CPL Automation — Beginner-Friendly Setup & User Guide

This guide is written for people with **no technical background**.

If you follow each step in order, you can install and run the app on a new computer.

What this app is for

This app helps compare:

- units from a student's external transcript (for example, Victoria University), and
- SHEA units,

then suggests likely credit matches with a confidence score.

Before you start (what you need)

- A computer with internet
- Python installed (version 3.11 or higher)
- This project folder (`cpl-automation`) copied to your computer
- Your SHEA file named exactly:

- `SHEA Course Data.xlsx`

Place that Excel file inside:

- `cpl-automation/data/`

Part A — First-time setup (one time only)

Step 1: Open Terminal

- **Mac:** open Spotlight → type `Terminal`
- **Windows:** open `PowerShell`

Step 2: Go to project folder

Example:

```
cd /path/to/cpl-automation
```

Step 3: Create app environment

Mac / Linux

```
python3 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
```

```
python -m playwright install
```

Windows PowerShell

```
py -3 -m venv .venv
.venv\Scripts\Activate.ps1
python -m pip install --upgrade pip
pip install -r requirements.txt
python -m playwright install
```

If all commands finish without errors, setup is complete.

Part B — Start the app each time you use it

Step 1: Start Streamlit app

In terminal (inside `cpl-automation`):

Mac / Linux

```
source .venv/bin/activate
streamlit run app.py --server.port 8503
```

Windows

```
.venv\Scripts\Activate.ps1
streamlit run app.py --server.port 8503
```

Step 2: Open app in browser

Open:

- `http://localhost:8503`

Part C — Run MCP server (required for external website retrieval)

You need this if you want the app to browse university websites and collect unit details.

Step 1: Open a second terminal window

Step 2: Go to MCP folder

```
cd /path/to/cplmcp
```

Step 3: Create MCP environment and start server

Mac / Linux

```
python3 -m venv .venv
source .venv/bin/activate
pip install --upgrade pip
pip install -r requirements.txt
python3 server.py
```

Windows PowerShell

```
py -3 -m venv .venv
.venv\Scripts\Activate.ps1
pip install --upgrade pip
pip install -r requirements.txt
```

```
python server.py
```

Keep this second terminal open while using the app.

Part D — How to use the app (normal workflow)

1) Load SHEA data

- In app sidebar, click:

- **Load SHEA units from local Excel**

This loads `data/SHEA Course Data.xlsx` into the app database.

2) Upload transcript

Go to page: **Upload Transcript**

- Upload transcript PDF
- Check extracted text
- Click **Parse and save external units**

3) Enrich external units from university website

Go to page: **CPL Suggestions**

- Choose university from dropdown OR paste course URL
- Click:

- **Run MCP check: crawl external course website**

This fills external unit descriptions/outcomes from the website.

4) Generate suggestions

- Click **Generate suggestions**

You will see:

- suggested SHEA match
- confidence percent
- calculation components

5) Review decisions

Go to page: **Review & Approval**

For each suggestion choose:

- approved

- rejected
- needs_review
- override

6) Export report

In suggestions page, click export:

- CSV
- Excel
- PDF

Export files are saved in:

- `cpl-automation/exports/`

Confidence score explained (simple)

Confidence is based on:

- title similarity
- description similarity
- learning outcome similarity
- credit similarity
- grade bonus
- retrieval bonus

You will also see these component percentages in output.

Important rule
If grade is **Fail / Not Competent / NYC**, suggestion is flagged and should **not** be auto-approved.

“I’m stuck” quick fixes

App does not open

- Make sure terminal command is still running
- Check URL is `http://localhost:8503`
- Try different port: `--server.port 8504`

MCP retrieval returns empty

- Confirm MCP terminal is running `python3 server.py`
- Confirm internet is working
- Re-run MCP check in app

University dropdown is empty

- Check file exists:

- `cpl-automation/data/university_registry.json`

SHEA not loading

- Check file name exactly:

- `SHEA Course Data.xlsx`

- Check file location:

- `cpl-automation/data/`

Database issue

Run this once:

```
python -c "from src.db import init_db; init_db()"
```

Recommended daily usage (very short)

- Start app terminal
- Start MCP terminal
- Open browser
- Load SHEA data
- Upload transcript
- Run MCP check
- Generate suggestions
- Review and export

MVC and data model (simple explanation)

MVC in this project

- **Model** = data and logic (`src/` files + SQLite tables)
- **View** = what user sees (`app.py` Streamlit pages)
- **Controller** = button actions in `app.py` that trigger model operations

Tables used by the app

- **`shea_units`**

- Stores official SHEA unit data

- Example info: unit code, name, description, learning outcomes, course level

- **`external_units`**

- Stores parsed transcript units and externally retrieved unit details

- Example info: institution, unit code, grade, semester, source URL, overview/outcomes

- **`suggestions`**

- Stores matching results between external and SHEA units

- Example info: confidence score, confidence %, explanation, score components

- **`decisions`**

- Stores reviewer decisions on each suggestion

- Example info: approved/rejected/needs_review/override, reviewer notes

- **`external_unit_url_cache`**

- Stores previously discovered unit URLs to speed up retrieval

Data flow (how records move)

- Load SHEA Excel → inserts into `shea_units`
- Upload transcript + enrichment → inserts/updates `external_units`
- Generate suggestions → inserts into `suggestions`
- Review page actions → inserts into `decisions`

Folder map (what is where)

- `app.py` → main app
- `data/` → input files + database
- `exports/` → output reports
- `src/` → internal app logic
- `docs/` → documentation

Final note

If a non-technical staff member follows this guide exactly, they should be able to run the app end-to-end.