

CMPSC 431W Spring 2020

Project Setup Tutorial

Overview

This tutorial aims to provide information about setting up the development environment to be used for project management and database application development in Phase 2, throughout the rest of the semester. Note that the project requires you to build a web site. The precise details of what you will be implementing will be shared later. The elements to be implemented include 1) the front-end web page; 2) the web/application server; and 3) the backend database. We first describe the system architecture of the database application in brief.

HTML (preferably coupled with Bootstrap) are used for front-end, which defines the look-and-feel of your website. For the programming language, we suggest you use Python. The installation instructions can be found in the section entitled “Install Python”. In addition, we recommend FLASK, a Python-oriented web development framework which helps to productively implement your website functionality. FLASK provides an API to receive requests from the website (such as form submissions, clicks, login request, etc.) and return responses back to the front-end (such as display the result of a login – success or failure, display information as requested by the user, etc.) Moreover, all SQL queries are issued by FLASK to the backend database in response to user requests. For example, consider an application function where a student asks for his grades to be displayed, FLASK issues an appropriate SQL query to the database server, which retrieves the data and return it to FLASK, which in turn sends the result (with proper rendering) to the client (browser) for display.

SQLite is suggested for use as the database management system in this project due to its portability. SQLite works by maintaining a local “database file” which will be queried by your application (i.e., the website). SQLite3 (a version of SQLite) execute a query and returns its result to FLASK. In the FLASK framework, there are some ‘@’ statements to define the front-end requests in various situations (login, display grades, show professor details, etc.) The functions under @ are called when the corresponding requests are received. That function in turn submits the corresponding query to the database and returns the result back to the front-end. Our starter code covers this routing mechanism.

In our starter code we use `render_template()` and `request()`. `render_template()` uses [Jinja2](https://jinja2.pocoo.org/), a full featured template engine, making this method very powerful because all you need to do is to provide the HTML template and the variables you want to pass. Jinja2 is inspired by Django’s templating system but with extension to an expressive language. **For the project assignment, this method saves you a lot of effort on communicating with front-end and back-end.** `request()`, for access inside the template, is a global object for taking requests from the webpage. We show access of the data in our starter code using the `form` attribute. For more details of FLASK, here is a link of its docs: <http://FLASK.pocoo.org/docs/1.0/>.

In this project, we recommend PyCharm as the IDE. PyCharm installation is instructed in the corresponding sections, as the self-explained section titles indicate. Finally, in the last section, you use the environment you set up (as described in the tutorial) for exercise. You need to correct the errors in the code and complete a commit-and-push. This whole set-up may take a while to complete for the first-time user, but it is very useful not only for reducing your effort/workload for the project but also beneficial later in your career.

1. Install Python

You should install Python if you don't have it installed on your computer. For the operating system that does not have a pre-installed Python package, you may download an installer from the [Python official website](#). For our course project, Python with version 3.5+ is recommended.

To make sure your Python installation functions properly, open your terminal and type:

```
1. Python3
```

Or, if that does not work, just type:

```
1. Python
```

Here is what you are expected to see:

```
1. PS C:\Users\431W> python
2. Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
3. Type "help", "copyright", "credits" or "license" for more information.
4. >>>
```

If you have trouble running the interpreter to check your Python version, please ask us for help via Canvas.

It is important to confirm that Python is installed on your system. To exit the interactive prompt, you can type:

```
1. exit()
```

and press Enter. On the Mac OS X and Linux versions of Python, you can also press Ctrl + D. On Windows, the exit shortcut is Ctrl + Z, followed by Enter.

2. Install PyCharm

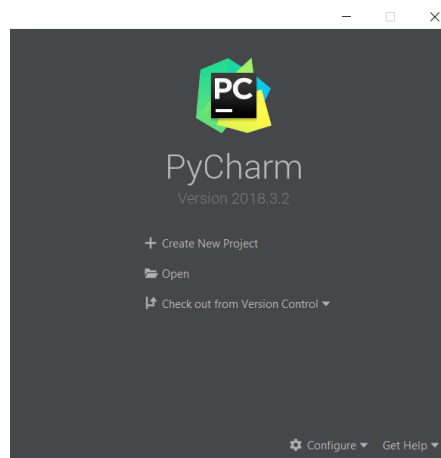
PyCharm is an integrated development environment for computer programming, specifically designed for the Python programming which are used for our project. PyCharm, supporting Python FLASK web framework along with frontend (client) and SQL (database) support, provides an ideal full-stack IDE for Python. PyCharm is compatible with Windows, MacOS, and Linux. Developed by the Czech company JetBrains, it is eligible for free license with a Penn State account.

For our project in this course, we recommend using PyCharm to complete your Phase 2 implementation. You may choose to use your preferred web development tools on the project assignment, but it is your own responsibility to make sure you are familiar with the tools you choose.

First, go to <https://account.jetbrains.com/signup> to register your account with PSU Email. **You need to remember your password since it is needed during the installation.** Please follow the instructions, go to <https://www.jetbrains.com/pycharm/download/> to download the software.

It is important to download PyCharm Professional. The machines in the CSE Computer Labs in Westgate are already equipped with the Professional version.

After the installation completed, you shall see the prompt as shown below, when you run PyCharm:



***For those who use machines in the lab:** you will need to bring your PSU ID card to enter. Also, you need to login with CSE password, which may be different from your Penn State Access account. If you need assistance on the account, please go to ITS Help Desk located at W100 Westgate.

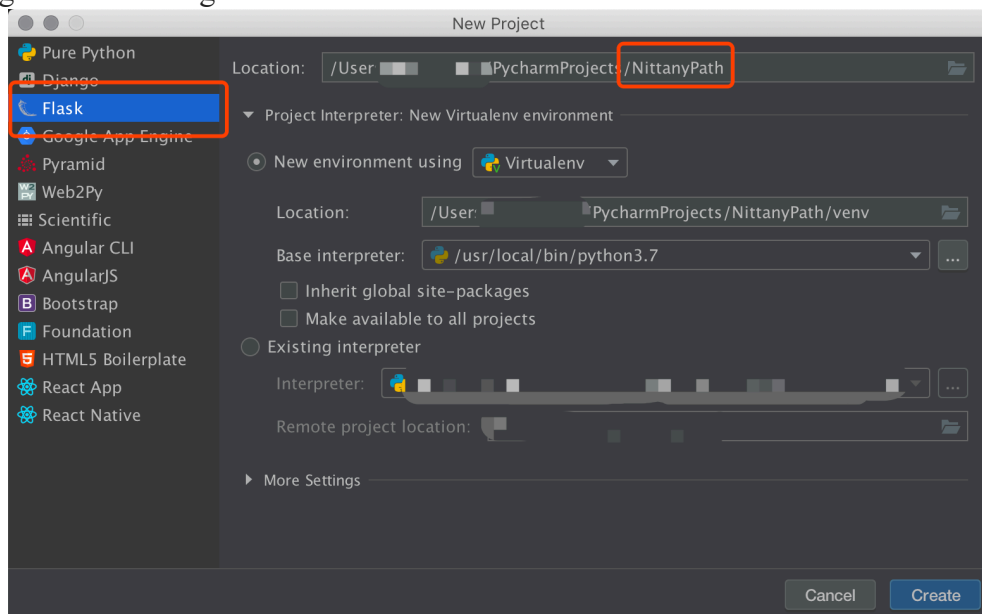
3. Create the Project

Now go back to PyCharm:

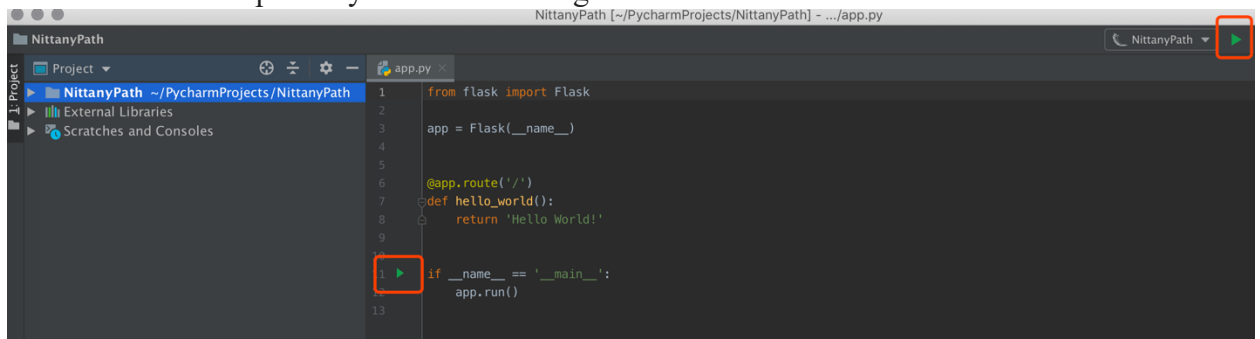
1. Click on Create New Project from the welcome window.



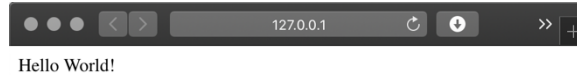
2. On you left panel, select **Flask** Configuration and name your project. Then choose the base Python interpreter from your system. It is recommended to create a new virtual environment to isolate the system environment. Click **Create** and you will see some packages are installing.



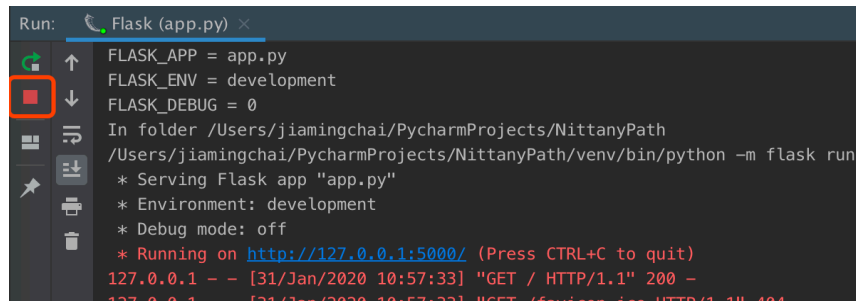
3. You should be able to see a workspace after setup is complete. Currently it includes a “Hello World” sample for you to run. Click green arrow to run.



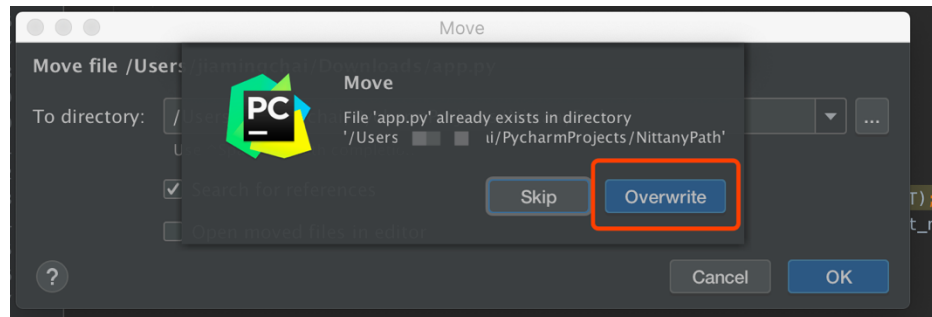
4. You should see an output window pops out, then click the local host url to test it. Your system browser will direct you to the page and Hello World shows on the top left corner.



5. After Hello World successfully shows up, go back to PyCharm and click the red square to stop it. Every time you want to see your change reflected on the webpage; you must rerun your project.



6. Now open the starter code you downloaded from canvas. Drag and Drop index.html and input.html under templates and app.py under project directory. Overwrite app.py if it asks.



7. Now rerun the project to see the new webpage where it asks first name and last name. Enter the name to see if the name can be inserted successfully. You can type a few names to see if it works properly. We use Bootstrap library and it has a very good documentation (<https://getbootstrap.com/docs/4.4/getting-started/introduction/>) on how to make your webpage look pretty.

Enter Your Name

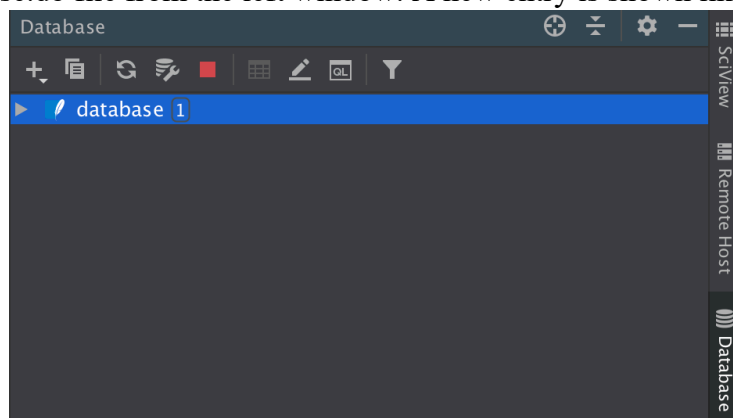
First Name

Last Name

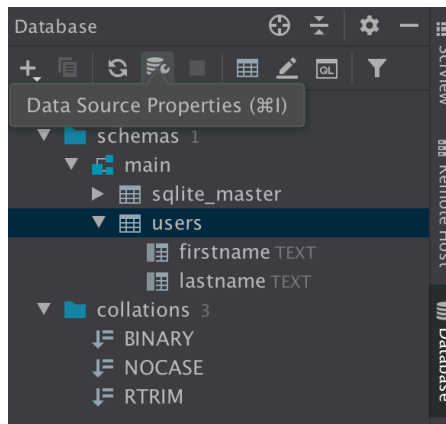
History

First Name	Last Name
ABC	1234
Test	123

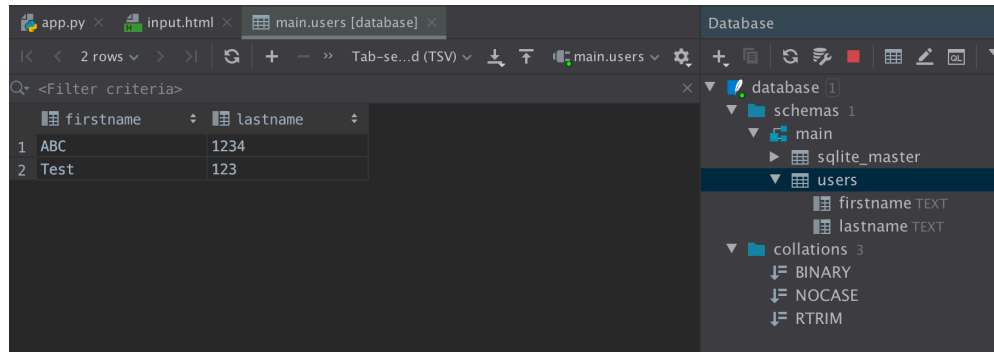
- Going back to the PyCharm, a new file called database.db is under your project directory. We can “visualize” the database by opening **Database** window on the right and double click the database.db file from the left window. A new entry is shown like this:



- Some machines do not have SQLite driver pre-installed. Click **Data Source Properties** to install the missing driver.



- Expand it and find the table called **users**. That is where we saved our first name and last name. Double click **users** you will be able to see all the names. This is where you “manage” the database visually.



The screenshot displays a database management interface. The main window shows a table named 'main.users' with two columns: 'firstname' and 'lastname'. The table contains two rows of data:

	firstname	lastname
1	ABC	1234
2	Test	123

The right sidebar, titled 'Database', shows the database structure. It includes a 'schemas' section with a 'main' schema containing a 'sqlite_master' table. Below this, the 'users' table is listed with its columns: 'firstname' (TEXT) and 'lastname' (TEXT). The 'collations' section shows three collations: 'BINARY', 'NOCASE', and 'RTRIM'.

11. Now the setup tutorial is complete. If you need help on the code and feel confused about Flask, please refer to the tutorial video on Canvas or message TAs for specific questions.