National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Tejas Bhanarkar
Student ID: x23287004

School of Computing
National College of Ireland

Supervisor: Vikas Tomer

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Tejas Bhanarkar
Student ID:	x23287004
Programme:	Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	Vikas Tomer
Submission Due Date:	12/09/2025
Project Title:	Configuration Manual
Word Count:	Aprox. 800
Page Count:	11

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:	
Date:	13th September 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

Tejas Bhanarkar
x23287004

1 Introduction

This manual shows us the full configuration and setup of a distributed Multi-Agent System, which is designed for static malware identification. This system uses autonomous agents that are developed using the SPADE framework with integration of machine learning and use of a watchdog-based file monitoring system to automate triggering. The agents coordinates using XMPP messaging via Openfire Software and emulates blockchain logging for secure threat evidence storage.

2 Environment Setup

2.1 Install Java

To install Java **On Windows:**
Step 1 is to download the JDK from Oracle Website or use OpenJDK as shown in the figure below.

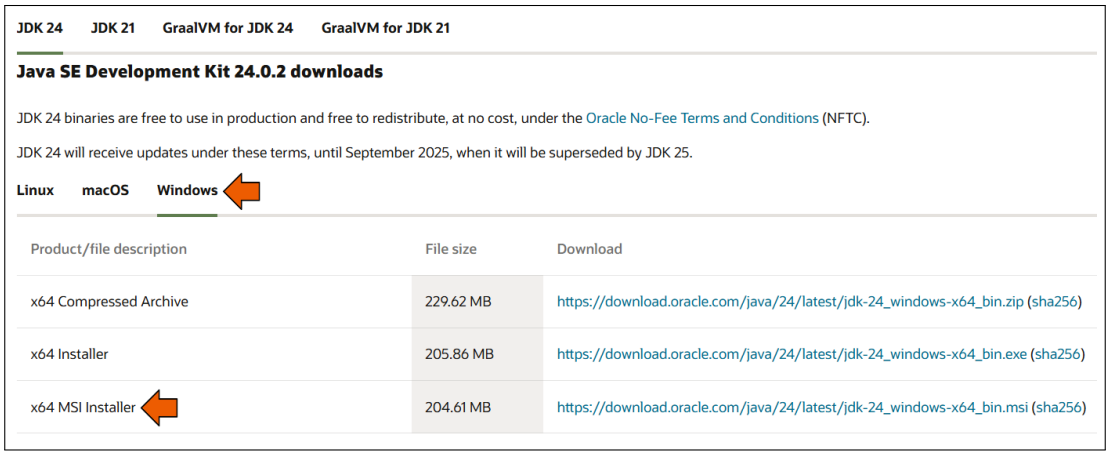

Figure 1: JDK Download from Oracle or OpenJDK

Step 2 is to run the installer and follow the instruction as displayed below.

Figure 2: Click Next

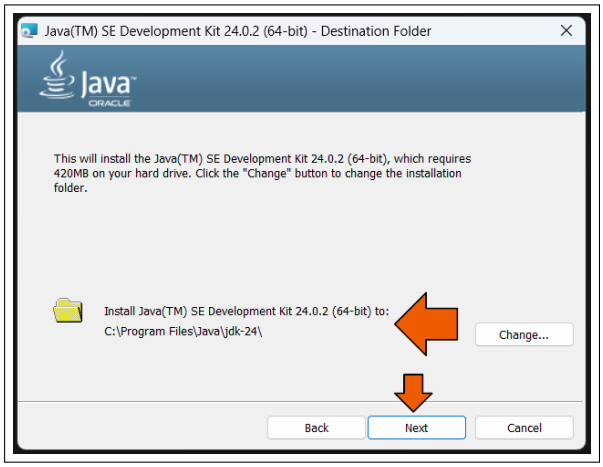

Figure 3: Don't Change the Directory and Click Next

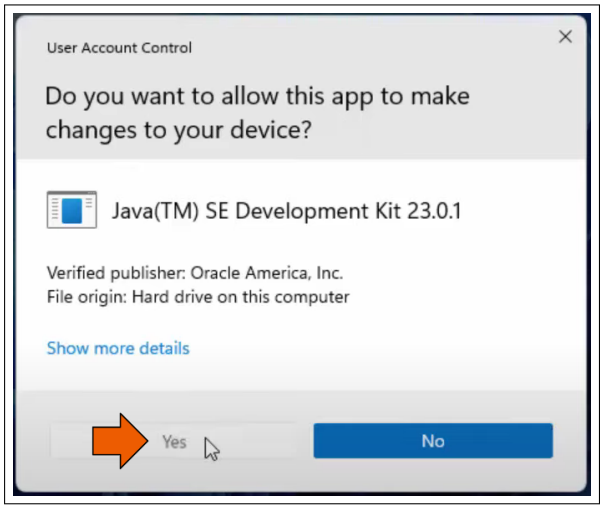

Figure 4: Click Yes

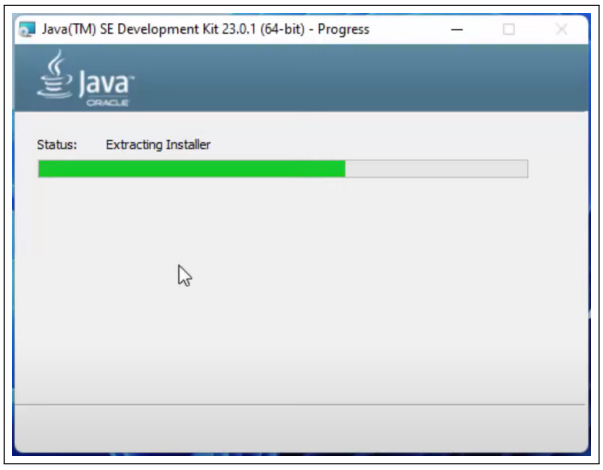

Figure 5: Under Progress

Figure 6: Sequential Steps in JDK Setup Process

After installation, Verify through the terminal with this command.

Figure 7: Verification Process

Final Step is to Set Environment Variable to avoid any type of hardcoding paths in your code or scripts. To complete the installation:

- Add `JAVA_HOME` pointing to the JDK installation directory.
- Add `%JAVA_HOME%\bin` to your `PATH`.

2.2 Install Openfire Architecture

Step 1 is to download the Openfire Software from Igniterealtime Website as shown in the figure below.

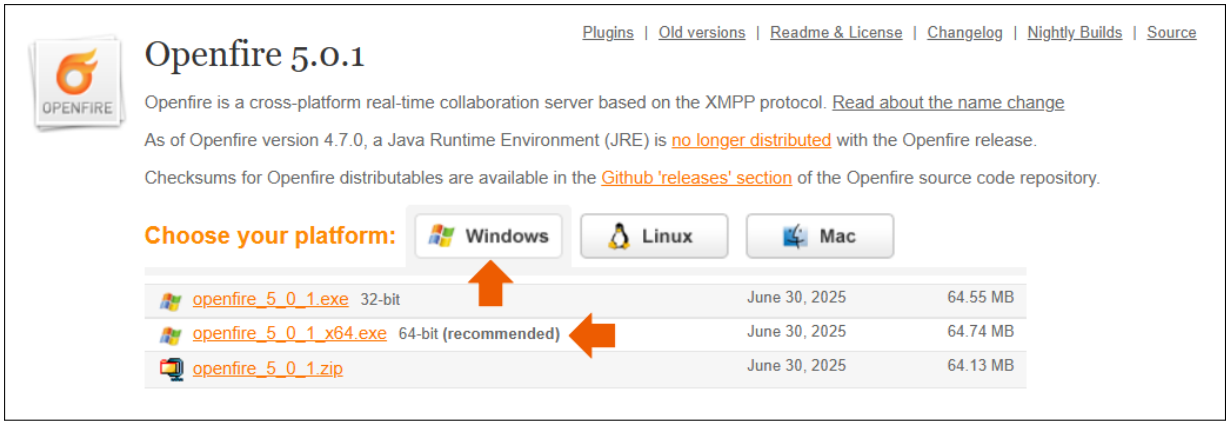

Figure 8: Download Openfire Software from Igniterealtime

Step 2 is to run the installer and follow the instruction as displayed below.

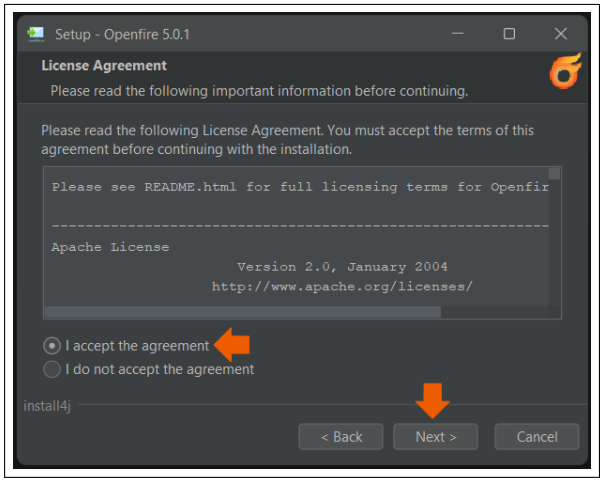

Figure 9: Accept and Click Next

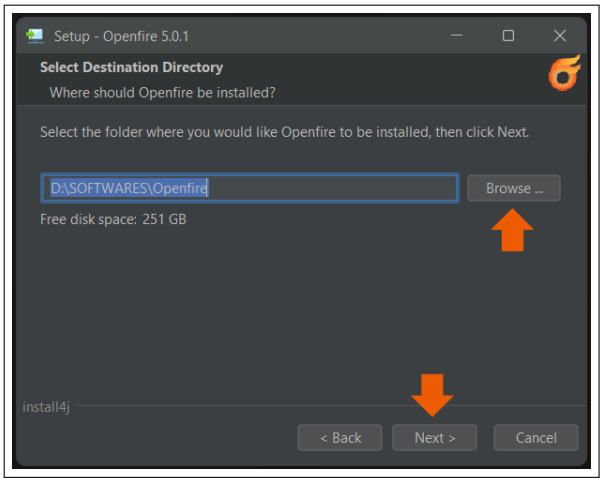

Figure 10: Click Next

After Installation launch the admin setup in any web browser (http://localhost:9090).

Step 4 is to do Initial Configuration (One-Time Setup). You have to go through these configuration steps:

1. **Choose Language** as per Preference
2. **Server Settings:**
   - Domain: `localhost` or your domain
   - Ports: leave default unless you need customization
3. **Database Settings:**
   - Option 1: Use embedded database (for testing)
   - Option 2: External DB like MySQL/PostgreSQL.
4. **Admin Account Setup:**
   - Provide an admin email and password

Step 5 is to Add Users and Configure Clients. To do this follow below steps.

1. First, Log in to the admin panel at `http://localhost:9090` in any web browser on your system.

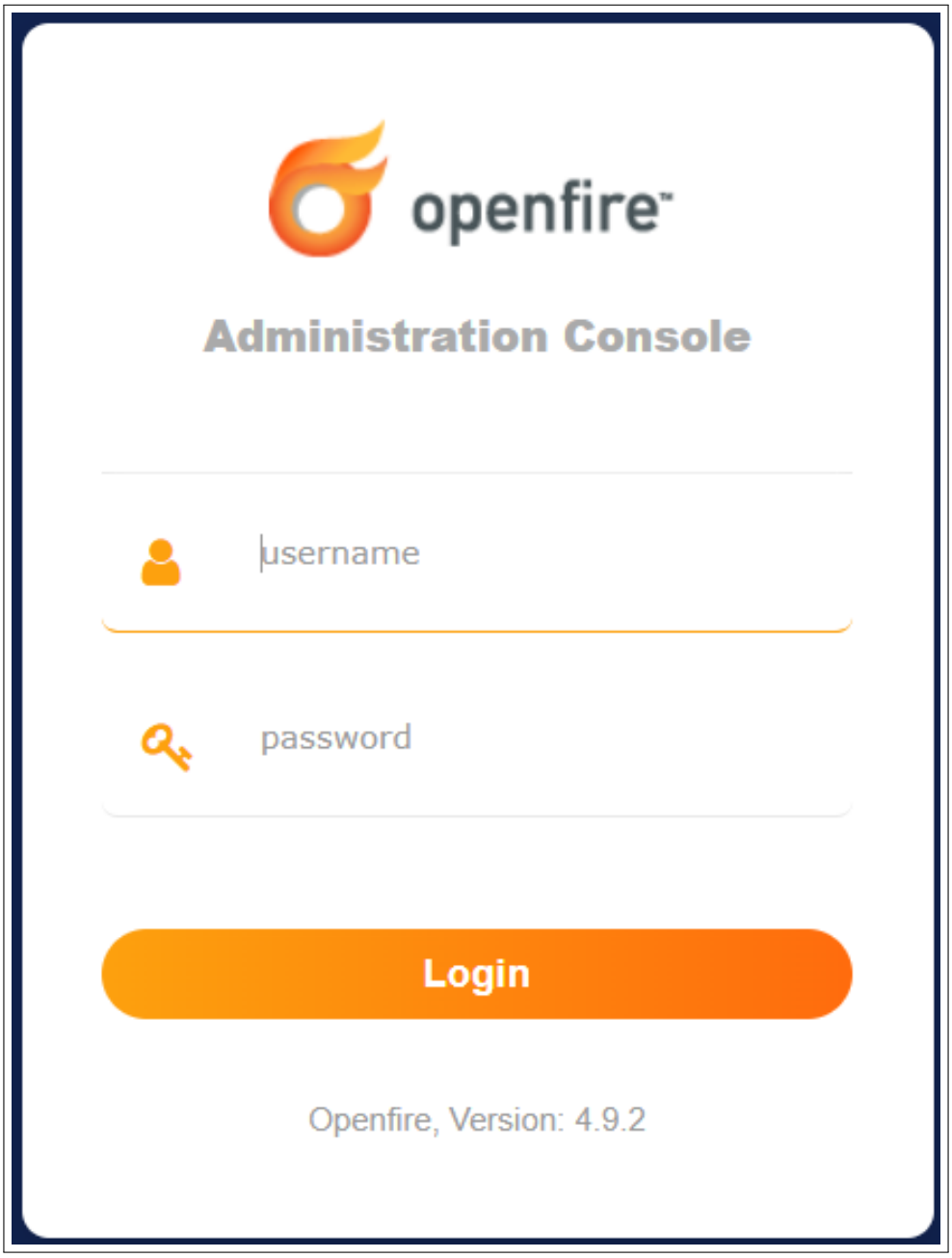

Figure 11: Login Panel

2. Second, Navigate to the **Users/Groups** section and from there click to **Add new users**.

3. Third, Configure your XMPP clients such as *Spark*, *Pidgin* or *Custom Agents* with:

   Go to **Server** through your Openfire domain or IP then add your **Username/Password** and keep the port 5222 (default).

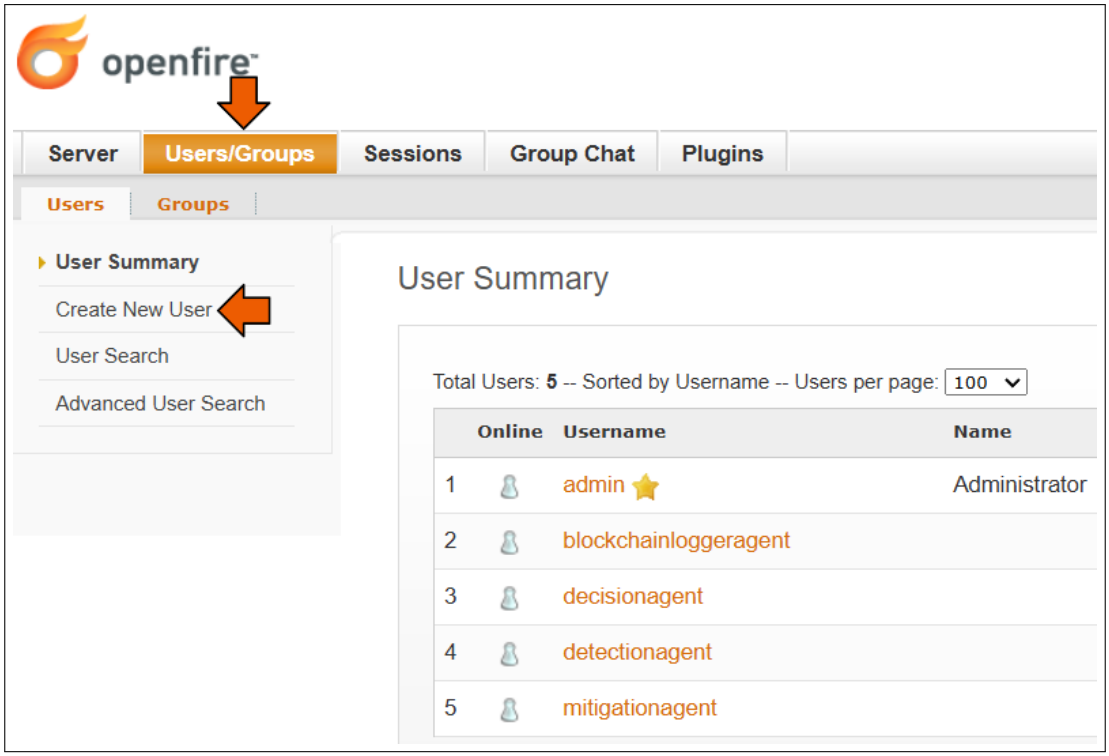

Figure 12: Create New Users

2.3 Verify XMPP Server Connectivity

From my MAS system which is in Python, try this quick test with SPADE:

```python
from spade.agent import Agent

class TestAgent(Agent):
    async def setup(self):
        print("✅ XMPP Connection Successful")

agent = TestAgent("detectionagent@10.33.66.165", "Tejas2k25")
agent.start(auto_register=True)
```

Figure 13: Verify Connectivity

2.4 Install and Setup Python Environment

To run the Multi-Agent system, Python 3.8 or later is required. It is recommended to use a virtual environment such as Anaconda to manage dependencies.

Step 1 is to install Python from Microsoft Store as shown in the figure below.

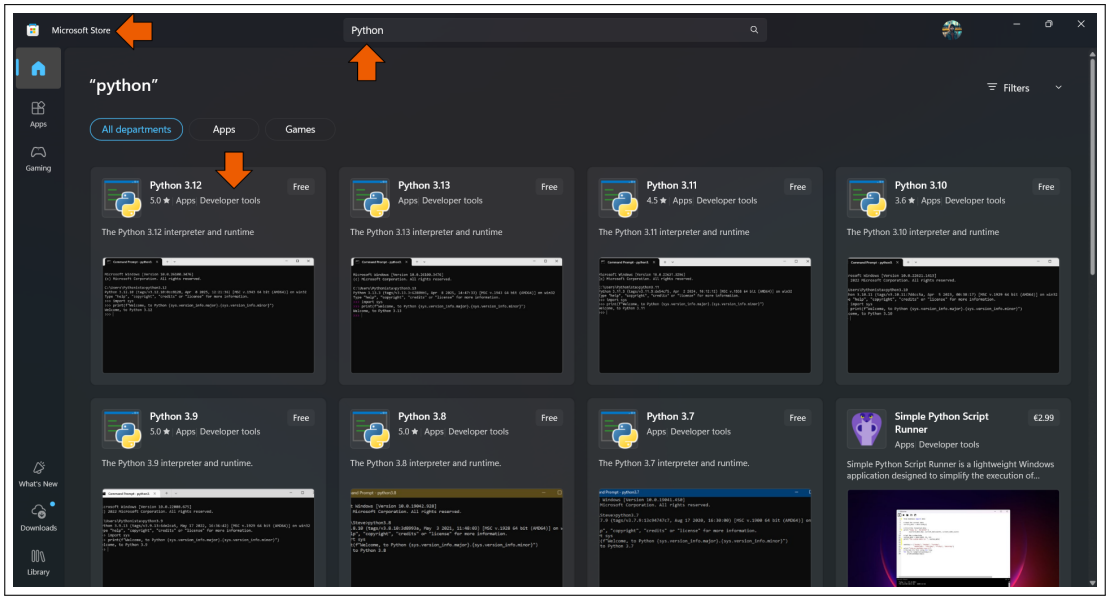

Figure 14: Search and Click Latest Python Version

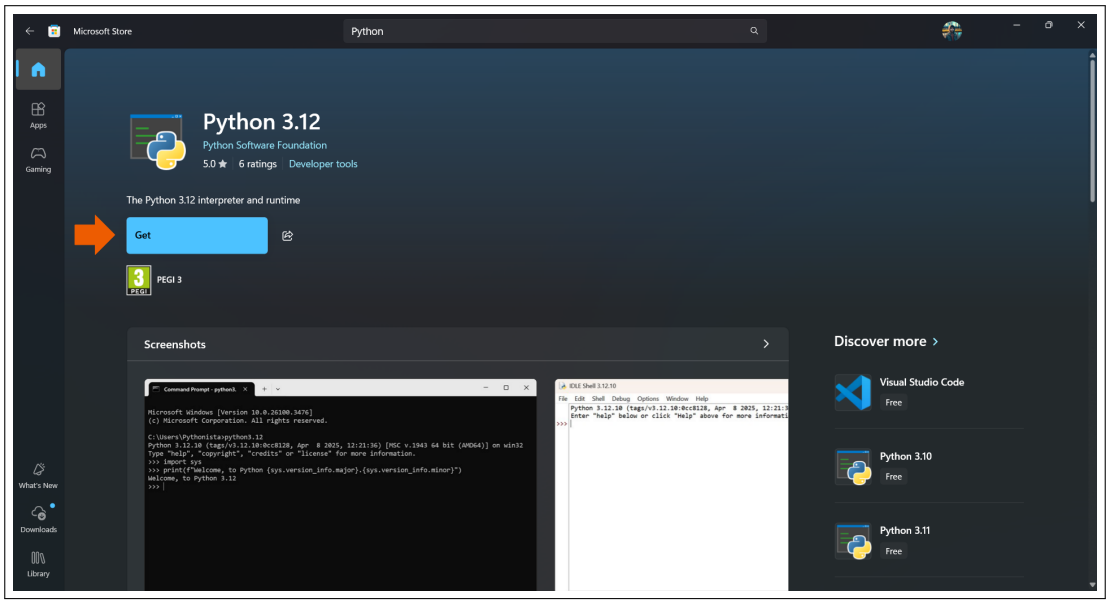

Figure 15: Click Get and Install

Step 2 is to verify Python is installed or not using the command as displayed below.

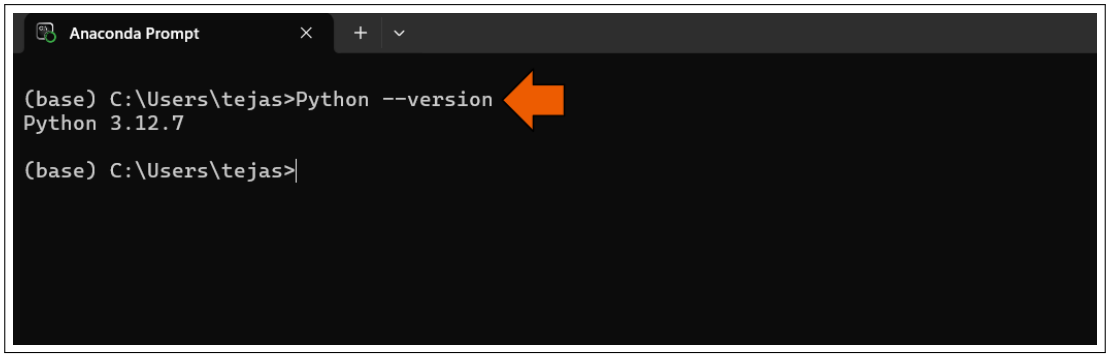

Figure 16: Verification Process

Step 3 is to install required Python libraries to run MAS system using the command as displayed below.

Figure 17: Install Python Libraries

2.5 Train the Model

This machine learning model used for malware detection is trained using a **Random Forest Classifier**. To do this **Data.csv** file has been use which contains static features extracted from executable files. The **Train.py** script includes the Python code executing data split action into training and testing sets, train the model, evaluate its accuracy and saves it as **Malware-Model.pkl**. It also exports a sample test rows to **Test-Data-With-Labels.csv** file for use by the Detection Agent.

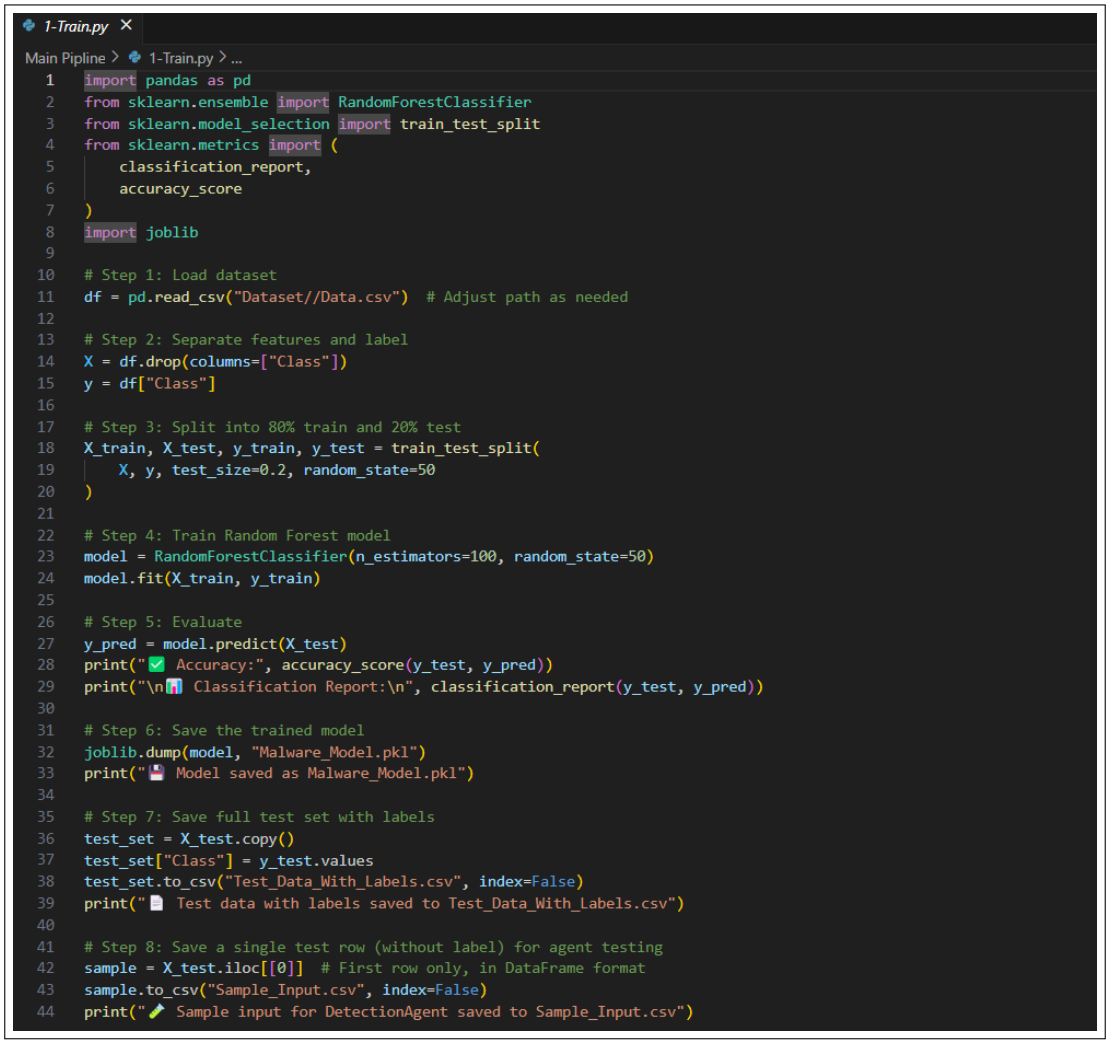

Figure 18: Train Model Using Random Forest Classifier

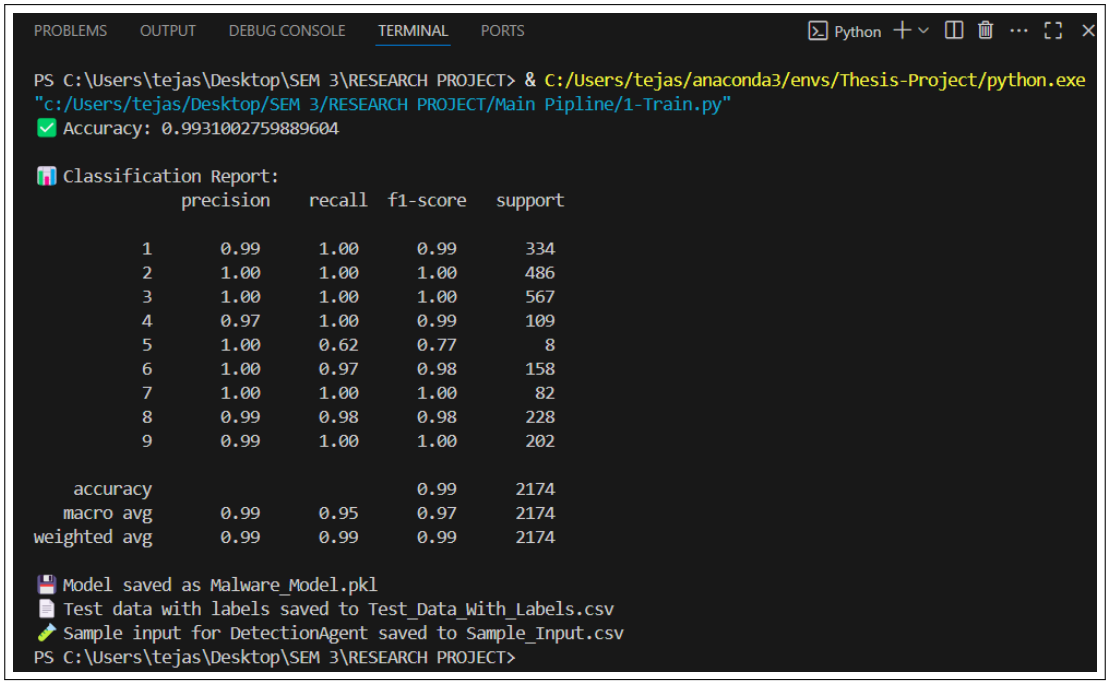

Figure 19: Model Output

2.6 Final Checklist Before Running MAS Architecture

Component	Status
Java Installed	Yes
Openfire Running	Yes
Agent Accounts Created	Yes
Python Environment Set	Yes
Model Trained	Yes
Monitor Ready	Yes

Table 1: System Readiness Checklist Before MAS Execution

2.7 Final Result of the Pipeline

This section demonstrates the working of the complete **MAS pipeline.** A **test .exe** file is used to trigger **detection, classification, mitigation, and logging.** Screenshots validate each agent's successful response and communication.

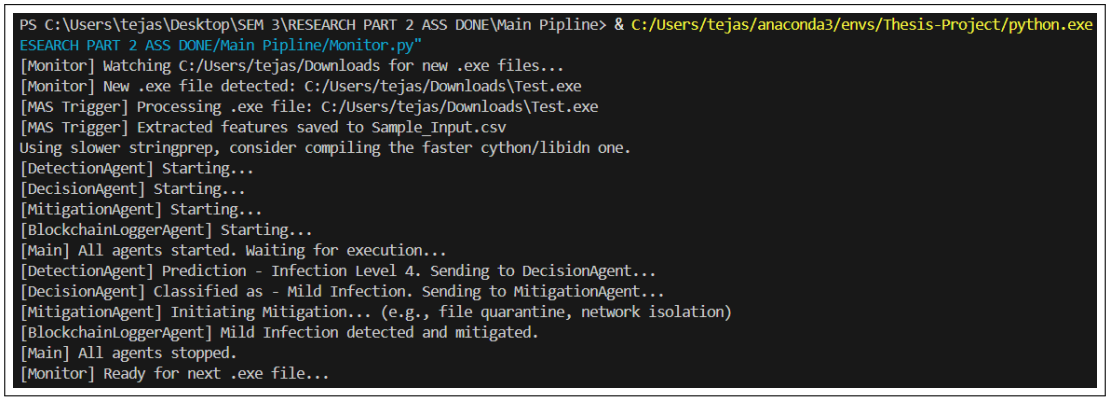

Figure 20: Final Pipeline Test