# Yoctopuce-Plot

## Getting started and things to do

You will need to complete these tasks before being able to run the application. Also included in this documentation are some suggestions for improvement.

### Database

You need to create the database for Yoctopuce-Plot before running the application. Currently there is no script for constructing the database tables. The dbconnect.py file contains the application's configuration for connecting to the database. The database name should be changed to something more meaningful than “pythonprogramming.” It would be helpful for someone to develop a script that can set up the database.

The database contains two tables: users and user\_devices.

The users table can be created with the following MySQL statement:

CREATE TABLE users (uid INT(11) AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(20), password VARCHAR(100), email VARCHAR(50), settings VARCHAR(32500), tracking VARCHAR(32500), rank INT(3));

The user\_devices table can be created with the following MySQL statement:

CREATE TABLE user\_devices (uid INT(11) AUTO\_INCREMENT PRIMARY KEY, username VARCHAR(20), device\_name VARCHAR(100));

Could be improved by linking the two tables together via the username field. Deleting a user from the users table should automatically remove their devices in user\_devices. The username and email fields should also be set to unique.

Lastly, the dbconnect.py file should not be including in the git repository. It is bad practice to include passwords in a repository. An example dbconnect.example.py file should be included with example data instead. A comment in the file can instruct the user to rename the file to dbconnect.py.

### Requirements

The repository contains a requirements.txt file. This file lists all of the Python libraries necessary for running the application. Create a virtual environment with either conda or virtualenv and install the requirements before attempting to run the application. This method is preferred over installing all of the libraries on your system's Python environment.

At this point you should be able to launch the application.

### Registration

The registration form needs to be revised. For example, the email field will accept any string regardless of whether or not it is an email address. There are no rules to encourage a stronger password. All user inputs need to be validated and cleaned of any problematic characters.

### Plot pages

After registering a user can view the plot pages. If the TinyDB database that records sensor data is empty, these pages will cause an Interal Error. There needs to be checks for the database's current contents and an appropriate response. TinyDB is also not a great long term solution. Switching to a database such as MongoDB will ultimately provide more reliable behavior.

### Update to socketio

Currently the website works by having the plotting page constantly checking the server for any new data that needs to be plotted. Additionally the Bokeh plot is constantly refreshing during this process and it can create a flickering appearance.

A better solution would be switching to using flask-socketio (<https://flask-socketio.readthedocs.io/en/latest/>). This would allow the server to broadcast new data to the clients only when new data has been received from the wireless hub. The Bokeh plot should then update with the new data point without redrawing the entire plot.