

## Proposed Solution for Message Classification and User Experience Enhancement

---

### 1. Introduction

This document presents a comprehensive AI/ML-powered solution designed to improve the classification and management of user messages. By categorizing messages into types such as spam, fraud, OTP, logistics, and transactional, the system aims to:

- Detect and mitigate malicious or fraudulent content.
  - Enhance user experience through actionable features like OTP auto-deletion and real-time logistics tracking.
  - Provide personalized recommendations based on user behavior.
- 

### 2. Solution Overview

#### Key Features

##### 1. Message Classification

- Categorizes messages into predefined categories (spam, fraud, OTP, etc.).
- Identifies suspicious or malicious content.

##### 2. User Interface Enhancements

- Highlights OTPs for quick access and deletes them after expiry.
- Provides real-time tracking for logistics messages within the app.
- Enables direct payment options for transactional messages.

##### 3. Behavioral Insights

- Tags users based on activities (e.g., frequent payers, shoppers).
- Suggests tailored promotions or services.

#### System Architecture

- **Input:** Raw text messages received by users.
  - **Processing:** AI/ML models classify the messages based on their content.
  - **Output:** Categorized messages are displayed with actionable buttons or suggestions.
- 

### 3. AI/ML Models and Techniques

#### Model Selection

- **Text Classification Model:** Uses supervised learning algorithms like Logistic Regression, Random Forest, or Transformer-based models (e.g., BERT) for accurate classification.
- **Feature Extraction:**
  - **TF-IDF:** Converts text into numerical features for traditional ML models.
  - **Embeddings:** Leverages contextual embeddings for deep learning models.

## Steps in the Process

### 1. Data Preparation

- Preprocess messages: Clean text, remove stop words, tokenize, and normalize.
- Label data into categories such as spam, fraud, OTP, etc.

### 2. Model Training

- Train the selected algorithm on labeled data to predict message categories.
- Use evaluation metrics like accuracy, precision, recall, and F1-score to measure performance.

### 3. Integration

- Develop a REST API to integrate the classification model with the app interface.

---

## 4. Use Cases and User Workflows

### Use Case 1: OTP Messages

- **Scenario:** A user receives an OTP.
- **Workflow:**
  1. The system detects and highlights the OTP.
  2. A "Copy OTP" button is displayed for user convenience.
  3. The message is auto-deleted after a predefined expiration time.

### Use Case 2: Logistics Messages

- **Scenario:** A user receives a package tracking update.
- **Workflow:**
  1. The system identifies the message as a logistics update.
  2. A "Track Order" button is displayed in the app.
  3. The user can view real-time tracking without leaving the app.

### Use Case 3: Transactional Messages

- **Scenario:** A user receives a bill payment reminder.
  - **Workflow:**
    1. The system classifies the message as transactional.
    2. A "Pay Now" button is displayed with integrated payment options.
    3. The user completes the transaction directly within the app.
- 

### 5. Benefits

- **Efficiency:** Automates message classification and improves user interaction.
  - **Convenience:** Provides actionable features like "Pay Now" and "Track Order."
  - **Personalization:** Tailors recommendations and offers based on user behavior.
- 

### 6. Next Steps

1. Enhance the dataset for better model accuracy.
2. Explore advanced AI techniques like deep learning for improved classification.
3. Test the system with real-world data to refine workflows and usability.