

# Online Payments Fraud Detection using Machine Learning




## Project Documentation format

### 1. Introduction

**Project Title:** Online Payments Fraud Detection using Machine Learning

- **Team Members** R VISHNU VARDHAN REDDY  
G. RAKESH SAI  
J.VISHNU VARDHAN REDDY  
G.SAI SUNDEEP.

### 2. Project Overview

- **Purpose:** The purpose of the **Online Payment Fraud Detection System** is to detect and prevent fraudulent online transactions using machine learning and rule-based validation techniques.. •
- **Features Real-Time Fraud Detection** – Analyzes transactions instantly.
- ☐ **Machine Learning Integration** – Classifies transactions as legitimate or fraudulent.
-  **Dashboard Analytics** – Displays fraud statistics and reports.
-  **User Authentication** – Secure login and registration system.
- ☐ **Transaction History** – Stores and tracks user payment records.
-  **Fraud Alerts** – Notifies admin or user about suspicious activity.

### 3. Architecture

**Frontend:** The frontend is developed using **React.js** and follows a component-based.

- **Backend** The backend is developed using **Node.js** with **Express.js** framework.

**Database:** MongoDB is used to store users and transaction data..

### 4. Setup Instructions

- **Prerequisites:** List software dependencies (e.g., Node.js, MongoDB). •
- Installation:** Step-by-step guide to clone, install dependencies, and set up the environment variables.

### 5. Folder Structure

- **Client:** Describe the structure of the React frontend.
- **Server:** Explain the organization of the Node.js backend.

### 6. Running the Application

- Provide commands to start the frontend and backend servers locally. ○
  - **Frontend:** `npm start` in the client directory.
  - **Backend:** `npm start` in the server directory.

## 7. API Documentation

- Document all endpoints exposed by the backend.
- Include request methods, parameters, and example responses.

## 8. Authentication

- Explain how authentication and authorization are handled in the project. •  
Include details about tokens, sessions, or any other methods used.

## **9. User Interface**

- Provide screenshots or GIFs showcasing different UI features.

## **10. Testing**

- Describe the testing strategy and tools used.

## **11. Screenshots or Demo**

- Provide screenshots or a link to a demo to showcase the application.

## **12. Known Issues**

- Document any known bugs or issues that users or developers should be aware of.

## **13. Future Enhancements**

- Outline potential future features or improvements that could be made to the project.