**ER Diagram Of LocaBusin Project**

Entity-Relationship (ER) diagram for your **LocaBusin** project, along with a step-by-step explanation. This diagram serves as the architectural blueprint for your database.

An **Entity-Relationship (ER) diagram** is a visual chart that illustrates the main "entities" (your database tables) and the relationships between them.

**Step 1: Identifying the Core Entities**

Based on the features we've built and planned, your project has four primary entities:

1. **Users** 👤: The individuals who register, log in, make bookings, and write reviews.
2. **Businesses** 🏢: The local establishments listed on the platform.
3. **Bookings** 📅: The appointments or reservations that connect a specific user to a specific business.
4. **Reviews** ⭐: The feedback and ratings that connect a specific user to a specific business.

**Step 2: Defining Attributes and Keys**

For each entity, we define its attributes (the columns in the table) and identify its keys.

* **Primary Key (PK)**: A unique identifier for each record in a table (e.g., user\_id).
* **Foreign Key (FK)**: A key used to link two tables together. It's a field in one table that uniquely identifies a row of another table.

**Step 3: The ER Diagram**

Here is the visual representation of your database structure using crow's foot notation, where the "crow's foot" (>-) symbol represents the "many" side of a relationship.

**ERDiagram**

USERS {

int user\_id PK

varchar name

varchar email UK

varchar password

timestamp created\_at

}

BUSINESSES {

int business\_id PK

varchar name

varchar category

varchar address

varchar phone

varchar image

text description

timestamp created\_at

}

BOOKINGS {

int booking\_id PK

int user\_id FK

int business\_id FK

datetime booking\_date

varchar status

timestamp created\_at

}

REVIEWS {

int review\_id PK

int user\_id FK

int business\_id FK

int rating

text comment

timestamp created\_at

}

USERS ||--o{ BOOKINGS : "makes"

BUSINESSES ||--o{ BOOKINGS : "receives"

USERS ||--o{ REVIEWS : "writes"

BUSINESSES ||--o{ REVIEWS : "receives"

**Step 4: Explaining the Relationships**

This is how the entities are connected:

**1. Users ↔ Bookings (One-to-Many)**

* **A single User can make many Bookings.**
* **A single Booking belongs to exactly one User.**
* **How it works:** The BOOKINGS table contains a user\_id column (a Foreign Key) that points to the user\_id (the Primary Key) in the USERS table. This creates a direct link, so you always know which user made which booking.

**2. Businesses ↔ Bookings (One-to-Many)**

* **A single Business can receive many Bookings.**
* **A single Booking is for exactly one Business.**
* **How it works:** The BOOKINGS table also contains a business\_id column (a Foreign Key) that points to the business\_id in the BUSINESSES table. This tells you which business each booking is for.

**3. Users ↔ Reviews (One-to-Many)**

* **A single User can write many Reviews** (for different businesses).
* **A single Review is written by exactly one User.**
* **How it works:** The REVIEWS table has a user\_id column (FK) that links to the USERS table, identifying the author of the review.

**4. Businesses ↔ Reviews (One-to-Many)**

* **A single Business can receive many Reviews** (from different users).
* **A single Review is for exactly one Business.**
* **How it works:** The REVIEWS table also has a business\_id column (FK) that links to the BUSINESSES table, ensuring each review is attached to the correct business listing.