BUSINESS REQUIREMENT DOCUMENT (BRD)

Project: A2B Restaurant Billing Automation System

1.Business Objective:

To build a Python-based billing and management system for A2B's restaurant chain, aiming to automate the manual billing process across all branches. The system will support both offline (branch) and online (home) operations with data synchronization to the central server when internet becomes available. This will:

- Increase billing speed and accuracy
- Provide real-time reports to head office
- Maintain operational continuity even in offline branches
- Support scalability for new branches

2.Goals:

- Build a robust **Point-of-Sale (POS)** system using Python.
- Enable **offline-first** operations with **local database (MySQL)** at branches.
- Sync data periodically to the **central home branch** server.
- Integrate **menu management**, **order processing**, and **billing**.
- Provide **dashboard reporting**, **inventory monitoring**, and **real-time analytics**.

3.Modules Overview:

A. Menu Management (Branch + Admin)

Business Description: Manage food items and pricing.

- Add/edit/delete menu items
- Update prices & availability
- Local CSV or MySQL storage

- Sync with central menu

Inspired by CLI-based module in the training system; implemented via Django admin panel or local UI.

B. Order & Billing System (POS)

Business Description: Take orders, calculate totals, apply discounts, and generate bills.

- Order input by item code/name
- Calculate subtotal, tax, and final bill
- Print/log bills locally
- Store orders in local DB (branch)
- REST API support for remote sync

Advanced version of CLI logic from the PDF, adapted for GUI with Django backend.

C. Offline Functionality

Business Description: Allow all operations without internet.

- Local hosting of UI and DB (MySQL)
- Log transactions locally
- Sync component to push/pull data

D. Data Synchronization Service

Business Description: Sync local data to central home server.

- Triggered when internet is available
- Fetch new menu updates, push new orders
- Schedule or real-time push
- Written in Python (Django REST/Node.js)

E. Admin Dashboard (Home Branch)

Business Description: Centralized monitoring and control.

- Monitor branch sales
- Review synchronized orders
- Update central menu
- Visual dashboards (Power BI)

F. Reporting & Analytics

Business Description: Analyze operations across branches.

- Daily/weekly/monthly sales
- Top-selling items
- Inventory depletion rate
- Branch-wise performance

G. User & Role Management

Business Description: Control user access and roles.

- Secure login system
- Role-based UI visibility (Admin, Cashier)
- Password and session management

4.Expected Deliverables:

Deliverable	Description
POS Software	GUI application for local billing (Python/Django)
Sync Engine	Sync tool for offline to central sync
Central Dashboard	Admin web portal (React + Django APIs)

Deliverable	Description
Database Schemas	MySQL DB for orders, menu, users, etc.
Reports	Power BI dashboards and auto-generated reports
Training Docs	User manual, SOPs for billing & sync
Deployment Package	Installation guide for local and server setup
Technical Support	Bug fixes, maintenance plan