

**As Per Documentation :**

1. How to write OneToOne , OneToMany, ManyToOne, ManyToMany relations. We just need to know the annotations.
2. How to call external APIs (3rd party call).

**As Per Kumar Shivam (Time Duration: 35 Minutes) :**

1. **Code:** Write a find by an employee id (findById) controller.
2. **Code:** Write a service layer for this findById controller call.
3. **Code:** Write a repository for this findById service call.
4. **Code:** Write an employee model. Try to use SequenceGenerator/UUID for employee id.
5. **Code:** Update employee details from spring boot, but you need to update specific fields, not all fields. Tell me the annotation for this. (Use Of Transient, JsonIgnore, Custom Query)
6. **Code:** Get employee details from DB, but not all field values. Tell me the annotation for this. (Use Of Transient, JsonIgnore, Custom Query)
7. How to write a singleton class? How can you define this?
8. Do you know how to write testing in spring boot mainly with MockMvc and JUnit?

**As Per Ipsita Ghosal (Time Duration: 1 hour) :**

1. Introduce yourself.
2. **Code:** Basic Java - Define an User class with a private String firstName, list of addresses.
3. **Code:** Basic java - Define a method where you can add an address to one user.
4. **Code:** Basic Java - Null pointer Exception when you do this below command, why? :  

```
public void method addAddress(Address address){
    User.add(address);
}
```
5. What is the annotation for a model in a JPA repository?
6. What is the annotation we can use for service? (I have used @Component/@Service)
7. If we are using @Autowired, is there any other approach other than this (Some constructor level code)?
8. Difference between constructor level approach and autowired, why are you using autowired but not a constructor to declare the same?
9. **Code:** How can we validate a field as not null and non-empty? (There are 3 options, one from front end, another one from model, another one from DAO / DTO / Repository / Persistence level. We can use @Valid too at the controller level. I have Mentioned @NOTNULL, @NOTEMPTY at the model level)
10. **Code:** Complete the model with proper annotations:  

```
Public class User{
    Private String Id;
    Private String name;
}
```

(Add @Entity. I have mentioned ID can't be string, we can do either int or UUID. Add @NotNull too for name field )

11. **Code:** How to declare a model with UUID as an Id declaration.
12. **Code:** Write a repository for a find by all users. (I have mentioned that it's not needed, from the service layer we can by default call the repository without mentioning any method here).
13. **Code:** Write a custom query to find all user lists in a repository. (Be very specific with your annotations)
14. **Code:** Write a service layer to call that user list repository.
15. Have you used ResponseEntity? When and why?
16. **Code:** Write a controller to call that service level method to get all users with url:  
<https://somecompany.com/api/users>
17. **Code:** There is an external API : <https://somecompany.com/getcities> . How can we explicitly call this from spring boot.
18. Which CI/CD you have used?
19. Can you configure the CI/CD pipeline?
20. If there is any rest api in the rest controller but you don't want that controller to be called from User privilege but from Admin privilege, then what can you do? Suppose only admin can call this API, not the users, then what can you do -  
<https://somecompany.com/api/saveUser>
21. Do you know any NoSQL DB?
22. Any cloud providers have you worked with like AWS, Azure?

**As Per Sanjog Bagchi (Time Duration: 50 Minutes) :**

*Most of the questions were similar to those of Ipsita's & Shivam's. Writing the different/new questions asked to me by Jason below.*

1. How to avoid writing getter and setter methods and constructors? Use Lombok dependency and add @Getter, @Setter, @AllArgsConstructor, @RequiredArgsConstructor, and @NoArgsConstructor annotations at class level.
2. How to make an ArrayList Read Only? Need to use Collections.unmodifiableList() method
3. Write a code to make a REST call to "https://demo-project/v1/users/17" to get data for the user with userId 17.
4. Have you used Spring Data JPA? If so then what is the Difference Between Hibernate and Spring Data JPA?
5. Write controller methods(Get and Post) to make call to service layer methods to get and post user data. Make use of @Valid annotation in front of @RequestBody annotation and also use @Validated annotation at controller level for validation purpose.
6. Write Basic Java code- Define an User class with a private int id, String firstName, list of addresses. Use validations like @NotBlank or @NotEmpty or @NotNull
7. Make changes to this User class to make it an entity and use OneToMany mapping for the addressList so that it is mapped by using "user". Be very careful about the annotations used.
8. Write an addAddress method inside the User class to add address to the addressList. Make the addressList an unmodifiableList.

## **8 days Preparation:**

### **Day1:**

Configuration -> STS, Mysql, jdk

Basic Java (Ipsita: 2,3,4)

### **Day2:**

Basic crud operation and DB connect (Ipsita: 11-16) (Shivam: 1-4)

### **Day3:**

Learn the annotations (Ipsita: 5,6,7,8,9,10) (Shivam: 5,6) (Documentation: 1)

### **Day4:**

External API call (Ipsita: 17) (Documentation: 2)

Singleton class (Shivam: 7)

### **Day5 (Basic Questions to Learn) :**

(Ipsita: 18-22)

### **Day6:**

Testing (Shivam: 8)