I. Project Title

The project's name : **Scan-Bondanh : Network Scanning & Security Assessment Tool**

II. Introduction

In Cyber Security , identifying active devices and open ports within a network is essential for securing systems, detecting threats, and performing routine network assessments. Beginners often lack simple, easy-to-use tools to understand how scanning works.

Scan-Bondanh aims to provide a lightweight, educational, and easy-to-use network scanning tool written in Python. The project helps students understand the fundamentals of network reconnaissance, port scanning, service detection, and vulnerability analysis.

III. Objectives

- Develop a Python-based tool capable of scanning IP ranges to find active hosts,and implement detailed port scanning and perform service detection.
- Store scan results and history through file handling (JSON/TXT).
- Apply core Python concepts including OOP, data structures, file handling, and error handling.

IV. Proposed Solution

1. Main Features

To address the cyber security need , we're building a python based-system that

1. **Host Discovery :** discover the active IP host in IP range
2. **Port Scanning :** Scan the port that open
3. **Report Generation**
4. **Banner Grabbing**: Attempts to identify services (e.g., SSH, HTTP).

2. Python Concept

Python Concepts Used:

1. **OOP Concept**
   - **Encapsulation**: : Bundle class and keeping the details of port scanning and service detection hidden.
   - **Inheritance**: Used to handle different scan types (host scanning, port scanning, service detection) by extending a base class.
   - **Polymorphism** (Optional): You could allow different types of scans (e.g., TCP, ICMP) to be treated as the same object type via method overriding, making the tool extensible for future scans.
2. **Data Structure:** Lists for ports, dictionaries for reporting results.
3. **File Handling:** Python's file handling functions will allow us to securely store and output the report.

V. Methodology

1. Outline

The Project will follow these step :

1. **Set Up the Environment:** Install necessary libraries like nmap , scrappy , socket etc, and set up a GitHub repository for version control.
2. **Develop MVP Core Feature** Build host scanner, Build port scanner and add banner grabbing
3. **Testing all features :** We test it by scanning multiple networks , testing against the firewall.
4. **Library**
    - **socket:** Used for network communication
    - **ipaddress:** Validates IP ranges and subnet inputs.
    - **subprocess:** Executes ping commands for host discovery.
    - **json:** Stores scan results, reports, and configurations in a structured format.
    - **time:** Tracks scan duration, adds delays, and handles timeouts.
    - **os:** Interacts with file paths and detects the OS for specific ping commands.

2. Technology used

1. **Python**
   Python is the main programming language used to develop the Scan-Bondanh tool. It is chosen for its simplicity, powerful networking libraries, and strong support for cybersecurity applications.
2. **Visual Studio Code (VS Code)**
   VS Code is used as the primary code editor and development environment. It provides:

- Syntax highlighting for Python
- Built-in terminal for testing
- Debugging tools
- Extension support for Git and Python
  VS Code helps improve productivity and ensures clean, well-organized code development.

3. **GitHub**
   GitHub is used for version control and project collaboration. It allows:

- Storing and managing source code securely
- Tracking changes and updates
- Team collaboration
- Backup and recovery of project files
  GitHub also serves as a platform for project submission and documentation.

4. **Telegram**
   Telegram is used as a communication and notification platform for the project. It helps in:

- Team communication and coordination
- Sharing updates and progress
- Sending scan notifications or alerts (optional future feature)
- Quick troubleshooting and feedback

Together, these technologies ensure that Scan-Bondanh is developed efficiently, securely, and collaboratively, while also being easy to maintain and extend in the future.

VI. Timeline

The project will be developed over four weeks, starting from **November 18, 2025**.

Week 1 (Nov 18 – Nov 24) : Research and complete the project proposal

Week 2 (Nov 25 – Dec 1) : begin implementing core scanning components.

- **Day 1–2:** Set up necessary libraries, project folder structure, and CLI
- **Day 3–5:** Implement host discovery using ping and TCP fallback.
- **Day 6–7:** Test host discovery across multiple networks.

Week 3 (Dec 2 – Dec 8)

Develop port scanning and service detection.

- **Day 8–10:** Implement threaded port scanning and service mapping.
- **Day 11–12:** Add FTP/Web service probe for active hosts
- **Day 13–14:**Integrate all modules

Week 4 (Dec 9 – Dec 15)

Finalize the tool, add logging, and complete full testing.

- **Day 15–17:** Implement logging (JSON/TXT) and error handling.
- **Day 18–19:** Perform complete testing against open networks and firewall-protected networks.
- **Day 20–21**: Final debugging, documentation, and project submission preparation.

VII. Expect Outcome

The Py-NetScan Network Scanning Tool will:

- Detect active hosts in an IP range , Scan and identify open ports
- Map common services (HTTP, HTTPS, FTP, SSH, etc.)
- Generate clear TXT/JSON scan reports
- Log all scanning activity for auditing and troubleshooting.
- Handle errors gracefully, including invalid ranges, unreachable hosts, and timeouts.

Check the Detail Proposal Here : **Detail_G1_T3_Proposal**