**Configure the Comparative Pathology Workbench (CPW) in a Development Environment**

This document describes how to set up and run the Software for the Comparative Pathology Workbench (CPW) in a DEVELOPMENT environment ONLY!

See the Document “CPW\_Configure-Production.docx” for instructions on how to setup the CPW in a simulated PRODUCTION environment.

Mike Wicks

19th May 2023

**Step 1 - Prerequisites**

Install Postgres and create a database with a database user.

Pull the software from the Git repository: <https://github.com/Comparative-Pathology/comparativepathologyworkbench>

Install “Pipenv” or other software that runs a virtual environment. Pipenv requires a “**Pipfile**” that specifies all the required Library dependencies. A “**requirements.txt**” file is also provided, should you use different virtual environment software.

**Step 2 – Set Up a Virtual Environment**

Move to the “**app**” folder - Create a virtual environment:

**pipenv install**

Start the a virtual environment:

**pipenv shell**

**Step 2 – Configure Software with a Database**

The system needs a file to hold environment variables.

In the “**app/config**” sub-folder create a “**.env**” file with the following environment variables.

An example “**.env**” file is shown here:

#

# CPW Environment Settings

#

SECRET\_KEY=a\_secret\_key

ENCRYPT\_KEY=an\_encrypt\_key

CPW\_CIPHER\_STRING=a\_cpw\_cipher\_string

#

DEBUG=True

ALLOWED\_HOSTS=localhost, 127.0.0.1

#

# PostGres Database Settings

#

DB\_ENGINE=django.db.backends.postgresql\_psycopg2

DB\_NAME=a\_schema\_name

DB\_USER=a\_database\_user

DB\_PASSWORD=a\_password

DB\_HOST=localhost

DB\_PORT=5432

DB\_ATOMIC\_REQUESTS=True

#

# Dummy Email File Backend

#

EMAIL\_BACKEND=django.core.mail.backends.filebased.EmailBackend

EMAIL\_FILE\_PATH=a\_folder\_somewhere

#

SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE=True

SESSION\_COOKIE\_AGE=86400

# Default: 1209600 (2 weeks, in seconds)

**Step 3 - Set up an Empty Database**

Run all Django Migrations, to set up an empty database.

**python manage.py migrate**

Create Superuser Account

**python manage.py createsuperuser**

**Step 4 - Set up Base Data in the Database**

The following commands populate the database with the basic data required for the system to function.

**python manage.py loaddata fixtures/protocol\_prod.json --app matrices.protocol**

**python manage.py loaddata fixtures/type\_prod.json --app matrices.type**

**python manage.py loaddata fixtures/location\_prod.json --app matrices.location**

**python manage.py loaddata fixtures/command\_prod.json --app matrices.command**

**python manage.py loaddata fixtures/blog\_prod.json --app matrices.blog**

**python manage.py loaddata fixtures/environment\_prod.json --app matrices.environment**

**python manage.py loaddata fixtures/authority\_prod.json --app matrices.authority**

**python manage.py loaddata fixtures/collectionauthority\_prod.json --app matrices.collectionauthority**

**python manage.py loaddata fixtures/server\_prod.json --app matrices.server**

**Step 5 – Run the System**

**python manage.py runserver**

Point a Browser at <http://localhost:8000>

You should now see the CPW Home Page

**WordPress Blogging Engine**

The Workbench requires access to a WordPress blogging instance to provide the full functionality of the system.

See the document “CPW\_Configure-Production.docx”, Part B, for instructions on how to setup a suitable WordPress “stack”.