Servlet Request Interface

→An object of Servelet Request is used to provide the client request information to servlet such as content type, content length, parameter names & values, header information, attributes etc.

Methods

Public String getParameter(String name) - used to obtain the value of parameter by name.

→ Container - different components i.e., multiple clients requests ~~go near~~ the server then the operation is performed in web container.

→ Println → Printstream → java.io

system.out → lang

→ res.setContentType("text/HTML"); → the content gets displayed in text or html method.

→ printwriter out = res.getwriter(); →

Example of Servlet Request to display the name of user

→ use getparameter method that returns the value for the given request parameter name

index.html

```
<form action="ServletReqDemo" method="get">
Enter yourname <input type="text" name="name"><br>
<input type="submit" value="login">
</form>
```

```
>servletexamples
  >src
   >main
     >webapp
       >index.html      } directory
```

```
public class ServletReqDemo extends HttpServlet {
    protected void doGiet (HttpServlet Request req, HttpServlet-
        Response res) throws ServletException, IoException
{
  res.setContentType("text/html");
  PrintWriter pw= res.getWriter();
  String name = req.getParameter("name");
   PW.println("Welcome "+name);
} }
```

- - - Arth.html

```
<form action="ArthDemo" method="get">
Enter first number: <input type="text" name="first"><br><br>
<input type="submit" name="add" value="add">
<input type="submit" name="sub" value="sub">
</form>
```

ArthDemo.java

```
 Public class ArthDemo extends


   String first = request.getParameter("first");
   String second = request.getParameter("second");
   if (first.equals("") || Second.equals(""))
           out.println("Enter both numbers for making
                                arithmetic operations");
   int fno = Integer.parseInt(first);
   int sno = Integer.parseInt(second);
   int result =0;
   if (request.getParamater ("add") != NULL)
```

```
{
result = fno +sno;
out.println ("<br> <br> Addition is : " +result);
}
```

Request Dispatcher in Servlet

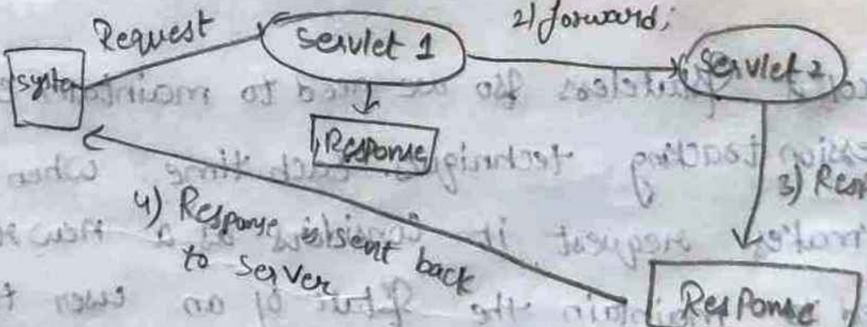
→ Provides the facility of dispatching the request to another resource it may be html, Servlet or JSP.

→ 2 methods

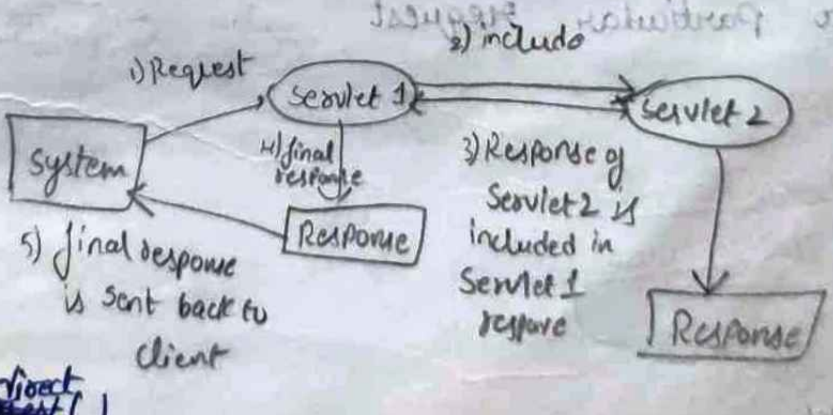
① public void forward (ServletRequest, ServletResponse response) throws ServletException, java.io.IOException: forwards a request from a Servlet to another resource.

② public void include (ServletRequest, ServletResponse response) throws ServletException, java.io.IOException

①



②



1) Request
servlet 1
2) include
servlet 2
System
4) final response
3) Response of Servlet 2 is included in Servlet 1 response
Response
5) final response is sent back to client
Response

Send Redirect()

The sendRedirect() method of HttpServletResponse interface can be used to redirect response to another resource, it may be Servlet (or) JSP (or) html file.

forward() method	SendRequest() method
Eg, request.get Request Dispatcher ("Servlet2"). forward (request, response);	Eg response. Send~~Request~~ Redirect ("Servlet2");
→ works at server side	→ works at Client side.

Session Tracking techniques

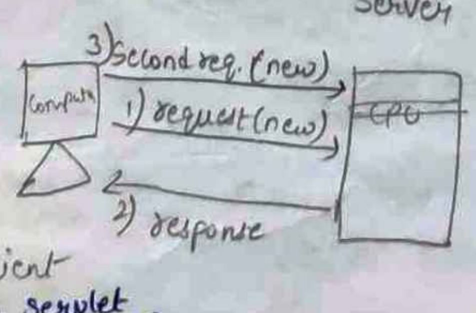
→ session tracking is used to recognize the Particular user.

4 techniques → Cookies
→ Hidden formfield
→ URL rewriting
} client side
→ HttpSession ——— serverside

→ Session means a particular interval of time.

→ Session Tracking is a way to maintain state(data) of an user. It is also known as session management in Servlet.

→ Http protocol is stateless so we need to maintain state using session tracking techniques. Each time when a server makes request it considers as a new request. So we need to maintain the state of an user to recognize to particular request (Cookies i.e, Accept cookies)

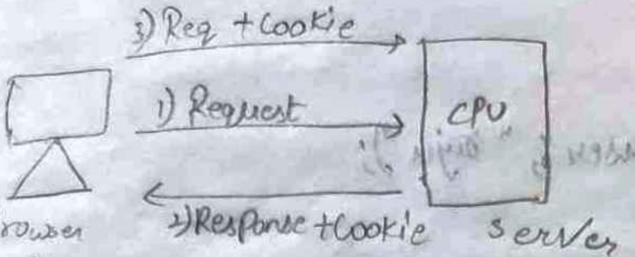


Cookie in servlet

→ Cookie is a small piece of information that is persisted between the multiple client requests.

→ It has a name, single value, optional attributes such as a comment, path & domain qualifiers, a maximum age & a version no.

→ Cookie is ~~s~~ stored in cache of browser.



↓

an app. software to access the web pages.

Types of Cookie

1. Non-Persistent - Valid for ~~s~~ single session Eg: otp
2. persistent - Valid for multiple sessions. Login & log out [then only it is removed by user]

   Eg: Fb, Insta..

Advantages → simplest technique of maintaing the state
↳ Cookie are maintained at client side

The current activity in a thread - state

Disadvantages → It will not work if Cookie is disabled from browser
↳ only textual information can be set in Cookie object.

Cookie class

→ javax.servlet.http.Cookie - provides functionality of using cookies

→ Cookie (String name, String value) - Constructs a Cookie with specified name & Value.

→ 4 methods

1. setname
2. getname
3. setValue
4. get Value

→ public void setMaxAge (int expiry) - Sets the maximum age of the cookie in seconds.

① Public void addCookie(Cookie ck); method of HttpServlet

Create Cookie

```
Cookie ck = new Cookie("user", "raju");
response.addCookie(ck);
```

Delete Cookie

```
Cookie ck=new Cookie("user", "");
ck.setMaxAge(0);
response.add Cookie (ck);
```

addCookies
response.addCookie(C);

get Cookie

```
Cookie ck[] = request.getCookies( );
for(int i=0; i<ck.length; i++) {
}   ~~out.println ("<br>" +ck[i].getValue());~~ Name
    Out.print("<br>" + ck[i].getName()+" "+ck[i].getValue());
 }
```

Pgm for Cookie in servlet

CookieIndex.html

```
<form action="CookieDemo1" method='Post'>
Name : <Input type="text" name="userName"><br>
<input type="submit" value="go">
</form>
```

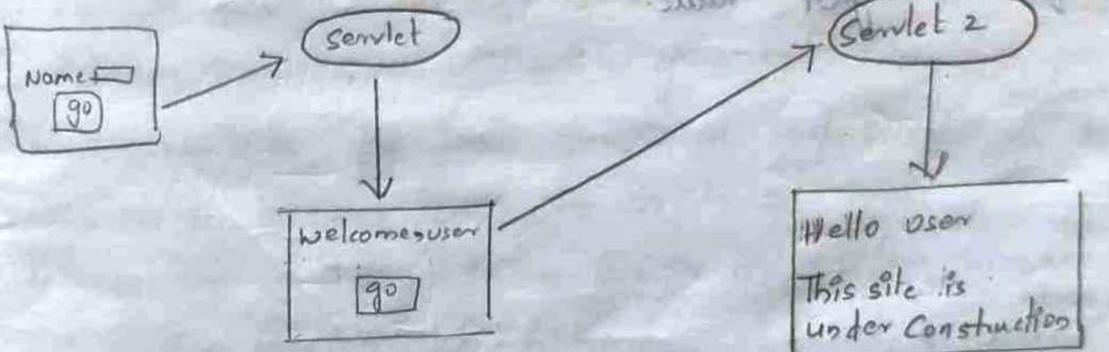
CookieDemo1.java

```
Public class CookieDemo1 extends HttpServlet {



   String n= request.getparameter ("userName");
   out.println ("welcome" +n);
Cookie ck=new Cookie ("uname", n);
response.addCookie(ck);

out.point("<foom action='CookieDemo2' method='post'>");
out.point("<input type='submit' Value='go'>");
out.point["</foom>");
}
}
```

CookieDemo2.Java

```
Cookie ck[]=request.getCookies();
out.point ('Hello" +ck[0].getValue());
```

→ for multiple cookies add another content into the Codes [Eg. Name, Password):

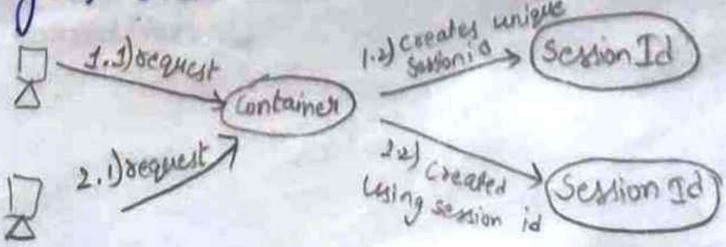


HttpSession Interyace

→ Container creates a session id for each user. The Container uses this id to identify the particular users.

→ object of HttpSession used to Peryorm 2 tasks,

1. bind objects
2. View & manipulate info abut a session such as session identifier, creation time, & last accesed time.



to get HttpSession object

The HttpServlet Request interface provides 2 methods to get the object of HttpSession

1. Public HttpSession getSession(): Returns the current Session associated with this request, or if the request doesnot have a session, creates one.

2. public HttpSession getSession (boolean create): Returns the Current HttpSession associated with this req., or if there is no session & create tue, returns a New Session.

Methods of HttpSession interface

1. Public String getId(): Returns a string containing the unique identifier value.

JDBC

→ Java database Connectivity – It is a Java-API to connect &/ execute the query with database.

→ Java.sql package provides predefined functionalities [Class & interfac]

→ API is a document that contains a description of all the features of a product or software.

→ It represents classes & interfaces that software programs can follow to communicate with each other.

→ An API can be created for applications, lib, OS etc. we can make communication with database.

JDBC API

① Connect to database
② Execute queries & update statements to database
③ Retrieve the result received from database.

→ JDBC Driver is a software component that enables java application to interact with the database

4 types —① JDBC - ODBC Bridge drivers [opendatabase Connectivity]
eg. MSAccess
② Native Driver
③ Network protocol Driver
④ Thin Driver.

→ 5 steps to connect any java application with database using JDBC

① Register the Driver class
② Create Connection
③ Create Statement
④ Execute queries
⑤ Close connection

Code in another book

Mysql

TEE D:\CSE20.TXT → It helps to create a file & stores the code which is written in mysql.

NOTEE; → Coming out from session

→ A JDBC project is a Simple Java project

Steps

1. File > New > Java project
2. ProjectName - JDBC CSE-D, finish
3. Right click on src, New > Java class >
4. Jarfile path

   mysql-connector-java-5.1.23-bin.jar - path

   Mysql 5.5 version need to be downloaded.

   we get an JDBC environment to write code
5. Run button - Run the java application

→ DATABASE (Paper) - logically correlated data offering wrt organizatn

→ DBMS (Pen) - software environment where the data can be stored & retrieved in convinent & efficient manner in an organization

→ MYSQL (Type of Pen) - Type of a DBMS - wrt to large project when we want to access multiple ~~functionalities~~ softwares (SQL, Mysql, Oracle, etc). So, multiple things can be used.

→ SQL - Interface - SQL is a interface b/w user & database.

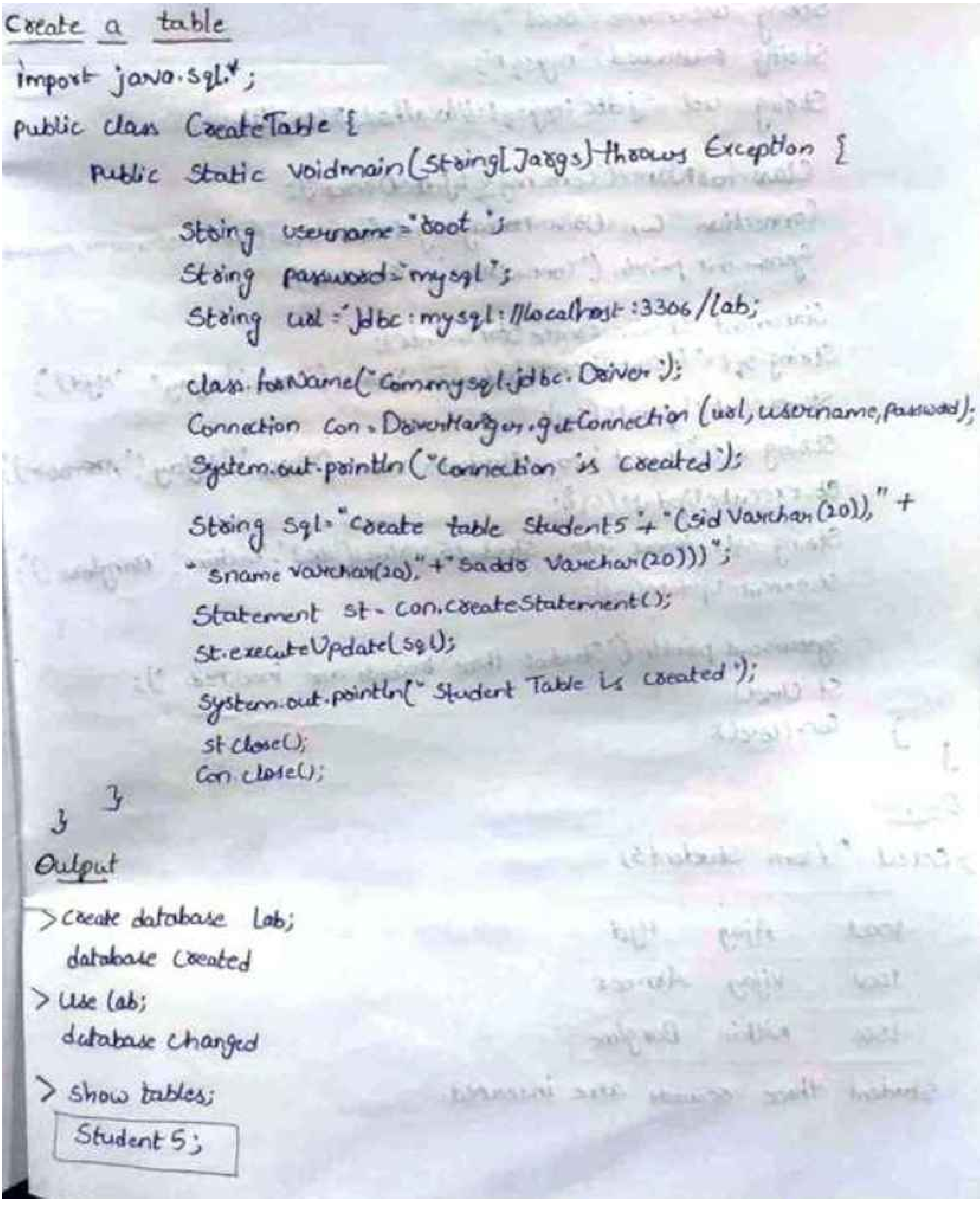
user — SQL — database

→ forName() is a static method of the java.lang.Class

→ getConnection() method of DriverManager class is used to establish Connection with the database.

→ The createStatement() method of Connection interface is used to create Statement.

→ Connection interface ~~is~~,



Create a table

```
import java.sql.*;
public class CreateTable {
    Public Static voidmain(String[]args) throws Exception {

        String username="root";
        String password="mysql";
        String url="jdbc:mysql://localhost:3306/lab;

        class.forName("Com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection(url,username,password);
        System.out.println("Connection is created");
        String sql="create table Student5"+"(sid Varchar(20)),"+
        "sname varchar(20),"+"saddr Varchar(20)))";
        Statement st=con.createStatement();
        St.executeUpdate(sql);
        System.out.println("Student Table is created");
        st.close();
        Con.close();
    }
}
```

Output

```
>Create database Lab;
 database created
>Use lab;
 database Changed
>show tables;
 Student 5;
```

Insert the Values

```
import java.sql.*;
Public class InsertData {
    Public static Voidmain (String[] args) throws Exception {
        String username = "root";
        String password = "mysql";
        String url = "jdbc:mysql://localhost:3306/lab;

        Class.forName("Com.mysql.jdbcDriver");
        Connection Con = DriverManager.get Connection (url, username, password);
        System.out.println ("Connection is Created");
       Statement st = Con.create Statement();
       String sql = "insert into Student5 Values ('1001', 'Ajay', 'Hyd')";
       St.executeUpdate(sql);
       String sql = "insert into Student5 values ('1002', 'Vijay', 'Armoor')";
       St.executeUpdate(sql);
       String sql = "Insert into Student5 Values('1003', 'Nithin', 'Banglore')";
       St.executeUpdate(sql);

       Systemout.println ("Student three records are inserted..");
       St.Close();
    }  Con.close();
}
```

Output

```
>Select * from Student5;
```

1001	Ajay	Hyd
1002	Vijay	Armoor
1003	Nithin	Banglore

Student three records are inserted..

Update Data

```
import java.sql.*;
Public class UpdateData {
    Public static Void main (String[ ]args) throws Exception {
        String username="root";
        String password="mysql";
        String url="jdbc:mysql://localhost:3306/lab";
        Class.forName("Com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection(url,username,
                                                   Password);
        System.out.println("Connection is created");
        Statement st=Con.createStatement();
        String sql="update Student Set saddr='Hyderabad' where
                                          Sid=1001";
        St.executeUpdate(sql);
        System.out.println("Student one record is updated...");
        St.close();
        Con.close();
    }
}
```

Output:

```
Select * from students;

    1001  Ajay    Hyderabad
    1002  Vijay   Armoor
    1003  Nithin  Banglore
Student one record is updated...
```

DeleteData

```
import java.sql.*;
Public class DeleteData {
    Public static Void main(String[]args)throws Exception {
            String username="root";
            String password="mysql";
            String url="jdbc:mysql://localhost:3306/lab;
            Class.forName("Com.mysql.jdbc.Driver");
            Connection con=DriverManager.getConnection(url,username,
                                                   Password);
            System.out.println("Connection is created");
            Statement st= Con.createStatement();
            String sql="delete from student where sid=1003";
            st.executeUpdate(sql);
            System.out.println("student one record is deleted...");
            St.close();
            Con.close();
    }
}
```

Output

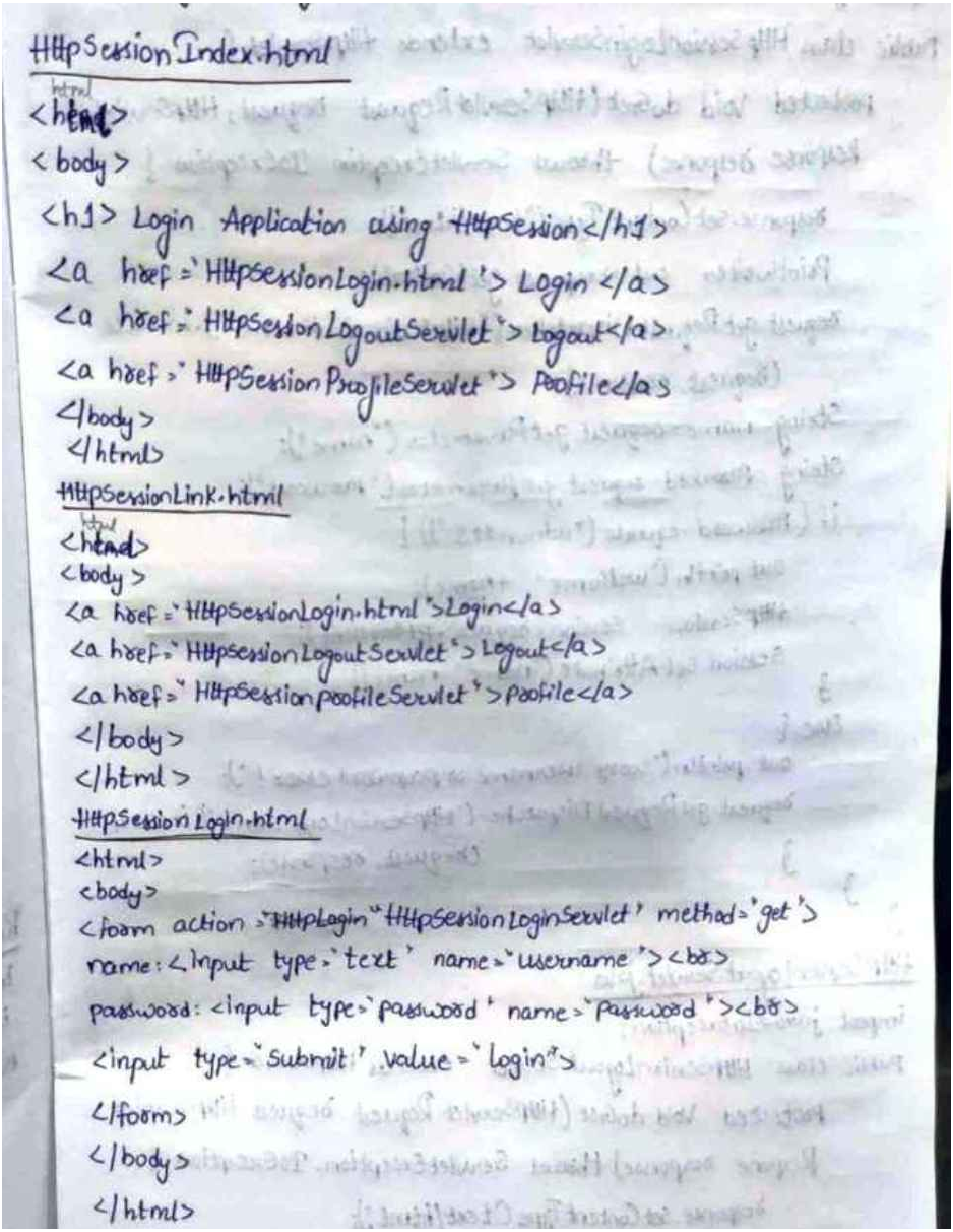
```
Select * from Students;

    1001    Ajay    Hyderabad
    1002    vijay   Armoor

Student one record is deleted...
```

SelectData

```
import java.sql.*;
Public Class SelectData {
        Public static voidmain (String[ ]args) throws Exception {
                String username = "root";
                String password = "mysql";
                String url = "jdbc:mysql://localhost:3306/lab";

                Class.forName("Com.mysql.jdbc.Driver");
                Connection Con = DriverManager.getConnection(url, username
                                                    password);
                system.out.println ("Connection is created");

                Statement St = Con.create Statement( );
                Result Set rs = St.executeQuery ("select *from students");
                System.out.println ("All the Employee Details\n");
                while(rs.next()) {
                        System.out.println("Student ID: " +rs.getString(1));
                        system.out.println("Student Name: " +rs.getString
                                                                  (2));
                        System.out.println ("Student Address: " +rs.getString
                                                                  (3));
                }
                St.close();
                Con.close();
        }
}
```

Output

```
All the Employee Details
Student ID: 1001
Student name: Ajay
Student Address: Hyderabad

Student ID: 1002
Student Name: Vijay
Student Address: Armoor
```



HttpSessionIndex.html

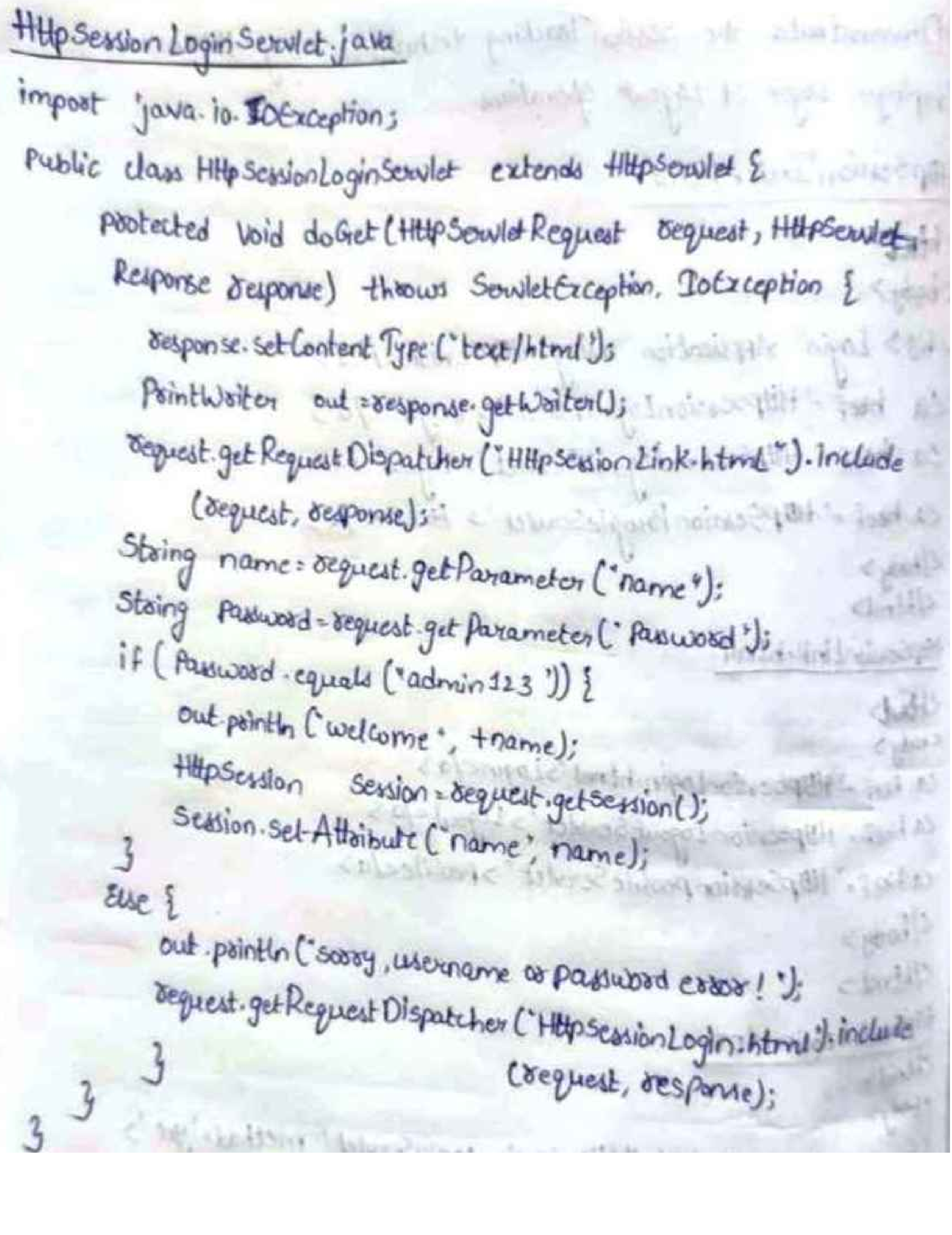
```
<html>
<body>
<h1> Login Application using HttpSession</h1>
<a href='HttpSessionLogin.html'> Login </a>
<a href='HttpSessionLogoutServlet'> Logout</a>
<a href='HttpSessionProfileServlet'> Profile</a>
</body>
</html>
```

HttpSessionLink.html

```
<html>
<body>
<a href='HttpSessionLogin.html'>Login</a>
<a href='HttpSessionLogoutServlet'>Logout</a>
<a href='HttpSessionprofileServlet'>Profile</a>
</body>
</html>
```

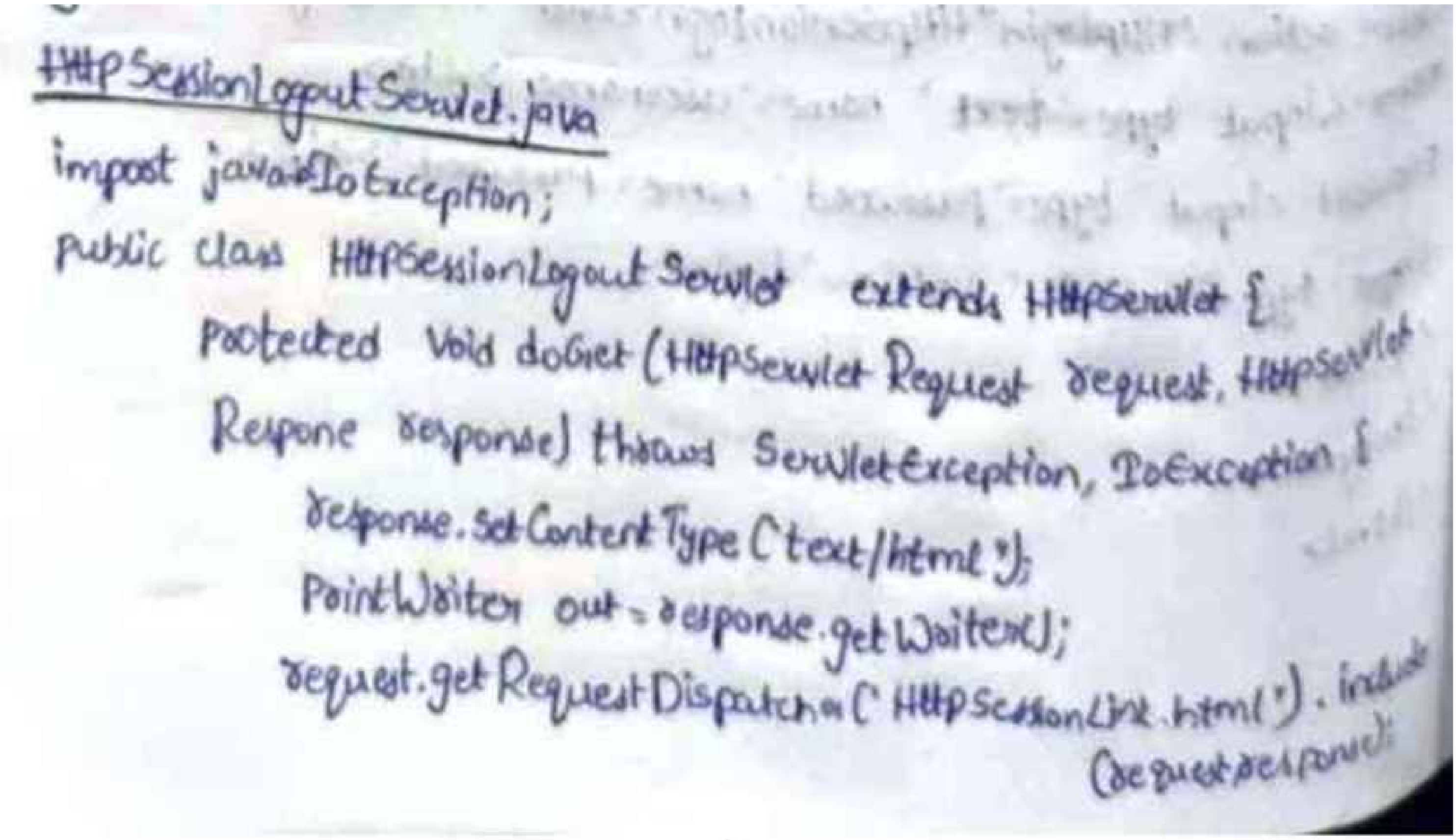
HttpSessionLogin.html

```
<html>
<body>
<form action='HttpSessionLoginServlet' method='get'>
name:<input type='text' name='username'><br>
password:<input type='password' name='password'><br>
<input type='submit' value='login'>
</form>
</body>
</html>
```



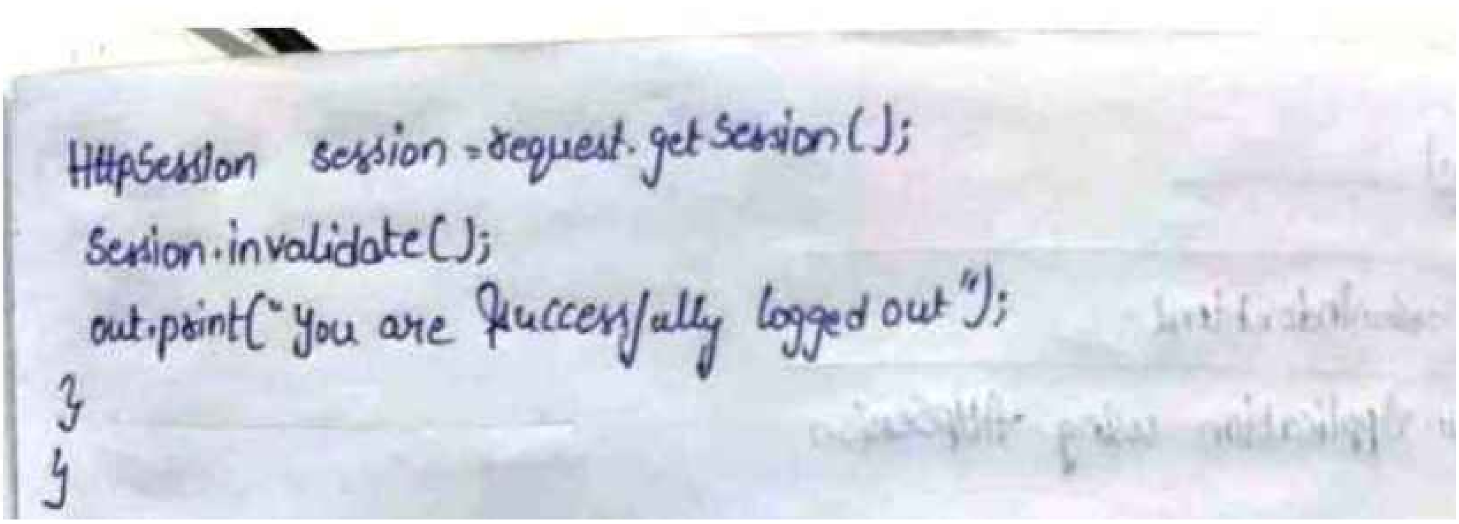
HttpSession LoginServlet.java

```
import java.io.IOException;
Public class HttpSessionLoginServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServlet
    Response response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out =response.getWriter();
       request.getRequestDispatcher("HttpSessionLink.html").include
            (request, response);
      String name= request.getParameter("name");
      String password=request.getParameter("password");
      if (Password.equals("admin123")) {
          out.println("welcome", +name);
          HttpSession   session=request.getSession();
          Session.set-Attribute("name", name);
      }
      Else {
          out.println("sorry,username or password error!");
          request.getRequestDispatcher("HttpSessionLogin.html").include
                                        (request, response);
      }
   }
}
```

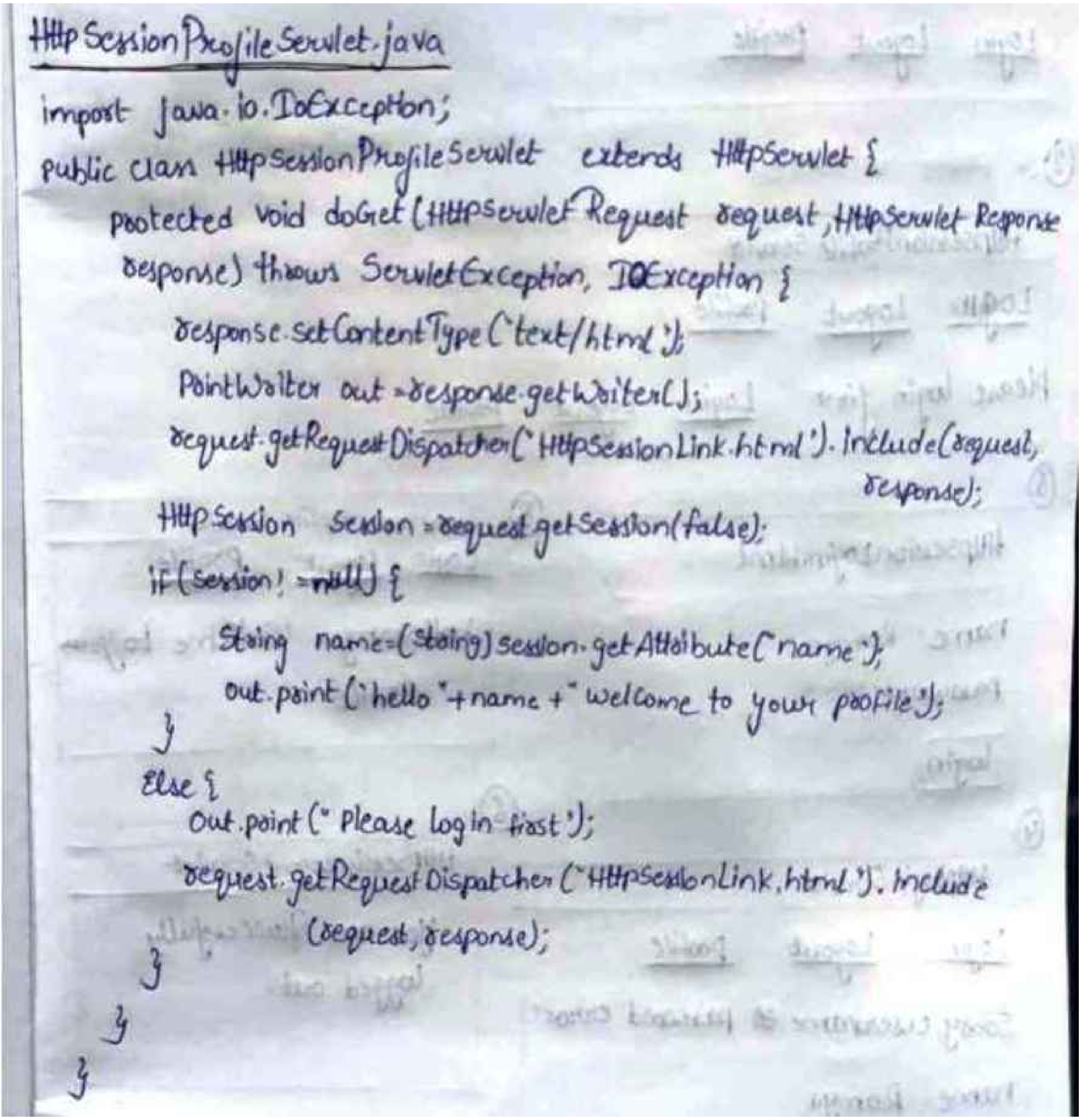


HttpSessionLogoutServlet.java

```
impost java.io.IoException;
public class HttpSessionLogoutServlet extends HttpServlet {
        protected void doGet (HttpServlet Request request, HttpServlet
        Respone response) throws ServletException, IoException {
              response.setContentType ("text/html");
              PrintWriter out = response.getWriter();
              request.getRequestDispatcher("HttpSessionLink.html").include
                                                          (request,response);
```



```
HttpSession session = request.getSession();
Session.invalidate();
out.print("You are successfully logged out");
}
}
```



HttpSessionProfileServlet.java

```
import java.io.IOException;
public class HttpSessionProfileServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
     response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        request.getRequestDispatcher("HttpSessionLink.html").include(request,
                                                              response);
      HttpSession session = request.getSession(false);
      if(session! =null) {
            String name=(String)session.getAttribute("name");
              out.print("hello "+name + " welcome to your profile");
      }
      Else {
         Out.print(" Please log in first");
         request.getRequestDispatcher("HttpSessionLink,html"). include
                  (request, response);
      }
   }
}
```

Output

① HttpsessionIndex.html

Login Application using HttpSession

Login Logout profile

② If profile is clicked

HttpsessionProfile Servlet

Login Logout profile

please login first Login Logout profile

③ HttpsessionLogin.html

name: Ramya

password: xxxx

login

④ HttpLogin Servlet

Login Logout profile

Sorry, username or password error!

name: Ramya

password: xxxxxxxx

login

- - - - - - - -

If

Welcome

Login Ramya

⑤ HttpSessionProfile Servlet

Login Logout Profile

helloramya welcome to yo

Profile

⑥ HttpsessionlogoutServlet

You are successfully logged out.