**⚙️ Phase 1: Core Technical & Python Foundations for Power Systems Applications**

**Duration:** 8 Weeks (can extend to 10 weeks if you need flexibility)

**Objective:** Build Python proficiency from zero to applied-level for electrical/power systems work.

**Outcome:** By end of Phase 1, you’ll be able to code analytical scripts for power flow, data visualization, and automation tasks related to your engineering work.

**🗓️ Week 1 – Environment Setup + Python Basics**

**Goal:** Get comfortable with the tools and basic syntax.

**Focus:** Smooth start — environment, basic programming, and working efficiently.

**✅ Tasks:**

* **Set up Tools**
  + Install Anaconda (for Python, Jupyter, and libraries).
  + Set up VS Code and learn to run Python scripts.
  + Learn how to use GitHub for version control (create your first repo for your learning journey).
* **Python Fundamentals (Day 2–6):**
  + Data types, variables, lists, tuples, dictionaries.
  + Loops and conditionals.
  + Functions — defining and calling them.
  + Basic input/output.
  + Simple exercises — temperature converter, unit converter (kV to V, MW to kW, etc.).
* **End-of-week mini project:**

→ Write a simple script that calculates transmission line parameters (R, X, Z) from conductor data.

**🗓️ Week 2 – NumPy, Pandas, and Data Handling**

**Goal:** Learn how to handle numerical and tabular data (core for all later analysis).

**✅ Tasks:**

* **NumPy:**
  + Arrays, indexing, slicing, reshaping.
  + Vectorized operations.
  + Solving linear equations with numpy.linalg.
* **Pandas:**
  + DataFrames, importing/exporting CSV/Excel.
  + Filtering, sorting, grouping data.
  + Handling missing data.
* **Mini project:**

→ Analyze transformer test data in Excel (e.g., open circuit/short circuit test) using Pandas to compute efficiency and regulation.

**🗓️ Week 3 – Data Visualization & Engineering Plotting**

**Goal:** Learn to visualize results like load flow outputs or time-series data.

**✅ Tasks:**

* **Matplotlib + Seaborn:**
  + Plot line, bar, and scatter charts.
  + Add labels, legends, subplots.
  + Customize plots for professional reports.
* **Plotly (optional but recommended):**
  + Create interactive charts for power system visualizations.
* **Mini project:**

→ Visualize daily load curve (CSV data) using Pandas and Matplotlib. Add annotations for peak/off-peak hours.

**🗓️ Week 4 – Power System Engineering Refresher + Python Integration**

**Goal:** Revisit essential power system formulas and calculations — but automate with Python.

**✅ Topics:**

* Per-unit system and base conversions.
* Line impedance and admittance matrices.
* Bus admittance (Y-bus) formation.
* Complex power, current, voltage relations.

**✅ Practical work:**

* Write Python functions to:
  + Convert actual system data to per-unit.
  + Build Y-bus for a 3-bus or 4-bus system.
* Verify manually with textbook/Matlab-style solution.

**Mini project:**

→ Automate Y-bus formation for any number of buses using user inputs (Python script).

**🗓️ Week 5 – Power Flow (Load Flow) Computations**

**Goal:** Implement core algorithms like Gauss-Seidel and Newton-Raphson in Python.

**✅ Tasks:**

* Understand mathematical foundation:
  + Power mismatch equations.
  + Jacobian matrix structure.
  + Iterative solution flow.
* Code implementation:
  + Write functions for bus classification (PQ, PV, Slack).
  + Code Gauss-Seidel method.
  + Visualize convergence.
* **Mini project:**

→ Perform load flow for 3-bus system and verify against textbook/MATLAB result.

**🗓️ Week 6 – Power System Data Analytics**

**Goal:** Bridge Python data science tools with power system insights.

**✅ Tasks:**

* Time-series analysis of load data (daily/monthly).
* Correlation analysis — e.g., temperature vs load.
* Anomaly detection in power measurements using rolling mean.
* Use Pandas + NumPy to generate synthetic load profiles.
* **Mini project:**

→ Analyze 1-year load data for a substation and identify seasonal patterns.

**🗓️ Week 7 – Automating Engineering Workflows**

**Goal:** Learn automation skills to reduce repetitive engineering tasks.

**✅ Tasks:**

* Automate Excel report generation using openpyxl or xlsxwriter.
* Generate formatted Power System Summary Reports.
* Read equipment data (transformers, lines, breakers) from Excel and compute:
  + Total losses.
  + Loading summary.
* **Mini project:**

→ Build an “Equipment Performance Report Generator” using Excel + Python.

**🗓️ Week 8 – Integration + Portfolio Project**

**Goal:** Combine all you learned into one professional-level project.

**✅ Capstone Project:**

* Build **“Mini Power System Analysis Tool”** using Python:
  + Input: Excel sheet with bus, line, and load data.
  + Output: System losses, voltages, convergence plots.
  + Visualization: Bus voltage profile, load flow convergence.
  + Export: Excel summary report.

**✅ Deliverables:**

* Fully documented code.
* GitHub repository.
* Optional short video demo (for Upwork/LinkedIn portfolio).

**📘 Additional Recommendations**

* Maintain a **learning log** (Google Sheet/Notion) to track progress weekly.
* Each mini-project → publish a short LinkedIn post to start building your **technical credibility**.
* By end of Phase 1, you’ll already have:
  + 5–6 mini projects.
  + 1 major capstone tool.
  + Solid command over Python, data handling, and applied power system automation.

**MacBook Token:**  
github\_pat\_11BZPRULY0KI7lkuUrIHl3\_RdQOultnkmfy8Vgpd1egsaqZG5ThPChN3ymAXL6D8NYZFMFTTWZ3CGarfXa

**MacBook Token: (Classic)**  
ghp\_4JwLJyVEmvEW9nk9o8BTPezF7LBNMn0JIczW

**For Synching Files, Every time:**

git add .

git commit -m "Your message"

git push