

Automated AI-Based Financial Transaction Categorisation

1. Technology Stack

- Python – Core development and ML pipeline
- Scikit-learn / LightGBM / TensorFlow – Model training & inference
- FastAPI / Flask – API for integration
- SQLite / PostgreSQL – Transaction, feedback, and logs storage
- JSON / YAML – Dynamic category taxonomy configuration
- SHAP / LIME – Explainability of predictions
- Docker / Kubernetes – Deployment & scalability

2. System Architecture

Modules Included

- **Input Preprocessing Layer**
 - Cleans raw transaction text
 - Removes noise, punctuation, casing
 - Expands abbreviations (e.g., “MC DLD” → “McDonald's”)
- **Feature Encoding Layer**
 - TF-IDF vectors or
 - Lightweight transformer embeddings
- **Classification Layer**
 - LightGBM model or shallow neural network
 - Outputs probability + predicted category
- **Config Manager**
 - Reads category taxonomy from JSON/YAML
 - Allows updates without code changes
- **Inference Pipeline**
 - Generates prediction + confidence score
 - Logs all predictions
 - Routes low-confidence results for human review
- **Explainability Module**
 - SHAP/LIME identifies influential words
 - Display top tokens that drive model decisions
- **Feedback Loop & Retraining**
 - Users correct wrong predictions
 - Corrections stored for incremental retraining

3. Data Model & Storage Tables

Transaction Table

- transaction_id
- raw_text
- cleaned_text
- predicted_category
- confidence_score
- timestamp

Category Config File (JSON/YAML)

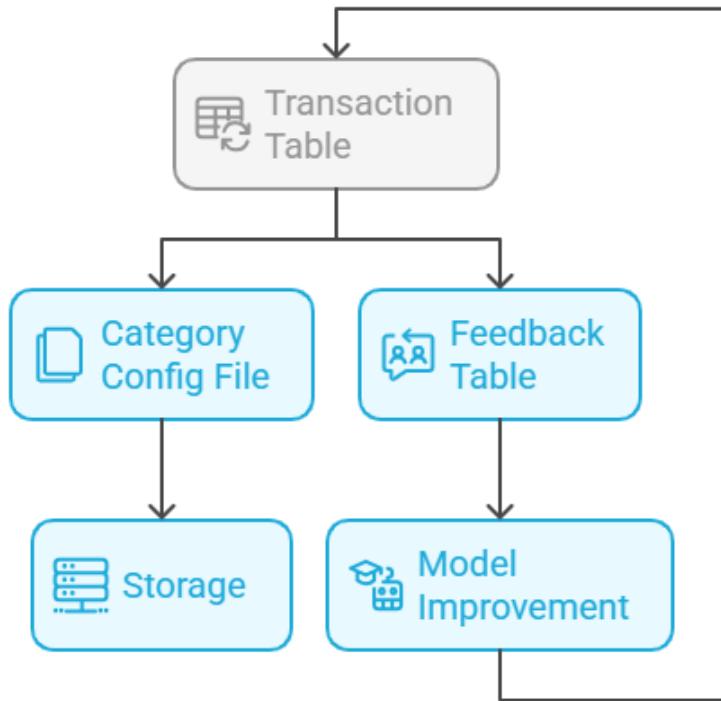
- List of categories
- Keywords / patterns (optional)
- Hierarchical structure (optional)

Feedback Table

- transaction_id
- model_prediction
- user_corrected_category
- review_status
- timestamp

Storage

- Local: SQLite
- Production: PostgreSQL
- Config: JSON/YAML
- Logs: Flat files / DB logs



4. AI / ML / Automation Components

Preprocessing

- Lowercasing
- Stopword removal
- Tokenization
- Abbreviation expansion
- Noise removal (extra spaces, numbers, symbols)

Feature Engineering

- TF-IDF for lightweight, fast inference
- Transformer embeddings for richer semantic understanding

Model

- Supervised ML classifier
- Options: LightGBM, Logistic Regression, Neural Network

Inference

- Category + Confidence score
- Threshold-based alerts for low confidence

Automation

- Auto-logging of predictions
- Scheduled retraining using feedback
- Dynamic taxonomy updates

5. Security & Compliance

- Entire inference pipeline runs locally → No external API calls
- Encryption
 - At rest (database encryption)
 - In transit (HTTPS/SSL)
- Role-Based Access Control (RBAC) for admin & reviewers
- GDPR-compliant design
 - User data can be deleted on request
 - Audit logs stored securely
- Explainable AI ensures transparency
- No third-party data leakage



6. Scalability & Performance

- Works in real-time for single transactions
- Supports batch processing for thousands of records
- Lightweight embeddings = fast CPU-only inference
- Horizontal scaling with Docker + Kubernetes
- Configurable pipeline for:
 - Model versioning
 - Rolling updates
 - A/B testing



7. Links

GitHub Repository:

<https://github.com/Esssp/inhouseTX-Tech4Good.git>

Demo Video (YouTube / Drive):

https://drive.google.com/file/d/1OhwEUuJpmMSlC0tTx2R_fZmNFkhgerKv/view?usp=sharing

8. Conclusion

The InHouseTX system is a fully local, transparent, and cost-efficient AI solution for financial transaction categorisation. It eliminates dependence on paid APIs, improves privacy, and allows quick customization of category taxonomies. With explainability, feedback loops, and scalable architecture, it is suitable for both academic demonstration and real-world financial applications.