National College of Ireland

Configuration Manual

MSc Research Project
MSc Cybersecurity

Sai Veeranjaneya Boppana

Student ID: x23200391

School of Computing
National College of Ireland

Supervisor: Niall Heffernan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: …..Sai Veeranjaneya Boppana……

Student ID: …x23200391……..……..…….

Programme: …MSc Cybersecurity…… **Year:** ……01/2024……..

Module: …..MSc Research Project……..……

Lecturer: ……Niall Heffernan……..……

Submission Due Date: ……29/01/2025……..……

Project Title: ENHANCING SECURITY IN IOT HOME AUTOMATION SYSTEMS. ……

Word Count: ……803…… **Page Count:** ……04……..……..……

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: …Sai Veerajaneya Boppana……..……

Date: …29/01/2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Sai Veeranjaneya Boppana

x23200391

1. Setting Up and Running the Node-RED Simulation

Option 1: Running Locally

Prerequisites Installation on Windows

1. **Install Node.js and npm**:
   - Download and install **Node.js** (version 18 or above) from the official website: https://nodejs.org/.
   - Verify installation:
     - Open the Command Prompt and run

```
node --version
npm –version
```

2. **Install Node-RED**:
   - Install Node-RED globally using npm:

```
npm install -g --unsafe-perm node-red
```

3. **Install Mosquitto MQTT Broker**:
   - Download and install the **Mosquitto MQTT Broker** from: https://mosquitto.org/download/.
   - Add the Mosquitto installation directory to the system PATH for easy access.
4. **Start Node-RED**:
   - Open a Command Prompt or PowerShell terminal and run:

```
node-red
```

   - Access the Node-RED editor in your browser at `http://localhost:1880`.

Importing the Simulation Flow

1. Access the Node-RED editor in your browser.
2. Click the menu icon (top-right corner), select **Import**, and upload the `flows.json` file provided in the project.

Option 2: Using Pre-Deployed Node-RED Instance (Preferred)

- **Access the Hosted Simulation**:
  - Node-RED simulation has been pre-configured and running as a service on an AWS EC2 instance. You can access it using the link: http://13.49.66.216:1880/#flow/1d8af67fc7ea6b91.
- **Managing the Node-RED Service on AWS Instance**:
  - To manage the Node-RED service running on the AWS instance, follow these steps:
    1. **Obtain Access**:
       - Ensure you have the `my_key.pem` file (key pair) for the AWS instance.
    2. **Connect to the Instance**:
       - Open a terminal and run, replace <key> with the actual path of key:

```
ssh -i <key> ec2-user@13.49.66.216
```

    3. **Start, Stop, or Check the Status of Node-RED**:
       - Start the service:

```
sudo systemctl start nodered.service
```

       - Stop the service:

```
sudo systemctl stop nodered.service
```

       - Check the service status:

```
sudo systemctl status nodered.service
```

    4. **Restart the Service**:
       - If any issues occur, restart Node-RED:

```
sudo systemctl restart nodered.service
```

- **Managing Mosquitto MQTT Broker**:
  - The Mosquitto service is also running on the AWS instance. Use similar commands to manage it:
    - Start the service:

```
sudo systemctl start mosquito
```

    - Check the service status:

```
sudo systemctl status mosquito
```

2. Python Environment Configuration

Opening and Configuring the Artefact

1. Open the project folder (containing the Python scripts and `requirements.txt`) in **Visual Studio Code (VS Code)**.
    - If you don't have VS Code installed, download it from: https://code.visualstudio.com/.
2. Open the VS Code terminal:
    - Go to the **Terminal** menu and select **New Terminal**.

Setting Up Python and Dependencies

1. Ensure Python **3.12.6** is installed:
    - Download Python from https://www.python.org/.
    - During installation, check **Add Python to PATH**.
    - Verify installation in the terminal:

```
python –version
```

2. Install required Python libraries:
    - In the terminal, navigate to the artefact folder (if not already there):

```
cd path_to_artefact_folder
```

    - Install dependencies from `requirements.txt`:

```
pip install -r requirements.txt
```

---

3. Running the Application

Streamlit Dashboard

1. **Run Locally in VS Code**:
    - Start the Streamlit dashboard by running:

```
streamlit run dashboard.py
```

    - Access the dashboard in your browser at: `http://localhost:8501`.
2. **Pre-Deployed Dashboard**:
    - Alternatively, access the deployed dashboard at: https://homesecurity.streamlit.app/.

Model Training and Evaluation

- **Training the Model**:
  - To retrain the model, run the `model.py` script:

    ```
    python model.py
    ```

  - This will save a trained model for future use.
- **Evaluating the Model**:
  - Use `evaluate.py` to test the model's performance:

    ```
    python evaluate.py
    ```

Additional Tips for Using VS Code

1. **Working with the Terminal**:
   - Use integrated terminal commands for smooth workflow.
   - Clear terminal history when necessary:

     ```
     cls
     ```

2. **Debugging**:
   - Launch Python scripts with debugging enabled via VS Code's **Run and Debug** tab.
3. **Managing Dependencies**:
   - If you need to add a library, install it directly from the terminal:

     ```
     pip install library name
     ```