**Milestone-3**

Requirements Document for Fraud Detection System UI

### 1. Introduction

The Fraud Detection System UI provides a user-friendly and professional interface for detecting potentially fraudulent transactions. It integrates a machine learning model to analyze and predict fraudulent activities in both individual and batch transactions. The interface is designed for ease of use, with clear navigation and visually appealing elements.

### 2. Functional Requirements

#### 2.1 Navigation Menu

- \*Purpose\*: Ensure easy access to all key sections of the app.

- \*Menu Options\*:

- Home

- Single-Predict

- File-Predict

- History

- About

- \*Icons\*: Use appropriate icons to visually represent each menu option.

- \*Design\*: Horizontal orientation with a clear indicator for the active tab.

#### 2.2 Home Page

- \*Header\*: Display a prominent title, “Online Payments Fraud Detection System,” centered at the top.

- \*Image Carousel\*:

- Rotate through pre-defined images relevant to online payment security.

- Automatic cycling with a configurable refresh interval.

- \*Feature Highlights\*:

- Real-time Fraud Detection

- User-Friendly Interface

- Advanced Machine Learning

- Prediction History

- \*Layout\*: Present features in well-organized, visually separated columns.

### 3. Single Transaction Prediction Page

- \*Form Inputs\*:

- Transaction Type (Select box with options such as CASH\_IN, CASH\_OUT, DEBIT, PAYMENT, TRANSFER)

- Amount (Numeric input field)

- Old Balance Original (Numeric input field)

- New Balance Original (Numeric input field)

- \*Validation\*:

- Ensure all required fields are completed before submission.

- Display error messages in red if fields are incomplete.

- \*Prediction Display\*:

- Display results in green for non-fraudulent predictions and red for fraudulent ones.

- \*Layout\*:

- Use two columns for input fields to ensure a balanced appearance.

### 4. Batch Transaction Prediction Page

- \*File Upload\*:

- Accept only CSV files.

- Validate the CSV structure to match the expected format (columns: type, amount, oldbalanceOrg, newbalanceOrig).

- \*File Processing\*:

- Display a preview of the uploaded data.

- Show the number of records and attributes in the file.

- \*Output\*:

- Display predictions in a table format.

- Include a download button for the processed CSV file with results.

### 5. History Page

- \*Content\*:

- Display a table containing past prediction records stored in session state.

- Include a download button to export history as a CSV file.

- \*Design\*:

- Use a clear, readable table format with alternating row colors.

### 6. About Page

- \*Content\*:

- Provide a detailed explanation of how to use the application and each feature.

- Include an overview of the machine learning model used (e.g., Random Forest algorithm).

- \*Layout\*:

- Use columns and visuals (e.g., images, icons) to enhance readability.

### 7. Non-Functional Requirements

- \*Performance\*:

- Ensure quick page loading and responsive interactions.

- \*Usability\*:

- Maintain a consistent and user-friendly design.

- Ensure all form labels and buttons are intuitive and descriptive.

- \*Accessibility\*:

- Use high-contrast text and distinguishable interactive elements.

- \*Scalability\*:

- Design the UI to accommodate future expansions (e.g., adding more transaction types or prediction options).

### 8. Technical Requirements

- \*Framework\*: Streamlit

- \*Libraries\*:

- streamlit\_option\_menu

- numpy

- pandas

- pickle

- PIL (Python Imaging Library)

- base64

- \*Styling\*: Custom CSS for enhanced UI aesthetics.

### 9. UI Styling Details

- \*Primary Color Scheme\*:

- Main theme color: #2874A6 (blue)

- Button hover color: #1B4F72

- \*Fonts\*:

- Use standard web-safe fonts such as Arial or Helvetica.

- \*Components\*:

- Styled buttons, form fields, and alert messages for better user experience.

### 10. Security Requirements

- \*Model Security\*:

- Ensure the predictive model is loaded securely without exposing sensitive data.

- \*File Upload Safety\*:

- Validate and sanitize all uploaded files to prevent code injection or malicious activities.

### 11. Future Enhancements

- \*User Authentication\*: Implement user login/logout functionality for enhanced security.

- \*Analytics Dashboard\*: Develop a dashboard for summarizing and visualizing historical prediction trends.