Credit Card Fraud Detection

Fraud Detection

Better Safe Than Sorry

**A Project Report**

Submitted in partial fulfillment of the Requirements for the award of the Degree of

**BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY)**

**By**

Chaahna Chandiramani

UID - 19BIT006

**Under the esteemed guidance of Mr. Wilson Rao and Ms. Bertilla Fernandez**

**DEPARTMENT OF INFORMATION TECHNOLOGY JAI HIND COLLEGE**

***(Autonomous)* MUMBAI, 400020 MAHARASHTRA 2021-22**

**JAI HIND COLLEGE**

***(Autonomous)***

**MUMBAI, 400020 MAHARASHTRA**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



**CERTIFICATE**

This is to certify that the project entitled, “Credit Card Fraud Detection”, is bonaﬁed work of Chaahna Chandiramani bearing UID (19BIT006) submitted in partial fulﬁllment of the requirements for the award of degree of BACHELOR OF SCIENCE in INFORMATION TECHNOLOGY from Jai Hind College Autonomous (University of Mumbai).

**Internal Guide Coordinator**

**External Examiner**

**Date: College Seal**

**DECLARATION**

I hereby declare that the project entitled, “Credit Card Fraud Detection” done at Jai Hind College, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulﬁllment of the requirements for the award of degree of BACHELOR OF SCIENCE (INFORMATION TECHNOLOGY) to be submitted as ﬁnal semester project as part of our curriculum.

**Name and Signature of the Student**

**ACKNOWLEDGEMENT**

I am extremely grateful for the guidance of our Head of Department (Information Technology & Software Development) **Mr. Wilson Rao. Sir** had great involvement in making sure my project is a well-rounded and a flawless system by constantly guiding us till the completion of our project work by providing all the necessary information for developing a good system.

I would like to express immense gratitude to the people who have helped me throughout the course of my project. I am grateful to **Prof. Ms. Bertilla Fernandes** for her constant encouragement and support.

I would also like to thank all of my friends and my seniors who supported and helped me in completing the project, where they all had their own interesting takes on the technology stack, and the own interesting ideas on how to finesse the system even further. I would also like to thank my family for their constant support and encouragement.

**ABSTRACT**

As technology advances the users increase, with this malicious attack that seeks to unlawfully access data, disrupt digital operations or damage information are also increasing. Among this is credit card fraud which is a type of identity theft that occurs when someone that is not you uses your credit card or account information for an unauthorized charge.

This Credit card fraud detection website features uses user behavior to check for unusual patterns. If any unusual pattern is detected, the fraud is detected. The system analyses user credit card data for various characteristics. Based upon previous data of that user the system recognizes unusual patterns in the payment procedure.

The major contribution of this project include reduction in number of fraud transactions. The user can safely use his /her credit card for online transaction. An Additional layer of security will protect the companies from losing customers and money.

**TABLE OF CONTENTS**

1. **INTRODUCTION 9**

1.1 **Background**………………………………………………………………………9

1.2 **Objectives**………………………………………………………………………..10

1.3 **Purpose, Scope, and Applicability**……………………………………………..11

1.3.1 Purpose…………………………………………………………………….11

1.3.2 Scope………………………………………………………………………11

1.3.3 Applicability………….……………………………………………………11

1.4 **Achievements**……………………………………………………………………11

1.5 **Organization of Report**…………………………………………………………11

1. **SURVEY OF TECHNOLOGIES 12**
2. **REQUIREMENT AND ANALYSIS 14**
   1. **Problem Definition**……………………………………………………………...14
   2. **Requirement Specification**……………………………………………………..15
   3. **Planning and Scheduling**……………………………………………………….16
   4. **Software and Hardware Requirements**………………………………………..18
   5. **Preliminary Product Description**……………………………………………....19
   6. **Conceptual Models**……………………………………………………………...20
3. **SYSTEM DESIGN 32**
   1. **Basic Modules**……………………………………………………………...........32
   2. **Data Design**……………………………………………………………...............33
      1. Schema Design…………………………………………………………...33
      2. Data Integrity and Constraints……………………………………………34
   3. **User Interface Design**…………………………………………….......................35
   4. **Security Issues**……………………………………………..................................40
   5. **Test Cases Design**…………………………………………….............................41
4. **IMPLEMENTATION AND TESTING 42**
   1. **Implementation Approach**……………………………………………………..42
   2. **Coding Details and Code Efficiency**…………………………………………..43
      1. Code Efficiency…………………………………………..........................61
   3. **Testing Approach**………………………………………….................................62
      1. Unit Testing…………………………………………................................62
      2. Integration Testing………………………………………….....................62
      3. System Testing…………………………………………...........................62
   4. **Modifications and Improvements**……………………………………………...63
5. **RESULTS AND DISCUSSION 64**
   1. **User Documentation**…………………………………………………...64
6. **CONCLUSION 69**
   1. **Conclusion**…………………………………………………………….69
   2. **Limitations of the System**………………………………………………69
   3. **Future Scope of the Project**…………………………………………….69

**References**

**List of Figures**

1. Gantt Chart……………………………………………………………...16
2. PERT Chart……………………………………………………………..17
3. Event Table……………………………………………………………...20
4. ER Diagram…………………………………………………………......21
5. Class Diagram…………………………………………………………..21
6. Use Case Diagram………………………………………………………22
7. Object Diagram…………………………………………………………23
8. Activity Diagram………………………………………………………..24
9. Sequence Diagram………………………………………………………25
10. State Chart Diagram……..……………………………………………...26
11. Package Diagram………………………………………………………..27
12. Component Diagram……………………………………………………28
13. Deployment Diagram…………………………………………………...29
14. Data Flow Level 0 Diagram…………………………………………….30
15. Data Flow Level 1 Diagram…………………………………………….31
16. Registration Page..……………………………………………………....35
17. Login Page……………………………………………………………....36
18. Data Entry Page…………..……………………………………………..37
19. Results Page…….……….………………………………………………38
20. About Page..……….………………………………………………….....39
21. Registration Page..……………………………………………………....64
22. Login Page……………………………………………………………....65
23. Data Entry Page…………..……………………………………………..66
24. Results Page…….……….………………………………………………67
25. About Page..……….………………………………………………….....68
26. **Introduction**

This is a credit card fraud detection website. The user must enter the amount and then model will classify the transaction as fraud or authentic.

The detection system is designed with a combination of a Random Forest model, Logistic regression model and Naïve Bayes model. This project can help detect fraudulent transactions in turn reducing fraud.

1. **Background**

From 2019 to 2020, the number of identity theft reports went up by 113% and the number of reports of identity theft by credit cards increased by 44.6%.

Credit card fraud accounted for 393,207 of the nearly 1.4 million reports of identity theft in 2020. This makes credit card fraud the second most common type of identity theft reported.

Credit Card fraud detection will use various data prediction methods which will have prevent such crimes, in turn keep people and their money safe.

1. **Objectives**

Objective of the project is to predict the fraud and fraud less transaction with respect to time and the amount of transaction using machine learning algorithms, statistics and calculus (diﬀerentiation, chain rule etc.) and linear algebra in building of the complex machine learning models for prediction and understanding of the data set.

1. **Purpose, Scope and Applicability**
   * 1. **Purpose**

As technology advances the users increase, with this malicious attack that seeks to unlawfully access data, disrupt digital operations or damage information are also increasing.

Among this is credit card fraud which is a type of identity theft that occurs when someone that is not you uses your credit card or account information for an unauthorized charge. Using this software users will be able to detect fraudulent activities and in turn stop it.

* + 1. **Scope**
* Credit Card Owners
* Credit Card Companies/Banks
* Other vendors can use our services to detect fraudulent transaction
  + 1. **Applicability**
* The focus is to identify if a transaction comes under normal class or fraudulent class using predictive models.
* Different sampling techniques will be implemented to tackle the class imbalance problem and series of machine learning algorithms like logistic regression, random forest etc will be implemented on the dataset, and the results will be reported.
* Users will be able to use this software to detect fraudulent transactions and in turn prevent them.

1. **Achievements**

We achieve a cost-effective, labour-saving and streamlined method of booking seats.

The project objectives are met successfully as it meets the needs.

* 1. **Organization of Report**

In the chapters that follows the hardware requirements, software requirements, system analysis, future enhancements, methodology, software languages and technologies, schema design, high level Architecture diagrams, testing methods and project contributions will be discussed.

1. **Survey of Technologies**

**Backend**

* **Python**
* Version -3.9
* Python is a dynamic (as opposed to static) language and reduces complexity when it comes to collaborating meaning you can implement functionality with less code. Can be platform specific, Python code can be run on pretty much any platform without wasting time on specific configurations.
* Has various necessary libraries like TensorFlow, Numpy, Skilearn, Pandas, Keras, Theano, PyTorch etc which are required.
* Python, has more edge and is more suitable for AI. With its simple syntax and readability promoting the rapid testing of complex machine learning algorithms and a thriving community bolstered by collaborative tools like Jupyter Notebooks and Google Colab.
* **Python Flask**
* Version - 2.0.2
* Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.
* **Python Flask**
* Version - 2.0.2
* SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine.
* The code for SQLite is in the public domain and is thus free for use for any purpose, commercial or private. SQLite is the most widely deployed database in the world with more applications than we can count, including several high-profile projects.
* SQLite is an embedded SQL database engine.
* Unlike most other SQL databases, SQLite does not have a separate server process. SQLite reads and writes directly to ordinary disk files. A complete SQL database with multiple tables, indices, triggers, and views, is contained in a single disk file.
* The database file format is cross-platform - you can freely copy a database between 32-bit and 64-bit systems or between big-endian and little-endian architectures.
* These features make SQLite a popular choice as an Application File Format.

**Frontend**

* **CSS (Cascading Style Sheets):**
* CSS is a computer language for laying out and structuring web pages (HTML or XML).
* This language contains coding elements and is composed of these “cascading style sheets” which are equally called CSS files
* CSS is the language for describing the presentation of Web pages, including colours, layout, and fonts.
* It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers.
* CSS is independent of HTML and can be used with any XML-based markup language.
* **Bootstrap:**
* Bootstrap is a front-end development framework.
* It is a free and open-source framework.
* It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.
* It is used to create modern websites and web apps.
* **HTML**
* HTML (stands for Hypertext Markup Language) is a computer language that makes up most web pages and online applications.
* A hypertext is a text that is used to reference other pieces of text, while a markup language is a series of markings that tells web servers the style and structure of a document.
* HTML is the combination of Hypertext and Markup language.
* Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages.
* This language is used to annotate (make notes for the computer) text so that a machine can understand it and manipulate text accordingly. Most markup languages (e.g. HTML) are human-readable.
* The language uses tags to define what manipulation has to be done on the text.

1. **REQUIREMENTS AND ANALYSIS**
   1. **Problem Definition**

* **Problem Statement**
* As technology advances the users increase, with this malicious attack that seeks to unlawfully access data, disrupt digital operations or damage information are also increasing.
* Among this is credit card fraud which is a type of identity theft that occurs when someone that is not you uses your credit card or account information for an unauthorized charge.
* **Existing System**
* In case of the existing system the fraud is detected after the fraud is done that is, the fraud is detected after the complaint of the card holder.
* In the existing system discriminate analysis and regression analysis are widely used which can detect fraud by credit rate for cardholder’s and credit card transaction.
* **Proposed System**
* In this proposed system we are doing the analysis of different techniques of fraud detection which is user friendly and secure.
* This system analyzes the feasibility of credit card fraud detection based on outlier mining, applies outlier detection mining based on distance sum into credit card fraud detection and proposes this detection procedures and its empirical process.
  1. **Requirement Specification**
* **Functional Requirements**
* The data cleaning should be done accurately.
* There shouldn’t be any missing values in the dataset.
* All the models should function seamlessly.
* Detection of fraud should occur prior to any damage being done.
* Accuracy of the calculations made should be high.
* High volumes of data can be stored with case.
* Maintenance of file is efficient and flexible.
* Manpower is reduced.
* **Non-Functional Requirements**

1. Reliability

The system is must be reliable and should not have frequent failures.

1. Availability

This system must be made available to all types of users with respective user controls.

1. Usability

The system must maintain usability requirements in such a manner that the interface is user friendly and user oriented.

1. Maintainability

The system must be easy to maintain in such a manner that any glitch or bug can be conveniently dealt with as well as frequent updates should not become tedious.

* 1. **Planning and Scheduling**

**• Gantt Chart**

****

Expected

Actual

**Figure 1**: GANNT Chart

* **PERT Chart**



**Figure 2**: PERT Chart

* 1. **Software and Hardware Requirements**

• **Hardware Requirements**

* Processor – i5
* Hard Disk – 5 GB
* Memory – 4GB RAM

**• Software Requirements**

- Operating system – windows 7 or greater

- Python

- SQLite

- HTML

- CSS

- Bootstrap

- Flask

* 1. **Preliminary Product Description**

1. **Registration**

User will have to create an account to access the facilities of the website.

1. **Login**

Once the End user is a registered customer then s/he can login to the system with his/her credentials as per their convenience and access the features of the website.

1. **Exploratory Data Analysis (ML)**

The data is investigated so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions. Along with this Principal component analysis will be done to simplify the complexity of the high-dimensional data while retaining trends and patterns. transforming the data into fewer dimensions, which act as summaries of features. This data is later feed to the system as training data.

1. **Random Forest Model (ML)**

For the classification problem statement, the output is the maximum occurrence outputs from each decision tree models inside the random forest.

1. **Logical Regression Model (ML)**

This algorithm works well with less amount of data set because of the use of sigmoid function if value the of sigmoid function is greater than 0.5 the output will 1 if the output the sigmoid function is less than 0.5 then the output is considered as the 0.

1. **Naïve Bayes Model (ML)**

Here, Naïve Bayes algorithm will be used for the classification problem, which work on the property of Bayes theorem. It can be implemented by using features in data set independent feature as input and dependent feature as a output.

1. **About Page**

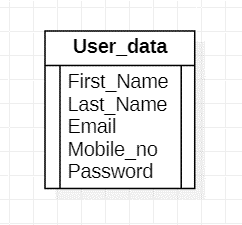
Describing various details about the website, functions available and about the developers.

* 1. **Conceptual Models**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S. No.** | **Event** | **Trigger** | **Source** | **Use Case** | **Response** | **Destination** |
| 1 | User Registration | Register data | Seller/Customer | Registration | Through login page | Seller/Customer |
| 2 | Login | User Id and Password | Admin/Seller/  Customer | Login | Home page | Admin/Seller/  Customer |
| 3 | Input | Button | Dataset | Access Data | Get Data | Program |
| 4. | Analyze Data | Button | Website | Input Analyzation | Fraud Detection | Website |

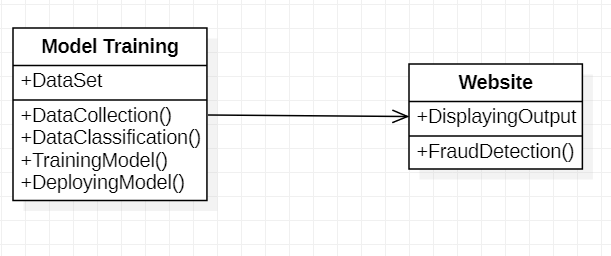
**Figure 3** Event Table

1. **Entity-Relationship Diagram**

****

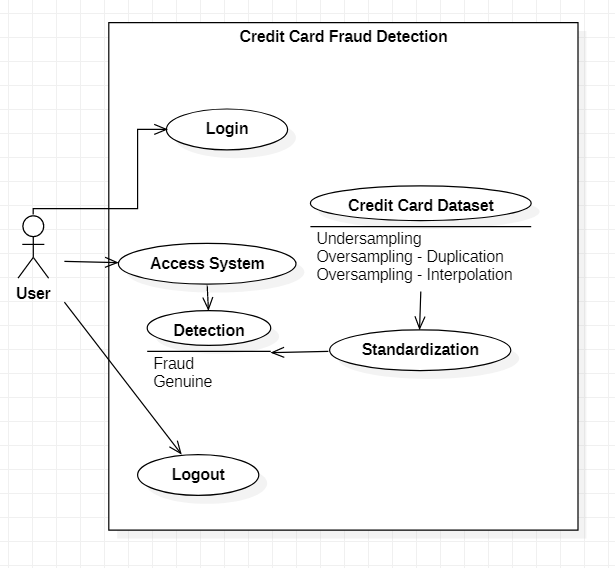
**Figure 4:** E-R Diagram

1. **Class Diagram**



**Figure 5:** Class Diagram

1. **Use Case Diagram**

****

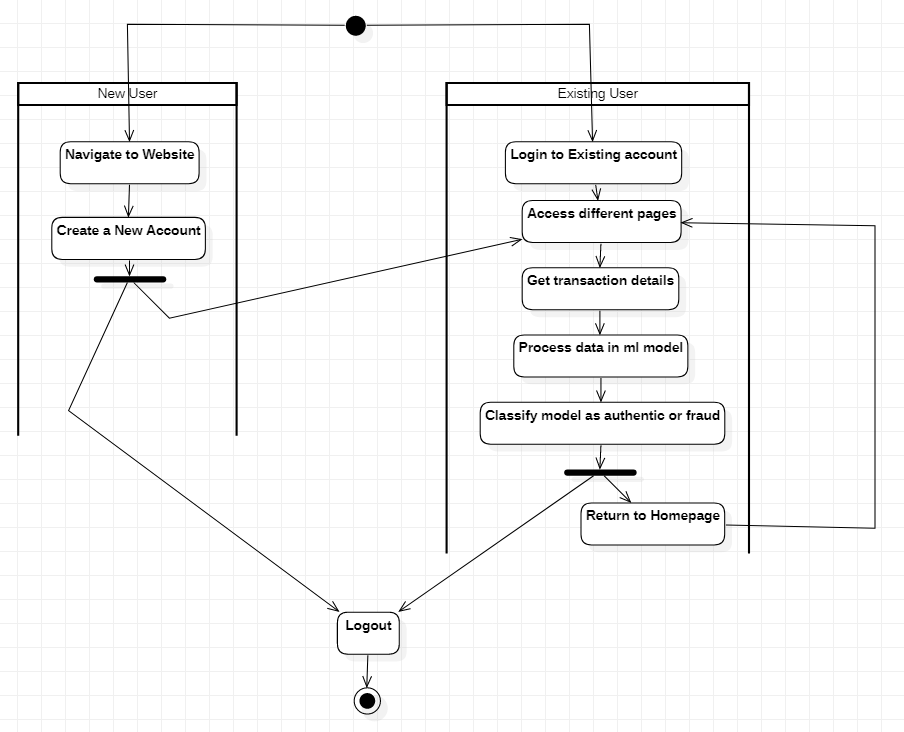
**Figure 6:** Use Case Diagram

1. **Object Diagram**



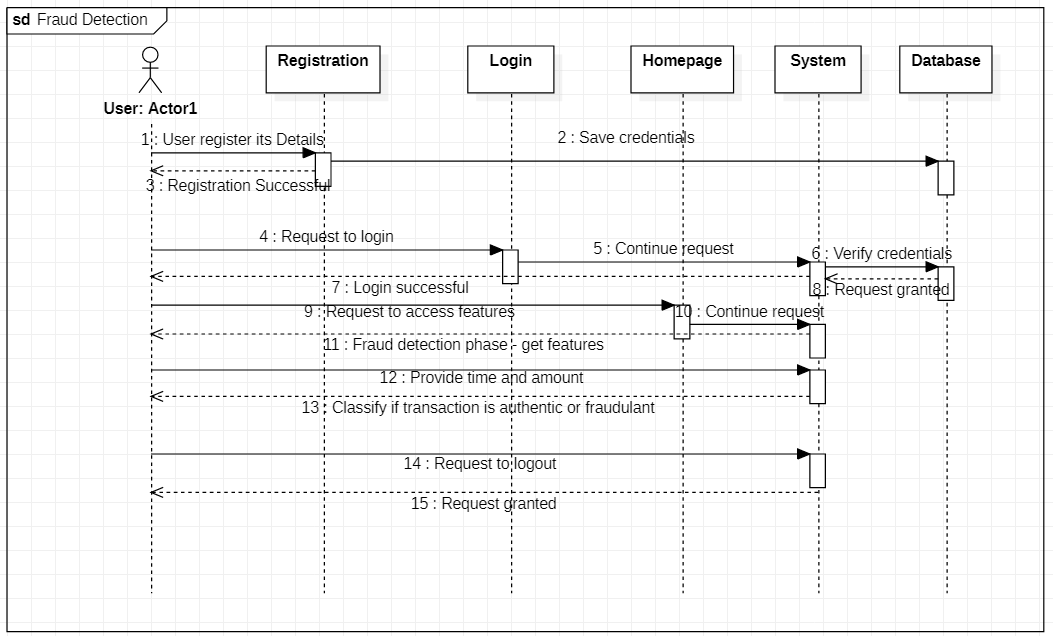
**Figure 7:** Object Diagram

1. **Activity Diagram**

****

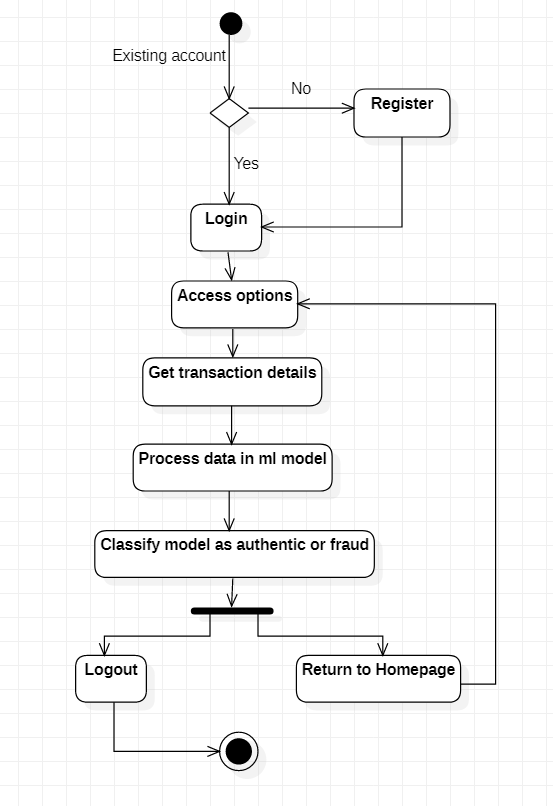
**Figure 8:** Activity Diagram

1. **Sequence Diagram**

****

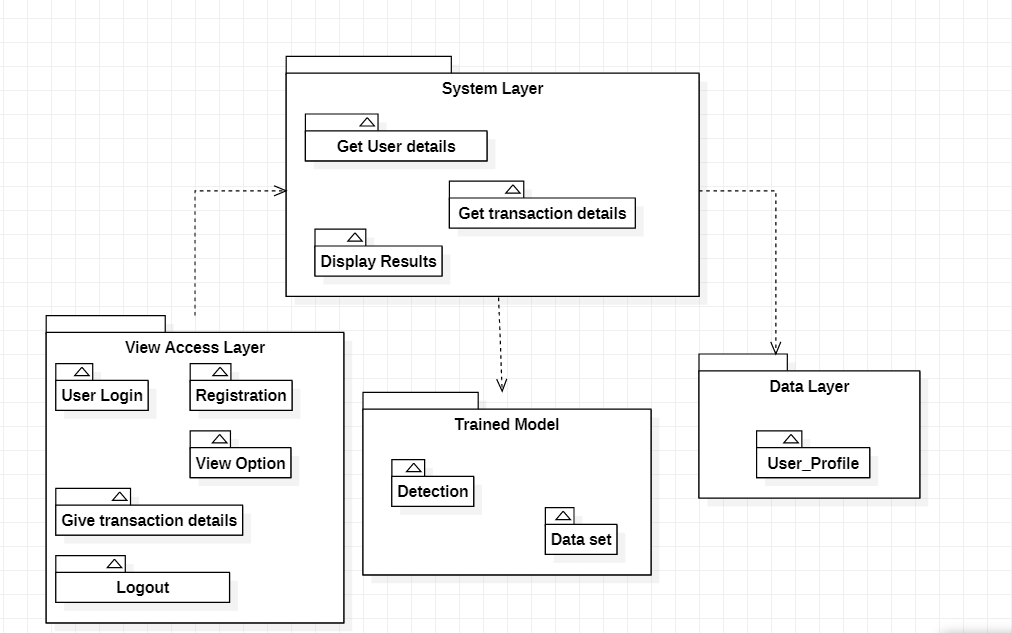
**Figure 9:** Sequence Diagram

1. **State Diagram**

**  
Figure 10:** State Chart Diagram

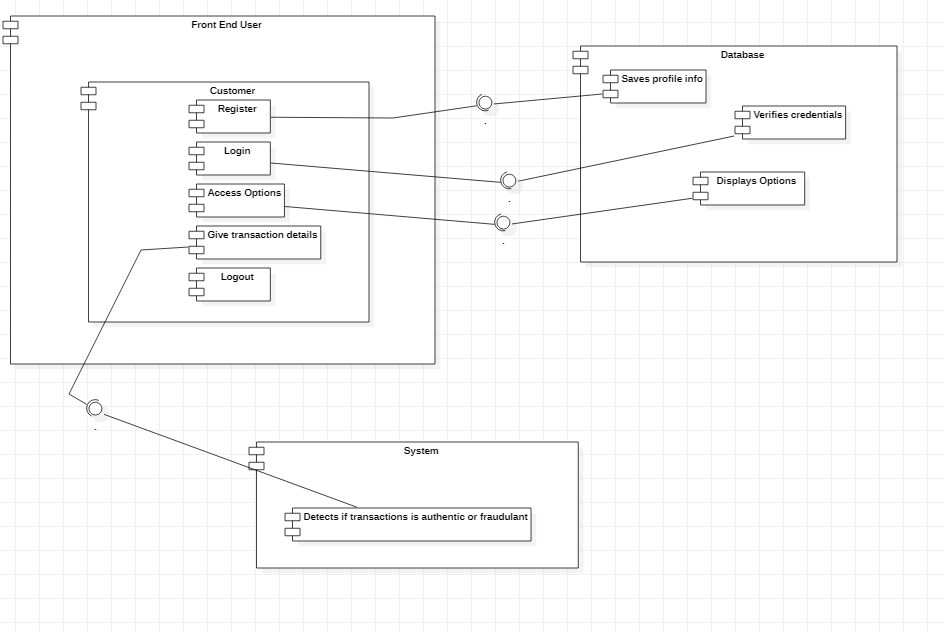
0

1. **Package diagram**



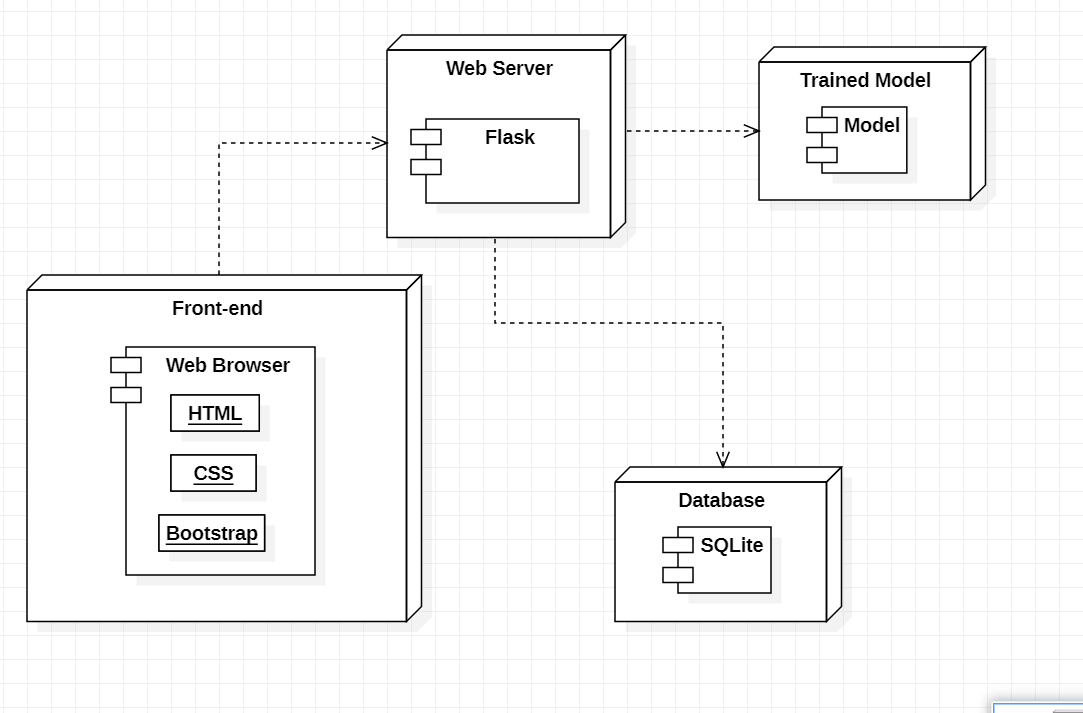
**Figure 11:** Package Diagram

1. **Component Diagram**

****

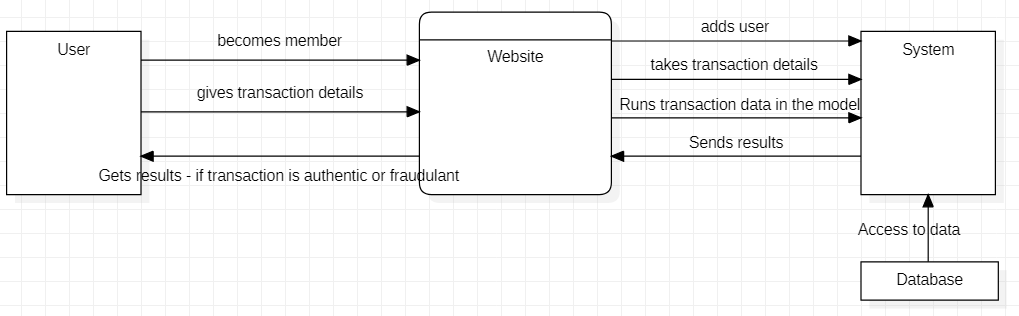
**Figure 12:** Component Diagram

1. **Deployment Diagram**



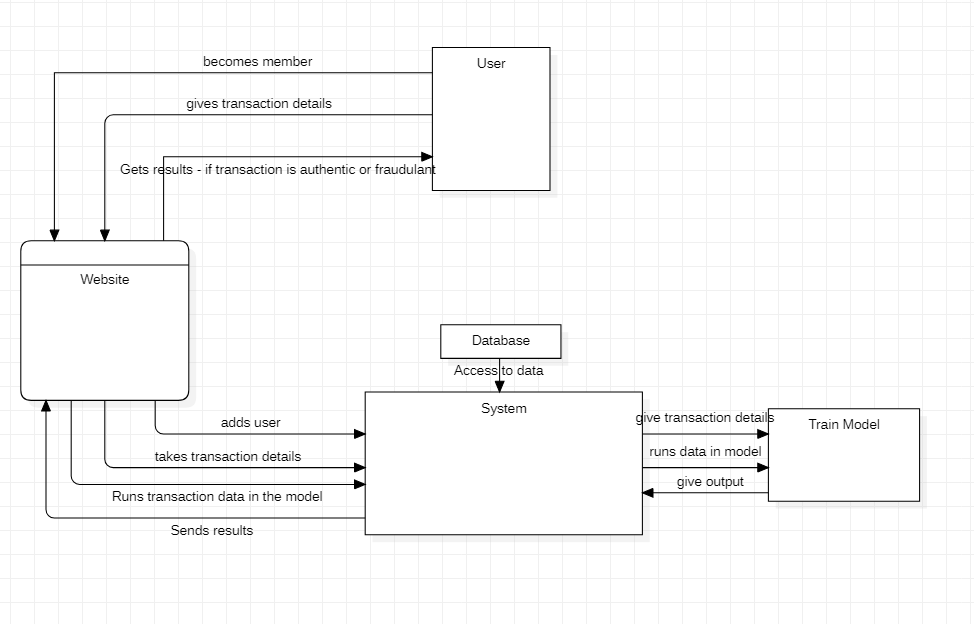
**Figure 13:** Deployment Diagram

1. **Data Flow Diagram Level 0**

****

**Figure 14:** Data Flow Diagram (Context Level)

1. **Data Flow Diagram Level 1**

****

**Figure 15:** Data Flow Diagram (Level 1)

1. **SYSTEM DESIGN**

**4.1 Basic Modules**

1. **Login /Register:**

The admin will be able to login to the system. S/he will have specific rights.

The Seller has to register after which on using the website later S/he can login.

The Customer also has to register and make an account to do any bookings.

After registration S/he can Login.

1. **Exploratory Data Analysis (ML)**

The data will be investigated so as to discover patterns, to spot anomalies, to test hypothesis and to check assumptions.

Along with this Principal component analysis will be done to simplify the complexity of the high-dimensional data while retaining trends and patterns. transforming the data into fewer dimensions, which act as summaries of features.

1. **Random Forest Model (ML)**

This will randomly select the features that are independent variables and also randomly selects the rows by row sampling and the number of decision tree can be determined by using hyper parameter optimization. For the classification problem statement, the output is the maximum occurrence outputs from each decision tree models inside the random forest.

1. **Logical Regression Model (ML)**

Logistic regression works with sigmoid function because the sigmoid function can be used to classify the output that is a dependent feature and it uses the probability for classification of the dependent feature. This algorithm works well with less amount of data set because of the use of sigmoid function if value the of sigmoid function is greater than 0.5 the output will 1 if the output the sigmoid function is less than 0.5 then the output is considered as the 0.

1. **Naïve Bayes Model (ML)**

Here, Naïve Bayes algorithm will be used for the classification problem, which work on the property of Bayes theorem. It can be implemented by using features in data set independent feature as input and dependent feature as a output.

1. **About Page:**

Describing various details about the website, functions available and about the developers.

**4.2 Data Design**

**4.2.1 Schema Design**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User\_Data** | | | | |
| **Sr. No.** | **Field** | **Data Type** | **Null** | **Constraint** |
| 1 | Username | varchar(30) | no | Primary Key |
| 2 | Last\_Name | varchar(30) | no |  |
| 3 | Email | varchar(50) | no | Unique |
| 4 | Phone | int(10) | no |  |
| 5 | password | varchar(10) | no |  |

**4.2.2 Data Integrity and Constraints**

* **Auto Increment Constraint**

Auto-increment allows a unique number to be generated automatically when a new record is inserted into a table. Often this is the primary key field that we would like to be created automatically every time a new record is inserted.

* **Primary Key Constraint**

In a database, a table can have only one primary key which cannot have a NULL value. It always represents a clustered index in a database table and helps to organize the sequence of clustered index.

Primary key can be related to another tables as a Foreign Key and you can apply Auto Increment value for a Primary key but auto increment is not mandatory.

* **Foreign key Constraint**

A foreign key is just a referential constraint between two tables. If a Primary key is used in another table, then it would be known in another table as Foreign key which can accept multiple null value. Foreign key always generates after Primary key. Thus, they do not automatically increment. It supports clustered or non-clustered indexes and you can have more than one foreign key in a table.

* **Check Constraint**

Adding a CHECK CONSTRAINT on a column of a table, you can limit the range of values allowed to be stored in that column.

* **Not NULL Constraint**

By default, the columns are able to hold NULL values.

A NOT NULL constraint in SQL is used to prevent inserting NULL values into the specified column, considering it as a not accepted value for that column.

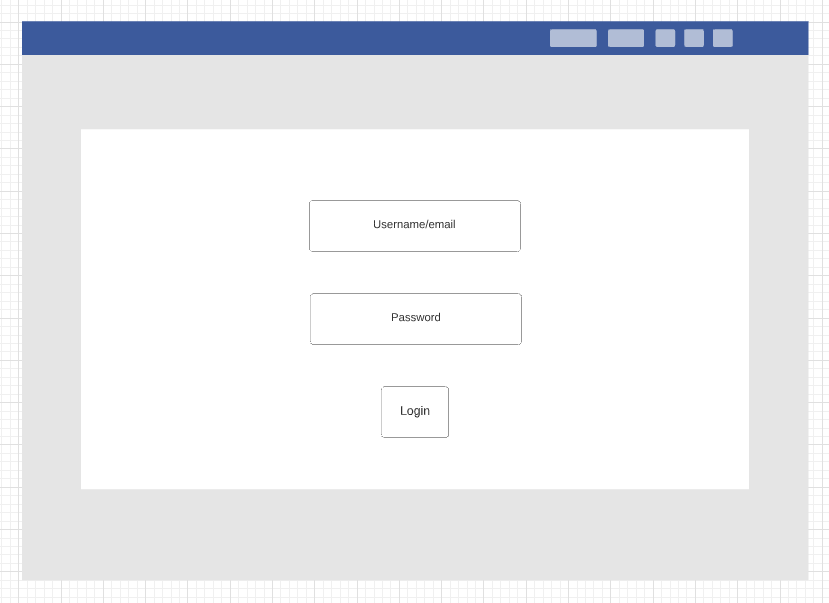
* **Uniqueness Constraint**

The UNIQUE constraint in SQL is used to ensure that no duplicate values will be inserted into a specific column or combination of columns that are participating in the UNIQUE constraint and not part of the PRIMARY KEY. In other words, the index that is automatically created when you define a UNIQUE constraint will guarantee that no two rows in that table can have the same value for the columns participating in that index, with the ability to insert only one unique NULL value to these columns, if the column allows NULL.

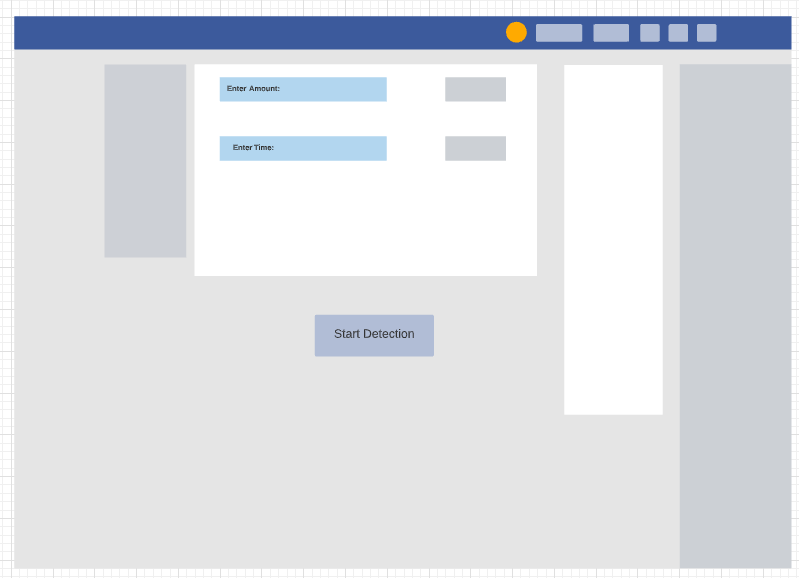
**4.3 User Interface Layouts**



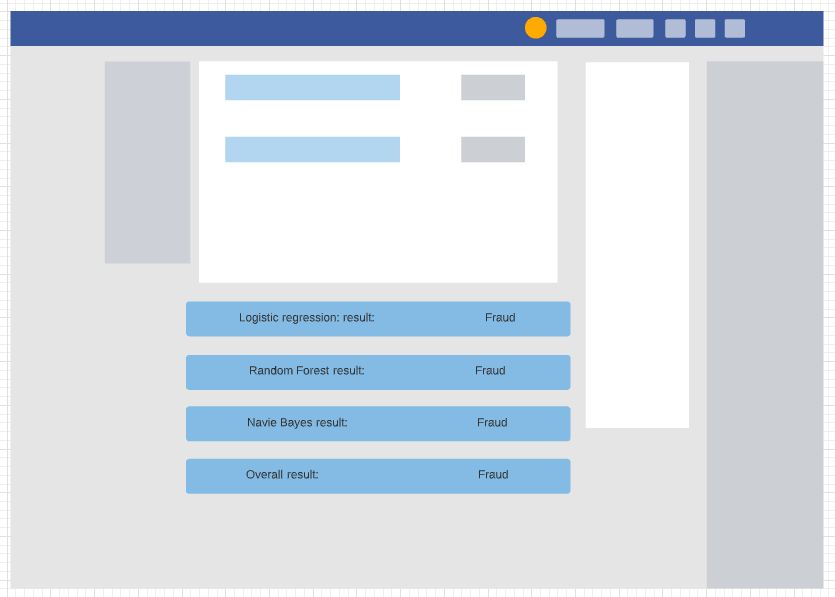
**Figure 16:** Registration Page



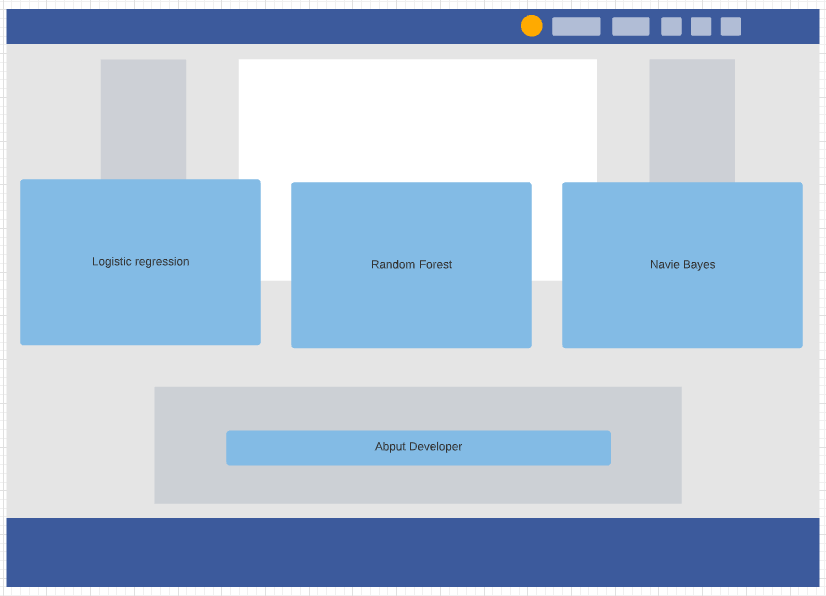
**Figure 17:** Login Page



**Figure 18:** Data Entry Page



**Figure 19:** Result Page



**Figure 20:** About Page

**4.4 Security Issues**

* **Password Encryption**

Password encryption is the translation of login data into a secret code. The industry-standard encryption is the most effective way to achieve data security because to read an encrypted password requires decryption.

* **Authorization**

Authorization is the function of specifying access rights/privileges to resources, which is related to general information security and computer security, and to access control in particular. More formally, "to authorize".

* **Authentication**

Authentication is the act of proving an assertion, such as the identity of a computer system user. In contrast with identification, the act of indicating a person or thing's identity, authentication is the process of verifying that identity. Here, will have to enter his/her email address which s/he used at the time of registration to authenticate her/himself. This email address will be checked and authenticated.

* **DOS attack prevention**

The objective of a DDoS attack is to prevent legitimate users from accessing your website. For a DDoS attack to be successful, the attacker needs to send more requests than the victim server can handle. Validation of users is done before hand to prevent such attacks.

**4.5 Test Case Design**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Condition** | **Input Selection** | **Expected** | **Actual** |
| User Login | Email="" | Email and Password fields are empty | Email and Password fields are empty |
| Password="" |
| Email="abc@gmail.com" | Verify and redirect | Verify and redirect |
| Password="abc" |
| Detect | Time="Current time" | Fraud Detected | Fraud Detected |
| Amount="22.04" |
| Time="Current time" | Not | Not |
| Amount="22.04" | Fraud | Fraud |

**Figure 24:** Test Case Design

**5. IMPLEMENTATION AND TESTING**

**5.1 Implementation Approach**

* + **Introduction**

To access the website the user must login. Before logging in the user should be registered with the website.

* + **Input output design implementation**

Firstly, the website functionalities have be segregated into various modules. These modules depend on the user and her/his accessibilities. This will be easier to understand with the help of the database design. The database has various tables that have been created specifically to cater to the needs of the users. The database works in such a manner that all the tables are linked. The admin can access the data from the tables.

* + **Code Module**

This Activity demonstrates the technologies used for the Front-End and Back-End and which Database. Python with Flask has been used along with HTML, CSS and bootstrap for the front end. SQLite has been used as the database.

* + **System Implementation**

The user must first create an account on the website, only then s/he will be able to login. Once logged in the user must provide specific details about the transaction. The model will take these details and run it model with it to classify the transaction. Once the detection is done the system will classify the transaction as authentic or fraud. This can later help a variety of people to solve credit card fraud cases.

* + **Project Summary**

As technology advances the users increase, with this malicious attack that seeks to unlawfully access data, disrupt digital operations or damage information are also increasing. Among this is credit card fraud which is a type of identity theft that occurs when someone that is not you uses your credit card or account information for an unauthorized charge. Using this software users will be able to detect fraudulent activities and in turn stop it.

**5.2 Code Details and** **Code Efficiency**

**Exploratory Data Analysis (ML)**

import numpy as np

import pandas as pd

# Metrics

from imblearn.metrics import classification\_report\_imbalanced

from sklearn.metrics import precision\_score, recall\_score, f1\_score, roc\_auc\_score, accuracy\_score, classification\_report

import warnings

warnings.filterwarnings("ignore")

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv('creditcard.csv')

print(data.head())

print(data.info())

print(data.describe())

**Random Forest Model (ML)**

data = pd.read\_csv(csvFilePath)

labels = {0: "Not Fraud", 1: "Fraud"}

def preprocessing(data):

X = data.drop('Class', axis=1)

y = data['Class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

robust\_scaler = RobustScaler().fit(X\_train)

X\_test = pd.DataFrame(robust\_scaler.transform(X\_test), columns=X.columns)

return X\_test, y\_test

X\_test, y\_test = preprocessing(data)

rf\_model = joblib.load(randomForestModelFilePath)

input\_tuple = X\_test.sample(1) # Select a random row from X\_test

rf\_prediction = labels[rf\_model.predict(input\_tuple)[0]]

y\_true = labels[y\_test.iloc[input\_tuple.index[0]]]

new\_input\_tuple = input\_tuple.to\_dict('records')[0] # Convert DataFrame to format => [{col\_name\_1: value\_1, col\_name\_2: value\_2, ...}, {second\_row}]

# Converting Amount & Time values to the ones before preprocessing so that it can be displayed on webpage

new\_input\_tuple['Amount'] = data.iloc[input\_tuple.index[0]]['Amount']

new\_input\_tuple['Time'] = data.iloc[input\_tuple.index[0]]['Time']

for col in new\_input\_tuple: # Round off each value to 4 decimal values so that it looks good on webpage

new\_input\_tuple[col] = round(new\_input\_tuple[col], 4)

print("The transcation is {0}, Transcation Amount: {1} EUR.".format(rf\_prediction,new\_input\_tuple['Amount']))

**Logical Regression Model (ML)**

data = pd.read\_csv(csvFilePath)

labels = {0: "Not Fraud", 1: "Fraud"}

def preprocessing(data):

X = data.drop('Class', axis=1)

y = data['Class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

robust\_scaler = RobustScaler().fit(X\_train)

X\_test = pd.DataFrame(robust\_scaler.transform(X\_test), columns=X.columns)

return X\_test, y\_test

X\_test, y\_test = preprocessing(data)

lr\_model = joblib.load(logisticRegressionModelFilePath)

input\_tuple = X\_test.sample(1) # Select a random row from X\_test

lr\_prediction = labels[lr\_model.predict(input\_tuple)[0]]

y\_true = labels[y\_test.iloc[input\_tuple.index[0]]]

new\_input\_tuple = input\_tuple.to\_dict('records')[0] # Convert DataFrame to format => [{col\_name\_1: value\_1, col\_name\_2: value\_2, ...}, {second\_row}]

# Converting Amount & Time values to the ones before preprocessing so that it can be displayed on webpage

new\_input\_tuple['Amount'] = data.iloc[input\_tuple.index[0]]['Amount']

new\_input\_tuple['Time'] = data.iloc[input\_tuple.index[0]]['Time']

for col in new\_input\_tuple: # Round off each value to 4 decimal values so that it looks good on webpage

new\_input\_tuple[col] = round(new\_input\_tuple[col], 4)

print("The transcation is {0}, Transcation Amount: {1} EUR.".format(lr\_prediction,new\_input\_tuple['Amount']))

**app.py**

from flask import \*

from flask import Flask, render\_template, url\_for

import sqlite3, hashlib, os

import joblib

import pandas as pd

import numpy as np

import datetime

from werkzeug.utils import secure\_filename

from pprint import pprint

from sklearn.preprocessing import RobustScaler

from sklearn.model\_selection import train\_test\_split

app = Flask(\_\_name\_\_)

app.secret\_key = 'random string'

UPLOAD\_FOLDER = 'static/uploads'

ALLOWED\_EXTENSIONS = set(['jpeg', 'jpg', 'png', 'gif'])

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

@app.route("/")

def root():

if 'email' in session:

return render\_template\_string("<h1>You are logged in</h1> </h1><a href=\"/index\">Get Started</a> </h1><a href=\"/changePassword\">change pass</a> <a href=\"/logout\">logout</a>")

else:

return redirect(url\_for('loginForm'))

@app.route("/changePassword", methods=["GET", "POST"])

def changePassword():

if 'email' not in session:

return redirect(url\_for('loginForm'))

if request.method == "POST":

oldPassword = request.form['oldpassword']

oldPassword = hashlib.md5(oldPassword.encode()).hexdigest()

newPassword = request.form['newpassword']

newPassword = hashlib.md5(newPassword.encode()).hexdigest()

with sqlite3.connect('database.db') as conn:

cur = conn.cursor()

cur.execute("SELECT userId, password FROM users WHERE email = '" + session['email'] + "'")

userId, password = cur.fetchone()

if (password == oldPassword):

try:

cur.execute("UPDATE users SET password = ? WHERE userId = ?", (newPassword, userId))

conn.commit()

msg="Changed successfully"

except:

conn.rollback()

msg = "Failed"

return render\_template("changePassword.html", msg=msg)

else:

msg = "Wrong password"

conn.close()

return render\_template("changePassword.html", msg=msg)

else:

return render\_template("changePassword.html")

@app.route("/loginForm")

def loginForm():

if 'email' in session:

return redirect(url\_for('root'))

else:

return render\_template('login.html', error='')

@app.route("/login", methods = ['POST', 'GET'])

def login():

if request.method == 'POST':

email = request.form['email']

password = request.form['password']

if is\_valid(email, password):

session['email'] = email

return redirect(url\_for('root'))

else:

error = 'Invalid UserId / Password'

return render\_template('login.html', error=error)

@app.route("/register", methods = ['GET', 'POST'])

def register():

if request.method == 'POST':

#Parse form data

password = request.form['password']

email = request.form['email']

firstName = request.form['firstName']

lastName = request.form['lastName']

phone = request.form['phone']

with sqlite3.connect('database.db') as con:

try:

cur = con.cursor()

cur.execute('INSERT INTO users (password, email, firstName, lastName, phone) VALUES (?, ?, ?, ?, ?)', (hashlib.md5(password.encode()).hexdigest(), email, firstName, lastName, phone))

con.commit()

msg = "Registered Successfully"

except:

con.rollback()

msg = "Error occured"

con.close()

return render\_template("login.html", error=msg)

@app.route("/registerationForm")

def registrationForm():

return render\_template("register.html")

@app.route("/logout")

def logout():

session.pop('email', None)

return redirect(url\_for('root'))

def is\_valid(email, password):

con = sqlite3.connect('database.db')

cur = con.cursor()

cur.execute('SELECT email, password FROM users')

data = cur.fetchall()

for row in data:

if row[0] == email and row[1] == hashlib.md5(password.encode()).hexdigest():

return True

return False

@app.errorhandler(404)

# inbuilt function which takes error as parameter

def not\_found(e):

# defining function

return render\_template("404.html")

@app.route('/home')

def home():

return render\_template('home.html')

@app.route('/index')

def index():

return render\_template('index.html')

def preprocessing(data):

X = data.drop('Class', axis=1)

y = data['Class']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3)

robust\_scaler = RobustScaler().fit(X\_train)

X\_test = pd.DataFrame(robust\_scaler.transform(X\_test), columns=X.columns)

return X\_test, y\_test

@app.route('/results', methods=["GET","POST"])

def results():

if request.method == "POST":

amount = request.form['amount']

now = datetime.datetime.now()

current\_time = now.strftime("%H:%M:%S")

print('Transcation Amount is {0}'.format(amount))

data = pd.read\_csv('creditcard.csv')

labels = {0: "Not Fraud", 1: "Fraud"}

X\_test, y\_test = preprocessing(data)

rf\_model = joblib.load('models/random\_forest\_model.pkl')

lr\_model = joblib.load('models/logistic\_regression\_model.pkl')

#nb\_model = joblib.load('models/naive\_bayes\_model.pkl')

input\_tuple = X\_test.sample(1) # Select a random row from X\_test

print(input\_tuple)

rf\_prediction = labels[rf\_model.predict(input\_tuple)[0]]

lr\_prediction = labels[lr\_model.predict(input\_tuple)[0]]

#nb\_prediction = labels[nb\_model.predict(input\_tuple)[0]]

y\_true = labels[y\_test.iloc[input\_tuple.index[0]]]

new\_input\_tuple = input\_tuple.to\_dict('records')[0] # Convert DataFrame to format => [{col\_name\_1: value\_1, col\_name\_2: value\_2, ...}, {second\_row}]

# Converting Amount & Time values to the ones before preprocessing so that it can be displayed on webpage

new\_input\_tuple['Amount'] = amount

new\_input\_tuple['Time'] = current\_time

# for col in new\_input\_tuple: # Round off each value to 4 decimal values so that it looks good on webpage

# new\_input\_tuple[col] = round(new\_input\_tuple[col], 4)

return render\_template("results.html", lr\_prediction=lr\_prediction, rf\_prediction=rf\_prediction,y\_true=y\_true, input\_tuple=new\_input\_tuple, time=current\_time, amount=amount)

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**Registration.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sign Up</title>

<!-- Font Awesome -->

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/all.css">

<!-- Google Fonts -->

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap">

<!-- Bootstrap core CSS -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet">

<!-- Material Design Bootstrap -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.1/css/mdb.min.css" rel="stylesheet">

<!-- JQuery -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<!-- Bootstrap tooltips -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.4/umd/popper.min.js"></script>

<!-- Bootstrap core JavaScript -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.0/js/bootstrap.min.js"></script>

<!-- MDB core JavaScript -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.1/js/mdb.min.js"></script>

</head>

<body>

<!-- Default form register -->

<form class="text-center border border-light p-5" action="/register" method="POST">

<p class="h4 mb-4">Sign up</p>

<div class="form-row mb-4">

<div class="col">

<!-- First name -->

<input type="text" name="firstName" id="defaultRegisterFormFirstName" class="form-control" placeholder="First name">

</div>

<div class="col">

<!-- Last name -->

<input type="text" name="lastName" id="defaultRegisterFormLastName" class="form-control" placeholder="Last name">

</div>

</div>

<!-- E-mail -->

<input type="email" name="email" id="defaultRegisterFormEmail" class="form-control mb-4" placeholder="E-mail">

<!-- Password -->

<input type="password" name="password" id="defaultRegisterFormPassword" class="form-control" placeholder="Password" aria-describedby="defaultRegisterFormPasswordHelpBlock">

<small id="defaultRegisterFormPasswordHelpBlock" class="form-text text-muted mb-4">

At least 8 characters and 1 digit

</small>

<!-- Phone number -->

<input type="text" name="phone" id="defaultRegisterPhonePassword" class="form-control" placeholder="Phone number" aria-describedby="defaultRegisterFormPhoneHelpBlock">

<!-- Sign up button -->

<input class="btn btn-info my-4 btn-block" type="submit" value="Sign Up"></input>

<hr>

<!-- Register -->

<p>Already a member?

<a href="/loginForm">Login</a>

</p>

</form>

</body>

{% include 'footer.html' %}

</html>

**Login.html**

<!DOCTYPE html>

<!--style>

body {

background-image: url('bg.jpg');

}

</style-->

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Login</title>

<!-- Font Awesome -->

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.8.2/css/all.css">

<!-- Google Fonts -->

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Roboto:300,400,500,700&display=swap">

<!-- Bootstrap core CSS -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.0/css/bootstrap.min.css" rel="stylesheet">

<!-- Material Design Bootstrap -->

<link href="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.1/css/mdb.min.css" rel="stylesheet">

<!-- JQuery -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

<!-- Bootstrap tooltips -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.4/umd/popper.min.js"></script>

<!-- Bootstrap core JavaScript -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/twitter-bootstrap/4.5.0/js/bootstrap.min.js"></script>

<!-- MDB core JavaScript -->

<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/mdbootstrap/4.19.1/js/mdb.min.js"></script>

</head>

<body background-color: linear-gradient(to bottom right, #c9eaff, #65c2fc);>

<!-- Default form login -->

<form class="text-center border border-light p-5" action="/login" method="POST">

{% if error != "" %}

<div class="alert alert-danger" role="alert">

{{error}}

</div>

{% else %}

<div></div>

{% endif %}

<p class="h4 mb-4">Login</p>

<!-- Email -->

<input type="email" name="email" id="defaultLoginFormEmail" class="form-control mb-4" placeholder="E-mail">

<!-- Password -->

<input type="password" name="password" id="defaultLoginFormPassword" class="form-control mb-4" placeholder="Password">

<input class="btn btn-info btn-block my-4" type="submit" value="Login"></input>

<!-- Register -->

<p>Not a member?

<a href="/registerationForm">Register</a>

</p>

</form>

</body>

{% include 'footer.html' %}

</html>

**Home.html**

<link href="https://fonts.googleapis.com/css?family=B612&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<body class="text-center bg-dark text-light" style="font-family: 'B612', sans-serif;">

<div class="container">

<img src="/static/background.jpg" style="margin: 50px auto 50px auto">

</div>

<div class="container">

<a class="btn btn-lg btn-primary" href="{{url\_for('results')}}"> Predict Results </a>

</div>

</body>

**Results.html**

<link href="https://fonts.googleapis.com/css?family=B612&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh" crossorigin="anonymous">

<body class="text-center bg-dark text-light" style="font-family: 'B612', sans-serif;">

<div class="container">

<img class="img-responsive" src="/static/background.jpg" style="margin: 50px auto 50px auto">

</div>

<div class="container" id="input\_display">

<h2> <u> Features of the tuple </u> </h3>

<h5>

Time of transcation: {{time}}

</h5>

<h5>

Amount: {{amount}}

</h5>

</div>

<div class="container" style="margin: 20px auto 20px auto;">

<h2> <u> Prediction Results </u> </h2>

<h4> Logistic Regression Result : {{lr\_prediction}} </h2>

<h4> Random Forest Result : {{rf\_prediction}} </h2>

<h4> Naive Bayes Result : {{nb\_prediction}} </h2>

<h4> Overall Result : {{y\_true}} </h2>

</div>

<a class="btn btn-lg btn-primary" href="{{url\_for('root')}}"> Back </a>

</body>

**5.2.1 Code Efficiency**

The system uses python with flask. Python, is most suitable for this project as it has simple syntax and readability promoting the rapid testing of complex machine learning algorithms and a thriving community bolstered by collaborative tools like Jupyter Notebooks and Google Colab. The primary advantages of choosing Flask are built-in fast debugger, secure cookies and unit testing support, Unicode basis, and more. The framework allows developers to decide how exactly to build the application needed.

**5.3 Testing Approach**

**5.3.1 Unit Testing**

Testing each component or module of your software project is known as unit testing. To perform this kind of testing, knowledge of programming is necessary. So only programmers do this kind of tests, not testers.

**5.3.2 Integration Testing**

After integrating the modules, you need to see if the combined modules work together or not. This type of testing is known as integration testing. You need to perform fewer integration tests than unit tests.

**5.3.3 System Testing**

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements. System testing takes, as its input, all of the integrated components that have passed integration testing.

**5.4 Modification and Improvements**

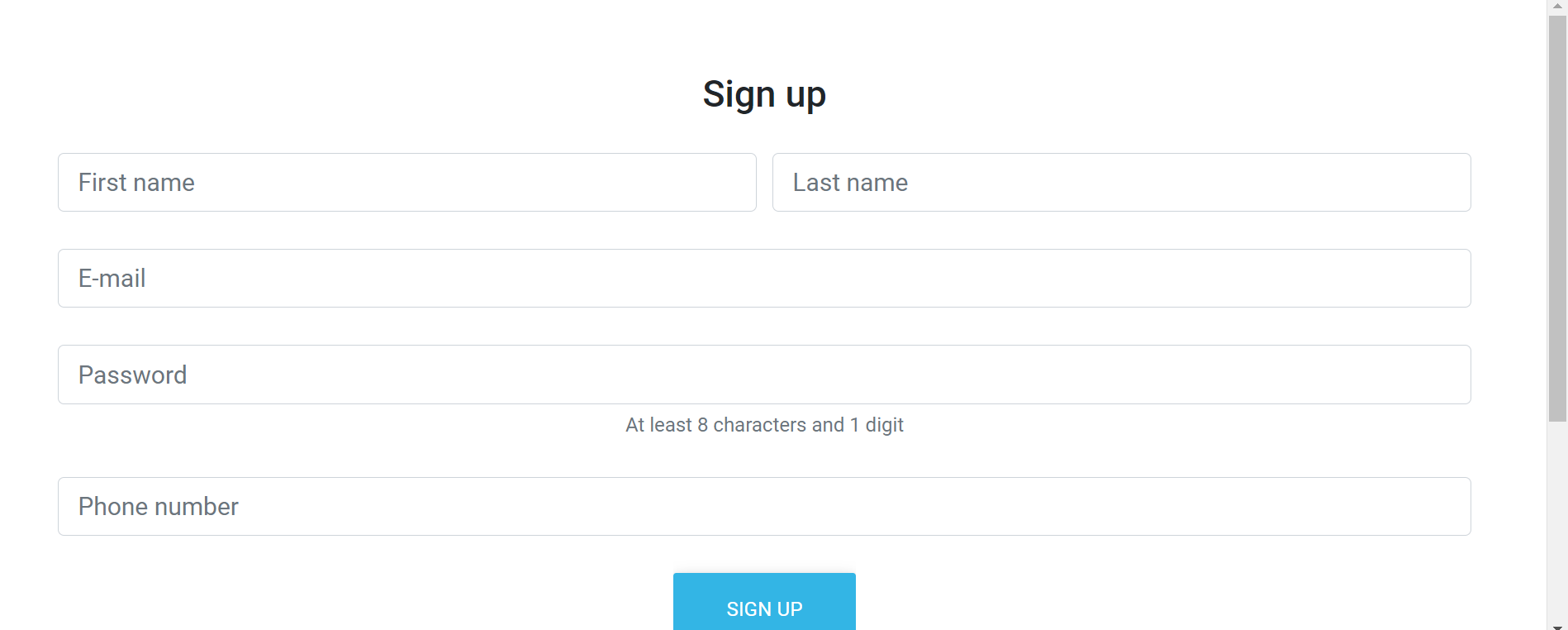
After testing and handling all errors, exceptions and bugs, errors were solved by troubleshooting, exceptions were handled by try catch block at the correct places avoiding redundancy of code from index files. Bugs can never be removed completely but most of them are solved and remaining are under development. The project can be further improved by adding the functionality of creating an arrangement for multiple examinations with multiple courses.

**6. RESULTS AND DISCUSSION**

**6.1 User documentation**

**Module 1**

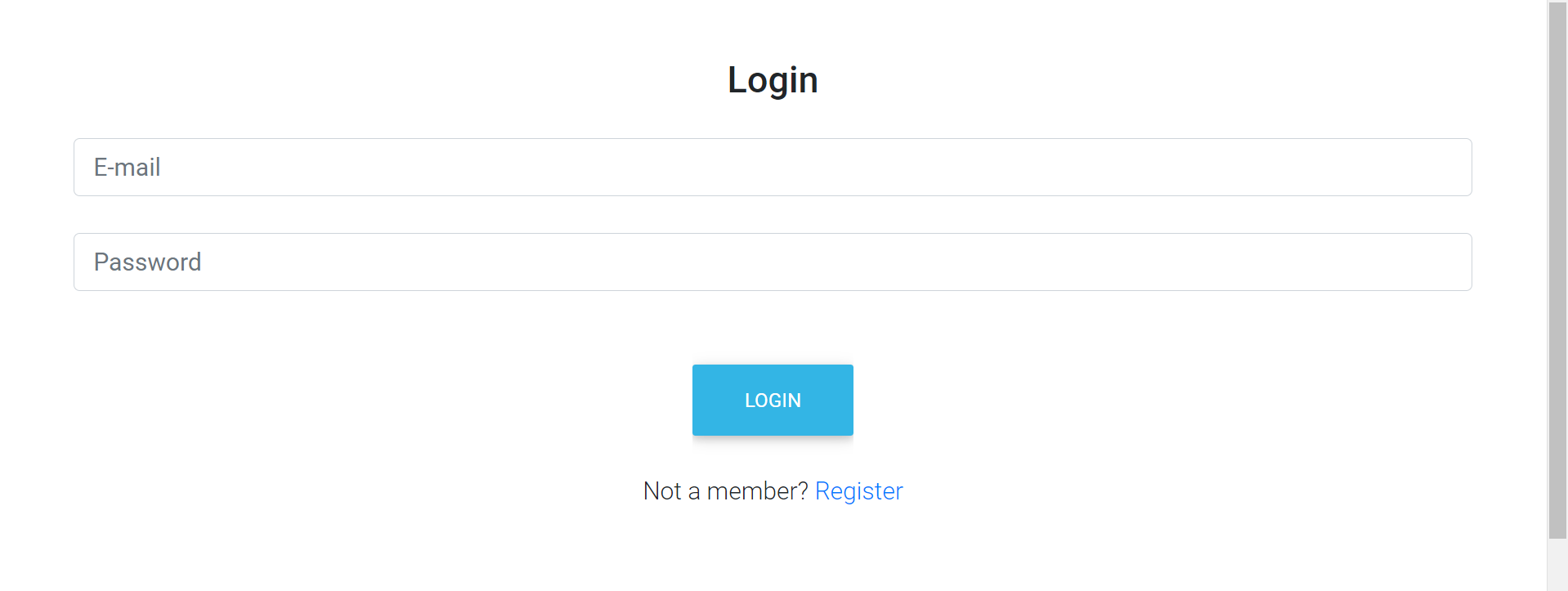
Sign up



**Figure 19:** Registration Page

**Module 2**

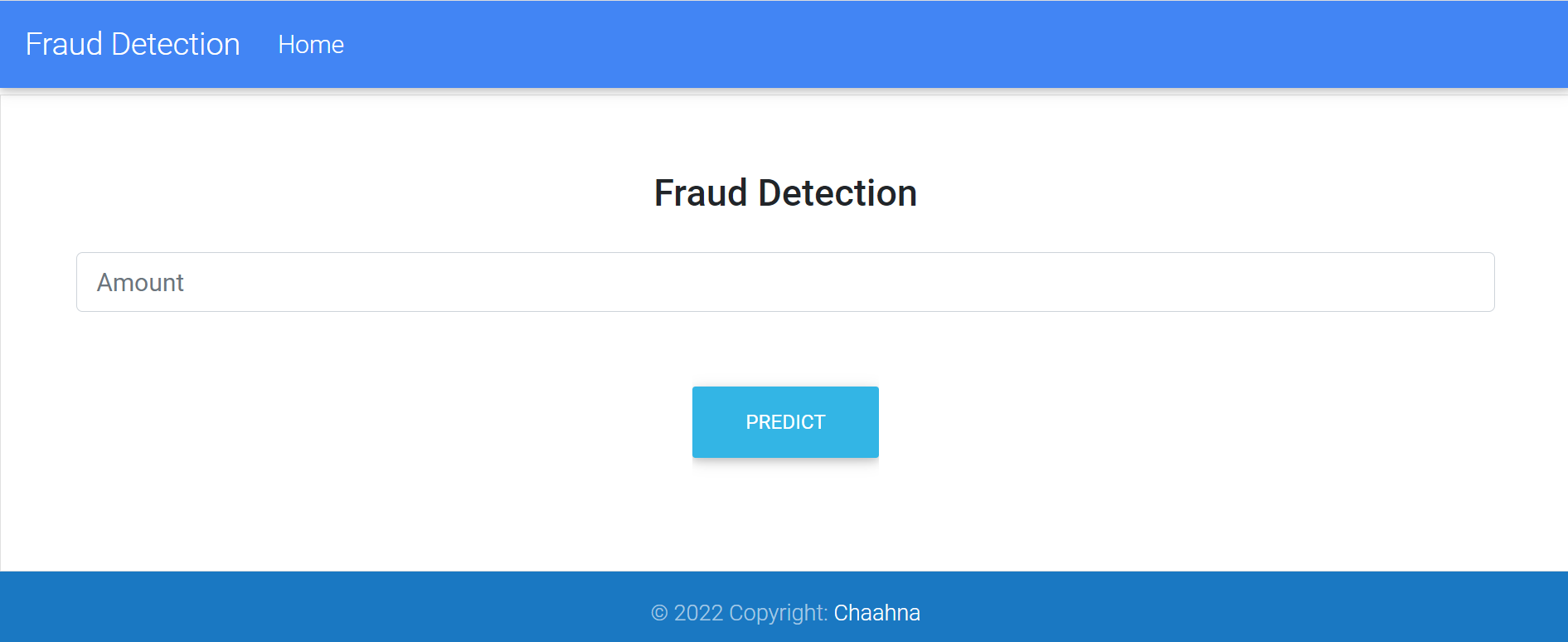
Login



**Figure 20:** Login Page

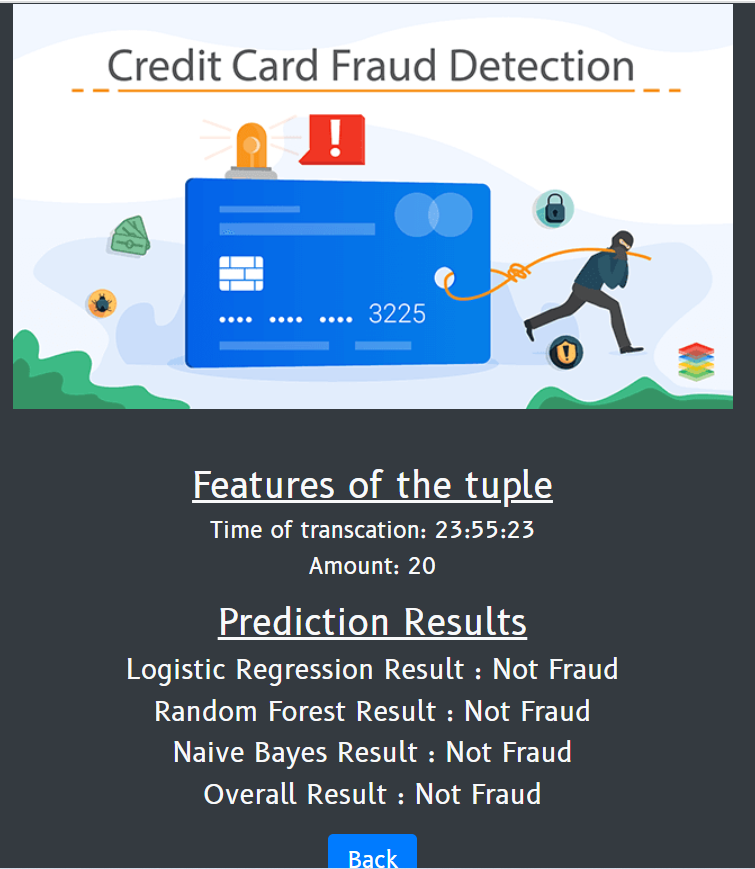
**Module 3**

**Detection with logical regression, random forest and naïve bayes.**



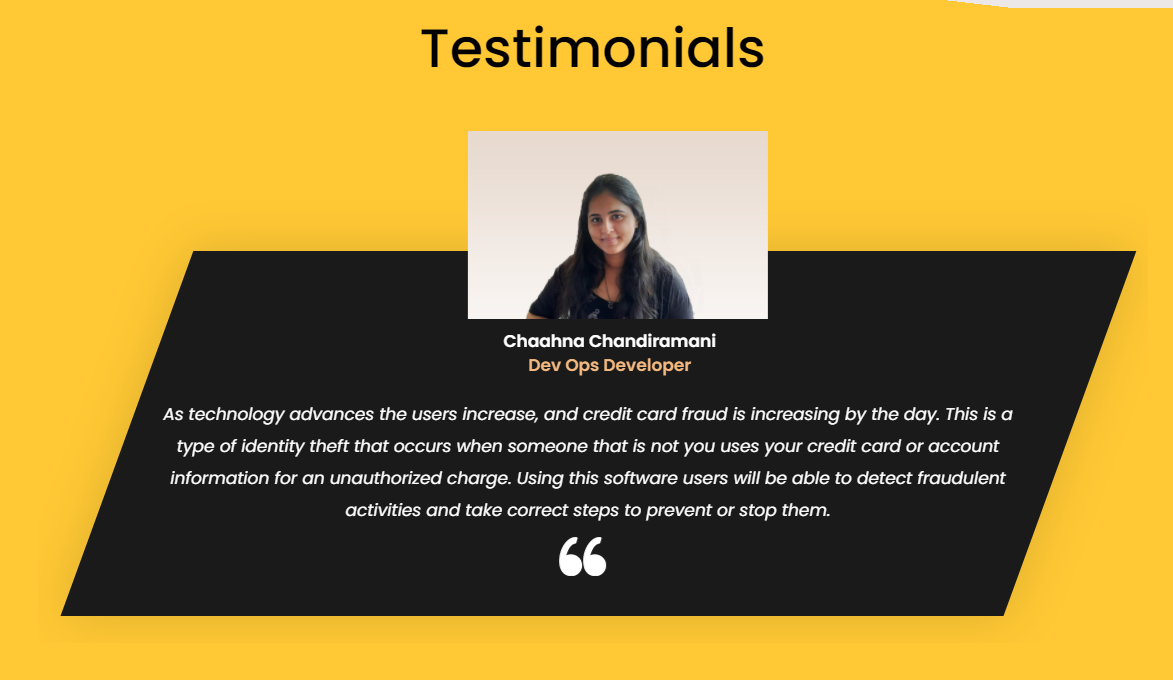
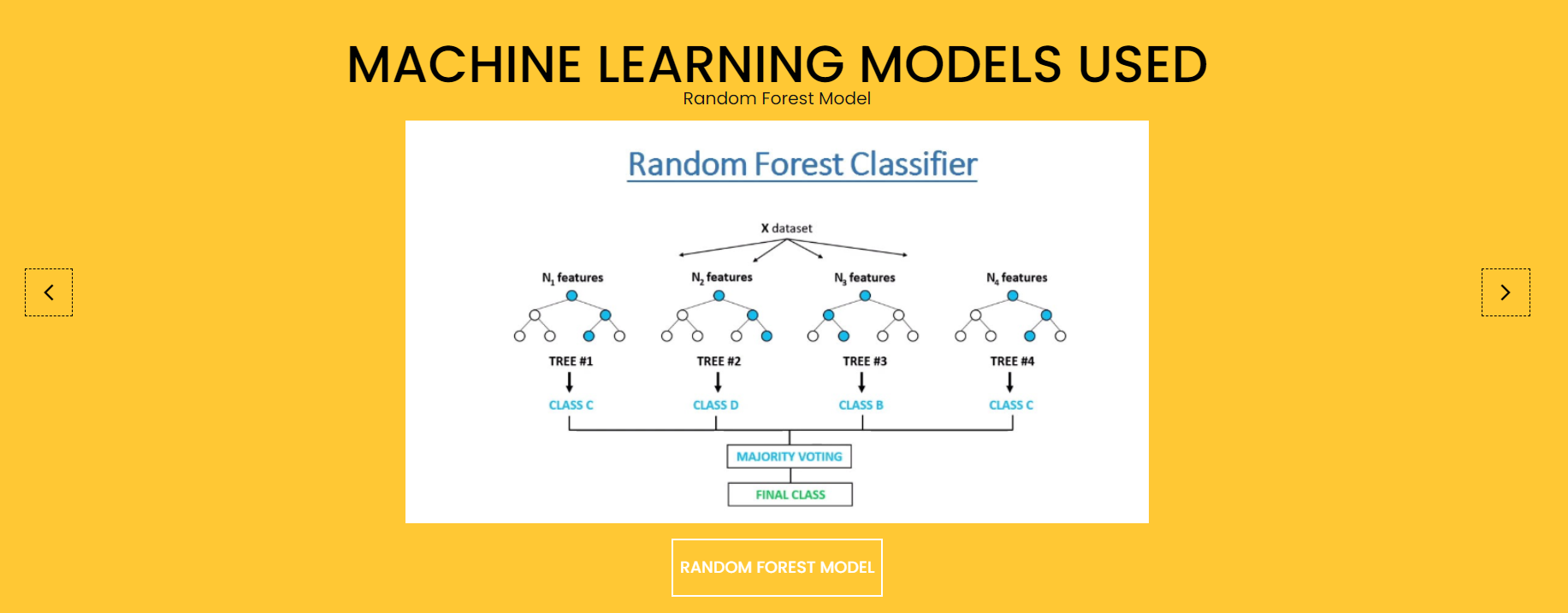
**Figure 21:** Data Entry Page

**Module 4**



**Figure 22:** Result Page

**Module 5**



**Figure 23:** About Page

**7 CONCLUSIONS**

**7.1 Conclusion**

- As technology advances the users increase, and credit card fraud is increasing by the day. This is a type of identity theft that occurs when someone that is not you uses your credit card or account information for an unauthorized charge.

- Using this software users will be able to detect fraudulent activities and take correct steps to prevent or stop them.

**7.2 Limitations**

- User must provide the required information with accuracy. Incase the user is unable to provide the details detection will noy be possible.

- Nonexistence of standard and comprehensive credit card benchmark or dataset Credit card is inherently private property, so creating a proper benchmark for this purpose is very difficult. Incomplete datasets can cause fraud detection system to learn fraud tricks or normal behavior partially. On the other hand, lack of a standard dataset makes comparison of various techniques problematic or impossible. Many researchers used datasets that are only permitted to authors and cannot be published in order to privacy considerations.

**7.3 Future scope of the project**

* The future scope of the project will include integration with different applications such that the user is provided with options to detect fraud within the same system during or after a transaction.

**References**

* <https://www.geeksforgeeks.org/frontend-vs-backend/>
* <https://lucid.app/documents#/templates?folder_id=home>
* <https://www.kaggle.com/mlg-ulb/creditcardfraud>
* <https://www.ijream.org/papers/INJRV01I02004.pdf>
* <https://inloop.github.io/sqlite-viewer/>
* <https://en.wikipedia.org/wiki/Credit_card_fraud>
* <https://mint.intuit.com/blog/planning/credit-card-fraud-statistics/>
* <https://www.researchgate.net/publication/261048703_Payment_Card_Fraud_Challenges_and_Solutions>
* <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.677.970&rep=rep1&type=pdf#:~:text=Credit%20card%20fraud%20detection%20is,time%2C%20address%20of%20the%20merchant>
* <https://www.mygreatlearning.com/blog/credit-card-fraud-detection/>
* <https://www.ijert.org/credit-card-fraud-detection-using-machine-learning>