

Introduction

Computerised devices are being used in every part of our lives. With the exponential growth of digital data, businesses and individuals face the challenge of organising and categorizing vast amounts of files. Intelligent agents can process and analyse large volumes of data much faster and more efficiently than humans (Russell, 2021). Also our digital assets are vulnerable to attack from malicious files. There are several file types that can be dangerous to our computer system (Hoffman, 2017) which may be difficult for an average computer user to differentiate between ordinary and malicious files. The aim of our project is to deploy intelligent agents to automate the process of classifying files. This project design proposal provides an outline plan for designing and developing intelligent agents to classify files based on the file name extension.

Objectives

The main objectives of this project are:

* To design and develop intelligent agents to be able to analyse the file name extension for classification with various file types such as documents, images, audio and video files (Examples: JPEG, PNG, PDF, DOCX, XLXS, MP3, WAV, etc.
* To develop an interface for users to interact with the agent to review and validate the classification results and provide feedback.
* To perform testing and validation to ensure accuracy and efficiency of the intelligent agents

Methodology

Requirement Analysis

Defining file types: Define the file types and extensions to be classified

Data preparation: Identifying features ( file name extension)

Choice of Python libraries: The following Python libraries will be used in this project.

Flask – useful for building web interface for users to interact with the system

werkzeug – useful for ensuring security in file handling, providing secure file names

os – useful for various operating system-dependent functionalities such as managing file and directory operations

shutil – useful for performing high level operation on files and in automated file management

Project Design and Architecture

Reactive agent

This project would deploy multi-agent system as it could be a powerful approach for file classification. Peng et al. (2001) had compared single and multi agent performance based on criteria such as response time, quality of classification and economic/privacy considerations for the classification of computer science and Medline documents. Their results indicate that collaborative multi-agent system performs better than single agent system. Multi-agent systems involve the coordination and interaction of multiple agents to achieve collective goals. These systems address complex problems that require cooperation, communication, and negotiation among agents. Techniques for multi-agent systems include protocols for communication, coordination algorithms, and game theory.

File Upload Agent: This agent is responsible for accepting file uploads from the user through the web interface. It leverages Flask's capabilities to handle file uploads securely.

File Classification Agent: This agent's function is to classify the uploaded files based on their extensions. It categorizes files as either 'safe' or 'suspicious' based on a predetermined list of file extensions.

File Movement Agent: Post-classification, this agent moves files to their designated folders ('Safe' or 'Suspicious'), and within these, to specific sub-folders based on file type. It uses the Python 'os' and 'shutil' libraries for these operations.

Two different graphical designs – UML model

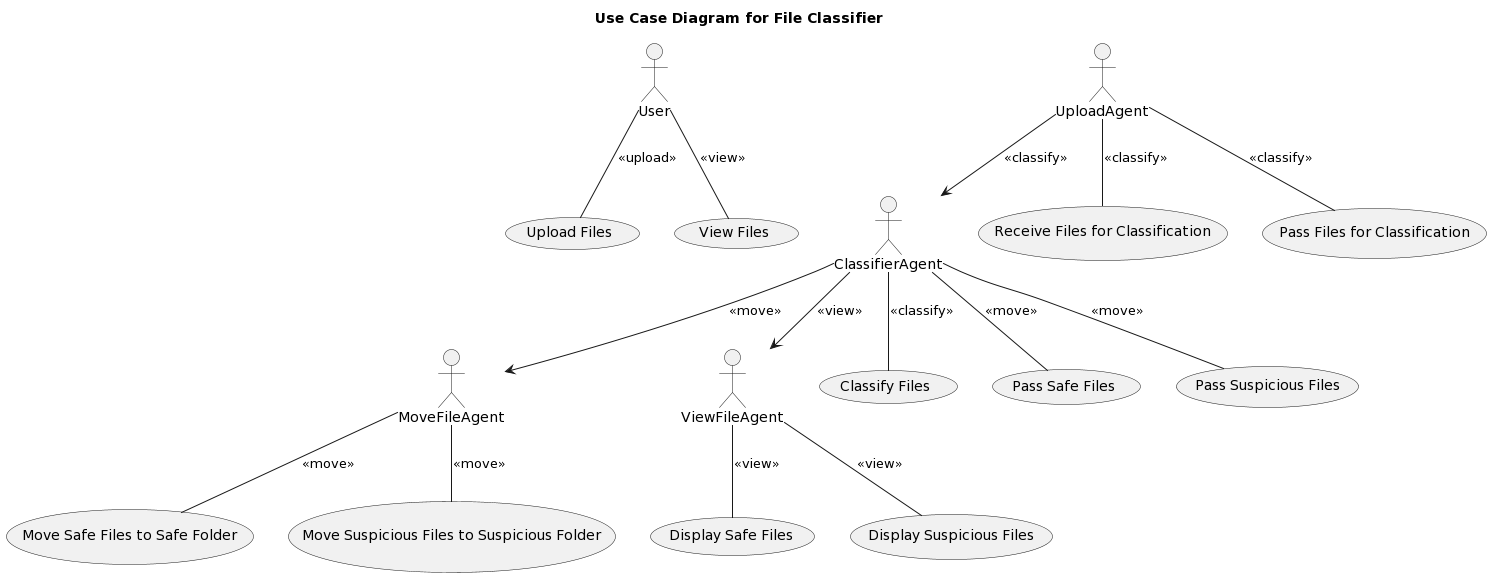


Figure 1. Use Case UML

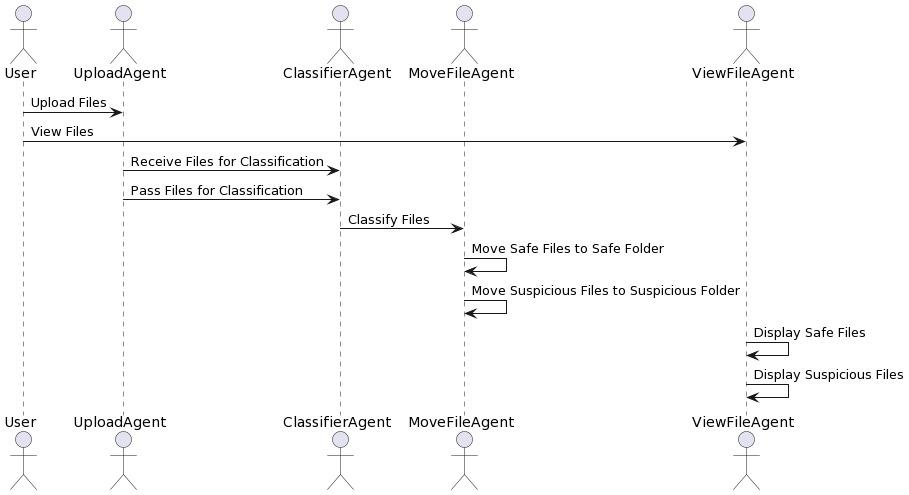


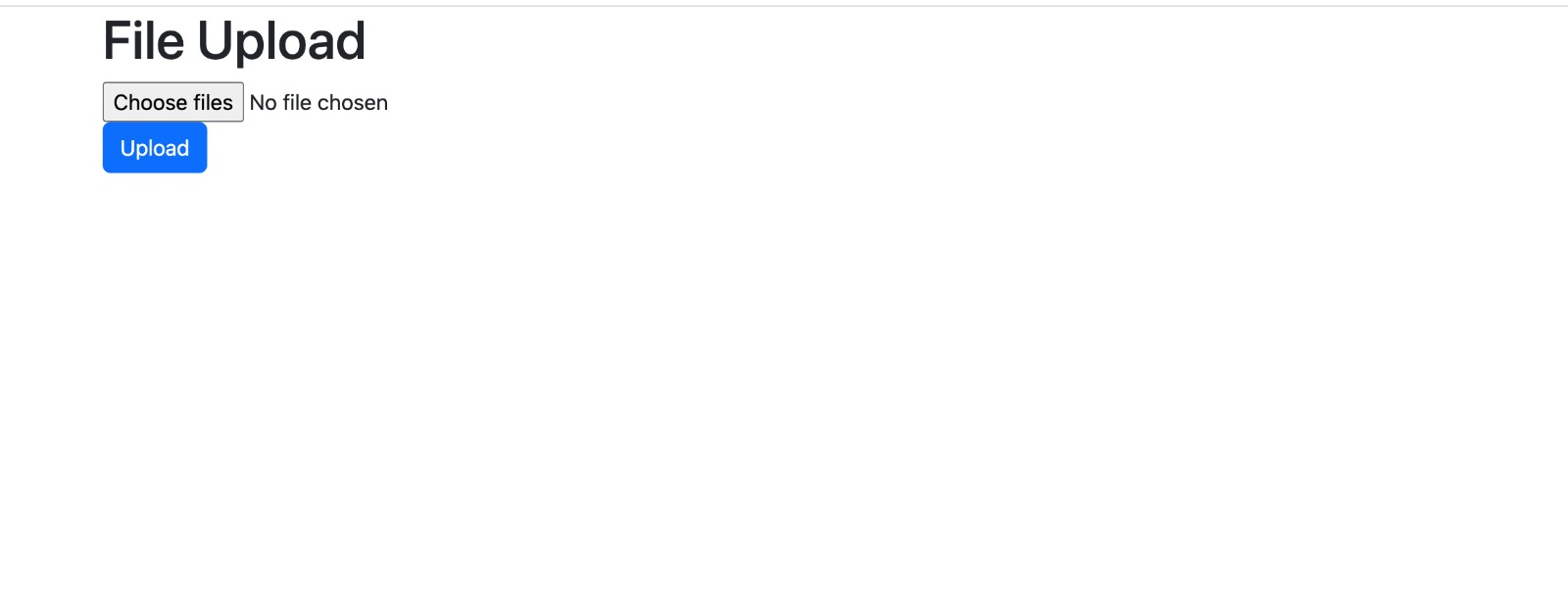
Figure 2. Sequence UML

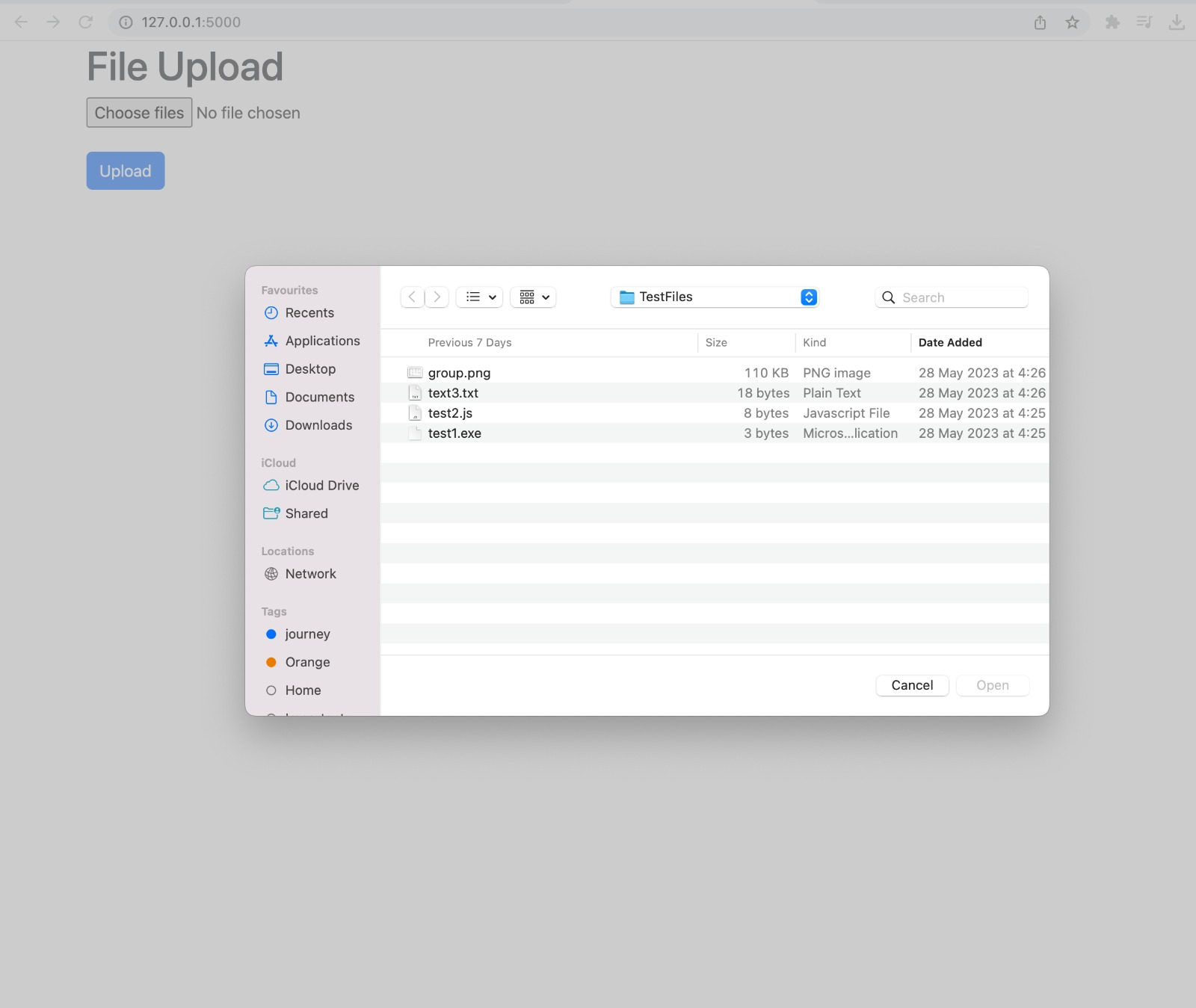
Development and Implementation

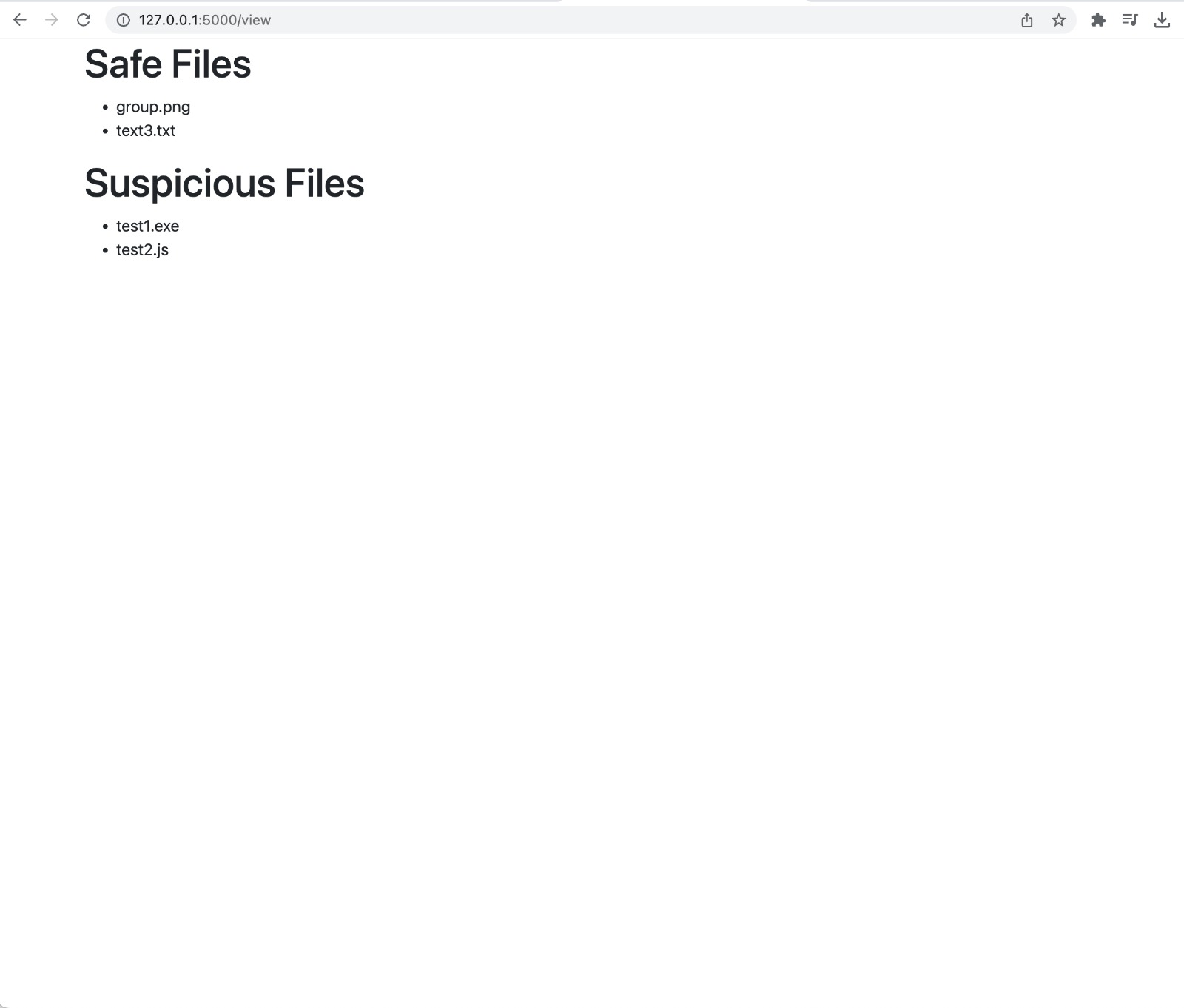
Testing and Evaluation

Deployment: interface or API that allows users to input files, and the intelligent agent will provide predictions or classifications results

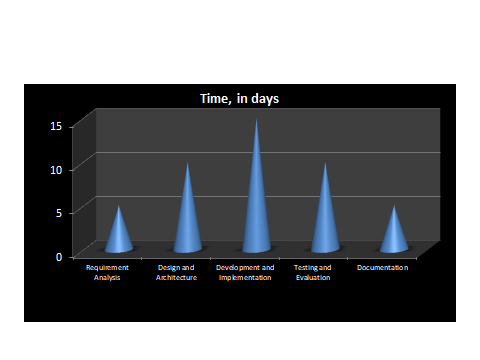
Add screenshot of interface







Timeline



Resources

3-member student team

Python libraries

Datasets for training and testing

Challenges encountered: (any issues around identifying and retrieving the data? Highlight (briefly) what paradigm(s), pattern(s), theories and practices you intend to utilise on this project to address the challenges. Justify your approaches supported by literature.)

Database for training and testing

A significant challenge in the development process would be to ensure the accurate classification of files. The current approach uses file extensions, which can be misleading if a malicious file is disguised with a benign extension (Casey, 2011). Further, ensuring secure file upload to prevent directory traversal attacks is a critical concern (Stuttard & Pinto, 2011).

Expected Outcomes

A functional intelligent agent capable of classifying files accurately based on their file name extension.

Improved data organisation and retrieval efficiency for users, reducing manual efforts in file management.

Conclusion

This project proposal outlines the plan for designing and developing an intelligent agent focused on file classification. By implementing machine learning techniques, the agent aims to automate the organisation and categorization of files, enhancing data organization and retrieval efficiency for users. Successful implementation of this project will contribute to streamlined file management and improved productivity in various domains.

References

Russell, S., Norvig, P., Canny, J., Malik, J. (2021). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.

Hoffman., C (2017) 50+ File Extensions that are potentially dangerous on Windows. Available from: <https://www.howtogeek.com/137270/50-file-extensions-that-are-potentially-dangerous-on-windows/>

S. Peng, S. Mukhopadhyay, R. Raje, and M. Palakal. 2001. A Comparison Between Single-agent and Multi-agent Classification of Documents. In Proceedings of the 10th Heterogeneous Computing Workshop â"" HCW 2001 (Workshop 1) - Volume 2 (IPDPS '01). IEEE Computer Society, USA, 20090.2.

Casey, E. (2011). Digital Evidence and Computer Crime. Academic Press. ISBN 978-0-12-374268-1.

Stuttard, D., & Pinto, M. (2011). The Web Application Hacker's Handbook. Wiley. ISBN 978-1-119-96327-1.