**Project Structure:**

online payments fraud detection/

├── data/

│ └── PS\_20174392719\_1491204439457\_logs.csv

│

├── flask/

│ ├── templates/

│ │ ├── home.html

│ │ ├── predict.html

│ │ └── submit.html

│ ├── app\_ibm.py

│ ├── app.py

│ └── payments.pkl

│

├── training/

│ ├── ONLINE PAYMENTS FRAUD DETECTION.ipynb

│ └── payments.pkl

│

└── training\_ibm/

└── online payments fraud prediction using ibm.ipynb

**1.app.py :**

This is the main Flask application file.

from flask import Flask, request, jsonify, render\_template import pickle import numpy as np import logging from utils import preprocess\_input

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

# Configure logging logging.basicConfig(level=logging.INFO)

# Load the trained model

try: with open('fraud\_detection\_model.pkl', 'rb') as file:

model = pickle.load(file) logging.info("Model loaded successfully.") except Exception as e:

logging.error(f"Error loading model: {e}") model = None

@app.route('/') def home():

return render\_template('index.html')

@app.route('/predict', methods=['POST']) def predict(): try:

data = request.form.to\_dict() logging.info(f"Received data: {data}") data = preprocess\_input(data) prediction = model.predict([data])

result = 'Fraudulent' if prediction[0] == 1 else 'Not Fraudulent' logging.info(f"Prediction: {result}") return jsonify({'prediction': result}) except Exception as e:

logging.error(f"Error during prediction: {e}") return jsonify({'error': str(e)}), 500

if \_\_name\_\_ == '\_\_main\_\_': app.run(debug=True)

**2.utils.py :**

This file contains utility functions for data preprocessing.

import numpy as np

def preprocess\_input(data): try:

processed\_data = [ float(data['amount']), int(data['transaction\_type']),

# Add more fields as required by your model

]

return np.array(processed\_data) except KeyError as e:

raise ValueError(f"Missing value for: {e}") except ValueError as e:

raise ValueError(f"Invalid value: {e}")

1. **requirements.txt :**

Flask==2.0.2numpy==1.21.2 scikit-learn==0.24.2

1. **fraud\_detection\_model.apk :**

Serialized machine learning model. Example code to create this file:

import pickle

from sklearn.ensemble import RandomForestClassifier

# Sample data and labels (replace with your actual dataset) X = [[0.1, 1], [0.2, 0], [0.3, 1], [0.4, 0]] y = [0, 1, 0, 1]

model = RandomForestClassifier() model.fit(X, y)

# Save the model with open('fraud\_detection\_model.pkl', 'wb') as file:

pickle.dump(model, file)

**Home.html & css & js**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Online Payments Fraud Detection</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Arial', sans-serif;

background: linear-gradient(135deg, #87CEEB 0%, #4682B4 100%);

min-height: 100vh;

position: relative;

overflow-x: hidden;

}

/\* Background decorative elements \*/

.bg-decoration {

position: absolute;

opacity: 0.3;

z-index: 1;

}

.gear {

width: 60px;

height: 60px;

border: 4px solid #fff;

border-radius: 50%;

position: absolute;

top: 20%;

left: 10%;

}

.gear::before {

content: '';

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 20px;

height: 20px;

background: #fff;

border-radius: 50%;

}

.chart-bars {

position: absolute;

bottom: 20%;

left: 8%;

display: flex;

align-items: flex-end;

gap: 8px;

}

.bar {

width: 20px;

background: #4169E1;

border-radius: 2px;

}

.bar:nth-child(1) { height: 40px; }

.bar:nth-child(2) { height: 60px; }

.bar:nth-child(3) { height: 30px; }

.credit-card {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 120px;

height: 80px;

background: linear-gradient(45deg, #4169E1, #87CEEB);

border-radius: 12px;

border: 2px solid #fff;

}

.credit-card::before {

content: '';

position: absolute;

top: 20px;

left: 20px;

width: 30px;

height: 20px;

background: #FFD700;

border-radius: 4px;

}

.credit-card::after {

content: '---- ---- ---- ----';

position: absolute;

bottom: 15px;

left: 15px;

color: #fff;

font-size: 10px;

font-family: monospace;

}

.shield {

position: absolute;

bottom: 15%;

right: 15%;

width: 80px;

height: 90px;

background: linear-gradient(45deg, #4169E1, #87CEEB);

clip-path: polygon(50% 0%, 100% 25%, 100% 75%, 50% 100%, 0% 75%, 0% 25%);

border: 3px solid #fff;

}

.shield::before {

content: '🔒';

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 24px;

color: #fff;

}

/\* Main content \*/

.container {

position: relative;

z-index: 10;

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

.header {

display: flex;

justify-content: flex-end;

gap: 15px;

margin-bottom: 40px;

}

.nav-button {

padding: 12px 24px;

border: none;

border-radius: 8px;

font-size: 16px;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

text-decoration: none;

display: inline-block;

text-align: center;

}

.home-btn {

background: #5DADE2;

color: #fff;

box-shadow: 0 4px 15px rgba(93, 173, 226, 0.4);

}

.predict-btn {

background: #2E86AB;

color: #fff;

box-shadow: 0 4px 15px rgba(46, 134, 171, 0.4);

}

.nav-button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 20px rgba(0, 0, 0, 0.2);

}

.main-title {

text-align: center;

font-size: 3.5rem;

font-weight: bold;

color: #2C3E50;

margin: 60px 0 40px;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);

letter-spacing: -1px;

}

.content-box {

background: rgba(255, 255, 255, 0.95);

backdrop-filter: blur(10px);

border-radius: 20px;

padding: 40px;

margin: 40px auto;

max-width: 900px;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);

border: 1px solid rgba(255, 255, 255, 0.3);

}

.content-text {

font-size: 1.2rem;

line-height: 1.8;

color: #555;

text-align: justify;

font-style: italic;

}

.highlight {

color: #2E86AB;

font-weight: 600;

}

/\* Floating elements animation \*/

@keyframes float {

0%, 100% { transform: translateY(0px); }

50% { transform: translateY(-10px); }

}

.gear {

animation: float 3s ease-in-out infinite;

}

.shield {

animation: float 4s ease-in-out infinite 1s;

}

/\* Responsive design \*/

@media (max-width: 768px) {

.main-title {

font-size: 2.5rem;

margin: 40px 0 30px;

}

.content-box {

padding: 30px 20px;

margin: 20px;

}

.content-text {

font-size: 1.1rem;

}

.nav-button {

padding: 10px 20px;

font-size: 14px;

}

}

@media (max-width: 480px) {

.main-title {

font-size: 2rem;

}

.header {

justify-content: center;

}

}

</style>

</head>

<body>

<!-- Background decorative elements -->

<div class="bg-decoration gear"></div>

<div class="bg-decoration chart-bars">

<div class="bar"></div>

<div class="bar"></div>

<div class="bar"></div>

</div>

<div class="bg-decoration credit-card"></div>

<div class="bg-decoration shield"></div>

<div class="container">

<!-- Navigation Header -->

<div class="header">

<a href="Home.html" class="nav-button home-btn">Home</a>

<a href="predict.html" class="nav-button predict-btn">Predict</a>

</div>

<!-- Main Title -->

<h1 class="main-title">Online Payments Fraud Detection</h1>

<!-- Content Box -->

<div class="content-box">

<p class="content-text">

The objective of this article is to <span class="highlight">predict online payments fraud</span> given the various parameters. This will be a <span class="highlight">classification problem</span> since the target or dependent variable is the fraud (categorical values). The purpose of fraud of online payments are to separate the available supply of potable online payments into classes differing in superiority. We will be using <span class="highlight">classification algorithms</span> such as <span class="highlight">Decision tree, Random forest, SVM, and Extra tree classifier</span>. We will train and test the data with these algorithms.

</p>

</div>

</div>

</body>

</html>

**predict.html,css and js**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Online Payments Fraud Detection - Predict</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Arial', sans-serif;

background: linear-gradient(135deg, #87CEEB 0%, #4682B4 100%);

min-height: 100vh;

position: relative;

overflow-x: hidden;

}

/\* Background decorative elements \*/

.bg-decoration {

position: absolute;

opacity: 0.3;

z-index: 1;

}

.hacker-silhouette {

position: absolute;

top: 20%;

right: 20%;

width: 200px;

height: 250px;

background: rgba(255, 255, 255, 0.2);

border-radius: 50%;

overflow: hidden;

}

.hacker-silhouette::before {

content: '👤';

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 120px;

color: rgba(255, 255, 255, 0.3);

}

.gear-left {

width: 50px;

height: 50px;

border: 3px solid rgba(255, 255, 255, 0.4);

border-radius: 50%;

position: absolute;

top: 60%;

left: 15%;

}

.gear-right {

width: 40px;

height: 40px;

border: 3px solid rgba(255, 255, 255, 0.4);

border-radius: 50%;

position: absolute;

top: 65%;

left: 20%;

}

.chart-bars {

position: absolute;

bottom: 25%;

left: 10%;

display: flex;

align-items: flex-end;

gap: 6px;

}

.bar {

width: 15px;

background: rgba(65, 105, 225, 0.6);

border-radius: 2px;

}

.bar:nth-child(1) { height: 35px; }

.bar:nth-child(2) { height: 50px; }

.bar:nth-child(3) { height: 25px; }

.credit-card {

position: absolute;

bottom: 30%;

right: 25%;

width: 120px;

height: 80px;

background: linear-gradient(45deg, #4169E1, #87CEEB);

border-radius: 12px;

border: 2px solid rgba(255, 255, 255, 0.5);

}

.credit-card::before {

content: '';

position: absolute;

top: 15px;

left: 15px;

width: 25px;

height: 15px;

background: #FFD700;

border-radius: 3px;

}

.credit-card::after {

content: '---- ---- ---- ----';

position: absolute;

bottom: 10px;

left: 10px;

color: rgba(255, 255, 255, 0.8);

font-size: 8px;

font-family: monospace;

}

.shield {

position: absolute;

bottom: 10%;

right: 10%;

width: 70px;

height: 80px;

background: linear-gradient(45deg, #4169E1, #87CEEB);

clip-path: polygon(50% 0%, 100% 25%, 100% 75%, 50% 100%, 0% 75%, 0% 25%);

border: 2px solid rgba(255, 255, 255, 0.5);

}

.shield::before {

content: '🔒';

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 20px;

}

.magnifying-glass {

position: absolute;

top: 40%;

right: 15%;

width: 40px;

height: 40px;

border: 3px solid rgba(65, 105, 225, 0.6);

border-radius: 50%;

}

.magnifying-glass::after {

content: '';

position: absolute;

bottom: -10px;

right: -5px;

width: 15px;

height: 3px;

background: rgba(65, 105, 225, 0.6);

transform: rotate(45deg);

}

/\* Main content \*/

.container {

position: relative;

z-index: 10;

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 30px;

}

.main-title {

font-size: 2.5rem;

font-weight: bold;

color: #2C3E50;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);

}

.nav-buttons {

display: flex;

gap: 15px;

}

.nav-button {

padding: 12px 24px;

border: none;

border-radius: 8px;

font-size: 16px;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

text-decoration: none;

display: inline-block;

text-align: center;

}

.home-btn {

background: #5DADE2;

color: #fff;

box-shadow: 0 4px 15px rgba(93, 173, 226, 0.4);

}

.predict-btn {

background: #2E86AB;

color: #fff;

box-shadow: 0 4px 15px rgba(46, 134, 171, 0.4);

}

.nav-button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 20px rgba(0, 0, 0, 0.2);

}

.form-container {

background: rgba(255, 255, 255, 0.95);

backdrop-filter: blur(10px);

border-radius: 20px;

padding: 40px;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);

border: 1px solid rgba(255, 255, 255, 0.3);

max-width: 500px;

margin: 0 auto;

}

.form-group {

margin-bottom: 25px;

}

.form-label {

display: block;

font-size: 1.1rem;

font-weight: 600;

color: #2C3E50;

margin-bottom: 8px;

}

.form-input {

width: 100%;

padding: 12px 16px;

border: 2px solid rgba(93, 173, 226, 0.3);

border-radius: 10px;

font-size: 1rem;

background: rgba(255, 255, 255, 0.9);

transition: all 0.3s ease;

color: #666;

}

.form-input:focus {

outline: none;

border-color: #5DADE2;

box-shadow: 0 0 0 3px rgba(93, 173, 226, 0.1);

transform: translateY(-1px);

}

.form-input::placeholder {

color: #999;

font-style: italic;

}

.predict-button {

width: 100%;

padding: 15px;

background: linear-gradient(45deg, #2E86AB, #5DADE2);

color: white;

border: none;

border-radius: 10px;

font-size: 1.2rem;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

margin-top: 20px;

box-shadow: 0 4px 15px rgba(46, 134, 171, 0.3);

}

.predict-button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 25px rgba(46, 134, 171, 0.4);

}

.predict-button:active {

transform: translateY(0);

}

.predict-button:disabled {

opacity: 0.7;

cursor: not-allowed;

transform: none;

}

/\* Loading animation \*/

.loading {

display: none;

text-align: center;

margin-top: 20px;

}

.spinner {

border: 3px solid rgba(46, 134, 171, 0.3);

border-top: 3px solid #2E86AB;

border-radius: 50%;

width: 30px;

height: 30px;

animation: spin 1s linear infinite;

margin: 0 auto 10px;

}

@keyframes spin {

0% { transform: rotate(0deg); }

100% { transform: rotate(360deg); }

}

/\* Floating animations \*/

@keyframes float {

0%, 100% { transform: translateY(0px) rotate(0deg); }

50% { transform: translateY(-8px) rotate(5deg); }

}

@keyframes pulse {

0%, 100% { opacity: 0.3; }

50% { opacity: 0.6; }

}

.gear-left, .gear-right {

animation: float 4s ease-in-out infinite;

}

.gear-right {

animation-delay: 1s;

}

.hacker-silhouette {

animation: pulse 3s ease-in-out infinite;

}

/\* Responsive design \*/

@media (max-width: 768px) {

.header {

flex-direction: column;

gap: 20px;

text-align: center;

}

.main-title {

font-size: 2rem;

}

.form-container {

padding: 30px 20px;

margin: 20px;

}

.nav-button {

padding: 10px 20px;

font-size: 14px;

}

}

@media (max-width: 480px) {

.main-title {

font-size: 1.8rem;

}

.form-container {

padding: 25px 15px;

}

}

</style>

</head>

<body>

<!-- Background decorative elements -->

<div class="bg-decoration hacker-silhouette"></div>

<div class="bg-decoration gear-left"></div>

<div class="bg-decoration gear-right"></div>

<div class="bg-decoration chart-bars">

<div class="bar"></div>

<div class="bar"></div>

<div class="bar"></div>

</div>

<div class="bg-decoration credit-card"></div>

<div class="bg-decoration shield"></div>

<div class="bg-decoration magnifying-glass"></div>

<div class="container">

<!-- Header -->

<div class="header">

<h1 class="main-title">Online Payments Fraud Detection</h1>

<div class="nav-buttons">

<a href="home.html" class="nav-button home-btn">Home</a>

<a href="predict.html" class="nav-button predict-btn">Predict</a>

</div>

</div>

<!-- Prediction Form -->

<div class="form-container">

<form id="fraudPredictionForm">

<div class="form-group">

<label for="step" class="form-label">Step</label>

<input type="number" id="step" name="step" class="form-input"

placeholder="step represents a unit of time when..." required>

</div>

<div class="form-group">

<label for="type" class="form-label">Type</label>

<select id="type" name="type" class="form-input" required>

<option value="">type of online transaction</option>

<option value="PAYMENT">PAYMENT</option>

<option value="TRANSFER">TRANSFER</option>

<option value="CASH\_OUT">CASH\_OUT</option>

<option value="CASH\_IN">CASH\_IN</option>

<option value="DEBIT">DEBIT</option>

</select>

</div>

<div class="form-group">

<label for="amount" class="form-label">Amount</label>

<input type="number" id="amount" name="amount" class="form-input"

placeholder="the amount of the transaction" step="0.01" required>

</div>

<div class="form-group">

<label for="oldbalanceOrg" class="form-label">OldbalanceOrg</label>

<input type="number" id="oldbalanceOrg" name="oldbalanceOrg" class="form-input"

placeholder="balance before the transaction" step="0.01" required>

</div>

<div class="form-group">

<label for="newbalanceOrg" class="form-label">NewbalanceOrg</label>

<input type="number" id="newbalanceOrg" name="newbalanceOrg" class="form-input"

placeholder="balance after the transaction" step="0.01" required>

</div>

<div class="form-group">

<label for="oldbalanceDest" class="form-label">OldbalanceDest</label>

<input type="number" id="oldbalanceDest" name="oldbalanceDest" class="form-input"

placeholder="initial balance of recipient before the..." step="0.01" required>

</div>

<div class="form-group">

<label for="newbalanceDest" class="form-label">NewbalanceDest</label>

<input type="number" id="newbalanceDest" name="newbalanceDest" class="form-input"

placeholder="the new balance of recipient after th..." step="0.01" required>

</div>

<button type="submit" class="predict-button" id="predictBtn">Predict Fraud</button>

</form>

<div class="loading" id="loading">

<div class="spinner"></div>

<p>Analyzing transaction data...</p>

</div>

</div>

</div>

<script>

document.getElementById('fraudPredictionForm').addEventListener('submit', function(e) {

e.preventDefault();

// Show loading animation

const loadingDiv = document.getElementById('loading');

const predictBtn = document.getElementById('predictBtn');

predictBtn.disabled = true;

predictBtn.textContent = 'Analyzing...';

loadingDiv.style.display = 'block';

// Get form data

const formData = new FormData(this);

const data = {};

for (let [key, value] of formData.entries()) {

data[key] = value;

}

// Simple fraud detection logic (for demonstration)

let fraudScore = 0;

// Check for suspicious patterns

const amount = parseFloat(data.amount);

const oldBalanceOrg = parseFloat(data.oldbalanceOrg);

const newBalanceOrg = parseFloat(data.newbalanceOrg);

const oldBalanceDest = parseFloat(data.oldbalanceDest);

const newBalanceDest = parseFloat(data.newbalanceDest);

// Fraud indicators

if (amount > 100000) fraudScore += 30;

if (data.type === 'CASH\_OUT' || data.type === 'TRANSFER') fraudScore += 20;

if (Math.abs((oldBalanceOrg - amount) - newBalanceOrg) > 0.01) fraudScore += 25;

if (oldBalanceOrg === 0 && newBalanceOrg === 0) fraudScore += 35;

if (oldBalanceDest === 0 && newBalanceDest === 0) fraudScore += 30;

if (amount === oldBalanceOrg) fraudScore += 20;

// Determine result based on fraud score

let result;

if (fraudScore >= 50) {

result = 'fraud';

} else if (fraudScore >= 25) {

result = 'warning';

} else {

result = 'safe';

}

// Simulate processing time

setTimeout(() => {

// Redirect to submit.html with the result

window.location.href = `submit.html?result=${result}&score=${fraudScore}`;

}, 2000); // 2 second delay to show loading animation

});

</script>

</body>

</html>

**submit.html,css and js:**<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Online Payments Fraud Detection - Result</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Arial', sans-serif;

background: linear-gradient(135deg, #87CEEB 0%, #4682B4 100%);

min-height: 100vh;

position: relative;

overflow-x: hidden;

}

/\* Background decorative elements \*/

.bg-decoration {

position: absolute;

opacity: 0.3;

z-index: 1;

}

.hacker-silhouette {

position: absolute;

top: 20%;

right: 20%;

width: 200px;

height: 250px;

background: rgba(255, 255, 255, 0.2);

border-radius: 50%;

overflow: hidden;

}

.hacker-silhouette::before {

content: '👤';

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 120px;

color: rgba(255, 255, 255, 0.3);

}

.gear-left {

width: 50px;

height: 50px;

border: 3px solid rgba(255, 255, 255, 0.4);

border-radius: 50%;

position: absolute;

top: 60%;

left: 15%;

}

.gear-right {

width: 40px;

height: 40px;

border: 3px solid rgba(255, 255, 255, 0.4);

border-radius: 50%;

position: absolute;

top: 65%;

left: 20%;

}

.chart-bars {

position: absolute;

bottom: 25%;

left: 10%;

display: flex;

align-items: flex-end;

gap: 6px;

}

.bar {

width: 15px;

background: rgba(65, 105, 225, 0.6);

border-radius: 2px;

}

.bar:nth-child(1) { height: 35px; }

.bar:nth-child(2) { height: 50px; }

.bar:nth-child(3) { height: 25px; }

.credit-card {

position: absolute;

bottom: 30%;

right: 25%;

width: 120px;

height: 80px;

background: linear-gradient(45deg, #4169E1, #87CEEB);

border-radius: 12px;

border: 2px solid rgba(255, 255, 255, 0.5);

}

.credit-card::before {

content: '';

position: absolute;

top: 15px;

left: 15px;

width: 25px;

height: 15px;

background: #FFD700;

border-radius: 3px;

}

.credit-card::after {

content: '---- ---- ---- ----';

position: absolute;

bottom: 10px;

left: 10px;

color: rgba(255, 255, 255, 0.8);

font-size: 8px;

font-family: monospace;

}

.shield {

position: absolute;

bottom: 10%;

right: 10%;

width: 70px;

height: 80px;

background: linear-gradient(45deg, #4169E1, #87CEEB);

clip-path: polygon(50% 0%, 100% 25%, 100% 75%, 50% 100%, 0% 75%, 0% 25%);

border: 2px solid rgba(255, 255, 255, 0.5);

}

.shield::before {

content: '🔒';

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

font-size: 20px;

}

.magnifying-glass {

position: absolute;

top: 40%;

right: 15%;

width: 40px;

height: 40px;

border: 3px solid rgba(65, 105, 225, 0.6);

border-radius: 50%;

}

.magnifying-glass::after {

content: '';

position: absolute;

bottom: -10px;

right: -5px;

width: 15px;

height: 3px;

background: rgba(65, 105, 225, 0.6);

transform: rotate(45deg);

}

/\* Main content \*/

.container {

position: relative;

z-index: 10;

max-width: 1200px;

margin: 0 auto;

padding: 20px;

}

.header {

display: flex;

justify-content: space-between;

align-items: center;

margin-bottom: 30px;

}

.main-title {

font-size: 2.5rem;

font-weight: bold;

color: #2C3E50;

text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.1);

}

.nav-buttons {

display: flex;

gap: 15px;

}

.nav-button {

padding: 12px 24px;

border: none;

border-radius: 8px;

font-size: 16px;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

text-decoration: none;

display: inline-block;

text-align: center;

}

.home-btn {

background: #5DADE2;

color: #fff;

box-shadow: 0 4px 15px rgba(93, 173, 226, 0.4);

}

.predict-btn {

background: #2E86AB;

color: #fff;

box-shadow: 0 4px 15px rgba(46, 134, 171, 0.4);

}

.nav-button:hover {

transform: translateY(-2px);

box-shadow: 0 6px 20px rgba(0, 0, 0, 0.2);

}

.result-container {

background: rgba(255, 255, 255, 0.95);

backdrop-filter: blur(10px);

border-radius: 20px;

padding: 40px;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1);

border: 1px solid rgba(255, 255, 255, 0.3);

max-width: 800px;

margin: 0 auto;

text-align: center;

}

.result-icon {

font-size: 4rem;

margin-bottom: 20px;

display: block;

}

.result-title {

font-size: 2.2rem;

font-weight: bold;

margin-bottom: 15px;

text-shadow: 1px 1px 2px rgba(0, 0, 0, 0.1);

}

.result-description {

font-size: 1.3rem;

margin-bottom: 25px;

line-height: 1.6;

color: #555;

}

.fraud-score {

background: rgba(255, 255, 255, 0.8);

border-radius: 15px;

padding: 20px;

margin: 25px 0;

border-left: 5px solid;

}

.score-label {

font-size: 1rem;

color: #666;

margin-bottom: 5px;

}

.score-value {

font-size: 1.8rem;

font-weight: bold;

}

.recommendations {

background: rgba(255, 255, 255, 0.8);

border-radius: 15px;

padding: 25px;

margin: 25px 0;

text-align: left;

}

.recommendations h3 {

color: #2C3E50;

margin-bottom: 15px;

font-size: 1.3rem;

}

.recommendations ul {

list-style: none;

padding: 0;

}

.recommendations li {

padding: 8px 0;

padding-left: 25px;

position: relative;

color: #555;

line-height: 1.5;

}

.recommendations li::before {

content: '▶';

position: absolute;

left: 0;

color: #5DADE2;

font-size: 0.8rem;

}

.action-buttons {

display: flex;

gap: 20px;

justify-content: center;

margin-top: 30px;

flex-wrap: wrap;

}

.action-btn {

padding: 15px 30px;

border: none;

border-radius: 10px;

font-size: 1.1rem;

font-weight: 600;

cursor: pointer;

transition: all 0.3s ease;

text-decoration: none;

display: inline-block;

min-width: 180px;

}

.new-prediction-btn {

background: linear-gradient(45deg, #2E86AB, #5DADE2);

color: white;

box-shadow: 0 4px 15px rgba(46, 134, 171, 0.3);

}

.home-redirect-btn {

background: linear-gradient(45deg, #27AE60, #58D68D);

color: white;

box-shadow: 0 4px 15px rgba(39, 174, 96, 0.3);

}

.action-btn:hover {

transform: translateY(-2px);

box-shadow: 0 6px 25px rgba(0, 0, 0, 0.2);

}

.action-btn:active {

transform: translateY(0);

}

/\* Status-specific styles \*/

.fraud-result {

--status-color: #E74C3C;

}

.warning-result {

--status-color: #F39C12;

}

.safe-result {

--status-color: #27AE60;

}

.fraud-result .result-title,

.warning-result .result-title,

.safe-result .result-title {

color: var(--status-color);

}

.fraud-result .fraud-score,

.warning-result .fraud-score,

.safe-result .fraud-score {

border-left-color: var(--status-color);

}

.fraud-result .score-value,

.warning-result .score-value,

.safe-result .score-value {

color: var(--status-color);

}

/\* Floating animations \*/

@keyframes float {

0%, 100% { transform: translateY(0px) rotate(0deg); }

50% { transform: translateY(-8px) rotate(5deg); }

}

@keyframes pulse {

0%, 100% { opacity: 0.3; }

50% { opacity: 0.6; }

}

@keyframes slideIn {

from {

opacity: 0;

transform: translateY(30px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

.gear-left, .gear-right {

animation: float 4s ease-in-out infinite;

}

.gear-right {

animation-delay: 1s;

}

.hacker-silhouette {

animation: pulse 3s ease-in-out infinite;

}

.result-container {

animation: slideIn 0.8s ease-out;

}

/\* Responsive design \*/

@media (max-width: 768px) {

.header {

flex-direction: column;

gap: 20px;

text-align: center;

}

.main-title {

font-size: 2rem;

}

.result-container {

padding: 30px 20px;

margin: 20px;

}

.result-title {

font-size: 1.8rem;

}

.result-description {

font-size: 1.1rem;

}

.action-buttons {

flex-direction: column;

align-items: center;

}

.action-btn {

width: 100%;

max-width: 250px;

}

.nav-button {

padding: 10px 20px;

font-size: 14px;

}

}

@media (max-width: 480px) {

.main-title {

font-size: 1.8rem;

}

.result-container {

padding: 25px 15px;

}

.result-title {

font-size: 1.5rem;

}

.result-description {

font-size: 1rem;

}

}

</style>

</head>

<body>

<!-- Background decorative elements -->

<div class="bg-decoration hacker-silhouette"></div>

<div class="bg-decoration gear-left"></div>

<div class="bg-decoration gear-right"></div>

<div class="bg-decoration chart-bars">

<div class="bar"></div>

<div class="bar"></div>

<div class="bar"></div>

</div>

<div class="bg-decoration credit-card"></div>

<div class="bg-decoration shield"></div>

<div class="bg-decoration magnifying-glass"></div>

<div class="container">

<!-- Header -->

<div class="header">

<h1 class="main-title">Online Payments Fraud Detection</h1>

<div class="nav-buttons">

<a href="home.html" class="nav-button home-btn">Home</a>

<a href="predict.html" class="nav-button predict-btn">Predict</a>

</div>

</div>

<!-- Result Container -->

<div class="result-container" id="resultContainer">

<div class="result-icon" id="resultIcon">🔍</div>

<h2 class="result-title" id="resultTitle">Transaction Analysis Complete</h2>

<p class="result-description" id="resultDescription">

The fraud detection system has analyzed the transaction data.

</p>

<!-- Fraud Score Display -->

<div class="fraud-score">

<div class="score-label">Fraud Risk Score</div>

<div class="score-value" id="fraudScore">0 / 100</div>

</div>

<!-- Recommendations -->

<div class="recommendations" id="recommendations">

<h3>Recommendations</h3>

<ul id="recommendationsList">

<li>Review transaction details carefully</li>

<li>Monitor account activity regularly</li>

<li>Enable transaction notifications</li>

</ul>

</div>

<div class="action-buttons">

<a href="predict.html" class="action-btn new-prediction-btn">New Prediction</a>

<a href="home.html" class="action-btn home-redirect-btn">Back to Home</a>

</div>

</div>

</div>

<script>

document.addEventListener('DOMContentLoaded', function() {

// Get URL parameters

const urlParams = new URLSearchParams(window.location.search);

const fraudResult = urlParams.get('result') || 'safe';

const fraudScore = parseInt(urlParams.get('score')) || 0;

// Get DOM elements

const resultContainer = document.getElementById('resultContainer');

const resultIcon = document.getElementById('resultIcon');

const resultTitle = document.getElementById('resultTitle');

const resultDescription = document.getElementById('resultDescription');

const fraudScoreElement = document.getElementById('fraudScore');

const recommendationsList = document.getElementById('recommendationsList');

// Update fraud score

fraudScoreElement.textContent = fraudScore + ' / 100';

// Configure result based on fraud status

if (fraudResult === 'fraud') {

resultContainer.className = 'result-container fraud-result';

resultIcon.textContent = '⚠️';

resultTitle.textContent = 'FRAUD DETECTED';

resultDescription.textContent = 'This transaction shows high risk indicators and may be fraudulent. Immediate action is recommended.';

recommendationsList.innerHTML = `

<li>Block this transaction immediately</li>

<li>Contact your bank or financial institution</li>

<li>Review recent account activity</li>

<li>Change account passwords and PINs</li>

<li>Monitor credit reports for suspicious activity</li>

<li>Consider freezing your credit temporarily</li>

`;

} else if (fraudResult === 'warning') {

resultContainer.className = 'result-container warning-result';

resultIcon.textContent = '⚡';

resultTitle.textContent = 'REQUIRES REVIEW';

resultDescription.textContent = 'This transaction has some suspicious characteristics. Additional verification is recommended.';

recommendationsList.innerHTML = `

<li>Verify transaction details with the recipient</li>

<li>Check for unusual transaction patterns</li>

<li>Enable additional security measures</li>

<li>Monitor account closely for 24-48 hours</li>

<li>Contact bank if you notice any irregularities</li>

<li>Keep transaction records for reference</li>

`;

} else {

resultContainer.className = 'result-container safe-result';

resultIcon.textContent = '✅';

resultTitle.textContent = 'TRANSACTION SAFE';

resultDescription.textContent = 'This transaction appears to be legitimate and shows no significant fraud indicators.';

recommendationsList.innerHTML = `

<li>Transaction can proceed normally</li>

<li>Continue regular account monitoring</li>

<li>Keep transaction receipt for records</li>

<li>Maintain good security practices</li>

<li>Report any unusual activity promptly</li>

<li>Review monthly statements regularly</li>

`;

}

// Add click effects to buttons

const buttons = document.querySelectorAll('.action-btn');

buttons.forEach(button => {

button.addEventListener('click', function(e) {

this.style.transform = 'scale(0.95)';

setTimeout(() => {

this.style.transform = '';

}, 150);

});

});

// Add hover effects to navigation buttons

const navButtons = document.querySelectorAll('.nav-button');

navButtons.forEach(button => {

button.addEventListener('mouseenter', function() {

this.style.transform = 'translateY(-2px)';

});

button.addEventListener('mouseleave', function() {

this.style.transform = 'translateY(0)';

});

});

});

</script>

</body>

</html>

**Running the Application**

1.Navigate to the project directory

2.Install packages

pip install -r requirements.txt

1. Run the Flask application:

python app.py

1. Open the web browser and go to [**http://127.0.0.1:5000/**](http://127.0.0.1:5000/) **t**o see the web interface and make predictions.