APPENDIX A - USER MANUAL

This user manual assumes that Python3 and PyCharm are installed.

Downloading the Tool

The complete tool developed for this project is available on GitHub at: https://github.com/CSIRO-enviro-informatics/shacl-form

It can be obtained by selecting 'clone or download' at the top-right and clicking 'Download ZIP' as shown in Figure 16.

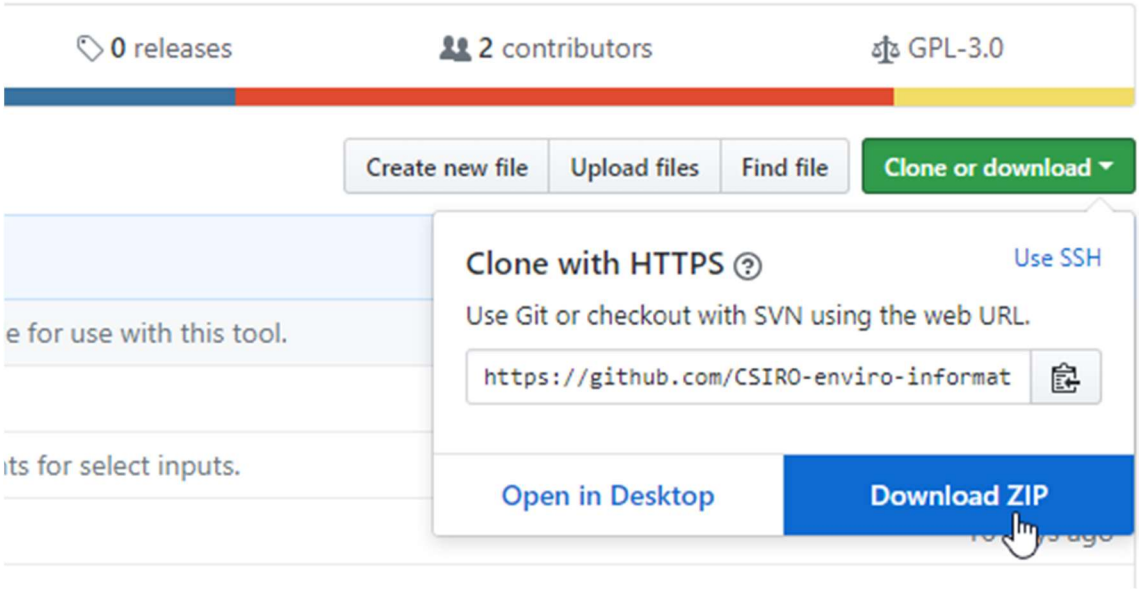


Figure 16. How to download this tool

After downloading, unzip the files by right-clicking and selecting 'Extract All…'.

Creating a Python Virtual Environment

Open PyCharm. On the start screen, click 'Open', navigate to the folder that was extracted and then click 'OK'. This will open the project.

At the top of the window, select 'File -> Settings…'. On the left side of the window, select 'Project: shacl-form-master -> Project Interpreter'. Click on the cog in the upper-right and select 'Add'. Click 'OK', then 'OK' again to exit the settings. A new virtual environment has been created.

Installing Required Modules

Open ‘requirements.txt’ to view the required modules of this project. A yellow banner will open at the top of the application listing the modules that are not yet installed. On this banner, click ‘Install Requirements’, then click ‘Install’ to verify. It may take a minute or two to install all the required modules.

Generating a Webform

In the ‘form-generator’ directory, right click the ‘main.py’ file and select “Create ‘main’…” (shown in Figure 17). This will open a configuration that will allow parameters to be supplied when running the application.

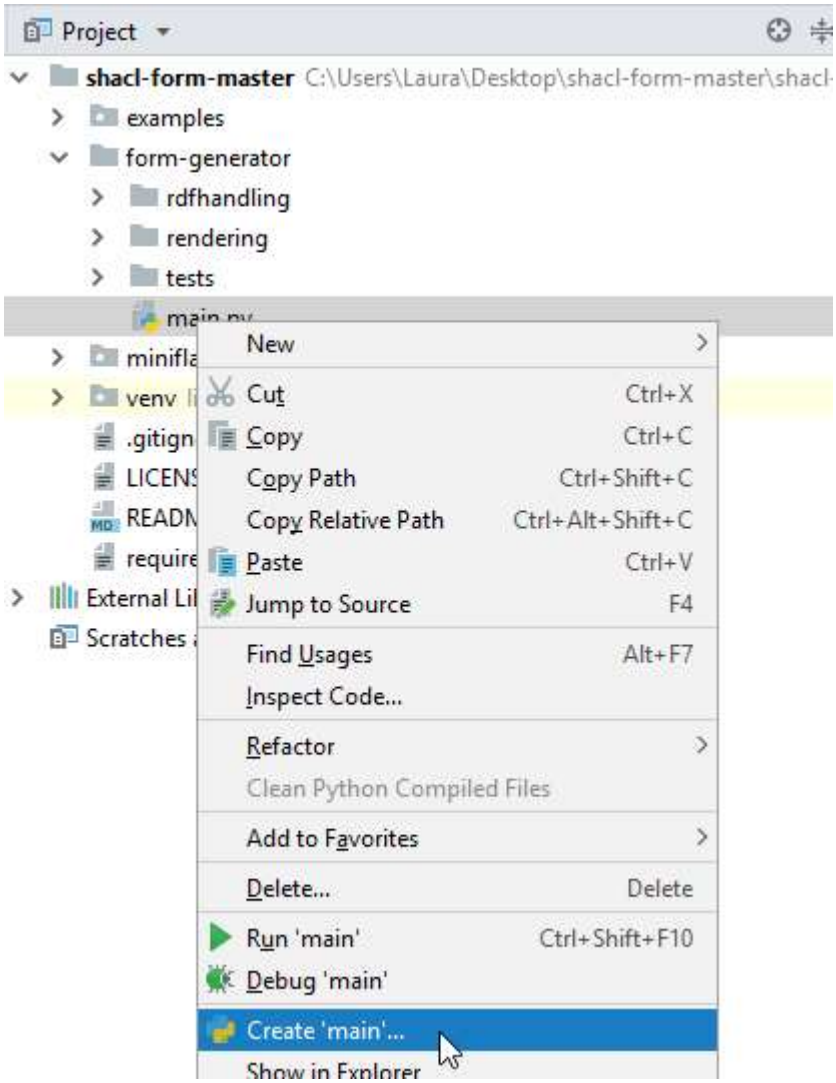


Figure 17. Creating a new configuration

The usage of ‘main.py’ is as follows:

python main.py <SHACL file path> <optional: destination>

In this guide, an example SHACL file that is supplied with the project will be used. This file is located at ‘examples/Person.ttl’. An optional destination will not be specified, since the default destination is ideal for this example. In the ‘Parameters’ input field, enter ‘../examples/Person.ttl’ as shown in Figure 18. Click ‘OK’.

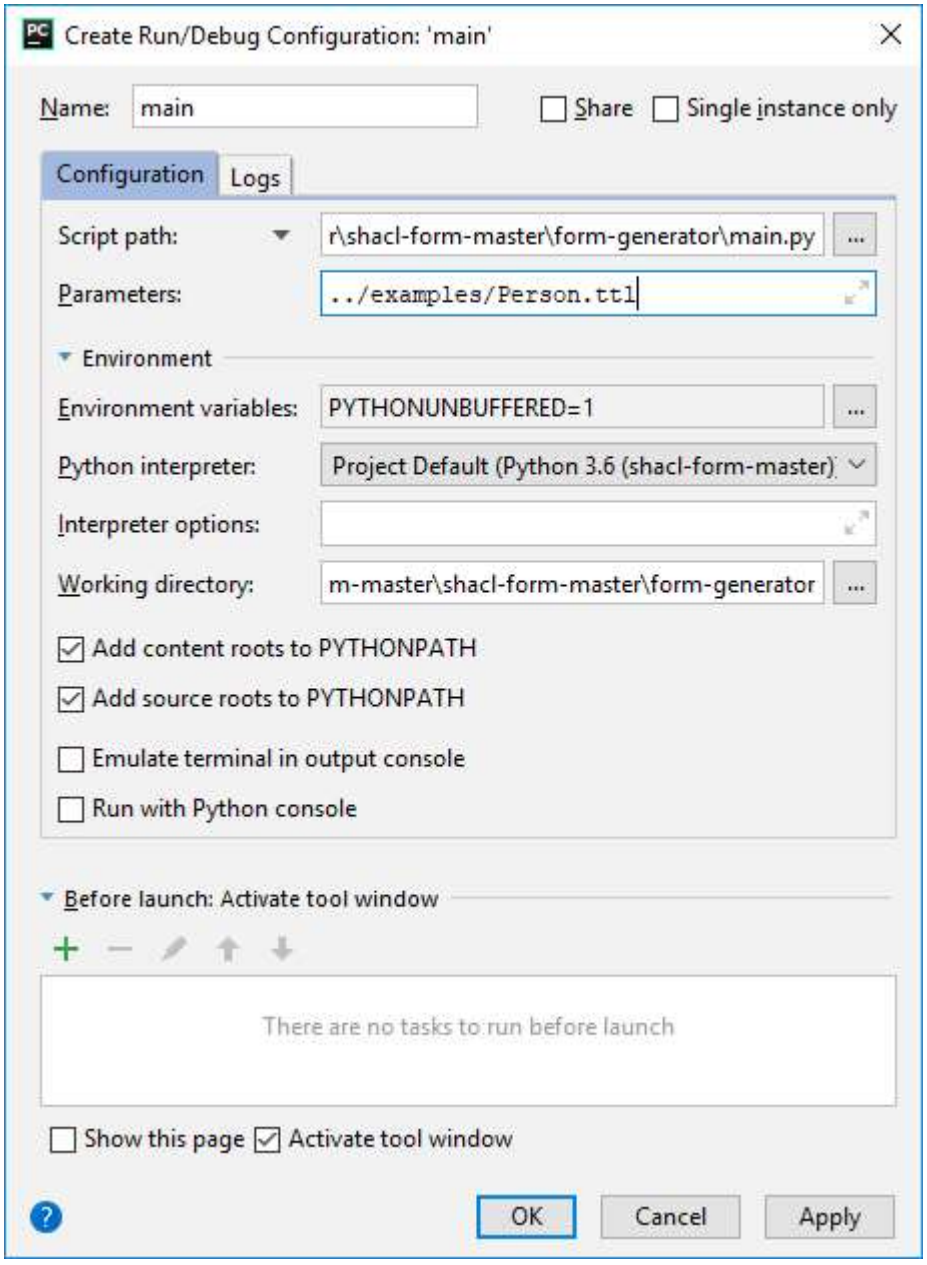


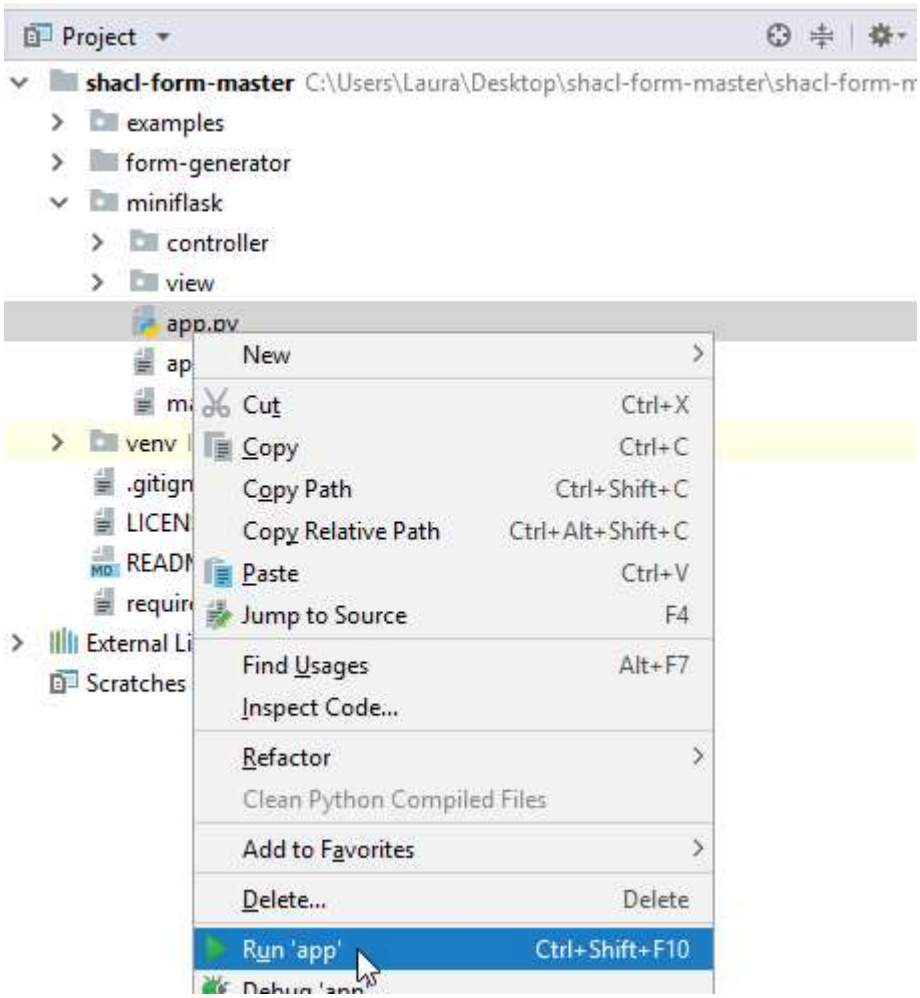
Figure 18. Correct configuration for this example

Now that the parameters have been set, run the webform generator by clicking the green ‘Run’ icon in the upper left, as shown in Figure 19. The webform has been generated and placed in the Flask framework, ready to be viewed.

Figure 19. The Run icon

Viewing the Webform

In the ‘miniflask’ directory, right click ‘app.py’ and select “Run ‘app’.



Open a browser and, type ‘localhost:5000’ in the address bar, and hit Enter. This will open the index page, which has a link leading to the webform. In this example, the webform should look like the webform in Figure 20.

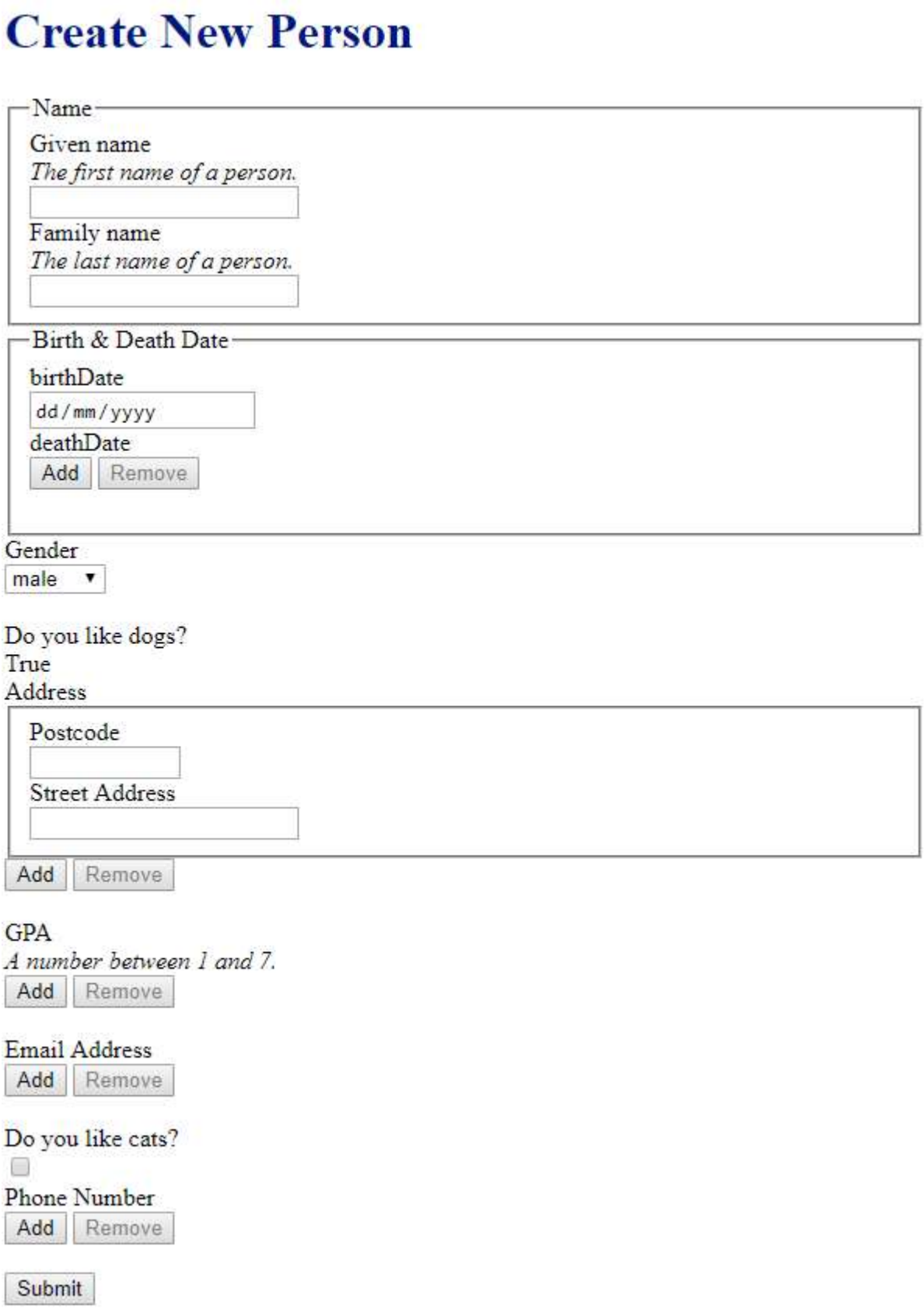


Figure 20. Webform produced by the 'Person.ttl' example SHACL file

Creating RDF data

Type some information into the webform and click ‘Submit’. In PyCharm, there will be a new directory called ‘entries’. Open the ‘.ttl’ file inside this directory. It will contain the data that was entered in RDF format. An example is shown in Figure 21.

e5655d8f-3f18-4294-8c8c-07c0d3fba00d.ttl

```
@prefix : <http://example.org/ex#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

:e5655d8f-3f18-4294-8c8c-07c0d3fba00d a schema:Person ;
    :likesDogs "on" ;
    schema:address [ schema:postalCode "10000" ;
            schema:streetAddress "34 Grey Street" ] ;
    schema:birthDate "2018-05-10" ;
    schema:familyName "Power" ;
    schema:gender "male" ;
    schema:givenName "Max" .
```

Figure 21. Example RDF data from data submitted though the webform