**Web UI Automation Manual**

**Table of Contents**

1. Introduction
2. System Overview
3. File Structure
4. Installation
5. Usage Guide
   * Running the Main Application
   * Creating and Managing Scripts
   * Executing Test Queues
   * Viewing Logs
6. Module Breakdown
   * script\_application.py
   * main\_window.py
   * script\_generator.py
   * action\_functions.py
   * log\_handler.py
7. Troubleshooting

**Introduction**

This application is built to automate web UI interactions with wall controllers. Users input a series of commands through a graphical interface, which are then translated into predefined actions and executed in sequence. Real-time logging and queue-based task management provide a structured, user-friendly experience.

**System Overview**

* **Input Commands:** Users provide commands via the GUI.
* **Script Generation:** Commands are translated into methods.
* **Action Execution:** Methods are executed to interact with the web UI.
* **Logging:** Actions and events are logged and viewable in real-time.

**File Structure**

* script\_application.py – Main entry point.
* main\_window.py – GUI logic and user interaction.
* script\_generator.py – Converts user commands into script actions.
* action\_functions.py – Defines executable methods for UI automation.
* log\_handler.py – Manages logging and log display.

**Installation**

Follow the same steps as outlined in the Log Viewer Manual.

**Usage Guide**

**Running the Main Application**

Start the application with:

python script\_application.py

**Creating and Managing Scripts**

1. Launch the GUI.
2. Enter automation commands in the provided input field.
3. Commands are translated into actionable scripts using script\_generator.py.

**Executing Test Queues**

* Commands are queued in the order entered.
* The queue system processes each task sequentially.
* Visual feedback is provided via the GUI.

**Viewing Logs**

* Logs are displayed for each wall controller.
* Accessible via dynamically generated log tabs.

**Module Breakdown**

**script\_application.py**

* **Role:** Main execution script.
* **Function:** Initializes the GUI and links modules.

**main\_window.py**

* **Role:** Manages the PyQt5 GUI.
* **Key Features:**
  + Handles user input.
  + Updates the test queue.
  + Connects with the log handler for real-time feedback.

**script\_generator.py**

* **Role:** Translates user commands to functions.
* **Responsibility:** Interfaces with action\_functions.py to perform actions.

**action\_functions.py**

* **Role:** Houses defined methods that mimic web UI actions.
* **Examples:** Login, navigate, click buttons, extract data.

**log\_handler.py**

* **Role:** Collects and displays logs.
* **Design:** Independent logging class used by main\_window.py.

**Troubleshooting**

* **Queue Not Processing:** Ensure commands are valid and correctly formatted.
* **Web UI Not Responding:** Check network access to wall controllers.
* **Missing Logs:** Verify log\_handler is correctly initialized.