

# 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



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

<b>Signature:</b>	
<b>Date:</b>	13th September 2025

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:**

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

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

<b>Office Use Only</b>	
Signature:	
Date:	
Penalty Applied (if applicable):	

# Configuration Manual

Tejas Bhanarkar  
x23287004

## 1 Introduction

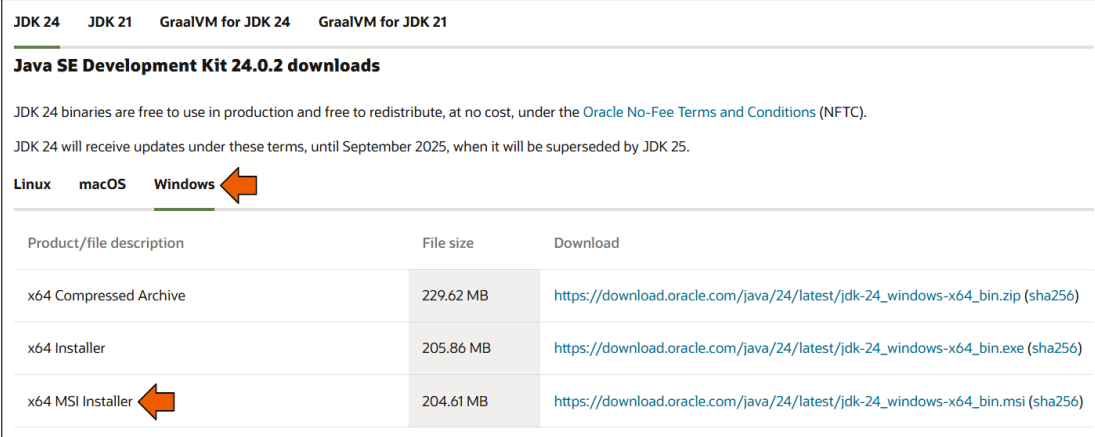
This manual shows us the full configuration and setup of a distributed Multi-Agent System, which is designed for static malware identification. This system uses autonomous agents that are developed using the SPADE framework with integration of machine learning and use of a watchdog-based file monitoring 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.



Product/file description	File size	Download
x64 Compressed Archive	229.62 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.zip">https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.zip</a> (sha256)
x64 Installer	205.86 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.exe">https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.exe</a> (sha256)
x64 MSI Installer	204.61 MB	<a href="https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.msi">https://download.oracle.com/java/24/latest/jdk-24_windows-x64_bin.msi</a> (sha256)

Figure 1: JDK Download from Oracle or OpenJDK

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

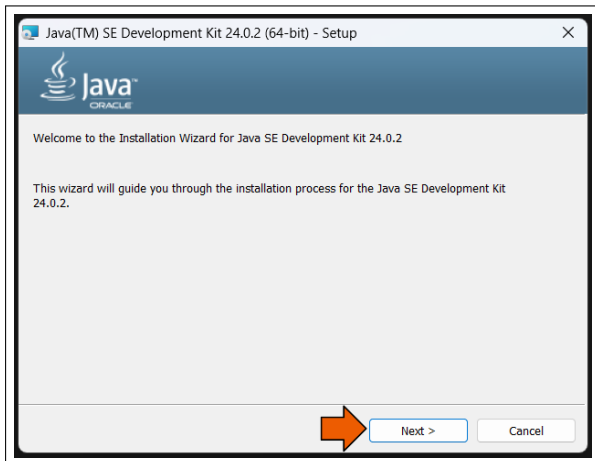


Figure 2: Click Next

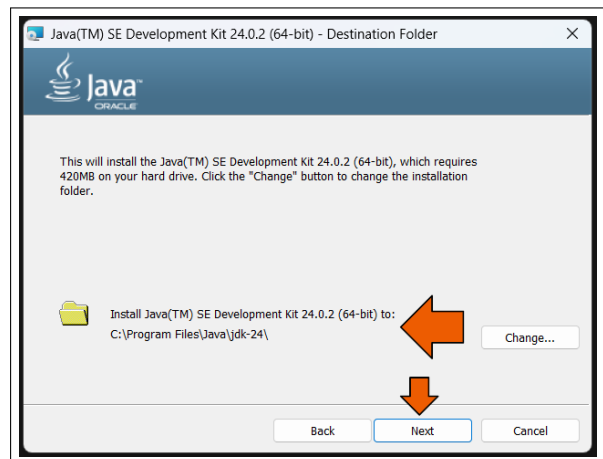


Figure 3: Don't Change the Directory and Click Next

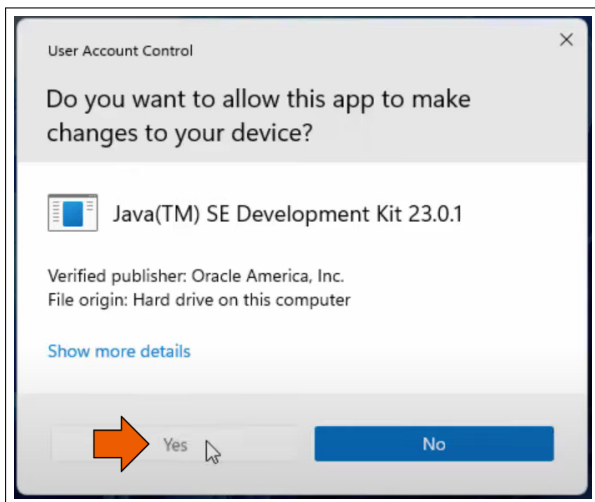


Figure 4: Click Yes

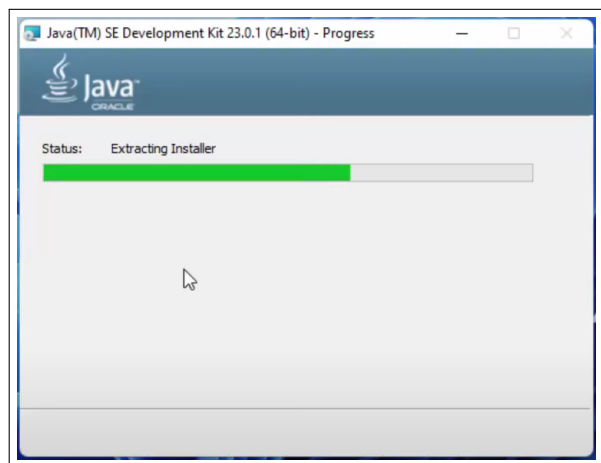


Figure 5: Under Progress

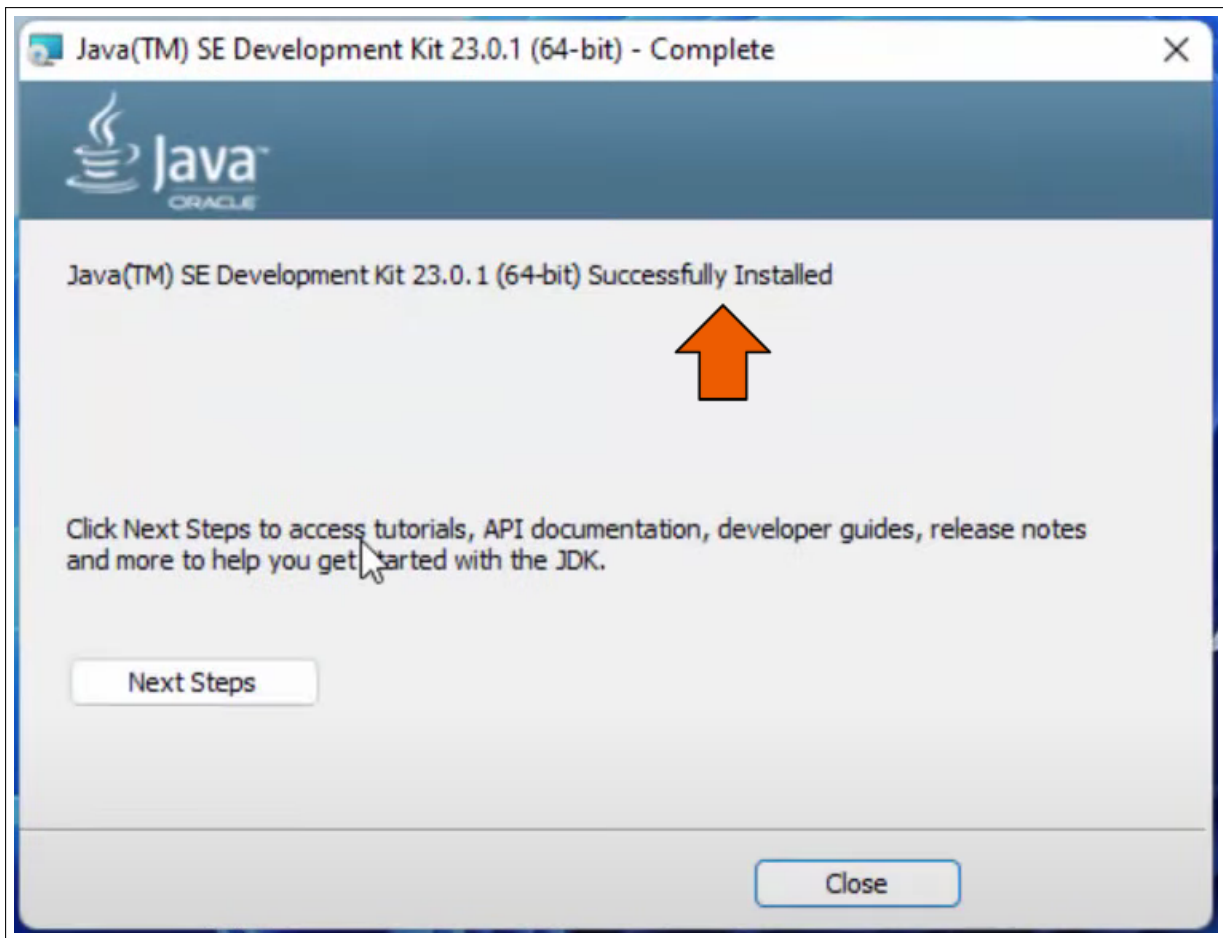


Figure 6: Sequential Steps in JDK Setup Process

**After installation**, Verify through the terminal with this command.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\tejas> java --version
java 24.0.1 2025-04-15
Java(TM) SE Runtime Environment (build 24.0.1+9-30)
Java HotSpot(TM) 64-Bit Server VM (build 24.0.1+9-30, mixed mode, sharing)
PS C:\Users\tejas> |
```

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.

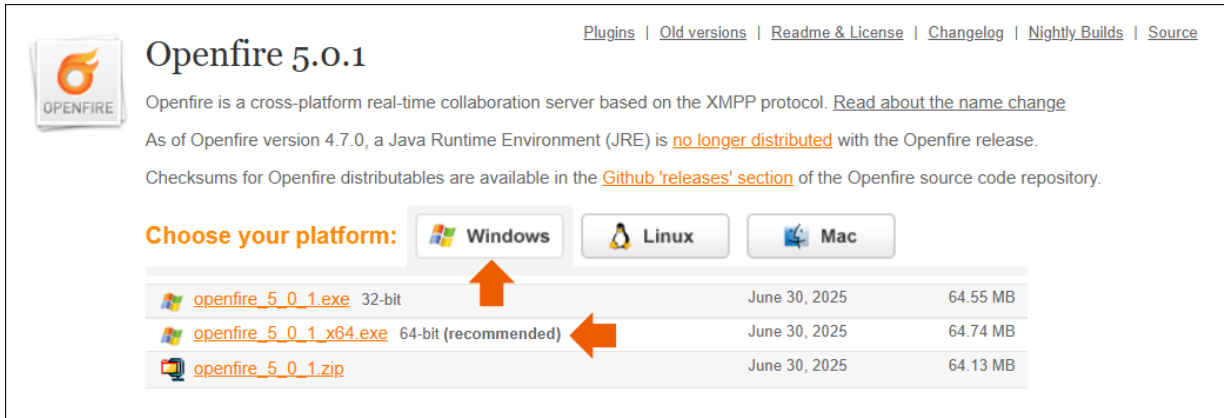


Figure 8: Download Openfire Software from Igniterealtime

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

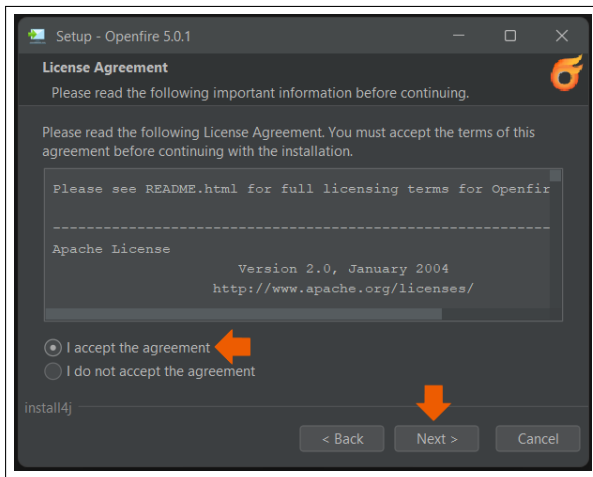


Figure 9: Accept and Click Next

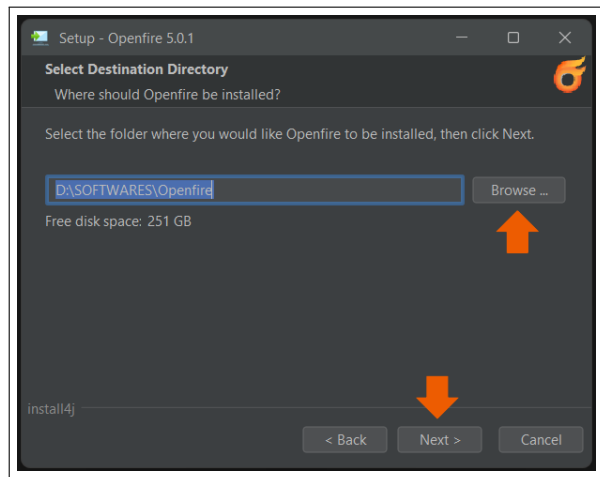


Figure 10: Click Next

**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.

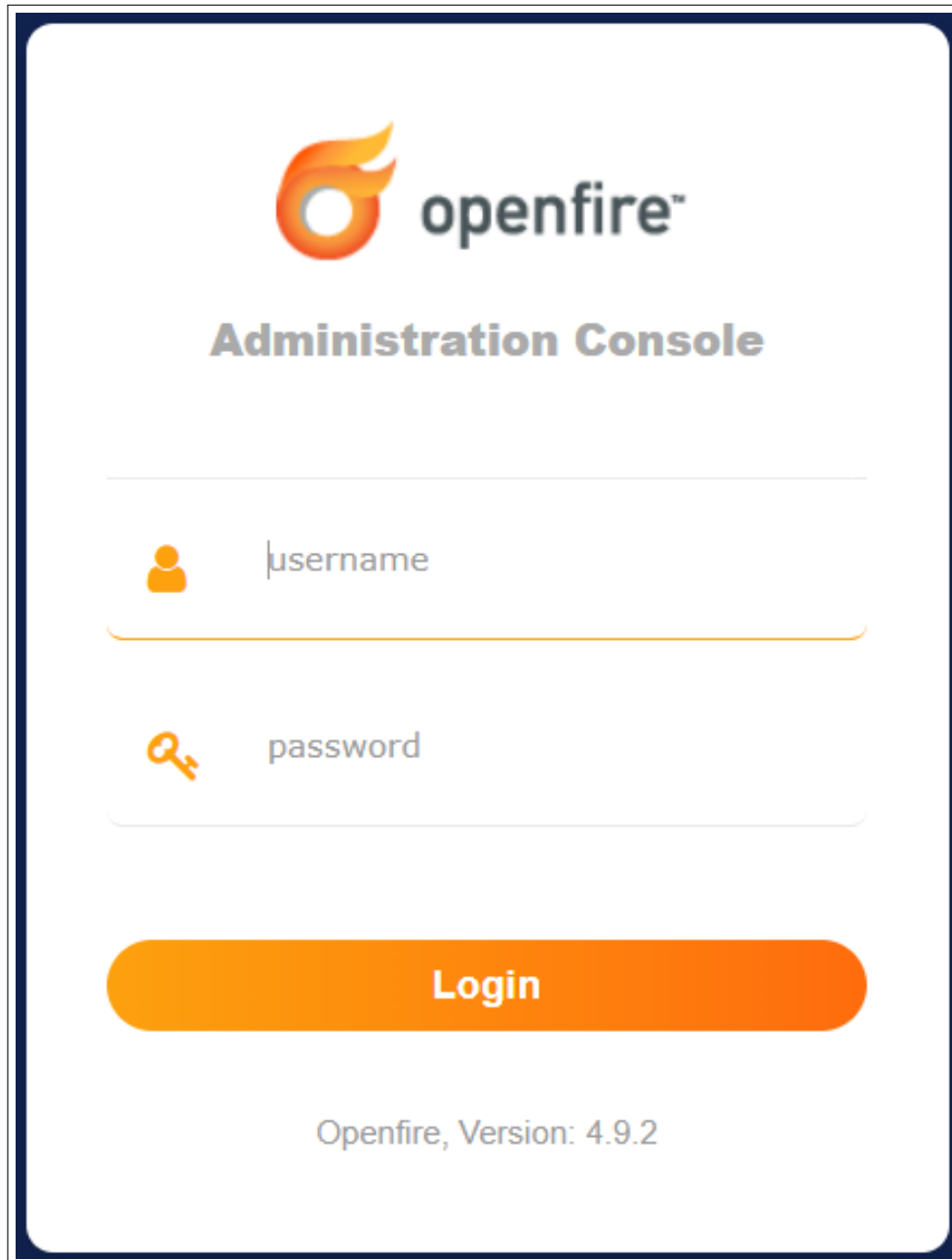


Figure 11: Login Panel

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).

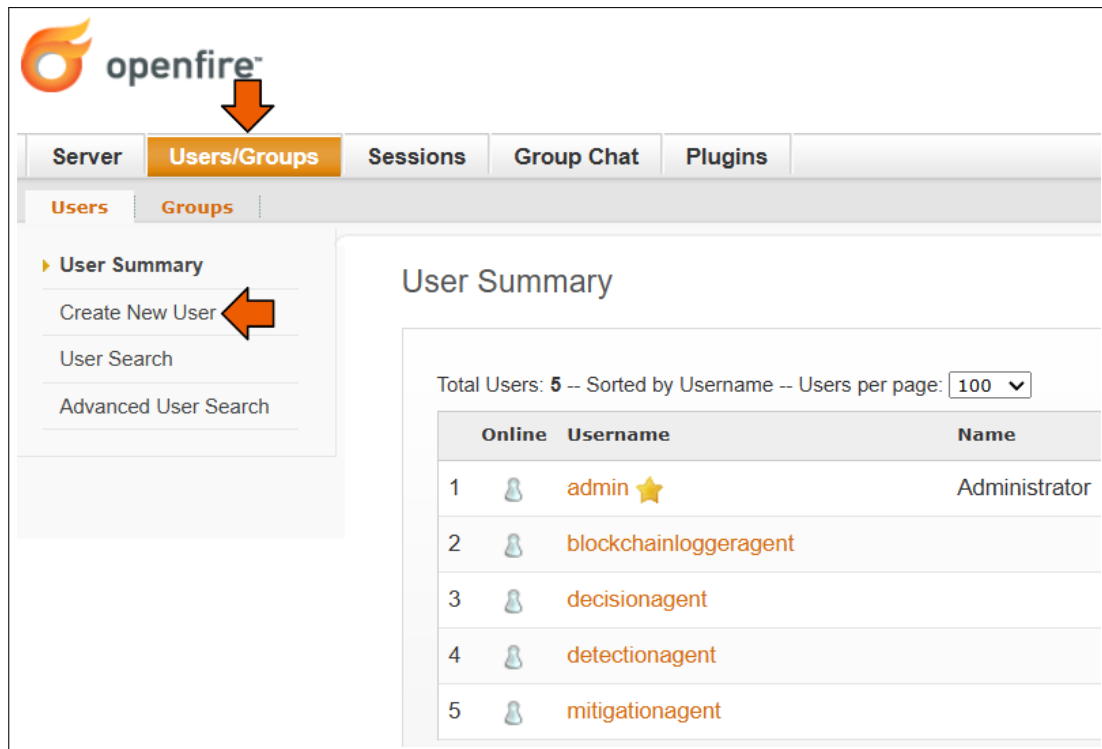


Figure 12: Create New Users

## 2.3 Verify XMPP Server Connectivity

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

```

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.

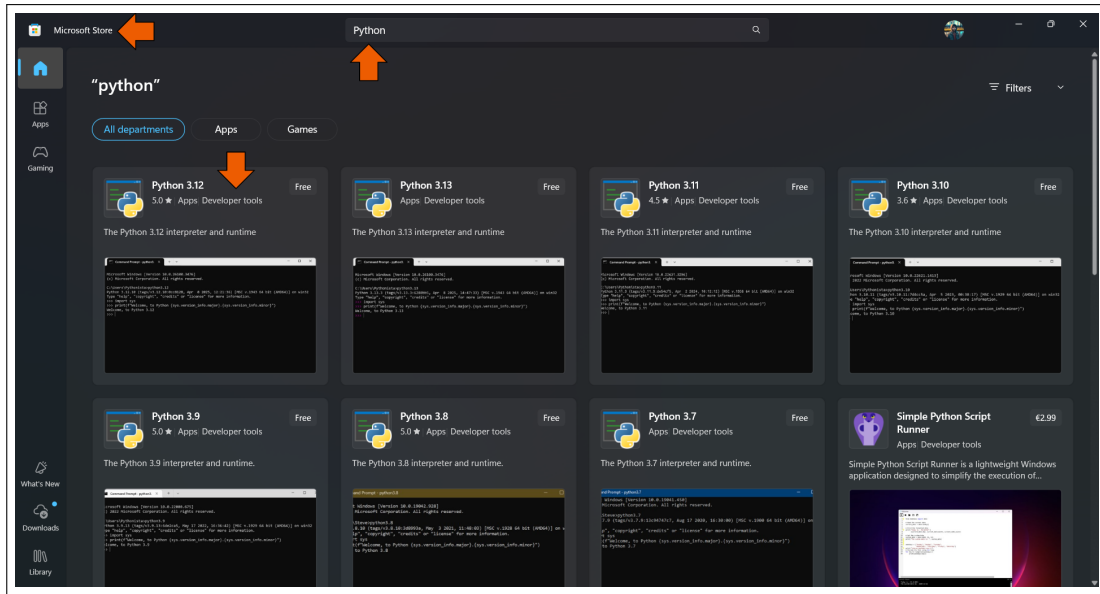


Figure 14: Search and Click Latest Python Version

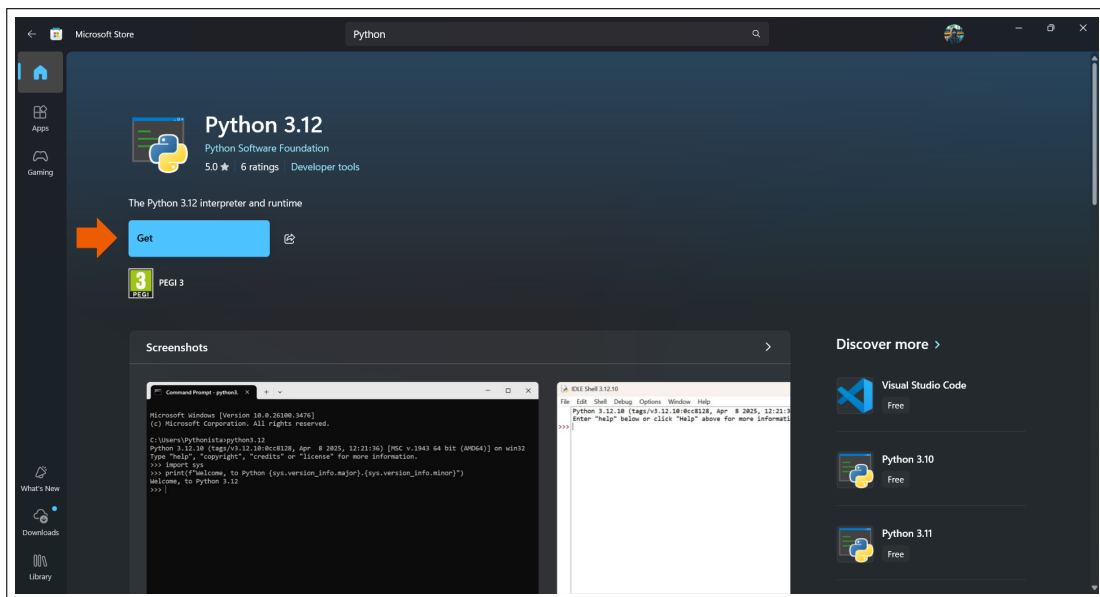
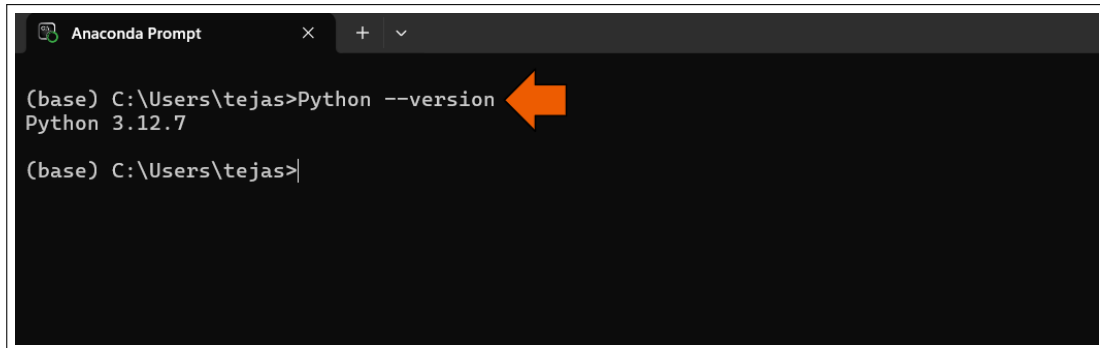


Figure 15: Click Get and Install

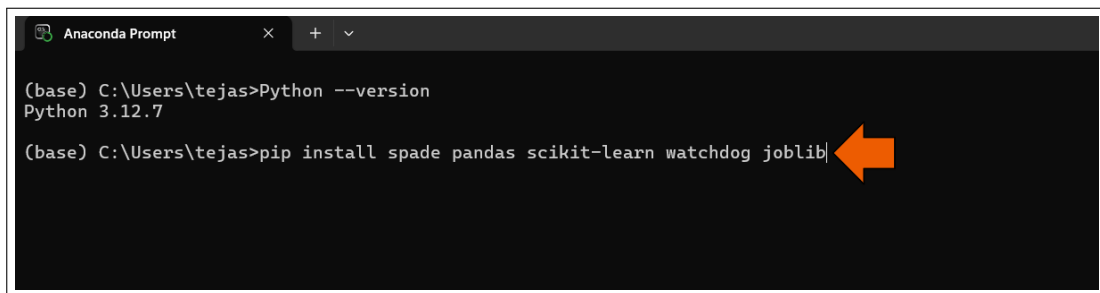
**Step 2** is to verify Python is installed or not using the command as displayed below.



```
Anaconda Prompt
(base) C:\Users\tejas>Python --version
Python 3.12.7
(base) C:\Users\tejas>
```

Figure 16: Verification Process

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



```
Anaconda Prompt
(base) C:\Users\tejas>Python --version
Python 3.12.7
(base) C:\Users\tejas>pip install spade pandas scikit-learn watchdog joblib
```

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.

```
1-Train.py X
Main Pipeline > 1-Train.py > ...
1 import pandas as pd
2 from sklearn.ensemble import RandomForestClassifier
3 from sklearn.model_selection import train_test_split
4 from sklearn.metrics import (
5     classification_report,
6     accuracy_score
7 )
8 import joblib
9
10 # Step 1: Load dataset
11 df = pd.read_csv("Dataset//Data.csv") # Adjust path as needed
12
13 # Step 2: Separate features and label
14 X = df.drop(columns=["Class"])
15 y = df["Class"]
16
17 # Step 3: Split into 80% train and 20% test
18 X_train, X_test, y_train, y_test = train_test_split(
19     X, y, test_size=0.2, random_state=50
20 )
21
22 # Step 4: Train Random Forest model
23 model = RandomForestClassifier(n_estimators=100, random_state=50)
24 model.fit(X_train, y_train)
25
26 # Step 5: Evaluate
27 y_pred = model.predict(X_test)
28 print("✅ Accuracy:", accuracy_score(y_test, y_pred))
29 print("\n📄 Classification Report:\n", classification_report(y_test, y_pred))
30
31 # Step 6: Save the trained model
32 joblib.dump(model, "Malware_Model.pkl")
33 print("📁 Model saved as Malware_Model.pkl")
34
35 # Step 7: Save full test set with labels
36 test_set = X_test.copy()
37 test_set["Class"] = y_test.values
38 test_set.to_csv("Test_Data_With_Labels.csv", index=False)
39 print("📄 Test data with labels saved to Test_Data_With_Labels.csv")
40
41 # Step 8: Save a single test row (without label) for agent testing
42 sample = X_test.iloc[[0]] # First row only, in DataFrame format
43 sample.to_csv("Sample_Input.csv", index=False)
44 print("📄 Sample input for DetectionAgent saved to Sample_Input.csv")
```

Figure 18: Train Model Using Random Forest Classifier



```
PS C:\Users\tejas\Desktop\SEM 3\RESEARCH PART 2 ASS DONE\Main Pipeline> & C:/Users/tejas/anaconda3/envs/Thesis-Project/python.exe
RESEARCH PART 2 ASS DONE/Main Pipeline/Monitor.py"
[Monitor] Watching C:/Users/tejas/Downloads for new .exe files...
[Monitor] New .exe file detected: C:/Users/tejas/Downloads/Test.exe
[MAS Trigger] Processing .exe file: C:/Users/tejas/Downloads/Test.exe
[MAS Trigger] Extracted features saved to Sample_Input.csv
Using slower stringprep, consider compiling the faster cython/libidn one.
[DetectionAgent] Starting...
[DecisionAgent] Starting...
[MitigationAgent] Starting...
[BlockchainLoggerAgent] Starting...
[Main] All agents started. Waiting for execution...
[DetectionAgent] Prediction - Infection Level 4. Sending to DecisionAgent...
[DecisionAgent] Classified as - Mild Infection. Sending to MitigationAgent...
[MitigationAgent] Initiating Mitigation... (e.g., file quarantine, network isolation)
[BlockchainLoggerAgent] Mild Infection detected and mitigated.
[Main] All agents stopped.
[Monitor] Ready for next .exe file...
```

Figure 20: Final Pipeline Test