

1. Explain POJO? with Example.

→ POJO :-

POJO in Java Stands for Plain Old Java Object. It is an ordinary object, which is not bound by any special restriction. The POJO file does not require any special classpath. It increases the readability & re-usability of a Java program.

POJOS are now widely accepted due to their easy maintenance. They are easy to read and write. A POJO class does not have any naming convention for properties and methods. It is tied to any Java framework; any Java Program can use it.

How to use POJO class in Java Program.

The POJO class created to use to the objects in other Java Programs. The major advantage of the POJO class is that we will not have to create objects every time in other Java programs. Simply we can access the objects by using the get() and set() methods.

To access the objects from the POJO class, follow steps:

- Create a POJO class objects.
- Set the values using the set() method.
- Get the values using the get() method.

Example :-

```
package Jtp.PojoDemo;
public class MainClass {
    public static void main (String args [ ] ) {
        // create an employee class object
        Employee obj = new Employee();
        obj.setName ("Alisha"); // setting the values using set
        obj.setId ("A0001");
        obj.setSal (200000);
        System.out.println ("Name:" + obj.getName ());
        System.out.println ("ID:" + obj.getId ());
        System.out.println ("Salary :" + obj.getSal ());
    }
}
```

Output :- Name : Alisha.

ID : A0001

Salary : 200000

2. Explain DTO ? with example.

→ It is used when we want to pass data with multiple attributes in one shot from client to server. Transfer object is a simple POJO class having getter/setter methods and is serialized so that it can be transferred over the network.

Server side business class normally fetches data from the database and fills the POJO and sends it to the client or passes it by value. For clients, the transfer object is read only. The client can create its own transfer object and pass it to the

server to update values in the database in one shot.

Following the entities of this type of design pattern:

- Transfer Object
- Business Object
- Client

Procedure:

- * Step 1: Create a Transfer object

Example:

Let's first introduce our simple CustomerDTO class.

```
public class CustomerDTO {  
    private final String id;  
    private final String firstName;  
    private final String lastName;
```

```
    public CustomerDTO(String id, String firstname,  
                       String lastName) {
```

```
        this.id = id;
```

```
        this.firstName = firstName;
```

```
        this.lastName = lastName;
```

```
    }  
    public String getId() {
```

```
        return id;
```

```
    }  
    public String getFirstName() {
```

```
        return firstName;
```

```
    }  
    public String getLastName() {
```

```
        return lastName;
```

```
}
```

```
?
```

CustomerResource class acts as the server for customer information.

```
public class CustomerResource {
    private final List<CustomerDTO> customers;
    this.customers = customers;
}
public List<CustomerDTO> getAllCustomers() {
    return customers;
}
public void save (CustomerDTO customer) {
    customers.add (customer);
}
public void delete (String customerId) {
    customers.removeIf (customer -> customer.getId()
        equals (customerId));
}
```

Now fetching customer information is easy since we have the DTOs.

```
var allCustomers = customerResource.getAllCustomers();
allCustomers.forEach (customer -> LOGGER.info (customer
    getFirstName ()));
```

3. Explain Java Bean & Spring

→ Java Bean

A JavaBean is a Java class that should follow the following conventions:

- It should have a no-arg constructor.
- It should be Serializable.
- It should provide methods to set and get the values of the properties, known as getter and setter methods.

Example of JavaBean class.

```
package mypack;
public class Employee implements java.io.Serializable {
    private int id;
    private String name;
    public Employee() {}
    public void setId (int id) {
        this.id = id;
    }
    public int getId () {
        return id;
    }
    public void setName (String name) {
        this.name = name;
    }
    public String getName () {
        return name;
    }
}
```

To access the JavaBean class, we should use getter and setter methods.

```
package mypack;
public class Test {
    public static void main (String args []) {
        Employee e = new Employee(); // object is created
    }
}
```

```
c.setName("Arjun"); //setting value to the object  
System.out.println(e.getName());  
}
```

Spring Bean

The objects that form the backbone of your application and that are managed by the Spring IOC container are called beans. A bean is an object that is instantiated, assembled, and otherwise managed by a spring IOC container. These beans are created with the configuration metadata that you supply to the container. For example, in the form of XML <bean/> definitions which you have already seen in the previous chapters.

Bean definition contains the information called configuration metadata.

4. What is Spring framework? Explain its features and advantages?

→ Spring is a lightweight framework. It can be thought of as a framework because It provides support to various frameworks such as Struts, Hibernate, Tapestry, EJB, JSF, etc. The framework, in broader sense, can be defined as a structure where we find solution of the various technical problems.

Advantages of Spring Framework

1) Predefined Templates :-

Spring framework provides templates for JDBC, hibernate, JPA etc. technologies. So there is no need to write the code of executing query only. Thus, it save a lot of JDBC code.

2) Loose Coupling :-

The spring applications are loosely coupled because of dependency injection.

3. Easy to Test :-

The Dependency Injection makes easier to test the application. The EJB or struts application require server to run the application but spring framework doesn't require server.

4. Lightweight :-

Spring framework is lightweight because of its POJO implementation. The spring framework doesn't force the programmer to inherit any class or implement any interface. That is why it is said non-invasive.

5. Fast development :-

The dependency injection feature of spring framework and its support to various frameworks makes the easy development JavaEE application.

6. Powerful abstraction :-

It provides powerful abstraction to JavaEE

specifications such as JMS, JDBC, JPA and JTA

7) Declarative support:-

It provides declarative support for catching, validation, transactions and formatting.

8. What is REST API and list its applications and uses?

→ A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for representational state Transfer and was created by computer scientist Roy Fielding.

6. What is POM.XML and Explain different dependencies in POM.XML.

→ POM is an acronym for Project Object Model. The POM.XML file contains information of project and configuration information for the maven to build the project such as dependencies, build directory, source directory, test source directory, plugin, goals etc.

Dependencies

* Project :-

It is the root element of pom.xml file.

* modelVersion :-

It is the sub element of project. It specifies the modelVersion. It should be set to 4.0.0.

* groupId :-

It is the sub element of project. It specifies the id for the project group.

* artifactId :-

It specifies the id for the artifact (project).

* Version :-

It specifies the version of the artifact under given group.

7. What is Enterprise web application in software development?

⇒ Web applications that assist in handling the internal and external operations of a company are enterprise web applications. Those web applications can meet the needs of the digital world by shifting the gear of digital transformation. When it comes to enterprise-oriented web applications, they are spread in a wide range and consist of simple chat applications for the inter-departmental communication channel to large applications like that can manage external services like ERP, CRM, etc.

8. Explain Annotations & Its advantages.

⇒ Java Annotation is a tag that represents the meta-data i.e attached with class, interface, methods or fields to indicate some additional information which can be used by Java compiler and JVM.

Annotations in Java are used to provide additional information, so it is an alternative option for XML and Java marker interfaces.

Advantages:-

- * They can produce informational messages for the developer at compile time.
- * detecting errors and suppressing warnings.

- * In addition, annotations can be processed to generate java source files or resources that can be used to modify annotated code.
- 3. Write a simple java program to demonstrate the annotation.

⇒ @override Annotation Example

```
class Animal {
```

```
    public void displayInfo() {
```

```
        System.out.println("I am an Animal");
```

```
}
```

```
class Dog extends Animal {
```

```
    public @Override
```

```
    public void displayInfo() {
```

```
        System.out.println("I am a dog");
```

```
}
```

```
class Main {
```

```
    public static void main (String args[]) {
```

```
        Dog d = new Dog();
```

```
        d.displayInfo();
```

```
}
```

```
}
```

Output :- I am a dog.

10. Write a java program to demonstrate List and
arraylist in Java.

⇒ Java List Example:-

```

import java.util.*;
public class ListExample1 {
    public static void main (String args []){
        // creating a list
        List <String> list = new ArrayList <String>();
        // adding elements in the list.
        list.add ("Mango");
        list.add ("Apple");
        list.add ("Banana");
        list.add ("Grapes");
        for (String fruit : list)
            System.out.println (fruit);
    }
}

```

Output :-

Mango
Apple
Banana
Grapes.

⇒ Java ArrayList Example:-

```

import java.util.*;
public class ArrayListExample {
    public static void main (String args []){
}

```

```
ArrayList<String> list = new ArrayList<String>();
list.add("carrot");
list.add("cucumber");
list.add("pumpkin");
list.add("tomato");
for (String vegetable : list) {
    System.out.println(vegetable);
}
```

Output : [carrot, cucumber, pumpkin, tomato].

11. Write a Java program to demonstrate for-each loop.

```
class ForEachExample {
    public static void main (String args[]) {
        int arr[] = {12, 13, 15, 44, 52};
        for (int i : arr) {
            System.out.println(i);
        }
    }
}
```

Output :-

12
13
15
44
52.

JavaScript Calculator operation

```
<!DOCTYPE html>
<html>
<body>
<script type = "text/javascript">
function multiply () {
    a=Number (document.my-cal.first.value);
    b=Number (document.my-cal.second.value);
    c=a * b;
    document.my-cal.total.value = c;
}

function addition () {
    a=Number (document.my-cal.first.value);
    b=Number (document.my-cal.second.value);
    c=a + b;
    document.my-cal.total.value = c;
}

function subtraction () {
    a=Number (document.my-cal.first.value);
    b=Number (document.my-cal.second.value);
    c=a - b;
    document.my-cal.total.value = c;
}

function division () {
    a=Number (document.my-cal.first.value);
    b=Number (document.my-cal.second.value);
    c=a / b;
    document.my-cal.total.value = c;
}

function modulus () {
    a=Number (document.my-cal.first.value);
}
```

```
b = Number(document.my.cal.second.value);  
c = a % b;  
document.my.cal.total.value = c;  
}
```

```
</script>
```

```
<form name = "my.cal">
```

```
Number 1 : <input type = "text" name = "first">
```

```
<br>
```

```
Number 2 : <input type = "text" name = "second">
```

```
<br> <br>
```

```
<input type = "button" value = "ADD"
```

```
onclick = "javascript: addition(); ">
```

```
<input type = "button" value = "SUB"
```

```
onclick = "javascript: subtraction(); ">
```

```
<input type = "button" value = "MUL"
```

```
onclick = "javascript: multiply(); ">
```

```
<input type = "button" value = "DIV"
```

```
onclick = "javascript: division(); ">
```

```
<input type = "button" value = "MOD"
```

```
onclick = "javascript: modulus(); ">
```

```
<br> <br>
```

```
Get Result : <input type = "text" name = "total">
```

```
</body>
```

```
</html>
```

Output :-

aa6496966

Number 1 :	2			
Number 2 :	3			
<input type="button" value="ADD"/>	<input type="button" value="SUB"/>	<input type="button" value="MUL"/>	<input type="button" value="DIV"/>	<input type="button" value="MOD"/>
Get	Result :	5		

JavaScript Form validation

→ <html>

<head>

<title> Form Validation </title>

<script type = "text/javascript">

</script>

</head>

<body>

<form action = "/cgi-bin/test.cgi" name = "myform"

onsubmit = "return validate();">

<table cellspacing = "1" cellpadding = "1" border = "1">

<tr>

<td> align = "right" > Name </td>

<td> <input type = "text" name = "Name" /> </td>

</tr>

<tr>

<td align = "right" > Email </td>

<td> <input type = "text" name = "Email" /> </td>

</tr>

<tr>

<td> align = "right" > Zip code </td>

<td> <input type = "text" name = "Zip code" /> </td>

```
</tr>
```

```
<tr>
```

```
<td align = "right"> Country </td>
```

```
<td>
```

```
<select name = "country">
```

```
<option value = "1" selected> [choose yours] </option>
```

```
<option value = "1"> USA </option>
```

```
<option value = "2"> UK </option>
```

```
<option value = "3"> INDIA </option>
```

```
</select>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td align = "right"> </td>
```

```
<td> <input type = "submit" value = "submit"/> </td>
```

```
</tr>
```

```
</table>
```

```
</form>
```

```
</body>
```

```
</html>
```

Output:-

Form validation

Name	
Email	
Zip code	
Country	[choose yours] <input checked="" type="checkbox"/>
	<input type="button" value="Submit"/>

* Full stack Development :-
⇒ It refers to the development of both front end (client side) and back end (server side) portions of web application.

* Full stack web Developers :-
⇒ Full stack web Developers have the ability to develop complete web applications and websites. They work on the frontend, backend, database and debugging of web applications or websites.

* Front end :-
⇒ It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website.

Front end languages :-

- HTML (HyperText Markup Language)
- CSS (Cascading Style sheets).
- Javascript

Front end Frameworks and Libraries:-

- Angular JS
- React JS
- Bootstrap
- jQuery
- SASS

* Back end :-

It refers to the server side development of web application or website with a primary focus on how the website works.

Back end Languages:-

- PHP
- C++
- Java
- Python
- Javascript
- Node.js.

Back End frameworks.

Express, Django, Rails, Laravel, spring.

* Database :-

Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from the database and organizes the data in the form of tables, views, schemas, reports etc.

- Oracle
- MongoDB
- Sql (Structured query language).

Full Stack Development

- 1.a) Digital Transformation is creating new - or modifying existing - business processes, culture, and customer experiences to meet changing business and market requirements.

Explain how digital transformation has brought revolution in retail purchases with an example.

→ Digital Transformation in Retail.

Retail Businesses are fighting to meet customer expectations due to the massive push towards ecommerce in the market.

Walmart

- The Walmart has undergone a transformation to keep up with Amazon and other grocery stores turning to things like online ordering and delivery services.
- The latest digital transformation success: voice ordering. Not only shifted customer to shop from comfort to their homes, but they can do voice command devices such as a Google Home.
- They are also adding a visual delivery service, providing customers updates on their deliveries, and allowing visual connectivity so you can see them putting the order away in your kitchen.

The Walmart mobile app now offers visual search, allowing customers to type in the item they are searching for and locate it on the store map.

Digital transformation for healthcare supply Covid-19.

Has there been a modern medical event as monumental as the covid-19 pandemic.

2. a) Identify the tasks involved in the process of receiving Admission ticket for semester exams in your college. Which of the identified tasks can be automated and illustrate automation of the task.

→ Tasks involve the process of receiving Admission ticket.

- * Verifying attendance, voltage
- * Taking No due certificate.
- * Eligible and Not eligible subject
- * Checking criteria, Fees payment Both exam and Admission

All tasks can be Automated for the one process :-

- * Download the Application Exam cell.
- * Currently Admission ticket receiving process manually and is paper based.
- * The Application aims to bring in a system that will ensure the activities in the context examination that can be effectively managed.
- * This system allows students to enroll themselves into the system by their names or sharing details to admin.

- * This is done by providing their personal and all the necessary details like name, email, examination, Register number, score, Branch name, etc.
- * The provided details then entered by admin into the system.
- * Student Admin can verifying the attendance. Student have 75% of attendance. He is eligible for examination.
- * Student have 60% of attendance, He is come to the with health certificate.
- * below 60% of attendance, students are not eligible for exam.
- * Admin verify the attendance data.
- * Admin check the library due certificate, Hostel due certificate, Student have the no due certificate. eligible for the examination.
- * Check the backlog subjects, Student have more than 4 backlog subjects. Students are not eligible for examination.
- * All the information was correct to create their Admission ticket and also created their login id and password.
- * After creating the Admission ticket, the system mails the link of soft copy

of every student who have registered.

- * Student containing link in the mail can view and print the Admission ticket.
- * This process we can achieve the automation of the Admin receiving Admission ticket.

2.b) Identify the following cloud service types and list their characteristics and advantages
Cisco webEx, Google App Engine, Amazon EC2.

⇒ 1. Cisco webEx

Service Types :-

- * webEx suite
- * Calling
- * meetings
- * messaging
- * events
- * CPaaS
- * Collaboration AI

Characteristics :-

- * HD video Conferencing
- * Platform Compatibility and Data protection
- * Convenient virtual meetings anytime, anywhere.
- * Integrated communication
- * Sharing Desktops and Documents

Advantages :-

- * Voice options from phone or computer audio, to video.
- * User can join calls on multiple devices, and can switch devices mid-call.
- * Annotation tools are available.
- * Calls can be recorded.
- * Chat and screen share features are available.
- * Screens can be controlled remotely.
- * Added security features.

2. Google App Engine:

Google App Engine (GAE) is a platform-as-a-service product that provides web app developers and enterprises with access to Google's scalable hosting and tier 1 internet service.

Services :-

- * Processor
- * Storage.
- * application programming interface calls.
- * Concurrent requests.

Characteristics :-

- * API selection.
- * Managed Infrastructure.
- * Several programming languages.
- * Support for legacy runtimes.
- * Application diagnostics.
- * Security features.
- * Traffic splitting.
- * Versioning

Advantages :-

- * makes it easy to use the platform.
- * which offers the flexibility to focus on other concurrent web applications and processes.
- * GAE automatically handles the traffic through patching, provisioning, and monitoring.

3. Amazon EC2

Instance Types

- * Memory-optimized instances
- * Compute optimized instances
- * General purpose Instances
- * Storage optimized instances
- * Accelerated computing instances

Characteristics :-

- * Global Infrastructure
- * Cost and capacity optimization
- * Storage
- * Networking
- * Operating Systems & Softwares

Advantages :-

- * Reliability
- * Security
- * flexibility
- * Cost saving
- * Complete computing solution
- * Elastic web scale computing
- * Completely controlled

User Story

Booking Function Hall

<1> Authentication.

i. Sign-up.

- As an unauthorized user.
- I want to sign up for the website.
- So that I can access or book an convention hall for an event

Acceptance criteria:

- When I sign up, I will use my name, username, Email and password with confirmed password.
- When I sign up successfully, I have to be automatically logged in to the website via.
- When I sign up with an existing email address, I have to receive an error message saying "email exists".

ii Login.

- As an unauthorized user.
- I want to able to login into convention hall booking website.
- So that I can access my profile.

Acceptance criteria:

- When I login, I will use my username & password.
- When I login with an incorrect password, I have to receive an error message saying "invalid password".

- If I forgot the login password. There should be a forgot password button. If you click on it, it will send OTP for your email after entering the correct OTP. It will redirect into the new page to create new password.
- When I login successfully, It will redirect into a home page.

iii. Log-out

- As an authorized user, I want to log out from the website.
- I want to be able to log out from the hall booking website.
- So that I can prevent unauthorized access of my details.

Acceptance criteria:

- When I log out of my account, It will be redirected to the log in page.

(2) Hall booking page

i. Main-page

- As I log in to the website.
- I want to be able to view a list of halls available.
- So that I can check new hall.

Acceptance criteria:

- As I log in, I had to see the hall image, name of the hall & place that where it is.
- As I log in as a user, when I click on an individual halls, It has to redirect to the hall page.

ii. View Hall page.

- As I log in.
- I want to able to view the details and review/rating for each individual hall page.

Acceptance criteria:

- When I click on a individual hall, then It has to redirect to new page, where i can see details about the particular hall.
- When I am on a individual hall page, If I willing to book an hall, it has to show the booking button.

iii. Booking Hall

- As I log in.
- If I am ok with that hall.
- I woud to able to book the hall for the specific date & time, also to pay advance.

Acceptance criteria:

- When I click on book hall button, It has redirect into a new page for booking which includes. time, date and also to accept payment as a advance. etc.

1. Explain how digital transformation can bring revolution in teaching learning process.

Digital Transformation in education is all about making changes to the way we calculate our students.

* This could mean anything from incorporating new technology into the class room, to changing the way we assess and track the student's progress. Whatever changes are made, they should aim to improve overall student outcomes.

* Nowadays teachers are using smart boards and interactive whiteboards as a way to engage students in the classroom.

* By the outbreak of covid-19, all the educational institutions started using digital technologies as a online classes, so that students could continue their education.

* Additionally, teachers can also assign homework that needs to be completed on online. And also students could submit there work online (only).

* The smart teaching technologies includes, google classroom, vidmate, Byjus., assomber, google meetings and more.

* So their are revolution brought in the process of teaching.

2. How DevOps enables faster development of new products and easier maintenance of existing deployments.

→ The word DevOps is nothing but, Development and operations. It aims to improve s/w development, testing, deployment, management and maintenance processes by streamlining, speeding them up, and increasing predictability, scalability, flexibility and website security.

DevOps helps in faster development of new products. Before were the days where a developer used to write a code for days and there have to wait for hours to get it deployed. DevOps has been the game changes in optimizing the entire effort of building and deploying the code. Efforts have come down from days to minutes. Before DevOps were introduced, the product deployment used to take a lot of time, where each other team used to wait for other team for the response.

DevOps helps in easier maintenance of the existing deployments. It helps in increasing the scalability and quality of the deployments. It helps in easy maintenance by providing features for the existing deployments.

There are different devops tools — Git, Github, kubernetes, Jenkins, Tira and more.

3. Passwords alone fail to provide protection because they can be guessed and phished. How to strengthen authentication for a banking portal.



* Passwords has to be very strong and length should contain all the different characters. Do not enter your personal details as a password.

* Enable two-factor Authentication —

Two-factor, or multifactor, authentication can add a second layer of security verification when logging in to your online or mobile banking account.

* First, you enter your login name and password and then you have to pass a second security test

* For example, you may need to enter a special code, verify your account through an automated phone call, use biometric verification or identify an image.

This makes it difficult for a hacker or identity thief to unlock your account even if they have your online or mobile banking password.

* You can also strengthen by enabling OTP, when user you log in.

* Simply you can also use a digital signature before making an transaction which gives a strong authentication.

4.

Ans:-

→ + The term "cloud services" refers to a wide range of services delivered on demand to companies and customers over the internet. These services are designed to provide easy, affordable access to applications and resources, without the need of intend infrastructure or hardware.

* There are 3 cloud services :-

◦ IaaS → Infrastructure as a service.

IaaS is also known as HaaS (Hardware as a service). IaaS is a cloud computing service where enterprises rent or lease services for compute & storage in the hardware.

◦ PaaS → Platform as a service.

PaaS provides a runtime environment. PaaS is a cloud computing model where a third-party provides devices h/w and s/w tools to users, once the internet

◦ SaaS → Software as a service.

SaaS is a way of delivering applications over the internet as a service.

As a consumer I would like to manage -

* IaaS → Applications, OS, data.

+ PaaS → Applications, data

* SaaS → _____.

1. Write a Test plan for online convention hall booking application.

⇒ Objectives

* Purpose :-

This is the plan for testing the architectural prototype of the online booking hall.

* Scope :-

This test plan describes the integration and system tests that will be conducted on the architectural prototype following integration of the subsystem.

The interfaces b/w the following subsystems will be tested :-

- 1) Login.
- 2) Home page with search tab.
- 3) Convention hall details tab.
- 4) Book the hall option.

The external interfaces to the following devices will be tested :-

- 1) Local PC's.
- 2) Remote PC's

The most critical performance measures to test are Response time for login, search tab, details tab, booking the hall.

Requirements for Test :-

* Data and Database integrity Testing

Verify → access to booking hall DB.

→ Simultaneous record. read. accessed.

→ lockout during booking hall details updates.

→ correct retrieval. of update of database

* User interface Testing :-

→ verify ease of navigation through a samples set of users [login page, home page etc].

→ Verify sample screens conform to GUI standard.

* Performance Testing :-

→ Verify response time to access online booking hall application and all its subsystems like -

Response time for login, searching, details and booking.

* Load Testing

→ Verify system response when loaded with 200 logged on users.

→ Verify system response when 50 simultaneous user access to the online booking hall application.

Security & access control Testing

→ Verify login from a local PC.

→ Verify login from a remote PC.

→ Verify login security through user name

and password mechanisms.

* Test Strategy :-

The main considerations for the Test strategy core - the techniques to be used and the criteria for knowing when the testing is completed.

* Testing Types.

→ Function Testing :- The goals of these tests are to verify proper date acceptance, processing and retrieval and the appropriate implementation of the business rules.

→ User Interface Testing :- User interface testing verifies a user's interaction with the software. The goal of UI Testing is to ensure that the user interface provides the user with the appropriate access and navigation through the functions of the applications.

* Tools used :-

Selenium is the tool used for the test management, Test design, Defect Tracking, Functional Testing, performance Testing.

* Resources :-

This section presents the recommended resources for testing the online hall booking application architectural prototype, these main responsibilities and their knowledge or skill set.

* Roles :-

- Test manager → provides management oversight
- Test Designer → identifies, prioritizes and implements test cases.
- System Test → Executes the tests
- Test System administrator → Ensures test requirement and assets are managed & maintained.
- Database Administration → Ensures test data environment and assets are managed and maintained.
- Designer → Identifies and defines the operations, attributes and associations of the test classes.
- Implementer → Implements and writes tests the test classes and test packages.

* Project milestones :-

Testing incorporates test activities for each of the test efforts identified in the previous sections.

Task (Test)	People	start date	end date
planning	2	March 12	March 14
Design	3	March 15	March 18
Development	4	March 19	March 23
Execution	3	March 24	March 28
Evaluation	1	March 29	March 30

* Deliverables :-

- * Test suite :- The test suit will define

the test cases and the test scripts which associated with each test case.

- * Defect Reports :- Rational clearquest will used for logging and tracking individual defects.

→ Project Tasks :-

- * plan test
- * Design test
- * Implement Test
- * Execute Test
- * Evaluate Test.

Sum
Goto Club

Online Convention Hall Booking Application

Test Case	Test data.	Expected Result.	Actual Result	Status	Date	Page
1. login page.	user = ABCD. password=1234	user should be able to log in.	user is navigating to dashboard with successful login	Pass.		
2. Find hall option	valid user.	user should be able to find hall.	user has found a hall using search button	Pass.		
3. Book the hall.	valid user.	user should be able to book hall.	user is navigating to details of hall page	Pass.		
4. display the booking details.	user = ABCD. password=1234	view the details.	user is navigating to hall details page.	Pass.		

Test Case Application.

Edesert

Test Steps	Test data.	Expected Result	Actual Result	Status Pass
1. Search the desert	valid user	user should search the user find the required desert	user is navigating pass.	Pass
2. Sort the desert	search desert	user should able to sort the desert from. to the quality, cost etc.	user is navigated pass.	Pass
3. Cost option.	valid user	user should be able to add the required desert to cart	user is navigating pass	Pass
4. Payment option	VPI	user should be able to access the bank account to epay or account pay.	user is navigating pass	Pass

Date _____
Page _____

2. Write a Test plan for online desert application.

→ Objectives :-

- * purpose :- This is the plan for testing the architectural prototype of the desert which is an online deserts shopping application.
- * Scope :- This Test plan describes the integration and system tests on the desert app.

The interfaces b/w the following subsystems will be tested :-

- 1) login
- 2) search the deserts
- 3) Sort the deserts
- 4) Cart option
- 5) payment mode/option.

The external interfaces to the following devices will be tested on :-

- local PC's
- Remote PC's

The most critical performance measures to test are :-

~~Response time for login, searching, sorting, cart addition and payment process.~~

→ Requirements for Test :-

- * Data and database integrity testing

Verify → access to edesert database.

→ Simultaneous record read accesses.

→ lockout during deserts details updates.

→ Correct retrieval of update of data.

* User interface Testing :-

→ Verify ease of navigation through a sample set of screens [login, search, sort etc].

→ Verify sample screens confirm GUI standards.

* Performance Testing :-

→ Verify response time to access online edesert application and all its subsystems like :-

Response time for login, searching, sorting, pushing to the cart and payment.

* Load Testing

→ Verify system response when loaded with 200 logged on users.

→ Verify system response when 50 simultaneous users access to the online edesert application.

* Security and Access control Testing :-

→ Verify from a local PC. (login process).

→ Verify login from a remote PC.

→ Verify login security through user name and password mechanisms.

* Test Strategy :-

The main considerations for the Test strategy are the techniques to be used and the criteria for knowing when the testing is completed.

* Testing types

- Function testing :- The goals of these tests are to verify proper data acceptance, processing and retrieval and the appropriate implementation of the business rules.
- User Interface Testing :- User interface testing verifies a user's interaction with the software. The goal of UI testing is to ensure that the user interface provides the user with the appropriate access and navigation through the functions of the applications.

→ Tools used.

Selenium is the tool used for test management, test design, defect tracking, functional testing, performance testing.

→ Resources :-

This section presents the recommended resources for testing the edesert application ~~architectural~~ prototype, their main responsibilities and their knowledge or skill set. [you can add names of the employee to be assigned].

→ Roles :-

- Test manager
- Test designer
- System Test
- Test system Administrator

- Database Administrator.
- Designer.
- Implementer

- Project milestone :-
Testing incorporates test activities for each of the test efforts identified in the previous sections.

Task (Test)	people involved	Start date	End date
planning	2	March 12	March 14
Design	3	March 15	March 16
Development	4	March 19	March 21
Execution	3	March 24	March 26
Evaluation	1	March 29	March 29

- Deliverables :-

- * Test suite :- The test suite will define all the test cases and the test scripts which are associated with each test case.
- * Defect Reports :- Rational clearQuest will be used for logging and tracking individual defects.

- Project Tasks :-

- * Plan test
- * Design test
- * Implement test
- * Execute test
- * Evaluate test

*Sum
V grud
R
n tutu*

React JS.

- * React is a library for building user interface.
 - * It is component based.
 - * React is declarative, Fast, Simple.
 - * JS library used on both mobile & Native, components.
 - * React comes under the view part.
 - * Jordan.
 - * It is a open source.
 - * provides a virtual DOM.
 - * browser is real DOM
 - * used to create dynamic web application
 - * Single page application.
 - * Reusable components.
 - * Unidirectional flow of data.
- features of React

- * JSX is syntax extension helps to React
- ```
const simple = <h1> hello </h1>;
```
- JS code                                                                          HTML code. JS.

- \* Virtual DOM.
  - document should be access, modify.
  - Represented in form of a tree.
  - Create virtual DOM exact copy of a real DOM.

When there is a change in web app, which is virtual all the objects in real DOM is updated.

- Virtual DOM compare to its previous state and then React gets to know which object ask to change.
  - It will be change only an object.
- 
- \* One way Data banding.
  - Info. flows in one way.  
They are displayed: modify there.  
They receive the inf through arguments.
  - \* Server side rendering application → Extension switch React as it → mobile & native application through React.
  - \* Components. →
  - \* property we can write. property derived.
  - \* Statement statements.



To types of components

- \* Class component.

```
class App extends React.Component {
```

```
 render () {
```

```
 return <h1>hello, world.</h1>;
```

```
}
```

```
{}
```

- \* Functional component.

```
const App = function () {
```

```
 return <h1> hello, world ! </h1>
```

```
}
```

```
const App = () => <h1>Hello, world</h1>;
```

\* Rendering  
→ ReactDOM.render(  
  <App />,  
  document.getElementById('root'));

\* Props:  
const welcome = { props: {  
  name: 'Erik' } };  
const element = <Welcome name={welcome.name} />;  
ReactDOM.render(  
  element,  
  document.getElementById('root'));

welcome.propTypes = {  
  name: React.PropTypes.string.isRequired  
};

SOS

## Component Lifecycle

- Mounting
- Updating
- Unmounting.
- Mounting → constructor()  
  componentWillMount()  
  render()  
  componentDidMount()
- Updating → componentWillMountReceiveProps()  
  shouldComponentUpdate()

componentDidUpdate()

render()

componentDidUpdate()

- Unmounting → componentWillUnmount()

- Data flow is unidirectional.  
All data flows down the component heirarchy.

### React strengths

- Performance
- Extremely easy to start
- Server side rendering
- Easy to read, maintain, debug.
- Compatible with MVC frameworks
- Compatible with building systems  
(RequireJS, Browserify, Webpack).
- A lot of ready to use components.

### Components

- It is a building block.
- multiple blocks
- each component can be used multiple times.

### State

- State is an object
- object can hold data of the object
- influence the output
- Rendering the object

### Property

- properties allows to pass arguments or data to components.
- helps to make component dynamic

## Design thinking for MRI Scan

- Empathize :- The MRI scanning room should be look like kids playing room.
- Define :- Diagnostic imaging procedure on cutting-edge technology, but at the same time they are a unpleasant experience for patients and even more for pediatric patients. Create a scanner experience that children loved.
- Ideate :- kids are scared by the MRI scanning so we need to design in the way that's look like playing room for kids with attractive and colorful tools in the MRI scanning room.
- Prototype :- prototype is nothing but a first or preliminary version. so we need to ready preliminary version of MRI scanning.
- Test :- you need to discover whether your ideas soever the user problem. by leaving the childrens into the MRI scanning room and note the childrens experience.

1. Demonstrate the use of arrow function and variable substitution.

### → \* Arrow functions

Arrow functions allows a short syntax for writing function expressions.

You don't need the function keyword, the return keyword, and the curly brackets.

example :-

<script>

```
const x = (x, y) => x * y;
document.getElementById("demo").innerHTML = x(5, 5);
</script>
```

### \* Variable Substitutions

Template literals allow variables in strings.

example :-

<script>

```
let firstName = "John";
let lastName = "Doe";
let text = `Welcome ${firstName}, ${lastName}`;
document.getElementById("demo").innerHTML = text;
</script>
```

Ques. Illustrate the working of Function and class component.

### \* Functional Components :-

Functional components are some of the more common components that will come across while working in React. These are simply Javascript functions. We can create a functional component in React by writing a Javascript function.

Example :-

```
import React, {useState} from "react";
```

```
const FunctionalComponent = () => {
```

```
 const [count, setCount] = useState(0);
```

```
 const increase = () => {
```

```
 setCount(count + 1);
```

```
}
```

```
 return <div>
```

```
 <div style={{margin: '50px'}}>
```

```
 <h1> Welcome to Geeks for Geeks </h1>
```

```
 <h3> Counter app using Functional
```

```
Component </h3>
```

```
 <h2> {count} </h2>
```

```
 <button onClick={increase}> Add </button>
```

```
</div>
```

```
}
```

```
export default FunctionalComponent;
```

## \* Class components

A class component must include the extends `React.Component` statement. The component also needs a `render()` method, this method returns HTML.

Example:-

```
import React from 'react';
import ReactDOM from 'react-dom/client';
```

```
class Car extends React.Component {
 render() {
 return <h2>Hi, I am a Car! </h2>
 }
}
```

```
const root = ReactDOM.createRoot(document);
getById('root'));
root.render(<Car/>);
```

## 3. Demonstrate the use of map and set.

### \* Map

- Creating a Map from an Array.
- A map holds key-value pairs where the keys can be any datatype.
- A map remembers the original insertion order.
- A map has a property that represents the size of the map.

Example :-

<script>

```
const fruits = new Map ([
 ["apples", 500],
 ["bananas", 300],
 ["oranges", 200]
]);
```

```
document.getElementById("demo").innerHTML =
fruits.get("apples");
</script>
```

\* Set.

- A Javascript set, is a collection of unique values.
- Each value can only occur once in a set
- A set can hold any value of any data type

example :-

<Script>

```
const letters = new Set(["a", "b", "c"]);
```

```
document.getElementById("demo").innerHTML =
letters.size;
```

```
</script>
```