

Project Context

This project is entirely **software-based**, using **DigSILENT PowerFactory** and **Python** to explore how different levels of solar integration affect voltage stability in an isolated small town.

I have the base PowerFactory model set up. What I need help with is building out the **Python automation and user interface**, making it smooth, reliable, and robust.

The final tool will allow us (and external users) to:

- Adjust solar penetration across suburbs.
- Run Quasi Dynamic Simulations automatically through the UI.
- Extract important results.
- Display these results in different formats, graphs, figures etc.

Collaboration Goals

I'd like us to work together on:

1. PowerFactory Integration

- Connect Python to Power Factory's API (mostly done).
- Automate the Quasi-Dynamic Simulation over a 24-hour period (possibly done, needs to be confirmed).
- Allow solar capacity to be set for each suburb (Seems to work well, this is done by adjusting the number of parallel inverters from 0 to 100, each inverter adds 6kW of installed PV. There is a 2 commercial and 1 industrial suburb that have make 100kW of PV and one school which has make 25kW, I will provide a spread sheet with all of the info).
- Extract min/max voltage levels for each suburb bus over sim period (p.u).
- Extract PV production over the sim period (kW/kWh).
- Extract line loading over the whole model (%).
- Extract Load profile of the sim period (kW/kWh).
- Extract Transformer loading over sim period (%).

2. Python UI Development

- A simple interface where solar levels (0–100%) can be adjusted with sliders & or increment arrows and display that percentage and kW of installed PV (mostly done).
- Needs a run button that will run the Quasi Dynamic Simulation in Power Factory (possibly done, although not 100% certain).
- Results need to be extracted and displayed, with the min/max voltages displayed by default (mostly done), and then the other variables displayed as each is selected by the user on a graph to compare results, just one suburb at a time. So if say just PV production is ticked, then only that will be displayed on the graph (not done).
- The map of the area should be displayed on the UI (mostly done, but poorly, needs work).
- Any results that are outside of the norm need to be flagged in some way
 - Min/max p.u – 0.95 min, 1.05 max
 - Line loading – anything over 95%
 - Tx Loading – anything over 95%
- (Push) be able to extract the results to .csv (would be good to have if we are already extracting all the results from power factory).

3. Performance Testing

- Run simulations for different solar penetrations.
- Display all required results in a clear and accurate way.

Technical Environment

- **PowerFactory**
- **Python 3.9** (anything higher and it upsets power factory)
- The main requirement is that the UI be easy to use, clear, and robust.

Deliverables

- A **Python codebase** (modular, well-documented).
- Working **PowerFactory automation** (script tested on my model).
- UI that can adjust solar penetration, run sims, and display/export results.

- Documentation & comments so I can work with the code after the project.
-

Timeline

- Project will need to be completed by 29.08.2025
-

Collaboration Style

I'm looking to work **iteratively** — e.g. start with a working script that connects to PowerFactory, then build up the UI, then add data processing and results export. I'll test things on my end as we go.

I'm happy to:

- Share the base PowerFactory model and existing Python scripts.
- Give feedback after each step.
- Work with you to make sure the solution is practical and well-aligned with my academic requirements.