

# Full-Stack Interview Preparation Guide

*Your Complete Roadmap to Success*

---

# Top Priority (Must-Know for Every Interview)

*These are non-negotiable — you must understand them clearly and deeply*

## 1. Project Structure & Flow

- Understand the full flow: from frontend form → API request → backend logic → database → response → frontend display
- Be able to explain: "How does a booking happen in your app?"

## 2. Authentication & Authorization

- What is authentication vs. authorization?
- How do login/register work?
- How are passwords stored? (bcrypt)
- How do you protect routes? (middleware like `isLoggedIn`)
- JWT vs Sessions (know what you're using and why)

## 3. CRUD Operations

- How to Create, Read, Update, and Delete listings/bookings
- Understand how CRUD is implemented in MongoDB + Mongoose + Express

## 4. MongoDB + Mongoose

- What is a schema?
- How do you define models for Listings, Users, Bookings?
- How do you relate users to listings?

## 5. React Basics + Data Fetching

- `useState`, `useEffect`, `props`, component structure
- How do you fetch data from your backend? (`axios`, `fetch`)
- Conditional rendering, loading states

## 6. Express.js Routing + Middleware

- Route structure: `/api/listings`, `/api/bookings`
- Custom middleware: `isLoggedIn`, error handlers
- How routes connect to controllers

## 7. Form Handling + Validation

- How did you build the forms in React?
- How do you validate inputs on frontend/backend?
- Example: price must be a number, email must be valid

## Medium Priority (Nice to Know, Asked Sometimes)

*These give your interview more depth if you can explain them*

### 8. Image Uploads

- Are you using Cloudinary or local storage?
- How are you sending files from frontend to backend?

### 9. Error Handling

- How do you catch async errors in Express?
- How do you display error messages in the frontend?

### 10. Environment Variables

- What is ``.env`` and why do we use it?
- Example: Mongo URI, Cloudinary keys

### 11. Deployment

- Where did you host frontend? (Vercel?)
- Where did you host backend? (Render/Railway?)
- How did you set up MongoDB in production?

## Optional Topics (Extra Confidence, Not Always Asked)

### 12. Reviews & Ratings System

- If you built it, understand how users post reviews
- How ratings are averaged and displayed

### 13. Search/Filter Feature

- How do you search listings?
- Did you use query params in routes?

### 14. Responsive UI / CSS Framework

- Tailwind / Bootstrap / custom CSS
- How did you make the UI mobile-friendly?

## Tip: Don't Just "Know" — Be Ready to Explain

For each of these topics, ask yourself:

- **What is it?**
- **Why did I use it?**
- **How does it work in my project?**

If you can do that — you're ready to impress any interviewer.