

1. Identify databases?

Oracle	Asp.Net	Java
HTML	Sql Server	My Sql
Java Script	Sqlite3	C#.NET
Mongo DB	CSS	ADO.NET

2. Identify the Java Framework from the given list?

Ado.net	JSP	JDBC
Hibernate	Spring	Spring Boot
RestAPI	Asp.net	Asp.net MVC

3. Identify the UI Technologies from the given list?

Java	C#.net	Html
Javascript	CSS	Python

4. Identify the Scripting Languages from the Given List?

C	HTML	Javascript
Python	TypeScript	VbScript
Angular	ReactJs	Nodejs

5. Identify the editors to write programs?

Notepad	Notepad++	Editplus
Googlechrome	Paint	Excel

6. Identify the Java IDE'S(Integrated Development Environment)

Eclipse	visualstudioEditor	Netbeans
STS	Inteelij	Visualstudiocode

7. Identify different servers

Tomcat	IIS	Internet
Google	FaceBook	JBoss

8. Desktop applications are used for users

- a. Single user b. Two Users c. Multi Users d. System purpose

9. Web Applications are used for Users

- a. Single user b. Two Users c. Multi Users d. System purpose

10. Identify Desktop Applications

Notepad	Antivirus	Facebook
Sathya Tech.com	Google chrome	Youtube
Amazon	Irctc.com	Visual studio

11. Identify Web Applications

Notepad	Antivirus	Facebook
Sathya Tech.com	Google chrome	Youtube
Amazon	Irctc.com	Visual studio

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

12. Html file extension is

- a. .html b. .js c. .cs d. asp e. .exe f.
.dll

13. Java script file extension is

- a. .html b. .js c. .cs d. asp e. .exe f.
.dll

14. To check syntax mistakes we have to use

- a. Compiler b. interpreter c. both d. none

15. Full form of .exe file

- a. Express file b. Executable filec. Error filed.none

16. Write POJO class Rules?

17. Why Framework?

18. What is framework?

19. what are java Framework?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

1. Write POJO class Rules?

2. Jar Stands for_____

3. War Stands for:_____

4. Ear Stands for:_____

5. MVC Stands for:_____

6. Purpose of Presentation Layer?

7. Purpose of Business Layer?

8. Purpose of Data Access Layer?

9. ORM stands for:_____

10. Why JDBC and Use of JDBC?

11. What is Hibernate?

12. Write Use of Hibernate? And write persistence methods also?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

13. Identify jar and war file dependencies web site name?

- a) <https://spring.io/>
- b) <https://start.spring.io/>
- b) <https://mvnrepository.com/>
- d) <https://www.baeldung.com/spring-boot>

14. The purpose of IDE

- a. to develop the application
- b. to run the application
- c. test the application
- d. build the application
- e. all the above

15. Identify the Build Tools from the below list?

- a. PostMan
- b. swagger
- c. Maven
- d. Junit
- e. Gradle

16. Abbreviate POM

- a. Properties Object Model
- b. POJO Object Model
- c. Project Object Model
- d. POJI Object Model

17. STS stands for: _____

18. What is Java Software Name?

19. Write JDBC 5 Steps with code?

20. Types of java Framework?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

1. What is Framework?

2. What is class?

3. What is interface?

4. What is invasive framework? Give examples?

5. What is Non-invasive framework? Give examples?

6. What are the spring Framework modules?

7. ORM stands for: _____

8. POM stands for: _____

9. STS stands for: _____

10. Identifies classes, methods, variables, keywords, identifiers from below code snippet?

```
import java.util.Scanner;
public class MyDetails {
    public static void main(String yakub[])
    {
        int rollno;
        String name;
        float fee;
        System.out.println("Enter Rollno,name and fee values");
        Scanner sc=new Scanner(System.in);
        rollno=sc.nextInt();
        name=sc.next();
        fee=sc.nextFloat();
        System.out.println(rollno+"\t"+name+"\t"+fee);
    }
}
```

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

11. Write POJO Class Rules? With example?

Rules

Example Code:

	<pre>Class Student{ Int stdNo; String stdName; Float stdFee; }</pre>
--	--

12. What is POJI?

S A T H Y A T E C H N O L O G Y

1. Which of the following statements is correct regarding the Spring Framework?

- A** Spring Framework is a heavy-weight solution.
B the Spring Framework is a light-weight solution.
C Both A and B are true.
D None of the above

2. Spring is a _____ framework?

- A** free **B** open source **C** under license **D** proprietary

3. What are the different types of Bean injection?

- A** constructor and setter **B** constructor and getter
C getter and setter **D** setter, getter and constructor

4. Which spring module provides the capability of DI or IOC?

- (A) Spring web (B) Core container (C) AOP (D) Data access

5. Identify the Web Frameworks in the below list?

- a. Hibernate b. spring c. Data JPA d. Spring Boot

6. Identify the Database Frameworks in the below list?

- a. Hibernate b. spring c. Data JPA d. Spring Boot

7. What is spring?**8. Write Spring framework Features?****9. What is class? Write how to create object for class and how many ways?**

SATHYATECHNOLGY

10. MVC implemented with which API's?

M:

V:

C:

11. Write Spring Core Modules?

12. Why spring is NON-INVASIVE Framework?

13. What is singleton class and its rules?

14. write return type of given methods?

public void m1(){ }	public float m1(){ }
public int m1(){ }	public Employee m1(){ }
public String m1(){ }	public Student m1(){ }

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

1. What are the different types of dependency injection?

2. What is singleton class and its rules?

3. Difference between setters methods and constructors?

Constructor	Setter methods

4. Why List Interface and its implemented classes?

5. Why set Interface and its implemented classes?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

6. identifies spring application required files?

- a) Bean.java b) MyClass.java c) Configuration.xml d) TestClass.java

7. Bean Configuration ways?

- a) 1 b) 2 c) 3 d) 4

8. Fill the below Bean class code?

```
Class MyBean {  
    int id;  
    String name;  
    String salary;
```

9. Consider Class name is MyBean, properties are id, name and salary.

Write the xml based Configuration code below? (Config.xml)

```
<XML><DTD>  
<beans>  
<bean>  
  
<bean>  
</beans>
```

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

1. What is IOC in spring?

2. What is IOC Life Cycle in spring?

3. What are Dependency Injection and Types Dependency Injection?

4. Identify spring configuration ways?

- a) Interface based configuration b) xml based configuration
- c) Java based configuration d) Annotation Based Configuration

5. Identify Setter Injection XML Tag?

- a). <property> </property>
- b) <constructor-arg/>
- c) <context:annotation-config/>
- d) <context:component-scan>

6. What is website name to download jars and wars ?

- A) <https://mvnrepository.com/>
- B) <https://mavenrepository.com/>
- C) <https://spring.io/>
- D) <https://start.spring.io/>

7. Identify the current version of spring Boot?

- A). 0.9 2003
- B). 1.0 March 24, 2004 First production release.
- C). 2.0 2006
- D). 3.0 2009
- E). 4.0 2013
- F). 5.0 2017

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

8). Identifies Spring Container Types?

- | | |
|---------------------------------------|--------------------------------------|
| A). Spring Core Container | B). Spring Stable Container |
| C). Spring Secondary Container | D). Spring Advanced Container |

9. Write Difference between Spring Core and Advanced Container?

Spring Core Container	Spring Advanced Container

10. Write Syntax for TestClass.java with Advanced Container?

11. Identify Constructor Injection XML Tag?

- | | |
|---|--|
| A). <property> </property> | B) <constructor-arg/> |
| C) <context:annotation-config/> | D) <context:component-scan> |

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

1. Prepare Spring Core Application for Display Employee Details?

Fields are empId, empName, empSalary, empDesignation With Setter injection?

Employee.java

Spconfig.xml Test.Java

--	--

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

2. Prepare Spring Core Application for Display Employee Details?

**empId,empName,empSalary,empDesignation With Constructor
injection?**

1. Employee.java

Spconfig.xml Test.Java

--	--

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

1. Write Steps to create spring core application with maven in eclipse IDE?

2. Identify spring core application required jars?

- A). Spring core b) spring aop c) junit d) spring context

3. Identify list collection xml tags in .xml file?

- A) <set><value/> B). <map><entry key="" value="" />
C) <props> <prop key="">value</prop> D). <list> <value/>

4. Identify set collection xml tags in .xml file?

- A) <set><value/> B). <map><entry key="" value="" />
C) <props> <prop key="">value</prop> D). <list> <value/>

5. Identify Properties collection xml tags in .xml file?

- A) <set><value/> B). <map><entry key="" value="" />
C) <props> <prop key="">value</prop> D). <list> <value/>

6. Identify map collection xml tags in .xml file?

- A) <set><value/> B). <map><entry key="" value="" />
C) <props> <prop key="">value</prop> D). <list> <value/>

7). which xml Tag is used to inject Object reference from one class to Another classs?

- A) <property> B)<constructor-arg> C) <property name="fiedname">
ref ="object_name"/> D) < constructor-arg ref ="object_name"/>

8). what is map in collection and its rules and its implemented class names?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

9). Write use of POM.xml?

10). What is Use of `toString()` method in bean class?

11). what is bean auto wiring?

12). Identify auto wiring annotation from below option?

- a) `@Required` b) `@Scope` c) `@Autowired` d) `@Qualifier`

13) What is Annotations?

14). what is IOC Lifecycle?

15. Differentiate between BeanFactory and ApplicationContext.

BeanFactory	ApplicationContext

S
A
T
H
Y
A

T
E
C
H
N
O
L
O
G
Y

1). what is bean auto wiring?

2). Identify auto wiring annotation from below option?

- a) @Required B) @Scope C) @Autowired D) @Qualifier

3). Write use of @Qualifier Annotation?

4). Write use of @Required Annotation?

5). Write use of @Configuration annotation?

6). Write use of @Bean Annotation?

7). Identifies java based configuration annotations?

- a). @Component b). @ComponentScan c). @Configuration
d). @Controller e). @Bean f). @Service

8).Identifies annotation based configuration annotations?

- a). @Component b). @ComponentScan c). @Configuration
d). @Controller e). @Bean f). @Service
g). @Repository h). @Autowired i). @Qualifier

8). Write use of @Controller Annotation?

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

9). How to Enable Annotation in Spring IOC container?

10).which implemented class is used to load java configuration file into Spring IOC container and write code below?

11).which implemented class is used to load .xml configuration file into Spring IOC container and write code below?

12. @SpringBootApplication annotation is a combination of how many annotations and writes purpose also?

S
A
T
H
A

T
E
C
H
N
O
L
O
G
Y

1). Create Spring core Application for Student Marks, Total marks, average and grade with setter injection with .xml based configuration?

Variables are: stdId, stdName, stdFee, sub1, sub2, sub3.

Output: Print in following format

StudentId:101

StudentName: Aaryan

StudentFee:50000

Sub1marks:80

Subject2 marks:90

Subject3 marks:70

Total: 280

Average: 75.00

Grade:A Grade

2). Create Spring core Application for Student Marks, Total marks, average and grade with constructor injection with .xml based configuration?

Variables are: stdId, stdName, stdFee, sub1, sub2, sub3.

Output: Print

StudentId:101

StudentName: Aaryan

StudentFee:50000

Sub1marks:80

Subject2 marks:90

Subject3 marks:70

Total: 280

Average: 75.00

Grade:A Grade

3) Create spring core application for Dependency injection in the form of collection by using Java Based Configuration.

Input Objects are:

List EmpNames;

Set EmpIds;

Map<String, Integer> DesignationDetails;

Properties DepartInfos;

OutPut:

-----Employee Names Are-----

Yakub

KannaBabu

Srinivas

Venkat

-----Employee Ids Are-----

101

102

103

104

-----Employee Designation Details with No of Employees Are-----

Accounts: 120

Software Engineers: 240

-----Employee DepartInfos Are-----

Accounts: Vikram

Software Engineers: Yakub

Hardware: Aryan Rathod

4. Create Spring Application with Following Details By Using Setter Injection.

Consider below example with LIST And Map Collection Objects

Class Name: Employee

Instance Variables Are:empId

List<Integer> empProjects; // Number of Projects

Map<Integer, String> empMap //Number of projects and Company Name

OutPut

Employee Id: 1001

List Of Projects:[20,40,10,60,100]

Number of Projects With Company Name:{20:"TCS",10:"Wiprow",3:"SathyaTech"}

5. Create Spring Application with following Details by Using Java Based Configuration?

Consider below example with LIST And Map Collection Objects

Class Name: Employee

Instance Variables Are: empId

List<Integer> empProjects; // Number of Projects

Map<Integer, String> empMap //Number of projects and Company Name

OutPut

Employee Id: 1001

List Of Projects:[20,40,10,60,100]

Number of Projects With Company Name:{20:"TCS",10:"Wiprow",3:"SathyaTech"}

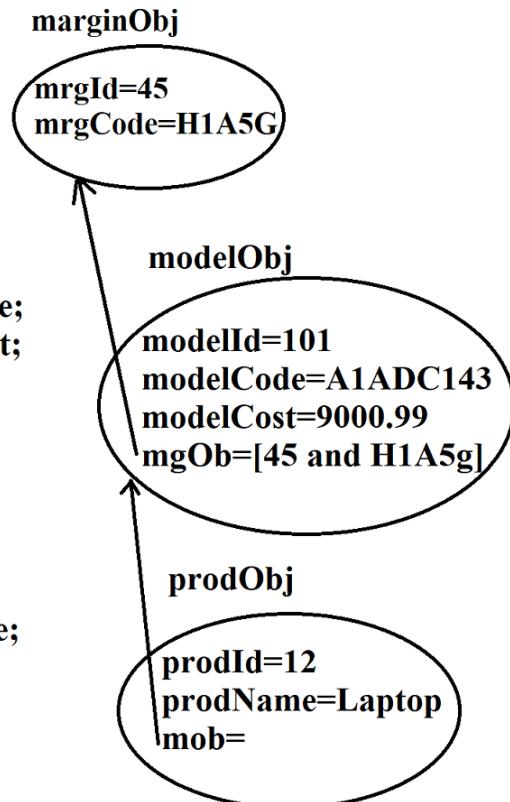
6. Create a spring Application with following specification by using Java Based Configuration? Inputs Takes form Key board?

Consider The Following Three Classes and Create a Spring Application

```
class Margin{
    private int mrgId;
    private String mrgCode;
}
```

```
class Model {
    private int modelId;
    private String modelCode;
    private double modelCost;
    private Margin mgOb;
}
```

```
class Product{
    private int prodId;
    private String prodName;
    private Model mob;
}
```



7. Create a Spring Application By Using Setter In injection?**Application-3: Consider Below example With List of Two Models**

```
class Product{
    int productId;
    String productName;
    List<Model> modelObjs;
}
```

Here Prepare Business Method
Print Product Details

```
class Model {
    int modelId;
    String modelCode;
    double modelCost;
    Map<Integer,String> marginObjs;
}
```

Note: Required toString() method

```
class Margin {
    int marginId;
    String marginCode;
}
```

Note: Required toString() method

8. Create a Spring Application with following Details by Using Java Based configuration. And Takes Dynamic values from key board?**Application-3: Consider Below example With List of Two Models**

```
class Product{
    int productId;
    String productName;
    List<Model> modelObjs;
}
```

Here Prepare Business Method
Print Product Details

```
class Model {
    int modelId;
    String modelCode;
    double modelCost;
    Map<Integer,String> marginObjs;
}
```

Note: Required toString() method

```
class Margin {
    int marginId;
    String marginCode;
}
```

Note: Required toString() method

9. Purpose of @Configuration Annotation?**10. Purpose of @Bean Annotation?****11. Purpose of @Scope Annotation and Scope Modes?**

1. Identifies Spring IOC configuration ways?

- A). xml based configuration**
- B). Interface based configuration**
- C). java Based configuration**
- D) Annotation Based Configuration**

2). write use of @Controller Annotation?

3). write use of @Service Annotation?

4). write use of @Repository Annotation?

5). write use of @ComponentScan Annotation?

6). write use of @Value Annotation?

7). write Spring IOC Scope modes? And Annotation name with values?

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

8. Map the below Annotations to valid Bean classes

@Controller	@Bean	@Service	@Configuration
@Repository	@Component	@ComponentScan	@Autowired
@Qualifier	@Value		

```
Class public TestApp{

Public static void main(String args[]){
}

Public myconfigarator{

Public MyBean createBean(){

Return new MyBean();
}

public class MyController
{
public ModelAndView handler(HttpServletRequest
req,HttpServletResponse res){
    return new ModelAndView("success");
}
}

Public class MyService{
Public void SaveEmployee(Employee e)
{ -----code----- }

}

Public interface MyRepositiry implements CrudRepository
{



Private MyBean mybean;

}
```

1 Why Spring DAO and its method?

2 Why Spring ORM and its method?

3 Write DriverManagerDataSource properties Syntax?

4. Use of @Controller Annotation in spring Boot?

5. Use of @Service Annotation in spring Boot?

6. Use of @Repository Annotation in spring Boot?

7. Use of @SpringBootApplication Annotation in spring Boot?

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

8. Prepare Spring core Application. For create, insert and select operation for product Table?

Fields are: proId, proName, proPrice, proQuantity.

Sathya Technology

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

1 Write Steps to create Spring Boot Application with spring initializer?

2. Write Steps to create Spring Boot Application with STS tool?

3. Use of @SpringBootApplication annotation in Spring Boot?

4. Write Spring WEB-MVC Flow?

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

5. identifies spring boot web application required jars?

- A) Spring Core B) **spring-boot-starter-web** C) **spring-boot-devtools**
D) **spring-boot-starter-data-jpa** E) **com.h2database**

6. Identifies Spring boot applications IDE's?

- A). notepad B). VS code C) eclipse D) STS D) intellij

7. Identifies default database of Spring Boot application?

- A). Mysql B) Oracle C) h2 Database C). Mysql Server

8. Write purpose of application.properties file in spring boot?

9. Write model class Rules?

10. Write the Spring Boot Application Architecture with folder names and file names?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

1. create spring boot web application to given number even or odd.

Check The Given Number is Even Or Odd

Enter a Value :

25 is Positive Number

2. Create Spring Boot web application for registration page?

User Registration Form		Out Put
Name:	<input type="text"/>	
Password:	<input type="text"/>	
Confirm Password:	<input type="text"/>	
Email:	<input type="text"/>	
Address:	<input type="text"/>	
Mobile Number:	<input type="text"/>	
Gender:	<input type="radio"/> Male	<input type="radio"/> Female
Hobbies:	<input type="checkbox"/> Gaming	<input type="checkbox"/> Programing
City:	<input type="text" value="--Select--"/>	
State:	<input type="text" value="--Select--"/>	
<input type="button" value="Register"/>		

My Name: Yakub
PassWord: Yakub
Email: yakub@gmail.com
Address: Hyderabad
Mobile Number: 9876543210
Gender: Male
Hobbies: Gmaing, programing
City: Hyderabad
State: Telangana

S
A
T
H
Y
A

T
E
C
H
N
O
L
O
G
Y

3. Create spring boot web application to print Invoice?

Total Bill Program

Enter Item Number :

Enter Item Name :

Enter Item Quantity :

Enter Item Cost :

Item Number : 1
Item Name : sugar
Item Quantity : 2
Item Cost : 40
Total Bill Amount is : 80

4. Create Spring Boot web application to print employee Salary?

Employee Invoice Application	Conditions are:
Employee No: <input type="text"/>	TA DA HRA PF
Employee Name: <input type="text"/>	if salary<30000 7% 9% 11% 15%
Employee Salary: <input type="text"/>	if salary>=30000 and salary<50000 12% 13% 17% 22%
	if salary>=50000 17% 19% 21% 25%
	Output: print in following format

	Employee No: 120 Employee Name: Aaryan Employee Salary: 45000 TA: 1000 DA: 2000 HRA: 1500 PF: 2000 Gross Salary: 49500 Net Salary: 47500

5. Spring Boot Application for Login Page:

New User Login Here

User Name:

Password:

if username and password is correct
print sucess message.
else
print wrong message

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

- 1. Create spring boot web application for Print Electricity Bill.**
Write Required Files are Jars

Spring Boot Electricity Bill Consumer Id: <input type="text"/> Consumer Name: <input type="text"/> Current Reading: <input type="text"/> Previous Reading: <input type="text"/> <input type="button" value="PrintBill"/>	OutPut Format Consumer Id:1002 Consumer Name: Aryan Current Reading: 450 Previous Reading: 50 Units:400 Total Bill: 1200.00 <u>Conditions</u> if units<300 --> rate 1.75 if units>=300 and units<500 --- rate --> 3.25 if units>=500 rate ----> 7.25
--	---

- 2. Create Spring Boot web application Print Amazon Invoice with following conditions?**
Write Required Files are Jars

Amazon Product Invoice Costomer Id: <input type="text"/> Costomer Name: <input type="text"/> Mobile Number: <input type="text"/> Product Name: <input type="text"/> Price: <input type="text"/> Quantity: <input type="text"/> <input type="button" value="PrintInvoice"/>	output format Customer Id: 1001 Name: Yakub Mobile No: 9876543210 ProductName: Dell LapTab Price: 40000.00 Quantity:1 Total Amount: 40000.00 Discount Amount: 3000.00 GST: 1000.00 Invoce_Bill: 38000.00 <u>Conditions Are:</u> ----- if total<2500 --> 10% discount on total price if total>=2500 and total<5000 --> 15% if total>=5000 --> 30%
--	--

3. Create Spring Boot web Application for Employee Invoice with following Conditions?

Write here required files are Jars

Employee Invoice Application

Employee No:

Employee Name:

Employee Salary:

PrintInvoice

Conditions are:

	TA	DA	HRA	PF
if salary<30000	7%	9%	11%	15%
if salary>=30000 and salary<50000	12%	13%	17%	22%
if salary>=50000	17%	19%	21%	25%

Output: print in following format

Employee No: 120
 Employee Name: Aaryan
 Employee Salary: 45000
 TA: 1000
 DA: 2000
 HRA: 1500
 PF: 2000
 Grass Salary: 49500
 Net Salary: 47500

4. Spring Boot Application for Login Page:

Write Required Files are Jars

New User Login Here

User Name:

Password:

Login

**if username and password is correct
 print sucess message.
 else
 print wrong message**

1. What is Annotations in Spring Boot?

2). Purpose of @Controller Annotation?

3). Purpose of @Service Annotation?

4). Purpose of @Repository Annotation?

5). what is Auto wiring in Spring Boot?

6). what is role of model class in MVC. Write Rules?

7) Explain Spring WEB-MVC execution Flow?

8). what is Spring Boot and Use of Spring Boot?

9). What MVC and Use of MVC in Spring Boot?

10). What Spring Boot starters for Develop Web application?

11). Write the steps to create Spring Boot application In STS?

12). Write completes Spring Boot Application Project Structure with All file names?

13). Purpose of Spring Dev Tools Starters?

14.) Purpose of Sprig Web Starters?

15). Purpose of Data JPA Starter?

16). Write about POM.xml file?

17). Difference between spring and Spring Boot?

Spring	Spring Boot

18). Purpose of @RequestMapping Annotation?

19). Purpose of @RequestParam Annotation?

20). Purpose of @RequestBody Annotation?

21).Purpose of @PathVariable Annotation?

1. Define DATA JPA and Purpose of DATA JPA?

2). Write Data JPA Interfaces and its Implemented Methods?

3). what is Web Application?

4). Define API and Purpose of API?

5). what is web API?

6). what is a web services?

7) SOAP Stands for. _____

8) REST Stands for. _____

9). MVC stands for. _____

10). STS stands for. _____

11).What is REST?

12). what is RESTFull API?

13.). what is RESTFull web services?

14). Write REST API call Request Types?

15). Difference between CrudRepository and JpaRepository?

CrudRepository	JpaRepository

1) SOAP Stands for. _____

2) REST Stands for. _____

3). MVC stands for. _____

4).What is REST?

5). what is RESTFull API?

6). what are RESTFull web services?

7). Purpose of @PostMapping?

8). Purpose of @GetMapping?

9). Purpose of @PutMapping?

10). Purpose of @PatchMapping?

11). Purpose of @RequestBody?

12). Purpose of @RequestMapping?

13). Purpose of @PathVariable?

14). Difference between CrudRepository and JpaRepository?

CrudRepository	JpaRepository

15). Difference between spring and Spring Boot?

Spring	Spring Boot

1 Write Steps to create Spring Boot Application with spring initializer?

2. Write Steps to create Spring Boot Application with STS tool?

3. Use of @SpringBootApplication annotation in Spring Boot?

4. Write Spring WEB-MVC Flow?

S
A
T
H
A
T
E
C
H
N
O
L
O
G
Y

5. identifies spring boot web application required jars?

- A) Spring Core B) **spring-boot-starter-web** C) **spring-boot-devtools**
D) **spring-boot-starter-data-jpa** E) **com.h2database**

6. Identifies Spring boot applications IDE's?

- A). notepad B). VS code C) eclipse D) STS D) intellij

7. Identifies default database of Spring Boot application?

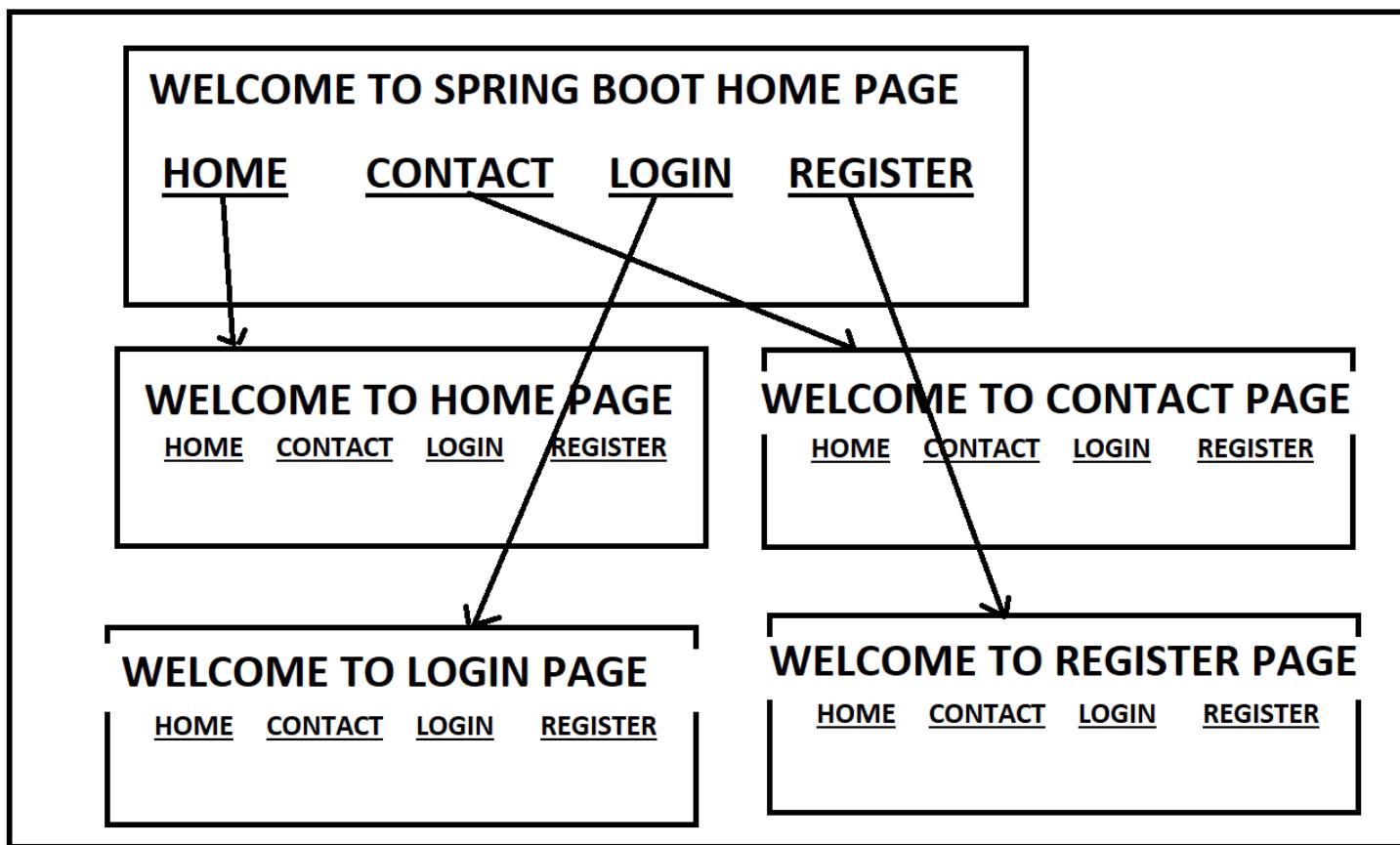
- A). Mysql B) Oracle C) h2 Database C). Mysql Server

8. Write purpose of application.properties file in spring boot?

9. Write model class Rules?

10. Write the Spring Boot Application Architecture with folder names and file names?

S
A
T
H
Y
A
T
E
C
H
N
O
L
O
G
Y

1. Spring Boot Web Application For View To View Communication?

Write Above Application Project folder Structure

2. Spring Boot Web Application For View To View Communication

By using following JSP web Forms.

Create Spring Boot Web application for View To View Communication

home.jsp

login.jsp

EMPLOYEE MANAGEMENT SYSTEM

[About](#) [Contact](#) [Login](#) [Register](#)

Welcome To Home Page

EMPLOYEE MANAGEMENT SYSTEM

[About](#) [Contact](#) [Login](#) [Register](#)

Employee Login Form

Name:

Password:

Login

EMPLOYEE MANAGEMENT SYSTEM

[About](#) [Contact](#) [Login](#) [Register](#)

Welcome To Contact Page

Mobile:

Website Name:

mail-id:

conatct.jsp

EMPLOYEE MANAGEMENT SYSTEM

[About](#) [Contact](#) [Login](#) [Register](#)

Employee Registartion Form

Name:

Password:

Email:

Gender: Male Female

Mobile No:

Address:

Register

about.jsp

register.jsp

1. Spring Boot Web Application CRUD FOR Electricity Bill Invoice

Spring Boot Electricity Bill

Consumer Id:

Consumer Name:

Current Reading:

Previous Reading:

SaveBill

Conditions

if units<300 --> rate 1.75
 if units>=300 and units<500 --- rate --> 3.25
 if units>=500 rate ----> 7.25

Table:ElectricityBill

cid	name	current_reading	previous_reading	units	service_charges	total_bill	Delete	Edit
11	viondo	900	50	850	120.00	1500.99	Delete	Edit

2. Spring boot web application CRUD for Product Invoice

Amazon Product Invoice

Customer Id:

Customer Name:

Mobile Number:

Product Name:

Price:

Quantity:

PrintInvoice

create Table With :product (Model class Name Also:Product)

cid	name	mobile_no	prodName	price	quantity	total	discount	gst_amt	invoice_bill
10	Dilip	9876543210	Dell	35000.00	1	35000.00	3000.00	1000.00	33000.00

Conditions Are:

 if total<2500 --> 10% discount on total price
 if total>=2500 and total<5000 --> 15%
 if total>=5000 --> 30%

Perfrom Complete CRUD Operations

prodView.jsp

cid	name	mobile_no	prodName	price	quantity	total	discount	gst_amt	invoice_bill	Delete	Update
10	Dilip	9876543210	Dell	35000.00	1	35000.00	3000.00	1000.00	33000.00	Delete	Update

3. Spring Boot web Application CRUD For Employee Invoice

Employee Invoice Application

Employee No:

Employee Name:

Employee Salary:

SaveInvoice

Conditions are:

	TA	DA	HRA	PF
if salary < 30000	7%	9%	11%	15%
if salary >= 30000 and salary < 50000	12%	13%	17%	22%
if salary >= 50000	17%	19%	21%	25%

Table: Employee_Invoice

empId	empName	salary	TA	DA	HRA	PF	GROSS_SALARY	NET_SALARY	DELETE	EDIT
10	yakub	90000	3000	5000	2000	5000	100000	95000	Delete	Edit

4. Create Spring Boot Web Application and Rest API Application for Student Marks Result.

Spring Boot Web Appn and Rest API for Student Marks

Sid:

First Name:

Last Name:

sub1:

sub2:

sub3:

PrintResult

Table: Student (Model class Name: Student)

sid	fullname	sub1	sub2	sub3	total	percentage	grade	Delete	Update
101	Aryan	90	80	70	270	72.00	A grade	Delete	Update
102	Ramesh	90	50	60	200	65.00	B Grade	Delete	Update

Perform Complete CRUD Operations

Conditions Are

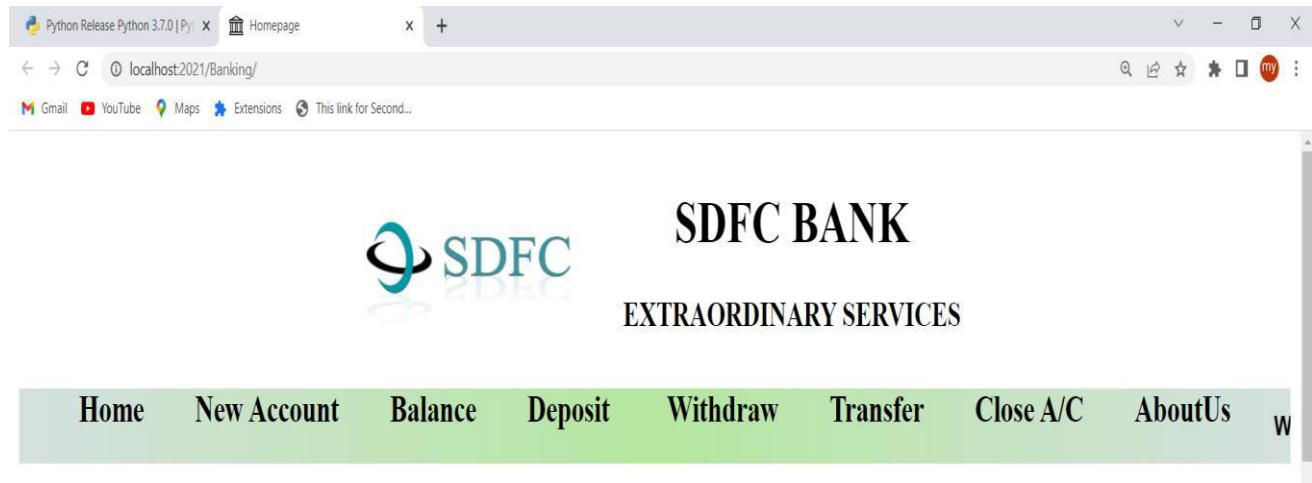
```

if total>=70 ---> A grade
if total>=60 ---> B grade
if total>=50 ---> C grade
else
Failed

```

SPRING BOOT BANKING WEBAPPLICATION APPLICATION TASK **Sathya Technologies**

Home page



Here create Home page with Following Links like NewAccount, Balance, Deposit, Withdraw Transfer and CloseA/c.

New Account Page.



Banking
Banking
Autopay
Banking
Banking
Banking
Banking
Banking
Banking
Banking
Banking

Account Number:	<input type="text"/>
Name:	<input type="text"/>
Password:	<input type="text"/>
Confirm_Password:	<input type="text"/>
Amount:	<input type="text"/>
Address:	<input type="text"/>
Mobile No:	<input type="text"/>

Say hello to a unique
your requirements and
We offer a wide range
the major curr

Accounts Pay

QNB provides a cor
services, so please take
more familiar with t
branches or contact u
comments and

Here New Account user create his/her account and save data in database.

If you have already an account you can check balance, deposit, with-draw, money transfer and can close your account by clicking on respective link.

View Balance



SDFC BANK

EXTRAORDINARY SERVICES

Home

New Account

Balance

Deposit

Withdraw

Transfer

Close A/C

AboutUs

ICES

BALANCE FORM

WE

pp Banking
ep Banking
Autopay
e Banking
e Banking
e Banking
h Banking
en Banking
Banking
e Banking

Account No.:

Name:

Password:



Say hello to a unique bank requirements and guarantee a wide range of accounts in major currencies.

Accounts Paym

QNB provides a comp

Here User check his balance with valid account no, name and password details.

1. On filling Correct Detail:

NEW ACCOUNT	BALANCE	DEPOSIT	WITHDRAW	TRANSFER	CLOSE A/C	Contact Us										
Services <ul style="list-style-type: none"> www.javatpoint.com www.javacstpoint.com www.javatpoint.com/forum.jsp 	Welcome Rishi kumar <table border="1"> <thead> <tr> <th>ACCOUNT NO</th><th>USERNAME</th><th>AMOUNT</th><th>ADDRESS</th><th>PHONE</th></tr> </thead> <tbody> <tr> <td>1</td><td>Rishi kumar</td><td>50000</td><td>delhi</td><td></td></tr> </tbody> </table>	ACCOUNT NO	USERNAME	AMOUNT	ADDRESS	PHONE	1	Rishi kumar	50000	delhi						
ACCOUNT NO	USERNAME	AMOUNT	ADDRESS	PHONE												
1	Rishi kumar	50000	delhi													



Accelerate your career with the top-notch software training institute in Hyderabad

**JAVA FULL STACK
.NET FULL STACK**

DAILY 8 HOURS TRAINING | CLASSROOM TRAINING |
100 % JOB GUARANTEE

[Join Now](#)

[sathyatech.com](#)

[9100920092 - 9100940094](#)



2. Can deposit money:



SDFC BANK
EXTRAORDINARY SERVICES

[Home](#) [New Account](#) [Balance](#) [Deposit](#) [Withdraw](#) [Transfer](#) [Close A/C](#) [AboutUs](#)

ICES

app Banking
ep Banking
Autopay
e Banking
e Banking
te Banking
h Banking
en Banking
e Banking
e Banking

DEPOSITE FORM

Account No.:

Name:

Password:

Amount:

WE



Say hello to a unique bank requirements and guarantee offer a wide range of accounts major currencies

Accounts Payments

ONLINE services & more

On filling correct Detail:

NEW ACCOUNT	BALANCE	DEPOSIT	WITHDRAW	TRANSFER	CLOSE A/C	Contact Us
Services	<div style="border: 1px solid red; padding: 5px; text-align: center;"> Account Info your balance has increased 60000 </div>					
www.javatpoint.com						



Accelerate your career with the top-notch software training institute in Hyderabad

JAVA FULL STACK
.NET FULL STACK

DAILY 8 HOURS TRAINING | CLASSROOM TRAINING |
100 % JOB GUARANTEE

[Join Now](#) | sathyatech.com | [9100920092 - 9100940094](tel:9100920092)



3. Can withdraw money:


SDFC BANK
 EXTRAORDINARY SERVICES

[Home](#)
[New Account](#)
[Balance](#)
[Deposit](#)
[Withdraw](#)
[Transfer](#)
[Close A/C](#)
[About Us](#)

SERVICES

- WhatsApp Banking
- Doorstep Banking
- UPI Autopay
- Mobile Banking
- Online Banking
- Remote Banking
- Branch Banking
- NextGen Banking
- Phone Banking
- Online Banking

WITHDRAW FORM

Account No.:

Name:

Password:

Amount:

Submit
Clear

WEI



Say hello to a unique bank requirements and guarantee offer a wide range of accounts major currencies

Accounts Payments

QNB provides a comprehensive banking services please take a look

On filling Correct Detail:

NEW ACCOUNT	BALANCE	DEPOSIT	WITHDRAW	TRANSFER	CLOSE A/C	Contact Us
<p>Services</p> <p>www.javatpoint.com</p>	<p>Account Info</p> <p>your balance has decreased 20000</p>					

Accelerate your career with the top-notch software training institute in Hyderabad

JAVA FULL STACK .NET FULL STACK

DAILY 8 HOURS TRAINING | CLASSROOM TRAINING |
100 % JOB GUARANTEE

Join Now
|
sathyatech.com
|
9100920092 - 9100940094




4. Can transfer Money:



SDFC BANK
EXTRAORDINARY SERVICES

[Home](#)
[New Account](#)
[Balance](#)
[Deposit](#)
[Withdraw](#)
[Transfer](#)
[Close A/C](#)
[AboutUs](#)

SERVICES

- WhatsApp Banking
- Doorstep Banking
- UPI Autopay
- Mobile Banking
- Online Banking
- Remote Banking
- Branch Banking
- NextGen Banking
- Phone Banking
- Online Banking

TRANSFER FORM

Account No.:	<input type="text"/>
Name:	<input type="text"/>
Password:	<input type="text"/>
Target Account No.:	<input type="text"/>
Amount:	<input type="text"/>

Submit
Clear

WELCOME..



Say hello to a unique banking experience that meets all your requirements and guarantees personalised service. We offer a range of accounts, deposits and in all the major currencies, including the below:

On filling Correct Detail:

NEW ACCOUNT	BALANCE	DEPOSIT	WITHDRAW	TRANSFER	CLOSE A/C	Contact Us
Services				Accounts Info the target account balance is 13000 reduced balance is 10000		
www.javatpoint.com www.javacstpoint.com						

Accelerate your career with the top-notch software training institute in Hyderabad

JAVA FULL STACK .NET FULL STACK

DAILY 8 HOURS TRAINING | CLASSROOM TRAINING | 100 % JOB GUARANTEE

Join Now
sathyatech.com | 9100920092 - 9100940094




SPRING BOOT BANKING WEBAPPLICATION APPLICATION TASK **Sathya Technologies**

5. Can close Account:

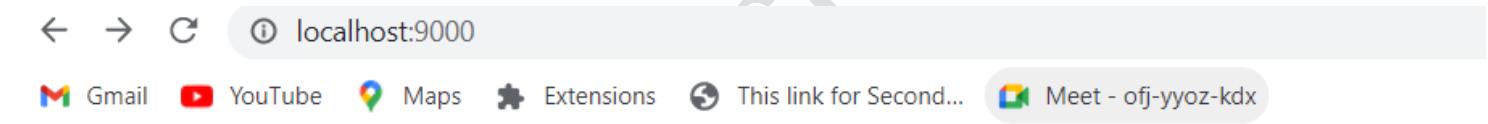
The screenshot shows a web application for SDFC Bank. At the top, there's a navigation bar with links for Home, New Account, Balance, Deposit, Withdraw, Transfer, Close A/C, and AboutUs. Below this is a section titled 'EXTRAORDINARY SERVICES' with a list of banking services. To the right, there's a 'CLOSE ACCOUNT' form with fields for Account No., Name, and Password, along with 'Submit' and 'Clear' buttons. On the far right, there's a 'WELCOME..' section featuring a group of professionals and a descriptive paragraph about the bank's services. The browser's address bar shows the URL 'localhost:2021/Banking/Close.jsp'.

On filling Correct Detail:

The screenshot shows a website for Sathya Technologies. The header has a navigation menu with links for New Account, Balance, Deposit, Withdraw, Transfer, Close A/C, and Contact Us. Below the menu, there's a 'Services' section and a welcome message for a user named Rishi kumar. The main content area features a large banner for Java and .NET full stack training, which includes a woman holding a laptop and the text 'ALL THE BEST'.

- 1. Spring Boot Web Application For Registration And Login and Logout?
Write Below Required Files and Jars**

Files Are	Jars Are

Home.jsp**Welcome To Home Page**[Add New Customer](#) [Login Here](#)**Register.jsp**

← → ⌛ ⓘ localhost:9000/add

Gmail YouTube Maps Extensions This link for Second... Meet - ofj-yyoz-kdx

Fill below form to register

Enter your Id:

Enter your name:

Enter your Password:

Register

Login.jsp

← → ⌛ ⓘ localhost:9000/loginform

Gmail YouTube Maps Extensions This link for Second... Meet - ofj-yyoz-kdx

Customer Login Page

Enter your name:	
Enter your Password:	
Login	

← → ⌛ ⓘ localhost:9000/login

Gmail YouTube Maps Extensions This link for Second... Meet - ofj-yyoz-kdx

Welcome To Cutomer Home Home Page

[Add New Customer](#) [Login Here](#)

1. Spring Boot Web Application CRUD FOR Electricity Bill Invoice

Spring Boot Electricity Bill

Consumer Id:

Consumer Name:

Current Reading:

Previous Reading:

SaveBill

Conditions

if units<300 --> rate 1.75
 if units>=300 and units<500 --- rate --> 3.25
 if units>=500 rate ----> 7.25

Table:ElectricityBill

cid	name	current_reading	previous_reading	units	service_charges	total_bill	Delete	Edit
11	viondo	900	50	850	120.00	1500.99	Delete	Edit

2. Spring boot web application CRUD for Product Invoice

Amazon Product Invoice

Customer Id:

Customer Name:

Mobile Number:

Product Name:

Price:

Quantity:

PrintInvoice

create Table With :product (Model class Name Also:Product)

cid	name	mobile_no	prodName	price	quantity	total	discount	gst_amt	invoice_bill
10	Dilip	9876543210	Dell	35000.00	1	35000.00	3000.00	1000.00	33000.00

Conditions Are:

 if total<2500 --> 10% discount on total price
 if total>=2500 and total<5000 --> 15%
 if total>=5000 --> 30%

Perfrom Complete CRUD Operations

prodView.jsp

cid	name	mobile_no	prodName	price	quantity	total	discount	gst_amt	invoice_bill	Delete	Update
10	Dilip	9876543210	Dell	35000.00	1	35000.00	3000.00	1000.00	33000.00	Delete	Update

3. Spring Boot web Application CRUD For Employee Invoice

Employee Invoice Application

Employee No:

Employee Name:

Employee Salary:

SaveInvoice

Conditions are:

	TA	DA	HRA	PF
if salary < 30000	7%	9%	11%	15%
if salary >= 30000 and salary < 50000	12%	13%	17%	22%
if salary >= 50000	17%	19%	21%	25%

Table: Employee_Invoice

empId	empName	salary	TA	DA	HRA	PF	GROSS_SALARY	NET_SALARY	DELETE	EDIT
10	yakub	90000	3000	5000	2000	5000	100000	95000	Delete	Edit

4. Create Spring Boot Web Application and Rest API Application for Student Marks Result.

Spring Boot Web Appn and Rest API for Student Marks

Sid:

First Name:

Last Name:

sub1:

sub2:

sub3:

PrintResult

Table: Student (Model class Name: Student)

sid	fullname	sub1	sub2	sub3	total	percentage	grade	Delete	Update
101	Aryan	90	80	70	270	72.00	A grade	Delete	Update
102	Ramesh	90	50	60	200	65.00	B Grade	Delete	Update

Perform Complete CRUD Operations

Conditions Are

```

if total >= 70 ---> A grade
if total >= 60 ---> B grade
if total >= 50 ---> C grade
else
Failed

```



Name :

Ph no :

Email :

Create REST full API using the bellow technology

• Java 17 with Spring Boot and React JS

- 1. Create Employee entity and save below Employee in database (useany database of your choice).**
- 2. You should be able to perform CRUD operations on Employeeentity to add/update/delete/get/get All Employees.**
- 3. Optionally add React UI for adding and removing Employee with following output screen shots. With React Single Page Applications**
- 4. Add React Form validations also.**

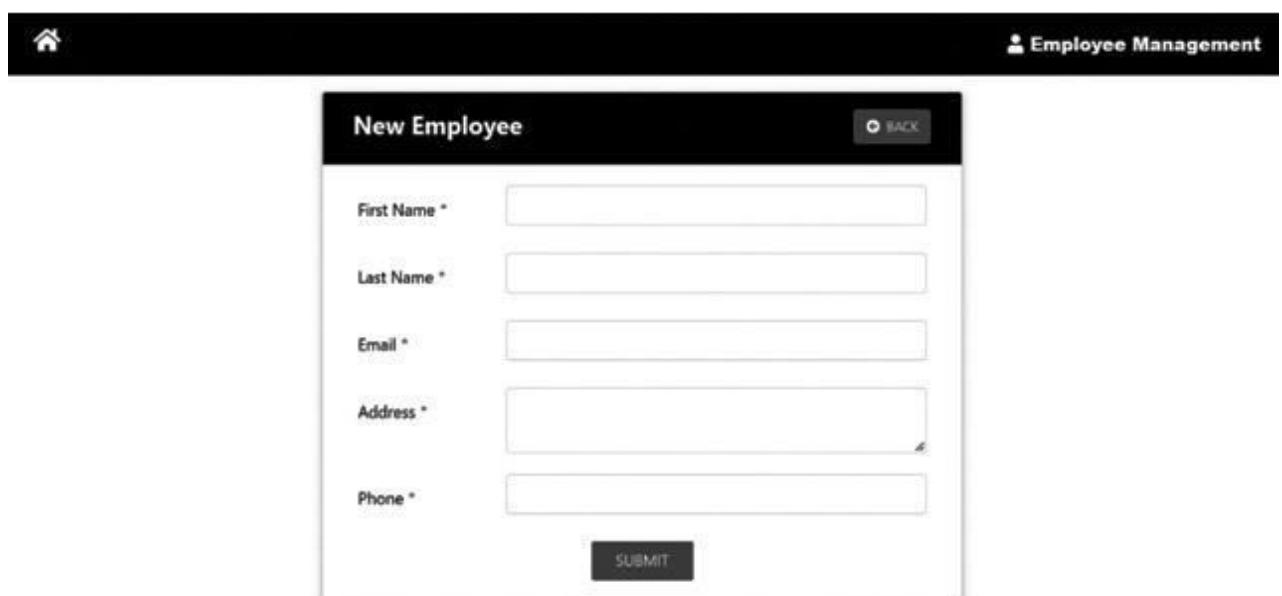
a). Let's see the employee list page screenshot now.



The screenshot shows a web application titled "Employee Management". The main title bar has a house icon and the text "Employee Management". Below it, a sub-header says "Manage Employees". A "ADD" button with a plus sign is located in the top right corner of the header. The main content area is a table with the following columns: First Name, Last Name, Email, Address, Phone, and Actions. There are three rows of data:

First Name	Last Name	Email	Address	Phone	Actions
Test101	Test101	Test101@yopmail.com	Test101 Address	1111111111	
Test102	Test102	Test102@yopmail.com	Test102 Address	2122222222	
Test103	Test103	Test103@yopmail.com	Test103 Address	1111112232	

**b). Add New Employee with Following Output Screen
(add form validation every field mandatory) *.**



The screenshot shows a modal dialog titled "New Employee". The title bar includes a house icon and the text "Employee Management". The main content area is a form with five fields: "First Name *", "Last Name *", "Email *", "Address *", and "Phone *". Each field has a corresponding input text box. At the bottom right of the form is a "SUBMIT" button.



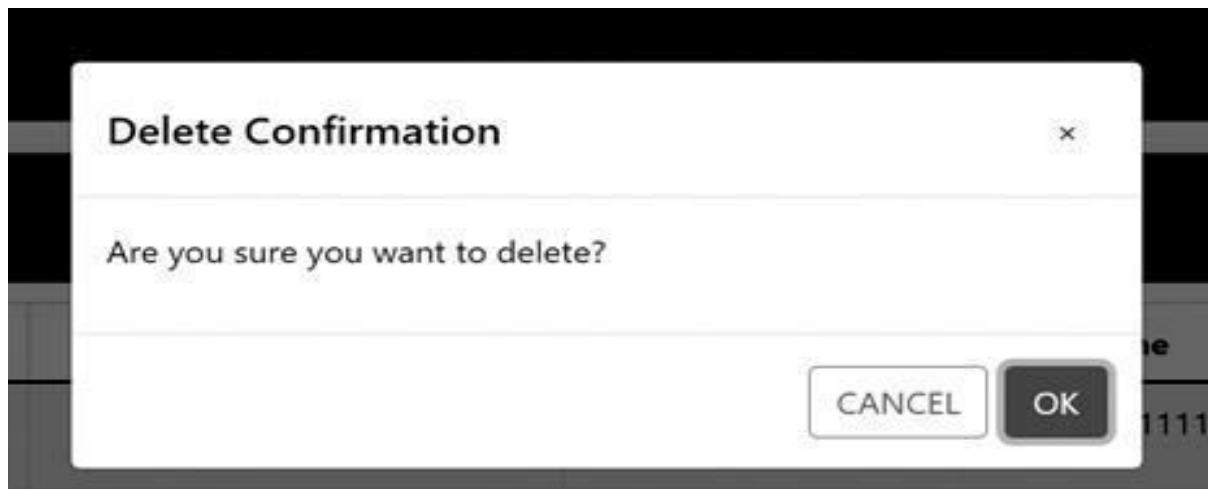
c). Get One Record. And if Required Update here and save back to Data base

The screenshot shows a mobile application interface titled "Employee Management". At the top, there is a navigation bar with a home icon and the text "Employee Management". Below this is a modal window titled "View Employee". Inside the modal, there is a table with the following data:

First Name	Test101
Last Name	Test101
Email	Test101@yopmail.com
Address	Test101 Address
Phone	1111111111

At the bottom of the modal is a large "Update" button.

d). Delete Confirmation Screen Shots

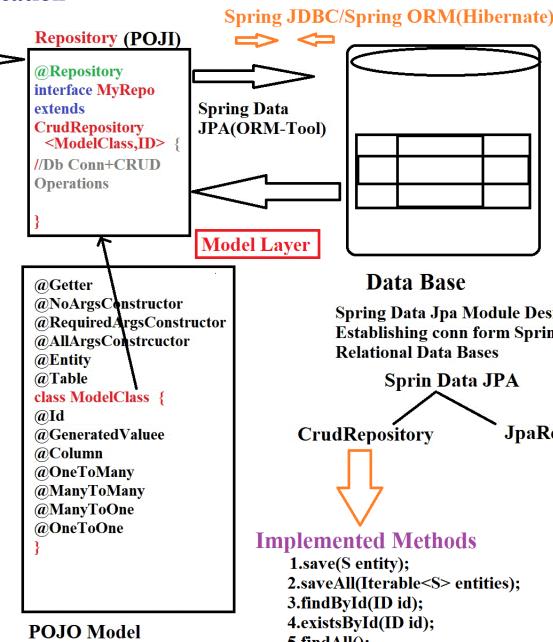
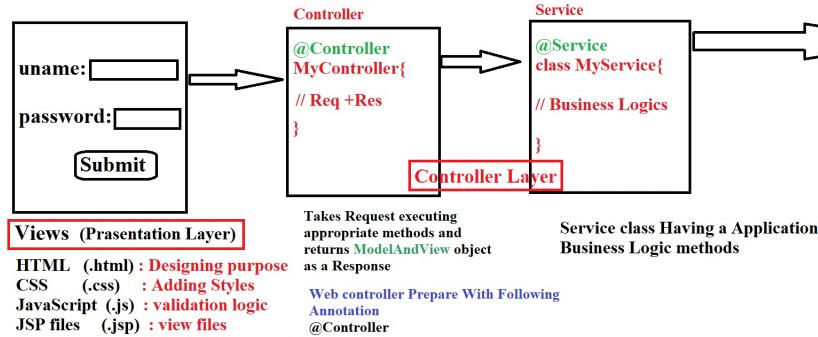


e). Success Messages for CRUD Operations

Three screenshots of a mobile application interface titled "Manage Employees". Each screenshot shows a success message in a dark grey box with a checkmark icon and the text: "Employee was added successfully", "Employee was updated successfully", and "Employee was deleted successfully". The "Manage Employees" title bar and a "ADD" button are visible at the top of each screen.

Spring and Spring boot Complete Application Structure

Spring Web-MVC Module For Developing WebApplication And REST API Application



Spring and Spring Boot All Topics With Diagrammatically For Our Application

Starting Point Of Spring Application

@SpringBootApplication
Application.java {.....}

--> IOC
--> IOC-Life Cycle
--> Dependency Injection
--> Spring and Spring Boot Configuration
--> Runners



Spring Core Modules

Implemented Methods

- 1.save(S entity);
- 2.findAll(Iterable<S> entities);
- 3.findById(ID id);
- 4.existsById(ID id);
- 5.findAll();
- 6.findAllById(Iterable<ID> ids);
- 7.count();
- 8.deleteById(ID id);
- 9.delete(Entity)
- 10.deleteAll(Iterable<? extends T> entities);
- 11.deleteAll()

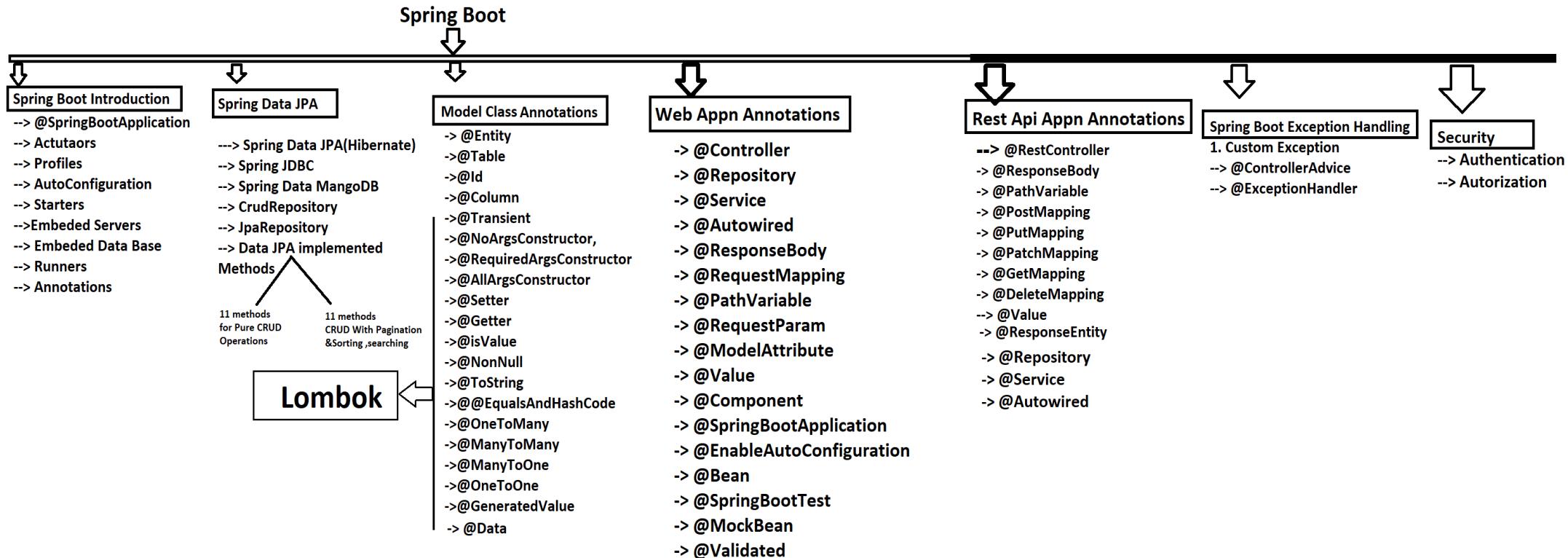
PagingAndSortingRepository

- 1.findAll(Sort sort);
- 2.findAll(Pageable pageable); pagination

Implemented Methods

- 1.findAll(Sort sort);
- 2.findAll(Iterable<T> entities);
- 3.findAllById(Iterable<ID> ids);
- 4.findAll(Iterable<S> entities);
- 5.flush();
- 6.saveAndFlush(S entity);
- 7.deleteInBatch(Iterable<T> entities);
- 8.deleteAllInBatch();
- 9.findOne(ID id);
- 10.findAll(Example<S> example);
- 11.findAll(Example<S> example, Sort sort);

Spring Boot Web Applications And REST API Application Annotations



Create Spring-Boot web Application for IPL-POINT-TABLE with all matches result in tables format

POS	TEAM	P	W	L	NR	NRR	FOR	AGAINST	PTS
1	—  GT	11	8	3	0	0.951	1873/209.0	1757/219.2	16
2	—  CSK	11	6	4	1	0.409	1835/194.3	1799/199.2	13
3	▲  MI	11	6	5	0	-0.255	2001/212.3	2052/212.1	12
4	▼  LSG	11	5	5	1	0.294	1715/196.0	1687/199.3	11
5	▼  RR	11	5	6	0	0.388	2020/219.2	1882/213.2	10
6	▼  KKR	11	5	6	0	-0.079	1992/216.0	1961/210.5	10
7	▼  RCB	11	5	6	0	-0.345	1947/216.2	1992/213.1	10
8	▼  PBKS	11	5	6	0	-0.441	1966/215.3	2026/211.5	10

By Using Below Forms Insert Match Details

IPL-TTABLE-FORM

Pos:

Team Name:

Played Matches

Win Matches:

Loss Matches:

Total Runs Winning Matches:

Total Runs Loss Matches:

Save

Reset

For Project Required Files and Jars with Folder Structure

1.For Above Application How many Controllers are required?

2.For Above Application How many Service classes are required and how?

3.For Above Application How many Repository classes are required?

4.Where we can write databases connection code?

5.Where we can write port number code?

6.Where we can write view Resolver code and in what files?

7.How to makes links from one view to another views?

8.How to maps url to our methods?

9.What are REST API call methods Annotations?

10.What is Autowiring Purpose of @Autowired Annotation?

11.Purpose of @Controller Annotation?

12. purpose of @Service Annotation?

13. purpose of @Repository Annotation?

14. purpose of Entity class and write Entity class Annotation?

15. What is Annotation?

16. Every Annotation is a _____

17. Annotation Start with _____ Symbol.

18. purpose of @SpringBootApplication?

19. purpose of application.properties file?

20. what is DATA JPA in spring boot and Purpose of DATA JPA?

Create RESTfull API using any of the bellow technology

- Java 8 with Spring Boot Or
- Node and ExpressJs

1. Create Product entity and save below product in database (use any database of your choice).
2. You should be able to perform CRUD operations on Product entity to add/update/delete products.
3. getProducts method will calculate the final price of each product based on discount and other charges table .
4. Optionally add Angular/React UI for adding and removing products within the cart.

Product ID	Product Name	Product Type	Product Category	Product Price
1001	Lenovo Yoga	Laptop	Electronics	45000
1002	LG Washing Machine	Washing Machine	Home Appliances	25000
1003	HP Pavilion 360	Laptop	Electronics	60000
1004	US Polo Assn	T-Shirt	Clothing	1200
1005	Neelkamal Been Bag	Sofa Set	Furniture	350

Disconts, Tax and Delivery charges (Per Category)

Product Category	Discount	GST	Delivey Charge (Rs)
Electronics	15%	18%	350
Home Appliances	22%	24%	800
Clothing	40%	12%	0
Furniture	10%	18%	300

get Products JSON output as given below.

```
{  
    "productId": "1001",  
    "name": "Lenovo Yoga",  
    "productType": "Laptop",  
    "category": "Electronics",  
    "basePrice": "45000.00",  
    "discount": 6750.00,  
    "charges": {  
        "gst": "6885.00",  
        "delivery": "350.00"  
    },  
    "finalPrice": "45485.00"  
}
```

Final price is calculated excluding discounts and including GST and Delivery Charges which are different for each product categories.

Important, Please Read Carefully:

- Do Add a README file and explain how your application can be started
- Do check-in the code to github, keep doing frequent check-ins.
- Do add Unit Test cases along with solutions.
- Do use Java 8 feature (when coding in Java) or Arrow functions when using NodeJS
- Do Add Swagger Document for API.
- Do Use any database along with your code, H2 or Mongo in memory would be a good choice.
- Do Not take help from other individuals however, you can use Google, Stack Overflow, YouTube or other online tutorials for making the solution.
- Make sure your code works before sending it for assessment.

Wish you good luck!

Thanks & Regards,

Assignment:

We have created some API's that you have to integrate.

1. Authenticate user using the credentials specified. API uses Bearer authentication to authenticate further API calls. Authentication API will return a bearer token which you will have to pass in subsequent API calls.

Path: https://qa2.sunbasedata.com/sunbase/portal/api/assignment_auth.jsp

Method: POST

Body:

```
{  
    "login_id" : "test@sunbasedata.com",  
    "password" :"Test@123"  
}
```

This will return you a token, that you will have to pass in as Bearer token for further calls.

2. Create a new Customer:

Path: <https://qa2.sunbasedata.com/sunbase/portal/api/assignment.jsp>

Method: POST

Parameters:

cmd : create

Header:

Authorization: Bearer token_recieved_in_authentication_API_call

Body:

```
{  
    "first_name": "Jane",  
    "last_name": "Doe",  
    "street": "Elvnu Street",  
    "address": "H no 2 ",  
    "city": "Delhi",  
    "state": "Delhi",  
    "email": "sam@gmail.com",  
    "phone": "12345678"
```

}

first_name and last_name are mandatory parameters.

Response:

Success: 201, Successfully Created

Failure: 400, First Name or Last Name is missing

3. Get customer list:

Path: <https://qa2.sunbasedata.com/sunbase/portal/api/assignment.jsp>

Method: GET

Parameters:

cmd : get_customer_list

Header:

Authorization: Bearer token_recieved_in_authentication_API_call

Response: 200

```
[{
    "first_name": "Jane",
    "last_name": "Doe",
    "street": "Elvnu Street",
    "address": "H no 2",
    "city": "Delhi",
    "state": "Delhi",
    "email": "sam@gmail.com",
    "phone": "12345678"
}]
```

4. Delete a customer

Path: <https://qa2.sunbasedata.com/sunbase/portal/api/assignment.jsp>

Method: POST

Parameters:

cmd : delete

uuid : uuid of a specific customer

Header:

Authorization: Bearer token_recieved_in_authentication_API_call

Response:

200, Successfully deleted

500, Error Not deleted

400, UUID not found

5. Update a customer

Path: <https://qa2.sunbasedata.com/sunbase/portal/api/assignment.jsp>

Method: POST

Parameters:

cmd : update

uuid : uuid of a specific customer

Header:

Authorization: Bearer token_recieved_in_authentication_API_call

Body:

```
{  
  "first_name": "Jane",  
  "last_name": "Doe",  
  "street": "Elvnu Street",  
  "address": "H no 2 ",  
  "city": "Delhi",  
  "state": "Delhi",  
  "email": "sam@gmail.com",  
  "phone": "12345678"
```

}

Response:

200, Successfully Updated

500, UUID not found

400, Body is Empty

Notes:

if you are passing in wrong value for cmd parameter, you will get 500, Invalid Command

If for any API authentication fails you will get 401, Invalid Authorization

You can test tool like postman to test these API's.

Basic steps include:

- create a basic UI in html. does not have to look pretty. basic Html table , form and buttons would do.

- Basically we need ability to Login, view the list of customer, create a new customer, delete and update an existing customer.

It will consist of 3 screens:

- Login Screen

Login Page

login id

Password

Submit

- Customer List Screen

Customer List							Delete this customer	Edit this customer
First Name	Last Name	Address	City	State	Email	Phone	Action	
Sakshi	Sharma	Hno 1	Delhi	Delhi	sam@gmail.com	10258		
Nikhil	Sharma	Hno 1	Delhi	Delhi	sam@gmail.com	10232		
Roneet	Sharma	Hno 1	Delhi	Delhi	sam@gmail.com	10267		
Jane	Jane	Hno 1	Delhi	Delhi	sam@gmail.com	10244		
Tom	Tom	Delhi	Delhi	Delhi	sam@gmail.com			

- Add a new customer

Customer Details

First name

Last name

Street

Address

City

State

Email

Phone

Submit

Create a new customer
on click of this button

1. What is Security?

2. What is Spring Security?

3. What are Spring Security Key points?

4. What is Authentication in Spring Boot?

5. How to Enable Security in Spring Boot Application?

6. What is Authorization in spring boot Application?

7. What is Knowledge based security? And give examples?

8. What is Possession based Security? And give examples?

9. Define multifactor based security? And give examples?

10. What are Granted Authorities in spring boot? Give some examples?

11. different between Authentication and Authorization?

Authentication	Authorization

12. How Spring Security will work explain with diagram?

13. Explain Spring Security Execution Flow?

14. How to Add Customize Security in spring boot application? And where we can add?

15. What are default Spring Security user name and password?

16. How to Perform Login operation in Post Man Tool with Spring Security?

17. What is 401 errors in spring Security?

18. What is 403 Error in Spring Security?

19. To Prepare Customize Security Java class. Which security class extends?

20. What annotations are using Enabled Customize Security?

21. Purpose of @EnableWebSecurity Annotation?

22. Purpose of @Configuration Annotation?

23. Purpose of .and().formLogin() method: _____

23. Purpose of .and().httpBasic() method: _____

1. Postman Tutorial

Postman is one of the most popular software testing tools which is used for API testing. With the help of this tool, developers can **easily create, test, share, and document APIs.**

Introduction to Postman

- Postman is a standalone software testing API (Application Programming Interface) platform to build, test, design, modify, and document APIs. It is a simple Graphic User Interface for sending and viewing HTTP requests and responses.
- While using Postman, for testing purposes, one doesn't need to write any HTTP client network code. Instead, we build test suites called collections and let Postman interact with the API.
- In this tool, nearly any functionality that any developer may need is embedded. This tool has the ability to make various types of HTTP requests like GET, POST, PUT, PATCH, and convert the API to code for languages like JavaScript and Python.

Terminologies Related to Postman

API

Application Programming Interface (API) is software that acts as an intermediary for two apps to communicate with each other. We use APIs whenever we use an application like Twitter, Facebook, sending text messages, or checking the weather over the phone.

HTTP

HTTP (Hypertext Transfer Protocol) is the collection of rules for the transmission of data on the World Wide Web, like graphic images, text, video, sound, and other multimedia data. The Web users implicitly make use of HTTP as soon as they open their Web browser.

2. What is REST

REST (Representational State Transfer) paradigm and how rest architecture streamlines communication between web components. Making it easier for systems to communicate with each other.

3. What RESTful API and RESTful web services?

API developers can design APIs using several different architectures. APIs that follow the REST architectural style are called REST APIs. Web services that implement REST architecture are called RESTful web services. The term RESTful API generally refers to RESTful web APIs. However, you can use the terms REST API and RESTful API interchangeably.

4. What does the RESTful API server response contain?

REST principles require the server response to contain the following main components:

Response Codes

Responses from the server contain status codes to alert the client to information about the success of the operation. As a developer, you do not need to know every status code (there are many of them), but you should know the most common ones and how they are used:

Status code	Meaning
200 (OK)	This is the standard response for successful HTTP requests.
201 (CREATED)	This is the standard response for an HTTP request that resulted in an item being successfully created.
204 (NO CONTENT)	This is the standard response for successful HTTP requests, where nothing is being returned in the response body.
400 (BAD REQUEST)	The request cannot be processed because of bad request syntax, excessive size, or another client error.
403 (FORBIDDEN)	The client does not have permission to access this resource.
404 (NOT FOUND)	The resource could not be found at this time. It is possible it was deleted, or does not exist yet.
500 (INTERNAL SERVER ERROR)	The generic answer for an unexpected failure if there is no more specific information available.

5. REST API-CRUD Operations

Operation	HTTP Methods	Description
Create	POST	Creating/Posting/Inserting Data
Read	GET	Reading/Getting/Retrieving Data
Update	PUT, PATCH	Updating Data
		Complete Update - PUT Partial Update - PATCH
Delete	DELETE	Deleting Data

@Autowired: Spring provides annotation-based auto-wiring by providing `@Autowired` annotation. It is used to autowire spring bean on setter methods, instance variable, and constructor. When we use `@Autowired` annotation, the spring container auto-wires the bean by matching data-type.

@Configuration: It is a class-level annotation. The class annotated with `@Configuration` used by Spring Containers as a source of bean definitions.

@ComponentScan: It is used when we want to scan a package for beans. It is used with the annotation `@Configuration`. We can also specify the base packages to scan for Spring Components.

@Bean: It is a method-level annotation. It is an alternative of XML `<bean>` tag. It tells the method to produce a bean to be managed by Spring Container.

@Component: It is a class-level annotation. It is used to mark a Java class as a bean. A Java class annotated with `@Component` is found during the classpath. The Spring Framework pick it up and configure it in the application context as a **Spring Bean**.

@Controller: The `@Controller` is a class-level annotation. It is a specialization of `@Component`. It marks a class as a web request handler. It is often used to serve web pages. By default, it returns a string that indicates which route to redirect. It is mostly used with `@RequestMapping` annotation.

@Service: It is also used at class level. It tells the Spring that class contains the **business logic**.

@Repository: It is a class-level annotation. The repository is a **DAOs** (Data Access Object) that access the database directly. The repository does all the operations related to the database

Spring Boot Annotations

- **@EnableAutoConfiguration:** It auto-configures the bean that is present in the classpath and configures it to run the methods. The use of this annotation is reduced in Spring Boot 1.2.0 release because developers provided an alternative of the annotation, i.e. `@SpringBootApplication`.
- **@SpringBootApplication:** It is a combination of three annotations `@EnableAutoConfiguration`, `@ComponentScan`, and `@Configuration`.

Spring MVC and REST Annotations

- **@RequestMapping:** It is used to map the **web requests**. It has many optional elements like **consumes**, **header**, **method**, **name**, **params**, **path**, **produces**, and **value**. We use it with the class as well as the method.
- **@GetMapping:** It maps the **HTTP GET** requests on the specific handler method. It is used to create a web service endpoint that **fetches** It is used instead of using: **@RequestMapping(method = RequestMethod.GET)**
- **@PostMapping:** It maps the **HTTP POST** requests on the specific handler method. It is used to create a web service endpoint that **creates** It is used instead of using: **@RequestMapping(method = RequestMethod.POST)**
- **@PutMapping:** It maps the **HTTP PUT** requests on the specific handler method. It is used to create a web service endpoint that **creates** or **updates** It is used instead of using: **@RequestMapping(method = RequestMethod.PUT)**
- **@DeleteMapping:** It maps the **HTTP DELETE** requests on the specific handler method. It is used to create a web service endpoint that **deletes** a resource. It is used instead of using: **@RequestMapping(method = RequestMethod.DELETE)**
- **@PatchMapping:** It maps the **HTTP PATCH** requests on the specific handler method. It is used instead of using: **@RequestMapping(method = RequestMethod.PATCH)**
- **@RequestBody:** It is used to **bind** HTTP request with an object in a method parameter. Internally it uses **HTTP MessageConverters** to convert the body of the request. When we annotate a method parameter with **@RequestBody**, the Spring framework binds the incoming HTTP request body to that parameter.
- **@ResponseBody:** It binds the method return value to the response body. It tells the Spring Boot Framework to serialize a return an object into JSON and XML format.

- **@PathVariable:** It is used to extract the values from the URI. It is most suitable for the RESTful web service, where the URL contains a path variable. We can define multiple @PathVariable in a method.
- **@RequestParam:** It is used to extract the query parameters form the URL. It is also known as a **query parameter**. It is most suitable for web applications. It can specify default values if the query parameter is not present in the URL.
- **@RequestHeader:** It is used to get the details about the HTTP request headers. We use this annotation as a **method parameter**. The optional elements of the annotation are **name**, **required**, **value**, **defaultValue**. For each detail in the header, we should specify separate annotations. We can use it multiple time in a method
- **@RestController:** It can be considered as a combination of **@Controller** and **@ResponseBody** annotations. The **@RestController** annotation is itself annotated with the **@ResponseBody** annotation. It eliminates the need for annotating each method with **@ResponseBody**.
- **@RequestAttribute:** It binds a method parameter to request attribute. It provides convenient access to the request attributes from a controller method. With the help of **@RequestAttribute** annotation, we can access objects that are populated on the server-side.

Spring Boot Starters or Dependencies

1. Spring Boot Starter Web

There are two important features of spring-boot-starter-web:

- It is compatible for web development
- Auto configuration

If we want to develop a web application, we need to add the following dependency in pom.xml file:

1. **<dependency>**
2. **<groupId>org.springframework.boot</groupId>**
3. **<artifactId>spring-boot-starter-web</artifactId>**
4. **<version>2.2.2.RELEASE</version>**
5. **</dependency>**

Starter of Spring web uses Spring MVC, REST and Tomcat as a default embedded server. The single spring-boot-starter-web dependency transitively pulls in all dependencies related to web development. It also reduces the build dependency count. The spring-boot-starter-web transitively depends on the following:

- org.springframework.boot:spring-boot-starter
- org.springframework.boot:spring-boot-starter-tomcat
- org.springframework.boot:spring-boot-starter-validation
- com.fasterxml.jackson.core:jackson-databind
- org.springframework:spring-web
- org.springframework:spring-webmvc

2. Spring Boot DevTools

DevTools stands for **Developer Tool**. The aim of the module is to try and improve the development time while working with the Spring Boot application. Spring Boot DevTools pick up the changes and restart the application.

We can implement the DevTools in our project by adding the following dependency in the pom.xml file.

```
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<scope>runtime</scope>
</dependency>
```

3. What is the H2 DATABASE?

H2 is an **embedded**, **open-source**, and **in-memory** database. It is a relational database management system written in [Java](#). It is a **client/server** application. It is generally used in **unit testing**. It stores data in memory, not persist the data on disk.

Advantages

- Zero configuration
- It is easy to use.
- It is lightweight and fast.

- It provides simple Configuration to switch between a real database and in-memory database.
- It supports standard SQL and JDBC API.
- It provides a web console to maintain in the database.

Persistence vs. In-memory Database

The persistent database persists the data in physical memory. The data will be available even if the database server is bounced. Some popular persistence databases are **Oracle**, **MySQL**, **Postgres**, etc.

In the case of the **in-memory database**, data store in the **system memory**. It lost the data when the program is closed. It is helpful for **POCs** (Proof of Concepts), not for a production application. The widely used in-memory database is **H2**.

Spring Boot – Difference Between CrudRepository and JpaRepository

CrudRepository	JpaRepository
It is a base interface and extends Repository Interface.	It extends PagingAndSortingRepository that extends CrudRepository.
It contains methods for CRUD operations. For example save(), saveAll(), findById(), findAll(), etc.	It contains the full API of CrudRepository and PagingAndSortingRepository. For example, it contains flush(), saveAndFlush(), saveAllAndFlush(), deleteInBatch(), etc along with the methods that are available in CrudRepository.
It doesn't provide methods for implementing pagination and sorting	It provides all the methods for which are useful for implementing pagination.
It works as a marker interface.	It extends both CrudRepository and PagingAndSortingRepository.
To perform CRUD operations, define repository extending CrudRepository.	To perform CRUD as well as batch operations, define repository extends JpaRepository.
Syntax: public interface CrudRepository<T, ID> extends Repository<T, ID>	Syntax: public interface JpaRepository<T, ID> extends PagingAndSortingRepository<T, ID>, QueryByExampleExecutor<T>

22. Profiles

Spring Profiles provide a way to segregate parts of your application configuration and make it only available in certain environments. Any @Component or @Configuration can be marked with @Profile to limit when it is loaded:

```
@Configuration  
{@Profile("production")  
public class ProductionConfiguration {  
    // ..  
}}
```

In the **normal Spring way**, you can use a **spring.profiles.active Environment property to specify which profiles are active**. You can specify the property in any of the usual ways, for example you could include it in your application.properties:

spring.profiles.active=dev,hsqldb

1.2 What are Profiles in Spring boot?

Profiles in simple term can be termed as the different environments which every application comprises of. For example – DEV, QA, or PROD. Each environment has its own specific requirement i.e. database connection, pipeline, and deployment configuration. In spring we maintain profiles with the help of properties and file files.

application. properties

```
## Spring boot application name  
spring.application.name=Profiles  
## Profiles are activated using this property  
## Tells Spring which profiles to use  
## E.g. - Here we set the profile as "qa"  
#spring.profiles.active=dev  
spring.profiles.active=qa  
#spring.profiles.active=prod
```

Spring Boot Test

1. Spring vs. Spring Boot?
2. Features of Spring Boot?
3. What is IOC? IOC Life cycle?
4. Dependency Injection? Types?
5. Setter Injection
6. Constructor Injection
8. IOC Container names?
9. Explain Spring modules?
10. Purpose Pom.xml?
11. Explain about application. Properties file?
12. How to set port no for application?
13. In which file we have to register views folder in spring Boot?
14. In which file we have to register database Connection strings in spring Boot write code also?
15. What is annotation?
16. Why we use annotations?
17. What are the different places we can declare annotations?
19. Explain @SpringBootApplication annotation?
20. @SpringBootApplication is the combination of ----- Annotations?

21. Starting point of Spring Boot application?
22. Explain about @Configuration?
23. Explain about @Component can?
24. Explain about @Component?
25. Explain about @Controller?
26. Explain about @Service?
27. Explain about @Repository?
28. Explain about Request Mapping Annotations?
 - a. @RequestMapping
 - b. @GetMapping

Spring Boot Test

c. `@PostMapping`

d. `@PutMapping`

e. `@DeleteMapping`

29. `@RequestBody`

30. `@ResponseBody`

31. `@PathVariable`

31. `@Requestparam`

32. `@RestController`

34. purpose of `<spring boot starter-web>`

36. Explain spring boot Dev tools?

37. What are H2 databases?

39. What is the embedded server name in spring boot Application?

41. How to establish connectivity between layers in spring boot application?

43. What Logic we write in Presentation Layer?

44. What Logic we write in Controller Layer?

45. What Logic we write in Service Layer?

46. What Code we write in Repository Layer?

47. Controller Consists of ----- Methods?

49. How to create Repository?

50. What is POJO and how to create POJO?

53. What is purpose of Model class?

55. How to create Entity class?

56. What is the purpose of `@Qualifier` annotation?

58. What is the starting point of spring boot application?

59. What is Singleton class and it's Rules?

60. What is default scope of bean class?

61. What is purpose of prototype scope?

63. What is the purpose of View Resolver?

64. What is the purpose of Handler Mapper?

Spring Boot Test

- 60. What is default scope of bean class?**
- 61. What is purpose of prototype scope?**
- 62. How many scopes are there in spring boot?**
- 63. What is the purpose of View Resolver?**
- 64. What is the purpose of Handler Mapper?**
- 65. How to create Rest Controller?**
- 66. What is the purpose of @Query annotation?**
- 69. What is Profile in Spring Boot?**
- 70. What is Actuator? Purpose of Actuator.**
- 71. What is starter's dependency in spring boot?**
- 72. What is spring auto-configuration?**
- 73. What does @EnableAutoConfiguration annotation do?**
- 74. What are the different ways to create spring boot application?**
- 75. What is Spring Boot Starter Parent?**
- 76. What is loosely coupling?**
- 77. What is swagger in spring boot?**
- 78. What is the purpose of postman?**
- 79. Explain about MVC 3-tier architecture in spring boot application?**
- 80. What is spring data JPA?**
- 81. What are interface names to extend in repository layer?**
- 82. How many ways we can handle incoming http request method?**
- 83. Explain about 5 http request method?**
- 84. What is marker interface?**
- 85. Explain about marker interface? Give an example marker interface which is used in Spring Data JPA?**
- 86. Name few annotation used in Entity class?**
- 87. What is URL?**
- 89. What are Http error explain?**

200,201,400,404,403,500

Spring Boot Test

90. How to create user defined exception in Spring Rest and write the snippet code with annotation?

- 91. What is bean life cycle?
 - 92. What is AOP?
 - 93. What is the purpose of AOP?
 - 94. What is the cross site origin?
 - 95. What is Aspect in AOP?
 - 96. What is Cross-cutting concern?
 - 97. What is pointcut, joinpoint,advice?
 - 98. What is purpose of kafka?
 - 100. What is Model, ModelMap and ModelAndView?
 - 101. What is Eager loading and Lazy loading?
 - 103. What is the purpose of @Controller Advice?
- ## **Microservices**
- 1. What are Micro services?
 - 2. What is the purpose of service registery?
 - 3. How to implement service registery?
 - 4. What is the purpose of load balancing in microservices?
 - 5. How to implement load balancing ?
 - 6. What is difference between Monolithic application and Micro services?
 - 7. What is circuit breaker design pattern?
 - 8. Write snippet code to implement circuit breaker pattern?
 - 9. What is purpose of API Gateway?
 - 10. Write a snippet code to implement API Gate?
 - 11. What is Zipkin Server?
 - 12. How to Implement Zipkin server?
 - 13. What is cloud config server?
 - 15. Expalin about Microservices acrchitecture?

Tools

- What is unit testing?
- What is purpose of Unit testing?
- What is Mockito?
- What is power Mockito?
- What is purpose of power Mockito?
- What is Github?
- What is purpose of Github?
- What is code coverage tool?
- What is purpose of code coverage tool?
- What is docker?
- What is purpose of docker?
- What is Jenkins?
- What is purpose of Zenkins?
- What is maven?
- What is built tool?