Real-Time Network Intrusion Detection Using Wireshark and Advanced Ensemble Learning Techniques

Design Document

Version 1.0

Group Id: **F24PROJECTA686A**
Supervisor Name: **Laraib Sana**

Revision History

Date (dd/mm/yyyy)	Version	Description	Author
13/01/25	1.0	First version of design document for the project “real-time network intrusion detection system using Wireshark and machine learning models”. This document cover all demanded design elements.	BC210427835

Table of Contents

1. Introduction of Design Document

A. Overview

The purpose of this design document is to outline the architectural and design considerations of my project Real-Time Network Intrusion Detection Using Wireshark and Advanced Ensemble Learning Techniques and its Django Web Application. This document provides a detailed structure of the system, including its entities, relationships, components, and interactions. It serves as a blueprint for developers, ensuring a systematic and well-structured approach to software development.

B. Scope

My this Web Application is designed to allow users to securely upload network traffic capture files (.pcap/.pcapng), convert them into CSV format, and analyze the data using machine learning models. The Machine learning models used in this project are TabNet, CatBoost, LightGBM. The results are then processed using ensemble techniques and presented in a user-friendly report, which can be downloaded in PDF format. The system also incorporates user authentication, file management, intrusion detection, and result logging to enhance security and usability.

C. Purpose of the Design Document

This document serves several key purposes:

1. **Provide a Clear System Structure**: Define the key entities, relationships, and workflows within the system.
2. **Guide to the Development Process**: Serve as a reference point for developing the project as well as for university supervisor to know my clarity in project.
3. **Prevents Costly Errors**: Another purpose of well-documented design reduces the risk of errors during development and deployment.
4. **Improves Code Quality**: Establishing a clear structure ensures clean, modular, and maintainable code.
5. **Optimizes Performance**: Well-designed architecture ensures efficient data flow and processing.
6. **Ensuring Consistency**: Helps maintain standardization across the entire development lifecycle.

D. Key Components Included in this Document

This document will cover the following essential design aspects:

1. **Entity Relationship Diagram (ERD)**: ERD shows entities and relationships between entities. I used Chen notion for the ERD of my project.
2. **Sequence Diagrams**: Illustrates step-by-step interactions between users and system components, along with the lifeline.
3. **Architecture Design Diagram**: Provides a high-level view of system components and their interactions.
4. **Class Diagram**: Represents system classes, attributes, methods, and relationships following OOP principles.
5. **Database Design**: Used crow foot notation to illustrate the details of database schema at physical level for efficient data management.
6. **Interface Design**: Showcases the graphical user interface (GUI) structure of my Django application.
7. **Test Cases**: Test scenarios to validate system functionality and performance.

2. Entity Relationship Diagram (ERD)

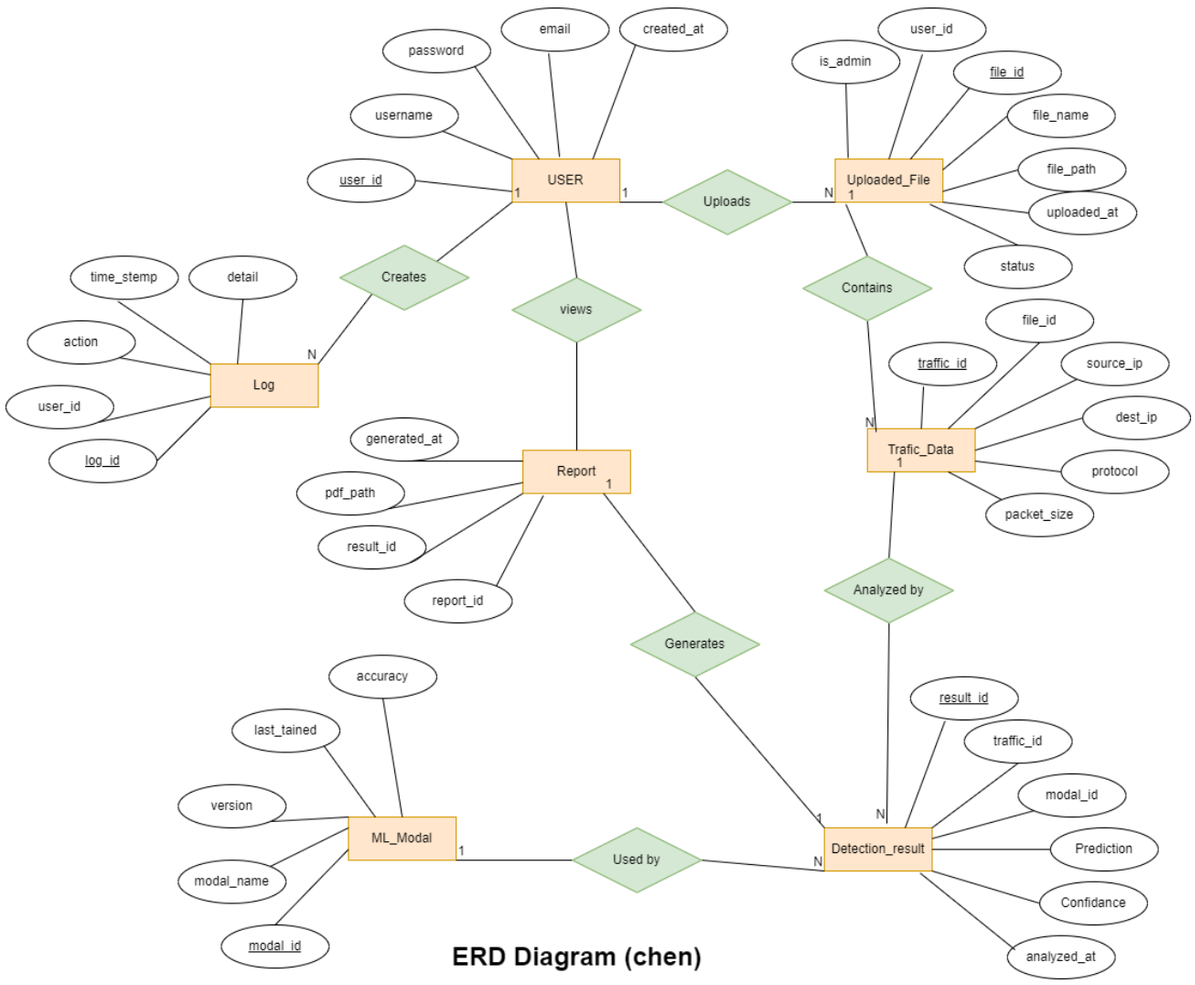

Figure 1Entity Relationship diagram in chen notation

3. Sequence Diagram

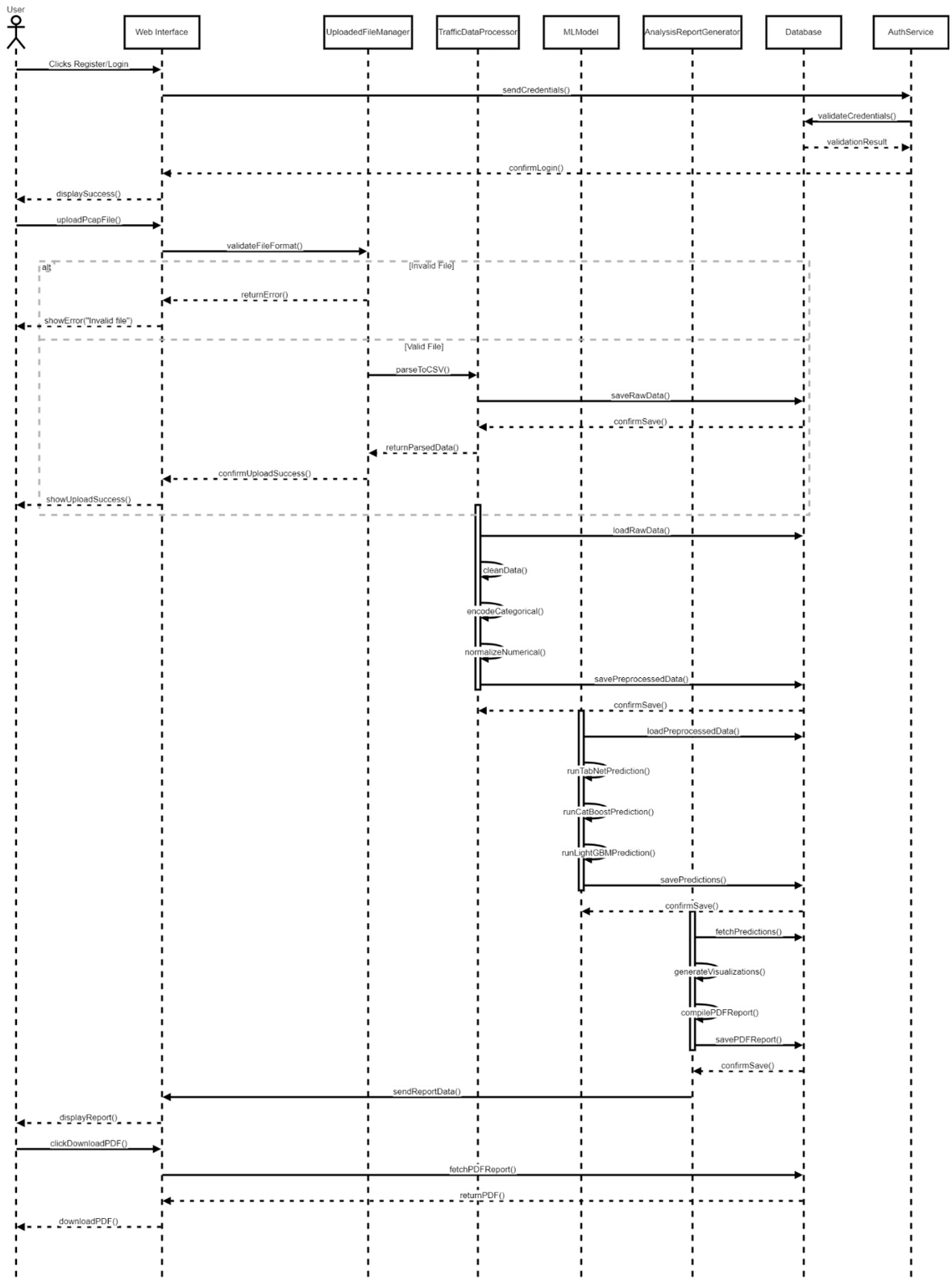

Figure 2 Sequence Diagram

4. Architecture Design Diagram

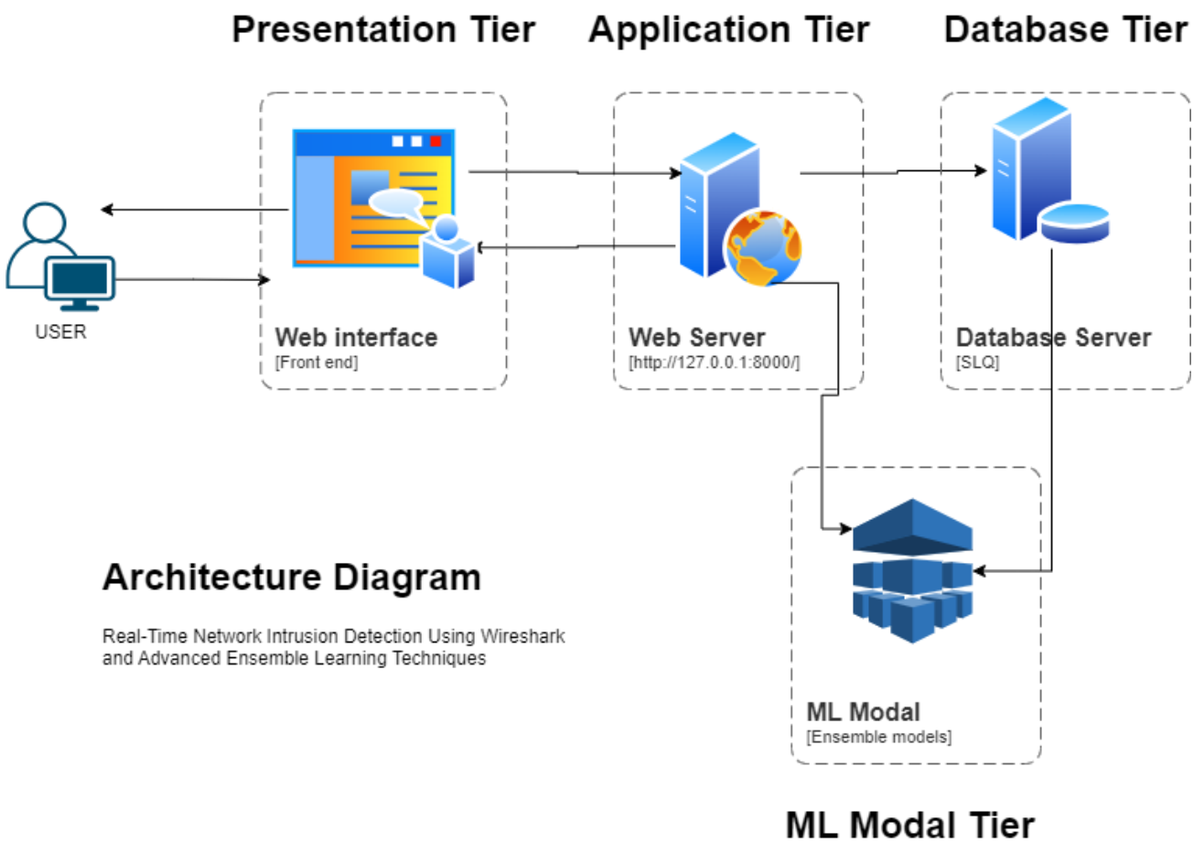

Figure 3 Architecture Design Diagram

5. Class Diagram

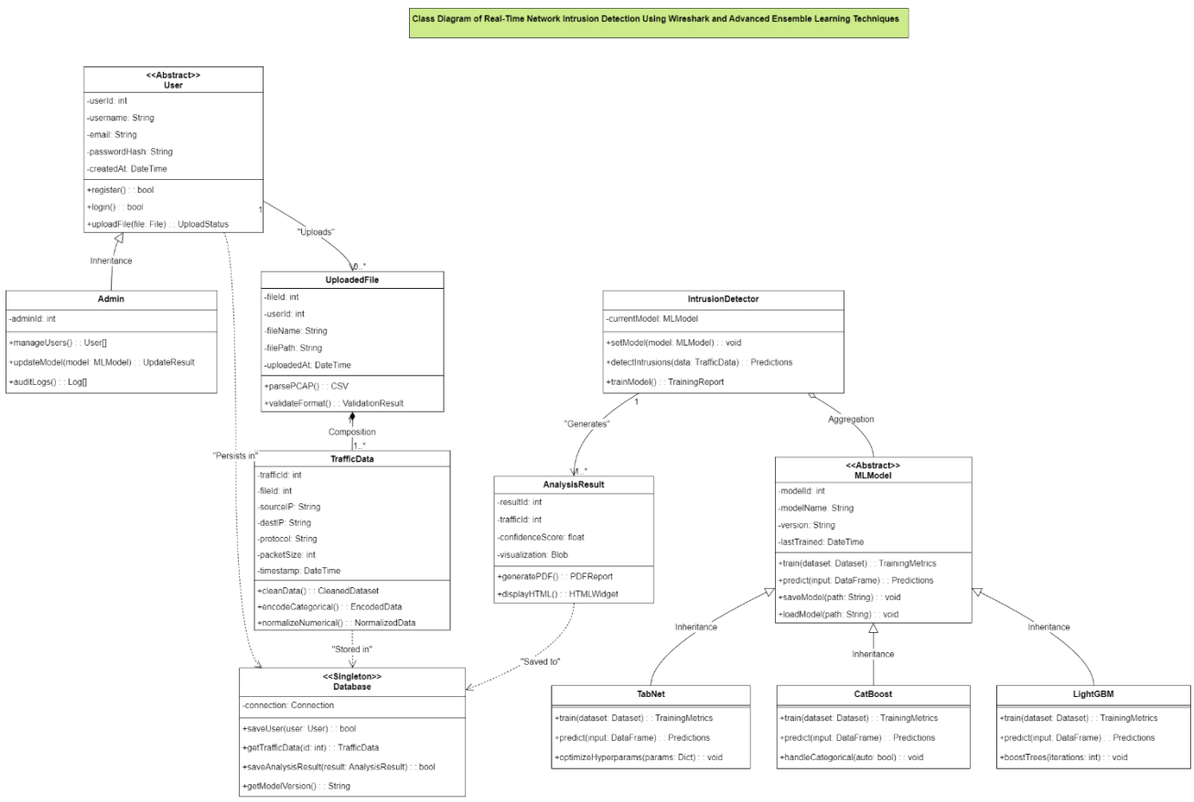

Figure 4 Class Diagram

6. Database Design

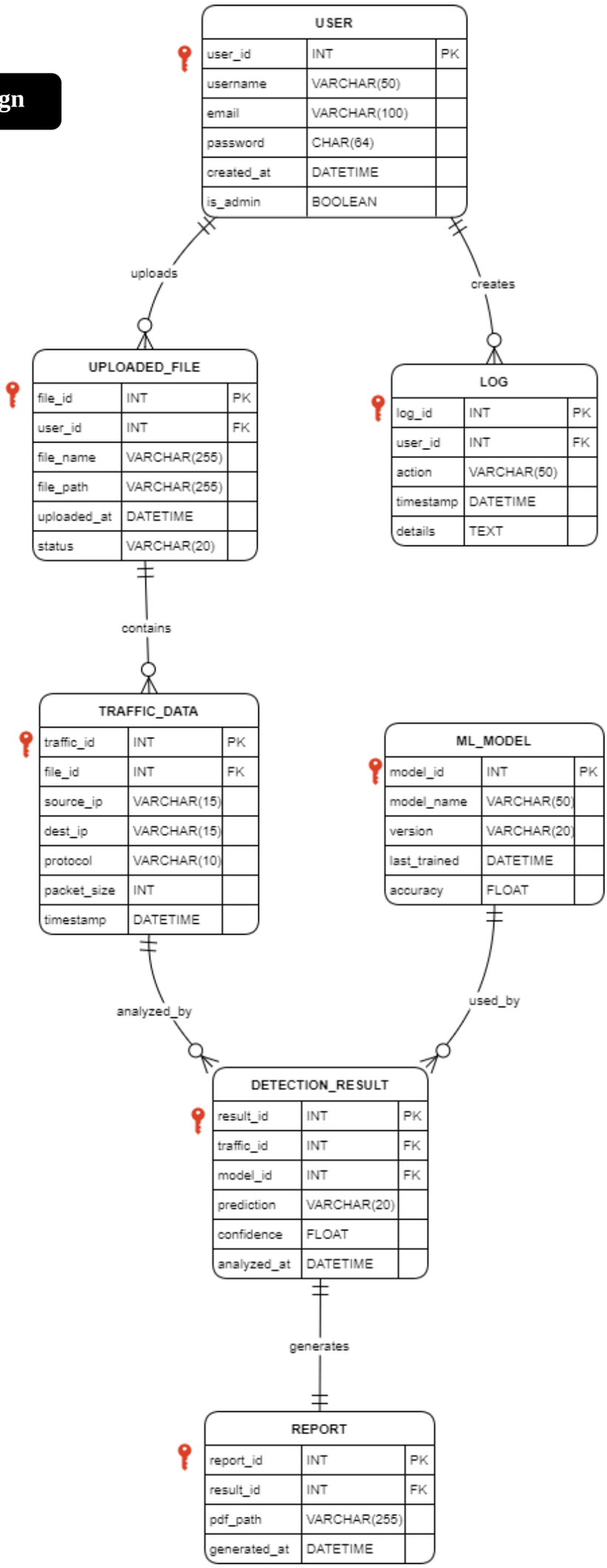

Figure 5 Database Design

7. Interface Design

Figure 6 Home.html

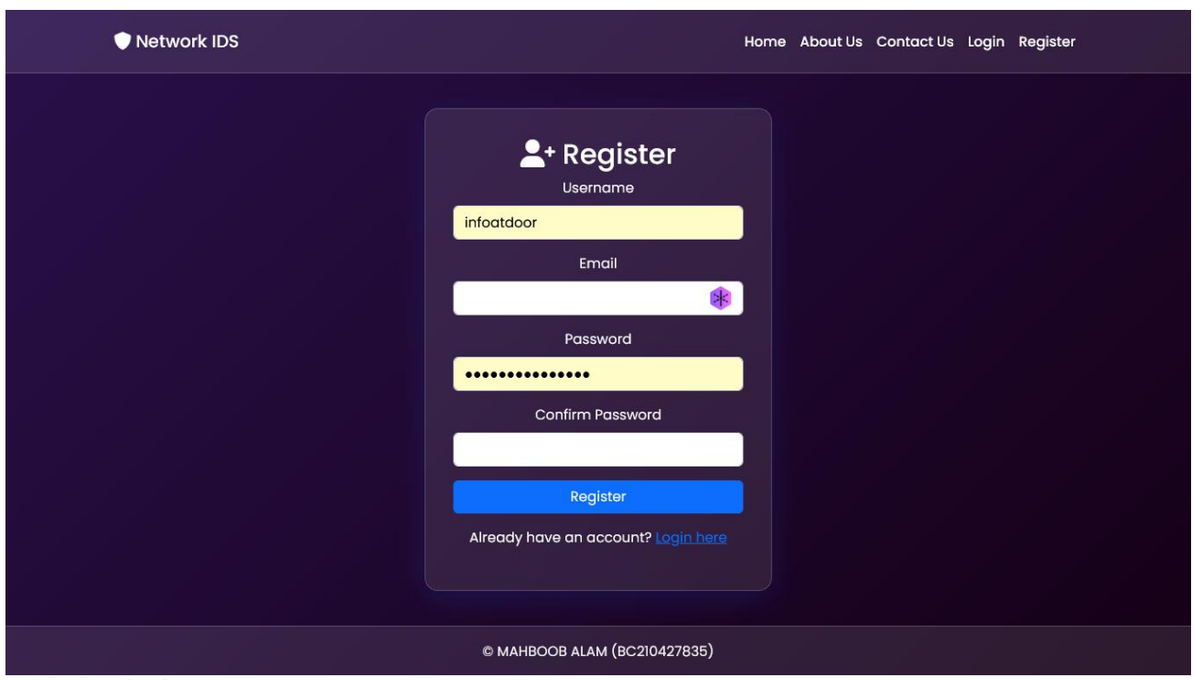

Figure 7 Register.html

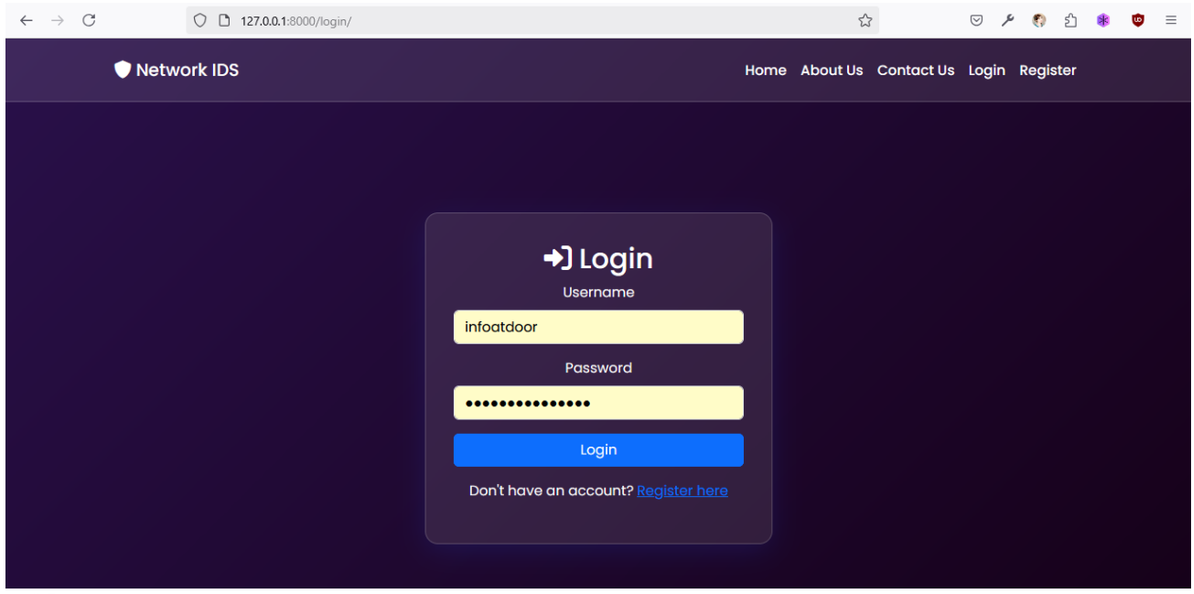

Figure 8 Login.html

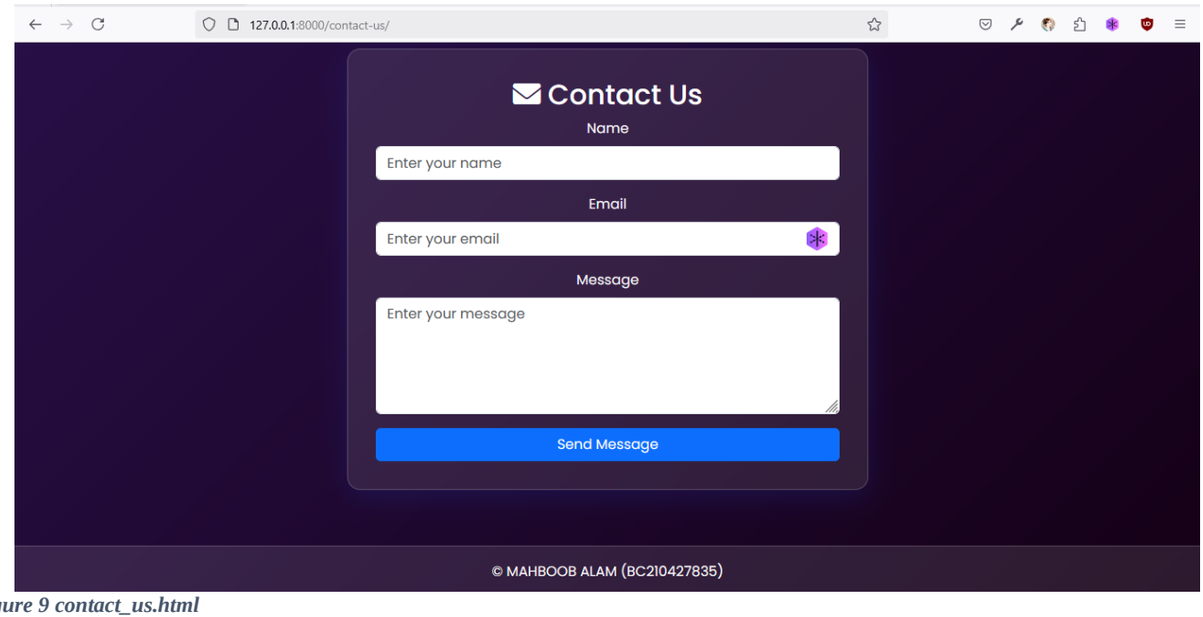

Figure 9 contact_us.html

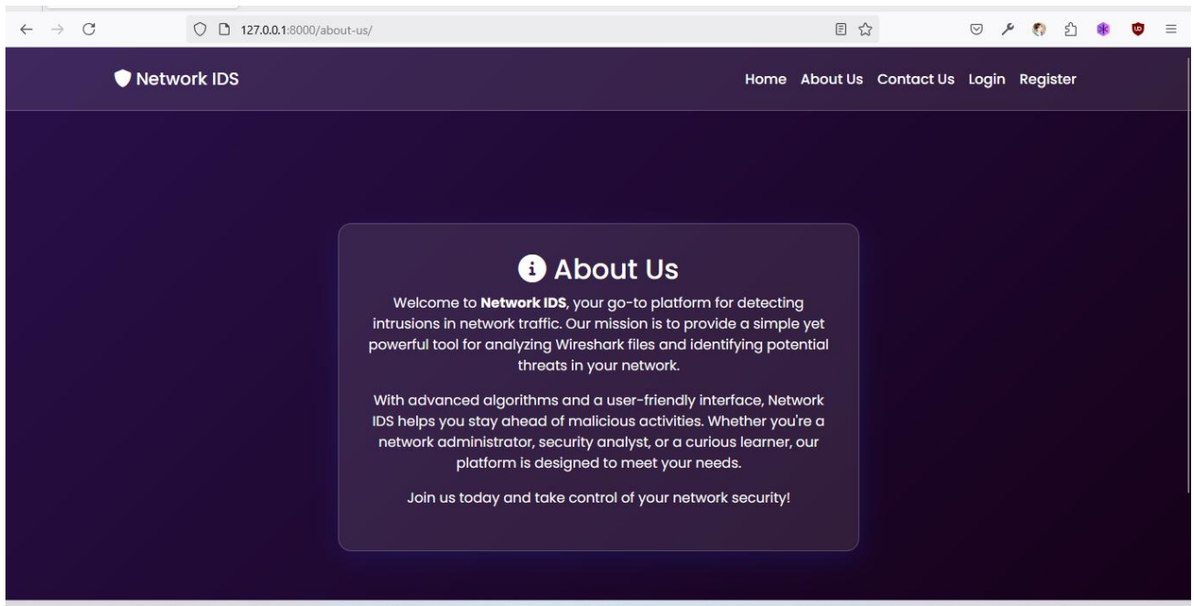

Figure 10 about_us.html

8. Test Cases

Table 1: Test Case User Registration

Test Case Name	User Registration
Test Case ID	TC-01
Preconditions	None.
Steps	1. Navigate to the registration page. 2. Enter valid details (username, email, password). 3. Click 'Register'.
Expected Result	User is registered, and a confirmation message is displayed.
Authority	User
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	The user successfully registers an account using valid credentials. The system ensures that all required fields are filled and validated.

Table 2: Test Case User Login

Test Case Name	User Login
Test Case ID	TC-02
Preconditions	User is registered.
Steps	1. Navigate to the login page. 2. Enter valid credentials (username/email, password). 3. Click 'Login'.
Expected Result	User is logged in, and the dashboard is displayed.
Authority	User
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	A registered user logs into the system using correct credentials. The system authenticates the user and grants access.

Table 3: Test case Upload file

Test Case Name	Upload .pcap File
Test Case ID	TC-03
Preconditions	User is logged in.
Steps	1. Navigate to the file upload page. 2. Select a valid '.pcap' file. 3. Click 'Upload'.
Expected Result	File is uploaded, and a success message is displayed.
Authority	User
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	The user uploads a valid network capture file for intrusion detection analysis. The system processes the file successfully.

Table 4: Test case file validation

Test Case Name	Uploaded Invalid File
Test Case ID	TC-04
Preconditions	User is logged in.
Steps	1. Navigate to the file upload page. 2. Select an invalid file (any format other than '.pcap'). 3. Click 'Upload'.
Expected Result	System displays an error message (e.g., 'Invalid file format').
Authority	User
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	The system rejects unsupported file formats and notifies the user with an appropriate error message.

Table 5: Test case Preprocess Data

Test Case Name	Preprocess Data
Test Case ID	TC-05
Preconditions	A valid '.pcap' file is uploaded.
Steps	1. The System parses the '.pcap' file into CSV format. 2. The system cleans the data (e.g., removes missing values). 3. The system encodes categorical features (e.g., protocol types). 4. The System normalizes numerical features (e.g., packet sizes).
Expected Result	Preprocessed data is saved in the database.
Authority	System
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	The uploaded file is converted into a structured dataset, ready for machine learning analysis.

Table 6: Test case ML Analysis

Test Case Name	Run ML Analysis
Test Case ID	TC-06
Preconditions	Preprocessed data is available.
Steps	1. System loads preprocessed data. 2. System runs predictions using TabNet, CatBoost, and LightGBM. 3. System saves predictions and confidence scores.
Expected Result	Predictions are saved in the database with accuracy > 90%.
Authority	System
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	Machine learning models analyze network traffic data to detect potential intrusions.

Table 7: Test case Generate Report

Test Case Name	Generate Report
Test Case ID	TC-07
Preconditions	Predictions are available.
Steps	1. System fetches predictions from the database. 2. System generates visualizations (e.g., traffic patterns, affected IPs). 3. System compiles a PDF report.
Expected Result	Report is generated and saved in the database.
Authority	System
Modification History	1.0
Author	F24PROJECTA686A (BC210427835)
Description	The system generates an analysis report containing details about detected intrusions.