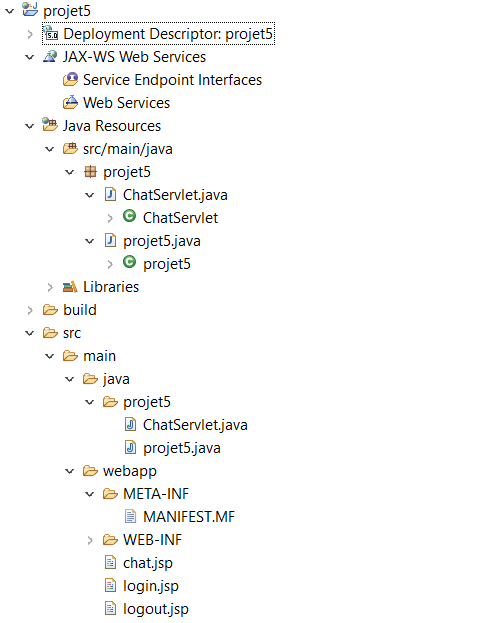
**SUJET PROJET : WebSocket(avec JSP/Servlet)  
  
Objectif du projet** : Quel est le but de votre application WebSocket ?   
Le but de votre application WebSocket est chat, notifications en temps réel

**Technologies utilisées** :

* Langage de programmation : Java, JavaScript
* Framework ou bibliothèques TOMCAT.10.1.36

Voici la Structurer le projet :  
  
  
**Page de connexion**  
login.jsp :

voici le code:

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html lang="fr">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Connexion</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

background-color: #f4f4f9;

}

h1 {

color: #333;

}

.login-container {

width: 300px;

padding: 20px;

border: 1px solid #ccc;

background-color: #fff;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

}

input[type="text"], input[type="password"] {

width: 100%;

padding: 10px;

margin: 10px 0;

border: 1px solid #ccc;

outline: none;

}

button {

width: 100%;

padding: 10px;

background-color: #28a745;

color: white;

border: none;

cursor: pointer;

}

button:hover {

background-color: #218838;

}

</style>

</head>

<body>

<h1>Connexion</h1>

<div class="login-container">

<form action="projet5" method="post">

<input type="text" name="username" placeholder="Nom d'utilisateur" required>

<input type="password" name="password" placeholder="Mot de passe" required>

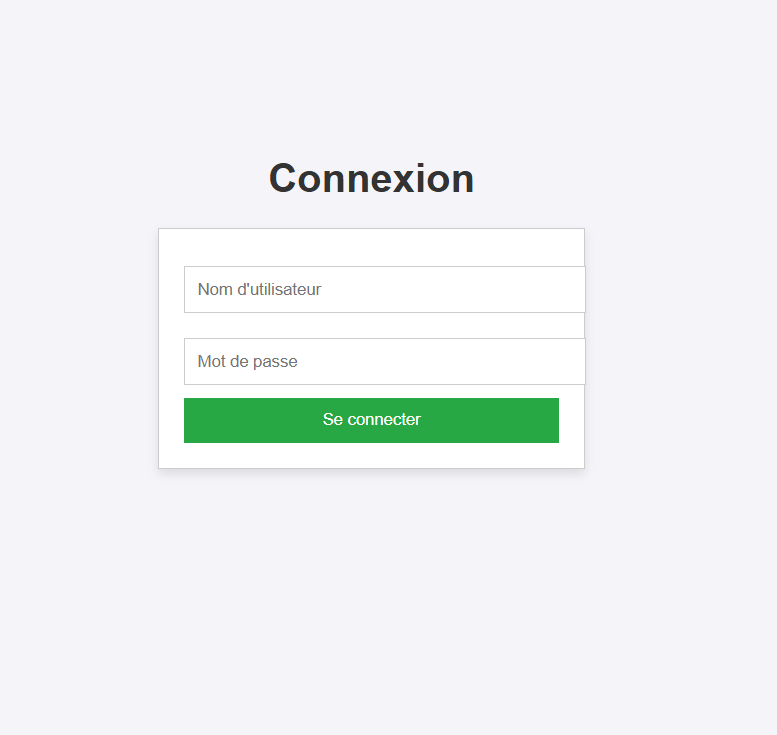
<button type="submit">Se connecter</button>

</form>

</div>

</body>

</html>

voici l'image:  
  
  
  
**Page de déconnexion**  
logout.jsp :  
  
<%@page contentType=*"text/html"* pageEncoding=*"UTF-8"*%>

<!DOCTYPE html>

<html lang=*"fr"*>

<head>

<meta charset=*"UTF-8"*>

<meta name=*"viewport"* content=*"width=device-width, initial-scale=1.0"*>

<title>Déconnexion</title>

<style>

**body** {

font-family: *Arial, sans-serif*;

margin: *0*;

padding: *0*;

display: *flex*;

flex-direction: *column*;

align-items: *center*;

justify-content: *center*;

height: *100vh*;

background-color: *#f4f4f9*;

}

**h1** {

color: *#333*;

}

*.logout-container* {

width: *300px*;

padding: *20px*;

border: *1px solid #ccc*;

background-color: *#fff*;

box-shadow: *0 4px 8px rgba(0, 0, 0, 0.1)*;

}

**button** {

width: *100%*;

padding: *10px*;

background-color: *#dc3545*;

color: *white*;

border: *none*;

cursor: *pointer*;

}

**button***:hover* {

background-color: *#c82333*;

}

</style>

</head>

<body>

<h1>Déconnexion</h1>

<div class=*"logout-container"*>

<form action=*"projet5"* method=*"post"*>

<input type=*"hidden"* name=*"action"* value=*"logout"*>

<button type=*"submit"*>Se déconnecter</button>

</form>

</div>

</body>

</html>  
  
ChatServlet.java :  
  
**package** projet5;

**import** java.util.Collections;

**import** java.util.Set;

**import** java.util.concurrent.ConcurrentHashMap;

**import** jakarta.websocket.OnClose;

**import** jakarta.websocket.OnMessage;

**import** jakarta.websocket.OnOpen;

**import** jakarta.websocket.Session;

**import** jakarta.websocket.server.ServerEndpoint;

@ServerEndpoint("/chat")

**public** **class** ChatServlet {

**private** **static** Set<Session> *userSessions* = Collections.*newSetFromMap*(**new** ConcurrentHashMap<Session, Boolean>());

@OnOpen

**public** **void** onOpen(Session curSession) {

*userSessions*.add(curSession);

System.***out***.println("Nouvelle connexion : " + curSession.getId());

}

@OnClose

**public** **void** onClose(Session curSession) {

*userSessions*.remove(curSession);

System.***out***.println("Connexion fermée : " + curSession.getId());

}

@OnMessage

**public** **void** onMessage(String message, Session userSession) {

System.***out***.println("Message reçu de " + userSession.getId() + " : " + message);

**for** (Session ses : *userSessions*) {

ses.getAsyncRemote().sendText(message);

}

}

}

projet5.java:  
  
package projet5;

import jakarta.servlet.ServletException;

import jakarta.servlet.annotation.WebServlet;

import jakarta.servlet.http.HttpServlet;

import jakarta.servlet.http.HttpServletRequest;

import jakarta.servlet.http.HttpServletResponse;

import jakarta.servlet.http.HttpSession;

import java.io.IOException;

@WebServlet("/projet5")

public class projet5 extends HttpServlet {

private static final long serialVersionUID = 1L;

public projet5() {

super();

}

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String action = request.getParameter("action");

if ("logout".equals(action)) {

HttpSession session = request.getSession(false);

if (session != null) {

session.invalidate();

}

response.sendRedirect("login.jsp");

} else {

response.sendRedirect("login.jsp");

}

}

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

String username = request.getParameter("username");

String password = request.getParameter("password");

String action = request.getParameter("action");

if ("logout".equals(action)) {

HttpSession session = request.getSession(false);

if (session != null) {

session.invalidate();

}

response.sendRedirect("login.jsp");

} else {

// Simuler une authentification simple

if ("admin".equals(username) && "admin".equals(password)) {

HttpSession session = request.getSession();

session.setAttribute("username", username);

response.sendRedirect("chat.jsp");

} else {

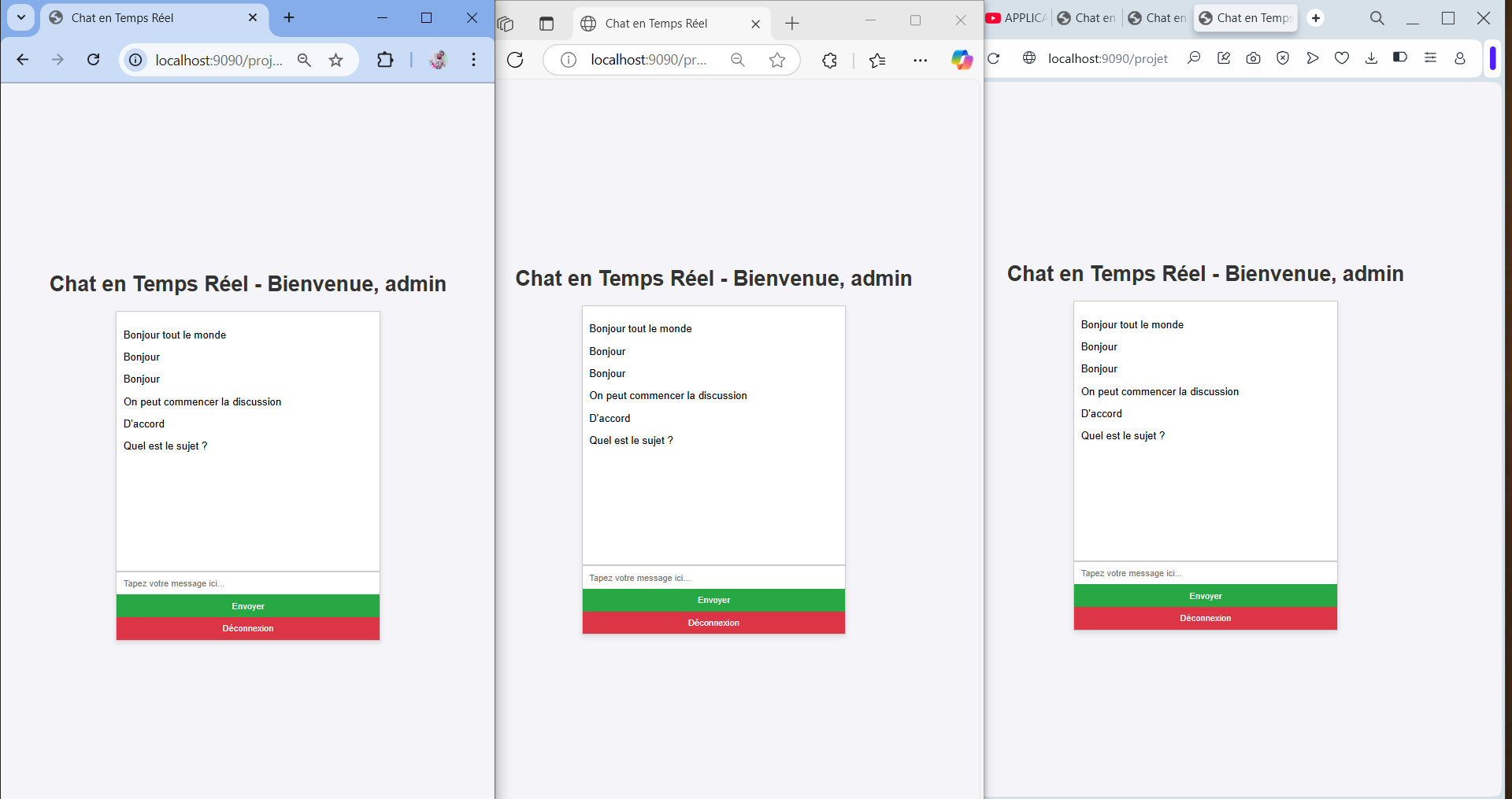
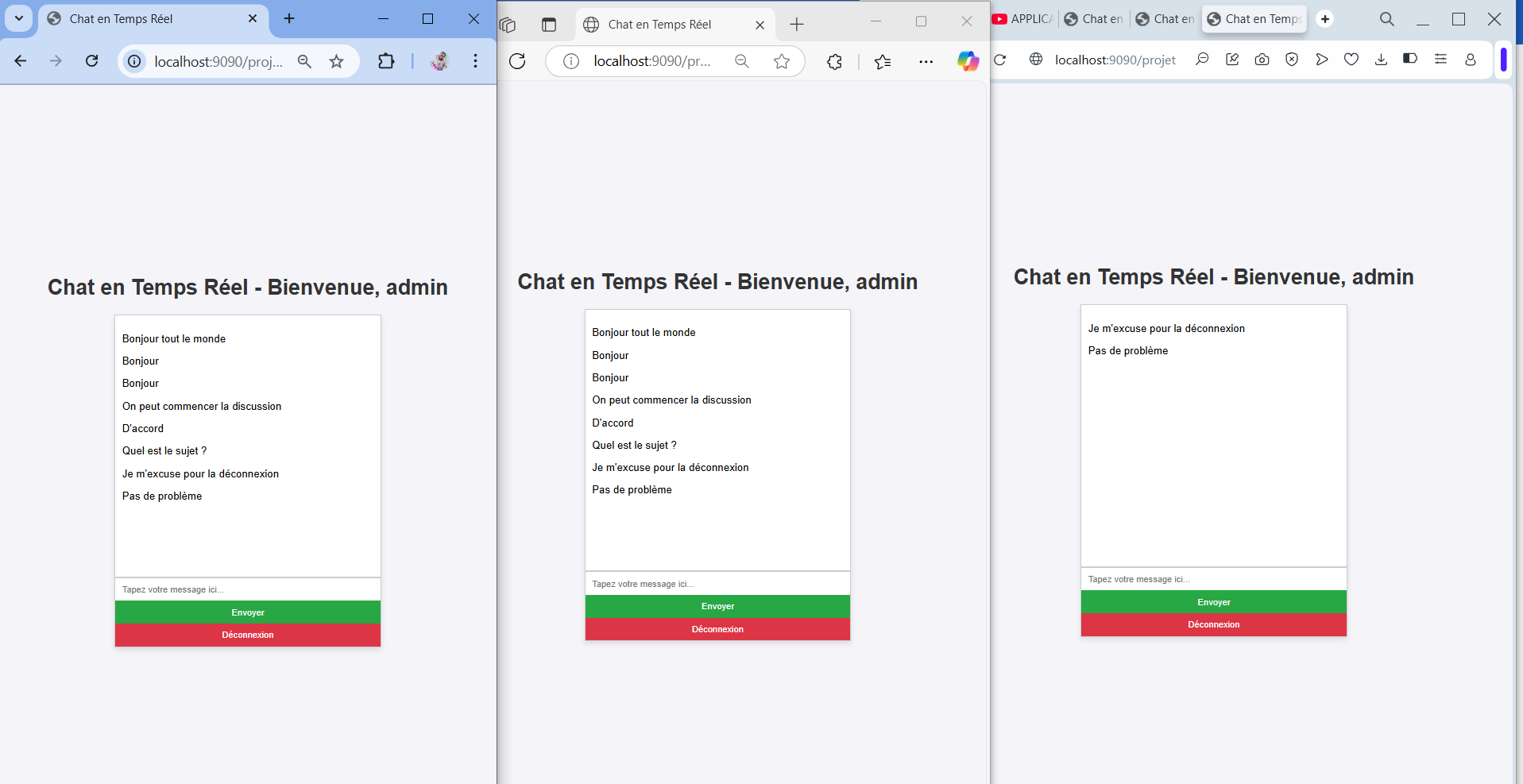
response.sendRedirect("login.jsp");

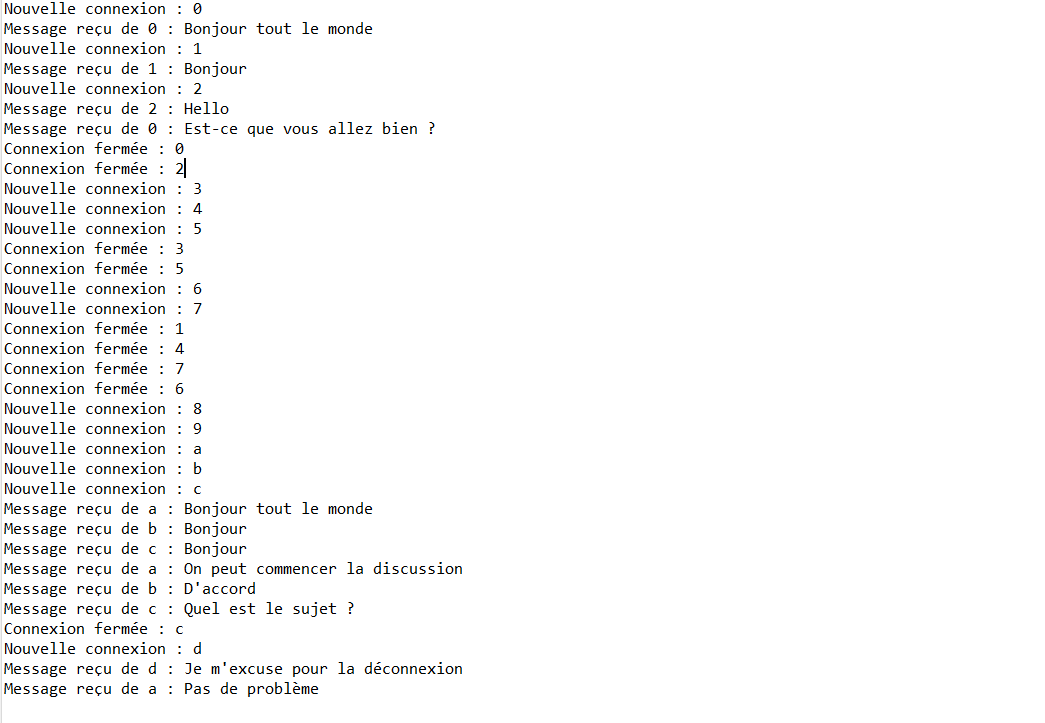
}

}

}

}

**Discussions en temps réel :  
   
  
  
  
Discussions personnalisées :  
**

****