

Aplikacja: Rezerwacja Pokoi

Opis funkcjonalności: sesja użytkownika

Wykorzystane technologie: Java (servlet, jsp), jQuery, Bootstrap

System logowania został oparty na sesjach z wykorzystaniem Cookies (Ciasteczek). Przy ładowaniu stron za pomocą JSP jest sprawdzane czy istnieje Cookie odpowiedzialne za sesję użytkownika (Listing 1).

```
<%
String userName = null;
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for (Cookie cookie : cookies) {
        if (cookie.getName().equals("user")) {
            userName = cookie.getValue();
        }
    }
}
%>
```

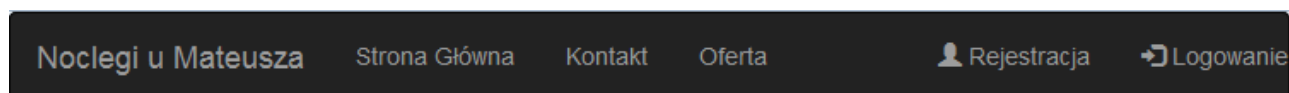
Listing 1: Fragment kodu pliku menu.jsp sprawdzający czy w przeglądarce jest Cookie odpowiedzialne za logowanie.

W zależności od istnienia Cookie tworzone są dwie wersje paska nawigacji za pomocą kodu JSP z listingu 2.

```
<%if (userName != null) {%>
<li><a><span class="glyphicon glyphicon-user"></span> <%= userName%> </a></li>
<li><a id="logowanie2" data-toggle="modal" data-target="#Wylogowanie">
    <span class="glyphicon glyphicon-log-in"></span> Wyloguj </a></li>
<% } else {%>
<li><a id="rejestracja" data-toggle="modal" data-target="#Rejestracja">
    <span class="glyphicon glyphicon-user"></span> Rejestracja</a></li>
<li><a id="logowanie1" data-toggle="modal" data-target="#Logowanie">
    <span class="glyphicon glyphicon-log-in data-"></span> Logowanie </a></li>
<% }%>
```

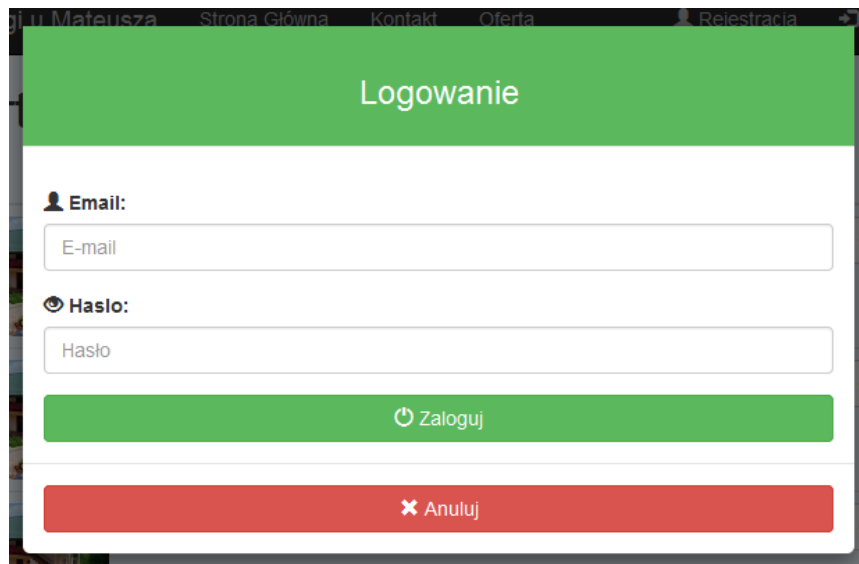
Listing 2: Fragment kodu pliku menu.jsp odpowiedzialny za wyświetlenie odpowiednich opcji na pasku nawigacyjnym.

W wypadku braku odpowiedniego Cookie logowania użytkownikowi zostanie wyświetlony pasek nawigacji z rysunku 1.



Rysunek 1. Pasek nawigacji niezalogowanego użytkownika.

W celu zalogowania się należy na pasku nawigacji nacisnąć opcję „Logowanie”. Wyświetlone zostanie okno utworzone przy pomocy framework Bootstrap zawierające formularz logowania (rysunek 2).



Rysunek 2. Okno logowania.

Po uzupełnieniu pól oraz wciśnięciu przycisku „Zaloguj” zostaje uruchomiony skrypt jQuery (listing 3) zastępujący standardowe przesyłanie formularza. Do zmiennych *email* oraz *hasło* pobierane są wartości odpowiednich pól, które następnie są wykorzystane w żądaniu do serwletu *logowanie.java*, wykonanego za pomocą metody *get* funkcji AJAX zaimplementowanej w jQuery. Gdy logowanie się powiedzie to nastąpi odświeżenie strony.

```
$(document).on("click", "#logowanieSubmit", function (event) {
    event.preventDefault();
    var email = $('#emailL').val();
    var haslo = $('#hasloL').val();
    $.get("logowanie", {
        action: action,
        email: email,
        haslo: haslo
    }, function (responseText) {
        var logowanie = $.trim(responseText);
        if (logowanie != '0')
            $("#komunikatL").html(responseText);
        else
            location.reload();
    });
});
```

Listing 3: Skrypt jQuery w *menu.jsp* odpowiedzialny za przekazywanie danych logowania.

Przy otrzymaniu żądania serwlet *logowanie.java* sprawdza w bazie danych czy istnieje użytkownik o podanych parametrach email oraz hasło (listing 4).

```
String email = request.getParameter("email");
String haslo = request.getParameter("haslo");
//połączenie z bazą danych
Connection con = PolaczenieDB.getConnection();
//szukanie użytkownika
PreparedStatement st = con.prepareStatement("select * from Uzytkownik where email=? and haslo=?");
st.setString(1, email);
st.setString(2, haslo);
ResultSet rs=st.executeQuery();
status=rs.next();
```

Listing 4: Fragment kodu w *logowanie.java* sprawdzający, czy użytkownik jest w bazie danych.

Gdy użytkownik zostanie znaleziony w bazie danych to serwlet *logowanie.java* tworzy Cookie, które następnie jest wysyłane do przeglądarki (listing 5). Poprawne logowanie

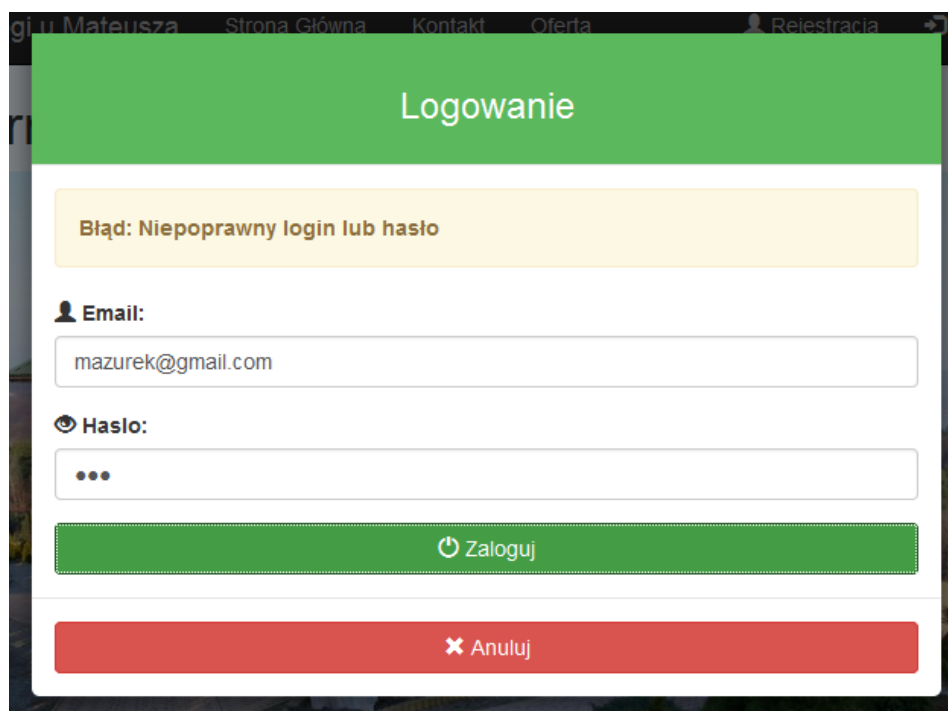
```
if(status){
    Cookie logowanie = new Cookie ("user", email);
    logowanie.setMaxAge(60*10);
    response.addCookie(logowanie);
    out.println("0");
}
```

Listing 5: Fragment kodu w logowanie.java tworzący Cookie.

Jeżeli podane dane były niepoprawne to serwlet *logowanie.java* wysyła wiadomość (listing 6) która zostanie wyświetlona w oknie logowania (rysunek 3) poprzez skrypt z listingu 3.

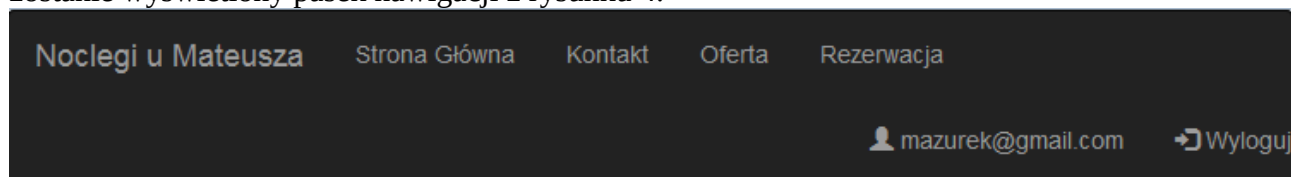
```
else{
    out.println("<div class =\"alert alert-warning\">");
    out.println("<strong>Błąd:<strong> Niepoprawny login lub
hasło</font>");
    out.println("</div>");
}
```

Listing 6: Fragment kodu w logowanie.java odpowiedzialny za treść błędu.



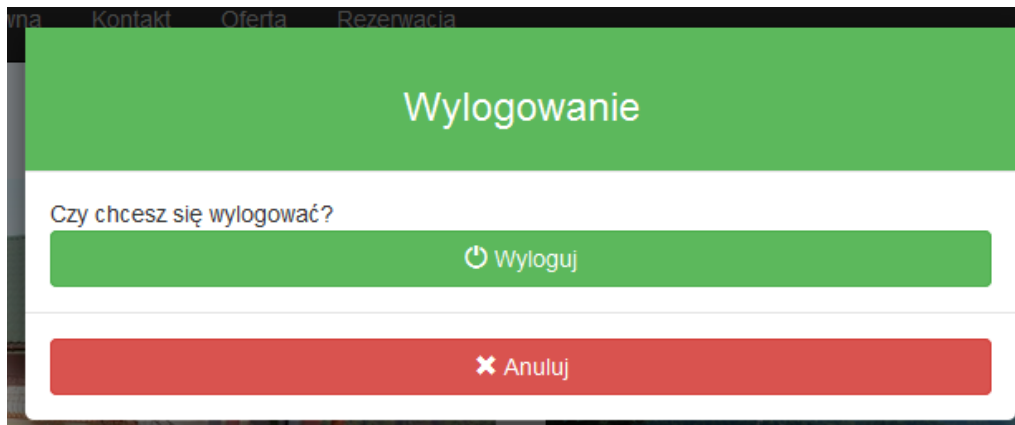
Rysunek 3. Okno logowania z wyświetlonym błędem.

Jeżeli proces logowania przebiegł pomyślnie oraz w przeglądarce znajduje się Cookie odpowiedzialne za logowanie to po odświeżeniu strony przez skrypt z listingu 3 użytkownikowi zostanie wyświetlony pasek nawigacji z rysunku 4.



Rysunek 4. Pasek nawigacji zalogowanego użytkownika.

Wciśnięcie przycisku wyloguj spowoduje wyświetlenie okna z rysunku 5.



Rysunek 5. Okno wylogowania.

Wciśnięcie przycisku „Wyloguj” spowoduje wysłanie żądania do serwletu *wylogowanie.java*. Wylogowanie polega na pobraniu odpowiedniego ciasteczka z przeglądarki, ustawienie jego czasu ważności na 0 oraz przesłanie ciasteczka do przeglądarki (listing 5). Zmodyfikowane ciasteczko zostanie natychmiast usunięte z przeglądarki.

```
Cookie loginCookie = null;
Cookie[] cookies = request.getCookies();
if (cookies != null) {
    for (Cookie cookie : cookies) {
        if (cookie.getName().equals("user")) {
            loginCookie = cookie;
            break;
        }
    }
}
if (loginCookie != null) {
    loginCookie.setMaxAge(0);
    response.addCookie(loginCookie);
}
```

Listing 5: Fragment kodu pobierający Cookie oraz zmieniający jego czas na 0.