

## Demo Tips & Code Walkthrough Guide (In 10 minutes)

### 1. Walk Through Key Features

- Highlight **user input on the frontend**, **API calls to the backend**, and **data persistence in the database**.
- Demonstrate **end-to-end flow**, for example:  
**Create a record:**  
React form → Node handler → DB insert
- Also cover **Read / Update / Delete** flows if possible.

### 2. Frontend Focus: React + JS

#### a. Key Components

Explain clearly:

- `useState` holds component data
- `useEffect` fetches API data once on mount
- `map()` renders lists dynamically

#### b. Event Handling / Form Submission

- Show how **form input triggers a `fetch()` POST request**
- Explain **controlled inputs** and why they are used

**Tip:** Show only **key code parts**, not the entire component.

### 3. Backend Focus: Custom Node Server

#### Key Points to Explain

- **Manual routing** without frameworks
- How **request data is collected and parsed** (`req.on('data')`, `req.on('end')`)
- **Sending JSON responses**
- Difference between **GET and POST** requests

**Tip:** Highlight the minimal code that achieves the functionality.

#### 4. Database Interaction

- Show simple DB queries (SQL or NoSQL)
- Explain:
  - Where data lives
  - How the backend communicates with the DB
  - Example of CRUD operation

**Tip:** Keep it short; focus on **data flow and logic** rather than schema details.

#### 5. End-to-End Feature Walkthrough

Pick **one feature** and demonstrate the **full request-response cycle**.

**Example: Adding a user**

1. React form → fetch("/api/users", POST)
2. Node server → reads req.body
3. DB → adds user
4. Node server → responds with JSON
5. React → updates UI dynamically

**Tip:** Optionally, use a **diagram** to visually show the flow.

#### 6. High-Weight Focus Areas

- Ability to explain **why your code works**
- Understanding **data flow**:  
UI → API → DB → Response → UI
- Manual Node server & routing logic
- React **state, props, and rendering**
- Simple **DB queries and CRUD operations**

**Avoid:** Just running the app without explaining the code.