



# MIDTERM EXAM REPORT

NGUYEN ANH DUNG'S PERSONAL WEB  
APPLICATION

## Abstract

Nguyen Anh Dung's Personal Web Application is the author's midterm assignment project of the course Web Application Development taught by Dr. Nguyen Van Sinh. The content of this web application is mainly the author's personal information, hobbies, interest, etc as requested by the lecturer.

## Table of Contents

<b>I.</b>	<b>Introduction .....</b>	<b>3</b>
1)	Introduction to the project .....	3
2)	Author .....	3
<b>II.</b>	<b>Description of the personal web application .....</b>	<b>3</b>
1)	Assumption.....	3
2)	Content.....	4
3)	Tools .....	4
4)	Techniques.....	4
5)	Database .....	4
<b>III.</b>	<b>Design, source codes and demos .....</b>	<b>5</b>
1)	Navigation bar and footer .....	5
2)	Homepage.....	5
a.)	Explanation.....	5
b.)	Source code.....	6
c.)	Demo .....	7
3)	About page.....	8
a.)	Explanation.....	8
b.)	Source code .....	8
c.)	Demo .....	9
4)	Projects page.....	12
a.)	Explanation.....	12
b.)	Source code .....	12
c.)	Demo .....	14
5)	Hobbies page .....	16
a.)	Explanation.....	16
b.)	Source code .....	16
c.)	Demo .....	18
6)	Contacts page.....	19
a.)	Explanation.....	19
b.)	Source code .....	19

c. Demo.....	21
7) Review page .....	22
a. Explanation.....	22
b. Source code .....	23
c. Demo.....	24
d. Servlet (review.java). ....	26
8) Direct page.....	28
a. Explanation.....	29
b. Source code .....	29
c. Demo.....	30
d. Servlet (dms.java). ....	33
9) Structural design .....	35
10) Cascading Style Sheets.....	35
 <b>IV. Use case diagram .....</b>	 <b>39</b>
1) Diagram.....	39
2) Explanation.....	40
 <b>V. Login and registration page.....</b>	 <b>40</b>
1) Assumption .....	40
2) LoginPage.jsp and CheckLogin Servlet.....	41
a. Source code .....	41
b. Demo.....	43
3) Registration.jsp .....	46
a. Source code .....	46
b. Demo.....	48
4) Store Servlet.....	49
a. Source code .....	49
b. Demo.....	50
5) JavaBean.....	52
a. Source code .....	52
b. Demo.....	54
 <b>VI. References .....</b>	 <b>56</b>

# I. INTRODUCTION

## 1) Introduction to the project:

This project is part of my midterm examination assignment given by Dr.Nguyen Van Sinh in the course Web Application Development. The premise of this assignment is to design and test a web application that can handle basic tasks like displaying text, images, videos as well as taking, handling and displaying input data on the web application using html , css, jsp , servlet, javabean and database. Along side basic functionalities, the web application also fulfilled some requirements given by the lecturer, which will be discussed in later parts of this report.

## 2) Author:

- *Full name:* Nguyễn Anh Dũng
- *Student ID:* ITITIU17073
- *Lecturer:* Dr. Nguyễn Văn Sinh
- *Course:* Web Application Development (Thursday morning class)

# II. DESCRIPTION OF THE PERSONAL WEB APPLICATION

## 1) Assumption:

- Nguyen Anh Dung's personal website is a collection of several different web pages (having both static and dynamic pages) that display my real personal information (except my student ID which is included in this report).
- All of the web pages are created using the knowledge that I learn in the Web Application Development course, combining with some expertise tips and tricks that I can find on Google (but not anything outside of what I have learnt).
- There will be links to my social medias (Facebook, Instagram) in some pages that will require internet connection.
- Some images and texts are outsourcing from Google for free so if any of them violating copyright issues please note that this project is only for educational purposes and I do not own any of them. However, the pictures of me, my personal contacts details and videos/images on the Project page are all mine (more on the source codes and demos).
- There will be the appearance of the login and registration functionalities in question 4, but for part II to IV we will not be counting them as part of the project.
- Some parts of the projects are solely for fulfilling the assignment's requirements so it may not look professionally made or well design, but it does fulfill what I intended.

- As I am still learning on web application development, please note that this project is not very professional, having several problems that I will point out in later parts.

## 2) Content:

- Nguyen Anh Dung's personal website contains 5 html pages that are static web pages with the main purpose of displaying texts, images and videos and 2 jsp pages with 2 servlets that acts as dynamic pages.
- The main contents of the html pages are my own actual information, contact details, some of my hobbies, projects and an index homepage for welcoming user.
- The 2 jsp pages are used for the purpose of interacting with the user: 1 page is to get user's text and display, the other one display some movie reviews on user's button click. Both tasks are done in conjunction with 2 servlets that handle data processing.
- Other than the actual contents of the pages, all pages will have a navigation bar on top and a footer at the bottom. The navigation bar will have an IU logo on it.

## 3) Tools:

- Editor: Netbeans IDE
- Server: Apache Tomcat 8.5
- Programming language: HTML5, CSS3 for front-end development. Java servlet, Java Server Page for back-end development. There's also a use of Java beans for the Login and Registration questions.
- Outsourcing method: Google.com, stackoverflow, etc.

## 4) Techniques:

- The pages are design by using html in conjunction with css to layout and format the contents. The main technique for formatting containers and objects in a page is using tables (tables and tables within tables) with variety of backgrounds and sizes.
- Input data are handle using <form> tag on jsp pages and pass to Java servlets for data handling.
- Each and every page will have a navigation bar that grant access to other pages and a credit footer with my name on it. The navigation bar contains an IU logo as well as a collection of links that lead to other pages and the footer is just basic text.
- All pages are design with a format: (navbar) – (main content) – (footer). The main contain, as mention above, is layout with tables in variety of sizes and a picture of International University as a background.

## 5) Database:

- As I would preferably use MySQL (because I already learnt to use it in another course) to manage database, this project doesn't require much data so I will not be using any database. Although there's an exception of the Review page that can benefit from using a database, but I just use a collection of static data in a servlet (more on this in the demo of Review page).

### III. DESIGN, SOURCE CODES AND DEMOS

#### 1) Navigation bar and footer:

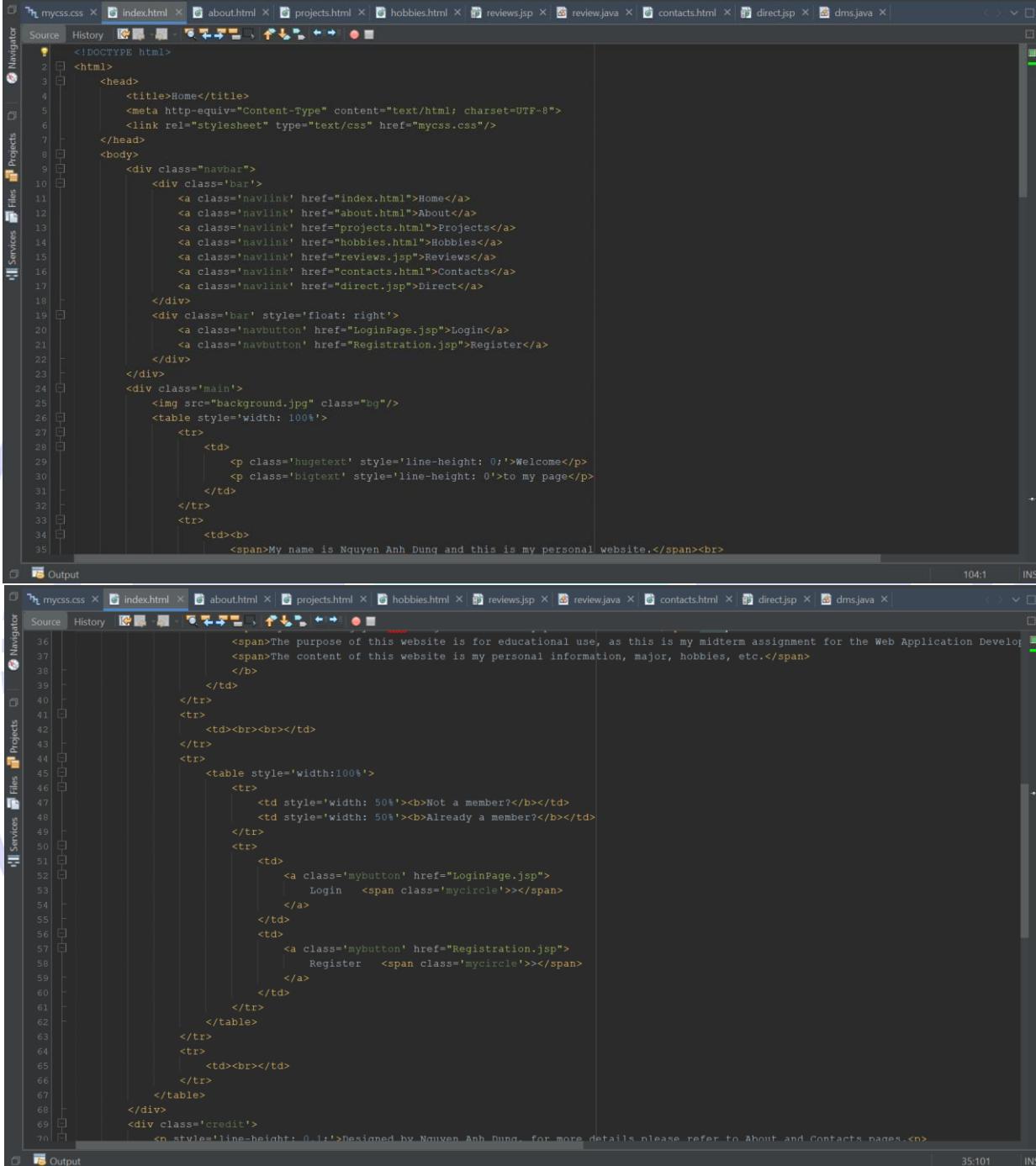
As mentioned before, every page will be designed with a format (navbar) – (main content) – (footer). The navbar (stands for navigation bar) is a container as wide as the page itself and contains an IU logo, the links to other pages on the left and 2 buttons (login and register) on the right with the buttons themselves being just links with borders. The footer is also a container as wide as the page containing a text in the middle with my name on it for citation. The navbar is located at the top of the page and the footer is located at the bottom. Navbar is created using the classes .navbar, .bar, .navlink and .navbutton; but the footer doesn't have anything special. Both of them are colored maroon to match IU's main theme color and texts/links are in white.

As they are in every page, no demo or source code will be attached here because they are in every other parts of this report (see other parts and css).

#### 2) Homepage:

##### a) *Explanation:*

- Content: The Home page (index.html) is the cover/homepage/index page of the website, it's the site that you first visit when you enter the website, the beginning of the project. The main content of this page is a greeting message, a description on what the website is about and 2 buttons that take you to the login page and registration page (but these 2 pages are irrelevant to the project for now so they're just there for decoration).
- Design: Other than the navbar and footer, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a table in the middle for layout and the texts are styled with .bigtext, .hugetext classes.

*b) Source code:*


The screenshot shows a Java IDE interface with two tabs open: 'mycss.css' and 'index.html'. The 'index.html' tab is active, displaying the following HTML code:

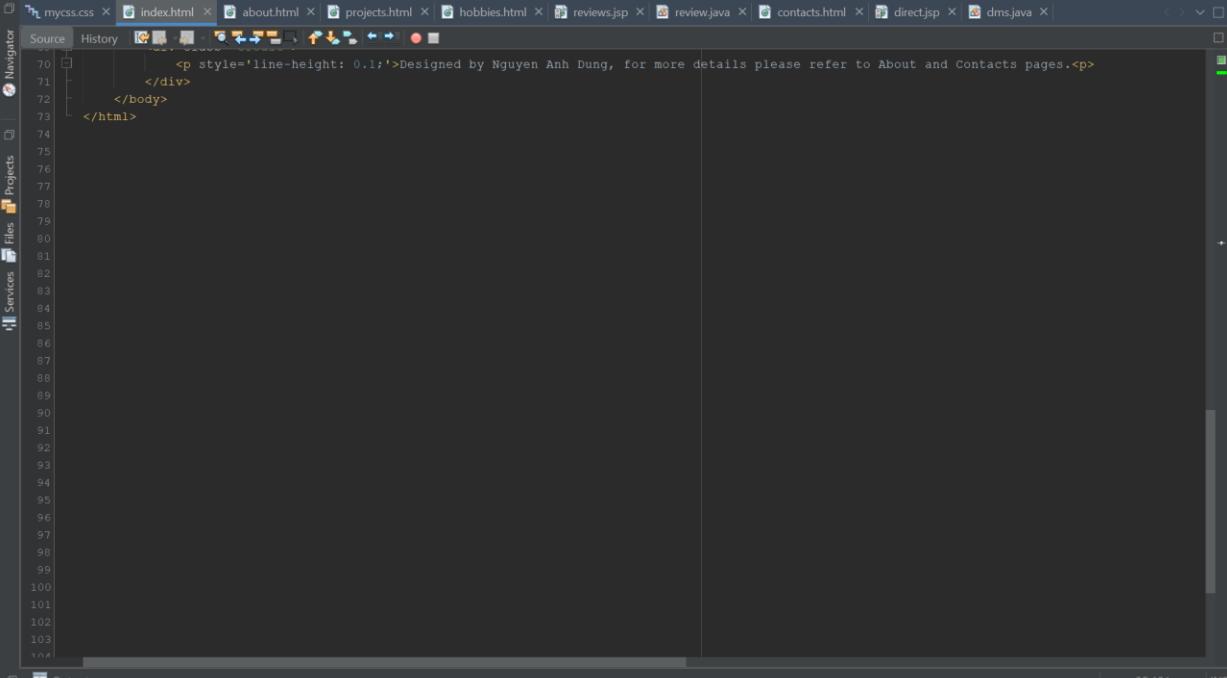
```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Home</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <link rel="stylesheet" type="text/css" href="mycss.css"/>
7   </head>
8   <body>
9     <div class="navbar">
10       <div class="bar">
11         <a class="navlink" href="index.html">Home</a>
12         <a class="navlink" href="about.html">About</a>
13         <a class="navlink" href="projects.html">Projects</a>
14         <a class="navlink" href="hobbies.html">Hobbies</a>
15         <a class="navlink" href="reviews.jsp">Reviews</a>
16         <a class="navlink" href="contacts.html">Contacts</a>
17         <a class="navlink" href="direct.jsp">Direct</a>
18       </div>
19       <div class="bar" style="float: right">
20         <a class="navbutton" href="LoginPage.jsp">Login</a>
21         <a class="navbutton" href="Registration.jsp">Register</a>
22       </div>
23     </div>
24     <div class="main">
25       
26       <table style='width: 100%'>
27         <tr>
28           <td>
29             <p class="hugetext" style='line-height: 0;'>Welcome</p>
30             <p class="bigtext" style='line-height: 0;'>to my page</p>
31           </td>
32         </tr>
33         <tr>
34           <td><b>
35             <span>My name is Nguyen Anh Dung and this is my personal website.</span><br>

```

The 'Output' tab is also visible at the bottom of the IDE.

Figure 2.1 and 2.2: Source codes for index.html



```

70      <p style='line-height: 0.1;'>Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
71      </div>
72    </body>
73  </html>
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103

```

Figure 2.3: Source codes for index.html

### c) Demo:

You can see the messages in the middle, the opaque background, the navbar and footer. When you click on to the links in navbar they will take you to the corresponding pages (Home will take you to this same page), same thing for the login and registration buttons.

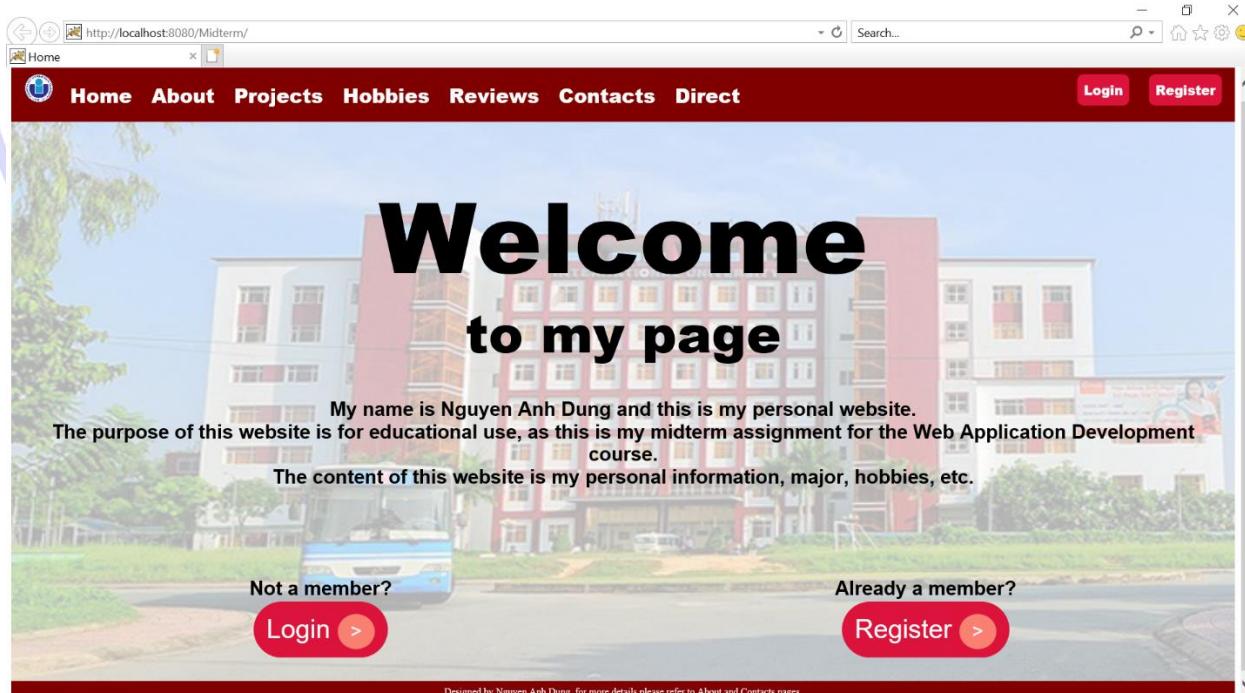


Figure 2.4: Demo of the index/home page

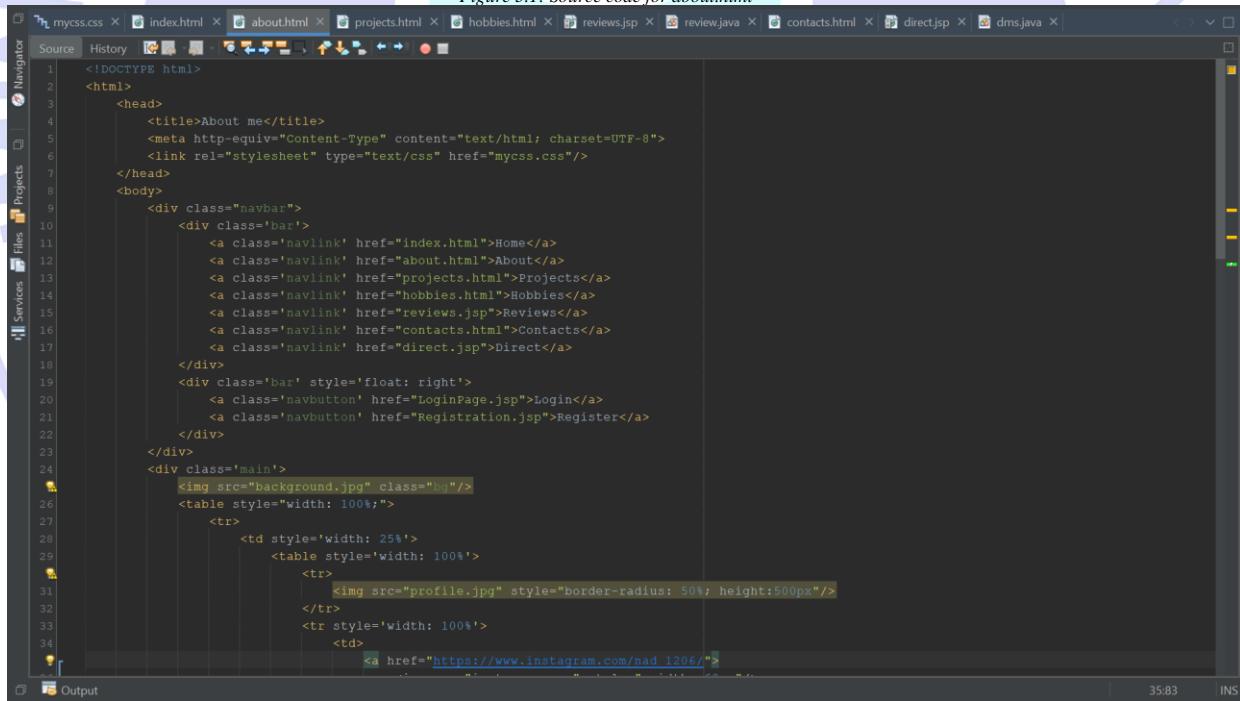
### 3) About page:

#### a) Explanation:

- Content: The About page (about.html) is the site where I show all my personal information, my pictures and my social media. The main content of this page is a title of the page, a real picture of me, 2 images that are links that leads to my actual Instagram and Facebook when you click on them on the left, as well as a table that contains my information: my name, date of birth, gender, real address, occupation, education and employment on the right.
- Design: Other than the navbar, the footer and the title established, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a main table taking 100% of the middle portion, containing 2 smaller tables 1 for pictures and 2 links that takes 25% width of the main table, 1 for information that takes 75% width of the main table. The tables are styled with css (see css part) and the information table is color salmon with some rows being lightsalmon. The title is styled with the class .bigtext in css.

#### b) Source code:

Figure 3.1: Source code for about.html



```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>About me</title>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <link rel="stylesheet" type="text/css" href="mycss.css"/>
7      </head>
8      <body>
9          <div class="navbar">
10             <div class="bar">
11                 <a class="navlink" href="index.html">Home</a>
12                 <a class="navlink" href="about.html">About</a>
13                 <a class="navlink" href="projects.html">Projects</a>
14                 <a class="navlink" href="hobbies.html">Hobbies</a>
15                 <a class="navlink" href="reviews.jsp">Reviews</a>
16                 <a class="navlink" href="contacts.html">Contacts</a>
17                 <a class="navlink" href="direct.jsp">Direct</a>
18             </div>
19             <div class="bar" style='float: right;'>
20                 <a class="navbutton" href="LoginPage.jsp">Login</a>
21                 <a class="navbutton" href="Registration.jsp">Register</a>
22             </div>
23         </div>
24         <div class="main">
25             
26             <table style="width: 100%;">
27                 <tr>
28                     <td style="width: 25%">
29                         <table style="width: 100%">
30                             <tr>
31                                 <td style="width: 100%">
32                                     <a href="https://www.instagram.com/nad_1206/">
33                                         
34                                     </a>
35                                 </td>
36                             </tr>
37                         </table>
38                     </td>
39                     <td style="width: 75%">
40                         <table border="1" style="width: 100%; border-collapse: collapse;">
41                             <tr style="background-color: #f2e0d2;">
42                                 <td style="width: 50%; padding: 5px; text-align: center;">
43                                     Name: Nadia
44                                 </td>
45                                 <td style="width: 50%; padding: 5px; text-align: center;">
46                                     Date of Birth: 12/06/1998
47                                 </td>
48                             </tr>
49                             <tr style="background-color: #ffccbc;">
50                                 <td style="width: 50%; padding: 5px; text-align: center;">
51                                     Gender: Female
52                                 </td>
53                                 <td style="width: 50%; padding: 5px; text-align: center;">
54                                     Address: 123 Main Street, Anytown, USA
55                                 </td>
56                             </tr>
57                             <tr style="background-color: #ffccbc;">
58                                 <td style="width: 50%; padding: 5px; text-align: center;">
59                                     Occupation: Student
60                                 </td>
61                                 <td style="width: 50%; padding: 5px; text-align: center;">
62                                     Education: High School Graduate
63                                 </td>
64                             </tr>
65                         </table>
66                     </td>
67                 </tr>
68             </table>
69         </div>
70     </body>
71 </html>

```

```

36         
37     </a>
38   </td>
39   <td>
40     <a href="https://www.facebook.com/nguyen.dunganh.754">
41       
42     </a>
43   </td>
44 </tr>
45 </table>
46 </td>
47 <td style='width: 75%'>
48   <h1 class="bigtext">Personal Information</h1><br>
49   <table style='width: 100%;text-align: left; background-color: salmon; border-radius: 10px'>
50     <tr>
51       <td><b>Full name</b></td>
52       <td>Nguyen Anh Dung</td>
53     </tr>
54     <tr style='background-color: lightsalmon;'>
55       <td><b>Gender</b></td>
56       <td>Male</td>
57     </tr>
58     <tr>
59       <td><b>Date of birth</b></td>
60       <td>June 12th, 1999</td>
61     </tr>
62     <tr style='background-color: lightsalmon;'>
63       <td><b>Address</b></td>
64       <td>15/10A C1 street, Tan Binh district, Ho Chi Minh city</td>
65     </tr>
66     <tr>
67       <td><b>Occupation</b></td>
68       <td>Student at International University - VNU - HCM</td>
69     </tr>
70   <tr style='background-color: lightsalmon;'>
71     <td><b>Education</b></td>
72     <td>
73       <p>Major in Network Engineering at International University - VNU - HCM (2017-current)</p>
74       <p>Student at Nguyen Thuong Hien high school (2014-2017)</p>
75       <p>Student at Ngo Si Lien secondary school (2011-2014)</p>
76       <p>Student at Le Van Si primary school (2007-2011)</p>
77     </td>
78   </tr>
79   <tr>
80     <td><b>Employment</b></td>
81     <td>None</td>
82   </tr>
83 </table>
84   </td>
85 </tr>
86 </table>
87 </div>
88   <p style='line-height: 0.1;'>Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
89 </div>
90 </body>
91 </html>
92
93
94
95
96
97

```

Figure 3.2 and 3.3: Source code for about.html

c) Demo:

You can see the title of the page “Personal information” on the top, my picture in the left with 2 social media icons under it. The icons are clickable links so when you click on them, they will take you to my actual Instagram and Facebook pages (requires internet connection). Other than that, the information table on the right are just my personal information.

**Personal Information**

<b>Full name</b>	Nguyen Anh Dung
<b>Gender</b>	Male
<b>Date of birth</b>	June 12th, 1999
<b>Address</b>	15/10A C1 street, Tan Binh district, Ho Chi Minh city
<b>Occupation</b>	Student at International University - VNU - HCM
<b>Education</b>	Major in Network Engineering at International University - VNU - HCM (2017-current) Student at Nguyen Thuong Hien high school (2014-2017) Student at Ngo Si Lien secondary school (2011-2014) Student at Le Van Si primary school (2007-2011)

Figure 3.4 and 3.5: Demo for the About page.

**Personal Information**

<b>Full name</b>	Nguyen Anh Dung
<b>Gender</b>	Male
<b>Date of birth</b>	June 12th, 1999
<b>Address</b>	15/10A C1 street, Tan Binh district, Ho Chi Minh city
<b>Occupation</b>	Student at International University - VNU - HCM
<b>Education</b>	Major in Network Engineering at International University - VNU - HCM (2017-current) Student at Nguyen Thuong Hien high school (2014-2017) Student at Ngo Si Lien secondary school (2011-2014) Student at Le Van Si primary school (2007-2011)
<b>Employment</b>	None

Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.

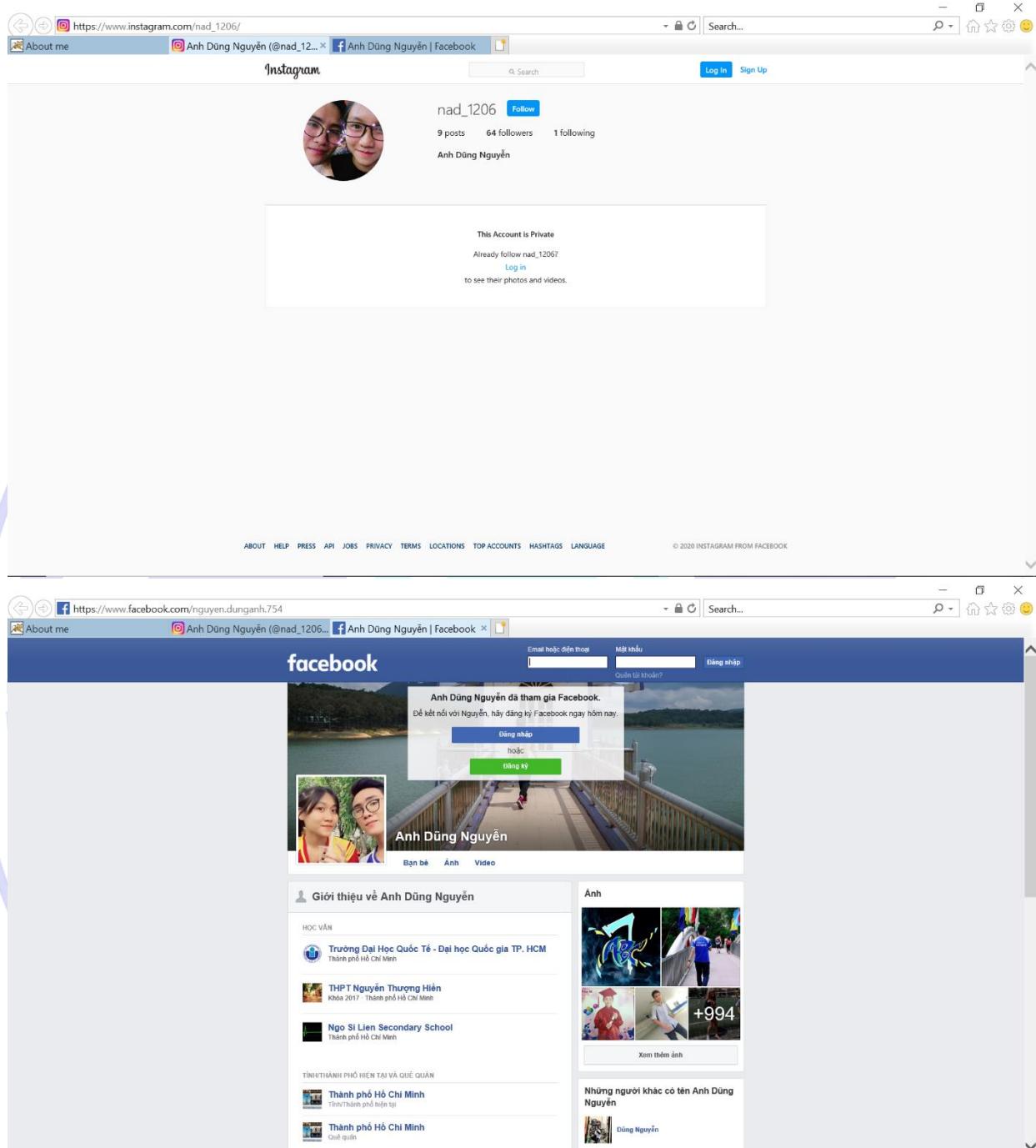


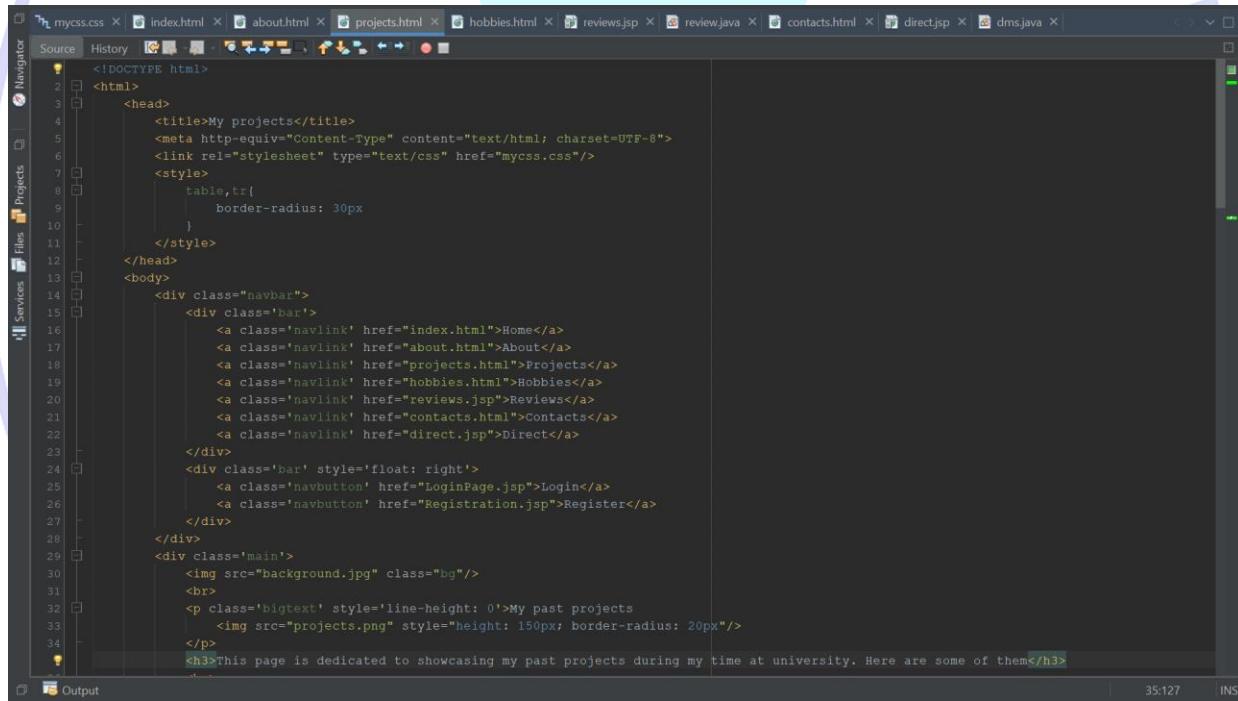
Figure 3.6 and 3.7: Facebook and Instagram appear when clicking the links/icons

## 4) Projects page:

### a) Explanation:

- Content: The Projects page (projects.html) is the site where I show all my past projects that I have done in my years being an IT student. The main content of this page is a title of the page, a picture for decoration next to it, a description of the page under it and a table that shows 3 projects with their own descriptions and videos/images of the projects.
- Design: Other than the navbar, the footer and the title established, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a main table taking 100% of the middle portion, containing 3 rows showcasing 3 projects. Each row having 2 columns, 1 will be containing a picture/video of the project and the other containing a smaller table that shows the project's description. The small tables are style like the information table on the information page, all tables are styled with css (see css part) and all images/videos are just resized to fit.

### b) Source code:



```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My projects</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <link rel="stylesheet" type="text/css" href="mycss.css"/>
7     <style>
8       table, tr {
9         border-radius: 30px;
10      }
11    </style>
12  </head>
13  <body>
14    <div class="navbar">
15      <div class="bar">
16        <a class="navlink" href="index.html">Home</a>
17        <a class="navlink" href="about.html">About</a>
18        <a class="navlink" href="projects.html">Projects</a>
19        <a class="navlink" href="hobbies.html">Hobbies</a>
20        <a class="navlink" href="reviews.jsp">Reviews</a>
21        <a class="navlink" href="contacts.html">Contacts</a>
22        <a class="navlink" href="direct.jsp">Direct</a>
23      </div>
24      <div class="bar" style="float: right">
25        <a class="navbutton" href="LoginPage.jsp">Login</a>
26        <a class="navbutton" href="Registration.jsp">Register</a>
27      </div>
28    </div>
29    <div class="main">
30      
31      <br>
32      <p class="bigtext" style="line-height: 0">My past projects
33        
34      </p>
35      <h3>This page is dedicated to showcasing my past projects during my time at university. Here are some of them</h3>
36    </div>
37  </body>
38</html>

```

Figure 4.1: Source code for projects.html

The figure consists of two screenshots of a Java IDE (NetBeans) showing the source code for a file named "projects.html".

**Screenshot 1 (Top):**

```

36         <br>
37         <table width='100%'>
38             <tr>
39                 <td style='background-color: black'><video height='500px' autoplay controls>
40                     <source src="DSA.mp4" type="video/mp4">
41                     Your browser does not support the video tag.
42                 </video>
43             </td>
44             <td>
45                 <table width='100%' style='border: 1px solid black; background-color:antiquewhite; text-align: left'>
46                     <tr style='background-color:lightsalmon'>
47                         <td><b>Project:</b></td>
48                         <td>2048 game</td>
49                     </tr>
50                     <tr>
51                         <td><b>Course:</b></td>
52                         <td>Data Structure and Algorithm</td>
53                     </tr>
54                     <tr style='background-color:lightsalmon'>
55                         <td><b>Programming language:</b></td>
56                         <td>Java</td>
57                     </tr>
58                     <tr>
59                         <td><b>Showcasing content:</b></td>
60                         <td>Demo video</td>
61                     </tr>
62                     <tr style='background-color:lightsalmon'>
63                         <td><b>Tools:</b></td>
64                         <td>Netbeans IDE</td>
65                     </tr>
66                 </table>
67             </td>
68         </tr>
69     <td>

```

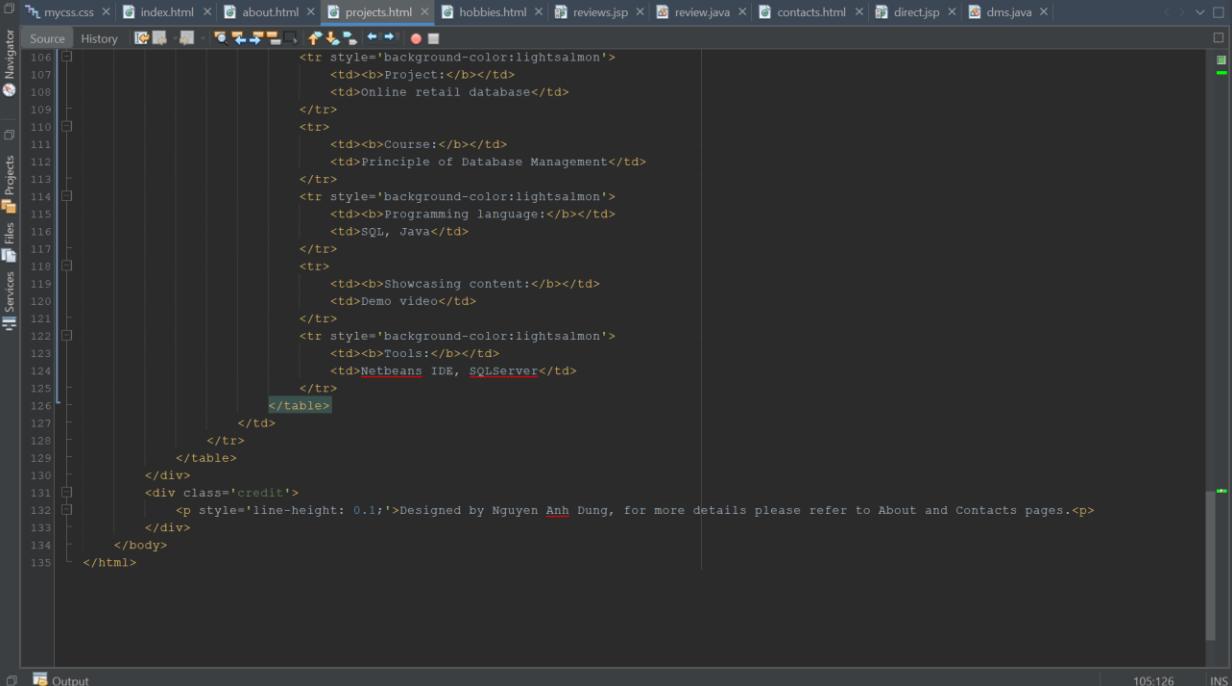
**Screenshot 2 (Bottom):**

```

71             <table width='100%' style='text-align: left; border: 1px solid black; background-color:antiquewhite'>
72                 <tr>
73                     <td><b>Project:</b></td>
74                     <td>Online booking system</td>
75                 </tr>
76                 <tr style='background-color:lightsalmon'>
77                     <td><b>Course:</b></td>
78                     <td>Software Engineering</td>
79                 </tr>
80                 <tr>
81                     <td><b>Programming language:</b></td>
82                     <td>None</td>
83                 </tr>
84                 <tr style='background-color:lightsalmon'>
85                     <td><b>Showcasing content:</b></td>
86                     <td>Use Case Diagram</td>
87                 </tr>
88                 <tr>
89                     <td><b>Tools:</b></td>
90                     <td>Draw.io</td>
91                 </tr>
92             </table>
93         </td>
94         <td>
95             
96         </td>
97     </tr>
98     <tr>
99         <td style='background-color: black'><video height='500px' autoplay controls>
100            <source src="database.mp4" type="video/mp4" />
101            Your browser does not support the video tag.
102        </video>
103    </td>
104    <td>

```

Figure 4.2 and 4.3: Source code for projects.html



The screenshot shows the NetBeans IDE interface with the source code for `projects.html`. The code is an HTML table listing various projects. The table has three columns: Project, Description, and Tools. Several rows have a light salmon background color. The code includes CSS styles for the table rows and columns, and a credit section at the bottom.

```

<tr style="background-color:lightsalmon">
    <td><b>Project:</b></td>
    <td>Online retail database</td>
</tr>
<tr>
    <td><b>Course:</b></td>
    <td>Principle of Database Management</td>
</tr>
<tr style="background-color:lightsalmon">
    <td><b>Programming language:</b></td>
    <td>SQL, Java</td>
</tr>
<tr>
    <td><b>Showcasing content:</b></td>
    <td>Demo video</td>
</tr>
<tr style="background-color:lightsalmon">
    <td><b>Tools:</b></td>
    <td>Netbeans IDE, SQLServer</td>
</tr>
</table>
</td>
</tr>
</table>
</div>
<div class="credit">
    <p style="line-height: 0.1;">Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
</div>
</body>
</html>

```

Figure 4.4: Source code for projects.html

### c) Demo:

You can see the title and a display image at the top, a description under it and then a list of projects with their demos and descriptions. All videos included are autoplay and can be pause, rewind and fast forward.

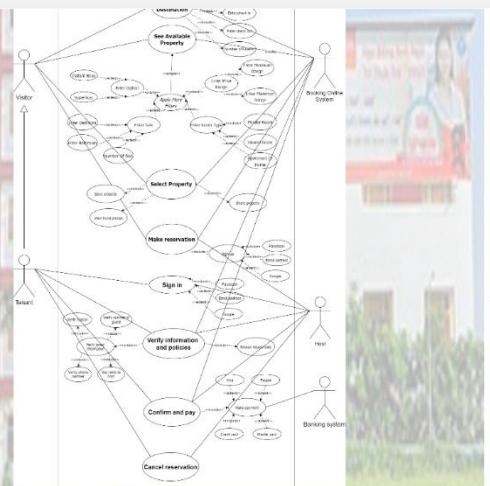


The screenshot shows a web browser displaying the `projects.html` page. The header includes a logo, navigation links (Home, About, Projects, Hobbies, Reviews, Contacts, Direct), and user authentication buttons (Login, Register). The main content features a large image of a university building, a title "My past projects", and a sidebar with a 2048 game. To the right is a table summarizing the project details.

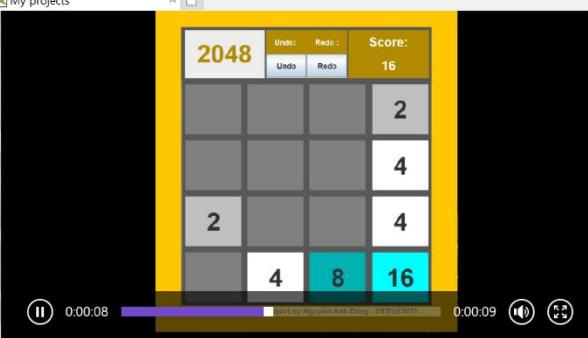
<b>Project:</b>	2048 game
<b>Course:</b>	Data Structure and Algorithm
<b>Programming language:</b>	Java
<b>Showcasing content:</b>	Demo video
<b>Tools:</b>	Netbeans IDE

Figure 4.5: Demo for the Projects page

**Project:** Online booking system  
**Course:** Software Engineering  
**Programming language:** None  
**Showcasing content:** Use Case Diagram  
**Tools:** Draw.io



**Project:** Online retail database  
**Course:** Data Structure and Algorithm  
**Programming language:** Java  
**Showcasing content:** Demo video  
**Tools:** Netbeans IDE



**Project:** Online booking system  
**Course:** Software Engineering  
**Programming language:** None

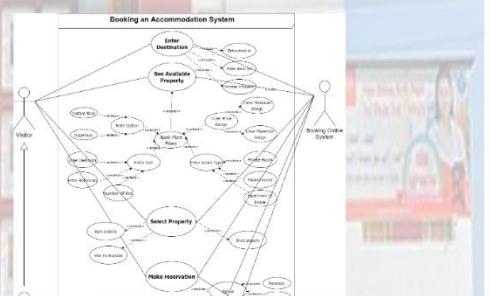


Figure 4.6 and 4.7: Demo for the Projects page

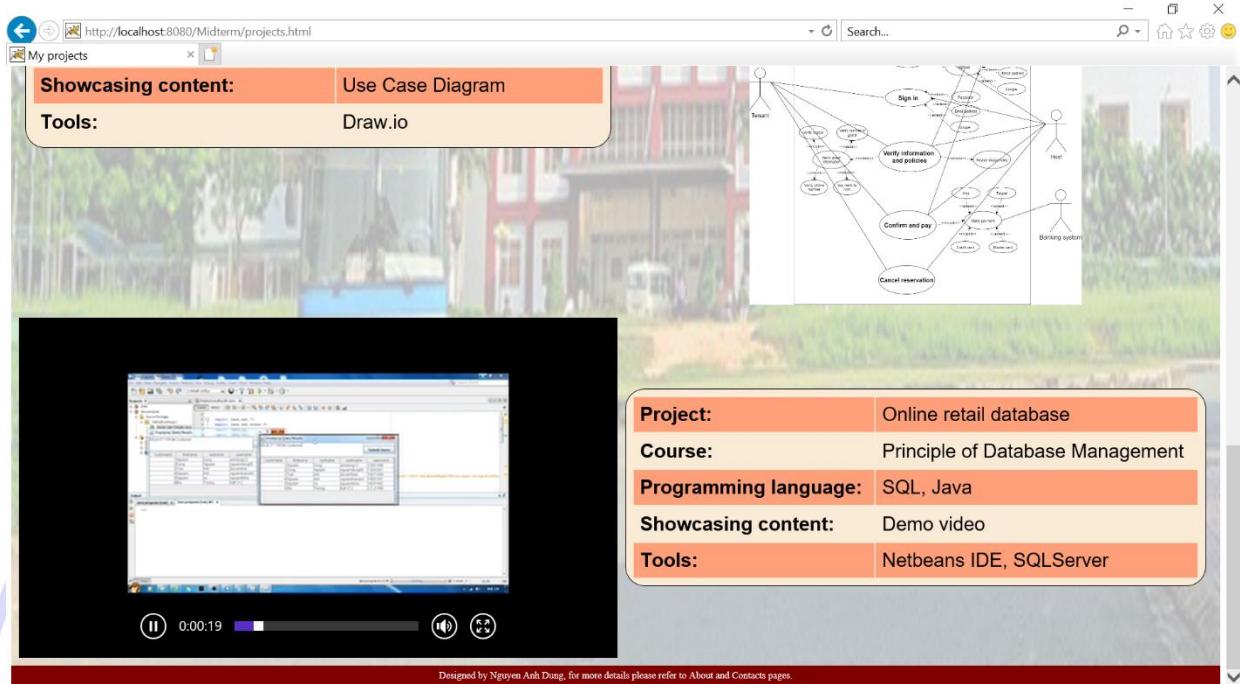


Figure 4.8: Demo for the Projects page

## 5) Hobbies page:

### a) Explanation:

- Content: The Hobbies page (hobbies.html) is the site where I show all my hobbies that I'm really interested in. The main content of this page is a title of the page, a picture for decoration next to it, a description of the page under it and a table that shows 3 hobbies with my thoughts and images of the hobbies. (similar to the Project page)
- Design: Other than the navbar, the footer and the title established, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a main table taking 100% of the middle portion, containing 3 rows showcasing 3 hobbies. Each row having 2 columns, 1 will be containing a picture of the hobby and the other containing a smaller table that shows what I think about it. The small tables are style like the information table on the information page, all tables are styled with css (see css part) and all images are just resized to fit. (similar to the Projects page).

### b) Source code:

Figure 5.1 shows the source code for the hobbies.html file. The code includes an external CSS file (mycss.css) for styling tables and borders. The HTML structure features a navigation bar with links to various pages, a main content area with a background image, and a section titled "My hobbies" containing three items: Music, Traveling, and Gaming.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Direct messages</title>
5          <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6          <link rel="stylesheet" type="text/css" href="mycss.css"/>
7          <style>
8              table, tr{
9                  border-radius: 20px
10             }
11             </style>
12     </head>
13     <body>
14         <div class="navbar">
15             <div class="bar">
16                 <a class="navlink" href="index.html">Home</a>
17                 <a class="navlink" href="about.html">About</a>
18                 <a class="navlink" href="projects.html">Projects</a>
19                 <a class="navlink" href="hobbies.html">Hobbies</a>
20                 <a class="navlink" href="reviews.jsp">Reviews</a>
21                 <a class="navlink" href="contacts.html">Contacts</a>
22                 <a class="navlink" href="direct.jsp">Direct</a>
23             </div>
24             <div class="bar" style='float: right'>
25                 <a class="navbutton" href="loginPage.jsp">Login</a>
26                 <a class="navbutton" href="Registration.jsp">Register</a>
27             </div>
28         </div>
29         <div class="main">
30             
31             <br>
32             <h1 class="bigtext" style="line-height: 0.1;'>My hobbies
33                 
34             </h1>
35             <h3>These are some of the things I like to do!</h3>
36         </div>
37     </body>
38 </html>

```

Figure 5.2 shows the source code for the hobbies.html file, focusing on the content of the "Music" section. It displays a table with a background color of light salmon, containing a title and a descriptive paragraph about the speaker's fondness for music.

```

36     <br>
37     <table width='100%'>
38         <tr>
39             <td>
40                 
41             </td>
42             <td>
43                 <table width='100%' style='border: 1px solid black; background-color:antiquewhite; text-align: left'>
44                     <tr style='background-color:lightsalmon'>
45                         <td><b>Music:</b></td>
46                     </tr>
47                     <tr>
48                         <td>Music has always been a favorite hobby of mine since i was in 6th grade, back in the days of Westlife, Backstreet Boys, and NSYNC. I still listen to them today and they bring me back to those carefree days of my life.</td>
49                     </tr>
50                 </table>
51             </td>
52         </tr>
53         <tr>
54             <td>
55                 <table width='100%' style='text-align: left; border: 1px solid black; background-color:antiquewhite'>
56                     <tr style='background-color:lightsalmon'>
57                         <td><b>Traveling:</b></td>
58                     </tr>
59                     <tr>
60                         <td>I feel like traveling is both overrated and underrated at the same time. It's the best way to spend time to relax and explore new places, but it can also be expensive and time-consuming. I've traveled to many countries and cities, and I always come back with great memories and new experiences.</td>
61                     </tr>
62                 </table>
63             </td>
64             <td>
65                 
66             </td>
67         </tr>
68         <tr>
69             <td>
70                 
71             </td>
72         </tr>
73     </table>
74 
```

Figure 5.1 and 5.2: Source code for the hobbies.html

```

71 <td>
72   <td>
73     <table width='100%' style='text-align: left; border: 1px solid black; background-color:antiquewhite'>
74       <tr style='background-color:lightsalmon'>
75         <td><b>Gaming:</b></td>
76       </tr>
77       <tr>
78         <td>Everybody seems to think that gaming is an addiction and a bad thing. While that is true in some cases (especi.
79       </tr>
80     </table>
81   </td>
82 </tr>
83 </table>
84 <br>
85 </div>
86 <div class='credit'>
87   <p style='line-height: 0.1;'>Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
88 </div>
89 </body>
90 </html>

```

Figure 5.3: Source code for the hobbies.html

**c) Demo:**

You can see the title and a display image at the top, a description under it and then a list of hobbies with my own thoughts on them next to their pictures.

**My hobbies**

These are some of the things I like to do!

**Music:**

Music has always been a favorite hobby of mine since I was in 6th grade, back in the days of Westlife, Backstreetboys, etc. I listen to all kinds of genre like pop, R&B, rock and especially hip hop. For me music is a kind of necessity in life as it helps me focus when I need to and relax when I take a break. In this crazy, chaotic and fast-paced world, some good tunes can be a great thing to have!

Figure 5.4: Demo for the Hobbies page

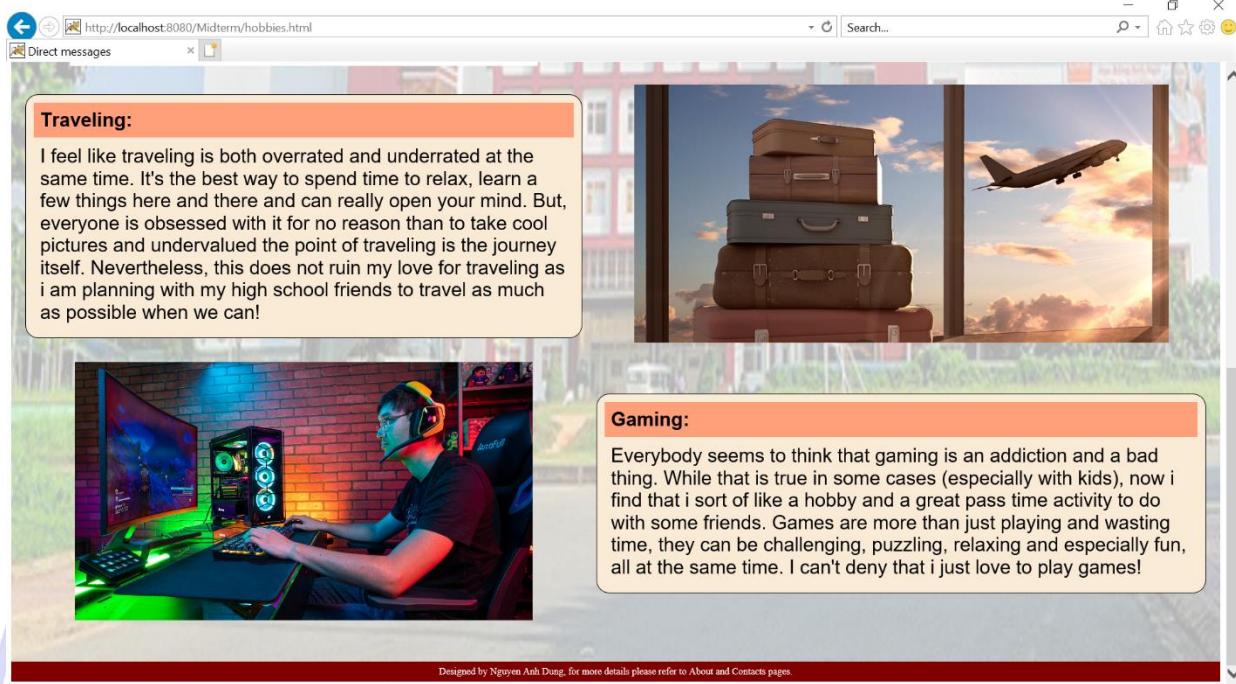


Figure 5.5: Demo for the Hobbies page

## 6) Contacts page:

### a) Explanation:

- Content: The Contacts page (contacts.html) is the site where I show all my contact details, links to my personal social media accounts. The main content of this page is a title of the page, a picture for decoration next to it and a table that shows 5 ways to contact me: via email, facebook, Instagram, phone number and direct message. The links to facebook and Instagram are actual clickable links that takes you to my facebook and Instagram, as well as the link “Go to direct message” is a link that takes you to the Direct page (more on this page in the Direct page part). Other than those 5 ways, I also show a picture of google map and an address that is my actual location.
- Design: Other than the navbar and the footer, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a main table taking 100% of the middle portion, containing 3 rows. The first row contains the title and a decoration image of the page, the second row is a small table that contains the contact details with their matching icons, the last row is a table that have 1 row, 2 columns: 1 column with width = 25% containing my address and the other column with width = 75% containing a picture of my house on google maps. The small tables are style like the information table on the information page, all tables are styled with css (see css part) and all images are just resized to fit.

### b) Source code:

The image shows two side-by-side screenshots of a code editor interface, likely Eclipse or similar, displaying the source code for a web application.

**Top Screenshot (contacts.html):**

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Contact details</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <link rel="stylesheet" type="text/css" href="mycss.css"/>
7     <style>
8       table, tr{
9         border-radius: 20px
10      }
11    </style>
12  </head>
13  <body>
14    <div class="navbar">
15      <div class="bar">
16        <a class="navlink" href="index.html">Home</a>
17        <a class="navlink" href="about.html">About</a>
18        <a class="navlink" href="projects.html">Projects</a>
19        <a class="navlink" href="hobbies.html">Hobbies</a>
20        <a class="navlink" href="reviews.jsp">Reviews</a>
21        <a class="navlink" href="contacts.html">Contacts</a>
22        <a class="navlink" href="direct.jsp">Direct</a>
23      </div>
24      <div class="bar" style="float: right">
25        <a class="navbutton" href="loginPage.jsp">Login</a>
26        <a class="navbutton" href="Registration.jsp">Register</a>
27      </div>
28    </div>
29    <div class="main">
30      
31      <table style="width: 100%; border:none">
32        <tr style="border:none">
33          <td style="border:none"><h1 class='bigtext'>My contact details</h1></td>
34          <td style="border:none"></td>
35        </tr>

```

**Bottom Screenshot (contacts.jsp):**

```

36      </tr>
37      <tr>
38        <td style="width: 60%; background-color: antiquewhite; margin: 0 auto; text-align: left" >
39          <table style="width: 100%; border:none">
40            <tr style="background-color: lightsalmon">
41              <td><b>Follow me on Instagram at:</b></td>
42              <td>
43                <a href="https://www.instagram.com/nad_1206/" style="color: blue;">
44                  
45                  @nad_1206
46                </a>
47              </td>
48            </tr>
49            <tr>
50              <td><b>Follow me on Facebook at:</b></td>
51              <td>
52                <a href="https://www.facebook.com/nguyen.dunganh.754" style="color: blue;">
53                  
54                  @nguyen.dunganh.754
55                </a>
56              </td>
57            </tr>
58            <tr style="background-color: lightsalmon">
59              <td><b>Email me at:</b></td>
60              <td>
61                
62                dungennoi@gmail.com
63              </td>
64            </tr>
65            <tr>
66              <td><b>(For work only) Contact me:</b></td>
67              <td>
68                
69                +84902701624
70              </td>
71            </tr>
72          <tr style='background-color: lightsalmon'>

```

Figure 6.1 and 6.2: Source code for the contacts.html

Screenshot of a code editor showing the source code for 'contacts.html'. The code includes a table for contact details, a geographical location section with an address and map image, and a credit section for the designer.

```

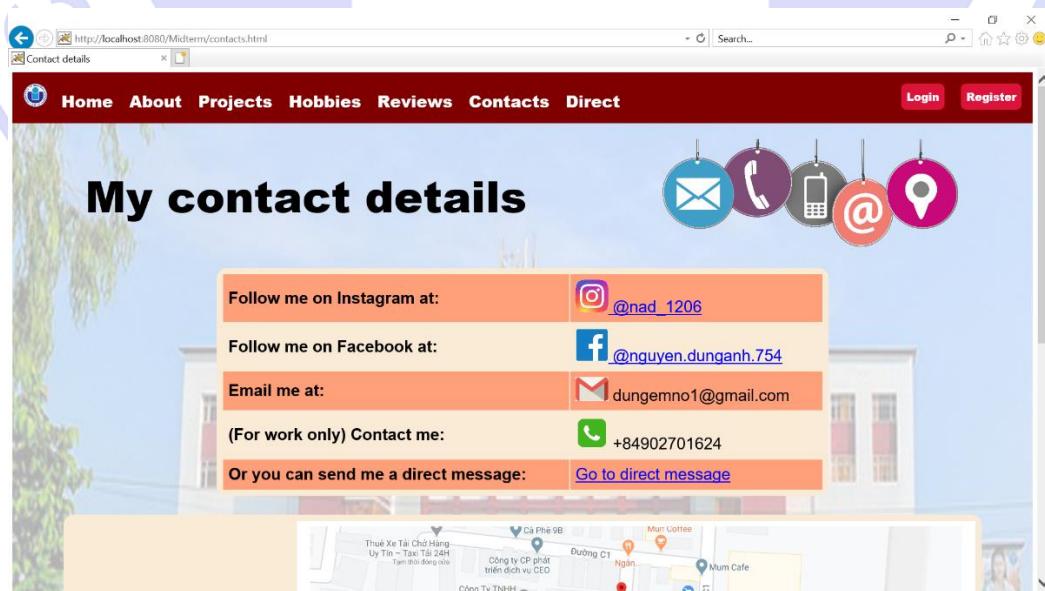
1 <td><b>Or you can send me a direct message:</b></td>
2 <td>
3 <a href="direct.jsp" style="color: blue; ">
4 Go to direct message
5 </a>
6 </td>
7 </tr>
8 </tr><td><br></td></tr>
9 <tr>
10 <td style="width: 50%; background-color: antiquewhite; margin: 0 auto">
11 <tr>
12 <td style="width: 25">
13 <B>My geographical location:</B>
14 <p>
15 15/10A C1 street, Tan Binh district, Ho Chi Minh city</p>
16 </td>
17 <td style="width: 75%;background-color: white"></td>
18 </tr>
19 </table>
20 </td>
21 </tr>
22 <tr><td><br></td></tr>
23 </table>
24 </div>
25 <div class='credit'>
26 <p style="line-height: 0.1;">Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.</p>
27 </div>
28 </body>
29 </html>

```

Figure 6.3: Source code for the contacts.html

### c) Demo:

You can see the title and a display image at the top row, then a list of contact details related to me in a small table in the middle of a row of the big table. Scroll down you will see the geographical location (address) of my house and a picture of it on google maps. Also, the links are clickable just like on the Information page so I won't be showcasing them again, same goes for the direct message link that it will take you to the Direct page and I will show it in the later part.



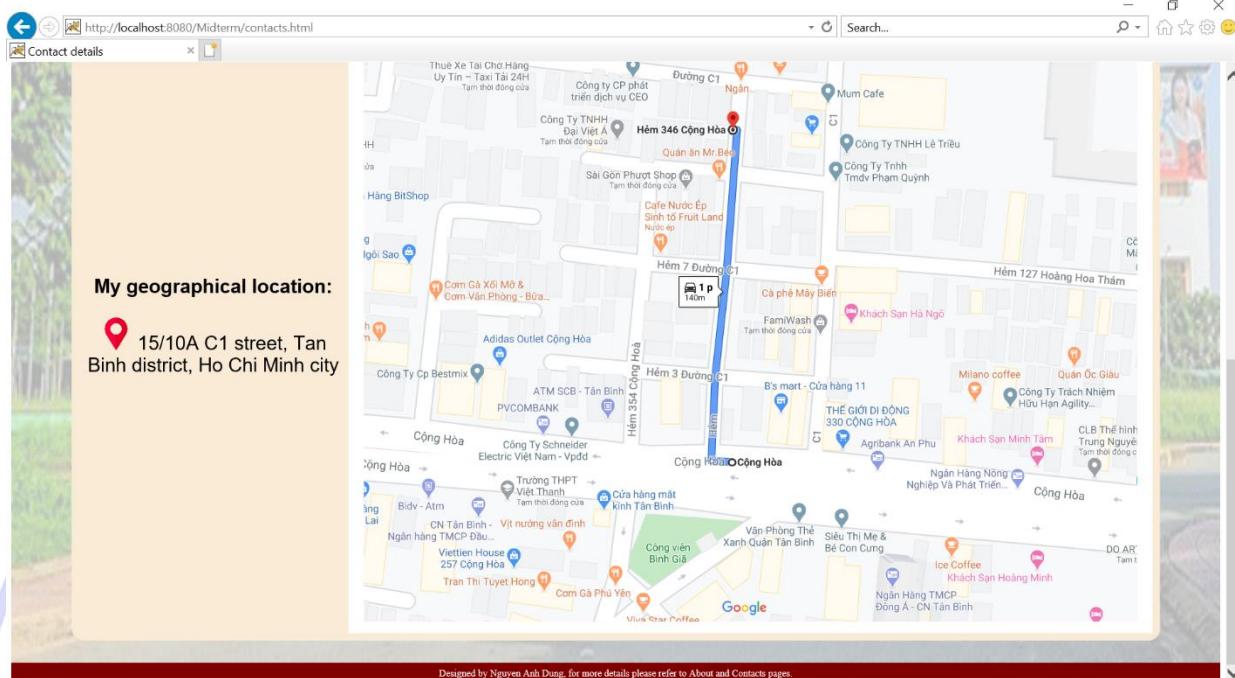


Figure 6.4 and 6.5: Demo for the Contacts page

## 7) Review page:

### a) Explanation:

- Content: The Reviews page (reviews.jsp) is the site where I show the reviews on some of my favorite movies. The main content of this page is a title of the page, a picture for decoration next to it and a table that shows a variety of clickable buttons with movie titles on them on the left of the page, as well as the chosen movie poster, rating and review on the right of the page when click. This page is interactive so if you want to see the review on what movie available you just click and it will appear on the right.
- Design: Other than the navbar, the footer and the title established, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a main table taking 100% of the middle portion, containing 1 row and 2 columns. The first column the submit buttons of the movie titles that are clickable and link to the servlet review.java by using <form> tag and all are style by using .mymovie class in css. The other row containing a small table that also have a smaller table on the first row and the reviews on the second row. The smallest table is divided into 25% and 75% columns with each of them containing rating and movie poster of the respective movie. All tables are styled with css (see css part) and all images are just resized to fit.
- Premise: The premise of this page is to have a list of buttons that have movie titles that once clicked, will send their respective value to review.java and review.java will return the poster, rating and review back to the this page immediately to display.

**b) Source code:**

```

1 <%@page import="java.util.Date"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <title>Direct messages</title>
6     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7     <link rel="stylesheet" type="text/css" href="mycss.css"/>
8     <style>
9       table, tr {
10         border-radius: 20px;
11       }
12     </style>
13   </head>
14   <body>
15     <div class="navbar">
16       <div class="bar">
17         <a class="navlink" href="index.html">Home</a>
18         <a class="navlink" href="about.html">About</a>
19         <a class="navlink" href="projects.html">Projects</a>
20         <a class="navlink" href="hobbies.html">Hobbies</a>
21         <a class="navlink" href="reviews.jsp">Reviews</a>
22         <a class="navlink" href="contacts.html">Contacts</a>
23         <a class="navlink" href="direct.jsp">Direct</a>
24       </div>
25       <div class="bar" style="float: right;">
26         <a class="navbutton" href="loginPage.jsp">Login</a>
27         <a class="navbutton" href="Registration.jsp">Register</a>
28       </div>
29     </div>
30     <div class="main">
31       
32       <br>
33       <h1 class="bigtext" style="line-height: 0.1;">Movie reviews
34       
35     </h1>
36   </div>
37 
```

```

36   <h3>These are the reviews on some of the movies that i really like! </h3>
37   <br>
38   <table size="100%">
39     <tr>
40       <td style="width: 40%; vertical-align: text-top;">
41         <%
42           String p = (String)session.getAttribute("poster");
43           String c = (String)session.getAttribute("content");
44           String r = (String)session.getAttribute("rating");
45           if(p==null || c==null || r==null){
46             session.setAttribute("poster","none");
47             session.setAttribute("content","No record");
48             session.setAttribute("rating","No record");
49           }
50         %>
51         <b>Click on the movie title to see review: </b><br><br>
52         <form method="post" action="review">
53           <input type="submit" value="Avengers: Endgame" name="movie" class="mymovie"><br><br>
54           <input type="submit" value="American Psycho" name="movie" class="mymovie"><br><br>
55           <input type="submit" value="Avengers: Infinity War" name="movie" class="mymovie"><br><br>
56           <input type="submit" value="Parasite" name="movie" class="mymovie"><br><br>
57           <input type="submit" value="Joker" name="movie" class="mymovie"><br><br>
58           <input type="submit" value="Aquaman" name="movie" class="mymovie"><br><br>
59           <input type="submit" value="Inception" name="movie" class="mymovie"><br><br>
60           <input type="submit" value="Us" name="movie" class="mymovie"><br><br>
61           <input type="submit" value="Get Out" name="movie" class="mymovie"><br><br>
62           <input type="submit" value="Spider-Man: Into the Spider-Verse" name="movie" class="mymovie"><br><br>
63         </form>
64       </td>
65       <td style="width: 60%;">
66         <table style="width: 100%; background-color: antiquewhite;">
67           <tr>
68             <td>
69               <table style="width: 100%; background-color: lightsalmon;">
70             <tr>
```

Figure 7.1 and 7.2: Source code for the reviews.jsp

```

71 <td style="width: 25%">
72   <h1>Rating: </h1>
73   <h1><%=session.getAttribute("rating")%></h1>
74 </td>
75 <td style="width: 75%">
76   .jpg" style="height: 400px"/>
77 </td>
78 </tr>
79 </table>
80 </td>
81 <tr style="width: 100%; background-color: antiquewhite;">
82   <td><p style="text-align: left"><%=session.getAttribute("content")%></p></td>
83 </tr>
84 </table>
85 </td>
86 </tr>
87 </table>
88 <br>
89 </div>
90 <div class='credit'>
91   <p style='line-height: 0.1;'>Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
92 </div>
93 </div>
94 </body>
</html>

```

Figure 7.3: Source code for the reviews.jsp

### c) Demo:

As you can see, other than the title and the description, there are a list of buttons that represents the movies' titles on the left. Before clicking the page will show no record and no picture on the right. After one is clicked on, the page will show the corresponding movie's rating, poster and review that the user chose on the left.



Figure 7.4: Demo for the Reviews page

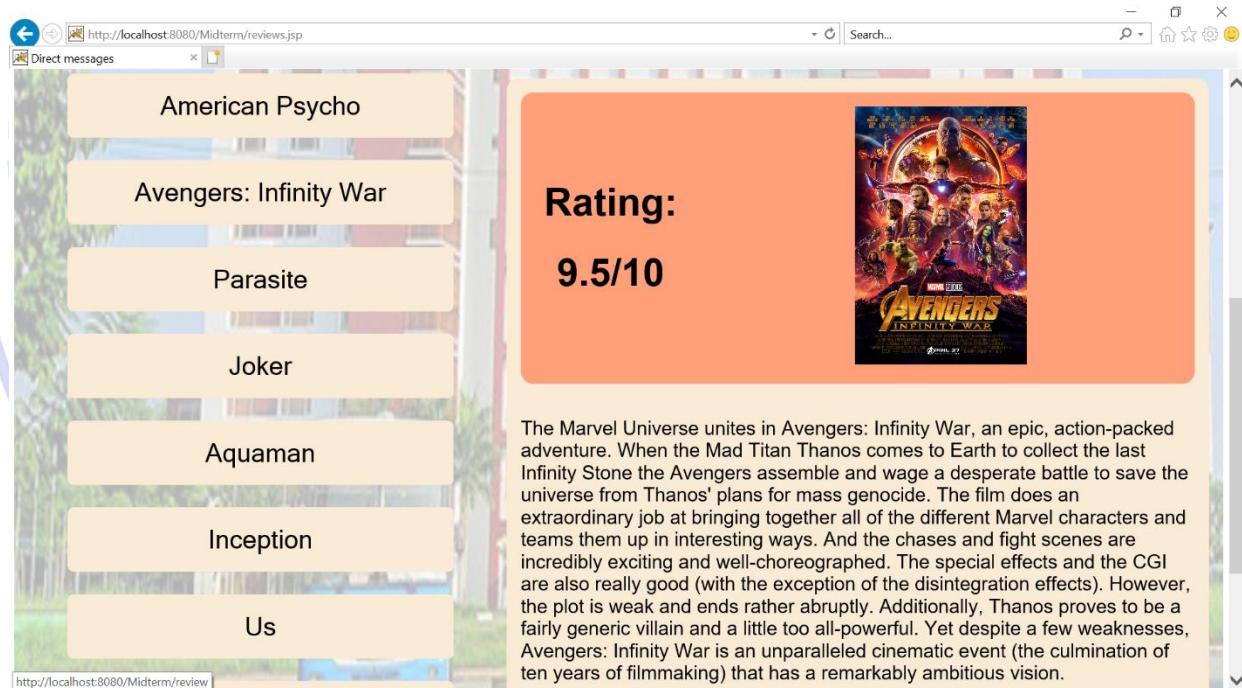
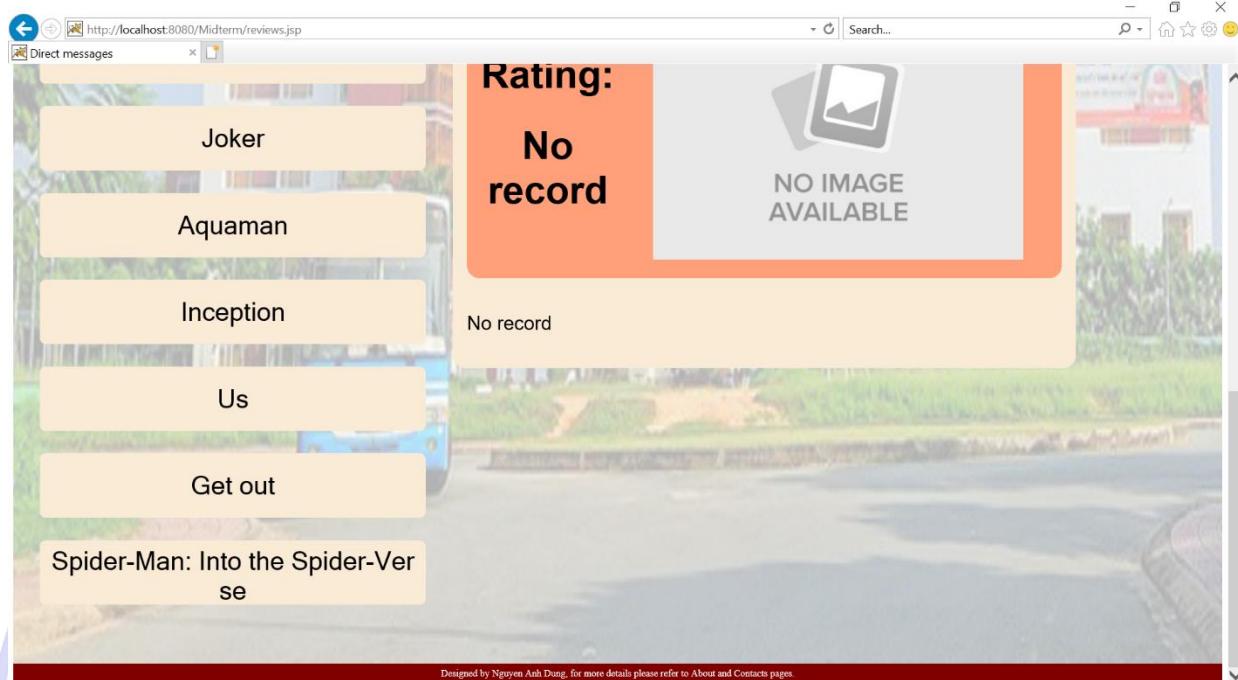


Figure 7.5 and 7.6: Demo for the Reviews page (before and after clicking on Avenger: Infinity War)



Figure 7.7: Demo for the Reviews page (after clicking on Parasite)

#### d) Servlet (review.java):

As state before, the reviews.jsp page will send data to review.java for data handling and return the movie poster, rating and review. How this is done is the servlet will take the value that the buttons sent with `request.getParameter("movie")`. From then, the servlet will call a switch for all cases of movie titles and send back the according poster, rating and review with a session through the attributes poster, content and rating. After setting the attribute, the servlet will `sendRedirect("reviews.jsp")` to redirect back to the jsp and the jsp will call these attributes out to get the values that the servlet has compiled.

```
mycss.css x index.html x about.html x projects.html x hobbies.html x reviews.jsp x review.java x contacts.html x direct.jsp x dms.java x
```

Source History

```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7 import java.io.IOException;
8 import java.io.PrintWriter;
9 import javax.servlet.ServletException;
10 import javax.servlet.annotation.WebServlet;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import javax.servlet.http.HttpSession;
15
16 /**
17 *
18 * @author lenovo
19 */
20 @WebServlet(urlPatterns = {"review"})
21 public class review extends HttpServlet {
22
23 /**
24 * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
25 * methods.
26 *
27 * @param request servlet request
28 * @param response servlet response
29 * @throws ServletException if a servlet-specific error occurs
30 * @throws IOException if an I/O error occurs
31 */
32 protected void processRequest(HttpServletRequest request, HttpServletResponse response)
33 throws ServletException, IOException {
34     response.setContentType("text/html;charset=UTF-8");
35     try (PrintWriter out = response.getWriter()) {
36         /* TODO output your page here. You may use following sample code. */
37         out.println("<!DOCTYPE html>");
38         out.println("<html>");
39         out.println("<head>");
40         out.println("<title>Servlet review</title>");
41         out.println("</head>");
42         out.println("<body>");
43         out.println("<h1>Servlet review at " + request.getContextPath() + "</h1>");
44         out.println("</body>");
45         out.println("</html>");
46
47         //Review contents/
48         HttpSession session = request.getSession(true);
49         String movie = (String)request.getParameter("movie");
50         switch(movie){
51             case "Avengers: Endgame":
52                 session.setAttribute("poster","movie1");
53                 session.setAttribute("rating","9.8/10");
54                 session.setAttribute("content","It all comes to an end in the epic superhero film Avengers: Endgame. In the wake of Thanos' m");
55                 break;
56             case "American Psycho":
57                 session.setAttribute("poster","movie2");
58                 session.setAttribute("rating","8.5/10");
59                 session.setAttribute("content","A surprisingly influential film. American Psycho gives us a dark, funny and amazing performance");
56                 break;
57             case "Avengers: Infinity War":
58                 session.setAttribute("poster","movie3");
59                 session.setAttribute("rating","8.5/10");
60                 session.setAttribute("content","The Marvel Universe unites in Avengers: Infinity War, an epic, action-packed adventure. When t");
61                 break;
62             case "Parasite":
63                 session.setAttribute("poster","movie4");
64                 session.setAttribute("rating","10/10");
65                 session.setAttribute("content","Writer/director Bong Joon Ho (\\"The Host\\" \\"Snowpiercer\\" \\"Okja\\") has been on a roll of lat");
66                 break;
67         }
68     }
69 }
70 }
```

Output

```
mycss.css x index.html x about.html x projects.html x hobbies.html x reviews.jsp x review.java x contacts.html x direct.jsp x dms.java x
```

Source History

```
36         /* TODO output your page here. You may use following sample code. */
37         out.println("<!DOCTYPE html>");
38         out.println("<html>");
39         out.println("<head>");
40         out.println("<title>Servlet review</title>");
41         out.println("</head>");
42         out.println("<body>");
43         out.println("<h1>Servlet review at " + request.getContextPath() + "</h1>");
44         out.println("</body>");
45         out.println("</html>");
46
47         //Review contents/
48         HttpSession session = request.getSession(true);
49         String movie = (String)request.getParameter("movie");
50         switch(movie){
51             case "Avengers: Endgame":
52                 session.setAttribute("poster","movie1");
53                 session.setAttribute("rating","9.8/10");
54                 session.setAttribute("content","It all comes to an end in the epic superhero film Avengers: Endgame. In the wake of Thanos' m");
55                 break;
56             case "American Psycho":
57                 session.setAttribute("poster","movie2");
58                 session.setAttribute("rating","8.5/10");
59                 session.setAttribute("content","A surprisingly influential film. American Psycho gives us a dark, funny and amazing performance");
56                 break;
57             case "Avengers: Infinity War":
58                 session.setAttribute("poster","movie3");
59                 session.setAttribute("rating","8.5/10");
60                 session.setAttribute("content","The Marvel Universe unites in Avengers: Infinity War, an epic, action-packed adventure. When t");
61                 break;
62             case "Parasite":
63                 session.setAttribute("poster","movie4");
64                 session.setAttribute("rating","10/10");
65                 session.setAttribute("content","Writer/director Bong Joon Ho (\\"The Host\\" \\"Snowpiercer\\" \\"Okja\\") has been on a roll of lat");
66                 break;
67         }
68     }
69 }
70 }
```

Output

*Figure 7.8 and 7.9: Source code for the review.java*

```

71     case "Joker":
72         session.setAttribute("poster","movie5");
73         session.setAttribute("rating","10/10");
74         session.setAttribute("content","A movie I've had to sit on for several days and I think I'll have to sit on even longer. Cont");
75         break;
76     case "Aquaman":
77         session.setAttribute("poster","movie6");
78         session.setAttribute("rating","8.5/10");
79         session.setAttribute("content","DC makes a hard turn toward the fantastical with James Wan's Aquaman. When King Orm attempts i");
80         break;
81     case "Inception":
82         session.setAttribute("poster","movie7");
83         session.setAttribute("rating","9/10");
84         session.setAttribute("content","A truly sensational film. I hadn't rewatched this for years and almost forgot how outstanding");
85         break;
86     case "Us":
87         session.setAttribute("poster","movie8");
88         session.setAttribute("rating","9.3/10");
89         session.setAttribute("content","Jordan Peele follows up his breakout hit, Get Out, with the psychological horror-thriller Us.");
90         break;
91     case "Get out":
92         session.setAttribute("poster","movie9");
93         session.setAttribute("rating","9.5/10");
94         session.setAttribute("content","The less you know about this movie before you watch it, the better. It's a creepy, delicious t");
95         break;
96     case "Spider-Man: Into the Spider-Verse":
97         session.setAttribute("poster","movie10");
98         session.setAttribute("rating","10/10");
99         session.setAttribute("content","Fully embraces its comic origins and the realization that most viewers know the ins and outs o");
100        break;
101    default:
102        session.setAttribute("poster","none");
103        session.setAttribute("content","none");
104        session.setAttribute("rating","none");
105    }
106}
107}
108}
109}
110}
111}
112}
113}
114}
115}
116}
117}
118}
119}
120}
121}
122}
123}
124}
125}
126}
127}
128}
129}
130}
131}
132}
133}
134}
135}
136}
137}
138}
139}
140}
141}
142}
143}
144}
145}
146}
147}
148}
149}
150}

```

HttpServlet methods. Click on the + sign on the left to edit the code.

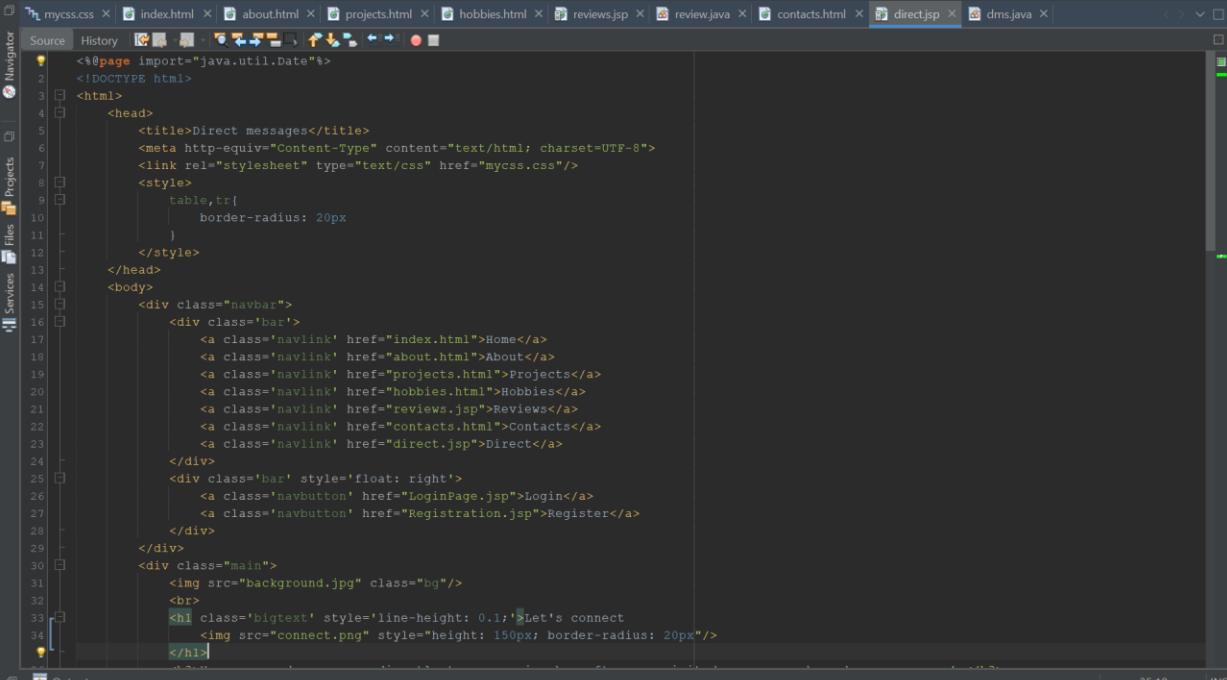
Figure 7.10 and 7.11: Source code for the review.java

## 8) Direct page:

*a) Explanation:*

- Content: The Direct page (direct.jsp) is the site where I connect with the user by letting them type in a name and send messages directly to me. The main content of this page is a title of the page, a picture for decoration next to it, a description of the page, a text box to get user's name, a text area to get user's text, a submit button and a table that shows the user's session details like: date time of first and last access, the last sent message and name along with total number of messages sent.
- Design: Other than the navbar, the footer and the title established, the main content of the page is styled with .main class in css, having a picture of IU as an opaque background using the class .bg, as well a main table taking 100% of the middle portion, containing 1 row and 2 columns. The first prompt the user with a text box and a text area with a submit button by using <form> tag. The other row containing a small table that are styled like the information table, containing session details. All tables are styled with css (see css part) and all images are just resized to fit.
- Premise: The premise of this page is to have the page display the session information inside the jsp itself. While the details like first and last access can be done with jsp alone, the message, name and message count will require jsp to call dms.java to handle data and send back the message, name and count the number of messages sent in a session.

*b) Source code:*



```

1 <%@page import="java.util.Date"%>
2 <!DOCTYPE html>
3 <html>
4     <head>
5         <title>Direct messages</title>
6         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7         <link rel="stylesheet" type="text/css" href="mycss.css"/>
8         <style>
9             table, tr{
10                 border-radius: 20px
11             }
12         </style>
13     </head>
14     <body>
15         <div class="navbar">
16             <div class="bar">
17                 <a class="navlink" href="index.html">Home</a>
18                 <a class="navlink" href="about.html">About</a>
19                 <a class="navlink" href="projects.html">Projects</a>
20                 <a class="navlink" href="hobbies.html">Hobbies</a>
21                 <a class="navlink" href="reviews.jsp">Reviews</a>
22                 <a class="navlink" href="contacts.html">Contacts</a>
23                 <a class="navlink" href="direct.jsp">Direct</a>
24             </div>
25             <div class="bar" style="float: right">
26                 <a class="navbutton" href="LoginPage.jsp">Login</a>
27                 <a class="navbutton" href="Registration.jsp">Register</a>
28             </div>
29         </div>
30         <div class="main">
31             
32             <br>
33             <h1 class="bigtext" style="line-height: 0.1;">Let's connect
34             
35         </div>

```

Figure 8.1: Source code for the direct.jsp

The image shows two side-by-side code editors, likely from an IDE like Eclipse or NetBeans, displaying the same JSP file, `direct.jsp`. The top editor shows the upper portion of the code, and the bottom editor shows the lower portion. Both editors have a dark theme and show the following code:

```

<hr><br>You can send a message directly to me or view how often you visited my page and send me a message! </hr>
<br>
<table style="width: 95%; background-color: antiquewhite; margin: 0 auto">
    <tr>
        <td style="width: 60%">
            <form method="post" action="dms">
                <p>
                    <label for="n">Your name (or leave blank): </label>
                    <input id="n" type="text" name="name" style="height:50px; width: 500px; font-size: 30px">
                </p>
                <textarea rows="15" name="dm" cols="55" style='font-size: 30px'></textarea>
                <br><br>
                <input type="submit" value='Send' style="height: 50px; width: 150px; font-size: 20px">
            </form>
        </td>
        <td>
            <%>
                session.setAttribute("first", new Date(session.getCreationTime()));
                session.setAttribute("last", new Date(session.getLastAccessedTime()));
            <%>
            <table style="width: 100%; background-color: salmon; text-align: left">
                <tr>
                    <td><b>First access: </b></td>
                    <td><%=session.getAttribute("first")%></td>
                </tr>
                <tr style="background-color:lightsalmon">
                    <td><b>Last access: </b></td>
                    <td><%=session.getAttribute("last")%></td>
                </tr>
                <tr>
                    <td><b>Last name used: </b></td>
                    <td><%=session.getAttribute("name")%></td>
                </tr>
                <tr style="background-color:lightsalmon">
                    <td><b>Last message: </b></td>
                    <td><%=session.getAttribute("message")%></td>
                </tr>
            </table>
        </td>
    </tr>
</table>
<br>
<div class='credit'>
    <p style='line-height: 0.1;'>Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
</div>

```

Figure 8.2 and 8.3: Source code for the direct.jsp

### c) Demo:

You can see the title, icon and description of page clearly at the top main content part of the page. In the middle of the page there's a text box that can be enter by your name, same goes for the text area below it and a submit button on the left side. Once submitted, the text will be shown on the right side of the page

along with total number of message sent, date time of first and last access in a session in this page. If you submit nothing and the counter doesn't go up, if you submitted some text with no name then it will prompt your name as Anonymous.

**Screenshot 1 (Top):**

<b>First access:</b>	Wed Apr 29 17:21:23 ICT 2020
<b>Last access:</b>	Wed Apr 29 17:21:47 ICT 2020
<b>Last name used:</b>	dung
<b>Last message:</b>	123123

**Screenshot 2 (Bottom):**

<b>First access:</b>	Wed Apr 29 17:21:23 ICT 2020
<b>Last access:</b>	Wed Apr 29 17:21:47 ICT 2020
<b>Last name used:</b>	dung
<b>Last message:</b>	123123
<b>Number of message sent:</b>	1

Figure 8.4 and 8.5: Demo for the Direct page (Before submitting any text)

**You can send a message directly to me or view how often you visited my page and send me a message!**

Your name (or leave blank):

**First access:** Wed Apr 29 22:37:52  
ICT 2020

**Last access:** Wed Apr 29 22:37:55  
ICT 2020

**Last name used:** dung

**Last message:** No message

**Number of message sent:** 0

**Send**

Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.

**You can send a message directly to me or view how often you visited my page and send me a message!**

Your name (or leave blank):

**First access:** Wed Apr 29 22:37:52  
ICT 2020

**Last access:** Wed Apr 29 22:38:23  
ICT 2020

**Last name used:** Anonymous

**Last message:** sdfsdfffsdffd

**Number of message sent:** 1

**Send**

Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.

Figure 8.6 and 8.7: Demo for the Direct page (After submitting. With no text is doesn't count and with no name it will count but set the name to anonymous)

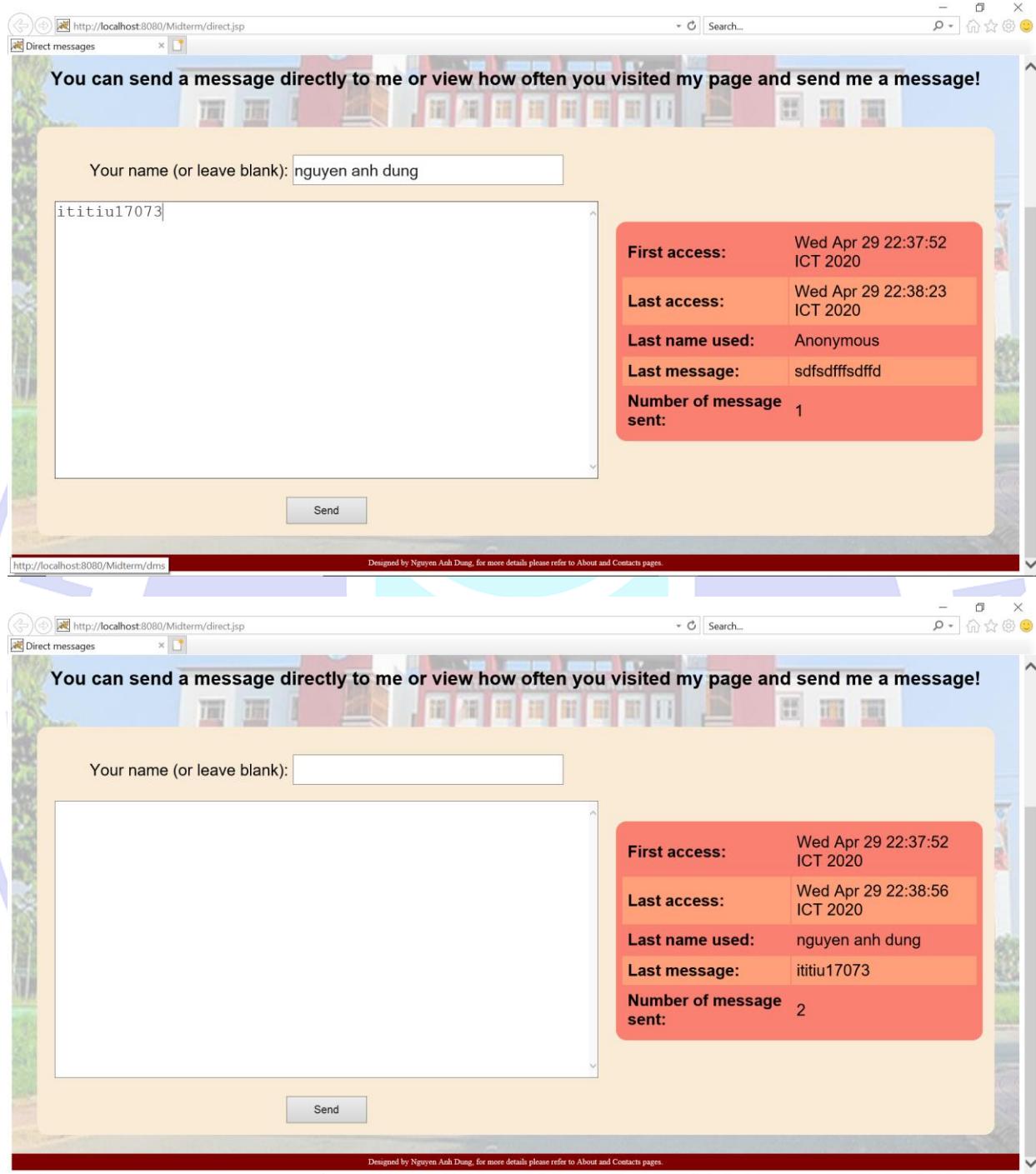


Figure 8.8 and 8.9: Demo for the Direct page (Before and after submitting a text with a name)

d) Servlet (dms.java):

As mentioned before, dms.java will receive the message and name from the form in direct.jsp and will return them as well as counting the total number of messages and send back to the jsp. It does this by creating a session with 3 attributes “count” “message” and “name” after receiving the data from the jsp.

It will then check the data and if the name is blank (name equals "") then the "name" will be set as "Anonymous". Same thing goes for the message, if it's blank then it will be set as "no message" as well as not counting up the counter. If the counter is null then the session is newly created so it will set to 0 for new users that visited the page. After all that is done, the servlet redirects back to the jsp and jsp will call these attributes through session to get the values. Although this method only works with each session.

```

1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6
7 import java.io.IOException;
8 import java.io.PrintWriter;
9 import java.util.Date;
10 import javax.servlet.ServletException;
11 import javax.servlet.annotation.WebServlet;
12 import javax.servlet.http.HttpServlet;
13 import javax.servlet.http.HttpServletRequest;
14 import javax.servlet.http.HttpServletResponse;
15 import java.util.LinkedList;
16 import java.util.Queue;
17 import javax.servlet.RequestDispatcher;
18 import javax.servlet.http.HttpSession;
19
20 /**
21 *
22 * @author lenovo
23 */
24 @WebServlet(urlPatterns = {"/dms"})
25 public class dms extends HttpServlet {
26
27     /**
28      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
29      * methods.
30      *
31      * @param request servlet request
32      * @param response servlet response
33      * @throws ServletException if a servlet-specific error occurs
34      * @throws IOException if an I/O error occurs
35     */
36
37     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
38         throws ServletException, IOException {
39         response.setContentType("text/html;charset=UTF-8");
40         try (PrintWriter out = response.getWriter()) {
41             /* TODO output your page here. You may use following sample code. */
42             out.println("<!DOCTYPE html>");
43             out.println("<html>");
44             out.println("<head>");
45             out.println("<title>Servlet dms</title>");
46             out.println("</head>");
47             out.println("<body>");
48             out.println("</body>");
49             out.println("</html>");
50
51             //Session//
52             HttpSession session = request.getSession(true);
53             Integer accesscount=(Integer)session.getAttribute("count");
54             if(accesscount == null){
55                 accesscount=0;
56             }
57             String name=(String)request.getParameter("name");
58             if(name.equals("")){
59                 name="Anonymous";
60             }
61             String msg = (String)request.getParameter("dm");
62             if(msg.equals ""){
63                 msg="No message";
64             }
65             else{
66                 accesscount++;
67             }
68             session.setAttribute("count",accesscount);
69             session.setAttribute("name",name);
70             session.setAttribute("message",msg);
71             response.sendRedirect("direct.jsp");
72         }
73     }
74
75 }

```

Figure 8.10 and 8.11: Source code for the dms.java

## 9) Structural design:

- Nguyen Anh Dung's Personal Website uses a navigation bar to set a connection between pages. Each and every page is embedded with a navigation bar, including the statics html pages and the dynamic jsp pages.
- Each page has a link to the others, the excluding ones are the servlets which are only connected to their according jsp pages to handle data when called.
- Another exception is the Contact page where it has another direct link to the Direct page if provoked.
- Considered all the factors above, this is the structural design of the web application:

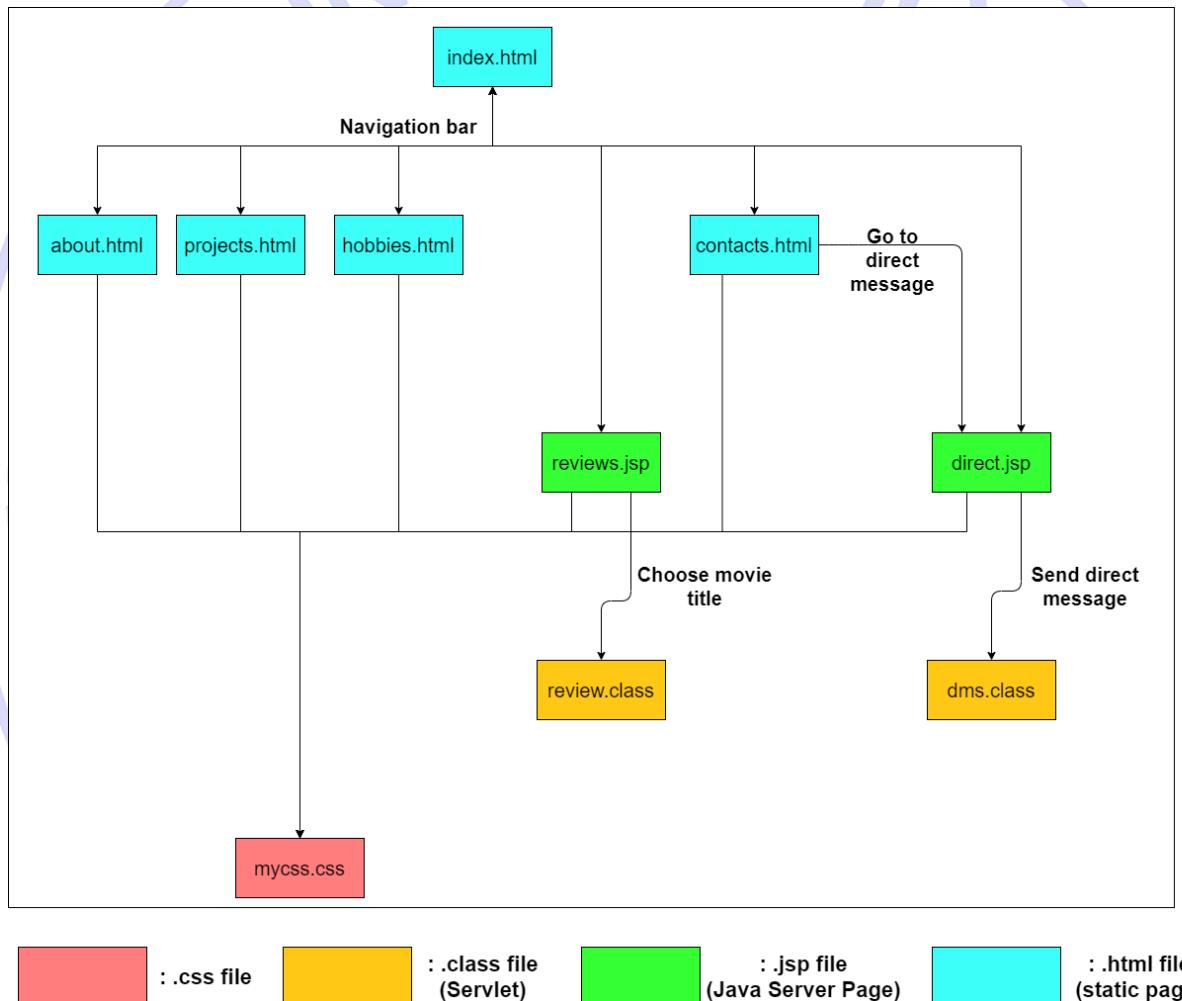
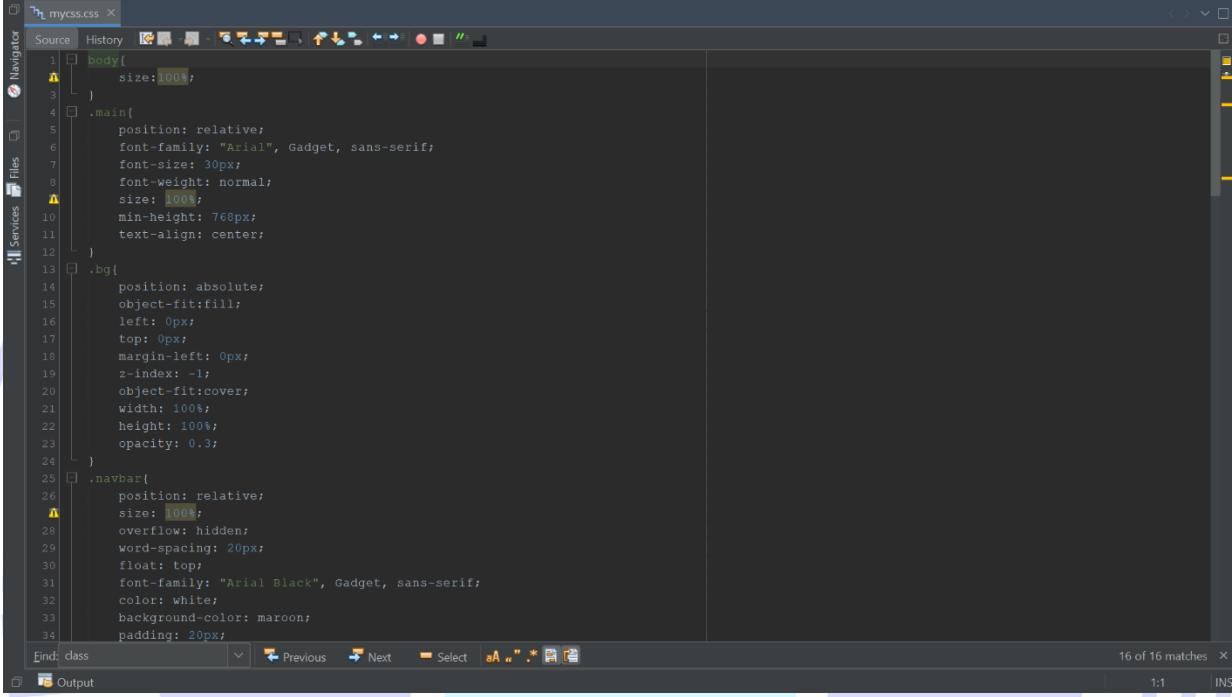


Figure 9.1: Nguyen Anh Dung's Personal Website structural design (with a glossary at the bottom)

## 10) Cascading Style Sheets:

- As mention many times in the previous parts, I used a dedicated css file to design, layout and decorate the front-end of my website (mainly the main content in the middle and the tables). The css file I use for part II to IV is mycss.css
- Most of the classes in mycss.css are for positioning, resize, borders, colors, spacing, font, tables, etc. I don't use any extra libraries like Bootstrap so nothing special.

Source code:



```
body{ size:100%; }
.main{ position: relative; font-family: "Arial", Gadget, sans-serif; font-size: 30px; font-weight: normal; size: 100%; min-height: 760px; text-align: center; }
.bg{ position: absolute; object-fit:fill; left: 0px; top: 0px; margin-left: 0px; z-index: -1; object-fit:cover; width: 100%; height: 100%; opacity: 0.3; }
.navbar{ position: relative; size: 100%; overflow: hidden; word-spacing: 20px; float: top; font-family: "Arial Black", Gadget, sans-serif; color: white; background-color: maroon; padding: 20px; }
```

Figure 10.1: Source code for mycss.css

The image shows two side-by-side panels of a code editor, both displaying the same CSS file named `mycss.css`. The left panel contains the original CSS code, while the right panel shows the same code with some styles removed or commented out.

```

35 }
36 .navbar:before{
37   display: inline-block;
38   content: url(logo.png);
39   zoom: 30%;
40 }
41 .bar{
42   word-spacing: 20px;
43   text-align: center;
44   font-size: 30px;
45   color: white;
46   margin-bottom: 0px;
47   display: inline-block;
48 }
49 .navlink{
50   color: white;
51   text-decoration: none;
52 }
53 .navlink:hover {
54   text-decoration: underline;
55 }
56 .navbutton{
57   margin-left: 0px;
58   font-size: 20px;
59   background-color: crimson;
60   padding: 10px;
61   border-radius: 10px;
62   color: white;
63   text-decoration: none;
64 }
65 table, th, td {
66   padding: 10px;
67 }
68 table{
69   text-align: center;
70 }
71 .bigtext{
72   font-family: "Arial Black", Gadget, sans-serif;
73   font-size: 80px;
74   font-weight: bold;
75 }
76 .hugetext{
77   font-family: "Arial Black", Gadget, sans-serif;
78   font-size: 150px;
79   font-weight: bold;
80   text-transform: capitalize;
81 }
82 .credit{
83   left: 0;
84   right: 0;
85   text-align: center;
86   font-size: 15px;
87   bottom: 0;
88   position: relative;
89   background-color: maroon;
90   padding: 0.1px;
91   color: white;
92 }
93 .mybutton{
94   font-size: 40px;
95   background-color: crimson;
96   padding: 20px;
97   border-radius: 50px;
98   color: white;
99   text-decoration: none;
100 }

```

**Left Panel (Original CSS):**

- `.navbar`: Contains a pseudo-element `:before` with a URL for the logo and a zoom value of 30%.
- `.bar`: A class with `word-spacing: 20px`, `text-align: center`, `font-size: 30px`, `color: white`, and `margin-bottom: 0px`.
- `.navlink`: A class with `color: white` and `text-decoration: none`.
- `.navlink:hover`: A class with `text-decoration: underline`.
- `.navbutton`: A class with `margin-left: 0px`, `font-size: 20px`, `background-color: crimson`, `padding: 10px`, `border-radius: 10px`, `color: white`, and `text-decoration: none`.
- `table, th, td`: A selector with `padding: 10px`.
- `table`: A selector with `text-align: center`.

**Right Panel (Modified CSS):**

- `.bigtext`: A class with `font-family: "Arial Black", Gadget, sans-serif`, `font-size: 80px`, and `font-weight: bold`.
- `.hugetext`: A class with `font-family: "Arial Black", Gadget, sans-serif`, `font-size: 150px`, `font-weight: bold`, and `text-transform: capitalize`.
- `.credit`: A class with `left: 0`, `right: 0`, `text-align: center`, `font-size: 15px`, `bottom: 0`, `position: relative`, `background-color: maroon`, `padding: 0.1px`, and `color: white`.
- `.mybutton`: A class with `font-size: 40px`, `background-color: crimson`, `padding: 20px`, `border-radius: 50px`, `color: white`, and `text-decoration: none`.

Figure 10.2 and 10.3: Source code for mycss.css

```

102 .mycircle{
103     padding: 10px;
104     padding-left: 20px;
105     padding-right: 20px;
106     font-size: 30px;
107     line-height: 0;
108     background-color: salmon;
109     border-radius: 100%;
110     color: white;
111 }
112 .mymovie{
113     border-radius: 10px;
114     word-wrap: break-word;
115     background-color: antiquewhite;
116     border: none;
117     color: black;
118     padding: 10px 10px;
119     text-decoration: none;
120     cursor: pointer;
121     height: 100px;
122     width: 600px;
123     font-size: 40px
124 }
125
126

```

Figure 10.4: Source code for mycss.css

*Explanation:*

- body is to set the body equals to full size of the page.
- .main is to set the main content container in the middle, with full 100% size, font ariel 30px, with a minimum height and in the middle of the page.
- .bg is to set an opaque background image that only blurs the image not the other contents, set to fill entire main container of the page and behind all other contents.
- .navbar, .bar, .navlink, .navbutton are to set the navigation bar and all of its contents with a fixed font size, position at the top of page, fixed spacing, maroon background and margin accordingly. Moreover, .navbar:before is to set the IU logo on the left of the navbar and .navlink:hover is to remove the underline the the links.
- table, th , td are to set the layout and alignment of the tables use in all pages.
- .bigtext and .hugetext are just 2 types of text with different sizes.
- .credit is to define the decoration of the footing/credits bar with maroon background and white text in the middle.
- .mybutton and .mycircle are custom designs for the 2 buttons Login and Registration in the Home page with crimson background and rounded borders (even turn into a circle).
- .mymovie is for the movie title buttons on the Review page, with antiquewhite background and a fixed font, no underline and a fixed size.

## IV. USE CASE DIAGRAM

### 1) Diagram:

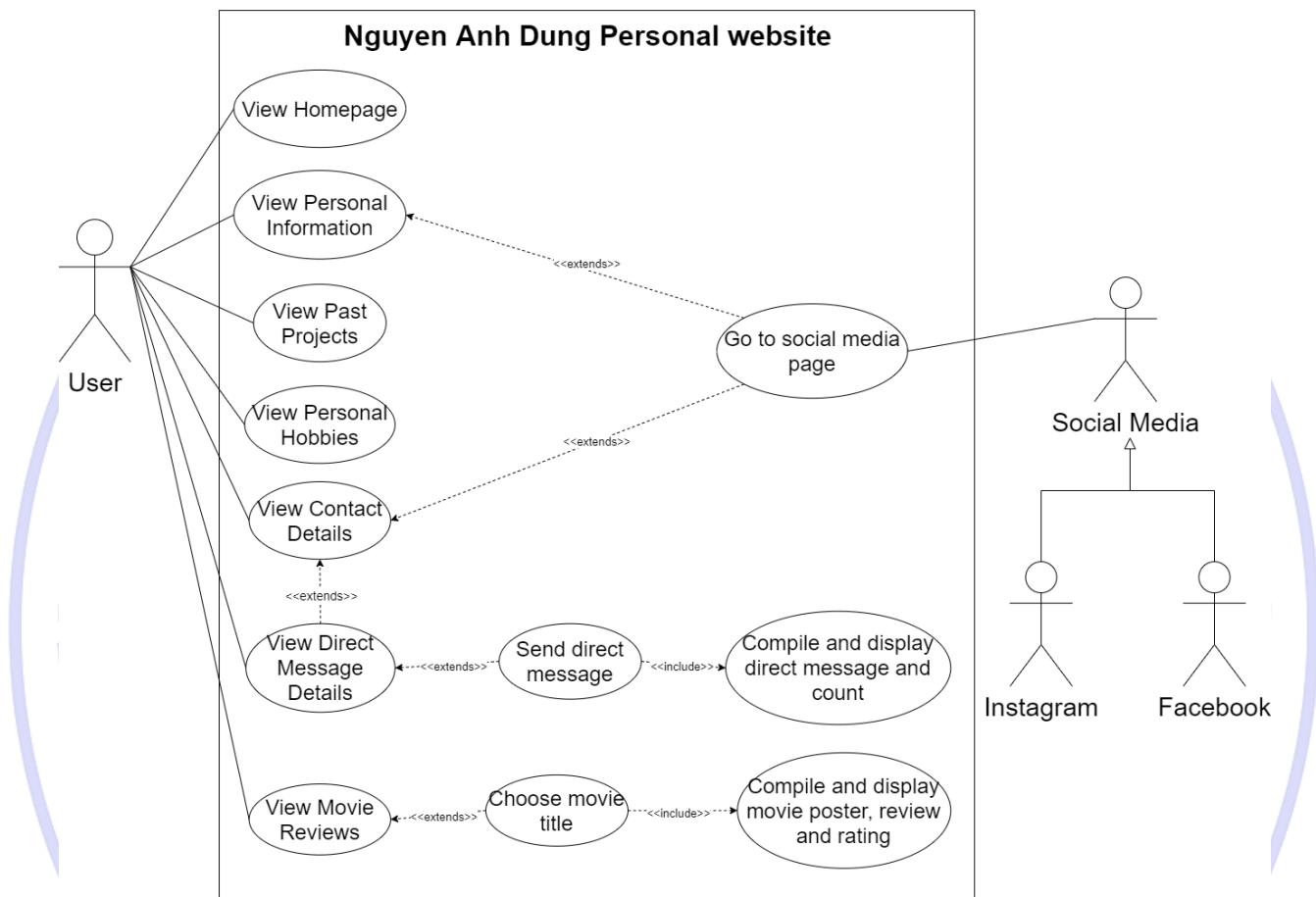


Figure 1: Nguyen Anh Dung Personal Website use case diagram.

- Actors: User, Social Media (Instagram and Facebook)
- Use cases:
  - View Homepage,
  - View Personal Information
  - View Past Project
  - View Personal Hobbies
  - View Contact Details
  - Go to social media page (extends from both View Personal Information and View Contact details)
  - View Direct Message Details (extends from View Contact Details), Send direct message (extends from View Direct Message Details), Compile and display direct message and count (included in Send direct message)

- View Movie Reviews, Choose movie title (extends from View Movie Reviews), Compile and display movie poster, review and rating (included in Choose movie title).

## 2) Explanation:

- The actors that involves in using this website are only the main user who views the pages and the social medias that the sites linked to (because social medias are their own external systems). Most of the functionalities on this website are just displaying or back-end server data handling so no other actors are required, even admins. Facebook and Instagram are specializations of the Social Media actor.
- There are total of 7 pages (not counting the login and registration pages in part V) corresponding with 7 use cases of the user viewing those pages. Technically each of the pages have the navigation bar that have the links to the others, so they can extends from each other, but that will make the diagram too complicated so let's assume that each page is separate and are functionalities themselves.
- However, the Contacts page and the About page do contain links to my Facebook and Instagram so there is a use case that extends from them which represent the action of going to the social media pages. This use case is connected to the Social Media (and Instagram/Facebook) actor.
- The 5 html pages are static, meaning they just displaying content, so they don't have any extra functionality. The 2 jsp pages are dynamic and are connected to their own servlets which do some functions so they will have a few use cases extends from them.
- The Contacts page, despite being a static web page, also have a link to the Direct page, but since we already have a use case for the Direct page (View Direct Message Details), let that use case be extends from the View Contact Details use case.
- The Direct page allows user to send a direct message and then the server will compile the message, the total message count and name to send back. By this logic, there will be 1 use case extends from the View Direct Message Details use case that will handles sending the direct message to the servlet and another use case which is included in the extended use case, which handles the compilation of the data and send back. These two are: "Send direct message" and "Compile and display direct message and count".
- The Review page allows user to click on a movie title/button to display information on that movie. By this logic, there will be 1 use case extends from the View Movie Reviews use case that will handles the act of choosing a movie title then send that data to the servlet and another use case which is included in the extended use case, which handles the compilation of the data and send back the corresponding output. These two are: "Choose movie title" and "Compile and display movie poster, review and rating".

## V. LOGIN AND REGISTRATION PAGE

### 1) Assumption:

As state in the first part's assumption, I designed a login and registration page in advanced but they're practically non-functional up until part IV, so the previous diagrams and structures do not include them. Now as we move into part V, I would like to point out that while the requirement is only

the right login and password can get you to the registration page, I will still keep the link to the registration page throughout for fast checking. Other than that, there will still be an authentication in the LoginPage.jsp (using servlet) which will only grant access to the registration page with the right username and password.

Moreover, in the question regards to the JavaBean, I will need to create a test.jsp to test the property handling capability of the JavaBean. Initially, the Registration.jsp file will be directed to the Store.java Servlet in question b and c, but to test the JavaBean I will slightly change the code to direct it to test.jsp, and the test.jsp is solely for testing the JavaBean so there will be no decoration or css to it, just raw data.

## **2) LoginPage.jsp and CheckLogin Servlet:**

- Requirement: Design the LoginPage.jsp and use a Servlet (CheckLogin) to check: if login successfully with username (midtermOnline) and password (23042020), then go to the registration page (Registration.jsp, question b); else, print-out “Sorry username or password error !”.
- Implementation: The LoginPage.jsp will be embedded within the website along with other pages, using the same layout format and css, only changing in the content. The main content of this page will be a login window in the middle of the page that prompt the user to enter their username and password. Before submitting anything, the page will create an attribute called “login” in the session with the value set to “” (blank) if that attribute is null and try to print it below the submit button. After the user submitted, the form will be directed to the CheckLogin.java Servlet where there the username and password will be check if they’re match with “midtermOnline” and “23042020”. If username and password doesn’t match, the servlet will call the attribute “login” and set it to “Sorry username or password error !” and directed back to the LoginPage.jsp, this will let jsp print out the “Sorry...” message because “login” is not null now. If username and password does match, the servlet will set “login” to “Successful” and redirect to Registration.jsp rather than back to the login page.

a) Source code:

The image shows two panels of a Java IDE displaying the source code for `LoginPage.jsp`. The top panel shows the initial structure of the page, including the header, navigation bar, and main content area. The bottom panel shows the detailed implementation of the login form and session handling logic.

```

Figure 2.1 and 2.2: Source code for LoginPage.jsp

```

**Panel 1 (Top):**

```

1 <%@page import="java.util.Date"%>
2 <!DOCTYPE html>
3 <html>
4     <head>
5         <title>Direct messages</title>
6         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
7         <link rel="stylesheet" type="text/css" href="mycss.css"/>
8     </head>
9     <body>
10        <div class="navbar">
11            <div class="bar">
12                <a class="navlink" href="index.html">Home</a>
13                <a class="navlink" href="about.html">About</a>
14                <a class="navlink" href="projects.html">Projects</a>
15                <a class="navlink" href="hobbies.html">Hobbies</a>
16                <a class="navlink" href="reviews.jsp">Reviews</a>
17                <a class="navlink" href="contacts.html">Contacts</a>
18                <a class="navlink" href="direct.jsp">Direct</a>
19            </div>
20            <div class="bar" style="float: right">
21                <a class="navbutton" href="LoginPage.jsp">Login</a>
22                <a class="navbutton" href="Registration.jsp">Register</a>
23            </div>
24        </div>
25        <div class="main">
26            
27            <br>
28            <h1 class="bigtext" style="line-height: 0.1;">
29                This is the login page!
30            </h1>
31            <br>
32            <table style="margin: 0 auto; background: white; padding: 10px; width: 30%; line-height: 0.1">
33                <tr>
34                    <td><h1>Login</h1></td>
35                </tr>
36            </table>
37        </div>
38    </body>
39 </html>

```

**Panel 2 (Bottom):**

```

36             <tr style="text-align: left">
37                 <td>
38                     <form method="post" action="CheckLogin">
39                         <p>Name</p>
40                         <input type="text" name="name" style="width: 100%; font-size: 30px; height: 50px;">
41                         <p>Password</p>
42                         <p><input type="password" name="pass" style="width: 100%; font-size: 30px; height: 50px;"></p>
43                         <input type="submit" value="Login" style="font-size: 30px;">
44                     </form>
45                 </td>
46             </tr>
47             <tr style="line-height: 2;">
48                 <% String s = (String)session.getAttribute("login");
49                     if(s==null){
50                         session.setAttribute("login","");
51                     }
52                 %>
53                 <td>
54                     <i style="color: red"><%=session.getAttribute("login")%</i>
55                 </td>
56             </tr>
57         </table>
58         <br>
59     </div>
60     <div class="credit">
61         <p style="line-height: 0.1;">Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
62     </div>
63 </body>
64 </html>

```

```

1 /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5 */
6
7 import java.io.IOException;
8 import java.io.PrintWriter;
9 import javax.servlet.ServletException;
10 import javax.servlet.annotation.WebServlet;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14 import javax.servlet.http.HttpSession;
15
16 /**
17  * 
18  * @author lenovo
19  */
20 @WebServlet(urlPatterns = {"/CheckLogin"})
21 public class CheckLogin extends HttpServlet {
22
23     /**
24      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
25      * methods.
26      *
27      * @param request servlet request
28      * @param response servlet response
29      * @throws ServletException if a servlet-specific error occurs
30      * @throws IOException if an I/O error occurs
31     */
32     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
33             throws ServletException, IOException {
34         response.setContentType("text/html;charset=UTF-8");
35         try (PrintWriter out = response.getWriter()) {
36
37             /* TODO output your page here. You may use following sample code. */
38             HttpSession session = request.getSession(true);
39             String n = (String)request.getParameter("name");
40             String p = (String)request.getParameter("pass");
41             if(n.equals("midtermonline") && p.equals("23042020")){
42                 session.setAttribute("login", "Successful");
43                 response.sendRedirect("Registration.jsp");
44             }
45             else{
46                 session.setAttribute("login", "Sorry username or password error !");
47                 response.sendRedirect("LoginPage.jsp");
48             }
49         }
50     }
51
52     //HttpServlet methods. Click on the + sign on the left to edit the code.
53
54 }

```

Figure 2.3 and 2.4: Source code for CheckLogin.java Servlet

b) Demo:

You can see from the demo that there's a login window in the middle of the page that prompt the user to enter username and password. If incorrect, the message "Sorry..." will appear in red at the bottom, if not then the user is taken to the Registration page in question b.

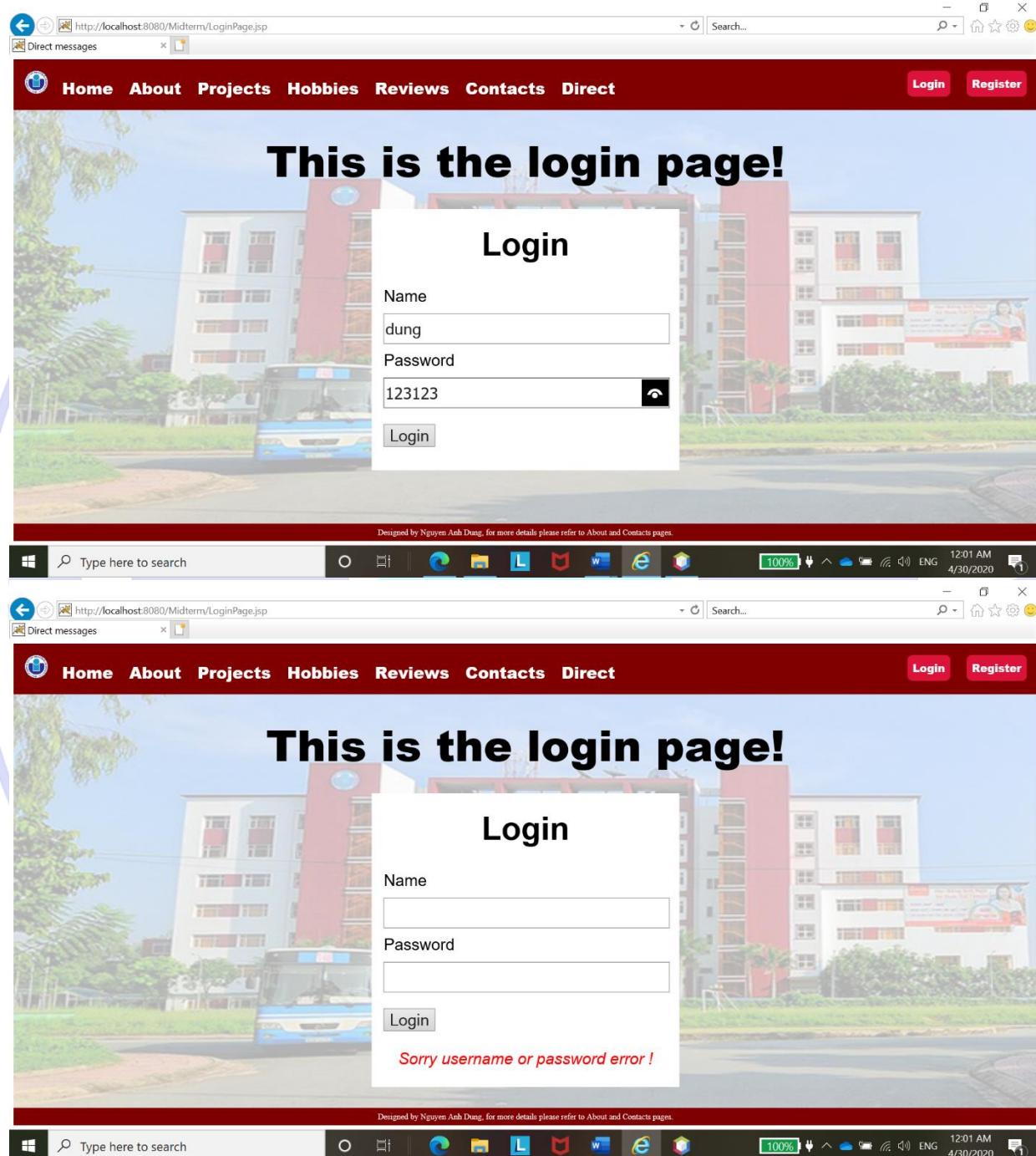


Figure 2.5 and 2.6: Demo for Login page before and after submitting the wrong username and password

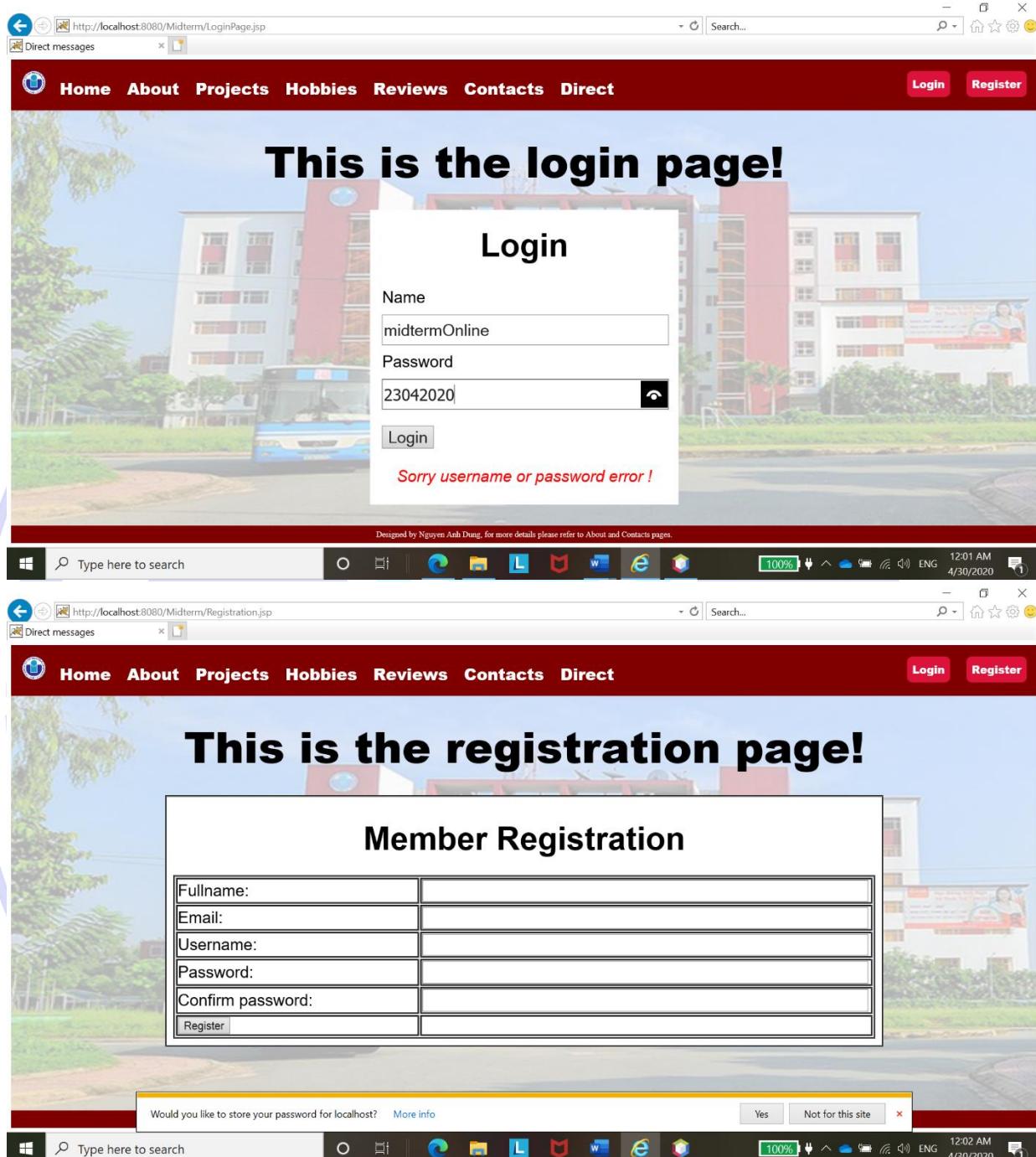
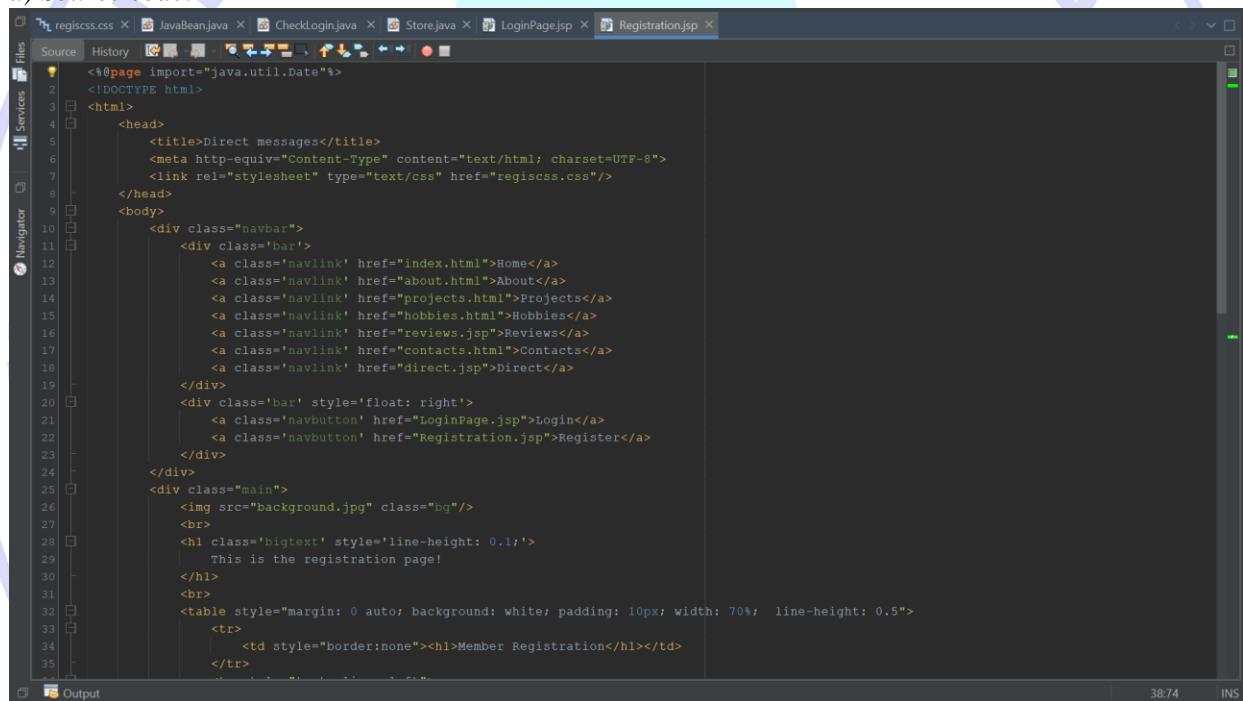


Figure 2.5 and 2.6: Demo for Login page before and after submitting the accurate username and password (taken to the Registration page)

### 3) Registration.jsp:

- Requirement: Write source code of the page Registration.jsp as in the Figure 1 (Member Registration) using a data form. (show source code). The format of this page is also used from the .css page.
- Implementation: The Registration.jsp will also be embedded within this website along with other pages, so it will also have the same layout and format but different content. Also, this page will have its own css called regiscss.css which is just a copy of the mycss.css but with a different table style to match the one in the figure 1. The main content will be a table that has a top row with the title and the bottom with a smaller table inside that contains input requirements and a submit button on the left columns and the data form to get the user input on the right columns. The big table will be in the middle of the page with white background, groove border and the inner table be the same but with inset border style rows and columns.

#### a) Source code:



```

<%@page import="java.util.Date"%>
<!DOCTYPE html>
<html>
    <head>
        <title>Direct messages</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <link rel="stylesheet" type="text/css" href="regiscss.css"/>
    </head>
    <body>
        <div class="navbar">
            <div class='bar'>
                <a class="navlink" href="index.html">Home</a>
                <a class="navlink" href="about.html">About</a>
                <a class="navlink" href="projects.html">Projects</a>
                <a class="navlink" href="hobbies.html">Hobbies</a>
                <a class="navlink" href="reviews.jsp">Reviews</a>
                <a class="navlink" href="contacts.html">Contacts</a>
                <a class="navlink" href="direct.jsp">Direct</a>
            </div>
            <div class='bar' style='float: right'>
                <a class="navbutton" href="LoginPage.jsp">Login</a>
                <a class="navbutton" href="Registration.jsp">Register</a>
            </div>
        </div>
        <div class="main">
            
            <br>
            <h1 class='bigtext' style='line-height: 0.1;'>
                This is the registration page!
            </h1>
            <br>
            <table style="margin: 0 auto; background: white; padding: 10px; width: 70%; line-height: 0.5">
                <tr>
                    <td style="border:none"><h1>Member Registration</h1></td>
                </tr>
            </table>
        </div>
    </body>
</html>

```

Figure 3.1: Source code for Registration.jsp

```

36 <tr style="text-align: left">
37   <td style="padding:0px; border-style: groove">
38     <form method="post" target="_blank" action="Store" >
39       <table style=" text-align: left; width: 100%; border: none" >
40         <tr>
41           <td style="width: 35%">Fullname:</td>
42           <td><input type="text" name="fname" style="width: 99%; font-size: 30px"></td>
43         </tr>
44         <tr>
45           <td style="width: 35%">Email:</td>
46           <td><input type="email" name="email" style="width: 99%; font-size: 30px"></td>
47         </tr>
48         <tr>
49           <td style="width: 35%">Username:</td>
50           <td><input type="text" name="user" style="width: 99%; font-size: 30px;"></td>
51         </tr>
52         <tr>
53           <td style="width: 35%">Password:</td>
54           <td><input type="password" name="pass" style="width: 99%; font-size: 30px"></td>
55         </tr>
56         <tr>
57           <td style="width: 35%">Confirm password:</td>
58           <td><input type="password" name="cpass" style="width: 99%; font-size: 30px"></td>
59         </tr>
60         <tr>
61           <td style="width: 35%"><input type="submit" value='Register' style='font-size: 20px'></td>
62           <td></td>
63         </tr>
64       </table>
65     </form>
66   </td>
67 </tr>
68 </table>
69 <br>
70 </div>
71 <div class='credit'>
72   <p style="line-height: 0.1;">Designed by Nguyen Anh Dung, for more details please refer to About and Contacts pages.<p>
73 </div>
74 </body>
75 </html>

```

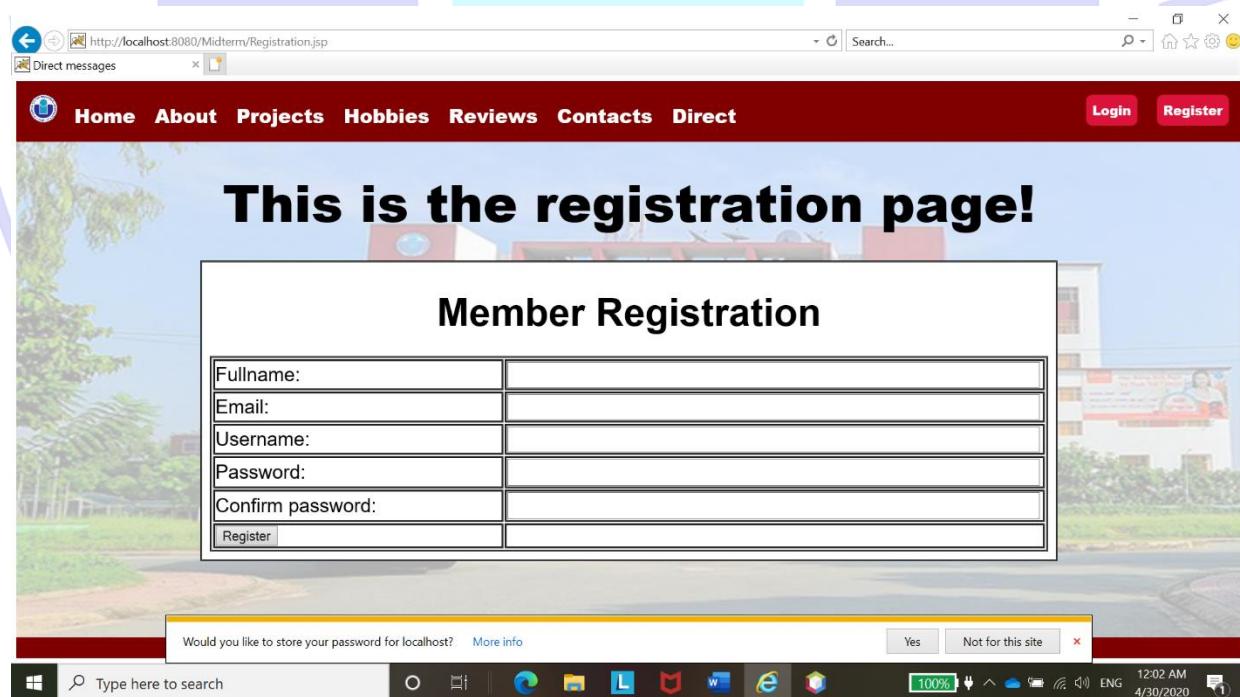
Figure 3.2 and 3.3: Source code for Registration.jsp

```
regiscss.css X JavaBean.java X CheckLogin.java  
Source History  Files Services Navigator  
1 border-radius: 10px;  
2 color: white;  
3 text-decoration: none;  
4 }  
5 td{  
6 border-style: inset;  
7 line-height: 0.8;  
8 }  
9 table{  
10 border-style: groove;  
11 text-align: center;  
12 }  
13 
```

Figure 3.4: Source code for regiscss.css (same as mycss.css only different in td,th and table style). Refer to mycss.css for the rest of the code.

b) Demo:

You can see from the demo that there's a registration table in the middle of the page that prompt the user to enter username, password, confirm password, fullname, email and a Register button.

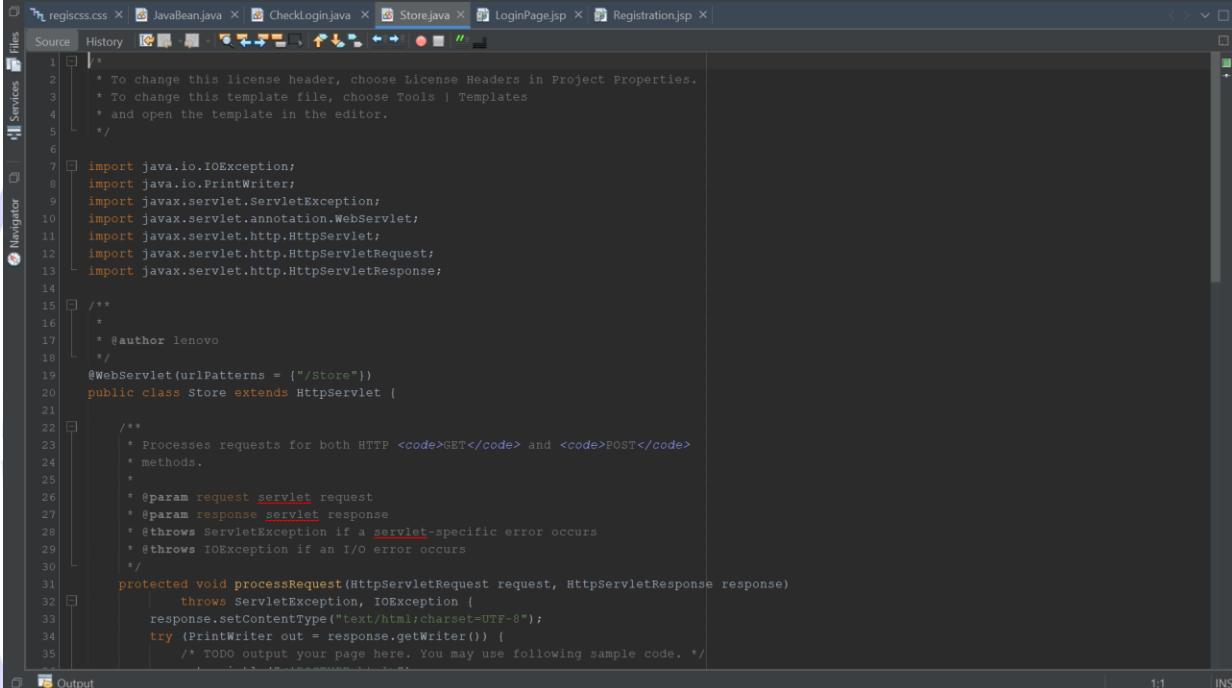


*Figure 3.5: Demo for the Registration.jsp page*

#### 4) Store Servlet:

- Requirement: After click the button “Register” (in question b), it will call a Servlet “Store”. Write source code for this Servlet to get all information in the data form of question b and display in a new page (defined by using html in the servlet. (show source code and interface).
- Implementation: The Store.java Servlet will be display in a new page and defined using html in the Servlet. This can be achieve by setting “target = \_blank” in the form in Registration.jsp and call request.getParameter() on the data that we want to get in the form. The servlet will be hard to decorate using html in servlet so I will only make a simple table with background color that display the information gathered.

##### a) Source code:



```

1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  import java.io.IOException;
8  import java.io.PrintWriter;
9  import javax.servlet.ServletException;
10 import javax.servlet.annotation.WebServlet;
11 import javax.servlet.http.HttpServlet;
12 import javax.servlet.http.HttpServletRequest;
13 import javax.servlet.http.HttpServletResponse;
14
15 /**
16  *
17  * @author lenovo
18  */
19 @WebServlet(urlPatterns = {" /Store"})
20 public class Store extends HttpServlet {
21
22     /**
23      * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
24      * methods.
25      *
26      * @param request servlet request
27      * @param response servlet response
28      * @throws ServletException if a servlet-specific error occurs
29      * @throws IOException if an I/O error occurs
30      */
31     protected void processRequest(HttpServletRequest request, HttpServletResponse response)
32             throws ServletException, IOException {
33         response.setContentType("text/html; charset=UTF-8");
34         try (PrintWriter out = response.getWriter()) {
35             /* TODO output your page here. You may use following sample code. */
36         }
37     }
38
39 }

```

Figure 4.1: Source code for Store.java Servlet

```

36     out.println("<!DOCTYPE html>");
37     out.println("<html>");
38     out.println("<head>");
39     out.println("<title>Registration Information</title>");
40     out.println("</head>");
41     out.println("<body>");
42     out.println("<table style='margin: 0 auto; background-color: antiquewhite'>");
43     out.println("<tr>");
44     out.println("<td>Member registration data:</td>");
45     out.println("</td>");
46     out.println("</tr>");
47     out.println("<tr>");
48     out.println("<td>Fullname: <b></b></td>");
49     out.println("<td>" + request.getParameter("fname") + "</td>");
50     out.println("</tr>");
51     out.println("<tr>");
52     out.println("<td>Email: <b></b></td>");
53     out.println("<td>" + request.getParameter("email") + "</td>");
54     out.println("</tr>");
55     out.println("<tr>");
56     out.println("<td>Username: <b></b></td>");
57     out.println("<td>" + request.getParameter("user") + "</td>");
58     out.println("</tr>");
59     out.println("<tr>");
60     out.println("<td>Password: <b></b></td>");
61     out.println("<td>" + request.getParameter("pass") + "</td>");
62     out.println("</tr>");
63     out.println("<tr>");
64     out.println("<td>Confirm password: <b></b></td>");
65     out.println("<td>" + request.getParameter("cpass") + "</td>");
66     out.println("</tr>");
67     out.println("</table>");
68     out.println("</body>");
69     out.println("</html>");

60     out.println("<td>Username: <b></b></td>");
61     out.println("<td>" + request.getParameter("user") + "</td>");
62     out.println("</tr>");
63     out.println("<tr>");
64     out.println("<td>Password: <b></b></td>");
65     out.println("<td>" + request.getParameter("pass") + "</td>");
66     out.println("</tr>");
67     out.println("<tr>");
68     out.println("<td>Confirm password: <b></b></td>");
69     out.println("<td>" + request.getParameter("cpass") + "</td>");
70     out.println("</tr>");
71     out.println("</table>");
72     out.println("</td>");
73     out.println("</tr>");
74     out.println("<table>");
75     out.println("<tbody>");
76     out.println("<tr>");
77     out.println("    <td>HttpServlet methods. Click on the + sign on the left to edit the code.</td>");
78   }
79 }
80 
```

Figure 4.2 and 4.3: Source code for Store.java Servlet

### b) Demo:

As you can see from the demo below, when you input the data and hit Register, the browser will open another tab that shows the information that you input in a basic html table by using Servlet.

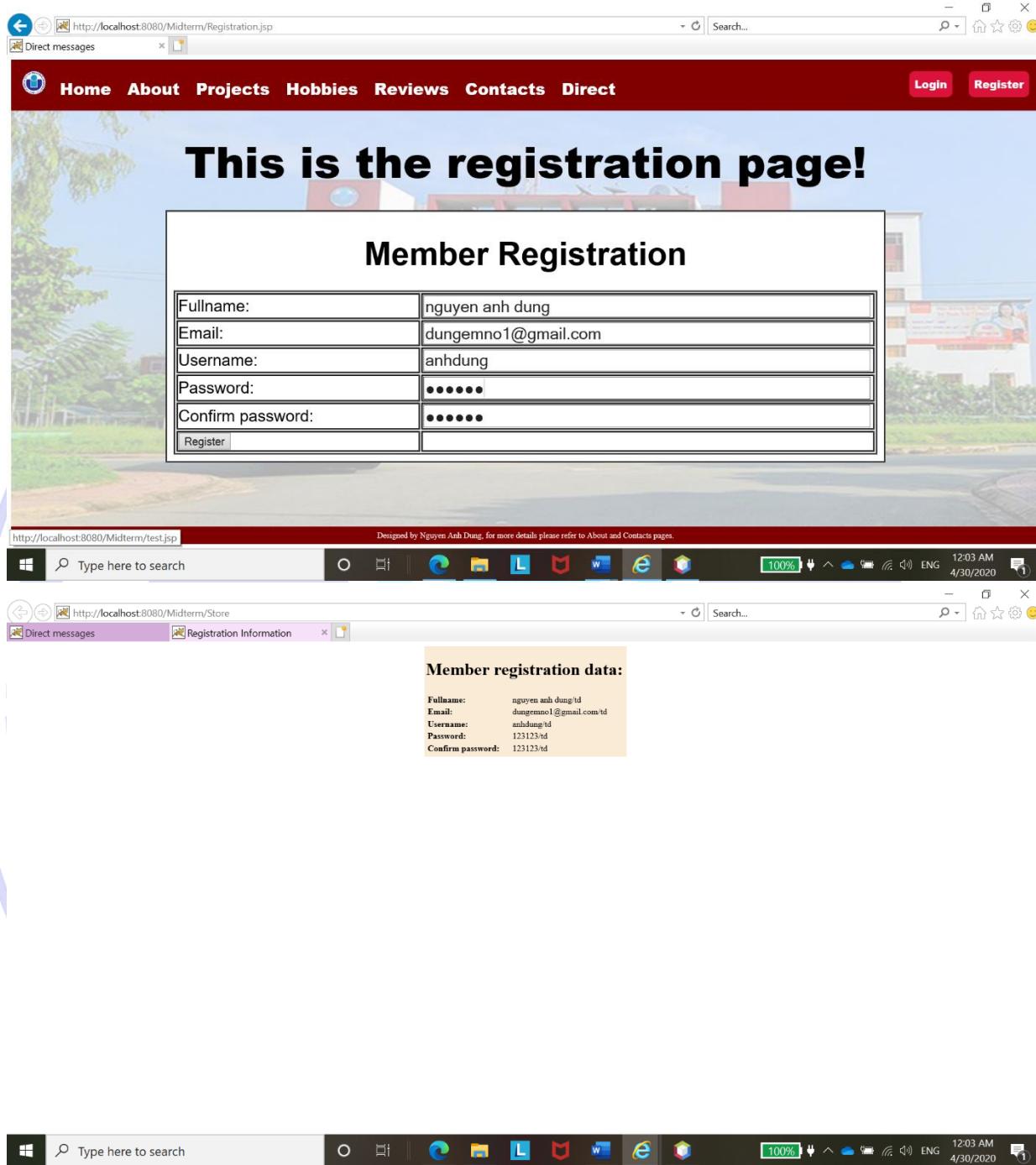
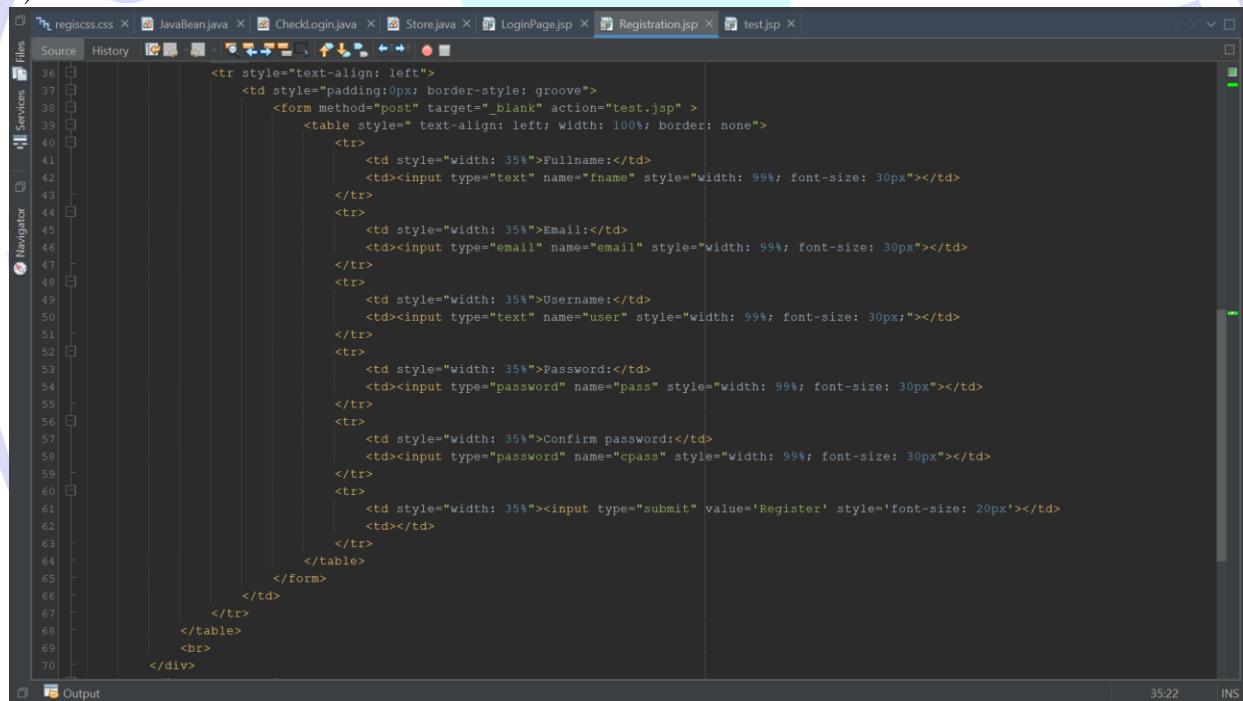


Figure 4.4 and 4.5: Before and after submitting the data to the Servlet

## 5) JavaBean:

- Requirement: Write a JavaBean that can handle all properties of the Registration.jsp.
- Implementation: For this requirement, as mention in the assumption of this part, along side the JavaBean itself I will also create a jsp called test.jsp to test the property handling capability of the JavaBean. The test.jsp will be created very simple with raw data output, no css do style no decoration because its purpose is only to test the output. Moreover, I will have to change the form action of the Registration.jsp from the Store.java Servlet to test.jsp so the JavaBean in test.jsp can handle the data in Registration.jsp. The JavaBean that I created will be named JavaBean.java with declare variables to match the data needed to collect, having a constructor and all getter setter to handle the properties. JavaBean.java will be created in a package called “bean”. To make it simple, I will also be using the name of the input types in the form as the name for the declare variables as well as the properties so I don’t have to set parameters myself. Registration.jsp will send data via the form to test.jsp, test.jsp will implement the JavaBean.java to initially set all of the properties to their according data. Then it will get all the properties that are set and display them on the page.

a) Source code:



```

36         <tr style="text-align: left">
37             <td style="padding:0px; border-style: groove">
38                 <form method="post" target="_blank" action="test.jsp" >
39                     <table style=" text-align: left; width: 100%; border: none">
40                         <tr>
41                             <td style="width: 35%">Fullname:</td>
42                             <td><input type="text" name="fname" style="width: 99%; font-size: 30px"></td>
43                         </tr>
44                         <tr>
45                             <td style="width: 35%">Email:</td>
46                             <td><input type="email" name="email" style="width: 99%; font-size: 30px"></td>
47                         </tr>
48                         <tr>
49                             <td style="width: 35%">Username:</td>
50                             <td><input type="text" name="user" style="width: 99%; font-size: 30px;"></td>
51                         </tr>
52                         <tr>
53                             <td style="width: 35%">Password:</td>
54                             <td><input type="password" name="pass" style="width: 99%; font-size: 30px"></td>
55                         </tr>
56                         <tr>
57                             <td style="width: 35%">Confirm password:</td>
58                             <td><input type="password" name="cpass" style="width: 99%; font-size: 30px"></td>
59                         </tr>
60                         <tr>
61                             <td style="width: 35%"><input type="submit" value='Register' style="font-size: 20px"></td>
62                         </tr>
63                     </table>
64                 </form>
65             </td>
66         </tr>
67     </table>
68     <br>
69 </div>
70

```

Figure 5.1: Source code for Registration.jsp (again). Notice the action= "test.jsp"

The image shows two side-by-side Java code editors from an IDE. Both editors have a dark theme and a top navigation bar with tabs for various files like regcss.css, JavaBean.java, CheckLogin.java, Store.java, LoginPage.jsp, Registration.jsp, and test.jsp.

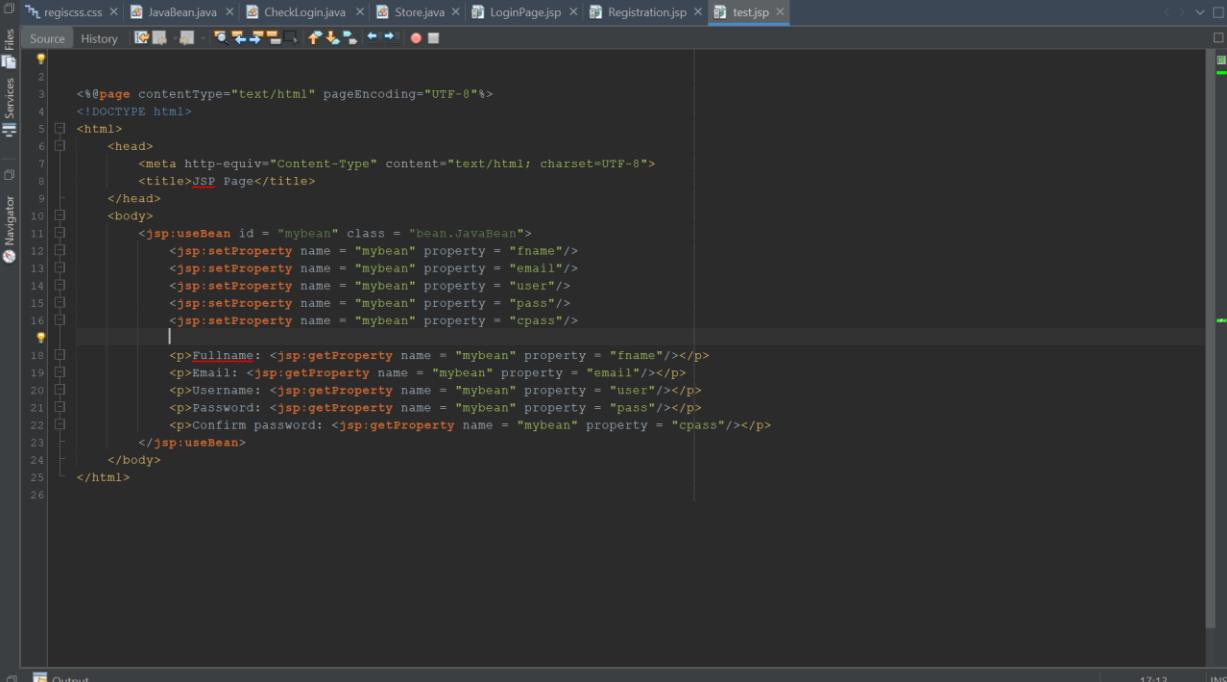
**Editor 1 (Left):**

```
1 package bean;
2 public class JavaBean {
3     String fname;
4     String user;
5     String pass;
6     String cpass;
7     String email;
8
9     public JavaBean() {
10 }
11
12     public String getFname() {
13         return fname;
14     }
15
16     public void setFname(String fname) {
17         this.fname = fname;
18     }
19
20     public String getUser() {
21         return user;
22     }
23
24     public void setUser(String user) {
25         this.user = user;
26     }
27
28     public String getPass() {
29         return pass;
30     }
31
32     public void setPass(String pass) {
33         this.pass = pass;
34     }
35 }
```

**Editor 2 (Right):**

```
32     public void setPass(String pass) {
33         this.pass = pass;
34     }
35
36     public String getCpass() {
37         return cpass;
38     }
39
40     public void setCpass(String cpass) {
41         this.cpass = cpass;
42     }
43
44     public String getEmail() {
45         return email;
46     }
47
48     public void setEmail(String email) {
49         this.email = email;
50     }
51
52
53
54 }
```

Figure 5.2 and 5.3: Source code for JavaBean.java



```
1 <%@page contentType="text/html" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <title>JSP Page</title>
7   </head>
8   <body>
9     <jsp:useBean id = "mybean" class = "bean.JavaBean">
10       <jsp:setProperty name = "mybean" property = "fname"/>
11       <jsp:setProperty name = "mybean" property = "email"/>
12       <jsp:setProperty name = "mybean" property = "user"/>
13       <jsp:setProperty name = "mybean" property = "pass"/>
14       <jsp:setProperty name = "mybean" property = "cpass"/>
15     |
16     <p>Fullname: <jsp:getProperty name = "mybean" property = "fname"/></p>
17     <p>Email: <jsp:getProperty name = "mybean" property = "email"/></p>
18     <p>Username: <jsp:getProperty name = "mybean" property = "user"/></p>
19     <p>Password: <jsp:getProperty name = "mybean" property = "pass"/></p>
20     <p>Confirm password: <jsp:getProperty name = "mybean" property = "cpass"/></p>
21   </jsp:useBean>
22   </body>
23 </html>
```

Figure 5.4: Source code for test.jsp

**b) Demo:**

As you can see from the demo below, the registration page is the same as in the previous part. But when you click register with the input values, this time the Registration.jsp will call test.jsp which implemented JavaBean.java to display the data rather than the Store.java Servlet.

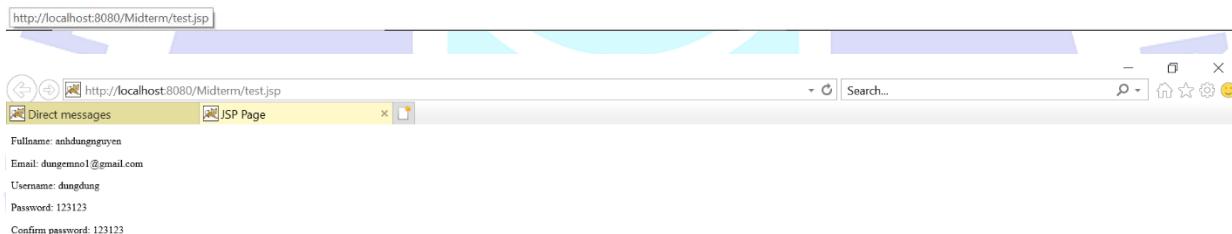
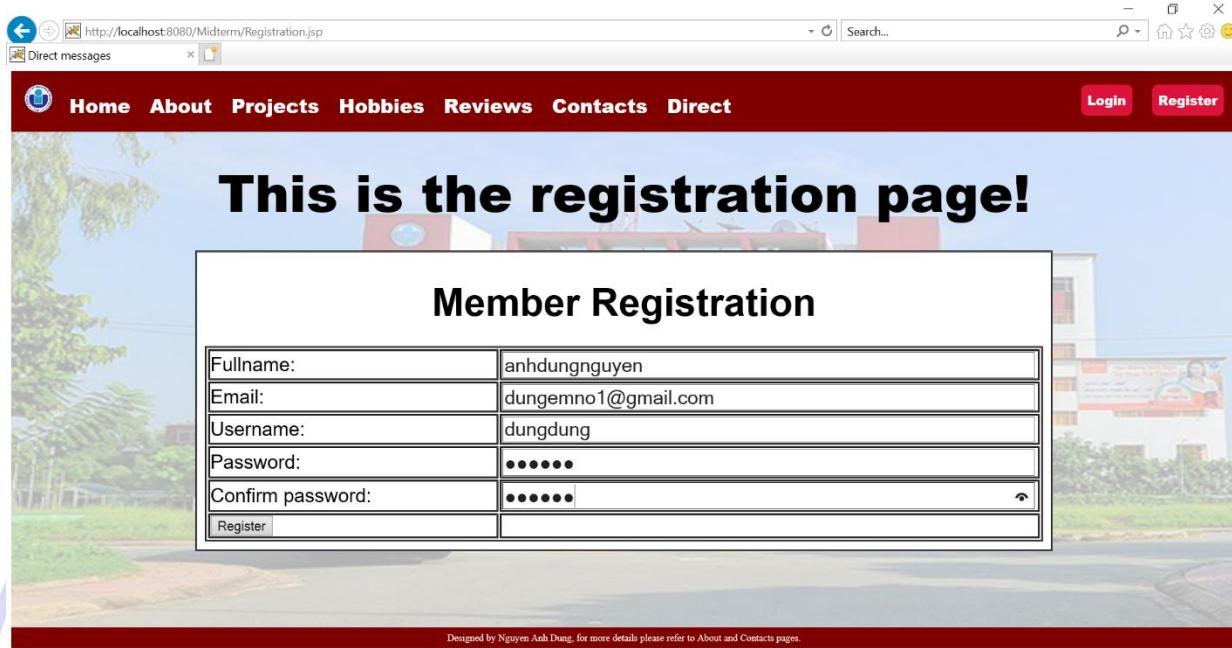


Figure 5.5 and 5.6: Demo for Registration.jsp sending data to test.jsp that implemented JavaBean.java to setProperty and getProperty to display them

## VI. REFERENCES

- Materials: Web Application Development lectures – Dr. Nguyen Van Sinh
- Drawing Use case and Structural design: draw.io
- External links: facebook.com, Instagram.com
- Debugging and problem solving:
  - <https://stackoverflow.com/questions/3024507/jasperexception-the-value-for-the-usebean-class-attribute-is-invalid/13512012>
  - <https://stackoverflow.com/questions/8329451/cant-use-jspusebean-bean-cannot-be-resolved-to-a-type>
  - <https://stackoverflow.com/questions/933320/javabean-value-for-the-usebean-class-attribute-classes-userdata-is-invalid>
  - (mostly stackoverflow) etc.
- Examples:
  - <https://way2java.com/jsp/title-2-javabean-in-jsp-client-jsp-javabean-example/>
  - [https://www.tutorialspoint.com/spring/spring\\_beans\\_scopes.htm](https://www.tutorialspoint.com/spring/spring_beans_scopes.htm)
  - [https://www.w3schools.com/html/html\\_classes.asp](https://www.w3schools.com/html/html_classes.asp)
  - [https://www.w3schools.com/tags/att\\_input\\_type\\_email.asp](https://www.w3schools.com/tags/att_input_type_email.asp)
  - etc.