**BISTRO 92 CHALLENGE: TRANSFORM THE DINING EXPERIENCE**

**A) Quick Fixes**

**Q1:**

**1. Live Order Progress Tracker for Customers**

Why it matters:  
 Customers love knowing what’s happening with their food. A simple visual tracker on the table device (like a pizza tracker) could show stages:  
 **“Order received → Being prepared → Ready to serve → On the way”**

Benefit:  
This keeps customers informed about their order progress, reducing uncertainty and inquiries to staff. It improves satisfaction by managing expectations and allows staff to focus on other tasks, streamlining operations.

### **2. Smart Dietary Filter and Allergen Alert System**

Why it matters:  
 Many diners have dietary restrictions or allergies. The device could offer a feature where customers can filter menu items by dietary preference (vegan,allergies) and get alerts if their selection contains any flagged ingredients.

Benefit:  
 Boosts safety and personalizes the dining experience for health-conscious or allergic customers — a feature many systems overlook.

**3.Real-Time Staff Assistance Summon**:

Why it matters:

Implement a feature that lets customers request immediate staff assistance (e.g., for spills or special requests) via a dedicated button, sending an alert to the admin dashboard with the table number.

Benefit:

This improves satisfaction by ensuring prompt service and optimizes staff response times without disrupting the order processing workflow.

**Q2:**

**1.Multilingual Interface with Bangla Localization:**

* **Description**: The interface offers a home screen language selector with Bangla prominently featured, enabling one-tap switching to Bangla for all text, menus, and instructions (e.g., “মেনু খুলুন” for “Open Menu”). It uses simple, colloquial Bangla and a clear font, with culturally relevant item descriptions.
* **Impact for Diverse Users**: Bangla localization ensures accessibility for users with limited English or tech skills, boosting confidence and satisfaction while maintaining efficient cloud-based order processing.

**2. Intuitive Ordering Process with Visual Cues and Voice Assistance:**

* **Description:** The interface guides users step-by-step with a progress bar and visual cues, like arrows, to direct them to the next action. A voice assistant provides clear instructions in multiple languages, ensuring users always know what to do next.
* **Impact for Diverse Users:** This system makes the ordering process simple and accessible, helping even tech novices follow along with ease and confidence.

**Q3:**

|  |  |  |
| --- | --- | --- |
| **Vulnerability** | **Description** | **Solution** |
| Device Theft | Someone could steal the table smart device and hack it offline to access sensitive systems. | set up remote wipe features if the device detects tampering or removal from the network**.** |
| Cloud Server Breach   |  | | --- | |  | | If the central cloud database is hacked, all orders, sales records, and customer preferences could leak. | Encrypt all sensitive data at rest (inside the database) and in transit (during communication). Also, regularly audit and patch the server software. |
| Denial of Service (DoS) Attack | Someone could flood the system with fake orders to crash it during peak hours | Implement rate limiting (limit how many requests one device can send per minute) and firewall rules to detect abnormal traffic patterns. |

**Q4:**

### **1. Load Balancing (as before)**

**Explanation**:  
 Use multiple servers with a load balancer in front of them.  
 When hundreds of customers press buttons and order food at once, the load balancer splits the work across multiple servers instead of overloading a single one.

**Result**:

* Faster response time.
* No downtime even during peak 7–9 PM rush.

### **2.Queue Management for Orders**

**Explanation**:  
 When too many orders arrive at the same time, instead of trying to process all immediately, you can **add orders to a queue** (like a task list).  
 The server **processes them one by one** or **in small batches**, preventing crashes.

Use **message queues** like:

* RabbitMQ
* AWS SQS
* Redis Pub/Sub

**Result**:

* Smooth processing even if 100 orders come at once.
* No system freezing.

**Q5:**

**Use an Automatic Data Synchronization Approach**

Instead of real-time integration, sync the smart ordering system and the inventory system at regular intervals through scheduled data updates (like every 1–5 minutes).

### How It Works:

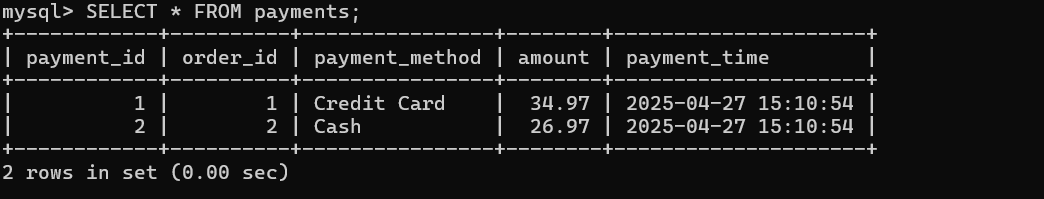
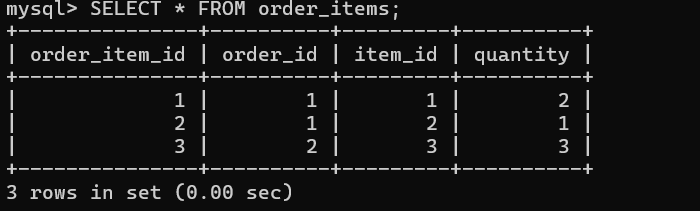
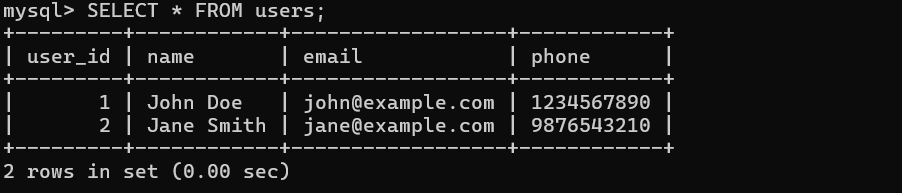
* Orders are stored temporarily in a separate smart system database.
* A synchronization script (small program) runs automatically every few minutes.
* It collects new orders, updates the inventory system based on items sold, and resets its temporary storage for the next cycle.

### Why This Method?

* No Load on Main System: Inventory system isn't overwhelmed by every tiny order event.
* Easier Rollback: If something goes wrong, it's easy to debug within short time windows (e.g., only last 5 minutes).
* Controlled Impact: If syncing fails once, the system retries in the next cycle without disrupting restaurant service.
* Flexibility: Frequency of updates can be adjusted based on peak and non-peak hours.

**B) Tech Tricks**

**Q1**



**Q2**

**SQL Query to Retrieve Orders from the Last Hour**

SELECT

t.table\_number,

oi.item\_name,

o.order\_time

FROM

orders o

JOIN

order\_items oi ON o.order\_id = oi.order\_id

JOIN

tables t ON o.table\_id = t.table\_id

WHERE

o.order\_time >= NOW() - INTERVAL 1 HOUR

ORDER BY

o.order\_time DESC;

**Q4**

### **Cloud-Based System Architecture for Real-Time Updates, Data Storage, and Smart Pad Communication**

#### Overview

This system enables real-time order updates, communication with kitchen staff, and data storage for Bistro 92. The architecture ensures fast, reliable service with low latency and high availability.

### 1. Components

1. Smart Pads (Customer Devices):  
   * These are the devices on the tables where customers place orders. They can be simple touchscreen tablets or web-based systems. The smart pad will communicate with the backend to send orders and receive updates.
2. Backend Server (API Gateway):  
   * This acts as the core of the system where orders are processed, data is stored, and real-time notifications are managed. Think of it as the server that connects everything.
3. Database:  
   * A cloud-based relational database (like MySQL or PostgreSQL) that stores customer details, orders, menu items, and other restaurant-related data.
4. Real-Time Notifications:  
   * This ensures that when a customer places an order, kitchen staff get notified immediately. We use WebSockets or event-driven systems like Kafka or Google Pub/Sub to manage these notifications.
5. Admin Dashboard:  
   * This is where the restaurant staff can monitor orders in real-time, update the menu, or check inventory.

### 2. Flow of Data

1. Order Placed: The customer places an order on the smart pad.
2. Order Sent to Backend: The order is sent to the backend server.
3. Data Stored: The backend saves the order in the database (like MySQL).
4. Kitchen Notified: The kitchen staff are instantly notified that a new order has arrived through a real-time messaging service (like WebSocket).
5. Admin Notified: The admin dashboard receives the same update so staff can monitor it.
6. Order Ready: Once the order is ready, updates are sent back to the smart pad and the kitchen.

### 3. Tech Stack

1. Cloud Provider: AWS, Google Cloud, or Microsoft Azure
2. Backend Services: Node.js or Python with Express/Django to create APIs.
3. Database: AWS RDS for MySQL or Google Cloud SQL for PostgreSQL.
4. Real-Time Communication: Use WebSockets for low-latency communication.
5. Notification System: Kafka or Google Pub/Sub to handle events and updates.
6. Admin Dashboard: React or Vue.js for building the staff interface.

### 4. Low Latency and High Availability

* Auto-Scaling: The system scales automatically to handle peak times.
* Database Replication: To avoid downtime, the database will replicate across regions.
* Caching: Use Redis for quick access to frequently needed data like menu items.

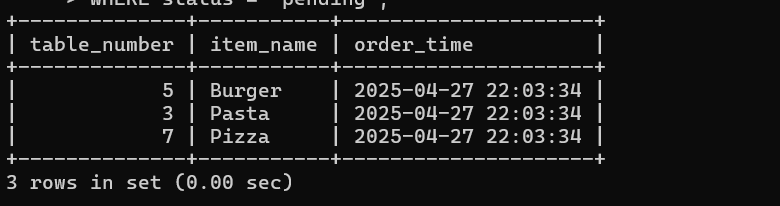
### 5. Benefits

* Real-Time Updates: Orders are processed and updated instantly.
* Scalability: The system grows easily as the restaurant expands.
* Reliability: Built to handle large volumes of orders without crashing.

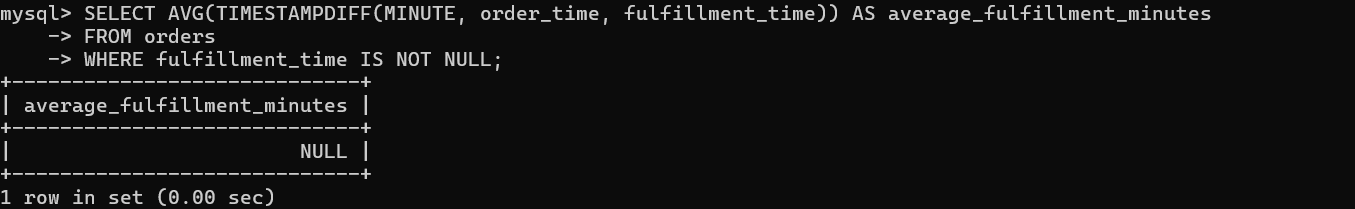
This approach combines fast communication, efficient data storage, and scalable architecture to ensure smooth operations. The real-time notifications and updates to both kitchen and customers make the ordering process seamless and efficient.

**Q5**

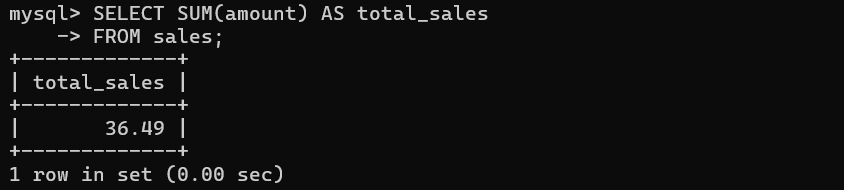
See **Pending Orders**:



See **Average Fulfillment Time** (example):



See **Total Sales**:



**D) Big Idea**

**Enhanced Solution: AI-Driven Features for Review Aggregation, Food Recommendations, and Personalized Suggestions**

#### **Problem Identified:**

Bistro 92 currently lacks a personalized customer experience and effective systems for tracking customer satisfaction. Furthermore, customers often have difficulty making decisions on what food to order, leading to order regrets. These issues can impact customer retention and overall business performance.

#### **Proposed Solution:**

We introduce AI-powered review aggregation, personalized food recommendations, and a visual meal preview feature to enhance the customer experience. This tech-driven approach will enable Bistro 92 to build better customer engagement, improve decision-making during the ordering process, and drive higher satisfaction levels.

### **1. AI-Powered Review Aggregation & Rating System**

Problem:  
 Manual review management is inefficient and doesn't provide an instant way to calculate accurate ratings or identify areas of improvement.

Solution:  
 Implement an AI-driven review aggregation system to automatically analyze customer feedback, categorize it, and generate an average star rating for each menu item. The AI will categorize reviews based on specific aspects (e.g., food quality, service, ambiance), providing actionable insights for Bistro 92.

#### How it works:

* Customer Reviews: After each meal, customers can leave a review (e.g., "Excellent food, but slow service").
* AI Analysis: The AI system will analyze the review using Natural Language Processing (NLP) to determine the sentiment (positive, neutral, or negative).
* Star Rating: The AI will calculate a star rating for each menu item based on the sentiment of the reviews.
* Categorized Feedback: Reviews are categorized into areas like food quality, service, ambiance, etc., enabling Bistro 92 to focus on specific improvement areas.
* Review Display: The average ratings are displayed on the menu, and detailed reviews are visible to customers browsing the restaurant's offerings.

#### Technology Stack:

* Natural Language Processing (NLP) for sentiment analysis (e.g., spaCy or Google Cloud NLP API).
* Backend: Node.js and Express.js for building APIs to collect and process reviews.
* Database: MySQL or PostgreSQL for storing customer reviews and ratings.

### **2. AI-Powered Food Recommendation System**

Problem:  
 Customers often find it difficult to decide what to order, and the restaurant misses opportunities to upsell complementary dishes that would improve the meal experience.

Solution:  
 Create an AI-powered food recommendation engine that suggests complementary items based on the customer's order. If a customer orders a dish like Paratha, the AI would suggest dishes like Kebabs or Curry that go well with it, enhancing the customer's meal.

#### How it works:

* Menu Data: Each menu item is linked to complementary items based on flavor profiles, popularity, and previous customer orders.
* AI Algorithm: The recommendation engine uses Machine Learning to identify patterns in customer orders and suggest side dishes or drinks that pair well with the selected main dish.
* Personalized Recommendations: The system will tailor suggestions based on customer preferences (e.g., vegetarian, spicy).
* Real-time Display: Recommendations are shown to customers as they make their order, encouraging upsells and making decision-making easier.

#### Technology Stack:

* Machine Learning: TensorFlow or scikit-learn for building the recommendation model.
* Backend: Node.js and Express.js for handling API requests for recommendations.
* Database: MySQL or PostgreSQL to store food pairings, preferences, and historical order data.
* Frontend: React Native or Flutter for displaying recommendations in the app.

### **3. Augmented Reality (AR) Meal Preview Feature**

Problem:  
 Customers often face decision fatigue when choosing items from the menu, leading to dissatisfaction with their final selection.

Solution:  
 Implement an AR-based meal preview feature that allows customers to see a 3D version of their meal before ordering. This will help customers visualize the food, enhancing their confidence in their choice and reducing order regrets.

#### How it works:

* AR Integration: Using ARKit (iOS) or ARCore (Android), a 3D model of the dish is displayed when the customer selects an item on the menu in the app.
* Real-time Interaction: Customers can rotate and zoom in on the food to get a better idea of the dish's appearance.
* User Engagement: The AR preview boosts excitement and engagement, helping the customer make a more informed decision.

#### Technology Stack:

* AR Development: ARKit (for iOS) or ARCore (for Android).
* Frontend: React Native or Flutter for building the mobile app.
* Backend: Node.js with Express.js for managing app requests.

### **4. Integrated Dashboard for Restaurant Management**

Problem:  
 Bistro 92 lacks a centralized system for managing orders, customer feedback, and sales in real-time.

Solution:  
 Develop a real-time dashboard for restaurant management that tracks pending orders, average fulfillment time, and total sales.

#### How it works:

* Pending Orders: Display a real-time list of orders that need to be fulfilled, sorted by table and priority.
* Average Fulfillment Time: Track the time it takes for orders to be completed and alert the staff if an order is taking too long.
* Sales Metrics: Display total sales in real-time, broken down by item, day, or week.

#### Technology Stack:

* Frontend: React.js or Vue.js for building the dashboard interface.
* Backend: Node.js with Express.js for API development.
* Database: MySQL or PostgreSQL to store and query orders, sales, and fulfillment times.

### Benefits of These Features:

1. Enhanced Customer Experience:  
   * AI-driven food recommendations and AR previews enhance the decision-making process, making it easier for customers to choose dishes they’ll love.
2. Increased Sales and Engagement:  
   * The recommendation engine helps upsell complementary items, leading to higher order value.
   * AR previews create excitement, encouraging customers to try more items.
3. Improved Operational Efficiency:  
   * The review aggregation and real-time dashboard provide restaurant managers with actionable insights into customer satisfaction, staff performance, and sales trends.
4. Actionable Insights:  
   * The AI-powered review system helps Bistro 92 gather customer feedback in real-time and make data-driven decisions for improving food quality and service.

### Tech Stack for Implementation:

* Frontend:  
  + React Native or Flutter for mobile app development.
  + React.js or Vue.js for web-based interfaces.
  + ARKit/ARCore for augmented reality features.
* Backend:  
  + Node.js with Express.js for building RESTful APIs.
  + Python (optional) for machine learning algorithms.
* Database:  
  + MySQL or PostgreSQL for storing order data, reviews, menu items, and customer preferences.
* AI/ML:  
  + Google Cloud NLP or spaCy for sentiment analysis.
  + TensorFlow or scikit-learn for food recommendation algorithms.
* Real-time Communication:  
  + Firebase or Socket.io for real-time updates to the kitchen staff, and pushing notifications to the app.

### Conclusion:

This combined solution addresses both operational and customer experience issues at Bistro 92. The AI-powered review aggregation, personalized food recommendations, and AR-based meal previews work together to boost customer engagement, streamline operations, and improve sales. The real-time dashboard further aids restaurant managers in tracking performance, allowing for proactive decision-making. By adopting these technologies, Bistro 92 will enhance its customer experience, increase revenue, and establish a competitive edge in the market.