User Reference Guide

Distribution System Protection Simulator (DSPS)

Saumil Shah
Swaraj Narasimha Paladi
Mansi Shah

Under guidance of :
Dr. Mesut Baran (NC State University)
Mr. James Stoupis (ABB US Research Centre)

Table of Contents

Summary

The Distribution System Protection Simulator (DSPS) was originally developed as a capstone project for fulfilment of MS-EPSE requirements for core course ECE584: Electric Power Engineering Practicum II at NC State University, Raleigh with association and guidance from ABB US Research Centre. The objective of this project was to develop a standalone GUI program that can model a standard distribution system such as IEEE 123 bus system and which can provide the users with the flexibility to interactively change the penetration levels for DERs such as PVs and show its effect on the protection elements present in the system. The vision for the current initiative is to provide a solid foundation and accelerate such studies in industry and academia. The current framework is made extensible following standard software engineering practices to make future updates to the current code base possible with minimum efforts.

Setup installation

General requirements:

1) Python version requirements: 3.10
2) Recommended python IDE: PyCharm from JetBrains.
3) Recommended web browser: Google Chrome

Setting up python environment:

1) Open the PyCharm IDE once the installation is complete. The main page will look similar as below:

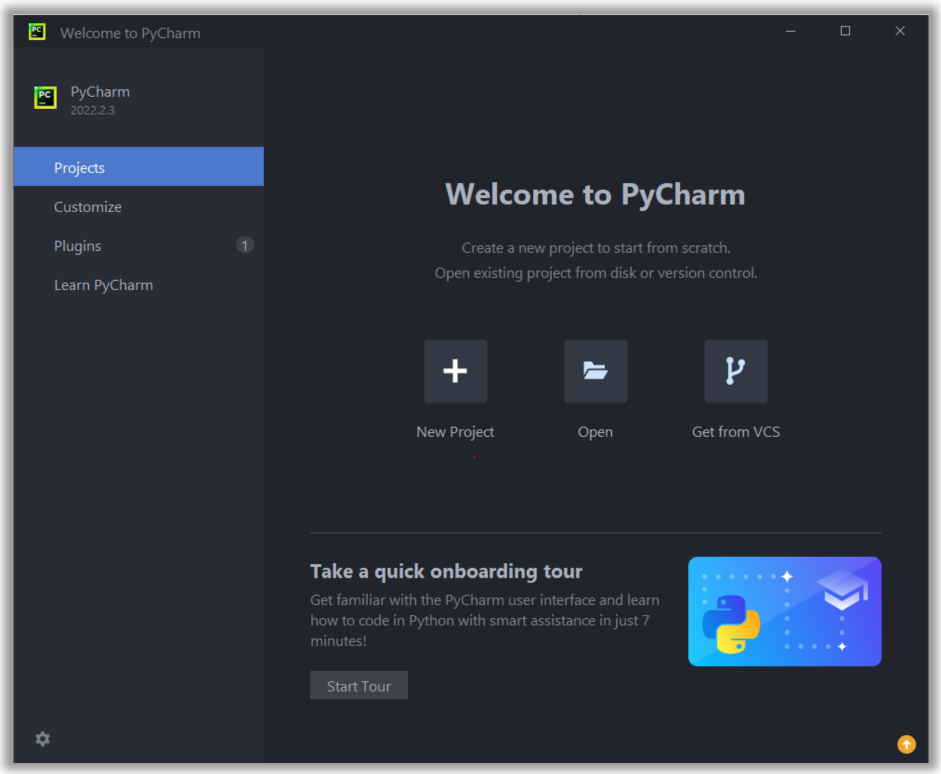

Figure 1: PyCharm welcome page

2) Click and open a new project.

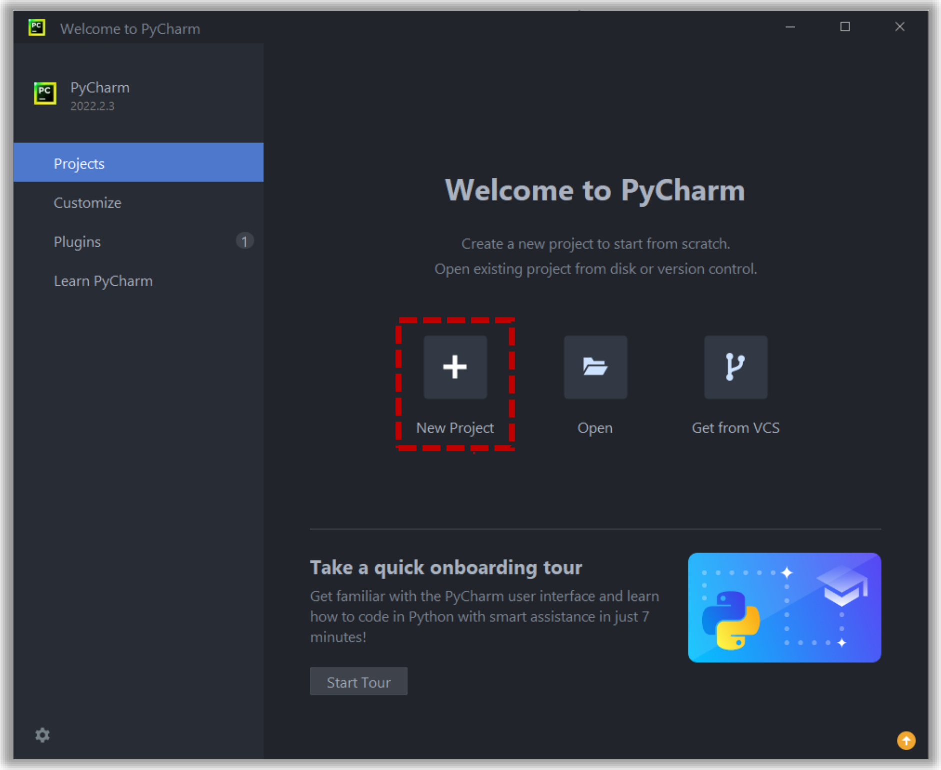

Figure 2: Creating a new PyCharm project

3) Specify path for the project and select Conda environment with python version: 3.10 before creating the project.

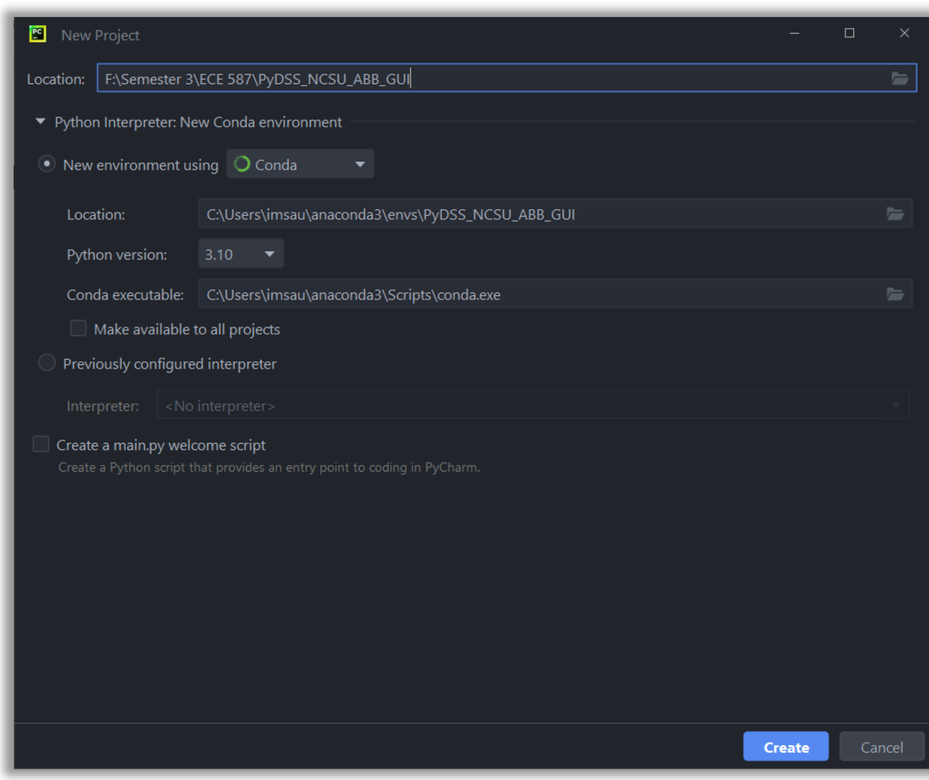

Figure 3: Setting project configuration

4) Open the terminal from PyCharm project window.

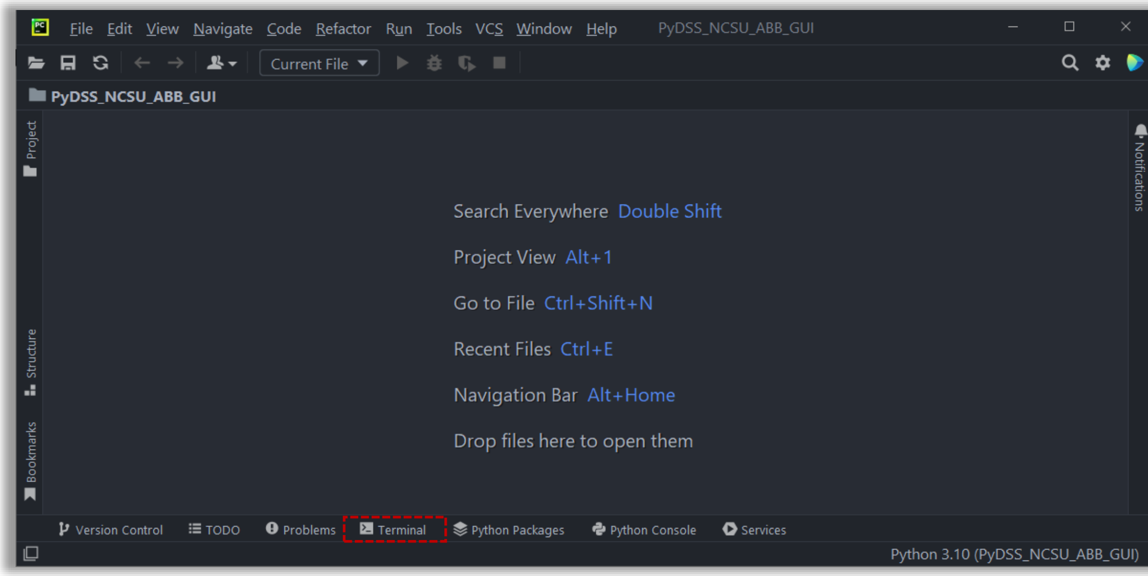

Figure 4: Using Command terminal within Pycharm

5) Verify activated virtual environment on the left and clone the codebase from GitHub.

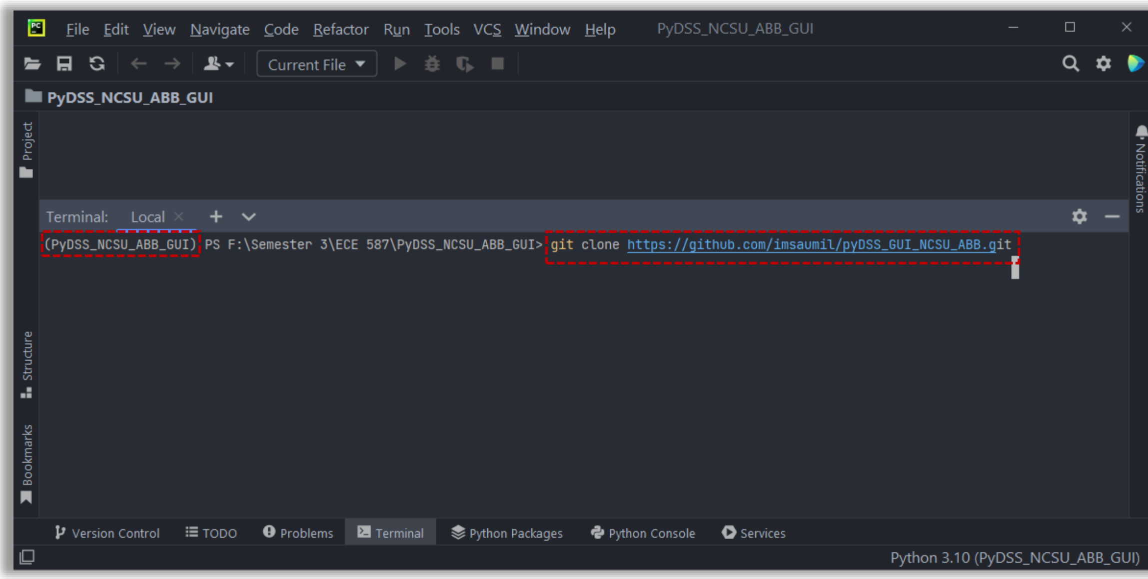

Figure 5: Fetching the required GUI code from GitHub

Enter below mentioned command in the terminal after verifying the activated conda environment:

COMMAND: git clone https://github.com/imsaumil/pyDSS_GUI_NCSU_ABB.git

Note: If it does not work, might require installation of git command line, which is a simple installation and will automatically integrate with the IDE. Please follow the link.

6) Verify the downloaded files as below in cloned subfolder pyDSS_GUI_NCSU_ABB. Also, check for the **requirements.txt** file in the cloned folder.

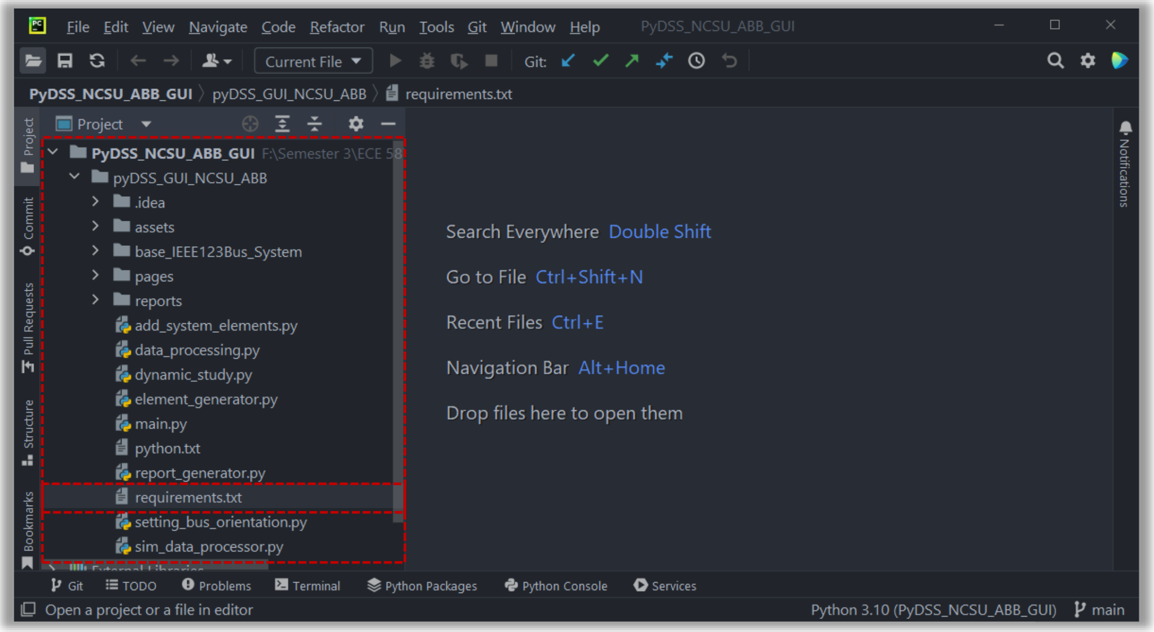

Figure 6: Verifying cloned files in the codebase

7) Click again on the terminal window from the navigation bar at the bottom and use command: **cd .\pyDSS_GUI_NCSU_ABB** to change directory to the cloned project subfolder.

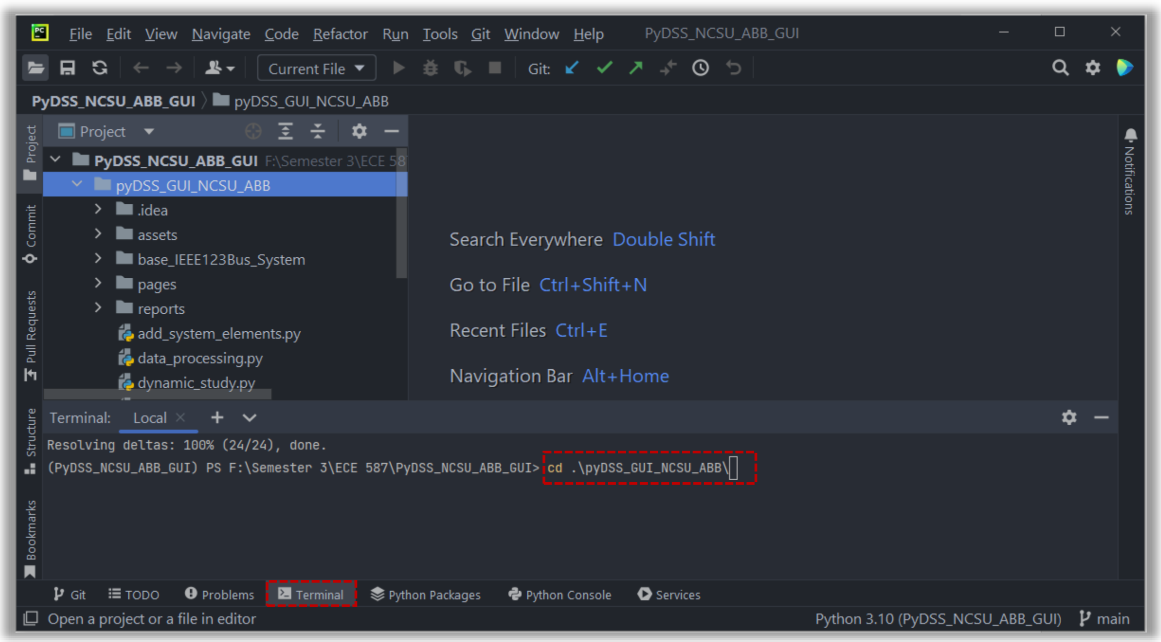

Figure 7: Changing directory to the cloned project subfolder

8) Once in the project subfolder, the next step is to install all the required dependencies for running the developed GUI. Use command: **pip install -r .\requirements.txt**

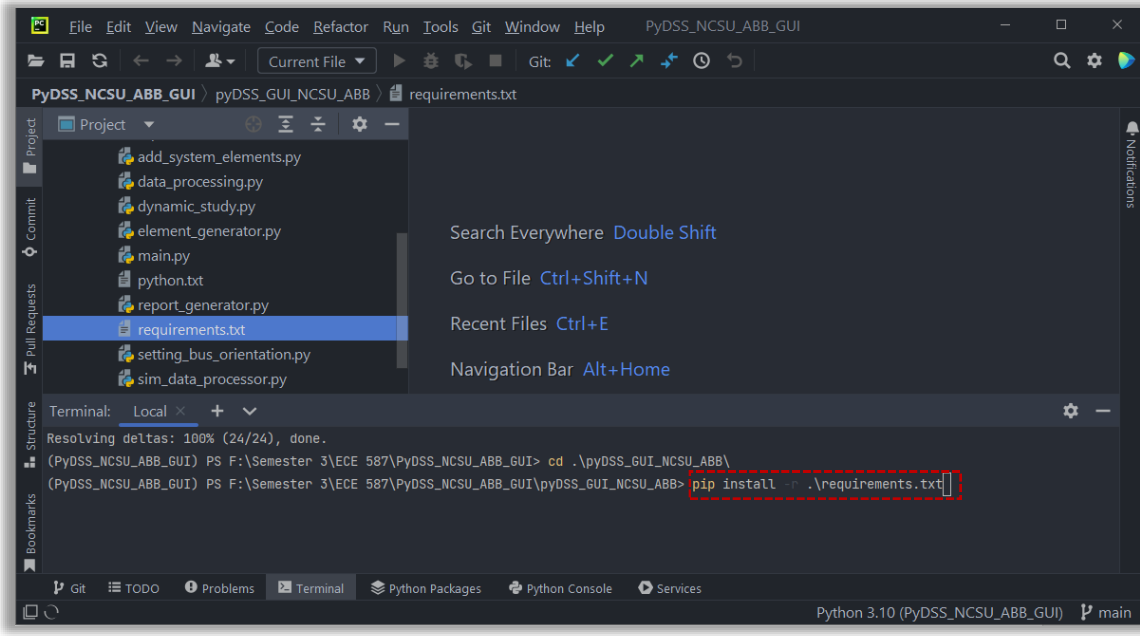

Figure 8: Installing required project dependencies

9) Next is to verify the installation of required libraries in the terminal without any error. Please open an issue on GitHub using the following link if facing any error.

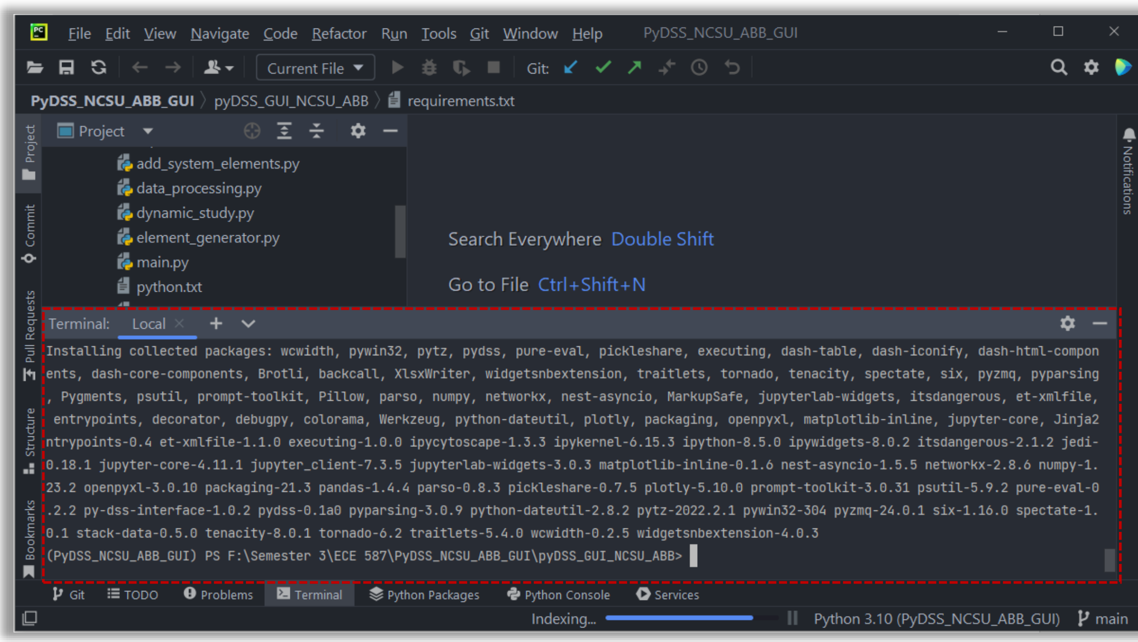

Figure 9: Verifying installed libraries

10) Finally, the environment is all set to run the developed GUI and it can be invoked to run on the local server. Using the following command: **python main.py**

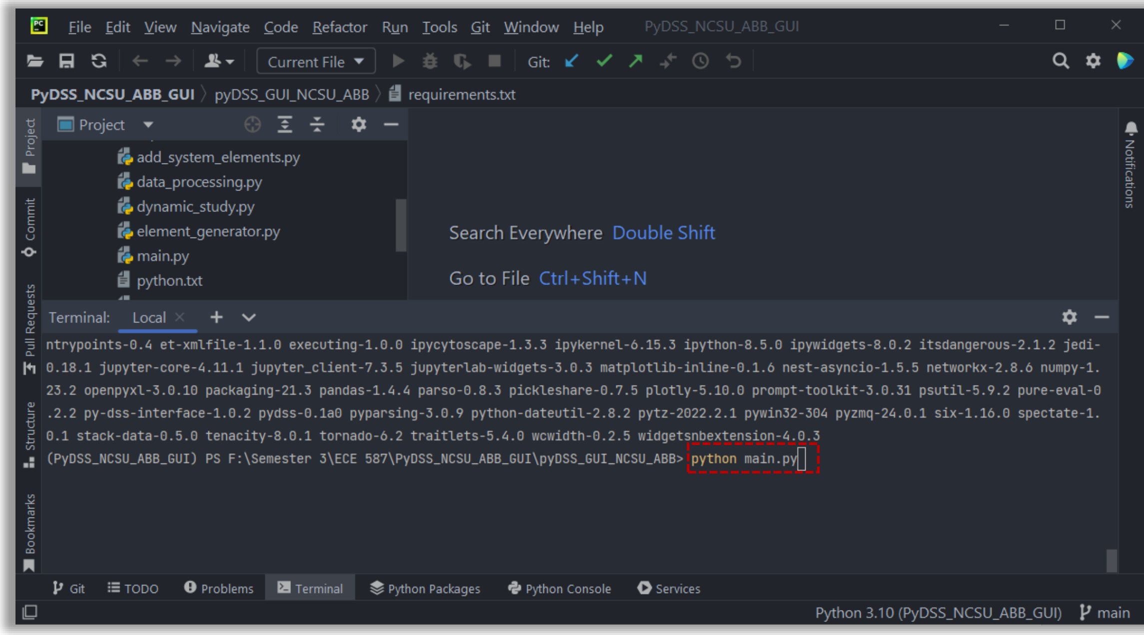

Figure 10: Running main.py file to start the GUI

11) Once the GUI is hosted on the local server, the terminal will pop up the links and user can click on either of the given links to access the GUI on any web browser.

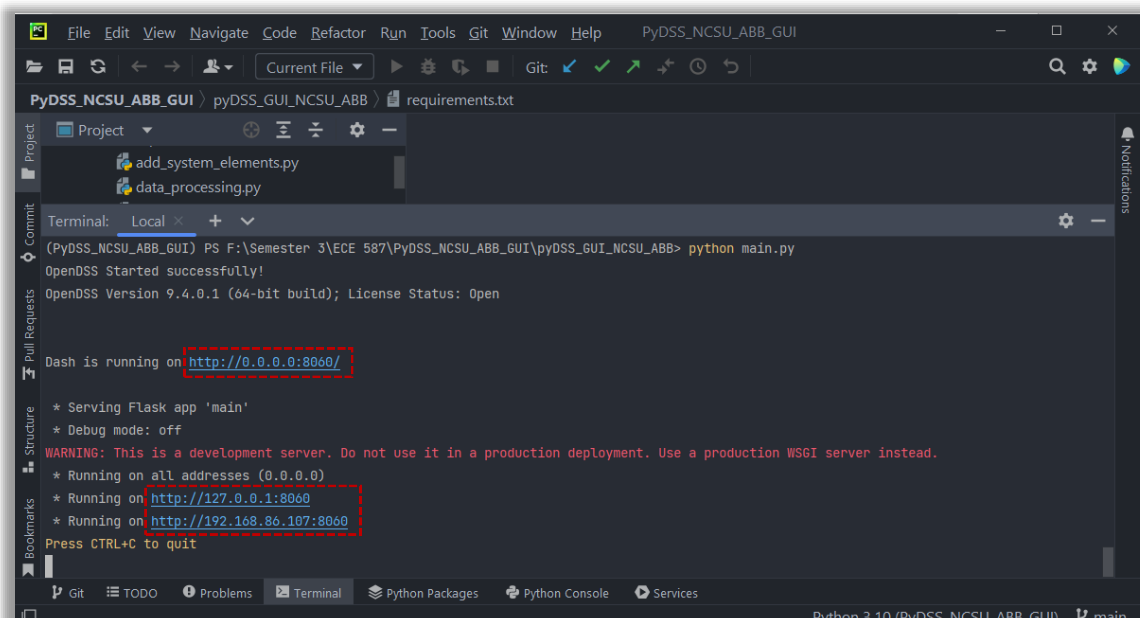

Figure 11: Accessing the GUI hosted on local server

12) Finally, this is how the developed GUI will look on the browser upon starting up.

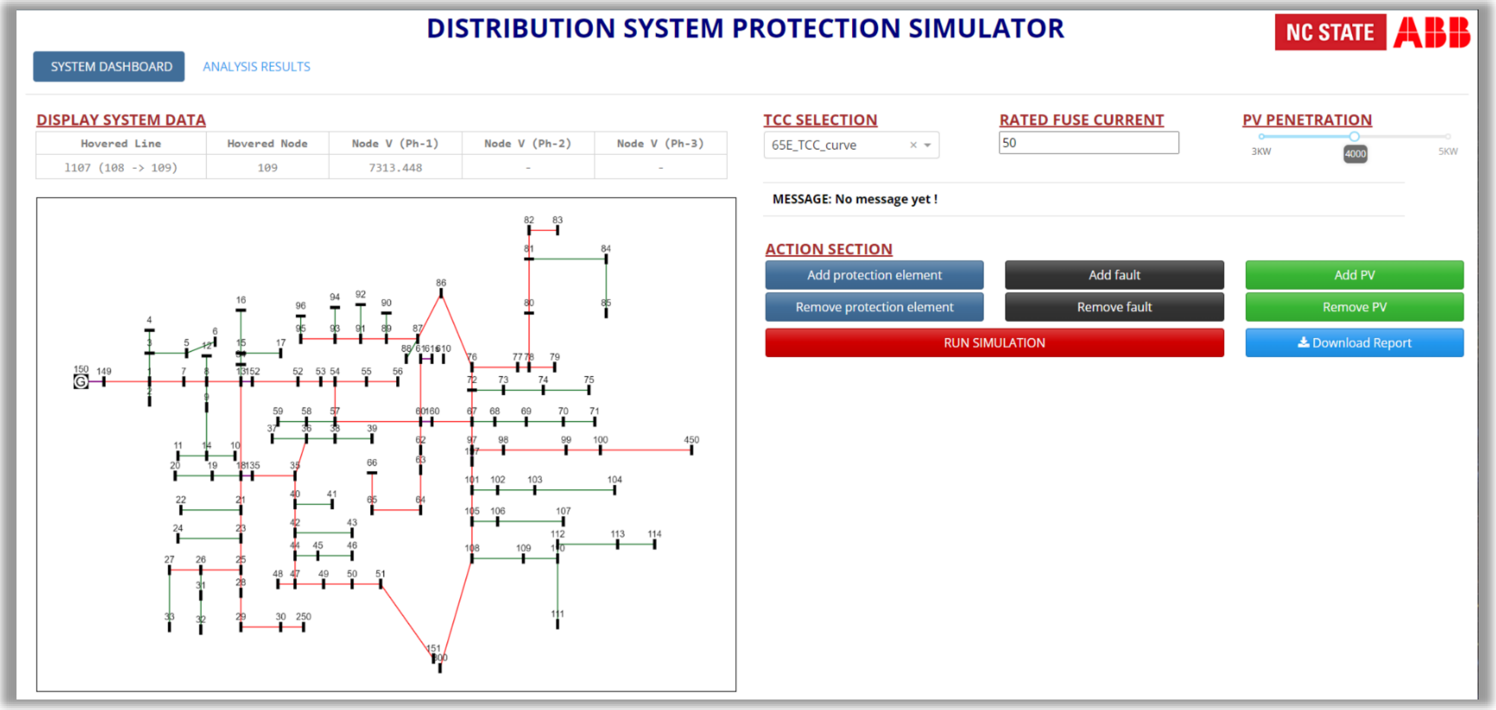

Figure 12: First look of the developed interactive GUI

Sample demonstration

Introduction

The presented sample demonstration will provide a quick walkthrough on how a user can use the standard IEEE 123 bus system embedded in the GUI to interactively hover and get details like node voltages, add protection element such as fuse, change the fuse current rating and TCC curves for individual fuses, add DERs such as PVs, change individual PV penetration levels using a slider and on top add a 3-phase fault to simulate and analyze the effects of varying DER penetration levels on the protection elements in the system. For the present demonstration case, we add two fuses in the system, add two PVs at different locations with default penetration levels and introduce a fault in the system to get the relative contribution of fault current from each individual PV added in the system.

Demonstration Walkthrough:

1) The user can simply start by hovering the mouse over the GUI to observe the dynamically changing data in "Display System Data" tab to check the individual node voltages. It also allows user to zoom in to the interactive nodal map of IEEE 123 bus system on the left and start adding a protection element such as fuse on any line in the system. On top right, the default value of TCC curve for the fuse is set to 65E and the default fuse rated current is set to 50 A. The user needs to select the desired line location and change these values to the desired values prior to inserting a fuse using the "Add protection element" button from the action section. In any case, if user needs to change the position of fuse, change the current rating or TCC curve or remove the fuse altogether, it can be removed by first selecting the fuse to be removed from the GUI and then pressing the "Remove protection element" button from the action section. Here on the GUI, Two fuses are added at location sw2 and sw3 with default TCC curve and the rated fuse current. The message box confirms the successful addition of each fuse with specified rated current and TCC curve.

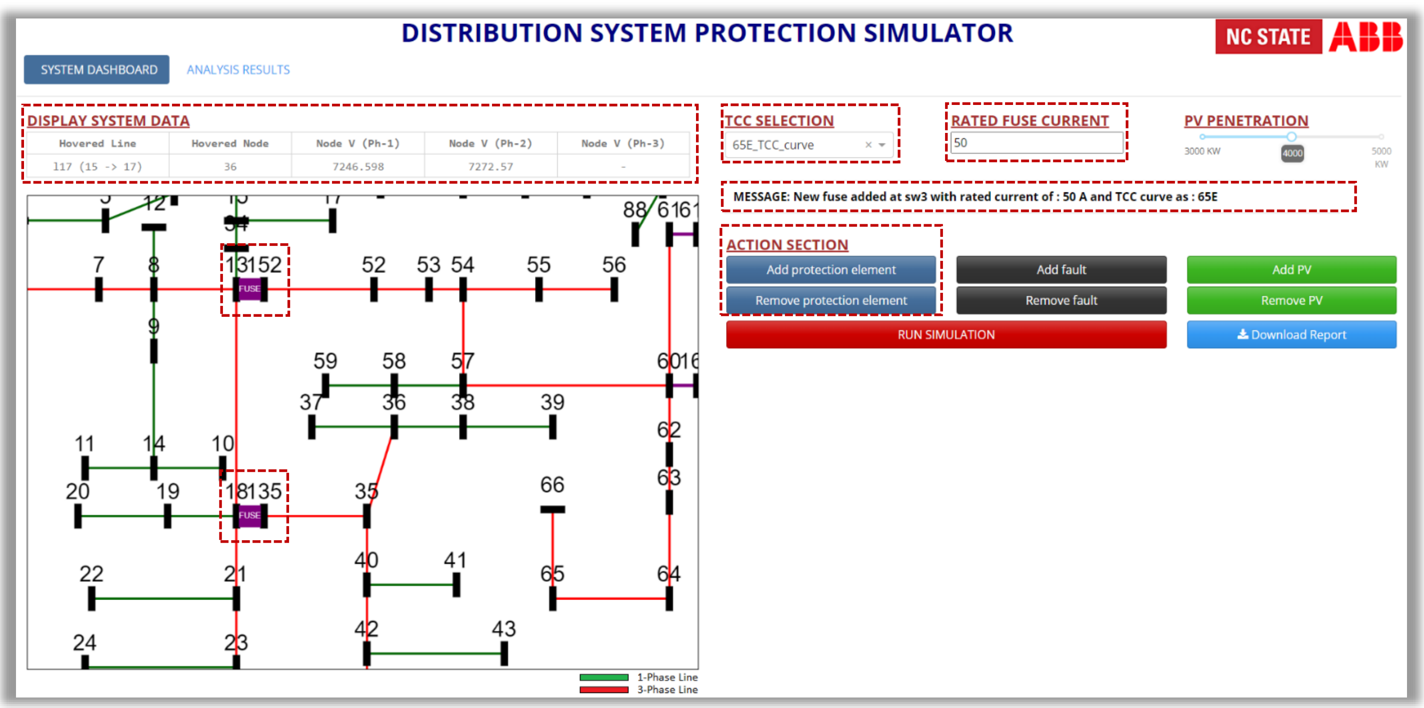

Figure 13: Inserting fuse in the system with specific TCC curve and rated current

2) Next task is to introduce a 3-phase fault in the system and the user can start by just selecting desired bus and clicking the "Add fault" button from the action section. Again, the user can choose to change the fault location or remove it entirely by using the "Remove fault" button from the same section. The message box will confirm the successful addition or removal of fault with the bus location details.

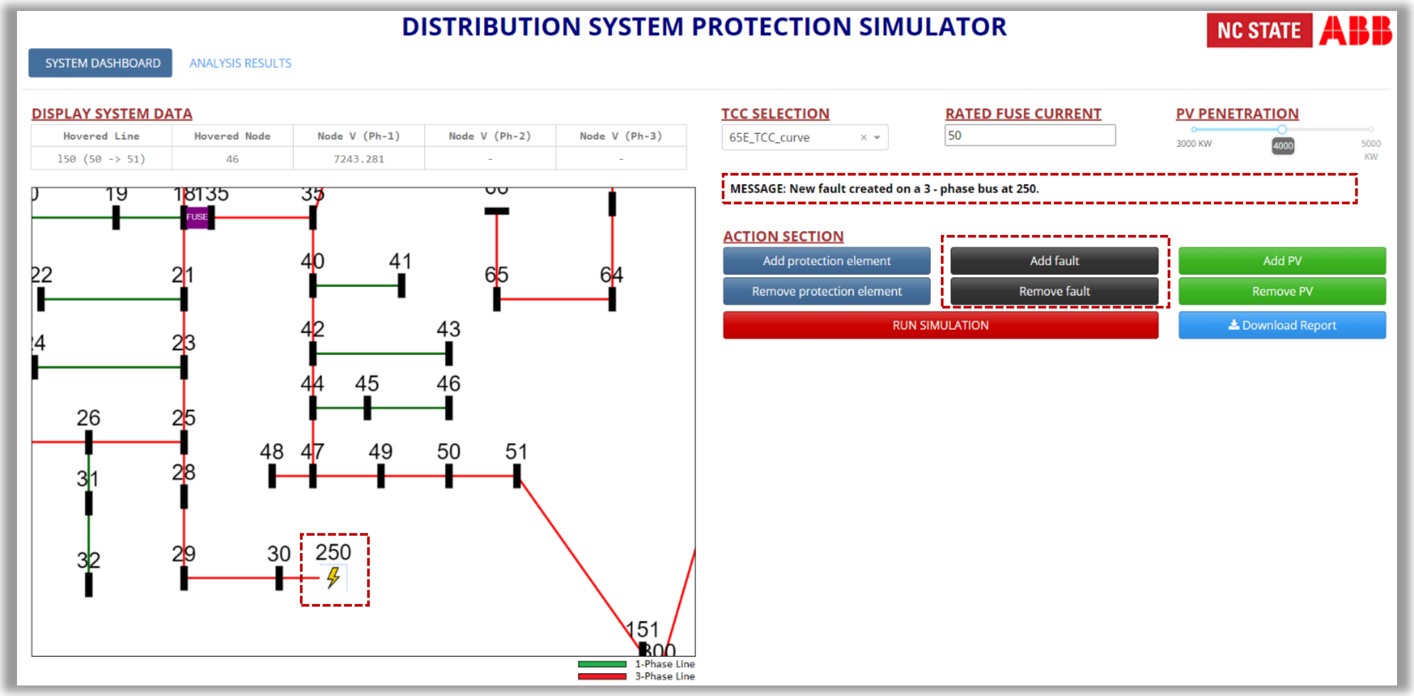

Figure 14: Inserting fault interactively through the GUI

3) Next, the DERs such as PVs can be added by the user interactively on the GUI by selecting the desired bus for the integration and changing the slider to desired PV penetration and pressing the "Add PV" button from action section. Again, the user has the flexibility to change the position of PV at any given time or can choose to remove it from the system by simply selecting the PV to be removed/changed and press "Remove PV" button. The message box will confirm the addition or removal of PV with location and penetration level details.

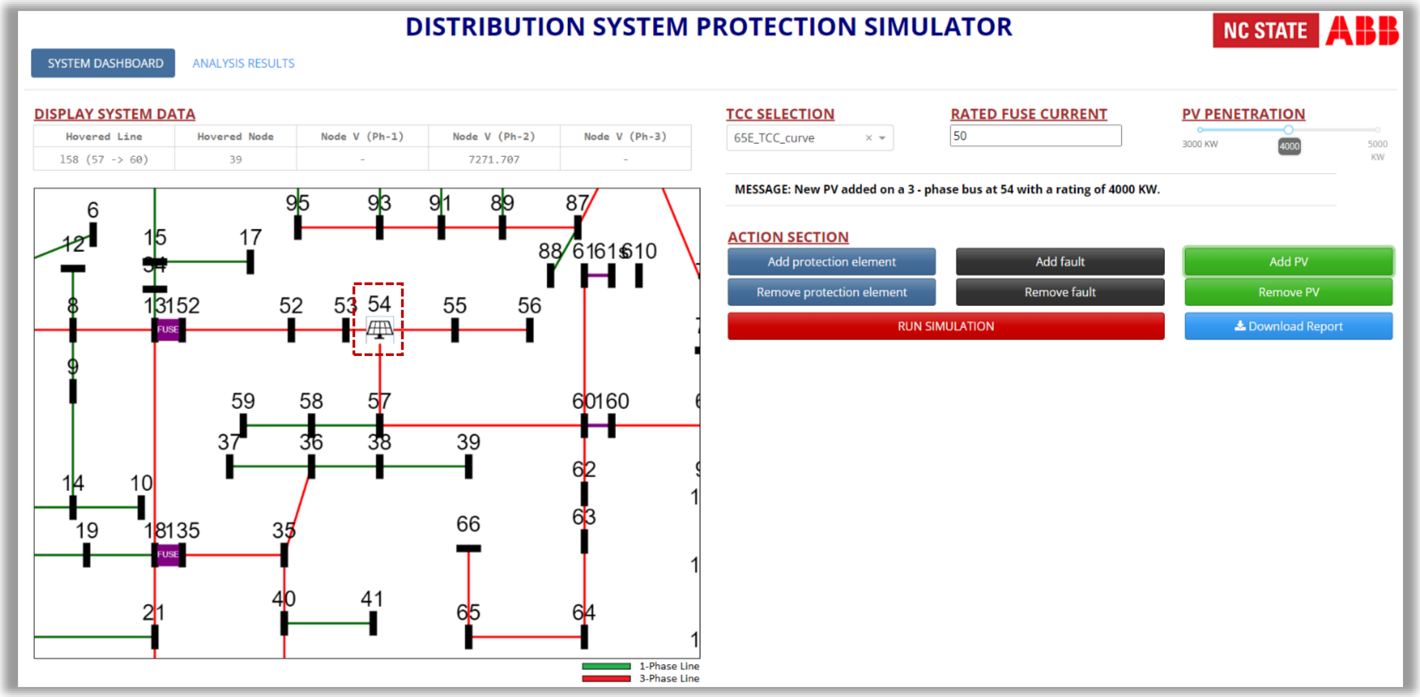

Figure 15: Adding a PV at bus 54 with specific PV penetration levels.

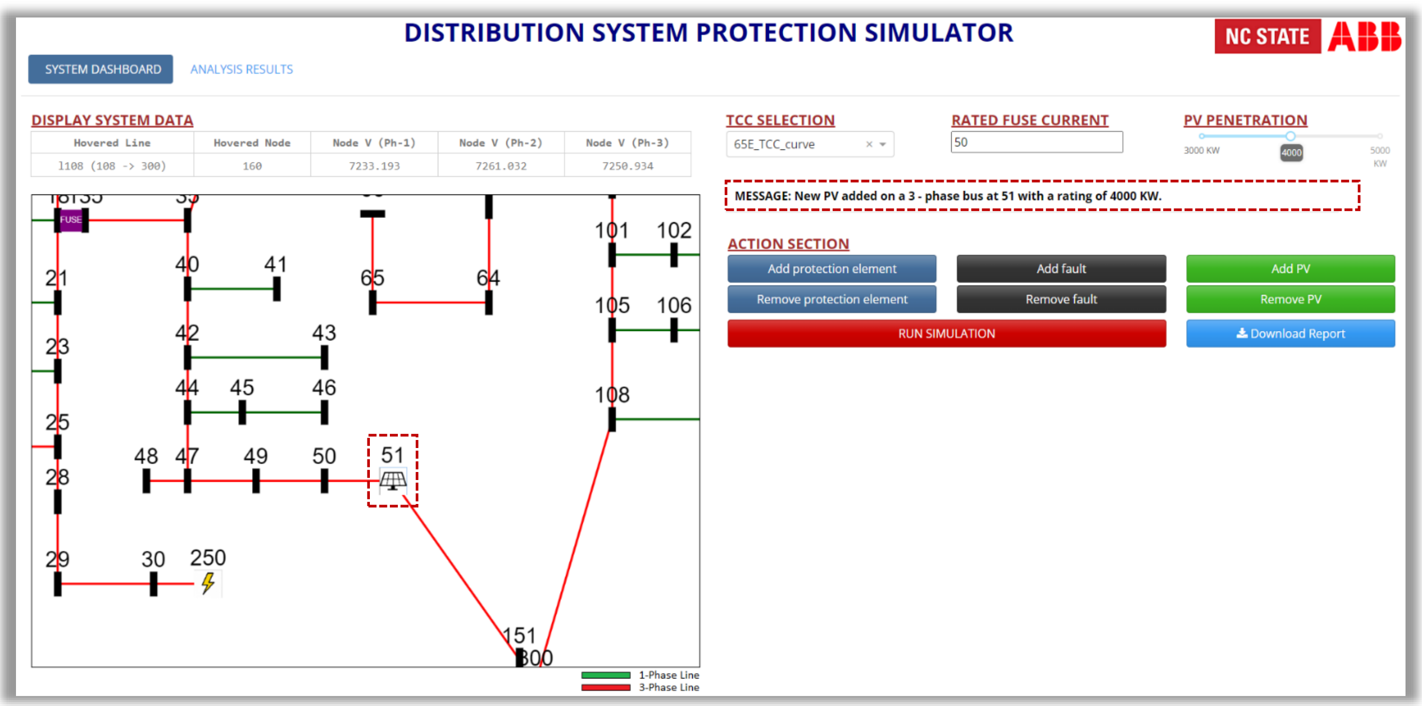

Figure 16: Adding a PV at bus 51 with specific PV penetration levels

4) Finally, user can simply just press the "RUN SIMULATION" button to invoke the developed GUI simulation engine and get the results in terms of intermediate results from each case run and the relative fault current contributions in percentages from each individual PV. The user can also choose to save these results in a formatted xlsx sheet using "Download Report" button.

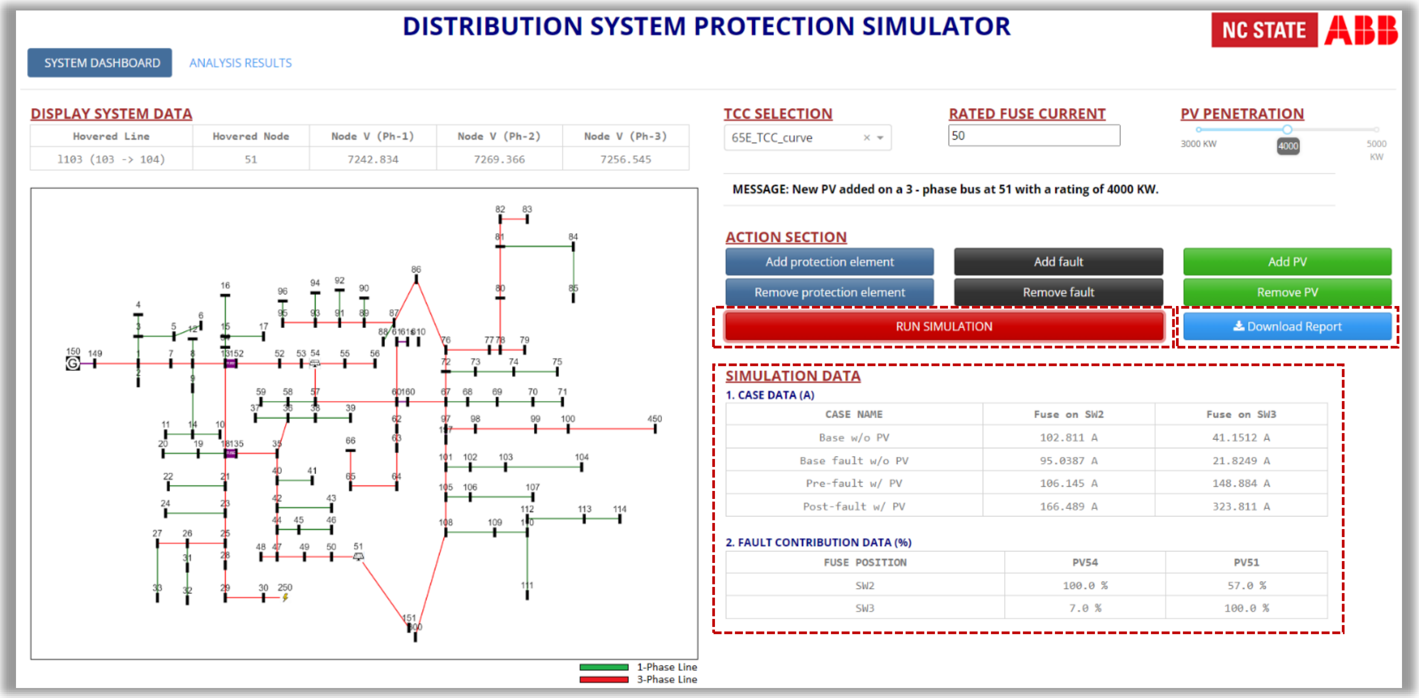

Figure 17: Running the simulation with added components and fault.

5) The downloaded xlsx report file looks as follows:

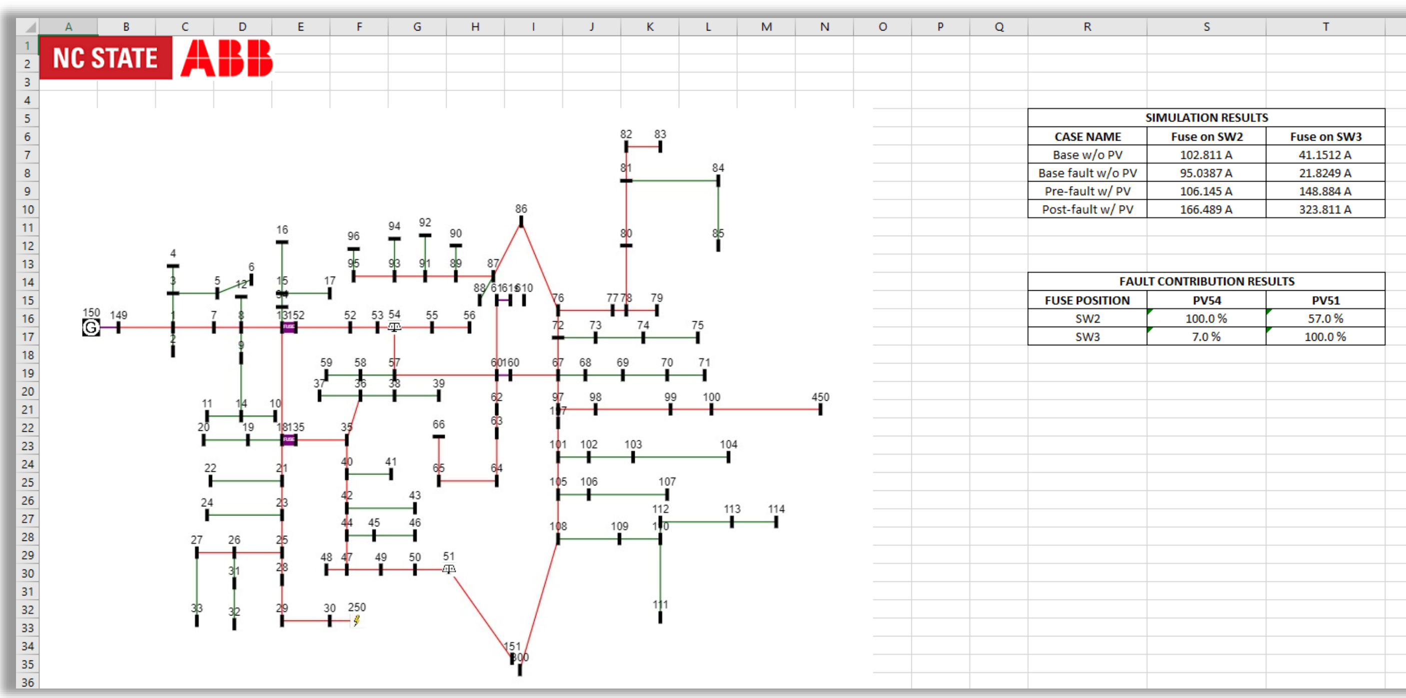

Figure 18: Overview of the GUI generated report