

AI SHOPPING ASSISTANT

**Integrating Azure Speech & Document Intelligence for
Automated Expense Tracking**

PRESENTED BY

STUDENT NAME: PRANAY MAKTHALA

**COLLEGE NAME: GURUNANAK INSTITUTE OF
TECHNOLOGY**

**DEPARTMENT: ELECTRONICS AND
COMMUNICATION**

EMAIL ID: PRANAYMAKTHALA@GMAIL.COM



OUTLINE:

- **Problem Statement**
- **Proposed System/Solution**
- **System Development Approach**
- **Algorithm & Deployment**
- **Result (Output Image)**
- **Conclusion**
- **Future Scope**
- **References**

PROBLEM STATEMENT:

- **Manual Data Entry Fatigue**
- Typing every item, price, and tax into budgeting apps is tedious and inconsistent.
- **Lack of Real-time Verification**
- Shopping lists rarely confirm if an item was purchased or if the paid price changed at checkout.
- **Unstructured Receipt Data**
- Paper receipts lock critical fields (GST, unit price, quantity) in images that aren't analyzed for budgeting.
- **Accessibility Barriers**
- Constant typing is not ideal for users on the go or those needing hands-free interaction.

PROPOSED SOLUTION:

- Build a voice-first, receipt-aware assistant that digitizes receipts, cross-references planned lists, and stores structured expense data for analytics.

Voice-First Interaction

“Voice Add” via Azure Speech SDK for hands-free list creation and rapid item capture.

Automated Receipt Digitization

Azure Document Intelligence (prebuilt-receipt model) converts images into structured item lists and key-value pairs.

Intelligent Cross-Referencing

Backend logic auto-checks planned items by matching scanned receipt entries to spoken/listed items.

Centralized Cloud Analytics

All processed receipts and voice logs are stored in Azure PostgreSQL to generate monthly expense reports in ₹.

SYSTEM APPROACH:

Backend Architecture

Spring Boot 4.0.3 + Java 25 using
Spring Web MVC to expose RESTful
APIs for voice, image upload, and
matching services.

Maven manages dependencies
(Azure SDKs, Lombok, JPA).

Cloud Intelligence & Data

Azure AI Speech for real-time
microphone streams and Neural TTS.
Azure Document Intelligence (prebuilt-
receipt) for OCR & KV extraction.
Azure Database for PostgreSQL
(flexible server) for relational
persistence and analytical queries

Data Pipeline & Preprocessing

1. Capture

Photo uploaded as MultipartFile; voice captured via microphone stream.

2. Convert & Analyze

Image → BinaryData for Azure SDK; SyncPoller runs prebuilt receipt model to extract item lists and totals.

3. Store & Index

Parsed fields inserted into PostgreSQL with normalized item, price, qty, and GST columns.

4. Match & Update

String-matching (ContainingIgnoreCase) aligns spoken items with receipt descriptions and flips checked status when matched.

ALGORITHM & DEPLOYMENT:

Algorithm Selection

Use rule-based string matching augmented by fuzzy matching heuristics to reconcile spoken names with OCR text.

Data Inputs

Features: item name, unit price, quantity, GST, timestamp, store metadata, day-of-week, and optional location.

Training & Tuning

Not model-heavy; tune matching thresholds and normalize tokens (stopwords, punctuation, transliterations).

Precision Handling

Java wrapper classes with `.intValue()` / `.doubleValue()` preserve decimal precision for accurate GST and totals.

Algorithm Selection

Use rule-based string matching augmented by fuzzy matching heuristics to reconcile spoken names with OCR text.

RESULT:

Voice Capture

Neural models achieved 100% accuracy for single-item name captures in controlled tests.

OCR Extraction

High-fidelity extraction of complex lines, e.g., "Chocolate — ₹979.85" and "Notebook — ₹68.67", including GST parsing.

Auto-Check Integration

System reliably flips checked status from false → true upon matching receipt entries to planned items.

Final Aggregation

Generated formatted summaries: "Total Amount Spent: ₹1048.52", aggregating prices and taxes across receipts.

CONCLUSION:

The integration of Azure's speech and document AI with a Spring Boot backend reduces friction in expense tracking by transforming manual entry into an automated, accurate workflow. The system improved data integrity, speed, and accessibility for on-the-go users.

Key Impact

Fewer manual errors, faster receipt processing, and hands-free interaction improve adoption and budgeting reliability.

FUTURE SCOPE:

Threshold Alerts

Push notifications when users exceed monthly budgets (example threshold: ₹5000).

Store Price Comparison

Aggregate receipts to suggest lowest-price stores for frequently purchased items.

Frontend Integration

Build React or Flutter mobile UI for the API to provide a polished user experience.

Edge & Advanced ML

Explore on-device preprocessing and advanced ML for offline OCR and privacy-preserving analytics.

REFERENCES:

- **References**

- Microsoft Azure SDK for Java — Documentation and examples
- Spring Boot Reference Guide
- Azure AI Document Intelligence — prebuilt-receipt model docs
- AICTE Internship Project Guidelines

- **Contact**

- Email: Pranay Makthala
- Department: Electronics and Communication
- GitHub Link : <https://github.com/PranayChary-M/AI-Shopping-Assistant>

Thank You