

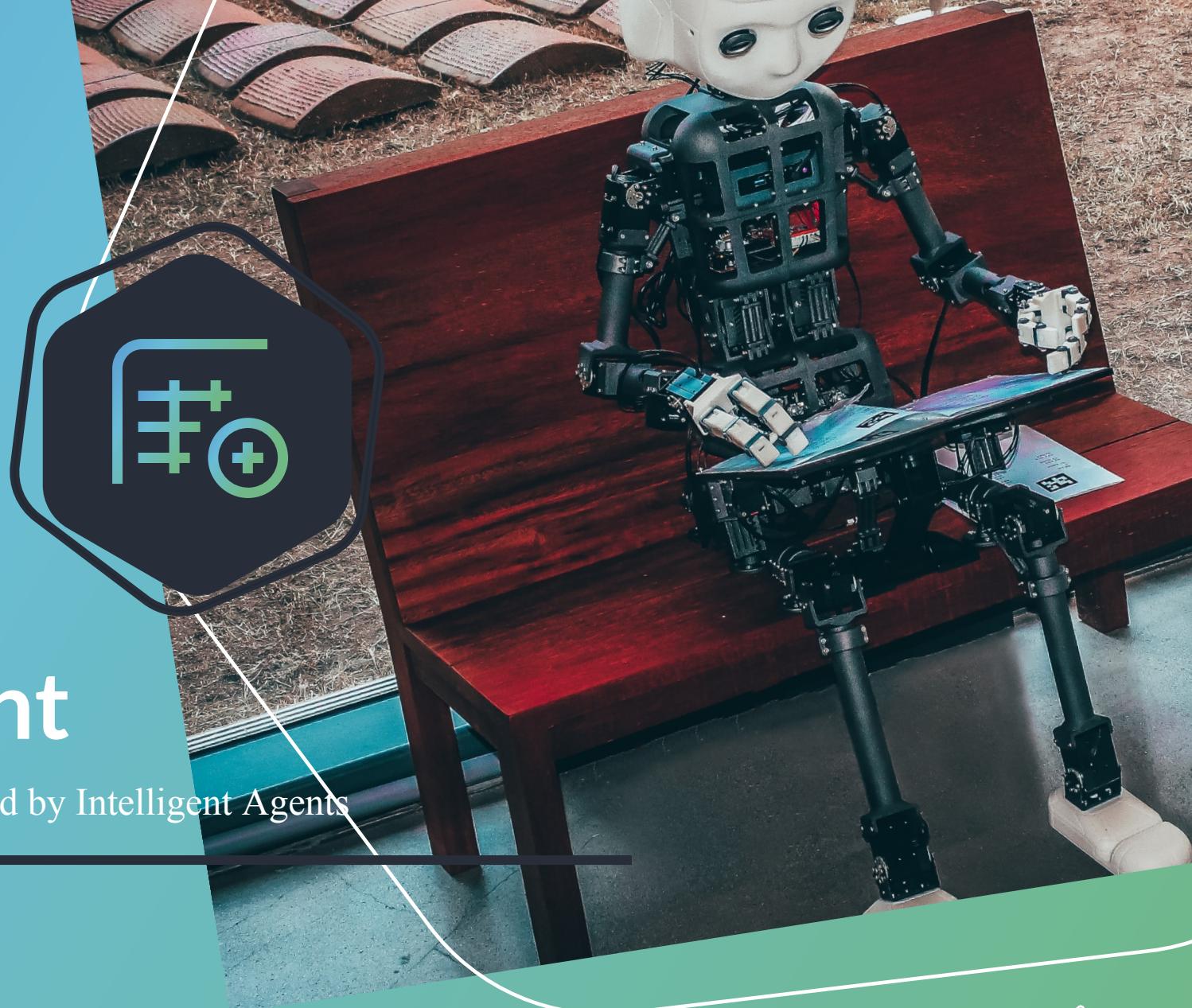
Intelligent Agent

File Segregation System Powered by Intelligent Agents

Challenges with digital data

File Classification - organisation and segregation

Automate using intelligent agents





Background

Intelligent Agents in AI

Architecture
Communication
Collaboration

Perceive
Reason
Extract knowledge
Inference
Derive conclusion
Autonomous decisions

Architecture: Reactive

Communication: protocols, languages and frameworks

Collaboration with users and other agents to achieve shared goals



Objectives



To develop intelligent agents - to analyse file name extensions for classification - > 10 different file types - accuracy rate > 95% - timeline 10 days.

To create a user-friendly interface - users to review and validate the classification results - provide feedback on any misclassifications - timeframe 10 days.

To conduct testing and validation processes - dataset of 1,000 files representing a diverse range of file types - To achieve an accuracy rate of >90% - ensure classification process < 5 seconds per file - timeline 5 days



Methodology

Multi Agent

Multi-agent system for file classification



Collaboration

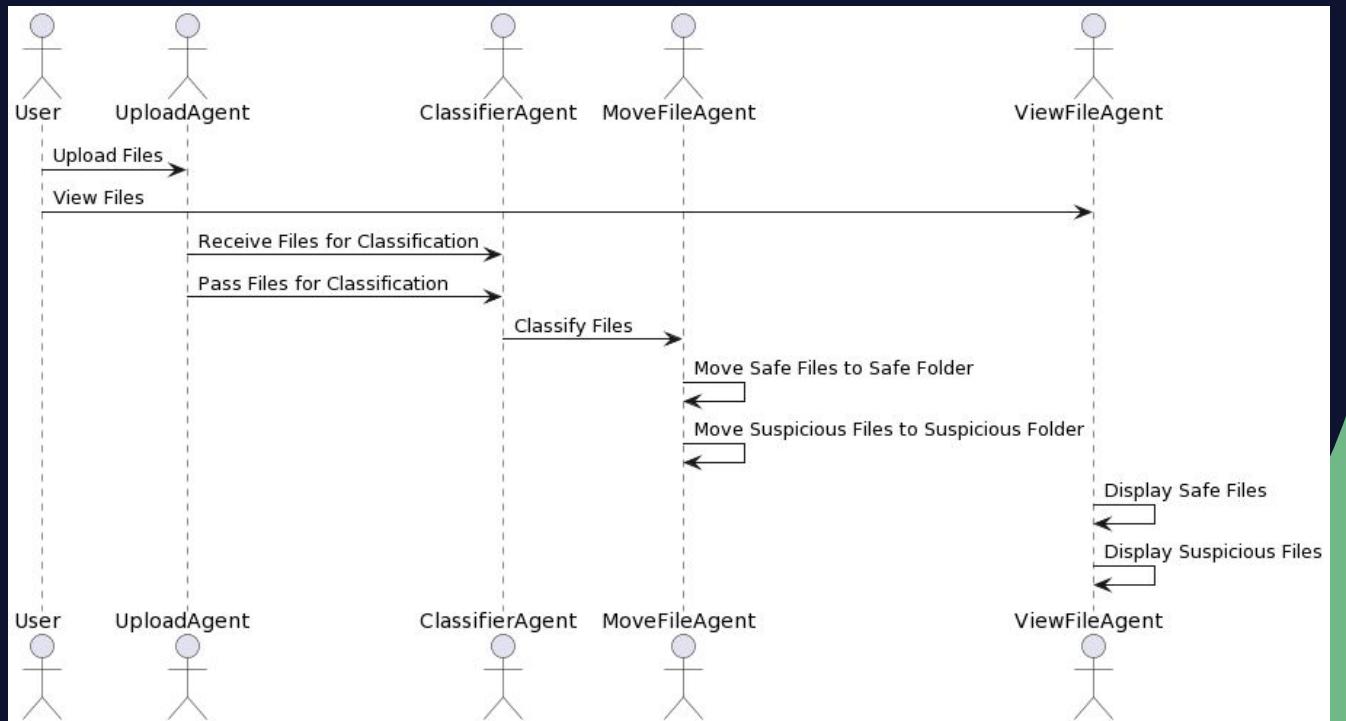
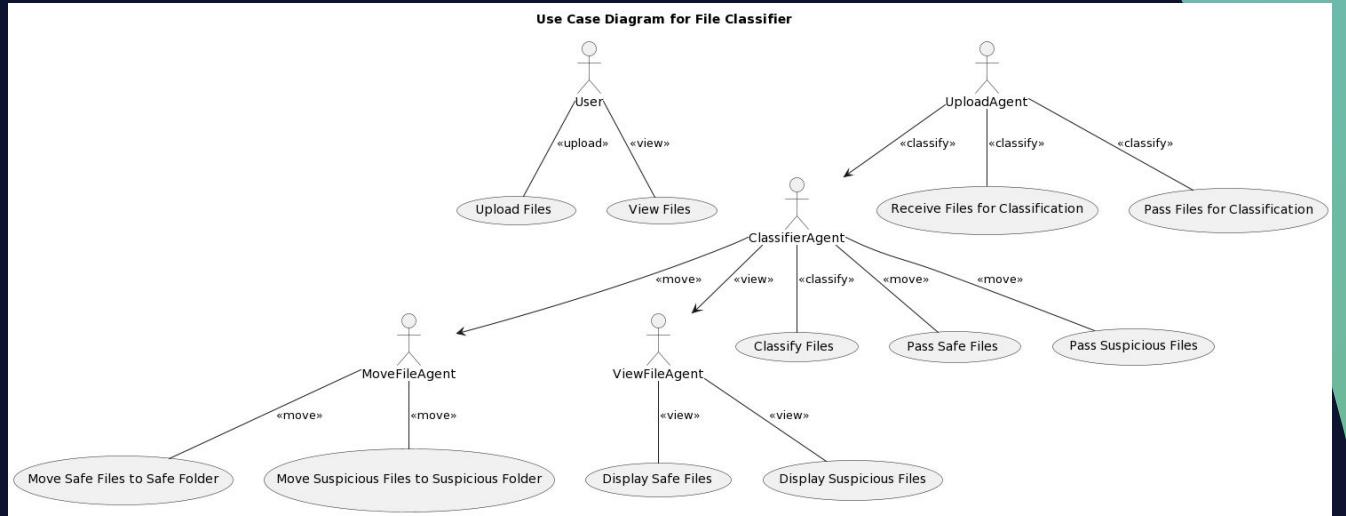
Collaboration and interaction among agents



Reactive Agent

Use of reactive agents for fast and efficient response





Project Design

Use Case UML

Use case diagrams serve to illustrate the broad functionalities and scope of a system.

Sequence UML

Sequence diagrams employed to highlight specific interaction patterns within the context of the use case.



Developments



Upload Agent

Secure file uploads

```
@app.route('/')
def upload_file():
    return render_template('upload.html')

@app.route('/upload', methods = ['GET', 'POST'])
def upload_files():
    if request.method == 'POST':
        files = request.files.getlist('files[]')
        for file in files:
            UploadAgent(file)
        return redirect(url_for('view_files'))
    else:
        return 'No file uploaded!'
```

```
def UploadAgent(file):
    filename = secure_filename(file.filename)
    if not os.path.exists('uploads'):
        os.makedirs('uploads')
    file.save(os.path.join('uploads', filename))
    ClassifierAgent(filename)
```



Classifier Agent

Classify files based on extensions

```
def ClassifierAgent(filename):
    # Checking if file extension is in the list of suspicious file types
    if any(filename.endswith(ft) for ft in suspicious_file_types):
        MoveFileAgent(filename, 'suspicious')
    else:
        MoveFileAgent(filename, 'safe')
```



Move File Agent

Move files to designated folders

```
def MoveFileAgent(filename, classification):
    if not os.path.exists(classification):
        os.makedirs(classification)
    shutil.move(os.path.join('uploads', filename), os.path.join(classification, filename))
```



View File Agent

Display files to the user

```
@app.route('/view', methods = ['GET'])
def view_files():
    safe_files = os.listdir('safe') if os.path.exists('safe') else []
    suspicious_files = os.listdir('suspicious') if os.path.exists('suspicious') else []
    return render_template('view.html', safe_files=safe_files, suspicious_files=suspicious_files)
```



Testing and Evaluation

- **Objective:** Ensuring functionality, identifying issues, evaluating performance, user satisfaction.
- **Test Types:**
 - **Unit:** Checks functions.
 - **Integration:** Reveals interaction faults.
 - **System:** Confirms compatibility.
 - **Performance:** Analyzes behavior under load.
 - **UAT:** User satisfaction confirmation.
- **Tools:**
 - Python's unittest for test cases.
 - pytest-cov and coverage to measure test coverage

```
chuehue@Chues-MBP FileClassifier % pytest --cov=app
=====
test session starts =====
platform darwin -- Python 3.11.4, pytest-7.4.0, pluggy-1.2.0
rootdir: /Users/chuehue/Desktop/MSC_AI/IntelligentAgent/Assignments/FileClassifier
plugins: cov-4.1.0
collected 7 items

app_test.py ......

[100%]

----- coverage: platform darwin, python 3.11.4-final-0 -----
Name      Stmts   Miss  Cover
app.py       38      4    89%
TOTAL        38      4    89%
=====
7 passed in 0.13s =====
```

Figure 7.1 Unit Test Result

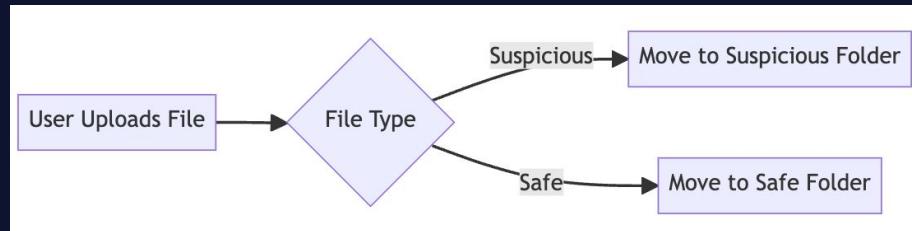


Figure 7.2 System Process

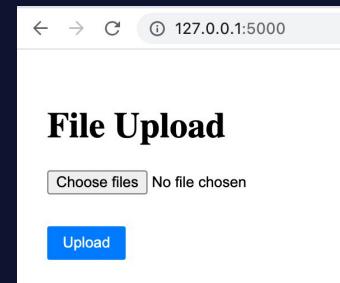


Figure 7.3 File Upload UI

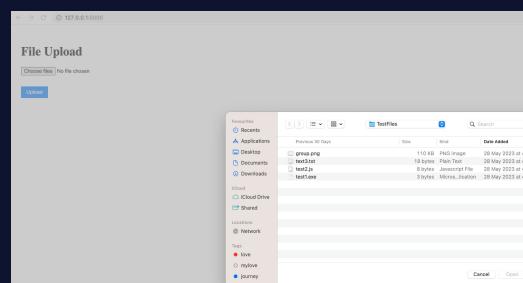


Figure 7.4 File Browser UI

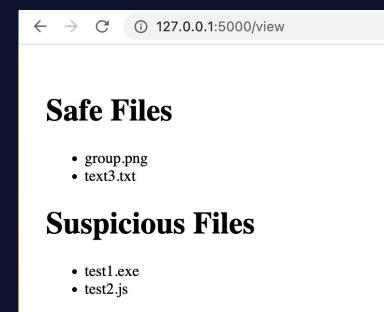


Figure 7.5 Result of Classified Files UI



Programming Approach

```

# app.py
1  from flask import Flask, render_template, request, redirect, url_for
2  from werkzeug.utils import secure_filename
3  import os
4  import shutil
5
6  app = Flask(__name__)
7
8  # The list of file extensions is chosen based on common malicious file types.
9  # This list can be modified to fit the specific security needs of the application.
10 suspicious_file_types = ['.exe', '.com', '.bat', '.cmd', '.scr', '.msi', '.dizf', '.gadget',
11  '.js', '.jse', '.vb', '.vbe', '.ws', '.wst', '.wsc', '.wsh', '.psl', '.psxml', '.ps2', '.ps2xml',
12  '.psx', '.ps1', '.ps2', '.ashx', '.ashx2', '.ashx3', '.ashx4', '.ashx5', '.ashx6', '.ashx7',
13  '.ashx8', '.ashx9', '.ashx10', '.ashx11', '.ashx12', '.ashx13', '.ashx14', '.ashx15',
14  '.ashx16', '.ashx17', '.ashx18', '.ashx19', '.ashx20', '.ashx21', '.ashx22', '.ashx23', '.ashx24',
15  '.ashx25', '.ashx26', '.ashx27', '.ashx28', '.ashx29', '.ashx30', '.ashx31', '.ashx32', '.ashx33',
16  '.ashx34', '.ashx35', '.ashx36', '.ashx37', '.ashx38', '.ashx39', '.ashx40', '.ashx41', '.ashx42',
17  '.ashx43', '.ashx44', '.ashx45', '.ashx46', '.ashx47', '.ashx48', '.ashx49', '.ashx50', '.ashx51',
18  '.ashx52', '.ashx53', '.ashx54', '.ashx55', '.ashx56', '.ashx57', '.ashx58', '.ashx59', '.ashx60',
19  '.ashx61', '.ashx62', '.ashx63', '.ashx64', '.ashx65', '.ashx66', '.ashx67', '.ashx68', '.ashx69',
20  '.ashx70', '.ashx71', '.ashx72', '.ashx73', '.ashx74', '.ashx75', '.ashx76', '.ashx77', '.ashx78',
21  '.ashx79', '.ashx80', '.ashx81', '.ashx82', '.ashx83', '.ashx84', '.ashx85', '.ashx86', '.ashx87',
22  '.ashx88', '.ashx89', '.ashx90', '.ashx91', '.ashx92', '.ashx93', '.ashx94', '.ashx95', '.ashx96',
23  '.ashx97', '.ashx98', '.ashx99', '.ashx100', '.ashx101', '.ashx102', '.ashx103', '.ashx104', '.ashx105',
24  '.ashx106', '.ashx107', '.ashx108', '.ashx109', '.ashx110', '.ashx111', '.ashx112', '.ashx113', '.ashx114',
25  '.ashx115', '.ashx116', '.ashx117', '.ashx118', '.ashx119', '.ashx120', '.ashx121', '.ashx122', '.ashx123',
26  '.ashx124', '.ashx125', '.ashx126', '.ashx127', '.ashx128', '.ashx129', '.ashx130', '.ashx131', '.ashx132',
27  '.ashx133', '.ashx134', '.ashx135', '.ashx136', '.ashx137', '.ashx138', '.ashx139', '.ashx140', '.ashx141',
28  '.ashx142', '.ashx143', '.ashx144', '.ashx145', '.ashx146', '.ashx147', '.ashx148', '.ashx149', '.ashx150',
29  '.ashx151', '.ashx152', '.ashx153', '.ashx154', '.ashx155', '.ashx156', '.ashx157', '.ashx158', '.ashx159',
30  '.ashx150', '.ashx151', '.ashx152', '.ashx153', '.ashx154', '.ashx155', '.ashx156', '.ashx157', '.ashx158',
31  '.ashx159', '.ashx160', '.ashx161', '.ashx162', '.ashx163', '.ashx164', '.ashx165', '.ashx166', '.ashx167',
32  '.ashx168', '.ashx169', '.ashx170', '.ashx171', '.ashx172', '.ashx173', '.ashx174', '.ashx175', '.ashx176',
33  '.ashx177', '.ashx178', '.ashx179', '.ashx180', '.ashx181', '.ashx182', '.ashx183', '.ashx184', '.ashx185',
34  '.ashx186', '.ashx187', '.ashx188', '.ashx189', '.ashx190', '.ashx191', '.ashx192', '.ashx193', '.ashx194',
35  '.ashx195', '.ashx196', '.ashx197', '.ashx198', '.ashx199', '.ashx200', '.ashx201', '.ashx202', '.ashx203',
36  '.ashx204', '.ashx205', '.ashx206', '.ashx207', '.ashx208', '.ashx209', '.ashx210', '.ashx211', '.ashx212',
37  '.ashx213', '.ashx214', '.ashx215', '.ashx216', '.ashx217', '.ashx218', '.ashx219', '.ashx220', '.ashx221',
38  '.ashx222', '.ashx223', '.ashx224', '.ashx225', '.ashx226', '.ashx227', '.ashx228', '.ashx229', '.ashx230',
39  '.ashx231', '.ashx232', '.ashx233', '.ashx234', '.ashx235', '.ashx236', '.ashx237', '.ashx238', '.ashx239',
40  '.ashx240', '.ashx241', '.ashx242', '.ashx243', '.ashx244', '.ashx245', '.ashx246', '.ashx247', '.ashx248',
41  '.ashx249', '.ashx250', '.ashx251', '.ashx252', '.ashx253', '.ashx254', '.ashx255', '.ashx256', '.ashx257',
42  '.ashx258', '.ashx259', '.ashx260', '.ashx261', '.ashx262', '.ashx263', '.ashx264', '.ashx265', '.ashx266',
43  '.ashx267', '.ashx268', '.ashx269', '.ashx270', '.ashx271', '.ashx272', '.ashx273', '.ashx274', '.ashx275',
44  '.ashx276', '.ashx277', '.ashx278', '.ashx279', '.ashx280', '.ashx281', '.ashx282', '.ashx283', '.ashx284',
45  '.ashx285', '.ashx286', '.ashx287', '.ashx288', '.ashx289', '.ashx290', '.ashx291', '.ashx292', '.ashx293',
46  '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292',
47  '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291',
48  '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290',
49  '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299',
50  '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298',
51  '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297',
52  '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296',
53  '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295',
54  '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294',
55  '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293',
56  '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292',
57  '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291',
58  '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290',
59  '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299',
60  '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298',
61  '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297',
62  '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296',
63  '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295',
64  '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294',
65  '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293',
66  '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292',
67  '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291',
68  '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290',
69  '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298', '.ashx299',
70  '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297', '.ashx298',
71  '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296', '.ashx297',
72  '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295', '.ashx296',
73  '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294', '.ashx295',
74  '.ashx296', '.ashx297', '.ashx298', '.ashx299', '.ashx290', '.ashx291', '.ashx292', '.ashx293', '.ashx294',
75
    
```

app.py

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Upload File</title>
5      <style>
6          /* Styling for the body of the webpage, padding is added for better visual appearance */
7          body {
8              padding: 20px;
9          }
10         /* Styling for the file input and submit button, height and margin are adjusted for better visual appearance */
11         input[type=file], input[type=submit] {
12             height: 30px;
13             margin-bottom: 20px;
14         }
15         /* Styling for the submit button, width, background color, text color, border style, and border radius are adjusted for better visual appearance and user experience */
16         input[type=submit] {
17             width: 70px;
18             background-color: #007bff;
19             color: white;
20             border: none;
21             border-radius: 2px;
22         }
23         /* styling for the text/javascript */
24         script type="text/javascript">
25             // Function to check if a file is selected when the user selects a file
26             function checkFile() {
27                 // Get the file input element
28                 var fileInput = document.getElementById('fileInput');
29
30                 // Check if a file has been selected
31                 if (fileInput.files.length === 0) {
32                     // If no file is selected, the upload button is disabled to prevent empty submissions
33                     document.getElementById('uploadButton').disabled = true;
34                 } else {
35                     // If a file is selected, the upload button is enabled allowing the user to submit the form
36                     document.getElementById('uploadButton').disabled = false;
37                 }
38             }
39         </script>
40     </head>
41     <body>
42         <div class="container">
43             <h3>Upload</h3>
44             <!-- Form for file upload, it sends a POST request to the /upload route on submission -->
45             <form method="POST" enctype="multipart/form-data">
46                 <!-- File input field, it allows multiple file selection and triggers the checkFile function when a file is selected or deselected -->
47                 <input type="file" id="fileInput" name="files" multiple="" onchange="checkFile()">
48                 <br>
49                 <!-- Submit button for the form, it is disabled by default and will be enabled once a file is selected -->
50                 <input type="submit" id="uploadButton" value="Upload" disabled="disabled" />
51             </form>
52         </div>
53     </body>
54 </html>
55 
```

upload.html

view.html

```

templates > view.html ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <!-- Styling for the body of the webpage, padding is added for better visual appearance -->
5      <style>
6          body {
7              padding: 20px;
8          }
9      </style>
10 </head>
11 <body>
12     <!-- Container for the file lists -->
13     <div class="container">
14         <!-- Heading for the list of safe files -->
15         <h3>Safe Files</h3>
16         <!-- Unordered list for displaying the safe files -->
17         <ul>
18             <!-- Loop through each file in the safe_files list passed from the server -->
19             {% for file in safe_files %}
20                 <li>{{ file }}</li>
21             {% endfor %}
22         </ul>
23         <!-- Heading for the list of suspicious files -->
24         <h3>Suspicious Files</h3>
25         <!-- Unordered list for displaying the suspicious files -->
26         <ul>
27             <!-- Loop through each file in the suspicious_files list passed from the server -->
28             {% for file in suspicious_files %}
29                 <li>{{ file }}</li>
30             {% endfor %}
31         </ul>
32     </div>
33 </body>
34 </html>
35 
```

app.py

- Programming approach: Python programming
- Flask: Lightweight framework for building web interfaces
- Werkzeug: Ensures security in file handling and communication
- os: Handles operating system-dependent functionalities
- shutil: Performs high-level file operations and management

upload.html

- HTML template for the file upload page.
- Allows users to select and upload files.
- Includes JavaScript code for handling file selection and enabling/disabling the upload button.

view.html

- HTML template for the file viewing page.
- Displays the list of files categorized as "safe" and "suspicious".
- Utilizes Flask template syntax to dynamically populate the file list.



Challenges and Solution

Guaranteeing accurate file classification and securing file uploads are paramount challenges in our development process

Accuracy

Accurate file classification

Security

Secure file upload to prevent attacks

Validation

Testing and validation





Developed multi-agent system for file
classification and segregation - Created
user interface - Testing and validation

Limitations – filename extension basis only

Further study – classification based on content
and user defined attributes – use machine
learning and natural language processing to
enhance performance

Thanks

