

EE417 Assignment 3 – Report

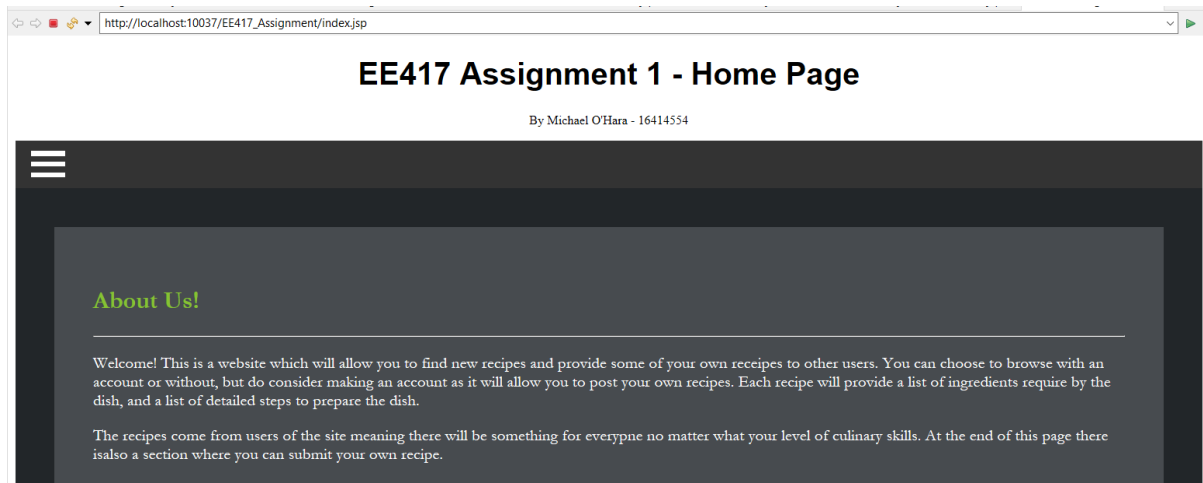
By Michael O'Hara

Convert Pages to JSP

I converted 5 pages to JSPs these pages are index, register, breakfast, lunch, and dinner. These pages were converted by simply creating JSP files inside of eclipse and adding the html code to the file alone with this line:

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>
```

As shown below is my index running as a JSP page rather than a static html page.



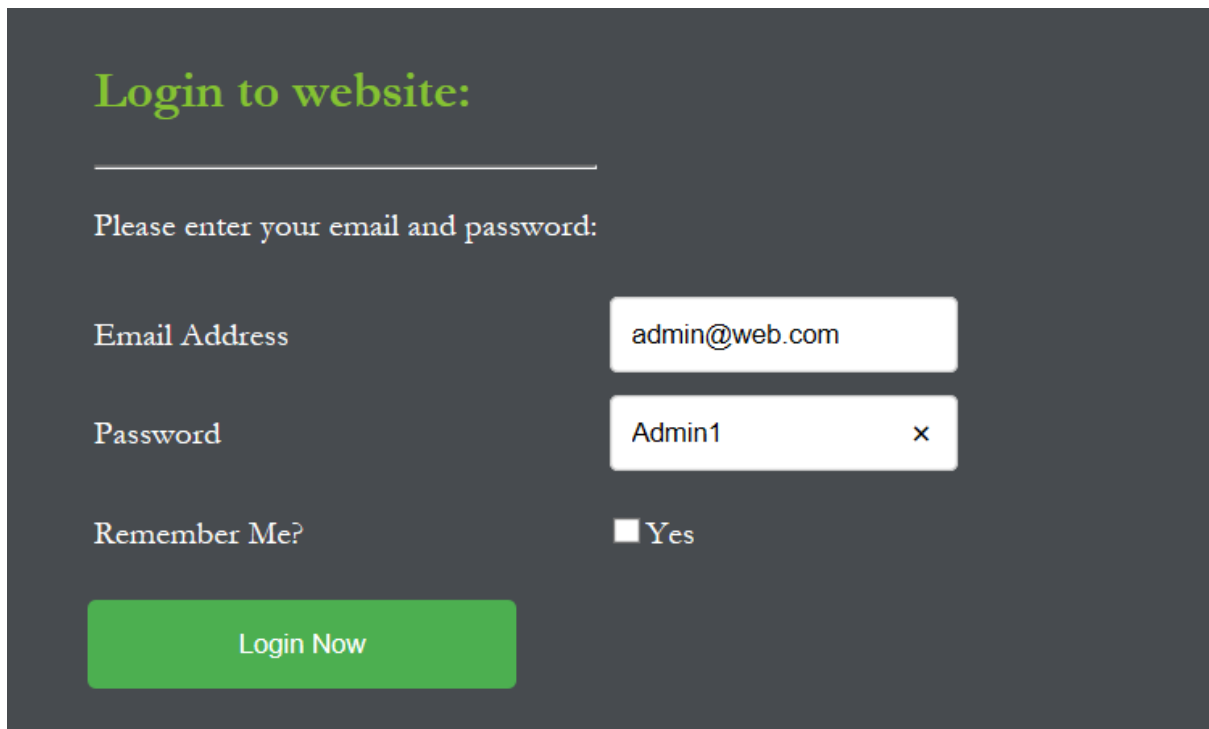
Login System for Website:

In previous versions of the assignment, I had already created a login page which meant all I had to do was add the functionality to the page. This is done using servlets in conjunction with called user.java.

```
User user = validateUser(request.getParameter("email"), request.getParameter("password"));
if(user!=null) {
    HttpSession session = request.getSession();
    session.setAttribute("theUser", user);
    if(user.isAdmin().equals(true)) {
        //out.print("<script type='text/javascript'>document.getElementById(\"admin\").style.display = \"block\";</script>");
        request.getRequestDispatcher("/Admin/AdminPanel.jsp").forward(request, response);
    }
    else {
        request.getRequestDispatcher("index.jsp").forward(request, response);
    }
}
else {
    response.sendRedirect("login.html");
}
out.close();
```

Above is the code snippet of the servlet which decides which page the user will be redirected to on login. If the Boolean value isAdmin() in the user class is set to true, then the user will be redirected

to the AdminPanel.jsp from there they can access a number of different pages such as a page to change passwords for other users. But if the user is not an admin, they will be redirected to the index page of the site. If the credentials, the user enters are incorrect it will redirect to the "login.html" page.



The image shows a login form on a dark grey background. At the top, the text "Login to website:" is written in a green, serif font. Below this is a horizontal line. The instruction "Please enter your email and password:" is in a light blue, serif font. There are two input fields: "Email Address" with the value "admin@web.com" and "Password" with the value "Admin1". To the right of the password field is a small 'x' icon. Below the password field is a "Remember Me?" checkbox, which is currently unchecked, followed by the text "Yes". At the bottom is a large green button with the text "Login Now" in white.

http://localhost:10037/EE417_Assignment/LoginServlet

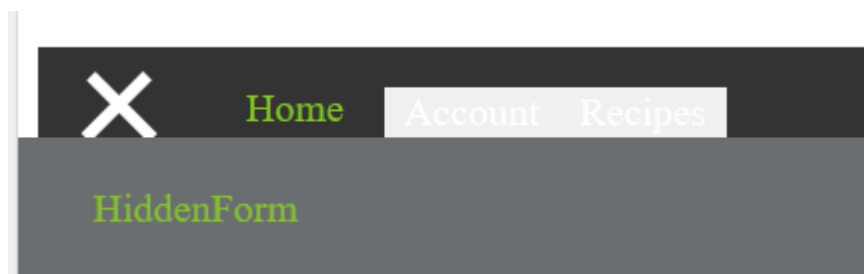
EE417 Assignment 1 - Home Page

By Michael O'Hara - 16414554



Welcome to the Admin Panel for the website. From here you can access functions which regular users can't. You are able to reset usernames and passwords for regular users should they forget their login details. This can be done from the "Hidden Form" page in the navbar, which regular users cannot see.

Above is a small section of the webpage which the admin user will be redirected to upon logging in and should they click on the menu bar they can see links that a non admin user cannot see. (This screen grab was taken before adding the other session management functions).



Below is the login details for a non admin member of the website and then the result of the login, that is the redirect to the index page. As you can see in the search bar the URL mentions the Login

Servlet.

The image shows two screenshots of a web application. The top screenshot is a login page with a dark grey background. It features a green heading 'Login to website:', a white line, and a prompt 'Please enter your email and password:'. Below this are two input fields: 'Email Address' with the value 'jon@web.com' and a clear button 'x', and 'Password' with the value 'Mypass1'. There is a checkbox for 'Remember Me?' which is checked, with the label 'Yes' next to it. A green 'Login Now' button is at the bottom. The bottom screenshot is a browser window showing the 'EE417 Assignment 1 - Home Page'. The browser address bar shows 'http://localhost:10037/EE417_Assignment/LoginServlet'. The page has a dark header with a hamburger menu icon. Below the header, there is a section titled 'About Us!' in green. The text below reads: 'Welcome! This is a website which will allow you to find new recipes and provide some of your own receipes to other users. You can choose to browse with an account or without, but do consider making an account as it will allow you to post your own recipes. Each recipe will provide a list of ingredients require by the dish, and a list of detailed steps to prepare the dish.' and 'The recipes come from users of the site meaning there will be something for everypne no matter what your level of culinary skills. At the end of this page there isalso a section where you can submit your own recipe.'

Login to website:

Please enter your email and password:

Email Address jon@web.com x

Password Mypass1

Remember Me? ☒ Yes

Login Now

EE417 Assignment 1 - Home Page

By Michael O'Hara - 16414554

About Us!

Welcome! This is a website which will allow you to find new recipes and provide some of your own receipes to other users. You can choose to browse with an account or without, but do consider making an account as it will allow you to post your own recipes. Each recipe will provide a list of ingredients require by the dish, and a list of detailed steps to prepare the dish.

The recipes come from users of the site meaning there will be something for everypne no matter what your level of culinary skills. At the end of this page there isalso a section where you can submit your own recipe.

Session Management Techniques:

Hidden Form:

The first implemented technique was the hidden form technique. This was done using a html form and 2 servlets. Code snippets of which can seen below.

HiddenForm1:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    PrintWriter out = response.getWriter();
    String UN = request.getParameter("username");

    out.println("<html><head><title>Hidden Form</title></head>");
    out.println("<body><h1>EE417 Assignment 1 - Hidden Form Page</h1><p>By Michael O'Hara - 16414554</p>");
    out.println("<h1>Currently editing: " + UN + " </h1>");
    out.println("<form method='post' " + "action='\"HiddenForm2\"' name='\"myform\"'>");
    out.println("<br/>Now enter your new password: ");
    out.println("<input name='\"password\"'>");

    out.println("<input type='\"hidden\"' name='\"username\"' " + " value='\"" + UN + "\">");
    out.println("<br/><br/><input type='\"submit\"' value='\"Submit Form\"'>");
    out.println("<input type='\"reset\"' value='\"Reset\"'>");
    out.println("</body><footer><p>Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1</p> </footer></html>");
    out.close();
}
```

In the above snippet you can see the code produces a form hidden from the user but not hidden from the server and passes the information from this to the second servlet seen below.

HiddenForm2:

```
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    doGet(request, response);

    response.setContentType("text/html");
    PrintWriter out = response.getWriter();


    String UN = request.getParameter("username");
    String PW = request.getParameter("password");

    out.println("<html><head><title>Hidden Forms Example</title></head>");
    out.println("<body><h1>EE417 Assignment 1 - Hidden Form Page</h1><p>By Michael O'Hara - 16414554</p>");
    out.println("<h1>You have selected username = " + UN + "</h1>");
    out.println("<br/><br/><p>The new password for this user is" + PW + "</p>");
    out.println("</body><footer><p>Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1</p> </footer></html>");
    out.close();
}
```

The next number of screen grabs are the output of this technique:

EE417 Assignment 1 - Home Page

By Michael O'Hara - 16414554



Change Account Details:

Please enter your username:

Username:

Submit

Above is the page the user will be redirected to when clicking on the “Hidden Form” link. Below is the page the first servlet will create when the user clicks the submit button.

EE417 Assignment 1 - Hidden Form Page

By Michael O'Hara - 16414554

Currently editing: admin

Now enter your new password:

Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1

Below is the result of clicking submit on the first servlet page the and as you can see it uses the details from the first page to output onto the second page. Unfortunately I was unable to get the link to my style sheet to correctly display the styles on these pages

Served at: /EE417_Assignment

EE417 Assignment 1 - Hidden Form Page 2

By Michael O'Hara - 16414554

You have selected username = admin

The new password for this user is: newpass1

An Automated email shall be sent to the email associated with this account to note the changes to the password

PLEASE NOTE: This is only a dummy change as there is no database behind the application yet.

Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1

[URL Rewriting:](#)

Below is the code snippet which is used to rewrite the URL by appending the username the entered by the user to the end of the URL. This page can only be accessed if the user is logged as an admin user otherwise the link to the page is hidden.

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    // TODO Auto-generated method stub
    response.getWriter().append("Served at: ").append(request.getContextPath());
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

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

    out.println("<html><head><title>URL Rewriting Example</title></head>");
    out.println("<body><h1>Rewrite URL using Username</h1>");

    if (request.getParameter("username") == null) {
        out.println("User logged in - generating unique sessionID");
        UN = "user1";
    }
    else {
        out.println("User logged in getting username");
    }

    out.println("<br><br>URL end = " + UN);
    out.println("<br><br>As seen in the URL the users username is appended to the end of the URL. This can be used to determine whihc "
        + "user is logged in at any time.");
    out.println("</body></html>");
    out.close();
}

```

Below is proof of the URL being changed to append the users entered username.

HTTP Sessions:

Below is the code snippet of a servlet that will track the amount of times a logged in user visits that specific page and outputs it to the screen along with the users username.

```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    response.getWriter().append("Served at: ").append(request.getContextPath());
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();

    HttpSession session = request.getSession(true);
    String UN = request.getParameter("username");

    out.println("<html><head><title>HTTP Sessions</title></head>");
    out.println("<body><h1>Track a users session</h1>");
    out.println("<br>Session ID = " + session.getId());

    Integer count = (Integer) session.getAttribute("count.value");
    if (count == null) {
        count = 1;
    }
    else {
        count = count + 1;
    }

    session.setAttribute("count.value", count);
    out.println("Selected user is: " + UN);
    out.println("<br><br>" + UN + " has visited this page " + count + " time(s)");
    out.println("</body><footer><p>Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1</p></footer></html>");
    out.close();
}

```

Below are screen grabs of the result of visiting this page a number of times and as you can see the integer displaying the amount of visits is increasing.

http://localhost:10037/EE417_Assignment/HTTP?username=admin

Served at: /EE417_Assignment

Track a users session

Session ID = 660419A65079CAB6C317CE640B592D22 Selected user is: admin

admin has visited this page 1 time(s)

http://localhost:10037/EE417_Assignment/HTTP?username=admin

Served at: /EE417_Assignment

EE417 Assignment 1 - Home Page

By Michael O Hara - 16414554

Track a users session

Session ID = 660419A65079CAB6C317CE640B592D22 Selected user is: admin

admin has visited this page 7 time(s)

Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1

Cookies:

Using the cookie example from loop the last session management technique I tried to implement was getting cookies however I was unable to get it to be implemented correctly. It would output the user that was logged in and session ID but would not display the cookies correctly. Output can be seen below:

Served at: /EE417_Assignment

EE417 Assignment 1 - Home Page

By Michael O Hara - 16414554

Getting Cookies

Getting cookies for: admin

Name: JSESSIONID

Value: 660419A65079CAB6C317CE640B592D22

Copyright © 2021 Michael O Hara EE417 Assignment 1 Part 1