Interview System Design

What your interviewer is expecting and how you should give your answers.
In System design interviews there are no right or wrong answer. It's about the mindset

1. Why System design interviews feel harder?
- There is no direct questions and answers. Say we want to build LMS (Learning Management System). Interviewers don't expect you to know everything about LMS, instead they want you to be curious and ask questions to them.
- You have to gather information
- You know the concepts, you don't have to pull all the concepts together, instead you have to understand requirements and design accordingly.
- Unstructured
- No fixed syllabus
- You don't recall, you choose

Interviewer wants to see one thing: Decision-making, not knowledge tests

2. What interviewers really wants?
- Are you able to understand the problem?
- Are you able to get the requirements?
Say the interviewer wants you to build a clone of Facebook. You have to list down the features and ask them what they want
    Facebook features – Auth
    - Profile, Feed (Live, Video, Text), Friends, Chat
Then ask interviewer which feature he wants to pick. Say Feed.
- Data models / Entities: say for Ticket booking application (what entities we need? User and Ticket)
- Once this part is over, next part is Distributed environment of the application, so CAP / PACELC comes into picture. For the given system, you got to choose whether Availability is more important or Consistency. Accordingly we will choose the resources. Say if you are using cache, you got to say Cache could sometimes have stale data and in this system, Availability is more important the Consistency.
- Correctness awareness. That we are able to justify our design for the given requirements
- How will you maintain the application post-production? That means interviewer is interested in observability / maintainability.
Hack is, it is wise to use fewer components in the system.

3. What are good interview traits?
- You are framing the problem correctly.
- You are deciding your priorities (SQL / NoSQL)
- Pick the defaults after discussion, not assuming anything (you got to pick correct number of users for that application. Say we are getting 9k requests / second, how are we going to handle them?)
- System should be able to adjust itself to scale
- Call out the risks
- Communicating trade-offs. You have to be confident enough to speak about your components for the feature
- Say you have left-out certain things out of consideration and if interviewer is interested in that then estimate and say it will take this much time and it is unnecessary.
- They may ask how to integrate AI features into LMS

4. Good design grows, Bad design starts big
5. Trade-off:
- Communicate them
- Discuss the plus and minus
- Communicate what you are choosing and what you are not choosing

6. Mistakes:
- You will jump directly into DB schema like I want to create User table immediately
- Don’t start with DB
- Designing before discussing
- Assumption about requirements
- Being quiet
- Treating your interviewer as a judge not as a collaborator

7. Other suggestions:
- If you are preparing for interviews, you don’t have to learn new topic
- Keep all the topics in your head
- Revisit the requirements, trade-offs, failures, logs, bottlenecks classes
- Try to read different system designs, share knowledge on LinkedIn, YouTube, Blogs
- Create notes of your knowledge
- Practice creating designs and post online
- While practicing stop yourself from over-engineering
- You know the system design, from now your growth depends upon calm decisions under ambiguity