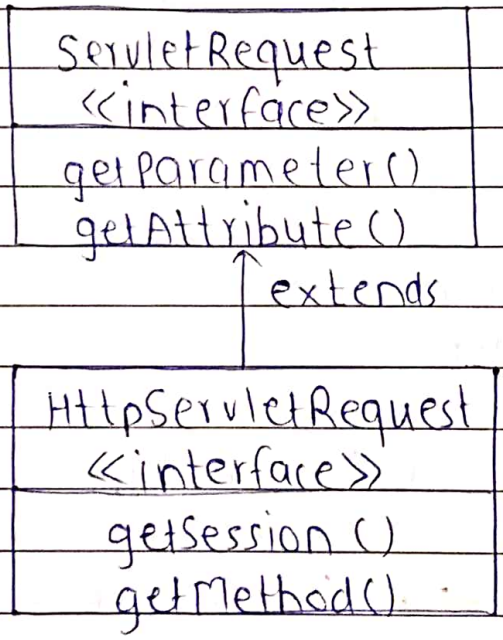
WT
Unit - 3

Q1. Explain HttpServlet Request and HttpServlet Response with suitable examples.

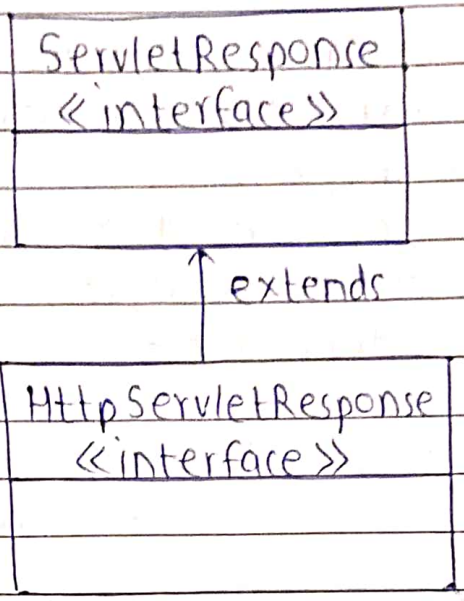
→ HttpServlet Request :-

- Servlet API provides two important interfaces javax.servlet.ServletRequest and javax.servlet.http.HttpServletRequest to encapsulate client request.
- Implementation of these interfaces provide imp information about client request to a servlet HttpServletRequest interface adds the methods that relates to the HTTP protocol.
- The servlet container creates an HttpServletRequest object & passes it as an argument to servlet's service methods (doGet, doPost, etc).



HttpServletResponse :-

- Servlet API provides two important ServletResponse and HttpServletResponse to assist in sending response to client.
- HttpServletResponse interface adds the methods that relates to the Http response.

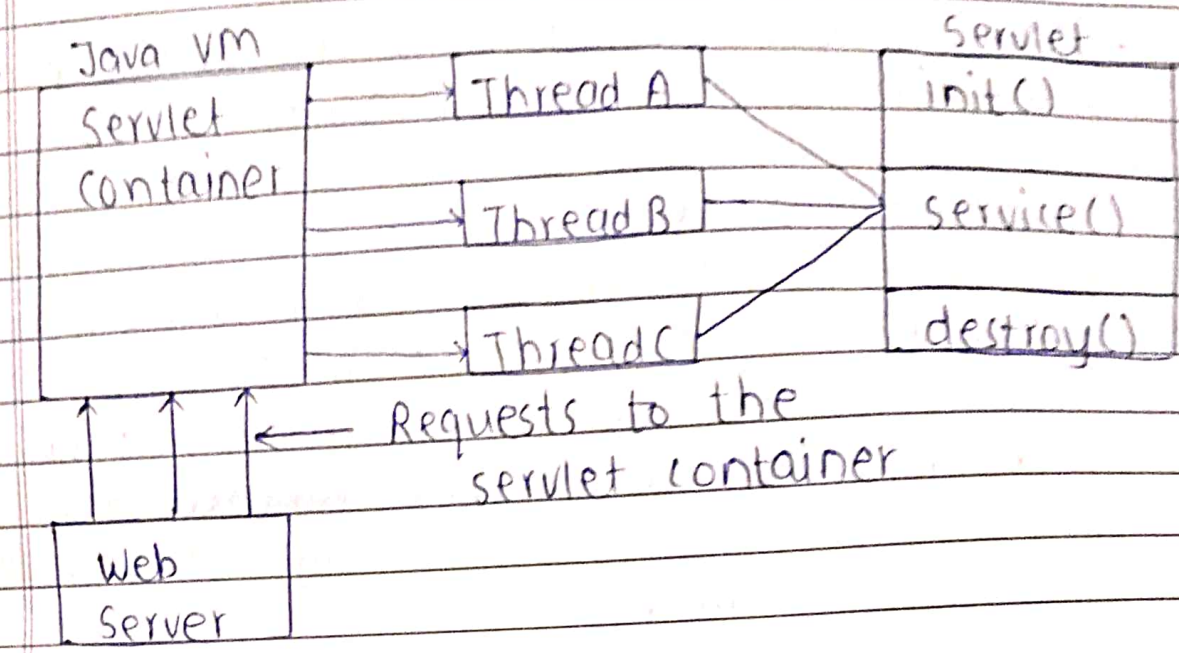


Q.2 Write a simple servlet to print "Hello, Good Day"

→
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Hello extends HttpServlet {
public void init() throws ServletException {}

public void doGet (HttpServletRequest request,
HttpServletResponse response) throws ServletException,
IOException
{
  response.SetContentType ("text/html");
  PrintWriter out = response.GetWriter();
  out.println ("<h1> Hello, Good Day </h1>");
}
  public void destroy() {}
}
```

Q.3 Explain the lifecycle of Servlet. OR
How does Servlet works? OR
Explain the functionality of Servlet



- First the Http Requests coming to the server are delegated to the servlet container.
- The servlet container loads the servlet before invoking the service() method.
- Then the servlet container handles multiple requests by spawning multiple threads, each thread executing the service() method of single instance of the servlet.

Servlet has three methods :-

1) init() method :-
- The servlet is initialized by calling init() method.
- The init() method is called only once.
- It is called only when the servlet is created.
- The init() method simply creates or loads some data that will be used throughout the file of the servlet.

2) service() method :-
- The servlet calls service() method to process a client's request.

- The service() method is the main method to perform the actual task.

i) doGet() method:-
A GET request results from a normal request for a URL or from an HTML form that has no method specified & it should be handled by doGet() method.

ii) doPost() method:-
A POST request results from an HTML form that specifically lists POST as the method & it should be handled by doPost() method.

3) destroy() method:-
- The servlet is terminated by calling the destroy() method.
- The destroy() method is called only once at the end of the life cycle of a servlet

Q.4. Write a servlet which will accept username and password in a form, which will compare both in the code to display success or failure.

OR

Write a servlet to accept username and password from client and verify if username = "ADMIN" & password = "ADMIN" then redirect to success page otherwise redirect to failure page.

→ Html code - S1.html

```
<html>
<body>
<form method="post" action="s2">
Enter Username:
<input type="text" name="t1">
Enter Password:
<input type="password" name="t2">
<input type="submit" value="login">
</form>
</body>
</html>
```

- Servlet file s2.java -

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class s2 extends HttpServlet {

public void doPost (HttpServletRequest request,
HttpServletResponce response) throws ServletException
IOException {
response.SetContentType ("text/html");
PrintWriter out = response.getWriter();
String q1 = request.getParameter ("t1");
String q2 = request.getParameter ("t2");
if (q1.equals ("ADMIN") && q2.equals ("ADMIN"))
   out.println ("<h3> Login successful </h3>");
else
   out.println ("<h3> Login Unsuccessful </h3>");
}
}
```

Q.5. Write a java servlet which will display "Welcome to servlet" message.

→
```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class Hello extends Httpservlet {

public void doGet(HttpservletRequest request,
HttpServletResponse response) throws ServletException,
IOException
{
  response.setContentType("text/html");
  PrintWriter out = response.getWriter();
  out.println("<h1> Welcome to Servlet </h1>");
}
}
```

Q.6. Explain Session and Cookies in servlets. Write any one program to demonstrate session or cookies.

→ • Session Tracking (management):-

- Session Tracking is a way to maintain state of an user. It is also known as session management in servlet.
- We need to maintain the state of an user to recognize to particular user.
- most of the time we don't want to only track the session, we have to store some data into the session that we can use in future requests. This will require a lot of effort if we try to implement this.

- That's why we need session management API and J2EE Servlet technology.
- Servlet API provides session management through HttpSession interface.
- HttpSession allows us to set objects or attributes that can be retrieved in future requests.
- HttpSession.getSession() - This method always returns a HttpSession object. If the request has no session attached, then it creates a new session & return it.

• Cookies :-
- A cookie is a small piece of information that is persisted bet^n the multiple client requests.
- Cookies are the mostly used technology for session tracking.
- Cookie is a key value pair of information sent by the server to the browser. Server can identify the client using the cookie.

Types of Cookie :-
① Non-persistent cookie - valid for single session.
② Persistent cookie - valid for multiple session.

Advantages of Cookies :-
- Simplest technique of maintaining the state.
- Cookies are maintained at client side.

Disadvantages of Cookies :-
- It will not work if cookie is disabled from browser.
- Only textual information can be set in cookie object.

Q.7. Write a short note on Servlet Concurrency.

→ Concurrency means multiple computations are happening at the same time.

- On a server side, multiple requests to the same servlet may be executed at the same time.
- So, concurrency container or web server is multithreaded.
- A thread is a single execution process.
- It is basic unit of CPU utilization consisting of own program counter, a stack and a set of registers.
- A program is multithreaded when multiple threads execute a single instance of a program.
- A servlet must be capable of serving more than one client at a time.
- If several clients issue requests at the same time, methods will serve each client in a different thread.
- service(), doGet(), doPost() can handle many concurrent clients.
- It uses lock mechanism to synchronize the threads.

Q.8. What is XML? OR
What are the strengths of XML technology? Explain the need for XML. OR
What are the strengths of XML technology? Also list the limitations of using XML.

→ • XML :-
- XML stands for Extensible Markup Language.
- XML is text based markup language that enables you store data in structured format by using meaningful tags.
- XML is a cross-platform, software & hardware independent tool for transmitting information.

• Strengths of XML technology :-
1) XML is platform independent & programming language independent.
2) It supports unicode.
3) The data stored & transported using XML can be changed at any point of time without affecting the data presentation.
4) It simplifies data sharing.
5) It allows validation using DTD & schema.
6) XML is readable & understandable.

• Limitations of XML :-
1) XML syntax is verbose & redundant.
2) The redundancy in XML causes higher storage & transportation cost.
3) XML document is less readable
4) XML doesn't support an array.
5) File size are usually very large.

Q9. Write difference beth HTML and XML

HTML	XML
→ 1) Tags are predefined	Tags are not predefined.
2) HTML is static in nature	XML is dynamic

3)	HTML stands for Hyper text markup language	XML stands for extensible markup language
4)	It is used to display the data & control how data is displayed.	XML is used to describe data & focus on what data is (describe data).
5)	It is not mandatory to close tag.	It is mandatory to close tag
6)	Stylesheet for HTML are optional.	Stylesheet for XML called XSL are compulsory for formatting data
7)	Some HTML elements can be improperly nested within each other.	XML elements must be properly nested.
8)	HTML is prerentation language.	XML is neither programming nor presentation language

Q.10. What are components of XML Document. Explain any two.

→ An XML document is composed of a number of components that can be used for representing information in a hierarchical order.

XML Components Are -

① XML declaration
② Tags
③ Elements
④ Attributes
⑤ Comments

1) XML declaration :-

An XML document usually begins with XML document declaration statement.

- An xml declaration declares the version of xml document used to define the document.
- It may also indicate the character encoding used to store or transfer the document.
- It is written as
<?xml version = "1.0" encoding = "UTF-8" ?>

2) Tags :-
- Tags are means of identifying data.
- Tags consist of opening and closing angular bracket (< >).
- Tags occurs in pairs i.e. start tags & end tags.
- These are three types:
i) Start tag (Ex. <book-name>)
ii) End tag (Ex. </book-name>)
iii) Empty-element tag. - This is also known as bodyless tag. ~~(Ex.<~~
Ex. <book-image src = "images/myxml.gif" />

Q.11. Discuss the XSLT technology with an example.
→ - XSL stands for extensible stylesheet language. It is a styling language for xml.
- XSL is the intermediary bet^ the xml elements & the browser. This is the language that tells the browser how to display the various elements, such as <table>.
- XSL is a stylesheet that can be used to transform xml documents into other document types & to format the output.
- The XSL document provides the browser with information on how to display an xml document.

- XSL is a family of recommendations for defining xml document transformation & presentation.
- XSLT stands for XSL Transformations.
- This language takes existing xml documents and transforms them into another documents.
- It allows an xml author to link a very specific locations within an xml document.
- The root element that declares the document to be an stylesheet is <xsl:stylesheet> or <xsl:transform>

Example:
<?xml:stylesheet type="text/xsl" href="class.xsl" ?>
This line indicates that this xml document should be transformed using class.xsl.

Q.12 What is server side scripting & client side scripting? OR
Difference betn server side & client side scripting.

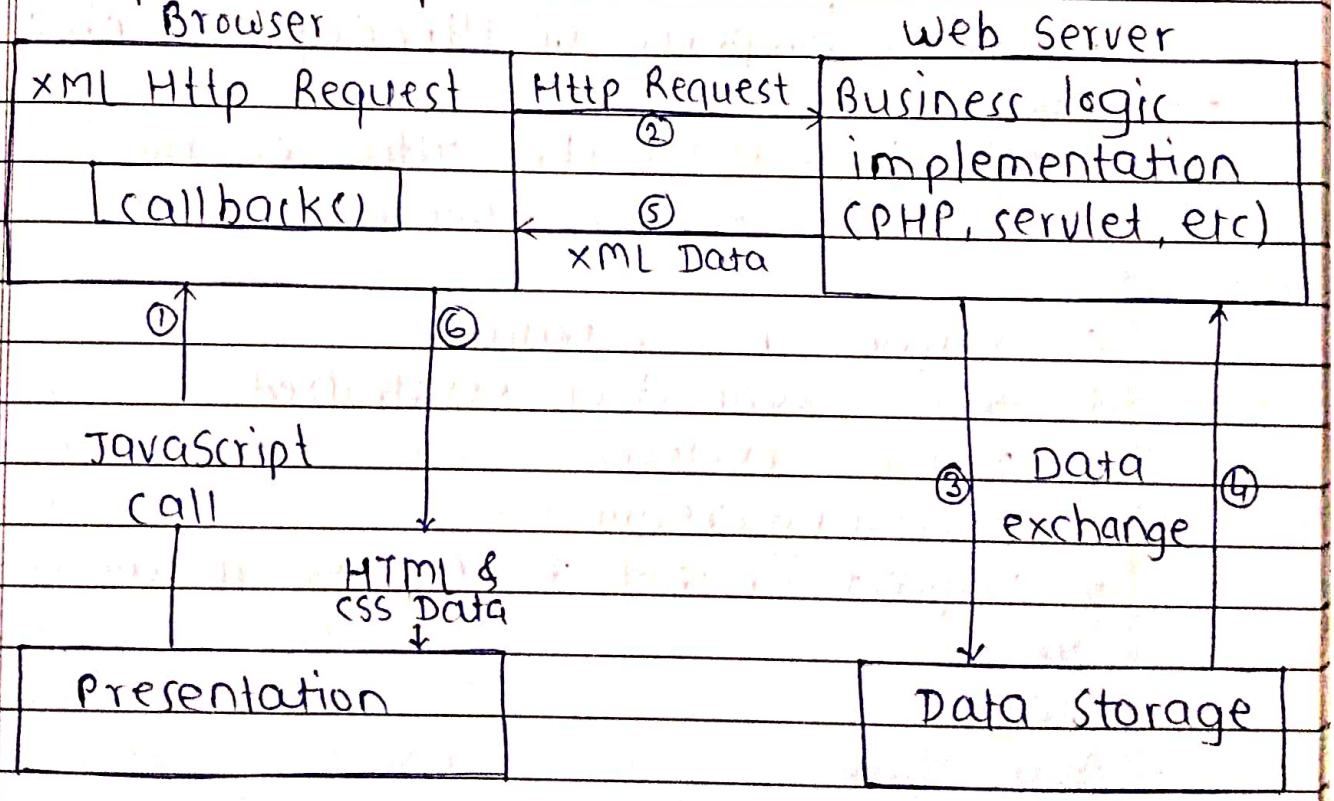
→

	Client side scripting	Server side scripting
1)	In client side scripting the data will access from the browser.	side scripting accepts the data which is submitted by the side scripting.
2)	Validates the input provided by the user.	Creates dynamic HTML as per the input provided by the user.

3) client side script is browser dependent.	server side scripts are browser independent.
4) source code is visible to the user.	source code is not visible to the user because its output of server side is an HTML page
5) No need of interaction with the server.	It is all about interaction with the servers
6) HTML, CSS & javascript are used.	PHP, python, Java, Ruby are used.
7) less security for data	More security for data

Q.13 Draw & explain how Ajax works with the help of example.

→



- User sends a request from the UI & javascript call goes to xMLHttpRequest object.
- Http Request is send to the server by xML HttpRequest object.

- Server interact with the database using JSP, PHP, servlet, ASP.net etc
- Data is retrieved
- Server sends XML data or JSON data to the XMLHttpRequest callback().

Some web applications that make use of Ajax :-

① Google Maps
② Google Suggest
③ Gmail
④ Yahoo Maps

Q.13. List & explain different values of readystate & status property of HttpRequest object.

→ ~~List :-~~

readystate - Holds the status of the XMLHttpRequest.

- 0: request not initialized
- 1: server connection established
- 2: request recieved
- 3: processing request
- 4: request finished & response is ready

Status:

- 200: "OK"
- 403: "Forbidden"
- 404: "Page not found"

Q.14. Write difference betn XML and XSLT.

XML	XSLT
1) XML stands for extensible markup language.	XSLT stands for XSL Transformations
2) XML is used for storing data in a structured format.	XSLT is used for transforming and also for formatting XML file
3) XML does not perform transformation of data.	XSLT performs transformation of one XML document into different XML document or HTML document.
4) XPath is a specialized language that is used to address portions of the XML file	XSLT will be using XPath for transformation and formatting XML file.
5) XML is neither programming nor presentation language.	XSLT is a programming language for processing XML data.
6) XML looks like content.	XSLT looks like template.

Q.15. What are the DTDs? Explain how do they work.

→ - DTD is a Document Type Definition.
- A DTD defines the structure & the legal elements and attributes of XML document.
- DTDs check the validity of structure & vocabulary of an XML document.
- An XML DTD can be either specified ~~eith~~ inside the document or it can be kept in a seperate document and then liked seperately.
- DTD defines following three rules -
1) Specifies the tags & attributes that can be

used to create XML document
2) How to tags combines and reuse
3) Specifies the entities which are represent the special characters

- Syntax of DTD is a follows:

```
<!DOCTYPE element DTD identifier
[
  declaration 1
  declaration 2
  ....
]>
```

- When to use a DTD?
1) With a DTD, you can verify that the data you receive from outside world is valid.
2) You can also use a DTD to verify your own data.

Q.16. Explain difference betn internal & external DTDs.

→

Internal DTD	External DTD
1) In internal DTD, elements are declared inside the XML file.	In External DTD, elements are declared outside the XML file
2) standalone attribute in XML declaration must be set to yes.	standalone attribute in XML declaration must be set to no.
3) You can write rules inside XML document using <!DOCTYPE...> declaration	You can write rules in a seperate file (with .dtd extension)

4)	Syntax of internal DTD is - <!DOCTYPE root-element [element-declarations]>	Syntax of external DTD is - <!DOCTYPE root-element SYSTEM "file-name">
5)		

Q.17. What are xml schemas? How are they better than DTDs?

→ xml Schemas :-

WT
Unit - 4

Q.1. What is JSP? Write advantages of JSP over servlets.

→ JSP :-

- Java server Pages (JSP) technology enables you to mix regular, static HTML with dynamic content.
- JSP technology integrates various java appl^n technologies such as Java servlet, JavaBeans, JDBC & Enterprise JavaBeans.

Advantages of JSP over servlets :-

1) Auto compilation (translation) -
- When the JSP changes, its corresponding servlet are automatically regenerated and jsp container automatically reloads them.

2) Platform independence -
- JSP technology can run on any web server.
- It is supported by wide variety of tools from multiple vendors.

3) Easy and Rapid Web Development -
- JSP server pages simplify and speed up development process.
- JSP provide reusable component (JavaBean, custom JSP tags, Enterprise JavaBeans) which speed up development process.

4) Easy maintenance -
JSP pages are easy to maintain because of seperation of the application logic & page design.

5) Power of server side java-
- With JSP it is possible to use all features of Java into web pages.
- Java platform provides extensibility into the enterprise.

Q.2 Explain life cycle of JSP.
→ A JSP page have ~~seven~~ six phases in its lifecycle-
1. Translation
2. Compilation
3. Loading ~~the class~~ and instantiating
4. _jspInit()
5. _jspservice()
6. _jspDestroy()

1. Translation :-
- The process of JSP page translation is determined by the semantics of a JSP page.
- These semantics includes directives, actions and custom action in JSP page.
- In this phase, the JSP page is read, parsed and validated.

2. Compilation :-
- The java file created in the traslation phase is compiled into a class file.
- All the java code is validated and syntax errors are reported in this phase

3. Loading and Instantiating :-
The servlet class is loaded into memory and

and after successful loading JSP container create instance of the servlet class.

4. jspInit() :-
- The jspInit() method is called only once in the life of the servlet.
- It is the method that we perform any initializations required for the servlet.

5. jspService() :-
- The request and response objects are passed to this method when each client request is received for the JSP page.
- JSP scriptlets & expressions are processed and included in this method.

6. jspDestroy() :-
- The jspDestroy() method is called when the servlet instance is taken out of service.
- Any cleanup operation can be performed in this method.

Q.3 List & ellaborate any 5 jsp implicit objects with example.

→ Implicit objects :- Implicit objects are pre-defined variables used to access request & application data.

List :-
① out ② request ③ response
④ session ⑤ page ⑥ exception

1) out :-
Type - JspWriter
Example - PrintWriter out = response.getWriter();

2) request :-
Type - HttpServletRequest
Example - request.getParameter("t1");

3) response :-
Type - HttpServletResponse
Example - response.sendRedirect("www.google.com");

4) session :-
Type - HttpSession
Example - session.setAttribute("user", name);

5) page :-
Type - object
Example - Object page = this;

6) exception :-
Type - Throwable
Example - <%= exception %>

Q4. Write a JSP program to demonstrate any two components from (Page directive, Scriptlet, Expression and Comment).

→1) Comment :-
- Used to add comments
- Syntax -
<%-- Comment --%>

2) Directive :-
- Directives controls the processing of entire JSP page.
- Syntax - <%@directive %>

3) Declaration :-
- Used to declare variable or method in JSP
- Syntax - <%! declaration %>

4) Scriptlet :-
- Java code is written inside scriptlet tag
- Syntax - <% scriptlet %>

5) Expression :-
- Used to represent the expression in JSP page
- Syntax - <%= expression %>

- JSP Program :-

```
<%@ page language = "java" content Type =
  "text / html" %>
<html>
<body>
<% = (10 + 20) ; %>
</body>
</html>
```

Here, I have added
i) page directive
ii) expression
To provide the information about the page & to execute the expression.

Q5 Draw & discuss MVC architecture in details

→

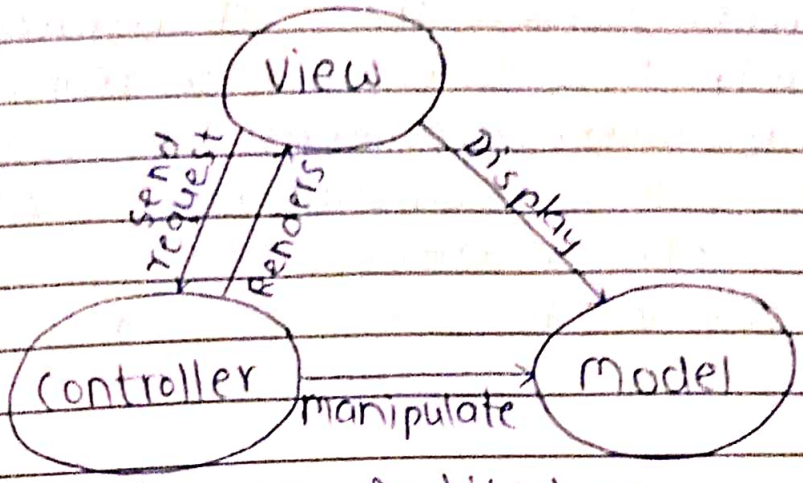


Fig. MVC Architeiture

- The Model-View-Controller (MVC) is an architectural pattern that seperates an application into three main logical components - model, view & controller.
- MVC is one ~~of~~ of the most frequently used industry-standard web depelopment framework to create scalable & extensible projects.

1) Model :-
- Model represents shape of the data & buriness logic.
- It maintains the data of the application.
- Model objects retrieve and store node state in a database.
- Model is a data & business logic.

2) View :-
- View is a user interface.
- View display data using model to the user & also enables them to modify the data.

3) Controller:
- Controller handler the user request.
- Typically, user interact with the view, which in turn raises appropriate URL request, this request will be handled by a controller.
- The controller renders the appropriate view with the model data as a response.
- Controller is a request handler.

Q.6. What are the web services?
→ - Web service is a client server application or application component for communication.
- It is a method of communication betⁿ two devices over network.
- It is a collection of standards or protocols for exchanging information betⁿ two devices or application.
- It is available over the internet or private (intranet) networks.
- It uses a standardized XML messaging system
- It is not tied to any one operating system or programming language.
- It is self-describing via a common XML grammar.
- It is discoverable via a simple find mechanism
- Examples of web services are.
1) OTP Generation System
2) credit card validation system
3) Weather forecast system

Q7 What are the general features / characteristics of web services?

→ 1) XML - Based :-
- Web services uses XML to represent the data and data transportation layers.
- Using XML eliminates any networking, operating system, or platform sort of dependency.

2) Loosely coupled :-
- Loosely coupled means that the web service are not bound to each other, which means that even if the web service changes over time, it should not change the way the client calls the web service.

3) Synchronous or Asynchronous functionality :-
- Synchronicity refers to the binding of the client to the execution of the service.
- Asynchronous operations allow a client to invoke a service & then execute other functions in parallel.

4) Supports Document exchange :-
- One of the key benefits of XML is its generic way of representing not only data but also complex documents.

5) Coarse-grained :-
- In this, a few objects hold a lot of related data.
- It provides broader functionality as compared to fine-grained service.

Q.8 Explain components of web services. OR List & explain layers in protocol stack of web service architecture.

→ Components of Web services :-

1) Web Service Discovery & Publication (UDDI)
2) Web service Description (WSDL)
3) XML Based Messaging (SOAP)
4) Data level Description (XML)

1) UDDI :-

- UDDI stands for Universal Description, Discovery and Integration.
- UDDI is an XML-based standard for describing, publishing, and finding web services.
- UDDI is platform independent, open framework.
- UDDI uses WSDL to describe interfaces to web services.

2) WSDL :-

- WSDL stands for Web services Description language.
- WSDL is an XML-based language for describing web services and how to access them.
- It was developed jointly by Microsoft & IBM.
- It is standard format for describing web service.
- WSDL is the language that UDDI uses.

3) SOAP :-

- SOAP stands for simple Object Access Protocol.
- SOAP is an XML-based protocol for exchanging information betn computers.
- SOAP is a communication protocol.
- SOAP is a format for sending messages.
- SOAP is simple & extensible.

4) Data level description :-
- XML stands for extensible markup language
- XML used to tag the data.
- It is a text-based markup language.
- XML is used to describe the data.

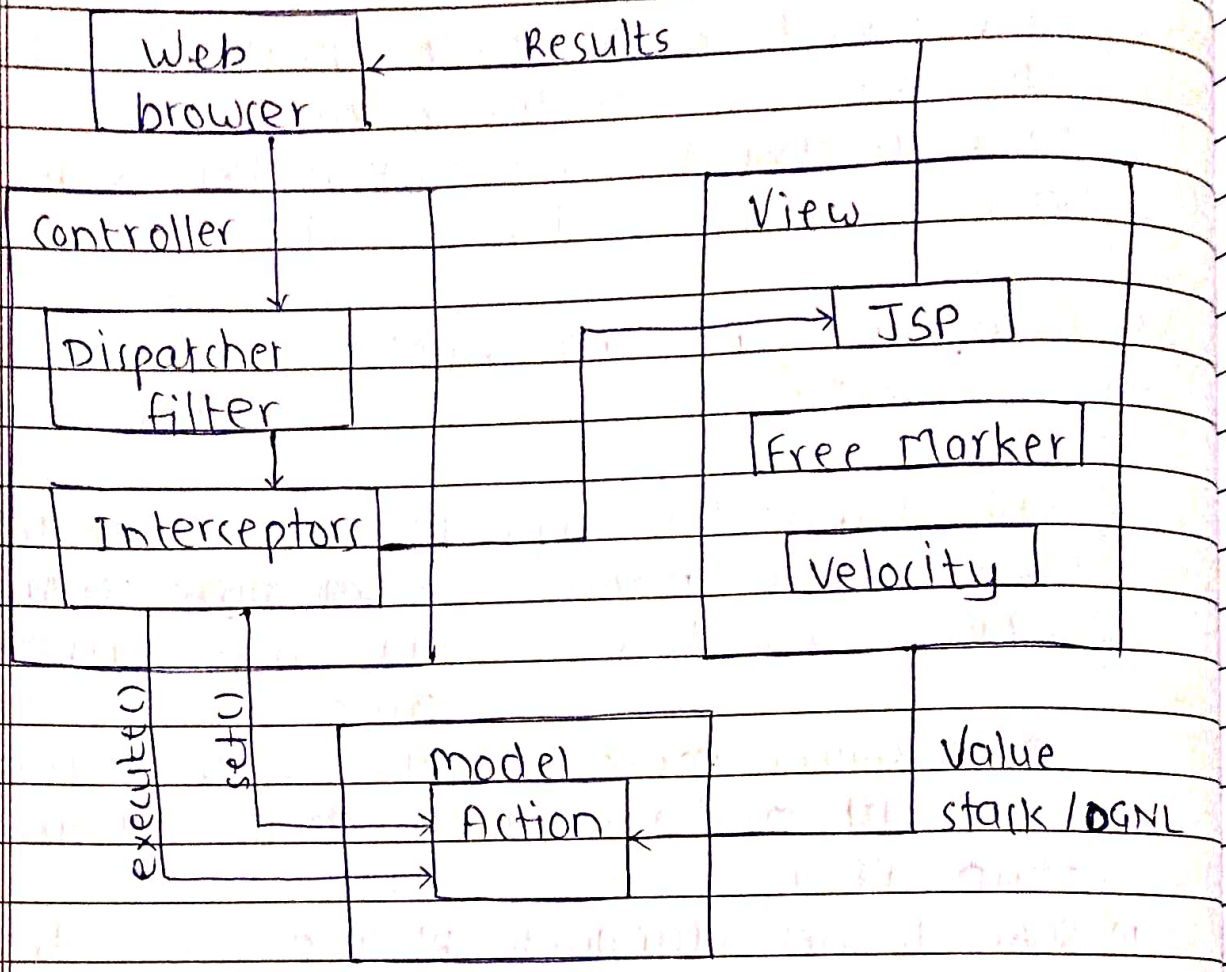
Q.9. Compare SOAP and REST.

→

SOAP	REST
1) SOAP stands for Simple Object Access Protocol.	REST stands for Representational state Transfer.
2) SOAP is a protocol.	REST is an architectural style
3) SOAP requires more bandwidth and resource than REST.	REST requires less bandwidth and resource than SOAP.
4) SOAP defines standards to be strictly followed.	REST does not define too much standards like SOAP.
5) JAX-WS is the Java API for SOAP web services.	JAX-RS is the Java API for RESTful web services
6) SOAP uses services interfaces to expose the business logic.	REST uses URI to expose business logic.

Q.10. Draw and explain neat diagram which depicts MVC to the struts architecture.

→



The MVC framework in Struts 2 has five components:

① Struts 2 Interceptors
② Actions
③ Value stack / OGNL
④ Results
⑤ View

1) Struts 2 Interceptors :-

- Interceptors can be configured according to every action.
- They can execute a code before & after an Action is called.

- Interceptors help in implementing type conversion, object population, validation, file upload, etc.

2) Actions:-
- Actions are the core basic unit of work in Struts 2 framework.
- Each action provides the processing logic for specific URL with which it is linked.
- Actions are mostly associated with a HTTP request of user.

3) Value Stack / OGNL:-
- OGNL is a open source framework used to get properties from Java Beans.
- OGNL provides way to access objects within the value stack
- Value stack is a stack of objects.

4) View:-
- Finally, the result is prepared by the view and returns the result to the user.

Q.11. What are the different configuration files are require to develop any struts application? Explain each configuration file.

→ Configuration files that are required to develop a strut application are:
① web.xml
② struts.xml
③ struts-config.xml
④ struts-properties

1) The web.xml file :-
The web.xml file is a J2EE configuration file that determines how elements of the Http request are processed by the servlet container.

2) struts.xml file :-
The struts.xml file contains the config information that you will be modifying as actions are developed. The file can be created under the folder WEB-INF / classes.

3) struts-config.xml file :-
The struts-config.xml configuration file is a link bet^n the view & model component in the web client but you would not have to touch these settings of your project.

4) struts-properties file :-
The configuration file provides a mechanism to change the default behaviour of the framework

Q.12. Explain various JSP directives.
→ There are three types of directives :-
1. Page directives
2. Include directives
3. Tag library directive

1. Page directive :-
- Page directive gives high-level information about the servlet that will result from the page

- Various attributes of page directives are:
1) import 2) language 3) contentType 4) extends
5) session 6) buffer 7) info 8) errorPage
- Example:
1) <%@ page import = "java.io.*" %>
2) <%@ page language = "java" contentType = "text/html" %>

2 Include directive:-
- It is used to copy the content of one JSP page to another
- This directive is useful if you have a common source that will be used by more than one JSP page.
- Example:
<%@ include file = "header.html" %>

3 taglib directive:-
- Allow user to create custom tags in JSP
- Tag libraries are easy to maintain and reuse.
- Example:
<%@ taglib uri = "http://www.jsp1.com" prefix = "mytag" %>

Q13. Difference betn JSP and servlet

→	JSP	Servlet
1)	It is html based code	Servlet is a java code.
2)	JSP is easy to code	Writing code is hard.
3)	JSP is slower than servlet	Servlet is faster than JSP.
4)	In MVC, JSP acts as view.	In MVC, servlet acts as controller.
5)	Can build custom tags	Cannot build custom tags
6)	Session management is enabled	Session management is not enabled.
7)	JSP modification is fast, just need to click refresh button	Modification is servlet is a time consuming task.

WT
Unit - 5

Q.1 Classify data type of PHP and describe variable data types in each type.

→ Data types of PHP are classified as follows

1) Scalar types → i) Boolean
→ ii) Integer
→ iii) Float
→ iv) string

2) Compound types → i) Array
→ ii) Object

3) Special types → i) Resources
→ ii) Null

1) Scalar types :-

i) Integer :-

- Integer is stored as signed integers with 32 bits.
- The PHP var_dump() function returns the data type and value of variable
- Example -

```
<?php
$x = 5985;
var_dump($x);
?>
```

Output :
Int(5985)

ii) Float :-

- A floating point number is a number with a decimal point
- It is stored with 64 bits.
- Example :-

```
<?php
$x = 10.45;
var_dump($x);
?>
```

Output :
float(10.45)

iii) String:-
- String is a data type representing textual data. It is a sequence of 8-bit characters.
- It can be specified in three ways- ' ', " ", <<<
- Example:-

```
<?php
$x = "Shweta Vyavahare";
echo $x;
?>
```

Output:
Shweta Vyavahare

iv) Boolean:-
- Boolean has two literal values 'true' & 'false'.
- Booleans are used in conditional statements.

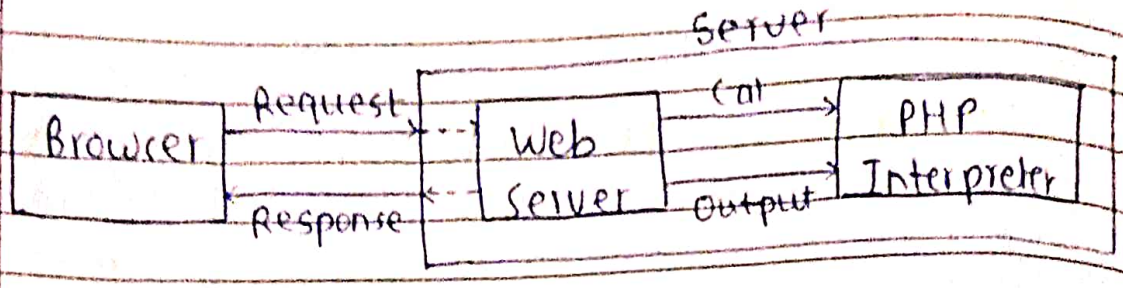
Example:

```
<?php
$male = false;
if ($male) {
  echo "Your name is Om\n";
} else
{
  echo "Your name is sakshi\n";
}
?>
```

Output:
Your name is sakshi

Q2 Explain server side include in PHP with sample code

→



- Browser make a request for web page. This request is passed to the web server. The web server analyze it & determine what to do with it.
- If the request is for PHP file, then the web server will pass that file to PHP interpreter.
- The PHP interpreter will read the PHP file, parse it and then execute it.
- After finishing execution, it'll return an output. The web server will take the output and send it back as response to browser.

PHP code - hello.php

```
<html>
<head>
<title> PHP test </title>
</head>
<body>
<? php echo "<p>Hello World</p>"; ?>
</body>
</html>
```

- The web server is asked to interpret PHP code & convert it to regular HTML code before it sends web page to requesting browser.

- If everything is configured correctly, this file will be parsed & following output will sent to your browser.

```
<html>
<head>
<title> PHP test </title>
</head>
<body>
<p>Hello World </p>
</body>
</html>
```

03. Explain all types of array in PHP. OR
What is Associative arrays in PHP? Explain it with simple PHP code. OR
What is Multidimensional arrays in PHP? Explain it with simple PHP code.

→ In PHP, there are three types of arrays:-
1) Indexed arrays - Arrays with numeric index.
2) Associative arrays - Arrays with named keys
3) Multidimensional arrays - Arrays containing one or more arrays.

1) Indexed arrays:-
- The objects in array having integer index called indexed arrays.
- There are two ways to create indexed arrays:
i) Index can be assigned automatically:

```
$cars = array ("Volvo", "BMW", "Audi");
```

ii) Index can be assigned manually:

```
$cars[0] = "Volvo";
$cars[1] = "BMW";
```

Example:
To create & print array

```
<?php
<html>
<body>
<?php
$cars = array ("Audi", "BMW");
echo "cars are". $cars[0]. $cars[1];
?>
</body>
</html>
```

Output:
Cars are Audi BMW

2) Associative array:
- Associative arrays are arrays that use named keys.
- Both the keys and values can be of any data type in the same array.
- There are two ways to create array:

i) Method 1:

```
$age = array ("Shweta" => 20, "Mayuri" => 19);
```

ii) method 2:

```
$age ['Shweta'] = 20;
$age ['Mayuri'] = 19;
```

Example:

```
<html>
<body>
<? php
```

```
$age = array("Shweta" => 20, "Pratima" => 21);
echo "Shweta is". $age['Shweta']. "years old";
?>
```

output:
Shweta is 20 years old

3) Multidimensional arrays:
- Arrays containing one or more arrays called as multidimensional arrays.
- PHP understands multidimensional arrays that are two, three, four, five or more levels deep.
- Values in multidimensional array are accessed using multiple index.

Example:
Consider the following table

Name	~~Car~~ Stock	Sold
Audi	27	18
BMW	22	10

```
<html>
<body>
<? php
$cars = array(
        array ("Audi", 27, 18),
        array ("BMW", 22, 10));
echo "car name is". $cars[0][0]. "car stock is".
    $cars[0][1]. "cars sold are". $cars[0][2];
?>
</body>
</html>
```

output:
car name is Audi cars ~~sold~~ are 27 cars sold are 18

Q.4 Write PHP script to display the squares & cubes of 1 to 10 numbers.

→
```
<html>
<body>
<table>
<tr>
    <th> Integers 1 to 10 </th>
    <th> Square </th>
    <th> Cube </th>
</tr>
<?php
for ($i=1; $i<=10 ; $i++)
{
  $a = $i * $i ;
  $b = $i * $i * $i ;
    echo "<tr>";
    echo "<td>" $i. "</td>";
    echo "<td>" $a. "</td>";
    echo "<td>" $b. "</td>";
    echo "</tr>";
}
?>
</table>
</body>
</html>
```

OR

```
<?php
  for ($i=1; $i<=10; $i++)
  {
echo "square of $i is". $i * $i;
echo "cube of $i is". $i*$i*$i;
} ?>
```

Q.5. What are cookies ? Explain cookies in PHP.

→ - Cookies are small bits of textual information that a web server sends to a browser.
- A cookie is often used to identify a user.
- A cookie is a small file that the server

embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too.
- With PHP you can both create and retrieve cookie values.

Advantages of cookies:-
1) Remembering username and passwords
2) Focusing advertising
3) Customizing sites
4) Identifying user during E-commerce session.

Example: Create / Retrieve a cookie

```
<? php
setcookie ("name", "Amit", time()+(86400*30), "/");
?>
```

```
<? php
if (isset ($_COOKIE["name"]))
{
 $nm = $_COOKIE["name"];
 echo "Hello", $nm;
}
else { echo "Cookie is not set"; }
?>
```

Q6. Explain session management techniques in PHP.
→ - When you work with an application, you open it, do some changes, and then you close it. This is much like a session.
- The computer knows who you are. It knows

when you start the appl^ & when you end. But, on the internet there is one problem: the web server does not known who you are or what you do, because the Http address doesn't maintain state.

- Session variables solve this problem by storing user information to be used across multiple pages. By default, session variables last until the user close the browser.
- So, session variables hold information about one single user, and are available to all pages in one application.
- A session is started with session_start() function
- Session variables are set with the PHP global variable: $_SESSION

Example:

```
<? php
session_start(); // Start the session
?>
<html>
<body>
<? php
$_SESSION["favcolor"] = "green"; //set session variable
echo "Session variables are set";
?>
</body>
</html>
```

Q.7. List & Explain steps involved in connecting to mySQL with PHP.

→ Following are the five steps:
1) Connect to the DBMS (mysql)
2) Select the database required
3) Run the query
4) Retrieve and process result
5) close the DBMS connection.

1) Connect to the DBMS (mysql) :-
- First step is open a connection to the mysql server. The function used to connect to mysql is called mysql_connect.
- Syntax :
mysql_connect (hostname, username, password);

2) Select the database :-
- Once you've connected to database, select a database to work with.
- Select a database using mysql select db() function.
- Here we assume that ness is name of database created
$selected = mysql_select_db ("ness", $dbhandle)

3) Run the Query :-
- Run a query on the database using mysql query() functions.
- Following functions are used in running query:
1) mysql_query (SQL, conn) - Executes the query & returns query handler
2) mysql_affected_rows - Returns the num of row affected for the last query.
3) mysql_num_rows (query handler) - Returns the no of records in the resultant for the select query.

4) Retrieve and Process result:
- To retrieve and process result php uses a function mysql_fetch_array()
- It's a good practice to release cursor memory at the end of each SELECT statement by using function mysql_free_result()

5) Close the DBMS connection:-
- Finally, PHP will automatically close the connection when the script ends.
- DBMS connection is closed using mysql_close()
- Syntax:
  mysql_close($dbhandle);

Q.8. Write a short note on Node JS.
→ - Node JS is a runtime library & environment
- It is cross-platform & used for creating running Javascript appln outside the browser
- It is free & open-source and utilized for creating server-side JS applications.
- Node JS allows developers to execute their code on the server-side
- It provides a faster way to write scripts that are scalable and light.
- Node JS can generate dynamic page content
- Node JS runs on various platforms (Windows, Linux, Unix, Mac OS, etc)
- Node.js is used by many large companies such as Paypal, LinkedIn, Mozilla, etc

real-time applications of Node JS are.
1) Chat applications
2) Game servers
3) Advertisement servers
4) Streaming servers.

22 What are various stages in ASP.net life cycle? Explain each stage in short.

→ Stages in ASP.NET life cycle are.
① Page request
② Page Utilization
③ starting of page life cycle
④ Page load
⑤ Validation
⑥ Post back event handling
⑦ Page Rendering - info is saved & result is sent to user
⑧ Upload - removing all unwanted objects

1 Page request :-
When ASP.NET gets a page request, it decides whether to parse & compile the page or there would be a catched version of the page, accordingly the responce is sent.

2 Page Utilization :-
At this stage, the controls on the page are assigned unique ID by setting the unique ID properly & the themes are applied.

3 starting of page life cycle :-
At this stage the Request & Response objects are set.

4. Page load :-
At this stage, control properties are set using the new state & control state value.

5. Validation :-
Validate method of validation control is called & on its successful execution the IsValid property of page is set to true.

6. Post back event handling :-
If the request is a post back, the related event handler is invoked.

.NET framework Architecture

Q.10. What is .NET framework? Explain various components of .NET framework 4.0.

→ .NET framework :-
- .NET framework is a framework designed to run application on windows operating system.
- .NET creates two types of application -
i) form based application
ii) web based application.

There are three components of .NET framework
1) CLR (Common language Runtime) :-
- "Common language Infrastructure" or CLI is a platform in .NET architecture on which the .NET programs are executed.
- CLI has following key features -
1. Exception Handling
2. Language
3. Compiler
4. Common language Interpreter

2) Class library :-
- The .NET framework includes a set of standard class libraries
- A class library is a collection of methods & functions.

3) Languages :-
There are three languages of .NET framework:
i) WinForm : It is used to create form-based applications in .NET framework
Example - Notepad, Notepad++

ii) ASP.Net : It is used to create web-based applications in .NET framework
Example - Firefox, Chrome

iii) ADO.Net - It is used to create application having database connections.
Example - Microsoft, SQL server.

Q1) What are ASP.NET Server controls ?
→ - Server controls are tags that are understood by the server.
- ASP.NET server control is a tag written in a web page to represent programmable server-side object.
- ASP.NET server controls are tags that can be understood by the server.
- They are coded in .aspx file and expose properties, methods & events of the control
- There are three kinds of server controls:

1) HTML Server Controls:
- Traditional HTML tags
- Example:

Control name	HTML tag
HtmlHead	<head> element
HtmlInputButton	<input type = button>
HtmlInputFile	<input type = file>
HtmlText	<input type = text>
HtmlImage	<img> element

2) Web Server Controls:
- New ASP.NET tags
- ~~Example~~ : Types:
i) List Controls - Supports binding to collection
ii) Rich Controls - calendal control
iii) Validation Controls - Used to validate values

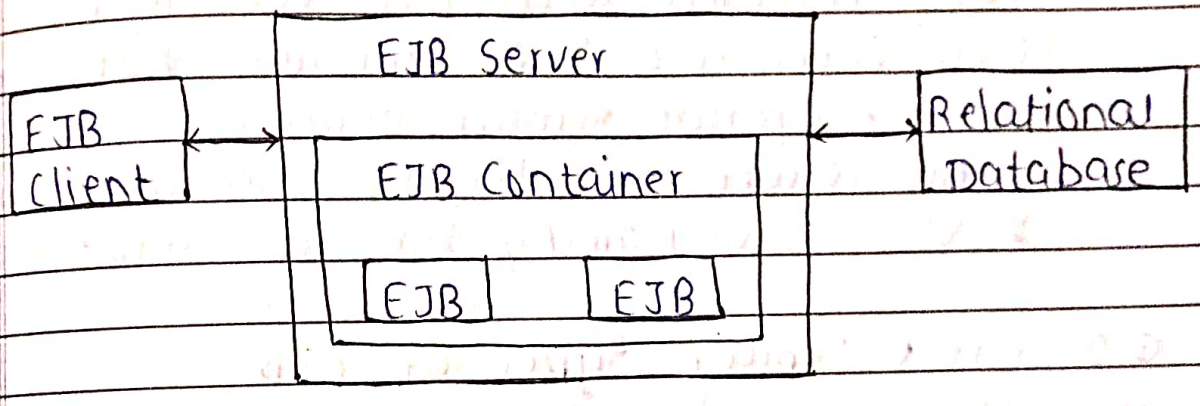
3) Custom Controls:
- User defined controls
- Useful if you need to combine functionality of two or more built-in web controls.
- Useful if you need to extend the functionality of a built-in control.

Unit - 6

Q1 What is EJB ? Draw & explain main components of EJB architecture

→ EJB :-
- EJB stands for Enterprise Java Beans
- EJB provider an architecture to develop component based enterprise applications with robustness & high scalability.
- There are three types of EJB :-
a) Session Bean b) Message Driven Bean
c) Entity Bean

• EJB Architecture :-



1. EJB Server :-
- It is the outermost container of EJB architecture
- It manages EJB container
- It provides runtime environment
- It performs services like - process & thread management, system resource management, Database connection.

2. EJB container :-
- It contains Enterprise beans
- It provides services, life style management, Transaction management

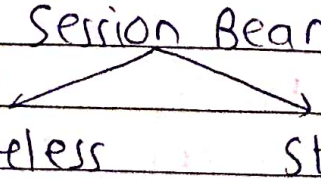
- It provides security.

3. EJB :-
- They are Enterprise Java Beans.
- There are various types of EJB :
  ① Session Beans → Stateless
  └→ Stateful
  ② Message Driven Beans
  ③ Entity Beans → Bean Managed Persistence
  └→ Container Managed Persistence

4. EJB Clients :-
- They interacts with EJB container.
- EJB Client is a local program which can call & operate remote beans.
- Client locater an Enterprise Java bean through JNDI (Java Naming Directory Services).

Q.2. List & explain types of EJB.
→ There are three types of EJB :-
① Session Beans
② message driven Beans
③ Entity Beans

1. Session Beans :-
- Session bean stores data of a particular user for a single session.



Stateless session bean :-
- Stateless session bean is business object that represents business logic only & doesn't have state.
- Conversational state between multiple method calls is not maintained by the container.
- Annotation used - @Stateless

Stateful session bean :-
- Stateful session bean is business object that represents business logic & maintains state.
- Conversational state between multiple method calls is maintained by the container.
- Annotation used - @Stateful

2. Message Driven Bean :-
- A message driven bean is a bean that contains business logic. But, it is invoked by passing the message.
- MDB receives message from queue & process it.
- MDB is like stateless session bean that represents business logic & doesn't maintain state.

3. Entity Beans :-
i) Container-managed persistence :-
- The container manages data by saving it to the database

ii) Bean-Managed persistence :-
- The bean implementation manages data within callback methods.
- The container invoke there methods when necessary.

Q.3 What are the uses of Ruby?

→ Uses of Ruby are -

1) Object :-
As Ruby is an object oriented programming language, everything is object.

2) Code development :-
Development of code is much faster than any other programming languages.

3) Maintability :-
Ruby makes developers for easy to maintain and understand. It makes the code to run faster.

4) Code Quality :-
Ruby provides a good quality of code to the application & it is easy & simple to read.

5) Security & Performance :-
Ruby ensures the high performance & secured deliverable application for better customer experience.

6) Dynamic Typing :-
It results to programs being more simple, faster & dynamic to code.

Q.4. Explain data types of Ruby. Also explain different operations that can be performed on scalar data types.

→ There are 5 data types of Ruby:-

① Scalar → Numeric
→ String

② Arrays
③ Boolean
④ Hashes
⑤ Symbol

1. Scalar :-

Scalar have 2 data types-

a) Numeric : They are digits or integers with decimal point (if needed)

Ex - int_num = 34;

b) String : They are serier of alphabets called strings

Ex - my_str = "Hello";

2. Arrays :-

They are used to store objects of different data types.

Ex - My_arr = ["Hi", 22, "A", true];

3. Boolean :-

- Boolean data type represents only 1 bit of information & returns value if True or False
- There are three boolean operators:
  NOT (!), AND (&&), OR (||)

4. Hashes :-

- A hash stores key-value pairs
- Assigning value is done by using => sign
- Ex - My_Hash = {"name" => "ABC", "Rno" => 21}

5. Symbols :-

- Symbols are lighter form of strings.
- They are preceded by a colon (:) & used instead of strings.
- Ex - my_sym = {:ap => "Apple", :bn => "Banana"}

• Operations on Scalar data types :-

1. Numbers :-

i) [.even?] → it is used to check whether given integer is even or not.
Ex - 7.even? # => false

ii) [.odd?] → it is used to check whether given integer is odd or not
Ex - 7.odd ? # => true

iii) [.pred] → it is used to give no. preceding given integer.
Ex - 15.pred # => 14

2. String :-

i) [str.size] → it is used to measure the size of string.
Ex. "Hello".size
Output : 5

ii) [str.reverse] → it is used to reverse the given string
Ex. "Hello".reverse
output : "olleH"

iii) [str.upcase] → It is used to make whole string in upcase
Ex. "hello".upcase
output : HELLO

Q5. Explain Arrays in Ruby with example how to create array?

→ Ruby - Arrays :-

- Ruby arrays are ordered, integer-indexed collections of any object. Each element in an array is associated with & referred to by an index.
- An index of -1 indicates the last element of the array, -2 is the next to last element in the array, and so on.
- Ruby arrays can hold objects such as string, Integer, Fixnum, hash, symbol, even other Array objects.

• Creating Arrays :-

Method 1 : cars = Array.new
⇒ Creating array using new method

Method 2 : cars = Array.new(20)
⇒ You can set the size of an array at the ~~car~~ time of creating arrays.

Method 3 : cars = Array.new(3, "Audi")
⇒ This will produce o/p
["Audi", "Audi", "Audi"]

Q6 What are Hashes in Ruby? Explain with example how to create hashes.

→ Hashes :-

- A Hash is a collection of key-value pairs like this : "employee" => "salary". It is similar to an array. except that indexing is done via arbitary keys of any object type, not an integer index
- If you attempt to access a hash with a key that does not exist, the method will return nil.
- Key-value pairs are seperated by commas & all pairs are enclosed within curly brackets.

• Creating Hashes :-

Method 1 : months = Hash.new

⇒ Creating hash using new method

Method 2 : months = Hash.new ("month")

⇒ Hash with a default value

Method 3 : H = Hash["a" => 100, "b" => 200]

⇒ puts "#{H['a']}"

This will produce OIP - 100

Q.7 How to take input and output in Ruby? Explain with example.

→ 1) Input :-

- gets method is used to take input from keyboard.
- This method reads line of input.
- gets.to_i to take integer value & gets.to_f to take float value.

Syntax:
val = gets

Example:
num1 = gets.to_i
num2 = gets.to_i
sum = (num1) + (num2)
puts "The sum is #{sum}"

2) output :-
- puts method is used to display the output on the screen
- Operands for the puts method is string
- To display variable value use #{...}

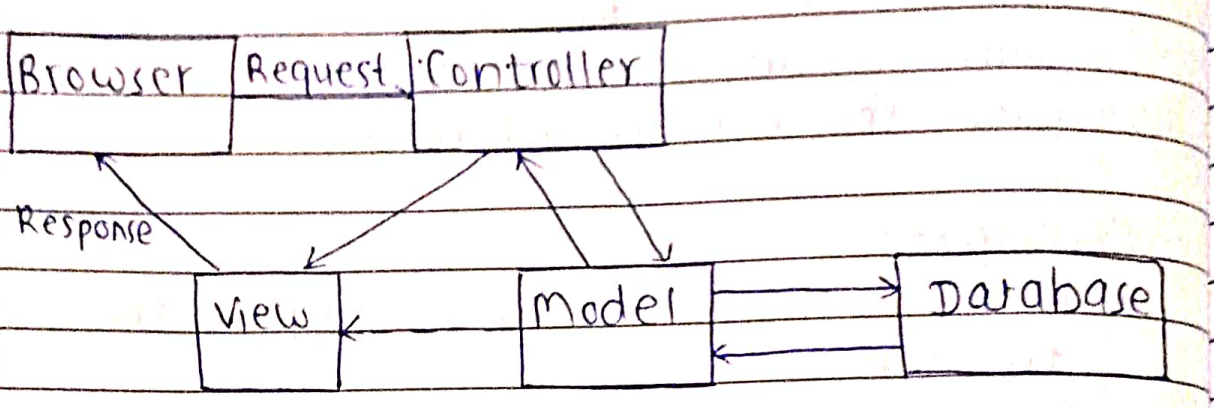
Syntax:
puts "val"

Example:
val1 = "This is Ruby"
puts "The statement is #{val1}"

Q.8 Write short note on Rail
→ • Rail :-
- Ruby on Rails is a server-side web application development framework
- It is based on MVC pattern.
- You could develop a web appl^{r} at least ten times faster with Rails than you could with a typical java framework
- It is an open source Ruby framework for

developing database backed web applications
- Configure your code with Database Schema
- No compilation phase required.

- Document Request :-



- Advantages of Ruby on Rail :-

1) Cost-effective - Ruby on Rails is free & open-source framework
2) Secure - Using Ruby on Rails is very secure
3) Flexibility - Creating web applications is simpler and easier.
4) Productivity - Ruby is incredibly fast from another language. Its productivity is high
5) Code Quality - Ruby code quality significantly higher than PHP or NodeJs
6) Tooling - Ruby provides tooling that helps us to deliver more features in less time.

- Dicadvantages Of Ruby on Rail :-

1) Runtime speed is slow as compare to NodeJs
2) lack of Flexibility
3) Boot speed
4) Documentation
5) Threading.

- Uses :-

1) In long term project which needs large transformation.
2) in the project that has heavy traffic.
3) to develop a short prototype.
4) in a project that requires wide range of complex function.

Q9. Explain Rails with Ajax

→ Ajax enables you to retrieve data for a web page without having to refresh the contents of the entire page.

- Rails has a simple, reliable & model for how it implements Ajax operations. Once the browser has rendered & displayed the initial web page, different user actions cause it to display a new web page

- Some trigger fires - This trigger could be the user clicking on a button or link.
- The web client calls the server - A Javascript method, XMLHttpRequest sends data associated with the trigger to an action handler on the server.
- The server does processing - The server-side action handler does something with the data & returns an HTML fragment to the web client.
- The client receives the response - The client-side Javascript receives the HTML fragment & uses it to update a specified part of current page's HTML.