Sankar Padala
Python Developer
Email: padala.sankarreddy@gmail.com
Mobile: 8919721999
LinkedIn: SankarPadala

Professional Summary:

- Having around **8 years** of experience in Python scripting, which includes **Development**, **Enhancement Maintenance.**
- Hands-on Experience in **Python Automation script development with DEV Tools** and various **Intel** product-based components like **TTK3, TTK2, One Box** devices, etc.
- Familiarity in working with **Continuous Integration** Builds using Python
- Good Exposure to **OOPS** concepts with Python
- Proven ability to deliver quality products within tight deadlines.
- An effective communicator with excellent team-building and relationship-management skills.
- Familiarity with development best practices such as **code reviews** and **unit testing.**
- Hands-on experience in **Team City, Artifactory** build creation, and creating a **chocolate NuGet** package.
- Ability to learn new technologies with minimal time and excellence in teamwork with multi-location teams.
- Developed **Fast API** interface/API commands for the framework.
- Worked on the **GIT** Version Control Tool and the tracking tool **JIRA**
- Following the **AGILE** software development life cycle process.
- **Experience in Intel test executors:** TWS, OAP Portal, CLA, NGA
- **Possess knowledge of the tools used at Intel** Wimager OS flashing, Onebox Recovery, Xmlcli bios settings, System Scope tool, System cycling tool, Touchless tool kit, CCBSDK tool, Python_efi

Professional Experience:

- Currently working as a **Senior Developer** at **Tech Mahindra** with client **Intel** since July 14, 2021.
- Previously worked as a **Product Developer** at **UST Global** with client **Intel** from April 4, 2018, to July 13, 2021**.**
- Worked as a **Junior Developer** at **APSSDC** from June 27, 2015, to May 31, 2016**.**

Educational Qualification:

- Completed Bachelor of Technology (B-tech) from Jawaharlal Nehru Technological University, Kakinada from 2011-2015

Project Name: **Autoflow 2.0 Automation**
Role: **Python Automation developer, Team lead**
Client**: Intel**

Autoflow 2.0 Automation framework involves a set of interface commands to fetch and execute IFWI BIOS OS provisions with the help of TWS (test web services) Fetching the SUT list dynamically based on SUT properties/attributes and also fetching the firmware/image/test content dynamically from artifactory and network share path and provides API/ Interfaces for remote firmware flashing solution (hardware - TTK2/3/Y2Prog and software – FFT) and OS Imaging solution (WIMAGER) provides API/ Interfaces for test execution using different test frameworks (RAILS, STANDALONE, ONEMAP, TCSS) and also provides API/ Interfaces for Splunk and One BKC reporting

Roles & Responsibilities:

- Involving in functional discussions and understanding the requirements from the project flow architecture
- Developing interface/API commands for the AF2.0 framework
- Exploring and integrating new features as per the client's request
- Creating chocolaty NuGet packages and deploying them to dedicated Artifactory locations
- Written **scripts** for validation of Tools as per client requirements, effectively using Python, deploying the project without any issues
- Run the framework on different platforms and fix the scripts in case of any tool/test case change.
- Integrating and deploying APIs as a backend for OAP portal execution

Project Name: **Autoflow Recovery Automation**
Role: **Python Automation developer, Team Lead**
Client**: Intel**

The primary objective of this project is to outline the procedures for restoring the System Under Test (SUT) to a state suitable for automated operations. Given the frequent failures encountered, it is crucial to establish a more robust and configurable recovery process. The SUT can experience various types of failures, each necessitating a specific recovery method. By implementing these recovery strategies, we aim to ensure that the SUT is consistently restored to an operational state, thereby enhancing the stability and reliability of the testing environment. Below is a comprehensive list of common failures and their respective recovery methods to List of failures Panic or crash, Failure to boot into the operating system, Failure to launch the service that is required for automation, Bad network connectivity or unavailability, System hang, System stuck in the EFI shell

Roles & Responsibilities:

- Designed and developed the recovery flow architecture which includes the recovery methods
- Creating command line interface and API commands for the recovery
- Involving as POC for this product to the customers to fix and enhance if any new feature request
- Involving in code review and addressing, and resolving any issues or conflicts that arise within the team

Project Name: **Nickel Automation**
Role: **Python Automation developer**
Client**: Intel**

Nickel automation is an End-to-end Automation. The main moto of this project is to test platform stability which involves IFWI Flashing by using the FFT tool or through TTK devices and flashing of platform operating system with the latest WIM file then after creating and installing the required intel tool and environment to test without any involvement of human interruption. Whenever any new build is released it will trigger automatically from team city, and it will send result reports to mail automatically to all team members.

Roles & Responsibilities:

- Automating tools and connected devices based on the Requirement Creating and modifying team city build steps in Python environment Flashing **IFWI and OS** imaging on platforms with Python scripts ● Creating chocolaty builds and enabling artifactory by using a chocolaty environment.
- Directing and implementing training programs.
- Identifying areas for modification in existing programs and subsequently developing these modifications ● Integrating test case scenarios with test executor **TWS.**
- Fixing issues in case of any tool/test case in the project

Project Name: **Tools Automation**
Role: **Python Automation developer**
Client**: Intel**

Tools automation validates the new tools released by the INTEL tools development team. Automated test cases can be executed on different platforms. Tools Automation has covered GUI and CLI test cases of TAT, SST, and SX Cycling. Automated test cases will be well arranged in the desired manner as scenario JSON files. Scenarios can be executed using a TWS test executor and generate the html file with pass/fail information. Automated test cases will move to production at COSMOS once they are under acceptance criteria. Automated test cases will be utilized in the tool Validation process covering all the possible scenarios related to the tool.

Roles & Responsibilities:

- Involving in functional discussions and understanding the requirements Identifying and developing **Python scripts** for various tools
- Implementing new Python libraries for all automatable functionalities, so that the whole team can utilize the same and the code base can be well-maintained
- Written **scripts** for validation of Tools as per client requirements using Python from the design database
- Automating GUI application tools by **Send Keys** module Integrating test case scenarios with test executor **TWS**
- Controlling the test machine (SUT) from a remote server and running scripts to get hardware and software information of **SUT** using python
- Run the scripts on different platforms and fix the scripts in case of any tool/test case change.
- Maintenance of In-Production scripts in **Jama Contour** serve

DECLARATION:

I hereby declare that the details furnished above are true to the best of my knowledge and belief.

Sankar Padala