**Payment Anomaly Detection Project - Context Document**

**Project Overview**

**Project Name**: Payment Anomaly Detection MVP  
**Developer**: Gaurav (AI Solution Architect)  
**Company**: BranchSpace  
**Client**: British Airways  
**Domain**: Digital Payments Settlement & Reconciliation

**Current Status**

* ✅ **GitHub Repository**: <https://github.com/gaurav0325/payment-anomaly-mvp>
* ✅ **Live Deployment**: https://[your-app-name].streamlit.app
* ✅ **Local Development**: Running on localhost:8502
* ✅ **Technology Stack**: Streamlit, Python, Pandas, Plotly, scikit-learn

**Project Architecture**

**Tech Stack**

* **Frontend**: Streamlit (cloud-deployed)
* **ML Algorithm**: Isolation Forest (statistical + planned upgrade)
* **Data Processing**: Pandas, NumPy
* **Visualization**: Plotly, Seaborn
* **Version Control**: GitHub
* **Deployment**: Streamlit Cloud
* **Local Development**: Windows, PowerShell, VS Code (planned)

**File Structure**

payment-anomaly-mvp/

├── streamlit\_app.py # Main Streamlit application

├── requirements.txt # Dependencies

├── README.md # Project documentation

├── data/

│ └── sample\_data/

│ └── dart\_312\_313\_sample.csv # Sample payment data

├── src/

│ ├── data\_processor.py # Data processing logic

│ ├── ml\_engine.py # ML model implementation

│ └── anomaly\_detector.py # Anomaly detection algorithms

└── .gitignore

**Key Features Implemented**

1. **Dashboard**: Interactive Streamlit web interface
2. **Anomaly Detection**: Multi-dimensional analysis
   * Amount anomalies (IQR method)
   * Settlement delays (>7 days)
   * Reconciliation issues (unmatched)
   * Timing anomalies (weekend/off-hours)
3. **Severity Classification**: LOW, MEDIUM, HIGH
4. **Filtering**: By severity, merchant, date range
5. **Visualization**: Plotly charts, timeline analysis
6. **Export**: CSV download functionality
7. **Sample Data**: DART 312/313 compliant synthetic data

**Data Schema (DART 312/313 Format)**

transaction\_id: Unique identifier

timestamp: Transaction timestamp

amount: Transaction amount (GBP)

currency: Currency code

merchant\_id: Merchant identifier

payment\_method: CARD/BANK\_TRANSFER/DIGITAL\_WALLET/DIRECT\_DEBIT

settlement\_date: Expected settlement date

actual\_settlement\_date: Actual settlement date

reconciliation\_status: MATCHED/UNMATCHED

processing\_fee: Associated fees

reference\_number: External reference

settlement\_delay\_days: Calculated delay

**Current Anomaly Detection Logic**

python

*# Amount anomalies: IQR method*

Q1 = df['amount'].quantile(0.25)

Q3 = df['amount'].quantile(0.75)

IQR = Q3 - Q1

anomaly = (amount < Q1 - 1.5\*IQR) | (amount > Q3 + 1.5\*IQR)

*# Settlement delays: >7 days*

settlement\_anomaly = settlement\_delay\_days > 7

*# Reconciliation issues: unmatched status*

reconciliation\_anomaly = status == 'UNMATCHED'

*# Timing anomalies: weekend processing*

timing\_anomaly = day\_of\_week in [5, 6]

**Development Environment**

* **OS**: Windows
* **IDE**: VS Code (planned setup)
* **Python**: Virtual environment (venv)
* **Local Port**: 8502 (avoiding conflict with 8501)
* **Git**: GitHub integration

**Recent Achievements**

1. **Deployed to Streamlit Cloud**: Live public URL
2. **GitHub Repository**: Version controlled
3. **Local Development**: Running on port 8502
4. **Professional Dashboard**: Interactive anomaly detection

**Next Steps Discussed**

1. **Enhanced ML**: Real Isolation Forest implementation
2. **Real-time Features**: Auto-refresh, alerts
3. **Advanced Analytics**: Predictive trends, merchant risk scoring
4. **API Integration**: RESTful endpoints
5. **Professional Features**: Authentication, configuration

**Business Value**

* **Early Detection**: Identify payment anomalies before impact
* **Operational Efficiency**: Reduce manual reconciliation by 70%
* **Risk Mitigation**: Catch fraud and system errors
* **Career Advancement**: Showcase AI architecture expertise

**Common Commands**

bash

*# Local development*

cd "C:\Gaurav\Gaurav CV et al\AI project\3rd project\_anomalies in S and R\Claude\payment-anomaly-mvp"

venv\Scripts\activate

streamlit run streamlit\_app.py --server.port 8502

*# Git workflow*

git add .

git commit -m "Enhancement description"

git push origin main

*# VS Code (when setup)*

code .

**Key Contacts & Context**

* **Purpose**: Portfolio project for AI Architect positioning
* **Target Audience**: British Airways, payment industry
* **Goal**: Demonstrate end-to-end AI solution capability
* **Timeline**: MVP completed, enhancement phase ongoing

**Troubleshooting Notes**

* **Port conflicts**: Use 8502 instead of 8501
* **VS Code setup**: Need to install and add to PATH
* **GitHub auth**: Already configured for gaurav0325
* **Streamlit Cloud**: Auto-deploys on git push

**Project Unique Identifiers**

* **GitHub**: gaurav0325/payment-anomaly-mvp
* **Domain**: Payment anomaly detection
* **ML Algorithm**: Isolation Forest
* **Data Format**: DART 312/313
* **Client**: British Airways via BranchSpace

**Instructions for Claude**: This is Gaurav's payment anomaly detection project. He's an experienced IT Solution Architect transitioning to AI Architect role, focusing on digital payments for British Airways through BranchSpace. The project demonstrates end-to-end AI/ML capability from data processing to cloud deployment. Use professional, direct communication style with technical depth.