Software Design Document

Group Name: *TeamOne*

ID	Team member Name
7741923	Abdulhayye Maricar
7811056	Abdul Rahim Kalsekar
8061634	Ali Sina Mohammad Arif
7869472	Mansoor Kalemzai
8044661	Varun Tulsiani

UNIVERSITY OF WOLLONGONG IN DUBAI

Proposal Cover Sheet

Subject Code: CSIT321
Subject Name: Project
Submission Type: Design document (High & low level design)
Project Title: Software Design Document

Student/Team Name:

Team Name	*TeamOne*	
Team members		
Student Name	Student ID	Role
Ali Sina Mohammad Arif	8061634	Leader
Mansoor Kalemzai	7869472	Presentation Coordinator
Varun Tulsiani	8044661	Management Tool Coordinator
Abdul Rahim Kalsekar	7811056	Submission Coordinator
Abdulhayye Maricar Asfaq Ahamed	7741923	Scribe

Student Phone/Mobile No. +971568648103, +971569879006, +971562859550, +971524628293, +971507907565

Student E-mail: ama438@uowmail.edu.au, mk085@uowmail.edu.au, vmt979@uowmail.edu.au, arrk807@uowmail.edu.au, amaa959@uowmail.edu.au

Lecturer Name: Dr. May El Barachi
Due Date: 27 Nov, 2024
Date Submitted: 27 Nov, 2024

PLAGIARISM:
The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.
The University of Wollongong in Dubai also endorses a policy of non-discriminatory language practice and presentation.

PLEASE NOTE: STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED

DECLARATION:
I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism. We/I declare that the project proposal has not been used in any UOWD courses before and that this project idea is a total new idea of the project team

Signature of Student: Ali Sina

Optional Marks:

Comments:

Lecturer Project Proposal Receipt (To be filled in by student and retained by Lecturer upon return of assignment)

Subject: **Project Title:**
Student / Team Name: **Student Number:**
Due Date: **Date Submitted:**
Signature of Student:

Student Project Proposal Receipt (To be filled in and retained by Student upon submission of assignment)

Subject: **Project Title:**
Student/Team Name: **Student Number:**
Due Date: **Date Submitted:**
Signature of Lecturer:

Table of Contents

Table of Figures

1. Introduction

1.1 Purpose

The Software Design Document (SDD) details the architecture and design of **Protego: AI Model Vulnerability Scanner for Query-Based Model Inversion Attacks**, developed collaboratively by TeamOne and DTS Solutions. It aims to assist system architects, developers, security engineers, and compliance professionals in securely deploying and assessing AI systems by scanning CNN models for vulnerabilities related to model inversion attacks.

1.2 Scope

Protego is designed to evaluate Keras-trained Convolutional Neural Networks (CNNs) against two major query-based model inversion attacks: **Score-Based** and **Boundary-Based**. Key features include:

- Detection of vulnerabilities caused by Score-Based and Boundary-Based Model Inversion attacks.
- Offline analysis of MobileNet, ResNet, and EfficientNet models (.h5 format)
- Generation of detailed reports with mitigation strategies

1.3 Overview

This document outlines:

- System architecture, layers, and data flow
- Input validation and result processing
- UI design and user interactions
- Compliance mapping and traceability to functional requirements

1.4 Reference Material

The following materials were referenced for the design and development of this system:

- Planning and Feasibility Report – TeamOne (2024)
- Requirements Analysis Document – TeamOne (2024)
- IEEE Standard for Software Design Descriptions (IEEE 1016-1998)
- OWASP Guidelines for Model Inversion Attacks

1.5 Definitions and Acronyms

AI: Artificial Intelligence.

CNN: Convolutional Neural Network, a class of deep neural networks frequently used in image analysis and targeted in Model Inversion attacks.

ResNet: Residual Network, a CNN architecture that uses skip connections to overcome vanishing gradient issues.

MobileNet: A lightweight CNN architecture optimized for mobile and edge devices.

EfficientNet: A CNN architecture that optimizes accuracy and efficiency by scaling network dimensions.

Model Inversion Attacks: Techniques allowing attackers to deduce sensitive training data by analyzing model outputs.

Score-Based Attack: Exploits model confidence scores to infer training data.

Boundary-Based Attack: Maps decision boundaries to deduce training data properties.

GDPR: General Data Protection Regulation, a European Union law governing data protection and privacy.

HIPAA: Health Insurance Portability and Accountability Act, a U.S. law governing the privacy and security of health information.

2. System Overview

2.1 Product Context

Protego enhances AI security by integrating into AI pipelines and providing proactive protection for CNNs. It complements tools like IBM ART and TensorFlow Privacy by focusing on adversarial simulation. Protego works both as a standalone scanner and in integration with AI deployment pipelines.

- It supports industries handling sensitive data (e.g., healthcare, finance, law) by mitigating vulnerabilities in AI models.
- Designed to work independently or in conjunction with existing AI monitoring solutions, the system enhances the robustness of deployed AI applications.

2.2 Major Functional Components

The system is composed of three primary components, each designed to fulfill critical aspects of vulnerability detection, monitoring, and reporting:

1. **Scanning Engine**
    - Simulates **Score-Based** and **Boundary-Based** attacks on **CNNs**
    - Assesses security posture of **MobileNet**, **ResNet**, **EfficientNet** models
2. **Monitoring Module**
    - Tracks AI robustness post-deployment
3. **Reporting Module**
    - Generates compliance reports aligned with regulations such as GDPR and HIPAA.
    - Offers detailed insights into the vulnerabilities detected and recommended mitigation strategies.

2.3 Performance Goals

The **AI Vulnerability Scanner** is designed to meet the following high-level performance expectations:

- **Latency:** Analyze models with a lesser processing time on standard GPU configurations.

- **Scalability:** Efficiently handle multiple models simultaneously, supporting diverse architectures like ResNet, MobileNet, and EfficientNet.
- **Accuracy:** Achieve a high detection rate for Model Inversion vulnerabilities while minimizing false positives.
- **Usability:** Deliver actionable insights and easy-to-interpret resilience ratings for organizations with varying levels of AI expertise.

3. System Architecture

The AI Vulnerability Scanner is thoughtfully designed to tackle vulnerabilities in Convolutional Neural Networks (CNNs) through a modular and comprehensive approach. The architecture strikes a balance between scalability, ease of maintenance, and functionality. It seamlessly integrates into diverse AI pipelines while delivering high-performance scanning and reporting features. Below, we'll dive deep into the system's structure, including its layered design, major subsystems, and the reasoning behind our design decisions.

3.1 Architectural Design

The system's architecture is built around four distinct layers, each with a specific role. This layered design ensures clear separation of tasks, enhances modularity, and facilitates smooth data flow across the components. Let's break down the purpose and functionality of each layer:

Layer 1: Data Ingestion

- Imports .h5 CNN models and input data
- Validates and standardizes metadata and datasets

Layer 2: Scanning Engine

- Runs Score-Based and Boundary-Based attack simulations
- Analyzes outputs for anomalies, patterns, and reconstruction risk

Layer 3: Reporting Module

- Generates PDF/HTML reports
- Assigns resilience scores, risk levels, and mitigations

Layer 4: User Interface

- React-based dashboard with real-time results, scan history, and model summaries

Diagram Representation

Here's a simple view of the architecture, showing how each layer builds on the previous one:

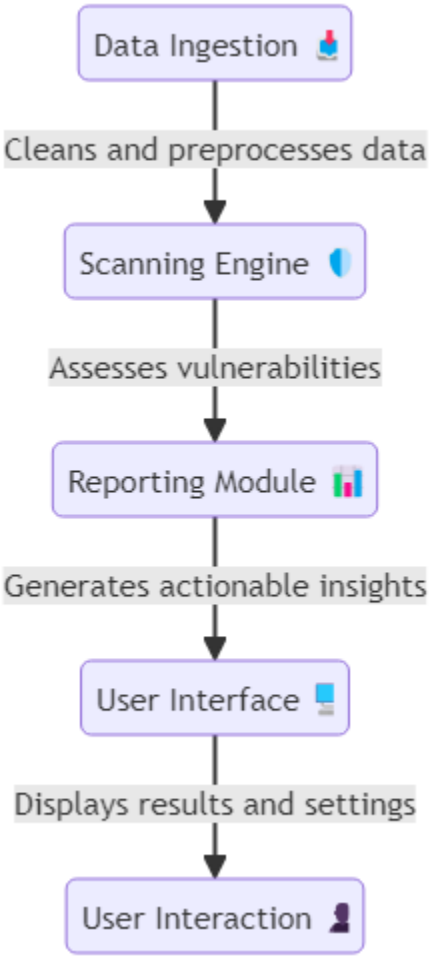

Figure 1 Architectural Design

3.2 Subsystem Overview

Each layer contains smaller subsystems, each designed for specific tasks. These subsystems work together seamlessly, ensuring smooth workflows and accurate results.

- **Data Validation**: Processes and prepares CNN model structure and attack inputs
- **Attack Simulation**: Executes Score-Based and Boundary-Based query attacks
- **Risk Assessment**: Computes resilience and assigns mitigation actions

3.3 Design Rationale

Our design choices reflect the need for efficiency, security, and scalability when addressing CNN vulnerabilities.

- **Choice of Architectures**:
    - **ResNet**: Given its popularity in image analysis, ResNet often becomes a target for attackers, making robust detection vital.
    - **MobileNet**: Frequently used in mobile devices, its lightweight design necessitates a focus on security in resource-constrained settings.

    - **EfficientNet**: As a cutting-edge architecture, its balance of accuracy and efficiency makes its protection a priority.

- **Advantages of a Modular Approach**:
    - **Scalability**: Each layer operates independently, allowing the system to handle larger workloads or added features with ease.
    - **Maintainability**: Subsystems are decoupled, simplifying updates and debugging without disrupting the rest of the system.
    - **Flexibility**: The design easily integrates into existing AI pipelines, adapting to different organizational needs.
    - **Performance**: Each layer focuses on a specialized role, ensuring fast and accurate results.

4. Data Design

4.1 Data Description

The information domain of the AI Vulnerability Scanner is transformed into data structures to support the detection, analysis, and mitigation of vulnerabilities in Convolutional Neural Network (CNN) architecture. The system manages multiple types of data related to model inputs, outputs, attack characteristics, and mitigation actions.

Key aspects of data storage, processing, and organization include:

- **Model Metadata** (architecture, layers, activation functions)
- **Attack Profiles** (score-based, boundary-based)
- **Scan Results** (extracted features, boundary mappings, vulnerabilities)
- **Compliance Info** (GDPR, HIPAA fields)

Storage:

- Relational DB for metadata and results
- File storage for logs and full reports
- Real-time pipelines for cloud monitoring (if enabled)

4.2 Data Dictionary

Table 1: Entity Data Dictionary

Entity Name	Attribute	Type	Description
AttackProfile	attackType	String	Type of model inversion attack (e.g. Score-Based, Boundary-Based)
	parameters	JSON	Configuration parameters specific to the attack type

ComplianceReport	reportID	Integer	Unique identifier for each compliance report
	generatedDate	DateTime	Timestamp when the compliance report was generated
	regulationType	String	Type of regulation (e.g. HIPAA, GDPR)
ModelMetadata	modelID	Integer	Unique identifier for each CNN model scanned
	architecture	String	CNN architecture type (e.g. ResNet, MobileNet)
	parameters	JSON	Architecture-specific parameters (e.g. # of layers, activation functions)
ScanResult	scanID	Integer	Unique identifier for each scan instance
	vulnerabilityType	String	Type of detected vulnerability
	riskLevel	String	Severity level of the vulnerability (e.g. High, Medium, Low)
	recommendation	Text	Suggested mitigation actions for the detected vulnerability
User	userID	Integer	Unique identifier for users of the system
	role	String	Role of the user (e.g. System Architect, Security Engineer)

Table 2: Functions Data Dictionary

Function Name	**Parameter**	**Type**	**Description**
scanModel	modelID	Integer	ID of the CNN model to be scanned
	scanOptions	JSON	Options specifying attack types and parameters to be scanned
generateReport	scanID	Integer	ID of the scan for which a compliance report will be generated
	reportFormat	String	Format of the report (e.g. PDF, HTML)

5. Component Design

5.1 Login Phase

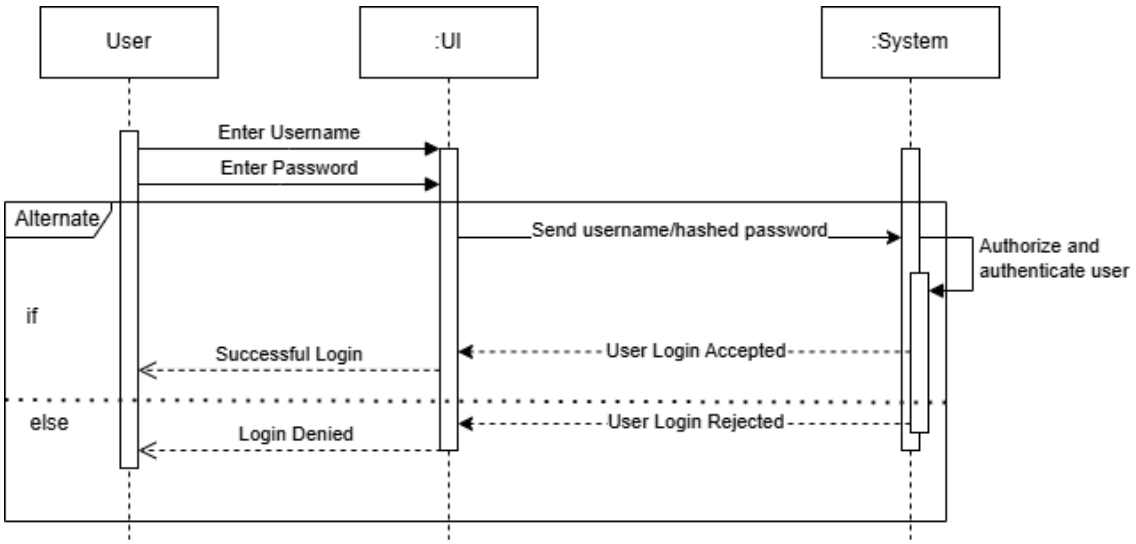

Figure 2 Sequence diagram for Login

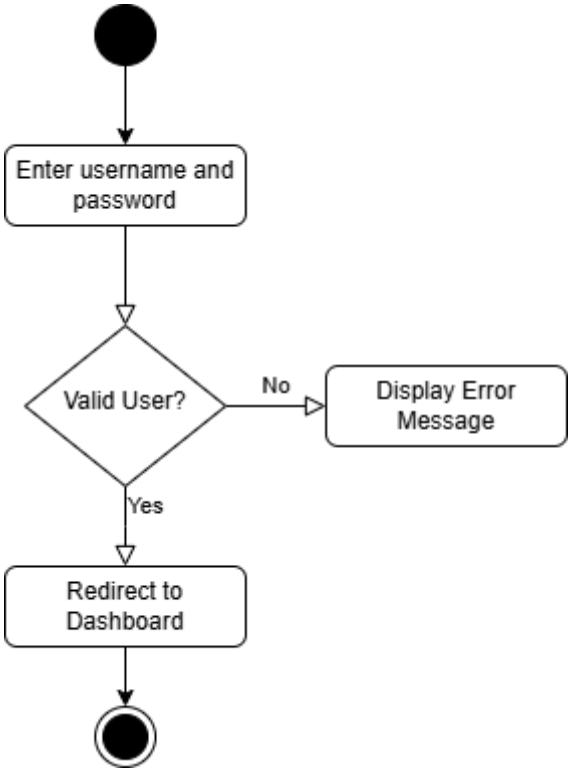

Figure 3 Activity Diagram for Login

5.2 Main Page (Launch Phase)

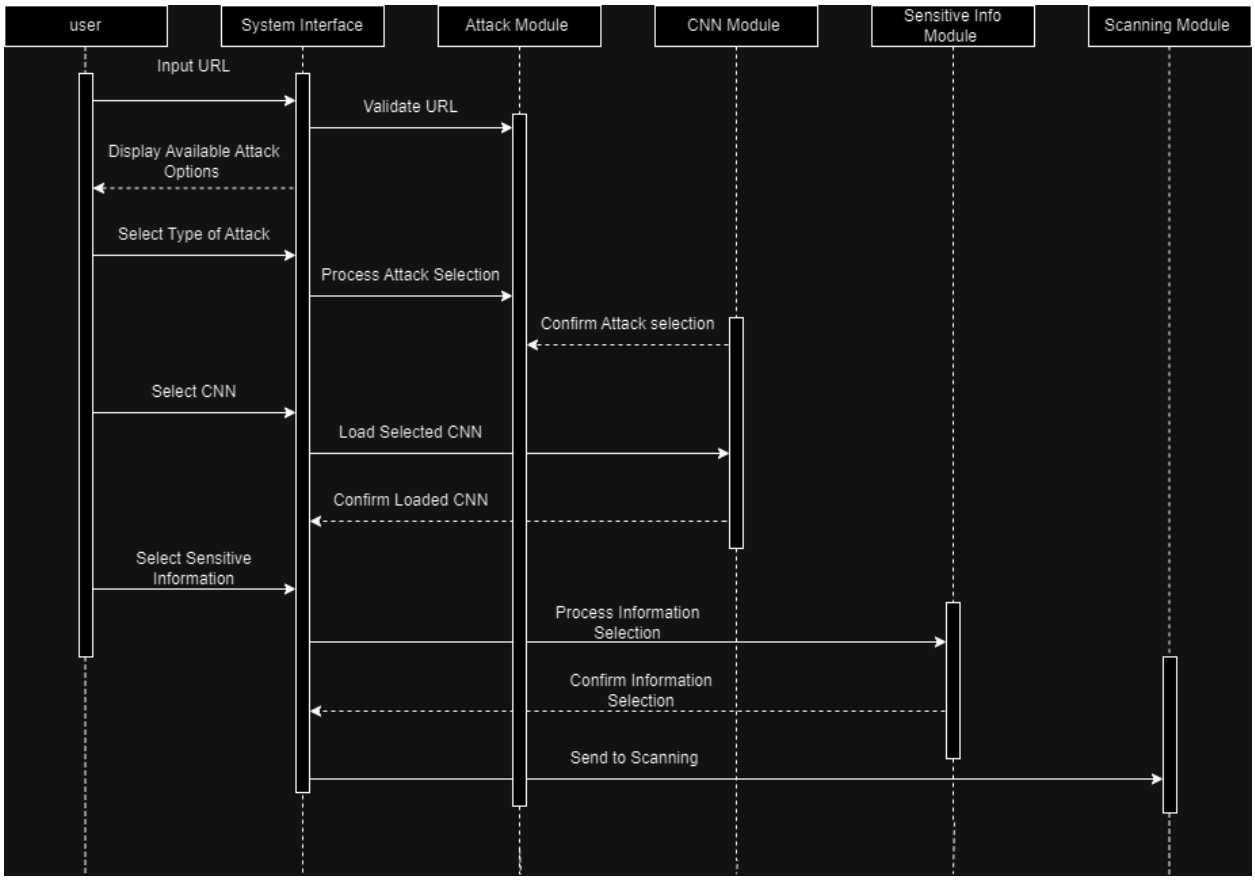

Figure 4 Main Page Sequence Diagram

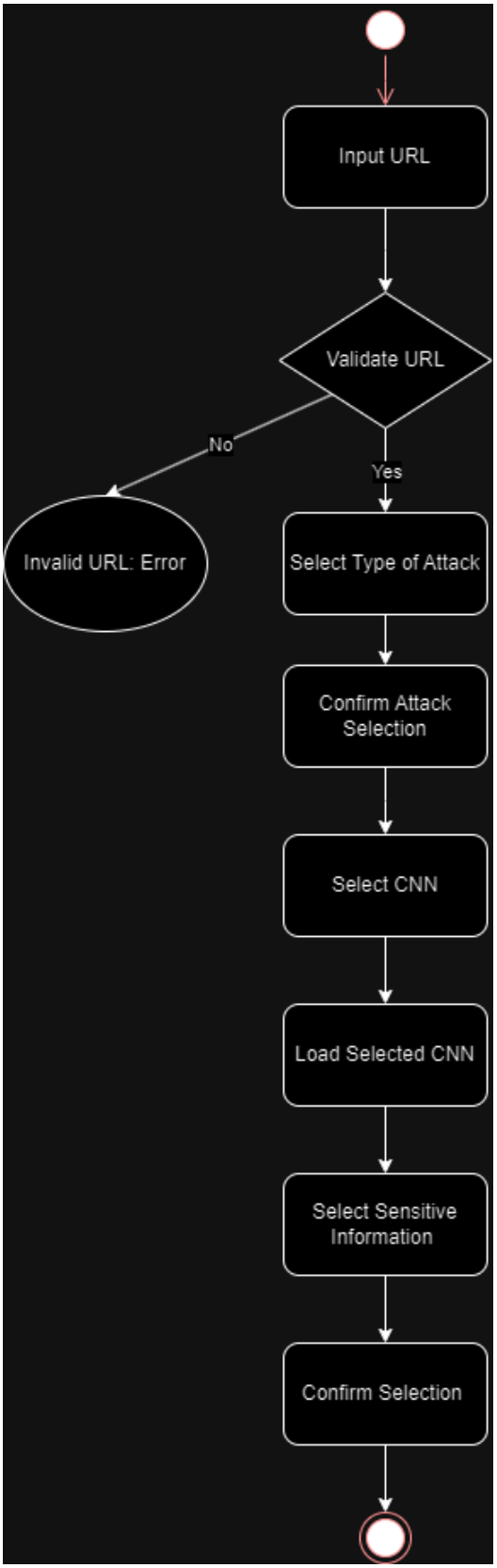

Figure 5 Activity Diagram for Main Page

5.3 Scanning Phase

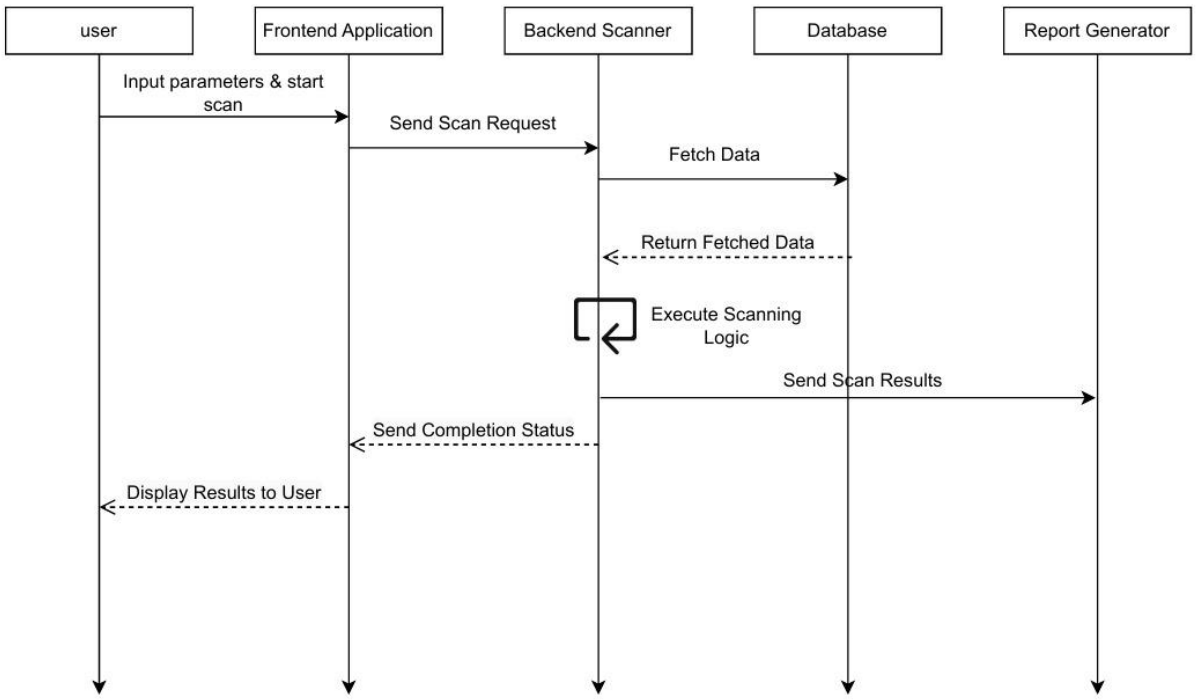

Figure 6 Sequence Diagram for Scanning

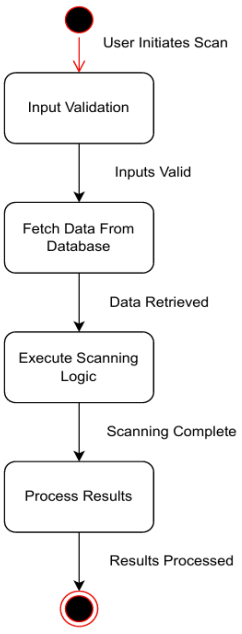

Figure 7 Activity Diagram for Scanning

5.4 Report Generation Phase

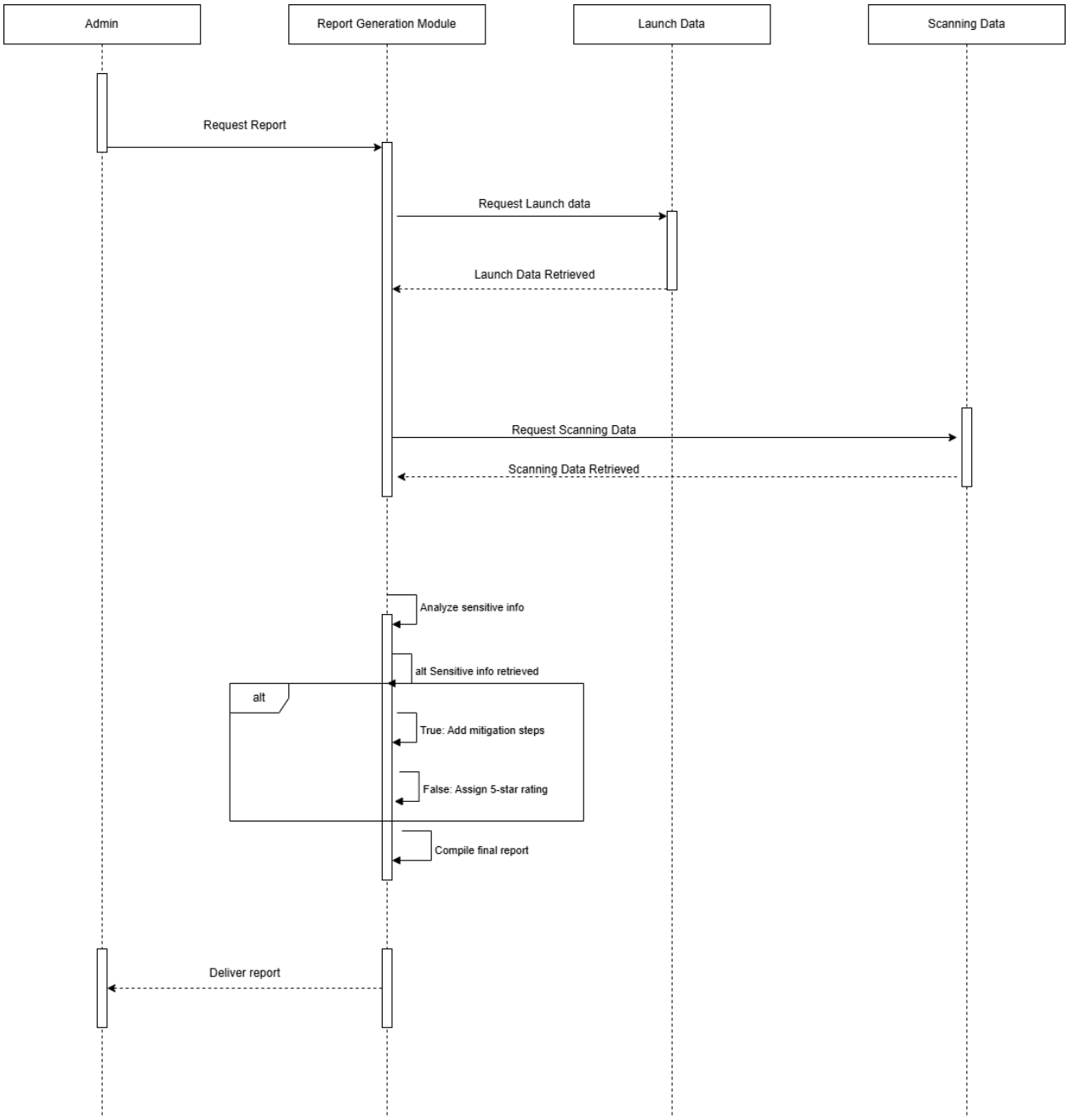

Figure 8 Sequence Diagram for Generating Report

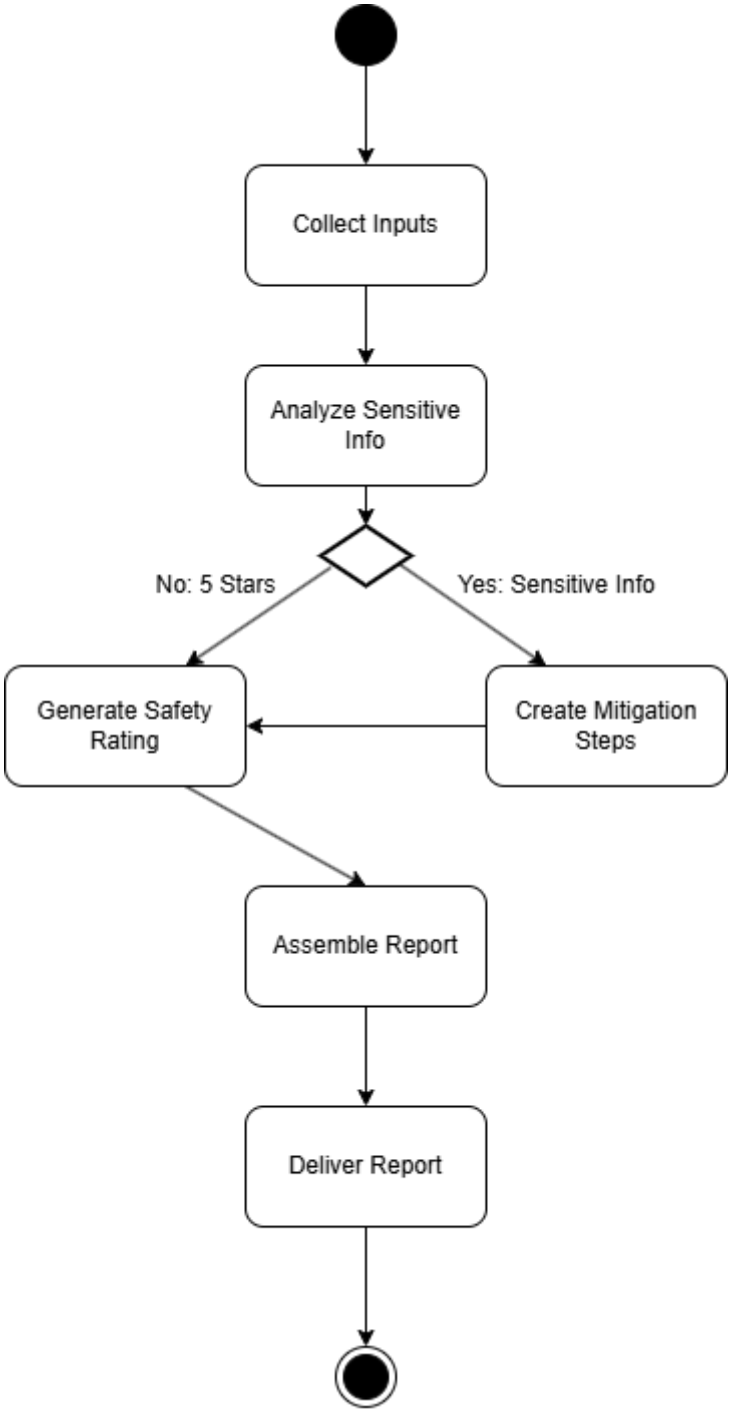

Figure 9 Activity Diagram for Generating Report

5.5 Dashboard (Output Phase)

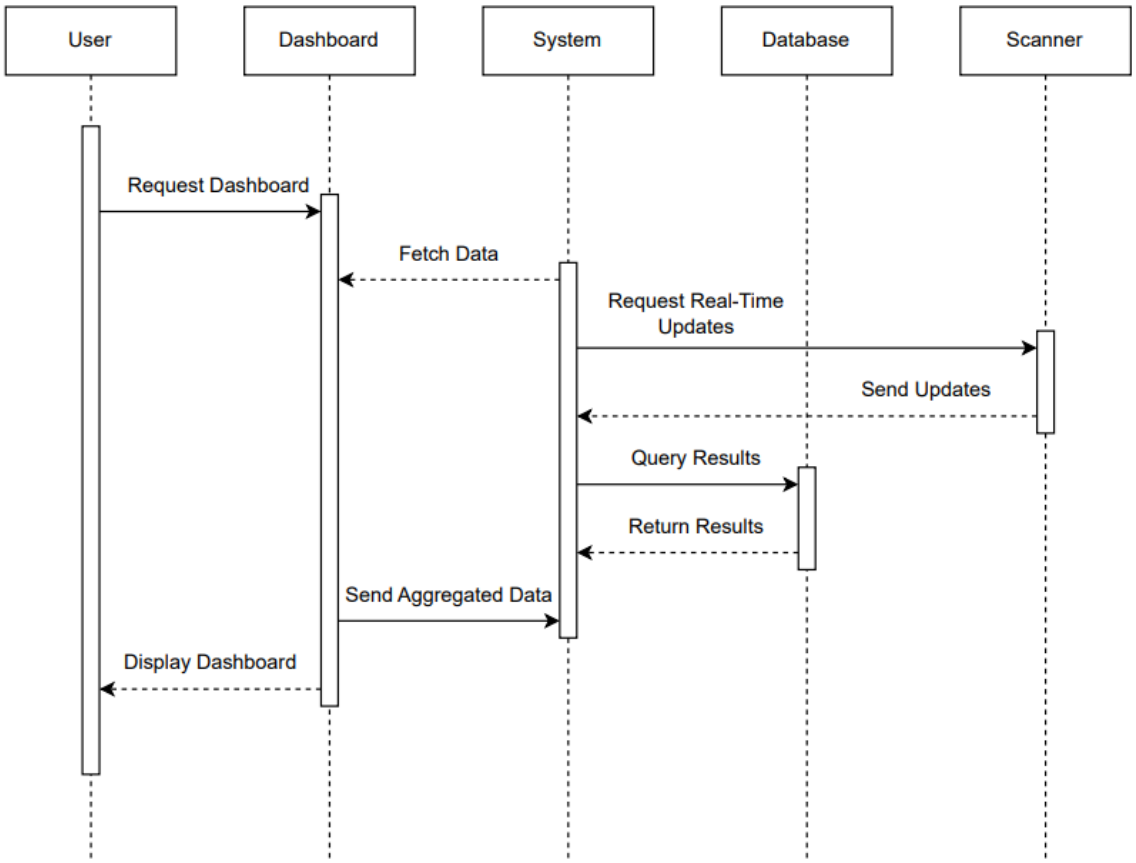

Figure 10 Sequence Diagram for Dashboard

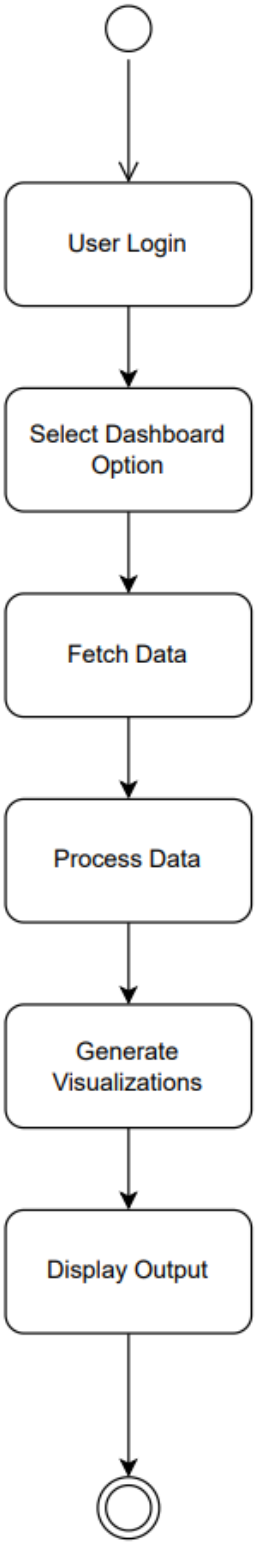

Figure 11 Activity Diagram for Dashboard

6. Human Interface Design

6.1 Overview of User Interface

The user interface (UI) for the AI Vulnerability Scanner is designed to be intuitive and accessible, providing clear navigation for different users. The system supports two primary user roles: Security Engineer and Compliance Officer, each with tailored functionalities to meet their needs.

Security Engineers:

- Upload models, configure scan parameters
- View scan progress and detailed results

Compliance Officers:

- Download compliance reports
- Monitor resilience metrics

6.2 Screen Images

6.2.1 Main Page

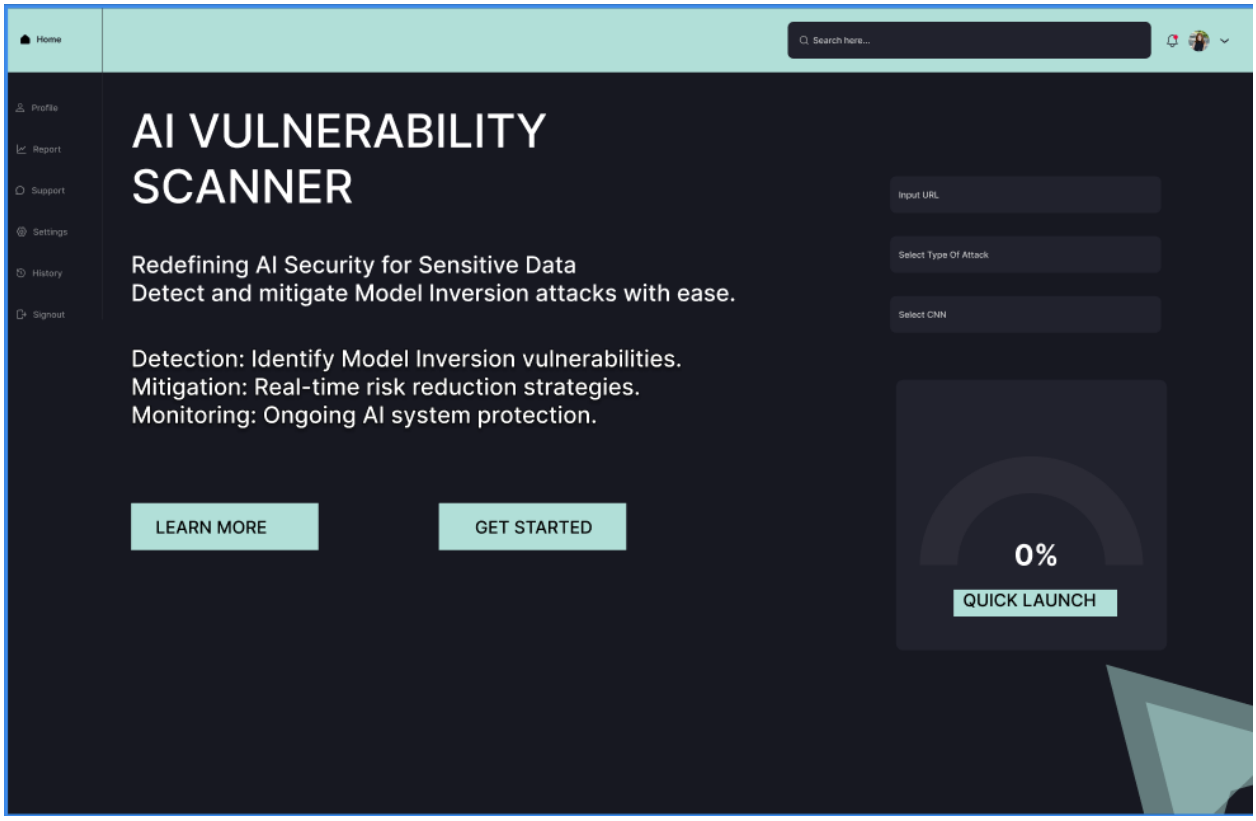

Figure 12 Main Page

The Main page is the first page we are shown once we open the Scanner and it has clear information and instructions about the Scanner and the option to start scanning where we must provide the AI model, the type of CNN Model, the type of Attack and the Sensitive information or Vulnerability we would like to Scan for then once everything is finalised the Scanner will start scanning and take us to the next page.

6.2.2 Login Page

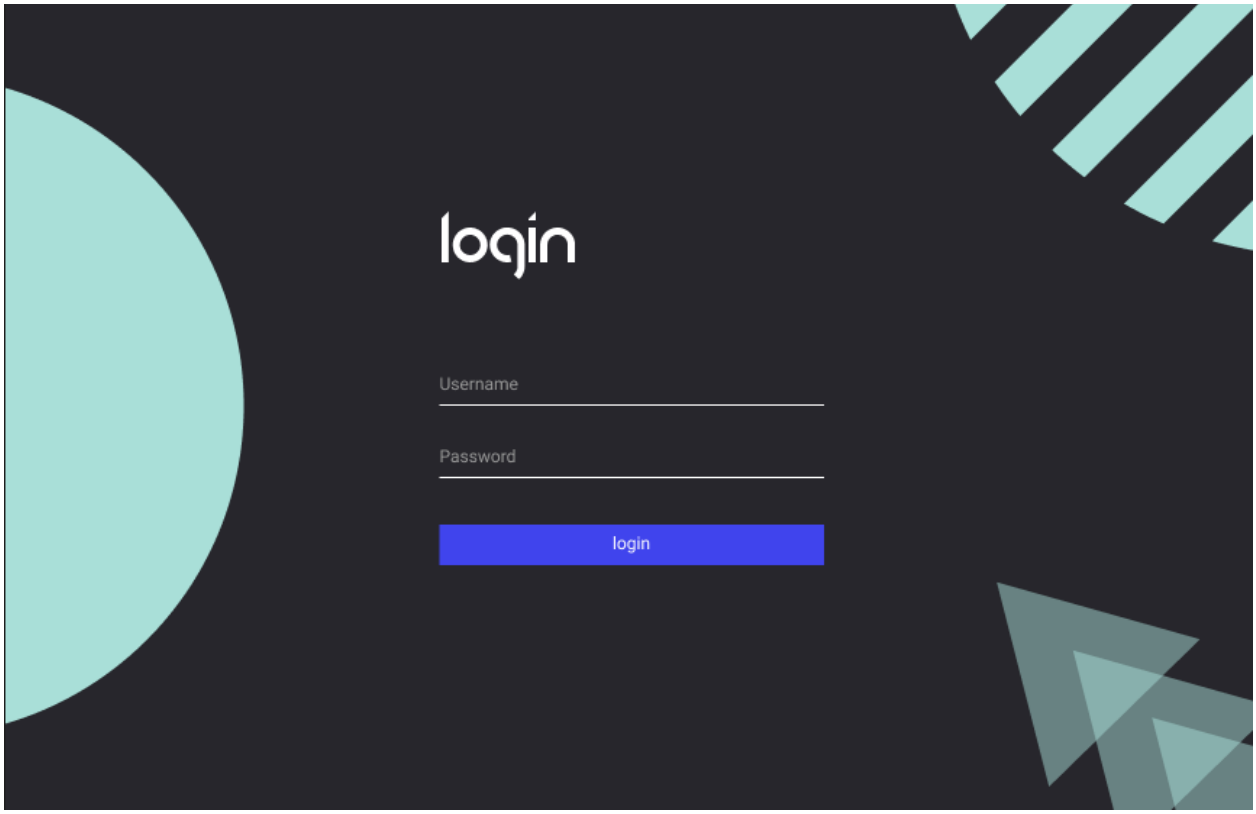

Figure 13 Login Page

The login page where users can enter their usernames and passwords to gain access to the scanning, dashboard and report pages according to their access level, Admins can access all pages while One-time logins are provided to other users and have access to Dashboard and Report pages only.

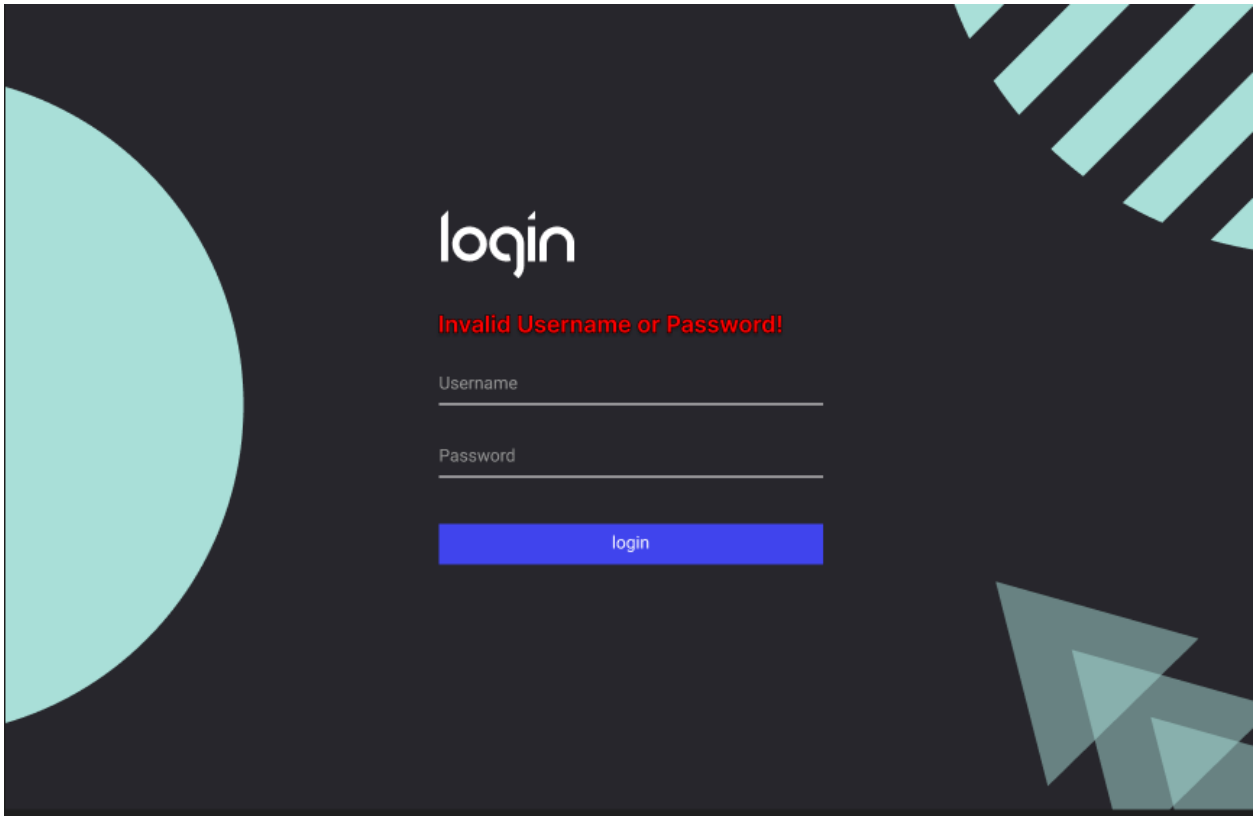

Figure 14 Failed Login Page

Failed login page is shown when a user enters invalid username or password, after multiple failed logins they should contact the Admins for assistance.

6.2.3 Scanning Page

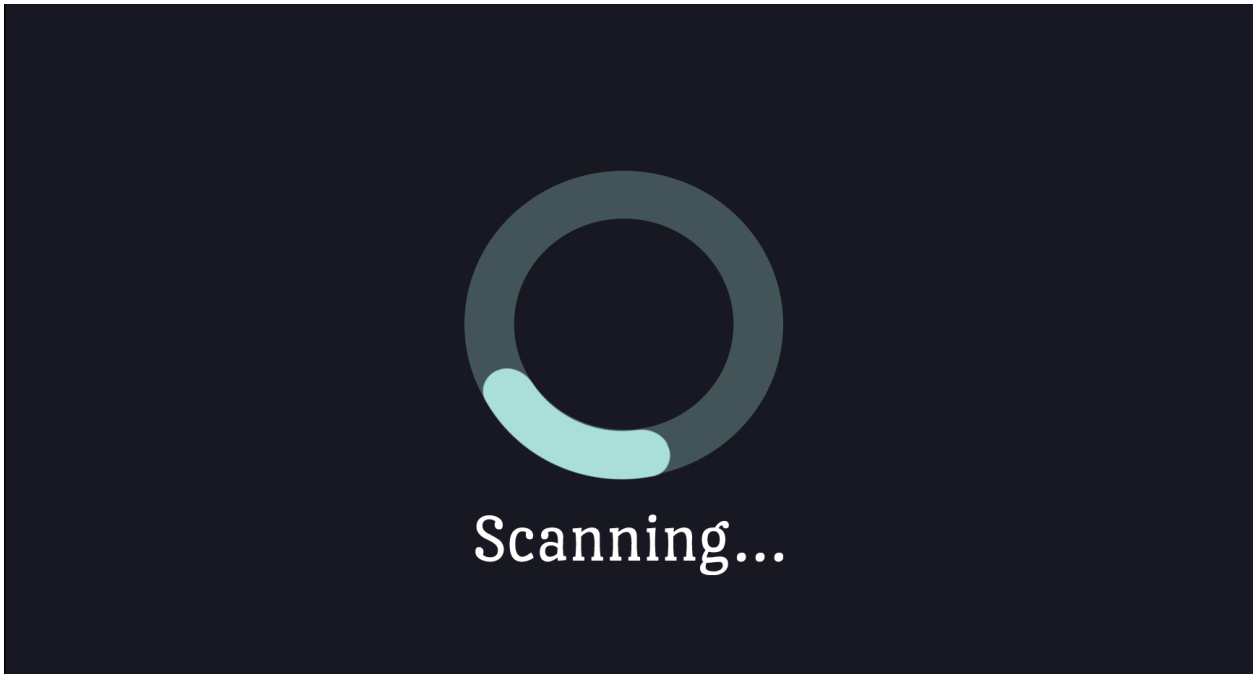

Figure 15 Scanning Page

The Scanning page appears while a Scan is initiated and this page is loaded until the scan is complete.

6.2.4 Report Page

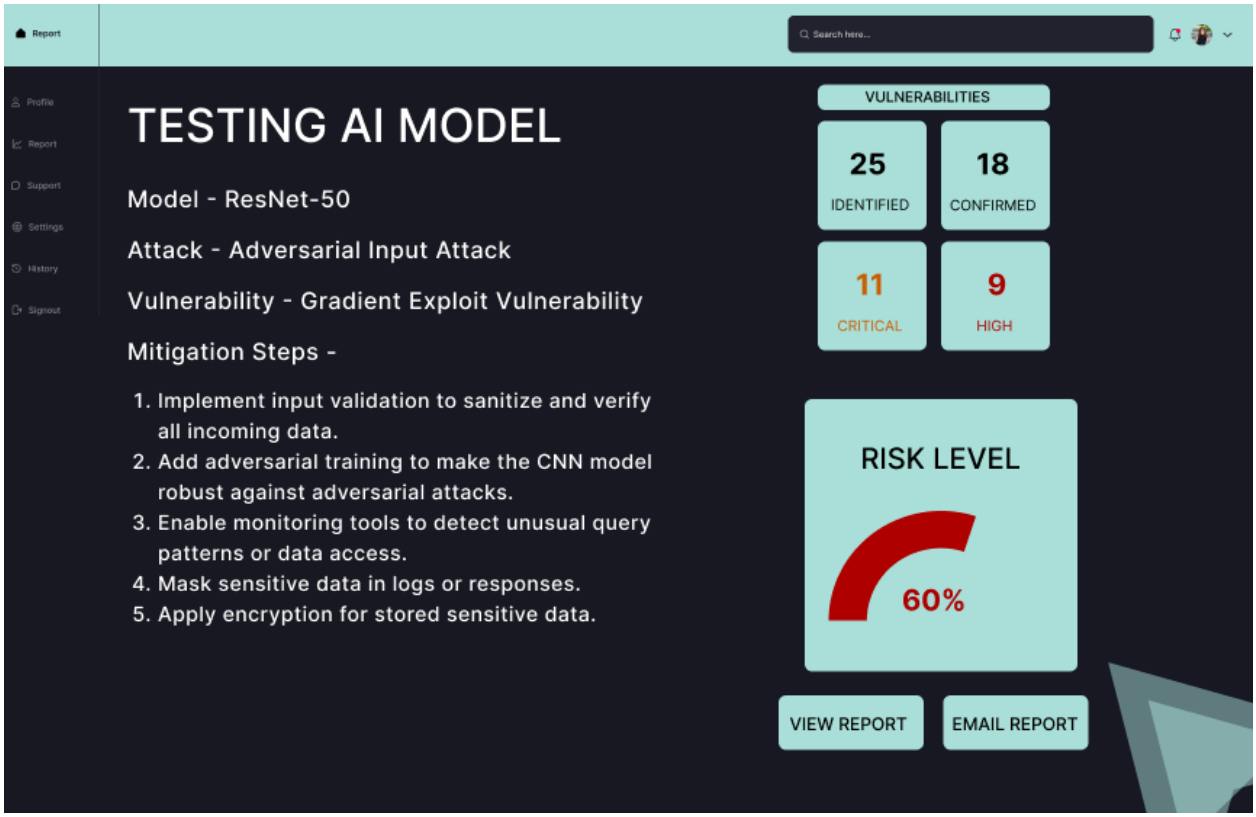

Figure 16 Report Page

The report page is shown after a scan is completed, it displays the AI model name, CNN Model type, Attack type and Vulnerability type chosen and mitigation steps needed to secure the AI, there is also useful analytics for risk level and Vulnerabilities and the complete report can be viewed and emailed in pdf format in which all the above data is shown as well as all queries used by the Scanner and all outputs received from the model along with sensitive information gathered.

6.2.5 Dashboard Page

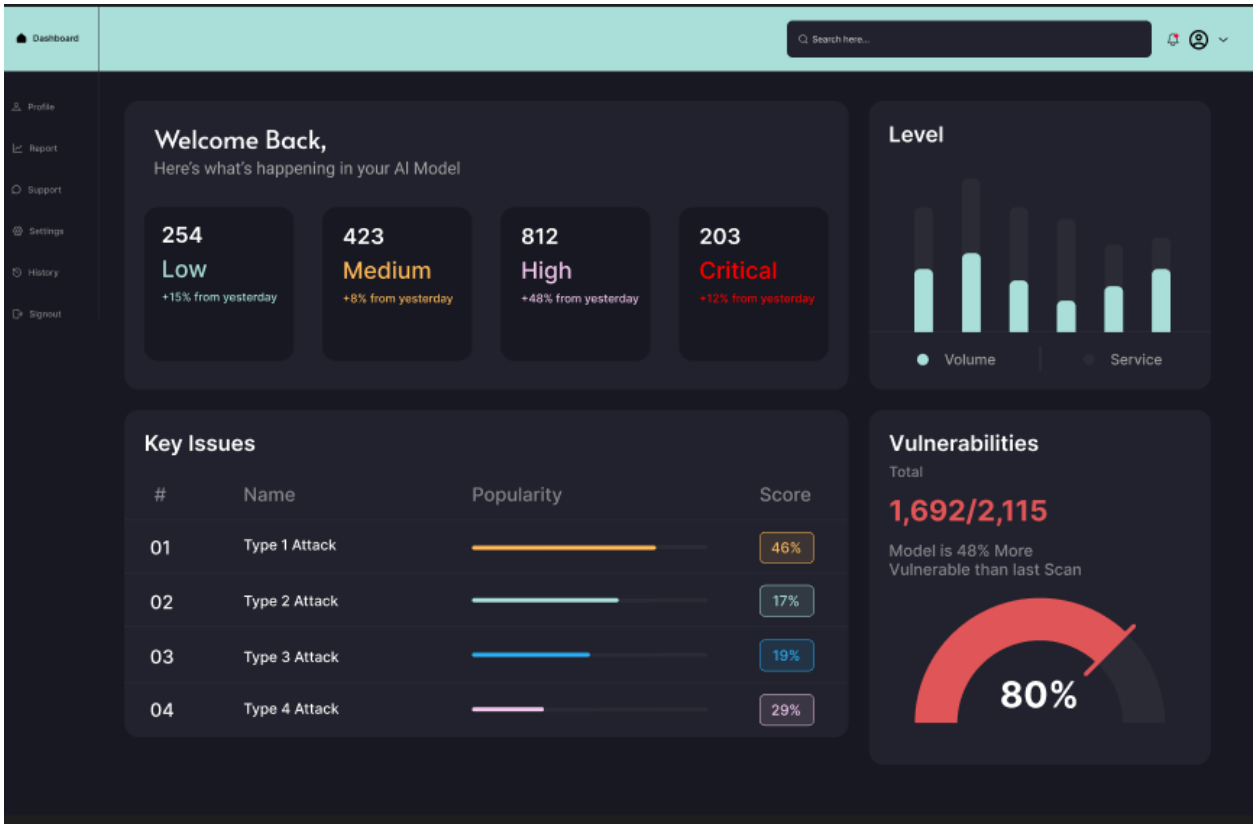

Figure 17 Dashboard Page

The Dashboard page displays key analytics for the models tested and is equally useful for Admins as well as interested users.

6.3 Screen Objects and Actions

1. Main Page

- **Purpose:** Introduce the AI Vulnerability Scanner and guide users to learn more or get started with the system.
- **Key Components:**
    - **Header:** Displays the application name or logo at the top.
    - **Learn More Button:**
        1. **Purpose:** Redirects users to an informational page or modal explaining the scanner's features and benefits.
        2. **Action:** On click, navigates to the Learn More page or opens a modal overlay.
    - **Get Started Button:**
        1. **Purpose:** Directs the user to the Login Page.

2. **Action:** On click, navigate to the Login Page.

2. Login Page

- **Purpose:** Secure access to the application by authenticating users.
- **Key Components:**
    - **Text Field (Username):**
        1. **Purpose:** Allows the user to input their username or email.
        2. **Behavior:**Input validation ensures the field is not empty.May provide error messages for invalid usernames (e.g., "User not found").
    - **Password Field:**
        1. **Purpose:** Allows the user to input their password securely.
        2. **Behavior:** Masks input for privacy and provides error messages if incorrect credentials are entered.
    - **Login Button:**
        1. **Purpose:** Initiates the login process.
        2. **Behavior:** Validates credentials and redirects the user to the dashboard on successful login.
        3. **Error Handling:** Displays an error message (e.g., "Invalid username or password") for failed attempts.

3. Scanning Page

- **Purpose:** Displays a scanning animation or loop while the vulnerability scanner operates.
- **Key Components:**
    - **Loading/Scanning Animation:**
        1. **Purpose:** Visually indicates that the scanner is processing inputs and performing its analysis.
        2. **Behavior:** A continuous loop with text such as "Scanning in progress…" or a progress bar if applicable.
    - **Status Text:**
        1. **Purpose:** Updates the user on the scanning status (e.g., "Analyzing queries…" or "Generating results…").
    - **Action Handling:**

1. No direct user interactions; the page transitions automatically to the next page (e.g., Results Dashboard or Report Page) when scanning is complete.

4. Report Page

- **Purpose:** Presents the scan results in an organized, analytical format with options to view or share the report.
- **Key Components:**
    - **Analytics Panel:**
        1. **Purpose:** Displays key metrics and graphs from the scan results.
        2. **Elements:**Charts and graphs summarizing vulnerabilities detected, safety ratings, and attack success rates.Text summaries highlighting sensitive data retrieved and mitigation steps.
    - **Show Report Button:**
        1. **Purpose:** Displays the detailed report within the application.
        2. **Action:** On click, opens a detailed report.
    - **Email Report Button:**
        1. **Purpose:** Allows the user to email the report in PDF format.
        2. **Action:** On click, prompts for an email address to send the PDF.

5. Dashboard

- **Purpose:** Serves as the central hub for users to view overall analytics, access recent scans, and navigate to other features.
- **Key Components:**
    - **Analytics Section:**

Purpose: Displays system-wide summaries such as:

1. Number of scans performed.
2. Average safety rating.
3. Most frequently targeted sensitive data types.

- **Recent Activity Panel:**
    1. **Purpose:** Lists recently completed scans with timestamps and summary results.
    2. **Action:** Allows users to click on a specific scan to navigate to the report page for that scan.
- **Navigation Drawer:**

1. **Purpose:** Provides quick access to other pages (e.g., Scan Configuration Page, Report Page).
2. **Elements:**Menu items for each page, such as Dashboard, Reports, and Login. Highlights the current page on top left for easy navigation.

7. Resulted Final Design

7.1 Landing Page

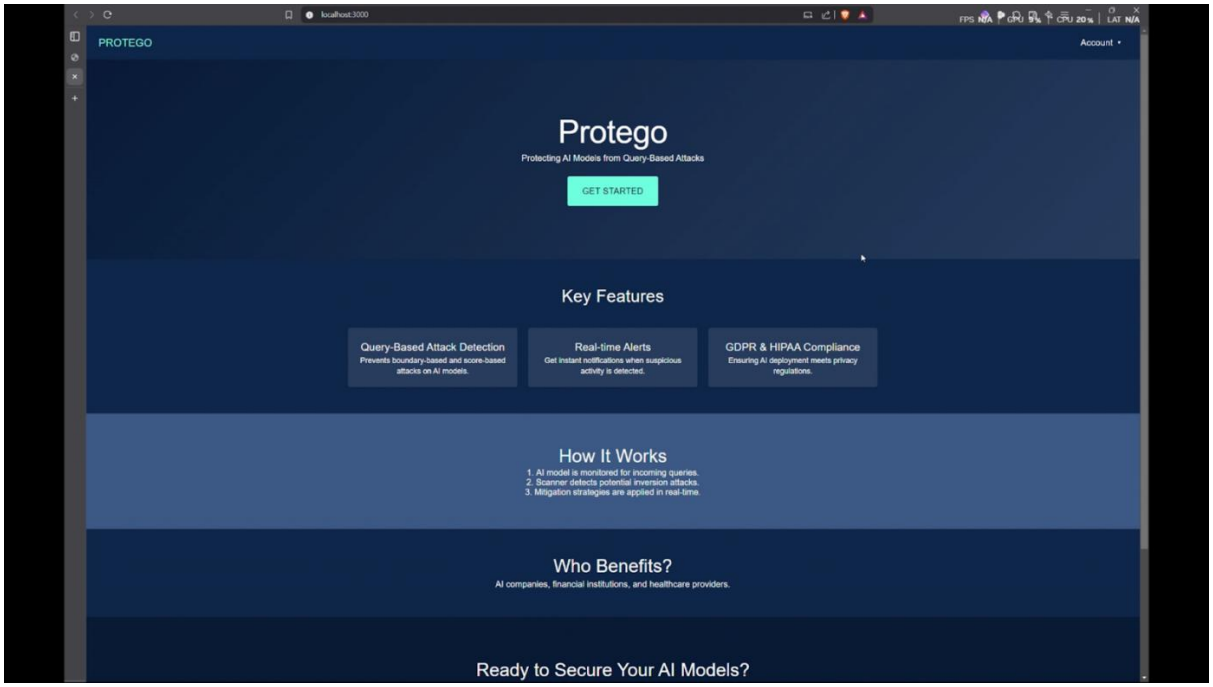

Figure 18 Landing Page

This showcases Protego's primary landing page. It highlights the scanner's key features, briefly explains how Protego works, and provides quick links to **Register** or **Login**. The page serves as the user's first point of contact, introducing the system's main functionalities—such as query-based attack detection and compliance adherence—and guiding new users to create an account or sign in to begin scanning their AI models.

7.2 Register Page

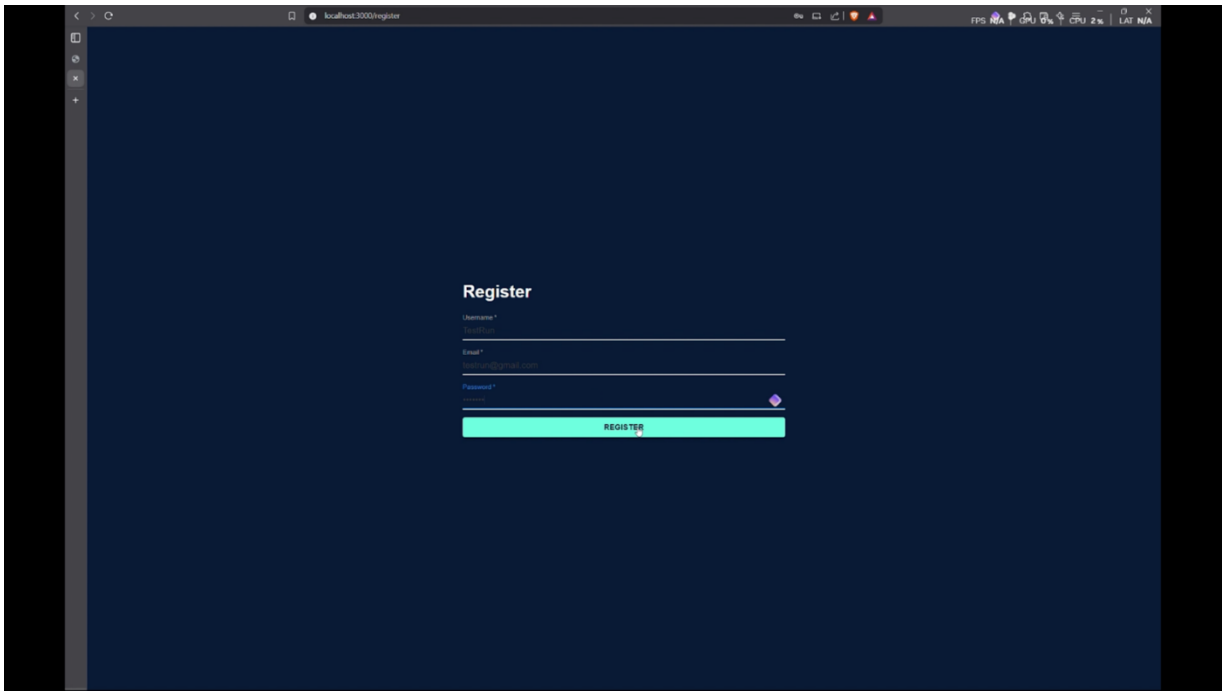

Figure 19 Register Page

This displays the Protego registration page. New users are prompted to create an account by providing necessary details. The page features a clean, user-friendly design that emphasizes security and ease of onboarding.

7.3 Login Page

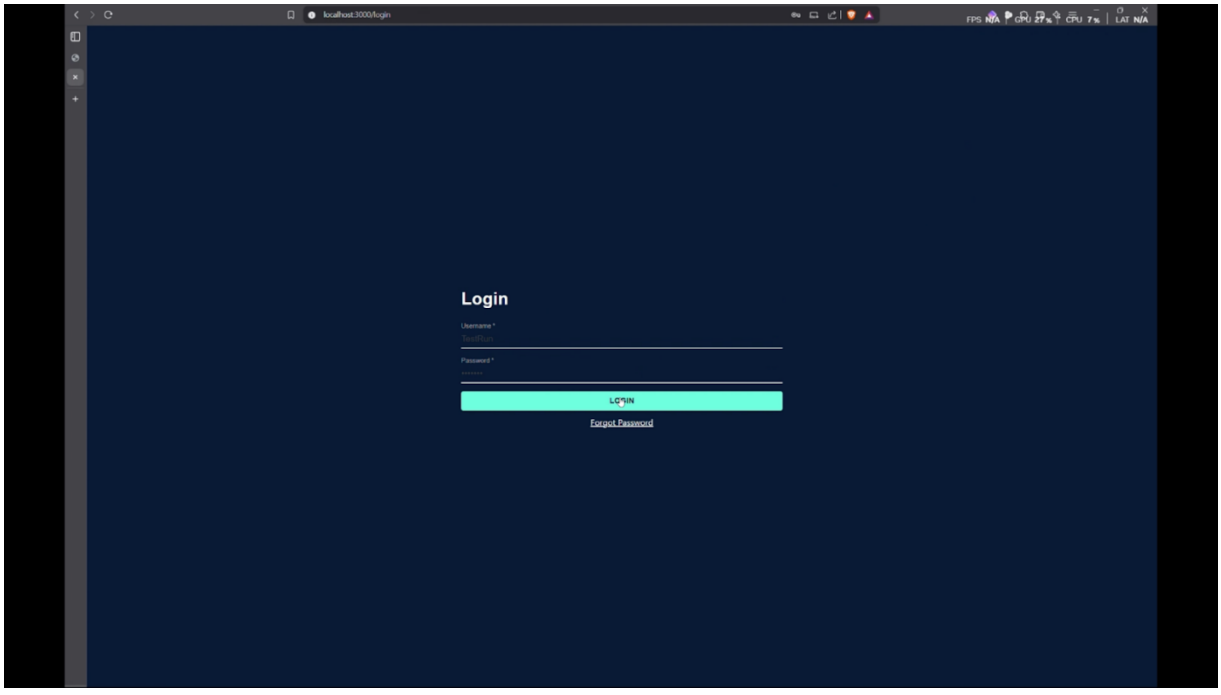

Figure 20 Login Page

This shows the login page for Protego. Users enter their credentials to access the system. The interface is straightforward, ensuring secure access for authorized users.

7.4 Testing AI Model

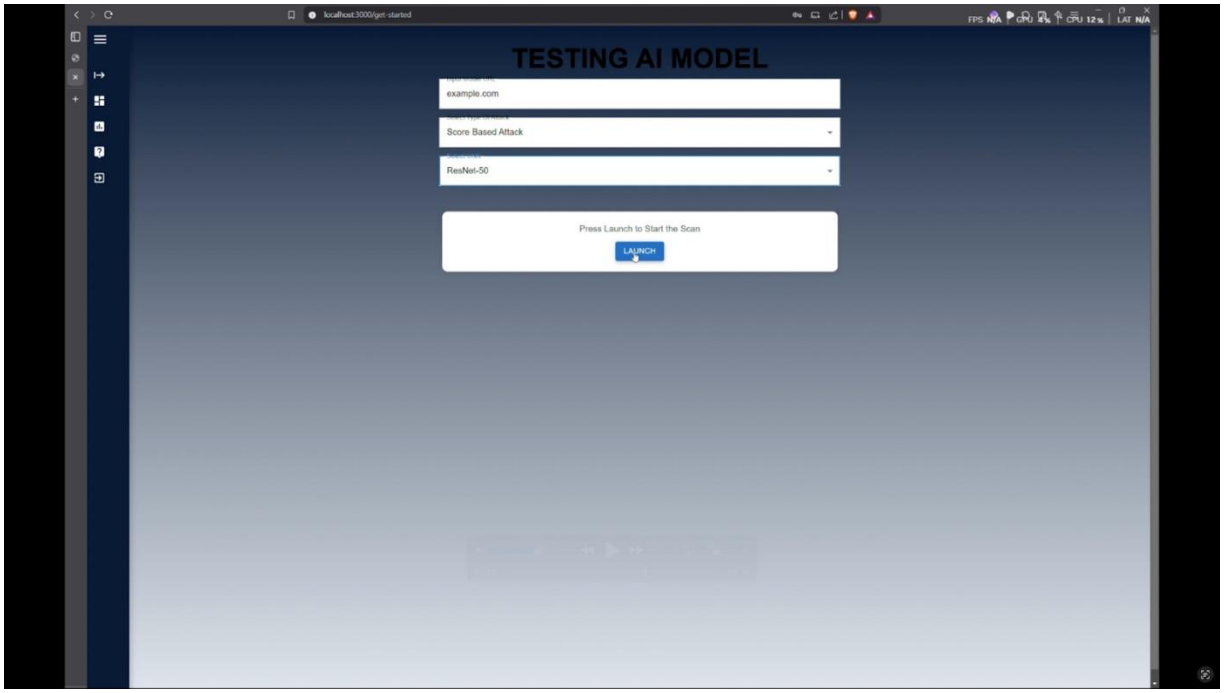

Figure 21 Testing AI Model

This illustrates the section where a user is testing an AI model. The interface allows users to upload a pretrained CNN model (e.g., ResNet, MobileNet, EfficientNet) for vulnerability scanning, highlighting the system's focus on model-based assessments.

7.5 Attack in Progress

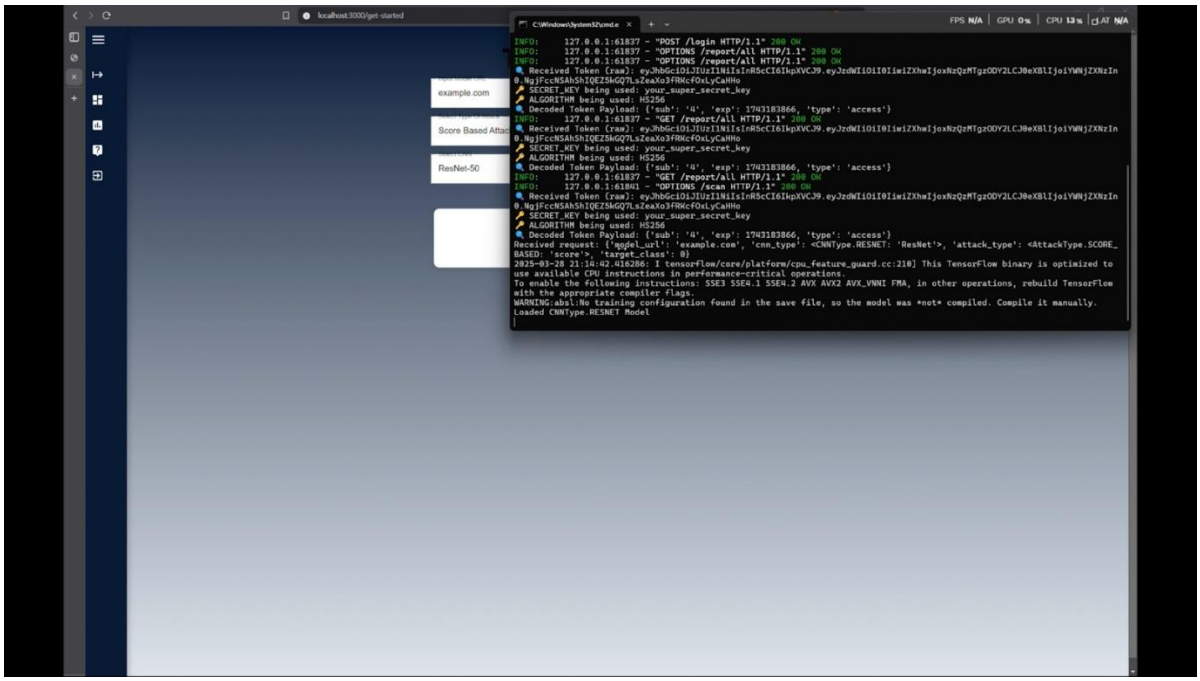

Figure 22 Attack in Progress

This is the Protego scanning engine in action. It shows the simulation of model inversion attacks (score-based and boundary-based) on the selected AI model, with visual indicators demonstrating that the attack is actively in progress.

7.6 Password-Protected Report (PDF Format)

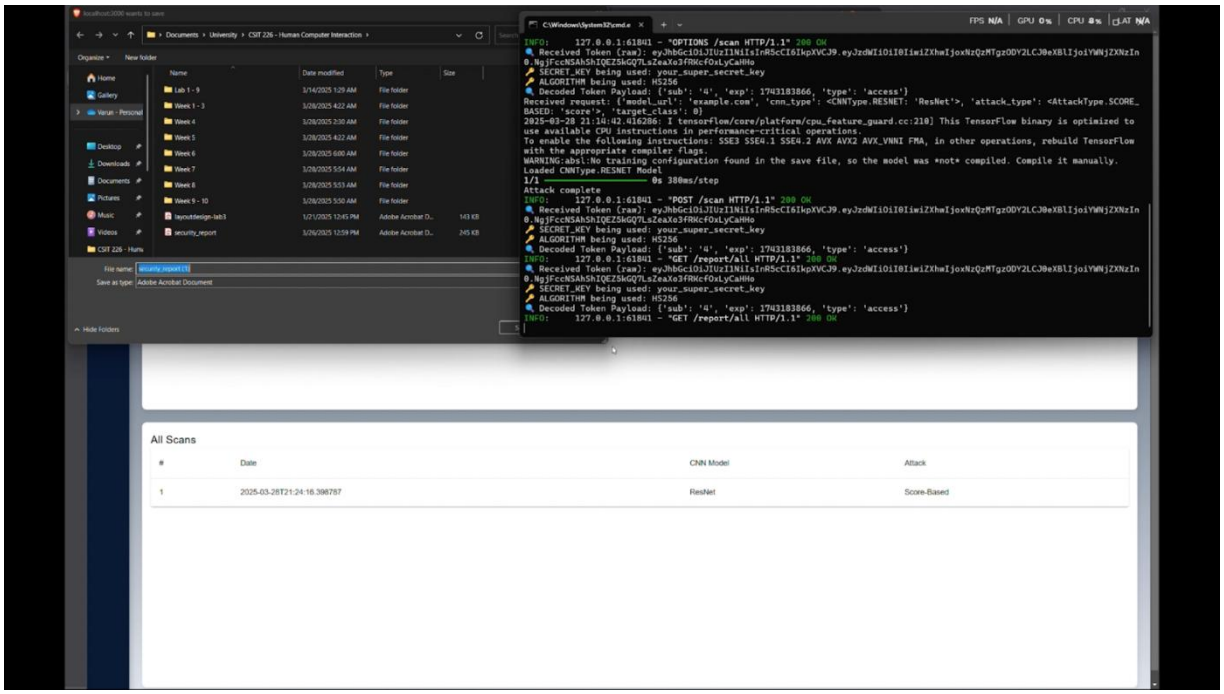

Figure 23 Password-Protected Report (PDF Format)

This displays the generated vulnerability report in PDF format. The report is securely produced and password-protected, ensuring that sensitive findings and recommendations remain confidential.

7.7 Report Access Authentication

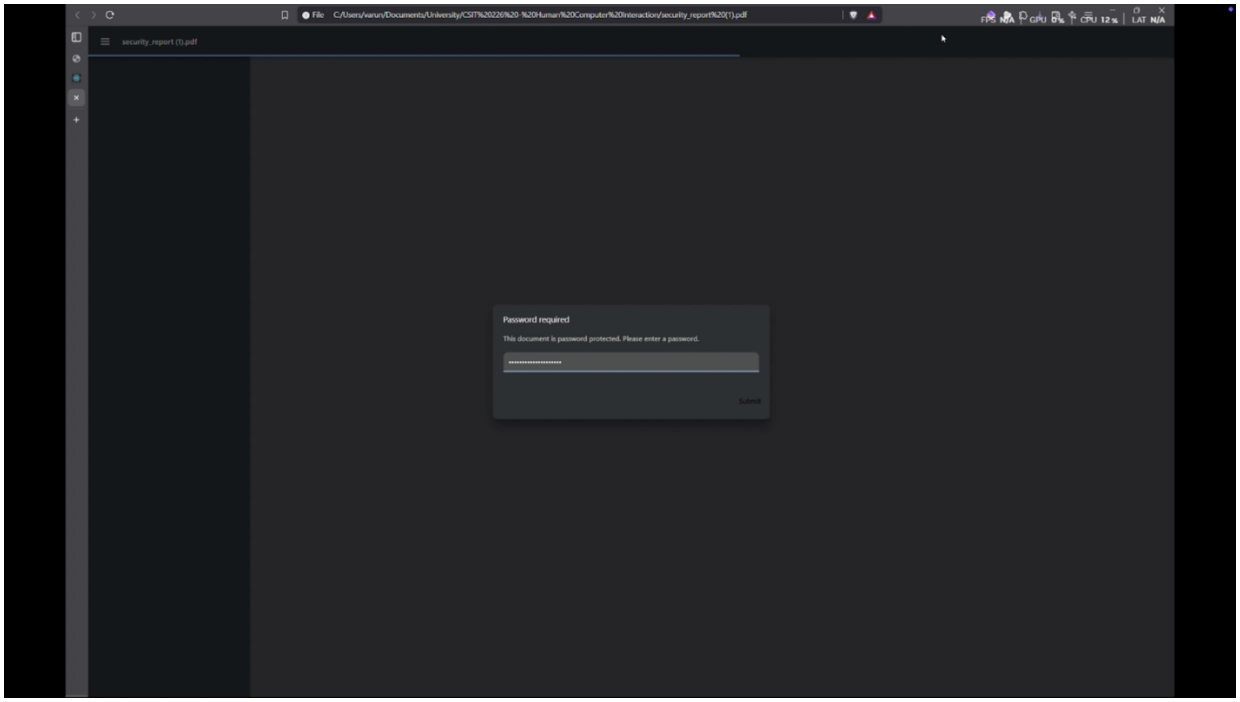

Figure 24 Report Access Authentication

This demonstrates the access control mechanism for the generated report. Users are required to enter a password to view the detailed report, ensuring that only authorized personnel can access sensitive scan results.

7.8 Comprehensive Report

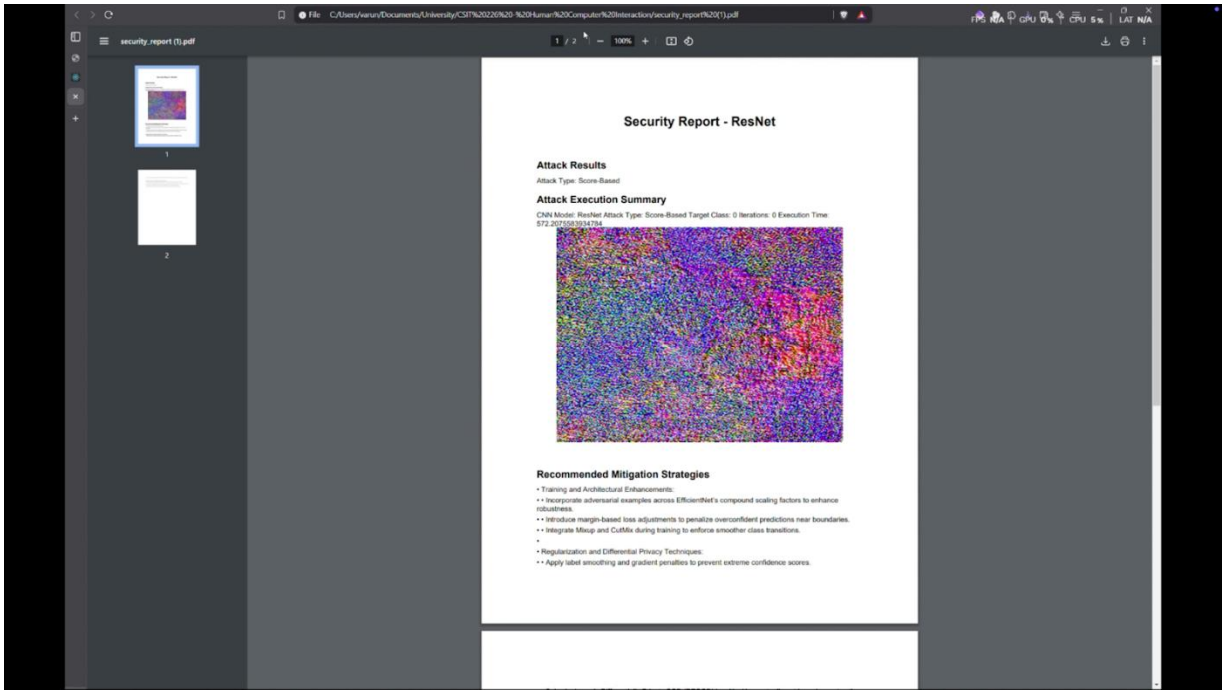

Figure 25 Comprehensive Report

This shows the actual detailed report generated by Protego. It includes a comprehensive summary of the attack execution, details of the type of model inversion attack performed, and the corresponding mitigation steps recommended to secure the model.

7.9 Post Attack Dashboard

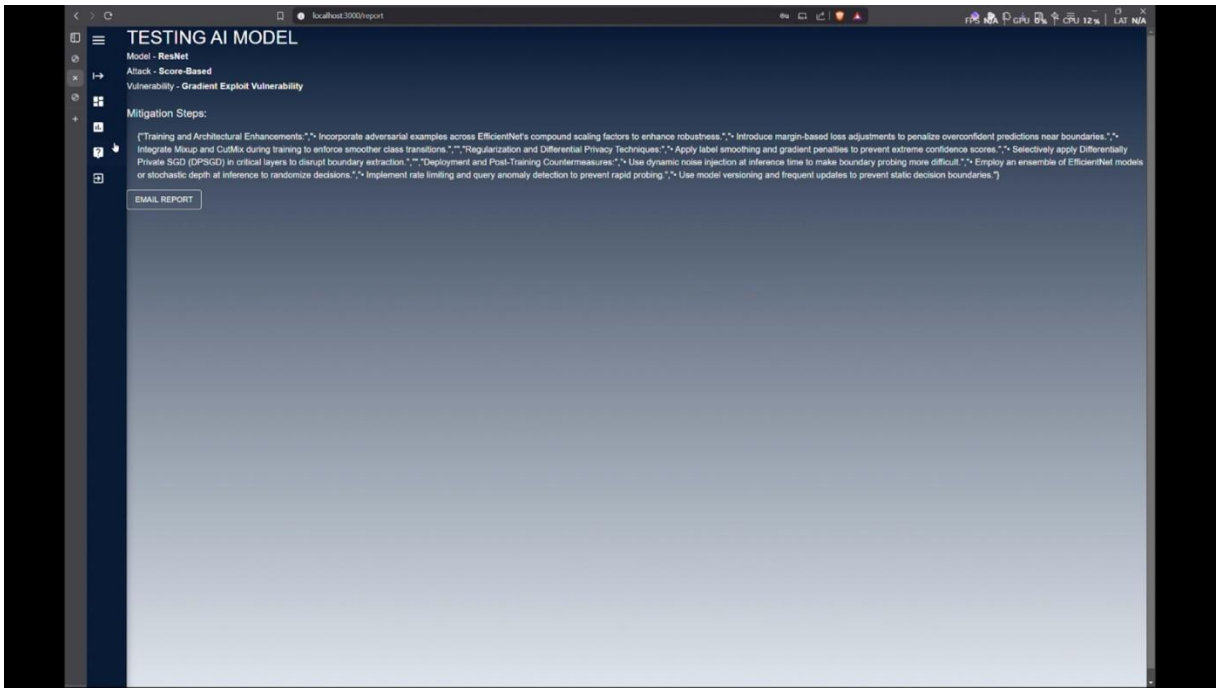

Figure 26 Post Attack Dashboard

This illustrates the Protego dashboard after an attack has been completed. The dashboard displays key information such as model vulnerabilities, executed attack details, and mitigation

recommendations. Additionally, it features an option to send the generated report via email for further review and collaboration.

7.10 FAQ Page

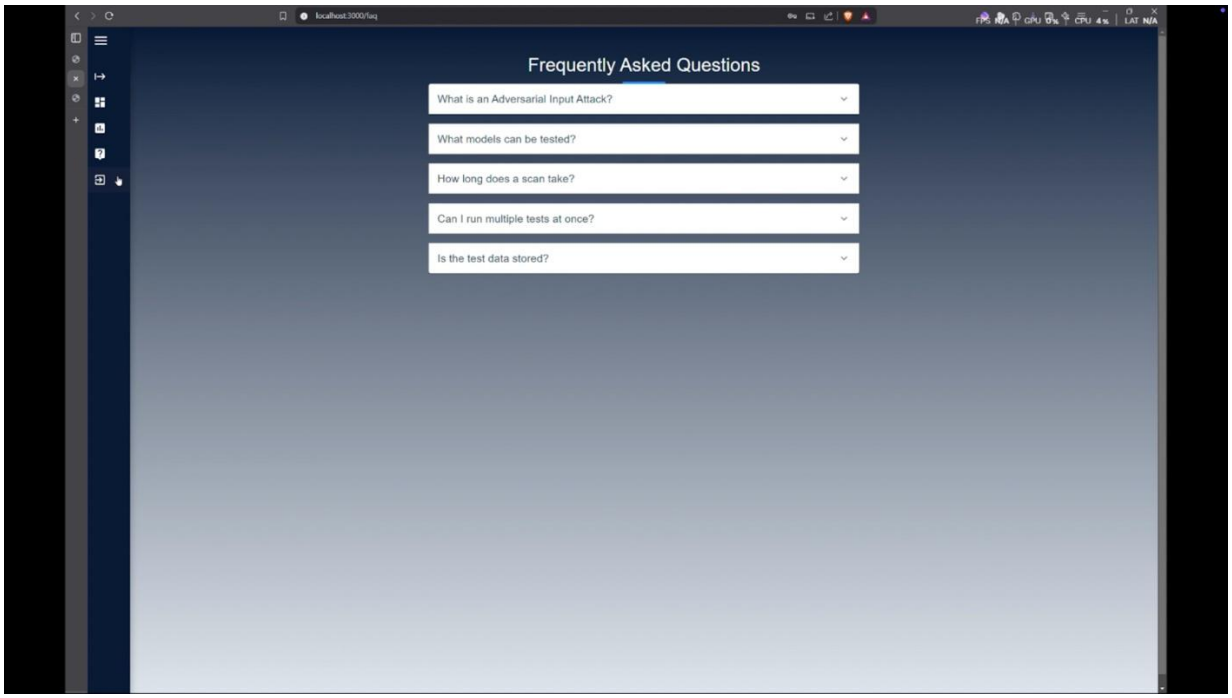

Figure 27 FAQ Page

This shows the Frequently Asked Questions (FAQ) page within Protego. It provides users with detailed answers and guidance on common queries related to the scanner's functionality, usage, and troubleshooting.

8. Requirement matrix

Requirement ID	Requirement Description	Mapped component
FR-01	Detect query-based model inversion attacks (score-based, boundary-based).	Scanning Engine
FR-02	Quantify attack impact and suggest mitigations.	Risk Assessment and reporting
FR-03	Monitor AI/ML models to detect anomalies or threats.	Monitoring Module

Requirement ID FR-01: Detect model inversion vulnerabilities.

Objective: The system must detect vulnerabilities linked to model inversion attacks, in which adversaries extract sensitive information from trained models.

Mapped Component: Scanning Engine

This component will use powerful algorithms to scan AI/ML models, assess their behaviour, and identify potential model inversion risks.

Key capabilities include pattern recognition, sensitivity analysis, and anomaly detection in trained models.

Requirement ID: FR-02 Provide risk evaluations and recommendations for mitigation techniques.

Objective: Provide extensive assessments of identified risks, quantify their severity using risk ratings, and recommend practical mitigation methods.

Mapped Component: Risk Assessment and Reporting

This module will collect risk data, compute risk levels based on predefined parameters, and provide detailed reports with prioritized suggestions.

Severity rankings, risk dashboard visualizations, and personalized action plans are among the key outputs.

Requirement ID FR-03: Monitor AI/ML models.

Objective: Continuously monitor model performance and identify abnormalities.

Mapped Component: Monitoring Module

Key features include anomaly detection and seamless connection with the scanning engine.

Document Update Notice
This **Software Design Document** has been updated as of **25 March 2025**.