Project Title: Product Dissection of Zomato – A Food Delivery Platform

Step 1 Overview:

This project is about understanding the features and data structure of **Zomato**, a popular food delivery platform.

We will study how Zomato works, what problems it solves for users, and how users interact with the app. Based on this understanding, we will design a **relational database schema** using tables and relationships.

The final goal is to show how data is stored and managed inside a big platform like Zomato, and how that data supports the smooth working of the app.

Step 2: Understanding Zomato's Core Features and User Interaction

Zomato is a popular food delivery and restaurant discovery platform. It helps users find nearby restaurants, check menus, read reviews, and order food online. Below are the main features and how users interact with them:

1. Restaurant Search – Users can search restaurants by name, cuisine, or location.

2. Online Food Ordering – Users can place food orders for home delivery.

3. Menu Display – Each restaurant shows its menu with item details and prices.

4. Ratings and Reviews – Users can read other people's reviews and write their own.

5. Order Tracking – Users can track their food delivery in real-time.

6. Online Payments – Users can pay via UPI, card, wallet, or choose cash on delivery.

7. Offers and Coupons – Users get discounts through promo codes and special deals.

8. Favorites/Bookmarks – Users can save restaurants for future orders.

These features make Zomato convenient and user-friendly, solving many food-related problems with just a few clicks.

Step 4: Case Study – Zomato Solving a Real User's Problem

Content:
Let's take a case study of a user named *Ravi*, a software engineer who works long hours and lives alone in Gurugram.

Problem Ravi Faces:

● He has no time to cook due to late office work.

- He often skips meals because nearby restaurants are closed late at night.

- When he tries a new restaurant, he sometimes gets poor food quality.

How Zomato Helps Ravi:

1. **Late-Night Food Access:**
   Zomato shows restaurants open 24/7, helping Ravi order food even at midnight.

2. **Reliable Choices with Ratings:**
   Zomato highlights user ratings and hygiene levels, helping Ravi choose safe and tasty food.

3. **Delivery to Home:**
   Ravi orders food while working, and Zomato delivers directly to his door.

4. **Time-Saving Filters:**
   He quickly finds veg, non-veg, or specific dishes using filters.

5. **Affordability:**
   Zomato's offers and discounts help Ravi manage food expenses on a budget.

Result:

Ravi now eats on time, feels less stressed, and trusts Zomato for everyday meals.

Step 5: Schema Design – Zomato Platform Database Structure

To design the schema for Zomato's platform, we imagine how data is stored and connected in different tables in a relational database.

There are seven important entities (tables) that Zomato's system needs:

1. **Users** – This table stores details of the customers such as their ID, name, email, phone number, and location.

2. **Restaurants** – It stores restaurant-specific information like restaurant ID, name, location, rating, and cuisine type.

3. **Menu Items** – Every restaurant offers different food items. This table contains the item name, price, category (veg or non-veg), and which restaurant it belongs to.

4. **Orders** – Whenever a user places an order, that information is saved here. This includes order ID, user ID, restaurant ID, order date, total amount, and status.

5. **Order Items** – One order can include multiple items. This table lists the items inside each order with their quantity.

6. **Delivery** – This table keeps track of order delivery, including delivery status and delivery time.

7. **Reviews** – After receiving food, users can review the restaurant. This table stores ratings and comments along with user and restaurant IDs.

Each table is connected through **primary keys** and **foreign keys**. For example, the user_id in the Users table is used as a foreign key in the Orders table to show which user placed the order. Similarly, restaurant_id connects Restaurants to Menu Items and Reviews.

This schema helps organize and manage Zomato's data efficiently.

Step 6: Entity-Relationship (ER) Diagram – Zomato Platform

The Entity-Relationship (ER) Diagram visually represents how the main entities in Zomato interact with each other. Since this document does not include diagrams, the relationships are described in text format below:

A **User** can place multiple **Orders**.

Each **Order** contains one or more **Order Items**.

Each **Order Item** is linked to a specific **Menu Item**.

A **Menu Item** belongs to a specific **Restaurant**.

A **User** can write multiple **Reviews**.

A **Restaurant** can receive multiple **Reviews**.

Each **Order** has one **Delivery** record associated with it.

These relationships show how Zomato connects users, restaurants, menu items, and delivery processes in a structured relational database model.

Step 7: Case Study – Solving User Problems on Zomato

User Scenario:

Rahul is a working professional living in a metro city. After a long day at work, he doesn't feel like cooking. He opens the Zomato app to order dinner. But he faces a few challenges:

He wants something affordable and tasty.

He is unsure which restaurant is best rated nearby.

He wants the food delivered within 30 minutes.

How Zomato Solves These Problems:

The **Search and Filter** features allow Rahul to sort restaurants by rating, delivery time, and cost.

User Reviews and Ratings help Rahul trust the quality of food and service.

Estimated Delivery Time gives him an idea of how fast the food will arrive.

Offers and Discounts make his order more affordable.

Live Order Tracking keeps him updated on the progress of his delivery.

Outcome:

Rahul orders from a top-rated nearby restaurant, gets his food on time, enjoys it, and leaves a review. Zomato successfully solves his problems through its smart features.

Step 8: ER Diagram (Entity-Relationship Diagram)

In this step, I have created an ER diagram to represent the data structure of the Zomato-like platform.

This diagram helped me visualize how different entities like Users, Restaurants, Food Items, and Orders are connected with each other.

For example, one user can place multiple orders, each order can contain multiple food items, and each food item belongs to a specific restaurant.

The ER diagram includes key entities such as User, Restaurant, Food Item, Order, and Delivery, along with their important attributes.

igned this ER diagram using an online tool (draw.io), then exported # I desit as an image and added it below in this document.

This step gave me a better understanding of how relational databases are structured in real-life applications like Zomato.