

CSE 565 - Software Verification Validation and Testing

Assignment 5

Graphical User Interface Testing

Kedar Sai Nadh Reddy Kanchi

School of Computing and Augmented Intelligence
Arizona State University, Tempe
kkanchi@asu.edu

Version 1 GUI

I have made a web application using HTML and CSS for a user to register, login and give review of my application. I have salvaged the underlying code framework for a login form from a GitHub repository [8] and then worked on it to create the two additional pages for sign up and review.

Page 1

The Sign Up form is the first page of the application where the user can register themselves by providing specific information such as their username, email, phone number, gender, age, and a password for the account.

The form has a small user avatar image at the top and a title “Create Account” just below the image. The sign up form includes text input boxes for the fields username, email, phone number, password, and confirm password. The form contains a group of radio buttons for the gender field, allowing the user to select their sex, and a slider for the age field, allowing the user to set their age by sliding the rounded element left and right. There are also two buttons on the sign-up form:

- Sign Up Button:** When the user clicks on this button, the application will perform some validation based upon the data entered in the form and if all the validations have been successfully passed, then the application will create an account for the user and reroute them to the login page immediately so that the user can login into the application and leave a feedback on the review form page.
- Login Button:** If the user already has an account, and if he or she has landed on the sign up page by mistake, then the application also provides a login button at the bottom. When the user clicks on this button, the application will reroute them to the login page immediately so that the user can login into the application and leave a feedback on the review form page.

Page 2

The second page of the application form is the Login Form where the user can login by entering their username or email and the password. Similar, to the sign up form, the login form also has a small user avatar image at the top and a title “Login” just below the image. Apart from the image, title, input fields for login credentials, the login form has 4 other components which includes two buttons as well.

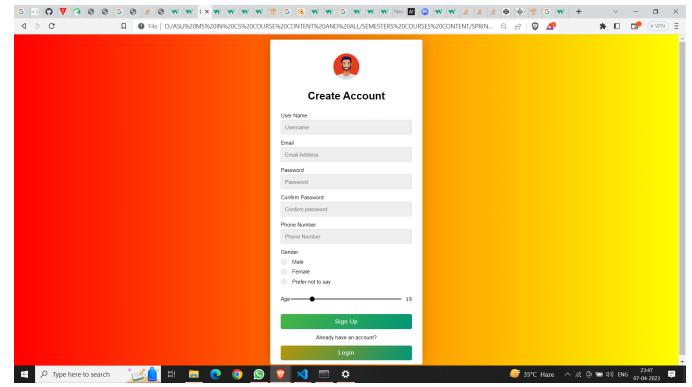


Figure 1: Form Page 1 : Registration / Sign Up Form

```
107 <div id="Page2">
108   <div class="contains">
109     <form class="Form" id="createAccount">
110       
111       <h1 class="Form__title" id="signup_create_account_header">
112         Create Account
113       </h1>
114       <div class="Form__message form__message--error"></div>
115       <div class="Form__group">
116         <span style="margin-bottom: 5px">
117           <label id="signup_userNameLabel"> User Name </label>
118           <input type="text" id="signup_username" class="Form__input" style="margin-top: 0.3rem" autofocus
placeholder="Username" required />
119           <div class="Form__input-error-message"></div>
120         </div>
121         <div class="Form__input-group">
122           <span style="margin-bottom: 5px">
123             <label id="signup_userEmailLabel"> Email </label>
124             <input type="text" class="Form__input" style="margin-top: 0.3rem" autofocus
placeholder="Email Address" id="signup_email" required />
125             <div class="Form__input-error-message"></div>
126           </div>
127         <div class="Form__input-group">
128           <span style="margin-bottom: 5px">
129             <label id="signup_userPassword_label"> Password </label>
130             <input type="password" class="Form__input" style="margin-top: 0.3rem" autofocus
placeholder="Password" id="signup_password" required />
131             <div class="Form__input-error-message"></div>
132           </div>
133         <div class="Form__input-group">
134           <span style="margin-bottom: 5px">
135             <label id="signup_userConfirmPassword_label"> Confirm Password </label>
136             <input type="password" class="Form__input" style="margin-top: 0.3rem" autofocus
placeholder="Confirm Password" id="signup_confirm_password" required />
137             <div class="Form__input-error-message"></div>
138           </div>
139         <div class="Form__input-group">
140           <span style="margin-bottom: 25px">
141             <label id="signup_userPhoneNumber_label" style="margin-bottom: 2px">
```

Figure 2: Code Part 1 for the Sign Up Form

- Scroll Box with a Label:** The user must read and accept the terms and conditions in this scroll box before logging in to the application. This scroll box is configured to match the width of the form as well as a predefined height, allowing the user to easily scroll through the information. There is a label “Please read the terms and conditions below” just above the scroll box which makes the user aware of what is the information specified with the scroll area.

```

151     <label id="signup_user_phone_number_label"> Phone Number </label>
152     <input type="text" class="form__input" style="margin-top: 0.3em" autofocus
153     placeholder="Phone Number" id="signup_user_phonenumber" required />
154     <div class="form__input-error-message"></div>
155   </div>
156
157   <div>
158     <label id="signup_gender_label">Gender </label>
159     <div>
160       <label class="radio-container" id="male_radio_label">
161         Male
162         <input type="radio" name="radio" id="male_radio_label_button" required />
163         <span class="radio-checkmark"></span>
164       </label>
165     </div>
166
167     <div>
168       <label class="radio-container" id="female_radio_label">
169         Female
170         <input type="radio" name="radio" id="female_radio_label_button" required />
171         <span class="radio-checkmark"></span>
172       </label>
173     </div>
174
175     <div>
176       <label class="radio-container" id="other_gender_radio_label">
177         Prefer not to say
178         <input type="radio" name="radio" id="other_gender_radio_label_button" required />
179         <span class="radio-checkmark"></span>
180       </label>
181     </div>
182   </div>
183
184
185   <div class="slider">
186     <div>
187       <label id="slider_label_for_ape">Age</label>
188       <input type="range" id="ape_range_slider" min="0" max="100" value="18" />
189       <span>rangeValue.innerText = this.value; />
190       <p id="rangeValue">18</p>
191     </div>
192   </div>
193
194   <div class="button" id="button_signup">
195     <a href="#" onclick="return validateSignupForm('Page1', 'Page2');">
196       <input type="submit" value="Sign Up" id="user_signup_button" />
197     </a>
198   </div>

```

Figure 3: Code Part 2 for the Sign Up Form

```

198   <div class="button" id="navigation_to_login_page">
199     <p class="form__text" id="label_for_login_button_on_sign_up_page">
200       Already have an account?
201     </p>
202     <a href="#" onclick="return show('Page1', 'Page2');">
203       <input type="submit" value="Login" id="login_button_on_sign_up_page" />
204     </a>
205   </div>
206 </div>
207 </div>
208 </div>

```

Figure 4: Code Part 3 for the Sign Up Form

- Radio Button with a Label:** This radio button is included in the form because the user can only login if he or she agrees to the application's terms and conditions. To do so, the user must select the radio button with the label "I agree to the terms and conditions" which informs the application that the user has accepted the terms and conditions of the application.
- Sign Up Button:** If the user does not already have an account and would like to create one, they can do so by clicking the Sign Up button at the very bottom of the form, which will take them to the Sign Up Page.
- Login Button:** When a user clicks the login button, the application validates the form to see if they have entered their credentials and checked the box. If these validations pass, the application will proceed to the review form page, where the user can provide feedback.

Page 3

The third page of this application is a review form. I created a review page after being inspired by the comments section of the ASU Canvas Grades Section. In this feedback form, I'm asking the user to rate this web application based on its UI, functionality implementation, and content. The feedback components are:

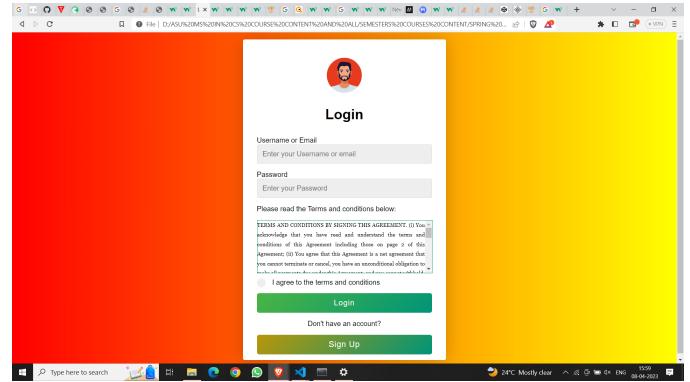


Figure 5: Form Page 2 : Login Form

```

20   <div id="Page1" style="display: none">
21     <div class="container">
22       <form class="form" id="login" action="#">
23         
24         <h1 Class="form__title" id="login_page_title">Login</h1>
25         <div class="form__message form__message--error"></div>
26         <div class="form__input-group">
27           <span style="margin-bottom: 5px;">
28             <label id="usernameEmail">Username or Email </label>
29           </span>
30           <input type="text" class="form__input" autofocus placeholder="Enter your Username or email" id="usernameEmail_input" required />
31           <div class="form__input-error-message"></div>
32         </div>
33         <div class="form__input-group">
34           <span style="margin-bottom: 5px;">
35             <label id="user_password">Password </label>
36           </span>
37           <input type="password" class="form__input" autofocus placeholder="Enter your Password" id="password_input" required />
38           <div class="form__input-error-message"></div>
39         </div>
40         <p id="label_for_user_login_terms_and_conditions">
41           Please read the Terms and conditions below:
42         </p>
43         <div class="scrollarea" id="login_scroll_area">
44           TERMS AND CONDITIONS BY SIGNING THIS AGREEMENT. (i) You acknowledge
45           that you have read and understand the terms and conditions of this
46           Agreement including those on page 2 of this Agreement; (ii) You agree
47           that this Agreement is a net agreement that you cannot terminate or
48           cancel, you have an unconditional obligation to make all payments due
49           under this Agreement, and you cannot withhold, set off or reduce such
50           payments for any reason; (iii) You agree to pay all amounts due for
51           business expenses and in compliance with all applicable laws and
52           regulations; (iv) you acknowledge that if this Agreement is replacing
53           an existing agreement, the new Installment Payment may include the
54           balance of that existing agreement and result in a greater aggregate
55           product cost to you; and (v) You agree that by providing a telephone
56           number to a cellular or other wireless device, you are expressly
57           consenting to receiving communications from us, our affiliates and
58           agents (for non-marketing purposes) at that number, including, but not
59           limited to, prerecorded and artificial voice messages, text messages,
60           and calls from automated telephone dialing systems; these calls may
61           incur fees from your cellular provider; and this consent applies to
62           each such telephone number you provide to us now or in the future
63           </div>
64       </div>

```

Figure 6: Code Part 1 for the Login Form

- Image and a Title with a SubTitle:** Similar, to the first two forms, the feedback form has a relevant image at the top and a title "We appreciate your review!" just below the image. But the feedback form also a subtitle "Your review will help me to improve in my future assignments" written below the title to inform the user as to why this review or feedback is important to us.
- Input Text Boxes:** So, like the first two forms, this feedback form too has two input text boxes for the username and email fields.
- Rating Header Label:** The form then has a kind of rating header label "Please rate your service experience for the following parameters" which asks the user to rate the applications for the following mentioned factors as they see fit.
- Rating sections:** There are a total 5 rating sections.
 - Rating Section 1:** The application asks the user to rate

```
  <!-- secondpage.html -->  <!-- firstpage.html -->  <!-- TestFile.py -->  <!-- checkbox.cs -->  <!-- demo.cs -->  <!-- radiobutton.cs -->  <!-- main.cs -->
  <!-- firstpage.html -->  <body>  <div>page1</div>  <div>container</div>  <form>login form</form>  <div>form_input_group</div>
  66    <div>
  67      <label class="radio-container">  <!-- radio_label_for_user_login_terms_and_conditions -->
  68        I agree to the terms and conditions
  69        <input type="radio" name="radio" id="login_terms_and_conditions" required />
  70        <span class="radio-checkmark"></span>
  71      </label>
  72    </div>
  73
  74    <div class="button" id="button_login">
  75      <a href="#" onclick="return validateForm('Page3','Page1', 'userNameOrEmail_input', 'password_input');">
  76        <input type="submit" value="Login" id="user_login_button" />
  77      </a>
  78    </div>
  79
  80    <div class="button" id="create">
  81      <p>  <!-- label_for_sign_up_on_login_page -->
  82        Don't have an account?
  83      </p>
  84      <a href="#" onclick="return show('Page2','Page1');">
  85        <input type="submit" value="Sign Up" id="sign_up_button_on_login_page" />
  86      </a>
  87    </div>
  88
  89  </form>
 90 </div>
 91 </div>
```

Figure 7: Code Part 2 for the Login Form

the application for the following factor “Selection of the testing tool for this assignment?”

- **Rating Section 2:** The application asks the user to rate the application for the following factor “Assessment of the submitted report in terms of set of grammar or formatting errors, information about the selected testing tool and the description of the Test Cases executed as part of this assignment?”
 - **Rating Section 3:** The application asks the user to rate the application for the following factor “How did you like the implementation of the application and the UI submitted as part of this assignment?”
 - **Rating Section 4:** The application includes a text area box in which the user can provide personal feedback on what they liked about the application and what could have been done better in terms of the web application as a whole, the test cases developed, and the submitted report.
 - **Rating Section 5:** The application includes a slider that asks the user to rate their overall experience with the web application, test cases developed, and report submitted on a scale of 1 to 5.
 - **Submit Review Button:** The user can click this button after providing all of the feedback inputs. The review is submitted and the user is logged out when you click this button, and the application returns to the login page.
 - **Logout Button:** If the user is not yet ready to leave feedback, he or she can click on this button, which will redirect the user to the login page, where he or she can re-login once ready to leave feedback.

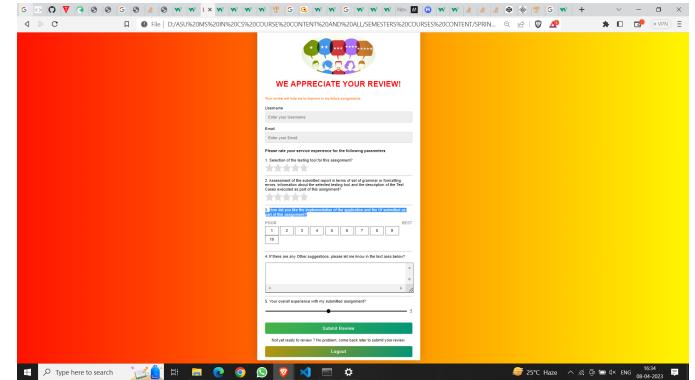


Figure 8: Form Page 3 : Feedback / Review Form

```
① secondpage.html ② firstpage.html ③ TestFilePage3 # checkbox.cs # demo.cs # radioButton.cs # main.cs # reviewpage.cs
④ firstpage.html > body > div#body > div#review_container.container > form#loginForm > div.form_input-group

  <div id="Page3" style="display: none">
    <div class="container" id="review_container">
      <form class="form" id="login_form" action="">
        
        <h1 class="form_title" id="review_page_title">
          Review  
ME APPRECIATE YOUR REVIEW!
        </h1>
        <h2 id="review_title_subtext">
          Your review will help me to improve in my future assignments.
        </h2>
        <div class="form_message form_message_error"></div>
        <div class="form_input-group">
          <span style="margin-bottom: 5px">
            <label id="review_username_label"> Username </label>
          </span>
          <input type="text" class="form_input" style="margin-top: 0.3em" autofocus
            placeholder="Enter your Username" id="review_username_input" required>
          <div class="form_input-error-message"></div>
        </div>
        <div class="form_input-group">
          <span style="margin-bottom: 5px">
            <label id="review_user_email"> Email </label>
          </span>
          <input type="text" class="form_input" style="margin-top: 0.3em" autofocus
            placeholder="Enter your Email" id="review_user_email_input" required>
          <div class="form_input-error-message"></div>
        </div>
        <br>
        <div id="rate_your_service">Please rate your service experience for the following parameters:</div>
        <br>
        <div>
          <label style="margin: 10px 0px 0px 0px" id="review_selection_tool_rating_label">1. Selection of the
            testing tool for this assignment.</label><br>
          <span class="star-rating" id="review_page_overall_experience_star_rating">
            <input type="radio" name="rating1" value="1"><i>1</i>
            <input type="radio" name="rating1" value="2"><i>2</i>
            <input type="radio" name="rating1" value="3"><i>3</i>
            <input type="radio" name="rating1" value="4"><i>4</i>
            <input type="radio" name="rating1" value="5"><i>5</i>
          </span>
        </div>
        <div class="clear"></div>
        <div class="survey-line">Review homeaportal line one</div>
```

Figure 9: Code Part 1 for the Feedback Form

```
 248      
```

249 **div** class="row">250 **div** style="width: 100%;">251 **div** style="margin-bottom: 10px;">252 **checkbox** checked="" name="checkbox1"/> 253 **checkbox** name="checkbox2"/> 254 **checkbox** name="checkbox3"/> 255 **checkbox** name="checkbox4"/> 256 **checkbox** name="checkbox5"/> 257 **checkbox** name="checkbox6"/> 258 **checkbox** name="checkbox7"/> 259 **checkbox** name="checkbox8"/> 260 **checkbox** name="checkbox9"/> 261 **checkbox** name="checkbox10"/> 262 **checkbox** name="checkbox11"/> 263 **checkbox** name="checkbox12"/> 264 **checkbox** name="checkbox13"/> 265 **checkbox** name="checkbox14"/> 266 **checkbox** name="checkbox15"/> 267 **checkbox** name="checkbox16"/> 268 **checkbox** name="checkbox17"/> 269 **checkbox** name="checkbox18"/> 270 **checkbox** name="checkbox19"/> 271 **checkbox** name="checkbox20"/> 272 **checkbox** name="checkbox21"/> 273 **checkbox** name="checkbox22"/> 274 **checkbox** name="checkbox23"/> 275 **checkbox** name="checkbox24"/> 276 **checkbox** name="checkbox25"/> 277 **checkbox** name="checkbox26"/> 278 **checkbox** name="checkbox27"/> 279 **checkbox** name="checkbox28"/> 280 **checkbox** name="checkbox29"/> 281 **checkbox** name="checkbox30"/> 282 **checkbox** name="checkbox31"/> 283 **checkbox** name="checkbox32"/> 284

285

286 **div** style="margin-bottom: 10px;">287 **label** style="margin-right: 10px;">288 **radio** name="rating1" value="1"/>**radio** name="rating1" value="2"/>**radio** name="rating1" value="3"/>**radio** name="rating1" value="4"/>**radio** name="rating1" value="5"/>

289 **label** style="margin-right: 10px;">290 **radio** name="rating2" value="1"/>**radio** name="rating2" value="2"/>**radio** name="rating2" value="3"/>**radio** name="rating2" value="4"/>**radio** name="rating2" value="5"/>

291 **label** style="margin-right: 10px;">292 **radio** name="rating3" value="1"/>**radio** name="rating3" value="2"/>**radio** name="rating3" value="3"/>**radio** name="rating3" value="4"/>**radio** name="rating3" value="5"/>

293 **label** style="margin-right: 10px;">294 **radio** name="rating4" value="1"/>**radio** name="rating4" value="2"/>**radio** name="rating4" value="3"/>**radio** name="rating4" value="4"/>**radio** name="rating4" value="5"/>

295 **label** style="margin-right: 10px;">296 **radio** name="rating5" value="1"/>**radio** name="rating5" value="2"/>**radio** name="rating5" value="3"/>**radio** name="rating5" value="4"/>**radio** name="rating5" value="5"/>

297

298 **div** style="clear: both;">

299 **div** class="survey-line" style="border-top: 1px solid black; margin-top: 10px;">

300 **div** style="text-align: center;">301 **label** id="review_implementation_3rd_question_label1">3. How did you like the implementation of the application and the UI submitted as part of this assignment?
input type="radio" name="rating6" value="1"/>**input** type="radio" name="rating6" value="2"/>**input** type="radio" name="rating6" value="3"/>**input** type="radio" name="rating6" value="4"/>**input** type="radio" name="rating6" value="5"/>

302 **label** id="review_implementation_3rd_question_label2">POOR
input type="radio" name="rating7" value="1"/>**input** type="radio" name="rating7" value="2"/>**input** type="radio" name="rating7" value="3"/>**input** type="radio" name="rating7" value="4"/>**input** type="radio" name="rating7" value="5"/>

303 **label** id="review_implementation_3rd_question_label3">BEST
input type="radio" name="rating8" value="1"/>**input** type="radio" name="rating8" value="2"/>**input** type="radio" name="rating8" value="3"/>**input** type="radio" name="rating8" value="4"/>**input** type="radio" name="rating8" value="5"/>

304

305 **div** style="margin-top: 10px;">306 **label** id="review_implementation_3rd_question_label4">4. How did you like the rating scale used?
input type="radio" name="rating9" value="1"/>**input** type="radio" name="rating9" value="2"/>**input** type="radio" name="rating9" value="3"/>**input** type="radio" name="rating9" value="4"/>**input** type="radio" name="rating9" value="5"/>

307 **label** id="review_implementation_3rd_question_label5">5. How did you like the rating scale used?
input type="radio" name="rating10" value="1"/>**input** type="radio" name="rating10" value="2"/>**input** type="radio" name="rating10" value="3"/>**input** type="radio" name="rating10" value="4"/>**input** type="radio" name="rating10" value="5"/>

308

309 **div** style="text-align: right; margin-top: 10px;">310 **button** type="button" style="width: 100px; height: 30px; border-radius: 15px; border: none; background-color: #007bff; color: white; font-weight: bold; font-size: 14px;">SUBMIT

311

312

Figure 10: Code Part 2 for the Feedback Form

```

Go Run Terminal Help
firstpage.html login-signup-form-master - Visual Studio Code
secondpage.html firstpage.py TestFile.py checkbox.css demo.css radiobutton.css main.css reviewpage.css scrollarea.css slider.css
① firstpage.html ② secondpage.html ③ checkbox.css ④ demo.css ⑤ radiobutton.css ⑥ main.css ⑦ reviewpage.css ⑧ scrollarea.css ⑨ slider.css
⑩ firstpage.html ⑪ body ⑫ div#page>div#review_container>div#review_container>form#loginform>div.form_input-group>div.form_input-error-message
⑬ </div>
⑭ <label value="">>
⑮ <input type="radio" name="rating">
⑯ <label style="width:180px;"><label>
⑰ </label>
⑱ <label value="7">
⑲ <input type="radio" name="rating">
⑳ <label style="width:180px;"><label>
⑲ </label>
⑳ <label value="8">
⑴ <input type="radio" name="rating">
⑵ <label style="width:180px;"><label>
⑴ </label>
⑵ <label value="9">
⑶ <input type="radio" name="rating">
⑷ <label style="width:180px;"><label>
⑶ </label>
⑷ <label value="10">
⑸ <input type="radio" name="rating" value="10">
⑹ <label style="width:180px;"><label>
⑸ </label>
⑹ </span>
⑺ </div>
⑻ <div class="clear"></div>
⑼ <div class="survey_hn" id="review_horizontal_line_three"></div>
⑽ </div>
⑾ <div id="custom_review_text_area_box">
⑿ <label id="custom_review_text_area_label" for="box_desc_4" if=" there are any Other suggestions, please let me know what you think!><label>
⑿ <input type="text" id="box_desc_4" style="width:100%; height: 50px; border: 1px solid #ccc; border-radius: 5px; font-size: 14px; padding: 5px; margin-bottom: 10px; " />
⑿ <div id="custom_review_text_area_box_html">
⑿ <pre></pre>
⑿ </div>
⑿ <div class="clear"></div>
⑿ <div class="survey_hn" id="review_horizontal_line_four"></div>
⑿ </div>
⑿ <div id="review_page_overall_experience" 5. Your overall experience with my submitted assignment></div>
⑿ <div class="slide" id="slide_for_overall_rating" for="overall_rating">
⑿ <label id="slide_label_for_overall_rating" for="age"><label>
⑿ <input type="range" value="1" max="10" value="5" />

```

Figure 11: Code Part 3 for the Feedback Form

```

Go Run Terminal Help
firstpage.html login-signup-form-master - Visual Studio Code
secondpage.html firstpage.py TestFile.py checkbox.css demo.css radiobutton.css main.css reviewpage.css scrollarea.css slider.css
① firstpage.html ② secondpage.html ③ checkbox.css ④ demo.css ⑤ radiobutton.css ⑥ main.css ⑦ reviewpage.css ⑧ scrollarea.css ⑨ slider.css
⑩ firstpage.html ⑪ body ⑫ div#page>div#review_container>div#review_container>form#loginform>div.form_input-group>div.form_input-error-message
⑬ <input type="range" min="1" max="10" value="5" />
⑭ <input type="range" id="review_overall_rating_slider" rangeValue="innerText = this.value" />
⑮ <p id="review_overall_rating_slider_rangeValue">5</p>
⑯ </div>
⑰ <div class="button" id="button_submit_review">
⑱ <cp class="form_text" id="label_for_submit_button_on_review_page">
⑲ Note you will be logged out as soon as you submit the review.
⑲ </cp>
⑲ <a href="#" onclick="return show('Page1', 'Page3');">
⑲ <!-- validateForm('Page2', 'Page1', 'usernameEmail_input', 'password_input'); -->
⑲ <input type="submit" value="Submit Review" id="user_submit_review_button" />
⑲ </a>
⑲ </div>
⑲ <div class="button" id="navigation_to_login_page_from_the_review_page">
⑲ <cp class="form_text" id="label_for_login_button_on_review_page">
⑲ Not yet ready to review? No problem, come back later to submit your review.
⑲ </cp>
⑲ <a href="#" onclick="return show('Page1', 'Page3');">
⑲ <input type="submit" value="Logout" id="logout_button_on_review_page" />
⑲ </a>
⑲ </div>
⑲ </div>
⑲ </div>

```

Figure 12: Code Part 4 for the Feedback Form

Version 2 GUI

As mentioned on the assignment description page on Canvas, I changed the size, location, and orientation of some of the GUI elements on all three pages.

Page 1

The Sign Up form is the first page of the application where the user can register their account by providing specific information such as their username, email, phone number, gender, age, and a password.

The changes made to this form are as follows:

- Image :** Initially, the sign up form had the small user avatar image at the top and a title “Create Account” just below the image. But in this updated GUI, there are two changes that have been done:
 - Location of the Image :** The title “Create Account” has been pushed to the top and the image has been pushed to be placed below the title.
 - Size of the image :** The height and width CSS attributes have been updated to make the image larger, as shown in the image below.

Orientation of the Input Text Box Fields : All fields were initially implemented in a downward flow. However, four input fields, namely the username, email, password, and confirm password fields, have been placed side by side in groups of two in the updated GUI.

Width of the form: The width of the form was originally set to 400px, but in the updated GUI, the width of the entire form is set to 600px, which is why the width of the form is clearly visible in the image attached below.

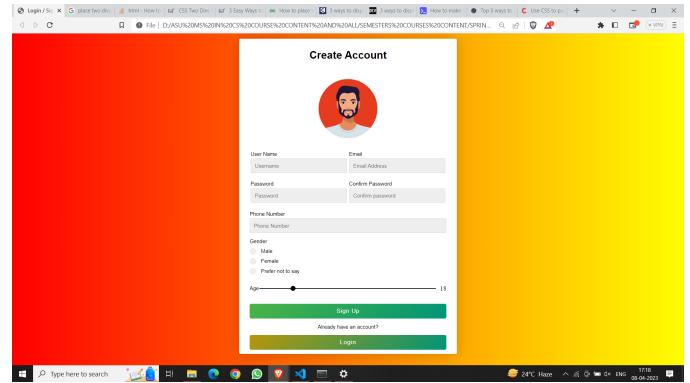


Figure 13: Updated Page 1 : Registration / Sign Up Form

```

108 <div id="Page2">
109 <div class="container" id="sign_up_form_page_container">
110 <form class="form" id="createAccount">
111 <!-- img src="Avatar.png" alt="Avatar" class="avatar" id="signup_homeimage" /-->
112 <h3 id="form_title" id="signup_create_account_header">
113 Create Account
114 </h3>
115 
116 <div class="form_message_form_message_error"></div>
117 <div id="input_section_parent">
118 <div class="form_input_group" id="input_section_parent_child_left">
119 <span style="margin-bottom: 5px;">
120 <label id="signUp_username_email" for="User Name"> User Name </label>
121 </span>
122 <input type="text" id="signUpUsername" class="form_input" style="margin-top: 0.3em" autofocus
123 placeholder="Username" required>
124 <div class="form_input_error_message"></div>
125 </div>
126 <div class="form_input_group" id="input_section_parent_child_right">
127 <span style="margin-bottom: 5px;">
128 <label id="signUp_user_email_label" for="Email"> Email </label>
129 </span>
130 <input type="text" class="form_input" style="margin-top: 0.3em" autofocus
131 placeholder="Email Address" id="signUpEmail_email" required>
132 <div class="form_input_error_message"></div>
133 </div>
134 <div id="input_section_parent">
135 <div class="form_input_group" id="input_section_parent_child_left">
136 <span style="margin-bottom: 5px;">
137 <label id="signUp_user_password_label" for="Password"> Password </label>
138 </span>
139 <input type="password" class="form_input" style="margin-top: 0.3em" autofocus
140 placeholder="Password" id="signUpPassword_password" required>
141 <div class="form_input_error_message"></div>
142 </div>
143 <div class="form_input_group" id="input_section_parent_child_right">
144 <span style="margin-bottom: 5px;">
145 <label id="signUp_user_confirm_password_label" for="Confirm Password"> Confirm Password </label>
146 </span>
147 <input type="password" class="form_input" style="margin-top: 0.3em" autofocus
148 placeholder="Confirm Password" id="signUpConfirmPassword_confirmPassword" required>
149 <div class="form_input_error_message"></div>
150 </div>
151 </div>
152 </div>
153 </div>
154 </div>
155 </div>
156 </div>
157 </div>
158 </div>
159 </div>
160 </div>
161 </div>
162 </div>
163 </div>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 </div>
171 </div>
172 </div>
173 </div>
174 </div>
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>
185 </div>
186 </div>
187 </div>
188 </div>
189 </div>
190 </div>
191 </div>
192 </div>
193 </div>
194 </div>
195 </div>
196 </div>
197 </div>
198 </div>
199 </div>
200 </div>
201 </div>
202 </div>
203 </div>
204 </div>
205 </div>
206 </div>
207 </div>
208 </div>
209 </div>
210 </div>
211 </div>
212 </div>
213 </div>
214 </div>
215 </div>
216 </div>
217 </div>
218 </div>
219 </div>
220 </div>
221 </div>
222 </div>
223 </div>
224 </div>
225 </div>
226 </div>
227 </div>
228 </div>
229 </div>
230 </div>
231 </div>
232 </div>
233 </div>
234 </div>
235 </div>
236 </div>
237 </div>
238 </div>
239 </div>
240 </div>
241 </div>
242 </div>
243 </div>
244 </div>
245 </div>
246 </div>
247 </div>
248 </div>
249 </div>
250 </div>
251 </div>
252 </div>
253 </div>
254 </div>
255 </div>
256 </div>
257 </div>
258 </div>
259 </div>
260 </div>
261 </div>
262 </div>
263 </div>
264 </div>
265 </div>
266 </div>
267 </div>
268 </div>
269 </div>
270 </div>
271 </div>
272 </div>
273 </div>
274 </div>
275 </div>
276 </div>
277 </div>
278 </div>
279 </div>
280 </div>
281 </div>
282 </div>
283 </div>
284 </div>
285 </div>
286 </div>
287 </div>
288 </div>
289 </div>
290 </div>
291 </div>
292 </div>
293 </div>
294 </div>
295 </div>
296 </div>
297 </div>
298 </div>
299 </div>
300 </div>
301 </div>
302 </div>
303 </div>
304 </div>
305 </div>
306 </div>
307 </div>
308 </div>
309 </div>
310 </div>
311 </div>
312 </div>
313 </div>
314 </div>
315 </div>
316 </div>
317 </div>
318 </div>
319 </div>
320 </div>
321 </div>
322 </div>
323 </div>
324 </div>
325 </div>
326 </div>
327 </div>
328 </div>
329 </div>
330 </div>
331 </div>
332 </div>
333 </div>
334 </div>
335 </div>
336 </div>
337 </div>
338 </div>
339 </div>
340 </div>
341 </div>
342 </div>
343 </div>
344 </div>
345 </div>
346 </div>
347 </div>
348 </div>
349 </div>
350 </div>
351 </div>
352 </div>
353 </div>
354 </div>
355 </div>
356 </div>
357 </div>
358 </div>
359 </div>
360 </div>
361 </div>
362 </div>
363 </div>
364 </div>
365 </div>
366 </div>
367 </div>
368 </div>
369 </div>
370 </div>
371 </div>
372 </div>
373 </div>
374 </div>
375 </div>
376 </div>
377 </div>
378 </div>
379 </div>
380 </div>
381 </div>
382 </div>
383 </div>
384 </div>
385 </div>
386 </div>
387 </div>
388 </div>
389 </div>
390 </div>
391 </div>
392 </div>
393 </div>
394 </div>
395 </div>
396 </div>
397 </div>
398 </div>
399 </div>
400 </div>
401 </div>
402 </div>
403 </div>
404 </div>
405 </div>
406 </div>
407 </div>
408 </div>
409 </div>
410 </div>
411 </div>
412 </div>
413 </div>
414 </div>
415 </div>
416 </div>
417 </div>
418 </div>
419 </div>
420 </div>
421 </div>
422 </div>
423 </div>
424 </div>
425 </div>
426 </div>
427 </div>
428 </div>
429 </div>
430 </div>
431 </div>
432 </div>
433 </div>
434 </div>
435 </div>
436 </div>
437 </div>
438 </div>
439 </div>
440 </div>
441 </div>
442 </div>
443 </div>
444 </div>
445 </div>
446 </div>
447 </div>
448 </div>
449 </div>
450 </div>
451 </div>
452 </div>
453 </div>
454 </div>
455 </div>
456 </div>
457 </div>
458 </div>
459 </div>
460 </div>
461 </div>
462 </div>
463 </div>
464 </div>
465 </div>
466 </div>
467 </div>
468 </div>
469 </div>
470 </div>
471 </div>
472 </div>
473 </div>
474 </div>
475 </div>
476 </div>
477 </div>
478 </div>
479 </div>
480 </div>
481 </div>
482 </div>
483 </div>
484 </div>
485 </div>
486 </div>
487 </div>
488 </div>
489 </div>
490 </div>
491 </div>
492 </div>
493 </div>
494 </div>
495 </div>
496 </div>
497 </div>
498 </div>
499 </div>
500 </div>
501 </div>
502 </div>
503 </div>
504 </div>
505 </div>
506 </div>
507 </div>
508 </div>
509 </div>
510 </div>
511 </div>
512 </div>
513 </div>
514 </div>
515 </div>
516 </div>
517 </div>
518 </div>
519 </div>
520 </div>
521 </div>
522 </div>
523 </div>
524 </div>
525 </div>
526 </div>
527 </div>
528 </div>
529 </div>
530 </div>
531 </div>
532 </div>
533 </div>
534 </div>
535 </div>
536 </div>
537 </div>
538 </div>
539 </div>
540 </div>
541 </div>
542 </div>
543 </div>
544 </div>
545 </div>
546 </div>
547 </div>
548 </div>
549 </div>
550 </div>
551 </div>
552 </div>
553 </div>
554 </div>
555 </div>
556 </div>
557 </div>
558 </div>
559 </div>
560 </div>
561 </div>
562 </div>
563 </div>
564 </div>
565 </div>
566 </div>
567 </div>
568 </div>
569 </div>
570 </div>
571 </div>
572 </div>
573 </div>
574 </div>
575 </div>
576 </div>
577 </div>
578 </div>
579 </div>
580 </div>
581 </div>
582 </div>
583 </div>
584 </div>
585 </div>
586 </div>
587 </div>
588 </div>
589 </div>
590 </div>
591 </div>
592 </div>
593 </div>
594 </div>
595 </div>
596 </div>
597 </div>
598 </div>
599 </div>
600 </div>
601 </div>
602 </div>
603 </div>
604 </div>
605 </div>
606 </div>
607 </div>
608 </div>
609 </div>
610 </div>
611 </div>
612 </div>
613 </div>
614 </div>
615 </div>
616 </div>
617 </div>
618 </div>
619 </div>
620 </div>
621 </div>
622 </div>
623 </div>
624 </div>
625 </div>
626 </div>
627 </div>
628 </div>
629 </div>
630 </div>
631 </div>
632 </div>
633 </div>
634 </div>
635 </div>
636 </div>
637 </div>
638 </div>
639 </div>
640 </div>
641 </div>
642 </div>
643 </div>
644 </div>
645 </div>
646 </div>
647 </div>
648 </div>
649 </div>
650 </div>
651 </div>
652 </div>
653 </div>
654 </div>
655 </div>
656 </div>
657 </div>
658 </div>
659 </div>
660 </div>
661 </div>
662 </div>
663 </div>
664 </div>
665 </div>
666 </div>
667 </div>
668 </div>
669 </div>
670 </div>
671 </div>
672 </div>
673 </div>
674 </div>
675 </div>
676 </div>
677 </div>
678 </div>
679 </div>
680 </div>
681 </div>
682 </div>
683 </div>
684 </div>
685 </div>
686 </div>
687 </div>
688 </div>
689 </div>
690 </div>
691 </div>
692 </div>
693 </div>
694 </div>
695 </div>
696 </div>
697 </div>
698 </div>
699 </div>
700 </div>
701 </div>
702 </div>
703 </div>
704 </div>
705 </div>
706 </div>
707 </div>
708 </div>
709 </div>
710 </div>
711 </div>
712 </div>
713 </div>
714 </div>
715 </div>
716 </div>
717 </div>
718 </div>
719 </div>
720 </div>
721 </div>
722 </div>
723 </div>
724 </div>
725 </div>
726 </div>
727 </div>
728 </div>
729 </div>
730 </div>
731 </div>
732 </div>
733 </div>
734 </div>
735 </div>
736 </div>
737 </div>
738 </div>
739 </div>
740 </div>
741 </div>
742 </div>
743 </div>
744 </div>
745 </div>
746 </div>
747 </div>
748 </div>
749 </div>
750 </div>
751 </div>
752 </div>
753 </div>
754 </div>
755 </div>
756 </div>
757 </div>
758 </div>
759 </div>
760 </div>
761 </div>
762 </div>
763 </div>
764 </div>
765 </div>
766 </div>
767 </div>
768 </div>
769 </div>
770 </div>
771 </div>
772 </div>
773 </div>
774 </div>
775 </div>
776 </div>
777 </div>
778 </div>
779 </div>
780 </div>
781 </div>
782 </div>
783 </div>
784 </div>
785 </div>
786 </div>
787 </div>
788 </div>
789 </div>
790 </div>
791 </div>
792 </div>
793 </div>
794 </div>
795 </div>
796 </div>
797 </div>
798 </div>
799 </div>
800 </div>
801 </div>
802 </div>
803 </div>
804 </div>
805 </div>
806 </div>
807 </div>
808 </div>
809 </div>
810 </div>
811 </div>
812 </div>
813 </div>
814 </div>
815 </div>
816 </div>
817 </div>
818 </div>
819 </div>
820 </div>
821 </div>
822 </div>
823 </div>
824 </div>
825 </div>
826 </div>
827 </div>
828 </div>
829 </div>
830 </div>
831 </div>
832 </div>
833 </div>
834 </div>
835 </div>
836 </div>
837 </div>
838 </div>
839 </div>
840 </div>
841 </div>
842 </div>
843 </div>
844 </div>
845 </div>
846 </div>
847 </div>
848 </div>
849 </div>
850 </div>
851 </div>
852 </div>
853 </div>
854 </div>
855 </div>
856 </div>
857 </div>
858 </div>
859 </div>
860 </div>
861 </div>
862 </div>
863 </div>
864 </div>
865 </div>
866 </div>
867 </div>
868 </div>
869 </div>
870 </div>
871 </div>
872 </div>
873 </div>
874 </div>
875 </div>
876 </div>
877 </div>
878 </div>
879 </div>
880 </div>
881 </div>
882 </div>
883 </div>
884 </div>
885 </div>
886 </div>
887 </div>
888 </div>
889 </div>
890 </div>
891 </div>
892 </div>
893 </div>
894 </div>
895 </div>
896 </div>
897 </div>
898 </div>
899 </div>
900 </div>
901 </div>
902 </div>
903 </div>
904 </div>
905 </div>
906 </div>
907 </div>
908 </div>
909 </div>
910 </div>
911 </div>
912 </div>
913 </div>
914 </div>
915 </div>
916 </div>
917 </div>
918 </div>
919 </div>
920 </div>
921 </div>
922 </div>
923 </div>
924 </div>
925 </div>
926 </div>
927 </div>
928 </div>
929 </div>
930 </div>
931 </div>
932 </div>
933 </div>
934 </div>
935 </div>
936 </div>
937 </div>
938 </div>
939 </div>
940 </div>
941 </div>
942 </div>
943 </div>
944 </div>
945 </div>
946 </div>
947 </div>
948 </div>
949 </div>
950 </div>
951 </div>
952 </div>
953 </div>
954 </div>
955 </div>
956 </div>
957 </div>
958 </div>
959 </div>
960 </div>
961 </div>
962 </div>
963 </div>
964 </div>
965 </div>
966 </div>
967 </div>
968 </div>
969 </div>
970 </div>
971 </div>
972 </div>
973 </div>
974 </div>
975 </div>
976 </div>
977 </div>
978 </div>
979 </div>
980 </div>
981 </div>
982 </div>
983 </div>
984 </div>
985 </div>
986 </div>
987 </div>
988 </div>
989 </div>
990 </div>
991 </div>
992 </div>
993 </div>
994 </div>
995 </div>
996 </div>
997 </div>
998 </div>
999 </div>
1000 </div>
1001 </div>
1002 </div>
1003 </div>
1004 </div>
1005 </div>
1006 </div>
1007 </div>
1008 </div>
1009 </div>
1010 </div>
1011 </div>
1012 </div>
1013 </div>
1014 </div>
1015 </div>
1016 </div>
1017 </div>
1018 </div>
1019 </div>
1020 </div>
1021 </div>
1022 </div>
1023 </div>
1024 </div>
1025 </div>
1026 </div>
1027 </div>
1028 </div>
1029 </div>
1030 </div>
1031 </div>
1032 </div>
1033 </div>
1034 </div>
1035 </div>
1036 </div>
1037 </div>
1038 </div>
1039 </div>
1040 </div>
1041 </div>
1042 </div>
1043 </div>
1044 </div>
1045 </div>
1046 </div>
1047 </div&gt
```

```

51     placeholder="Confirm password" id="signup_confirm_password" required/>
52   
```

```

20 <div id="login_form" style="display: none">
21   <div class="form_group">
22     <form id="login" action="#">
23       <!-- 
24       
25       <div class="form_message form_message_error"></div>
26     </div>
27     <div class="form_input_group">
28       <span style="margin-bottom: 5px">
29         <label id="username_email_label" style="margin-bottom: 2px">
30           Phone Number
31         </label>
32       </span>
33       <input type="text" class="form_input" style="margin-top: 0.3rem" autofocus
34         placeholder="Phone Number" id="signup_phonenumber" required/>
35     </div>
36     <div class="form_input_group">
37       <span style="margin-bottom: 5px">
38         <label id="signup_gender_label" style="margin-bottom: 2px">
39           Male
40         </label>
41         <input type="radio" name="radio" id="male_radio_label" required />
42         <span class="radio-checkmark"></span>
43       </span>
44     </div>
45     <div class="form_input_group">
46       <span style="margin-bottom: 5px">
47         <label class="radio-container" id="female_radio_label">
48           Female
49         </label>
50         <input type="radio" name="radio" id="female_radio_label_button" required />
51         <span class="radio-checkmark"></span>
52       </span>
53     </div>
54     <div class="form_input_group">
55       <span style="margin-bottom: 5px">
56         <label id="other_gender_label">
57           Prefer not to say
58         </label>
59         <input type="radio" name="radio" id="other_gender_radio_label_button" required />
60         <span class="radio-checkmark"></span>
61       </span>
62     </div>
63   </div>
64   <div class="form_input_group">
65     <span style="margin-bottom: 5px">
66       <label id="slider_label_for_age" style="margin-bottom: 2px">
67         Age
68       </label>
69       <input type="range" id="age_slider_id" min="0" max="100" value="18" />
70       <span style="margin-left: 10px">18</span>
71     </span>
72   </div>

```

Figure 15: Code Part 2 for the Updated Sign Up Page

```

198   <p id="rangeValue">18</p>
199   </div>
200
201   <div class="button" id="button_signup">
202     <a href="#" onclick="return validateSignUpForm('Page1', 'Page2');">
203       <!-- validateForm('Page2', 'Page1', 'userNameorEmail_input', 'password_input'); -->
204     </a>
205   </div>
206
207   <div class="button" id="navigation_to_login_page">
208     <p class="form_text" id="label_for_login_button_on_sign_up_page">
209       Already have an account?
210     </p>
211     <a href="#" onclick="return show('Page1', 'Page2');">
212       <input type="submit" value="Login" id="login_button_on_sign_up_page" />
213     </a>
214   </div>
215
216 </form>
217 </div>
218 </div>
219

```

Figure 16: Code Part 3 for the Updated Sign Up Page

- Size of the image :** The height and width CSS attributes have been updated to make the image larger, as shown in the image below.
- Spacing between the Labels and Input Text Boxes :** I have added a “margin-top” CSS property to the input field text boxes to add some spacing between them and their respective labels.
- Scroll Area Box :** The height of the Scroll Area Box has been increased in comparison to its initial height.

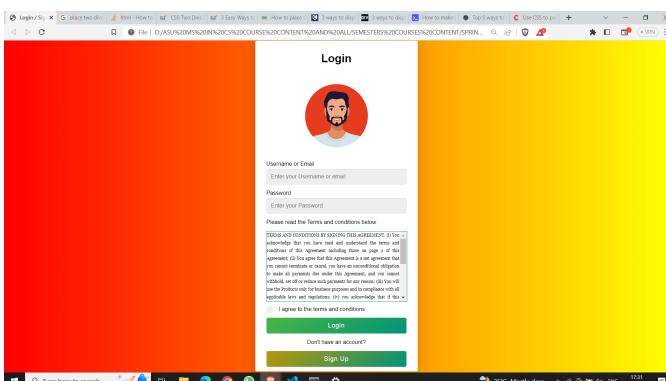


Figure 17: Updated Page 2 : Login Form

```

20 <div id="login_form" style="display: none">
21   <div class="checkboxes" style="display: flex; justify-content: space-around; margin-bottom: 10px">
22     <input type="checkbox" id="checkbox1" checked="" style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 50%;"/>
23     <input type="checkbox" id="checkbox2" checked="" style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 50%;"/>
24     <input type="checkbox" id="checkbox3" checked="" style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 50%;"/>
25   </div>
26   <div class="form_group">
27     <form id="login" action="#">
28       <!-- 
29       
30       <div class="form_message form_message_error"></div>
31     </div>
32     <div class="form_input_group">
33       <span style="margin-bottom: 5px">
34         <label id="username_email_label">
35           Username or Email
36         </label>
37       </span>
38       <input type="text" class="form_input" style="margin-top: 0.3rem" autofocus
39         placeholder="Enter your Username or Email" id="userEmailInput" required/>
40     </div>
41     <div class="form_input_group">
42       <span style="margin-bottom: 5px">
43         <label id="user_password">
44           Password
45         </label>
46       </span>
47       <input type="password" class="form_input" style="margin-top: 0.3rem" autofocus
48         placeholder="Enter your Password" id="passwordInput" required/>
49     </div>
50     <div class="form_input_group">
51       <span style="margin-bottom: 5px">
52         <label id="checkbox_label" style="margin-bottom: 2px">
53           Agree to the terms and conditions
54         </label>
55         <input type="checkbox" name="checkbox" id="checkbox_terms_and_conditions" required />
56         <span class="checkbox-checkmark"></span>
57       </span>
58     </div>
59     <div class="button" id="button_login">
60       <a href="#" onclick="return validateForm('Page3', 'Page1', 'userEmailInput', 'passwordInput');">
61         <input type="submit" value="Login" id="userLoginButton" />
62       </a>
63     </div>
64     <div class="button" id="create">
65       <p class="form_text" id="label_for_sign_up_on_login_page">
66         Don't have an account?
67       </p>
68       <a href="#" onclick="return show('Page2', 'Page1');">
69         <input type="submit" value="Sign Up" id="signUpButtonOnLoginPage" />
70       </a>
71     </div>
72   </div>
73 </div>

```

Figure 18: Code Part 1 for the Updated Login Page

```

64   <div id="login_form" style="display: none">
65     <div class="checkboxes" style="display: flex; justify-content: space-around; margin-bottom: 10px">
66       <input type="checkbox" id="checkbox1" checked="" style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 50%;"/>
67       <input type="checkbox" id="checkbox2" checked="" style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 50%;"/>
68       <input type="checkbox" id="checkbox3" checked="" style="width: 20px; height: 20px; border: 1px solid #ccc; border-radius: 50%;"/>
69     </div>
70     <div class="form_group">
71       <form id="login" action="#">
72         <!-- 
73         
74         <div class="form_message form_message_error"></div>
75       </div>
76       <div class="form_input_group">
77         <span style="margin-bottom: 5px">
78           I agree to the terms and conditions
79         </span>
80         <input type="radio" name="radio" id="login_terms_and_conditions" required />
81         <span class="radio-checkmark"></span>
82       </div>
83       <div class="button" id="button_login">
84         <a href="#" onclick="return validateForm('Page3', 'Page1', 'userEmailInput', 'passwordInput');">
85           <input type="submit" value="Login" id="userLoginButton" />
86         </a>
87       </div>
88       <div class="button" id="create">
89         <p class="form_text" id="label_for_sign_up_on_login_page">
90           Don't have an account?
91         </p>
92         <a href="#" onclick="return show('Page2', 'Page1');">
93           <input type="submit" value="Sign Up" id="signUpButtonOnLoginPage" />
94         </a>
95       </div>
96     </div>
97   </div>
98 </div>

```

Figure 19: Code Part 2 for the Updated Login Page

Page 3

The third page of this application is a review form. I created a review page after being inspired by the comments section of the ASU Canvas Grades Section. In this feedback form, I'm asking the user to rate this web application based on its UI, functionality implementation, and content. This form has undergone the following modifications:

- Third Rating Section:** I have done three changes in this section.
 - Interchanged the labels;** I have interchanged the positions of the POOR and the BEST labels.
 - And since to match the labels, I have also reordered the rating values from 10 - 1 instead of 1 - 10.
- Text Area Box :** I increased the number of rows for the text area box, which increased its height in comparison to its initial height.
- Flow Change:** Since, it was stated in the assignment de-

scription that one flow change was required. So, instead of navigating to the Login page as was originally implemented, when the user clicks on the Logout button at the bottom, the application will now navigate to the Sign Up page.



WE APPRECIATE YOUR REVIEW!

Your feedback will help us to improve in our future assignments.

Username	<input type="text" value="Enter your Username"/>
E-mail	<input type="text" value="Enter your Email"/>
Please rate your service experience for the following parameters 1. Satisfaction with the assignment?	
<input checked="" type="radio"/> Very Satisfied <input type="radio"/> Satisfied <input type="radio"/> Neutral <input type="radio"/> Dissatisfied <input type="radio"/> Very Dissatisfied	
2. Assessment of the submitted report in terms of quality or technology used? (Please mark all applicable) <input checked="" type="checkbox"/> The report is well written and the description of the Test Cases mentioned as part of the assignment.	
3. How satisfied are you with the representation of the association and the UI submitted as part of this assignment?	
<input checked="" type="radio"/> Very Satisfied <input type="radio"/> Satisfied <input type="radio"/> Neutral <input type="radio"/> Dissatisfied <input type="radio"/> Very Dissatisfied	
4. If there are any other suggestions, please let me know in the box below? <div style="border: 1px solid #ccc; height: 40px; width: 100%; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 15px; width: 100%;"></div>	
5. Your overall experience with my submitted assignment? <div style="border: 1px solid #ccc; height: 15px; width: 100%; margin-bottom: 5px;"></div> <div style="border: 1px solid #ccc; height: 15px; width: 100%;"></div>	
<input style="background-color: #0070C0; color: white; padding: 5px; border-radius: 5px; border: none; font-weight: bold; width: 100px;" type="button" value="Submit Review"/>	
Not yet ready to review? No problem, come back later to submit your review.	
Logout	

Figure 20: Updated Page 3 : Feedback / Review Form

Figure 21: Code Part 1 for the Updated Review Form

```
secondpage.html ● TestFileTyp3 # checkbox.css # demo.css # radioButtons.css # main.css # reviewpage.css # scrolla
○ secondpage.html > body > div#page3 > div#review_container.container > form#loginForm > div.form_input_group
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
```

Figure 22: Code Part 2 for the Updated Review Form

```
  # secondpage.html • ④ TestFile.py 3  # checkbox.css  # demo.css  # radiobutton.css  # main.css  # reviewpage.css  # scrollarea.css

④ secondpage.html > body > div#page3 > div#review_container.container > #form#loginForm > #form_input-group
293     <label style="width:100%;">
294         <input type="radio" name="rating" value="5">
295         <label style="width:100%;">
296             <input type="radio" name="rating" value="4">
297             <label style="width:100%;">
298                 <input type="radio" name="rating" value="3">
299                 <label style="width:100%;">
300                     <input type="radio" name="rating" value="2">
301                     <label style="width:100%;">
302                         <input type="radio" name="rating" value="1">
303                         <label style="width:100%;">
304                             <input type="radio" name="rating" value="0">
305                             <label style="width:100%;">
306                         </label>
307                     </label>
308                 </label>
309             </label>
310         </label>
311     </label>
312     <div class="clear"></div>
313     <hr id="review_horizontal_line_three" style="border-top: 1px solid black; margin-top: 10px;">
314 </div>

315     <div id="custom_review_text_area_box">
316         <label id="custom_reviewing_text_area_box_label">4. If there are any Other suggestions, please let me
317             know in the text area below:</label><br><br />
318         <textareas cols="50" rows="8" style="overflow:scroll; width: 100%;"
319             id="custom_reviewing_text_area_box_html">
320             </textareas>
321         <div class="clear"></div>
322         <hr id="review_horizontal_line_four" style="border-top: 1px solid black; margin-top: 10px;">
323     </div>

324     <div id="review_page_overall_experience">5. Your overall experience with my submitted assignment?</div>
325     <div id="review_label_14c511f8-11e6-4600-a9c9-53d02a2a0003" style="margin-top: 10px;">
326         <input type="radio" name="experience" value="5" checked="" style="width: 100%; height: 100%; border: none; background-color: transparent;"/>
327         <label style="width: 100%; height: 100%; font-size: 10px; color: #ccc; border: none; background-color: transparent; position: absolute; left: -10px; top: -10px; font-weight: bold; font-style: italic; transform: rotate(-15deg);">
328             5
329         </label>
330     </div>
331     <div id="review_label_14c511f8-11e6-4600-a9c9-53d02a2a0004" style="margin-top: 10px;">
332         <input type="radio" name="experience" value="4" style="width: 100%; height: 100%; border: none; background-color: transparent;"/>
333         <label style="width: 100%; height: 100%; font-size: 10px; color: #ccc; border: none; background-color: transparent; position: absolute; left: -10px; top: -10px; font-weight: bold; font-style: italic; transform: rotate(-15deg);">
334             4
335         </label>
336     </div>
337     <div id="review_label_14c511f8-11e6-4600-a9c9-53d02a2a0005" style="margin-top: 10px;">
338         <input type="radio" name="experience" value="3" style="width: 100%; height: 100%; border: none; background-color: transparent;"/>
339         <label style="width: 100%; height: 100%; font-size: 10px; color: #ccc; border: none; background-color: transparent; position: absolute; left: -10px; top: -10px; font-weight: bold; font-style: italic; transform: rotate(-15deg);">
340             3
341         </label>
342     </div>
343     <div id="review_label_14c511f8-11e6-4600-a9c9-53d02a2a0006" style="margin-top: 10px;">
344         <input type="radio" name="experience" value="2" style="width: 100%; height: 100%; border: none; background-color: transparent;"/>
345         <label style="width: 100%; height: 100%; font-size: 10px; color: #ccc; border: none; background-color: transparent; position: absolute; left: -10px; top: -10px; font-weight: bold; font-style: italic; transform: rotate(-15deg);">
346             2
347         </label>
348     </div>
349     <div id="review_label_14c511f8-11e6-4600-a9c9-53d02a2a0007" style="margin-top: 10px;">
350         <input type="radio" name="experience" value="1" style="width: 100%; height: 100%; border: none; background-color: transparent;"/>
351         <label style="width: 100%; height: 100%; font-size: 10px; color: #ccc; border: none; background-color: transparent; position: absolute; left: -10px; top: -10px; font-weight: bold; font-style: italic; transform: rotate(-15deg);">
352             1
353         </label>
354     </div>
355     <div id="review_label_14c511f8-11e6-4600-a9c9-53d02a2a0008" style="margin-top: 10px;">
356         <input type="radio" name="experience" value="0" style="width: 100%; height: 100%; border: none; background-color: transparent;"/>
357         <label style="width: 100%; height: 100%; font-size: 10px; color: #ccc; border: none; background-color: transparent; position: absolute; left: -10px; top: -10px; font-weight: bold; font-style: italic; transform: rotate(-15deg);">
358             0
359         </label>
360     </div>
361 
```

Figure 23: Code Part 3 for the Updated Review Form

```

① secondepage.html • ② testfileipy 3 # checkbox.cs # demo.css # radiobutton.css # main.css # reviewpage.css # scrollarea
③ secondepage.html > ④ body > ⑤ div#page3 > ⑥ div#review_container.container > ⑦ form#login_form > ⑧ div.form_input_group
332   <div class="clear"></div>
333   <hr class="survey_he" id="review_horizontal_line_four">
334   </div>
335
336   <div id="review_page_overall_experience">5. Your overall experience with my submitted assignment?</div>
337   <div class="slider"> ⑨ <div id="overall_label_for_review" style="display: inline-block; vertical-align: middle; margin-right: 10px;">
338     <input type="range" id="review_overall_rating_slider" min="1" max="10" value="5" oninput="review_overall_rating_slider.rangeValue.innerText = this.value" />
339     <p id="review_overall_rating_slider_rangeValue">5</p>
340   </div>
341
342   <div class="button" id="button_submit_review">
343     <div class="form_text" id="label_for_submit_review_button_on_review_page">
344       <p>Not yet ready to review? No problem, come back later to submit your review.</p>
345     </div> -->
346     <a href="#" onclick="return show('Page1', 'Page3');">
347       <!-- validateForm('Page2', 'Page1', 'usernameOrEmail_input', 'password_input'); -->
348       <input type="submit" value="Submit Review" id="user_submit_review_button" />
349     </a>
350   </div>
351
352   <div class="button" id="navigation_to_login_page_from_the_review_page">
353     <div class="form_text" id="label_for_login_button_on_review_page">
354       <p>Not yet ready to review? No problem, come back later to submit your review.</p>
355     </div> -->
356     <a href="#" onclick="return show('Page1', 'Page3');">
357       <input type="submit" value="Logout" id="logout_button_on_review_page" />
358     </a>
359   </div>
360
361 </div>
362
363 </form>
364 </div>
365 </div>
366

```

Figure 24: Code Part 4 for the Updated Review Form

Description of the Selected Tool

Selenium is a popular open-source library for automating web browsers, including web application testing, web scraping, and repetitive task automation. The Selenium tool provides a set of libraries and tools to simulate user interactions with web browsers, which can be useful for testing web applications and other tasks such as web scraping and web automation.

Selenium can work with a variety of programming languages, including Python. Selenium is used in Python in conjunction with a WebDriver.

Selenium WebDriver is a tool for automating web application testing by using a programmatic interface to control a web browser. This means that developers and testers can use Selenium WebDriver to create functional tests that automate browser actions. The Selenium WebDriver platform automates interactions with web elements like clicking buttons, filling out forms, and navigating between pages, as well as a set of APIs for interacting with web browsers like Chrome, Firefox, and Safari.[5]

Scope of the Tool

The scope of Selenium WebDriver is to automate functional tests, which are used to verify the behavior of a web application. It is not designed to perform load testing, security testing, or other types of performance testing.

Purpose of the Tool

The primary goal of Selenium with Python is to automate web testing by controlling the browser and making assertions about the web application's behavior. Selenium allows testers to run the same tests repeatedly and ensure consistent results. This facilitates the identification and correction of bugs, as well as ensuring that new changes to the application do not disrupt existing functionality.

Example of the tool's Usage with a Use Case

Let's say you are testing a web application that has a login page [5]. You want to verify that the login process works

correctly. Here's how you could use Selenium to automate this process:

```

① exampleusecase.py > ...
1   from selenium import webdriver
2
3   # Create an instance of the webdriver
4   driver = webdriver.Chrome()
5
6   # Navigate to the login page
7   driver.get("https://example.com/login")
8
9   # Find the username and password fields and enter values
10  username_field = driver.find_element_by_name("username")
11  username_field.send_keys("myusername")
12
13  password_field = driver.find_element_by_name("password")
14  password_field.send_keys("mypassword")
15
16  # Click the login button
17  login_button = driver.find_element_by_xpath("//button[contains(text(), 'Login'))")
18  login_button.click()
19
20  # Wait for the page to load
21  driver.implicitly_wait(10)
22
23  # Verify that the user is logged in
24  welcome_message = driver.find_element_by_xpath("//h1[contains(text(), 'Welcome'))")
25  assert welcome_message.is_displayed()
26
27  # Close the browser
28  driver.quit()
29

```

Figure 25: Code Example of the Selenium WebDriver

In this example, we use Selenium WebDriver to automate the following steps:

- We create a new instance of the Chrome driver.
- Navigate to the login page of an example website
- We then find the email and password fields on the page and enter credentials.
- We submit the login form by clicking on the Login button.
- Verify that the user is logged in by checking the page title.
- Finally if everything goes well, we close the browser window.

By using Selenium WebDriver to automate this test, we can run it repeatedly to ensure that the login functionality is working correctly, without the need for manual testing.

Description of the Test Cases

According to the assignment description, the test cases should be created to test the existence of a GUI element, its correct location, the size of the element, the content of the GUI element, and the link's correctness (flow from page to another).

As a result of these guidelines, I primarily created three types of Test Cases.

- **Category 1:** This category of test cases checks for the **existence, location and the size** of the element using either it's unique ID or CLASS_NAME.
 - The command “**I = browser.find_element(By.ID, “signup_homeimage”)**” checks for the existence of the element.
 - The commands “**location = l.location**”, “**x = location.get(“x”)**” and “**y = location.get(“y”)**” retrieve the x and y coordinates of the element.

- The commands “size = l.size”, and “w, h = size[“width”], size[“height”]” retrieve the width and the height of the element.
- **Category 2:** This category of test cases checks for the **Content of the GUI element** using either it’s unique ID or CLASS_NAME.
- This is possible with the help of the command “s = l.text”. This command is used to get the label content of the GUI element.
- **Category 3:** This category of test cases checks for the **Flow of the GUI** from one link to another using either it’s unique ID or CLASS_NAME.
- This is implemented by using the command “**browser.find_element(By.ID, “user_signup_button”).click()**”.
- The application is then navigated to the next page after clicking a button, so an additional code line is implemented to check for an element that is present in the next page to validate whether or not the application has truly navigated.

```

26 def testcase10(self):
27     try:
28         l = browser.find_element(By.ID, "signup_homenage") code line to find the existence of the element
29         s = l.text
30         print(s) # location code line to find the location of the element
31         l.size = l.size # code line to find the size of the element
32         w, h = l.size["width"], l.size["height"] code line to find the width and the height of the element
33         tw = 100
34         th = 100
35         x0 = 40
36         y0 = 48
37         x = location.get("x") code lines to get the x-coordinate
38         y = location.get("y") and the y-coordinate of the element
39         print("width - ", w)
40         print("height - ", h)
41         print("xco - ", x)
42         print("yco - ", y)
43         self.assertEqual(w, h), [tw, th]
44         self.assertEqual((x, y), (x0, y0))
45         print("Testcase 1 Passed")
46     except NoSuchElementException:
47         print("Element does not exist")
48

```

Figure 26: Code Example of the Test Case Category 1

```

100 def testcase10(self):
101     try:
102         l = browser.find_element(By.ID, "signup_usernameEmail") code line to find the existence of the element in the GUI
103         s = l.text # code line to get the label content of the GUI
104         value = "User Name"
105         self.assertEqual(s, value) code lines to check if the expected
106         print("Testcase 4 Passed") Value is same as the actual value
107
108     except NoSuchElementException:
109         print("Element does not exist")
110

```

Figure 27: Code Example of the Test Case Category 2

```

836 def testcase10(self):
837     try:
838         username = browser.find_element(By.ID, "signupUsername")
839         username.clear()
840         username.send_keys("entering my username here as a value")
841
842         email = browser.find_element(By.ID, "signupEmail")
843         email.clear()
844         email.send_keys("entering my username here as a value")
845
846         password = browser.find_element(By.ID, "signupPassword")
847         password.clear()
848         password.send_keys("entering my password here as a value")
849
850         confirm_password = browser.find_element(By.ID, "signupConfirmPassword")
851         confirm_password.clear()
852         confirm_password.send_keys("entering my password here as a value")
853
854         phone_number = browser.find_element(By.ID, "signupPhoneNumber")
855         phone_number.clear()
856         phone_number.send_keys("8135630372")
857
858         browser.find_element(By.ID, "male_radio_label_button").click()
859         browser.find_element(By.ID, "user_signup_button").click() find the sign up button and click on it
860
861         i = 0
862         while i < 20:
863             i = i + 1
864
865             l = browser.find_element(By.ID, "homenage") check if the image on the login page which is the next
866             s = "round" page is coming up or not to validate the navigation
867             self.assertEqual(s, "round")
868             print("Testcase 41 Passed")
869
870     except:
871         print("Element does not exist")
872

```

Figure 28: Code Example of the Test Case Category 3

Results of the Developed Test Cases

For the whole application, I have developed a total of 109 test cases basing on the initial GUI.

Test Cases Results for the Original GUI

All the 109 test cases have successfully passed after a successful execution of the test cases.

```

Selenium WebDriver (Advanced)
[INFO] D:\Vedio\CS 695 Software Verification Validation and Testing\Assignment\Assignment 5\login-signup-form-master.py
[INFO] TestCasesList.py
[INFO] TestCases listening on ws://127.0.0.1:57332/devtools/browser/2f0bded0-dba7-4c39-aec1-93177930d0d
[INFO] Testcase 1 Passed
[INFO] Testcase 2 Passed
[INFO] Testcase 3 Passed
[INFO] Testcase 4 Passed
[INFO] Testcase 5 Passed
[INFO] Testcase 6 Passed
[INFO] Testcase 7 Passed
[INFO] Testcase 8 Passed
[INFO] Testcase 9 Passed
[INFO] Testcase 10 Passed
[INFO] Testcase 11 Passed
[INFO] Testcase 12 Passed
[INFO] Testcase 13 Passed
[INFO] Testcase 14 Passed
[INFO] Testcase 15 Passed
[INFO] Testcase 16 Passed
[INFO] Testcase 17 Passed
[INFO] Testcase 18 Passed
[INFO] Testcase 19 Passed
[INFO] Testcase 20 Passed
[INFO] Testcase 21 Passed
[INFO] Testcase 22 Passed
[INFO] Testcase 23 Passed
[INFO] Testcase 24 Passed
[INFO] Testcase 25 Passed
[INFO] Testcase 26 Passed
[INFO] Testcase 27 Passed
[INFO] Testcase 28 Passed
[INFO] Testcase 29 Passed
[INFO] Testcase 30 Passed
[INFO] Testcase 31 Passed
[INFO] Testcase 32 Passed
[INFO] Testcase 33 Passed
[INFO] Testcase 34 Passed
[INFO] Testcase 35 Passed
[INFO] Testcase 36 Passed
[INFO] Testcase 37 Passed
[INFO] Testcase 38 Passed
[INFO] Testcase 39 Passed
[INFO] Testcase 40 Passed
[INFO] Testcase 41 Passed
[INFO] Testcase 42 Passed
[INFO] Testcase 43 Passed
[INFO] Testcase 44 Passed
[INFO] Testcase 45 Passed

```

Figure 29: Results of the Test Cases on the Initial GUI - 1

```

Selenium WebDriver (Advanced)
[INFO] D:\Vedio\CS 695 Software Verification Validation and Testing\Assignment\Assignment 5\login-signup-form-master.py
[INFO] TestCasesList.py
[INFO] TestCases listening on ws://127.0.0.1:57332/devtools/browser/2f0bded0-dba7-4c39-aec1-93177930d0d
[INFO] Testcase 1 Passed
[INFO] Testcase 2 Passed
[INFO] Testcase 3 Passed
[INFO] Testcase 4 Passed
[INFO] Testcase 5 Passed
[INFO] Testcase 6 Passed
[INFO] Testcase 7 Passed
[INFO] Testcase 8 Passed
[INFO] Testcase 9 Passed
[INFO] Testcase 10 Passed
[INFO] Testcase 11 Passed
[INFO] Testcase 12 Passed
[INFO] Testcase 13 Passed
[INFO] Testcase 14 Passed
[INFO] Testcase 15 Passed
[INFO] Testcase 16 Passed
[INFO] Testcase 17 Passed
[INFO] Testcase 18 Passed
[INFO] Testcase 19 Passed
[INFO] Testcase 20 Passed
[INFO] Testcase 21 Passed
[INFO] Testcase 22 Passed
[INFO] Testcase 23 Passed
[INFO] Testcase 24 Passed
[INFO] Testcase 25 Passed
[INFO] Testcase 26 Passed
[INFO] Testcase 27 Passed
[INFO] Testcase 28 Passed
[INFO] Testcase 29 Passed
[INFO] Testcase 30 Passed
[INFO] Testcase 31 Passed
[INFO] Testcase 32 Passed
[INFO] Testcase 33 Passed
[INFO] Testcase 34 Passed
[INFO] Testcase 35 Passed
[INFO] Testcase 36 Passed
[INFO] Testcase 37 Passed
[INFO] Testcase 38 Passed
[INFO] Testcase 39 Passed
[INFO] Testcase 40 Passed
[INFO] Testcase 41 Passed
[INFO] Testcase 42 Passed
[INFO] Testcase 43 Passed
[INFO] Testcase 44 Passed
[INFO] Testcase 45 Passed

```

Figure 30: Results of the Test Cases on the Initial GUI - 2

```
Selected Awesomes Project (Awesomes)

Textcase 67 Passed
Textcase 68 Passed
Textcase 69 Passed
Textcase 70 Passed
Textcase 71 Passed
Textcase 72 Passed
Textcase 73 Passed
Textcase 74 Passed
Textcase 75 Passed
Textcase 76 Passed
Textcase 77 Passed
Textcase 78 Passed
Textcase 79 Passed
Textcase 80 Passed
Textcase 81 Passed
Textcase 82 Passed
Textcase 83 Passed
Textcase 84 Passed
Textcase 85 Passed
Textcase 86 Passed
Textcase 87 Passed
Textcase 88 Passed
Textcase 89 Passed
Textcase 90 Passed
Textcase 91 Passed
Textcase 92 Passed
Textcase 93 Passed
Textcase 94 Passed
Textcase 95 Passed
Textcase 96 Passed
Textcase 97 Passed
Textcase 98 Passed
Textcase 99 Passed
Textcase 100 Passed
Textcase 101 Passed
Textcase 102 Passed
Textcase 103 Passed
Textcase 104 Passed
Textcase 105 Passed
Textcase 106 Passed
Textcase 107 Passed
Textcase 108 Passed
Textcase 109 Passed
Textcase 110 Passed

Run 109 tests in 0.125s

D:\USU MS IN CS COURSE CONTENT AND ALL SEMESTERS COURSES CONTENT\SPRING 2023\CS 965 Software Verification Validation and Testing\ASSESSMENTS\ASSESSMENT 5\Login-signup-Form-master>
[base] > Type here to search

[base] D:\USU MS IN CS COURSE CONTENT AND ALL SEMESTERS COURSES CONTENT\SPRING 2023\CS 965 Software Verification Validation and Testing\ASSESSMENTS\ASSESSMENT 5\Login-signup-Form-master>
```

Figure 31: Results of the Test Cases on the Initial GUI - 3

Test Cases Results for the Updated GUI

Out of the developed 109 test cases, 55 test cases failed when executed on the updated GUI.

```
[base] D:\udit M IN COURSE CONTENT AND ALL SEMESTERS COURSES CONTENT\SPRING 2023\CSE 965 Software Verification Validation and Testing\ASSIGNMENTS\ASSIGNMENT 5\login-signup-form-master\py  
thon testfile.py  
[base] D:\udit M IN COURSE CONTENT AND ALL SEMESTERS COURSES CONTENT\SPRING 2023\CSE 965 Software Verification Validation and Testing\ASSIGNMENTS\ASSIGNMENT 5\login-signup-form-master\py  
thon3.8.5 (v3.8.5:43b675d1c4f3, Mar 28 2020, 16:07:35) [MSC v.1916 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license" for more information.  
[base] D:\udit M IN COURSE CONTENT AND ALL SEMESTERS COURSES CONTENT\SPRING 2023\CSE 965 Software Verification Validation and Testing\ASSIGNMENTS\ASSIGNMENT 5\login-signup-form-master\py  
doctests listing on ws://127.0.0.1:83805/doctools/browser?9484a29a-caf7-46f7-a817-2c86c4318870  
....  
ffTestcase 7 Passed  
ffTestcase 8 Passed  
ffTestcase 12 Passed  
ffTestcase 13 Passed  
ffTestcase 14 Passed  
ffTestcase 15 Passed  
ffTestcase 16 Passed  
ffTestcase 17 Passed  
ffTestcase 18 Passed  
ffTestcase 19 Passed  
ffTestcase 20 Passed  
ffTestcase 21 Passed  
ffTestcase 22 Passed  
ffTestcase 23 Passed  
ffTestcase 24 Passed  
ffTestcase 25 Passed  
ffTestcase 26 Passed  
ffTestcase 27 Passed  
ffTestcase 28 Passed  
ffTestcase 29 Passed  
ffTestcase 30 Passed  
ffTestcase 31 Passed  
ffTestcase 32 Passed  
ffTestcase 33 Passed  
ffTestcase 34 Passed  
ffTestcase 35 Passed  
ffTestcase 36 Passed  
ffTestcase 37 Passed  
ffTestcase 38 Passed  
ffTestcase 39 Passed  
ffTestcase 40 Passed  
ffTestcase 41 Passed  
ffTestcase 42 Passed  
ffTestcase 43 Passed  
ffTestcase 44 Passed  
ffTestcase 45 Passed  
ffTestcase 46 Passed  
ffTestcase 47 Passed  
ffTestcase 48 Passed  
ffTestcase 49 Passed  
ffTestcase 50 Passed  
ffTestcase 51 Passed  
ffTestcase 52 Passed  
ffTestcase 53 Passed  
ffTestcase 54 Passed  
ffTestcase 55 Passed  
ffTestcase 56 Passed  
ffTestcase 57 Passed  
ffTestcase 58 Passed  
ffTestcase 59 Passed  
ffTestcase 60 Passed  
ffTestcase 61 Passed  
ffTestcase 62 Passed  
ffTestcase 63 Passed  
ffTestcase 64 Passed  
ffTestcase 65 Passed  
ffTestcase 66 Passed  
ffTestcase 67 Passed  
ffTestcase 68 Passed  
ffTestcase 69 Passed  
ffTestcase 70 Passed  
ffTestcase 71 Passed  
ffTestcase 72 Passed  
ffTestcase 73 Passed  
ffTestcase 74 Passed  
ffTestcase 75 Passed  
ffTestcase 76 Passed  
ffTestcase 77 Passed  
ffTestcase 78 Passed  
ffTestcase 79 Passed  
ffTestcase 80 Passed  
ffTestcase 81 Passed  
ffTestcase 82 Passed  
ffTestcase 83 Passed  
ffTestcase 84 Passed  
ffTestcase 85 Passed  
ffTestcase 86 Passed  
ffTestcase 87 Passed  
ffTestcase 88 Passed
```

Figure 32: Results of the Test Cases on the Updated GUI - 1

Figure 33: Results of the Test Cases on the Updated GUI - 2

Understanding of the Results of the Test Cases for the Updated GUI

- **Test Cases with respect to the Sign Up Page:** For the Sign Up page, 41 test cases have been created. Because of

```
Python Prompt (Annotated)
>>> FULL: testcases#01 (..., main_, TestStringMethods)

Traceback (most recent call last):
File "testfile.py", line 99, in testcase#01
    assertEqual((213, 987), (316, 237))
AssertionError: tuples differ: (213, 987) != (316, 237)

First differing element: 0
#316
#313
+ (213, 987)
| 131, 237)
| 131, 237)
| 131, 237)

Full: testcases#01 (..., main_, TestStringMethods)

Traceback (most recent call last):
File "testfile.py", line 99, in testcase#01
    assertEqual((w, h), (l, m))
AssertionError: tuples differ: (299, 44) != (100, 44)

First differing element: 0
#299
#100
+ (299, 44)
| 100,
| 100,
| 100, 44)
?  **

Full: testcases#05 (..., main_, TestStringMethods)

Traceback (most recent call last):
File "testfile.py", line 103, in testcase#05
    assertEqual(x, y, "(x,y) in testcase#05")
AssertionError: tuples differ: (316, 267) != (316, 321)

First differing element: 0
#316
#316
+ (316, 267)
+ (316, 321)
```

Figure 34: Results of the Test Cases on the Updated GUI - 3

```
Microsoft Windows [Version 10.0.19041]
Copyright (C) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Acer\PycharmProjects\Avacode\src> python -m unittest testStringMethods.py

FALSY: testcase01() _main_.testStringMethods
Traceback (most recent call last):
File "C:\Users\Acer\PycharmProjects\Avacode\src\testStringMethods.py", line 10, in testcase01
    self.assertEqual(w, h), (lw, lh))
AssertionError: tuples differ: (256, 48) != (400, 48)
First differing element 0:
IOB
IOB

+ (256, 48)
| +
| (256, 48)
| ++
| (400, 48)

FALSY: testcase02() _main_.testStringMethods
Traceback (most recent call last):
File "C:\Users\Acer\PycharmProjects\Avacode\src\testStringMethods.py", line 21, in testcase02
    File "Textfile_py", line 212, in testcase02
        self.assertEqual(w, h), (lw, lh))
AssertionError: tuples differ: (210, 48) != (210, 48)
First differing element 0:
IOB
IOB

+ (210, 48)
| +
| (210, 48)

FALSY: testcase03() _main_.testStringMethods
Traceback (most recent call last):
File "C:\Users\Acer\PycharmProjects\Avacode\src\testStringMethods.py", line 32, in testcase03
    File "Textfile_py", line 216, in testcase03
        self.assertEqual(w, h), (lw, lh))
AssertionError: tuples differ: (204, 48) != (400, 48)
First differing element 0:
IOB
IOB

+ (204, 48)
| +
| (204, 48)
```

Figure 35: Results of the Test Cases on the Updated GUI - 4

```
 1 // AcmeCorp Prompt (AcmeCorp)
 2
 3 //-----[Output]-----[Input]-----[Method]
 4 public TestResult TestTextFile()
 5 {
 6     File file = new File("Textfile.txt");
 7     List<String> lines = file.readLines();
 8     assertEquals("Line 1", lines.get(0));
 9     assertEquals("Line 2", lines.get(1));
10     assertEquals("Line 3", lines.get(2));
11
12     insertionError = lines.get(3);
13     assertEquals("Line 4", lines.get(4));
14
15     firstDifferingElement = lines.get(5);
16
17     return TestResult.OK;
18 }
19
20 //-----[Output]-----[Input]-----[Method]
21 public TestResult TestTextFileWithAssert()
22 {
23     File file = new File("Textfile.txt");
24     List<String> lines = file.readLines();
25
26     assertEquals("Line 1", lines.get(0));
27     assertEquals("Line 2", lines.get(1));
28     assertEquals("Line 3", lines.get(2));
29
30     insertionError = lines.get(3);
31     assertEquals("Line 4", lines.get(4));
32
33     firstDifferingElement = lines.get(5);
34
35     self.assertEqual("Line 5", lines.get(6));
36
37     return TestResult.OK;
38 }
39
40 //-----[Output]-----[Input]-----[Method]
41 public TestResult TestTextFileWithAssert2()
42 {
43     File file = new File("Textfile.txt");
44     List<String> lines = file.readLines();
45
46     assertEquals("Line 1", lines.get(0));
47     assertEquals("Line 2", lines.get(1));
48
49     insertionError = lines.get(2);
50     assertEquals("Line 3", lines.get(3));
51
52     firstDifferingElement = lines.get(4);
53
54     self.assertEqual("Line 5", lines.get(5));
55
56     return TestResult.OK;
57 }
58
59 //-----[Output]-----[Input]-----[Method]
60 public TestResult TestTextFileWithAssert3()
61 {
62     File file = new File("Textfile.txt");
63     List<String> lines = file.readLines();
64
65     assertEquals("Line 1", lines.get(0));
66     assertEquals("Line 2", lines.get(1));
67
68     insertionError = lines.get(2);
69     assertEquals("Line 3", lines.get(3));
70
71     firstDifferingElement = lines.get(4);
72
73     self.assertEqual("Line 5", lines.get(5));
74
75     self.assertEqual("Line 6", lines.get(6));
76
77     return TestResult.OK;
78 }
79
80 //-----[Output]-----[Input]-----[Method]
81 public TestResult TestTextFileWithAssert4()
82 {
83     File file = new File("Textfile.txt");
84     List<String> lines = file.readLines();
85
86     assertEquals("Line 1", lines.get(0));
87     assertEquals("Line 2", lines.get(1));
88
89     insertionError = lines.get(2);
90     assertEquals("Line 3", lines.get(3));
91
92     firstDifferingElement = lines.get(4);
93
94     self.assertEqual("Line 5", lines.get(5));
95
96     self.assertEqual("Line 6", lines.get(6));
97
98     self.assertEqual("Line 7", lines.get(7));
99
100    return TestResult.OK;
101 }
```

Figure 36: Results of the Test Cases on the Updated GUI - 5

the changes made to this page, such as changing the position of the image and title, as well as changing the format of the form, all of the test cases pertaining to location and size, totaling 28 cases, have failed. Twelve of the thirteen test cases passed because they were checking for the content of the GUI elements, and since I had not changed any of the elements' content, all of those test cases passed. Finally, the last test case that passed was about the GUI flow, and it passed because the flow from the Sign Up page to the Login page was not hampered or modified in this case.

```

[1] 100% [Anconda]
File: testfile16.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

[1] 100% [Anconda]
File: testfile17.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (210, 623) != (310, 693)
First differing element: 0
210
+
310
(210, 623)
+
(310, 693)

[1] 100% [Anconda]
File: testfile18.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

[1] 100% [Anconda]
File: testfile19.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (210, 623) != (310, 693)
First differing element: 0
210
+
310
(210, 623)
+
(310, 693)

```

Figure 37: Results of the Test Cases on the Updated GUI - 6

```

[1] 100% [Anconda]
File: testfile20.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (230, 626) != (230, 627)
First differing element: 0
230
+
230
(230, 626)
+
(230, 627)

[1] 100% [Anconda]
File: testfile21.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

[1] 100% [Anconda]
File: testfile22.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

[1] 100% [Anconda]
File: testfile23.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

[1] 100% [Anconda]
File: testfile24.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

[1] 100% [Anconda]
File: testfile25.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (400, 18) != (400, 19)
First differing element: 0
400
+
400
(400, 18)
+
(400, 19)

```

Figure 38: Results of the Test Cases on the Updated GUI - 7

- Test Cases with respect to the Login Page:** For the Login page, 20 test cases have been created. Because of the changes made to this page, such as changing the position of the image and title, as well as adding some space between the labels and fields, all of the test cases pertaining to location and size, totaling 13 cases, have failed. 6 of the seven test cases passed because they were checking for the content of the GUI elements, and since I had not changed any of the elements' content, all of those test cases passed. Finally, the last test case that passed was about the GUI flow, and it passed because the flow from the Login page to the Feedback form page was not hampered or modified in this case.

- Test Cases with respect to the Feedback Form Page:** The Feedback Form page has 49 test cases. However, because I did not change the position of the image at the top of this page, all subsequent location and size test cases did not fail and instead passed. The failed test cases are associated with the “POOR” and “BEST” labels, which were relocated. Following these test cases, the text area box test case fails because its height was increased. As a result, some of the following 11 test cases concerning location and size failed. Finally, since I have made a change in the flow where when the user submits the feedback form, the application is rerouted to the sign up page instead of the login page as per the initial implementation.

```

[1] 100% [Anconda]
File: testfile25.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (210, 777) != (310, 797)
First differing element: 0
210
+
310
(210, 777)
+
(310, 797)

[1] 100% [Anconda]
File: testfile26.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

[1] 100% [Anconda]
File: testfile27.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

[1] 100% [Anconda]
File: testfile28.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

[1] 100% [Anconda]
File: testfile29.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

```

Figure 39: Results of the Test Cases on the Updated GUI - 8

```

[1] 100% [Anconda]
File: testfile30.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (210, 766) != (310, 777)
First differing element: 0
210
+
310
(210, 766)
+
(310, 777)

[1] 100% [Anconda]
File: testfile31.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

[1] 100% [Anconda]
File: testfile32.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

[1] 100% [Anconda]
File: testfile33.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

[1] 100% [Anconda]
File: testfile34.py, line 61, in testStringMethods
    File "testfile.py", line 21, in __init__
        self.assertEqual((x, y), (x0, y0))
AssertionError: tuples differ: (310, 2) != (310, 1)
First differing element: 0
310
+
310
(310, 2)
+
(310, 1)

```

Figure 40: Results of the Test Cases on the Updated GUI - 9

As a result, we have 55 failing test cases, with the majority of them related to location and size, with the exception of one that failed due to a change in the application’s flow. Only two of the 54 passed test cases check for the flow of the GUI application, while the majority of the 54 passed test cases check for the content of the GUI elements.

```

[...] Anconda Prompt (Anconda)
[...]
[...] testcases14 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 172, in testcases14
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (600, 45) != (400, 45)
First differing element:
[0]:
  (600, 45)
  (400, 45)
[1]:
  [...]
[...] testcases25 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 183, in testcases25
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (600, 45) != (400, 45)
First differing element:
[0]:
  (600, 45)
  (400, 45)
[1]:
  [...]
[...] testcases36 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 194, in testcases36
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (600, 45) != (400, 45)
First differing element:
[0]:
  (600, 45)
  (400, 45)
[1]:
  [...]

```

Figure 41: Results of the Test Cases on the Updated GUI -10

```

[...] Anconda Prompt (Anconda)
[...]
[...] testcases15 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1020, in testcases15
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 250) != (310, 296)
First differing element:
[0]:
  (310, 250)
  (310, 296)
[1]:
  [...]
[...] testcases17 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1020, in testcases17
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 250) != (310, 294)
First differing element:
[0]:
  (310, 250)
  (310, 294)
[1]:
  [...]
[...] testcases19 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1080, in testcases19
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 250) != (310, 294)
First differing element:
[0]:
  (310, 250)
  (310, 294)
[1]:
  [...]

```

Figure 44: Results of the Test Cases on the Updated GUI -13

```

[...] Anconda Prompt (Anconda)
[...]
[...] testcases36 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 194, in testcases36
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (600, 45) != (400, 45)
First differing element:
[0]:
  (600, 45)
  (400, 45)
[1]:
  [...]
[...] testcases37 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 194, in testcases37
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (600, 18) != (400, 18)
First differing element:
[0]:
  (600, 18)
  (400, 18)
[1]:
  [...]
[...] testcases39 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 82, in testcases39
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (600, 45) != (400, 45)
First differing element:
[0]:
  (600, 45)
  (400, 45)
[1]:
  [...]

```

Figure 42: Results of the Test Cases on the Updated GUI -11

```

[...] Anconda Prompt (Anconda)
[...]
[...] testcases50 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1080, in testcases50
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (400, 202) != (400, 122)
First differing element:
[0]:
  (400, 202)
  (400, 122)
[1]:
  [...]
[...] testcases53 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1100, in testcases53
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 771) != (310, 562)
First differing element:
[0]:
  (310, 771)
  (310, 562)
[1]:
  [...]
[...] testcases57 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1120, in testcases57
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (596, 774) != (596, 595)
First differing element:
[0]:
  (596, 774)
  (596, 595)
[1]:
  [...]

```

Figure 45: Results of the Test Cases on the Updated GUI -14

```

[...] Anconda Prompt (Anconda)
[...]
[...] testcases11 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 111, in testcases11
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (220, 220) != (100, 100)
First differing element:
[0]:
  (220, 220)
  (100, 100)
[1]:
  [...]
[...] testcases12 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 115, in testcases12
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 45) != (310, 165)
First differing element:
[0]:
  (310, 45)
  (310, 165)
[1]:
  [...]
[...] testcases13 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 115, in testcases13
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 35) != (310, 237)
First differing element:
[0]:
  (310, 35)
  (310, 237)
[1]:
  [...]

```

Figure 43: Results of the Test Cases on the Updated GUI -12

```

[...] Anconda Prompt (Anconda)
[...]
[...] testcases55 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1163, in testcases55
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 805) != (310, 596)
First differing element:
[0]:
  (310, 805)
  (310, 596)
[1]:
  [...]
[...] testcases56 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1163, in testcases56
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 805) != (310, 596)
First differing element:
[0]:
  (310, 805)
  (310, 596)
[1]:
  [...]
[...] testcases57 (_main__ TestStringMethods)
Traceback (most recent call last):
File "Testfile.py", line 1211, in testcases57
    self.assertEqual((x,y), (x0,y0))
assertionError: tuples differ: (310, 866) != (310, 697)
First differing element:
[0]:
  (310, 866)
  (310, 697)
[1]:
  [...]

```

Figure 46: Results of the Test Cases on the Updated GUI -15

Figure 47: Results of the Test Cases on the Updated GUI -16

Figure 50: Results of the Test Cases on the Updated GUI -19

Figure 48: Results of the Test Cases on the Updated GUI -17

Figure 51: Results of the Test Cases on the Updated GUI -20

Figure 49: Results of the Test Cases on the Updated GUI -18

Figure 52: Results of the Test Cases on the Updated GUI-21

Assessment of the tool in my own words

Selenium WebDriver is a popular open-source testing framework for automating web applications. Selenium WebDriver is relatively simple to use for testers with programming experience because it provides a simple and intuitive API for web browser automation.

In terms of features and functionalities, Selenium WebDriver provides a comprehensive set of tools for efficient and effective web application testing across multiple platforms and browsers. Here is an evaluation of Selenium WebDriver based on the aforementioned criteria:

- **Set of features and functionalities provided:** Selenium WebDriver provides a rich set of features and functionalities that allow for web browser automation. [6]
 - It supports various programming languages such as Java, Python, C#, Ruby, etc., and allows testers to perform actions like clicking, typing, selecting, etc. on web elements.
 - It also provides capabilities for handling alerts, pop-ups, frames, windows, etc.
 - It supports different types of locators such as ID, name, class name, tag name, link text, partial link text, and CSS selectors.
 - It has the ability to handle different types of web GUI elements like buttons, links, text boxes, dropdowns, and checkboxes, among others. However, it may not support some advanced GUI elements like canvas elements, audio and video elements, and 3D graphics.
 - Additionally, it has the ability to automate different browsers (Chrome, Firefox, Safari, etc.), and support for parallel test execution by running multiple tests in parallel, which can significantly reduce the overall test execution time.
 - **Type of coverage:** Selenium WebDriver provides extensive coverage for web application testing, as it allows you to interact with various GUI elements and test the functionality of web pages [5].
 - This is because Selenium WebDriver allows for comprehensive testing of web applications including functional testing, regression testing, and performance testing.
 - It supports testing on different operating systems, multiple browsers including Chrome, Firefox, Edge, and Safari), and mobile devices. This allows testers to ensure that their web applications are compatible across different browsers and operating systems.
 - **Reuse of test cases:** By providing a programming interface that allows testers to create reusable functions and libraries that can be used across multiple test cases that can be executed across multiple browsers and platforms, Selenium WebDriver encourages and supports the development of reusable test cases. Testers can create test cases once and run them in multiple environments without having to modify them. Because testers can reuse the same test cases for multiple scenarios, they can save time and effort in developing new test scripts for each new application release. [5]
- **Test results produced:** Selenium WebDriver generates detailed test results that aid in the identification of application problems and flaws. The test results can be exported in a variety of formats, including HTML, XML, and JSON, making integration with various reporting tools simple. These reports include detailed information about test execution, such as each test case's pass/fail status, logs, screenshots, and so on.[3]
 - **Ease of usage:** Selenium WebDriver is simple to use and has an API that allows both novice and experienced users to write and maintain test scripts. It also has good documentation with a large set of examples and a large active community of users who help others with any issues, making it simple for beginners to get started with automation testing.[5]
 - **Type of GUI elements that can be tested:** Selenium WebDriver can be used to test a wide range of GUI elements, including text boxes, buttons, forms, links, dropdowns, checkboxes, radio buttons, and more.[4]
 - It can also interact with iframes, pop-ups, and alerts.
 - It can also handle complex GUI components like calendars, sliders, grids, drag and drop, file upload, and captcha.
 - However, it may not be suitable for testing non-HTML GUI components such as desktop applications or mobile apps.

To summarize, Selenium WebDriver is a robust automation tool with numerous features and functionalities for automating web application testing. It is easy to use and offers numerous opportunities for reusing test cases. It is a popular choice among testers due to its support for multiple browsers, reusability of test cases, and detailed test reports. Its simplicity of use and support for testing various GUI elements make it a valuable tool for developers and quality assurance engineers. Furthermore, its ability to generate detailed test results and support for various types of testing make it an excellent choice for testing web applications. It may not, however, be suitable for testing non-HTML GUI components.

References

- Prof. James Collofello, CSE 565 Lecture Videos, 2023
- Prof. Ayca Tuzmen, CSE 565 Lectures, 2023
- Selenium with Python
- The Selenium Browser Automation Project
- Selenium WebDriver Tutorial - A Comprehensive Guide to WebDriver Automation
- Selenium WebDriver
- Webdriver
- Sample Code for the assignment