**Basis of Design**

**1. Development of Python Script**

**a. Translating Pseudocode to Python Script**

The script will be developed based on the given pseudocode, ensuring that all required functionalities are implemented using Python's built-in libraries and essential language elements. The script will:

* Accept user input for hostnames.
* Store and manipulate the entered data.
* Resolve hostnames to IPv4 addresses.
* Handle user selection through index-based input.
* Implement error handling for invalid inputs.

**b. Code Documentation with Comments**

The script will include comprehensive inline comments explaining:

* Purpose and functionality of each function.
* Data processing and manipulation steps.
* Error handling mechanisms.
* Expected output for each block of the code.

**2. Debugging and Testing**

**c. Reviewing and Debugging**

The script will be reviewed for logical and syntax errors, ensuring correct operation. The debugging process will include:

* Step-by-step execution to identify any issues.
* Use of try-except blocks for error handling.
* Printing debug statements where necessary.
* Conducting test cases as outlined below.

**Test Cases**

The script will be tested under the following scenarios:

|  |  |
| --- | --- |
| **Test Case** | **Description** |
| 1 | User fails to enter any hostnames (empty input). |
| 2 | User enters examplex.org scanme.org hackthissite.org, then selects examplex.org for resolution. |
| 3 | User enters examplex.org scanme.org hackthissite.org, selects scanme.org, then hackthissite.org for resolution. |
| 4 | User enters examplex.org scanme.org hackthissite.org and inputs an out-of-range index. |
| 5 | User enters examplex.org scanme.org hackthissite.org and inputs a non-integer index. |

Each test will verify that the script correctly processes input and returns appropriate results or error messages.

**3. User Documentation**

**d. Script Operation**

**i. Launching the Script**

* The script is executed using Python 3.
* Required libraries: socket (for hostname resolution).
* Command to run: python script.py

**ii. Entering Data**

* Users will be prompted to enter hostnames separated by spaces.
* After entering hostnames, they will be asked to select one by entering its index.

**iii. Error Handling**

|  |  |  |
| --- | --- | --- |
| Error Message | Meaning | Resolution |
| "No hostnames entered." | User did not provide any input. | Enter at least one hostname. |
| "Invalid index. Out of range." | Entered index is beyond available options. | Enter a valid index within the range. |
| "Invalid input. Please enter a number." | User inputted a non-integer index. | Enter a valid numerical index. |
| "Hostname resolution failed." | Unable to resolve domain to IP. | Check network connection or domain validity. |

**iv. Explanation of Results**

* If successful, the script will display the resolved IPv4 address of the selected hostname.
* If unsuccessful, appropriate error messages will be shown.

This document ensures a structured approach to developing, testing, and documenting the Python script for hostname resolution.