Server Based Malware Scanner

BY

Sajid Hasan Akhand

ID: 192-16-449

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computing and Information System

Supervised By

Md. Sarwar Hossain Mollah

Associate Professor and Head

Department of CIS

Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

APPROVAL

This Project titled "**Server Based Malware Scanner**", submitted by **Sajid Hasan Akhand**, ID: 192-16-449 to the Department of Computing and Information System, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computing and Information System and approved as to its style and contents. The presentation was held on 23-09-2024.

BOARD OF EXAMINERS

Md Sarwar Hossain Mollah — **Chairman**
Associate Professor and Head
Department of Computing and Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Nasimul Kader
Assistant Professor
Department of Computing and Information System
Faculty of Science & Information Technology
Daffodil International University

Internal Examiner

Md. Mehedi Hassan
Lecturer
Department of Computing and Information System
Faculty of Science & Information Technology
Daffodil International University

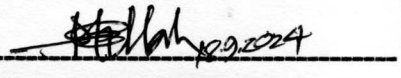
External Examiner

Dr. Saifuddin Md. Tareeq
Professor
Department of Computer Science and Engineering
University of Dhaka, Dhaka

Declaration

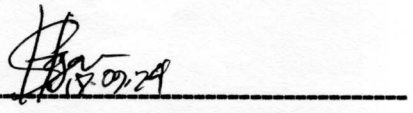
I hereby declare that; this project has been done by me under supervision of **Md. Sarwar Hossain Mollah, Associate Professor and Head,** Department of Computing and Information System (CIS) of Daffodil International University. I am also declaring that this project or any part of it has never been submitted anywhere else for the award of any educational degree like, B.Sc., M.Sc., Diploma or other qualifications.

Supervised By



Md. Sarwar Hossain Mollah
Associate Professor and Head
Department of CIS
Daffodil International University

Submitted By



Name: Sajid Hasan Akhand
ID: 192-16-449
Department of CIS
Daffodil International University

ACKNOWLEDGEMENT

First, I express our heartiest thanks and gratefulness to almighty Allah for His divine blessing making us possible to complete the final year project successfully.

I am grateful and wish my profound indebtedness to**, Md. Sarwar Hossain Mollah, Associate Professor & Head, Department of CIS Daffodil International University, Dhaka.** The Deep knowledge & keen interest of my supervisor in the field of "Web development & Cybersecurity" helped to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude again to **Md. Sarwar Hossain Mollah, Head,** Department of CIS, for his kind help to finish my project and also to other faculty members and the staff of the CIS department of Daffodil International University.

I would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, I must acknowledge with due respect the constant support and patients of our parents.

ABSTRACT

You may read the findings that go along with my concept, a "server based malware scanner," in full. This report includes an extensive discussion of the techniques that helped transform the idea into a working website. One module can be distinguished among users inside the system: dashboard for the admin. In this project there has been a login system for scanning specific locations of computers. Two scanners have been used in this system like: “Normal Scan” and “Deep scan”. Normal scan finds the initial problem of malware and gives the solution according to ChatGPT. Deep scanners can find out all problems; include normal scanner problems and give the solution according to ChatGPT. Also have configuration of the scan specific portion. Code checker functions have been used to check for specified code portions. I have used an HTML user interface, CSS style pages, JS, and PHP web application with a reliable backend with XAMPP. To set up our system application, all that is required are desktop computers and online access; costly software or computer components are not required. If you have the right login credentials, you may utilize our infrastructure as a worldwide repository and as easily accessible apps from anywhere. With a typical internet-accessible network, almost any user, from anywhere, at any time, may utilize our platform-independent solution. They may modify my system to meet specific needs as well.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Introduction

An application that shows the relationships and interactions between other apps is called a system. Computers' "System" page includes programming connections, programs, and system management tools. While the term "system" may have varied meanings depending on the situation, the idea is basically the same. The "Serve based malware scanner" develops an extensive framework by integrating several technologies. The limits established for each system under consideration are applied by the several modules that make up this framework. There are several systems in each module.

The "Server based malware scanner " project is an important instrument intended to identify and eliminate malicious software threats that are directed at servers in a networked environment. In contrast to conventional antivirus software, which targets specific endpoints like PCs or laptops, server-based malware detection programs are designed to safeguard the computer systems themselves, which frequently house vital information, software, and services. keeps an eye out for indications of abnormal activity or malware infection in server operations and incoming data. is a crucial part of an all-encompassing cybersecurity plan for businesses that depend on servers to keep running and hold important data. Its proactive identification and remediation features contribute to the protection of server-based services from constantly changing malware threats.

Fundamentally, the goal of how I constructed them is to automate tedious manual tasks, which is precisely what systems are meant to do. The needs of the user dictate how the system is set up; for example, a manufacturing company may build a plant that automates manual tasks. To automate the manual registration process, I have turned to computer programming. Our lives have drastically altered as a result of computer advancement. When computers are utilized correctly, they can help us learn new skills, protect our personal information, save time and energy, and access the data we need whenever we need it.

CHAPTER 2

Initial Study

2.1 Project Proposal

Objectives

The goal of the current project is to create a server-based malware scanning system that makes use of technology to find malware and find intelligent solutions to problems. This project has one dashboard: administrators for the delivery of sophisticated malware scanners. It has complete control over all aspects of malware issues. The GPT discussion goes into further depth about how to solve malware problems. This suggested web application system's computerization of both malware scanner database is one of its goals:

- To automate admin and malware scanning databases.
- To preserve the integrity and consistency of data.
- Automate the registration process to eliminate the need for in-person communication.
- Scan specific location of PC to detect malware.
- Every record has been kept, and the administrator may examine every instance of malware on the system.

Benefits of the website:

- Scan malware problem.
- Find out specific malware problems.
- Find any location of PC malware problem.

2.2 Background of the server-based malware scanner system

Customers' ways of getting goods and services have been revolutionized as a result of the rise in the use of smartphones, web browsing, and e-commerce pages. These systems offer efficiency, accuracy, and enhanced departmental communication, which makes them essential for handling the complex and varied operations of the parking system. The technical foundation of the

server-based malware scanning system is becoming more and more difficult in terms of administrative responsibilities.

Early structures were frequently specialized and confined, serving specific purposes like several kinds of virus detection systems. The majority of server-based virus scanners for training purposes are made to help consumers become more tech-savvy. This server-based malware scanner has taken into account several factors, including A standard scan identifies the first malware issue and provides a remedy based on ChatGPT. Any problem, even common scanner issues, may be identified by a deep scanner, which can also provide a solution based on ChatGPT.

2.3 Problem Area

A server-based malware scanner's issue statement focuses on the particular difficulties and specifications involved in safeguarding servers from harmful software threats. includes several different difficulties, such as proactive threat identification, performance optimization, flexibility, legal compliance and integration. Taking care of these issues guarantees the creation of a reliable, strong, and efficient solution that protects vital server infrastructures from malware's constant danger. Here's a methodical way to formulate the problem statement:

- Vulnerability Identification
- Immediate Danger Identification
- Enhancing Performance
- All-Inclusive Safety Measures
- Maintenance and Scalability
- Adherence to Regulations
- Reaction and Corrective Action
- Consolidation and Flexibility.

2.4 Possible Solution

The array of server-based malware scanners consists of a dynamic collection of online resources and practical strategies designed to revolutionize the user-friendliness and productivity of malware scanner maintenance. The central component of the system is a web-based platform or

mobile application that serves as the hub for all things connected to malware scanners. A server-based malware scanner's scope includes detection and prevention of malware threats, efficiency improvements, central management, compliance requirements, and integration with pre-existing infrastructure. It also includes protection measures that are specifically designed for servers. Organizations may protect their vital server assets from prospective cyber-attacks by identifying these characteristics and putting an effective security policy in place.

CHAPTER 3

Literature Review

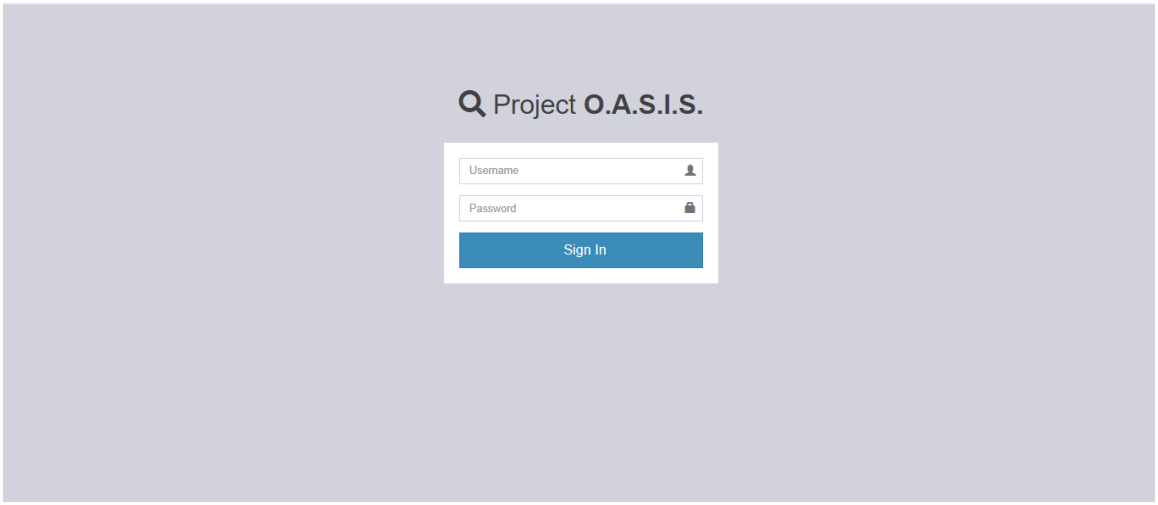
3.1 Discussion on problem domain based on published articles

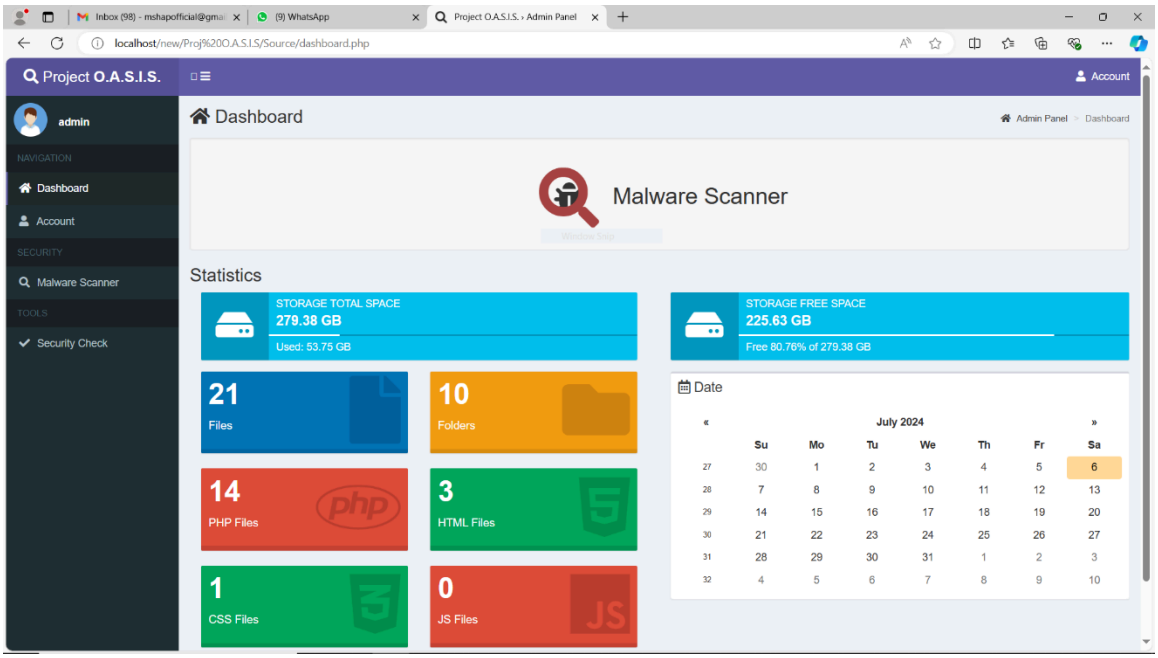
The principles for the admin systems are defined in this section. Examine the accompanying relevant works as well. Included in the scope of a server-based malware scanner are compliance requirements, efficiency gains, central management, the identification and avoidance of malware threats, and compatibility with pre-existing technology.

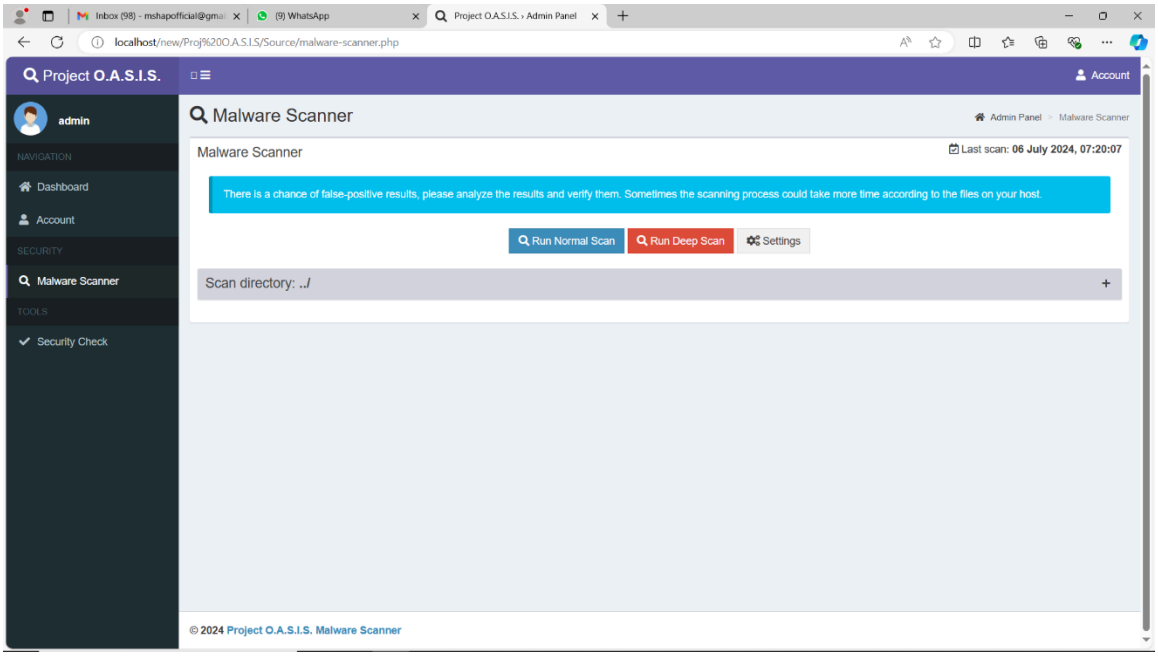
3.2 Discussion on problem solutions based on published articles

A specialized software application called a server-based malware scanner is made to guard servers against different kinds of harmful software, or malware. In contrast to conventional antivirus programs, which are mainly concerned with safeguarding individual endpoints such as laptops or desktop computers, server-based malware detectors are designed to satisfy the particular security requirements of servers in business networks. A log-in mechanism was used in this project to scan a particular computer location. This system makes use of two scanners: "Normal Scan" and "Deep scan."

3.3 Comparison of three/four leading solutions







Best Features:

- Scanning malware at any location.
- Configuration settings.
- Security check
- Malware problem solution

Limitations

Although the features and efficacy of server-based malware scanners differ greatly, they usually have a few similar drawbacks and difficulties. largely relies on signatures of known malware, which may not be able to identify zero-day threats or malware that is polymorphic—that is, changes its code to avoid detection—or both. It is possible for newly developed malware variations to evade signature-based detection, making servers open to highly skilled attackers. When scanning, server-based scanners can be quite resource-intensive, using a lot of CPU, RAM, and disk I/O resources. This may result in a decline in the scanned servers' performance, which would impact the availability and responsiveness of vital services. Updating malware signatures on a regular basis is essential to preserving detection effectiveness. Particularly during times of fast malware evolution, delays in maintaining signatures may provide holes in defense

against new malware variants. The shortcomings of current server-based malware scanners must be addressed via ongoing innovation and adaptability to changing cyberthreats. Important areas for improvement include expanding the ability to detect beyond signature-based techniques, increasing scalability, optimizing resource efficiency, and integrating easily with larger IT infrastructure. Organizations may enhance their server infrastructures' security against malware and preserve operational effectiveness and regulatory compliance by reducing these constraints. Despite their many advantages, server-based virus scanning solutions still have several disadvantages.

- Dependency on technology;
- Costs associated with internet access;
- Difficulty with user acceptance and interface;
- Concerns about safety and confidentiality

3.4 Recommended Approach

Table 1: Modules descriptions

Actuators	Functions
ADMIN	● Admin dashboard. ● Log in and out. ● See all types of files which have been located. ● Change Password ● Two types of Malware scan: Normal & Deep scan. ● Solved the problem using chat GPT after scanning. ● Configuration for the directory search. ● Security Check.

CHAPTER 4
Methodology

4.1 What to Use

The process for creating a server-based malware scanner system which allows administrators to scan different locations on the computer drive is described in this section. The method includes the theoretical basis and architecture of the system, the procedure for identifying attributes pertinent to this project, and a log-in system for scanning a particular computer site to resolve the computer's virus issue. I'll go over how to accomplish a project's aim with the malware scanner in this chapter. I am aware that the accepted technique is the SDLC life cycle model, which is a comprehensive framework for design, development, etc. I am

familiar with several SDLC model types. The method of developing software includes models such as the Big Bang, Spiral, Waterfall, Agile, Iterative, and Dynamic System Development models, among others. Each model provides a framework and means of guiding the development and implementation of the vehicle components platform. The unique need of the SDLC model will tailor the malware scanner domain in order to provide an efficient development process that aligns with the goals of the malware detect project.

4.2 Why to use

The system architecture was created as the initial stage of the development process. Determining the parts and their interactions was a part of this. Top considerations in the architecture of the system infrastructure were security, dependability, and scalability. It entailed keeping database administration and the product's back-end operations apart from the user interface. In order to safeguard computer data and guarantee safe transactions, the design also included the necessary security measures. Every software project has to use the agile process. I am familiar with a number of agile technique names, including feature-driven development, scrum, crystal, dynamic system development method, and kanban. However, I applied the DSDM approach to my project. The DSDM approach is useful for a variety of reasons. Iterative development is developed using the dynamic system development method, which allows for flexibility in adjusting requirements. This approach works well when prompt delivery is needed. The Dynamic System Development Method requires the frequent delivery of functional software projects. This agility strategy accepts companies to use valuable sooner, reduces time to market, and improves reporting to market shifts. This agile method helps team members find probability hazards early in the project lifecycle by organizing risk management strategies into its actions. Project failure decreases after taking proactive steps to manage risks. For these reasons, I approached my project using the DSDM agile process.

4.3 Section of methodology

Pre-Project Phase:

- **Feasibility Study**: The project concept's operational, financial, and technical feasibility are evaluated at this point. It comprises assessing the potential costs, benefits, and hazards associated with the project.
- **Conditions Collecting**: The requirements for the program are gathered and recorded at this step. In order to specify the project's scope, it is necessary to comprehend the user's expectations, business requirements, and restrictions.
- **Planning:** The planning stage entails drafting an endeavor plan that details the goals of the undertaking, its schedule, the resources needed, and deliverables. It entails determining the parties involved in the project, outlining roles and duties, and putting in place a plan for communication and risk management.

Project Lifecycle Phase:

- **Design:** Using the requirements acquired, the software design is built in this phase. Architectural, database, and design of user interfaces are only a few examples of the high-level and comprehensive design tasks it involves.
- **Development**: Utilizing the design guidelines as a guide, the software is coded during this phase. To construct a software product that functions, developers write the source code, carry out unit tests, and integrate components.
- **Testing**: This stage is all about putting the software through its paces to make sure it works and is of high quality. Numerous testing techniques are covered, including system, user acceptability, integration, and unit testing.
- **Deployment**: After undergoing extensive testing and receiving approval, the program is introduced into the operational environment. The program must be installed, configured, and set up in the intended environment.

Post-Project Phase:

- **Maintenance**: The software goes into the maintenance phase following its deployment. To make sure the program stays functioning and satisfies evolving requirements, this phase entails continuing maintenance, bug repairs, and

upgrades.

- **Evaluation**: Comparing the project's actual results to its predetermined goals allows for an assessment of the project's success. For next projects, it aids in pinpointing areas in need of improvement and lessons gained.

- **Closure:** The project officially ends at this phase. It entails completing all project documentation, preserving project artifacts, and performing a project review or post-mortem.

These parts cover everything from preliminary planning to post-implementation assistance, offering an organized method for managing software development projects and assisting in achieving successful results.

4.4 Implementations plans

At this point in the project, the application is released to the general public for use. The new system must be put into service as soon as a flaw is discovered and fixed. In this part, the settings, methods, and release standards are selected. The new system is then put to the test and put into use if everything goes as planned.

Features:

- Admin can generate two types of malware scan.
- Solved the Malware problem.
- Code checker system.
- Configure the specific location to scan.

CHAPTER 5

Planning

5.1 Project Plan

Advanced detection methods, scalable architecture, automatic reaction mechanisms, real-time monitoring capabilities, and extensive administration tools should all be included in a suggested server-based malware scanner system. Organizations may increase their entire cybersecurity posture and successfully secure key assets by attending to these components, which will improve their capacity to identify, respond to, and neutralize complex malware attacks targeting their server infrastructures. Every project must have its opportunity, expense, schedule, risk management, and communication server procedures determined

before it can be developed. Planning is crucial before a project is started in order to reduce any risks that might jeopardize the developer's ability to complete the project. Establishing goals and objectives, reducing risk, adhering to deadlines, and other things can all be included in project planning. Gantt charts and time boxes, which are frequently incorporated in software project plans, are standard tools for project planning. These technologies are crucial for methodically organizing intuitive car parts tasks in a systematic manner.

5.1.1 Management plan

Describe the project team's duties and responsibilities as well as the project management methodology. To guarantee successful cooperation, set up the reporting and communication routes. Determine the issue-resolution process's stages of decision-making and escalation protocols.

Table 2: Management Planning

No	Task Name	Duration	Start Date	End Date
1	Introduction	5	01/01/2024	05/01/2024
2	Initial Study	4	06/01/2024	09/01/2024
3	Literature Review	4	10/01/2024	13/01/2024
4	Methodology	3	14/01/2024	16/01/2024
5	Planning	10	17/01/2024	26/01/2024
6	Feasibility	15	27/01/2024	10/02/2024
7	Foundation	5	11/02/2024	15/02/2024
8	Exploration	14	16/02/2024	29/02/2024
9	Engineering	30	01/03/2024	30/03/2024
10	Deployment	28	30/03/2024	27/04/2024
11	Testing	15	28/04/2024	12/05/2024
12	Implementation	5	13/05/2024	17/05/2024
13	Critical Appraisal and Evaluation	9	18/05/2024	26/05/2024
14	Lessons Learning	3	27/05/2024	29/05/2024
15	Conclusion	1	30/05/2024	31/05/2024

	Total	151 days	01/01/2024	31/05/2024

5.1.2 Resource Allocation

Determine all of the resources needed for the project, such as staff, tools, and software. Make the right resource allocations depending on the workload and project schedule. Assign tasks and duties to team members and make sure they possess the required knowledge and abilities.

Table 3: Resource Allocation

No	Task Name	Duration	Resource
1	Introduction	5	End User
2	Initial Study	4	Analyst
3	Literature Review	4	Analyst
4	Methodology	3	Analyst
5	Planning	10	Analyst, Designer, Developer
6	Feasibility	15	Analyst
7	Foundation	5	Designer
8	Exploration	14	Designer , Developer
9	Engineering	30	Developer
10	Deployment	28	Analyst, Developer
11	Testing	15	Analyst, Developer, Tester, Users
12	Implementation	5	Analyst, Developer
13	Critical Appraisal and Evaluation	9	Analyst, Tester and Developer
14	Lessons Learning	3	Analyst, Users
15	Conclusion	1	Analyst
	Total	151 days	

5.1.3 Time Boxing

To make development and testing easier, divide the project into discrete time frames or iterations. Determine the length of each time box as well as the assignments and outputs that must be finished in each iteration. For every time box, assign resources and establish distinct objectives.

Table 4: Time Boxing

Time -Box	Task Name	Duration	Resource
TB1	Introduction	5	End Users, Analyst
	Initial Study	4	Analyst
	Literature Review	4	Analyst
TB2	Methodology	3	Analyst
	Planning	10	Analyst, Designer, Developer
	Feasibility	15	Analyst
TB3	Foundation	5	Designer
TB4	Exploration	14	Designer, Developer
	Engineering	30	Developer
TB5	Deployment	28	Analyst, Developer
	Testing	15	Analyst, Developer, Tester, Users
TB6	Implementation	5	Analyst, Developer
TB7	Critical Appraisal and Evaluation	9	Analyst, Tester and Developer
	Lessons Learning	3	Analyst, Users
TB8	Conclusion	1	Analyst
	Total	151 days	

CHAPTER 6
Feasibility

6.1 All possible types of feasibility

6.1.1 Operational feasibility

Feasibility study gauges the possibility that all pertinent elements—such as engineering, planning, lawful and financial considerations—will be taken into account for a project to be completed successfully. Operational feasibility is a type of degree that demonstrates how well a system matures and benefits from the scope set during the definition of opportunities and how well it satisfies the criteria defined during the project or system development requirement analysis phase. Web-based application that uses a server-based virus scan. This is an easy-to-use tool that helps administrators identify PCs. This idea develops malware detection by utilizing offline procedures to ensure operational viability.

6.1.2 Technical feasibility

Hardware	Software
ASUS Laptop, Wi-Fi, Router, Cable, Android Phone	Visual Studio Code, Google Chrome Browser, Windows, MS Word

6.1.3 Technology

Front End	Back end
HTML, CSS, JavaScript.	PHP

6.2 Cost Benefit Analysis

Analyzing the advantages and drawbacks of various project paths, including transactions, activities, business demands, and investments, is done in project management using the idea of Cost Benefit Analysis. This analysis provides an efficient path of action for achieving my objective while expanding a minimal amount of money out of the possibilities that are accessible.

Project Name: Server based malware scanner

Table 5: Cost Benefit

Equipment	1st Year	2nd Year	3rd Year	4th Year	Total
Web Based Application	20000				20000
Domain & Hosting		10000	10000	10000	30000
Software	1000				1000
Internet	2000	2000	2000	2000	8000
Training	5000				5000
Development		5000			5000

Maintenance	10000	10000	10000	1000 0	4000
Total					73,000 BDT.

6.3 DSDM Dynamic System Development Method (DSDM)

Rather than being a particular tool or technology for developing applications, DSDM (Dynamic Systems Development Method) is essentially an organizational structure for agile management of projects and software development. It emphasizes cooperation between development teams and business stakeholders, regular delivery of functional software, and iterative development methods. It is crucial to remember that DSDM does not mandate the use of certain tools or technologies, such as HTML, CSS, or PHP. These tools are widely used in web development and are definitely applicable in the context of a DSDM project.

CHAPTER 7

Foundation

7.1 Some potential approaches

7.1.1 Interview

Interview scheduling is crucial for any project. Through conducting interviews, I am able to get insight into the needs of users, compile information on malware scanners, and interact with knowledgeable personnel to enable users to utilize the server-based malware scanner system to solve malicious problems. I can gather a ton of information about malware problems by conducting interviews. I can learn about all requirements and communication barriers with resource distributions by participating in interviews.

7.1.2 Observation

Visit various types of malware scanner research portion that's why to encourage how to make the system to select the malware problem.

7.2 Specific problem are identification and description

The issues and requirements relevant to protecting servers from malicious software threats are the subject of the problem statement of a server-based malware scanner. incorporates a

variety of challenges, including integration, performance optimization, adaptability, proactive threat identification, and legal compliance. Taking care of these problems ensures the development of a dependable, robust, and effective solution that shields crucial server infrastructures from the ongoing threat posed by malware.

7.3 Possible solution

Generally, developing and managing malware threat scanning technologies such as admin scanners will be simplified by the "Server Based Malware Scanner" effort. Using a log-in procedure, a particular computer address was scanned in this investigation. This system makes use of two scanners: "Normal Scan" and "Deep scan." A standard scan identifies the original malware issue and provides a solution, according to ChatGPT.A deep scanner is capable of identifying nearly any problem, including common issues with scanners, and may provide a solution according to ChatGPT. Additionally, include a customized section pertinent to the scan. To verify a specific code portion, a code checker approach has been employed. Application programming languages (APIs) and secure communication technologies were used to provide a seamless integration between the online system and its infrastructure of support.

7.4 Overall Requirement List

- Functional Requirements
- Non-Functional Requirements.

7.4.1 Functional Requirements

7.4.1.1 Admin

- Admin dashboard.
- Log in and out.
- See all types of files which have been located.
- Change Password
- Two types of Malware scan: “Normal Scan” & “Deep scan”.
- Solved the problem using chat GPT after scanning.
- Configuration for the directory search.
- Security Check.

7.4.2 Non-Functional Requirements

7.4.2.1 Security

Each user on the system has an account, and the only people who may access it are those who have both authorization and a password. Both PHP and JavaScript are used to encrypt the passwords.

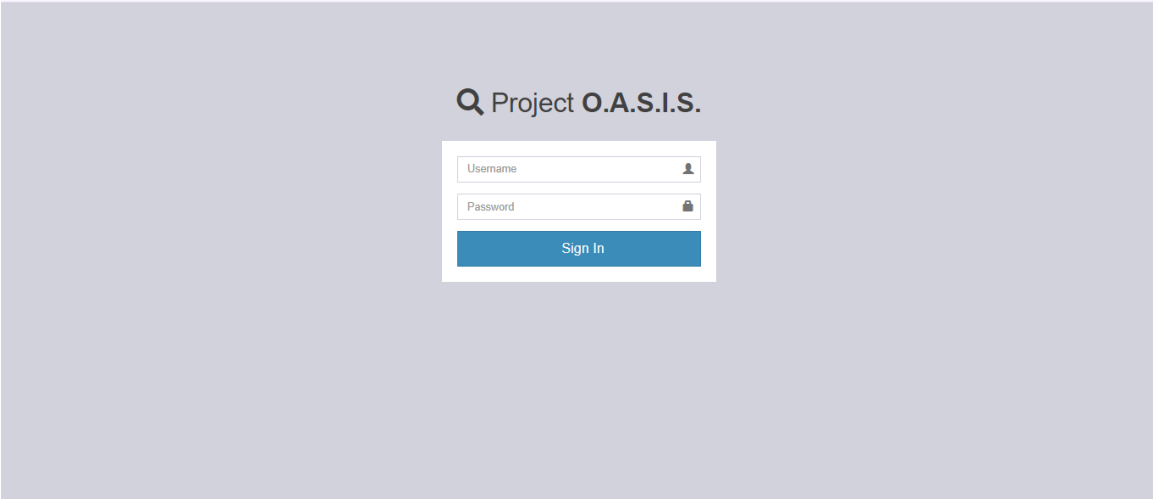


Fig 1: User Login Interfaces

7.4.2.2 Performance

Maintaining and updating records is simple.

7.4.2.3 Availability

Users only need a PC with an Internet connection to access the system from anywhere at any time. Several web browsers are compatible with the system, including Internet Explorer, Mozilla, Opera, and Chrome.

7.4.2.4 User Friendly

The technology is very interesting and has an easy-to-use user interface.

7.5 Which technology to be implemented

My web project is a totally web based application. I am using HTML, CSS, JavaScript, PHP to develop my project.

HTML: Web pages may be created using markup languages like HTML. HTML files may be seen and understood by web browsers. The core of any website is made up of HTML elements. enables the usage of HTML components and graphics, giving you the ability to create engaging content. In addition, it can produce lists, quotes, chapters, links, and titles in [2].

CSS: I may further design the information on our webpage by modifying the fonts, colors, and layouts using CSS. This improves the consistency and aesthetic appeal of our website. Consequently, it makes the website look more appealing. In [2]

PHP: A language that servers are free to use. I am able to construct dynamic web pages with this. PHP-powered web pages . This is a well-liked programming language. In [3]

MySQL Database: An open-source database that is widely used is MySQL. Performance, usability, and dependability define the MySQL database. Prominent online applications including Twitter, Facebook, YouTube, Yahoo, and Facebook are made with this. In [3]

JavaScript: JavaScript is currently a widely used programming language. I also utilize JavaScript as a web development language. A layer of common web technology is developed in this way.[4]

Bootstrap 4: The current version of Bootstrap is 4.0. With Bootstrap 4, you can develop responsive websites with the HTML, CSS, and JavaScript elements you need. As a result, I gave people on my website accounts.[5]

7.6 Recommendation and justifications

Advanced detection methods, scalable architecture, automatic reaction mechanisms, real-time monitoring capabilities, and extensive administration tools should all be included in a suggested server-based malware scanner system. Organizations may increase their entire cybersecurity posture and successfully secure key assets by attending to these components, which will improve their capacity to identify, respond to, and neutralize complex malware attacks targeting their server infrastructures.

In my project using html, CSS, bootstrap to develop the front end. An essential tool for locating and removing malicious software threats aimed at servers in networked environments is the "Server based malware scanner" project. Whereas traditional antivirus software focuses on a particular endpoint, such as a PC or laptop, server-based malware detection tools are intended to protect the systems of computers and their families, which often include important data, applications, and services. Within the system, one module stands out to users: the administrator's dashboard. This project included a log-in mechanism for scanning a particular computer location. This system uses two scanners, referred to as "Normal Scan" and "Deep scan." A typical scan identifies the first malware issue and provides a fix based on ChatGPT. All issues, even those common to scanners, may be discovered by deep scanning, which also provides solutions based on ChatGPT. possess the scan-specific part configured as well. The code checker function has been utilized to verify specific code segments. A dependable MYSQL database backend with XAMPP has been utilized in conjunction with an HTML user interface, CSS style pages, JavaScript, and PHP web application.

CHAPTER 8

Exploration

8.1 Use case

In order to examine both functional and non-functional needs, this part makes use of use-case data and diagrams.

Admin:

Following their login, the administrator can carry out the following tasks:

- Log in and out.
- See all types of files which have been located.
- Change Password
- Two types of Malware scan: “Normal Scan” & “Deep scan”.
- Solved the problem using ChatGPT after scanning.
- Configuration for the directory search.
- Security Check.

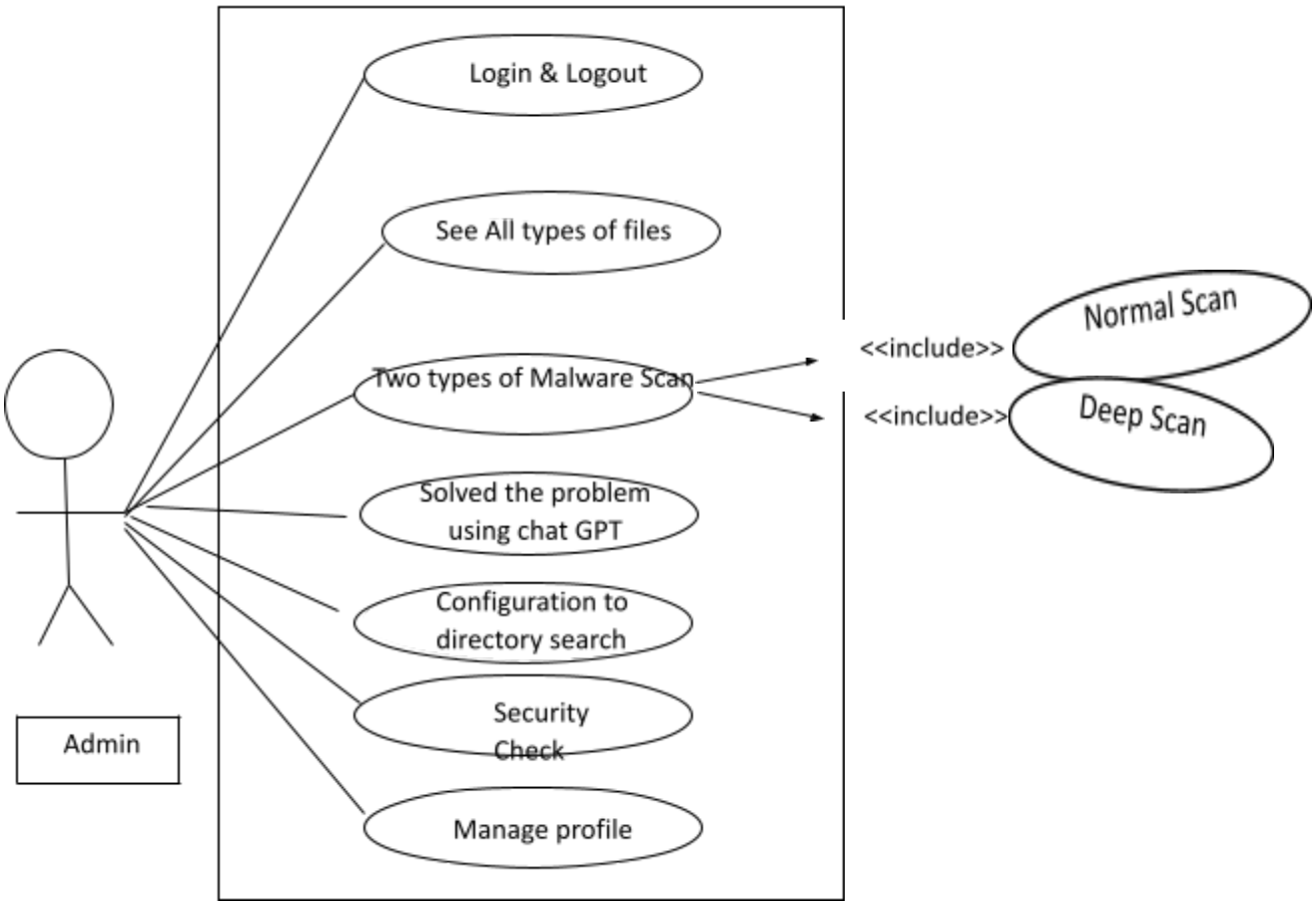


Fig 2: Use case Diagram

8.2 Activity diagram

Describe the dynamic aspects of the system. It's basically a flow chart that shows how one activity leads to another. The activity can be used to describe how the system is operating. This leads to the transfer of control across operations. This is the whole activity diagram for each module: Administrator.

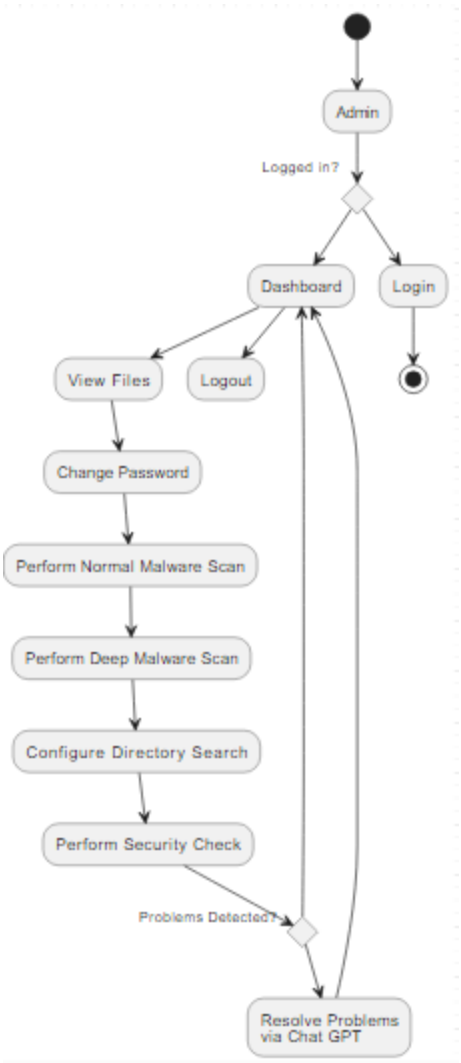


Fig 3: Activity Diagram.

8.3 Requirement Catalog

Functional requirements:

- FR1: User Login for the modules for: admin.
- FR2: See all types of files which have been located.
- FR3: Change Password of the admin.
- FR3: Two types of Malware scan: “Normal Scan” & “Deep scan”.
- FR4: Solved the problem using chat GPT after scanning.
- FR5: Configuration for the directory search.
- FR6: Security Check of the all execution.

Non-Functional Requirements:

- **NFR1:** Each user on the system has an account, and the only people who may access it are those who have both authorization and a password. Both PHP and JavaScript are used to encrypt the passwords.
- **NFR2:** Maintaining and updating records is simple.
- **NFR3:** Users only need a PC with an Internet connection to access the system from anywhere at any time. Several web browsers are compatible with the system, including Internet Explorer, Mozilla, Opera, and Chrome.
- **NFR3:** The technology is very interesting and has an easy-to-use user interface.

User Interface Requirements:

- **UIR1:** Simple and straightforward navigation that makes it simple to access various features and capabilities.
- **UIR2**: responsive design, which adjusts to various device sizes and screen sizes.
- **UIR3:** Icons offer visual clues to help with system usage and comprehension.

Security and Privacy Requirements:

- **SR1:** Safe authorization and authentication procedures to safeguard user information and accounts.
- **SR2:** Secure all the data from malware problems.

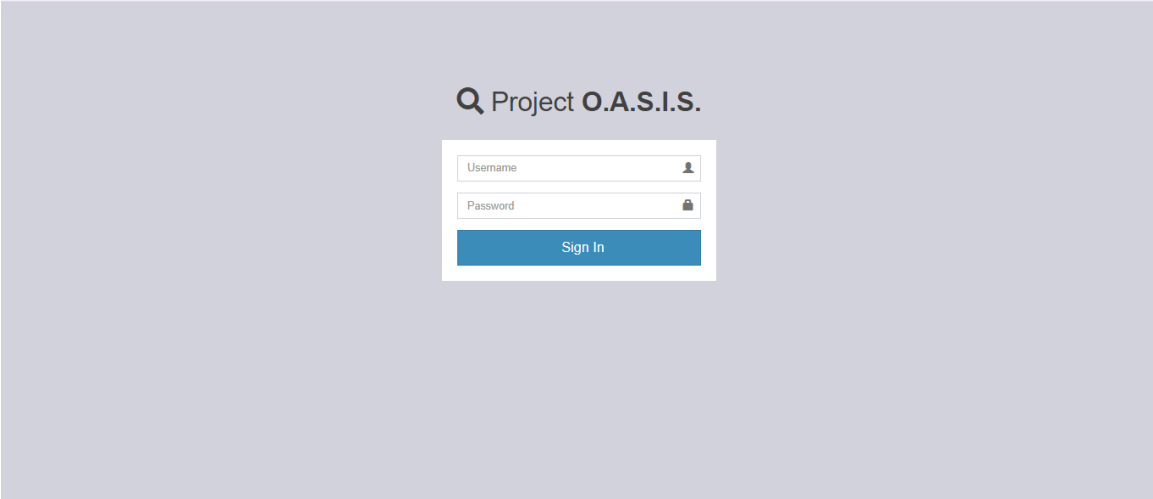
8.4 Prioritized Requirement List (PRL)

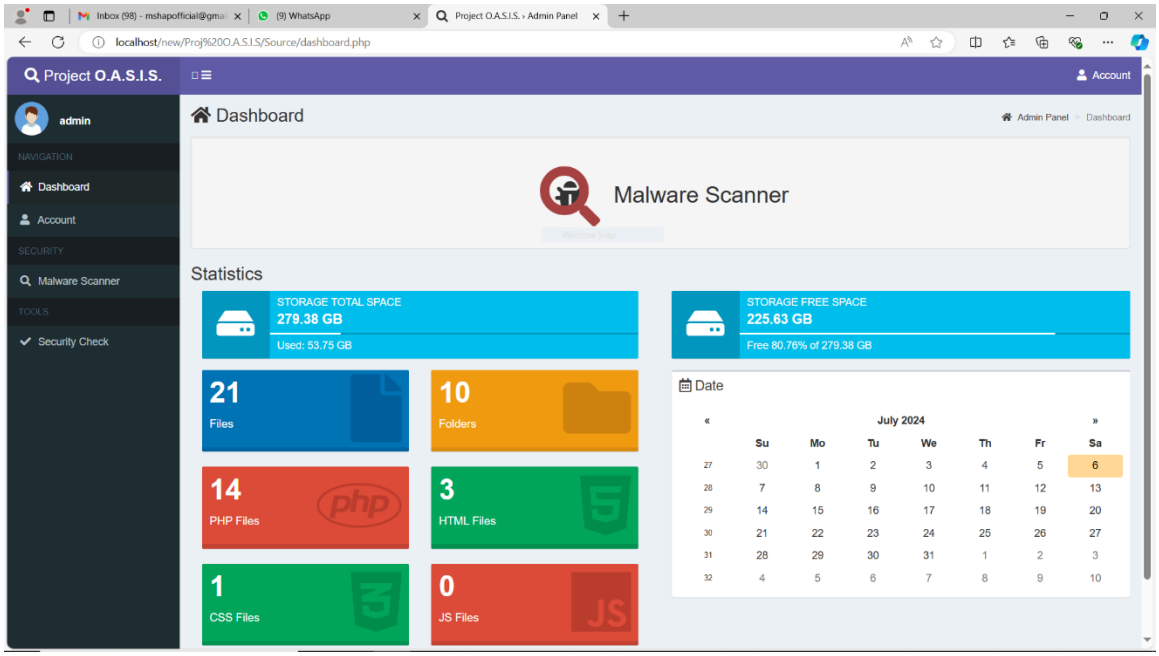
Table 6: Prioritized requirement list

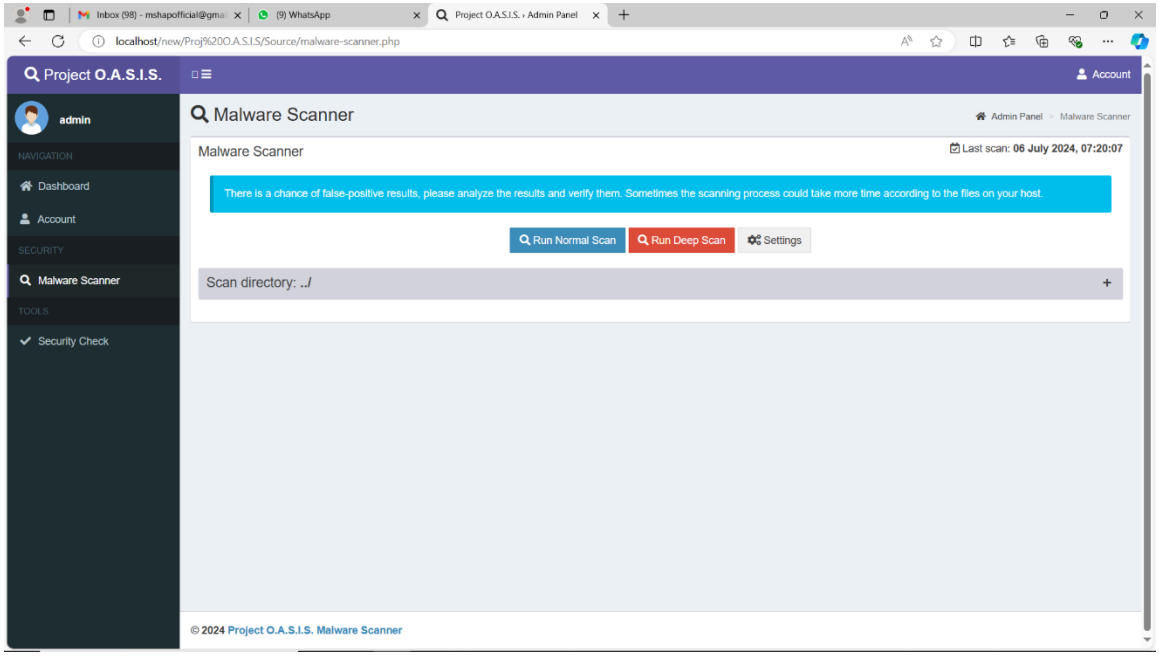
Requirement ID	Requirement Description	Priority	Dependencies	Status	Validation Criteria
RQ1	Admin login functionality	High			Admin can login successfully

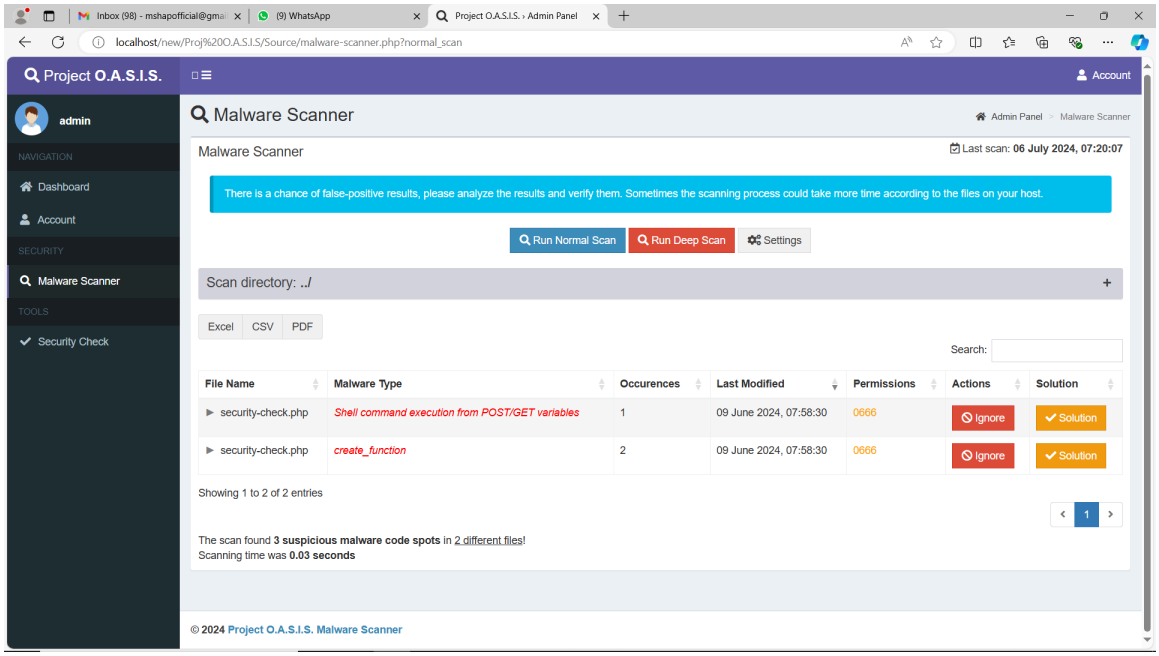
RQ2	View all the data files of located path.	Medium			All types of files viewed.
RQ3	Two types of Malware scan: “Normal Scan” & “Deep scan”.	High			Two types of scan.
RQ4	Solved the problem using c GPT after scanning.	High	RQ3		Problem solved using chat GPT.
RQ5	Security Check of the all execution.	High			Successful for all execution.

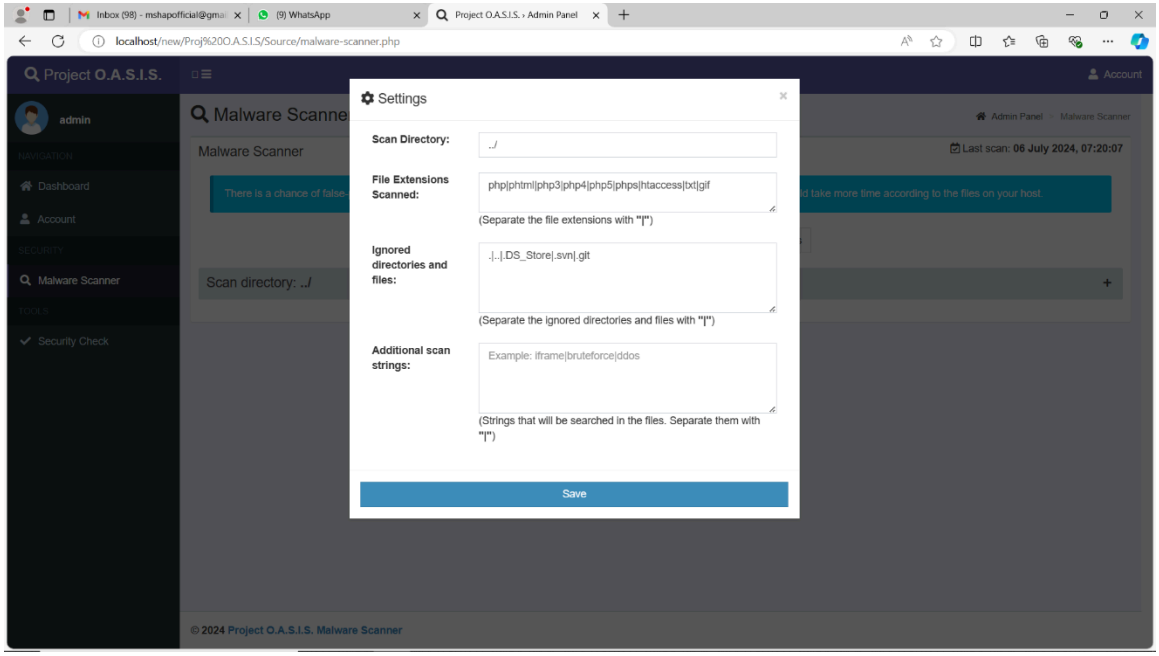
8.5 Prototype of new system

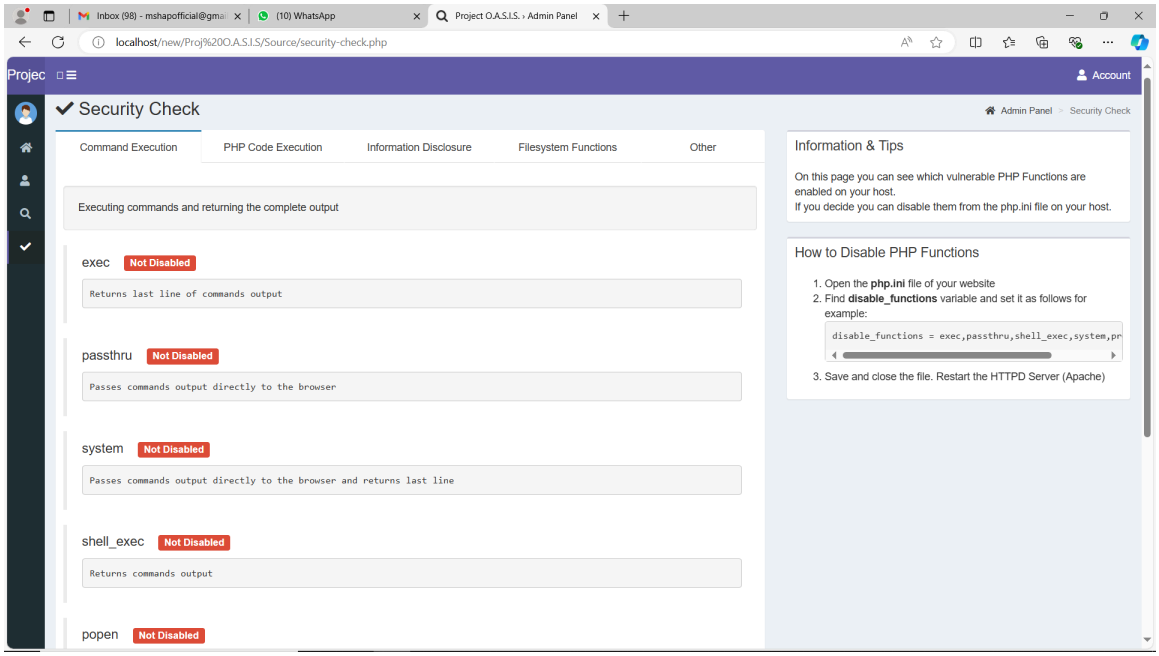


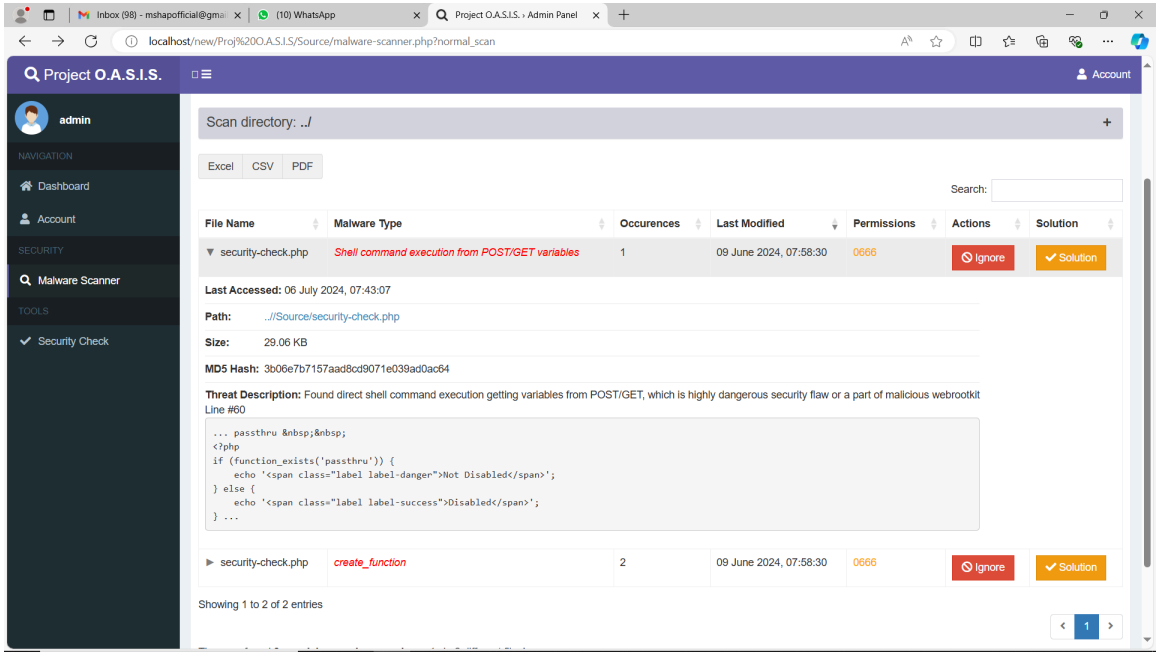


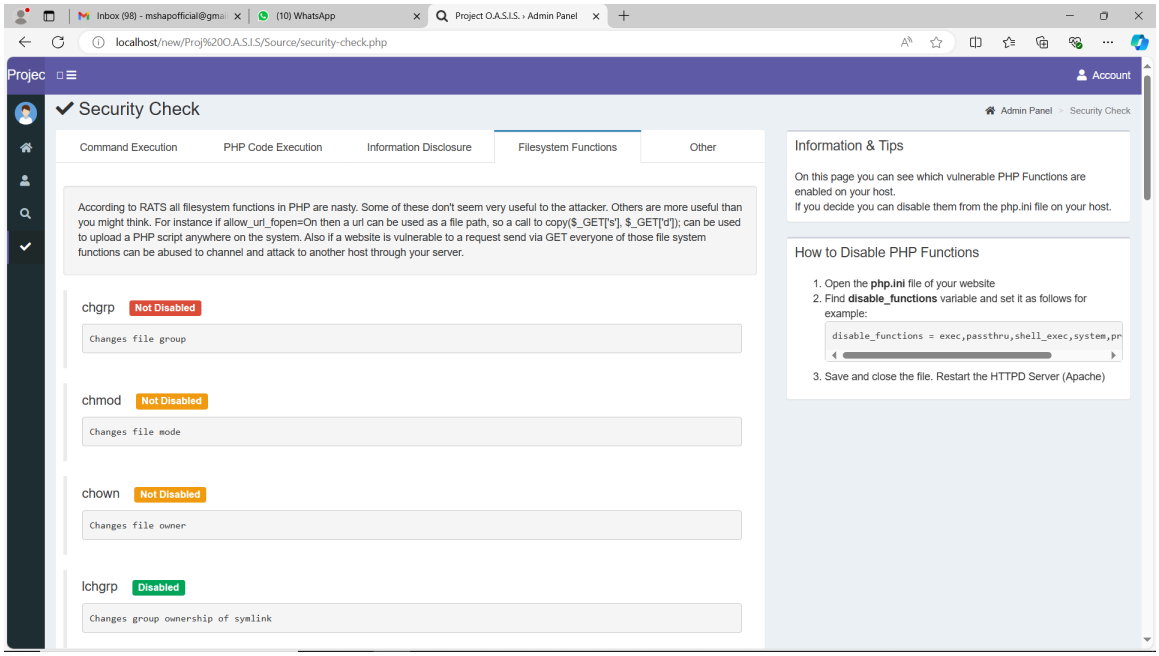


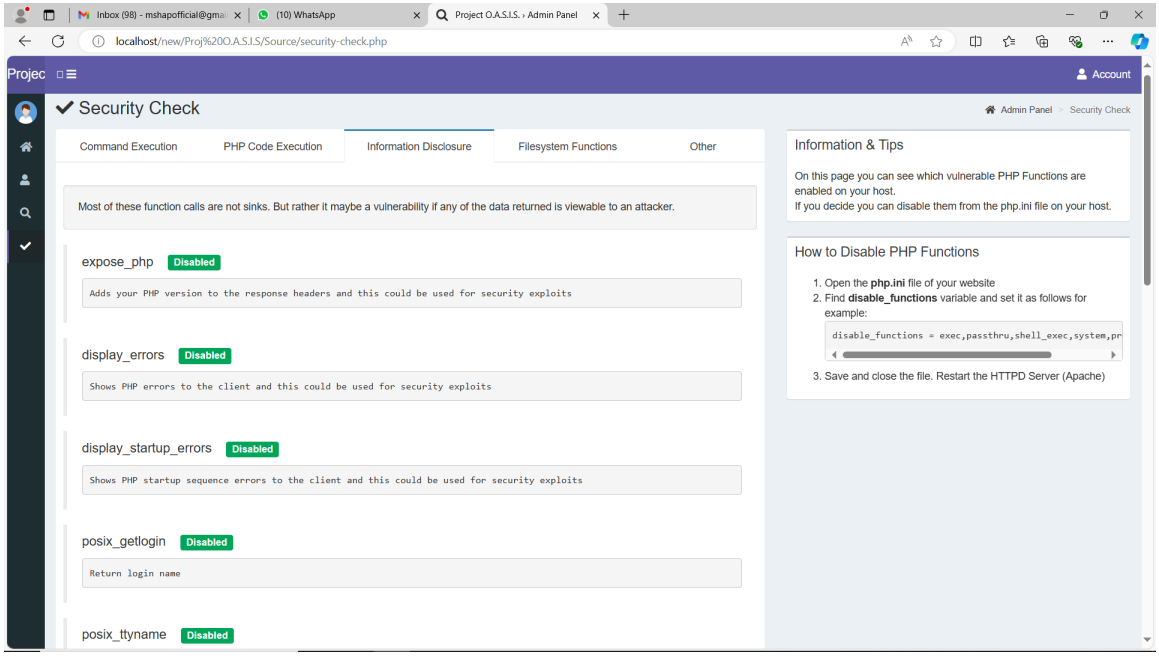


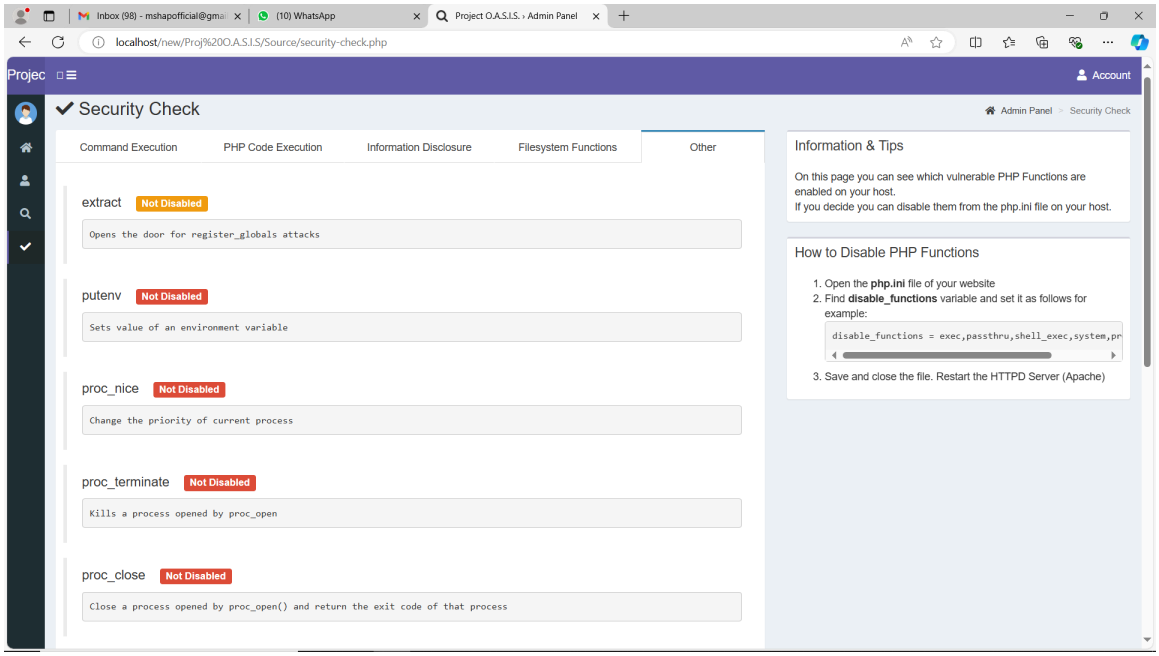


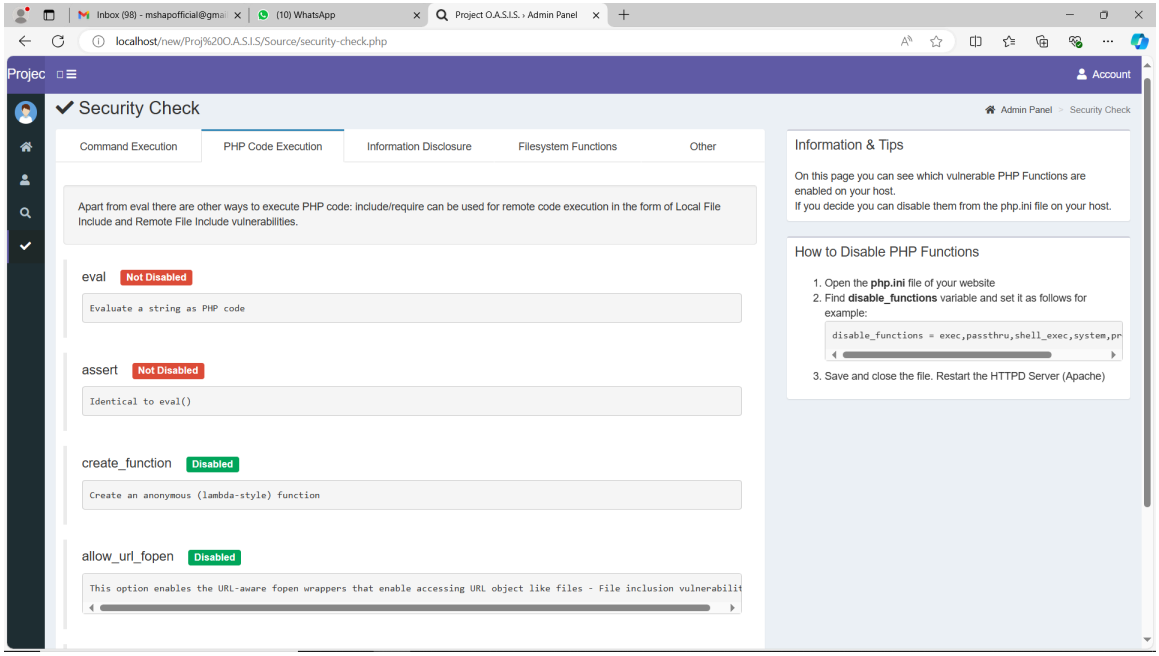












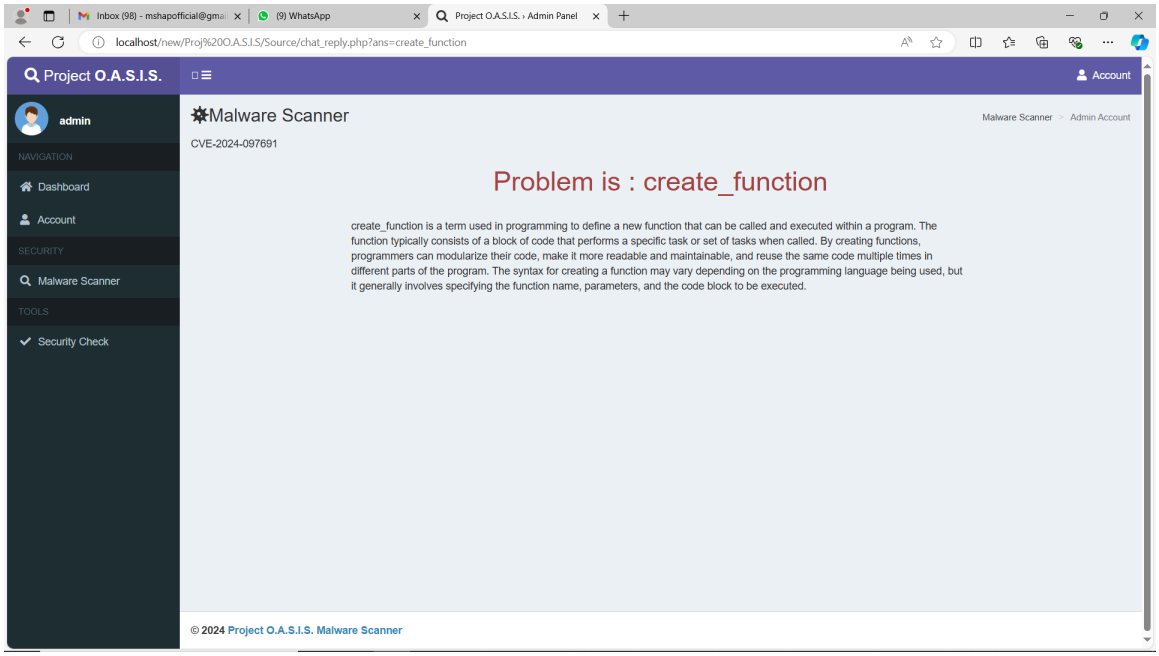


Fig 4: Interfaces for admin account

CHAPTER 9
Engineering

This project has a single dashboard called admin. Features like these on this dashboard Configuration and code checkers are the two sorts of scans. In this experiment, a specific computer location was scanned using a log-in technique. The two scanners used by this system are the "Normal Scan" and the "Deep scan." Based on ChatGPT, a typical scan finds the initial malware problem and offers a fix. A deep scanner can detect almost any issue, including typical scanner problems, and can provide an answer based on ChatGPT. Moreover, offer a customized part relevant to the scan. A code checker method has been used to confirm a certain code section.

Features:

- Admin login to perform scanning malware problems.
- Admin can generate two types of malware scan.
- Solved the Malware problem.
- Code checker system.
- Configure the specific location to scan.

9.1 Class Diagram

A class was built in order to display the interclass connections' source material. Here, the class describes an entity's variables and operations as a single code representation or as a distinct entity within a program.

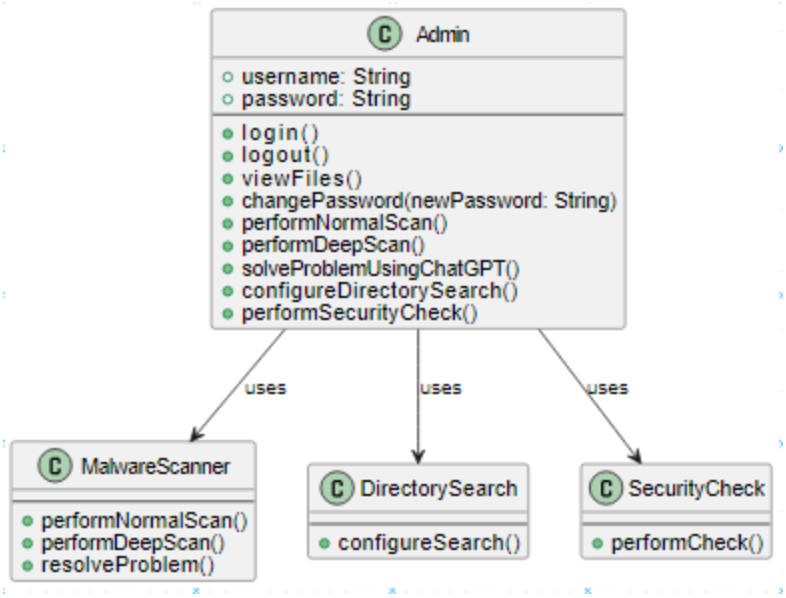


Fig 5: Class Diagram

9.2 ER diagram

Known by several names such as the ER model, ER Diagram, and ERD, institutional methods of communication are a sort of structural application utilized in design. The primary components of the restricted system and the relationships among these organizations are the two primary bits of data that the ERD interchanges and differs in presenting. This is all the information for one module's entity-relationship diagram: Administrator.

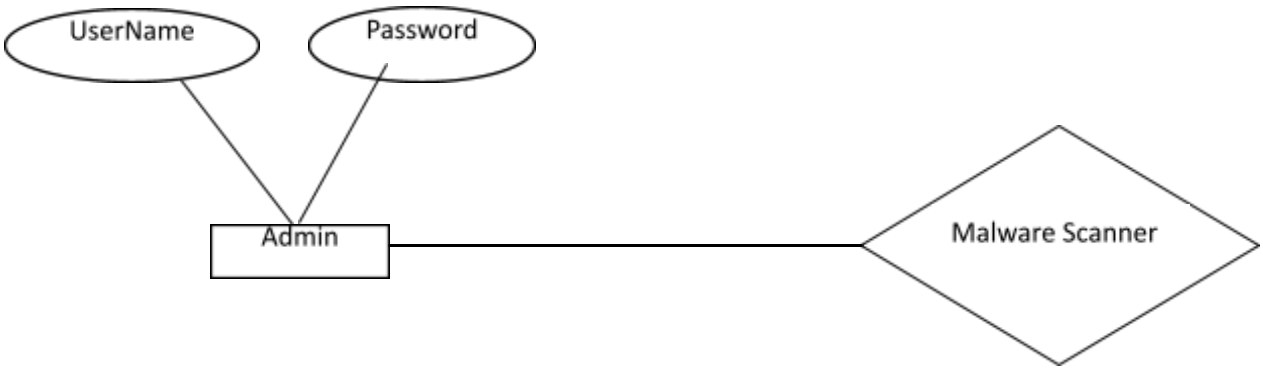


Fig 6: ER Diagram.

9.3 Sequence Diagram

This is all about one module's sequence diagram: Administrator.

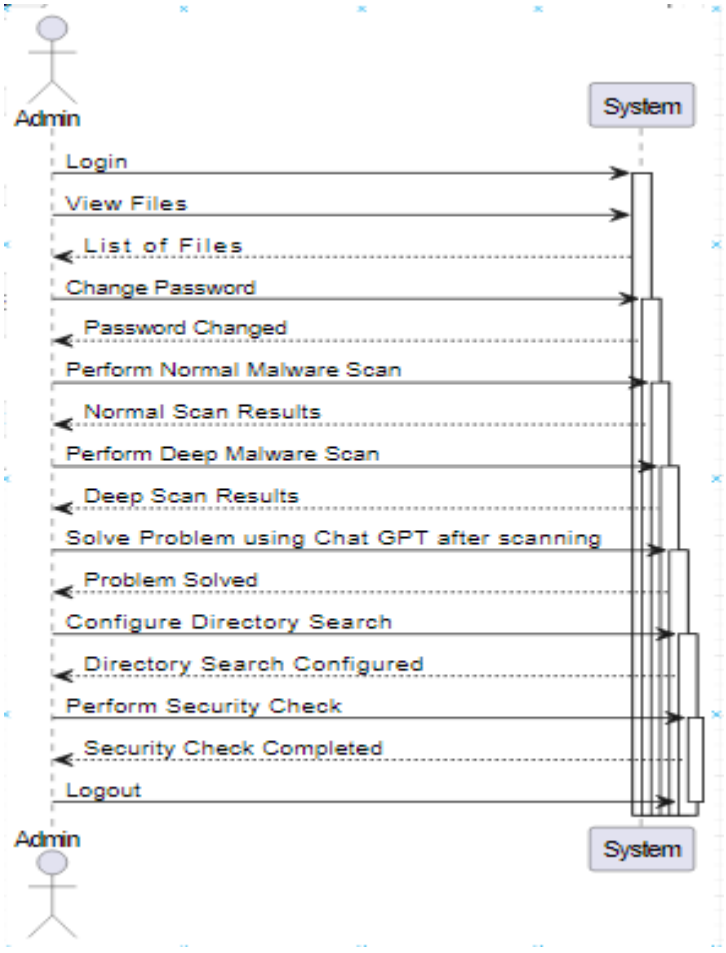
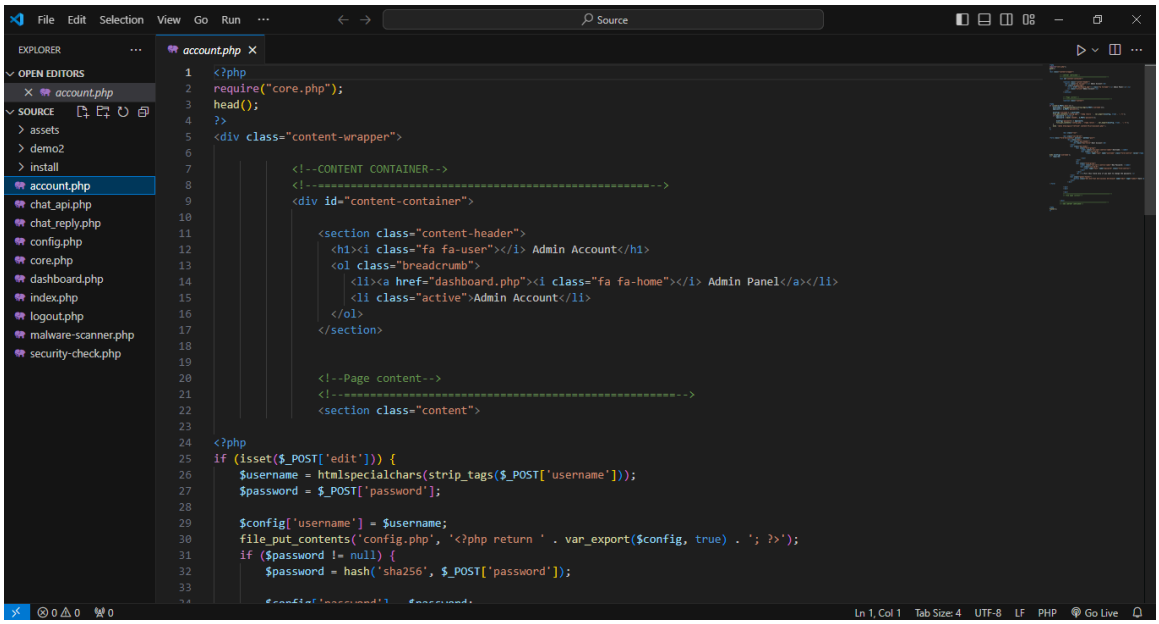


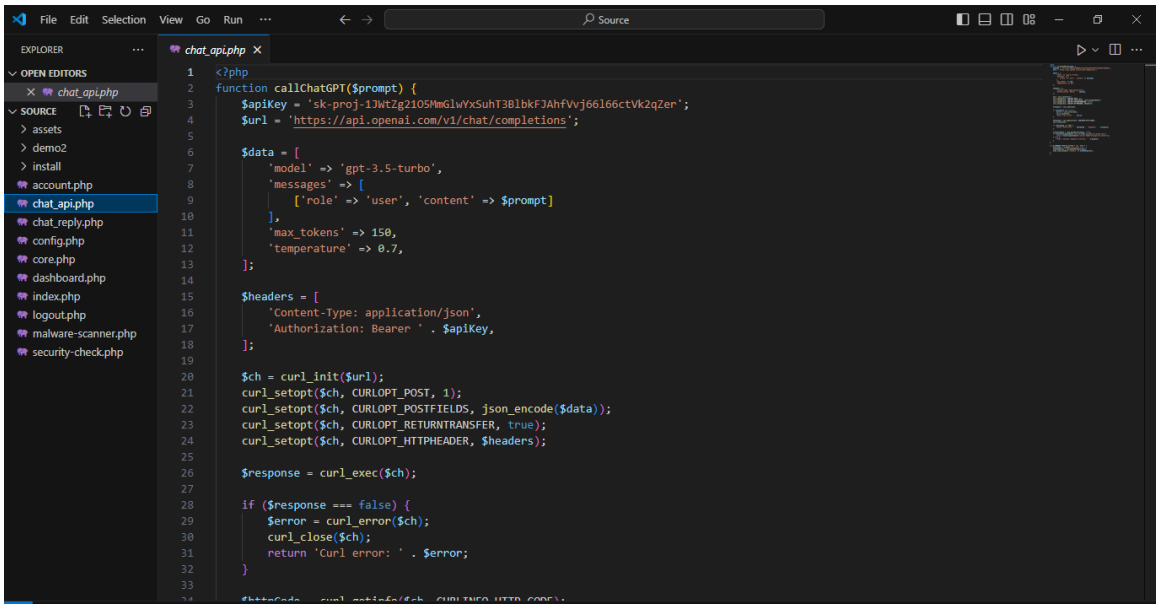
Fig 7: Sequence Diagram

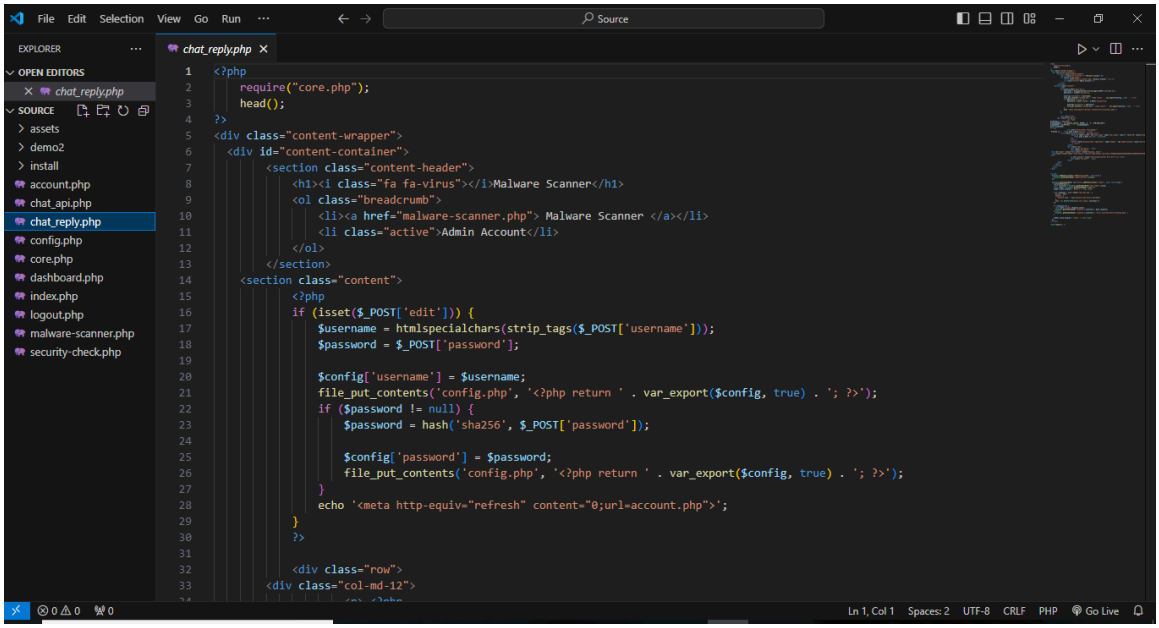
CHAPTER 10

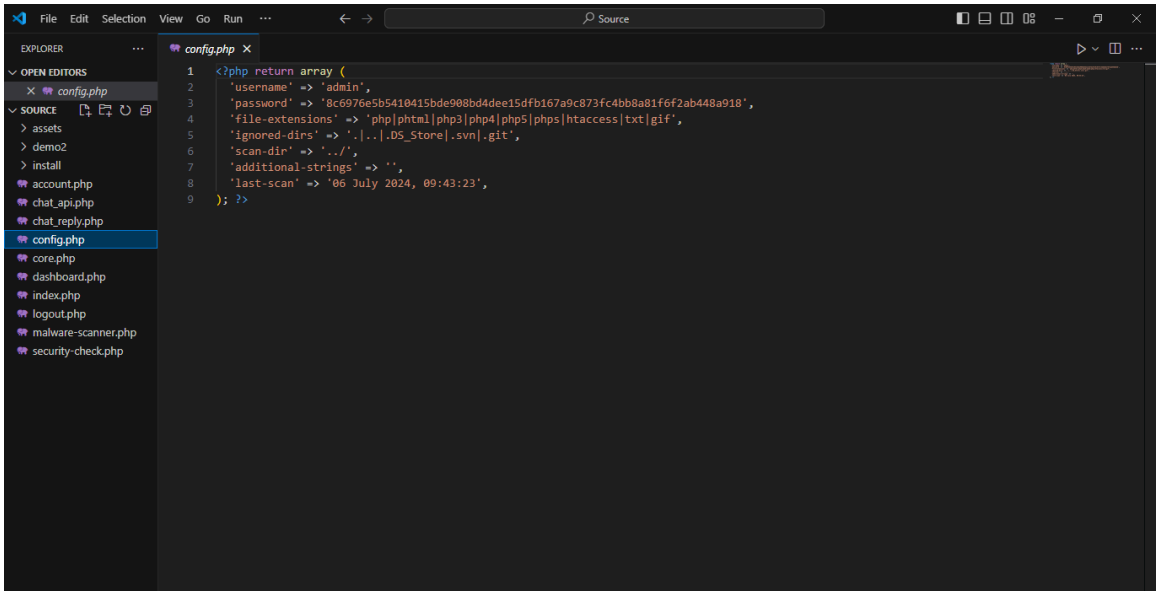
Development

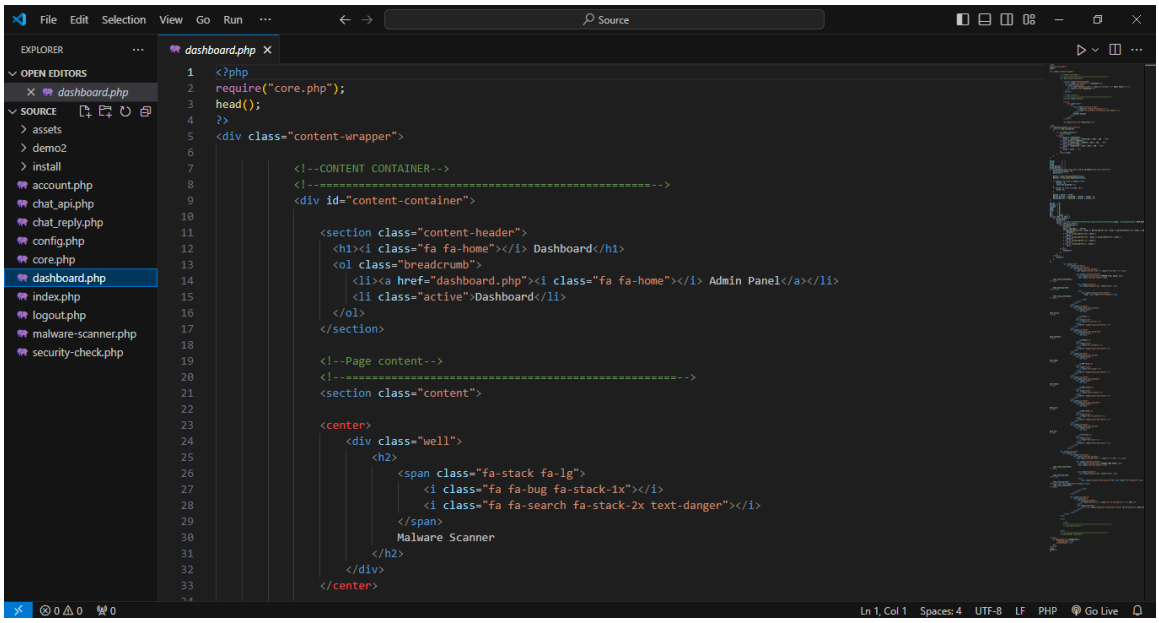
10.1 Core Module Samples

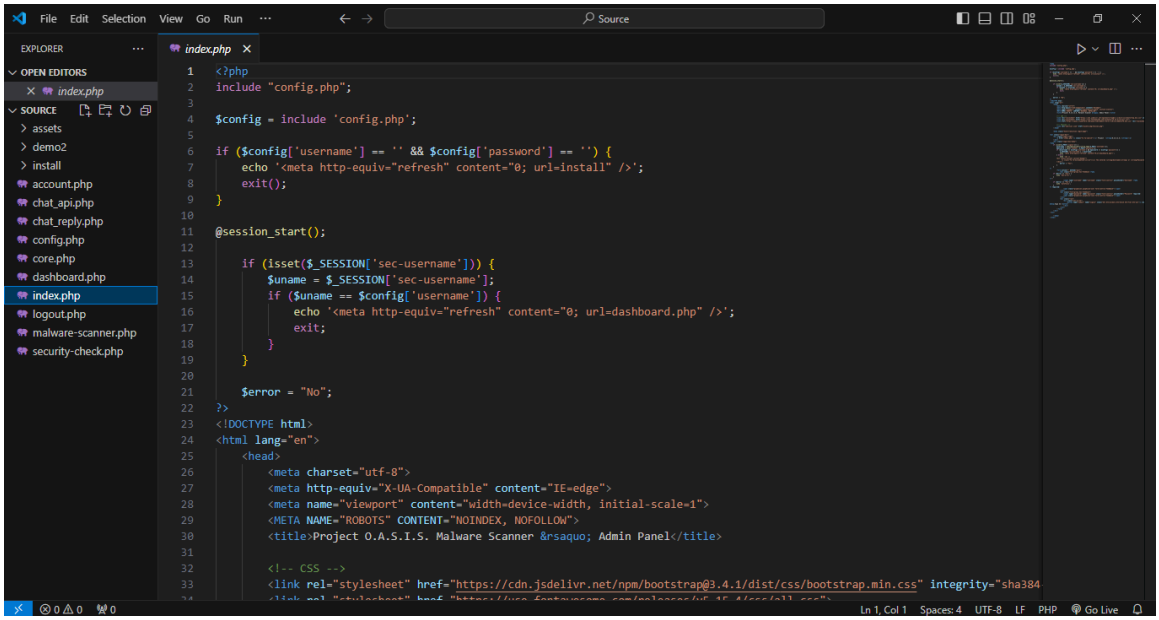


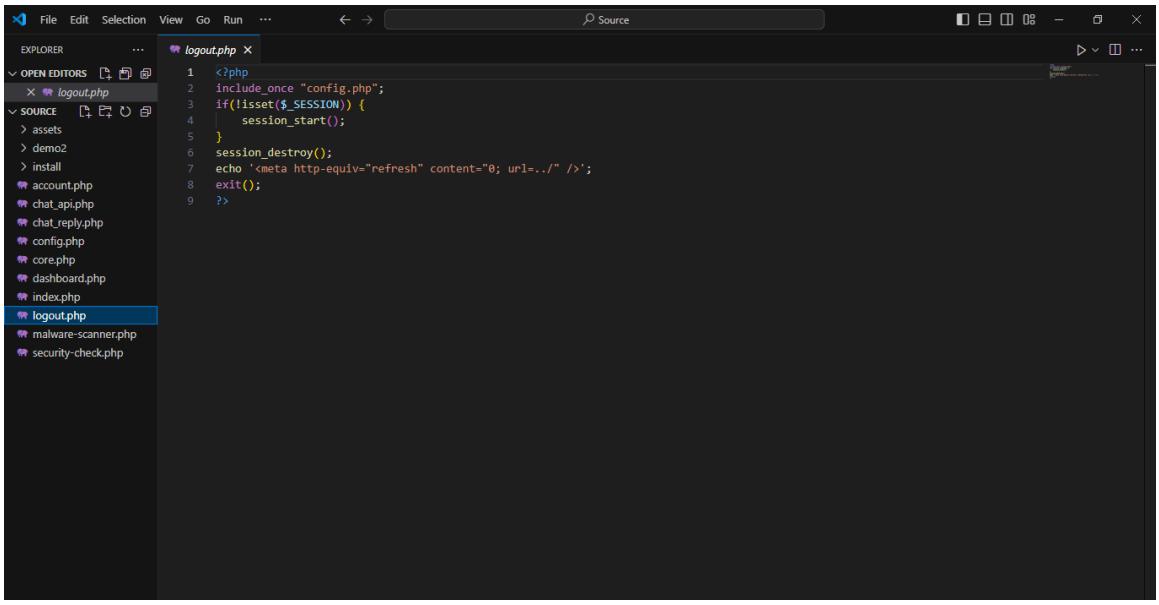


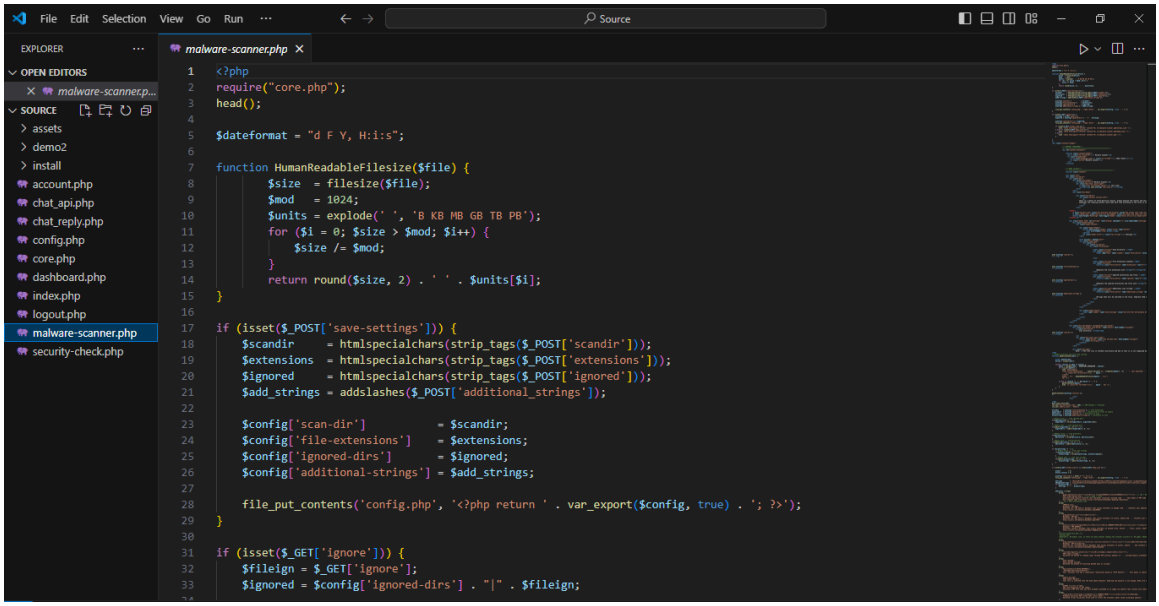












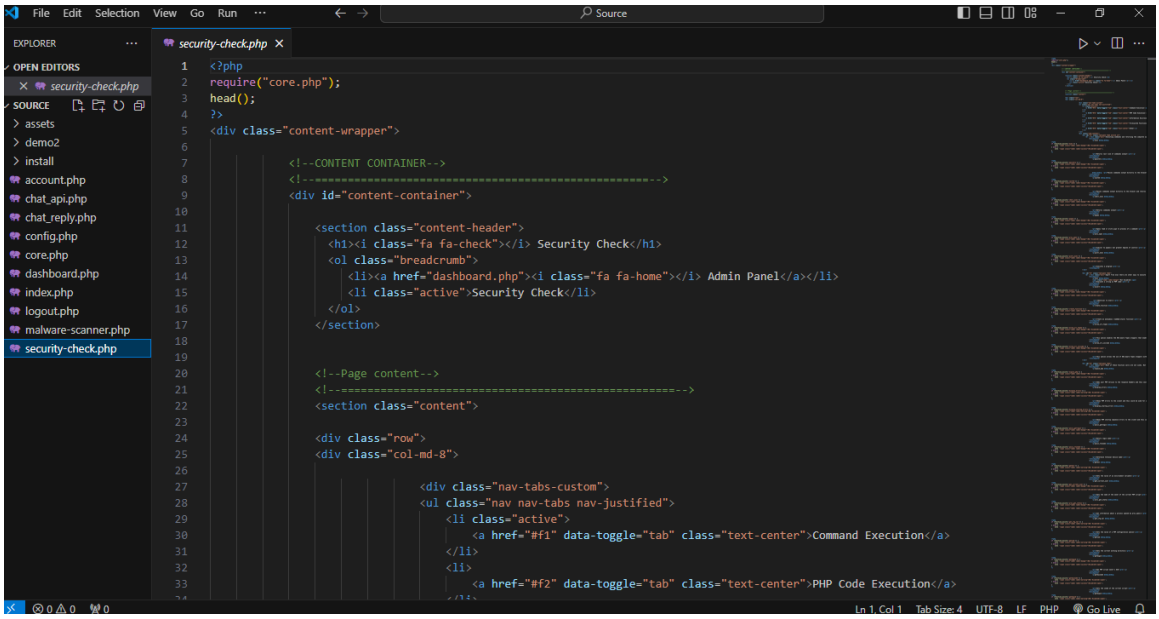


Figure 8: Code Sample

10.2 Probability problem break down

- **Uploading and downloading files:** Uploading files that include malware or executable programs is known as malicious file uploading.

Incorrect File Permissions: Providing unapproved access to files that have been uploaded.

Downloading malicious files: Giving users or additional systems access to compromised files.

- **Consistency problems:** confirming that the scanner is capable of precisely identifying and analyzing malware and vulnerabilities in a variety of web technologies, including PHP, JavaScript, HTML, CSS, and well-known frameworks (including React, Angular, and Vue.js).

  consistency in identifying vulnerabilities, which might change according to the web application's particular technological stack.
- **Objection of performance**: My project is a normal html and php based web application. Processing and scanning huge, complicated websites with several codebases and relationships in an effective way. Encouraging numerous scans at once without sacrificing efficiency or generating resource conflicts.
- **Data security and protection**: Making sure that scanned material is dealt with securely to avoid unwanted access or exposure, especially source code and setting files. Protecting data while it's in transit and at rest by putting robust encryption techniques in place, especially when transferring scan findings or keeping them in databases.
- **Debugging and Testing**: It's critical to find and address issues, such as front-end and back-end defects, while developing online or mobile applications. Automation testing is utilized during the coding process in both the html, css and the JS backend framework php, which aids in finding and fixing bugs.
- **Quality Control and Testing:** Throughout the development process, comprehensive testing and quality control techniques are employed to identify and rectify any possible faults or defects. Functionality, performance, and acceptance by users were all tested. One of the goals of security testing was to identify and address vulnerabilities. The opinions of administrators and stakeholders were solicited in order to make the required changes that would guarantee a dependable, high-quality system.

10.3 Prioritization while developing

Prioritization	Requirements and Explanation
Core Functionality	Issues with server-based malware scanners are to blame for this, examining the two different malware issues to find different kinds of issues. Make sure these essential features are simple to use and properly executed.
UX	A user-friendly interface that is standard for using pure data of all file kinds in the precise location of the computer. To always generate good usability, the usability has to be evaluated and feedback needs to be gathered.
Security and Data Management	Strong data encryption, safe authentication, and sustainable user access controls. To keep improving the usability of the web application, test it with actual users and get their input. Give careful consideration to safeguarding user data and maintaining data integrity. To preserve data security and safeguard admin user information, give careful consideration to putting strong data encryption, secure authentication for users, and suitable access controls into place.
Optimization Performance	The web application's performance was enhanced through performance optimization, which involved improving the network's coding performance and speeding up effective caching practices.
Integration with External Systems	In this work I need to be an admin sign in a database using an external system. Also access with the location of the computer.
Quality Assurance and Testing	I work on a project for a web application. I wrote a huge amount of code for this project. There have been a few mistakes in the coding. I utilized automated testing to find the solution to this bug. The project's quality is checked through the usage of automation testing. Thus, prior to project release, look for and identify issues.

User Feedback and Continuous Improvement	User feedback is an essential feature for every application, as it allows the development team to learn novel concepts and improves the overall efficiency of the program. It's crucial to fix errors and attempt to upgrade the application's functionality after publishing feedback.

CHAPTER 11

Testing

Project Name	Server Based Malware Scanner Web Application		
Name of product	Server Based Malware Scanner App		
Product description	Server Based Malware Scanner App		
Project description	HTML, CSS, JS, PHP based web application		
Project duration	Project Type	Testing/ Verification	
	Start date	End date	

11.1 Test Plan Acceptance

Specify the goals and parameters of the test. Determine the important parties and get their consent before implementing the test plan. Clearly define each testing phase's acceptance criteria.

11.2 Unit Testing

When performing unit testing, test each module independently and integrate the entire framework as a whole. Unit testing helps to focus verification efforts on the construction of software, the smallest unit of each module. An alternate phrase for this is module testing. Every module in the system is tested separately. Furthermore, make sure this technique works with all browsers.

Unit Test Case-01

Test Case Name	Unit Test
Test Class	Authentication
Test description	Testing the authentication of the system, ensuring the system is secure.

Source of Data	Test Steps	Expected Result	Actual Result
Admin	Input valid credentials “admin” (by default).	The app should authenticate the admin and redirect to the dashboard page.	The app successfully authenticated the user (admin) and redirected to the dashboard page.

Unit Test Case-02

Test Case Name	Unit Test		
Test Class	Scanning		
Test description	Testing the Scanning capabilities of the app		
Source of Data	Test Steps	Expected Result	Actual Result
Admin	Input a valid destination in the server’s file structure, and press “Normal Scan”	The app should scan the given destination and show a report of any flagged files.	The app successfully scanned the given destination and showed a report of the flagged files.

11.3 Validation Testing

Software testing uses validation and verification procedures to make sure a system meets requirements and performs as intended. Another name for it might be assurance of software quality.

11.4 Integration Testing

Integration testing addresses the challenges around the two issues pertaining to inspection and program generation. After integrating the program, a set of High order benchmarks are executed. The main objective of this testing technique is to use unit-tested modules to build a program structure according to design criteria.

Integration Test Case-01

Test Case Name	Integration Test		
Test Class	Deep Scanning		
Test description	Testing the Deep Scanning capabilities of the app		
Source of Data	Test Steps	Expected Result	Actual Result
Admin	Input a valid destination in the server's file structure, and press "Deep Scan"	The app should scan the given destination and show a detailed report of any flagged files. Resulting in more entries in the result.	The app successfully scanned the given destination and showed a detailed report of the flagged files. Which resulted in more entries in the result.

11.5 TEST CASES

Table 7: Test Case

Case Id	**CASE NAME**	**Expected Result**	**Actual Result**	**Result (Pass/Fail)**
1	User Name	Incorrect or empty names will result in a notice being shown.	Incorrect or empty names will result in a notice being shown.	Pass
2	Password	A notification will appear if the password is invalid or empty.	A notification will appear if the password is invalid or empty.	Pass
3	Scan	Scanning all the parts of a specific location and detecting malware.	Scanning all the parts of a specific location and detecting malware.	Pass

4	Solved Problem	Given a solution according to the malware problem.	Given a solution according to the malware problem using ChatGPT.	Pass

CHAPTER 12

Implementation

12.1 Training

User	Training	Time	Comment
Admin			

12.2 Big Bang Implementation

The Big Bang method of execution does not use a pilot program or parallel implementation; instead, the full system is deployed all at once. It is imperative to offer thorough training to all web app users, including admin for malware scan purposes. In this project there is one module: Admin. Numerous topics, including two malware scanner, solution of malware problem, responding to user queries, and any other pertinent features and capabilities, should be covered in the training program. Ensuring that everybody using it is conversant with the app & know how to utilize its features efficiently is crucial.

12.3 Scaling

An online application malware scanner must be scalable in order to efficiently manage growing scan volumes, a range of web application complexity, and a variety of vulnerability kinds. Scale the web-based malware scanner efficiently to accommodate increasing scan volumes, a variety of application designs, and a range of scanning complexity levels all while preserving security, dependability, and efficiency.

12.3.1 Design of scaling

A microservice should be created for each task the scanner handles, such as reporting, vulnerability detection, static and dynamic analysis, and analysis. Provide the architectural and development team with training on creating scalable and effective systems. Being aware of elements like caching, horizontal scaling, database optimization, and performance monitoring is part of this.

12.3.2 Testing Performance

To determine whether the app can withstand higher loads, do performance tests. During testing, teach the participants how to spot performance problems and blockages and adjust the program accordingly.

12.4 Load Balancing

To improve the overall efficiency of this application, load balancing is required.For this application, a few crucial considerations must be made.

CHAPTER 13
Critical Appraisal and Evaluation

13.1Objective that could be met

The following might be the goals of the server-based malware scanning application:

- Determine and identify the different kinds of malware that are contained in web-based files and components, such as Trojan horses, bugs, and ransomware.
- Look for vulnerabilities that an attacker may exploit in the third-party libraries, frameworks, and codebase of the web application.
- In order to identify and neutralize threats before they have a chance to compromise the application or server, perform real-time screening of incoming documents and components.
- Scheduled scans should be put in place to systematically check the whole web application for malware and security vulnerabilities.
- Assure compatibility with cutting-edge technology and contemporary web development methodologies to offer thorough coverage.

13.1.1 Success rate against each objective

A web-based malware scanner's efficacy in detection, mitigation, and response must be evaluated in order to meet each goal. Determine what proportion of recognized malware and defects the scanner found out of all the web apps it examined.

13.1.2 How much better could have been done

Enhancing a web application malware scanner's identification, efficiency, usability, and overall efficacy are all part of improving it. Expand scanning capability to provide full coverage of new web frameworks and technologies. Improve support for containerized apps, microservices, and serverless designs.

13.1.3 Why it could not be done

Creating detection systems that are capable of accurately identifying every malware variation calls for ongoing research, upgrades, and advanced methods such as behavioral analysis and heuristic analysis. Despite developments, certain new malware variations and zero-day vulnerabilities could remain unnoticed until they are located and examined.

13.1.4 Which objectives have been missed

For the reasons outlined above, it is possible that the application did not meet its goals regarding the successful rescue rate, database accuracy, and public awareness campaign reach.

13.1.5 Why these objectives have missed

These goals might not have been met because of practical difficulties, a shortage of specialist malware scanner resources in some areas, and all data of removable systems for detecting malware.

13.1.6 What could have been done to complete those objectives

Several tactics and techniques may be used to successfully fulfill the goals of a web application malware scanner, including malware detection, vulnerability identification, compatibility with contemporary technologies, and scheduled and real-time scans. Use real-time tests using lightweight scanning techniques to reduce the impact on application efficiency. Use parallel processing and effective resource management for scheduled scans to guarantee thorough coverage without using excessive amounts of resources.

13.1.7 How better is the features of the solution

Enhancing a web application malware scanner's characteristics includes making it more capable of integration, detection, prevention, and usability. Allow for the real-time analysis of incoming elements and documents to detect threats right away.

Set up periodic scans to methodically perform routine checks for malware and vulnerability

throughout the whole online application. Scanner performance should be minimized while maintaining thorough coverage by optimizing scanning algorithms.

13.1.8 Which features could not be touched

Although attempts are made to remain ahead of evolving threats, it may take significant resources to conduct continuing research, development, and interaction with cybersecurity specialists in order to continually adapt to new attack vectors, tactics, and strategies.

13.1.9 Why these features could not be touched

The team may not have been able to completely integrate these sophisticated features due to a lack of funding and development time.

13.1.10 What could be done to touch those features

The project might receive more financing and a specialized development team in order to incorporate the unfinished features. Implement strategies like load balancing, caching, and parallel processing to increase the performance of your scanning methods. These adjustments can lessen the impact on application efficiency and resource usage during planned and real-time scans. Frequent profiling and performance testing might find additional areas for development.

13.2 Objectives totally not met / touched

It is plausible that certain goals, such as obtaining Identification and combating complex, persistent hazards that use cutting-edge strategies to avoid detection and keep access to systems for an extended period of time. APTs frequently use covert tactics, such data exfiltration and lateral movement, making it challenging to identify them without specialized equipment and ongoing surveillance were not fulfilled as a result of major obstacles and constraints.

13.2.1 Why it could not be touched

These goals may not have been met because of significant financial or logistical obstacles, inadequate infrastructure, or low user uptake.

13.2.2 What could have been done

The project team may have obtained more funds and carried out in-depth feasibility studies in order to accomplish these unmet goals. This "Server Based Malware Scanner" initiative will, in general, make administering malware threat scanning technologies—like admin scanners—fewer complicated. An inquiry was conducted using a log-in process to scan a specific computer address.

13.2.3 Including software and documentation

To guarantee the efficacy and efficiency of the software development process, frequent testing, debugging, and user feedback should have been incorporated. In order to enable simple comprehension and future upgrades, the documentation had to have been thorough and include user manuals, technical directives, and service instructions.

CHAPTER 14

Lessons Learned

14.1 Pre-project

For the server-based malware scanner web application, I would normally specify the project's goals, parameters, and specifications at the pre-project stage. This might entail figuring out the features and functions the program needs, as well as performing market research and defining the intended user base. In addition, I might need to assemble staff and technological resources in addition to putting together a project strategy with deadlines and milestones. Staff and software technologies are included, as is creating a comprehensive project plan with deadlines.

14.2 Review

To evaluate the project's progress, spot departures from the plan, and make sure everything is moving forward as planned, frequent reviews are crucial. Through these assessments, I can assess whether the web application is accomplishing its objectives and whether it needs to be improved or adjusted. Informed decision-making and the identification of areas for development may be facilitated by regular input from stakeholders, including users.

14.3 Lessons Learned

I've studied a lot before creating this web application. I studied the design and analysis of systems before creating this application. This is the most difficult phase of system development. Accurate software cannot be produced without analysis. I then study the programming language JS. Learning is challenging Even though PHP is quite popular, learning and using it in its basic form can be challenging. In order to be considered for future projects, I identify my strengths and limitations as soon as I begin the learning process.

14.4 Problem Faced

I had a lot of difficulties when working on the project. I'll give you a few instances, involving how difficult it is to locate knowledgeable employees who are well-versed in solving malware-related issues at computer locations and how challenging it is to obtain reliable information regarding parts linked to malware scanners.

14.5 Problems That are solutions

It is difficult to tackle software tasks like this. Despite the fact that I create web applications. I encountered some typical issues when working on the project, but they are the really difficult parts. However, coming up with effective solutions is challenging. Acquiring accurate knowledge on every aspect of malware scanners, as well as remedies for malware issues, is a formidable task. Technical problems may have remedies, such as consulting an expert or looking at the application architecture. The major issue is a communication breakdown that the chat GPT system has to address in order to prevent viruses.

CHAPTER 15

Conclusion

15.1 Summary of the project

The server-based malware scanner web application for virus detection improves the present one in this digital era. One module inside the system—the dashboard for the admin—allows users (admins) to be separated from other users thanks to this new, necessary technology. A log-in mechanism was used in this project to scan a particular computer location. This system makes use of two scanners: "Normal Scan" and "Deep scan." Both standard and deep scans identify the first malware issue and provide a remedy based on ChatGPT.

15.2 Goal of the project

The goal of the present project is to develop a server-based malware scanning system that parks automobiles intelligently using technology. This project has a single dashboard: admin for malware detection and scanning across several computer locations. The scope of a server-based malware scanner covers compliance requirements, central management, efficiency enhancements, interaction with pre-existing infrastructure, and the detection and elimination of malware threats. The best malware scanner in the world is sent to my machine, helping us solve the malware issue and troubleshoot our system overall.

Subsequent sections of the report in question will go into greater detail on the methodology, application specifics, assessment outcomes, and conclusions of the research. This project is to investigate how tailored feature desires for students might revolutionize the dissemination of higher learning technologies, as well as provide a thorough examination of the built online platform and malware detection.

15.3 Success of the project

This initiative has been a big success. The primary acts are:

- Admin can generate two types of malware scan.
- Solved the Malware problem.
- Code checker system.
- Configure the specific location to scan.

15.4 Documentation

The following phases, tasks, and plans were probably addressed in the documentation:

- **Pre-Project Documentation:** This might consist of preliminary requirements collecting, feasibility studies, and project proposals.
- **Project plan:** The Project Plan usually means a written document which describes the goals, parameters, schedule, needed resources, and risk control techniques for the project.

- **Technical Specifications:** Comprehensive guidelines outlining the features, functionality, and architecture of the web application for a server-based malware scanner.
- **User Documentation**: Provides guidelines and instructions on how to use the program efficiently for all stakeholders, including the administrator who searches for malware.
- **Testing and Quality Assurance Instruction**: Recording test strategies, scenarios, and outcomes to guarantee the application satisfies quality requirements.
- **Deployment and Maintenance plans**: Plans for application deployment, maintenance, and upgrades, together with documentation detailing the procedures involved, are included.

15.5 Value of the project

The current project uses technologies to create a malware scanning system that is server-based, capable of detecting malware and intelligently resolving problems. This project's administrators are based on a single dashboard that handles the distribution of sophisticated malware scanners. It is fully in charge of all aspects of malware issues. The GPT discussion explores malware problem-solving in further detail.

15.6 My Experience

I have experienceed in project management, development, design, and other related tasks that I may apply to the server-based malware scanner web application project. With the help of my experience, I can now effectively manage project timeframes, collaborate with stakeholders, overcome technical obstacles, and put solutions in place that are tailored to the unique requirements of a small server cybersecurity management. In order to further improve my professional talents, I have also enhanced my documentation, teamwork, and problem-solving skills.

References

[1] K.N. Durai and K., Priyadharsini 2014, “A survey on security properties and web application scanner.” *International journal of computer science and mobile computing*, *3*(10), pp.517-527. Luke Welling, L., 2000. *PHP and MySQL Web Development*. 4th ed.

[2] Duckett, J., 2001. *Web Design with HTML, CSS, JavaScript and jQuery*. 1st ed.

[3] Luke Welling, L., 2000. *PHP and MySQL Web Development*. 4th ed.

[4] W3schools.com. 2021. *JavaScript Tutorial*. [Online] Available at: <https://www.w3schools.com/js/default.asp> [Accessed 17 December 2021].

[5] W3schools.com. 2021. *What is Bootstrap*. [Online] Available at: <https://www.w3schools.com/whatis/whatis_bootstrap.asp> [Accessed 17 December 2021].

192-16-449

by 2_ Sajid Hasan Akhand

Submission date: 02-Oct-2024 12:29PM (UTC+0600)
Submission ID: 2472458475
File name: 192-16-449_Final-for-plaigarism-report.pdf (2.1M)
Word count: 8609
Character count: 49835

192-16-449

ORIGINALITY REPORT

9%	8%	0%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	3%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	3%
3	Submitted to University of Greenwich Student Paper	1%
4	Submitted to Gulf College Oman Student Paper	<1%
5	Submitted to Xavier University Student Paper	<1%
6	Submitted to Central Queensland University Student Paper	<1%
7	Submitted to Brunel University Student Paper	<1%
8	Submitted to Visvesvaraya Technological University Student Paper	<1%
9	Submitted to CSU, San Jose State University Student Paper	<1%

10	Submitted to University of Southern Queensland Student Paper	<1%
11	hdl.handle.net Internet Source	<1%
12	microgliss.github.io Internet Source	<1%
13	www-dweb-cors.dev.archive.org Internet Source	<1%
14	www.coursehero.com Internet Source	<1%
15	www.theseus.fi Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

192-16-449

GRADEMARK REPORT

FINAL GRADE

/0

GENERAL COMMENTS

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

PAGE 25

PAGE 26

PAGE 27

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

PAGE 35

PAGE 36

PAGE 37

PAGE 38

PAGE 39

PAGE 40

PAGE 41

PAGE 42

PAGE 43

PAGE 44

PAGE 45

PAGE 46

PAGE 47

PAGE 48

PAGE 49

PAGE 50

PAGE 51

PAGE 52

PAGE 53

PAGE 54

PAGE 55

PAGE 56

PAGE 57

PAGE 58

PAGE 59

PAGE 60