Lec-05

"Java servlet"

Servlet: servlet is a technology is used to create web application (resides at server side and generates a dynamic webpage)

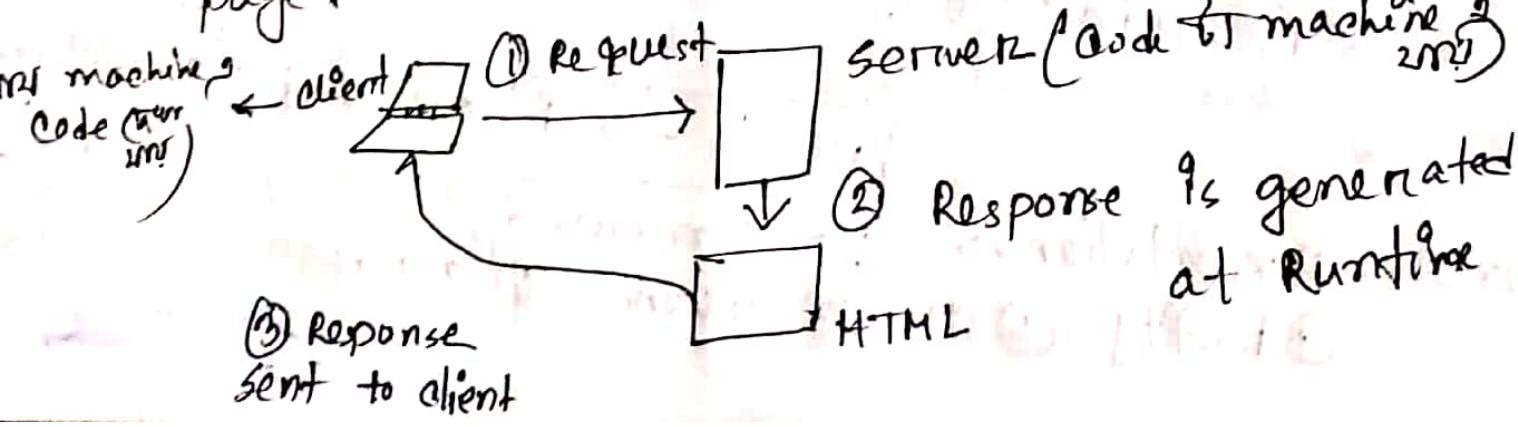
static web page/website
dynamic " "/web application

What is servlet?

→ Is an API that provides many Interface & classes including documantation.

→ Is an interface that must be implemented for creating any servlet.

→ Is a web component that is deployed on the server to create a dynamic web page.



(A) (1) Download and install XAMP

(2) Open & start Apache

(3) NOW you are a 'server'...

Apache Default part ; go

(B) What is a web application:

A who application is an application accessible from web.

A web application is composed of web component like servlet, JSP, filter etc and other element such as HTML, CSS, javascript. The web component typically execute in web server and respond to the HTTP request.

(C) Dynamic webpage created by

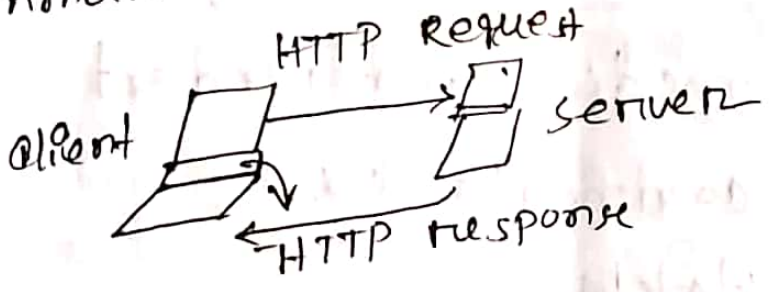
(1) Java
(2) php
(3) C#
(4) Python.

Static Vs Dynamic website :

Static	Dynamic
① Prebuilt content is same every time the page is loaded.	① Content is generated quickly and changes regularly.
② It was the HTML code for developing a website.	② It uses the server side language such as Java, PHP, ASP etc for developing a website.
③ It sends exactly the same response for every request.	③ It may generate Different HTML for each of the request.
④ The content only changed when someone publishes and update the file (HTML)	④ The page contains "serverside" code which allows the server to generate the unique content when is loaded.
⑤ Flexibility is the main advantage of static website	⑤ Content management system is the main advantage of Dynamic website

(A) HTTP (Hypertext transfer protocol):

The hypertext transfer protocol is application level protocol for collaborative, distributed, hypermedia information system. It is the data communication protocol used to establish communication between client server.



II Basic characteristics of HTTP:

(1) It is the protocol that allows web servers
(2) and browsers to exchange data over the web.
(2) It is a request response protocol.
(3) It was the reliable TCP connections by default on tcp port 80

(18)

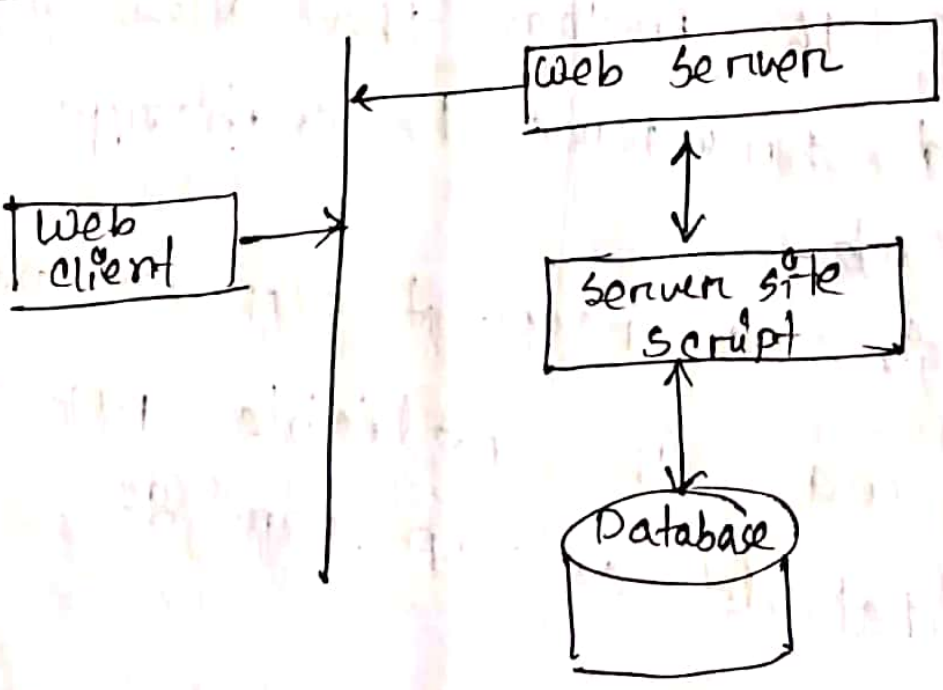
(4) Why HTTP is stateless protocol?

(4) If is stateless means each request is considered as the new request. In other words, server doesn't recognize the user by default.

Basic features of HTTP:

1. HTTP is media independent.
2. HTTP is connectionless.
3. HTTP is stateless.

Basic characteristic of web applications:



HTTP protocol

Ans

HTTP client sends the request to the server in the form of request message which includes the following:

1. The request-line.
2. The analysis of source IP address, proxy and port
3. The analysis of destination IP address, proxy
4. The requested URI (Uniform Resource Identifier)
5. The request method and content.
6. The User-Agent header.
7. The connection control header.
8. The cache control header.

Ans

HTTP request methods are:

- GET
- POST
- PUT
- HEAD
- TRACE
- DELETE
- OPTIONS

(A) XAMP file এ ঢুকো —

c:// xampp// htdocs → create your app main folder

↓ inside this folder create your html file

↓ That's it, you are Done.

Lec-06

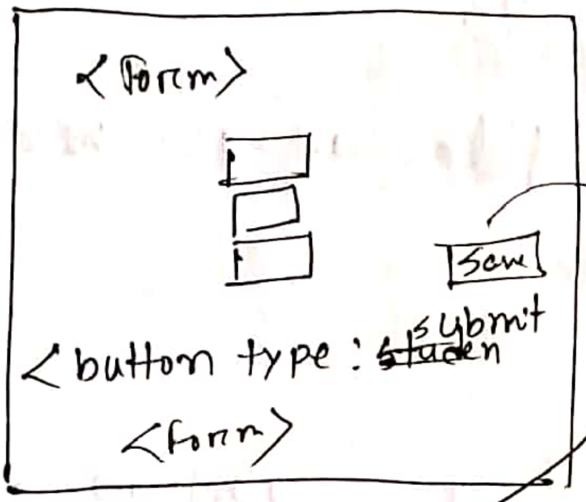
"GET Vs POST"

① Only Limited amount of data can be sent because data is sent as 'header'.	① Large amount of data can be sent because data is sent in 'body'.
② GET request is not secured because data is exposed in URL bar.	② Post request is secured because data is not exposed in URL bar
③ Get request can be bookmarked.	③ Post request can not be bookmarked.
④ GET request is independent. It means second request will be ignored until response of first request is delivered.	⑤ Post request is non-independent.
⑥ GET-request is more efficient and used more than post.	⑥ Post request is less efficient and used less than GET.

http://test/index.html?name=value → GET request → ২টা way যে data পাঠানোর জন্য

main URL

Query parameter



http:// locatted /

HTTP - POST - Request → 1টা way

Data পাঠানোর জন্য use হয়

http: Localtest /test/deb

↳ Path parameter

(*) আমরা GET request হলে URL bar দিয়ে
browser "
post " " Console দিয়ে

Anatomy of GET Request :

http: localhost/ Register JSP? user= Rone & id = 100 HTTP //1.1

Host: www.javatpoint.com
User-Agent: Mozilla/

w3school bootstrap.com

(4) Bootstrap.css library

(1) Download Bootstrap.css

(2) " " .js

(3) Import this two files at the header of your html file.

03.11.19

Lec-07

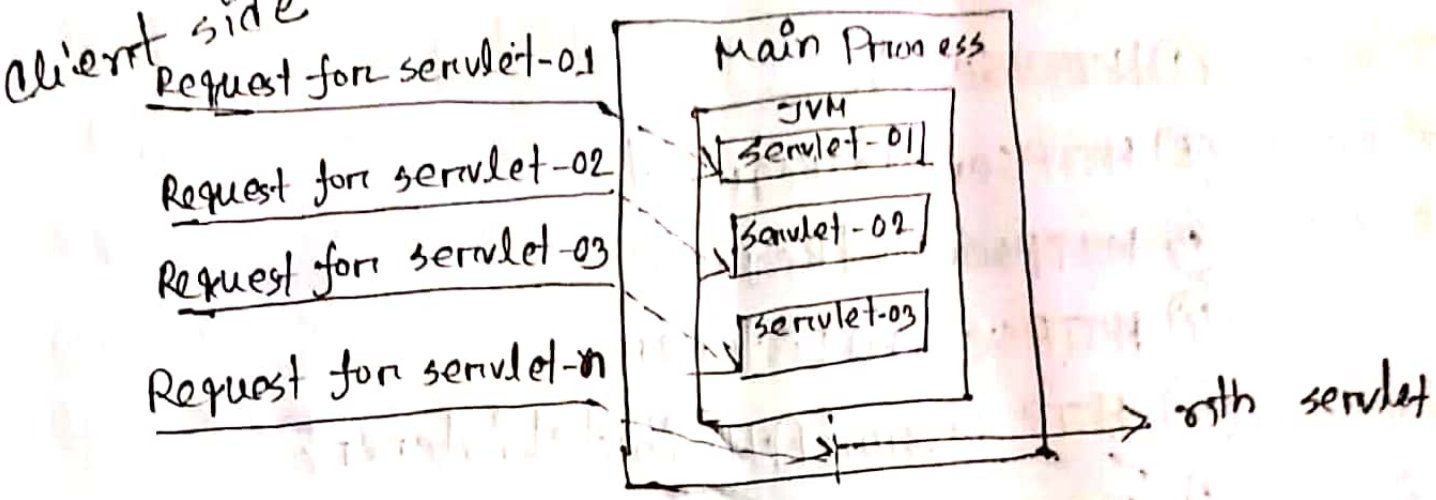
Web server:

It is also known as HTTP server. It can handle HTTP requests send by client and responds the requests with an HTTP response.

Web containers:

Also known as servlet container and servlet engine. It is a part of web server that interacts with servlets. This is the main component of web server that manages the life cycle of servlets.

[1] How servlet works?



[1] Interface in Java servlet package:

1. servlet
2. ServletRequest
3. ServletResponse
4. ServletConfig
5. ServletContext
6. SingleThreadModel
7. RequestDispatcher
8. servlet Request Listener
9. servlet Request Attribute Listener
10. Servlet Context Listener
11. servlet Context Attribute Listener
12. filter
13. FilterConfig
14. Filterchain.

Interfaces in javax.servlet.http package:

1. HTTPSession
2. HTTPServletRequest
3. HTTPServletResponse.
4. HTTPSessionListener
5. HTTPSessionAttributeListener.
6. HTTPSessionBindingListener.
7. HTTPSessionActivationListener.
8. HTTPSessionContext.

next

servlet life cycle:

① servlet class is loaded: The class loader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by web container.

② servlet Instance is created: The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.

③ Init method is invoked: The web container calls the init method only once after creating the servlet instance. The init method is used to initialize the servlet. IT is the life cycle method of the javax.servlet.servlet interface. Syntax of the Init method is:

```
Public void init (servletConfig Config) Throws Servlet
                                          Exception {}
```

④ Service method is invoked: The web container calls the service method each time when request for the servlet is received. If servlet is not initialized, it follows the first three(3) steps as described above then calls the service method. If servlet is initialized, it calls the service method. Notice that servlet is initialized only only once. Code is:

```
Public void service (servletRequest request,
        servletResponse response) Throws
        ServletException, IOException {
                                    }
```

(5) Destroy method is invoked: The web container call the destroy method before removing the servlet instance from the service it gives the opportunity to clean up any resource for example: memory, threads etc. Code is: Public void destroy();

Project Must have the following features libs:

1. Repository - GIT/Github
2. servlet - Java
3. Database: MySQL
4. Server: Apache Tomcat
5. Css Lib: Bootstrap
6. Email: API
7. SMS - API
8. Reporting Module
9. Forgot password Module
10. Email verification to active an user.
11. Front-end field validation
12. ~~fbay~~.
13. DataTable for every table view (Sorting, Searching, pagination.

4. Highchart ~~for~~ front end Report

Lab-03

(1) How to send SMS-API from Java

Step-01: We have choose greenweb vendor

Step-02: create an account to

Step-03: Read API Document.

Lec-08

Servlet project

```
public class ExampleHttpServlet extends HttpServlet
(1) @Override
    public void init() throws ServletException{
    }
(2) @Override
    public void doGet(HttpServletRequest request,
    HttpServletResponse response) throws ServletException,
    IOException{
    }
(3) @Override
    public void destroy(){
    }
```

Web.xml

<web-app>

<servlet>
<servlet-name>ExampleHttpServlet</servlet-name>
<servlet-class>ExampleHttpServlet</servlet-class>
xyz

</servlet>
<servlet-mapping>
<servlet-name>ExampleHttpServlet</servlet-name>
same
<servlet-pattern>/welcome</url pattern>
xyz

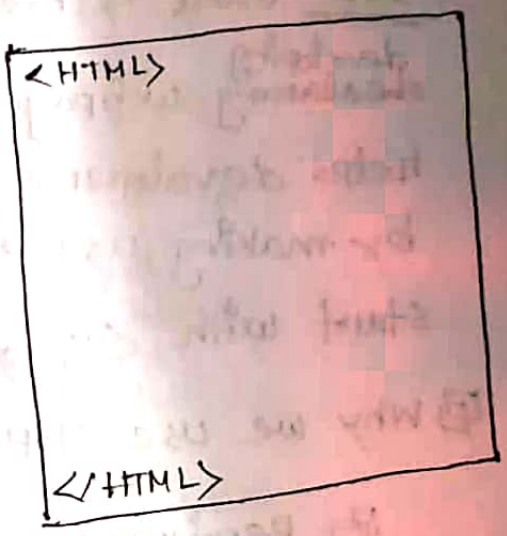
</servlet-mapping>

</web-app>

1 class 1 servlet

RequestDispatcher dispatcher = request.getRequestDispatcher("hello.html");
dispatcher.forward(request, response);
jsp

}



1. git fetch
2. git checked -b branch name.
3. git clone . projectUrl.
4. git rebase origin/branch name.
5. git pull.
6. git ~~push~~ push.
7. git commit.

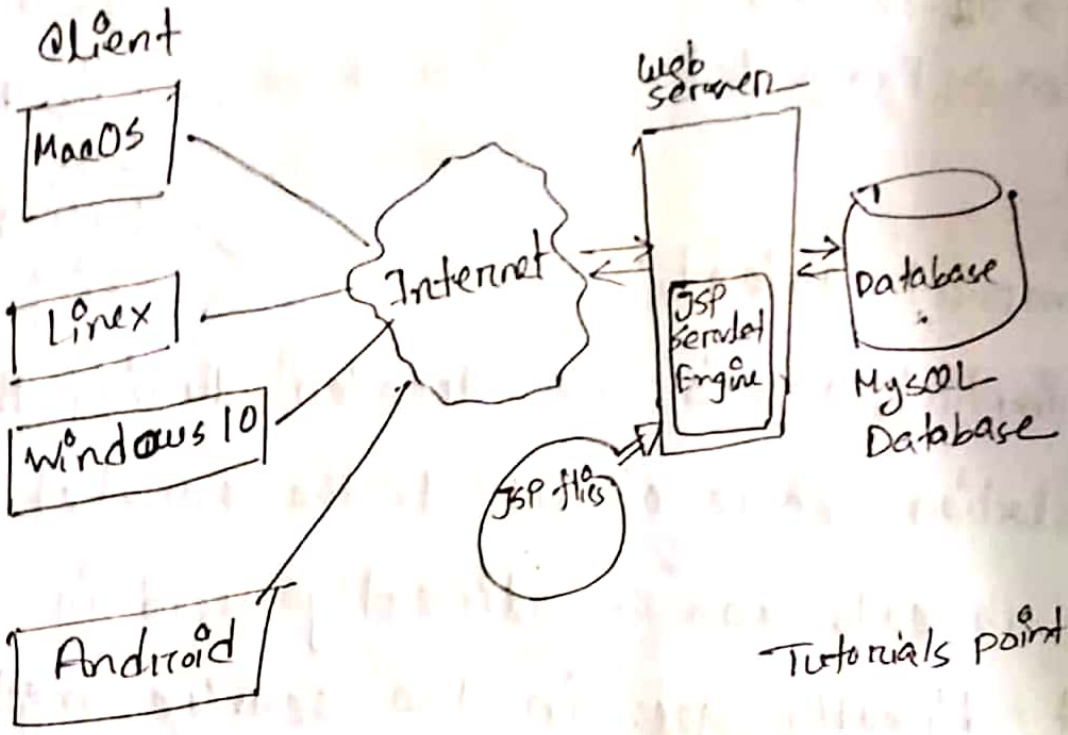
Lec-09

JSP: Java server ~~being~~ page is a technology for developing ~~declaring~~ webpages that supports dynamic content this helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>

Why we use JSP:

1. performance is significally better because JSP allows dynamic elements in HTML page itself instead of having separate CGI files.
2. JSP are always compiled before they are processed by the server.
3. JSP pages are built on top of the java servlet API so like servlets, JSP also has access to all the powerfull enterprize java APIs, including JDBC, JNDI, EJB, JAM etc.
4. JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engine.

JSP Architecture:



Tutorials point.

1. JSP lifecycle
2. JSP architecture

elements of JSP:

1. The Scriptlet <%... %>
2. JSP declaration
3. JSP expressions
4. JSP commants
5. JSP directives
6. JSP Action
7. JSP implicit objects
8. Control flow statements
9. Decision-making statements
10. Loop statements
11. JSP operations
12. JSP Literals

<%! int i=0; %>
<%! int a,b,c; %>

02

Lec-10

2 তারিখ ct

17.11.2019

CT-02 → 02-12-2019

Lab Quize → Exam date

A-10

Lab = 20

Lab = 20

Lab final = 40

Viva = 10

atleast 1 set

JSP Implict objects:

⇒ JSP implict objects are created during the translation phase of JSP to the servlet.

⇒ The objects can be directly used in scriptlets that directly goes in the service method.

⇒ They are created by the container automatically, and they can be accessed using objects.

How many implict objects are available in JSP? 9 types of implict objects available in The container.

1. out
2. request
3. response
4. config
5. application
6. session
7. pageContext.
8. page
9. Exception

① Out: Is the one of the implict objects to write the data to the buffer and send output to the client response.

⇒ Out objects allows us to access the servlets output stream.

⇒ Out is object of javax.servlet.jsp.jspwriter.

② Request: This is an instance of java.servlet.http.HttpServletRequest and it is one of the argument of service method.

→ It will be created by container for every request.

→ It will be used to request the information like parameters, header information, servername etc.

→ It uses the "getparameter()" to access the request parameter.

③ Response: Is an instance of a class which implements "HttpServletResponse" Interface

⇒ the object is created by the containers for each request.

⇒ It represents the response that can be given to the client.

⇒ the response implicit object is used to content type, cookie and redirect to response page.

Example: response.setContentType("text/html")

④ Config: Config is of the type java.servlet.servletConfig".

→ Created for each JSP page.

→ It is used to get the initialization parameter in web.xml.

Example: String servletName = Config.getServletName();

⑤ Application: It is instance of "javax.servletContext" and it is used to get the context information and attributes in JSP.

→ servlet Content object contains a set of methods which are used to interact with servlet container. We can find information about the servlet container.

Example: Application.getContextPath();

⑥ session: The session is holding HttpSessions object.

→ Series object is used to get, set and remove attributes to session scope and also used to get sessions information.

Example:

```
<% session.setAttribute("username", "id");
   session.setAttribute("isloggedin", true);
%>
```

Lab

1. MySQL connect with servlet,
2. Create, Read, Update, Delete.
3. Add Bootstrap theme with servlet