascript or jQuery functions

Currently only Tooling and Data Analysis databases use NPM and Grunt, so these directions only apply to the these database. You need to be located in either program directory before you run the 'grunt' command

To edit the js files you need to run a 'grunt' task after you change the files. You do that by navigating to the project folder location in the computers terminal and typing in `grunt`.

To run a Grunt task you need Node: Node.js

Npm: Npm beginners guide

And Grunt: Grunt - Getting started

Within the project folder, you need to run 'npm install' and that should install grunt and every grunt dependency you need. It does that because we have specified those dependencies in the package.json file.

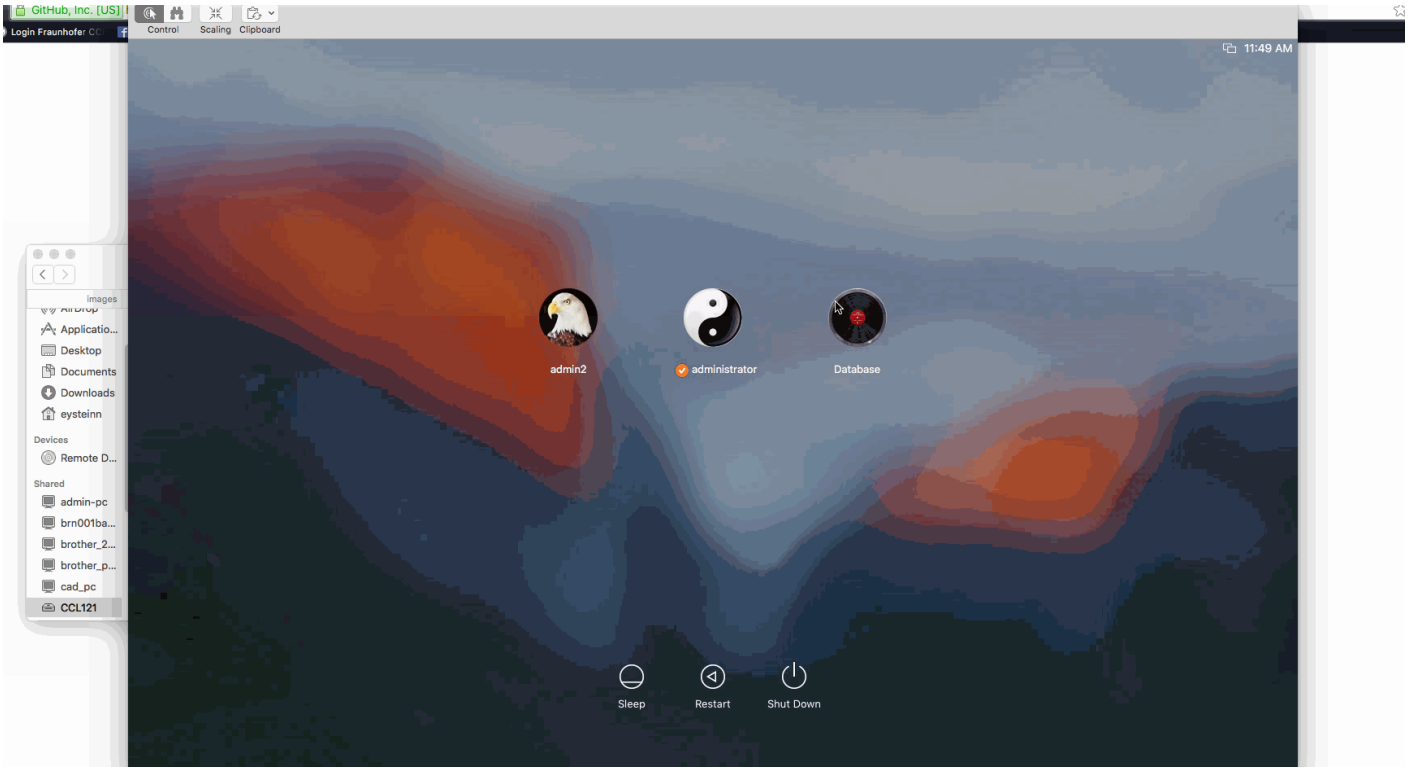
After that you should be all set to simply run 'grunt' in the terminal and that should compress all the javascript files in the js folder to the minified js file in the 'dest' folder. As well as checking if your JavaScript code is 'lint free' with the Jshint tool.

Instead of running the grunt command in your terminal after each javascript file modification, you can write 'grunt watch'. After that, every time you save your changes in a javascript file, the 'grunt' command is executed automatically.

The server computer should have everything installed for you to simply run the grunt command in the terminal after you edit the js files

Uploading changes to server

Originally this database was just hosted locally on the developers computer. As of November 2015, it is hosted on a server which Robert Rechenberg administrates. That means that you can host a testing database locally and when you are ready to publish the changes you replace the Fraunhofer file on the server with your Fraunhofer file. I like to put the recent version of the PHP/JavaScript in a backup folder first and then move the new version over.



The PHP/JavaScript/HTML files are all hosted on the Database guest account on the server. However, the MySQL code is all hosted on the Admin account so if you need to manipulate the database directly you will have to talk to Robert about that. Update 09/14/2016: All code is now hosted on the Admin account on the server. To upload your changes copy your local Fraunhofer folder to a usb stick and bring it to Robert. The parent folder is called Fraunhofer which includes two folders, one called Fraunhofer and another one called Fraunhoder Uploads. The former contains all the source code for the website the second all uploaded files by users of the Data Analysis Database.

I recommend exporting the MySQL code from the server Admin account regularly and import to your local SQL, so that your test database remains similar to the public database.

Guides

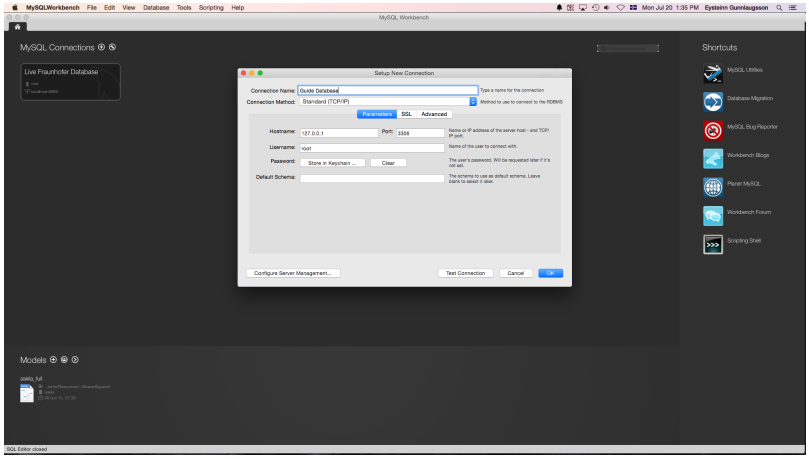
MySQL setup guide

After both the MySQL community server and MySQL workbench have been installed you can set up a connection to a new database.
To allow MySQL connections on OSX, go to system preferences and search for MySQL. From there it is possible to start a connection. To fill this database with the same data as is on the real running Fraunhofer database take the newest database dump (at the time writing it is stored on the shared drive under Perla Osk Hjartardottir. File name 'Dump + date created') and import it to a new server. You might need to add `USE Fraunhofer;` to the top of the file if you get an error saying that this database does not exists (see picture 5). You can now play around with this database to get to know the system without it having any effect on the real running database. If you want to start a new database with only the structure and without any data you can use the structure dump file 'StructureOnlyDump + date created'.

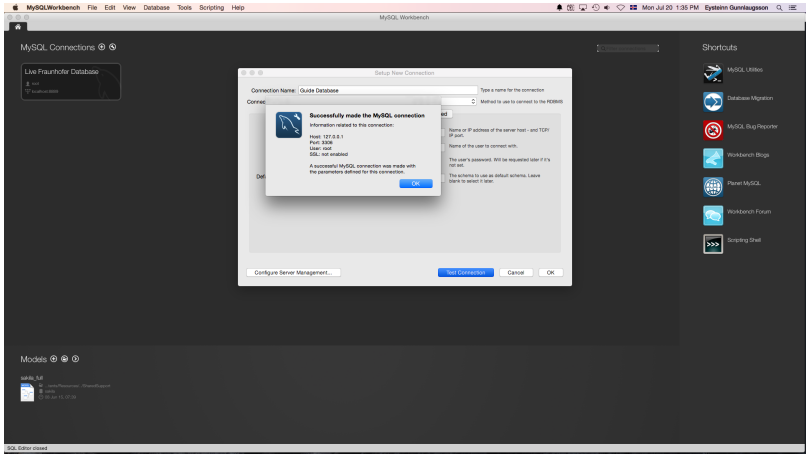
While developing I highly recommend testing everything on a mock database before saving the changes to the source code for the project.

Here is a picture guide on how to create a copy of the real database.

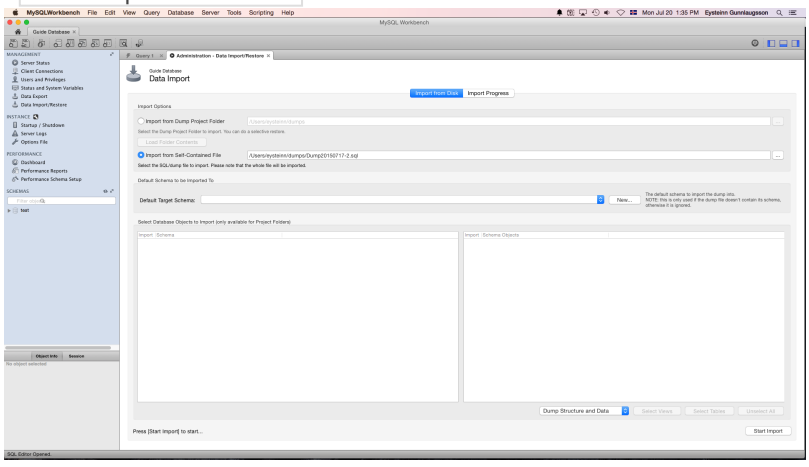
1. Create a new connection in MySQL.



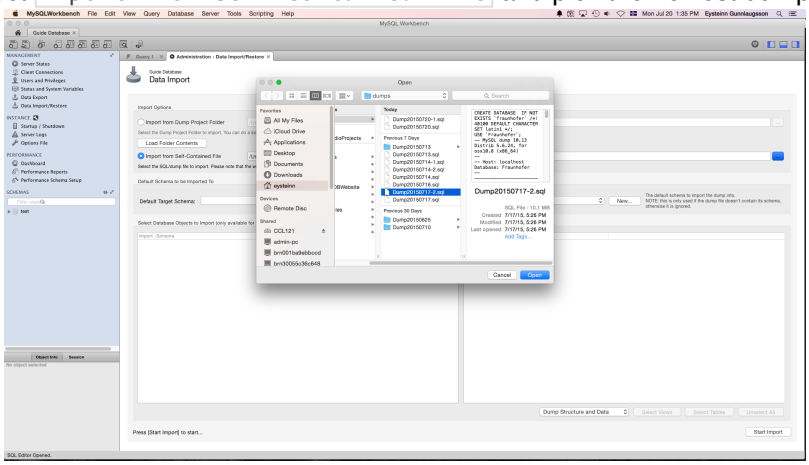
2. Click test connection to make sure everything is OK.



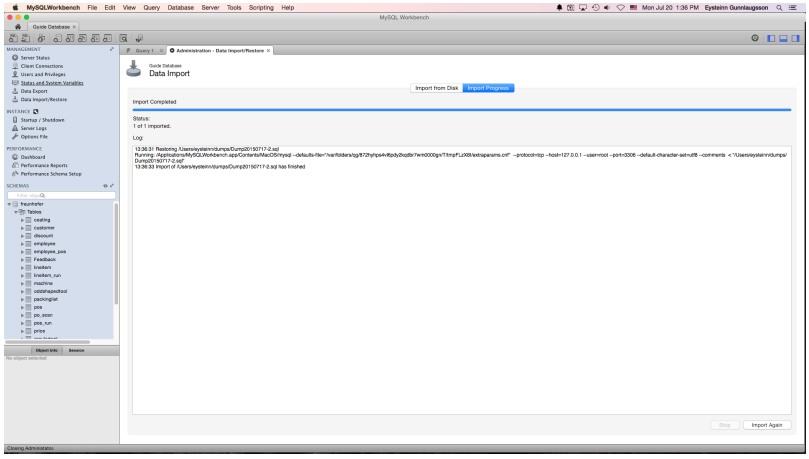
3. Click `Data Import/Restore`.



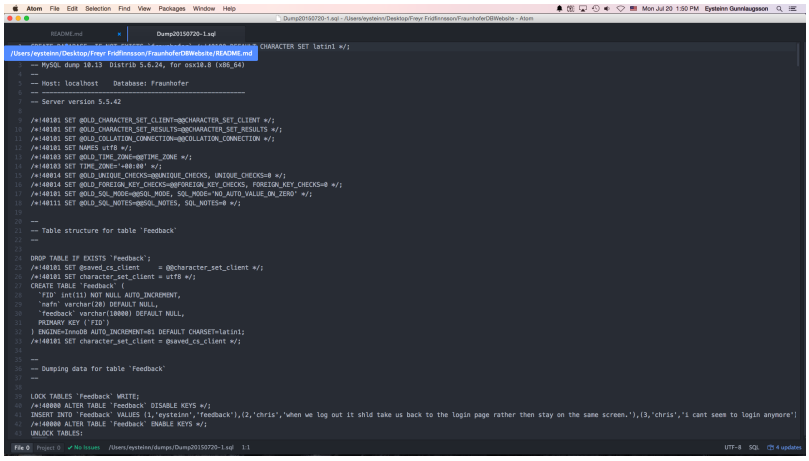
4. Select `Import from self contained file` and pick the newest dump file.



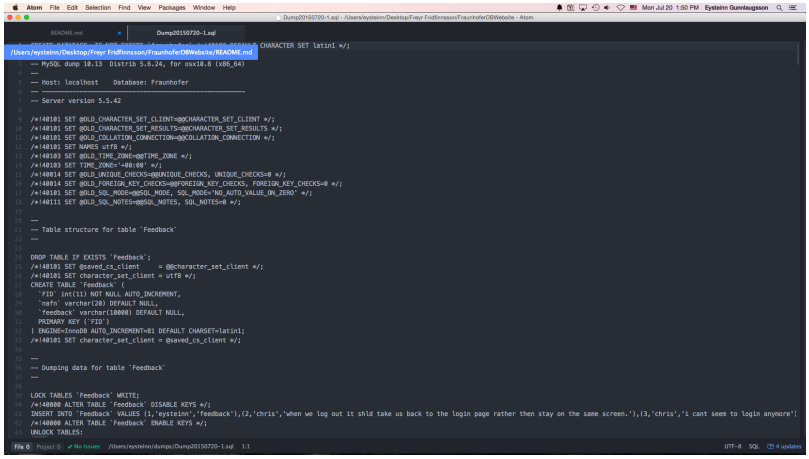
5. If there are no errors you should get a message like this.



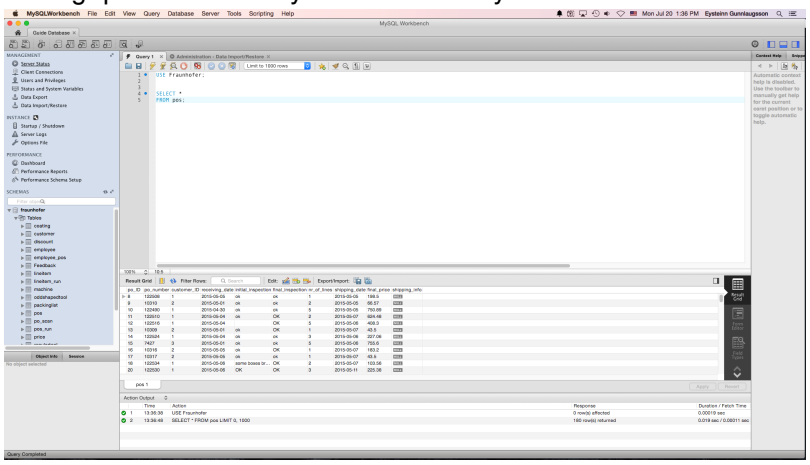
   - If you get an error saying that this database does not exists do the following changes to the dump file.
     - Before



- After (see line 23):



6. Try running queries to see if you have data in your database.



MAMP setup guide

Setting up MAMP is pretty straight forward. It is a good idea to set MAMP up to a mock database so you can test code there instead of the live database. After you download MAMP and setup MySQL there are a few settings that need to be configured:

- Set apache port to any available port. This will be the number you type in the URL to connect to the database. 8889 or 8887 should be free
  - It might be a good idea to pick a port number very different from the actual database so you do not accidentally mess with the live database. I used 3306 for my mock database. The live one is 8888
- Set MySQL port to the same port as your mock database
- Set PHP version to 5.6.2
- Set Web Server to Apache
- Set the document root to the project folder

The URL needed to access the webserver is a combination of three things:

- Your computers name or IP-address
  - You can find your computer name by going to `System Preferences/Sharing`. At the top you can see the computers name and an `Edit` button. Click the edit button and there you can see your computers webserver name. The one I use is `eysteinn.local`

    - You can see your IP-address here
- The port number you picked for MAMP, not the MySQL port number
- The filepath from the project root folder to the file you want to open

The URL should look something like this `http://35.9.146.192:8888/Fraunhofer`, `http://eysteinn.local:8888/Fraunhofer` or `http://localhost:8887/Fraunhofer/`.

You should now have a connection to the database you just made through Google Chrome. You can only have one active web server running per computer so do not change this if you are working on the server computer since it will remove access for other computers to the main database!