



INSTITUT NATIONAL DES POSTES ET TÉLÉCOMMUNICATIONS

RAPPORT DE PROJET

Dynamic Web project : Psychologue enline

Realisé par :
Driouch Meryam
Azzim Taha

Sous l'encadrement de :
Dr Mahmoud Rlhamlaoui

Année universitaire :2020-2021

Contents

1	Page Login	2
1.1	Page d'identification	2
1.2	Connexion avec une base de données MySQL : vérification des identifiants	3
1.2.1	Classe Session	3
1.2.2	Classe DB	4
1.2.3	Création de la base de données "userdb"	6
1.3	Login servlet	7
1.4	Teste de login	9
1.5	Configuration des differents acteurs : Psychologue, RH et Util- isateur	9
1.6	Partie CSS	13
2	Conception de la base de données	15

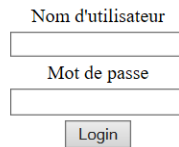
1 Page Login

1.1 Page d'identification

Créons premièrement un fichier Login.jsp dans lequel on aura la description de la page d'identification des différents utilisateurs.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Login</title>
6 </head>
7 <body>
8 <div align = "center">
9 <form action="login" method="post">
10 <table>
11 <tr>
12 <td align ="center">Nom d'utilisateur</td>
13 </tr>
14 <tr>
15 <td><input type="text" name="nom"></td>
16 </tr>
17 <tr>
18 <td align ="center">Mot de passe</td>
19 </tr>
20 <tr>
21 <td><input type="password" nom="mot de passe"></td>
22 </tr>
23 <tr>
24 <td align ="center"><input type="submit" value="Login"></td>
25 </tr>
26 </table>
27 </form>
28 </div>
29 </body>
30 </html>
```

On aura le resultat simple suivant



The image shows a simple login form. It consists of two text input fields stacked vertically. The first field is labeled 'Nom d'utilisateur' and the second is labeled 'Mot de passe'. Below the second field is a button labeled 'Login'.

Figure 1: Login.jsp

1.2 Connexion avec une base de données MySQL : vérification des identifiants

1.2.1 Classe Session

On aura besoin d'une classe *Session* (package : loginseccion) qui va récupérer pendant chaque identification le nom et le passe entrés.

```
1
2 package loginseccion;
3
4 public class session {
5     private String nom;
6     private String passe;
7     public String returnNom() {
8         return nom;
9     }
10    public String affecteNom(String nom) {
11        this.nom = nom;
12    }
13    public String returnPasse() {
14        return passe;
15    }
16    public String affectePasse(String passe) {
17        this.passe = passe;
18    }
19 }
```

1.2.2 Classe DB

La classe DB (Package : base_donnees) va permettre dans un premier lieux la connexion avec une base de données MySQL (userdb) qu'on va créer par la suite, puis vérifie si les identifiants (nom et mot de passe) entrés figurent dans cette base.

```
1 package base_donnees;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import loginsession.*;
8 public class DB {
9     private String dbUrl = "jdbc:mysql://localhost:3306/userdb?
10                             useJDBCCompliantTimezoneShift=true&
11                             useLegacyDatetimeCode=false&serverTimezone=UTC";
12     private String dbUname = "AzzimDriouich";
13     private String dbPassword = "0000";
14     private String dbDriver = "com.mysql.cj.jdbc.Driver";
15     public void loadDriver(String dbDriver)
16     {
17         try {
18             Class.forName(dbDriver);
19         } catch (ClassNotFoundException e) {
20             e.printStackTrace();
21         }
22     }
23     public Connection getConnection()
24     {
25         Connection con = null;
26         try {
27             con = DriverManager.getConnection(dbUrl, dbUname, dbPassword);
28         } catch (SQLException e) {
29             e.printStackTrace();
30         }
31     }
32 }
```

```

31         return con;
32     }
33     public boolean valider_donnees(Session session)
34     {
35         boolean status = false;
36
37         loadDriver(dbDriver);
38         Connection con = getConnection();
39         String sql = "SELECT *
40                     FROM login
41                     WHERE nom = ?
42                     AND mot_de_passe =?";
43         PreparedStatement ps;
44         try {
45             ps = con.prepareStatement(sql);
46             ps.setString(1, session.returnNom());
47             ps.setString(2, session.returnPasse());
48             ResultSet rs = ps.executeQuery();
49             status = rs.next();
50
51         } catch (SQLException e) {
52             e.printStackTrace();
53         }
54         return status;
55     }
56 }

```

1.2.3 Création de la base de données "userdb"

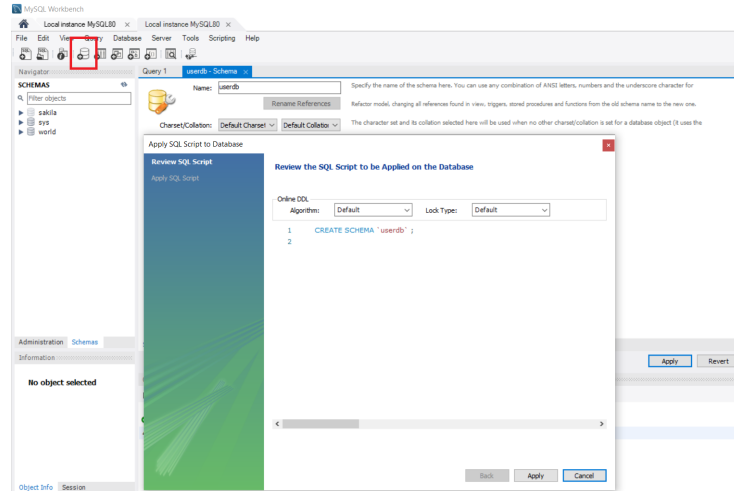


Figure 2: create schema

Après, on doit créer note tableau *login* avec les deux colonnes *nom* (Clé primaire et non null) et *mot_de_passe* (non null).

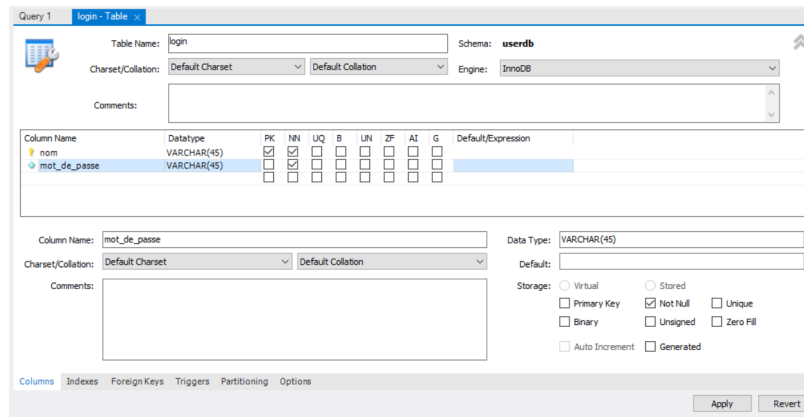


Figure 3: Création de la table

Afin de se connecter, on va insérer quelques utilisateurs à la table *login*.

Review the SQL Script to be Applied on the Database

```
1 INSERT INTO `userdb`.`login` (`nom`, `mot_de_passe`) VALUES ('Azzim', '111');
2 INSERT INTO `userdb`.`login` (`nom`, `mot_de_passe`) VALUES ('Driouich', '222');
3
```

Figure 4: Insertion des utilisateurs

1.3 Login servlet

Dans le package web, introduisant la première servlet qui va se servir de l'authentification et diriger l'utilisateur vers son compte si les identifiants sont corrects ou actualiser la page login sinon.

Pour cela, ajoutons un simple fichier Succes.jsp

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Succes</title>
6 </head>
7 <body>
8 Succes
9 </body>
10 </html>
```


Puis la servlet serait comme suit :

```
1 package web;
2
3 import java.io.IOException;
4 import javax.servlet.ServletException;
5 import javax.servlet.annotation.WebServlet;
6 import javax.servlet.http.HttpServlet;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9 import base_donnees.*;
10 import loginsession.*;
11
12 @WebServlet("/login")
13 public class LoginServlet extends HttpServlet {
14
15     protected void doPost(HttpServletRequest request,
16                           HttpServletResponse response)
17     throws ServletException, IOException {
18         String nom = request.getParameter("nom");
19         String passe = request.getParameter("mot de passe");
20
21         Session session = new Session();
22         session.affecteNom(nom);
23         session.affectePasse(passe);
24
25         DB connexion_db = new DB();
26         if(connexion_db.valider_donnees(session)) {
27             response.sendRedirect("Succes.jsp");
28         }
29         else {
30             response.sendRedirect("login.jsp");
31         }
32     }
33 }
```

1.4 Teste de login

Exécutons le programme et essayons une authentification avec l'un des utilisateurs déclarés dans la base de données :

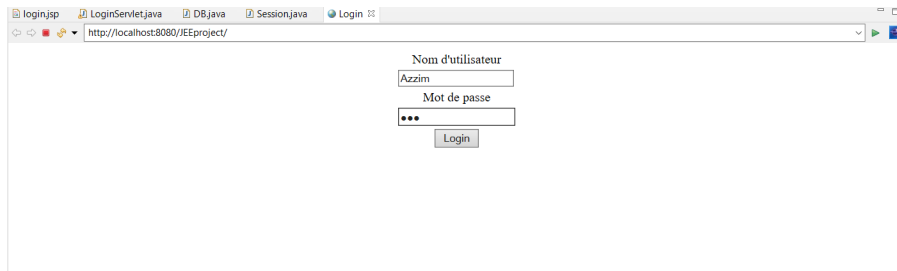


Figure 5: Exécution

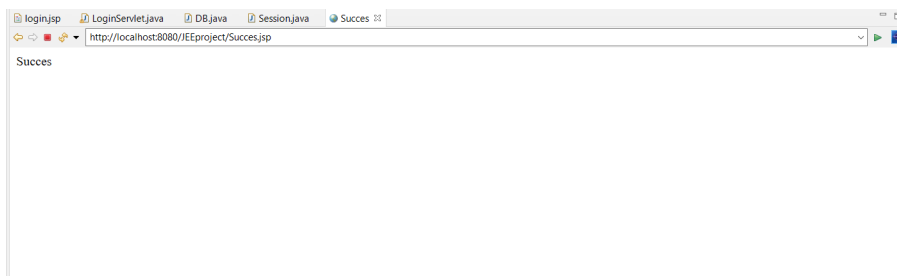
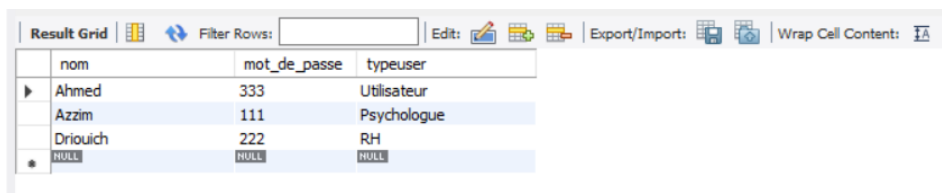


Figure 6: Authentification avec succès

1.5 Configuration des différents acteurs : Psychologue, RH et Utilisateur

Après l'authentification, l'inscrit doit être dirigé vers sa page personnelle. On distingue entre 3 type d'inscrits : Psychologue, RH et utilisateur. De ce fait, on doit modifier la table *login* en ajoutant la colonne *type*.



	nom	mot_de_passe	typeuser
▶	Ahmed	333	Utilisateur
	Azzim	111	Psychologue
	Driouch	222	RH
*	NULL	NULL	NULL

Figure 7: Table login

Ensuite on doit modifier la classe *Session* en ajoutant le String type et les méthodes affectType et returnType.

```
1 package loginsession;
2
3 public class Session {
4     private String nom;
5     private String passe;
6     private String typeUser;
7     public void affectType(String typeUser) {
8         this.typeUser = typeUser;
9     }
10    public String returnType() {
11        return typeUser;
12    }
13    public String returnNom() {
14        return nom;
15    }
16    public void affecteNom(String nom) {
17        this.nom = nom;
18    }
19    public String returnPasse() {
20        return passe;
21    }
22    public void affectePasse(String passe) {
23        this.passe = passe;
24    }
25 }
```

Pour la classe DB, pendant la validation des identifiants (valide_donnees), on récupère le type de l'inscrit et on affecte sa valeur à userType en faisant appel à affectType

```
1 public boolean valider_donnees(Session session)
2 {
3     boolean status = false;
4
5     loadDriver(dbDriver);
6     Connection con = getConnection();
7     String sql = "select * from login where nom =? and mot_de_passe =?";
8     PreparedStatement ps;
9     try {
10        ps = con.prepareStatement(sql);
11        ps.setString(1, session.returnNom());
12        ps.setString(2, session.returnPasse());
13        ResultSet rs = ps.executeQuery();
14        status = rs.next();
15        session.affectType(rs.getString("typeuser"));
16    } catch (SQLException e) {
17        e.printStackTrace();
18    }
19    return status;
20 }
```

Finalement la servlet doit diriger chaque type d'inscrit vers sa page personnelle (Psychologue.jsp, RH.jsp ou Utilisateur.jsp)

```
1  @WebServlet("/login")
2  public class LoginServlet extends HttpServlet {
3
4      protected void doPost(HttpServletRequest request,
5                             HttpServletResponse response)
6      throws ServletException, IOException {
7          String nom = request.getParameter("nom");
8          String passe = request.getParameter("mot de passe");
9
10         Session session = new Session();
11         session.affecteNom(nom);
12         session.affectePasse(passe);
13         DB connexion_db = new DB();
14         if(connexion_db.valider_donnees(session)) {
15             if(session.returnType().equals("Psychologue")) {
16                 response.sendRedirect("Psychologue.jsp");
17             }
18             else if(session.returnType().equals("Utilisateur")) {
19                 response.sendRedirect("Utilisateur.jsp");
20             }
21             else if(session.returnType().equals("RH")) {
22                 response.sendRedirect("RH.jsp");
23             }
24         }
25         else {
26             response.sendRedirect("login.jsp");
27         }
28     }
29
30 }
```

1.6 Partie CSS

Pour finir cette partie, introduisant un fichier css *loginCSS.css* pour les raisons esthétiques.

```
1  button {
2      background-color: white;
3      color: black;
4      padding: 14px 20px;
5      border-radius: 10px;
6      margin: 8px 0;
7      border: none;
8      cursor: pointer;
9      width: 100%;
10 }
11 input[type=text], input[type=password] {
12     width: 100%;
13     border-radius: 10px;
14     padding: 12px 20px;
15     margin: 8px 0;
16     display: inline-block;
17     border: 1px solid #ccc;
18     box-sizing: border-box;
19 }
20 button:hover {
21     opacity: 0.8;
22     background-color: black;
23     color: white;
24 }
25 body {
26     background-image: url("1.gif");
27     background-attachment: fixed;
28     background-repeat: no-repeat;
29     background-position: center;
30     background-size: cover;
31     background-color: white;
32 }
```

```

33
34 .login {
35     overflow: hidden;
36     opacity: 0.8;
37     background-color: #8e8e8e;
38     padding: 20px 30px 30px 30px;
39     border-radius: 10px;
40     top: 100px;
41     width: 400px;
42     box-shadow: 5px 10px 10px rgba(green, 0.2);
43 }

```

le fichier *login.jsp* sera aussi changé

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Login</title>
6 </head>
7 <link href="loginCSS.css" rel="stylesheet" type="text/css"><body>
8 <div align = "center">
9 <form class = login action="login" method="post">
10 <table>
11 <tr>
12 <td><input type="text" name="nom" placeholder="Nom d'utilisateur"></td>
13 </tr>
14 <tr>
15 <td><input type="password" name="mot de passe" placeholder="Password"></td>
16 </tr>
17 <tr>
18 <td align ="center"><button type="submit" value="Login">login</button></td>
19 </tr>
20 </table>
21 </form>
22 </div>
23 </body>
24 </html>

```

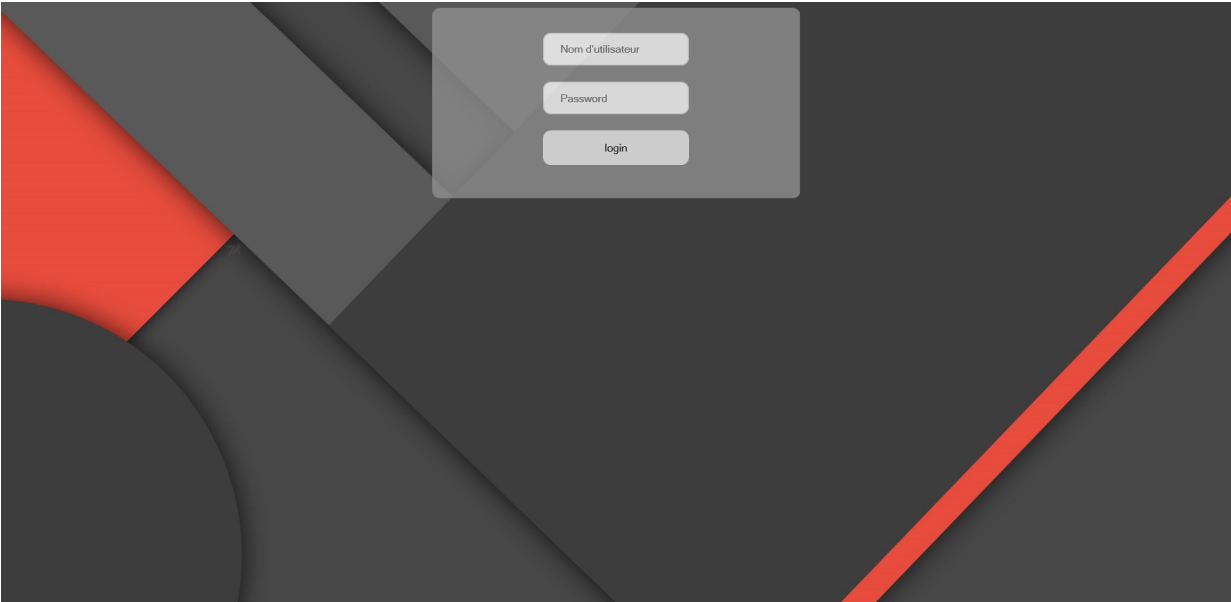


Figure 8: Page login

2 Conception de la base de données

De plus du tableau *login* on aura besoin d'autres pour stocker les questions posés par les psychologues et les réponses récupérés de la part des utilisateurs.

On propose l'ajout de deux tableaux :

- **Formulaires** : renferme l'id du formulaire (Clé primaire), le nom du psychologue (Créateur du formulaire), le nom d'utilisateur (Destinataire) et l'état du formulaire (Approuvé ou non par le RH)
- **Questions** : contient la question, son l'id (Clé primaire), l'id du formulaire où se trouve et les réponses fournies par les utilisateurs.

Formulaires			
id_formulaire	utilisateur	psychologue	etat
.	.	.	.
.	.	.	.
.	.	.	.

Questions			
id_question	question	id_formulaire	reponse
.	.	.	.
.	.	.	.
.	.	.	.