

**FINAL
PROJECT**

CDAC PROJECT LIST

REACT SPRINGBOOT MYSQL

Enroll

Now!

WWW.CODEWITHARRAYS.IN

Explore More

Subscription : Premium CDAC NOTES & MATERIAL



Contact to Join
Premium Group



Click to Join
Telegram Group

<CODEWITHARRAY'S/>

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

codewitharrays.in freelance project available to buy contact on 8007592194		
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySQL
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySQL
3	Tour and Travel management System	React+Springboot+MySQL
4	Election commition of India (online Voting System)	React+Springboot+MySQL
5	HomeRental Booking System	React+Springboot+MySQL
6	Event Management System	React+Springboot+MySQL
7	Hotel Management System	React+Springboot+MySQL
8	Agriculture web Project	React+Springboot+MySQL
9	AirLine Reservation System / Flight booking System	React+Springboot+MySQL
10	E-commerce web Project	React+Springboot+MySQL
11	Hospital Management System	React+Springboot+MySQL
12	E-RTO Driving licence portal	React+Springboot+MySQL
13	Transpotation Services portal	React+Springboot+MySQL
14	Courier Services Portal / Courier Management System	React+Springboot+MySQL
15	Online Food Delivery Portal	React+Springboot+MySQL
16	Municipal Corporation Management	React+Springboot+MySQL
17	Gym Management System	React+Springboot+MySQL
18	Bike/Car ental System Portal	React+Springboot+MySQL
19	CharityDonation web project	React+Springboot+MySQL
20	Movie Booking System	React+Springboot+MySQL

freelance_Project available to buy contact on 8007592194

21	Job Portal web project	React+Springboot+MySQL
22	LIC Insurance Portal	React+Springboot+MySQL
23	Employee Management System	React+Springboot+MySQL
24	Payroll Management System	React+Springboot+MySQL
25	RealEstate Property Project	React+Springboot+MySQL
26	Marriage Hall Booking Project	React+Springboot+MySQL
27	Online Student Management portal	React+Springboot+MySQL
28	Resturant management System	React+Springboot+MySQL
29	Solar Management Project	React+Springboot+MySQL
30	OneStepService LinkLabourContractor	React+Springboot+MySQL

31	Vehical Service Center Portal	React+Springboot+MySQL
32	E-wallet Banking Project	React+Springboot+MySQL
33	Blogg Application Project	React+Springboot+MySQL
34	Car Parking booking Project	React+Springboot+MySQL
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySQL
36	Society management Portal	React+Springboot+MySQL
37	E-College Portal	React+Springboot+MySQL
38	FoodWaste Management Donate System	React+Springboot+MySQL
39	Sports Ground Booking	React+Springboot+MySQL
40	BloodBank mangement System	React+Springboot+MySQL
41	Bus Tickit Booking Project	React+Springboot+MySQL
42	Fruite Delivery Project	React+Springboot+MySQL
43	Woodworks Bed Shop	React+Springboot+MySQL
44	Online Dairy Product sell Project	React+Springboot+MySQL
45	Online E-Pharma medicine sell Project	React+Springboot+MySQL
46	FarmerMarketplace Web Project	React+Springboot+MySQL
47	Online Cloth Store Project	React+Springboot+MySQL
48	Train Ticket Booking Project	React+Springboot+MySQL
49	Quizz Application Project	JSP+Springboot+MySQL
50	Hotel Room Booking Project	React+Springboot+MySQL
51	Online Crime Reporting Portal Project	React+Springboot+MySQL
52	Online Child Adoption Portal Project	React+Springboot+MySQL
53	online Pizza Delivery System Project	React+Springboot+MySQL
54	Online Social Complaint Portal Project	React+Springboot+MySQL
55	Electric Vehical management system Project	React+Springboot+MySQL
56	Online mess / Tiffin management System Project	React+Springboot+MySQL
57	Online Examination Portal Project	React+Springboot+MySQL
58	Lawyer / Advocate Appointment Booking System	React+Springboot+MySQL
59	Café Management System	React+Springboot+MySQL
60	Agriculture Product Rent system Portal	React+Springboot+MySQL

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/vXz0kZQi5to?si=IiOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7uiQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/lynLouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyl4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHxE?si=gDiAhKBowi2gNUKz

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrk7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizlyJlh
15	Employee management system project	https://youtu.be/ID1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWA OzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vn0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

Sr.No	Project Name	YouTube Link
21	Online Bus ticket booking system Project	https://youtu.be/FJ0RUzfMdv8?si=auHjmNgHMrpaNzvY
22	Online Mess / Tiffin Booking System Project	https://youtu.be/NTVmHFDowyl?si=yrvClbE6fdJ0B7dQ
23		
24		
25		

TAP ON THE ICONS TO JOIN!



1. Describe the problem statement of your Project briefly.?

- Nowadays drones are in very trend people wanted to catch their special moments or events via drones also by research we found that there is no website made in india who is giving their drone on rent on scheduled time also with services by thinking this we implemented our project and future scope of our project is
- We can provide drone training and guidance for user from drone experts.
- We can arrange online drone events.
- At the time of emergencies, we can make sure of drone availability in bulk quantity.
- We can provide service centres for customers.

2. Explain the flow of your project?

Admin Panel: First Admin is hard coded added using SQL query. Admin will login to our web application using role based login.

1. Admin can add/delete/update another admin's, dealer's, agent's information.
2. Admin can permit or reject the customers booking request and service request
3. Admin can check the status of agent
4. Admin can add/delete/update new stock
5. Admin will keep the records of everything like registered user and all

Customer Panel : At first customer will enter in our web application then home page of our web application will get loaded which will show him available drones if he wanted buy a drone we will find two options on home page register and login we have provided role base login there. If customer is already registered then he will go for login and if not he has to register first.

After login customer can buy drone or to take drone on rent or want service for his already borrowed drones or can go for orders where he can see his orders If customer wants buy a drone he can view products, its specification and category after deciding which drone he wants to buy he will book the drone or will add to cart the drone. In add to cart he can remove from cart, or can book drone also.

After booking he has to update his personal details like address mobile no etc. After that this request of booking will go to admin panel where admin can see availability of particular order if it is available he will allow the customer booking request. After that customer will select the payment mode if he selects online payment mode then after payment one invoice will generate in which payment details and on message "payment done successfully now wait for delivery" will get shown after delivery there is feedback form also available for customer where he can give his feedback and if he selects cash on delivery option again one invoice will generate in which booking details and on message "your booking is done wait for delivery" will get shown.

After delivery he has to do payment and after payment he will get invoice of payment details and on message "payment done successfully and please give your feedback".

If customer wants to take drone on rent he will go in rent section where rental drone home page will get loaded where he can see rental drone products if customers decides to take particular drone on rent he will select take drone on rent then he has to update his personal details like address mobile no etc. Also he has

to mention particular scheduled time on he want drone. After filling this all detail he has to do payment online only. After payment one invoice will get generate in which payment details and one message "payment done successfully wait for scheduled time." will get shown. At schedule time our agent will go to customer address with drone then he will perform his task. After performing task he will update to admin that task is done. Then customer can see feedback form.

If customer wants to return a drone then he will go for orders where he can see his recent orders. From there he can return his recently ordered drone if he selects return drone one query form will generate where customer has to select reason for return order. After filling it properly pick up of drone with verification as per selected reason will get done and if return order get successful one invoice will get generate in which drone details and one message "your drone amount N/A has been transferred to respected bank".

If customer wants service for his drone then he will click on service then there is repairing service is available for only those who are our existing customer. Then customer will enter his drone's product id and will wait for confirmation after that product Id will get matched in database, if match not found admin will reject for service and if match get found admit will fetch warranty details of this product and if the product is in warranty then only free service will get available for customer if not then customer have to pay for service. After getting matched product key admin will permit the customer for service. After that one invoice will get generate on customer side in which product warranty details with book your free service for in warranty product or do payment for your service for not in warranty product one of these two buttons will get displayed on customers side. If customer selects for free service. Service will get booked and one invoice will get generated in which service details and product details with one message "Service booked now wait for service" will get shown. and if he selects for do payment again one invoice will get generate in which payment details and on message "payment done successfully now wait for service" will get shown. After completion of service then customer can see feedback form.

Agent panel: Agent will login and will check for scheduled task if there is task he will fetch customer details after that we will reached the destination at scheduled time and will perform his task throughout this he will keep updating every status to admin

Dealer Panel: Dealer will login and then he can see two options where he can update his new available stock and order request. If he has ne stock then he will update it so that admin can see it and make available it for customer

3.What you have done in your project?

=>my project was divided into modules, and I have worked on owner module. whenever owner will register, all validation will check at frontend, like email is in valid format or not mobile number is of 10 digit or not, once all the details are filled he click submit button and his object get created, we have connected front end to backend through axios. in my owner controller I have written save method . In save method i have written I else statement in If statement I am checking if email is already register or not, if it is same it will return "email alredy register", if not then it flow will go to else statement and in that saveOwner method will get called which i have written in service class It wil create new Owner and all the details will save to data base throgh Owner entity in this entity class i have written all the respective column in which data will store in database.

now when owner try to login his data will come to backend and @RequestBody annotation will dsitribute the data in Login dto. now what ever data we get through login has save to loginDto. now validate method will get called which I have written in OwnerService , this validate method will take LoginDto (owner login data) and I am using JPA method findById if that particular over is present in database and I have written one If statement in that I am checking if owner is != null && owner.getPwd equals means the database password is equals entered password && owner isActive , if all are are true it will return owner to the controller, and through controller flow will return back to front end and it will show popup msg Owner is loggin. if any issue occurs it will throw msg invalid details.

4.OOps concepts used in your project?

Overloading kr rkha hai in admin and user login methods
Inheritance used in DTO's
Encapsulations in entities!
Abstraction in Repos!



5. What are the limitations of your project?

Talking about limitations as in our project we have not implemented synchronization part as a result if one customer is trying to book a particular drone and another customer is also trying to book same drone then it is not possible there will be some ambiguity error will come . These is the limitation of our project sir.

6. How will you improve the performance of your project? (memory related and response time)?

Using load balancing Using network cache Http caching Reverse proxy server catching Using database cache Optimizing the application Storing less data in session Avoiding running out of memory Using native SQL instead of running queries in a loop

7. Validation in project?

We have implemented validation in our project at front end using Java script and for back end we added one dependency called spring boot starter validation and after that in entity we added annotation like @not null @email @not empty @size etc.

freelance Project for buy contact us: - Ashok Pate (8007592194)/ (9284926333)

8. Where we add Configurations in spring project and list few names?

We added configuration in application.properties. To change the port number we used server.port , configured database using driver name , to change the user name we used spring.datasource.username , to change the password we used spring.datasource.password , to send mail of login info to the user we used SMTP server.

9. Mention the Technology used in Project. ?

For front end we used react js and for back end we used hibernate, spring boot and JPA and for database we used MySql .

10. How you all done the project module distribution?

My role in the project was back end developer . Firstly we implemented all the entities of our project together after that we segregated our task one will make one controller , one repository an one service like this.

11. As a leader which are the project leader qualities are?

I believe that by sharing knowledge , you learn more. Problem solving , helpful attitude, team management like delegate tasks to the appropriate team member, Develop team schedules and assist in the successful on boarding and training of team members....These qualities should be in project leader.

12. As a Team members what should be the role towards Project leader?

Executing all tasks assigned by the team leader diligently, on schedule, and to the highest standard. Working with team members to achieve daily, weekly, and monthly targets. Participating in meetings and voicing concerns as well as suggestions for improvement.

13. Any issue or discrepancy in the project where you disagree?

Yes sir , We were found disagreement while selecting project topic but then we solved it by holding grudges and giving chance to everyone for speak also and listening problem deeply and carefully and then finally we came to the solution

14. Have you used any Software Life cycle Model in your project?

Yes, we have used agile model in our project. We used four sprints. In first sprint we made ER-diagram, tables and analysed that what will be the flow of our project. In second sprint we made back end. In third sprint we made front end and in last sprint we did connectivity and implemented the part which we left to implement for some time

15. Is it possible to convert your project from Java to .NET or vice versa?

Yes sir , It is possible using tool like Java Language Conversion Assistant (JLCA) we can convert java project into dot net and for Stryon is providing iNet product so that no need to rewrite the code.

16. If you make use of Angular technology in your UI designing, what possible changes are required to be done?

1) React allows developers full freedom to define the structure of their projects. They can combine the templates, styles and business logic into a single Java script file, separate each into different files, or set an entirely new structure. In other hand Angular has a defined code structure, as components are broken into separate templates, styles, and JavaScript files. The template uses HTML, the styles use CSS/SCSS/SASS, and the JavaScript file uses Typescript. This structure allows backend developers to generate users interfaces with code that is familiar to them. we have to change this after that

2) Our second step is to translate React JSX syntax to Angular HTML syntax . React simply returns JSX without creating any extra wrapper elements. The Angular HTML syntax is not valid JSX and will throw an error. Therefore, we will need to rewrite this component to work with angular

17. Have you used any OOPs concepts in your Project?

Yes sir , our entity class which we made in our project describes encapsulation , interfaces which we made describes abstraction and whole project describes modularity and also in our project we are extending some super classes that shows polymorphism

18. Have you used any APIs in your project?

Mail sender api is used to send email Axios is used for connectivity of back end and front end Restful API like Do, Get, Post, Put to retrieve data

19. Have you used any Payment Gateway in your Project?

No sir, we have not used any payment gateway because to use this we have to borrow it so In our project when customers clicks on pay simply it shows one invoice .

20. Any errors that you face during project?

Spring boot allows adding Dependencies at the start but when we use to need any of the dependencies while making project then we have to add it manually , also mapping related errors and stack overflow error we faced .

21. What are the plus point of your team?

Overcome Obstacles :- When a team faces a challenge, they can utilize their varied learnings to come up with multiple solutions to tackle the problem. Teammates also help each other through their difficulties thus are better able to handle any hiccups along the way and could even proactively warn each other of foreseeable risks.

Trust :- Teamwork creates a strong bond and a team that enjoys working together. If you are able to trust a teammates, it provides a feeling of safety that enables teammates to open up and encourage each other. Trust in teammates also assists in open communication which could indirectly lead to increased trust in the organization and management too. These are the plus points of our team

22. What is the minus point of your team?

Minus point of our team is that we were taking too much time on discussion on particular concept

23. What is hibernet why u have used hibernet in project? and why not jdbc?

Hibernate can perform automatic object mapping. It maps the object model's data to the schema of the database itself with the help of annotations.

In JDBC, one needs to write code to map the object model's data representation to the schema of the relational model.

Whereas Hibernate is database independent and the same code can work for many databases with minor changes.

JDBC is database-dependent, meaning that developers are required to write different codes for different databases.

Hibernate provides good support for lazy loading. JDBC does not support lazy loading.

Hibernate manages exceptions itself by marking them as unchecked.

JDBC code needs to be written in a try-catch block as it throws checked exceptions.

24. Did u know JPA what is it and why u haven't used it?

Spring Boot JPA is a Java specification for managing relational data in Java applications. It allows us to access and persist data between Java object/ class and relational database. JPA follows Object-Relation Mapping (ORM). It is a set of interfaces. It also provides a runtime EntityManager API for processing queries and transactions on the objects against the database. It uses a platform-independent object oriented query language JPQL (Java Persistent Query Language).

25. Normalization is applicable to your project? (Yes/no). If so up to what level.

Yes sir , we have applied normalization till 1NF and 2NF

1NF – Each table cell should contain a single value. Each record needs to be unique.

2NF – Eliminate partial functional dependency

3NF – A relation will be in 3NF if it is in 2NF and not contain any transitive partial dependency.

3NF is used to reduce the data duplication. It is also used to achieve the data integrity.

If there is no transitive dependency for non-prime attributes, then the relation must be in third normal form.

26. Describe database connection of your project?

Driver class: The driver class for the mysql database is **com.mysql.jdbc.Driver**.

Connection URL: The connection URL for the mysql database is **jdbc:mysql://localhost:3306/elearningdb** where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and **elearningdb** is the database name. We may use any database, in such case, we need to replace the **elearningdb** with our database name.

Username: The default username for the mysql database is root.

Password: It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password.

27. Differentiate between .NET and JAVA Project database connections?

Driver class: The driver class for the mysql database is com.mysql.jdbc.Driver.

Connection URL: The connection URL for the mysql database is jdbc:mysql://localhost:3306/sonoo where jdbc is the API, mysql is the database, localhost is the server name on which mysql is running, we may also use IP address, 3306 is the port number and sonoo is the database name. We may use any database, in such case, we need to replace the sonoo with our database name.

Username: The default username for the mysql database is root.

Password: It is the password given by the user at the time of installing the mysql database. In this example, we are going to use root as the password

That's how we do database connection in java

- Create Visual Basic .NET Windows application.
- Create ADO.NET objects.
- Use the SqlConnection object to open SQL Server connection.
- Use the SqlDataReader object to retrieve data from SQL Server.
- View database in Server Explorer.
- Use Server Explorer to open SQL Server connection.

That's how we do database connection in dotnet

28. Is there any feature you have planned but not able to complete it?

Yes sir, we have not implemented synchronization part.

29. Annotation of hibernate ?

@Entity Annotation, @Table Annotation @Id and @GeneratedValue @Column Annotation

30. Annotation of spring?

@Controller @RequestMapping @PathVariable @RequestParam @ModelAttribute @RequestBody
@ResponseBody @RequestHeader and @ResponseHeader.

31. Annotation of http ?

@GET maps to the HTTP GET method. @HEAD maps to HTTP HEAD method. @POST maps to HTTP POST method. @PUT maps to HTTP PUT method. @PATCH maps to HTTP PATCH method. @DELETE maps to HTTP DELETE method.

32. Annotations used in project ?

@Entity @Table @Id and @GeneratedValue @Column @GET @POST @PUT @Controller @crossorigin
@RestController @RequestMapping @RequestParam @RequestBody and @ResponseBody.

33. What is serialization and deserialization? have you used in your project ?

Serialization is the process of converting an object into bytes, so that it can be transmitted over the network, or stored in a flat file and can be recreated later. Serialized object is an object represented as sequence of bytes that includes objects data, object type, and the types of data stored in the object. Yes sir we have used response body that is nothing but the serialization. The `@ResponseBody` annotation tells a controller that the object returned is automatically serialized into JSON and passed back into the `HttpServletResponse` object. Deserialization is the reverse process where the byte stream is used to recreate the actual Java object in memory.

Yes sir we have used request body that's nothing but the deserialization. The `@RequestBody` annotation maps the HTTP Request body to a transfer or domain object, enabling automatic deserialization of the inbound `HttpServletRequest` body onto a Java object.

34. What is session ?

A session is some data that is stored on the server. The server then provides an ID to the client, which the client can use to make requests back to the server.

35. How you handle the session in spring boot ?

Create Spring Boot project from Spring Initializer.

Add Spring Session jdbc dependency in `pom.xml`.

Add spring jdbc properties in `application.properties`.

Create rest end points to save, destroy/invalidate session.

36. How you did session management in react?

We used local storage to maintain the session.

When user logged in customer id will get stored into local storage using

`LocalStorage.setItem("customerId", Id);` method Key value

And that Id will get retrieve from local storage using `localStorage.getItem("customerId");` method Key

To clear the session we used local Storage. `removeItem("customerId");` If customer logged out.

37. What is synchronization ? have you used in your project ?

Synchronization is a technique used to ensure that when multiple threads are sharing same resources, there shouldn't be a "Race Condition" scenario. In other words, synchronization ensures that the lock on an object is acquired by only one thread at a time. No sir we have not used it in our project.

38. What is SRS table ?

Software Requirements Specifications, also known as SRS, is the term used to describe an in-depth description of a software product to be developed. It's considered one of the initial stages of the software development lifecycle (SDLC).

39. What is restful API ?

A Restful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating and deleting of operations concerning resources.

40. What is rest controller ?

Spring RestController annotation is used to create RESTful web services using Spring MVC. Spring RestController takes care of mapping request data to the defined request handler method. Once response body is generated from the handler method, it converts it to JSON or XML response

41. What is REST ?

REST is an acronym for Representational State Transfer and an architectural style for distributed hypermedia systems. REST is web standards based architecture and uses HTTP Protocol for data communication. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods.

42. What is REDUX?

Redux is a library used for front end development. It is a state container for JavaScript applications which should be used for the applications state management. You can test and run an application developed with Redux in different environments

43. What is web service?

A web service is a collection of open protocols and standards used for exchanging data between applications and applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer

44. How to create Restful web service ?

- Step 1: Initializing a Spring Boot Project
- Step 2: Connecting Spring Boot to the Database
- Step 3: Creating a User Model
- Step 4: Creating Repository Classes
- Step 5: Creating a Controller applying Rest controller, request mapping, get, put, post, delete annotations in controller
- Step 6: Compile, Build and Run
- Step 7: Testing the Spring Boot REST APIs This is how we can create Restful web Service

45. Spring MVC flow?

In a typical web application, the controller-service-model-repository architecture is a common design pattern. Here is an explanation of how it works:

Controller: The controller is responsible for receiving and processing incoming HTTP requests from the client. It is usually the first point of contact between the client and the application. The controller handles the incoming request, performs any necessary validation, and decides which service method to call.

Service: The service layer is responsible for implementing the business logic of the application. It receives requests from the controller, performs any necessary data processing, and communicates with the repository layer to retrieve or store data.

Model: The model layer contains the data model used by the application. It represents the domain entities and relationships between them. The model can be simple or complex depending on the requirements of the application.

Repository: The repository layer is responsible for communicating with the database. It provides an abstraction layer between the service and the database, allowing the service to work with objects instead of raw SQL statements. The repository layer typically implements the CRUD (Create, Read, Update, Delete) operations for each model entity.

Here is an example of how the flow works:

The client sends an HTTP request to the server.

The request is received by the controller, which extracts the necessary data from the request.

The controller validates the data and calls the appropriate service method.

The service method receives the request and performs any necessary business logic. It communicates with the repository layer to retrieve or store data as needed.

The repository layer communicates with the database to perform CRUD operations.

The results are returned to the service layer, which processes the data and returns a response to the controller.

The controller sends an HTTP response back to the client with the appropriate data.

46. React lifecycle?

Initialization: This is the stage where the component is constructed with the given Props and default state. This is done in the constructor of a Component Class.

Mounting: Mounting is the stage of rendering the JSX returned by the render method itself.

componentWillMount() Function: As the name clearly suggests, this function is invoked right before the component is mounted on the DOM i.e. this function gets invoked once before the render() function is executed for the first time.

componentDidMount() Function: Similarly as the previous one this function is invoked right after the component is mounted on the DOM i.e. this function gets invoked once after the render() function is executed for the first time

Updating: Updating is the stage when the state of a component is updated and the application is repainted.

componentWillUpdate() Function: As the name clearly suggests, this function is invoked before the component is rerendered i.e. this function gets invoked once before the render() function is executed after the updation of State or Props.

componentDidUpdate() Function: Similarly this function is invoked after the component is rerendered i.e. this function gets invoked once after the render() function is executed after the updation of State or Props.

Unmounting: As the name suggests Unmounting is the final step of the component lifecycle where the component is removed from the page.

• **componentWillUnmount() Function:** This function is invoked before the component is finally unmounted from the DOM i.e. this function gets invoked once before the component is removed from the page and this denotes the end of the lifecycle.

47. What is component in react?

We can say that every application you will develop in React will be made up of pieces called components. Components make the task of building UIs much easier. You can see a UI broken down into multiple individual pieces called components and work on them independently and merge them all in a parent component which will be your final UI. Components in React basically return a piece of JSX code that tells what should be rendered on the screen

Class component: The class components are a little more complex than the functional components. The functional components are not aware of the other components in your program whereas the class components can work with each other. We can pass data from one class component to other class components. We can use JavaScript ES6 classes to create class-based components in React.

Functional component: Functional components are simply javascript functions. We can create a functional component in React by writing a javascript function. These functions may or may not receive data as parameters.

48. What is axios ?

Axios: Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser and it supports the Promise API that is native to JS ES6. It can be used intercept HTTP requests and responses and enables client-side protection against XSRF. It also has the ability to cancel requests.

49. What is fetch?

Fetch: Fetch API provides a fetch() method defined on the window object. It also provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline (requests and responses). The fetch method has one mandatory argument- the URL of the resource to be fetched. This method returns a Promise that can be used to retrieve the response of the request.

50. State and pros ?

State is the local state of the component which cannot be accessed and modified outside of the component. It's equivalent to local variables in a function.

Props, on the other hand, make components reusable by giving components the ability to receive data from their parent component in the form of props.

51. Features in ES 6 ?

Default parameters

- Template literals (Template strings)
- Tagged Templates
- Destructuring assignment
- Arrow function expressions
- let and const
- Spread and Rest syntaxes (...)
- Object.assign() and Object.is()
- Classes

52. Virtual dom?

The virtual DOM (VDOM) is a programming concept where an ideal, or “virtual”, representation of a UI is kept in memory and synced with the “real” DOM by a library such as ReactDOM. This process is called reconciliation.

53. Explain registration page sign up and sign in in your project ?

For registration and login we have created components called register.js and login.js inside register.js we have taken some entities like name , email, phone number, gender, DOB, security question and also added a checkbox of accept policy , terms and condition so it is mandatory to new user to fill these all details also we have provided validation to our page using java script. For login page we have taken entities like email and password so that the old user can sign in easily.

54. Explain registration page sign up and sign in in your project ?

For registration and login we have created components called register.js and login.js inside register.js we have taken some entities like name , email, phone number, gender, DOB, security question and also added a checkbox of accept policy , terms and condition so it is mandatory to new user to fill these all details also we have provided validation to our page using java script. For login page we have taken entities like email and password so that the old user can sign in easily.

55. You did mail invoice ? Explain ?

For Email invoice we took reference from you tube and google. Firstly We have taken one dependency called spring-boot-starter-mail after that we created one email service class inside that we used java mail sender API and then created some methods like SendSimpleEmail(String toEmail, String body, String subject) and sendMailWithInlineResources(String to, String subject, String fileToAttach) which was throwing Messaging exception after that in user controller we imported email service class and overrided his methods and implemented code for it and finally in application.properties we added some configurations of SMTP server like spring.mail.host

spring.mail.port

spring.mail.properties.mail.smtp.auth=true

spring.mail.properties.mail.smtp.starttls.enable=true

56. How you perform role based login?

For role based login we have added extra columns in our user table i.e is_admin , is_agent, is_dealer and inside that column we put some admins, dealers and agents information. So at the time of login whoever user is, he will put his email address and password in login page then that email and password will get match with database credentials if we found particular matched admin or agent or dealer of our staff then respected mapped home page will get open for that user. That's how we done role based login in our project

57. Which advanced features have you used in your project ?

We used react for view, axios for front end and back end connectivity , we provided role based login , mail sender API to send mail , real time reporting , invoicing , to make our web application responsive we have used media min and max queries

58. What are cookies and how you created it ?

Cookies are the textual information that is stored in key-value pair format to the client's browser during multiple requests. It is one of the state management techniques in session tracking. Basically, the server treats every client request as a new one so to avoid this situation cookies are used. When the client generates a request, the server gives the response with cookies having an id which are then stored in the client's browser. Thus if the client generates a second request, a cookie with the matched id is also sent to the server. The server will fetch the cookie id, if found it will treat it as an old request otherwise the request is considered new.

- We use a Cookie class that is present in javax.servlet.http package.
- After that we created an object of Cookie class and passed a name and its value.
- To add cookie in response, we used addCookie(Cookie) method of HttpServletResponse interface.
- To fetch the cookie, We used getCookies() method of Request Interface is used.

59. Which Design pattern are used in your project?

Singleton pattern:- The singleton pattern is a mechanism that ensures only one instance of an object exists per application. This pattern can be useful when managing shared resources or providing cross-cutting services, such as logging.

60. Which database is used in your Project? Why?

We used SQL database because it provides

High speed, Using the SQL queries, the user can quickly and efficiently retrieve a large amount of records from a database.

No coding needed

Well defined standards

Portability

Interactive language

Multiple data view

61. How many web pages are present in your project? And in each module of your project?

There are total 18 Pages are present in our project

Login page, Register page, Admin Home Page, instructor Home Page, Student Home page, order page, Invoice page, Course page, about page , feedback page, payment page, Cart page. Approve request page, manageCourses page,manage users page,footer page, header page,navbar. Etc etc etc.

62. How did you implement look and feel of your web pages? Have you used any framework and why?

Look and feel basically means UI and we have used React JS to build our UI. Concepts like Routing, Nested-Routing, Functional components, Hooks, different libraries such as sweet alert etc. CSS properties, Bootstrap and React Bootstrap. Layout of the page were made through HTML(called JSX in React) and then applying different CSS properties, using inbuilt classes of Bootstrap and components in React Bootstrap, This is how we built UI.

63. Explain me any one part of your project from frontend to the database .?

In our Signup part there are various input field where we input all the appropriate data and when we click on Signup Button, an API call is made to the backend using post method of axios library. the control goes to the backend and it searches for the specified API in the Controller. after finding the API, it will execute that method and call another method from the Service class. The Service class will again call a method from the repository layer and execute that method. After which the repository layer will deal with the database and perform the mentioned statements

64. Now explain me the reverse flow of the data from database to UI ?

When we make an API call to get list of drones using get method of axios, control goes to the controller and then to the Service layer and in turn goes to the repository layer. In repository layer, it executes the method `findByItems(String items)`; Which executes the query mentioned in it and fetches list of drones from a particular location. This list is returned to the Service layer and then again passed on to the Controller and after that this List is sent as a response to the Frontend using `ResponseEntity<List<droneList, HttpStatus.OK>`. This List is received at the FrontEnd in `response.data` which is then used to update the UI to fill the information

65. Tell me which collection do you used in project and example of where and how you used?

In my project we mostly used List collection. Suppose you are building an online course platform where users can enroll in courses and view the course materials. One way to store the course materials is to use a list, where each item in the list represents a module or lesson within the course. Here's an example

66. Write User Controller , Service , Entity?

User Controller:

```
@RestController
@RequestMapping("/api/elearning")
public class MainController {

    @PostMapping("/register")
    public String userRegistration(@RequestBody User userToRegister) {
        System.out.println("request hit" + userToRegister.toString());
        return registerService.register(userToRegister);

    }

    @PostMapping("/login")
    public String userLogin(@RequestBody User obj) {
        System.out.println("inside login");
        String username = obj.getUserName();
        String uname = registerService.getUserName(username);
        System.out.println(username + " " + uname);
        if (uname == null) {
```

```
        return "";
    } else {
        if (uname.equals(username)) {
            String pass = obj.getPass();
            String pwd = registerService.getPassword(uname);

            if (pwd == null) {
                return "";
            }

            if (pwd.equals(pass)) {
                Integer roleid = registerService.getUserRoleId(uname);
                return uname;
            } else {
                return "";
            }
        } else {
            return "";
        }
    }
}

// get rolename by username
@PostMapping("/rolename")
public String getUserRoleName(@RequestBody User user) {
    System.out.println("rolename=" + user.getUserName());
    return registerService.getUserRoleName(user.getUserName());
}

@PostMapping("/user/getuid/{userInfo}")
public String getUserIdByUserName(@PathVariable String userInfo) {

    String result = registerService.getUserIDByUserName(userInfo);

    return result;
}
```

```
@PostMapping("/getuid")
public String getUserIdByUserName(@RequestBody User user) {
    System.out.println("username=" + user.getUserName());
    String result =
registerService.getUserIDByUserName(user.getUserName());
    System.out.println(result);
    return result;
}
```

Repository using JPA query

```
@Repository
public interface RegisterRepository extends JpaRepository<User, Long> {

    @Query(value = "SELECT username FROM elearningdb.users WHERE
username = ?1", nativeQuery = true)
    String findByUsername(String username);

    @Query(value = "SELECT pass FROM elearningdb.users WHERE username
= ?1", nativeQuery = true)
    String findByPassword(String username);

    @Query(value = "SELECT user_id FROM elearningdb.users WHERE
username = ?1", nativeQuery = true)
    String findUserId(String username);

    @Query(value = "select first_name from elearningdb.users where
username = ?1", nativeQuery = true)
    String findFirstName(String userName);

    @Query(value = "select last_name from elearningdb.users where
username = ?1", nativeQuery = true)
    String findLastName(String userName);

    @Query(value = "select email from elearningdb.users where username
= ?1", nativeQuery = true)
    String findMailByUserName(String userName);
```

```
//  
    @Query(value = "select category_id from elearningdb.users where  
    username = ?1", nativeQuery = true)  
    Integer findRoleIdByUsername(String userName);  
  
    @Query(value = "select name from elearningdb.category where id =  
    (select category_id from elearningdb.users where username = ?1)",  
    nativeQuery = true)  
    String findRoleNameByUsername(String userName);  
  
    @Query(value = "SELECT * FROM elearningdb.users", nativeQuery =  
    true)  
    public List<String> findAllUsers();  
  
    @Query(value = "SELECT count(*) FROM elearningdb.users where  
    category_id=?1", nativeQuery = true)  
    public String getUserCount(Integer id);  
}
```

Repository using JPA:

```
public interface RegisterService {  
  
    public String register(User reg);  
    public String getUserName(String username);  
    public String getPassword(String password);  
    public String getUserEmail(String username);  
    public Integer getUserId(String username);  
    public String getUserName(String username);  
  
    public String getUserIdByUserName(String username);  
  
    public String deleteUserByID(Long id);  
    public String getFirstName(String username);  
    public String getLastName(String username);  
  
    public List<User> getAllUsersList();  
}
```

Entity:-

```
@Entity
@Table(name = "users")
public class User {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long userId;

    @Column(name = "username")
    private String userName;

    @Column(name = "first_name")
    private String firstName;

    @Column(name = "last_name")
    private String lastName;

    @Column(name = "email")
    private String email;

    @Column(name = "pass")
    private String pass;

    //    @DateTimeFormat(pattern = "dd-MM-yyyy")
    //    @Column(name = "dob")
    //    private Date DOB;
    //

    @Column(name = "phone_no")
    private String phoneNo;

    @Column(name = "address")
    private String address;

    // @JsonIgnore
    // @ManyToOne
    // private Category category;
```

```
//@JsonIgnore
@OneToMany(mappedBy = "user")
private List<Course> courses;

@OneToMany(mappedBy = "user")
private List<Order> orders;

public User() {
    super();
    // TODO Auto-generated constructor stub
}

//added for store user id from course model
public User(Long l) {
    super();
    this.userId = l;
    // TODO Auto-generated constructor stub
}

public User(Long userId, String userName, String firstName, String
lastName, String email, String pass,
        String phoneNo, String address, Category category,
List<Course> courses) {
    super();
    this.userId = userId;
    this.userName = userName;
    this.firstName = firstName;
    this.lastName = lastName;
    this.email = email;
    this.pass = pass;
    this.phoneNo = phoneNo;
    this.address = address;
```

```
        this.category = category;
        Courses = courses;
    }
    public Long getUserId() {
        return userId;
    }

    public void setId(Long userId) {
        this.userId = userId;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }
```

```
public void setEmail(String email) {
    this.email = email;
}

public String getPass() {
    return pass;
}

public void setPass(String pass) {
    this.pass = pass;
}

public String getPhoneNo() {
    return phoneNo;
}

public void setPhoneNo(String phoneNo) {
    this.phoneNo = phoneNo;
}

public String getAddress() {
    return address;
}

public void setAddress(String address) {
    this.address = address;
}

public Category getCategory() {
    return category;
}

public void setCategory(Category category) {
    this.category = category;
}

public List<Course> getCourses() {
```

```
        return Courses;
    }

    public void setCourses(List<Course> courses) {
        Courses = courses;
    }

    public List<Order> getOrders() {
        return orders;
    }

    public void setOrders(List<Order> orders) {
        this.orders = orders;
    }

    @Override
    public String toString() {
        return "User [userId=" + userId + ", userName=" + userName +
", firstName=" + firstName + ", lastName="
                + lastName + ", email=" + email + ", pass=" + pass +
", phoneNo=" + phoneNo + ", address=" + address
                + ", category=" + category + "]";
    }
}
```

Project

when browser (client) send request to the server means the request is send from our frontend. we use React Js as our frontend technology. In React everything is acts as component. app.js is parent component and rest are its child component. according to the url request from app.js is goes to its particular child component then from child component through the axios request is forwarded to the backend server. axios is popular library is mainly used to send asynchronous HTTP request to Rest endpoints.

through this ^{axios} request frontend send request to the server. then server side we implement Rest API client-server architecture. then cors (cross origin resource) comes into picture (read cors working) & browser take the permission of server for the further request processing if server gives permission then process is continue if permission not given cors throw exception. after successfully get access permission first of all there is controller (presentation layer or API layer) its control all the activity. as per request url its finds its appropriate controller. controller maps all the request coming from client. i.e.

request is post, put, delete, patch.

all the mapping of request done by the controller.

then the controller dispatches the request to the service layer.

service layer contain application business logic. service layer consist of different services method according to request particular method executes.

then from service layer it goes to dao layer. in dao layer we have jpa repository. jpa provides lot of inbuilt method for us. jpa is its ^{just} specification hibernate provide implementation for this. dao layer communicates with database fetch the required data from database and provide data to the service layer for the executn of business logic after completn or successful completn of business logic response is send to controller then controller send the response to the frontend in the JSON or XML format. then the response is display in the form of UI.

**FINAL
PROJECT**

CDAC PROJECT LIST

REACT SPRINGBOOT MYSQL

Enroll

Now!

WWW.CODEWITHARRAYS.IN

Explore More

Subscription : Premium CDAC NOTES & MATERIAL



Contact to Join
Premium Group



Click to Join
Telegram Group

<CODEWITHARRAY'S/>

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

codewitharrays.in freelance project available to buy contact on 8007592194		
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySQL
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySQL
3	Tour and Travel management System	React+Springboot+MySQL
4	Election commition of India (online Voting System)	React+Springboot+MySQL
5	HomeRental Booking System	React+Springboot+MySQL
6	Event Management System	React+Springboot+MySQL
7	Hotel Management System	React+Springboot+MySQL
8	Agriculture web Project	React+Springboot+MySQL
9	AirLine Reservation System / Flight booking System	React+Springboot+MySQL
10	E-commerce web Project	React+Springboot+MySQL
11	Hospital Management System	React+Springboot+MySQL
12	E-RTO Driving licence portal	React+Springboot+MySQL
13	Transpotation Services portal	React+Springboot+MySQL
14	Courier Services Portal / Courier Management System	React+Springboot+MySQL
15	Online Food Delivery Portal	React+Springboot+MySQL
16	Municipal Corporation Management	React+Springboot+MySQL
17	Gym Management System	React+Springboot+MySQL
18	Bike/Car ental System Portal	React+Springboot+MySQL
19	CharityDonation web project	React+Springboot+MySQL
20	Movie Booking System	React+Springboot+MySQL

freelance_Project available to buy contact on 8007592194

21	Job Portal web project	React+Springboot+MySQL
22	LIC Insurance Portal	React+Springboot+MySQL
23	Employee Management System	React+Springboot+MySQL
24	Payroll Management System	React+Springboot+MySQL
25	RealEstate Property Project	React+Springboot+MySQL
26	Marriage Hall Booking Project	React+Springboot+MySQL
27	Online Student Management portal	React+Springboot+MySQL
28	Resturant management System	React+Springboot+MySQL
29	Solar Management Project	React+Springboot+MySQL
30	OneStepService LinkLabourContractor	React+Springboot+MySQL

31	Vehical Service Center Portal	React+Springboot+MySQL
32	E-wallet Banking Project	React+Springboot+MySQL
33	Blogg Application Project	React+Springboot+MySQL
34	Car Parking booking Project	React+Springboot+MySQL
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySQL
36	Society management Portal	React+Springboot+MySQL
37	E-College Portal	React+Springboot+MySQL
38	FoodWaste Management Donate System	React+Springboot+MySQL
39	Sports Ground Booking	React+Springboot+MySQL
40	BloodBank mangement System	React+Springboot+MySQL
41	Bus Tickit Booking Project	React+Springboot+MySQL
42	Fruite Delivery Project	React+Springboot+MySQL
43	Woodworks Bed Shop	React+Springboot+MySQL
44	Online Dairy Product sell Project	React+Springboot+MySQL
45	Online E-Pharma medicine sell Project	React+Springboot+MySQL
46	FarmerMarketplace Web Project	React+Springboot+MySQL
47	Online Cloth Store Project	React+Springboot+MySQL
48	Train Ticket Booking Project	React+Springboot+MySQL
49	Quizz Application Project	JSP+Springboot+MySQL
50	Hotel Room Booking Project	React+Springboot+MySQL
51	Online Crime Reporting Portal Project	React+Springboot+MySQL
52	Online Child Adoption Portal Project	React+Springboot+MySQL
53	online Pizza Delivery System Project	React+Springboot+MySQL
54	Online Social Complaint Portal Project	React+Springboot+MySQL
55	Electric Vehical management system Project	React+Springboot+MySQL
56	Online mess / Tiffin management System Project	React+Springboot+MySQL
57	Online Examination Portal Project	React+Springboot+MySQL
58	Lawyer / Advocate Appointment Booking System	React+Springboot+MySQL
59	Café Management System	React+Springboot+MySQL
60	Agriculture Product Rent system Portal	React+Springboot+MySQL

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/vXz0kZQi5to?si=IiOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7uiQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/lynLouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyl4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKz

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrk7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizlyJlh
15	Employee management system project	https://youtu.be/ID1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWA OzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vn0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

Sr.No	Project Name	YouTube Link
21	Online Bus ticket booking system Project	https://youtu.be/FJ0RUzfMdv8?si=auHjmNgHMrpaNzvY
22	Online Mess / Tiffin Booking System Project	https://youtu.be/NTVmHFDowyl?si=yrvClbE6fdJ0B7dQ
23		
24		
25		

TAP ON THE ICONS TO JOIN!

