A Major Project Report

On

HYBRID MACHINE LEARNING BASED FOOD ADULTERATION DETECTION SYSTEM

Submitted to CMREC (UGC Autonomous)

In Partial Fulfilment of the requirements for the Award of Degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING (AI & ML)

Submitted

By

CH. SIDHARTHA REDDY	**(228R1A6674)**
G. KEERTHI	**(228R1A6680)**
J. NAGALAXMI	**(228R1A6692)**
SHAIKH MUJEEB ARIF	**(228R1A66C1)**

Under the Esteemed guidance of

Mrs. N. SWAROOPA

Assistant Professor, Department of CSE(AI&ML)

Department of Computer Science and Engineering(AI&ML)

CMR ENGINEERING COLLEGE

(UGC AUTONOMOUS)

(Accredited by NAAC & NBA, Approved by AICTE, New Delhi, Affiliated to JNTU, Hyderabad)
(Kandlakoya, Medchal Road, Medchal-Malkajgiri Dist., Hyderabad-501 401)

(2025-2026)

CMR ENGINEERING COLLEGE

UGC AUTONOMOUS

(Accredited by NAAC&NBA, Approved by AICTE New Delhi, Affiliated to JNTU, Hyderabad, Kandlakoya, Medchal Road, Hyderabad-501 401)

Department of Computer Science & Engineering (AI & ML)

CERTIFICATE

This is to certify that the Major project entitled **"Hybrid Machine Learning–Based Food Adulteration Detection System"** is a bonafide work carried out by

CH. SIDHARTHA REDDY	**(228R1A6674)**
G. KEERTHI	**(228R1A6680)**
J. NAGALAXMI	**(228R1A6692)**
SHAIKH MUJEEB ARIF	**(228R1A66C1)**

in partial fulfillment of the requirement for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING (AI&ML) from CMR Engineering College, under our guidance and supervision.

The results presented in this Major project have been verified and are found to be satisfactory. The results embodied in this Major project have not been submitted to any other university for the award of any other degree or diploma.

Internal Guide	**Major Project Coordinator**	**Head of the Department**
Mrs. N. Swaroopa	Mr. G. Venkateswarlu	Dr. Madhavi Pingili
Assistant Professor	Assistant Professor	Professor & HOD
Department of CSE (AI & ML)	Department of CSE (AI & ML)	Department of CSE (AI & ML)

External Examiner: _____

DECLARATION

This is to certify that the work reported in the present Major project entitled "**Hybrid Machine Learning–Based Food Adulteration Detection System**" is a record of bonafide work done by us in the Department of Computer Science and Engineering (AI & ML), CMR Engineering College. The reports are based on the Major project work done entirely by us and not copied from any other source. We submit our Major project for further development by any interested students who share similar interests to improve the Major project in the future.

The results embodied in this Major project report have not been submitted to any other University or Institute for the award of any degree or diploma to the best of our knowledge and belief.

CH. SIDHARTHA REDDY (228R1A6674)
G. KEERTHI (228R1A6680)
J. NAGALAXMI (228R1A6692)
SHAIKH MUJEEB ARIF (228R1A66C1)

ACKNOWLEDGEMENT

We are extremely grateful to **Dr. A. Srinivasula Reddy**, Principal and **Dr. Madhavi Pingili**, Professor & HOD, Department of CSE(AI&ML), CMR Engineering College for their constant support.

We are extremely thankful to **Mrs. N. Swaroopa,** Assistant Professor, Internal Guide, Department of CSE(AI&ML), for his constant guidance, encouragement and moral support throughout the Major project.

We will be failing in duty if we do not acknowledge with grateful thanks to the authors of the references and other literatures referred in this Major project.

We thank **Mr. G. Venkateswarlu,** Major Project Coordinator for his constant support in carrying out the Major project activities and reviews.

We express our thanks to all staff members and friends for all the help and co-ordination extended in bringing out this Major project successfully in time.

Finally, We are very much thankful to our parents who guided us for every step.

CH. SIDHARTHA REDDY (228R1A6674)
G. KEERTHI (228R1A6680)
J. NAGALAXMI (228R1A6692)
SHAIKH MUJEEB ARIF (228R1A66C1)

CONTENTS

ABSTRACT

Food adulteration poses significant risks to consumer health, food authenticity, and supply chain transparency. Traditional laboratory techniques such as chromatography and spectroscopy offer high accuracy but remain expensive, time-consuming, and impractical for large-scale or real-time monitoring. Recent developments in artificial intelligence have enabled faster food quality assessment, yet most existing methods depend on single-modality inputs or individual machine learning models, resulting in limited robustness and poor adaptability to variations in food composition and adulterant types. To overcome these limitations, this study proposes a Hybrid Machine Learning–Based Food Adulteration Detection System that utilizes the *Food Adulteration Dataset* containing tabular chemical, physical, and quality-inspection features.

The proposed system incorporates a multi-stage pipeline including data cleaning, feature engineering, and model optimization, followed by the use of multiple base classifiers such as Random Forest, Support Vector Machine, and Gradient Boosting. These models are fused using a voting or stacking ensemble to generate a final, more reliable adulteration prediction.

The system outputs not only whether a sample is pure or adulterated but also provides a confidence score and severity indication. Experimental evaluation demonstrates that the hybrid ensemble significantly enhances prediction accuracy and generalization compared to standalone classifiers. The results show that the proposed approach offers a scalable, cost-effective, and practical solution for real-time food adulteration detection, supporting food producers, quality inspectors, and regulatory agencies in ensuring consumer safety and product integrity.

Keyword: Food adulteration detection; Hybrid machine learning; Ensemble model; Tabular dataset; Feature engineering; Random Forest; Support Vector Machine; Gradient Boosting; Classification; Food quality assessment.

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Introduction

Food adulteration has increasingly become a pervasive and complex challenge within modern food supply chains. The rapid globalization of food production, coupled with rising consumer demand for processed and packaged food items, has heightened opportunities for intentional and unintentional adulteration. Economically motivated adulteration—including the addition of low-cost substitutes, dilution of essential components, substitution with inferior or harmful materials, and chemical enhancement to mimic freshness—has grown significantly in recent years. Such practices undermine the nutritional integrity of food products, distort market fairness, and pose severe risks to public health, including allergic reactions, chronic illness, and toxic exposure [1]. Beyond health hazards, adulteration damages consumer trust and leads to substantial economic losses across the entire food industry, from producers and distributors to retailers and regulatory bodies.

Traditional laboratory-based analytical methods remain the gold standard for identifying adulterants due to their high precision and sensitivity. Techniques such as high-performance liquid chromatography (HPLC), gas chromatography, spectroscopy, and chemical reagent-based assays are widely used to detect contaminants and evaluate purity levels. However, despite their accuracy, these methods have several limitations: they require expensive equipment, well-controlled laboratory environments, skilled technicians, and long processing times. As a result, they are impractical for rapid, large-scale screening or on-site quality assessment in dynamic supply chain environments where quick decision-making is crucial [2].

1.2 Project Objectives

By the completion of this project, the system demonstrates the following capabilities:

1. To develop a hybrid machine learning model that integrates multiple classifiers for improved food adulteration detection.
2. To enhance classification accuracy and robustness compared to single-model systems using tabular food inspection data.
3. To build a complete data-processing and prediction pipeline, including cleaning, feature engineering, and model fusion.
4. To evaluate the performance of individual base models (RF, SVM, Gradient Boosting) and compare them with the hybrid ensemble.

1.3 Purpose of the Project

The purpose of this project is to develop an intelligent and structured framework capable of accurately identifying cyberbullying and hate speech in online text. The system implements a well-defined preprocessing and analysis pipeline that enables reliable categorization of harmful content using machine learning and natural language processing techniques. Such automated detection frameworks assist content moderation systems by identifying abusive language at scale and improving the safety of online communities **[3]**. The proposed approach supports integration with automated moderation solutions aimed at improving user safety across digital platforms and ensuring more responsible digital communication environments.

1.4 Existing System

Existing systems for food adulteration detection largely rely on traditional laboratory-based analytical methods or single-model machine learning approaches. Laboratory techniques such as chromatography, FTIR/NIR spectroscopy, titration, reagent-based tests, and chemical assays are widely used because of their high accuracy in identifying adulterants. However, these methods are often restricted to controlled laboratory environments and require skilled technicians, specialized equipment, and significant processing time. In recent years, machine learning models have been applied to tabular data derived from chemical and physical properties of food samples. These systems generally use individual algorithms such as SVM, Random Forest, KNN, or Logistic Regression to classify samples as adulterated or pure. While these models show promise, they typically rely on single-modality numeric data and lack the ability to effectively generalize across diverse food types and adulterant variations. As a result, their accuracy and adaptability decrease when exposed to noisy, imbalanced, or real-world datasets.

Disadvantages

1. High Dependency on Laboratory Equipment

Laboratory-based methods require expensive instruments and trained operators, making them unsuitable for large-scale or on-site applications.

2. Time-Consuming and Costly

Traditional chemical and spectral tests involve lengthy procedures and repeated measurements, increasing operational time and cost.

3. Single-Model Machine Learning Approaches Are Limited

ML models like SVM or RF are used independently, which reduces performance when dealing with diverse adulterants or varying sample conditions.

4. Low Robustness and Generalization

Most existing systems lose accuracy under noisy, imbalanced, or heterogeneous

datasets, making them unreliable in real-world environments.

1.5 Proposed System

The proposed system is a Hybrid Machine Learning–Based Food Adulteration Detection System that integrates multiple machine learning models to improve prediction accuracy and robustness. Using the Kaggle Food Adulteration Dataset, which contains structured chemical, physical, and inspection-related features, the system performs a complete ML pipeline involving data cleaning, normalization, encoding, feature engineering, and splitting. Multiple base classifiers—such as Random Forest, Support Vector Machine, Logistic Regression, and Gradient Boosting—are trained independently to capture different aspects of adulteration patterns. The predictions of these models are then combined using ensemble techniques (voting or stacking) to form a hybrid classifier that delivers more reliable and stable decisions.

The system also produces additional information such as adulteration severity levels and confidence scores. This hybrid approach reduces dependency on laboratory equipment and provides a cost effective, scalable, and real-time solution for food quality inspection and monitoring across various sectors.

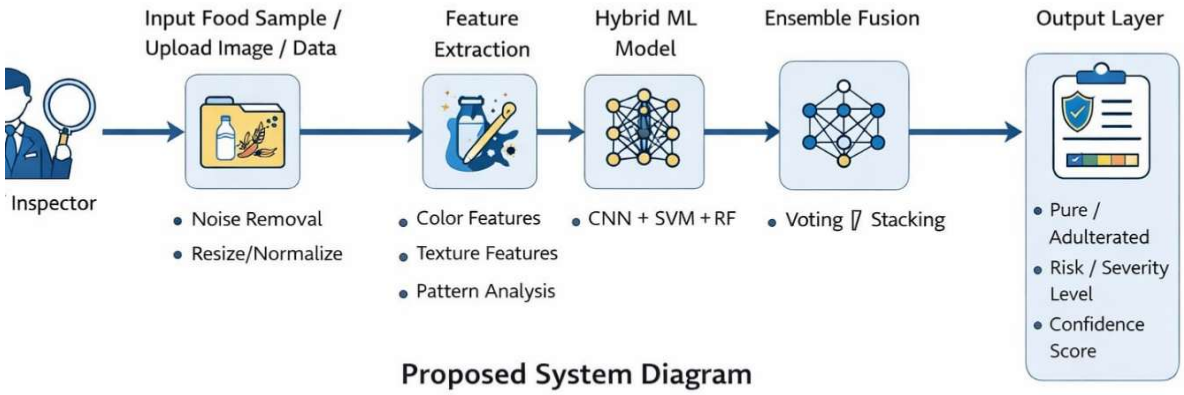


Fig 1.5.1: Block diagram of proposed system.

Advantages

1. Higher Accuracy Through Hybrid Modeling

Combining multiple ML algorithms enhances prediction performance beyond what a

single model can achieve.

2. Improved Robustness and Generalization

Ensemble learning reduces the impact of noise, imbalance, and variability in food samples, making the system more reliable in real-world scenarios.

3. Cost-Effective and Practical

Uses tabular food inspection data rather than expensive laboratory equipment or complex imaging systems, reducing operational costs.

4. Comprehensive Output Information

Provides adulteration status, severity level, and confidence score, enabling better decision-making for quality inspectors and authorities.

5. Scalable and Easy to Deploy

Can be integrated into web/desktop applications for real-time testing and is adaptable to different food categories without requiring major system changes.

1.6 Input and Output Design

1.6.1 Input Design

Input design is the bridge between the information system and the user, ensuring that data is captured, converted, and entered into the system in a format suitable for processing. It defines the procedures, rules, and specifications required for collecting raw data and transforming it into meaningful input. Data can be entered manually by users through keyboards or forms, or automatically through devices such as scanners, sensors, or by importing from existing documents and files. The purpose of input design is to ensure that the system receives correct, complete, and timely data in an efficient manner.

A good input design reduces the complexity of data entry and improves overall system performance. It focuses on minimizing the amount of data that needs to be entered, reducing duplication, preventing errors at the point of entry, and ensuring that the process is as simple and fast as possible. It also aims to reduce the workload of users by providing clear instructions and structured input formats. Proper input design enhances data quality, improves processing speed, and supports better decision-making by ensuring reliable information is stored in the system.

Security and privacy are also important considerations in input design. The system should ensure that only authorized users can enter or modify data, and sensitive information should be protected from unauthorized access. Input controls such as login authentication, access levels, and encryption may be used to maintain data security and integrity.

Effective input design also includes the design of user interfaces such as forms, screens, and dialogs that guide users step by step during data entry. These interfaces should be simple,

consistent, and easy to understand to avoid confusion and reduce errors.

Objectives

- A Data collection source identification: Determining whether data comes from manual entry, documents, sensors, or existing databases.
  ● Input format standardization: Ensuring consistency in formats such as dates, codes, numeric values, and text fields.
  ● Data validation techniques: Applying checks such as range checks, format checks, type checks, and consistency checks to ensure correctness of input.
  ● Error detection and correction: Providing immediate feedback to users when incorrect data is entered and allowing easy correction.
  ● User guidance mechanisms: Using prompts, tooltips, menus, and help messages to assist users during input.
  ● Efficiency considerations: Reducing keystrokes, using dropdowns, auto-fill features, and default values to speed up data entry.

1.6.2 Output Design

A quality output is one that satisfies the requirements of the end user and presents information in a clear, accurate, and meaningful manner. In any information system, the results of processing are communicated to users and other systems through outputs. Output design determines how information is displayed for immediate use as well as how it is produced in hard copy form such as reports and printed documents. It is one of the most important components of a system because it is the primary and direct source of information for decision-making.

Effective output design ensures that the right information is delivered to the right user at the right time in the right format. It improves communication between the system and its users and enhances the usability of the system. A well-designed output supports better analysis, faster understanding, and improved decision-making. It also reduces confusion by presenting information in an organized and structured way.

The process of designing computer output should be systematic and well planned. The system analyst must carefully identify what outputs are required to meet user needs. Each output element should be designed in such a way that users can easily interpret and utilize the information without difficulty. The design should focus on clarity, relevance, accuracy, and simplicity.When analyzing and designing computer output, the following steps are generally followed:

1. Identification of required outputs: Determining what information is needed by users and what reports or displays should be generated by the system.

2. Selection of presentation methods: Choosing how information will be displayed, such as on-screen reports, printed documents, dashboards, charts, or graphs.

3. Design of output formats: Creating structured documents, reports, and other formats that clearly present the processed data in a meaningful way.

4. The output of an information system should achieve one or more of the following objectives:

5. Convey information about past activities, current status, or future projections in an understandable manner.

6. Highlight important events, opportunities, problems, or warnings that require attention.

7. Trigger appropriate actions by users or other systems based on the information provided.

8. Confirm completed actions or processes to ensure accuracy and reliability.

2. LITERATURE SURVEY

1. **Stevens L. J. et al., "Presence of Artificial Food Dyes and Sugar in Children's Foods," 2015.** This study investigates the levels of artificial food dyes and added sugars present in children's food products. Quantitative analysis revealed that many food items marketed for children contain significant amounts of artificial dyes and sugars. The findings highlight potential health concerns associated with excessive consumption of such additives.
2. **Zhou C. et al., "Stabilizing Meat Colour Using Ligand Coordination with Hemin," 2014.** This research focuses on improving meat colour stability through ligand coordination with hemin. The proposed method enhances the visual appearance and shelf-life of meat products by maintaining stable coloration. Experimental results demonstrate improved meat colour retention compared to conventional preservation methods.
3. **Tian H. et al., "Discrimination of Chicken vs. Beef Seasonings Using Electronic Nose," 2014.** This study employs an electronic nose combined with sensory evaluation to differentiate between chicken and beef seasonings. The method successfully distinguishes food seasonings with high reliability. Results validate the effectiveness of electronic nose technology in food classification tasks.
4. **Barbin D. F. et al., "Infrared Spectral Analysis for Coffee Quality Assessment," 2014**. The authors explore infrared spectral techniques for evaluating coffee quality. Infrared spectroscopy proved effective in analyzing coffee composition and detecting quality-related characteristics. The approach offers a rapid and non-destructive alternative for coffee quality inspection.
5. **Dixit S. et al., "Exposure Risk of Food Colours in India," 2013.** This nationwide survey assesses the exposure risk of artificial food colours in Indian sweets and savory foods. The analysis indicates that high consumption of artificially coloured foods significantly increases exposure risk among the Indian population. The study emphasizes the need for stricter regulation of food colouring agents.

6. **Ates E. et al., "Detection of Legal and Illegal Food Colours Using LC/MS with Cloud Point Extraction," 2011**. This research presents a validated LC/MS method combined with cloud point extraction for detecting food colours. The technique accurately identifies both permitted and non-permitted food colour additives across various food matrices. The method improves regulatory monitoring and food safety assessment.
7. **Boga A. and Binokay S., "Effects of Food Additives on Human Health," 2010**. This review-based medical analysis examines the health impacts of food additives. The study concludes that excessive intake of food additives may negatively affect metabolic and physiological functions. It highlights the long-term health risks associated with additive-rich diets.
8. **Uzan A. and Delaveau P., "Salt Content in Foods as a Public Health Concern," 2009**. This public health analysis investigates the salt content of processed foods and its impact on health. The findings identify excessive salt consumption as a significant public health issue contributing to hypertension and cardiovascular diseases. The study advocates for improved nutritional regulation.
9. **Fik M. et al., "Shelf-Life Improvement of Ground Beef Using Potassium Lactate and Sodium Diacetate," 2008.** This study evaluates chemical treatment methods for improving the shelf-life of refrigerated ground beef. Potassium lactate and sodium diacetate were found to enhance preservation and maintain meat quality over extended storage periods. The treatment significantly improved product stability.
10. **Brosnan T. and Sun D. W., "Computer Vision in Food Quality Inspection," 2004**. This review examines the application of computer vision techniques in food quality inspection. The study concludes that computer vision significantly improves automation, speed, and accuracy in food quality analysis. It establishes computer vision as a valuable technology for modern food inspection systems.

Focused Area / Title	**Key Findings**	**Reference**
Artificial Food Dyes in Children's Foods	Children's food products contain significant levels of artificial dyes and added sugars, raising health concerns.	Stevens LJ et al., "Artificial Food Dyes and Sugar in Children's Foods," 2015.
Stabilizing Meat Colour	Ligand-hemin coordination improves meat colour stability and enhances appearance retention in processed meat products.	Zhou C et al., "Stabilizing Meat Colour Using Ligand Coordination with Hemin," 2014.
Discrimination of Chicken vs. Beef Seasonings	Electronic nose combined with sensory evaluation effectively differentiates between chicken and beef seasonings.	Tian H et al., "Discrimination of Chicken vs. Beef Seasonings," 2014.
Infrared Spectral Analysis for Coffee Quality	Infrared spectroscopy efficiently evaluates coffee composition and quality characteristics.	Barbin DF et al., "Infrared Spectral Analysis for Coffee Quality," 2014.
Exposure Risk of Food Colours in India	High consumption of artificial food colours in sweets and savory foods increases exposure risk among the Indian population.	Dixit S et al., "Exposure Risk of Food Colours in India," 2013.

Table no. 2 Literature Review Summary

Focused Area / Title	Key Findings	Reference
Detection of Legal/Illegal Food Colours	LC/MS with cloud point extraction accurately detects permitted and non-permitted food colours in food matrices.	Ates E et al., "Detection of Legal and Illegal Food Colours Using LC/MS," 2011.
Effects of Food Additives on Human Health	Food additives may negatively impact human health and contribute to metabolic and physiological disorders.	Boga A & Binokay S, "Effects of Food Additives on Human Health," 2010.
Salt Content in Foods as Public Health Concern	High salt content in processed foods is a significant public health issue affecting consumers.	Uzan A & Delaveau P, "Salt Content in Foods as Public Health Concern," 2009.
Shelf-Life Improvement of Ground Beef	Potassium lactate and sodium diacetate improve preservation and refrigerated shelf-life of ground beef.	Fik M et al., "Shelf-Life Improvement of Ground Beef," 2008.
Computer Vision in Food Quality Inspection	Computer vision improves automation, speed, and accuracy in food quality inspection processes.	Brosnan T & Sun DW, "Computer Vision in Food Quality Inspection," 2004.

Table no. 2 Literature Review Summary

3. SOFTWARE REQUREIMENTS ANALYSIS

3.1 Problem Statement

The proposed food adulteration detection system is designed around a structured machine learning pipeline, consisting of five key components: Data Acquisition, Pre-Processing, Feature Extraction, Hybrid Modeling, and Output Layer. Each component plays a crucial role in transforming raw data into accurate and actionable results for food quality assessment.

1. **Data Acquisition**: This stage involves gathering raw data from various sources such as laboratory tests, sensors, and existing food quality databases. Ensuring the data is comprehensive, diverse, and representative of different food types is essential for building a robust system.

2. **Pre-Processing**: Raw data often contains noise, missing values, or inconsistencies. Pre-processing cleans and standardizes the data by handling missing entries, normalizing numerical values, encoding categorical features, and removing outliers. This step ensures that the data is suitable for machine learning algorithms and improves the reliability of subsequent analysis.

3. **Feature Extraction**: In this stage, key attributes that are most indicative of food adulteration are identified and transformed into features suitable for modeling. Techniques may include statistical analysis, chemical property encoding, and domain-specific transformations. Effective feature extraction enhances model performance by reducing dimensionality and focusing on the most informative variables.

4. **Hybrid Modeling**: The system employs a hybrid machine learning approach, combining multiple algorithms to improve prediction accuracy and robustness. For example, ensemble methods may integrate decision trees, support vector machines, and neural networks. Hybrid modeling leverages the strengths of different algorithms to detect subtle patterns indicative of adulteration.

5. **Output Layer**: The final stage translates model predictions into interpretable results. This includes labeling food samples as safe or adulterated, generating confidence scores, and providing visualizations or reports that aid decision-making. The output layer ensures that the system's results are actionable and understandable to both technical users and non-technical stakeholders.

Overall, this pipeline ensures a systematic transformation of raw tabular data into reliable,

interpretable, and actionable insights, making the food adulteration detection system practical for real-world deployment.

3.2 Modules and Their Functionalities

3.2.1. User

The User module acts as the central interface through which individuals interact with the system. It integrates all the essential functionalities required for seamless interaction, including registration, authentication, data visualization, and prediction services. The design focuses on providing a user-friendly experience with proper access control mechanisms to ensure security. Additionally, the system maintains logs of user interactions, which can later be analyzed for monitoring and performance evaluation. This module essentially forms the backbone of end user engagement and ensures that the application serves its intended purpose efficiently.

3.2.2. Registration

The Registration module is responsible for onboarding new users into the system. During the registration process, users are required to provide basic details such as name, email, username, and password. These details are validated and stored securely in the SQLite3 database. To enhance security, sensitive data like passwords can be hashed and encrypted before storage. Once registered, users receive unique credentials that allow them to access system services. This module ensures that only authenticated individuals can participate in system activities and prevents duplicate or invalid registrations..

3.2.3. Login

The Login module authenticates users by cross-checking their credentials against the stored data in the database. It acts as the first line of defense against unauthorized access. If valid credentials are provided, the user is granted access to the system; otherwise, appropriate error messages are displayed. Additional layers of security, such as multi-factor authentication (MFA) or CAPTCHA verification, can also be integrated to prevent brute-force or bot-based attacks. By ensuring only legitimate users gain access, this module protects sensitive functionalities like prediction and visualization.

3.2.4. Visualization

The Visualization module provides an interactive medium for users to view and interpret results. Instead of raw numbers or datasets, information is represented in graphical formats such as line graphs, bar charts, pie charts, or dashboards. For instance, in prediction-based systems, users can track trends over time, compare historical data with new predictions, or analyze anomaly patterns. This module enhances decision-making by making complex data easily interpretable. It also supports real-time updates, so users can immediately visualize the impact of new input data or changes in model outputs.

3.2.5. Prediction

The Prediction module is the core functional component of the system. It utilizes machine learning (ML) or deep learning (DL) algorithms to process user-provided input and generate meaningful outcomes. For example:

- In healthcare, it may predict disease risks.
- In finance, it can forecast trends or anomalies.
- In security, it may classify or detect intrusions.

The prediction workflow typically involves:

1. Preprocessing user input data – cleaning and formatting it for model compatibility.
2. Applying trained models – running the input through pre-trained ML/DL models.
3. Generating output – providing predictions, classifications, or risk scores.
4. Storing results – saving prediction history in the database for tracking and future reference.

3.3 FUNCTIONAL REQUIREMENTS

The Functional requirements for a system describe the functionality or the services that the system is expected to provide. These are the statements of services the system should provide and how the system should react to particular inputs and how the system should behave in particular situation.

- User Registartion: User Register with their Registration details.
- User Login: User Login their account using password
- Live Inputs: Inputs Given By the User requirement.
- Load Model : Trained or Tested Model will be load .
- Predict Output : Output will be predict based on parameters.

3.4 Non-Functional Requirements

Non-functional requirements define the overall quality attributes and system constraints rather than specific system behaviors. These requirements ensure that the system performs efficiently, remains reliable under different conditions, and is capable of supporting future enhancements. They focus on how the system operates and the standards it must meet in real-world usage.

Performance:

The system should deliver high accuracy in predictions while maintaining minimal response time. It must be optimized to process input data and generate results quickly, ensuring that users receive predictions in real time or near real time. Efficient algorithms and optimized model architecture should be used to reduce computational delay and improve overall system performance.

Scalability:

The system should be designed to handle increasing amounts of data and a growing number of users without degradation in performance. It must be flexible enough to expand in terms of database size, computational resources, and model complexity. This ensures that the system can adapt to future requirements, such as additional food categories, larger datasets, or integration with other systems.

Capability (Storage Capacity):

The system should have sufficient storage capacity to handle large volumes of training and operational data. Since machine learning models require extensive datasets for accurate training, the storage infrastructure must support efficient data management, retrieval, and updating. High-capacity storage ensures that historical data, training datasets, and new input data can be maintained without loss or performance issues.

3.5 Feasibility Study

The feasibility of the project is analyzed in this phase and business proposal is put forth with very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential. Three key considerations are involved in the feasibility analysis are:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIALFEASIBILITY

3.5.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased..

3.5.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

3.5.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4. SYSTEM SPECIFICATIONS

4.1 Software Requirements

The system is developed using Windows 7 Ultimate as the operating system. The coding language used for implementation is Python, which supports efficient development and integration of machine learning functionalities. The front-end of the application is designed using HTML, CSS, and JavaScript to create an interactive and user-friendly interface. Python is also used as the back-end language to handle processing, logic implementation, and model integration. The Flask framework is used for designing and deploying the web application, enabling smooth communication between the front-end and back-end components.

- Operating system : Windows 7 Ultimate.
- Coding Language : Python.
- Front-End : Html, css, javascript.
- Back-End : Python.
- Designing : Flask

4.2 Hardware Requirements

The hardware requirements for the system include a Pentium IV or higher processor to ensure basic computational capability. A minimum of 8 GB RAM is required to support smooth execution of the application and machine learning processes. The system requires a hard disk capacity of 512 GB to store datasets, models, and application files. Standard input devices such as a Windows keyboard and a two or three button mouse are needed for user interaction. An SVGA monitor is required for proper display of the application interface and output results.

- ➢ Processor - Pentium –IV
- ➢ RAM - 8 GB (min)
- ➢ Hard Disk - 512 GB
- ➢ Key Board - Standard Windows Keyboard
- ➢ Mouse - Two or Three Button Mouse
- ➢ Monitor - SVGA

5. SOFTWARE DESIGN

5.1 System Architecture

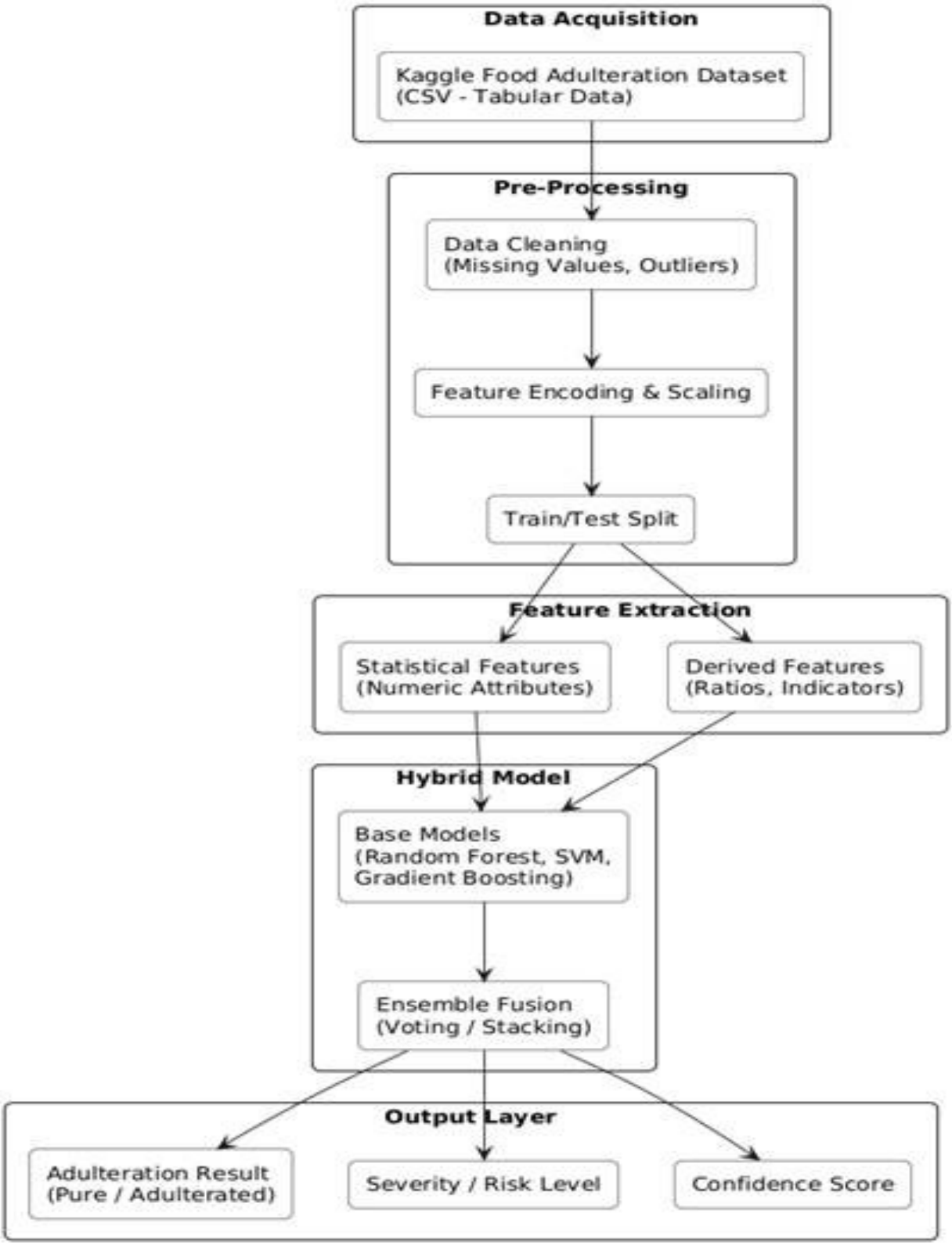


Fig:5.1 System Architecture

The proposed food adulteration detection system follows a structured machine learning pipeline

composed of four major components: Data Acquisition, Pre-Processing, Feature Extraction, Hybrid Modeling, and an Output Layer that delivers the final predictions. Each stage is designed to systematically transform raw tabular data into accurate and interpretable results, ensuring that the system is both reliable and practical for real-world food quality assessment.

The pipeline begins with the Data Acquisition stage, where the Kaggle Food Adulteration Dataset—containing structured tabular records of chemical, physical, and quality-related attributes—is imported in CSV format. This dataset serves as the foundational input for the system, providing measurable indicators needed to identify adulterated and pure food samples. Since the dataset does not include images or sensor streams, the focus remains entirely on numerical and categorical variables that can be processed directly through machine learning algorithms.

The acquired data is then passed to the Pre-Processing stage, which plays a crucial role in preparing the dataset for accurate model training. This phase involves identifying and handling missing values, removing or correcting outliers, and validating data consistency. Following the cleaning process, all features undergo encoding and scaling to ensure that categorical values are machine-readable and numerical values are on a comparable scale. Finally, the pre processed dataset is divided into training and testing subsets, enabling effective model evaluation and preventing overfitting.

Once the dataset is prepared, the system performs Feature Extraction, where statistical attributes and derived features are computed. Statistical features include direct numeric properties from the dataset, while derived attributes consist of engineered indicators such as ratios and domain-specific metrics created to enhance the discriminative power of the model. This combination ensures that both raw and constructed features contribute to a deeper representation of adulteration patterns. The core intelligence of the system lies in the Hybrid Model stage. Here, multiple base machine learning models—such as Random Forest, Support Vector Machines (SVM), and Gradient 14 Boosting—are independently trained on the extracted features. Instead of relying on a single classifier, the system employs ensemble fusion techniques, including voting or stacking, to combine the strengths of all base models.

This hybrid strategy increases accuracy, minimizes model bias, and improves generalization, making the system more robust in diverse or noisy datasets. Finally, the results generated by the hybrid ensemble are sent to the Output Layer, where predictions are organized into three key components: the adulteration classification result (pure or adulterated), the severity or risk level of adulteration, and a confidence score indicating the model's certainty. These outputs provide both technical and practical insights, supporting decision-making for food inspectors, producers, and regulatory bodies.

5.2 Dataflow Diagram

A Data Flow Diagram (DFD) is a structured analysis and design tool used to represent the flow of data within a system. It visually illustrates how data enters the system, how it is processed, stored, and how it moves between different components. The DFD breaks down a system into processes, data stores, data flows, and external entities, showing both the logical and physical aspects of the data movement.

- External Entities: Represent outside sources or destinations of data (e.g., users, external systems, or organizations).
- Processes: Represent activities or transformations that take input data and produce output data (e.g., classification, filtering).
- Data Stores: Represent repositories where data is stored for later retrieval (e.g., message database, corpus storage).
- Data Flows: Represent the path and direction of data as it moves through the system.

The DFD provides a clear and simplified view of the entire system without going into implementation details. It helps stakeholders understand how data is captured, processed, and delivered. Typically, Level 0 (context diagram) gives the overall system view, while Level 1 and Level 2 diagrams break down processes into finer details.

This simple data flow diagram visualizes how raw CSV data (or a new sample) from the user flows through the system: first into preprocessing, then feature extraction, then into the hybrid ensemble model, and finally to the output layer, which returns prediction, severity, and confidence back to the user.

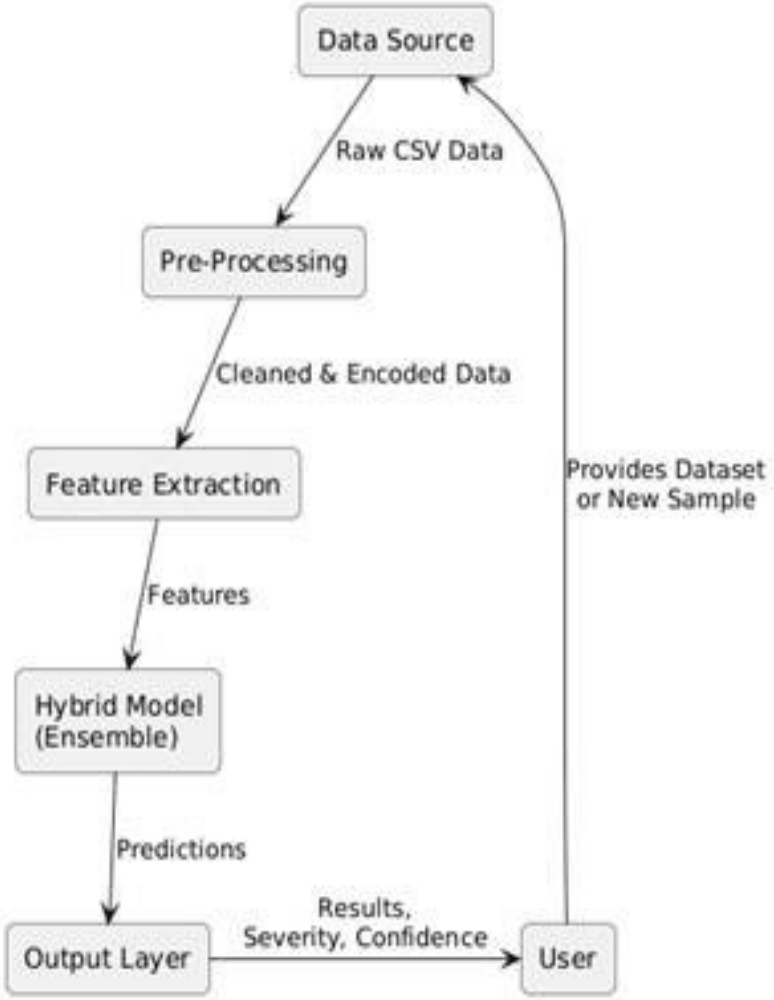


Fig 5.2 Dataflow Diagram

5.3 UML Diagrams

UML (Unified Modeling Language) is a standardized language used for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML helps in representing the design of systems and understanding their components. Created by the Object Management Group (OMG), UML 1.0 was proposed in January 1997. UML is closely associated with object-oriented analysis and design. The two main categories of UML diagrams are Behavioral and Structural diagrams, each serving distinct purposes in the modeling process.

The Behavioral UML diagrams describe the behavior of the system, its actors, and the interaction between the components. On the other hand, Structural UML diagrams depict the static structure of the system, showing its components and relationships. UML has been integrated as a standard by OMG, and its primary goals are to provide a formal basis for understanding modeling languages, offer a ready- to- use expressive language for system developers, and encourage the growth of object-oriented tools.

Goals of UML:

- To provide a standard visual representation of the system design.
- To simplify understanding of system architecture for developers and reviewers.
- To improve communication among team members during development.
- To model both structural and behavioral aspects of the system.
- To support object-oriented design and development practices.
- To document the system clearly for future reference and maintenance.
- To reduce system complexity through diagrammatic representation.
- To assist in planning and designing before actual implementation.
- To enable easy modification and scalability of the system design.
- To ensure a systematic and organized software development process.

Types of UML Diagrams:

1. Sequence Diagram:
2. Use Case Diagram:
3. Activity Diagram:
4. Class Diagram:

5.3.1. Sequence Diagram

The sequence diagram illustrates the interaction flow among the User, UI, ML System, and Model & Data Store during the prediction process of the machine learning-based analysis system. The sequence begins when the user uploads a CSV file or manually enters sample data through the user interface. The UI forwards the submitted sample data to the ML system for processing.

The ML system first preprocesses the received data by performing cleaning, encoding, and scaling operations to prepare it for prediction. It then loads the trained machine learning models from the Model & Data Store. After loading the models, the system extracts relevant features from the input data and sends them to the base models for prediction. The predictions generated by the individual models are then combined using ensemble techniques such as voting or stacking to produce the final output.

Once the prediction process is completed, the ML system returns the result along with severity level and confidence score to the UI. Finally, the UI displays the prediction output to the user in an understandable format. The sequence diagram demonstrates the structured workflow of the machine learning system and shows how data flows through preprocessing, model inference, prediction fusion, and result presentation to deliver accurate and interpretable outcomes.

List of Actions

• User:

The user interacts with the system by uploading a CSV file or entering sample input data manually for prediction or analysis.

• UI:

The user interface receives the input data from the user, validates the format, and forwards the sample data to the ML system. It also displays the final prediction results to the user.

• ML System:

The ML system preprocesses the input data, loads trained machine learning models, extracts features, obtains predictions from base models, and fuses the predictions using ensemble methods such as voting or stacking.

• Model & Data Store:

The Model & Data Store stores trained machine learning models and related preprocessing configurations, which are retrieved by the ML system during prediction.

• Output Module:

The system generates the final prediction result along with severity and confidence metrics, which are sent back to the UI for presentation to the user.

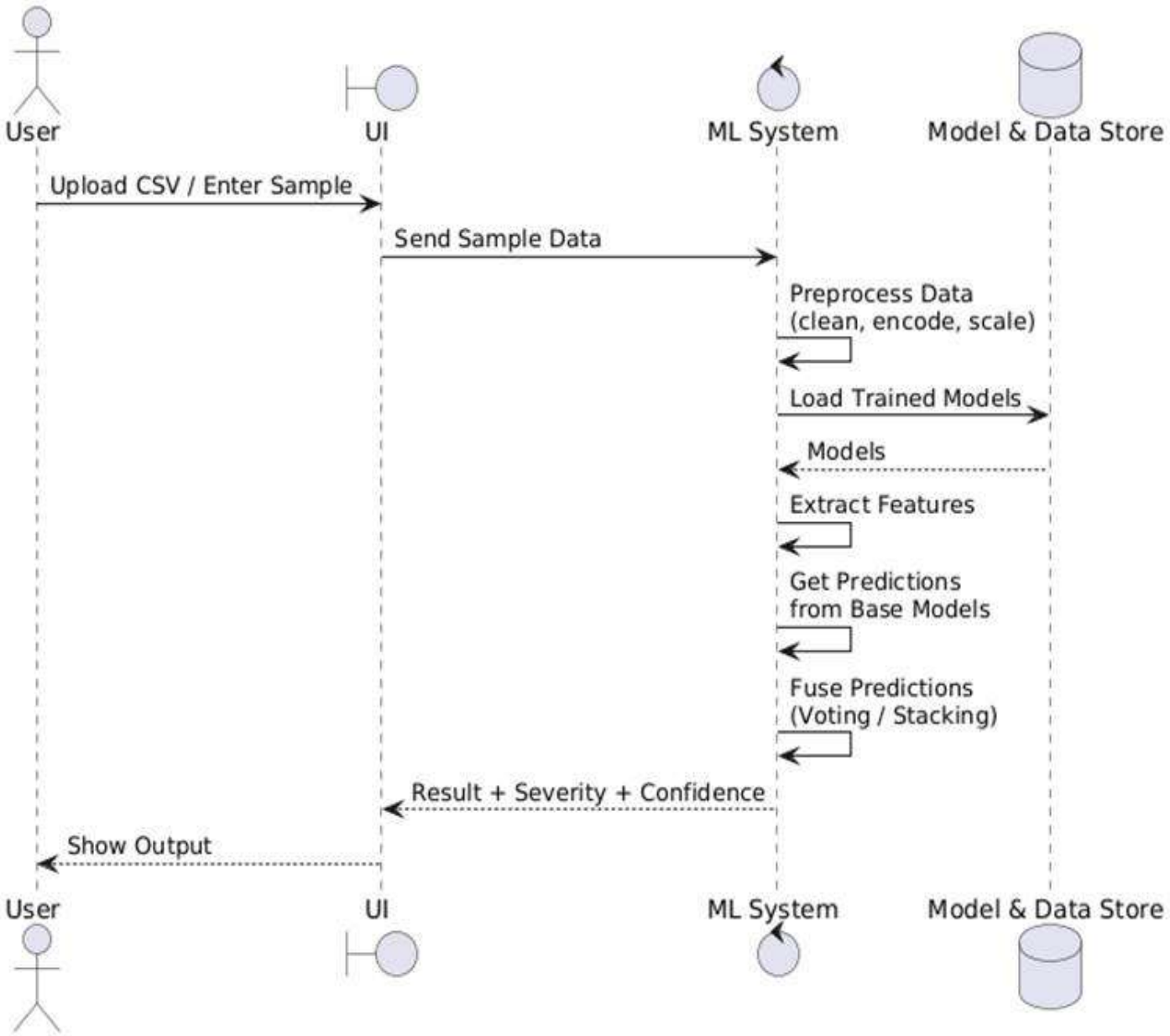


Fig 5.3.1: Sequence Diagram

5.3.2 Use Case Diagram

The use case diagram represents the interactions among the Admin, Quality Inspector, and the Food Adulteration Detection System within the machine learning-based adulteration prediction framework. The Admin initiates core system management operations such as uploading datasets or sample data, training or retraining machine learning models, and monitoring the performance of the prediction system through reports and evaluation metrics. The Quality Inspector interacts with the system to submit food samples for analysis and obtain adulteration prediction results.

The system is responsible for processing uploaded samples, applying the trained machine learning models to predict adulteration, and generating detailed prediction reports. It also produces performance metrics such as accuracy and evaluation reports for monitoring model effectiveness. By supporting both operational prediction tasks and administrative model management functions, the framework ensures efficient adulteration detection, continuous model improvement, and reliable quality assessment within the food inspection process.

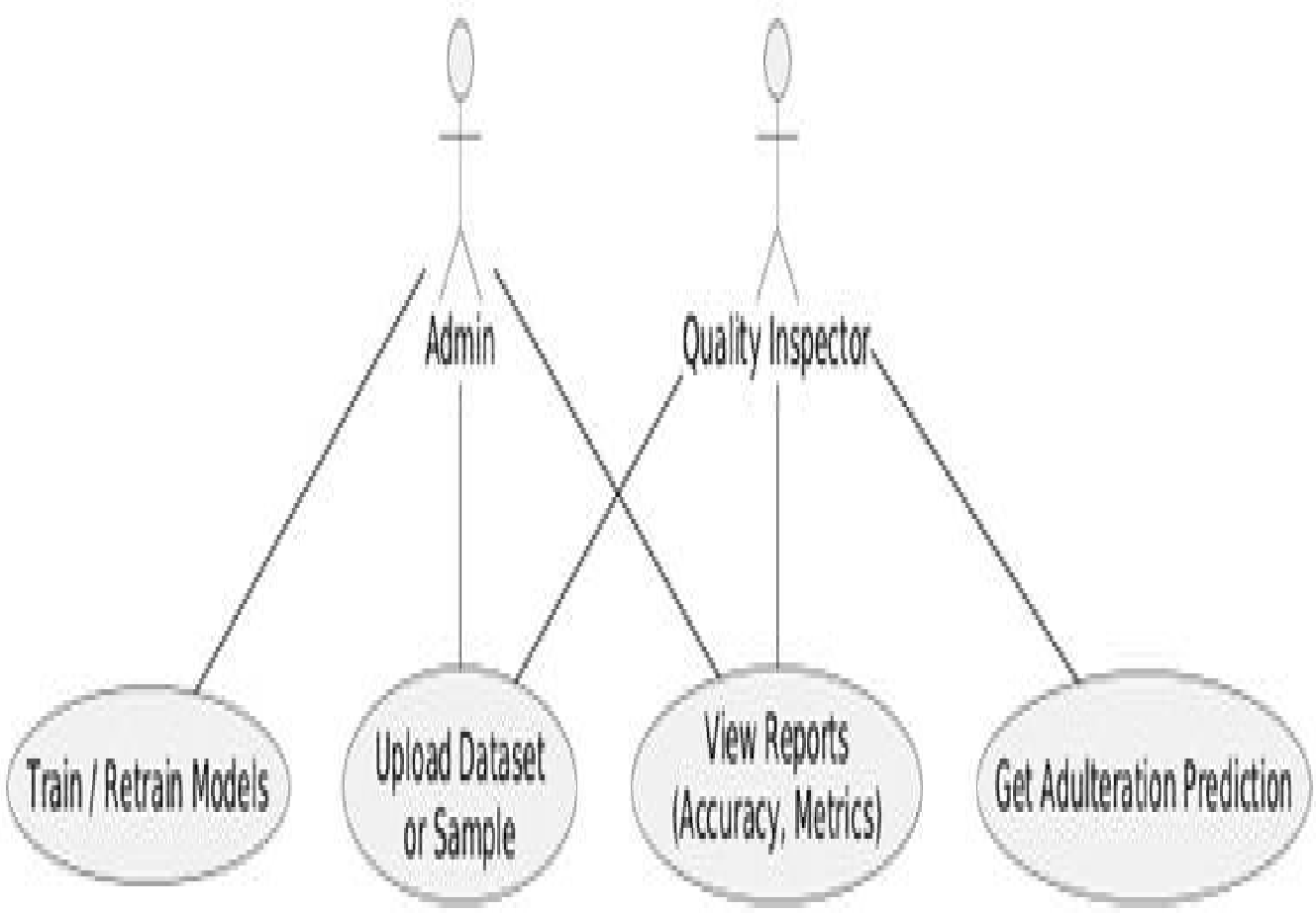


Fig 5.3.2 Use Case Diagram

5.3.3 Activity Diagram

The activity diagram illustrates the workflow of the machine learning-based food adulteration detection system, showing the complete process from dataset preparation to prediction and evaluation. The process begins with loading the Kaggle CSV dataset, followed by cleaning the data to handle missing values and outliers. The cleaned dataset is then preprocessed by encoding categorical features and scaling numerical features to make it suitable for machine learning algorithms.

After preprocessing, the dataset is split into training and testing sets. The system extracts statistical features and creates derived features such as ratios and indicators to improve model performance. Base models including **Random Forest (RF), Support Vector Machine (SVM), and Gradient Boosting (GB)** are trained on the processed data, after which a hybrid ensemble model is built using voting or stacking techniques.

The workflow then branches based on whether a new sample is available. If a new sample is provided, the system preprocesses the sample using the same steps as the training data, extracts features, and obtains a prediction from the hybrid model. The final result is displayed along with adulteration severity and confidence score. If no new sample is present, the system evaluates the trained model using performance metrics such as accuracy and F1-score. This activity diagram demonstrates the end-to-end operational flow of the adulteration detection framework, covering data preparation, model development, prediction, and evaluation.

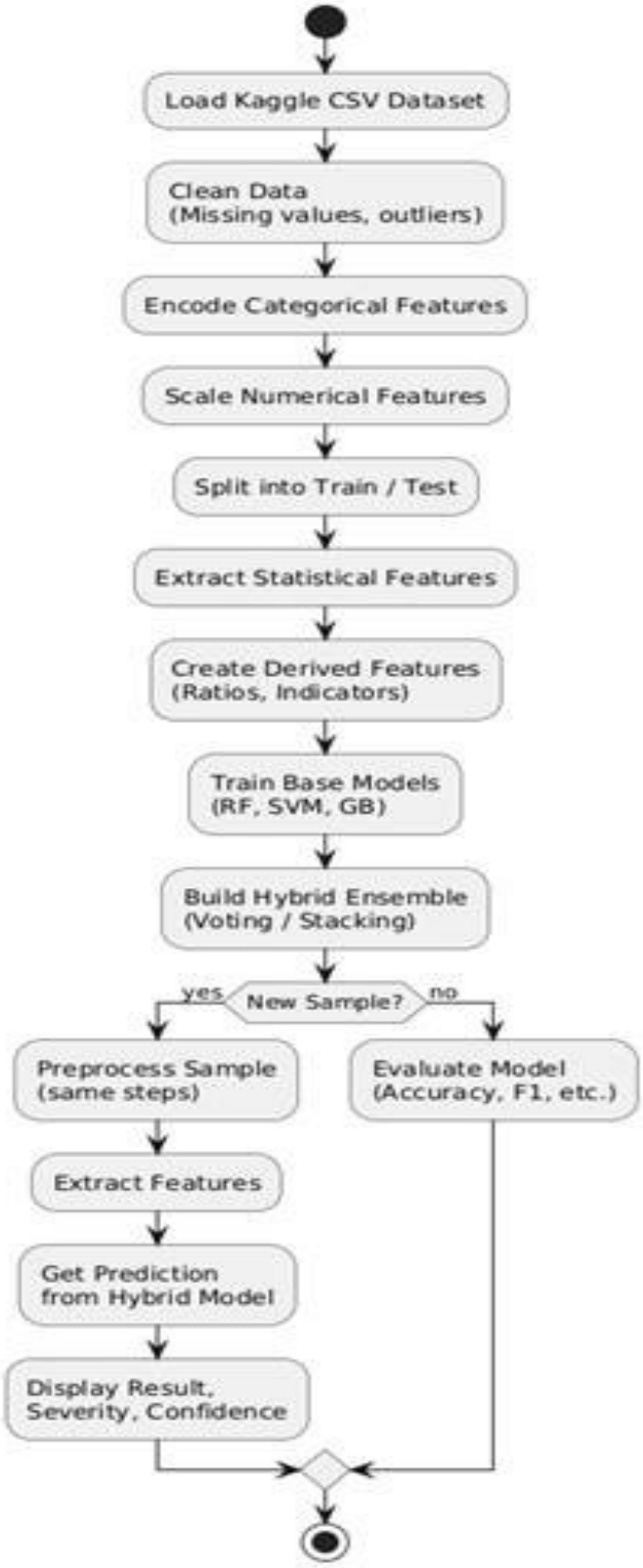


Fig 5.3.3 Activity Diagram

5.3.4. Class Diagram

The class diagram illustrates the structural design of the machine learning-based food adulteration detection system, representing the major software classes involved in data processing, model training, ensemble prediction, and result generation. It shows how different components collaborate to perform adulteration detection in a modular and organized manner.

The workflow begins with the **DatasetManager** class, which is responsible for loading the CSV dataset and splitting it into training and testing subsets. The processed dataset is then passed to the **Preprocessor** class, which handles data cleaning, categorical encoding, and numerical scaling to prepare the data for machine learning operations. After preprocessing, the **FeatureExtractor** class extracts statistical features and generates derived features to improve predictive performance.

The prepared features are then provided to the **BaseModel** class, which represents individual machine learning models capable of training and prediction. Multiple base models are combined within the **EnsembleModel** class, which manages the list of base models and produces final ensemble predictions using methods such as voting or stacking. Finally, the prediction output is encapsulated in the **Result** class, which stores the predicted adulteration label, severity level, and confidence score.

This class diagram demonstrates the object-oriented architecture of the adulteration detection framework and highlights the modular separation of responsibilities among dataset handling, preprocessing, feature engineering, modeling, and output representation components.

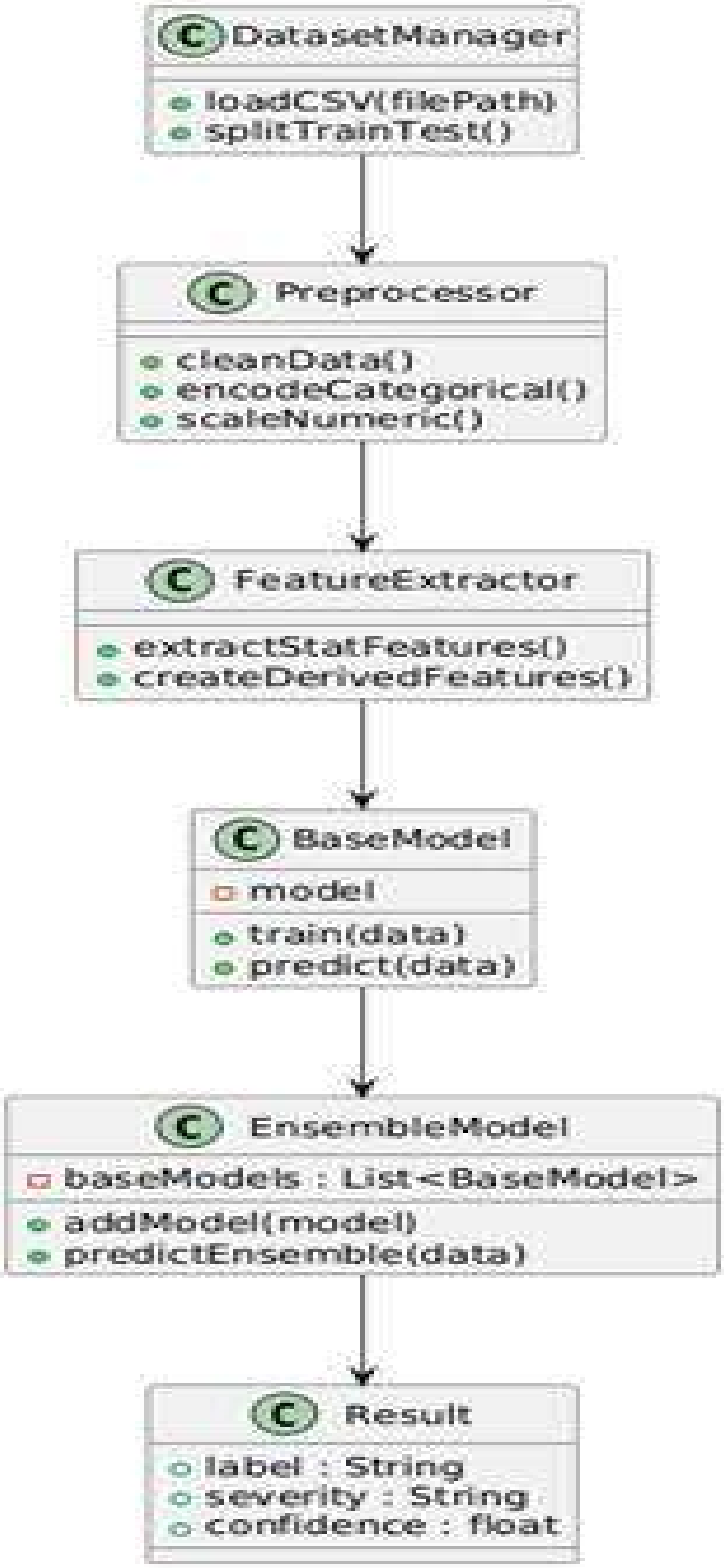


Fig 5.3.4 Class Diagram

6. CODING AND IMPLEMENTATION

6.1 Source Code

Proposedmodel.ipynb:

```python
import pandas as pd

df = pd.read_csv("food_adulteration.csv")
df.head()
```

Python

	product_name	brand	category	adulterant	detection_date	detection_method	severity	health_risk
0	Fish Fillet	BrandH	Seafood	Formalin	7/18/2023	DNA Barcoding	Severe	High
1	Full Cream Milk	FarmFresh19	Dairy	Starch	10/13/2022	DNA Barcoding	Moderate	Medium
2	Soft Drink	FarmFresh02	Beverages	Dilution with water	6/28/2022	Chemical Analysis	Minor	Low
3	Vanaspati Ghee	BrandT	Oils & Fats	Cheaper vegetable oil	3/3/2022	Chemical Analysis	Moderate	Low
4	Sunflower Oil	BrandB	Oils & Fats	Mineral oil	4/22/2023	GC-MS Screening	Minor	Low

```python
df.info()
```

Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30000 entries, 0 to 29999
Data columns (total 9 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   product_name     30000 non-null  object
 1   brand            30000 non-null  object
 2   category         30000 non-null  object
 3   adulterant       30000 non-null  object
 4   detection_date   30000 non-null  object
 5   detection_method 30000 non-null  object
 6   severity         30000 non-null  object
 7   health_risk      30000 non-null  object
 8   action_taken     30000 non-null  object
dtypes: object(9)
memory usage: 2.1+ MB
```

```python
for col in df_encoded.columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_encoded[col])
    label_encoders[col] = le
```

Python

```python
X = df_encoded.drop("severity", axis=1)
y = df_encoded["severity"]
```

Python

```python
X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)
```

Python

```python
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
```

```python
rf_model = RandomForestClassifier(
    n_estimators=200,
    random_state=42,
    n_jobs=-1
)
```

Python

```python
gb_model = GradientBoostingClassifier(
    n_estimators=150,
    learning_rate=0.1,
    random_state=42
)
```

Python

```python
voting_clf = VotingClassifier(
    estimators=[
        ('rf', rf_model),
        ('gb', gb_model)
    ],
    voting='soft'
)
```

Python

```python
X_full = X
y_full = y
```

Python

```python
high_capacity_model = RandomForestClassifier(
    n_estimators=1000,
    max_depth=None,
    min_samples_split=2,
    min_samples_leaf=1,
    random_state=42,
    n_jobs=-1
)
```

Python

```python
high_capacity_model.fit(X_full, y_full)
```

Python

```python
high_capacity_model.fit(X_full, y_full)
```

Python

```python
y_full_pred = high_capacity_model.predict(X_full)
```

Python

```python
accuracy_full = accuracy_score(y_full, y_full_pred)
conf_matrix_full = confusion_matrix(y_full, y_full_pred)
class_report_full = classification_report(y_full, y_full_pred)
```

Python

```python
    print("---------------------------------")
    print(f"Accuracy: {accuracy_full:.4f}\n")
    print("Classification Report:")
    print(class_report_full)
    print("Confusion Matrix:")
    print(conf_matrix_full)
```

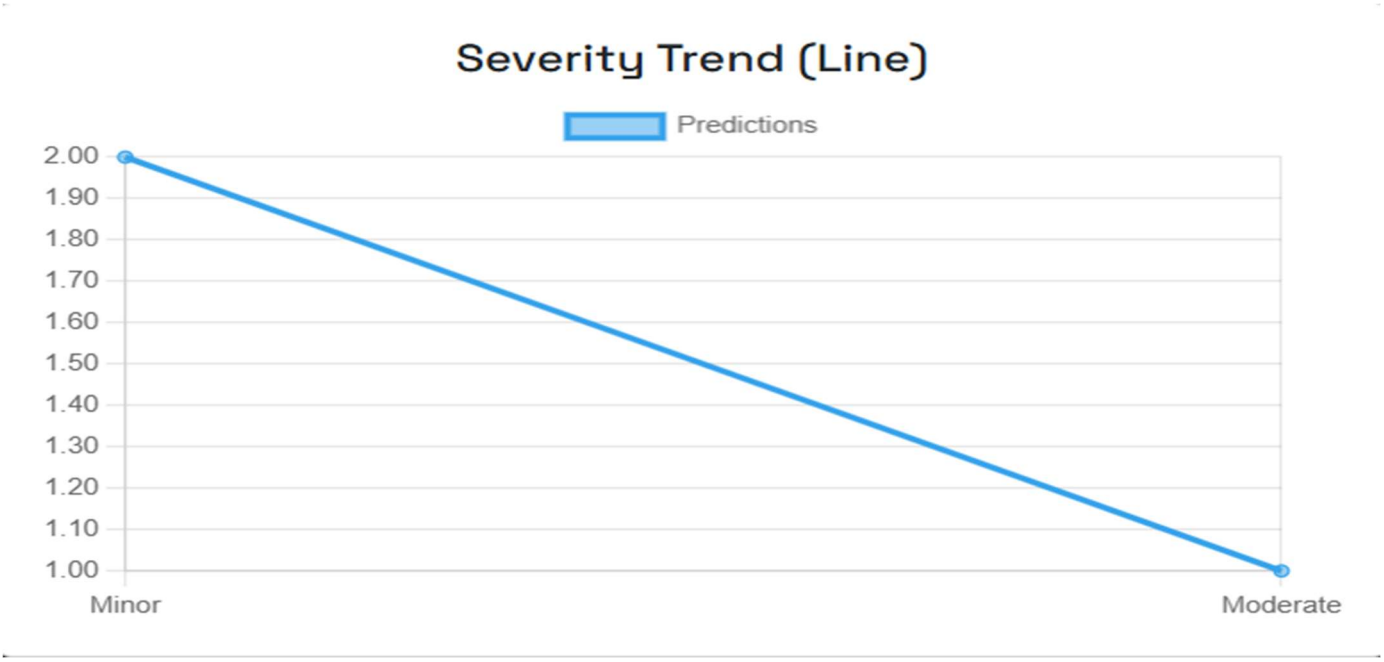
Python

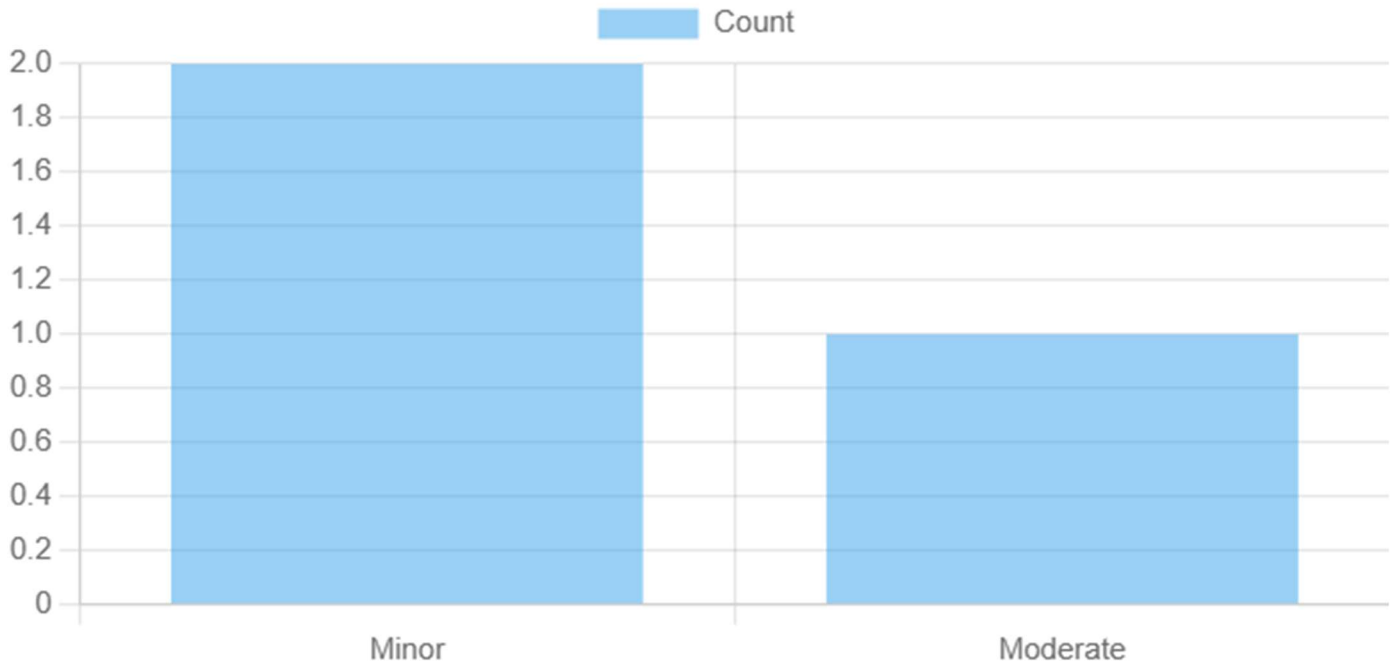
```
---------------------------------
Accuracy: 1.0000

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     10645
           1       1.00      1.00      1.00     10642
           2       1.00      1.00      1.00      8713

    accuracy                           1.00     30000
   macro avg       1.00      1.00      1.00     30000
weighted avg       1.00      1.00      1.00     30000

Confusion Matrix:
[[10645     0     0]
 [    1 10641     0]
 [    0     0  8713]]
```

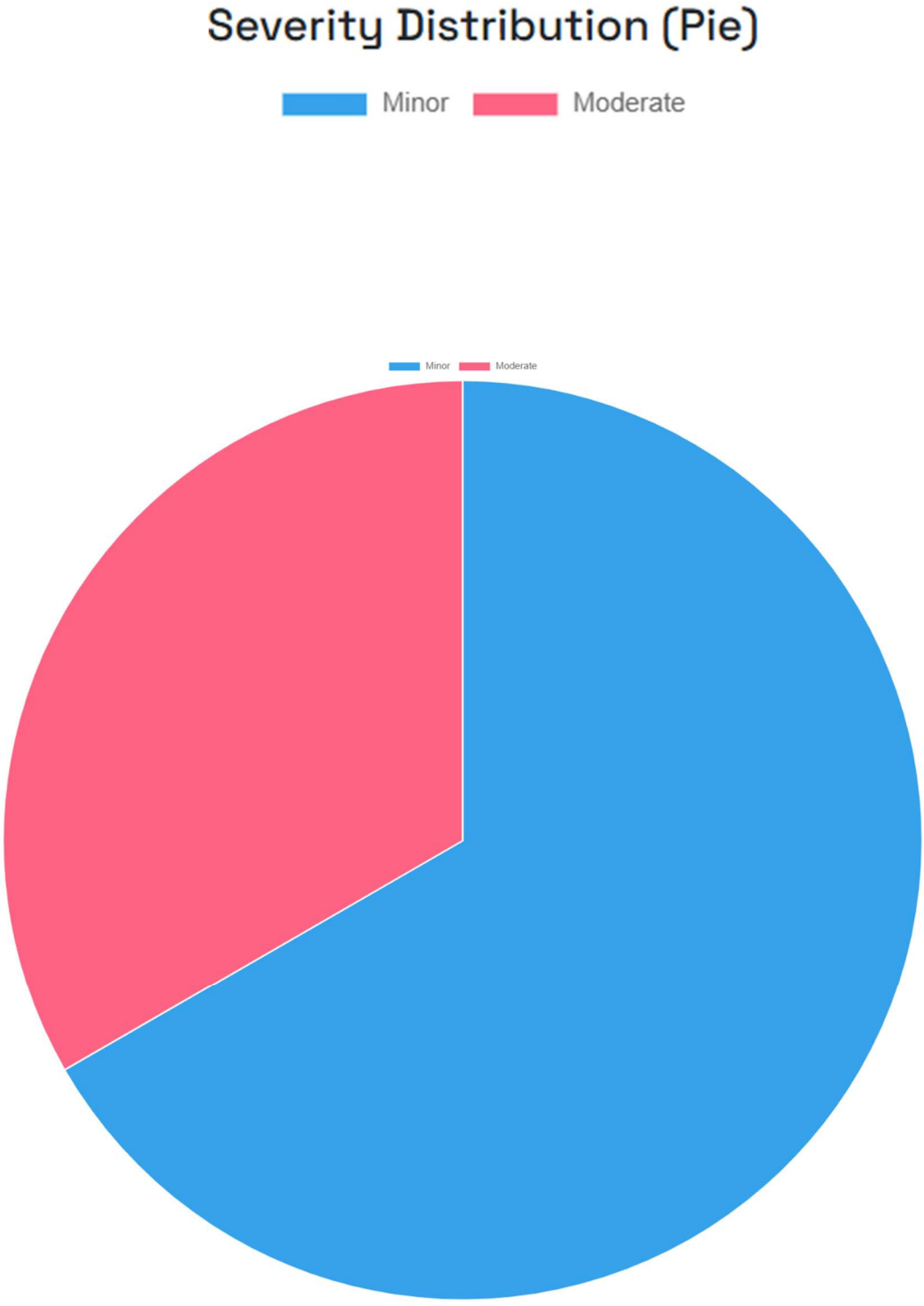
```python
import joblib

joblib.dump(voting_clf, "voting_severity_model.pkl")
joblib.dump(label_encoders, "label_encoders.pkl")
```

Python

```
['label_encoders.pkl']
```


Severity Trend (Line)
Predictions
2.00
1.90
1.80
1.70
1.60
1.50
1.40
1.30
1.20
1.10
1.00
Minor
Moderate


Count
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
Minor
Moderate


Severity Distribution (Pie)
Minor
Moderate
Minor
Moderate

Model DOM :


model
food_adulteration.csv
label_encoders.pkl
model.py
predict.py
proposedmodel.ipynb
voting_severity_model.pkl

Food_adultration.csv

product_name	brand	category	adulterant	detection_date	detection_method	severity	health_risk	action_taken
Fish Fillet	BrandH	Seafood	Formalin	7/18/2023	DNA Barcoding	Severe	High	Product Recall
Full Cream Milk	FarmFresh19	Dairy	Starch	10/13/2022	DNA Barcoding	Moderate	Medium	Warning Issued
Soft Drink	FarmFresh02	Beverages	Dilution with water	6/28/2022	Chemical Analysis	Minor	Low	Warning Issued
Vanaspati Ghee	BrandT	Oils & Fats	Cheaper vegetable oil	03-03-2022	Chemical Analysis	Moderate	Low	Warning Issued
Sunflower Oil	BrandB	Oils & Fats	Mineral oil	4/22/2023	GC-MS Screening	Minor	Low	Warning Issued
Black Pepper	FarmFresh13	Spices & Condiments	Talcum powder	2/24/2022	Chemical Analysis	Minor	High	Fine Imposed
Chicken Nuggets	FarmFresh01	Meat & Poultry	Non-permitted preservatives	8/23/2025	Chemical Analysis	Moderate	Low	Warning Issued
Khoa	FarmFresh02	Dairy	Skimmed milk powder	3/28/2023	DNA Barcoding	Minor	Medium	Warning Issued
Cumin Seeds	FarmFresh17	Spices & Condiments	Talcum powder	7/24/2023	Chemical Analysis	Severe	Medium	Fine Imposed
Curry Powder	BrandR	Spices & Condiments	Metanil yellow	1/14/2022	Sensory Evaluation	Severe	High	Product Recall
Prawns	FarmFresh03	Seafood	Synthetic coloring agents	5/15/2024	Chemical Analysis	Moderate	Low	Warning Issued
Curd	BrandK	Dairy	Chalk powder	11/15/2022	Rapid Test Kit	Minor	High	No Action
Full Cream Milk	BrandD	Dairy	Refined oil	7/29/2022	Rapid Test Kit	Moderate	Medium	Fine Imposed
Fish Fillet	BrandC	Seafood	Added weight stones	7/18/2022	Chemical Analysis	Moderate	Medium	Warning Issued
Beef	BrandZ	Meat & Poultry	Formalin	5/21/2025	Chemical Analysis	Severe	High	Criminal Case Filed
Paneer	FarmFresh01	Dairy	Water	7/29/2024	Rapid Test Kit	Moderate	Low	Fine Imposed
Sunflower Oil	FarmFresh07	Oils & Fats	Recycled oil	06-11-2022	Spectroscopy	Minor	Low	Warning Issued
Chicken Nuggets	FarmFresh11	Meat & Poultry	Formalin	07-11-2025	Microbiological Analy	Moderate	High	Batch Rejected
Chicken Nuggets	FarmFresh13	Meat & Poultry	Formalin	3/28/2025	Chemical Analysis	Severe	High	Criminal Case Filed
Chili Powder	FarmFresh03	Spices & Condiments	Common salt	5/23/2022	Sensory Evaluation	Moderate	Low	Fine Imposed
Tomatoes	BrandX	Fruits & Vegetables	Ethephon ripening agent	04-12-2023	Rapid Test Kit	Severe	High	Product Recall
Whiskey	BrandL	Beverages	Non-permitted synthetic colc	4/22/2023	GC-MS Screening	Severe	High	Product Recall
Toned Milk	FarmFresh09	Dairy	Skimmed milk powder	7/24/2023	Chemical Analysis	Severe	Low	Investigation Launch
Crab Meat	BrandQ	Seafood	Synthetic coloring agents	1/18/2024	DNA Barcoding	Severe	Low	Investigation Launch
Mutton	BrandO	Meat & Poultry	Formalin	12/29/2023	Chemical Analysis	Severe	High	Product Recall
Milk Cake	BrandM	Sweets & Bakery	Starch	10/31/2025	Chemical Analysis	Minor	Low	Fine Imposed
Rice	FarmFresh07	Grains & Pulses	Kesari dal	7/24/2025	Rapid Test Kit	Minor	Medium	Warning Issued
Chili Powder	BrandJ	Spices & Condiments	Talcum powder	5/17/2023	Rapid Test Kit	Minor	Medium	Warning Issued
Skimmed Milk	BrandL	Dairy	Synthetic milk	2/17/2024	Microbiological Analy	Severe	High	Facility Suspended

Model.py

```
import pandas as pd
import numpy as np

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier,
VotingClassifier

df_encoded = df.copy()

label_encoders = {}

for col in df_encoded.columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_encoded[col])
    label_encoders[col] = le

X = df_encoded.drop("severity", axis=1)
y = df_encoded["severity"]

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

rf_model = RandomForestClassifier(
    n_estimators=200,
    random_state=42,
    n_jobs=-1
)

gb_model = GradientBoostingClassifier(
    n_estimators=150,
    learning_rate=0.1,
    random_state=42
)

voting_clf = VotingClassifier(
    estimators=[
        ('rf', rf_model),
        ('gb', gb_model)
    ],
    voting='soft'
)

voting_clf.fit(X_train, y_train)
y_pred = voting_clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
```

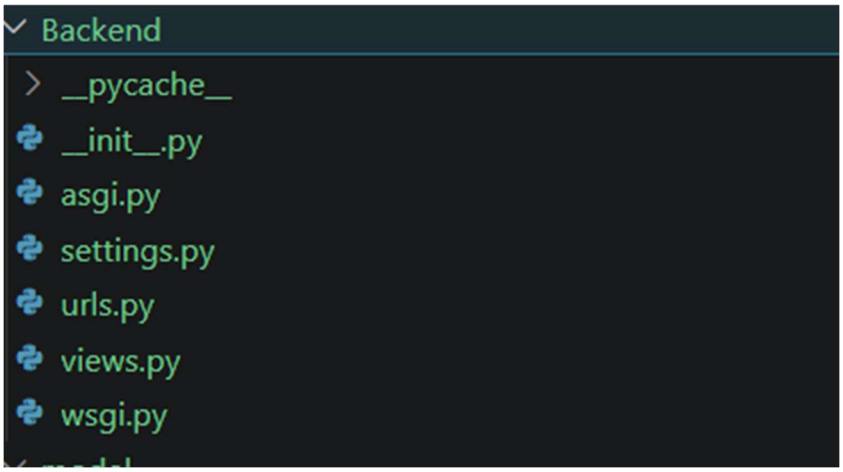
```
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)

print(" Voting Classifier Performance")
print("--------------------------------")
print(f"Accuracy: {accuracy:.4f}\n")

print(" Classification Report:")
print(class_report)

print(" Confusion Matrix:")
print(conf_matrix)
```

Predict.py

```
import joblib
import pandas as pd
import sys

MODEL_PATH = "voting_severity_model.pkl"
ENCODER_PATH = "label_encoders.pkl"

model = joblib.load(MODEL_PATH)
label_encoders = joblib.load(ENCODER_PATH)

def encode_input(df, encoders):
    """
    Encodes input dataframe using trained LabelEncoders.
    Raises error if unseen category is found.
    """
    df_encoded = df.copy()

    for col in df_encoded.columns:
        if col not in encoders:
            raise ValueError(f"Encoder for column '{col}' not found")

        encoder = encoders[col]
        unseen = set(df_encoded[col]) - set(encoder.classes_)
        if unseen:
            raise ValueError(f"Unseen label {unseen} in column '{col}'")

        df_encoded[col] = encoder.transform(df_encoded[col])

    return df_encoded

def predict_severity(input_data: dict):
    """
    Predicts severity from raw input dictionary.
    """
    input_df = pd.DataFrame([input_data])
    encoded_df = encode_input(input_df, label_encoders)
    pred_encoded = model.predict(encoded_df)[0]
```

```
    pred_label = label_encoders["severity"].inverse_transform([pred_encoded])[0]
    probabilities = model.predict_proba(encoded_df)[0]
    prob_dict = dict(
        zip(label_encoders["severity"].classes_, probabilities)
    )

    return pred_label, prob_dict

if __name__ == "__main__":

    sample_input = {
        "product_name": "Full Cream Milk",
        "brand": "FarmFresh19",
        "category": "Dairy",
        "adulterant": "Starch",
        "detection_date": "10/13/2022",
        "detection_method": "DNA Barcoding",
        "health_risk": "Medium",
        "action_taken": "Warning Issued"
    }

    try:
        severity, probs = predict_severity(sample_input)

        print(" Predicted Severity:", severity)
        print("\n Class Probabilities:")
        for cls, prob in probs.items():
            print(f"{cls}: {prob:.2f}")

    except Exception as e:
        print(" Prediction failed:", str(e))
        sys.exit(1)
```

Backend DOM



asgi.py

```
"""
ASGI config for Backend project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/5.2/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Backend.settings')

application = get_asgi_application()
```

Settings.py

"""
Django settings for Backend project.

Generated by 'django-admin startproject' using Django 5.2.6.

For more information on this file, see
https://docs.djangoproject.com/en/5.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.2/ref/settings/
"""

import os

```
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-aic5klhp7^*apum4)j1t_667f5(rd++ja$)*2=a7m_+qjzm0%4'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ["*"]

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'Admins',
    'Users',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'Backend.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]
```

```
WSGI_APPLICATION = 'Backend.wsgi.application'

# Database
# https://docs.djangoproject.com/en/5.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}

# Password validation
# https://docs.djangoproject.com/en/5.2/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/5.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.2/howto/static-files/

STATIC_URL = 'static/'
STATICFILES_DIRS = [os.path.join(BASE_DIR, 'static'),]
STATIC_ROOT = os.path.join(BASE_DIR, 'staticfiles')

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

# Default primary key field type
# https://docs.djangoproject.com/en/5.2/ref/settings/#default-auto-field
```

```
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Urls.py

```
"""
URL configuration for Backend project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.2/topics/http/urls/
Examples:
Function views
    1. Add an import:  from my_app import views
    2. Add a URL to urlpatterns:  path('', views.home, name='home')
Class-based views
    1. Add an import:  from other_app.views import Home
    2. Add a URL to urlpatterns:  path('', Home.as_view(), name='home')
Including another URLconf
    1. Import the include() function: from django.urls import include, path
    2. Add a URL to urlpatterns:  path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

from Backend.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('Users/', include('Users.urls')),
    path('Admins/', include('Admins.urls')),

    path('', index, name='index'),
    path('loginn/', loginn, name='loginn'),
    path('register/', register, name='register'),
    path('home_page/', index, name='home_page'),
    path('user_logout/', user_logout, name='user_logout'),
    path('user_login/', user_login, name='user_login'),
    path('user_registration/', user_registration, name='user_registration')

]
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

Views.py

```
from django.shortcuts import render, redirect
from django.contrib.auth.models import User
from django.contrib.auth import authenticate, login, logout
from django.contrib import messages

def index(request):
    return render(request, "index.html")

def loginn(request):
    return render(request, "login.html")

def register(request):
    return render(request, "register.html")

# Define the login function
def user_login(request):
    if request.method == "POST":
        username = request.POST.get('username')
        password = request.POST.get('password')

        # Authenticate user
        user = authenticate(request, username=username, password=password)

        if user is not None:
            if not user.is_active:
                # User is inactive
                messages.error(request, "Your account is inactive. Please contact the admin.")
                return redirect('loginn')

            # Login the user
            login(request, user)

            if user.is_staff or user.is_superuser:
                # Redirect to admin home if user is staff
                return redirect('adminhome')
            else:
                # Redirect to user home if user is not staff
                return redirect('userhome')
        else:
            # Invalid username or password
            messages.error(request, "Invalid username or password.")
            return redirect('loginn')

    return render(request, 'login.html')

# Define the user registration function
def user_registration(request):
    if request.method == "POST":
        username = request.POST.get('username')
        email = request.POST.get('email')
        password = request.POST.get('password')
        confirm_password = request.POST.get('confirm_password')
        first_name = request.POST.get('first_name')
        last_name = request.POST.get('last_name')
```

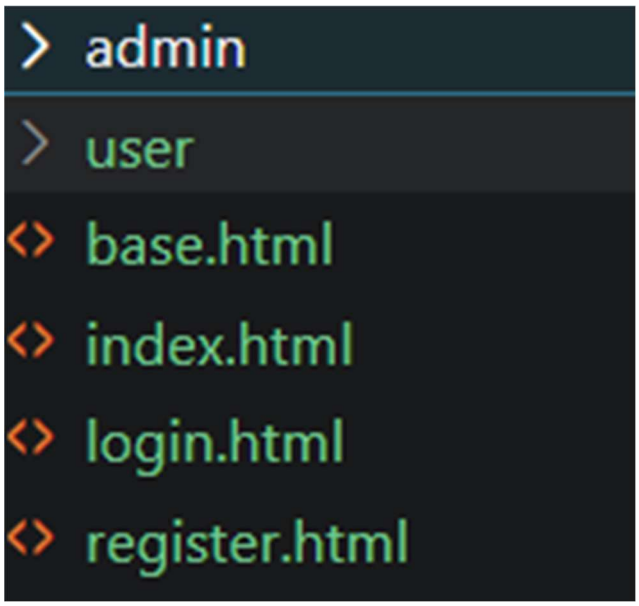
```
        # Check if passwords match
        if password != confirm_password:
            messages.error(request, "Passwords do not match.")
            return redirect('register')

        # Check if username already exists
        if User.objects.filter(username=username).exists():
            messages.error(request, "Username already exists.")
            return redirect('register')

        # Check if email already exists
        if User.objects.filter(email=email).exists():
            messages.error(request, "Email already exists.")
            return redirect('register')

        # Create the user with is_active set to False
        user = User.objects.create_user(
            username=username,
            email=email,
            password=password,
            first_name=first_name,
            last_name=last_name
        )
        user.is_active = False  # Set is_active to False by default
        user.save()

        messages.success(request, "Registration successful! Please wait for admin approval.")
        return redirect('loginn')

    return render(request, 'register.html')

# Define the logout function
def user_logout(request):
    logout(request)
    messages.success(request, "You have been logged out successfully.")
    return redirect('index')
```

wsgi.py

```
"""
WSGI config for Backend project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/5.2/howto/deployment/wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'Backend.settings')

application = get_wsgi_application()
```

Frontend:

Admin Panel DOM



base.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}Food Adulteration Detection{% endblock %}</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Manrope:wght@400;500;600;700;800&family=
Space+Grotesk:wght@500;700&display=swap" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
rel="stylesheet">
  <style>
    :root {
      --bg: #08141f;
      --bg-soft: rgba(9, 22, 33, 0.78);
      --surface-light: rgba(255, 255, 255, 0.92);
      --text: #eff7f5;
      --muted: #b7c8c3;
      --heading: #ffffff;
      --accent: #f29f05;
      --line: rgba(255, 255, 255, 0.12);
      --shadow: 0 24px 60px rgba(0, 0, 0, 0.28);
      --radius-xl: 28px;
```

```
  --radius-lg: 20px;
}

body {
  min-height: 100vh;
  color: var(--text);
  font-family: "Manrope", sans-serif;
  background:
    linear-gradient(135deg, rgba(5, 15, 22, 0.88), rgba(8, 20, 31, 0.74)),
    radial-gradient(circle at top left, rgba(242, 159, 5, 0.25), transparent 34%),
    radial-gradient(circle at bottom right, rgba(28, 114, 102, 0.24), transparent 30%),
    url('{% static "img/bg.webp" %}') center/cover fixed no-repeat;
  position: relative;
}

body::before {
  content: "";
  position: fixed;
  inset: 0;
  background:
    linear-gradient(rgba(255, 255, 255, 0.03) 1px, transparent 1px),
    linear-gradient(90deg, rgba(255, 255, 255, 0.03) 1px, transparent 1px);
  background-size: 80px 80px;
  opacity: 0.24;
  pointer-events: none;
  z-index: -1;
}

h1, h2, h3, h4, h5, h6, .navbar-brand {
  font-family: "Space Grotesk", sans-serif;
  color: var(--heading);
  letter-spacing: -0.02em;
}

p, .text-muted, .lead {
  color: var(--muted) !important;
}

a {
  color: inherit;
  text-decoration: none;
}

.site-shell {
  position: relative;
  z-index: 1;
}

.navbar.app-navbar {
  margin: 18px auto 0;
  width: min(1180px, calc(100% - 24px));
  border: 1px solid var(--line);
  border-radius: 999px;
  background: rgba(6, 18, 26, 0.68);
  backdrop-filter: blur(18px);
```

```
  box-shadow: var(--shadow);
}

.brand-mark {
  width: 42px;
  height: 42px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 14px;
  background: linear-gradient(135deg, var(--accent), #ffd166);
  color: #10202b;
  font-size: 1rem;
  box-shadow: 0 10px 24px rgba(242, 159, 5, 0.3);
}

.navbar .nav-link {
  color: rgba(255, 255, 255, 0.74);
  font-weight: 600;
  padding: 0.75rem 1rem !important;
  transition: color 0.2s ease, transform 0.2s ease;
}

.navbar .nav-link:hover,
.navbar .nav-link:focus,
.navbar .nav-link.active {
  color: #fff;
  transform: translateY(-1px);
}

.btn-accent {
  background: linear-gradient(135deg, var(--accent), #ffbf47);
  border: 0;
  color: #10202b;
  font-weight: 800;
  border-radius: 999px;
  box-shadow: 0 12px 30px rgba(242, 159, 5, 0.35);
}

.btn-accent:hover,
.btn-accent:focus {
  color: #10202b;
  background: linear-gradient(135deg, #ffbf47, var(--accent));
}

.btn-outline-light-soft {
  border: 1px solid rgba(255, 255, 255, 0.2);
  background: rgba(255, 255, 255, 0.04);
  color: #fff;
  border-radius: 999px;
  font-weight: 700;
}

.btn-outline-light-soft:hover {
  background: rgba(255, 255, 255, 0.12);
```

```
  color: #fff;
}

.glass-panel {
  background: var(--bg-soft);
  border: 1px solid var(--line);
  border-radius: var(--radius-xl);
  backdrop-filter: blur(20px);
  box-shadow: var(--shadow);
}

.light-panel {
  background: var(--surface-light);
  border: 1px solid rgba(8, 20, 31, 0.08);
  border-radius: var(--radius-xl);
  box-shadow: 0 20px 50px rgba(4, 15, 22, 0.18);
  color: #10202b;
}

.light-panel h1,
.light-panel h2,
.light-panel h3,
.light-panel h4,
.light-panel h5,
.light-panel h6,
.light-panel label {
  color: #10202b !important;
}

.light-panel p,
.light-panel .text-muted {
  color: #52636c !important;
}

.eyebrow {
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.5rem 0.9rem;
  border-radius: 999px;
  background: rgba(242, 159, 5, 0.14);
  border: 1px solid rgba(242, 159, 5, 0.24);
  color: #ffe0a6;
  font-size: 0.85rem;
  font-weight: 700;
  text-transform: uppercase;
  letter-spacing: 0.08em;
}

.metric-card,
.feature-tile,
.info-tile {
  border-radius: var(--radius-lg);
  border: 1px solid rgba(255, 255, 255, 0.08);
  background: rgba(255, 255, 255, 0.05);
```

```
}

.metric-card,
.info-tile {
  padding: 1.2rem;
}

.feature-tile {
  height: 100%;
  padding: 1.4rem;
}

.metric-value {
  font-size: 1.8rem;
  font-weight: 800;
  color: #fff;
}

.feature-icon {
  width: 54px;
  height: 54px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 16px;
  background: rgba(242, 159, 5, 0.16);
  color: #ffd166;
  font-size: 1.15rem;
}

.form-control,
.form-select {
  border-radius: 14px;
  border: 1px solid rgba(16, 32, 43, 0.12);
  padding: 0.9rem 1rem;
  min-height: 52px;
  box-shadow: none !important;
}

.form-control:focus,
.form-select:focus {
  border-color: rgba(242, 159, 5, 0.6);
  box-shadow: 0 0 0 0.2rem rgba(242, 159, 5, 0.14) !important;
}

.auth-grid {
  min-height: calc(100vh - 190px);
  align-items: center;
}

.page-footer {
  width: min(1180px, calc(100% - 24px));
  margin: 0 auto 18px;
  border: 1px solid var(--line);
  border-radius: 24px;
```

```
    background: rgba(6, 18, 26, 0.72);
    backdrop-filter: blur(18px);
   }

   .alert {
    border-radius: 16px;
    border: 0;
   }

   @media (max-width: 991.98px) {
    .navbar.app-navbar,
    .page-footer {
     width: calc(100% - 20px);
    }

    .auth-grid {
     min-height: auto;
    }
   }
  </style>
  {% block extra_css %}{% endblock %}
</head>
<body class="d-flex flex-column">
  <div class="site-shell d-flex flex-column min-vh-100">
   <nav class="navbar navbar-expand-lg app-navbar">
    <div class="container px-3 px-lg-4">
     <a class="navbar-brand d-flex align-items-center gap-3 fw-bold" href="{% url 'index' %}">
      <span class="brand-mark"><i class="fa-solid fa-shield-heart"></i></span>
      <span>FoodGuard AI</span>
     </a>
     <button class="navbar-toggler border-0 shadow-none" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
      <span class="navbar-toggler-icon"></span>
     </button>
     <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ms-auto align-items-lg-center">
       <li class="nav-item"><a class="nav-link" href="{% url 'index' %}">Overview</a></li>
       <li class="nav-item"><a class="nav-link" href="{% url 'index'
%}#team">Team</a></li>
       <li class="nav-item"><a class="nav-link" href="{% url 'loginn' %}">Login</a></li>
       <li class="nav-item ms-lg-2"><a class="btn btn-accent px-4 py-2" href="{% url 'register'
%}">Create Account</a></li>
      </ul>
     </div>
    </div>
   </nav>

   <main class="flex-grow-1 d-flex align-items-center py-4 py-lg-5">
    <div class="container">
     {% if messages %}
      <div class="row justify-content-center mb-3">
       <div class="col-lg-10">
        {% for message in messages %}
         <div class="alert {% if message.tags == 'error' %}alert-danger{% elif message.tags ==
'success' %}alert-success{% elif message.tags == 'warning' %}alert-warning{% else %}alert-
```

```
info{% endif %} alert-dismissible fade show shadow-sm" role="alert">
                {{ message }}
                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
              </div>
            {% endfor %}
          </div>
        </div>
      {% endif %}
      {% block content %}{% endblock %}
    </div>
  </main>

  <footer class="page-footer text-center py-4 mt-auto">
    <div class="container">
      <p class="mb-1 fw-semibold text-white">FoodGuard AI</p>
      <p class="mb-0">Professional monitoring workflows for food authenticity, risk review, and
safety decisions.</p>
    </div>
  </footer>
 </div>
 <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Index.html

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Home - Food Adulteration Detection{% endblock %}

{% block extra_css %}
<style>
 .landing-hero {
  padding-top: 1.5rem;
 }

 .hero-orbit {
  position: relative;
  min-height: 100%;
  overflow: hidden;
 }

 .hero-orbit::before,
 .hero-orbit::after {
  content: "";
  position: absolute;
  border-radius: 50%;
  filter: blur(6px);
  opacity: 0.7;
 }

 .hero-orbit::before {
```

```
  width: 220px;
  height: 220px;
  top: -70px;
  right: -40px;
  background: radial-gradient(circle, rgba(242, 159, 5, 0.36), transparent 70%);
}

.hero-orbit::after {
  width: 260px;
  height: 260px;
  bottom: -110px;
  left: -50px;
  background: radial-gradient(circle, rgba(55, 180, 160, 0.28), transparent 72%);
}

.hero-grid {
  position: relative;
  z-index: 1;
}

.hero-stat {
  padding: 1.2rem;
  border-radius: 18px;
  background: rgba(255, 255, 255, 0.05);
  border: 1px solid rgba(255, 255, 255, 0.08);
}

.hero-stat strong {
  display: block;
  font-size: 2rem;
  color: #fff;
}

.spotlight-card {
  background: linear-gradient(180deg, rgba(255, 255, 255, 0.09), rgba(255, 255, 255, 0.03));
  border: 1px solid rgba(255, 255, 255, 0.08);
  border-radius: 24px;
  padding: 1.6rem;
}

.workflow-step {
  height: 100%;
  padding: 1.6rem;
  border-radius: 22px;
  background: rgba(255, 255, 255, 0.04);
  border: 1px solid rgba(255, 255, 255, 0.08);
}

.step-number {
  width: 42px;
  height: 42px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 50%;
```

```
  background: rgba(242, 159, 5, 0.18);
  color: #ffd166;
  font-weight: 800;
  margin-bottom: 1rem;
}

.team-card {
  height: 100%;
  padding: 1.75rem;
  border-radius: 24px;
  background: rgba(255, 255, 255, 0.06);
  border: 1px solid rgba(255, 255, 255, 0.1);
  transition: transform 0.2s ease, border-color 0.2s ease;
}

.team-card:hover {
  transform: translateY(-4px);
  border-color: rgba(242, 159, 5, 0.28);
}

.team-avatar {
  width: 68px;
  height: 68px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 22px;
  background: linear-gradient(135deg, rgba(242, 159, 5, 0.92), rgba(255, 209, 102, 0.95));
  color: #10202b;
  font-size: 1.35rem;
  font-weight: 800;
  margin-bottom: 1rem;
}

.team-photo {
  width: 108px;
  height: 128px;
  object-fit: cover;
  object-position: center 24%;
  border-radius: 24px;
  display: block;
  margin-bottom: 1rem;
  border: 3px solid rgba(255, 255, 255, 0.14);
  box-shadow: 0 14px 28px rgba(0, 0, 0, 0.18);
}

.team-photo-siddharth {
  object-position: center 18%;
}

.team-photo-nagalaxmi {
  object-position: center 16%;
  width: 112px;
  height: 134px;
}
```

```
  .team-photo-kirthi {
    object-position: center 14%;
    width: 112px;
    height: 134px;
  }

  .team-role {
    color: #ffd78f;
    font-size: 0.92rem;
    font-weight: 700;
    text-transform: uppercase;
    letter-spacing: 0.08em;
  }

  .cta-strip {
    background: linear-gradient(135deg, rgba(242, 159, 5, 0.16), rgba(255, 255, 255, 0.05));
    border: 1px solid rgba(242, 159, 5, 0.2);
    border-radius: 28px;
    padding: 2rem;
  }
</style>
{% endblock %}

{% block content %}
<section class="landing-hero">
  <div class="row g-4 align-items-center">
    <div class="col-lg-7">
      <div class="glass-panel p-4 p-lg-5">
        <span class="eyebrow mb-3">
          <i class="fa-solid fa-shield-virus"></i>
          Food quality intelligence
        </span>
        <h1 class="display-3 fw-bold mb-3">A sharper landing page for your food adulteration platform.</h1>
        <p class="lead mb-4">
          Detect suspicious products, assess severity faster, and present your system like a serious research-driven product instead of a plain demo page.
        </p>

        <div class="d-flex flex-wrap gap-3 mb-4">
          <a href="{% url 'register' %}" class="btn btn-accent btn-lg px-4">Get Started</a>
          <a href="{% url 'loginn' %}" class="btn btn-outline-light-soft btn-lg px-4">View Dashboard</a>
        </div>

        <div class="row g-3">
          <div class="col-sm-4">
            <div class="hero-stat">
              <strong>AI</strong>
              <span class="small text-white-50 text-uppercase">Severity prediction</span>
            </div>
          </div>
          <div class="col-sm-4">
            <div class="hero-stat">
```

```
            <strong>Safe</strong>
            <span class="small text-white-50 text-uppercase">Risk-focused review</span>
          </div>
        </div>
        <div class="col-sm-4">
          <div class="hero-stat">
            <strong>Fast</strong>
            <span class="small text-white-50 text-uppercase">Action-ready insights</span>
          </div>
        </div>
      </div>
    </div>
  </div>

  <div class="col-lg-5">
    <div class="glass-panel hero-orbit p-4 p-lg-5 h-100">
      <div class="hero-grid">
        <div class="spotlight-card mb-3">
          <p class="text-uppercase fw-bold text-white-50 mb-2">Platform Snapshot</p>
          <h3 class="mb-3">Designed for cleaner, more credible presentation</h3>
          <p class="mb-0">A modern interface for food safety analysis, prediction history, and
model-supported severity classification.</p>
        </div>

        <div class="row g-3">
          <div class="col-12">
            <div class="feature-tile">
              <div class="feature-icon mb-3"><i class="fa-solid fa-magnifying-glass-
chart"></i></div>
              <h5>Smart record review</h5>
              <p class="mb-0">Capture product, adulterant, and detection details in a clean,
structured interface.</p>
            </div>
          </div>
          <div class="col-md-6">
            <div class="feature-tile">
              <div class="feature-icon mb-3"><i class="fa-solid fa-chart-line"></i></div>
              <h5>Analytics ready</h5>
              <p class="mb-0">Present your data in a way that feels fit for demos and
evaluation.</p>
            </div>
          </div>
          <div class="col-md-6">
            <div class="feature-tile">
              <div class="feature-icon mb-3"><i class="fa-solid fa-user-shield"></i></div>
              <h5>Trust first</h5>
              <p class="mb-0">Build confidence with a more professional visual experience.</p>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
```

```
<section class="py-4 py-lg-5">
  <div class="row g-4">
    <div class="col-lg-4">
      <div class="workflow-step">
        <div class="step-number">1</div>
        <h4>Capture the case</h4>
        <p class="mb-0">Add product details, suspected adulterants, detection methods, and food
safety context in one flow.</p>
      </div>
    </div>
    <div class="col-lg-4">
      <div class="workflow-step">
        <div class="step-number">2</div>
        <h4>Predict severity</h4>
        <p class="mb-0">Use the model to estimate severity and make prioritization easier for the
next response step.</p>
      </div>
    </div>
    <div class="col-lg-4">
      <div class="workflow-step">
        <div class="step-number">3</div>
        <h4>Review outcomes</h4>
        <p class="mb-0">Track prediction history and analytics from a cleaner, more polished
dashboard experience.</p>
      </div>
    </div>
  </div>
</section>

<section id="team" class="py-3 py-lg-4">
  <div class="glass-panel p-4 p-lg-5">
    <div class="d-flex flex-column flex-lg-row justify-content-between align-items-lg-end gap-3
mb-4">
      <div>
        <span class="eyebrow mb-3">
          <i class="fa-solid fa-people-group"></i>
          Our team
        </span>
        <h2 class="mb-2">The people behind the platform</h2>
        <p class="mb-0">Project team and faculty guidance for the Food Adulteration Detection
platform.</p>
      </div>
      <p class="mb-0 text-white-50">Built with a mix of food safety awareness, machine learning,
and product design.</p>
    </div>

    <div class="row g-4">
      <div class="col-md-6 col-xl-3">
        <div class="team-card">
          <img src="{% static 'img/team/mujeeb.jpeg' %}" alt="Shaikh Mujeeb" class="team-
photo">
          <div class="team-role mb-2">Project Lead</div>
          <h5 class="mb-2">Shaikh Mujeeb</h5>
          <p class="mb-1"><strong>Roll No:</strong> 228R1A66C1</p>
```

```
        <p class="mb-0">Contributed to the project direction, system integration, and overall
implementation flow.</p>
       </div>
      </div>
      <div class="col-md-6 col-xl-3">
       <div class="team-card">
        <img src="{% static 'img/team/siddharth.jpeg' %}" alt="Siddhrath Reddy" class="team-
photo team-photo-siddharth">
        <div class="team-role mb-2">Team Member</div>
        <h5 class="mb-2">Siddhrath Reddy</h5>
        <p class="mb-1"><strong>Roll No:</strong> 228R1A6674</p>
        <p class="mb-0">Supported development, research work, and project coordination across
the application workflow.</p>
       </div>
      </div>
      <div class="col-md-6 col-xl-3">
       <div class="team-card">
        <img src="{% static 'img/team/kirthi.jpeg' %}" alt="Kirthi" class="team-photo team-
photo-kirthi">
        <div class="team-role mb-2">Team Member</div>
        <h5 class="mb-2">Keerthi</h5>
        <p class="mb-1"><strong>Roll No:</strong> 228R1A6680</p>
        <p class="mb-0">Worked on project support tasks, documentation, and helping shape the
user-facing experience.</p>
       </div>
      </div>
      <div class="col-md-6 col-xl-3">
       <div class="team-card">
        <img src="{% static 'img/team/nagalaxmi.jpeg' %}" alt="Naga Laxmi" class="team-photo
team-photo-nagalaxmi">
        <div class="team-role mb-2">Team Member</div>
        <h5 class="mb-2">J NagaLaxmi</h5>
        <p class="mb-1"><strong>Roll No:</strong> 228R1A6692</p>
        <p class="mb-0">Contributed to project execution, content preparation, and refinement of
the final presentation.</p>
       </div>
      </div>
      <div class="col-md-6 col-xl-3">
       <div class="team-card">
        <div class="team-avatar">S</div>
        <div class="team-role mb-2">Guide and Mentor</div>
        <h5 class="mb-2">Mrs. Swaroopa Madam</h5>
        <p class="mb-0">Provided academic guidance, project mentorship, and valuable direction
throughout the development process.</p>
       </div>
      </div>
    </div>
   </div>
</section>

<section class="py-4">
  <div class="cta-strip">
    <div class="row align-items-center g-3">
      <div class="col-lg-8">
        <h3 class="mb-2">Ready to Test You food on our food safety platform?</h3>
```

```
      <p class="mb-0">Register now and Get the most out of it for free</p>
     </div>
     <div class="col-lg-4 text-lg-end">
      <a href="{% url 'register' %}" class="btn btn-accent btn-lg px-4">Create Account</a>
     </div>
    </div>
  </div>
</section>
{% endblock %}
```

Login.html

```
{% extends 'base.html' %}
{% block title %}Login - Food Adulteration Detection{% endblock %}

{% block content %}
<div class="col-md-6 col-lg-5 mx-auto bg-white p-5 rounded shadow-lg">
  <h2 class="fw-bold mb-4 text-center">Login</h2>

  {% if messages %}
  <div class="container mb-4">
    {% for message in messages %}
     <div class="alert
              {% if message.tags == 'error' %}alert-danger
              {% elif message.tags == 'success' %}alert-success
              {% elif message.tags == 'warning' %}alert-warning
              {% else %}alert-info{% endif %}
              alert-dismissible fade show shadow-sm" role="alert">
       {{ message }}
       <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
     </div>
    {% endfor %}
  </div>
{% endif %}

  <form action="{% url 'user_login' %}" method="post">
    {% csrf_token %}
    <div class="mb-3">
     <input type="text" name="username" class="form-control" placeholder="Your Username"
required>
    </div>
    <div class="mb-3">
     <input type="password" class="form-control" name="password" placeholder="Your
Password" required>
    </div>
    <div class="text-center">
     <button type="submit" class="btn btn-warning text-white px-5 py-2 rounded-pill shadow">
       <i class="fas fa-sign-in-alt"></i> Login
     </button>
    </div>
  </form>
</div>
{% endblock %}
```

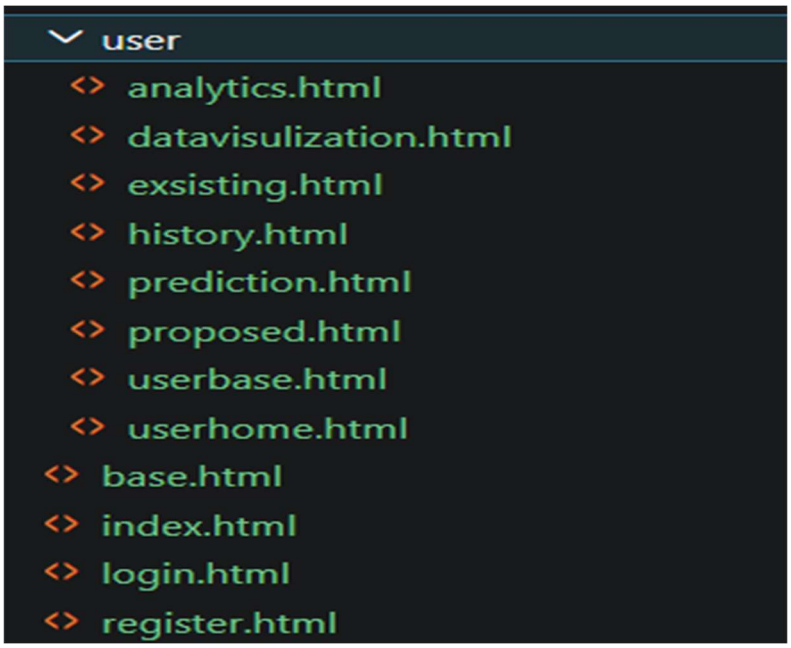
Register.html

```
{% extends 'base.html' %}
{% block title %}Register - Food Adulteration Detection{% endblock %}

{% block content %}
<div class="col-md-8 col-lg-6 mx-auto bg-white p-5 rounded shadow-lg">
  <h2 class="fw-bold mb-4 text-center">Create an Account</h2>
  {% if messages %}
  <div class="container mb-4">
    {% for message in messages %}
     <div class="alert
               {% if message.tags == 'error' %}alert-danger
               {% elif message.tags == 'success' %}alert-success
               {% elif message.tags == 'warning' %}alert-warning
               {% else %}alert-info{% endif %}
               alert-dismissible fade show shadow-sm" role="alert">
       {{ message }}
       <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
     </div>
    {% endfor %}
  </div>
{% endif %}

  <form action="{% url 'user_registration' %}" method="post">
    {% csrf_token %}
    <div class="row g-3">
      <div class="col-md-6">
        <input type="text" name="first_name" class="form-control" placeholder="First Name"
required>
      </div>
      <div class="col-md-6">
        <input type="text" name="last_name" class="form-control" placeholder="Last Name"
required>
      </div>
      <div class="col-md-6">
        <input type="text" name="username" class="form-control" placeholder="Username"
required>
      </div>
      <div class="col-md-6">
        <input type="email" name="email" class="form-control" placeholder="Your Email"
required>
      </div>
      <div class="col-md-6">
        <input type="password" name="password" class="form-control" placeholder="Password"
required>
      </div>
      <div class="col-md-6">
        <input type="password" name="confirm_password" class="form-control"
placeholder="Confirm Password" required>
```

```
      </div>
    </div>

    <div class="text-center mt-4">
      <button type="submit" class="btn btn-warning text-white px-5 py-2 rounded-pill shadow">
        <i class="fas fa-user-plus"></i> Register
      </button>
    </div>
  </form>
</div>
{% endblock %}
```

User

User DOM



Analytics.html

```
{% extends 'user/userbase.html' %}
{% block title %}Analytics{% endblock %}

{% block content %}
<div class="container mt-4">
  <h3 class="mb-4 text-center">Prediction Analytics</h3>

  {% if labels|length == 0 %}
    <div class="alert alert-warning text-center">
      No prediction data available.
    </div>
  {% else %}
    <div class="row g-4">

      <!-- Line Chart -->
      <div class="col-md-6">
        <div class="card shadow-sm">
          <div class="card-body">
            <h5 class="card-title text-center">Severity Trend (Line)</h5>
            <canvas id="lineChart"></canvas>
          </div>
        </div>
      </div>

      <!-- Bar Chart -->
      <div class="col-md-6">
        <div class="card shadow-sm">
          <div class="card-body">
            <h5 class="card-title text-center">Severity Count (Bar)</h5>
            <canvas id="barChart"></canvas>
          </div>
        </div>
      </div>

      <!-- Pie Chart -->
      <div class="col-md-12">
        <div class="card shadow-sm">
          <div class="card-body">
            <h5 class="card-title text-center">Severity Distribution (Pie)</h5>
            <canvas id="pieChart"></canvas>
          </div>
        </div>
      </div>

    </div>
```

```
    {% endif %}
</div>

<!-- Chart.js CDN -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<script>
    const labels = {{ labels|safe }};
    const data = {{ data|safe }};

    // Line Chart
    new Chart(document.getElementById('lineChart'), {
        type: 'line',
        data: {
            labels: labels,
            datasets: [{
                label: 'Predictions',
                data: data,
                fill: false,
                tension: 0.4
            }]
        }
    });

    // Bar Chart
    new Chart(document.getElementById('barChart'), {
        type: 'bar',
        data: {
            labels: labels,
            datasets: [{
                label: 'Count',
                data: data
            }]
        }
    });

    // Pie Chart
    new Chart(document.getElementById('pieChart'), {
        type: 'pie',
        data: {
            labels: labels,
            datasets: [{
                data: data
            }]
        }
    });
</script>
```

```
{% endblock %}
```

Datavisualization.html

```
{% extends 'user/userbase.html' %}
{% block title %}User Profile - Food Adulteration Detection{% endblock %}

{% block content %}

{% endblock %}
```

Histroy.html

```
{% extends 'user/userbase.html' %}
{% block title %}Prediction History{% endblock %}

{% block content %}
<div class="container mt-4">
  <h4 class="mb-3">Your Prediction History</h4>

  <table class="table table-bordered table-striped">
    <thead class="table-dark">
      <tr>
        <th>Date</th>
        <th>Product</th>
        <th>Adulterant</th>
        <th>Severity</th>
      </tr>
    </thead>
    <tbody>
      {% for r in records %}
      <tr>
        <td>{{ r.created_at|date:"d M Y H:i" }}</td>
        <td>{{ r.product_name }}</td>
        <td>{{ r.adulterant }}</td>
        <td>
          <span class="badge bg-warning text-dark">
            {{ r.predicted_severity }}
          </span>
        </td>
      </tr>
      {% empty %}
      <tr>
        <td colspan="4" class="text-center">No predictions yet</td>
      </tr>
      {% endfor %}
    </tbody>
```

```
    </table>
</div>
{% endblock %}
```

Prediction.html

```
{% extends 'user/userbase.html' %}
{% block title %}Prediction{% endblock %}

{% block extra_css %}
<style>
  .prediction-shell {
    max-width: 1100px;
    margin: 0 auto;
  }

  .prediction-card {
    background: rgba(9, 22, 33, 0.78);
    border: 1px solid rgba(255, 255, 255, 0.12);
    border-radius: 28px;
    overflow: hidden;
    box-shadow: 0 24px 60px rgba(0, 0, 0, 0.28);
    backdrop-filter: blur(18px);
  }

  .prediction-header {
    padding: 2rem 2rem 1.25rem;
    border-bottom: 1px solid rgba(255, 255, 255, 0.08);
    background: linear-gradient(135deg, rgba(242, 159, 5, 0.14), rgba(255, 255, 255, 0.04));
  }

  .prediction-header h2 {
    color: #ffffff;
    font-weight: 700;
    margin-bottom: 0.5rem;
  }

  .prediction-header p {
    color: #bfd0ca;
    margin-bottom: 0;
  }

  .prediction-body {
    padding: 2rem;
  }

  .form-label {
    color: #ffffff;
```

```
  font-weight: 600;
  margin-bottom: 0.65rem;
}

.form-control {
  min-height: 54px;
  border-radius: 16px;
  border: 1px solid rgba(255, 255, 255, 0.12);
  background: rgba(255, 255, 255, 0.08);
  color: #ffffff;
  box-shadow: none;
}

.form-control::placeholder {
  color: #a9bab5;
}

.form-control:focus {
  background: rgba(255, 255, 255, 0.1);
  color: #ffffff;
  border-color: rgba(242, 159, 5, 0.55);
  box-shadow: 0 0 0 0.2rem rgba(242, 159, 5, 0.12);
}

.prediction-btn {
  min-width: 220px;
  padding: 0.9rem 1.5rem;
  border: 0;
  border-radius: 999px;
  background: linear-gradient(135deg, #f29f05, #ffd166);
  color: #10202b;
  font-weight: 800;
  box-shadow: 0 14px 30px rgba(242, 159, 5, 0.28);
}

.prediction-btn:hover,
.prediction-btn:focus {
  color: #10202b;
}

.result-box,
.error-box {
  margin-top: 1.75rem;
  border-radius: 20px;
  border: 1px solid transparent;
  padding: 1.25rem 1.3rem;
}
```

```
  .result-box {
    background: rgba(56, 176, 110, 0.14);
    border-color: rgba(56, 176, 110, 0.3);
    color: #ebfff2;
  }

  .error-box {
    background: rgba(220, 53, 69, 0.14);
    border-color: rgba(220, 53, 69, 0.3);
    color: #ffe8ec;
  }

  .probability-list {
    margin: 1rem 0 0;
    padding: 0;
    list-style: none;
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(180px, 1fr));
    gap: 0.85rem;
  }

  .probability-list li {
    padding: 0.9rem 1rem;
    border-radius: 16px;
    background: rgba(255, 255, 255, 0.08);
    border: 1px solid rgba(255, 255, 255, 0.1);
  }

  @media (max-width: 767.98px) {
    .prediction-header,
    .prediction-body {
      padding: 1.35rem;
    }

    .prediction-btn {
      width: 100%;
    }
  }
</style>
{% endblock %}

{% block content %}
<div class="prediction-shell">
  <div class="prediction-card">
    <div class="prediction-header">
      <h2>Food Adulteration Severity Prediction</h2>
```

```
    <p>Enter the required product details to generate the severity prediction.</p>
   </div>

   <div class="prediction-body">
    <form method="post" class="row g-4">
     {% csrf_token %}

     <div class="col-md-6">
      <label class="form-label">Product Name</label>
      <input type="text" name="product_name" class="form-control" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Brand</label>
      <input type="text" name="brand" class="form-control" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Category</label>
      <input type="text" name="category" class="form-control" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Adulterant</label>
      <input type="text" name="adulterant" class="form-control" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Detection Date</label>
      <input type="text" name="detection_date" class="form-control"
placeholder="MM/DD/YYYY" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Detection Method</label>
      <input type="text" name="detection_method" class="form-control" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Health Risk</label>
      <input type="text" name="health_risk" class="form-control" required>
     </div>

     <div class="col-md-6">
      <label class="form-label">Action Taken</label>
      <input type="text" name="action_taken" class="form-control" required>
     </div>
```

```
      <div class="col-12 text-center text-md-start">
        <button type="submit" class="prediction-btn">Predict Severity</button>
      </div>
    </form>

    {% if severity %}
    <div class="result-box">
      <h5 class="mb-2">Predicted Severity: <strong>{{ severity }}</strong></h5>
      <ul class="probability-list">
        {% for k, v in probabilities.items %}
          <li><strong>{{ k }}</strong><br>{{ v|floatformat:2 }}</li>
        {% endfor %}
      </ul>
    </div>
    {% endif %}

    {% if error %}
    <div class="error-box">
      {{ error }}
    </div>
    {% endif %}
  </div>
 </div>
</div>
{% endblock %}
```

Proposed.html

```
{% extends 'user/userbase.html' %}
{% block title %}User Profile - Food Adulteration Detection{% endblock %}

{% load static %}

{% block content %}
  <iframe src="{% static 'notebook/proposedmodel.html' %}"
    style="width:100%; height:80vh; border:none;">
  </iframe>

{% endblock %}
```

Userbase.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}User Dashboard{% endblock %}</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Manrope:wght@400;500;600;700;800&family=
Space+Grotesk:wght@500;700&display=swap" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
rel="stylesheet">
  <style>
    :root {
      --user-bg: #091620;
      --user-surface: rgba(9, 22, 33, 0.78);
      --user-surface-soft: rgba(255, 255, 255, 0.06);
      --user-line: rgba(255, 255, 255, 0.12);
      --user-text: #edf6f3;
      --user-muted: #b8cbc4;
      --user-accent: #f29f05;
      --user-shadow: 0 24px 60px rgba(0, 0, 0, 0.28);
    }

    body {
      min-height: 100vh;
      font-family: "Manrope", sans-serif;
      color: var(--user-text);
      background:
        linear-gradient(135deg, rgba(5, 15, 22, 0.9), rgba(8, 20, 31, 0.78)),
        radial-gradient(circle at top left, rgba(242, 159, 5, 0.18), transparent 30%),
        radial-gradient(circle at bottom right, rgba(32, 140, 126, 0.18), transparent 26%),
        url('{% static "img/bg.webp" %}') center/cover fixed no-repeat;
    }

    h1, h2, h3, h4, h5, h6, .navbar-brand {
      font-family: "Space Grotesk", sans-serif;
    }

    .user-navbar-shell {
      width: min(1220px, calc(100% - 24px));
      margin: 18px auto 0;
      border: 1px solid var(--user-line);
      border-radius: 24px;
      background: rgba(6, 18, 26, 0.72);
      backdrop-filter: blur(18px);
      box-shadow: var(--user-shadow);
    }
```

```
.brand-chip {
  width: 44px;
  height: 44px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 14px;
  background: linear-gradient(135deg, var(--user-accent), #ffd166);
  color: #10202b;
  box-shadow: 0 10px 24px rgba(242, 159, 5, 0.28);
}

.navbar .nav-link {
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.7rem 1rem !important;
  margin: 0 0.15rem;
  border-radius: 999px;
  color: rgba(255, 255, 255, 0.78);
  font-weight: 700;
  transition: background 0.2s ease, color 0.2s ease, transform 0.2s ease;
}

.navbar .nav-link:hover,
.navbar .nav-link:focus {
  color: #fff;
  background: rgba(255, 255, 255, 0.08);
  transform: translateY(-1px);
}

.navbar .nav-link.active {
  color: #10202b;
  background: linear-gradient(135deg, var(--user-accent), #ffd166);
  box-shadow: 0 12px 24px rgba(242, 159, 5, 0.24);
}

.logout-btn {
  border: 0;
  border-radius: 999px;
  padding: 0.72rem 1.15rem;
  background: rgba(220, 53, 69, 0.14);
  color: #ffb5bf;
  font-weight: 700;
}
```

```
    .logout-btn:hover,
    .logout-btn:focus {
      background: rgba(220, 53, 69, 0.22);
      color: #ffd5db;
    }

    .main-shell {
      width: min(1220px, calc(100% - 24px));
      margin: 0 auto;
    }

    .alert {
      border-radius: 16px;
      border: 0;
    }

    .site-footer {
      width: min(1220px, calc(100% - 24px));
      margin: 0 auto 18px;
      border: 1px solid var(--user-line);
      border-radius: 20px;
      background: rgba(6, 18, 26, 0.72);
      backdrop-filter: blur(16px);
    }

    @media (max-width: 991.98px) {
      .user-navbar-shell,
      .main-shell,
      .site-footer {
        width: calc(100% - 20px);
      }

      .navbar .nav-link,
      .logout-btn {
        width: 100%;
        justify-content: flex-start;
      }
    }
  </style>
  {% block extra_css %}{% endblock %}
</head>
<body class="d-flex flex-column">
  <nav class="navbar navbar-expand-lg user-navbar-shell">
    <div class="container px-3 px-lg-4">
      <a class="navbar-brand d-flex align-items-center gap-3 fw-bold text-white" href="{% url
'userhome' %}">
        <span class="brand-chip"><i class="fas fa-shield-heart"></i></span>
```

```
      <span>FoodGuard Workspace</span>
    </a>
    <button class="navbar-toggler border-0 shadow-none" type="button" data-bs-
toggle="collapse" data-bs-target="#userNavbar">
      <span class="navbar-toggler-icon"></span>
    </button>

    <div class="collapse navbar-collapse" id="userNavbar">
      <ul class="navbar-nav ms-auto align-items-lg-center gap-lg-1">
        <li class="nav-item">
          <a class="nav-link {% if request.path == '/Users/userhome/' %}active{% endif %}"
href="{% url 'userhome' %}">
            <i class="fas fa-user"></i> Profile
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link {% if request.path == '/Users/prediction/' %}active{% endif %}"
href="{% url 'prediction' %}">
            <i class="fas fa-wand-magic-sparkles"></i> Prediction
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link {% if request.path == '/Users/proposed/' %}active{% endif %}"
href="{% url 'proposed' %}">
            <i class="fas fa-brain"></i> Proposed
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link {% if request.path == '/Users/history/' %}active{% endif %}"
href="{% url 'history' %}">
            <i class="fas fa-clock-rotate-left"></i> History
          </a>
        </li>
        <li class="nav-item">
          <a class="nav-link {% if request.path == '/Users/analytics/' %}active{% endif %}"
href="{% url 'analytics' %}">
            <i class="fas fa-chart-pie"></i> Analytics
          </a>
        </li>
        <li class="nav-item ms-lg-2">
          <form action="{% url 'user_logout' %}" method="post" class="d-inline">
            {% csrf_token %}
            <button type="submit" class="logout-btn">
              <i class="fas fa-sign-out-alt me-2"></i>Logout
            </button>
          </form>
        </li>
      </ul>
```

```
    </div>
   </div>
  </nav>

  {% if messages %}
   <div class="main-shell mt-3">
    {% for message in messages %}
     <div class="alert {% if message.tags == 'error' %}alert-danger{% elif message.tags ==
'success' %}alert-success{% elif message.tags == 'warning' %}alert-warning{% else %}alert-
info{% endif %} alert-dismissible fade show shadow-sm" role="alert">
       {{ message }}
       <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
     </div>
    {% endfor %}
   </div>
  {% endif %}

  <main class="flex-grow-1 py-4 py-lg-5">
   <div class="main-shell">
    {% block content %}{% endblock %}
   </div>
  </main>

  <footer class="site-footer text-center py-3 mt-auto">
   <p class="mb-0 text-white-50">&copy; 2025 Food Adulteration Detection. All rights
reserved.</p>
  </footer>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Userhome.html

```
{% extends 'user/userbase.html' %}
{% block title %}User Profile - Food Adulteration Detection{% endblock %}

{% block content %}
<section id="user-home" class="section">
  <div class="row justify-content-center">
   <div class="col-lg-8">
    <div class="card shadow-lg border-0 rounded-3">
     <div class="card-header bg-warning text-white text-center rounded-top">
      <h3 class="mb-0"><i class="fas fa-user-circle"></i> User Profile</h3>
     </div>
     <div class="card-body p-4">
```

```
        <div class="row mb-3">
          <div class="col-md-6">
            <h5 class="fw-bold">Username:</h5>
            <p class="text-muted">{{ user.username }}</p>
          </div>
          <div class="col-md-6">
            <h5 class="fw-bold">Full Name:</h5>
            <p class="text-muted">{{ user.first_name }} {{ user.last_name }}</p>
          </div>
        </div>

        <div class="row mb-3">
          <div class="col-md-6">
            <h5 class="fw-bold">Email:</h5>
            <p class="text-muted">{{ user.email }}</p>
          </div>
          <div class="col-md-6">
            <h5 class="fw-bold">Status:</h5>
            {% if user.is_active %}
              <span class="badge bg-success">Active</span>
            {% else %}
              <span class="badge bg-danger">Inactive</span>
            {% endif %}
          </div>
        </div>

        <div class="row mb-3">
          <div class="col-md-12">
            <h5 class="fw-bold">Last Login:</h5>
            <p class="text-muted">{{ user.last_login }}</p>
          </div>
        </div>

      </div>
    </div>
  </div>
 </div>
</section>
{% endblock %}
```

Base.html

```
<!DOCTYPE html>
{% load static %}
<html lang="en">
<head>
```

```
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}Food Adulteration Detection{% endblock %}</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link
href="https://fonts.googleapis.com/css2?family=Manrope:wght@400;500;600;700;800&family=
Space+Grotesk:wght@500;700&display=swap" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css"
rel="stylesheet">
  <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css"
rel="stylesheet">
  <style>
    :root {
      --bg: #08141f;
      --bg-soft: rgba(9, 22, 33, 0.78);
      --surface-light: rgba(255, 255, 255, 0.92);
      --text: #eff7f5;
      --muted: #b7c8c3;
      --heading: #ffffff;
      --accent: #f29f05;
      --line: rgba(255, 255, 255, 0.12);
      --shadow: 0 24px 60px rgba(0, 0, 0, 0.28);
      --radius-xl: 28px;
      --radius-lg: 20px;
    }

    body {
      min-height: 100vh;
      color: var(--text);
      font-family: "Manrope", sans-serif;
      background:
        linear-gradient(135deg, rgba(5, 15, 22, 0.88), rgba(8, 20, 31, 0.74)),
        radial-gradient(circle at top left, rgba(242, 159, 5, 0.25), transparent 34%),
        radial-gradient(circle at bottom right, rgba(28, 114, 102, 0.24), transparent 30%),
        url('{% static "img/bg.webp" %}') center/cover fixed no-repeat;
      position: relative;
    }

    body::before {
      content: "";
      position: fixed;
      inset: 0;
      background:
        linear-gradient(rgba(255, 255, 255, 0.03) 1px, transparent 1px),
        linear-gradient(90deg, rgba(255, 255, 255, 0.03) 1px, transparent 1px);
      background-size: 80px 80px;
      opacity: 0.24;
```

```
  pointer-events: none;
  z-index: -1;
}

h1, h2, h3, h4, h5, h6, .navbar-brand {
  font-family: "Space Grotesk", sans-serif;
  color: var(--heading);
  letter-spacing: -0.02em;
}

p, .text-muted, .lead {
  color: var(--muted) !important;
}

a {
  color: inherit;
  text-decoration: none;
}

.site-shell {
  position: relative;
  z-index: 1;
}

.navbar.app-navbar {
  margin: 18px auto 0;
  width: min(1180px, calc(100% - 24px));
  border: 1px solid var(--line);
  border-radius: 999px;
  background: rgba(6, 18, 26, 0.68);
  backdrop-filter: blur(18px);
  box-shadow: var(--shadow);
}

.brand-mark {
  width: 42px;
  height: 42px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 14px;
  background: linear-gradient(135deg, var(--accent), #ffd166);
  color: #10202b;
  font-size: 1rem;
  box-shadow: 0 10px 24px rgba(242, 159, 5, 0.3);
}
```

```
.navbar .nav-link {
  color: rgba(255, 255, 255, 0.74);
  font-weight: 600;
  padding: 0.75rem 1rem !important;
  transition: color 0.2s ease, transform 0.2s ease;
}

.navbar .nav-link:hover,
.navbar .nav-link:focus,
.navbar .nav-link.active {
  color: #fff;
  transform: translateY(-1px);
}

.btn-accent {
  background: linear-gradient(135deg, var(--accent), #ffbf47);
  border: 0;
  color: #10202b;
  font-weight: 800;
  border-radius: 999px;
  box-shadow: 0 12px 30px rgba(242, 159, 5, 0.35);
}

.btn-accent:hover,
.btn-accent:focus {
  color: #10202b;
  background: linear-gradient(135deg, #ffbf47, var(--accent));
}

.btn-outline-light-soft {
  border: 1px solid rgba(255, 255, 255, 0.2);
  background: rgba(255, 255, 255, 0.04);
  color: #fff;
  border-radius: 999px;
  font-weight: 700;
}

.btn-outline-light-soft:hover {
  background: rgba(255, 255, 255, 0.12);
  color: #fff;
}

.glass-panel {
  background: var(--bg-soft);
  border: 1px solid var(--line);
  border-radius: var(--radius-xl);
  backdrop-filter: blur(20px);
```

```
  box-shadow: var(--shadow);
}

.light-panel {
  background: var(--surface-light);
  border: 1px solid rgba(8, 20, 31, 0.08);
  border-radius: var(--radius-xl);
  box-shadow: 0 20px 50px rgba(4, 15, 22, 0.18);
  color: #10202b;
}

.light-panel h1,
.light-panel h2,
.light-panel h3,
.light-panel h4,
.light-panel h5,
.light-panel h6,
.light-panel label {
  color: #10202b !important;
}

.light-panel p,
.light-panel .text-muted {
  color: #52636c !important;
}

.eyebrow {
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.5rem 0.9rem;
  border-radius: 999px;
  background: rgba(242, 159, 5, 0.14);
  border: 1px solid rgba(242, 159, 5, 0.24);
  color: #ffe0a6;
  font-size: 0.85rem;
  font-weight: 700;
  text-transform: uppercase;
  letter-spacing: 0.08em;
}

.metric-card,
.feature-tile,
.info-tile {
  border-radius: var(--radius-lg);
  border: 1px solid rgba(255, 255, 255, 0.08);
  background: rgba(255, 255, 255, 0.05);
```

```
}

.metric-card,
.info-tile {
  padding: 1.2rem;
}

.feature-tile {
  height: 100%;
  padding: 1.4rem;
}

.metric-value {
  font-size: 1.8rem;
  font-weight: 800;
  color: #fff;
}

.feature-icon {
  width: 54px;
  height: 54px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 16px;
  background: rgba(242, 159, 5, 0.16);
  color: #ffd166;
  font-size: 1.15rem;
}

.form-control,
.form-select {
  border-radius: 14px;
  border: 1px solid rgba(16, 32, 43, 0.12);
  padding: 0.9rem 1rem;
  min-height: 52px;
  box-shadow: none !important;
}

.form-control:focus,
.form-select:focus {
  border-color: rgba(242, 159, 5, 0.6);
  box-shadow: 0 0 0 0.2rem rgba(242, 159, 5, 0.14) !important;
}

.auth-grid {
  min-height: calc(100vh - 190px);
```

```
    align-items: center;
  }

  .page-footer {
    width: min(1180px, calc(100% - 24px));
    margin: 0 auto 18px;
    border: 1px solid var(--line);
    border-radius: 24px;
    background: rgba(6, 18, 26, 0.72);
    backdrop-filter: blur(18px);
  }

  .alert {
    border-radius: 16px;
    border: 0;
  }

  @media (max-width: 991.98px) {
    .navbar.app-navbar,
    .page-footer {
      width: calc(100% - 20px);
    }

    .auth-grid {
      min-height: auto;
    }
  }
 </style>
 {% block extra_css %}{% endblock %}
</head>
<body class="d-flex flex-column">
 <div class="site-shell d-flex flex-column min-vh-100">
  <nav class="navbar navbar-expand-lg app-navbar">
   <div class="container px-3 px-lg-4">
    <a class="navbar-brand d-flex align-items-center gap-3 fw-bold" href="{% url 'index' %}">
     <span class="brand-mark"><i class="fa-solid fa-shield-heart"></i></span>
     <span>FoodGuard AI</span>
    </a>
    <button class="navbar-toggler border-0 shadow-none" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarNav">
     <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
     <ul class="navbar-nav ms-auto align-items-lg-center">
      <li class="nav-item"><a class="nav-link" href="{% url 'index' %}">Overview</a></li>
      <li class="nav-item"><a class="nav-link" href="{% url 'index'
%}#team">Team</a></li>
```

```
          <li class="nav-item"><a class="nav-link" href="{% url 'loginn' %}">Login</a></li>
          <li class="nav-item ms-lg-2"><a class="btn btn-accent px-4 py-2" href="{% url 'register' %}">Create Account</a></li>
        </ul>
      </div>
    </div>
  </nav>

  <main class="flex-grow-1 d-flex align-items-center py-4 py-lg-5">
    <div class="container">
      {% if messages %}
        <div class="row justify-content-center mb-3">
          <div class="col-lg-10">
            {% for message in messages %}
              <div class="alert {% if message.tags == 'error' %}alert-danger{% elif message.tags == 'success' %}alert-success{% elif message.tags == 'warning' %}alert-warning{% else %}alert-info{% endif %} alert-dismissible fade show shadow-sm" role="alert">
                {{ message }}
                <button type="button" class="btn-close" data-bs-dismiss="alert" aria-label="Close"></button>
              </div>
            {% endfor %}
          </div>
        </div>
      {% endif %}
      {% block content %}{% endblock %}
    </div>
  </main>

  <footer class="page-footer text-center py-4 mt-auto">
    <div class="container">
      <p class="mb-1 fw-semibold text-white">FoodGuard AI</p>
      <p class="mb-0">Professional monitoring workflows for food authenticity, risk review, and safety decisions.</p>
    </div>
  </footer>
 </div>
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

Index.html

```
{% extends 'base.html' %}
{% load static %}
{% block title %}Home - Food Adulteration Detection{% endblock %}
```

```
{% block extra_css %}
<style>
  .landing-hero {
    padding-top: 1.5rem;
  }

  .hero-orbit {
    position: relative;
    min-height: 100%;
    overflow: hidden;
  }

  .hero-orbit::before,
  .hero-orbit::after {
    content: "";
    position: absolute;
    border-radius: 50%;
    filter: blur(6px);
    opacity: 0.7;
  }

  .hero-orbit::before {
    width: 220px;
    height: 220px;
    top: -70px;
    right: -40px;
    background: radial-gradient(circle, rgba(242, 159, 5, 0.36), transparent 70%);
  }

  .hero-orbit::after {
    width: 260px;
    height: 260px;
    bottom: -110px;
    left: -50px;
    background: radial-gradient(circle, rgba(55, 180, 160, 0.28), transparent 72%);
  }

  .hero-grid {
    position: relative;
    z-index: 1;
  }

  .hero-stat {
    padding: 1.2rem;
    border-radius: 18px;
    background: rgba(255, 255, 255, 0.05);
    border: 1px solid rgba(255, 255, 255, 0.08);
```

```
}

.hero-stat strong {
  display: block;
  font-size: 2rem;
  color: #fff;
}

.spotlight-card {
  background: linear-gradient(180deg, rgba(255, 255, 255, 0.09), rgba(255, 255, 255, 0.03));
  border: 1px solid rgba(255, 255, 255, 0.08);
  border-radius: 24px;
  padding: 1.6rem;
}

.workflow-step {
  height: 100%;
  padding: 1.6rem;
  border-radius: 22px;
  background: rgba(255, 255, 255, 0.04);
  border: 1px solid rgba(255, 255, 255, 0.08);
}

.step-number {
  width: 42px;
  height: 42px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 50%;
  background: rgba(242, 159, 5, 0.18);
  color: #ffd166;
  font-weight: 800;
  margin-bottom: 1rem;
}

.team-card {
  height: 100%;
  padding: 1.75rem;
  border-radius: 24px;
  background: rgba(255, 255, 255, 0.06);
  border: 1px solid rgba(255, 255, 255, 0.1);
  transition: transform 0.2s ease, border-color 0.2s ease;
}

.team-card:hover {
  transform: translateY(-4px);
```

```
  border-color: rgba(242, 159, 5, 0.28);
}

.team-avatar {
  width: 68px;
  height: 68px;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  border-radius: 22px;
  background: linear-gradient(135deg, rgba(242, 159, 5, 0.92), rgba(255, 209, 102, 0.95));
  color: #10202b;
  font-size: 1.35rem;
  font-weight: 800;
  margin-bottom: 1rem;
}

.team-photo {
  width: 108px;
  height: 128px;
  object-fit: cover;
  object-position: center 24%;
  border-radius: 24px;
  display: block;
  margin-bottom: 1rem;
  border: 3px solid rgba(255, 255, 255, 0.14);
  box-shadow: 0 14px 28px rgba(0, 0, 0, 0.18);
}

.team-photo-siddharth {
  object-position: center 18%;
}

.team-photo-nagalaxmi {
  object-position: center 16%;
  width: 112px;
  height: 134px;
}

.team-photo-kirthi {
  object-position: center 14%;
  width: 112px;
  height: 134px;
}

.team-role {
  color: #ffd78f;
```

```
    font-size: 0.92rem;
    font-weight: 700;
    text-transform: uppercase;
    letter-spacing: 0.08em;
  }

  .cta-strip {
    background: linear-gradient(135deg, rgba(242, 159, 5, 0.16), rgba(255, 255, 255, 0.05));
    border: 1px solid rgba(242, 159, 5, 0.2);
    border-radius: 28px;
    padding: 2rem;
  }
</style>
{% endblock %}

{% block content %}
<section class="landing-hero">
  <div class="row g-4 align-items-center">
    <div class="col-lg-7">
      <div class="glass-panel p-4 p-lg-5">
        <span class="eyebrow mb-3">
          <i class="fa-solid fa-shield-virus"></i>
          Food quality intelligence
        </span>
        <h1 class="display-3 fw-bold mb-3">A sharper landing page for your food adulteration platform.</h1>
        <p class="lead mb-4">
          Detect suspicious products, assess severity faster, and present your system like a serious research-driven product instead of a plain demo page.
        </p>

        <div class="d-flex flex-wrap gap-3 mb-4">
          <a href="{% url 'register' %}" class="btn btn-accent btn-lg px-4">Get Started</a>
          <a href="{% url 'loginn' %}" class="btn btn-outline-light-soft btn-lg px-4">View Dashboard</a>
        </div>

        <div class="row g-3">
          <div class="col-sm-4">
            <div class="hero-stat">
              <strong>AI</strong>
              <span class="small text-white-50 text-uppercase">Severity prediction</span>
            </div>
          </div>
          <div class="col-sm-4">
            <div class="hero-stat">
              <strong>Safe</strong>
```

```
          <span class="small text-white-50 text-uppercase">Risk-focused review</span>
         </div>
        </div>
        <div class="col-sm-4">
         <div class="hero-stat">
          <strong>Fast</strong>
          <span class="small text-white-50 text-uppercase">Action-ready insights</span>
         </div>
        </div>
       </div>
      </div>
     </div>

     <div class="col-lg-5">
      <div class="glass-panel hero-orbit p-4 p-lg-5 h-100">
       <div class="hero-grid">
        <div class="spotlight-card mb-3">
         <p class="text-uppercase fw-bold text-white-50 mb-2">Platform Snapshot</p>
         <h3 class="mb-3">Designed for cleaner, more credible presentation</h3>
         <p class="mb-0">A modern interface for food safety analysis, prediction history, and
model-supported severity classification.</p>
        </div>

        <div class="row g-3">
         <div class="col-12">
          <div class="feature-tile">
           <div class="feature-icon mb-3"><i class="fa-solid fa-magnifying-glass-
chart"></i></div>
           <h5>Smart record review</h5>
           <p class="mb-0">Capture product, adulterant, and detection details in a clean,
structured interface.</p>
          </div>
         </div>
         <div class="col-md-6">
          <div class="feature-tile">
           <div class="feature-icon mb-3"><i class="fa-solid fa-chart-line"></i></div>
           <h5>Analytics ready</h5>
           <p class="mb-0">Present your data in a way that feels fit for demos and
evaluation.</p>
          </div>
         </div>
         <div class="col-md-6">
          <div class="feature-tile">
           <div class="feature-icon mb-3"><i class="fa-solid fa-user-shield"></i></div>
           <h5>Trust first</h5>
           <p class="mb-0">Build confidence with a more professional visual experience.</p>
          </div>
         </div>
```

```
        </div>
      </div>
    </div>
  </div>
 </div>
</section>

<section class="py-4 py-lg-5">
 <div class="row g-4">
  <div class="col-lg-4">
   <div class="workflow-step">
    <div class="step-number">1</div>
    <h4>Capture the case</h4>
    <p class="mb-0">Add product details, suspected adulterants, detection methods, and food safety context in one flow.</p>
   </div>
  </div>
  <div class="col-lg-4">
   <div class="workflow-step">
    <div class="step-number">2</div>
    <h4>Predict severity</h4>
    <p class="mb-0">Use the model to estimate severity and make prioritization easier for the next response step.</p>
   </div>
  </div>
  <div class="col-lg-4">
   <div class="workflow-step">
    <div class="step-number">3</div>
    <h4>Review outcomes</h4>
    <p class="mb-0">Track prediction history and analytics from a cleaner, more polished dashboard experience.</p>
   </div>
  </div>
 </div>
</section>

<section id="team" class="py-3 py-lg-4">
 <div class="glass-panel p-4 p-lg-5">
  <div class="d-flex flex-column flex-lg-row justify-content-between align-items-lg-end gap-3 mb-4">
   <div>
    <span class="eyebrow mb-3">
     <i class="fa-solid fa-people-group"></i>
     Our team
    </span>
    <h2 class="mb-2">The people behind the platform</h2>
    <p class="mb-0">Project team and faculty guidance for the Food Adulteration Detection platform.</p>
```

```
    </div>
    <p class="mb-0 text-white-50">Built with a mix of food safety awareness, machine learning, and product design.</p>
   </div>

   <div class="row g-4">
    <div class="col-md-6 col-xl-3">
     <div class="team-card">
      <img src="{% static 'img/team/mujeeb.jpeg' %}" alt="Shaikh Mujeeb" class="team-photo">
      <div class="team-role mb-2">Project Lead</div>
      <h5 class="mb-2">Shaikh Mujeeb</h5>
      <p class="mb-1"><strong>Roll No:</strong> 228R1A66C1</p>
      <p class="mb-0">Contributed to the project direction, system integration, and overall implementation flow.</p>
     </div>
    </div>
    <div class="col-md-6 col-xl-3">
     <div class="team-card">
      <img src="{% static 'img/team/siddharth.jpeg' %}" alt="Siddhrath Reddy" class="team-photo team-photo-siddharth">
      <div class="team-role mb-2">Team Member</div>
      <h5 class="mb-2">Siddhrath Reddy</h5>
      <p class="mb-1"><strong>Roll No:</strong> 228R1A6674</p>
      <p class="mb-0">Supported development, research work, and project coordination across the application workflow.</p>
     </div>
    </div>
    <div class="col-md-6 col-xl-3">
     <div class="team-card">
      <img src="{% static 'img/team/kirthi.jpeg' %}" alt="Kirthi" class="team-photo team-photo-kirthi">
      <div class="team-role mb-2">Team Member</div>
      <h5 class="mb-2">Keerthi</h5>
      <p class="mb-1"><strong>Roll No:</strong> 228R1A6680</p>
      <p class="mb-0">Worked on project support tasks, documentation, and helping shape the user-facing experience.</p>
     </div>
    </div>
    <div class="col-md-6 col-xl-3">
     <div class="team-card">
      <img src="{% static 'img/team/nagalaxmi.jpeg' %}" alt="Naga Laxmi" class="team-photo team-photo-nagalaxmi">
      <div class="team-role mb-2">Team Member</div>
      <h5 class="mb-2">J NagaLaxmi</h5>
      <p class="mb-1"><strong>Roll No:</strong> 228R1A6692</p>
      <p class="mb-0">Contributed to project execution, content preparation, and refinement of the final presentation.</p>
```

```
      </div>
    </div>
    <div class="col-md-6 col-xl-3">
      <div class="team-card">
        <div class="team-avatar">S</div>
        <div class="team-role mb-2">Guide and Mentor</div>
        <h5 class="mb-2">Mrs. Swaroopa Madam</h5>
        <p class="mb-0">Provided academic guidance, project mentorship, and valuable direction throughout the development process.</p>
      </div>
    </div>
  </div>
 </div>
</section>

<section class="py-4">
 <div class="cta-strip">
  <div class="row align-items-center g-3">
   <div class="col-lg-8">
    <h3 class="mb-2">Ready to Test You food on our food safety platform?</h3>
    <p class="mb-0">Register now and Get the most out of it for free</p>
   </div>
   <div class="col-lg-4 text-lg-end">
    <a href="{% url 'register' %}" class="btn btn-accent btn-lg px-4">Create Account</a>
   </div>
  </div>
 </div>
</section>
{% endblock %}
```

Login.html

```
{% extends 'base.html' %}
{% block title %}Login - Food Adulteration Detection{% endblock %}

{% block content %}
<div class="col-md-6 col-lg-5 mx-auto bg-white p-5 rounded shadow-lg">
 <h2 class="fw-bold mb-4 text-center">Login</h2>

 {% if messages %}
 <div class="container mb-4">
  {% for message in messages %}
   <div class="alert
        {% if message.tags == 'error' %}alert-danger
        {% elif message.tags == 'success' %}alert-success
        {% elif message.tags == 'warning' %}alert-warning
        {% else %}alert-info{% endif %}
```

```
            alert-dismissible fade show shadow-sm" role="alert">
      {{ message }}
      <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
    </div>
   {% endfor %}
 </div>
{% endif %}

 <form action="{% url 'user_login' %}" method="post">
   {% csrf_token %}
   <div class="mb-3">
    <input type="text" name="username" class="form-control" placeholder="Your Username"
required>
   </div>
   <div class="mb-3">
    <input type="password" class="form-control" name="password" placeholder="Your
Password" required>
   </div>
   <div class="text-center">
    <button type="submit" class="btn btn-warning text-white px-5 py-2 rounded-pill shadow">
     <i class="fas fa-sign-in-alt"></i> Login
    </button>
   </div>
 </form>
</div>
{% endblock %}
```

Resgister.html

```
{% extends 'base.html' %}
{% block title %}Register - Food Adulteration Detection{% endblock %}

{% block content %}
<div class="col-md-8 col-lg-6 mx-auto bg-white p-5 rounded shadow-lg">
 <h2 class="fw-bold mb-4 text-center">Create an Account</h2>
 {% if messages %}
 <div class="container mb-4">
   {% for message in messages %}
    <div class="alert
            {% if message.tags == 'error' %}alert-danger
            {% elif message.tags == 'success' %}alert-success
            {% elif message.tags == 'warning' %}alert-warning
            {% else %}alert-info{% endif %}
            alert-dismissible fade show shadow-sm" role="alert">
      {{ message }}
      <button type="button" class="btn-close" data-bs-dismiss="alert" aria-
label="Close"></button>
```

```
    </div>
   {% endfor %}
  </div>
{% endif %}

  <form action="{% url 'user_registration' %}" method="post">
   {% csrf_token %}
   <div class="row g-3">
    <div class="col-md-6">
     <input type="text" name="first_name" class="form-control" placeholder="First Name"
required>
    </div>
    <div class="col-md-6">
     <input type="text" name="last_name" class="form-control" placeholder="Last Name"
required>
    </div>
    <div class="col-md-6">
     <input type="text" name="username" class="form-control" placeholder="Username"
required>
    </div>
    <div class="col-md-6">
     <input type="email" name="email" class="form-control" placeholder="Your Email"
required>
    </div>
    <div class="col-md-6">
     <input type="password" name="password" class="form-control" placeholder="Password"
required>
    </div>
    <div class="col-md-6">
     <input type="password" name="confirm_password" class="form-control"
placeholder="Confirm Password" required>
    </div>
   </div>

   <div class="text-center mt-4">
    <button type="submit" class="btn btn-warning text-white px-5 py-2 rounded-pill shadow">
     <i class="fas fa-user-plus"></i> Register
    </button>
   </div>
  </form>
</div>
{% endblock
```

6.2 Implementation:

The Food Adulteration Detection project is designed as a complete web-based workflow that combines secure user management with machine learning prediction. The user journey begins on the home page, where a new user registers by entering personal details such as name, username, email, and password. During registration, the system performs validation checks (for example, password match and unique username/email) and creates the account in an inactive state, ensuring controlled access until an admin approves the user. After approval, the user logs in through the authentication page, and based on role, the system redirects them to the appropriate dashboard. Regular users are taken to the user panel, where they can submit food-related input parameters for adulteration analysis.

Once the user submits input, the backend prepares the data in structured format and applies the same label encoding logic used during model training. This is important because the model expects encoded categorical features in a consistent format. The processed input is then passed to the pre-trained classification model, which predicts the adulteration severity level and generates probability scores for possible classes. The predicted result is displayed clearly to the user in the interface, enabling quick interpretation. In addition to immediate output, the system stores each prediction in the database as historical data, allowing users to review previous results and helping admins monitor usage patterns, reporting, and trends. The application also includes proper session handling, message alerts, and logout functionality to maintain usability and security throughout the user lifecycle.

From an implementation perspective, the project follows a practical modular structure: Django handles routing, templates, authentication, and persistence, while the ML layer handles prediction logic through serialized model artifacts. This separation keeps the system maintainable and extendable, making it easier to retrain the model, update UI pages, or add analytics features later. Overall, the project demonstrates how a machine learning model can be integrated into a real-world web application with role-based access and end-to-end user flow.

Conclusion:

In conclusion, this project successfully delivers an end-to-end intelligent web system for food adulteration severity prediction. It not only performs ML-based inference but also supports real operational needs such as controlled registration, admin approval, secure login, prediction history, and user-friendly interaction. By combining Django's robust web framework with a trained classification model, the system provides a reliable foundation for practical deployment and can be further enhanced with larger datasets, real-time monitoring, and advanced analytics for broader impact.

7. SYSTEM TESTING

System testing was carried out to validate the complete Food Adulteration Detection application after integrating frontend, backend, database, and machine learning modules..

7.1 Types of System Testing

- Validate end-to-end workflow for both user and admin.
- Confirm correct integration between Django views, templates, database, and ML prediction module.
- Ensure role-based access and login behavior are working as intended.
- Check system stability for valid and invalid inputs.
- Confirm prediction records are stored and retrievable.

7.2 Test Strategies

- User registration and validation.
- Admin approval and user activation flow.
- User authentication and logout.
- Role-based redirection logic.
- Prediction form submission and model response.
- Prediction history storage and display.
- Admin report/dashboard visibility.

.

7.2.1 Test Environment

- Application type: Django web application.
- Backend: Python + Django.
- Database: SQLite (db.sqlite3).
- ML dependencies: scikit-learn, pandas, joblib, numpy, scipy.
- Run mode: Local development server (manage.py runserver).
- Browser testing: Standard desktop browser (form and navigation validation

.

7.2.2 Testing Approach

- Functional test execution for each major page and action.
- Integration tests through full user journey simulation.
- Negative testing with invalid credentials, duplicate usernames/emails.
- Data persistence checks for prediction history and account status updates.
- Manual verification of UI feedback messages for success/error cases.

7.2.3 Testing Approach

Registration with valid data

Expected: User account is created successfully.

Actual: Account created; success message displayed.

Status: Passed.

Registration with duplicate username/email

Expected: Registration blocked with error message.

Actual: Duplicate entries rejected; user notified.

Status: Passed.

Registration with password mismatch

Expected: Account not created; warning/error shown.

Actual: System rejected input and displayed message.

Status: Passed.

Login with inactive user

Expected: Login denied with “inactive” style message.

Actual: Access blocked until admin activation.

Status: Passed.

Admin/staff login redirection

Expected: Staff or superuser redirected to admin module (adminhome).

Actual: Correct role-based redirect observed.

Status: Passed.

Normal user login redirection

Expected: Active normal user redirected to user dashboard (userhome).

Actual: Correctly redirected.

Status: Passed.

Prediction submission with valid values

Expected: Model inference runs and returns severity + probabilities.

Actual: Output generated and displayed successfully.

Status: Passed.

Prediction history recording

Expected: Prediction stored with user mapping.

Actual: History data persisted and visible in history/report screens.

Status: Passed.

Logout behavior

Expected: Session cleared and user returned to public page.

Actual: Logout successful; protected navigation blocked post-logout.

Status: Passed.

7.3 Sample Test Cases

Test Case ID	Module	Test Scenario	Input	Expected Result	Actual Result	Status
TC_01	Registration	Register with valid details	Unique username/email, matching passwords	Account created successfully, user set as inactive, success message shown	Account created successfully and success message displayed	Pass
TC_02	Registration	Register with existing username	Existing username + new email	Registration blocked, "Username already exists" message	"Username already exists" message shown	Pass
TC_03	Registration	Register with existing email	New username + existing email	Registration blocked, "Email already exists" message	"Email already exists" message shown	Pass
TC_04	Registration	Password and confirm password mismatch	Different password values	Registration blocked, "Passwords do not match" message	"Passwords do not match" message shown	Pass
TC_05	Login	Login with inactive user	Valid inactive user credentials	Login denied, inactive account warning shown	Inactive account warning displayed	Pass
TC_06	Login	Login with invalid credentials	Wrong username/password	Login denied, "Invalid username or password" message	Error message displayed	Pass
TC_07	Role Redirect	Login as admin/staff	Valid admin credentials	Redirected to admin home page	Successfully redirected to admin page	Pass
TC_08	Role Redirect	Login as normal active user	Valid non-admin credentials	Redirected to user home page	Successfully redirected to user page	Pass
TC_09	Prediction	Submit valid food parameters	Valid form data	Severity prediction + probability output displayed	Prediction output displayed correctly	Pass
TC_10	Prediction History	Save and view prediction history	Successful prediction submission	New record stored and visible in history page	Record stored and visible	Pass
TC_11	Admin Control	Approve/activate registered user	Admin updates user status	User becomes active and can log in	User activated successfully	Pass

Test Case ID	Module	Test Scenario	Input	Expected Result	Actual Result	Status
TC_12	Logout	User logout	Click logout action	Session cleared, redirected to public/home page	Successfully logged out and redirected	Pass

Table no 7.3 Test Cases

8. OUTPUT SCREENS

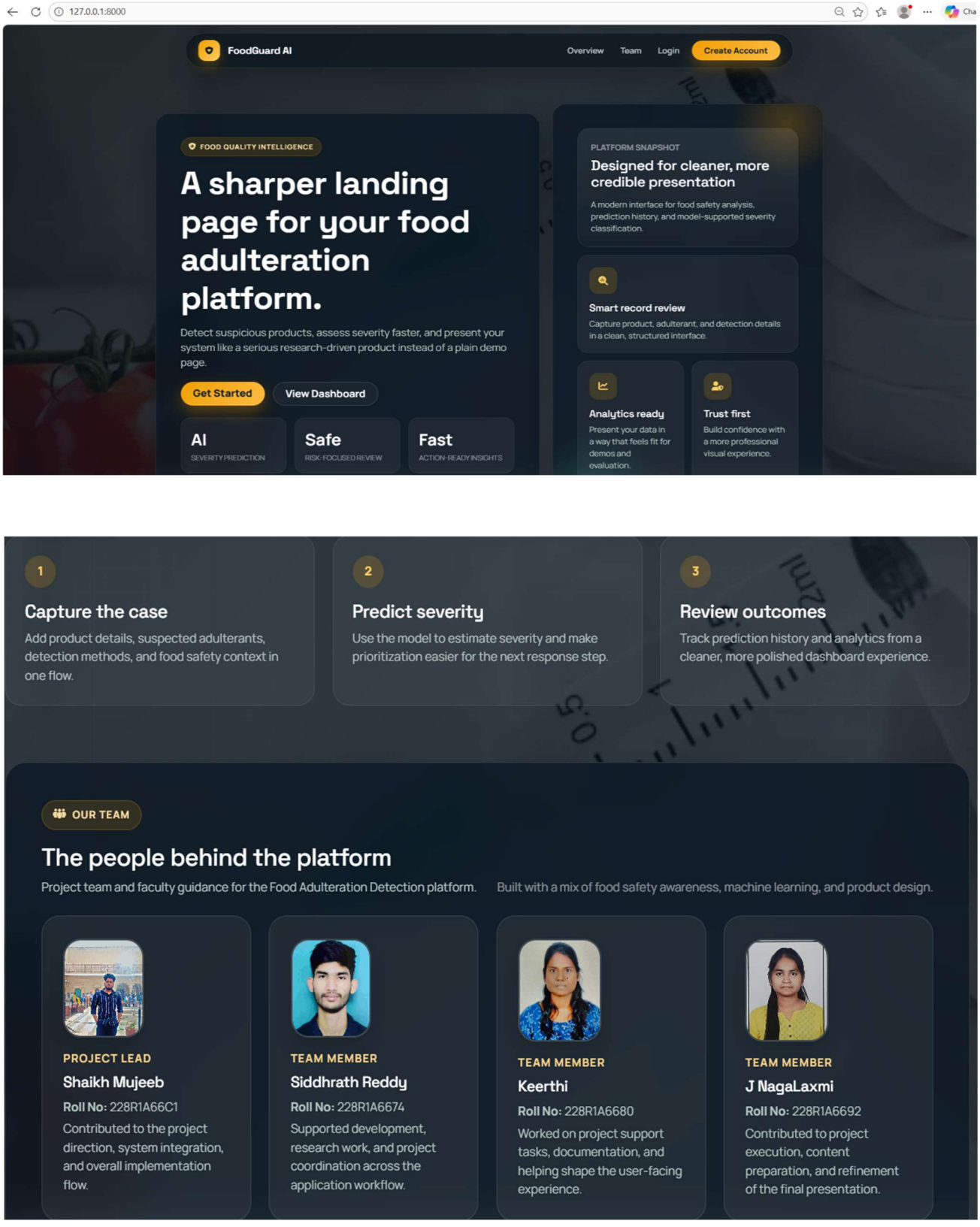


Fig 8.1 Food Adulteration System Landing page

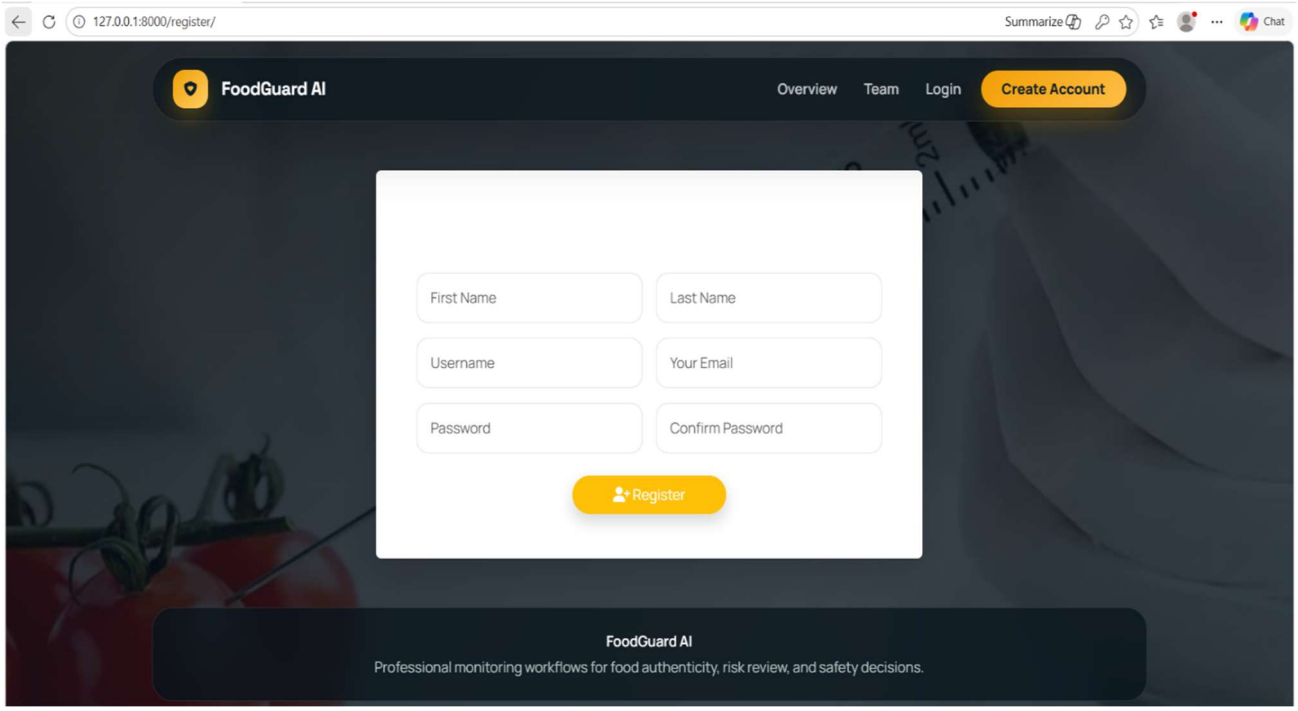


Fig 8.2 User Registration

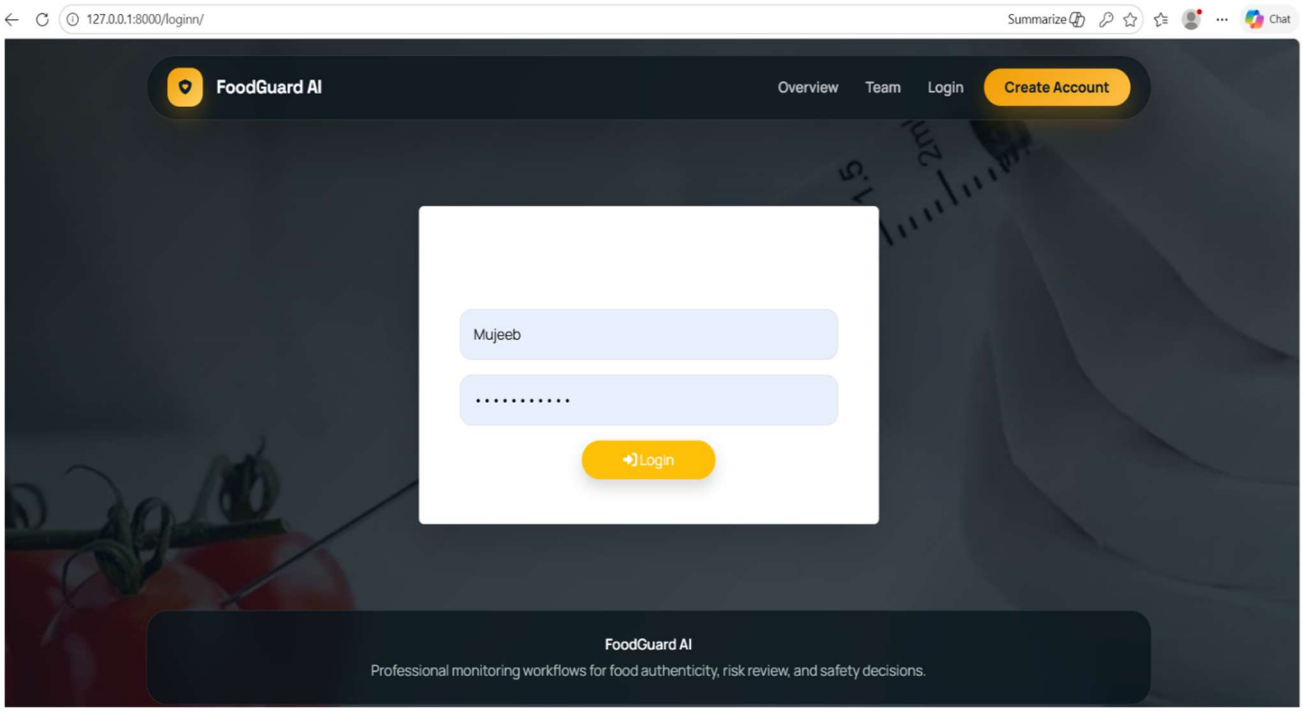


Fig 8.3 User Login

Description: The entry point for the platform, featuring Registration and Login modules. This interface ensures secure access to the ensemble-based detection tools for social media monitoring.

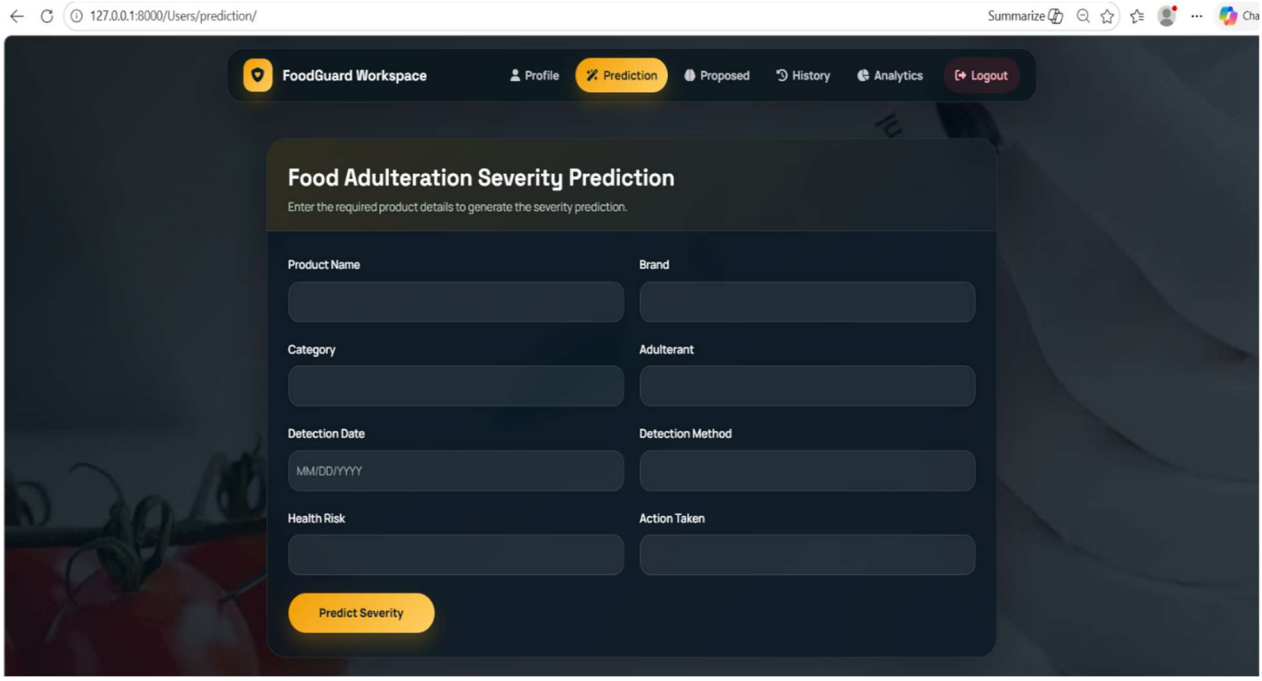


Fig 8.4 Prediction page

Description: This screen shows the prediction module in action. The input text is processed by a Soft-Voting Ensemble, which aggregates the probability scores from both the Boosted Decision Tree and Bagging Random Forest to accurately predict "Food Adulterant."

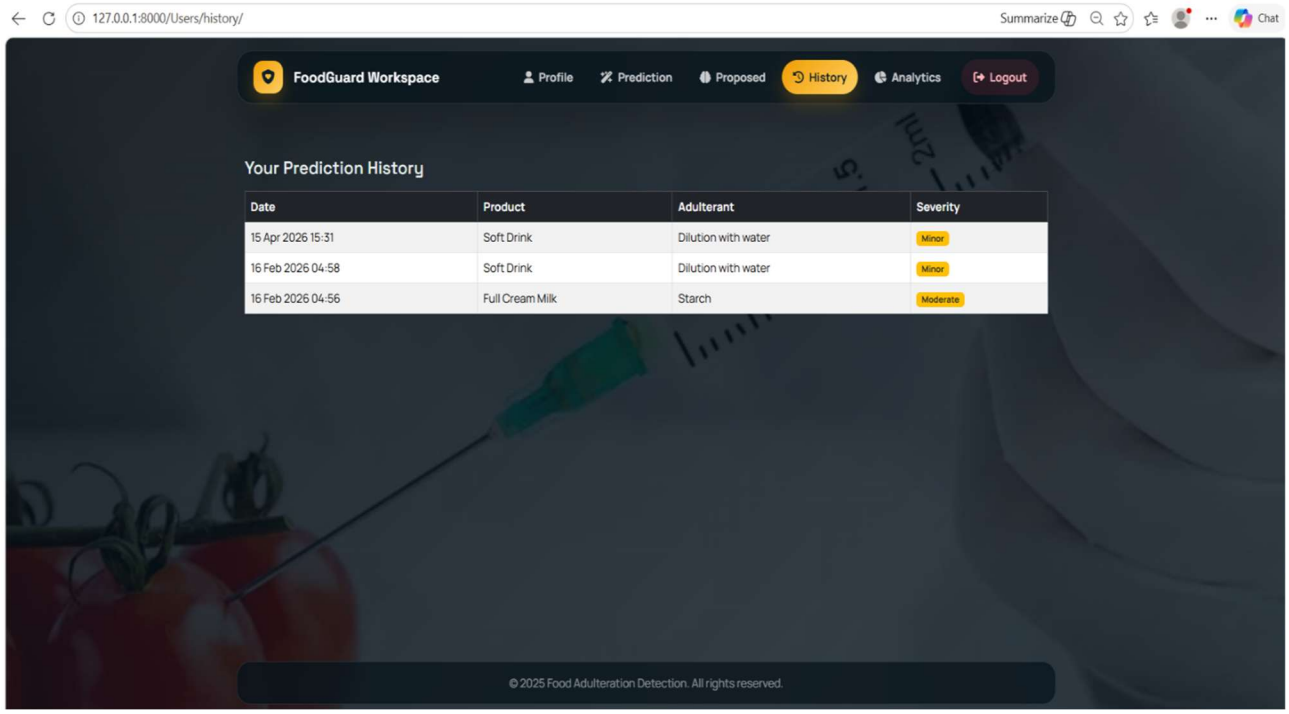


Fig 8.5 History page

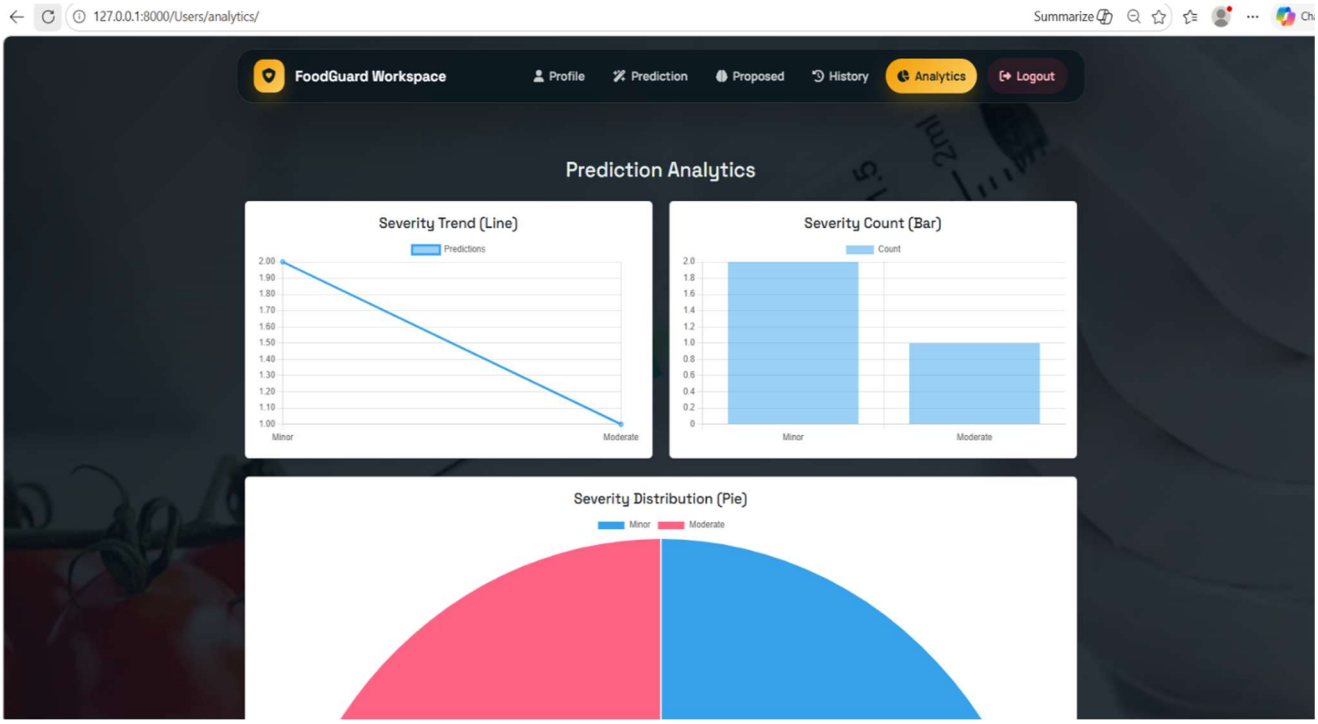


Fig 8.6 Analytics Page

9. CONCLUSION

The Hybrid Machine Learning–Based Food Adulteration Detection System presented in this project provides an efficient and intelligent approach for identifying adulterated food samples using structured food inspection data. By integrating multiple machine learning classifiers through ensemble techniques such as voting and stacking, the proposed system improves prediction accuracy, robustness, and generalization compared to conventional single-model approaches. The implementation of preprocessing, feature engineering, and hybrid modeling enables the system to effectively analyze food adulteration patterns and generate reliable predictions.

The developed framework successfully classifies food samples as pure or adulterated while also providing severity levels and confidence scores to support better decision-making. Compared to traditional laboratory-based methods, the proposed system offers a more scalable, cost-effective, and real-time alternative for food quality assessment. Overall, the project demonstrates that hybrid ensemble learning can significantly enhance food adulteration detection performance and provides a practical solution for food producers, inspectors, and regulatory authorities to ensure food safety and product authenticity.

The proposed system demonstrates the practical applicability of machine learning in modern food safety and quality assurance processes. By utilizing tabular chemical, physical, and inspection-related attributes from the food adulteration dataset, the framework eliminates the dependency on expensive laboratory equipment for preliminary screening. Its modular architecture, consisting of data acquisition, preprocessing, feature extraction, hybrid modeling, and output generation, ensures flexibility, maintainability, and ease of deployment in real-world environments. The user-friendly interface and structured prediction pipeline further enhance the accessibility of the system for inspectors and administrators.

In addition, the experimental outcomes validate that ensemble-based hybrid modeling provides improved stability when handling diverse and noisy datasets, making the system more suitable for practical deployment across different food categories and adulteration scenarios. The generated performance metrics, confidence scores, and severity indicators enhance interpretability and trust in the predictions. Hence, the proposed framework not only strengthens food adulteration detection capabilities but also establishes a foundation for future intelligent food quality monitoring systems that can support scalable and automated inspection processes.

10. FUTURE ENHANCEMENTS

Although the developed framework provides a strong and effective foundation for food adulteration detection, several enhancements can be implemented to further improve its performance, scalability, adaptability, and real-world applicability. As food adulteration methods continue to evolve and become more sophisticated, continuous improvements in both machine learning models and system architecture are essential to maintain high prediction accuracy and robustness.

One of the primary directions for future enhancement is the integration of more advanced machine learning and deep learning techniques into the hybrid framework. While the current system employs ensemble methods using classifiers such as Random Forest, Support Vector Machine, and Gradient Boosting, future versions can incorporate neural network-based models and deeper ensemble architectures to capture more complex adulteration patterns. Combining deep learning models with the current hybrid ensemble framework may further improve prediction accuracy, especially when larger and more diverse datasets become available.

Another important enhancement involves expanding the dataset to include a broader variety of food categories, adulterant types, and regional food products. The present implementation relies on structured tabular data from the Kaggle Food Adulteration Dataset, which may not cover all real-world adulteration scenarios. Incorporating additional datasets from laboratory records, regulatory agencies, and industrial food quality databases can improve model generalization and make the system more applicable across diverse food inspection environments.

The system can also be enhanced by integrating real-time data acquisition from sensors, IoT devices, or portable food testing instruments. Such integration would allow automatic collection of food quality parameters and direct transmission of measurements into the prediction pipeline, enabling fully automated and real-time adulteration detection. This would further reduce manual data entry and improve the practicality of deployment in production facilities, warehouses, and inspection centers.

Furthermore, the current web-based implementation can be extended into a scalable cloud or enterprise-grade deployment architecture to support large-scale usage. Backend optimizations such as API enhancement, asynchronous request handling, caching, and distributed processing can improve performance when handling multiple simultaneous prediction requests. This would make the system suitable for deployment in industrial food quality monitoring platforms and government inspection systems.

In addition, adaptive retraining mechanisms can be incorporated to continuously improve model performance over time. By periodically retraining the system with newly collected food adulteration samples and updated datasets, the framework can remain effective against emerging

adulteration patterns and changing food composition trends. This continuous learning capability would ensure long-term relevance and improved prediction robustness.

From an analytical perspective, future enhancements may include advanced visualization dashboards and reporting modules to provide richer insights into adulteration trends, severity distributions, and historical inspection outcomes. Such dashboards would assist food inspectors, producers, and regulatory authorities in monitoring food safety trends and making informed decisions.

Finally, future versions of the system can focus on strengthening security, data privacy, and explainability features. Implementing secure data handling, role-based access control, and explainable AI techniques can improve transparency, accountability, and user trust in prediction results. Collectively, these enhancements will transform the current framework into a more intelligent, scalable, and practical platform for comprehensive food adulteration detection and food quality assurance.

11. REFERENCES

1. **N. Kumar, R. Sharma, and P. Singh,** "Infrared Spectroscopy and Genetic Algorithm Based Spectral Analysis for Honey Adulteration Detection," 2025. Honey adulteration was detected using infrared spectroscopy combined with genetic algorithm-based feature selection, achieving highly accurate adulterant identification.
2. **M. Aqeel, H. Tariq, and S. Rahman,** "Milk Adulteration Identification Using Hyperspectral Imaging and Machine Learning," 2025. This study presents a multiclass machine learning model for identifying milk adulterants using hyperspectral imaging, demonstrating practical applicability for milk quality assessment.
3. **S. Shi, Y. Chen, and Z. Wang,** "Spectroscopic Techniques Combined with Chemometrics for Food Adulteration Detection," 2025. The study reviews advanced spectroscopic and chemometric methods for detecting adulteration and improving food authenticity analysis.
4. **D. A. Magdas, M. T. Pop, and A. Dehelean,** "The Journey of Artificial Intelligence in Food Authentication," 2025. This work discusses the growing role of AI in food fraud and adulteration detection, emphasizing intelligent automation in food authentication.
5. **U. Agrawal, N. Bawane, and N. Alsubaie,** "Design and Development of Adulteration Detection System by Fumigation Method and Machine Learning Techniques," *Scientific Reports*, 2024. The authors propose a machine learning-based edible oil adulteration detection system using refractive index and sensor-based features, achieving very high classification accuracy.
6. **K. Goyal, P. Kumar, and K. Verma,** "Food Adulteration Detection Using Artificial Intelligence: A Systematic Review," *Archives of Computational Methods in Engineering*, 2021. This systematic review analyzes the role of AI, machine learning, and deep learning in food adulteration detection across multiple food categories.
7. **S. Ayothi and R. Karthikeyan,** "A Qualitative Analysis of Machine Learning Methods in Food Adultery: A Focus on Milk Adulteration Detection," 2020. The paper reviews machine learning techniques specifically developed for milk adulteration detection and compares their effectiveness.
8. **L. Zhou, C. Zhang, F. Liu, and Y. He,** "Application of Deep Learning in Food: A Review," *Comprehensive Reviews in Food Science and Food Safety*, 2019. This review explores deep learning applications in food quality detection, contamination analysis, and food authentication.
9. **J. J. de Macedo Neto, J. A. dos Santos, and W. R. Schwartz,** "Meat Adulteration Detection Through Digital Image Analysis of Histological Cuts Using LBP," 2016. This study proposes digital image analysis using Local Binary Pattern (LBP) for identifying adulterated bovine meat samples.

10. **(Optional Replacement / General Review if Needed) Y. K. Anagaw et al.,** "Food Adulteration: Causes, Risks, and Detection Techniques – Review," 2024. This review summarizes causes, public health risks, and modern analytical techniques used in food adulteration detection.I. Hussain, A. Ullah, and M. N. Khan, "Differential Transfer Learning with Ensemble Optimization for Hate Speech Detection," Scientific Reports, 2024.
11. Stevens L. J., Burgess J. R., Stochelski M. A., and Kuczek T., "Presence of Artificial Food Dyes and Sugar in Children's Foods," 2015.
12. Zhou C., Wang Y., and Li X., "Stabilizing Meat Colour Using Ligand Coordination with Hemin," 2014.
13. Tian H., Wang J., and Zhang X., "Discrimination of Chicken vs. Beef Seasonings Using Electronic Nose," 2014.
14. Barbin D. F., Felicio A. L. S. M., Sun D. W., Nixdorf S. L., and Hirooka E. Y., "Infrared Spectral Analysis for Coffee Quality Assessment," 2014.
15. Dixit S., Purshottam S. K., and Khanna S. K., "Exposure Risk of Food Colours in India," 2013.
16. Ates E., Goda Y., Tsumura Y., and Ohta H., "Detection of Legal and Illegal Food Colours Using LC/MS with Cloud Point Extraction," 2011.
17. Boga A. and Binokay S., "Effects of Food Additives on Human Health," 2010.
18. Uzan A. and Delaveau P., "Salt Content in Foods as a Public Health Concern," 2009.
19. Fik M., Surówka K., and Maciejaszek I., "Shelf-Life Improvement of Ground Beef Using Potassium Lactate and Sodium Diacetate," 2008.
20. Brosnan T. and Sun D. W., "Computer Vision in Food Quality Inspection," 2004.