**Client Server Architecture**

It uses HTTP Protocol for communication between client and server

**Server-Side Programming**

**It can be done by using: -**

1. Java SE API’s – These are used for standalone applications
2. Common Gateway Interface- It help web servers generate dynamic web pages
3. Pulls down the performance

**Another Server-Side Programming**

1. **ASP.Net**
2. **Node.js**
3. **PHP**
4. **Servlets – We need API’s to do Server-side programming in JAVA EE**

**Servlets**

1. Server-Side Programming to generate dynamic web content
2. Efficient as compared to CGI – creates a thread for every new request
3. Works along all JVM Features – Platform Independence, Multithreading, object oriented
4. Servlets can work together with JAVA SE API’s like generics, exception handling, collections , multi-threading, JDBC

Java Servlet 3 for Java EE 6, JAVA SE 6 , JDK 1.6

Java Servlet 3.1 for JAVA EE 7

**In case of JAVA SE we already have Servlets and JSP files in our project structure but in case of our JAVA EE we need to add them.**

**Java Server Pages (JSP)**

**Problems with HTML**

1. Allows to render static data only
2. I/O operations to read and write HTML templates back is time consuming process.
3. Stuffing all HTML code (with CSS) code directly into servlet is a tedious task.
4. Specialized UI developers may not be equipped to handle servlet code.

**Java Server Pages (JSP)**

1. Allows dynamic content.
2. HTML + JAVA
3. Static Code + Dynamic code from server-side JAVA objects
4. JSP’s gets executed on the server side and gets rendered to the client.
5. JSP can work with the scripting language

Login.jsp

HTML + JAVA

HT

LoginServlet.class

LoginServlet.java

After Compilation

After Tanslation