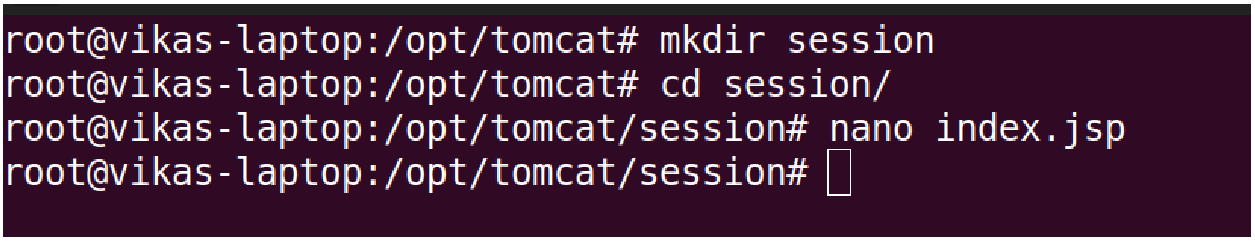
Tomcat Scenario Task:

Create a simple web application that uses session management to track the number of visits by a user.

SESSION MANAGEMENT:

Session management in Tomcat is a way for the server (Tomcat) to keep track of individual users' interactions with a web application. It allows the server to maintain specific information about each user, such as login status, preferences, or data that needs to be remembered during their visit.

STEP1:
Create a folder and index.jsp file inside tomcat folder

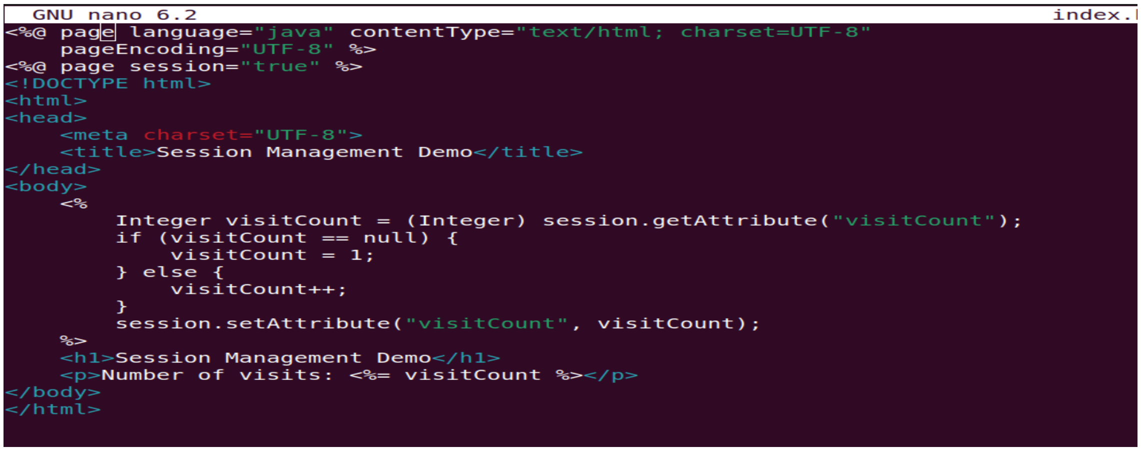


```
root@vikas-laptop:/opt/tomcat# mkdir session
root@vikas-laptop:/opt/tomcat# cd session/
root@vikas-laptop:/opt/tomcat/session# nano index.jsp
root@vikas-laptop:/opt/tomcat/session# 
```

STEP 2:
Add this code inside the index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<%@ page session="true" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Session Management Demo</title>
```

```
</head>
<body>
    <%
        Integer visitCount = (Integer) session.getAttribute("visitCount");
        if (visitCount == null) {
            visitCount = 1;
        } else {
            visitCount++;
        }
        session.setAttribute("visitCount", visitCount);
    %>
    <h1>Session Management Demo</h1>
    <p>Number of visits: <%= visitCount %></p>
</body>
</html>
```



```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" %>
<%@ page session="true" %>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Session Management Demo</title>
</head>
<body>
    <%
        Integer visitCount = (Integer) session.getAttribute("visitCount");
        if (visitCount == null) {
            visitCount = 1;
        } else {
            visitCount++;
        }
        session.setAttribute("visitCount", visitCount);
    %>
    <h1>Session Management Demo</h1>
    <p>Number of visits: <%= visitCount %></p>
</body>
</html>
```

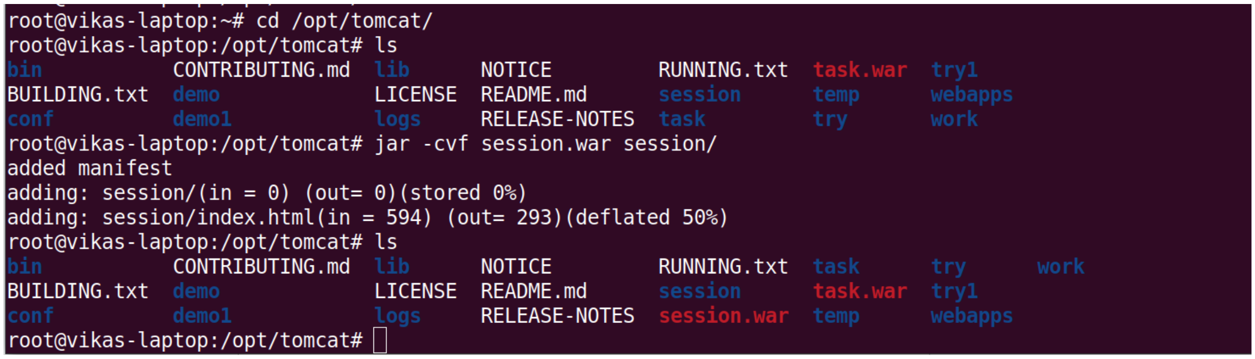
STEP 3:

Create a war file of this session directory and move it inside webapps ,which will automatically extract the directory.

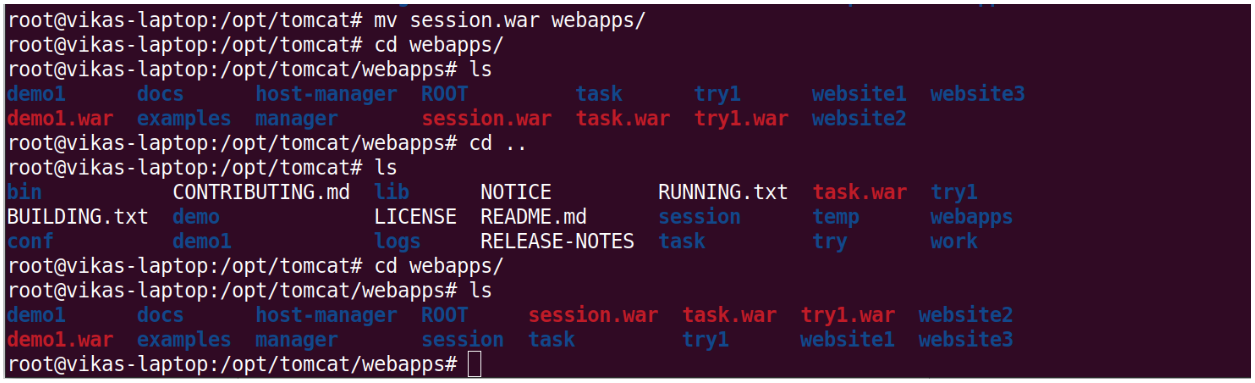
```
→cd /opt/tomcat
→jar -cvf session.war session
```

→mv session.war webapps

- Also move the index.jsp file which is from the extracted war file, directly inside the session folder which is created automatically.



```
root@vikas-laptop:~# cd /opt/tomcat/
root@vikas-laptop:/opt/tomcat# ls
bin           CONTRIBUTING.md  lib      NOTICE         RUNNING.txt  task.war  try1
BUILDING.txt  demo             LICENSE  README.md      session      temp      webapps
conf          demo1            logs     RELEASE-NOTES  task         try       work
root@vikas-laptop:/opt/tomcat# jar -cvf session.war session/
added manifest
adding: session/(in = 0) (out= 0)(stored 0%)
adding: session/index.html(in = 594) (out= 293)(deflated 50%)
root@vikas-laptop:/opt/tomcat# ls
bin           CONTRIBUTING.md  lib      NOTICE         RUNNING.txt  task      try       work
BUILDING.txt  demo             LICENSE  README.md      session      task.war  try1
conf          demo1            logs     RELEASE-NOTES  session.war  temp      webapps
root@vikas-laptop:/opt/tomcat# 
```



```
root@vikas-laptop:/opt/tomcat# mv session.war webapps/
root@vikas-laptop:/opt/tomcat# cd webapps/
root@vikas-laptop:/opt/tomcat/webapps# ls
demo1      docs      host-manager  ROOT         task      try1      website1  website3
demo1.war  examples  manager       session.war  task.war  try1.war  website2
root@vikas-laptop:/opt/tomcat/webapps# cd ..
root@vikas-laptop:/opt/tomcat# ls
bin           CONTRIBUTING.md  lib      NOTICE         RUNNING.txt  task.war  try1
BUILDING.txt  demo             LICENSE  README.md      session      temp      webapps
conf          demo1            logs     RELEASE-NOTES  task         try       work
root@vikas-laptop:/opt/tomcat# cd webapps/
root@vikas-laptop:/opt/tomcat/webapps# ls
demo1      docs      host-manager  ROOT     session.war  task.war  try1.war  website2
demo1.war  examples  manager       session  task         try1      website1  website3
root@vikas-laptop:/opt/tomcat/webapps# 
```

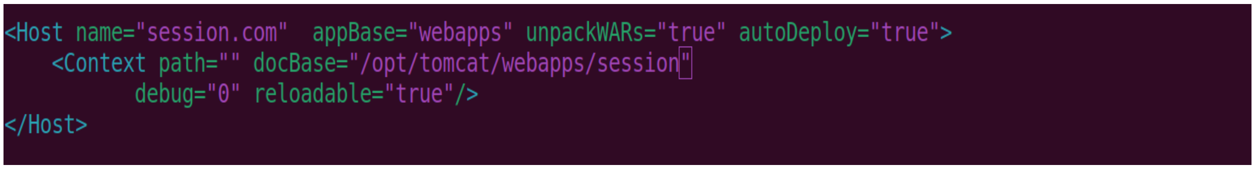
STEP 4:

- Open the server.xml file and configure the path for the new website.
  →nano /opt/tomcat/conf/server.xml

- Add the following lines to set the appropriate path, inside the server.xml

<Host name="session.com" appBase="webapps" unpackWARs="true" autoDeploy="true">

```
    <Context path="" docBase="/opt/tomcat/webapps/session"
          debug="0" reloadable="true"/>
</Host>
```

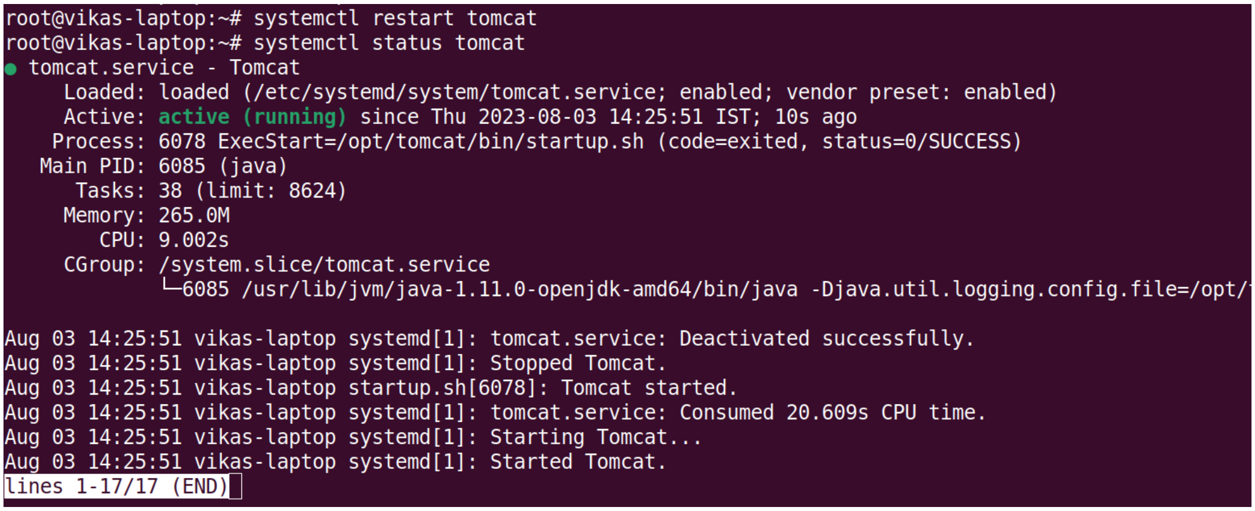


```
<Host name="session.com"  appBase="webapps" unpackWARs="true" autoDeploy="true">
    <Context path="" docBase="/opt/tomcat/webapps/session"
           debug="0" reloadable="true"/>
</Host>
```

- Save and exit

STEP 5:

Restart tomcat after updating the server.xml file



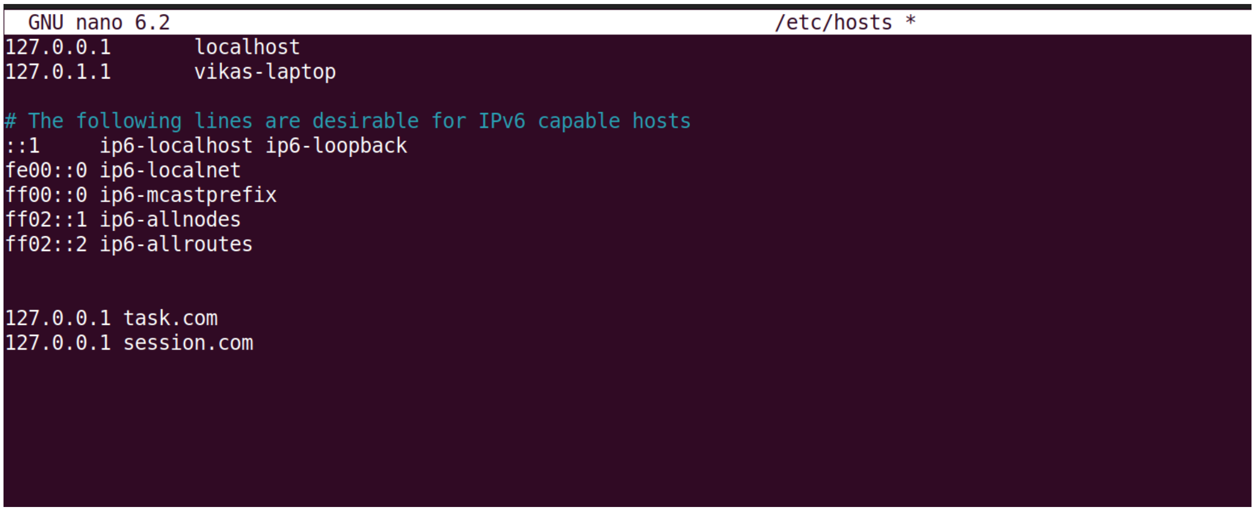
```
root@vikas-laptop:~# systemctl restart tomcat
root@vikas-laptop:~# systemctl status tomcat
● tomcat.service - Tomcat
     Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; vendor preset: enabled)
     Active: active (running) since Thu 2023-08-03 14:25:51 IST; 10s ago
    Process: 6078 ExecStart=/opt/tomcat/bin/startup.sh (code=exited, status=0/SUCCESS)
   Main PID: 6085 (java)
      Tasks: 38 (limit: 8624)
     Memory: 265.0M
        CPU: 9.002s
     CGroup: /system.slice/tomcat.service
             └─6085 /usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/

Aug 03 14:25:51 vikas-laptop systemd[1]: tomcat.service: Deactivated successfully.
Aug 03 14:25:51 vikas-laptop systemd[1]: Stopped Tomcat.
Aug 03 14:25:51 vikas-laptop startup.sh[6078]: Tomcat started.
Aug 03 14:25:51 vikas-laptop systemd[1]: tomcat.service: Consumed 20.609s CPU time.
Aug 03 14:25:51 vikas-laptop systemd[1]: Starting Tomcat...
Aug 03 14:25:51 vikas-laptop systemd[1]: Started Tomcat.
lines 1-17/17 (END)
```

STEP 6:

Open hosts file and add your machine ip and assign session.com to it as DNS name.

→ nano /etc/hosts



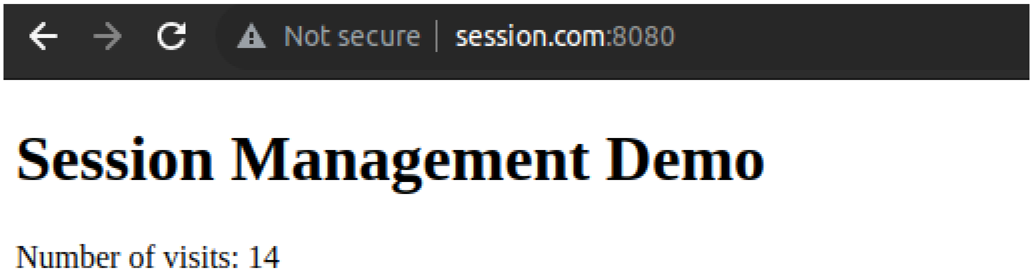
```
  GNU nano 6.2                                              /etc/hosts *
127.0.0.1       localhost
127.0.1.1       vikas-laptop

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allroutes


127.0.0.1 task.com
127.0.0.1 session.com
```

STEP 7:

Type session.com:8080 in the browser and then we will get the required page,refresh many times ,and the number of visits gets added up,this shows the session object is working.



Session Management Demo

Number of visits: 14