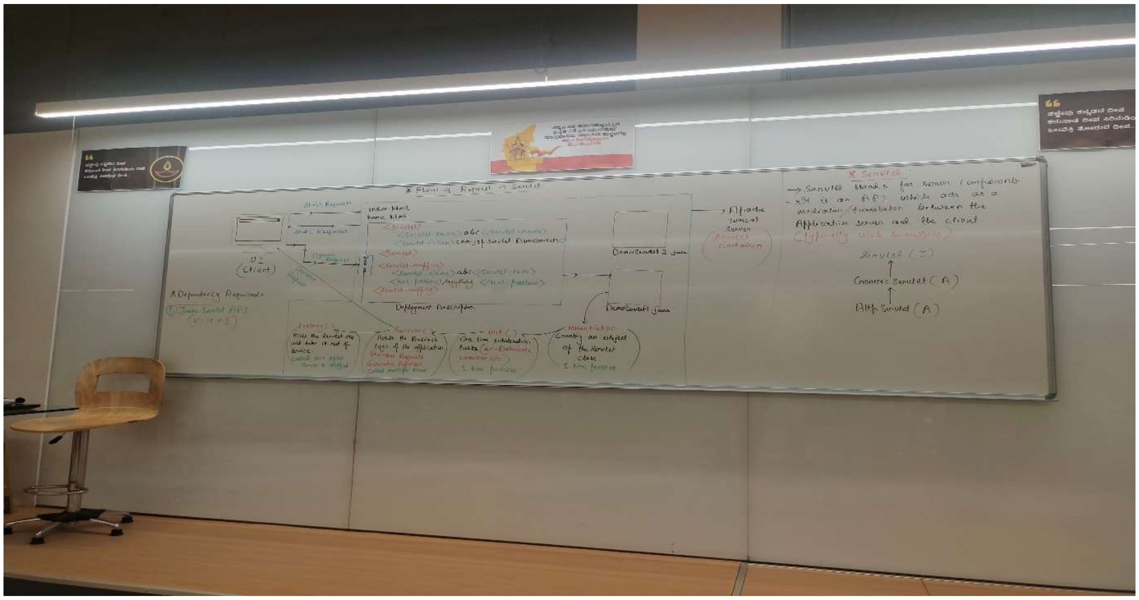

* Flow of Request in Servlet
Static Request
Static Response
Dynamic Request
UI
(Client)
* Dependency Required
1) Java Servlet API
(V - 4.0.1)
<servlet>
<servlet-name>abc</servlet-name>
<servlet-class>com.jsp.servlet.DemoServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>abc</servlet-name>
<url-pattern>/anything</url-pattern>
</servlet-mapping>
Deployment Descriptor
DemoServlet2.java
DemoServlet1.java
Apache
tomcat
server
Servlet
Container
destroy()
service()
init()
Instantiation
Creating an object of the Servlet class
1 time process
* Servlet
→ Servlet stands for server components.
→ It is an API which acts as a mediator/translator between the Application server and the client
Servlet (I)
Generic Servlet (A)
Http Servlet (A)

Servlet:-

- Servlet stands for server components.
- Servlet is an API which helps to interact between the browser and the application server.
- We can create a servlet by implementing the Servlet (I) interface er extending either of the two abstract class(GenericServlet(A) & HttpServlet(A)).

Servlet Interface:-

- It is an interface present inside javax.servlet package which is used to develop server side programming application.
- The as 5 unimplemented methods which define the lifecycle of a servlets.
  1. init() method:- This method is called by the servlet container to indicate that the servlet is being placed into service.
     It is used for one-time initialization tasks, such as loading configuration parameters or establishing database connections.
  2. service(ServletRequest req, ServletResponse res):- This method is called by the servlet container to handle each client request.
     It is the core method where developers Implement the logic to process Incoming requests and generate appropriate responses.
     Incoming requests and generate appropriate responses.
     The ServletRequest and ServletResponse parameters provide access to the client's request and the server's response, respectively.

  3. destroy():- This method is called by the servlet container to indicate that the servlet is being taken out of service.

Once this method is called, the servlet instance is no longer available for handling requests.

4. getServletConfig():- This method returns a ServletConfig object that contains the servlet's configuration information.

5. getServletInfo():- This method returns string containing Information about the servlet, such as its name and version.

GenericServlet (A)-

- It is an Abstract class present in javax.servlet package.
- It Implements the Servlet Interface and provides implementation to all the abstract methods

Configuration of a Static request:-

1. Whenever a request for a static content is given from the UI, the request enters the server checks for the for the requested file inside the Application deployed inside the server.

2. Only thing to take care is the server is in running state and the application where the requested file is present must be deployed into the server.

Configuration of a Dynamic request:-

1. Whenever a request for dynamic content is given from the UI, the request enters the server and checks for the web.xml file which is also known as the Deployment Descriptor.

2. In web.xml file it checks if there is a <servlet-mapping> tag with the requested "<url-pattern>".

3. Once the <url-pattern> matches then it reads the servlet-name given for the specific <servlet-mapping> tag

4. It carries the <servlet-name> and checks for a <servlet> tag with the same <servlet-name>.

5. If a <servlet> tag is found with the same name then it gets the information of the <servlet-class> from the <servlet> tag.

6. After this process the instantiation of the servlet is done and further code will be executed.

Servlet Requests-

- ServletRequest is an interface in the Java Servlet API present in javax.servlet package
- It provides away for servlets to read data from a client's request.
- It is implemented by the servlet container to encapsulate the details of as HTTP request sent by a client to a servlet.

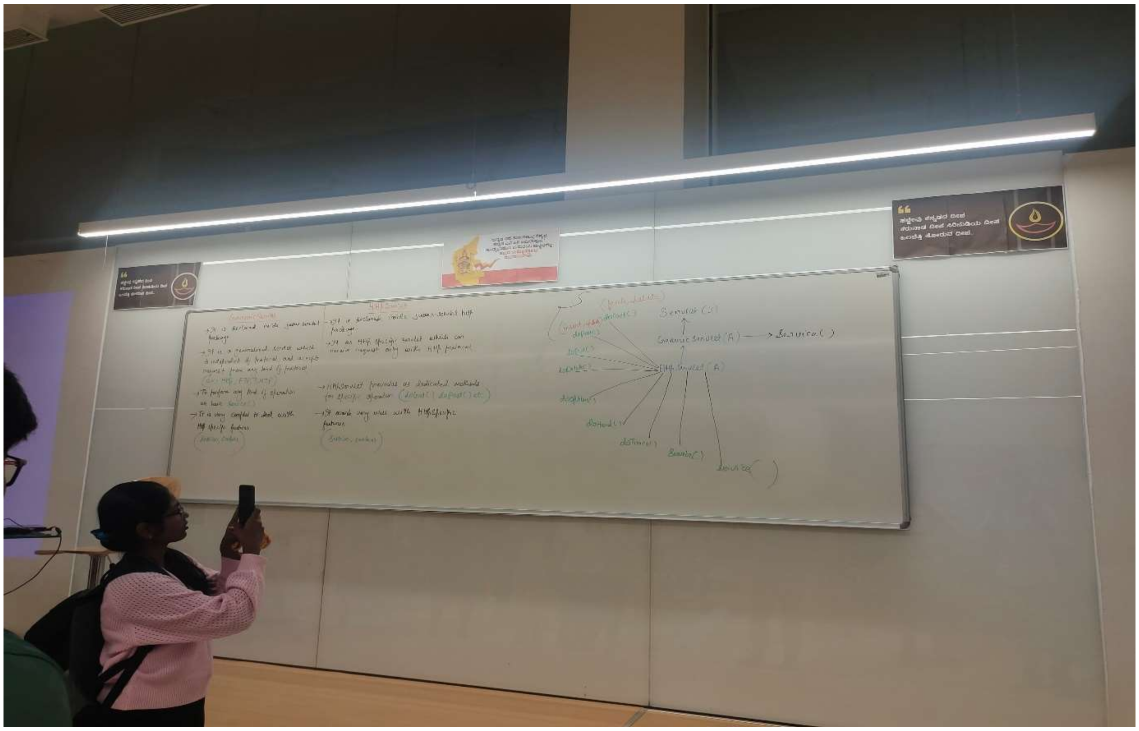
- The ServletRequest interface provides methods to obtain information about the parameters and other attributes of the client's request.

ServletResponse-

- It is an interface in the Java Servlet API present in javax.servlet package.
- It defines the methods that a servlet can use to send a response back to the client.
- It is implemented by the servlet container to provide an abstraction for handling responses In a servlet.

PrintWriter:-

- PrintWriter is a java class in the java.io. package that provides methods for printing formatted text to a file.
- ln the context servlets and we development, the PrintWriter class is frequently used to send text data (such as HTML content) as the response from a servlet to the client's browser.



JSP:-

- It stands for Java Server pages
- It is a type of view technology file which is similar to as that of HTML but provides some extra features to the user in perspective of java.
- It provides some specific tags using which we can write the java logic inside a JSP file.
- JSP allows for dynamic content generation by embedding Jave code within HTML-like tags.
- It understands java objects and allows the user to import java classes inside a JSP file.
- It can include another view technology file in a current JSP file.

Tag in JSP:-

1. Directive tag – It is used to add some external resources to the current JSP file (Import statements for Java, any other view technology file etc).
2. Declarative tag - This tag is used to declare Jave components inside the JSP file.
3. Expression tag – This tag is used as a print statement in JSP to access the variables declared.
4. Scriptlet tag - It is used to write the hard code java logic typically (looping statements, conditional statements, etc).

HttpServlet (A):-

- It is specialization of GenericServlet specifically designed to handle HTTP requests.
- It extends GenericServlet and provides additional methods specifically for handling HTTP requests such as doGet, doPost, doPut, doDelete, etc.
- Use HttpServlet when you are specifically building a servlet for handling HTTP requests.
- It provides convenience methods for handling common HTTP operations, making it easier to work with HTTP-specific features.

RequestDispatcher:-

- It is an Interface present in Javax.servlet package.
- It is used to forward or Include responses to other servlets, a JSP or an HTML file.
- It is a part of servlet API which allows servlets to collaborate in processing client requests.
- When you forward a request using RequestDispatcher, the original request URL remains unchanged in the client’s browser.
- Overall RequestDispatcher provides a way to modularize and organize code in servlet-based web applications.
- It has two methods using which we can forward or include the respose:
  1. forward(req, resp):-

- This is useful in cases where you want to perform some processing in one servlet and then hand over the request to another servlet for further processing or response generation.
- The control will be given back to the servlet again, but the req and resp object will no longer be available to use in the current servlet.

sendRedirect():-

- In servlets sendRedirect() is a method provided by the HttpServletResponse interfece which is used to redirect the client's browser to a different resource(such as another servlet, a JSP File, or an HTML file) on the server or even to an external URL.
- Unlike RequestDispatcher, sendRedirect() causes the client's browser to change the URL visible in the address bar.
- This method is typically used when you want to redirect the user to a different page or resource after some processing has been done, or when you want to send the user to a different part of the application based on certain conditions.

@WebServlet:-

- It is an annotation from javax.servlet.annotations package:
- It is used to declare and configure a servlet directly in the java code without needing to configure it in web. xml file (Deployment Descriptor).
- it simplifies the configuration process.

Lifecycle of a Servlet:

1. Loading and Instantiation

- The servlet container loads the servlet class Into memory when it receives the first
- request for the servlet or during server startup.
- Once loaded, the container Instantiates the servlet object.
- This is an one time process.

2. initialization (init() Method)

- The container invokes the servlet's init() method only once after instantiation.
- This is where the servlet initializes resources like database connections or configuration parameters.
- It is also an one time process.

3. Request Handling (service() Method)

- After Initialization, the container calls the service() method for each client request.

- The service() method determines the request type (eg, GET, POST) and calls the appropriate method (doGet(), doPost(), etc.).

4. Destruction(destroy() method)

- when the servlet is no longer needed (eg, the server is shutting down), the servlet container Invokes the destroy() method.
- Once the destroy() method is Invoked the servlet object will be killed and will no longer be available to handle requests.