MongoDbServlet

Day 2 Work

1- Install MongoDb and set classpath
2- Establishing connection between MongoDb and Java
3- Fetch Data and Insert Data into db
4- Make Login form

Story 1- Install MongoDb and set classpath

To install the MOngoDb in UBUNTU

Task 1 -- Import the public key used by the package management system.

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv
              9DA31620334BD75D9DCB49F368818C72E52529D4
```

Task 2--Create a list file for MongoDB

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
xenial/mongodb-org/4.0 multiverse" | sudo tee
/etc/apt/sources.list.d/mongodb-org-4.0.list
```

Task 3-- Reload local package database.

```
sudo apt-get update
```

Task 4--Install the MongoDB packages

```
sudo apt-get install -y mongodb-org
```

Task 5-- Install the MongoDB packages

```
systemctl start mongod
```

Task 6-- Check status of MongoDB

```
systemctl status mongod
```

Story 2- Establishing connection between MongoDb and Java

To connect with MongoDB database, Java project includes the following steps.

Task 1- Download jar file of MongoDb from this url

[Link to download Mongodb jar]

Task 2- Create a .java file named JavaMongoDemo In your root folder

Here we establish the connection between java program and Mongodb and insert a record to Mongodb

```
import com.mongodb.MongoClient;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.Document;
public class JavaMongoDemo {
public static void main(String[] args){
try{
//---------- Connecting DataBase -------------------------//
MongoClient mongoClient = new MongoClient( "localhost" , 27017 );
//---------- Creating DataBase ---------------------------//
MongoDatabase db = mongoClient.getDatabase("kls");
//---------- Creating Collection -------------------------//
MongoCollection<Document> table = db.getCollection("employee");
//---------- Creating Document ---------------------------//
Document doc = new Document("name", "Peter John");
doc.append("id",12);
//---------- Inserting Data -----------------------------//
table.insertOne(doc);
}catch(Exception e){
System.out.println(e);
}
}  }
```

Concept 1-For connecting Database we need port number and IPaddress. 27017 port number of mongoDb Database

```
MongoClient mongoClient = new MongoClient( "localhost" , 27017 );
```

Concept 2: For creating Database

```
MongoDatabase db = mongoClient.getDatabase("klst");
```

Concept 3: For creating collection

```
MongoCollection<Document> table = db.getCollection("employee");
```

\Concept 4: To insert data into Mongodb first we create Document class object and pass parameter

```
Document doc = new Document("name", "kls");
```

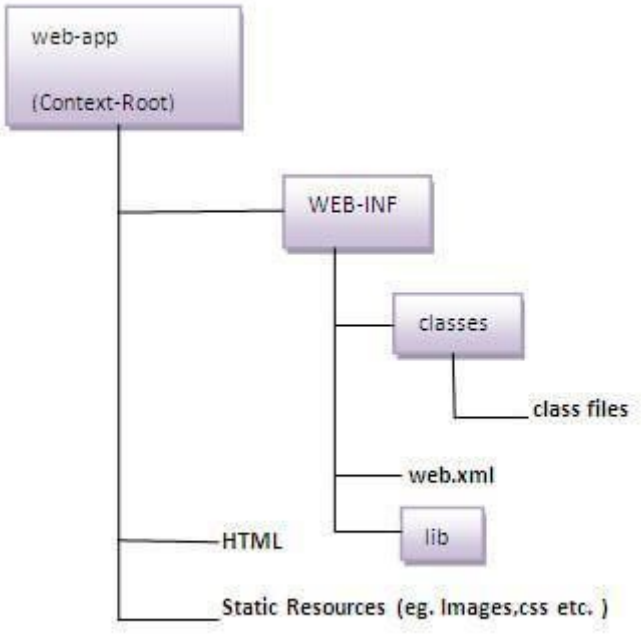
to insert the data

```
doc.insertOne(doc);
```

Story 3- Make Login Form

In this we create 3 .java file one login,verify and error page

Task 1 : first we create a directory structure



Task 2: After creating directory structure we create a .java file name loginServlet in root folder

```
 import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServlet;
```

```
import javax.servlet.http.HttpServletResponse;
import javax.servlet.ServletException;

import java.io.IOException;
import java.io.PrintWriter;

import java.io.IOException;
import com.mongodb.*;
import com.mongodb.DB;
import com.mongodb.BasicDBObject;
import com.mongodb.DBCollection;
```

```
import com.mongodb.MongoClient;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import org.bson.types.ObjectId;
public class Verify extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
String name=req.getParameter("name");
String pass=req.getParameter("pass");
String name1="";
String pass1="";
 try
{
```

```
  MongoClient mongoClient = new MongoClient( "localhost" , 27017
);
  DB db = mongoClient.getDB("at");
            DBCollection collection = db.getCollection("people");
            DBCursor dbo = collection.find();
  while(dbo.hasNext())
  {
           DBObject dbq=dbo.next();
```

```
 name1=(String) dbq.get("name");
       pass1=(String) dbq.get("pass");

}
if(name.equalsIgnoreCase(name1) && pass.equalsIgnoreCase(pass1))
{
     res.sendRedirect("login");
}
else
{
     res.sendRedirect("error");
}
}
```

```
catch(Exception e)
{
      out.println(e);
}
}
}
```

Task 3: After that we create another .java file name Verify to take input data from user and send to another servlet where we check that the user is valid or not

```
  import javax.servlet.http.*;
import javax.servlet.*;
import com.mongodb.client.MongoCollection;
import org.bson.*;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoDatabase;
import com.mongodb.DBCollection;
import com.mongodb.MongoClient;

import java.util.*;
```

```
import org.bson.types.ObjectId;
public class CheckServlet extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
```

```
out.println("<head><h1 align='center'
style='margin-top:30px;'><u>User LogIn</u></h1></head>");
out.println("<table width=30% style='margin-top:100px;height:200px'
align='center' border=1>");
out.println("<tr><td>Enter Username</td><td><input type='text'
name='name' ></td></tr>");
out.println("<tr><td>Enter Password</td><td><input type='password'
name='pass'></td></tr></table>");
out.println("<form action='login'><input type='submit'
style='color:blue;margin-left:580px;margin-top:30px;width:90px;hei
ght:40px;' value='LogIn'>");
}
}
```

Task 4 : We create a error page if user information incorrect we redirect to that page

```
 import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.ServletException;
```

```
import java.io.IOException;
import java.io.PrintWriter;

public class Error extends HttpServlet{
public void doGet(HttpServletRequest req,HttpServletResponse res)
throws ServletException,IOException
{
res.setContentType("text/html");
PrintWriter out=res.getWriter();
out.println("<html><body style='background-color: darkorange';>");
out.println("<head><h1 align='center'
style='margin-top:30px;'><u>UserName And Password
```

```
Incorrect</u></h1></head>");

}
}
```

Task 5 : After that we open terminal in root folder set classpath servlet and mongodb after that run this command to compile our .java file

```
Javac -d WEB-INF/classes *.java
```

Here * - denote to compile all java file

Task 6 : After that we map our all servlet in web.xml file

Task 7 : After that make war file in same terminal same as Servlet

```
Jar cvf  mongo.war *
```

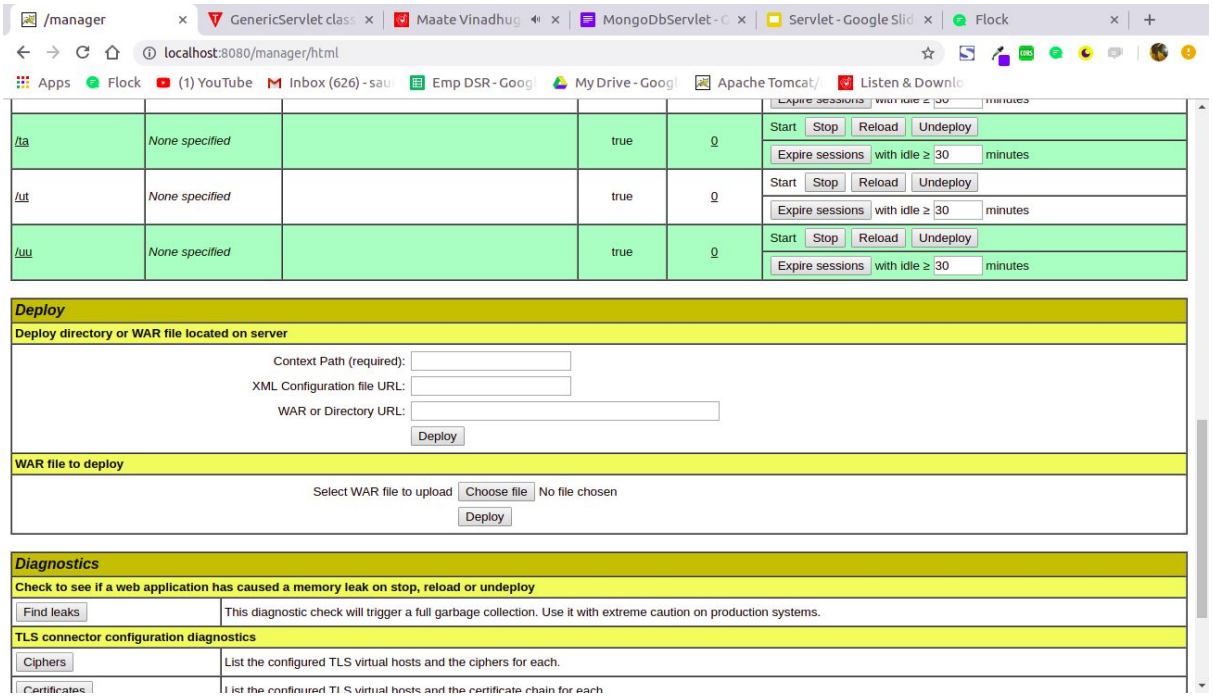
Task 8 : start your tomcat server by using this command

```
Systemctl start tomcat
```

Story 4 : To run our project

Task 1: Deploy your war file to Tomcat server goto the browser and type http://localhost:8080 in your address bar after that this page will

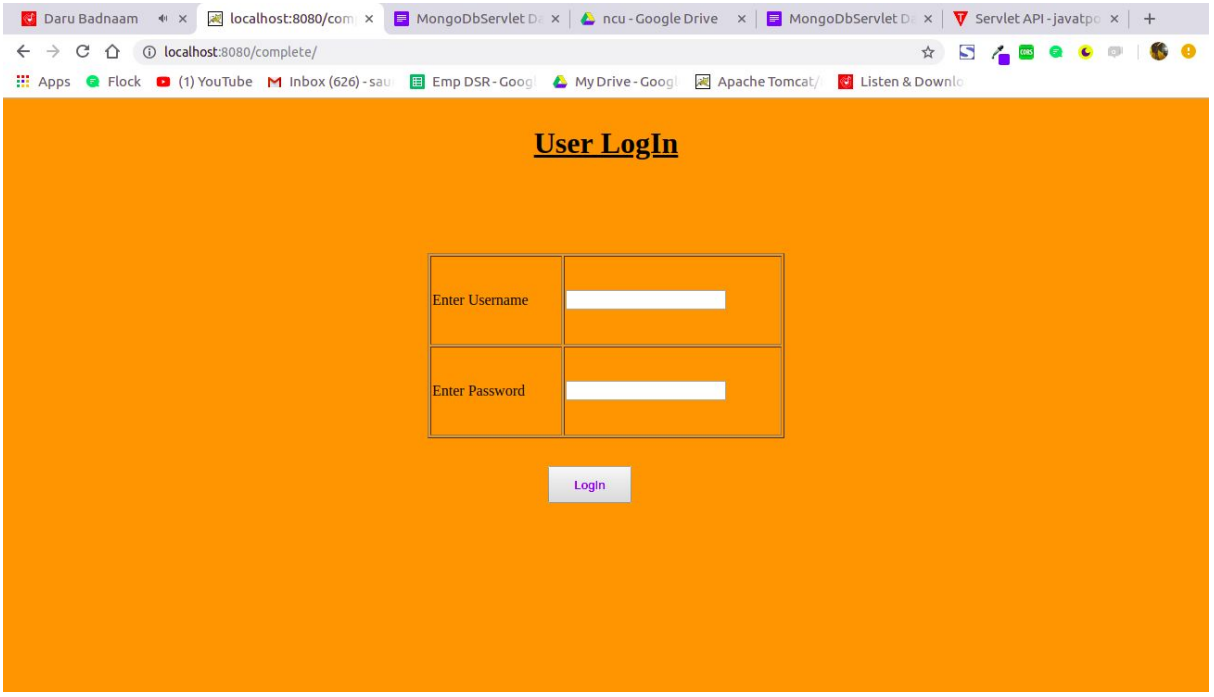
Task 2: deploy your war file



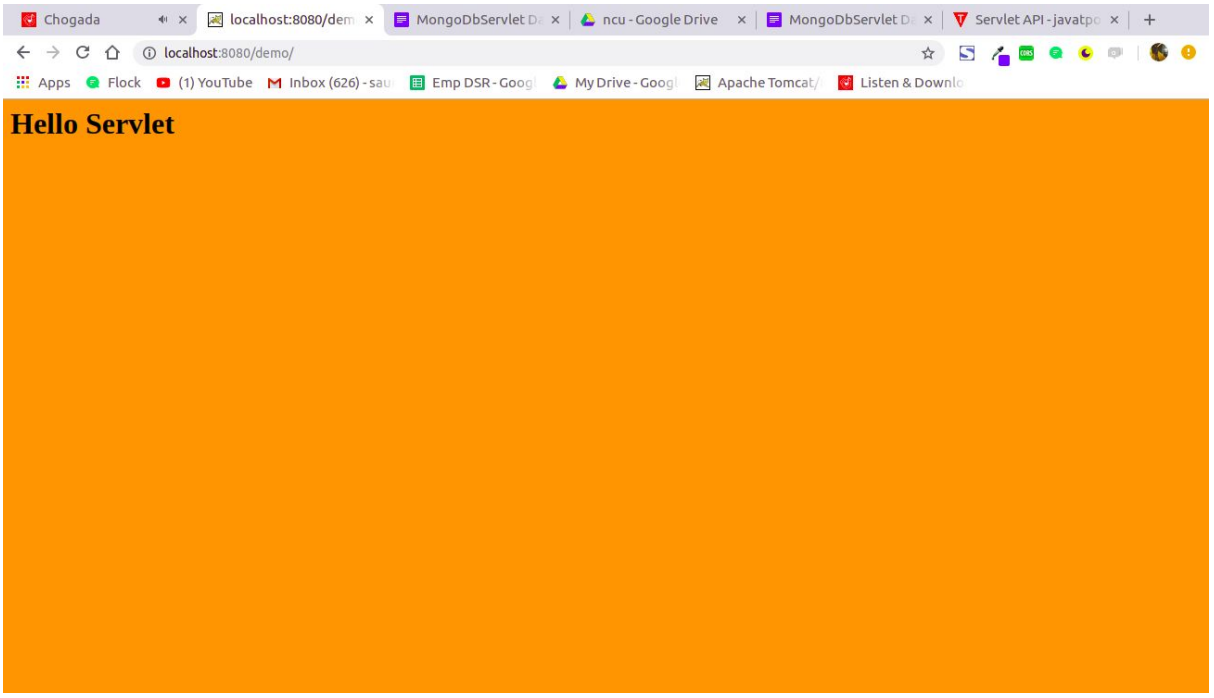
Task 3: Choose File-- Here you choose your war file and after that click on deploy Key

Task 4 : After succesfull deploy click on your war file to run your project

Task 5: After click your war file this page will open



Task 6: Enter your username and password if its correct this page will open login



Task 7: If username and password incorrect this page will open