A GUI-BASED WEB INTERFACE FOR CREDIT CARD FRAUD DETECTION SYSTEM

USER GUIDE

1. Introduction

This user guide provides step-by-step instructions on how to install, configure, and use the GUI-Based Web Interface for Credit Card Fraud Detection System. The application is built using Python, Flask, and machine learning techniques to detect fraudulent credit card transactions based on an imbalanced dataset. The guide will cover the installation, execution, and usage of the system.

2. System Requirements

Before installing the application, ensure that your system meets the following requirements:

Operating System: Windows, macOS, or Linux

Python Version: 3.7 or higher

Required Packages: Flask, scikit-learn, pandas, numpy

Browser: Google Chrome, Firefox, or any modern web browser

Tools: Anaconda, Jupyter Notebook, Spyder

3. Installation Guide

Step 1: Install Python

If Python is not installed on your machine, download and install the latest version from the official website:

Python Download

Step 2: Install Anaconda

Anaconda is a distribution of Python that simplifies package management and deployment. Install Anaconda to use tools like Spyder and Jupyter Notebook

Anaconda Download

Step 3: Install Required Packages

Install Flask and other necessary libraries by running:

```
pip install flask scikit-learn pandas numpy
```

Step 4: Clone or Download the Project Files

Download the project from the repository or transfer it to your local machine.

4. Execution Process

Step 1: Running the Jupyter Notebook

Open Jupyter Notebook:

Launch Jupyter Notebook from Anaconda Navigator or by running the following command in your terminal/command prompt:

```
jupyter notebook
```

Navigate to the Notebook:

Locate and open the notebook **Credit Card Fraud Detection.ipynb** provided in the project directory

Run the Notebook Cells:

Execute each cell in the notebook sequentially to train the machine learning model. The notebook will train a model (e.g., Random Forest Classifier) on the credit card fraud detection dataset.

Save the Model:

Once the model is trained, ensure that the trained model is saved in both .pkl and .joblib formats within the same project directory.

After saving the model files (credit-card-model.pkl and Credit_Card_Model.joblib), ensure they are stored in the project folder.

Step 2: Running the Tkinter GUI

Run the Tkinter GUI in Jupyter Notebook:

If you prefer a desktop application interface, the project also includes a Tkinter-based GUI for predicting credit card fraud.

Execute the Tkinter GUI Code:

In the notebook, run the code to launch the GUI

Use the Tkinter GUI:

The GUI will allow you to upload transaction data and receive fraud predictions directly through a desktop window, without needing to open a browser.

Step 3: Running the Flask Application in Spyder

Open Spyder IDE:

Open Spyder from Anaconda Navigator or by running the following command in your terminal/command prompt:

```
Spyder
```

Navigate to app.py:

In Spyder, open the **app.py** file located in the project directory. This file contains the Flask web server code responsible for running the web interface of the fraud detection system.

Run the Flask Application:

In Spyder, click the "Run" button or press F5 to execute the app.py script. Once the server starts, you will see a message in the Spyder console:

```
Running on http://127.0.0.1:5000/
```

Access the Web Interface:

Open your web browser and go to the provided link (e.g., http://127.0.0.1:5000/). You will see a website named **FraudShield Analytics**, where you can upload transaction data and receive predictions on whether the transactions are fraudulent.

This provides both GUI-based and Web-based user interfaces to interact with the credit card fraud detection system, allowing flexibility depending on the user's preference.

5. Features Overview

Real-Time Fraud Detection:

Enter transaction data manually through the provided input fields, and the system will analyze each entry using machine learning algorithms to detect fraudulent activity in real time.

User-Friendly Interface:

The system offers both web-based and desktop applications. The web interface is built using Flask, HTML, CSS, and JavaScript, ensuring a smooth and responsive experience. The Tkinter-based desktop GUI provides an intuitive alternative for local usage.

Result Display:

After submitting a transaction, the system instantly displays whether the transaction is fraudulent or legitimate based on the analysis.

Model Accuracy:

The application utilizes a trained Random Forest Classifier model, which achieves high accuracy, ensuring reliable and effective fraud detection.

6. Troubleshooting

Issue 1: Flask Server Not Starting

- Ensure that Flask is installed correctly. You can check by running pip list to see if Flask is included.
- Make sure the project's directory is activated.

Issue 2: Page Not Loading in Browser

- Ensure the correct URL (http://127.0.0.1:5000/) is entered in the browser.
- Check the terminal to ensure that the Flask server is running without errors.

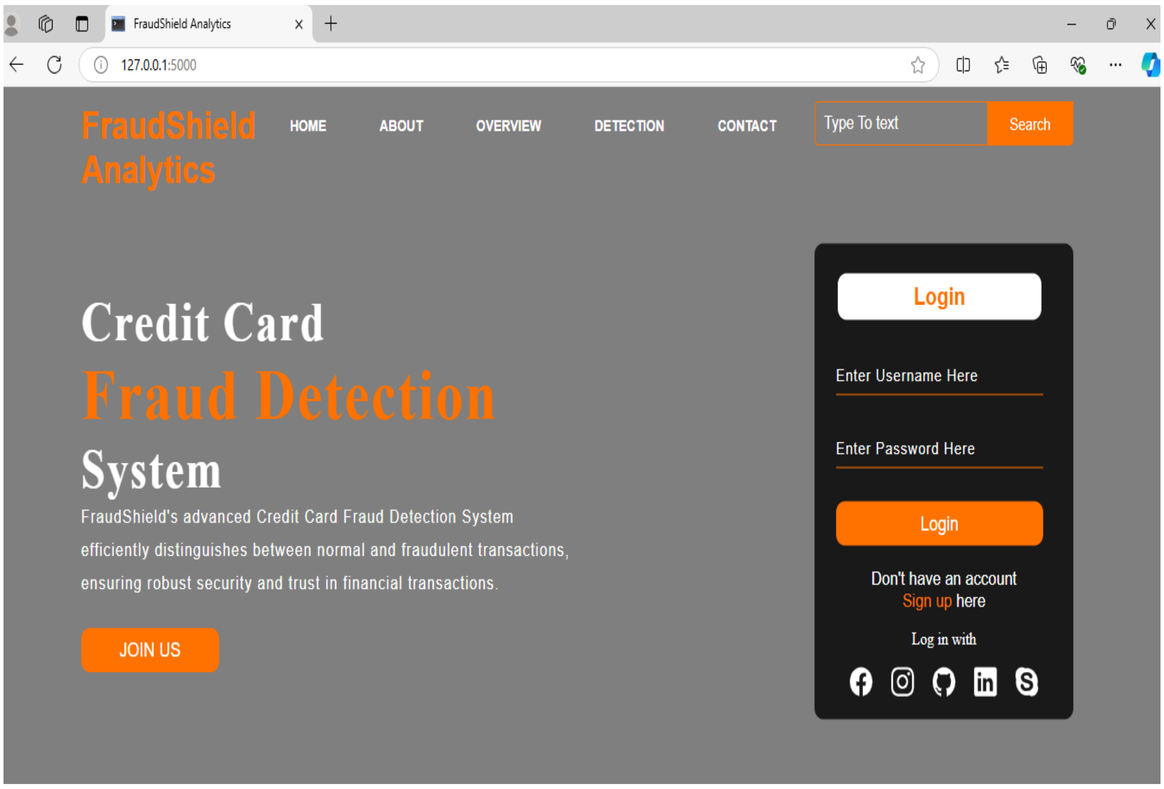
Issue 3: Prediction Results Not Displaying

- Ensure that the entered transaction details are in the correct format expected by the system (i.e., including relevant fields such as amount, transactions, etc.).
- Validate that the required dependencies (scikit-learn, pandas, etc.) are properly installed in your environment.
- Double-check the input values for any errors or inconsistencies that might prevent the prediction results from being displayed correctly.

7. Output

In this section, we showcase the output of the GUI-Based and Web-Based Interface for the Credit Card Fraud Detection System. Below are examples of what users can expect to see after entering transaction details and receiving predictions through both the GUI and web interfaces.

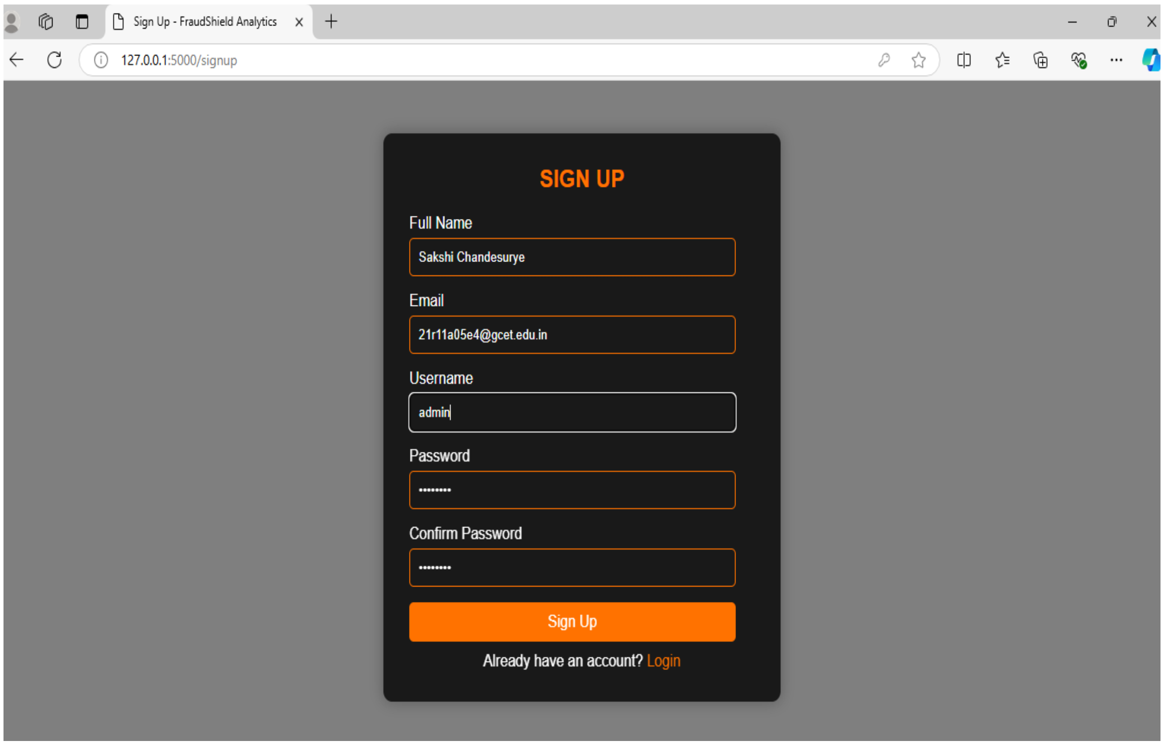
Web Interface



Credit Card
Fraud Detection
System

FraudShield's advanced Credit Card Fraud Detection System efficiently distinguishes between normal and fraudulent transactions, ensuring robust security and trust in financial transactions.

Fig 7.1 Home Page


Sign Up - FraudShield Analytics
127.0.0.1:5000/signup
SIGN UP
Sign Up
Already have an account? Login

Full Name: Sakshi Chandesurye

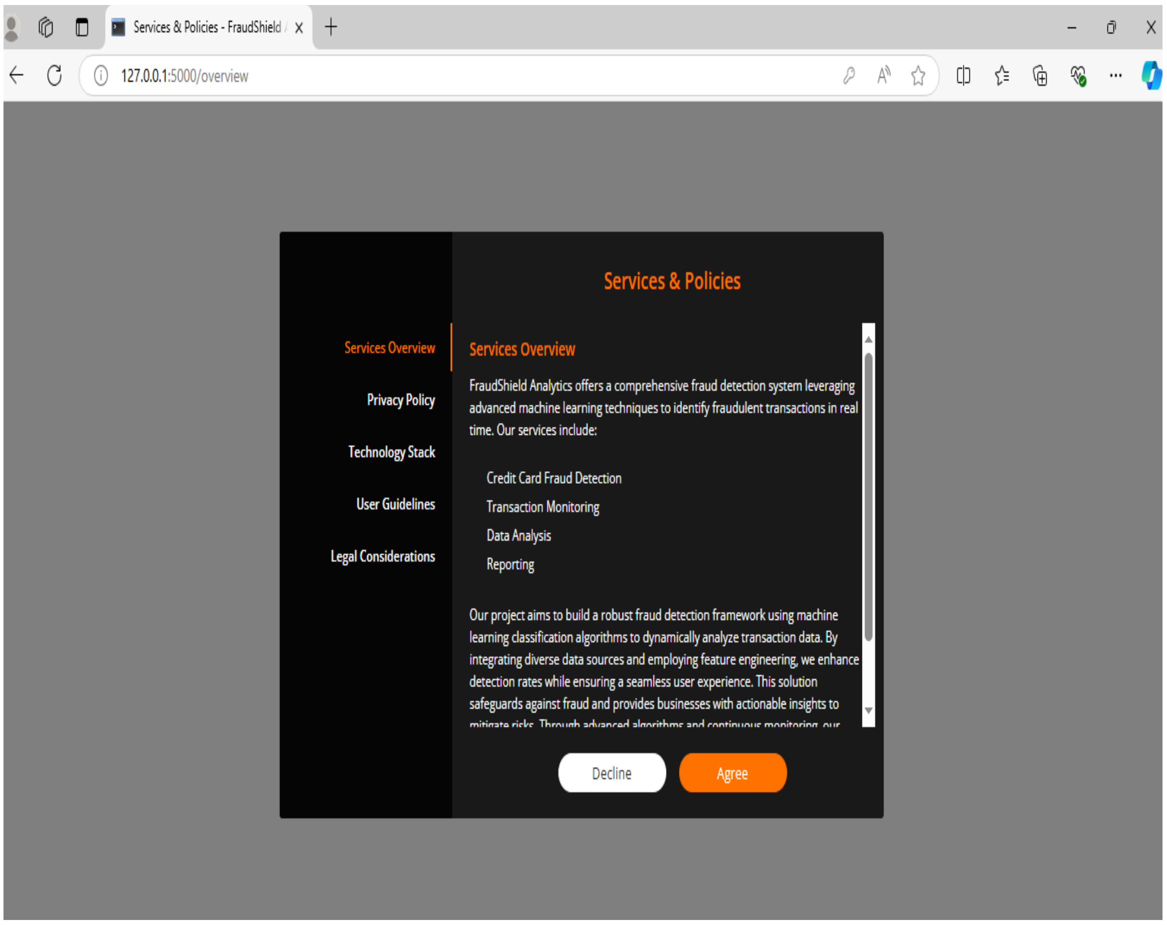
Email: 21r11a05e4@gcet.edu.in

Username: admin

Password: ••••••••

Confirm Password: ••••••••

Fig 7.2 Signup Page in progress



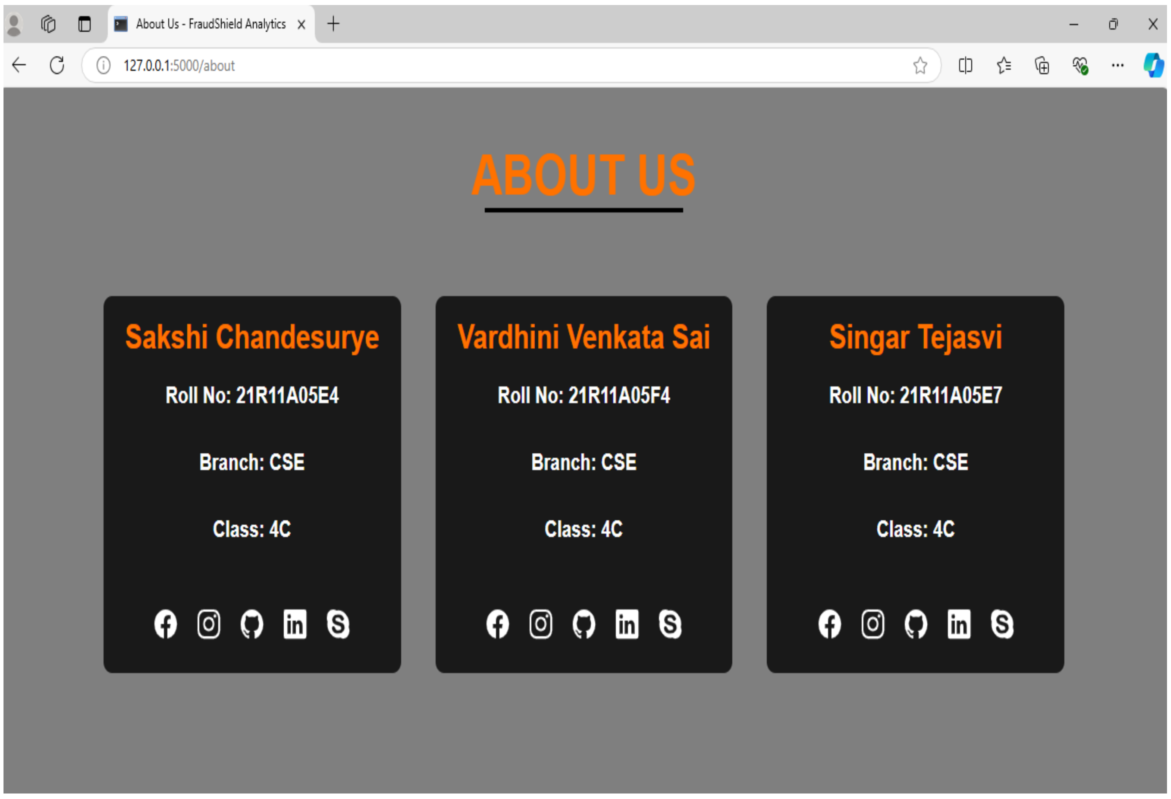
Services Overview

FraudShield Analytics offers a comprehensive fraud detection system leveraging advanced machine learning techniques to identify fraudulent transactions in real time. Our services include:

- Credit Card Fraud Detection
- Transaction Monitoring
- Data Analysis
- Reporting

Our project aims to build a robust fraud detection framework using machine learning classification algorithms to dynamically analyze transaction data. By integrating diverse data sources and employing feature engineering, we enhance detection rates while ensuring a seamless user experience. This solution safeguards against fraud and provides businesses with actionable insights to

Fig 7.3 Overview Page

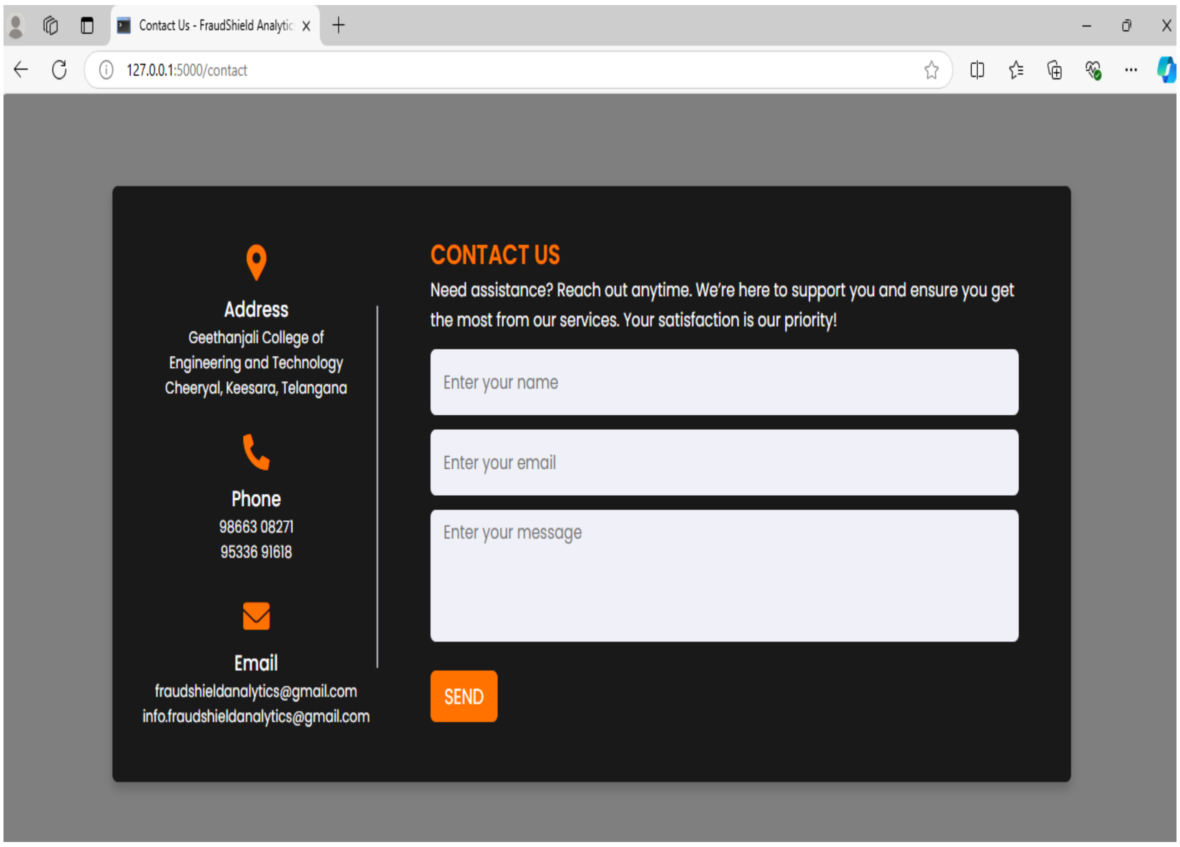


Sakshi Chandesurye
Roll No: 21R11A05E4
Branch: CSE
Class: 4C

Vardhini Venkata Sai
Roll No: 21R11A05F4
Branch: CSE
Class: 4C

Singar Tejasvi
Roll No: 21R11A05E7
Branch: CSE
Class: 4C

Fig 7.4 About Page



CONTACT US

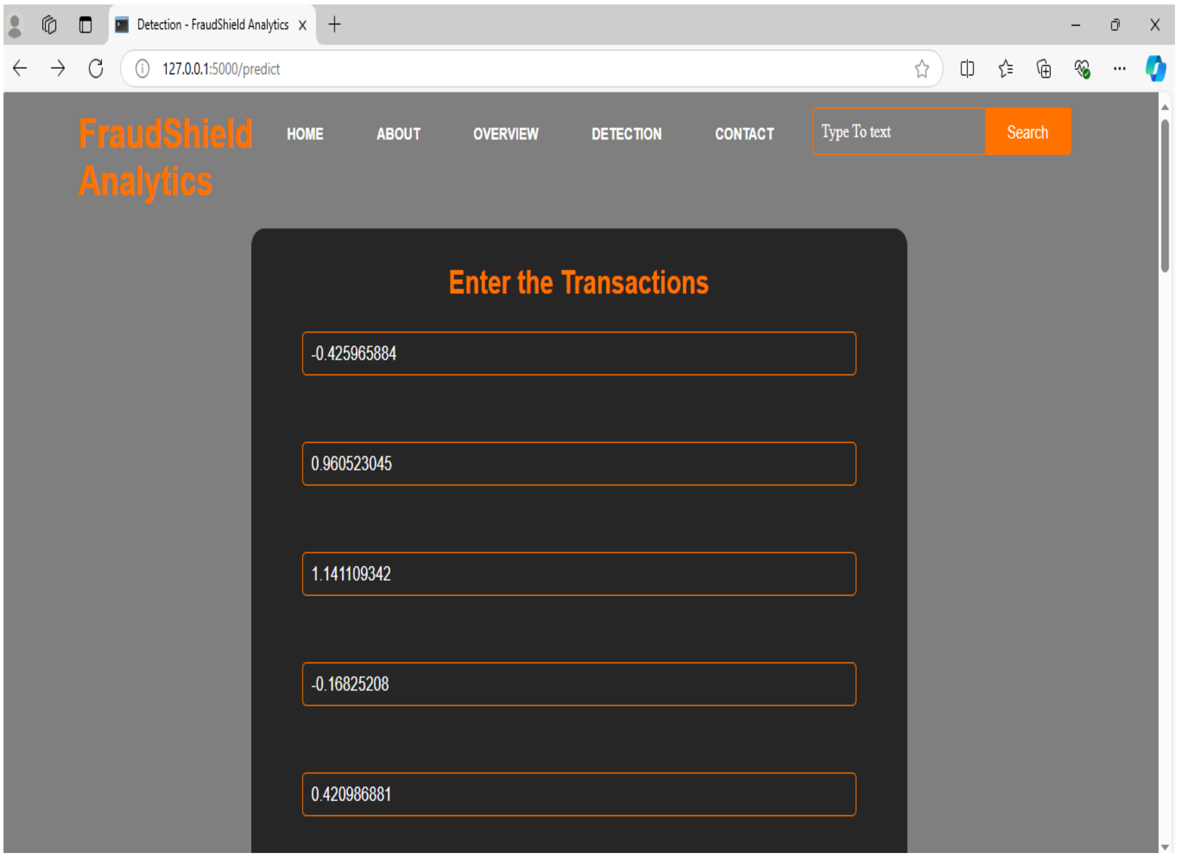
Need assistance? Reach out anytime. We're here to support you and ensure you get the most from our services. Your satisfaction is our priority!

Address
Geethanjali College of Engineering and Technology
Cheeryal, Keesara, Telangana

Phone
98663 08271
95336 91618

Email
fraudshieldanalytics@gmail.com
info.fraudshieldanalytics@gmail.com

Fig 7.5 Contact Page



Enter the Transactions

-0.425965884

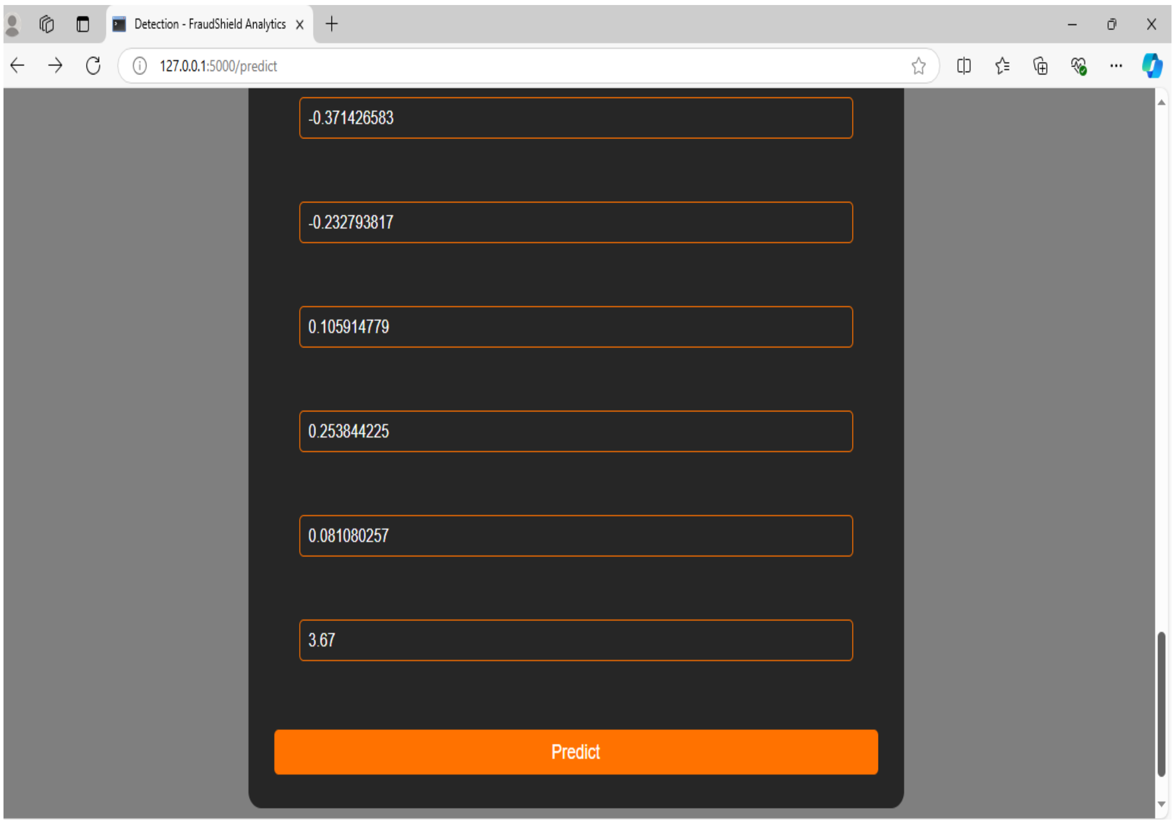
0.960523045

1.141109342

-0.16825208

0.420986881

Fig 7.6 Detection Page in Progress



-0.371426583

-0.232793817

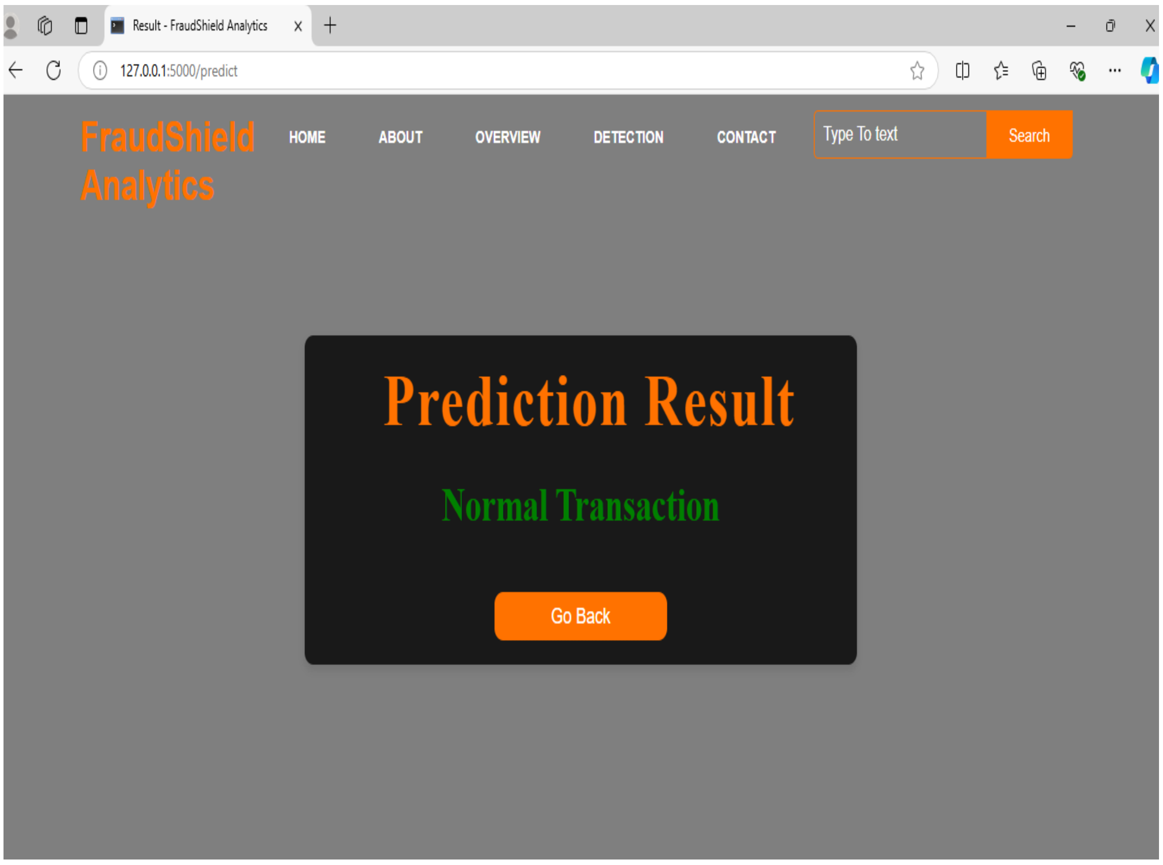
0.105914779

0.253844225

0.081080257

3.67

Fig 7.7 Continuation of the Detection Page in Progress

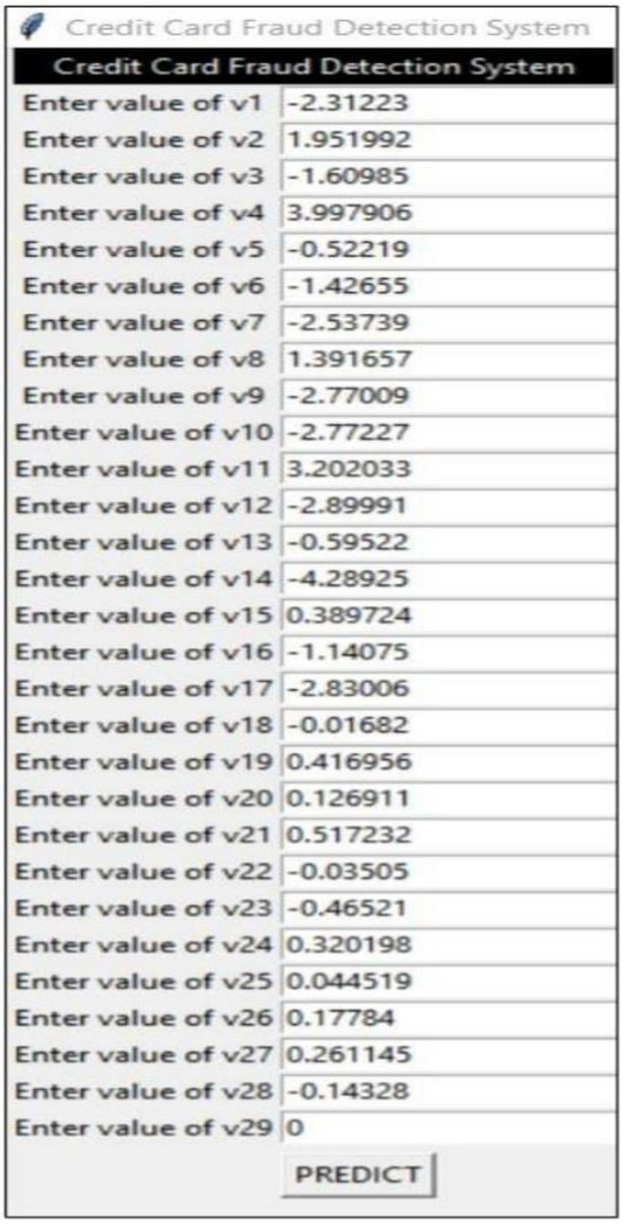


Prediction Result

Normal Transaction

Fig 7.8 Result Page

GUI Interface

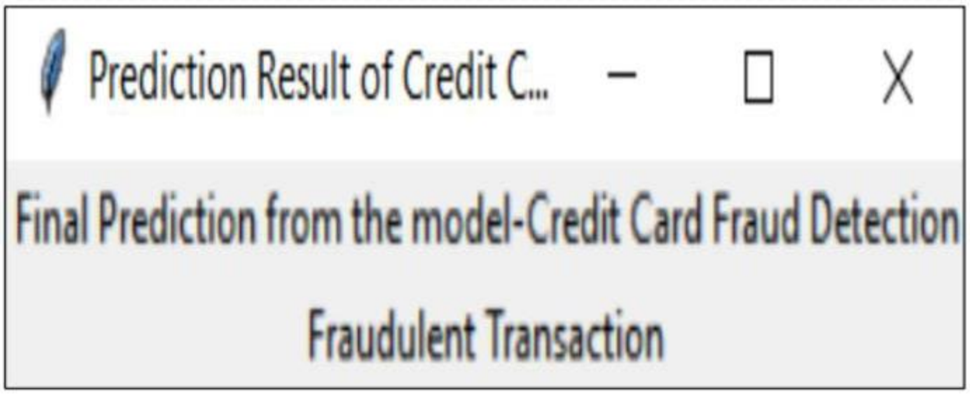


Credit Card Fraud Detection System

Credit Card Fraud Detection System

Enter value of v1	-2.31223
Enter value of v2	1.951992
Enter value of v3	-1.60985
Enter value of v4	3.997906
Enter value of v5	-0.52219
Enter value of v6	-1.42655
Enter value of v7	-2.53739
Enter value of v8	1.391657
Enter value of v9	-2.77009
Enter value of v10	-2.77227
Enter value of v11	3.202033
Enter value of v12	-2.89991
Enter value of v13	-0.59522
Enter value of v14	-4.28925
Enter value of v15	0.389724
Enter value of v16	-1.14075
Enter value of v17	-2.83006
Enter value of v18	-0.01682
Enter value of v19	0.416956
Enter value of v20	0.126911
Enter value of v21	0.517232
Enter value of v22	-0.03505
Enter value of v23	-0.46521
Enter value of v24	0.320198
Enter value of v25	0.044519
Enter value of v26	0.17784
Enter value of v27	0.261145
Enter value of v28	-0.14328
Enter value of v29	0

Fig 7.9 GUI Application of Credit Card Fraud Detection System



Prediction Result of Credit C...

Final Prediction from the model-Credit Card Fraud Detection

Fraudulent Transaction

Fig 7.10 Final Prediction of GUI Application

8. Conclusion

This user guide provides a comprehensive overview of the GUI-Based Web Interface for the Credit Card Fraud Detection System, outlining how to set up and run the application, navigate through its features, and troubleshoot common issues. By following these steps, users can easily detect fraudulent credit card transactions using machine learning models integrated into a user-friendly web interface.