Technical Design Document: A2B Restaurant Billing Automation System

1.Overview

The A2B Restaurant Billing Automation System is a Python-based solution designed to digitize and automate the billing and operations of A2B restaurant branches. The system supports offline-first architecture with synchronization capabilities to a central server. It uses a modular, scalable architecture powered by Django (backend), React (admin UI), MySQL (database), and REST APIs for cross-branch operations and syncing.

2. Architecture.

1.menu.py
Handles Add/Edit/Delete of menu items (CRUD operations).
Core module for menu management.

2.order.py
Manages customer orders: accepts input, links menu items, and prepares data for billing.
Essential for processing transactions.

3.billing.py
Calculates subtotal, tax, discounts, and final amount.
Performs core billing logic.

4.receipt.py
Generates and formats printed or digital bills.
Used at point-of-sale for output.

5.sync_service.py
Pushes and pulls data between local and central servers.
Enables offline-to-online sync capability.

6.auth.py
Manages login, session control, and secure access.
Critical for multi-role access security.

3. Module Descriptions

3.1 Menu Management

Add/Edit/Delete menu items, sync with central server, Django Admin Panel for control.

3.2 Order & Billing System

Take orders, apply tax/discounts, print bills, store in local DB.

3.3 Offline Functionality

Local server with MySQL, transaction logs, CSV backup.

3.4 Sync Engine

Push/pull data with Django REST, triggered on internet availability.

3.5 Admin Dashboard

React-based UI for analytics and centralized monitoring.

3.6 Reporting & Analytics

Sales reports, top items, Power BI integration.

3.7 User & Role Management

Django login, Admin/Cashier roles, session management.

4.Data Flow

1. Order placed via POS GUI
2. Stored in Local MySQL DB
3. Bill generated (PDF or Print)
4. Data queued for sync
5. Sync engine pushes to central server on internet availability
6. Admin dashboard fetches and displays reports

5.Technologies Used

Component	Technology
Backend	Python 3.x, Django

Frontend Dashboard	React.js
Database (Local/Central)	MySQL
Reporting	Power BI / Matplotlib
Sync API	Django REST Framework
Optional GUI (POS)	Django Template or PyQt

6. Future Enhancements

- QR-code-based order tracking
- Inventory and supplier integration
- Customer feedback and loyalty system
- Mobile version for order taking
- AI-based demand forecasting

7.Deliverables

Deliverable	Description
POS Software	GUI-based billing tool (Python/Django)
Sync Engine	Tool for syncing local data to central DB
Central Dashboard	React + Django Admin Panel
Database Schemas	MySQL schemas for menu, orders, users
Auto Reports	Power BI dashboards and downloadable PDFs
Training Docs	User manuals, setup guides

Deployment Package	Installer for branch and server setup
Maintenance Plan	Bug fixing and updates roadmap