

WEB TECHNOLOGY UNIT – 4 PYQ

➤ **MAY / JUN 2022**

Q3)

a) Explain life cycle of JSP. Write advantages of JSP over servlet

JSP Life Cycle

A JSP life cycle is defined as the process from its creation till the destruction. This is similar to a servlet life cycle with an additional step which is required to compile a JSP into a servlet. The following are the paths followed by a JSP:

1. Compilation

When a browser asks for a JSP, the JSP engine first checks whether it needs to compile the page. If the page has never been compiled or if the JSP has been modified since it was last compiled, the JSP engine compiles the page.

The compilation process involves three steps:

- **Parsing the JSP** – The JSP engine checks the syntax and structure of the JSP file.
- **Turning the JSP into a servlet** – The JSP is converted into an equivalent Java servlet (.java file).
- **Compiling the servlet** – The generated servlet code is compiled into a .class file.

2. Initialization

When a container loads a JSP, it invokes the `jspInit()` method before servicing any requests. If you need to perform JSP-specific initialization, you can override the `jspInit()` method:

```
java

public void jspInit() {
    // Initialization code (e.g., database connections, file handling)
}
```

3. Execution

This phase represents all interactions with requests until the JSP is destroyed.

- When a browser requests a JSP, and the page has been loaded and initialized, the JSP engine invokes the `_jspService()` method.
- This method handles **all HTTP requests** (GET, POST, DELETE, etc.) and generates the response.

```
java

void _jspService(HttpServletRequest request, HttpServletResponse response) {
    // Service handling code (request processing & response generation)
}
```

4. Cleanup (Destruction)

The destruction phase occurs when the JSP is removed from use by the container.

- The `jspDestroy()` method is called (similar to `destroy()` in servlets).
- Used for cleanup tasks like releasing database connections or closing files.

```
java

public void jspDestroy() {
    // Cleanup code (e.g., closing resources)
}
```

Advantages of JSP over Servlet:

1. **Ease of Use (Less Code):**
JSP allows embedding HTML directly with Java, making it easier and cleaner than writing HTML inside Java code in Servlets.
2. **Separation of Concerns:**
JSP focuses on the **presentation layer**, while Servlets are better suited for **business logic**. This promotes MVC architecture.
3. **Faster Development:**
Since most of the UI work is in HTML with embedded tags, JSP development is faster and more intuitive for front-end design.
4. **Built-in Features:**
JSP supports implicit objects (like request, response, session) and custom tags which simplify coding.

b) Write a short notes on : i) SOAP ii) WSDL

i) SOAP (Simple Object Access Protocol) :

1. SOAP is an XML-based protocol used for exchanging structured information in web services.
2. It is platform-independent and commonly uses HTTP or SMTP for message transmission.
3. SOAP messages consist of an envelope, a header, and a body, where the envelope defines the message boundaries, the header contains metadata, and the body holds the main content.

Feature :

- **Platform & Language Independent:** SOAP uses XML, making it compatible across different operating systems and programming languages.
- **Protocol Agnostic:** Although commonly used with HTTP, SOAP can also work over SMTP, FTP, and more.
- **Structured XML Messaging:** SOAP messages have a defined structure — Envelope, Header, and Body — for clear communication.
- **Extensibility:** Supports additional standards like WS-Security (security), WS-ReliableMessaging, and WS-AtomicTransaction.

Example: A simple SOAP message might request weather information for a city.

ii) WSDL (Web Services Description Language) :

1. WSDL (Web Services Description Language) is an XML-based language used to describe the functionality of web services.
2. It defines operations, message formats, and data types, and specifies how the web service is invoked.
3. WSDL serves as a contract between the client and the service provider, enabling automatic generation of client-side code.

Feature :

- **XML-based Description:** WSDL is written in XML, making it readable by both humans and machines.
- **Describes Web Services:** It defines operations, input/output messages, data types, and service location.
- **Binding Support:** Specifies how the service is bound to communication protocols like SOAP or HTTP.

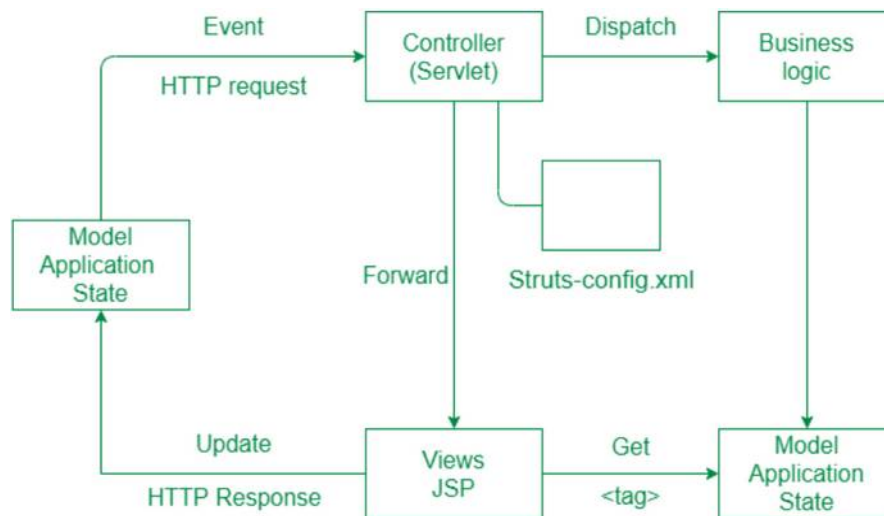
Example: A WSDL document for a weather service would define an operation for retrieving weather data, specify that the input is a city name, and describe the output as weather details.

Q4)

a) Explain struts framework with its components. Also explain interceptors.

[9]

Struts is an open-source web application framework used for building Java EE web applications. It is based on the **Model-View-Controller (MVC)** architecture, which separates the business logic, user interface, and control flow.



Struts Framework Components:

1. **Model:**
Represents the business logic and interacts with the database. It can be implemented using JavaBeans, EJB, or POJOs.
2. **View:**
The presentation layer responsible for displaying data to the user. JSP, HTML, and other front-end technologies are used here.
3. **Controller:**
Manages the flow of data between Model and View. In Struts, this is the ActionServlet, which uses a configuration file (struts-config.xml) to route requests.
4. **Action Class:**
A Java class that handles user requests, processes them, and returns the result (like success/failure).
5. **struts-config.xml:**
A configuration file that maps user requests to appropriate Action classes and defines result pages (JSPs).

Interceptors in Struts :

Interceptors are **reusable components** that execute **before and after an Action class**. They handle **cross-cutting concerns** (e.g., logging, validation, security).

How Interceptors Work?

1. **Request Received** → Interceptors pre-process the request.

2. **Action Execution** → Business logic runs.
3. **Response Sent** → Interceptors post-process the response.

Common Struts Interceptors :

Interceptor	Purpose
params	Binds request parameters to Action class fields.
validation	Validates user input using XML rules.
fileUpload	Handles file uploads.
logging	Logs action execution details.
authentication	Checks if a user is logged in.

b) Explain JSP Support for MVC (Model-View-Controller) in Web Application Development. [8]

JSP (JavaServer Pages) can be effectively used in the **MVC architecture** for building web applications by separating concerns into three layers: **Model, View, and Controller**.

1. Model

- Represents the business logic and database interaction.
- Usually implemented using JavaBeans, POJOs, or external Java classes.
- JSP does **not handle model logic directly** but can access model objects using tags or JSTL.

Example:

A JavaBean called User.java to store user data fetched from a database.

2. View

- The presentation layer is handled by **JSP files**.
- JSP displays dynamic content to users based on data provided by the Model.

- It should not contain any business logic — only display data using JSTL, EL (Expression Language), or custom tags.

Example:

welcome.jsp displays the username received from the model.

3. Controller

- Controls application flow and user input handling.
- Typically implemented using **Servlets** or frameworks like Struts/Spring MVC.
- The Controller receives requests, invokes the model, and forwards the result to a JSP view using `RequestDispatcher`.

Example:

`LoginServlet.java` processes form data, validates it, and forwards to `success.jsp` or `error.jsp`.

JSP supports MVC by cleanly separating the UI (View) from business logic (Model) and control logic (Controller), which makes the application more maintainable, reusable, and testable.

➤ NOV / DEC 2022

Q3)

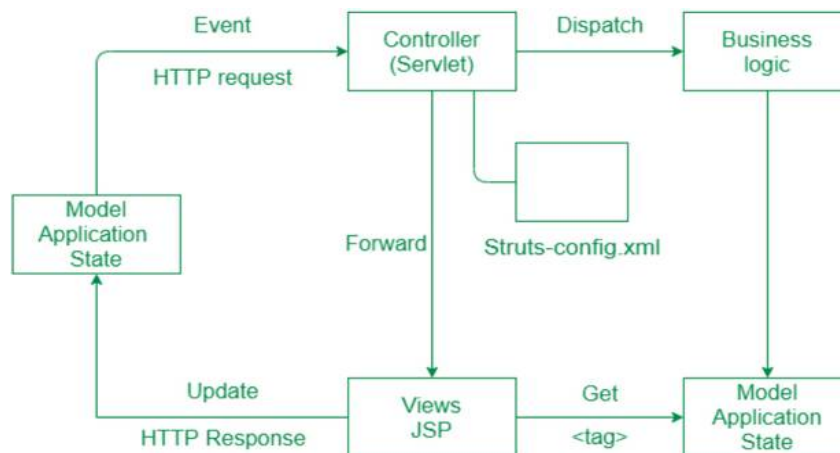
a) Write advantages of JSP over servlet and explain lifecycle of JSP. [8]

-----> Already Done !

b) Explain the Struts architecture with neat diagram and also explain the benefits of Struts. [9]

Struts Architecture

Struts is an open-source Java web application framework that follows the MVC (Model-View-Controller) design pattern. It separates the business logic, presentation, and control logic to enhance modularity and maintainability.



Components of Struts Framework:

1. Model

- Represents business logic and database interaction.
- Implemented using JavaBeans, EJBs, or POJOs.

2. View

- Presentation layer (JSP, HTML) that displays data to the user.

3. Controller (ActionServlet)

- Handles all client requests.
- Uses struts-config.xml to forward requests to Action classes.

4. Action Class

- Processes the request and returns result (success/failure) to view.

5. struts-config.xml

- Maps URLs to Action classes and defines navigation flow and result pages.

Benefits of Struts Framework:

- Based on MVC, improving separation of concerns.
- Supports form beans and input validation.
- Built-in tag libraries for faster development.
- Highly configurable using XML files.
- Simplifies development of large-scale applications.

Q4)

a) Write a JSP program to demonstrate use of page directive, Scriptlet Expression and Comment.
[9]

→ Do it by yourself !!

b) Write the benefits of Web services and explain SOAP, Rest and UDDI. [8]

Benefits of Web Services:

1. **Interoperability** – Web services allow different applications from different sources to communicate with each other without time-consuming custom coding.
2. **Platform & Language Independent** – Can be implemented in any language and run on any platform (Java, .NET, PHP, etc.).
3. **Reusability** – Once developed, web services can be reused by other applications over the network.

SOAP :

- SOAP is an XML-based protocol used for exchanging structured information in web services.
- It is platform-independent and commonly uses HTTP or SMTP for message transmission.
- SOAP messages consist of an envelope, a header, and a body, where the envelope defines the message boundaries, the header contains metadata, and the body holds the main content.

REST (Representational State Transfer):

- REST is an architectural style for designing web services.
- It uses standard HTTP methods like GET, POST, PUT, DELETE.
- Data is often transferred using JSON or XML.
- It is lightweight, faster, and easier to implement than SOAP.
- REST is commonly used in mobile and web applications.

UDDI (Universal Description, Discovery and Integration):

- UDDI is a registry that allows businesses to list themselves and their web services.
 - It provides a platform-independent way of discovering services.
 - UDDI enables service lookup based on company, service type, or technical details.
 - Works in coordination with SOAP and WSDL.
-

➤ **MAY / JUN 2023**

Q3

a) List & describe important interceptors provided by struts framework. [5]

Interceptors in Struts 2 are used to perform operations like input validation, file upload handling, and more before or after an action is executed. Below are some important built-in interceptors:

1. params Interceptor

- **Description:** This interceptor is responsible for mapping request parameters to Action class properties.
- **Functionality:** It automatically populates the Action class fields with the values from the request (form fields, query parameters).
- **Usage:** Essential for data binding between the front-end and back-end.

2. validation Interceptor

- **Description:** Performs input validation using XML-based or annotation-based validation rules.
- **Functionality:** Works with the validations.xml file or Java annotations to validate user input before executing the Action.
- **Usage:** Ensures that the submitted data meets specified constraints (e.g., required fields, email format).

3. fileUpload Interceptor

- **Description:** Handles file uploads in Struts applications.
- **Functionality:** Manages multipart requests, extracts uploaded files, and stores them in temporary storage.
- **Usage:** Used in forms with `<input type="file">` to process file submissions.

4. exception Interceptor

- **Description:** Catches and handles exceptions thrown during Action execution.
- **Functionality:** Redirects to an error page or performs custom exception handling.
- **Usage:** Useful for graceful error handling and displaying user-friendly error messages.

5. prepare Interceptor

- **Description:** Invokes the `prepare()` method in the Action class before execution.
- **Functionality:** Used to pre-load data (e.g., from a database) before the Action processes the request.

- **Usage:** Helps in initializing resources or data needed for the Action.

6. modelDriven Interceptor

- **Description:** Supports the ModelDriven pattern, allowing an Action to expose a model object directly.
- **Functionality:** Pushes the model object to the top of the ValueStack, making it accessible in forms.
- **Usage:** Useful when working with domain objects directly in forms.

These interceptors can be customized and configured in struts.xml under <interceptors> and <interceptor-ref> tags.

b) Identify & justify the benefits of using Web Services.

[5]

Web services enable communication between different applications over the internet using standardized protocols (e.g., **SOAP, REST**). Below are the key benefits of using web services:

1. Interoperability

- **Justification:** Web services use **platform-independent** standards like **XML, JSON, HTTP**, allowing different systems (Java, .NET, PHP) to communicate seamlessly.
- **Example:** A Java application can call a .NET web service without compatibility issues.

2. Reusability

- **Justification:** A single web service can be **used by multiple clients** (web, mobile, desktop apps), reducing redundant code.
- **Example:** A payment gateway web service can be reused across e-commerce platforms.

3. Loose Coupling

- **Justification:** Web services operate independently of the client's technology, ensuring that changes in one system **do not affect others**.
- **Example:** Updating a backend service doesn't require changes in mobile apps consuming it.

4. Scalability

- **Justification:** Web services can handle **high traffic** by deploying on multiple servers (load balancing).
- **Example:** Cloud-based REST APIs scale automatically based on demand.

5. Cost-Effective Integration

- **Justification:** Instead of building custom integration layers, businesses use **standardized web services**, reducing development time and costs.
- **Example:** Integrating a third-party weather API instead of developing a weather tracking system.

Additional Benefits

- **Supports Multiple Protocols** (HTTP, SMTP, JMS).
- **Enables B2B Communication** (e.g., e-commerce suppliers & retailers).
- **Enhances Security** (via **OAuth**, **SSL**, **WS-Security**).

c) Explain JSP life cycle with diagram.

[7]

----> Already done !!

Q4)

a) What is JSP? Enlist advantages of JSP over servlet?

[5]

JSP stands for JavaServer Pages. It's a server-side technology that combines HTML with Java code to create dynamic web applications.

Advantages of JSP over Servlet:

1. **Ease of Use (Less Code):**
JSP allows embedding HTML directly with Java, making it easier and cleaner than writing HTML inside Java code in Servlets.
2. **Separation of Concerns:**
JSP focuses on the presentation layer, while Servlets are better suited for business logic. This promotes MVC architecture.
3. **Faster Development:**
Since most of the UI work is in HTML with embedded tags, JSP development is faster and more intuitive for front-end design.
4. **Built-in Features:**
JSP supports implicit objects (like request, response, session) and custom tags which simplify coding.
5. **Easier Maintenance:**
UI changes can be done in JSP without modifying Java code, making maintenance simpler for designers and developers.

b) What is WSDL and SOAP? Explain WSDL in detail.

[5]

SOAP (Simple Object Access Protocol):

SOAP is a protocol used for exchanging structured information in web services using XML over HTTP or other protocols.

WSDL (Web Services Description Language):

WSDL is an XML-based language that describes the functionality offered by a web service, including its operations, input/output messages, and access details.

WSDL :

- a) WSDL (Web Services Description Language) is an XML-based language used to describe the functionality of web services.
- b) It defines operations, message formats, and data types, and specifies how the web service is invoked.
- c) WSDL serves as a contract between the client and the service provider, enabling automatic generation of client-side code.

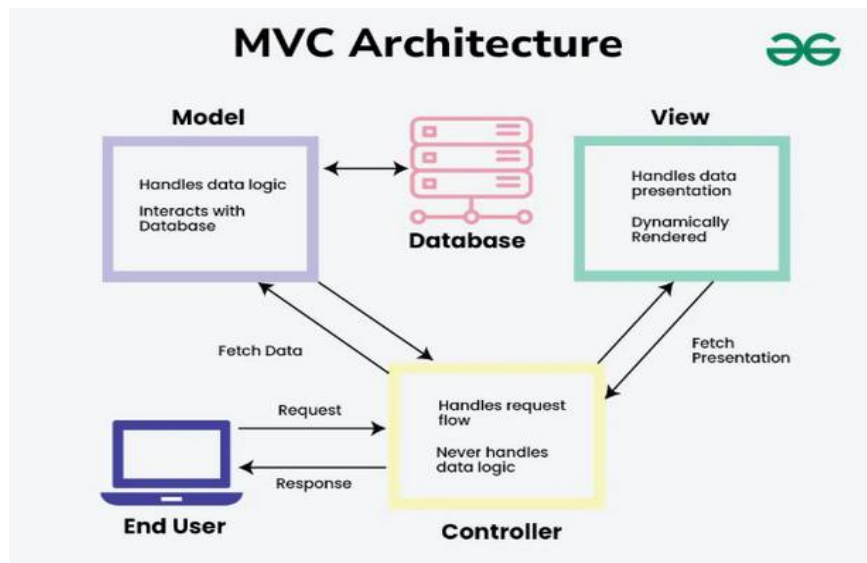
Components of WSDL:

- 1. **Types**
 - Defines the **data types** used in the service using XML Schema.
- 2. **Message**
 - Defines the **data elements** for each operation (input/output).
- 3. **PortType**
 - Defines a set of **operations (like a Java interface)** supported by the service.
- 4. **Binding**
 - Specifies the **protocol and data format** used (e.g., SOAP over HTTP).
- 5. **Service**
 - Specifies the **web service endpoint (URL)** where it can be accessed.
- 6. **Port**
 - Defines a **single endpoint** as an address for binding.

C) Draw and explain MVC architecture for developing web application.

[7]

MVC (Model-View-Controller) architecture is a universal pattern of a structure in which an application is divided into three parts which are all dedicated to certain parts of the whole application. This pattern is normally used in software development to create organized and easy-to-maintain code.



- **Model:** It is worth stating that the Model stands as the data layer for the application. It is directly involved in managing the data as well as the control of the application's logic and rules.
- **View:** The View is in the presentation tier. It plays a role of presenting the information given by the Model to the user and transferring the user commands to the Controller. The View is used to display the data to the user in a readable and manageable way using the interface created by the Controller.
- **Controller:** The Controller CE works in the middle between the Model and the View. It takes the input from the View, sometimes modifies it with the help of the Model, and sends it back to the View. the results back to the View.

Benefits of MVC Architecture :

1. **Separation of Concerns:** Divides application into Model, View, and Controller for better organization.
 2. **Reusability:** Allows reuse of business logic across multiple views.
 3. **Scalability:** Supports easy scaling and independent development of components.
 4. **Improved Testing:** Enables unit testing of individual components.
 5. **Multiple Views Support:** One model can serve multiple user interfaces.
-

➤ **NOV / DEC 2023**

Q3

- a) Explain the concept of JSP with syntax and sample example. Explain the analogy of JSP and Servlets . [9]

JavaServer Pages (JSP) is a server-side technology used to create dynamic web content by embedding Java code in HTML. It simplifies web development by separating presentation (HTML) from business logic (Java).

✓ **2. Syntax of JSP:**

```
jsp

<%@ page language="java" contentType="text/html" %>
<html>
<head><title>Sample JSP</title></head>
<body>
<%
    String name = "Himanshu";
%>
<h2>Hello, <%= name %>!</h2>
</body>
</html>
```

✓ 3. Sample Example:

Filename: `hello.jsp`

```
jsp

<%@ page language="java" contentType="text/html; charset=UTF-8" %>
<html>
<head><title>Welcome Page</title></head>
<body>
<%
    String user = request.getParameter("user");
%>
<h1>Welcome, <%= user %>!</h1>
</body>
</html>
```

- **Input URL:** `hello.jsp?user=Himanshu`
- **Output:** `Welcome, Himanshu!`

4. JSP and Servlet Analogy:

JSP

HTML with embedded Java

Easy for UI designers

Automatically compiled to servlet

Better for presentation layer

Servlet

Java code with embedded HTML (via out.println)

More suited for backend developers

Manually written Java class

Better for business logic

Analogy:

JSP is like writing content in a Word document with embedded instructions (Java), while Servlets are like writing a full instruction manual (pure Java code) to produce the output manually.

b) Explain the concept of WSDL and SOAP.

[8]

→ already done !!

Q4)

a) Explain JSP with support for Model View Controller.

[9]

→ already done !!

b) Explain the concept of Struts with architecture, actions, interceptors and exception handling. [9]

Struts Framework Overview:

Struts is an open-source framework used for developing Java EE web applications. It follows the **MVC (Model-View-Controller)** design pattern.

Architecture of Struts (with components):

1. **Model:**
Business logic, typically implemented using JavaBeans, EJB, or POJOs.
2. **View:**
JSP or HTML pages responsible for displaying data.
3. **Controller (ActionServlet):**
Handles user requests and decides which action to invoke using struts-config.xml.
4. **Action Class:**
A Java class that contains the business logic for a specific user request.
5. **struts-config.xml:**
XML file mapping user requests to actions and defining views (result pages).

Actions in Struts:

Action classes handle business logic. Each action corresponds to a user interaction (e.g., form submission).

It processes input, interacts with the model, and forwards the result to the view.

Interceptors in Struts 2:

Interceptors are objects that can intercept requests before or after an action executes.

Examples: Validation, file upload, logging.

They allow preprocessing/postprocessing without changing the action code.

Exception Handling in Struts:

Struts provides centralized exception handling using exception elements in struts-config.xml.

You can define custom error pages for specific exceptions using `<global-exceptions>` tag.

➤ **MAY / JUN 2024**

Q3)

a) Explain the JSP support for MVC paradigm.

[8]

→ Already done !!

b) Explain struts framework with respect to architecture, actions, interceptors & exception handling

[9]

→ Already done !!

Q4)

a) Explain JSP lifecycle. Differentiate JSP Vs Servlet. (Min.04).

[9]

JSP Life cycle ----> already done !!

Feature	JSP	Servlet
Type	Mainly for presentation layer	Mainly for processing logic
Code Style	HTML with Java (less complex)	Pure Java (more complex)
Translation	Translated into servlet by server	Written directly in Java
Ease of Use	Easier for designing UI	Suitable for processing & logic
Modification	Simple to modify UI	More effort to change presentation

b) Explain the concept of web services. Explain in brief WSDL & SOAP.

[8]

i) Web services are standardized ways of integrating web-based applications using open standards such as XML, SOAP, WSDL, and UDDI over an internet protocol backbone.

ii) They allow different applications to communicate and share data and services among themselves, regardless of the platform or language used.

i) SOAP (Simple Object Access Protocol) :

4. SOAP is an XML-based protocol used for exchanging structured information in web services.
5. It is platform-independent and commonly uses HTTP or SMTP for message transmission.
6. SOAP messages consist of an envelope, a header, and a body, where the envelope defines the message boundaries, the header contains metadata, and the body holds the main content.

Feature :

- **Platform & Language Independent:** SOAP uses XML, making it compatible across different operating systems and programming languages.
- **Protocol Agnostic:** Although commonly used with HTTP, SOAP can also work over SMTP, FTP, and more.
- **Structured XML Messaging:** SOAP messages have a defined structure — Envelope, Header, and Body — for clear communication.
- **Extensibility:** Supports additional standards like WS-Security (security), WS-ReliableMessaging, and WS-AtomicTransaction.

Example: A simple SOAP message might request weather information for a city.

ii) WSDL (Web Services Description Language) :

4. WSDL (Web Services Description Language) is an XML-based language used to describe the functionality of web services.
5. It defines operations, message formats, and data types, and specifies how the web service is invoked.
6. WSDL serves as a contract between the client and the service provider, enabling automatic generation of client-side code.

Feature :

- **XML-based Description:** WSDL is written in XML, making it readable by both humans and machines.
- **Describes Web Services:** It defines operations, input/output messages, data types, and service location.
- **Binding Support:** Specifies how the service is bound to communication protocols like SOAP or HTTP.

Example: A WSDL document for a weather service would define an operation for retrieving weather data, specify that the input is a city name, and describe the output as weather details.

➤ **NOV / DEC 2024**

Q3

a) Explain JavaBeans classes & JSP.

[8]

JavaBeans Classes:

JavaBeans are reusable components in Java that follow a specific design pattern. They encapsulate data in fields (variables) and provide public getter and setter methods to access and modify these fields. JavaBeans are commonly used in JSP applications to manage and share data between various components.

Key Features of JavaBeans:

1. **Serializable:** Implements `java.io.Serializable` so the bean can be persisted.
2. **No-argument constructor:** Must have a public no-arg constructor.
3. **Getter and Setter methods:** To access properties using methods like `getName()` and `setName()`.

JSP and JavaBeans:

JSP (JavaServer Pages) supports JavaBeans to follow the MVC pattern by separating business logic from presentation logic. Beans manage data (Model), JSP handles the view (HTML, CSS), and servlets often control the flow.

JSP Standard Actions for JavaBeans:

- `<jsp:useBean>` – Declares a JavaBean and optionally instantiates it.
- `<jsp:setProperty>` – Sets the property of the bean.
- `<jsp:getProperty>` – Retrieves the property of the bean.

Example in JSP:

jsp

Copy

Edit

```
<jsp:useBean id="user" class="com.example.UserBean" scope="session"/>
<jsp:setProperty name="user" property="name" value="Himanshu"/>
Hello, <jsp:getProperty name="user" property="name"/>
```

Advantages of using JavaBeans in JSP:

- Promotes code reusability.
- Clean separation of concerns.
- Simplifies data sharing across multiple JSPs.
- Reduces Java code in JSP, keeping presentation logic clean.

b) Already done !!

Q4

a) JSP vs Servlets (Min.04). Explain how JSPs are translated to Servlets.

[8]

Feature	JSP (JavaServer Pages)	Servlet
---------	------------------------	---------

Type	Used for presentation layer (view)	Used for business logic (controller)
Code	Mostly HTML with embedded Java code	Entirely Java code
Ease of development	Easier for UI design	More complex, better for processing logic
Modification	Changes in UI are easier to handle	Changes need re-compilation
Compilation	Compiled internally to servlet by container	Must be compiled manually before deployment

How JSP is Translated to Servlet:

1. When a JSP is accessed for the first time, the server converts it into an equivalent Java Servlet class.
2. This servlet is then compiled into a .class file.
3. The generated servlet handles the HTTP request/response lifecycle.

Example:

```

jsp
Copy Edit

<!-- index.jsp -->
<html>
<body>
<%= "Hello Himanshu!" %>
</body>
</html>

```

This is internally converted to:

```

java
Copy Edit

public class index_jsp extends HttpJspBase {
    public void _jspService(HttpServletRequest req, HttpServletResponse res) {
        res.getWriter().write("<html><body>");
        res.getWriter().write("Hello Himanshu!");
        res.getWriter().write("</body></html>");
    }
}

```

Q4

b) Write a note on Struts. Explain Struts with example code block. [9]

Struts Framework Overview:

Apache Struts is an MVC-based Java framework used for building scalable, maintainable web applications. It separates the application into:

- Model (business logic)
- View (JSP/UI)
- Controller (ActionServlet)

Key Components:

- ActionServlet: Controller that routes requests.
- struts-config.xml: Maps URLs to Action classes.
- Action class: Contains logic to process user requests.
- FormBean: Stores user input.
- JSP: Used for front-end UI (View).

Simple Flow:

1. User submits a form.
2. Request goes to ActionServlet.
3. Servlet consults struts-config.xml.
4. Action class executes business logic.
5. JSP displays result.

Code Example:**1. Form.jsp**

```
jsp Copy Edit  
  
<form action="login.do" method="post">  
  Username: <input type="text" name="username"/>  
  <input type="submit" value="Login"/>  
</form>
```

2. LoginAction.java

```
java Copy Edit  
  
public class LoginAction extends Action {  
  public ActionForward execute(ActionMapping mapping, ActionForm form,  
    HttpServletRequest request, HttpServletResponse response) {  
    // business logic here  
    return mapping.findForward("success");  
  }  
}
```

3. struts-config.xml

```
xml Copy Edit  
  
<action path="/login" type="com.example.LoginAction"  
  name="loginForm" scope="request" input="/login.jsp">  
  <forward name="success" path="/welcome.jsp"/>  
</action>
```