



# Mastering Projects

Everything You Need to Know for Your Resume and Interviews

---

**Presented By**

J B Mugundh

# What kind of projects can I have in my resume?

Any projects that showcase your skills, creativity, self-directed learning, hands-on experience, and ability to solve real-world problems.



# Transforming boring projects into extraordinary innovations

**Admin dashboard**

**Payment Integration**

**Chatbot**

**Automated Email**

**UI UX improvements**

**Custom Search Filters**

**Unit Tests and Functional Tests**

**CI / CD**

**Performance optimization**

**Personalized Recommendations**

# How to mention it in my resume?

## Problem Addressed

- Clearly state the issue or need the project aimed to solve.

## Solution Developed

- Describe the solution you created and its core functionalities.

## Unique Aspects

- Highlight distinctive features or approaches used in the project.

## Outcomes/Impact

- Share the results or impact of the project, including metrics.

# Amazon Lite | React.js, Node.js, Express.js, MySQL, Stripe API, AWS, Docker

---

- Developed a fully functional e-commerce website clone inspired by popular online marketplaces.
- Implemented advanced search filters for products to enhance user experience.
- Created an intuitive admin dashboard allowing for easy management of products and offers.
- Developed automated email notifications for key events, including new discount offers and abandoned cart reminders.
- Addressed slow page load times by 15% by optimizing database queries and implementing caching mechanisms.
- ~~Successfully integrated Stripe API for secure payment processing.~~
- ~~Deployed the application using AWS, with continuous integration and deployment pipelines set up for efficient updates.~~

# How to present / prepare the projects for interview?

- Follow STAR (Situation, Task, Action, Result) method.
- You know a lot of things and you want to say a lot of things, but that's not what the person at the other end wants to hear!
- Ask "Who's in front of me? What do they care about? And how do they like to receive information? How can I get them engaged"
- If they want more information, they'll ask for it. And this would end up in an interactive way. Remember, "Less is more"
- Make sure to be prepared beforehand for the follow-up questions and mention the challenges faced as well.

# EduConnect

---

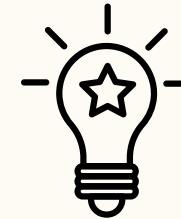
- You know how traditional learning can be pretty isolating, right? Students often struggle to find the right resources, connect with others, and pinpoint exactly what they need to work on to improve. We saw this as an opportunity to change the game. Our goal was to create a dynamic peer-to-peer learning platform where learners could collaborate in real-time, get personalized course recommendations, and bridge the skill gaps holding them back.
- As the lead of this group project, I took on the challenge of making this vision a reality. My focus was on building features that enabled real-time collaboration and designing an AI-driven system for personalized skill gap analysis while keeping the entire team on track and make sure all the pieces fit together smoothly.
- I implemented the features like video calls, screen sharing, file sharing, session recording, and chat using the Mirotalk API.

# EduConnect

---

- I also developed a system that utilized the Gemini LLM to analyze users' performance data and identify skill gaps. This system would then recommend personalized courses from our platform to help learners tackle their weak spots and make real progress.
- Beyond coding, I coordinated with the rest of the team to create courses portal, AI-powered mock tests, real-time chatbot assistance, etc.
- Finally, I took the lead in testing and integrating all these features.
- The end result? We successfully delivered a comprehensive P2P learning platform that received great feedback. The real-time collaboration features made learning together easy and engaging, while the skill gap analysis helped learners focus on their weaknesses and improve. The personalized learning paths we created truly made a difference, and it was incredibly rewarding to see the impact it had on users.

# How to approach Database design in interviews



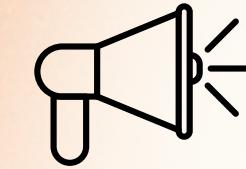
## Requirements Gathering

Identify the requirements and needs of the users and the business.



## Logical Design

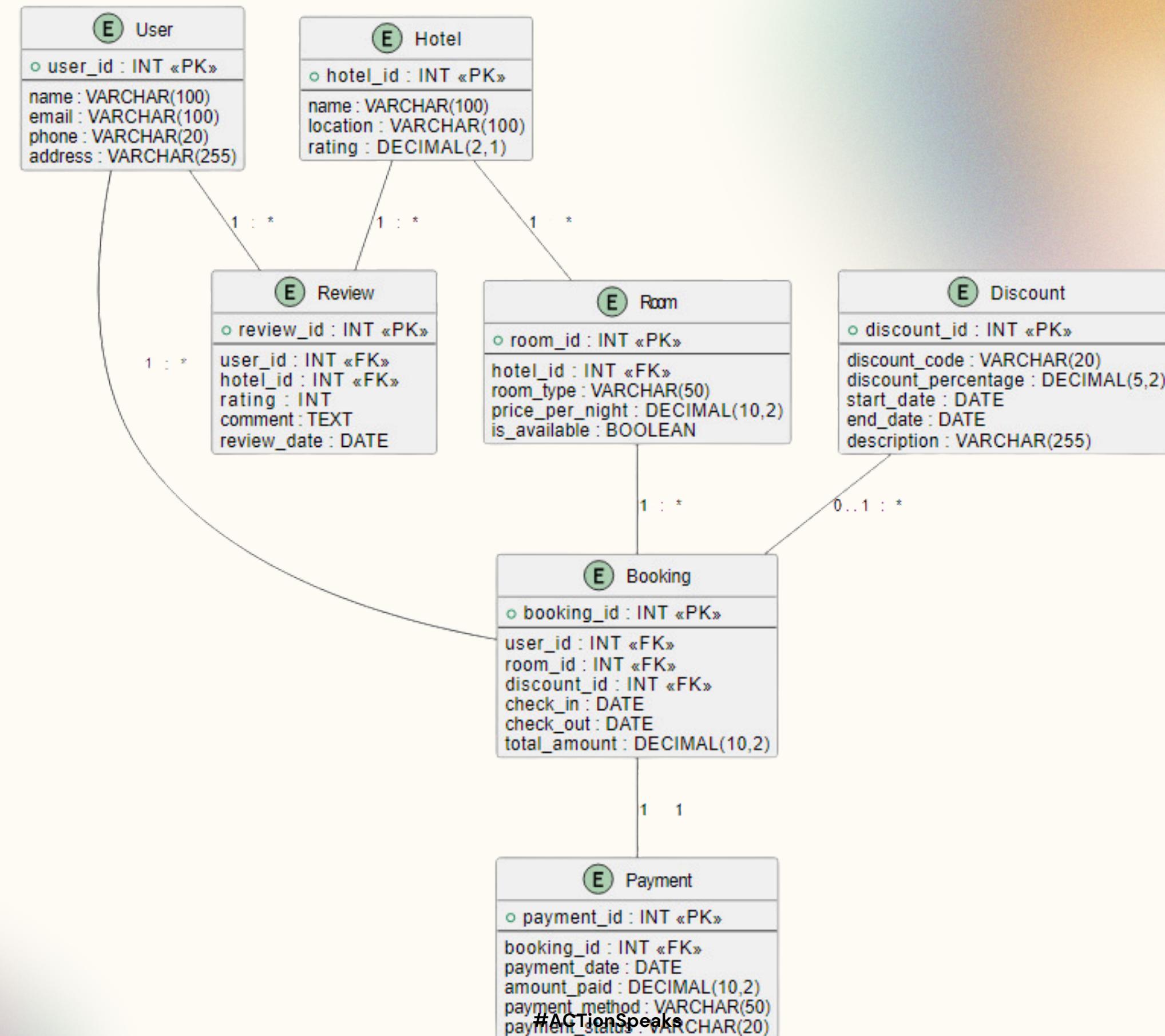
Define the structure of the database in terms of tables, columns, and relationships.



## Normalization

Apply normalization rules to organize the data and reduce redundancy.

# Hotel Management System (SQL)



# Hotel Management System (SQL)

---

```
CREATE TABLE User (
    user_id INT PRIMARY KEY,
    name VARCHAR(100),
    email VARCHAR(100),
    phone VARCHAR(20),
    address VARCHAR(255)
);
```

```
CREATE TABLE Hotel (
    hotel_id INT PRIMARY KEY,
    name VARCHAR(100),
    location VARCHAR(100),
    rating DECIMAL(2,1)
);
```

```
CREATE TABLE Room (
    room_id INT PRIMARY KEY,
    hotel_id INT,
    room_type VARCHAR(50),
    price_per_night DECIMAL(10,2),
    is_available BOOLEAN,
    FOREIGN KEY (hotel_id) REFERENCES Hotel(hotel_id)
);
```

```
CREATE TABLE Discount (
    discount_id INT PRIMARY KEY,
    discount_code VARCHAR(20),
    discount_percentage DECIMAL(5,2),
    start_date DATE,
    end_date DATE,
    description VARCHAR(255)
);
```

# Hotel Management System (SQL)

---

```
CREATE TABLE Booking (
    booking_id INT PRIMARY KEY,
    user_id INT,
    room_id INT,
    discount_id INT,
    check_in DATE,
    check_out DATE,
    total_amount DECIMAL(10,2),
    FOREIGN KEY (user_id) REFERENCES
    User(user_id),
    FOREIGN KEY (room_id) REFERENCES
    Room(room_id),
    FOREIGN KEY (discount_id) REFERENCES
    Discount(discount_id)
);
```

```
CREATE TABLE Payment (
    payment_id INT PRIMARY KEY,
    booking_id INT,
    payment_date DATE,
    amount_paid DECIMAL(10,2),
    payment_method VARCHAR(50),
    payment_status VARCHAR(20),
    FOREIGN KEY (booking_id) REFERENCES
    Booking(booking_id)
);
```

# Social Media Platform (NoSQL)

```
const userSchema = new Schema({
  username: { type: String, required: true, unique: true },
  email: { type: String, required: true, unique: true },
  password_hash: { type: String, required: true },
  profile_picture: { type: String },
  bio: { type: String },
  followers: [{ type: Schema.Types.ObjectId, ref: 'User' }],
  following: [{ type: Schema.Types.ObjectId, ref: 'User' }]
});
```

```
const messageSchema = new Schema({
  sender_id: { type: Schema.Types.ObjectId, ref: 'User',
    required: true },
  receiver_id: { type: Schema.Types.ObjectId, ref: 'User',
    required: true },
  content: { type: String, required: true },
  timestamp: { type: Date, default: Date.now }
});
```

```
const postSchema = new Schema({
  user_id: { type: Schema.Types.ObjectId, ref: 'User', required: true },
  content: { type: String, required: true },
  timestamp: { type: Date, default: Date.now },
  comments: [
    comment_id: { type: Schema.Types.ObjectId },
    user_id: { type: Schema.Types.ObjectId, ref: 'User' },
    content: { type: String, required: true },
    timestamp: { type: Date, default: Date.now }
  ]
});
```

```
const notificationSchema = new Schema({
  user_id: { type: Schema.Types.ObjectId, ref: 'User', required: true },
  type: { type: String, required: true }, // 'follow', 'like', 'comment'
  message: { type: String },
  is_read: { type: Boolean, default: false },
  timestamp: { type: Date, default: Date.now }
});
```

# Q&A

---

Thank you for listening!