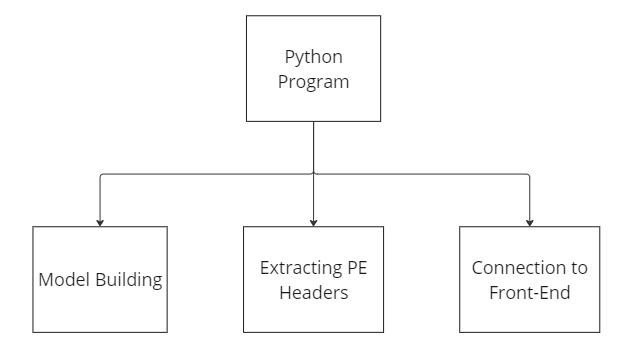
# Vasu Bansal

# 20BCE1072

# CSE4003

# Digital Assignment-2

# Proposed Methodology and Module Description:



The given diagram shows the modules that are a part of the main back-end application. It consists of the following main parts:

1. Model Building: This part involves training a machine learning model using PEFile features to detect malware. PEFile features can include characteristics such as file size, imports and exports, section details, resource information, and more. The model is built using a labeled dataset that contains both malware and benign PE files. The model learns from this dataset to classify new PE files as either malware or benign based on their extracted features.
2. Extracting PE Headers: This part involves extracting the headers of PE files, which contain information about the structure and attributes of the file. PE headers typically include details such as file type, entry point, section information, import and export tables, and more. Extracting these headers allows for the extraction of relevant features that can be used as input to the machine learning model for detecting malware.
3. Connection to Front-End: This part involves creating a connection between the back-end machine learning model and a front-end interface, such as a web application or a graphical user interface (GUI). The front-end allows users to interact with the model, providing input files for analysis, and receiving the output of the model's malware detection results. The connection may involve APIs or other communication mechanisms to pass files and results between the front-end and back-end components of the system

Diagram

Description automatically generated

The flow of the code goes something like this - The python code is kept running live using ‘uvicorn’, which is an ASGI (Asynchronous Server Gateway Interface) server implementation for Python. The user accesses the home page and uploads the file they feel requires some checking. The file is passed onto the backend and Python functions in the backend predict whether the given file contains a virus or not. If the uploaded file has a risk of having a virus, the user is suggested some remedies on how to check if their PC has been damaged. If not, they get a green light on opening and using the file.