**Automation software:**

**Prerequisites**

Ui path, Python, ChatGPT, Tkinter

Python – Language used

Tkinter – GUI for python

* From making the table to all the user interfaces

GPT3.5 turbo- SDK- programmable language specific (Used Python SDK)

UI path (open source automation tool)- For doing the automation of clicking like tasks(like if we need to upload the issue in zoho it will do automatically).

**Process we are going to cater it:**

* **We are going to make the process which replace the human thinking.**
* **Working with the issue (in ZOHO)**

**To do**

1. **Test cases (where can I create a case to automate)**
2. **Create a process of that will make the ppt of the whole day data (eg; everyday’s Battery status- how much is the distance travelled, time etc).**

**Working with the issue:**

**Five minutes before and after the occurrence of issue(example working with the issue in over current fault)- Sanjith made the ChatGPT write summary for what is the issue**

**CodeForAutomation.py**

1. **main\_folder\_path:**
   * **This variable specifies the path to the main folder that contains subfolders. Each subfolder is assumed to contain the data files (log and km files) that need to be analyzed.**
   * **The code iterates over the subfolders in main\_folder\_path using os.listdir(main\_folder\_path) to find the log and km files for analysis.**
2. **folder\_path:**
   * **Inside the loop that iterates over the subfolders (for subfolder in os.listdir(main\_folder\_path):), folder\_path is used to specify the path to the current subfolder being processed.**
   * **For each subfolder, the code looks for log and km files within that subfolder using os.listdir(subfolder\_path). This is where folder\_path is used to dynamically access the files in each subfolder.**

**Overall, the use of main\_folder\_path allows the code to specify the main directory where all the data is stored, while folder\_path is used to navigate to each subfolder containing the data files for analysis. This modular approach allows the code to handle multiple sets of data files in different subfolders within the main folder.**

**OS module** in Python provides functions for interacting with the operating system.

* Handling the Current Working Directory
* Creating a Directory
* Listing out Files and Directories with Python
* Deleting Directory or Files using Python

**Issue (Ticket Raising) automation:**

**def adjust\_current(row) function:**

This function is designed to set the current to zero if the RPM (revolutions per minute) remains zero for 10 or more consecutive points in the data.

**Os.listdir(folder path)**

This method returns the list of all files and directories in the specified path. The return type of this method is list.

**Currently Working:**

1. **Controller Over Temperature - ANNMON**
2. **CellUnderVoltageWarning- KAMAL**

**Issues rectified till now:**

1. **Controller\_Undervoltage\_408094978**
2. **CellUnderVolWarn\_9**
3. **ChgPeakProt\_9**
4. **Overcurrent\_Fault\_ 408094978**
5. **DriveError\_Controller\_OverVoltag\_408094978**
6. **Battery LOW SOC warning\_9**

**Doubt’s:**

1. **MOSFET\_protection**
2. **Drive Error Motor Hall**
   1. **Hall sensor not working.**
3. **Temperature sense Fault\_9**
4. **FET Temperature Protection\_9**
5. **Motor stalling**
6. **Motor Phase loss**

**To do:**

1. **Discharge Over Temperature Protection\_9**
   1. **8 temperature sensors - >45 degrees**
2. **Discharge Over Temperature Protection\_9**
   1. **8 temperature sensors - <20 degrees**
3. **Cell over Voltage warning**
4. **Throttle error**
5. **Cell Under Voltage Protection**
6. **Cell over Voltage Protection**
7. **Cell over Voltage Warning**

**Not an error:**

1. **Dchg FET status\_9**
2. **Chg FET status\_9**
3. **Ignition\_status\_12**
4. **Prechg\_FET status\_9**
5. **Latch protection**
6. **Pack Under Voltage protection**
7. **FET failure**
8. **Pack over Voltage protection**
9. **Cell Over Dev Protection**