Mearn:-

HR Interview Questions

Introduce yourself?

① describe ~~deepak as a person~~

Sir I have recently completed post graduation diploma in advance Computing from sunbeam pune, I have completed my engineering in electronics & telecommunicatn from sinhgad academy of Engineering pune,
I have completed 12th from SR Public school kota, completed tenth from NTPC korba, CG.

as a family we are two brothers 3 sisters, I belong to osmanabad maharastra,

---

Describe deepak as person?

① hardworking
② sincere
③ time bound & helping.
④ Quick learner
⑤ punctual
⑥ patience
⑦ initiator
⑧ strong wall.
⑨ optimisor
⑩ planned worker

① hard ~~hand~~ written notes all modules, ready to put an extra effort on concepts with doubt, almost all thaught hands on.

② love always before time, every activity in plan i always try to be on time and i put all extra time required for that sincerity.
→ sincerely following orders of trainer

③ planner → i love to plan before doing any activity, because i think we can achieve better result with planning
→ group leader electn → putting whatapp on group
→ 15min → preparing ppts for cordinatn
→ 15min → why I should be leader
↳ result of plann → leade.

→ Interview guidance Seminar
planning → asking mentors.
→ posting msg
→ boundry of event
→ managing google meet
→ reduce confusion
examinatn, tours, before presenting my thoug
→ programs at gharata prakop.

Initiator → interview guidance
→ nptel library
→ project starting
→ Group leader → assignment discussion
→ starting of day feed back
→ fodder based project adjob.
→ pcb design
→ course planning

---

helping → Group leader → helping nature installatn of software
→ many doubts related to technologies. i had made videos and posted
→ in team also if there is lack of knowledge of something, i used to study and arrange Seminar.
→ most of my friends approached me for doubts

patience :- after so many failures also standing with same enugy, IIT jee, Gate, bank exams
→ set targets → spring-security, jwt, docker-container
→ patiently studied all required components slowly-slowly implemented with immense searching
→ none of functionality was compromised
→ followed all trainers instructn
→ JPA, one to many, many to one

Optimiser → i love making things simpler, effective.
→ vscode transparency, deskpins.
→ time stamps in note book,
→ explore technology and try to make work simpler
→ linux module → laptop ram was insufficient for virtual box, used ec2 instance
↳ vim
→ short notes.
→ ~~all images on~~
direct deployement → server.jar → node server.js

Server.jar → containerise → mysql
build → memory consumptn → nginx
heavy → java

result → 600mb of ram in single instance → JPA, @one to many

Sticky to work :- once plan is ready, then giving whatever time and getting work done
↳ earlier plan → docker, spring security, jwt

→ at any cost studied, researched implemented all three
→ I make plan and always try to achieve it
→ for that i gain knowledge, do research of common doubts
→ i try every simple doubt on net

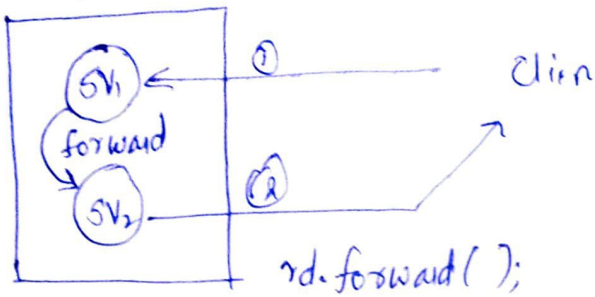
Time bound :- project was completed in January. target was april
→ every sunday, second half of saturday was given to project
→ planning of division
→ always project submission, assessment submissions were always ahead of time.

Organised → git, maintains folders of things i learned,

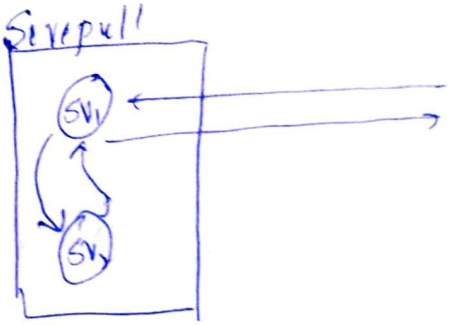
Curious → goDaddy.com, ec2 s3, @lombok, git in vscode, sts, mysql workbench, git GUI presentatn tool, obs, recording through zoom,
→ net speed disconnect detector
→ ec2 instance charge free
→ spring security → @annotatn

Advance java

Server pull



Servepull



rd.include(req, resp).

Spring → "forward: url"

Session tracking

→ rest servic → stateless

state-management → client info is maintained by web app

client
1. cookie
2. session or localstorag
3. hiddenfield
   input-hidden

Serve.
1. http session
2. application, ServletContext
3.

What is spring boot

↳ many framewor

Spring-framework
- ↳ web
- ↳ database
- ↳ AOP
- ↳ messaging

Spring-boot

spring rest | spring security
spring data | ...

spring-framework

Spring boot = spring + Embeddid server + auto configuratn + xml
Reduced effort.

e.g spring JPA
↳ preconfigured
↳ which hibernate
JPA
mysql

⊗ no requirem of motion version

What is auto configuratn?

↳ checks which class on classpath
- ↳ dependencies
- ↳ beans created

↳ automatically beans are created

Spring bean life cycle

↳ spring bean objects (singleton) are created

↳ while creating Object we can change Behavior that is given by life cycl

↳ Bean Post Process.

1. Instatiation
2. Setter based DI
3. BeanNameAware.setBeanName()
4. ApplicatContextAwar.setApplicat()
5. BeanPostProcesor.postProcessBefor Initiati

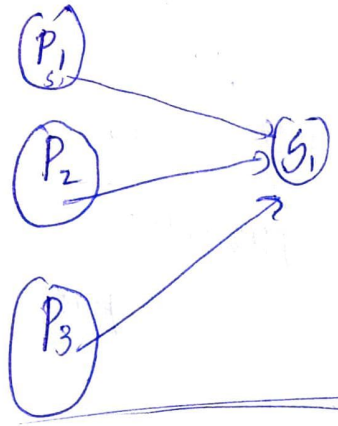
⑥ custom init() @postConstruct(e)
⑦ AfterPropertySet.
⑧ postAfterInitialisatn()

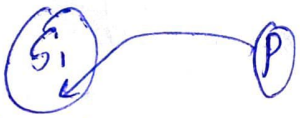
Bean Destruction

① custom destroy() @preDestroy)
② DisposableBean.destroy()
③ ob.finalize();

Bean Scope

@singleton in prototype bean.





by default only

whe ~~prototy~~ inner prototy bean access each time new bean should be given

↳ Inherit from ApplicatnContextAware

```
Inner  getInner(){
         return cts.getBean(innerclass)
       }
```

③ Method to achieve

@Lookup

```
@Lookup
Inner getInner(){
        return null
      }
```

Inversion of Control

```
Car c = new Car();
  c.setEngine(e)
  c.setChassis(ch)
  c.setWheel(w);
  .
  .
  c.drive();
```

reverse process follow
we just tell configuratn to create object for you

⇒ this process is autowiring

Autowiring

special DI, auto selecting bean object to be injected

field based di

```
class Account {          ← to tell spring
    @Value("101")        } dependency
    int id;
    double balance;      → only DI
    @autowired
    Person accHolder;
}
```

Autowiring
@configuratn
@Bean → DI + Autowiring

```
Person p1(){
    return new Person()
}
```

@Autowired types
↳ by type, by name, constructor
none.

@Autowired
↳ at field level
↳ at setter method level

Spring mvc

Rest Service :

→ automatically Jackson object created data can be converted to "json"

@RestController

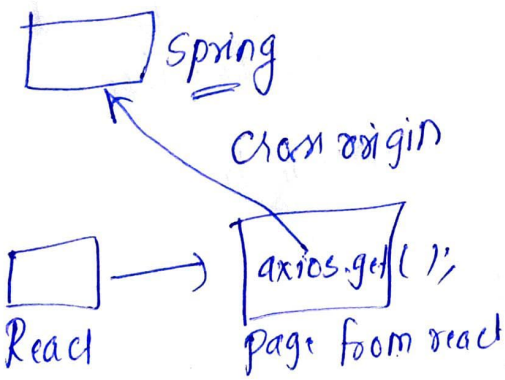
@ResponseBody + @Controller

@ Jackson message Converter

@CrossOrigin

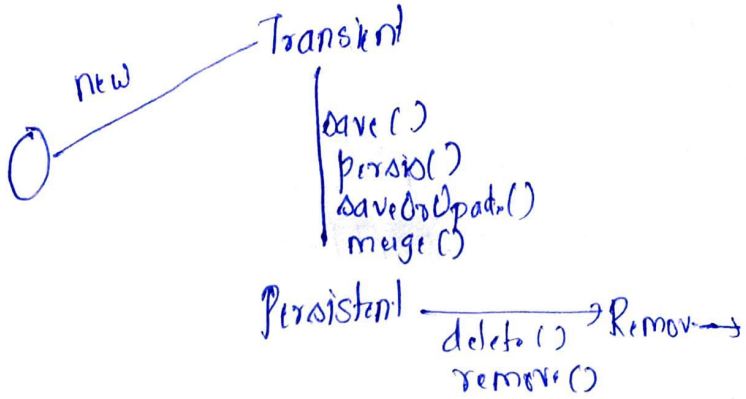
springserve → 8080
localhost:8080/ tomcat srv.

react server
localhost:3000



hibernate entity life cycl

similar to jpa life cy.



persist →

openSession → internally JDBC is attached

Opensession
→ demos → autowiring

Java - 11

⊗ collection.toArray(IntFunctio)
① default method

⊗ Thread.destroy() & Thread.stop()
removed

⊗ Why java 8 ?
↳ main aim was to bring consis-
tency in code
↳ Lambda expression
↳ functional programming.

features of java 8

① Lamda expression
② Stream API
③ Default methods, static methods
④ Functional Interface, optional, Date API

Date API

Lambda expression → is annonymous function (without name, returntype access modifi..

```
public void add(int a, int b){
    System.out.println(a)
}.

(a, b) → System.out.println(a+b)
```

* functional interface can point to Lambda function
* method referencing →
  ::

```
class test {
    public static void test1impl(){
      syso("implementation class");
    }
}

FunctionalInterface i = Test::test2impl;
```

(class name)

2 is referring to i.singleabstract()
↳ method of functional interface.

```
Function... — i = (a) → syso(a)
i.singleAbstractMethod();
```

```
name {"a", "b", "c", "d"}.
Arrays.stream(names).filter(
    x → x.startsWith("s").
sorted().ForEach(syso)
```

Stream

source → intermediate → called

↳ filter sort → map

↳ Collections, List, Sets int. longs, arrays, lines of fil..

Intermediate (Parallelpn

filter → sort →

terminal →

forEach → works on every element.

collect → into collects

→ count, max(), min()
reduce(), Summary Stat

```
IntStream.range(1, 10).forEach
(  )
```

(not inclu)

```
InStream.rang(1, 100).skip(5)
```

skip first five forEach(

```
Stream.of("Ava", "ansari",
    "Albero").sorted().
  findFirst().ifPresent(sysout
```

find first item in list

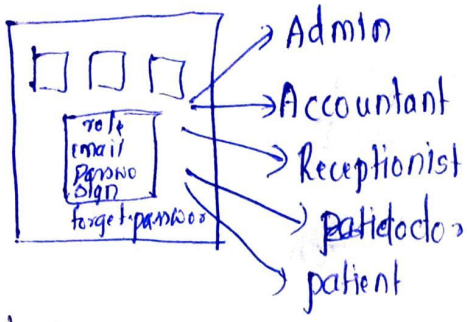
---

map

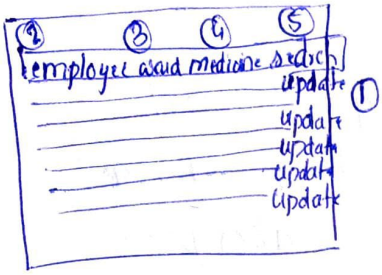
```
Array.stream(int{1, 2, 3..))
  .map(x→x*x).average()
  .ifPresent(sysout);
```

Project interview Questions

1. can you explain CDAC project? draw block diagram?



Admin - page



① update → except email, password
role other details are updatable
→ modal
→

② Employee → add Employee
↳ all details (required details Validated) and added to DB
modal

→ Remove Modal ↳ for doctor ↳ enter empid → remove
if checks if patient are there

③ ward → add ward
↳ ward Type
↳ ward charges
↳ add ward.

Remove ward
↳ checked if patient is present
↳ if present → not removed

④ Medicine → add medicine
↳ Name
↳ charges
↳ added.

Remove medicine
↳ click on Remove

⑤ search → map → filter → contains → accordingly details are filtered.

Accountant

↳ main page → display list
→ update status
change from ~~pa~~ unpaid → paid
→ this is padd then only reception can remove patient

Reception

↳ add patient/rent → validate → if bed is available
→ add patient.

↳ Remove patient if status is paid (auto)

Doctor

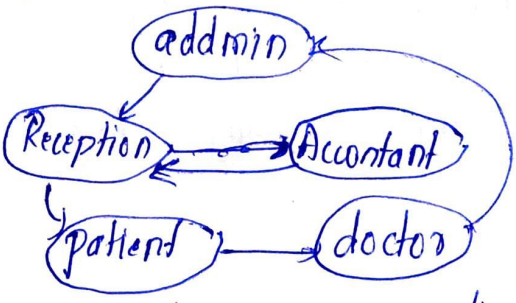
① add prescription → visit of doctor is considered

② add medicine with prescription
Remove medicine

Patient-Page

→ Read only data → Prescriptn, medicines, paid/unp

What's new in your project?
Technology wise → docker, spring-security, ↳ Role based authorisatn
Content wise → Authorised content management, that would do basic syncing of hospital work



→ easily linked with corrusponding medical store

③ Why did you select this technology & framework for this project

→ Why docker

→ Why java →

→ Why react →

Why java → java is easy to learn
java is easy to use
→ huge resources to learn due to old
→ high readability
isEmpty, isInstanceof

→ its object-oriented →
↳ it helps in achieving modularity and ↑ code reusability
↳ Java is platform-independance
↳ we require JRE for executing the java applicatns
↳ we require JDK for development
↳ .class file can be run on any system requirement is only JRE

→ ⊛ Awsome tools → high intellegence, easy maintainability, easy to run, powerful debugging capability,
↳ MAVEN, jenkins,
easy integratn with Desktop, web applicatn,

→ huge community → for every problem there is soluth on internet whereas other languages lag this featur

→ ⊛ huge documentatn

→ open source.

Why react →

→ fast and Responsive UI

→ Component (Reusable Compon

→ App

const function = () => {

Return ( );

}

⊛ no jquery required

⊛ when state is changed auto render takes place

⊛ flexibility to used Routing libraries, axios, no libraries.

⊛ easy → moving from
CSS+HTML+Javascript
→ JSX

⊛ Why docker → they start instantly

→ small images & containers.

→ easily transferrable

→ to overcome virtual machine problems large in size, time to start. each has, OS,

→ portability issue

Containers → easy maintainability

→ just docker-compose down

→ up --build

→ microservice architecture used.

Why gitlab → upto 10 Gb

Why STS open source
- ↳ inbuild starter, initialiser
- ↳ powerfull debugging & maven tools
- ↳ easy start of project.
- ↳

org.springframework.context.ApplicationContext → Spring IOC container → instantiating, configuring, assembling

it reads config metadata before creating object.

configuration → xml / annotatn / java code.

ApplicationContext

ClassPathXmlApplicationContext

→ web.xml → for creating beans

ApplicationContext ctx = new ClassPathXmlApplicationContext("service.xml", "");

How OOP concepts implemented in your project?

Abstraction, encapsulatn, inheritance polymorphism

Abstraction → @controller anotatn, in controller only methods of service layer are allowed, used,

encapsulatn :- in every class it data members & methods are bound together.

Inheritance

→ PatientDao implements JPARepository<>

→ UserServiceDetails

→ userDetails

Polymorphism

more than one constructor, @override tostring,

List → ArrayList<>

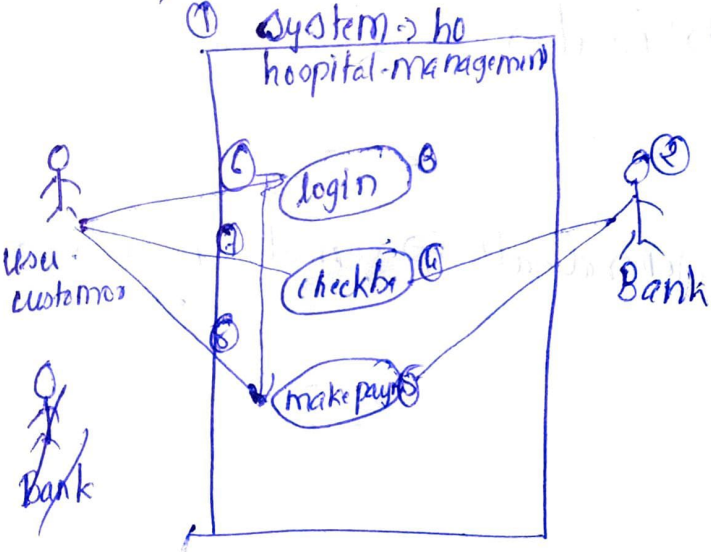
What is use case Diagram?

Actors → users that interact with the system. it can be person organisatn, external objects

System → actions b/w actors & system

Goals → end result of system

→ Actors, use cases, Relationship, syst



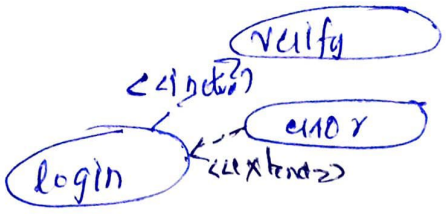
primary actor → user, customer

secondary actor → Bank

Relatn

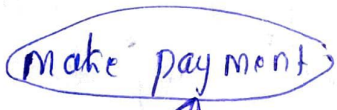
Include →

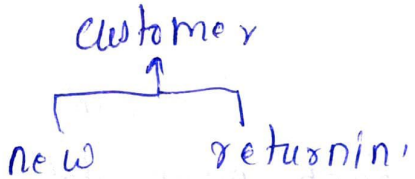
extend →



Include → always executed with base case or without Include base
not possible → every time

Extended → extend use case only if certain condition met

Generalisation → 



---

which advance feature used in your project?

① Docker-compose :- Volume
② Spring-security → multi role based authorisation, token
③ Lombok library →
④ Light. weight docker images → nginx, reverse proxy

spring-data JPA
→ open jdk, mvn, docker-container
→ AWS, ec2 deployment

Q) What was your role and what did you do in it?

Accountant → update status.

Spring-security, Docker part, selecting spring-starters,

spring-security → multi role authorisation

Docker part → reverse proxy in nginx, multi build structure,

→ Accountant → modals, axios, () => {}

component → patient, props, state. = < Navbar

Which development methodology used?

Agile → 2 week target with functional division was made,

sign in page → backend → frontend → integral

docker → Dashboard
→ add prescription
→ add medicine
→ Remove medicine.

Hms → doctor → Spring
→ Reception → =
→ admin → =
→ Patient → =

Storie.

How will you deploy your project?

first deployment → EC2 instance with
→ mysql, openJdk, Apache2, server.jar, build
→ docker container
→ docker-compose → git push git pull. nginx.

Which design pattern have you used in project?

M → pojo, Dao, entities, Dto
V → React
C → @Controller, servlet Dispatcher

Singleton → all @Autowired dependencies are singleton in nature.

template patter → strict naming convention are follow.

What are limitation of your project?

① scop extension of scope is required
② needs to add more beautification to the project
③ folder structure needs more segregation in frontend
④ more technologies such as jenkiens graphal is missing, clusture formation is missing.
⑤ some places comments are not written

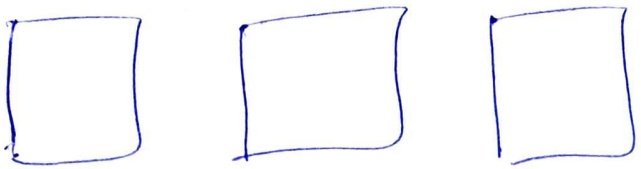
What are difficulties you faced? and how you overcome them?

→ JPA cascading implementation @manytoone @onetoone, @onetomany
git merge --abort.

→ nginx reverse proxy
→ creating docker containers.
→ deploying in Aws,
→ transfering data from local to Aws
→ data update on page refresh
→ Conditional rendering
→ size of ec2 Ram, docker + containers, → optimation
mysql →
small size images,

Docker commands

curl http://localhost:4000



Apache → by default JS is missing.

- from nginx directly using container http://product-service is not working (container)
- ping → database

→ mvn clean install

→ npm run build

→ axios don't work directly inside nginx → hence reverseproxy is used

→ Dockerfile: blueprint to create image.

environment variable → to set in container

→ RUN → execute any linux command ~~not~~ inside container

→ ~~inside~~

COPY . /home/app

↳ executes on host

CMD ["node", "server.js"]

↳ entry point after creating container

→ docker build -t my-app .

Entrypoint →

Entrypoint ["java", "-jar", "/mes.jar"]

↳ When container is booting.

Compose

- automatically N/w created and connection by name.
- Version:

services:

mongo:

host → container

Volume → data persistency

YAML (writing configurat°)

→ is serialisatn language

e.g yaml, json, xml

- very readable
- spaces & indent° are impt

" " → optional

comment

mixo → object

a:
b:
c: } → value