EtherGuard: Anti-Money Laundering System for Ethereum

User Guide & Manual

📑 Table of Contents

🔷 Introduction

EtherGuard is a blockchain-powered compliance and monitoring system designed to help financial institutions, cryptocurrency exchanges, and compliance teams detect and prevent **money laundering on the Ethereum network**.

With EtherGuard, you can:

- Monitor Ethereum transactions for suspicious behavior
- Track high-risk wallet addresses in real time

- Maintain AML-focused watchlists
- Process **KYC (Know Your Customer)** verifications
- Generate **regulatory-ready reports** for auditors and compliance teams

This guide walks you through the system's key features, setup, and day-to-day usage.

🔷 System Overview

EtherGuard is made up of several integrated modules, each supporting a different part of AML compliance:

1. **Dashboard** – At-a-glance view of suspicious activity, alerts, and system metrics.
2. **Wallet Tracking** – Investigate Ethereum addresses and analyze their transaction history.
3. **Watchlist** – Maintain continuous monitoring of flagged wallets.
4. **KYC Verification (Admin)** – Review and process customer verification requests.
5. **System Logs (Admin)** – Audit user activity and system events.
6. **Report Generation** – Export structured compliance reports (PDF).

🔍 Detection Mechanisms

EtherGuard uses **rule-based logic** to identify suspicious activity, including:

- **Large Transactions** → Transfers exceeding **10 ETH**
- **Suspicious Counterparties** → Interaction with blacklisted or high-risk addresses
- **High Frequency** → More than 10 transactions in a single hour
- **KYC Risk Scoring** → Automated scoring of user-provided identity data

---

🔷 Getting Started

System Requirements

- **Python 3.x**

- **Flask**
- **SQLite**
- **Web3.py** (Ethereum integration)
- **ReportLab** (PDF generation)
- Any modern browser (Chrome, Firefox, Safari, Edge)

Installation Steps

1. **Download Project**

   ```
   cd c:\Users\GPU Tech\Desktop\fyp
   ```

2. **Install Dependencies**

   ```
   pip install flask flask-sqlalchemy web3 reportlab requests
   ```

3. **Initialize Database**

   ```
   python init_db.py
   ```

4. **Create Admin User (Optional)**

   ```
   python make_admin.py
   ```

5. **Run Application**

   ```
   python app.py
   ```

6. Open `http://localhost:5000` in your browser.

First Login

- **Regular Users** → Sign up through the web portal.
- **Admin Access** → Use the admin account created via `make_admin.py`.

🔷 Dashboard

The **Dashboard** provides a high-level overview of Ethereum activity being monitored.

Key Features

- **KPI Cards** – Suspicious transactions, high-risk wallets, SARs filed
- **Risk Trends** – Charts showing changes in activity over time
- **Recent Alerts** – A quick snapshot of the latest flagged events
- **Transaction Volume Analysis** – Identify unusual spikes in transaction activity

How to Use:

1. Open **Dashboard** from the sidebar.
2. Review the KPIs and charts for quick insights.
3. Investigate recent alerts for immediate action.

🔷 Wallet Tracking

Track and analyze Ethereum wallet addresses in detail.

Features

- **Search any Ethereum address**
- **View full transaction history**
- **Flag suspicious behavior automatically**
- **Export findings into PDF reports**

Detection Criteria

- Transactions > 10 ETH
- Transactions with flagged addresses
- More than 10 transactions in one hour

How to Use:

1. Open **Track Wallet** from the sidebar.
2. Enter the Ethereum address.

3. Click **Track Wallet** to view its profile.
4. Export a compliance-ready PDF if required.

🔷 Watchlist

Keep continuous monitoring on selected wallet addresses.

Features

- Add or remove addresses from the watchlist
- Balance distribution chart across monitored wallets
- Automatic flagging of suspicious activity
- Real-time alerts

How to Use:

1. Go to **Watchlist** from the sidebar.
2. Add addresses of interest.
3. Monitor transaction alerts and system updates automatically.

🔷 KYC Verification (Admin)

Admins can review **identity verification requests**.

Features

- Stats for pending, approved, and rejected requests
- Review uploaded ID documents
- Automated **risk scoring** based on data completeness

Risk Scoring Rules

- Missing ID document → +50 points
- Short/invalid ID number → +30 points
- Missing Date of Birth → +20 points
- **Max score: 100 (high risk)**

🔷 System Logs (Admin)

Audit system activity for compliance and security.

Features

- Full activity logs (user & system)
- Log filtering by date, user, or severity
- Export logs for audits

🔷 User Account Management

Users can manage their own profile and security.

Features

- Update personal profile details
- Configure email alert settings
- Submit KYC documents
- Reset/change passwords

🔷 Report Generation

Generate structured PDF reports for compliance documentation.

Available Reports

- **Wallet Report** – Activity of a specific Ethereum address
- **Flagged Transactions Report** – List of suspicious activities
- **Watchlist Report** – Overview of monitored addresses
- **KYC Verification Report** – Status of verification requests

---

🔷 Technical Architecture

Frontend: HTML, CSS, JS, Bootstrap, Chart.js

Backend: Python Flask

Database: SQLite + SQLAlchemy

Blockchain Integration: Web3.py (Ethereum)

Reports: ReportLab (PDFs)

🔷 Troubleshooting

Issue	Possible Fix
Transaction data not loading	Check internet connection & API status
PDF not generating	Verify transaction history exists
KYC document upload failing	Ensure file < 5MB, valid format (JPG/PNG/PDF)
Login issues	Clear browser cache, reset password
Database errors	Re-run `python init_db.py` , check file permissions

📌 Support

For technical support, contact the system administrator or refer to the **developer documentation**.
