# Log-In Form Project Setup

How to set up your development environment using PyCharm

This tutorial starts your journey into creating database driven web applications. You will learn how to use the PyCharm to:

- ☑ Connect to GitHub and download a repository
- ☑ Set Up a virtual environment and add packages to run the program
- ☑ Configure environment variables and run the application in a web browser

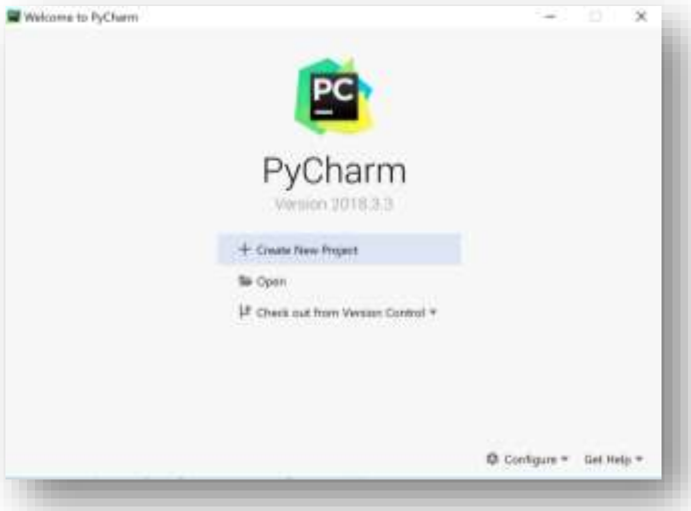
## Before you start

Ensure that you have the following programs installed on your computer:

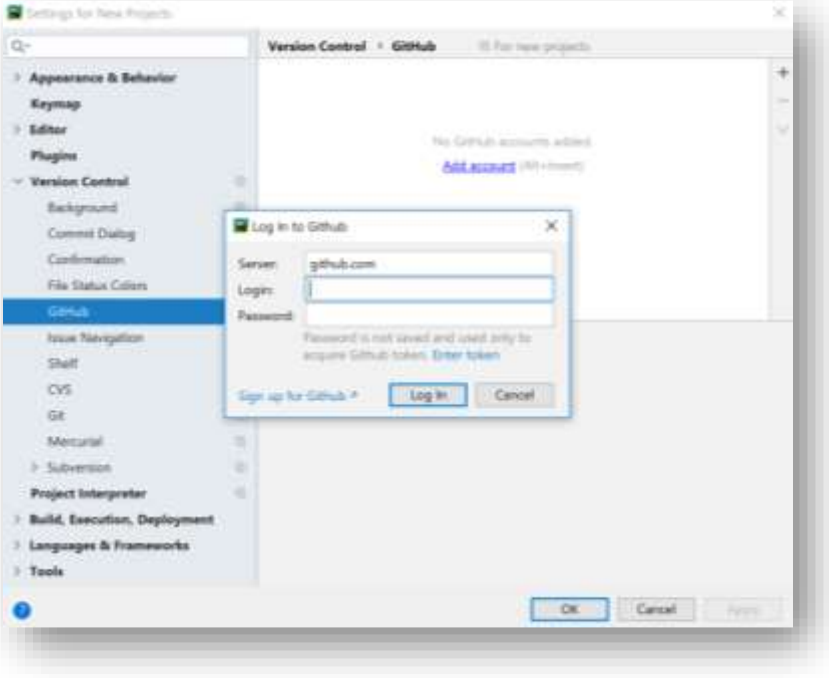
- ☑ Python 3
- ☑ PyCharm
- ☑ DB Browser for SQLite
- ☑ Brackets
- ☑ Git

## Setting up GitHub

Open git hub on your windows machine:



☑ **Step 1:** Click Configure → Settings



Server: github.com

Login:

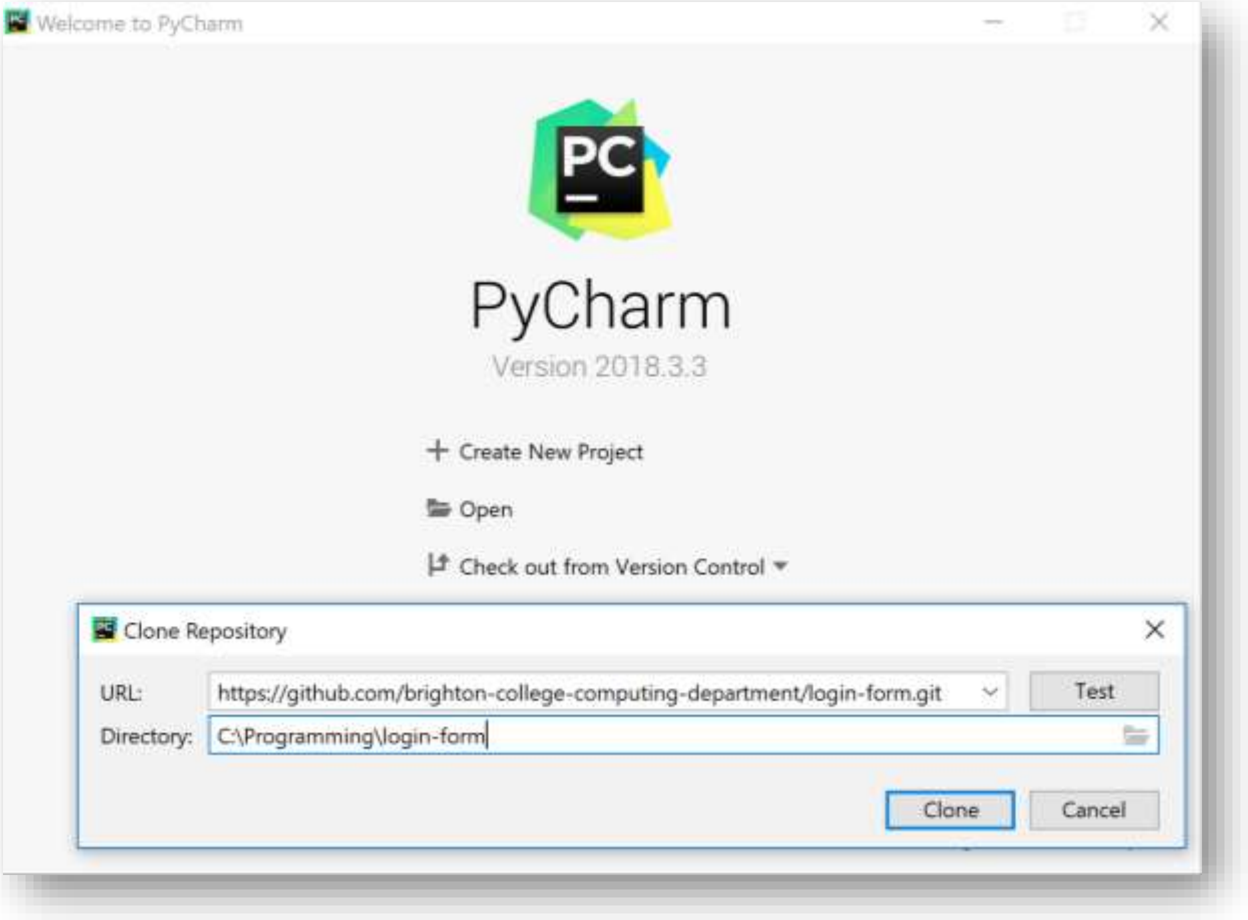
Password:

Password is not saved and used only to acquire Github token. Enter token

☑ **Step 2:** Click Version Control → GitHub → Add Account then enter your account details and *Log In* to the account.

## Cloning the GitHub repository

All the files that you need for this project live on the github code repository. They can be downloaded manually from https://github.com/brighton-college-computing-department/login-form but it is easier to set up if you let PyCharm do it for you.



URL: https://github.com/brighton-college-computing-department/login-form.git

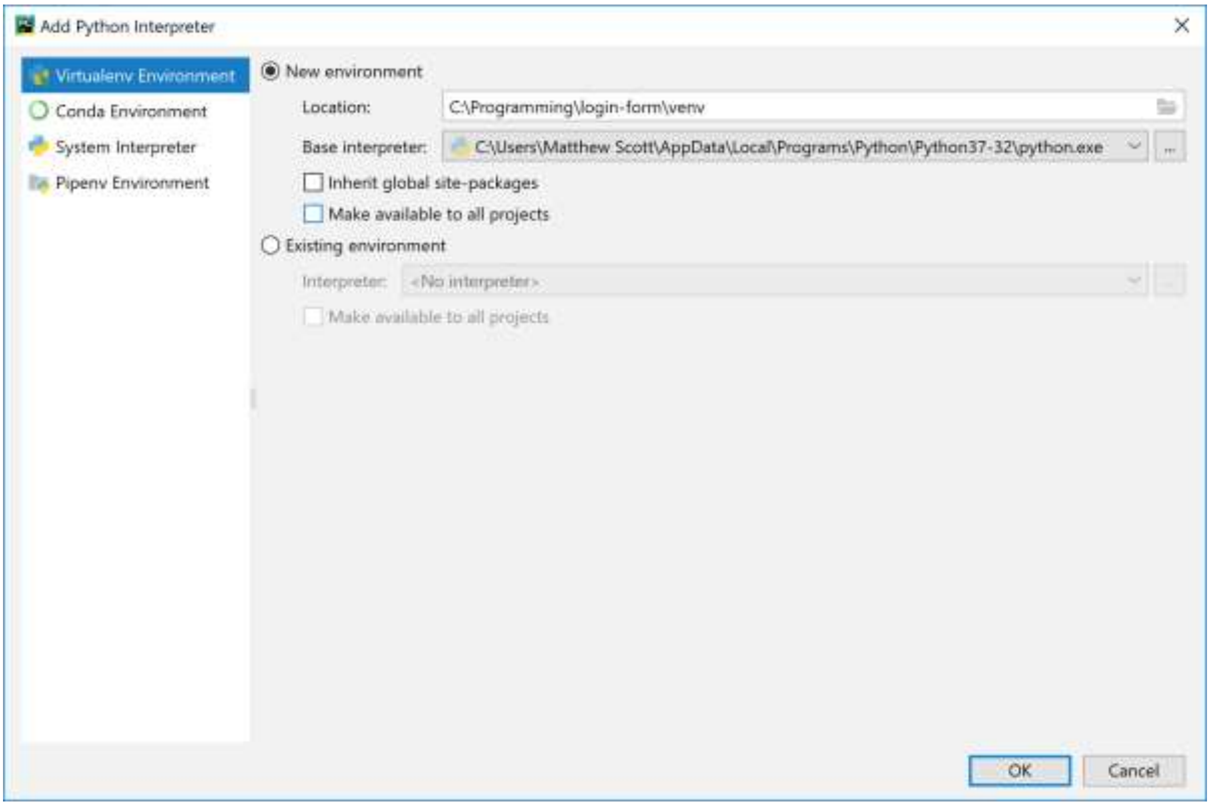
Directory: C:\Programming\login-form

☑ **Step 3:** Select *Check out from Version Control* → Git. Type the URL in as shown below. Choose the folder where you keep your programming projects then add the name of the project (login-form). Click *Clone*

## Set Up the Virtual Environment for the project

A virtual environment is where you create a stand-alone programming environment for a given project. Programmers do this as it prevents problems occurring when packages that a program is using become updated and over time the source code is no longer compatable with the packages that it is dependent on. This project will use Flask. As such we will create a folder, venv, that will store the current version of python along with the package, flask, that we are using and all of the other packages that flask is dependent upon.

☑ **Step 4:** Select File → Settings (Ctrl+Alt+S) → Project: login-form → Project Interpreter. Select the cog button and add.



Add Python Interpreter

- Virtualenv Environment
- Conda Environment
- System Interpreter
- Pipenv Environment

● New environment

Location: C:\Programming\login-form\venv

Base interpreter: C:\Users\Matthew Scott\AppData\Local\Programs\Python\Python37-32\python.exe

□ Inherit global site-packages

□ Make available to all projects

○ Existing environment

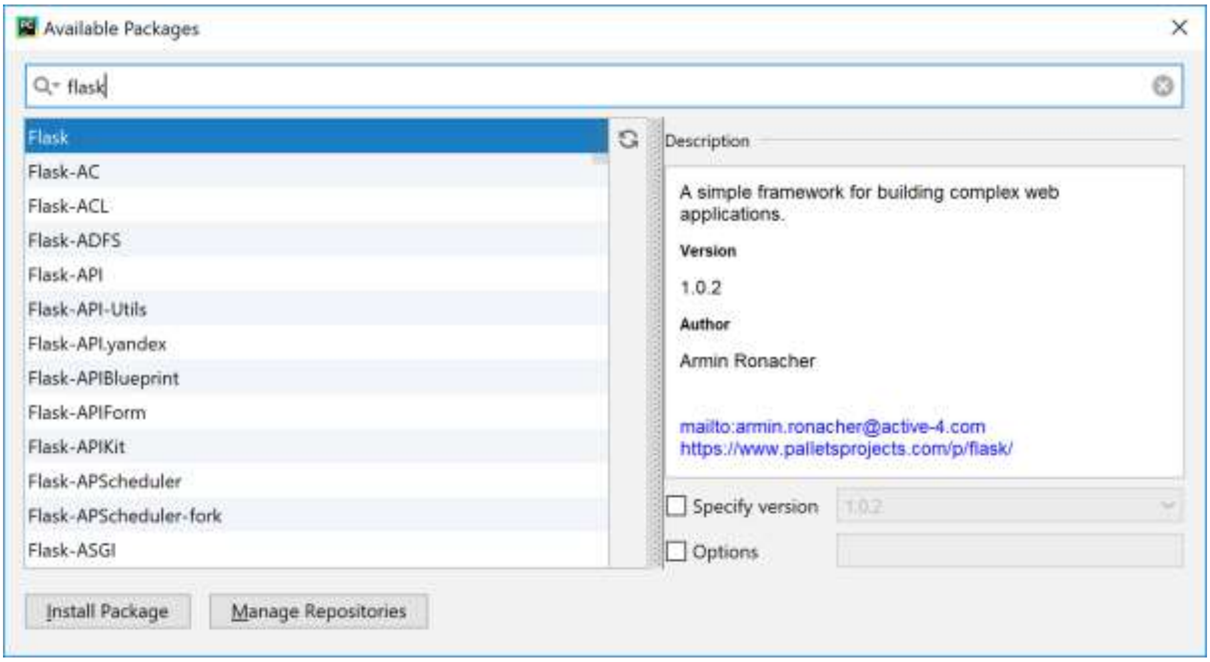
Interpreter: <No interpreter>

□ Make available to all projects

OK Cancel

☑ **Step 5:** Ensure that you have selected the correct directory and added venv on to the end of the path name then click OK **twice.**

☑ **Step 6:** You now need to add flask to the environment. Click the plus button (+), search for **flask** in the search bar and then click install package.



Available Packages

flask

- Flask
- Flask-AC
- Flask-ACL
- Flask-ADFS
- Flask-API
- Flask-API-Utils
- Flask-API.yandex
- Flask-APIBlueprint
- Flask-APIForm
- Flask-APIKit
- Flask-APScheduler
- Flask-APScheduler-fork
- Flask-ASGI

Description

A simple framework for building complex web applications.

Version

1.0.2

Author

Armin Ronacher

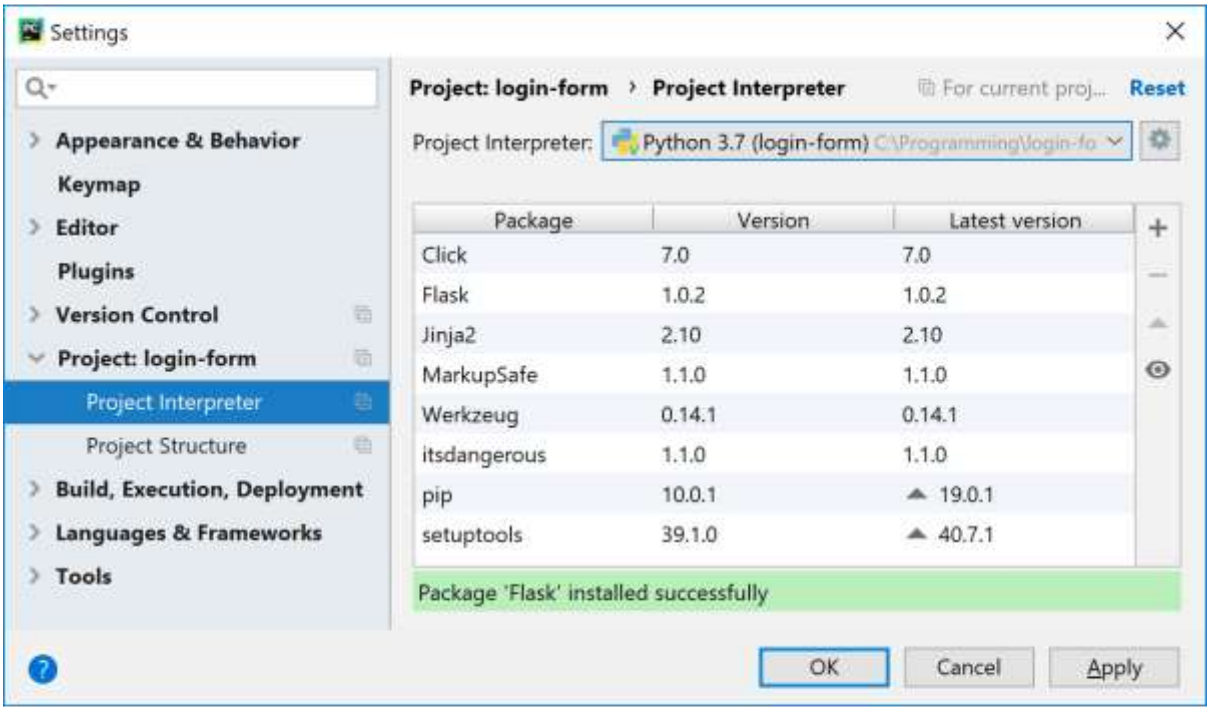
mailto:armin.ronacher@active-4.com
https://www.palletsprojects.com/p/flask/

□ Specify version 1.0.2

□ Options

Install Package Manage Repositories

You will note that the pip program that installs packages from the python package index (pypi) also has installed all of the other dependencies. These are the packages that flask in turn uses:



Project Interpreter: Python 3.7 (login-form) C:\Programming\login-fo

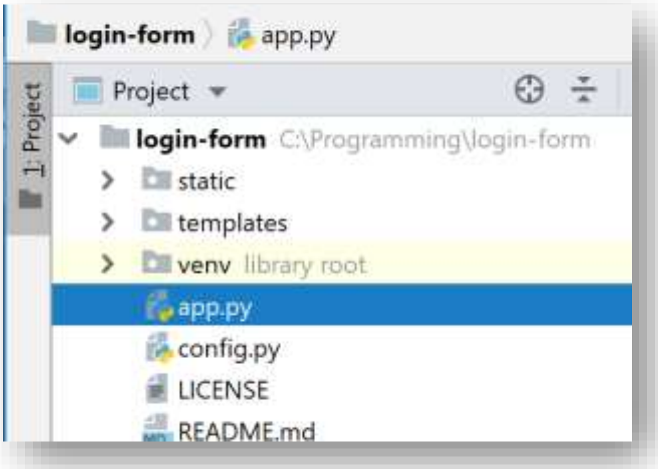
| Package | Version | Latest version |
|---|---|---|
| Click | 7.0 | 7.0 |
| Flask | 1.0.2 | 1.0.2 |
| Jinja2 | 2.10 | 2.10 |
| MarkupSafe | 1.1.0 | 1.1.0 |
| Werkzeug | 0.14.1 | 0.14.1 |
| itsdangerous | 1.1.0 | 1.1.0 |
| pip | 10.0.1 | ▲ 19.0.1 |
| setuptools | 39.1.0 | ▲ 40.7.1 |

Package 'Flask' installed successfully

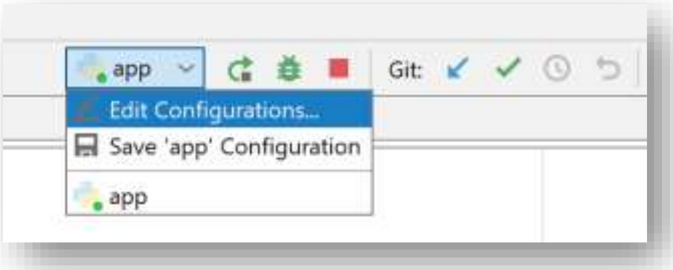
## Run in development mode

You are now almost ready to run your app. Before you do this you must add an environment variable.

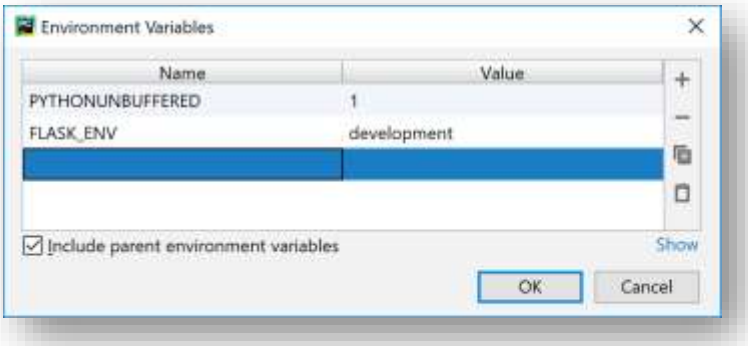
☑ **Step 7:** Select app.py from the project pane under login form:



☑ **Step 8:** Edit the configuration of how this module runs by selecting the option below:



☑ **Step 9: Click the folder next to the Environment Variables text box followed by the + button.** Add the following environment variable:



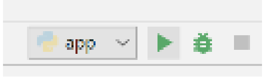
| Name | Value |
|---|---|
| PYTHONUNBUFFERED | 1 |
| FLASK_ENV | development |

☑ Include parent environment variables

The flask program will now run in development mode. This means that should you save you python file you will only need to refresh the browser to update the program, as opposed to restarting the program each time.

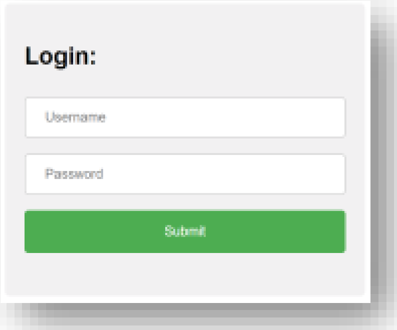
## Running the Login Form App

The final step is to start the application.



☑ **Step 10:** Click the run button on the top right hand of the screen. The console will open and you will select the hyperlink to http://127.0.0.1:5000/

The app should appear as below:


Login:
Username
Password
Submit