**CHAPTER 1**

**PROJECT PROFILE**

|  |  |
| --- | --- |
| **Project Title :-** | Ransomware Attack Detection Tool |
| **Objective :-** | The objective of this project is to develop a ransomware detection tool using machine learning to identify potential ransomware attacks in real-time and enhance cybersecurity. |
| **Platform :-** | Window OS |
| **Tools & Technologies :-** | Flask , Python , HTML , CSS , JavaScript , Machine Learning |
| **Developed by :-** | 1. Kaushal Patel ( 22082291024 ) 2. Dhruvil Raval ( 22082291032 ) |
| **Group No :-** | 22 |
| **Internal Guide :-** | Prof. Riya Patel |
| **Duration :-** | 90 days |
| **Submitted to :-** | Department of Computer Science, Ganpat  University, Kherva |

**Introduction**

* 1. **PROJECT PLAN**

## ABOUT THE PROJECT

This project involves developing a ransomware detection tool designed to identify potential ransomware attacks using machine learning. The tool was built by training a model with the XGBoost classifier, a powerful algorithm well-suited for classification tasks, and integrating it into a user-friendly web interface using the Flask framework.

The process began by gathering and preparing the dataset, which was loaded and preprocessed using pandas. The dataset was split into training and testing sets to ensure the model's effectiveness in predicting unseen data. The features used for training were carefully selected, excluding the target label, which represents whether a ransomware attack is present.

To achieve accurate predictions, the XGBoost classifier was employed. The model was trained on the prepared dataset, and its performance was evaluated using cross-validation techniques, which help in assessing the model's robustness and reliability. Several metrics, including accuracy, precision, recall, and F1 score, were computed to gauge the model’s performance.

The trained model was then saved using joblib for later use in the web application. The Flask-based web interface was developed to allow users to interact with the model easily. The application includes routes for login, logout, prediction, and a dashboard for visualizing results. Upon receiving user inputs through the interface, the model predicts whether the input data indicates a ransomware attack.

Additional analysis was conducted to assess feature importance and model performance, with visualizations created to illustrate the results. These include feature importance plots, confusion matrices, and model performance comparisons, providing insights into the tool’s effectiveness and the significance of various input features.

This project showcases the integration of machine learning with web technologies, resulting in a practical tool for detecting ransomware, with the potential for real-world application in cybersecurity.

## PURPOSE AND SCOPE

The primary purpose of this project is to develop a ransomware detection tool capable of identifying potential ransomware attacks in real-time, thereby enhancing cybersecurity measures. Ransomware attacks are increasingly prevalent and pose significant risks to individuals, businesses, and government institutions by encrypting valuable data and demanding ransom for its release. This tool aims to preemptively detect such malicious activities, allowing for prompt mitigation and protection of critical data.

The tool employs a machine learning model, specifically an XGBoost classifier, trained on a dataset of known ransomware behaviors. The model analyzes various features extracted from system activities to predict the likelihood of a ransomware attack. By leveraging the power of XGBoost, which is renowned for its performance and efficiency in classification tasks, the model can accurately differentiate between normal and malicious activities, minimizing false positives and enhancing detection accuracy.

The scope of this project extends beyond the model's training and prediction capabilities. It also includes the development of a user-friendly interface using the Flask web framework, allowing users to interact with the model seamlessly. The interface provides functionalities such as login authentication, real-time predictions, and a dashboard for monitoring activities. Additionally, the tool offers an API endpoint for integrating the detection capabilities with other systems, making it versatile and adaptable for various use cases.

This tool is designed to be scalable, allowing for future improvements, such as incorporating more advanced machine learning models or expanding the feature set used for predictions. The current implementation, while focused on ransomware, can be adapted to detect other forms of malware, providing a comprehensive security solution.

In summary, this project delivers a robust, accessible, and effective ransomware detection tool, contributing significantly to the ongoing efforts in cybersecurity.

## Front End

## 1.2.1 Introduction of HTML

HTML is an acronym which stands for **Hyper Text Markup Language** which is used for creating web pages and web applications. Let's see what is meant by Hypertext Markup Language, and Web page.

**Hyper Text:**

Hyper Text simply means "Text within Text." A text has a link within it, is a hypertext. Whenever you click on a link which brings you to a new webpage, you have clicked on a hypertext. Hyper Text is a way to link two or more web pages (HTML documents) with each other.

**Markup language:**

A markup language is a computer language that is used to apply layout and formatting conventions to a text document. Markup language makes text more interactive and dynamic. It can turn text into images, tables, links, etc.

**Web Page:**

A web page is a document which is commonly written in HTML and translated by a web browser. A web page can be identified by entering an URL. A Web page can be of the static or dynamic type. **With the help of HTML only, we can create static web pages**.

Hence, HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.

**Brief History of HTML:**

In the late 1980's , a physicist, Tim Berners-Lee who was a contractor at CERN, proposed a system for CERN researchers. In 1989, he wrote a memo proposing an internet based hypertext system.

Tim Berners-Lee is known as the father of HTML. The first available description of HTML was a document called "HTML Tags" proposed by Tim in late 1991. The latest version of HTML is HTML5, which we will learn later in this tutorial.

**HTML Versions:**

Since the time HTML was invented there are lots of HTML versions in market, the brief introduction about the HTML version is given below:

**HTML 1.0:** The first version of HTML was 1.0, which was the barebones version of HTML language, and it was released in1991.

**HTML 2.0:** This was the next version which was released in 1995, and it was standard language version for website design. HTML 2.0 was able to support extra features such as form-based file upload, form elements such as text box, option button, etc.

**HTML 3.2:** HTML 3.2 version was published by W3C in early 1997. This version was capable of creating tables and providing support for extra options for form elements. It can also support a web page with complex mathematical equations. It became an official standard for any browser till January 1997. Today it is practically supported by most of the browsers.

**HTML 4.01:** HTML 4.01 version was released on December 1999, and it is a very stable version of HTML language. This version is the current official standard, and it provides added support for stylesheets (CSS) and scripting ability for various multimedia elements.

**HTML5 :** HTML5 is the newest version of HyperText Markup language. The first draft of this version was announced in January 2008. There are two major organizations one is W3C (World Wide Web Consortium), and another one is WHATWG( Web Hypertext Application Technology Working Group) which are involved in the development of HTML 5 version, and still, it is under development.

**Features of HTML**

1) It is a very **easy and simple language**. It can be easily understood and modified.

2) It is very easy to make an **effective presentation** with HTML because it has a lot of formatting tags.

3) It is a **markup language**, so it provides a flexible way to design web pages along with the text.

4) It facilitates programmers to add a **link** on the web pages (by html anchor tag), so it enhances the interest of browsing of the user.

5) It is **platform-independent** because it can be displayed on any platform like Windows, Linux, and Macintosh, etc.

6) It facilitates the programmer to add **Graphics, Videos, and Sound** to the web pages which makes it more attractive and interactive.

7) HTML is a case-insensitive language, which means we can use tags either in lower-case or upper-case.

## Introduction of CSS

CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.

CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

* **What does CSS do**

You can add new looks to your old HTML documents.

You can completely change the look of your website with only a few changes in CSS code.

* **Why use CSS**

These are the three major benefits of CSS:

**1) Solves a big problem**

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation.

**2) Saves a lot of time**

CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

**3) Provide more attributes**

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

* **Advantages of CSS**

**CSS saves time** − You can write CSS once and then reuse same sheet in multiple HTML pages. You can define a style for each HTML element and apply it to as many Web pages as you want.

**Pages load faster** − If you are using CSS, you do not need to write HTML tag attributes every time. Just write one CSS rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

**Easy maintenance** − To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.

**Superior styles to HTML** − CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

**Multiple Device Compatibility** − Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.

**Global web standards** − Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

* **CSS Versions**

Cascading Style Sheets level 1 (CSS1) came out of W3C as a recommendation in December 1996. This version describes the CSS language as well as a simple visual formatting model for all the HTML tags.

CSS2 became a W3C recommendation in May 1998 and builds on CSS1. This version adds support for media-specific style sheets e.g. printers and aural devices, downloadable fonts, element positioning and tables.

**1.2.3 Introduction of JavaScript**

JavaScript is a light-weight object-oriented programming language which is used by several websites for scripting the webpages. It is an interpreted, full-fledged programming language that enables dynamic interactivity on websites when applied to an HTML document. It was introduced in the year 1995 for adding programs to the webpages in the Netscape Navigator browser. Since then, it has been adopted by all other graphical web browsers. With JavaScript, users can build modern web applications to interact directly without reloading the page every time. The traditional website uses js to provide several forms of interactivity and simplicity.

Although, JavaScript has no connectivity with Java programming language. The name was suggested and provided in the times when Java was gaining popularity in the market. In addition to web browsers, databases such as CouchDB and MongoDB uses JavaScript as their scripting and query language.

* **Features of JavaScript**

There are following features of JavaScript:

* All popular web browsers support JavaScript as they provide built-in execution environments.
* JavaScript follows the syntax and structure of the C programming language. Thus, it is a structured programming language.
* JavaScript is a weakly typed language, where certain types are implicitly cast (depending on the operation).
* JavaScript is an object-oriented programming language that uses prototypes rather than using classes for inheritance.
* It is a light-weighted and interpreted language.
* It is a case-sensitive language.
* JavaScript is supportable in several operating systems including, Windows, macOS, etc.
* It provides good control to the users over the web browsers.
* **History of JavaScript**

In 1993, **Mosaic**, the first popular web browser, came into existence. In the **year 1994**, **Netscape** was founded by **Marc Andreessen**. He realized that the web needed to become more dynamic. Thus, a 'glue language' was believed to be provided to HTML to make web designing easy for designers and part-time programmers. Consequently, in 1995, the company recruited **Brendan Eich** intending to implement and embed Scheme programming language to the browser. But, before Brendan could start, the company merged with **Sun Microsystems** for adding Java into its Navigator so that it could compete with Microsoft over the web technologies and platforms. Now, two languages were there: Java and the scripting language. Further, Netscape decided to give a similar name to the scripting language as Java's. It led to 'Javascript'. Finally, in May 1995, Marc Andreessen coined the first code of Javascript named '**Mocha**'. Later, the marketing team replaced the name with '**LiveScript**'. But, due to trademark reasons and certain other reasons, in December 1995, the language was finally renamed to 'JavaScript'. From then, JavaScript came into existence.

* **Application of JavaScript**
* JavaScript is used to create interactive websites. It is mainly used for:
* Client-side validation,
* Dynamic drop-down menus,
* Displaying date and time,
* Displaying pop-up windows and dialog boxes (like an alert dialog box, confirm dialog box and prompt dialog box),
* Displaying clocks etc.

## Back End

## 1.3.1 Introduction of Python

Python tutorial provides basic and advanced concepts of Python. Our Python tutorial is designed for beginners and professionals.

Python is a simple, general purpose, high level, and object-oriented programming language.

Python is an interpreted scripting language also. *Guido Van Rossum* is known as the founder of Python programming.

* **What is Python**

**Python** is a general purpose, dynamic, [high-level](https://www.javatpoint.com/classification-of-programming-languages), and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

Python is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.

Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc.

We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable.

Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

* **Python 2 vs. Python 3**

In most of the programming languages, whenever a new version releases, it supports the features and syntax of the existing version of the language, therefore, it is easier for the projects to switch in the newer version. However, in the case of Python, the two versions Python 2 and Python 3 are very much different from each other.

A list of differences between Python 2 and Python 3 are given below:

Python 2 uses **print** as a statement and used as print "something" to print some string on the console. On the other hand, Python 3 uses **print** as a function and used as print("something") to print something on the console.

Python 2 uses the function raw\_input() to accept the user's input. It returns the string representing the value, which is typed by the user. To convert it into the integer, we need to use the int() function in Python. On the other hand, Python 3 uses input() function which automatically interpreted the type of input entered by the user. However, we can cast this value to any type by using primitive functions (int(), str(), etc.).

In Python 2, the implicit string type is ASCII, whereas, in Python 3, the implicit string type is Unicode.

Python 3 doesn't contain the xrange() function of Python 2. The xrange() is the variant of range() function which returns a xrange object that works similar to Java iterator. The range() returns a list for example the function range(0,3) contains 0, 1, 2.

There is also a small change made in Exception handling in Python 3. It defines a keyword **as** which is necessary to be used. We will discuss it in Exception handling section of Python programming tutorial.

* **Python History**

Python was invented by **Guido van Rossum** in 1991 at CWI in Netherland. The idea of Python programming language has taken from the ABC programming language or we can say that ABC is a predecessor of Python language.

There is also a fact behind the choosing name Python. Guido van Rossum was a fan of the popular BBC comedy show of that time, **"Monty Python's Flying Circus"**. So he decided to pick the name **Python** for his newly created programming language.

Python has the vast community across the world and releases its version within the short period.

* **Where is Python used?**

Python is a general-purpose, popular programming language and it is used in almost every technical field. The various areas of Python use are given below.

* Data Science
* Date Mining
* Desktop Applications
* Console-based Applications
* Mobile Applications
* Software Development
* Artificial Intelligence
* Web Applications
* Enterprise Applications
* 3D CAD Applications
* Machine Learning

**1.3.2 Flask**

Flask is a lightweight and versatile web framework for Python, designed to be simple yet powerful, making it a popular choice for developers looking to build web applications quickly and efficiently. Released in 2010 by Armin Ronacher, Flask is classified as a microframework because it does not require specific tools or libraries, giving developers the freedom to structure their applications as they see fit. Despite its minimalistic core, Flask is highly extensible, allowing users to add functionality through various extensions as needed.

One of the key features of Flask is its simplicity and ease of use. Flask applications start with just a single Python file, making it easy for beginners to get started with web development. The core of Flask is built around WSGI (Web Server Gateway Interface), which serves as a standard for Python web application development. Flask also uses Jinja2 for templating, allowing developers to create dynamic HTML content by embedding Python expressions within HTML files.

Flask's routing system is another strong feature, enabling developers to define the routes (URLs) of their web application effortlessly. A route in Flask maps a URL to a Python function, allowing the function to handle requests to that URL. For instance, the @app.route("/") decorator binds the root URL ("/") to a specific function, which determines what the user sees when they visit that URL. This straightforward approach makes it easy to manage different parts of a web application.

Despite being a microframework, Flask supports essential web development features like session management, request handling, and form validation. Additionally, Flask provides built-in development servers and debugging tools, which are invaluable during the development process. The development server allows developers to run their application locally and make changes in real-time, while the debugging tool provides detailed error messages and stack traces when something goes wrong.

Flask's modular design is one of its greatest strengths. Developers can keep the core of their applications lean by only adding the necessary extensions. For example, Flask-SQLAlchemy can be used for database integration, Flask-WTF for handling web forms, and Flask-Login for user authentication. This modularity ensures that Flask applications remain lightweight, without unnecessary bloat, while still being capable of handling complex tasks.

In summary, Flask's simplicity, flexibility, and powerful features make it an excellent choice for both beginners and experienced developers. Whether building a small web service or a full-fledged web application, Flask provides the tools and structure needed to create efficient and maintainable code.

**1.3.3 Machine Learning**

In the real world, we are surrounded by humans who can learn everything from their experiences with their learning capability, and we have computers or machines which work on our instructions. But can a machine also learn from experiences or past data like a human does? So here comes the role of **Machine Learning**.

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own.

With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

## How does Machine Learning work

A Machine Learning system **learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it**. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm:

****

## Features of Machine Learning:

* Machine learning uses data to detect various patterns in a given dataset.
* It can learn from past data and improve automatically.
* It is a data-driven technology.
* Machine learning is much similar to data mining as it also deals with the huge amount of the data.

**CHAPTER 2**

**SOFTWARE DEVELOPMENT LIFE CYCLE**

## REQUIREMENT ANALYSIS PHASE

The Requirements Analysis Phase begins when the previous phase objectives have been achieved. Documentation related to user requirements from the Concept Development Phase and the Planning Phase shall be used as the basis for further user needs analysis and the development of detailed requirements. Multiple-release projects require only one iteration of the Requirements Analysis Phase, which should involve requirements definition for all planned releases.

The objective of this phase is to define in more detail the system inputs, processes, outputs and interfaces. At the end of this phase the system’s processes will be defined at the functional level, meaning the functions to be performed will be known, but not necessarily how they will be performed. Unless specifically constrained by the Project Charter, Requirements Analysis should not consider the computer programs, files and data streams.   
  
Requirements Analysis will identify and consider the risks related to how the technology will be integrated into the standard operating procedures. Requirements Analysis will collect the functional and system requirements of the business process, the user requirements and the operational requirements (e.g., when operational what is necessary to keep the system up and running).

## System Requirement Specification

Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function (in a market-driven project, these roles may be played by the marketing and development divisions). Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communications with the project team and customer throughout the software development process.

## Purpose

The purpose of this document is to give a detailed description of the requirements for the “Amazing Lunch Indicator” (ALI) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## Hardware and Software Requirement

#### Hardware Specification

* + - * RAM 4 GB
      * GPU

#### Software Requirements:

* + - * Python
      * PyCharm
      * Browser to Test

## Functional Requirement

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define whata system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements which impose constraints on the design or implementation.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non-functional requirements drive the technical architecture of a system.

## Non-Functional Requirement

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. They are contrasted with functional requirements that define specific behaviour or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture, because they are usually Architecturally Significant Requirements.

Broadly, functional requirements define what a system is supposed to do and non-functional requirements define how a system is supposed to be. Functional requirements are usually in the form of, an individual action or part of the system, perhaps explicitly in the sense of a mathematical function, a black box description input, output, process and control functional model or IPO Model. In contrast, non-functional requirements are in the form of, an overall property of the system as a whole or of a particular aspect and not a specific function. The system's overall properties commonly mark the difference between whether the development project has succeeded or failed.

Non-functional requirements are often called "quality attributes" of a system. Other terms for non-functional requirements are "qualities", "quality goals", "quality of service requirements", "constraints" and "non-behavioural requirements". Informally these are sometimes called the "ileitis", from attributes like stability and portability. Qualities—that is non-functional requirements—can be divided into two main categories:

1. Execution qualities, such as safety, security and usability, which are observable during operation.
2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the system.

## Feasibility Study

Feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer foresee the future of the project and the usefulness.

A feasibility study of a system proposal is according to its workability, which is the impact on the organization, ability to meet their user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

Following are its feature

**2.1.6 TECHNICAL FEASIBILITY**

The system must be evaluated from the technical point of view first. The assessment of this feasibility must be based on an outline design of the system requirement in the terms of input, output, programs and procedures.

Having identified an outline system, the investigation must go on to suggest the type of equipment, required method developing the system, of running the system once it has been designed.

Technical issues raised during the investigation are:

* Does the existing technology sufficient for the suggested one?
* Can the system expand if developed?

#### ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest.

One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

* The costs conduct a full system investigation.
* The cost of the hardware and software.
* The benefits in the form of reduced costs or fewer costly errors.

Since the system is developed as part of project work, there is no manual cost to spend for the proposed system. Also all the resources are already available, it give an indication of the system is economically possible for development

#### BEHAVIORAL FEASIBILITY

This includes the following questions:

* Is there sufficient support for the users?
* Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioural aspects are considered carefully and conclude that the project is behaviourally feasible.

**CHAPTER 3**

DIAGRAMS

**Implementation or Architecture Diagrams**

1. **Data Flow Diagram (DFD)**

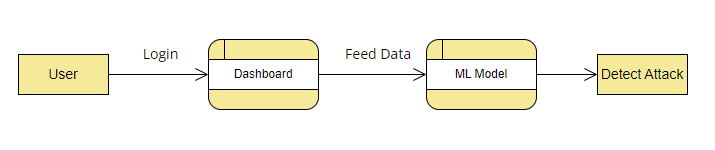
A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble chart, The data flow diagram may be used to represent a system or software at any level of abstraction. As information moves through software, it is modified by a series of transformations.

A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The basic form of a data flow diagram, also known as a data flow graph or a bubble chart. DFD is an abstract description of the system. The data flow diagram may be used to represent a system or software at any level of abstraction. DFDs may be partitioned into levels that represent increasing information flow and functional detail. Therefore, the DFD provides a mechanism for functional modeling as well as information flow modeling.

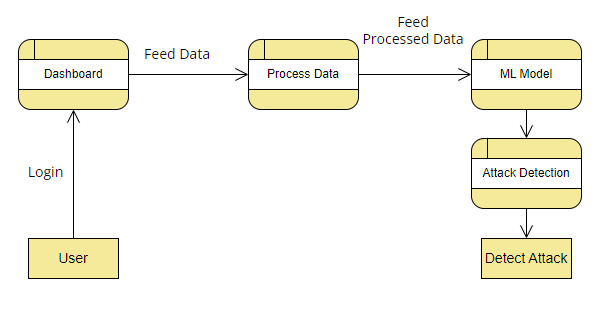
DFDs are very useful in understanding a system and can be effectively used during analysis. DFDs can be hierarchically organized, which helps in progressively partitioning and analyzing large systems. Such DFDs are called leveled DFDs.

.

**Level-0 DFD**

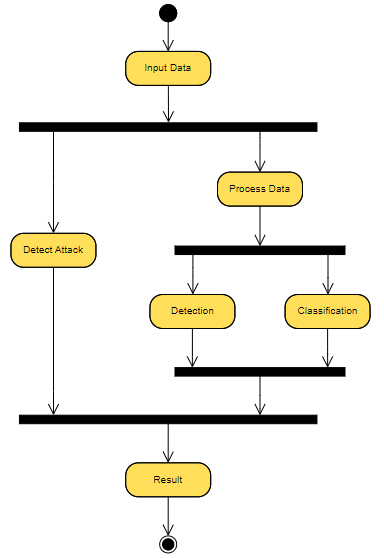
****

**Level-1 DFD**

****

**2. ACTIVITY DIAGRAM**

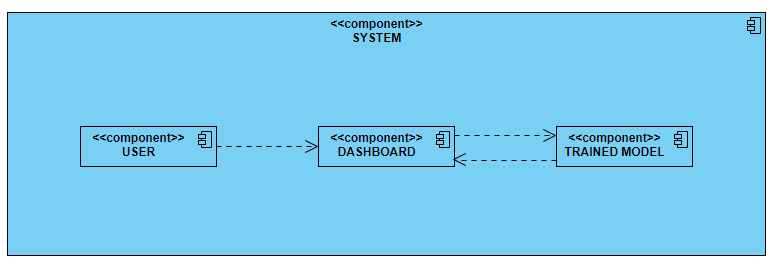
The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.



**3. COMPONENT DIAGRAM**

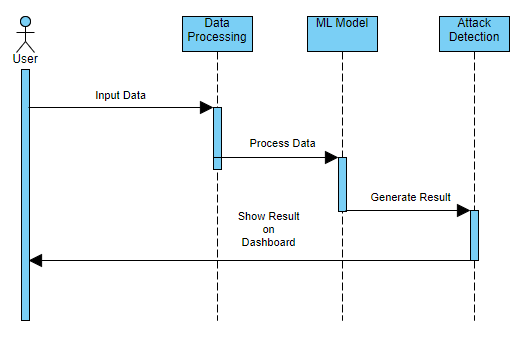
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

A component diagram allows verification that a system's required functionality is acceptable. These diagrams are also used as a communication tool between the developer and stakeholders of the system. Programmers and developers use the diagrams to formalize a roadmap for the implementation, allowing for better decision-making about task assignment or needed skill improvements. System administrators can use component diagrams to plan ahead, using the view of the logical software components and their relationships on the system.

****

**4. SEQUENCE DIAGRAM**

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.



**CHAPTER 4**

CODING

1. **Python Code for Main App (Controller)**

from flask import Flask,render\_template,redirect,request,session

import joblib

import numpy as np

model = joblib.load("../models/model1.pkl")

features = joblib.load("../models/features.pkl")

app = Flask(\_\_name\_\_)

app.secret\_key = 'asd84a5d6d6sd5a6dq4asc4d6wed4s65c5d4ed6as5sad46d56as5d56d56qd6eedwe4e8d'

@app.route("/")

def Home():

if "login" not in session:

return redirect("/login")

data = {"features":list(features)}

@app.route("/login",methods = ["POST","GET"])

def Login():

if request.method == "GET":

return render\_template("login.html")

return "Error"

@app.route("/logout")

def logout():

session.pop("login")

return redirect("/login")

@app.route("/predict",methods = ["POST"])

def Predict():

output = "null"

user\_inputs = request.form

vals = []

for f in features:

vals.append(user\_inputs[f])

vals = [np.array([int(x) for x in vals])]

@app.route("/api/predict",methods = ["POST"])

def PredictApi():

output = "null"

user\_inputs = request.form

vals = []

output = int(pred[0])

return {"value":output,"output": "Attack Detected" if output == 1 else "Not Detected"}

@app.route("/dashboard")

def Dashboard():

if "login" not in session:

return redirect("/login")

return render\_template("dashboard.html")

app.run(debug=True)

1. **HTML Home Page Code**

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Ransomware Detection Tool</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">

<style>

body {

background-image: url('{{ url\_for('static', filename='back.png') }}');

background-size: cover;

background-position: center;

background-repeat: no-repeat;

}

</style>

</head>

<body>

<nav class="navbar bg-dark border-bottom border-body justify-content-between" data-bs-theme="dark">

<div class="container-fluid">

<a class="navbar-brand" href="#">Ransomware Detection Tool</a>

<div>

<a class="navbar-brand" href="/logout">Logout</a>

<a class="navbar-brand" href="/dashboard">Model Analysis</a>

</div>

</div>

</nav>

<div class="container mt-4">

<h1 class="display-12 text-center mb-4 text-white">Detect Attack Here</h1>

<section>

<div class="row justify-content-center">

<div class="col-md-8 mb-6">

<div class="card mb-6">

<div class="card-header py-3">

<h5 class="mb-0">Provide Details for Detection</h5>

</div>

<div class="card-body">

<form action="/predict" method="post">

<div class="row row-cols-3">

{% for x in data.features %}

<div class="col">

<div class="input-group flex-nowrap my-1">

<span class="input-group-text" id="addon-wrapping">#</span>

<input type="text" required class="form-control" name={{x}} placeholder={{x}} aria-label="Username" aria-describedby="addon-wrapping">

</div>

</div>

{% endfor %}

</div>

<button type="submit" class="btn btn-dark mt-5">Predict</button>

</form>

</div>

</div>

</div>

</div>

</section>

<br>

</div>

<div class="container">

<h1 class="display-4 mb-5 text-center text-white">Output</h1>

{% if data.output == 0 %}

<h1 class="display-6 mb-5 text-center text-success fw-bold text-white">Not Detected</h1>

{% elif data.output == 1 %}

<h1 class="display-6 mb-5 text-center text-danger fw-bold text-white">Detected</h1>

{% endif %}

</div>

</body>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz" crossorigin="anonymous"></script>

</body>

</html>

**CHAPTER 5**

TESTING

## Testing Phase

Testing refers to test the software so it is also called software testing. **Software testing** is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test.[[1]](https://en.wikipedia.org/wiki/Software_testing#cite_note-Kaner_1-1) Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test-

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time
* is sufficiently usable
* can be installed and run in its intended environments, and
* achieves the general result its stakeholder’s desire.

## 5.1 Unit Testing

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Parameterized unit tests (PUTs) are tests that take parameters. Unlike traditional unit tests, which are usually closed methods, PUTs take any set of parameters. PUTs have been supported by Testing, JUnit and various .NET test frameworks.

Suitable parameters for the unit tests may be supplied manually or in some cases are automatically generated by the test framework. Testing tools like Quick Check exist to generate test inputs for PUTs.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it offers several benefits.

In our tool we tested each and every unit (or module) and it was successfully executed.

We perform attack detection process by our tool, and it is working fine, fulfill all criteria and give us correct result.

## Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing. Some different types of integration testing are big-bang, mixed (sandwich), risky- hardest, top-down, and bottom-up. Other Integration Patterns are collaboration integration, backbone integration, layer integration, client-server integration, distributed services integration and high-frequency integration. In the big-bang approach, most of the developed modules are coupled together to form a complete software system or major part of the system and then used for integration testing. This method is very effective for saving time in the integration testing process.

However, if the test cases and their results are not recorded properly, the entire integration process will be more complicated and may prevent the testing team from achieving the goal of integration testing. In our tool we tested all components by merging together and it passed all our criteria.

## System Testing

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black-box testing, and as such, should require no knowledge of the inner design of the code or logic.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system.

The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

After performing system testing with our tool, we find that it follows all requirements and working as per requirement, it takes the user’s inputs to process data and give result as a possible attack and suggest precaution measures.

## Functional Testing

Functional testing is a [quality assurance](https://en.wikipedia.org/wiki/Quality_assurance) (QA) process and a type of [black-box testing](https://en.wikipedia.org/wiki/Black-box_testing) that bases its test cases on the specifications of the software component under test.

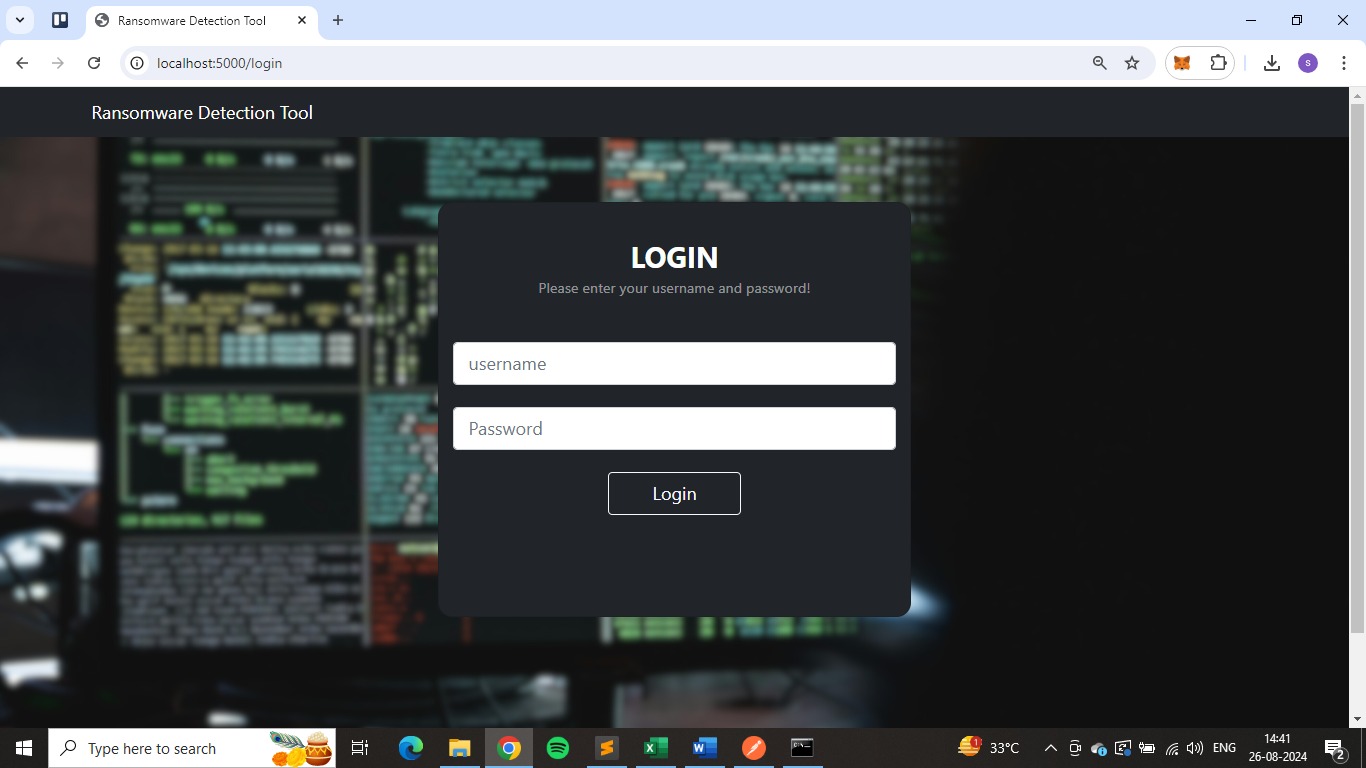
Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike [white-box testing](https://en.wikipedia.org/wiki/White-box_testing)). Functional testing is conducted to evaluate the compliance of a system or component with specified functional [requirements](https://en.wikipedia.org/wiki/Requirement). Functional testing usually describes what the system does.

All functionality of ransomware detection tools is working properly. It took input parameters properly and gives the output as per requirements.

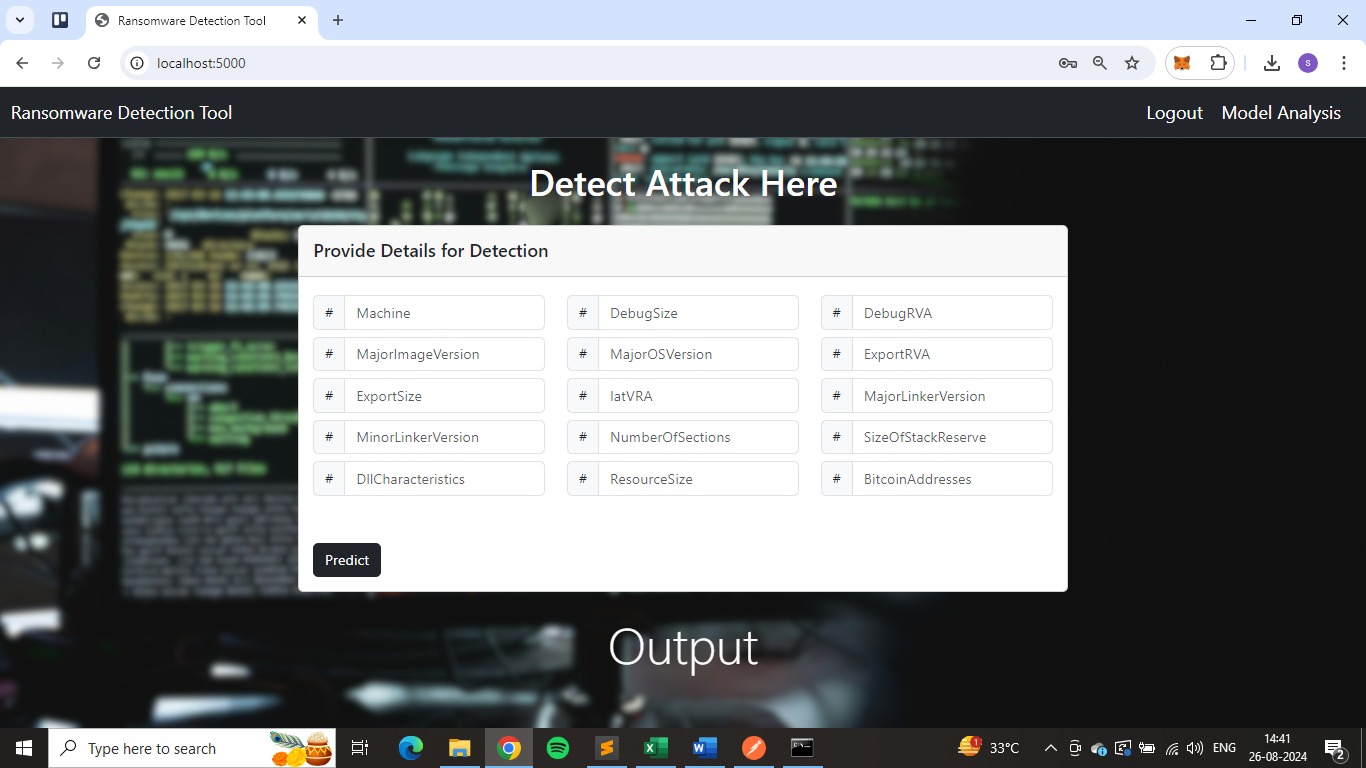
**CHAPTER 6**

SNAPSHOTS

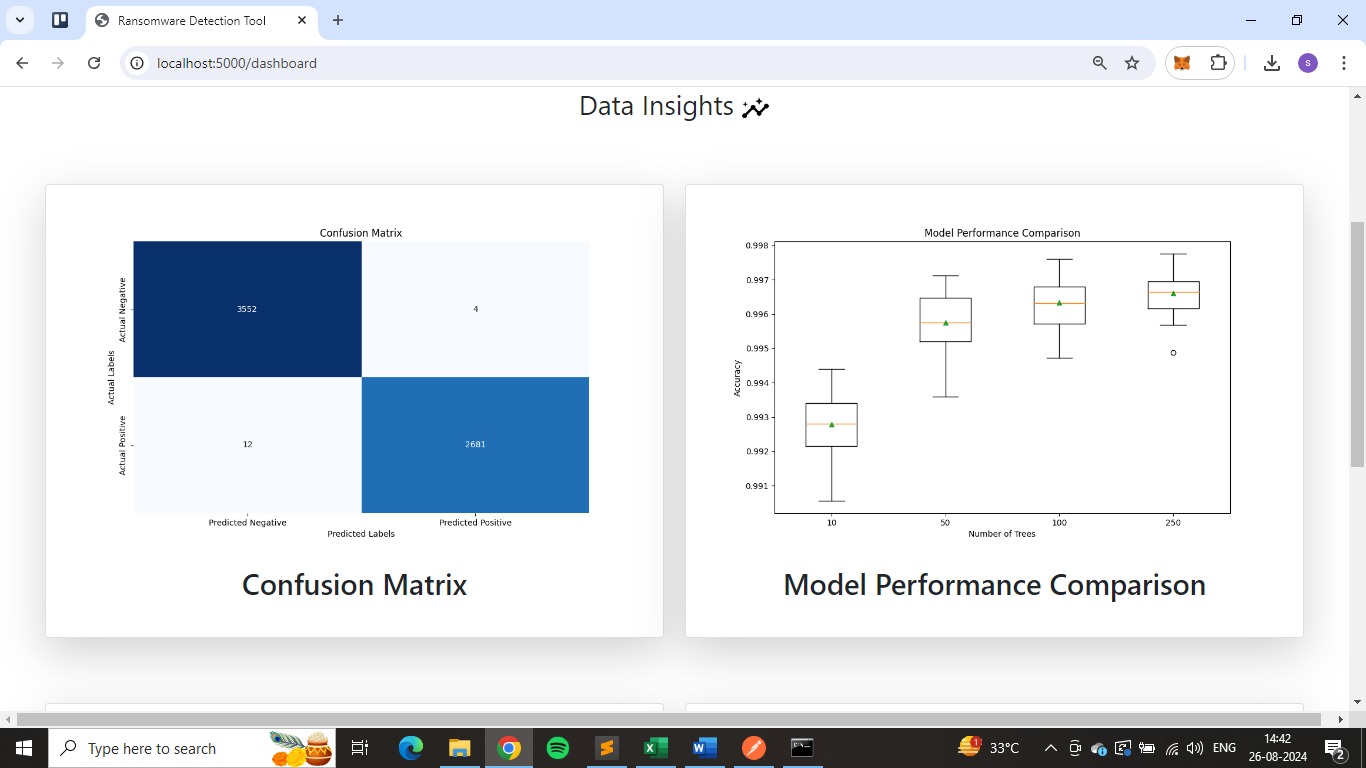
1. **Login Page**

****

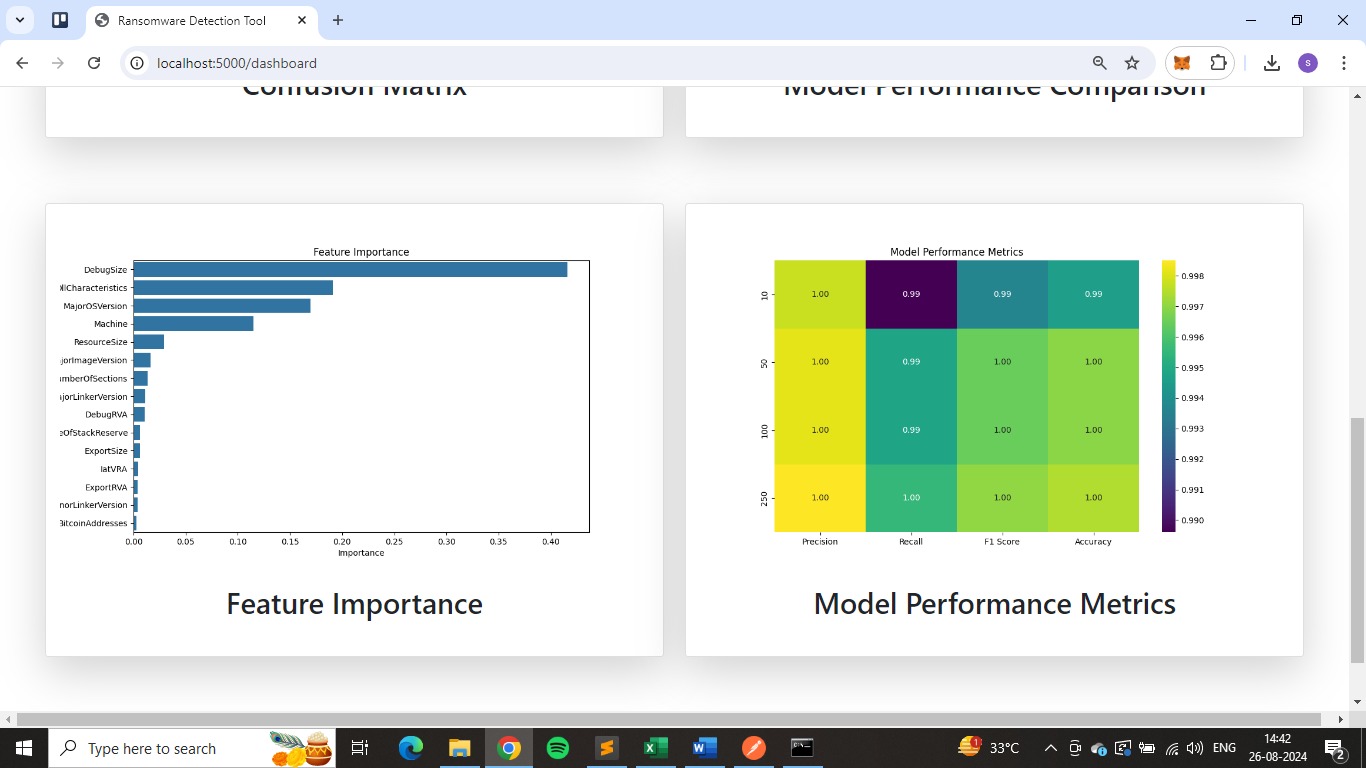
1. **Home Page**



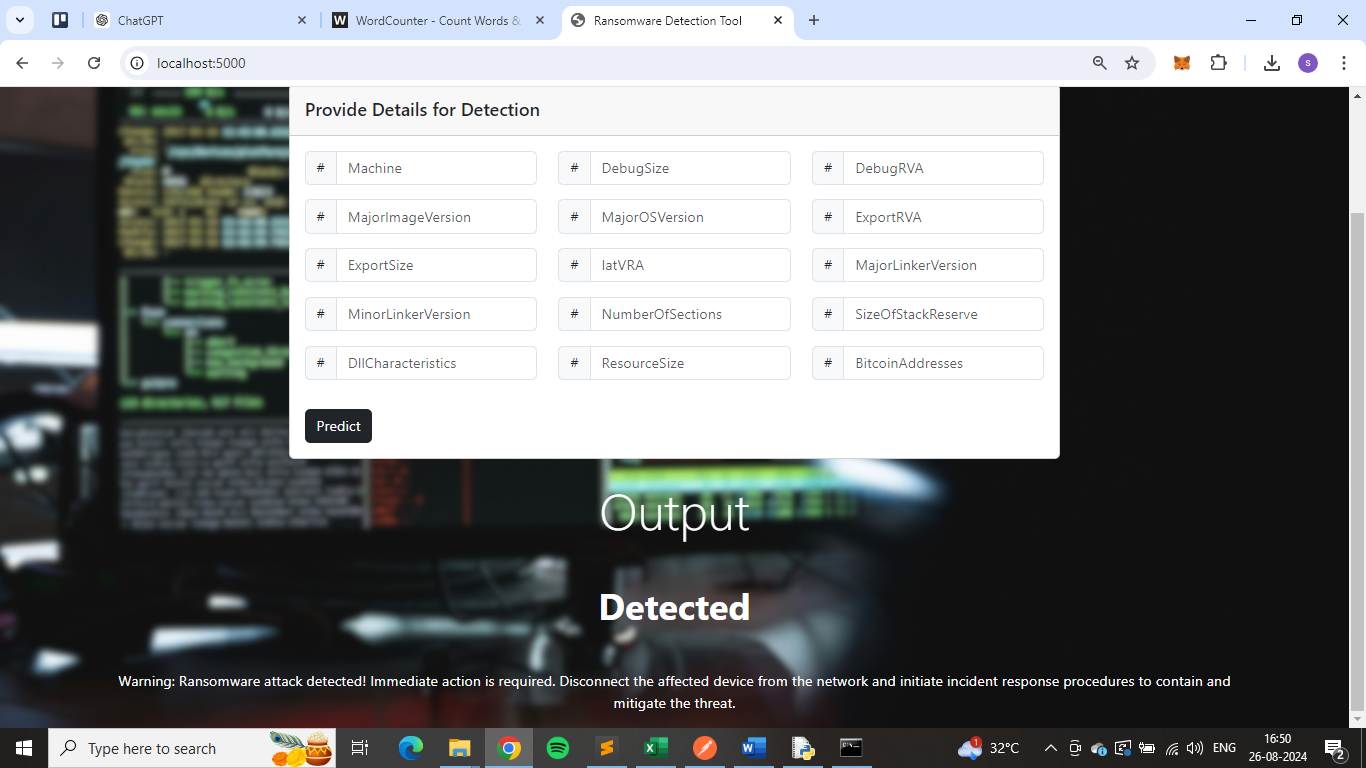
1. **Model Analysis Page**

****

1. **Model Analysis Page**

****

1. **Output Page**

****

**CHAPTER 7**

FUTURE ENHANCEMENT

**Future Enhancement**

In Future we will add some more features to tool like-

* **Improve Model Accuracy:** Experiment with different algorithms and hyperparameters to boost detection performance.
* **Add Real-Time Monitoring:** Implement real-time monitoring to detect and respond to ransomware attacks immediately.
* **Enhance User Interface:** Make the Flask app more user-friendly with interactive dashboards and better visualizations.
* **Integrate Threat Intelligence:** Use updated threat intelligence feeds to stay current with new ransomware threats.
* **Increase Scalability:** Deploy the tool using containers or cloud solutions to handle larger volumes and ensure reliability.

**REFERENCES**

#### BOOKS:

* Ian Somerville ‘Software engineering’
* Rajeev mall ‘Software engineering’

#### ONLINE REFERENCE:

1. [www.google.co.in](http://www.google.co.in)
2. <https://stackoverflow.com/>
3. <https://docs.djangoproject.com/en/3.2/>
4. <https://www.w3schools.com/>