

Unit Testing Documentation for Port Checker Application

1. Introduction

This document provides an overview of the unit testing strategy implemented for the Port Checker application. The tests are designed to validate core functionalities of the backend services and the GUI components of the application, ensuring robustness and reliability.

2. Tools and Frameworks

- **Python's unittest Framework:** Used for writing and running test cases.
- **PyQt5:** For GUI testing, simulating user interactions like clicks and input.
- **Mocking with unittest.mock:** Used to simulate certain behaviors in GUI tests without requiring manual intervention.

3. Test Structure

The tests are divided into two main categories:

1. **Backend Function Tests (test_port_checker.py):** Focus on testing individual backend functions like IP resolution, port checking, and network reachability.
2. **GUI Tests (test_port_checker_gui.py):** Focus on testing the GUI components of the application, such as buttons and dialogs.

4. Backend Function Tests (test_port_checker.py)

Test Cases Overview

1. test_get_workstation_ip

- **Purpose:** Verifies that the function returns a valid IP address string.
- **Assertions:**
 - The result is a string.
 - The result is not an empty string.

2. test_resolve_host_to_ip

- **Valid Case:**
 - **Input:** 'localhost'

- **Expected Output:** '127.0.0.1'

- **Invalid Case:**

- **Input:** 'invalid-host'
- **Expected Output:** None

3. **test_perform_nslookup**

- **Valid Case:**

- **Input:** 'localhost'
- **Expected Output:** Contains 'NSLookup Result'

- **Invalid Case:**

- **Input:** 'invalid-host'
- **Expected Output:** Contains 'NSLookup failed'

4. **test_check_local_tcp_port_open**

- **Purpose:** Tests the status of a local TCP port.
- **Assertions:**
 - Protocol is 'TCP'.
 - Status is one of 'Open', 'Closed', or 'Error'.

5. **test_check_remote_tcp_port_open**

- **Purpose:** Tests the status of a remote TCP port.
- **Assertions:**
 - Protocol is 'TCP'.
 - Status is one of 'Open' or 'Closed'.

6. **test_identify_service**

- **Purpose:** Identifies the service associated with a port number.
- **Assertions:**
 - For port 80, returns 'HTTP'.
 - For port 9999, returns 'Unknown'.

7. test_is_ip_reachable

- **Purpose:** Checks if an IP address is reachable.
- **Assertions:**
 - '8.8.8.8' (valid) returns True.
 - '256.256.256.256' (invalid) returns False.

5. GUI Tests (test_port_checker_gui.py)

Test Cases Overview

1. test_nslookup_button

- **Purpose:** Tests the NSLookup feature within the GUI using mocking to simulate dialog interaction.
- **Method:** Uses unittest.mock.patch to replace the exec_ method of QMessageBox with a mock that simulates clicking "OK".
- **Assertions:** Confirms the mock was called, indicating the dialog interaction occurred as expected.

2. test_dark_mode_toggle

- **Purpose:** Tests the dark mode toggle functionality.
- **Assertions:**
 - The checkbox is checked when clicked.
 - The checkbox is unchecked when clicked again.

3. test_text_box_inputs

- **Purpose:** Verifies that text boxes accept the correct input values.
- **Assertions:** Text boxes contain expected values after input.

4. test_buttons

- **Purpose:** Tests buttons for checking local and remote ports.
- **Assertions:** Confirms that the results table is updated correctly when buttons are clicked.

5. test_ping_button

- **Purpose:** Verifies the ping functionality within the GUI.
- **Assertions:** Confirms that the results table updates correctly after clicking the ping button.

6. Running the Tests

To run the tests, use the following command in the terminal:

```
python -m unittest discover -s tests -p "test_port_checker.py"
```

```
python -m unittest discover -s tests -p "test_port_checker_gui.py"
```

This command discovers and runs all test files starting with test_ in the tests directory.

7. Conclusion

These unit tests provide a robust way to validate the functionality of the Port Checker application, ensuring that both backend services and GUI components work as intended. By catching bugs early through automated testing, the application becomes more reliable and easier to maintain.