

JSP – introduction, Directives , Tags and Comments

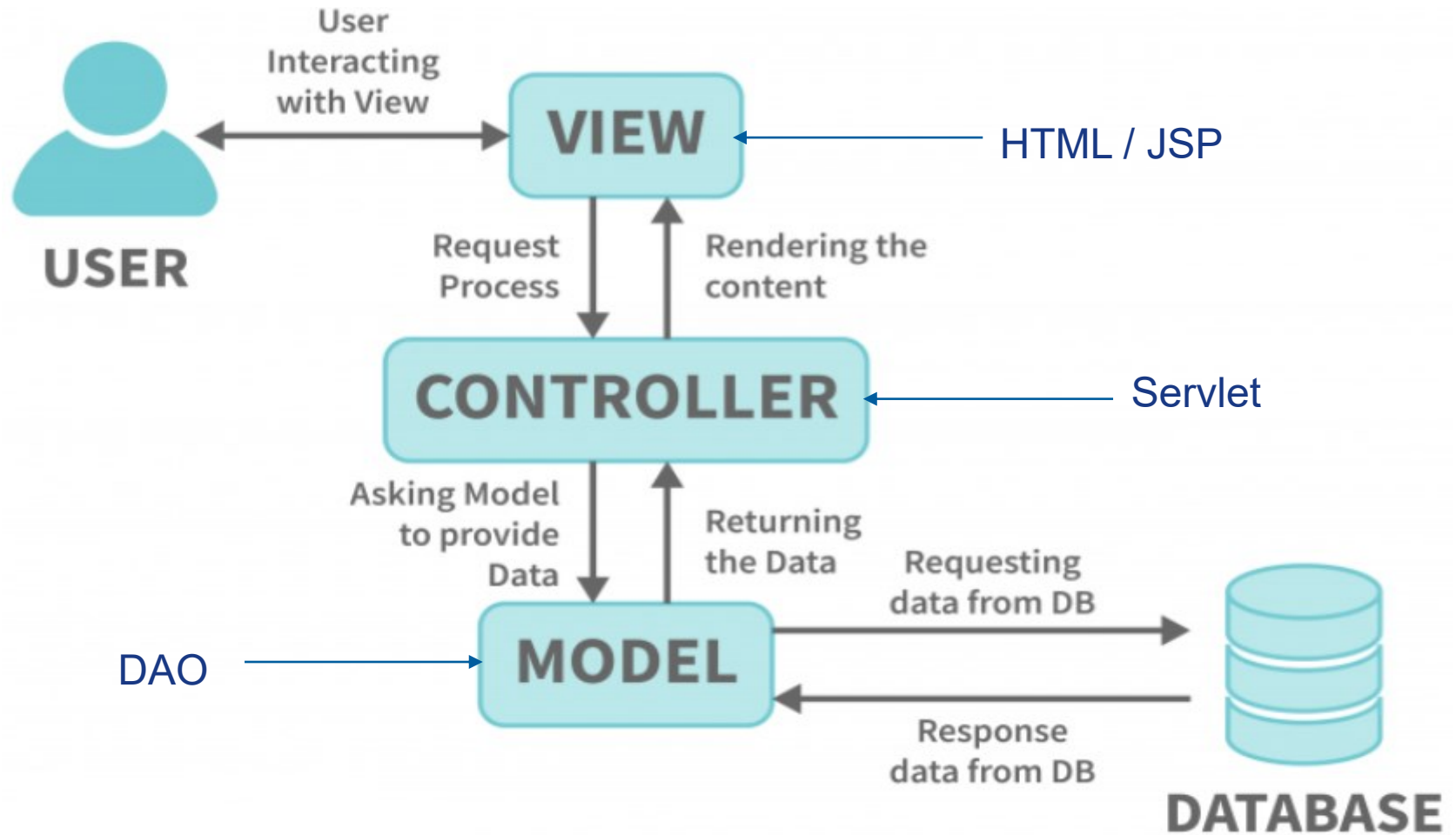
Presented by

MVC Architecture

- A Design Pattern
- Model -- represents state of the application e.g. form data ,database
- View -- looks at model and generates an appropriate user interface to present the data
- Controller -- receives user interface input, updates data model
- Ex:
 - Model – Java Beans / POJOs /DAO layer classes
 - View – HTML / JSP pages
 - Controller - Servlets



MVC Architecture

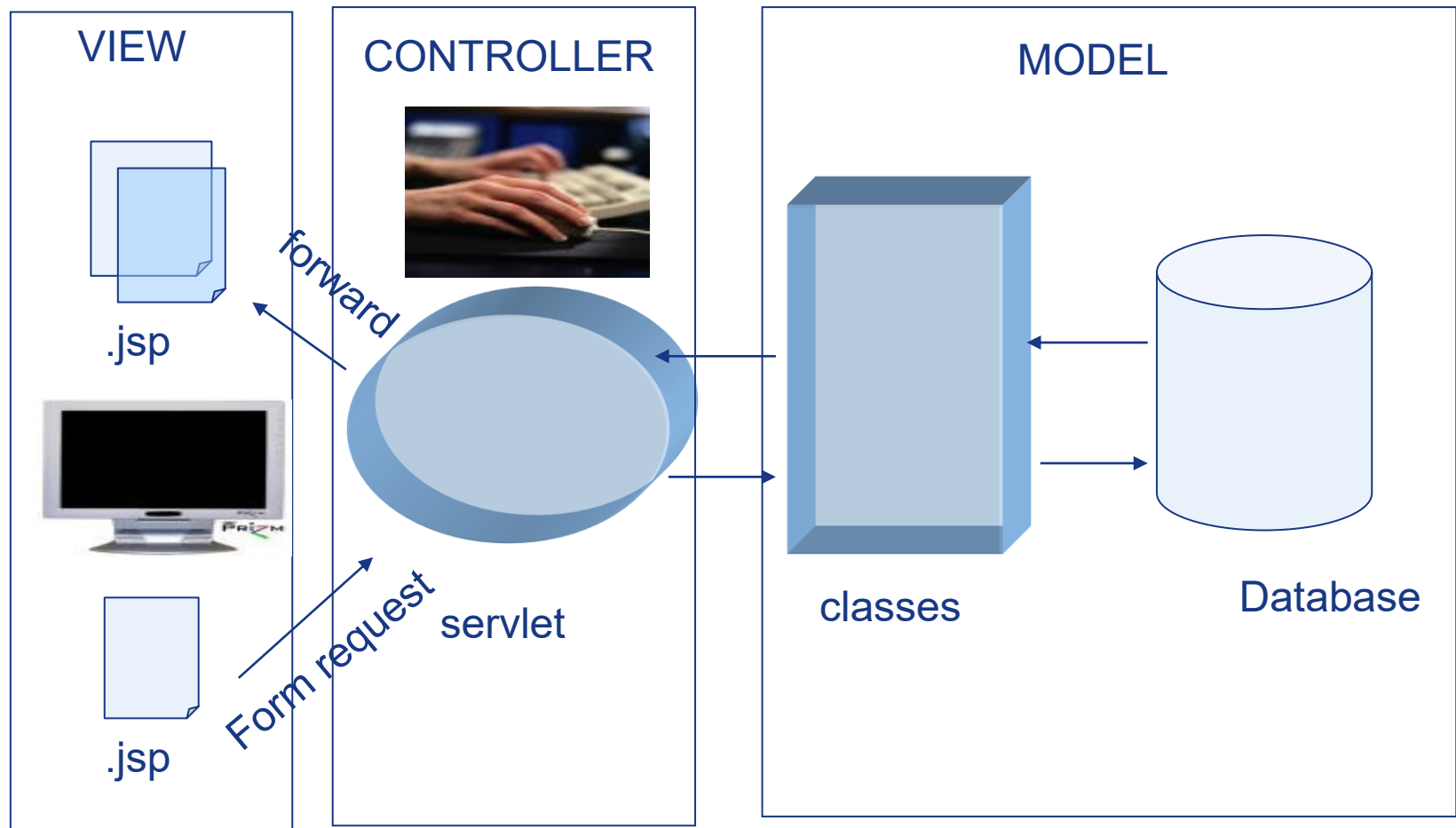


What is JSP?

- Mostly HTML page, with extension .jsp
- Include JSP tags to enable dynamic content creation
- Dynamic Content: The information generated and displayed at runtime is called 'dynamic content'. JSP uses Java to display dynamic content
- Translation: JSP → Servlet class
- Compiled at Request time
(first request, a little slow)
- Execution: Request → JSP Servlet's service method



MVC - Flow



JSP Elements – Directives - @page

Directives: <%@ %>

- Provide global information to the page
- Import statements

1. @page :

Provides information about the page
Imports the required API

```
<%@page contentType= "text/html" pageEncoding= "UTF-8"%>  
<%@ page language = "java"%>  
<%@ page language= "java" import= "java.sql. *" %>
```



JSP - @include

2. @include :

- Includes another resource (jsp / html / servlet) into the current page

```
<%@include file = "sample.html"%>
```



JSP Elements – Tags

1. Declarations: `<%! ...%>`

- For page-wide variable and method declarations

Example:

```
<%! int num;%>
```

2. Scriptlet : `<% ...%>`

- To embed Java code in the web page

Example:

```
<%  
    for (int num=5; num<5; num++) {  
        out.println(num);  
    }  
%>
```



JSP – Tags ...

3. Expressions: `<%= ... %>`

- Formats expression values into string in the output of the web page

Example:

```
<%= 1 + 1 %>
```

```
<%= new java.util.Date() %>
```

4. Comments : `<%-- ... --%>`

- To add user documented comments
Will not be parsed by the JSP compiler

Example:

```
<%-- Calculate the sum of 1 + 2 + 3 dynamically --%>
```

****Example:**



JSP – Implicit Objects



JSP – out, request, response

out :

- It is the most used implicit object that comes from `java.servlet.jsp.JspWriter`.
- It's work is to write data into a buffer which would be sent to the client as output.

Ex: `out.println("Hello, Guest!");`

request:

- This is an in-built implicit object of `javax.servlet.http.HttpServletRequest`.
- It is passed as a parameter to `jspService` method.
- This instance is created each time a request is generated.

Ex: `String name=request.getParameter("username");`

response :

- This is an object from `HttpServletResponse`.
- This object rather gets passed as a parameter to `jspService()` like request.
- It is the response given to the user.

Ex: `response.sendRedirect("http://www.google.com");`



JSP – session,config, pageContext

session :

session object is the implicit object under javax.servlet.http.HttpSession . It is used to get session information as it tracks sessions between client requests.

Ex: `session.setAttribute("user",user);`

config:

It is an object of javax.servlet.ServletConfig. It is used to get the information regarding configuration of JSP pages.

Ex: `String driver=config.getInitParameter("username");`

pageContext:

It is the implicit object from javax.servlet.jsp.PageContext implementation where pageContext is an abstract implementation.

Ex: `pageContext.setAttribute("client",myname,pageContext.SESSION_SCOPE);`



JSP – application, exception

application :

- It is an implicit object of ServletContext from javax.servlet.ServletContext implementation.
- This object is created by the web container only when an application is deployed.
- It is generated one per application.
 - Ex: `application.setAttribute("technology","Java FSD");`

Exception

- Exception is an implicit built-in object of java.lang.Throwable.
- This is useful for handling exceptions in JSP.
- As these pages handle errors, they can only be used for JSP error pages.
- This implies that if `<%@ page isErrorPage="true"%>`, then only you can use this object.



Thank You

