

# Graphical User Interface Testing

## Description of Version 1 GUI

I have developed a Library management website with three pages- login, personal page and explore more books page. This application can be run by just clicking on any of the html files. There are two folders 'library\_1' and 'library\_2' containing version 1 and 2 respectively. Each folder has the same test.py file. To run the test cases, open the respective folder in an IDE and use Safari browser, else line 12 of the code- `browser = webdriver.Safari()` must be modified according to the browser being used.

### Login Page

GUI used in this page are images, button, input box, radio button and checkbox. When one types the correct username (Junaita) and password (library123) the page flows to the next page, which is the personal information page.

# Login



Username:

Password:

Preferred Language:

- English
- French
- Spanish
- German

Keep me signed in

**Login**

## Personal Page

GUI used in this page are image, button, input box, radio button and list. Clicking on the preferred reading format radio buttons will display your preference. Using the enter genre inputbox, you can add to the list of favorite genres. Clicking on the explore books button present in the nav bar will land on the explore page.

Personal Profile

Explore Books



John Doe

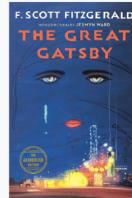
johndoe@example.com

+1 123-456-7890

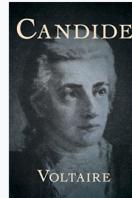
123 Main St, Anytown USA

Member ID: 123456

#### Books Read



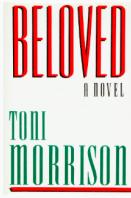
F. SCOTT FITZGERALD  
THE GREAT GATSBY



CANDIDE  
VOLTAIRE



STEPPING STONES  
Gulliver's Travels



BELIEVED  
A NOVEL  
TONI MORRISON

#### Favorite Genres

Mystery

Thriller

Romance

Add Favorite Genre

Enter genre

Add

#### Preferred Reading Format:

Print

Ebook

Audiobook

After giving your preferences-

Personal Profile

Explore Books



John Doe

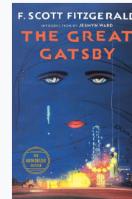
johndoe@example.com

+1 123-456-7890

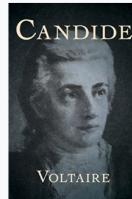
123 Main St, Anytown USA

Member ID: 123456

#### Books Read



F. SCOTT FITZGERALD  
THE GREAT GATSBY



CANDIDE  
VOLTAIRE



STEPPING STONES  
Gulliver's Travels



BELIEVED  
A NOVEL  
TONI MORRISON

#### Favorite Genres

Mystery

Thriller

Romance

Classic

Add Favorite Genre

Enter genre

Add

#### Preferred Reading Format:

Print

Ebook

Audiobook

Preferred Format: Ebook

## Explore Page

GUI used in this page are images, button, input box, checkbox and scroll. This page has cards of books. Clicking on the summary will expand the card. For the first book scroll GUI is implemented in the summary. On the nav bar there is a search feature which filters the books based on genre (fiction, classic and comedy). Clicking on the profile button on the nav bar will take you to the personal information page.

Book Library   Profile  

Display books based on genre preference

### Explore Books

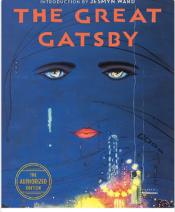
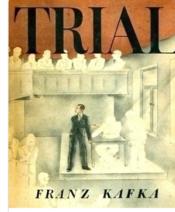
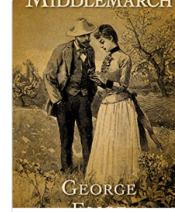
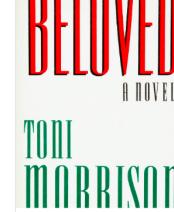
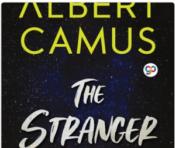
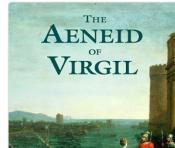
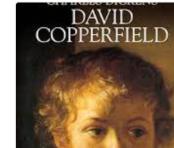
 The Great Gatsby Fiction <a href="#">Summary</a>	 H. G. WELLS The Invisible Man Fiction <a href="#">Summary</a>	 TRIAL FRANZ KAFKA Fiction <a href="#">Summary</a>	 MIDDLEMARCH GEORGE E. Fiction <a href="#">Summary</a>	 BELIEVED A NOVEL TONI MORRISON Fiction <a href="#">Summary</a>	 Gulliver's Travels Fiction <a href="#">Summary</a>
 ALBERT CAMUS The Stranger Fiction <a href="#">Summary</a>	 JANE EYRE CHARLOTTE BRONTE Fiction <a href="#">Summary</a>	 THE AENEID OF VIRGIL Fiction <a href="#">Summary</a>	 COLLECTED FICTION by Jorge Luis Borges Fiction <a href="#">Summary</a>	 DAVID COPPERFIELD Fiction <a href="#">Summary</a>	 LES MISÉRABLES by Victor Hugo Fiction <a href="#">Summary</a>

Screenshot after summary is clicked. It is seen that the scroll GUI element is present.

Book Library Profile Search by genre Search

### Explore Books

Display books based on genre preference

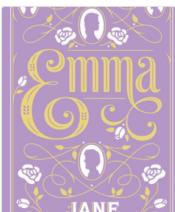
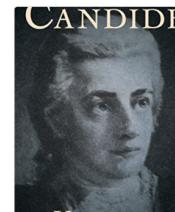
 The Great Gatsby Fiction <a href="#">Summary</a> portrays the decadence and excess of the Roaring Twenties	 H.G. WELLS The Invisible Man Fiction <a href="#">Summary</a>	 TRIAL FRANZ KAFKA Fiction <a href="#">Summary</a>	 MIDDLEMARCH GEORGE ELIOT Fiction <a href="#">Summary</a>	 BELIEVED A NOVEL TONI MORRISON Fiction <a href="#">Summary</a>	 Gulliver's Travels Fiction <a href="#">Summary</a>
 ALBERT CAMUS The Stranger <a href="#">Summary</a>	 JANE EYRE CHARLOTTE BRONTE <a href="#">Summary</a>	 THE AENEID OF VIRGIL <a href="#">Summary</a>	 COLLECTED FICTION by Jorge Luis Borges <a href="#">Summary</a>	 DAVID COPPERFIELD <a href="#">Summary</a>	 Les Misérables <a href="#">Summary</a>

Screenshot after filtering based on genre

Book Library Profile comedy Search

### Explore Books

Display books based on genre preference

 Emma Comedy <a href="#">Summary</a>	 CANDIDE VOLTAIRE Comedy <a href="#">Summary</a>
--	---

## **Description of changes in GUI- Version 2**

### **Change 1**

The password in the login page is placed after preferred languages radio buttons.

### **Change 2**

In the login page, the login button and keep me signed checkbox were initially in the same horizontal orientation. In Version 2 the keep me signed in checkbox was placed below the login button

### **Change 3**

The library image placed at the top is now placed at the bottom in the login page.

### **Change in flow**

When the login button is clicked, the flow is changed from personal page to explore page.

Screenshot of the login page after the changes in Version 2

# Login

Username:

Preferred Language:

- English
- French
- Spanish
- German

Password:

Keep me signed in

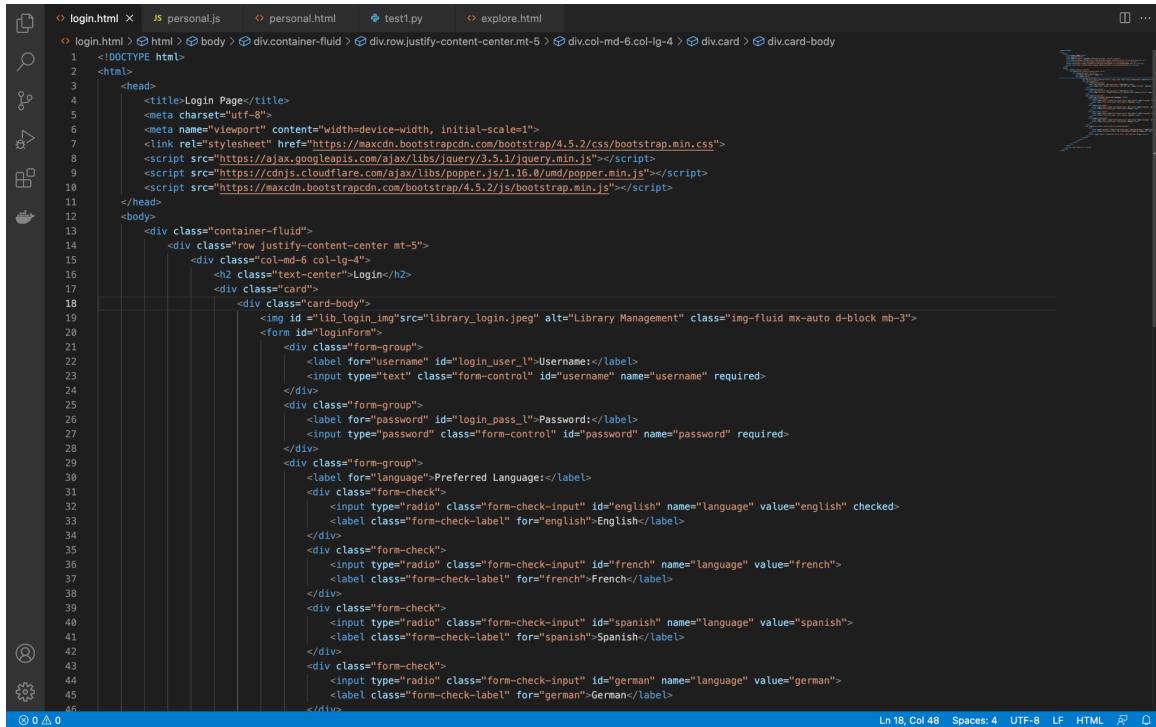


## Screenshots of Source Code

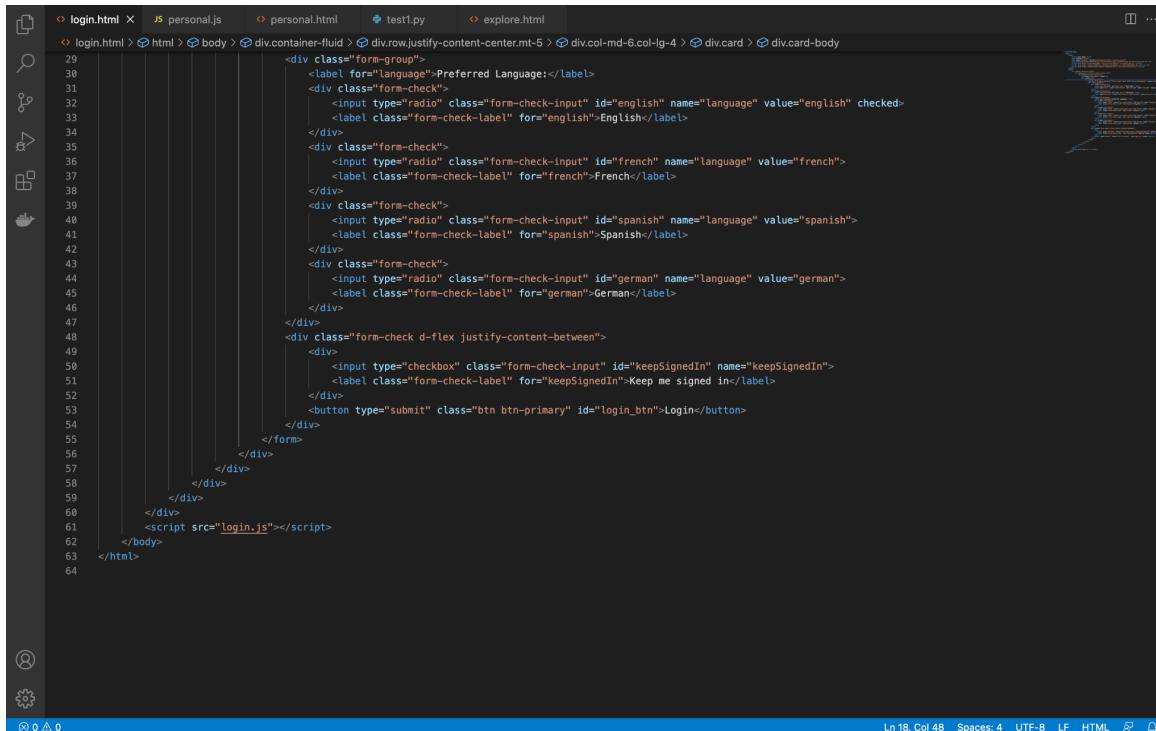
### Version 1

**Login page-** The screenshot present below are those of the html file and javascript file of the login page in Version 1

login.html

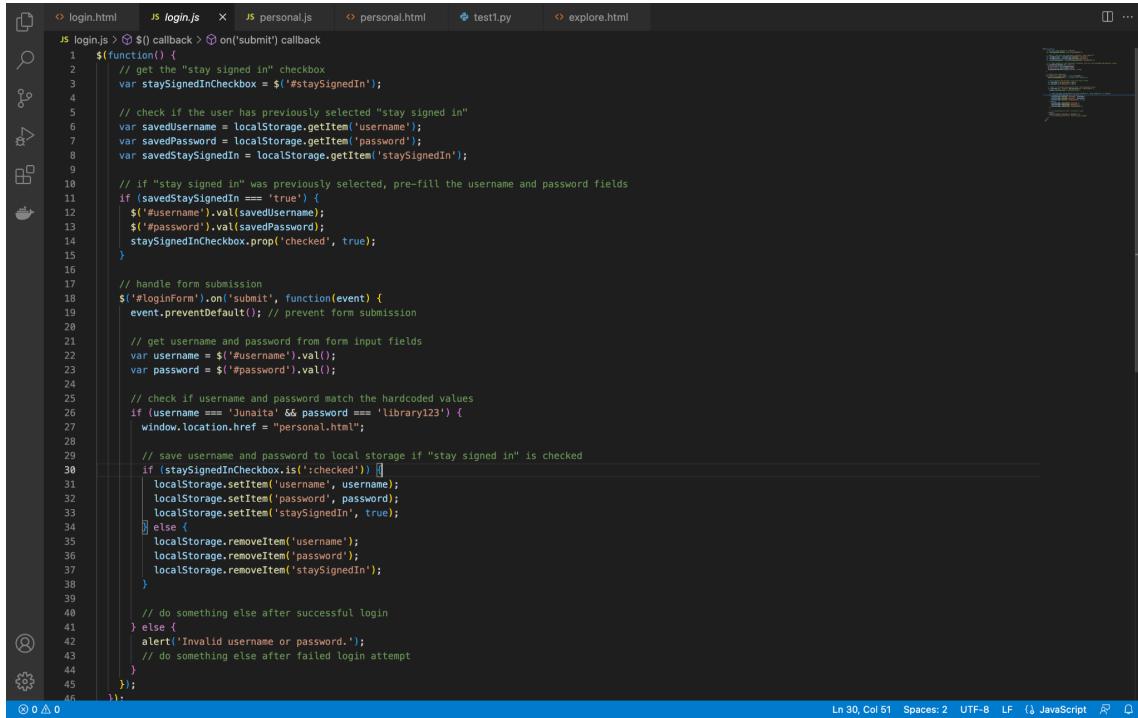


```
login.html x JS personal.js personal.html test1.py explore.html
login.html > html > body > div.container-fluid > div.row.justify-content-center.mt-5 > div.col-md-6.col-lg-4 > div.card > div.card-body
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Login Page</title>
5      <meta charset="utf-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1">
7      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
8      <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9      <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
10     <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
11   </head>
12   <body>
13     <div class="container-fluid">
14       <div class="row justify-content-center mt-5">
15         <div class="col-md-6 col-lg-4">
16           <h2 class="text-center">Login</h2>
17           <div class="card">
18             <div class="card-body">
19               
20               <form id="loginForm">
21                 <div class="form-group">
22                   <label for="username" id="login_user_l">Username:</label>
23                   <input type="text" class="form-control" id="username" name="username" required>
24                 </div>
25                 <div class="form-group">
26                   <label for="password" id="login_pass_l">Password:</label>
27                   <input type="password" class="form-control" id="password" name="password" required>
28                 </div>
29                 <div class="form-group">
30                   <label for="language">Preferred Language:</label>
31                   <div class="form-check">
32                     <input type="radio" class="form-check-input" id="english" name="language" value="english" checked>
33                     <label class="form-check-label" for="english">English</label>
34                   </div>
35                   <div class="form-check">
36                     <input type="radio" class="form-check-input" id="french" name="language" value="french">
37                     <label class="form-check-label" for="french">French</label>
38                   </div>
39                   <div class="form-check">
40                     <input type="radio" class="form-check-input" id="spanish" name="language" value="spanish">
41                     <label class="form-check-label" for="spanish">Spanish</label>
42                   </div>
43                   <div class="form-check">
44                     <input type="radio" class="form-check-input" id="german" name="language" value="german">
45                     <label class="form-check-label" for="german">German</label>
46                 </div>
47               </form>
48             </div>
49           </div>
50           <script src="login.js"></script>
51         </div>
52       </div>
53     </div>
54   </body>
55 </html>
```



```
login.html x JS personal.js personal.html test1.py explore.html
login.html > html > body > div.container-fluid > div.row.justify-content-center.mt-5 > div.col-md-6.col-lg-4 > div.card > div.card-body
29   <div class="form-group">
30     <label for="language">Preferred Language:</label>
31     <div class="form-check">
32       <input type="radio" class="form-check-input" id="english" name="language" value="english" checked>
33       <label class="form-check-label" for="english">English</label>
34     </div>
35     <div class="form-check">
36       <input type="radio" class="form-check-input" id="french" name="language" value="french">
37       <label class="form-check-label" for="french">French</label>
38     </div>
39     <div class="form-check">
40       <input type="radio" class="form-check-input" id="spanish" name="language" value="spanish">
41       <label class="form-check-label" for="spanish">Spanish</label>
42     </div>
43     <div class="form-check">
44       <input type="radio" class="form-check-input" id="german" name="language" value="german">
45       <label class="form-check-label" for="german">German</label>
46     </div>
47     <div class="form-check d-flex justify-content-between">
48       <div>
49         <input type="checkbox" class="form-check-input" id="keepSignedIn" name="keepSignedIn">
50         <label class="form-check-label" for="keepSignedIn">Keep me signed in</label>
51       </div>
52       <button type="submit" class="btn btn-primary" id="login_btn">Login</button>
53     </div>
54   </form>
55 </div>
56 </div>
57 </div>
58 </div>
59 </div>
60 </div>
61 <script src="login.js"></script>
62 </body>
63 </html>
```

## login.js



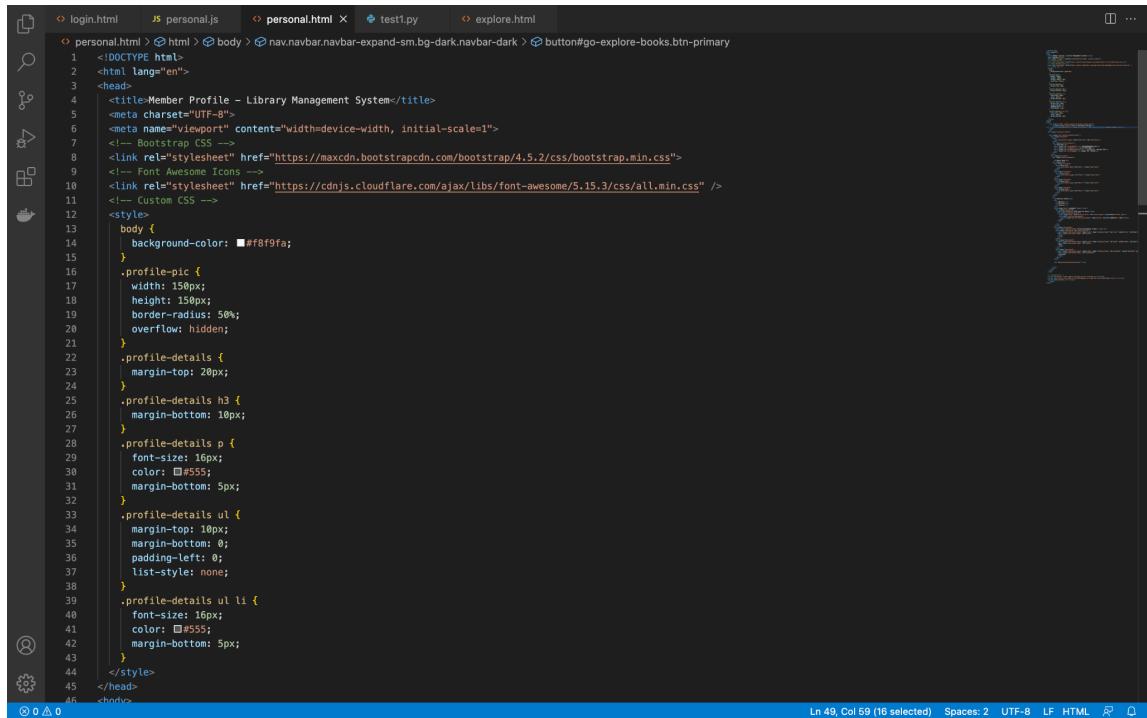
```
login.html JS login.js X JS personal.js personal.html test1.py explore.html ...  
JS login.js > $() callback > $(on('submit') callback  
1 $function() {  
2   // get the "stay signed in" checkbox  
3   var staySignedInCheckbox = $('#staySignedIn');  
4  
5   // check if the user has previously selected "stay signed in"  
6   var savedUsername = localStorage.getItem('username');  
7   var savedPassword = localStorage.getItem('password');  
8   var savedStaySignedIn = localStorage.getItem('staySignedIn');  
9  
10  // if "stay signed in" was previously selected, pre-fill the username and password fields  
11  if (savedStaySignedIn === 'true') {  
12    $('#username').val(savedUsername);  
13    $('#password').val(savedPassword);  
14    staySignedInCheckbox.prop('checked', true);  
15  }  
16  
17  // handle form submission  
18  $('#loginForm').on('submit', function(event) {  
19    event.preventDefault(); // prevent form submission  
20  
21    // get username and password from form input fields  
22    var username = $('#username').val();  
23    var password = $('#password').val();  
24  
25    // check if username and password match the hardcoded values  
26    if (username === 'Junaita' && password === 'Library123') {  
27      window.location.href = 'personal.html';  
28  
29      // save username and password to local storage if "stay signed in" is checked  
30      if (!staySignedInCheckbox.is(':checked')) {  
31        localStorage.setItem('username', username);  
32        localStorage.setItem('password', password);  
33        localStorage.setItem('staySignedIn', true);  
34      } else {  
35        localStorage.removeItem('username');  
36        localStorage.removeItem('password');  
37        localStorage.removeItem('staySignedIn');  
38      }  
39  
40      // do something else after successful login  
41    } else {  
42      alert('Invalid username or password.');  
43      // do something else after failed login attempt  
44    }  
45  });  
46 };
```

Ln 30, Col 51 Spaces: 2 UTF-8 LF ⓘ JavaScript ⌂ ⌂

## Personal page-

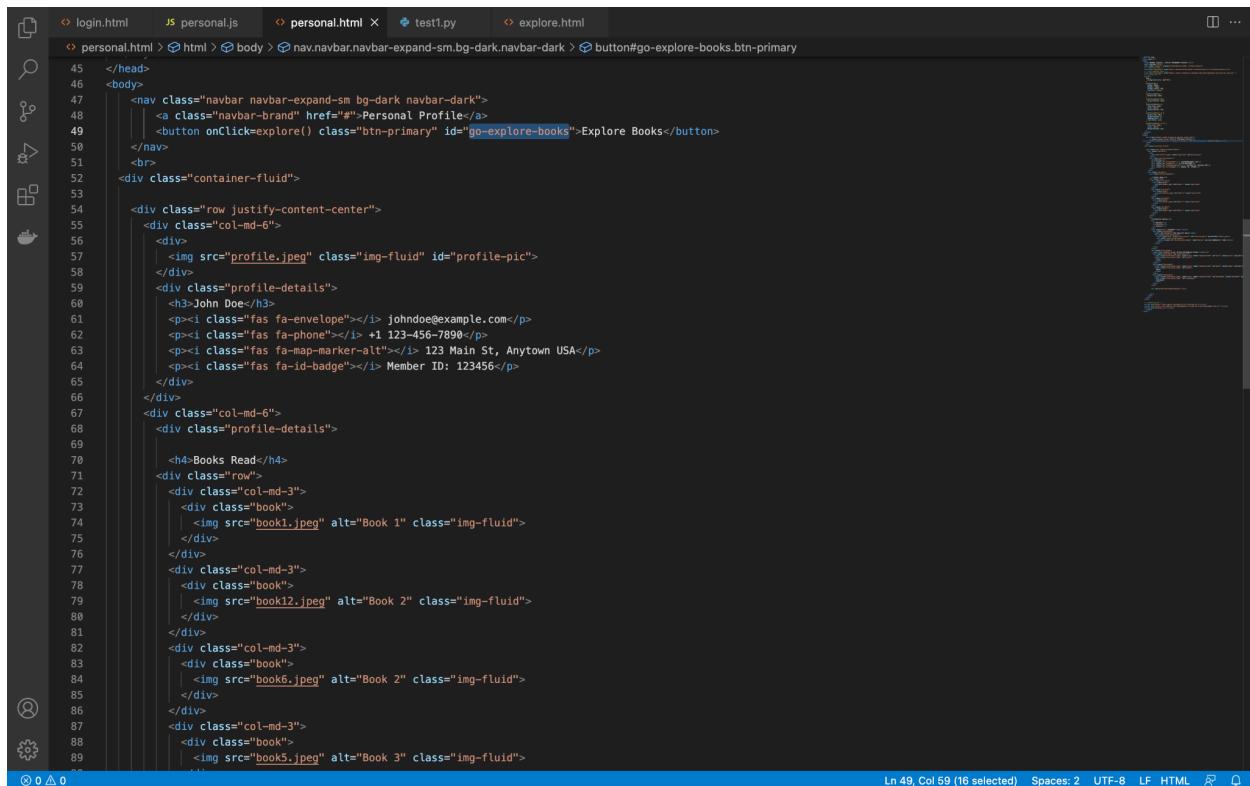
The screenshot present below are those of the html file and javascript file of the personal page in Version 1

## personal.html



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Member Profile - Library Management System</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- Bootstrap CSS -->
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <!-- Font Awesome Icons -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" />
    <!-- Custom CSS -->
    <style>
      body {
        background-color: #f8f9fa;
      }
      .profile-pic {
        width: 150px;
        height: 150px;
        border-radius: 50%;
        overflow: hidden;
      }
      .profile-details {
        margin-top: 20px;
      }
      .profile-details h3 {
        margin-bottom: 10px;
      }
      .profile-details p {
        font-size: 16px;
        color: #555;
        margin-bottom: 5px;
      }
      .profile-details ul {
        margin-top: 10px;
        margin-bottom: 0;
        padding-left: 0;
        list-style: none;
      }
      .profile-details ul li {
        font-size: 16px;
        color: #555;
        margin-bottom: 5px;
      }
    </style>
  </head>
  <body>
```

Ln 49, Col 59 (16 selected) Spaces: 2 UTF-8 LF HTML ⚙

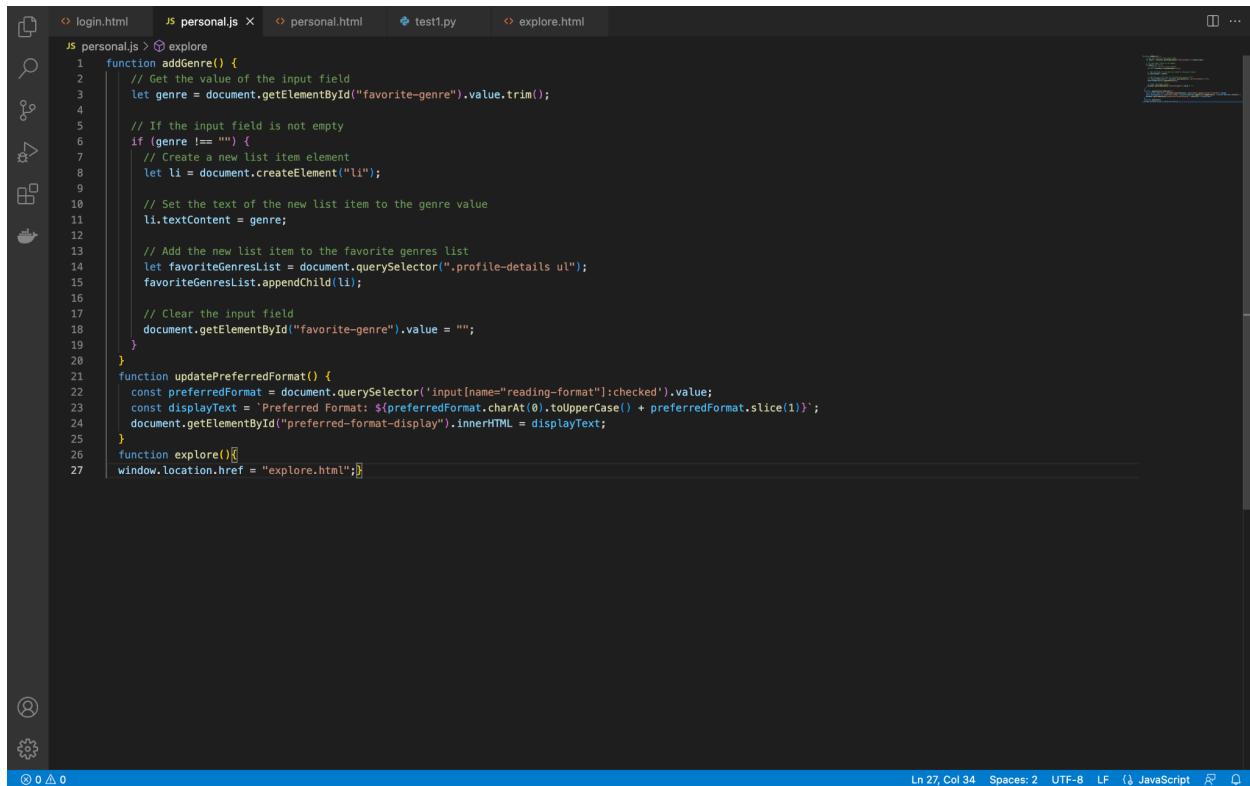


```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <a class="navbar-brand" href="#">Personal Profile</a>
  <button onClick=explore() class="btn-primary" id="go-explore-books">Explore Books</button>
</nav>
<br>
<div class="container-fluid">
  <div class="row justify-content-center">
    <div class="col-md-6">
      <div>
        
      </div>
      <div class="profile-details">
        <h3>John Doe</h3>
        <p><i class="fas fa-envelope"></i> johndoe@example.com</p>
        <p><i class="fas fa-phone"></i> +1 123-456-7890</p>
        <p><i class="fas fa-map-marker-alt"></i> 123 Main St, Anytown USA</p>
        <p><i class="fas fa-id-badge"></i> Member ID: 123456</p>
      </div>
    </div>
    <div class="col-md-6">
      <div class="profile-details">
        <h4>Books Read</h4>
        <div class="row">
          <div class="col-md-3">
            <div class="book">
              
            </div>
          </div>
          <div class="col-md-3">
            <div class="book">
              
            </div>
          </div>
          <div class="col-md-3">
            <div class="book">
              
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
```

Ln 49, Col 59 (16 selected) Spaces: 2 UTF-8 LF HTML ⚙



## personal.js



The screenshot shows a code editor interface with a dark theme. The central area displays the content of the `personal.js` file. The file contains JavaScript code for managing favorite genres. It includes functions for adding a genre, updating the preferred reading format, and navigating to an explore page. The code uses modern JavaScript features like arrow functions and template literals. The status bar at the bottom right indicates the file has 27 lines, 34 columns, and is saved in UTF-8 format.

```
login.html    JS personal.js > explore    personal.html    test1.py    explore.html
JS personal.js > explore
1  function addGenre() {
2      // Get the value of the input field
3      let genre = document.getElementById("favorite-genre").value.trim();
4
5      // If the input field is not empty
6      if (genre !== "") {
7          // Create a new list item element
8          let li = document.createElement("li");
9
10         // Set the text of the new list item to the genre value
11         li.textContent = genre;
12
13         // Add the new list item to the favorite genres list
14         let favoriteGenresList = document.querySelector(".profile-details ul");
15         favoriteGenresList.appendChild(li);
16
17         // Clear the input field
18         document.getElementById("favorite-genre").value = "";
19     }
20
21     function updatePreferredFormat() {
22         const preferredFormat = document.querySelector('input[name="reading-format"]:checked').value;
23         const displayText = `Preferred Format: ${preferredFormat.charAt(0).toUpperCase()} + ${preferredFormat.slice(1)}`;
24         document.getElementById("preferred-format-display").innerHTML = displayText;
25     }
26
27     function explore(){
28         window.location.href = "explore.html";
29     }

```

## Explore Page-

The screenshot present below are those of the html file and javascript file of the explore page in Version 1.

## explore.html

```
login.html test1.py explore.html
1 <!DOCTYPE html>
2 <html> <body> <div.container-fluid.my-3> <div.row> <div.col-md-3.col-lg-2.mb-3> <div.card.fiction>
3 <head>
4   <title>Explore Books</title>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
8   <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
9   <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>
10  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
11  <style>
12    .card-img-top {
13      height: 250px;
14      object-fit: cover;
15    }
16    .summary-arrow {
17      margin-left: 5px;
18      font-size: 0.8em;
19      display: inline-block;
20      transition: transform 0.2s ease-in-out;
21    }
22    .summary-arrow.collapsed {
23      transform: rotate(-90deg);
24    }
25    .summary-text {
26      margin-left: 5px;
27      font-size: 0.8em;
28      display: inline-block;
29    }
30  </style>
31 </head>
32 <body>
33   <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
34     <a class="navbar-brand" href="#">Book Library</a>
35     <a href="#">personal.html</a><button id="person-pro">Profile</button></a>
36   </nav>
37   <ul class="navbar-nav ml-auto">
38     <li class="nav-item">
39       <form class="form-inline">
40         <input class="form-control mr-sm-2" type="search" placeholder="Search by genre" aria-label="Search" id="searchInput">
41         <button class="btn btn-outline-success my-2 my-sm-0" type="button" id="searchButton">Search</button>
42       </form>
43     </li>
44   </ul>
45 </body>
46 </html>
```

Ln 104, Col 7 Spaces: 4 UTF-8 LF HTML ⚡

```
login.html test1.py explore.html
47 <div class="container-fluid my-3">
48   <h2 class="text-center" id="explore-title">Explore Books</h2>
49   <div class="form-check">
50     <input class="form-check-input" type="checkbox" id="genre-preference" name="genre-preference">
51     <label class="form-check-label" for="genre-preference">
52       | Display books based on genre preference
53     </label>
54   </div>
55
56   <hr>
57   <div class="row">
58
59     <div class="col-md-3 col-lg-2 mb-3">
60       <div class="card fiction">
61         
62         <div class="card-body">
63           <h5 class="card-title">The Great Gatsby</h5>
64           <p class="card-text">Fiction</p>
65           <a class="summary-link" href="#book1Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book1Summary">
66             <span class="summary-text">Summary</span>
67           </a>
68           <div class="collapse" id="book1Summary">
69             <div class="card-body">
70               <div class="summary-content" style="height: 100px; overflow-y: scroll;">
71                 <p>The Great Gatsby portrays the decadence and excess of the Roaring Twenties through the story of Jay Gatsby, a wealthy and enigmatic millionaire who is in love with Nick's wife, Nick's friend, Tom Buchanan's wife, and the mysterious Myrtle Wilson. The novel explores themes of American Dream, the corruption of the upper class, and the loss of innocence during the Jazz Age. The book is considered a classic of American literature and has been adapted into numerous films and stage plays over the years. It is also a great example of F. Scott Fitzgerald's writing style, which is characterized by its lyrical prose and social commentary. The book is available for download in PDF, EPUB, and MOBI formats. You can also purchase it in print or audiobook format. If you're interested in reading more books like The Great Gatsby, you might want to check out The Sun Also Rises by Ernest Hemingway or The Catcher in the Rye by J.D. Salinger. Both of these books are also great examples of American literature and have won numerous awards and critical acclaim. I hope you enjoy reading The Great Gatsby!</p>
72             </div>
73           </div>
74         </div>
75       </div>
76     </div>
77
78     <div class="col-md-3 col-lg-2 mb-3">
79       <div class="card fiction">
80         
81         <div class="card-body">
82           <h5 class="card-title">Invisible Man</h5>
83           <p class="card-text">Fiction</p>
84           <a class="summary-link" href="#book2Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book2Summary">
85             <span class="summary-text">Summary</span>
86           </a>
87           <div class="collapse" id="book2Summary">
88             <div class="card-body">
89               <div class="summary-content" style="height: 100px; overflow-y: scroll;">
90                 <p>Invisible Man is a novel by Ralph Ellison that was first published in 1952. It is considered one of the most important works of African American literature. The book follows the life of a black man named Invisible Man, who is trying to find his place in the world. He is a member of the Black Panthers and is involved in their struggle for civil rights. The book is a powerful exploration of race, identity, and the search for meaning in a world that often seems to be against the protagonist. It is also a critique of American society and its treatment of minorities. The book is available for download in PDF, EPUB, and MOBI formats. You can also purchase it in print or audiobook format. If you're interested in reading more books like Invisible Man, you might want to check out The Color Purple by Alice Walker or Beloved by Toni Morrison. Both of these books are also great examples of African American literature and have won numerous awards and critical acclaim. I hope you enjoy reading Invisible Man!</p>
91         </div>
92       </div>
93     </div>
94   </div>
95 </div>
```

Ln 104, Col 7 Spaces: 4 UTF-8 LF HTML ⚡

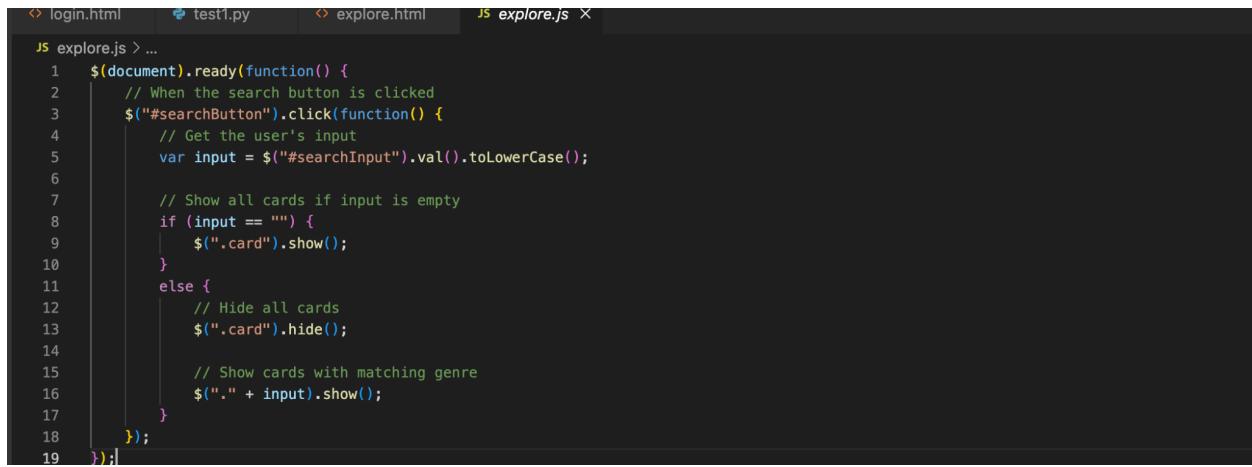
```
login.html test1.py explore.html
explore.html > html > body > div.container-fluid.my-3 > div.row > div.col-md-3.col-lg-2.mb-3 > div.card.fiction
 94
 95
 96
 97
 98
 99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
  </a>
  <div class="collapse" id="book2Summary">
    <p>Science fiction novel by H.G. Wells that tells the story of a scientist who discovers the secret of invisibility, but is driven mad by the
  </div>
</div>
<div class="col-md-3 col-lg-2 mb-3">
  <div class="card fiction">
    
    <div class="card-body">
      <h5 class="card-title">The Trial</h5>
      <p class="card-text">Fiction</p>
      <a class="summary-link" href="#book3Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book3Summary">
        <span class="summary-text">Summary</span>
      </a>
      <div class="collapse" id="book3Summary">
        <p>Novel by Franz Kafka that tells the story of a man named Josef K. who is arrested and put on trial for a crime that is never revealed.</p>
      </div>
    </div>
  </div>
</div>
<div class="col-md-3 col-lg-2 mb-3">
  <div class="card fiction">
    
    <div class="card-body">
      <h5 class="card-title">Middlemarch</h5>
      <p class="card-text">Fiction</p>
      <a class="summary-link" href="#book4Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book4Summary">
        <span class="summary-text">Summary</span>
      </a>
      <div class="collapse" id="book4Summary">
        <p>"Middlemarch" is a novel by George Eliot that portrays the lives and loves of the inhabitants of the fictitious English town of Middle
      </div>
    </div>
  </div>
</div>
<div class="col-md-3 col-lg-2 mb-3">
  <div class="card fiction">
    
    <div class="card-body">
      <h5 class="card-title">Beloved</h5>
      <p class="card-text">Fiction</p>
      <script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>
      <script src="explore.js"></script>
    </div>
  </div>
</div>
<script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>
<script src="explore.js"></script>
</body>
</html>
```

Ln 104, Col 7 Spaces: 4 UTF-8 LF HTML ⚡

```
login.html test1.py explore.html
explore.html > html > body > div.container-fluid.my-3 > div.row > div.col-md-3.col-lg-2.mb-3 > div.card.fiction
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 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
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
  <div class="col-md-3 col-lg-2 mb-3">
    <div class="card comedy">
      
      <div class="card-body">
        <h5 class="card-title">Emma</h5>
        <p class="card-text">Comedy</p>
        <a class="summary-link" href="#book15Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book15Summary">
          <span class="summary-text">Summary</span>
        </a>
        <div class="collapse" id="book15Summary">
          <p>"Emma" is a novel by Jane Austen that tells the story of Emma Woodhouse, a young woman who
        </div>
      </div>
    </div>
</div>
<div class="col-md-3 col-lg-2 mb-3">
  <div class="card comedy">
    
    <div class="card-body">
      <h5 class="card-title">Candide</h5>
      <p class="card-text">Comedy</p>
      <a class="summary-link" href="#book12Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book12Summary">
        <span class="summary-text">Summary</span>
      </a>
      <div class="collapse" id="book12Summary">
        <p>A satirical novel by Voltaire that follows the travels of a young man named Candide an
      </div>
    </div>
  </div>
</div>
<script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>
<script src="explore.js"></script>
</body>
</html>
```

Ln 104, Col 7 Spaces: 4 UTF-8 LF HTML ⚡

## explore.js

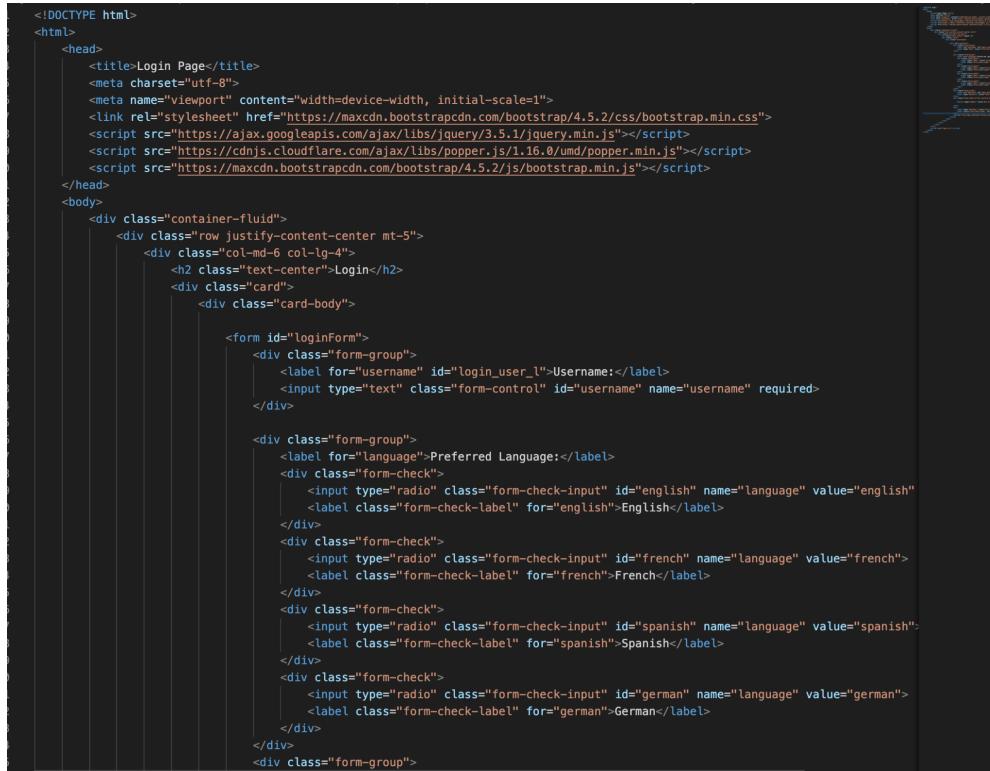


```
JS explore.js > ...
1  $(document).ready(function() {
2      // When the search button is clicked
3      $("#searchButton").click(function() {
4          // Get the user's input
5          var input = $("#searchInput").val().toLowerCase();
6
7          // Show all cards if input is empty
8          if (input == "") {
9              $(".card").show();
10         }
11     else {
12         // Hide all cards
13         $(".card").hide();
14
15         // Show cards with matching genre
16         $("." + input).show();
17     }
18 });
19});
```

## Version 2

**Login page-** The screenshot present below are those of the html file and javascript file of the login page in Version 2.

### login.html



```
<!DOCTYPE html>
<html>
  <head>
    <title>Login Page</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
  </head>
  <body>
    <div class="container-fluid">
      <div class="row justify-content-center mt-5">
        <div class="col-md-6 col-lg-4">
          <h2 class="text-center">Login</h2>
          <div class="card">
            <div class="card-body">
              <form id="loginForm">
                <div class="form-group">
                  <label for="username" id="login_user_l">Username:</label>
                  <input type="text" class="form-control" id="username" name="username" required>
                </div>
                <div class="form-group">
                  <label for="language">Preferred Language:</label>
                  <div class="form-check">
                    <input type="radio" class="form-check-input" id="english" name="language" value="english">
                    <label class="form-check-label" for="english">English</label>
                  </div>
                  <div class="form-check">
                    <input type="radio" class="form-check-input" id="french" name="language" value="french">
                    <label class="form-check-label" for="french">French</label>
                  </div>
                  <div class="form-check">
                    <input type="radio" class="form-check-input" id="spanish" name="language" value="spanish">
                    <label class="form-check-label" for="spanish">Spanish</label>
                  </div>
                  <div class="form-check">
                    <input type="radio" class="form-check-input" id="german" name="language" value="german">
                    <label class="form-check-label" for="german">German</label>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
```

```
html > html > body > div.container-fluid > div.row.justify-content-center.mt-5 > div.col-md-6.col-lg-4 > div.card > div.card-body >
</div>
<div class="form-check">
  <input type="radio" class="form-check-input" id="french" name="language" value="french">
  <label class="form-check-label" for="french">French</label>
</div>
<div class="form-check">
  <input type="radio" class="form-check-input" id="spanish" name="language" value="spanish">
  <label class="form-check-label" for="spanish">Spanish</label>
</div>
<div class="form-check">
  <input type="radio" class="form-check-input" id="german" name="language" value="german">
  <label class="form-check-label" for="german">German</label>
</div>
<div class="form-group">
  <label for="password" id="login_pass_l">Password:</label>
  <input type="password" class="form-control" id="password" name="password" required>
</div>
<div class="form-check d-flex justify-content-between">
  <button type="submit" class="btn btn-primary" id="login_btn">Login</button>
</div>
<div>
  <input type="checkbox" class="form-check-input" id="keepSignedIn" name="keepSignedIn">
  <label class="form-check-label" for="keepSignedIn">Keep me signed in</label>
</div>

</div>
</div>
</div>
<script src="login.js"></script>

```

## login.js

```
$(function() {
    // get the "stay signed in" checkbox
    var staySignedInCheckbox = $('#staySignedIn');

    // check if the user has previously selected "stay signed in"
    var savedUsername = localStorage.getItem('username');
    var savedPassword = localStorage.getItem('password');
    var savedStaySignedIn = localStorage.getItem('staySignedIn');

    // if "stay signed in" was previously selected, pre-fill the username and password fields
    if (savedStaySignedIn === 'true') {
        $('#username').val(savedUsername);
        $('#password').val(savedPassword);
        staySignedInCheckbox.prop('checked', true);
    }

    // handle form submission
    $('#loginForm').on('submit', function(event) {
        event.preventDefault(); // prevent form submission

        // get username and password from form input fields
        var username = $('#username').val();
        var password = $('#password').val();

        // check if username and password match the hardcoded values
        if (username === 'Junaita' && password === 'library123') {
            window.location.href = "personal.html";

            // save username and password to local storage if "stay signed in" is checked
            if (staySignedInCheckbox.is(':checked')) {
                localStorage.setItem('username', username);
                localStorage.setItem('password', password);
                localStorage.setItem('staySignedIn', true);
            } else {
                localStorage.removeItem('username');
                localStorage.removeItem('password');
                localStorage.removeItem('staySignedIn');
            }

            // do something else after successful login
        } else {
            alert('Invalid username or password.');
            // do something else after failed login attempt
        }
    });
});
```

**Personal Page-** The screenshot present below are those of the html file and javascript file of the personal page in Version 2.

## personal.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Member Profile - Library Management System</title>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1">
<!-- Bootstrap CSS -->
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<!-- Font Awesome Icons -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css" />
<!-- Custom CSS -->
<style>
body {
    background-color: #f8f9fa;
}
.profile-pic {
    width: 150px;
    height: 150px;
    border-radius: 50%;
    overflow: hidden;
}
.profile-details {
    margin-top: 20px;
}
.profile-details h3 {
    margin-bottom: 10px;
}
.profile-details p {
    font-size: 16px;
    color: #555;
    margin-bottom: 5px;
}
.profile-details ul {
    margin-top: 10px;
    margin-bottom: 0;
    padding-left: 0;
    list-style: none;
}
.profile-details ul li {
    font-size: 16px;
    color: #555;
    margin-bottom: 5px;
}
</style>
</head>
<body>
```

Ln 1, Col 1   Spaces: 2   UT

```
</head>
<body>
  <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
    |   <a class="navbar-brand" href="#">Personal Profile</a>
    |   <button onClick=explore() class="btn-primary" id="go-explore-books">Explore Books</button>
  </nav>
  <br>
  <div class="container-fluid">

    <div class="row justify-content-center">
      <div class="col-md-6">
        <div>
          |   
        </div>
        <div class="profile-details">
          |   <h3>John Doe</h3>
          |   <p><i class="fas fa-envelope"></i> john.doe@example.com</p>
          |   <p><i class="fas fa-phone"></i> +1 123-456-7890</p>
          |   <p><i class="fas fa-map-marker-alt"></i> 123 Main St, Anytown USA</p>
          |   <p><i class="fas fa-id-badge"></i> Member ID: 123456</p>
        </div>
      </div>
    </div>
    <div class="col-md-6">
      <div class="profile-details">
        |   <h4>Books Read</h4>
        <div class="row">
          <div class="col-md-3">
            <div class="book">
              |   
            </div>
          </div>
          <div class="col-md-3">
            <div class="book">
              |   
            </div>
          </div>
          <div class="col-md-3">
            <div class="book">
              |   
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

<ul>
    <li>Mystery</li>
    <li>Thriller</li>
    <li>Romance</li>
</ul>
<form class="mt-3" onsubmit="return false">
    <div class="form-group">
        <label id="add-genre">Add Favorite Genre</label>
        <div class="input-group mb-3">
            <input type="text" class="form-control" id="favorite-genre" placeholder="Enter genre">
            <div class="input-group-append">
                <button class="btn btn-outline-primary" type="button" onclick="addGenre()">Add</button>
            </div>
        </div>
    </div>
</form>
<div class="form-group">
    <label for="reading-format">Preferred Reading Format:</label><br>
    <div class="form-check"> <input class="form-check-input" type="radio" name="reading-format" id="print" value="print" onclick="updatePreferredFormat()"/> <label class="form-check-label" for="print"> Print </label>
    </div>
    <div class="form-check"> <input class="form-check-input" type="radio" name="reading-format" id="ebook" value="ebook" onclick="updatePreferredFormat()"/> <label class="form-check-label" for="ebook"> Ebook </label>
    </div>
    <div class="form-check"> <input class="form-check-input" type="radio" name="reading-format" id="audiobook" value="audiobook" onclick="updatePreferredFormat()"/> <label class="form-check-label" for="audiobook"> Audiobook </label>
    </div>
</div>
<div id="preferred-format-display"></div>

```

## personal.js

```

function addGenre() {
    // Get the value of the input field
    let genre = document.getElementById("favorite-genre").value.trim();

    // If the input field is not empty
    if (genre !== "") {
        // Create a new list item element
        let li = document.createElement("li");

        // Set the text of the new list item to the genre value
        li.textContent = genre;

        // Add the new list item to the favorite genres list
        let favoriteGenresList = document.querySelector(".profile-details ul");
        favoriteGenresList.appendChild(li);

        // Clear the input field
        document.getElementById("favorite-genre").value = "";
    }
}

function updatePreferredFormat() {
    const preferredFormat = document.querySelector('input[name="reading-format"]:checked').value;
    const displayText = `Preferred Format: ${preferredFormat.charAt(0).toUpperCase() + preferredFormat.slice(1)}`;
    document.getElementById("preferred-format-display").innerHTML = displayText;
}

function explore(){
    window.location.href = "explore.html";
}

```

## Explore Page

The screenshot present below are those of the html file and javascript file of the explore page in Version 2.

explore.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Explore Books</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"></script>
    <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
<style>
    .card-img-top {
        height: 250px;
        object-fit: cover;
    }
    .summary-arrow {
        margin-left: 5px;
        font-size: 0.8em;
        display: inline-block;
        transition: transform 0.2s ease-in-out;
    }
    .summary-arrow.collapsed {
        transform: rotate(-90deg);
    }
    .summary-text {
        margin-left: 5px;
        font-size: 0.8em;
        display: inline-block;
    }
</style>
</head>
<body>
    <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
        <a class="navbar-brand" href="#">Book Library</a>
        <a href="personal.html"><button id="person-profile">Profile</button></a>

        <ul class="navbar-nav ml-auto">
            <li class="nav-item">
                <form class="form-inline">
                    <input class="form-control mr-sm-2" type="search" placeholder="Search by genre" aria-label="Search" id="searchInput">
                    <button class="btn btn-outline-success my-2 my-sm-0" type="button" id="searchButton">Search</button>
                </form>
            </li>
        </ul>
    </nav>
</body>
```

```
test1.py          login.html        personal.html      explore.html <-->
explore.html > html > body > div.container-fluid.my-3 > div.row > div.col-md-3.col-lg-2.mb-3>
49   <div class="container-fluid my-3">
50     <h2 class="text-center" id="explore-title">Explore Books</h2>
51     <div class="form-check">
52       <input class="form-check-input" type="checkbox" id="genre-preference" name="genre-preference">
53       <label class="form-check-label" for="genre-preference">
54         | Display books based on genre preference
55       </label>
56     </div>
57
58
59     <hr>
60     <div class="row">
61
62       <div class="col-md-3 col-lg-2 mb-3">
63         <div class="card fiction">
64           
65           <div class="card-body">
66             <h5 class="card-title">The Great Gatsby</h5>
67             <p class="card-text">Fiction</p>
68             <a class="summary-link" href="#book1Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book1Summary">
69               <span class="summary-text">Summary</span>
70             </a>
71             <div class="collapse" id="book1Summary">
72               <div class="card-body">
73                 <div class="summary-content" style="height: 100px; overflow-y: scroll;">
74                   <p>The Great Gatsby portrays the decadence and excess of the Roaring Twenties through the story of Jay Gatsby, a wealthy and enigmatic millionaire who is obsessed with wealth and status. The novel explores themes of American Dream, the corruption of the upper class, and the loss of innocence in the early 20th century.</p>
75                 </div>
76               </div>
77             </div>
78           </div>
79         </div>
80
81       <div class="col-md-3 col-lg-2 mb-3">
82         <div class="card fiction">
83           
84           <div class="card-body">
85             <h5 class="card-title">Invisible Man</h5>
86             <p class="card-text">Fiction</p>
87             <a class="summary-link" href="#book2Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book2Summary">
88               <span class="summary-text">Summary</span>
89             </a>
90             <div class="collapse" id="book2Summary">
91               <div class="card-body">
92                 <div class="summary-content" style="height: 100px; overflow-y: scroll;">
93                   <p>Science fiction novel by H.G. Wells that tells the story of a scientist who discovers the secret of invisibility, but is driven mad by the power it gives him. The novel explores themes of identity, perception, and the nature of reality.</p>
94                 </div>
95               </div>
96             </div>
97           </div>
98         </div>
99
100      <div class="col-md-3 col-lg-2 mb-3">
101        <div class="card fiction">
102          
103          <div class="card-body">
104            <h5 class="card-title">The Trial</h5>
105            <p class="card-text">Fiction</p>
106            <a class="summary-link" href="#book3Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book3Summary">
107              <span class="summary-text">Summary</span>
108            </a>
109            <div class="collapse" id="book3Summary">
110              <div class="card-body">
111                <div class="summary-content" style="height: 100px; overflow-y: scroll;">
112                  <p>Novel by Franz Kafka that tells the story of a man named Josef K. who is arrested and put on trial for a crime that is never revealed.</p>
113                </div>
114              </div>
115            </div>
116          </div>
117        <div class="col-md-3 col-lg-2 mb-3">
118          <div class="card fiction">
119            
120            <div class="card-body">
121              <h5 class="card-title">Middlemarch</h5>
122              <p class="card-text">Fiction</p>
123              <a class="summary-link" href="#book4Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book4Summary">
124                <span class="summary-text">Summary</span>
125              </a>
126              <div class="collapse" id="book4Summary">
127                <div class="card-body">
128                  <div class="summary-content" style="height: 100px; overflow-y: scroll;">
129                    <p>"Middlemarch" is a novel by George Eliot that portrays the lives and loves of the inhabitants of the fictitious English town of Middlemarch. The novel explores themes of social class, gender roles, and the complexities of human relationships.</p>
130                  </div>
131                </div>
132              </div>
133            <div class="col-md-3 col-lg-2 mb-3">
134              <div class="card fiction">
135                
136                <div class="card-body">
137                  <h5 class="card-title">Beloved</h5>
```

```
test1.py          login.html        personal.html      explore.html <-->
explore.html > html > body > div.container-fluid.my-3 > div.row > div.col-md-3.col-lg-2.mb-3>
92   <div class="container-fluid my-3">
93     <span class="summary-text">Summary</span>
94   </a>
95   <div class="collapse" id="book25Summary">
96     <div class="card-body">
97       <p>Science fiction novel by H.G. Wells that tells the story of a scientist who discovers the secret of invisibility, but is driven mad by the power it gives him. The novel explores themes of identity, perception, and the nature of reality.</p>
98     </div>
99   </div>
100 </div>
101 <div class="col-md-3 col-lg-2 mb-3">
102   <div class="card fiction">
103     
104     <div class="card-body">
105       <h5 class="card-title">The Trial</h5>
106       <p class="card-text">Fiction</p>
107       <a class="summary-link" href="#book3Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book3Summary">
108         <span class="summary-text">Summary</span>
109       </a>
110       <div class="collapse" id="book3Summary">
111         <div class="card-body">
112           <div class="summary-content" style="height: 100px; overflow-y: scroll;">
113             <p>Novel by Franz Kafka that tells the story of a man named Josef K. who is arrested and put on trial for a crime that is never revealed.</p>
114           </div>
115         </div>
116       </div>
117     </div>
118   <div class="col-md-3 col-lg-2 mb-3">
119     <div class="card fiction">
120       
121       <div class="card-body">
122         <h5 class="card-title">Middlemarch</h5>
123         <p class="card-text">Fiction</p>
124         <a class="summary-link" href="#book4Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book4Summary">
125           <span class="summary-text">Summary</span>
126         </a>
127         <div class="collapse" id="book4Summary">
128           <div class="card-body">
129             <div class="summary-content" style="height: 100px; overflow-y: scroll;">
130               <p>"Middlemarch" is a novel by George Eliot that portrays the lives and loves of the inhabitants of the fictitious English town of Middlemarch. The novel explores themes of social class, gender roles, and the complexities of human relationships.</p>
131           </div>
132         </div>
133       </div>
134     <div class="col-md-3 col-lg-2 mb-3">
135       <div class="card fiction">
136         
137         <div class="card-body">
138           <h5 class="card-title">Beloved</h5>
```

```
test1.py      login.html      personal.html      explore.html x
explore.html > html > body > div.container-fluid.my-3 > div.row > div.col-md-3.col-lg-2.mb-3
136             <div class="card-body">
137                 <h5 class="card-title">Beloved</h5>
138                 <p class="card-text">Fiction</p>
139                 <a class="summary-link" href="#book5Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book5Summary">
140                     <span class="summary-text">Summary</span>
141                 </a>
142                 <div class="collapse" id="book5Summary">
143                     <p>Novel by Toni Morrison that tells the story of a former slave named Sethe who is haunted by the memory of her dead daughter</p>
144                 </div>
145             </div>
146         </div>
147     </div>
148     <div class="col-md-3 col-lg-2 mb-3">
149         <div class="card fiction">
150             
151             <div class="card-body">
152                 <h5 class="card-title">Gulliver's Travels</h5>
153                 <p class="card-text">Fiction</p>
154                 <a class="summary-link" href="#book6Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book6Summary">
155                     <span class="summary-text">Summary</span>
156                 </a>
157                 <div class="collapse" id="book6Summary">
158                     <p>Novel by Jonathan Swift that follows the adventures of a sailor named Lemuel Gulliver as he visits various imaginary lands with different cultures and customs</p>
159                 </div>
160             </div>
161         </div>
162     </div>
163     <div class="col-md-3 col-lg-2 mb-3">
164         <div class="card classic">
165             
166             <div class="card-body">
167                 <h5 class="card-title">The Stranger</h5>
168                 <p class="card-text">Classic</p>
169                 <a class="summary-link" href="#book7Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book7Summary">
170                     <span class="summary-text">Summary</span>
171                 </a>
172                 <div class="collapse" id="book7Summary">
173                     <p>Novel by Albert Camus that tells the story of a man named Meursault who kills an Arab man in Algiers and is put on trial for murder</p>
174                 </div>
175             </div>
176         </div>
177     </div>
178     </div>
179 </div>
180 </div>
```

Ln 85, Col 49 Spaces: 4 UTF-8 LF HTML ⚡

```
test1.py      login.html      personal.html      explore.html x
explore.html > html > body > div.container-fluid.my-3 > div.row > div.col-md-3.col-lg-2.mb-3
261             <div class="col-md-3 col-lg-2 mb-3">
262                 <div class="card comedy">
263                     
264                     <div class="card-body">
265                         <h5 class="card-title">Emma</h5>
266                         <p class="card-text">Comedy</p>
267                         <a class="summary-link" href="#book15Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book15Summary">
268                             <span class="summary-text">Summary</span>
269                         </a>
270                         <div class="collapse" id="book15Summary">
271                             <p>"Emma" is a novel by Jane Austen that tells the story of Emma Woodhouse, a young woman who is trying to find a suitable husband</p>
272                         </div>
273                     </div>
274                 </div>
275                 <div class="col-md-3 col-lg-2 mb-3">
276                     <div class="card comedy">
277                         
278                         <div class="card-body">
279                             <h5 class="card-title">Candide</h5>
280                             <p class="card-text">Comedy</p>
281                             <a class="summary-link" href="#book12Summary" data-toggle="collapse" aria-expanded="false" aria-controls="book12Summary">
282                                 <span class="summary-text">Summary</span>
283                             </a>
284                             <div class="collapse" id="book12Summary">
285                                 <p>A satirical novel by Voltaire that follows the travels of a young man named Candide and his journey through various countries and ideologies</p>
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     <script src="https://kit.fontawesome.com/yourcode.js" crossorigin="anonymous"></script>
302     <script src="explore.js"></script>
303 </body>
304 </html>
305 </div>
306 </div>
```

Ln 85, Col 49 Spaces: 4 UTF-8 LF HTML ⚡

## explore.js

```
JS explore.js > ...
1  $(document).ready(function() {
2      // When the search button is clicked
3      $("#searchButton").click(function() {
4          // Get the user's input
5          var input = $("#searchInput").val().toLowerCase();
6
7          // Show all cards if input is empty
8          if (input == "") {
9              $(".card").show();
10         }
11     else {
12         // Hide all cards
13         $(".card").hide();
14
15         // Show cards with matching genre
16         $(".." + input).show();
17     }
18 });
19});
```

## Description of the tool-

The tool that I selected to automate GUI testing is the selenium library in python.

### Scope-

The scope of Selenium Python is vast and encompasses various areas, such as web automation, testing, and web scraping. Using Selenium Python, it is feasible to automate common web browser operations like clicking links, filling forms, and browsing across web pages. In tasks that need repeating actions, like web testing or web scraping, this can reduce duration while improving efficiency. For automated web application testing, Selenium Python is commonly used. It can mimic user interactions with the website, such as completing forms and clicking buttons, to ensure that it functions as planned.

### Purpose-

The purpose of the Selenium library in Python is to provide a set of tools and APIs for web browser automation and testing. A set of tools and APIs for web browser automation and testing are provided by the Python Selenium package. It helps developers to automate browser actions like clicking links, completing forms, and navigating between web pages by using Python scripts to manage web browsers like Chrome, Firefox, and Safari. Selenium supports a variety of web

browsers, including Chrome, Firefox, Safari, and Internet Explorer. This provides a way for developers to create tests that work on multiple platforms and browsers. Support for multiple programming languages: Python, Java, C#, and Ruby are just a few of the languages that Selenium supports. This helps developers to select the language in which they are most at ease.

### **Use Case-**

The main use case of the selenium library in python is automation of testing. This simple python script attached below can be used for validation in a login page. This enters your password and username and clicks the login button. It then checks the destination url, if the credentials are correct all the test cases will pass and the destination url will be reached. [2]

```
from selenium import webdriver

# Set up the Chrome driver
driver = webdriver.Chrome('/path/to/chromedriver')

# Navigate to the login page
driver.get('https://www.example.com/login')

# Fill in the login details
username = driver.find_element_by_id('username')
username.send_keys('your_email@example.com')
password = driver.find_element_by_id('password')
password.send_keys('your_password')

# Click the "Login" button
login_button = driver.find_element_by_id('login-button')
login_button.click()

# Check if the login was successful
if driver.current_url == 'https://www.example.com/inbox':
    print('Login successful!')
else:
    print('Login failed!')

# Close the browser window
driver.quit()
```

## **Test Cases-**

## Screenshots of code-

The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is 'test1.py', which contains Python code for testing a login page using Selenium. The code includes imports for time, unittest, selenium.webdriver, selenium.common.exceptions, and selenium.webdriver.common.by. It defines a class 'TestStringMethods' that inherits from 'unittest.TestCase'. The class has two test methods: 'test1' and 'test2'. 'test1' checks the location and size of a login button. 'test2' checks the size and location of a user name label. Error handling is included for 'NoSuchElementException'. The right side of the screen shows a vertical stack of other code snippets, likely from other notebooks or files.

```
login.html  test1.py  explore.html  explore.js  from selenium import webdriver Untitled-1

test1.py > TestStringMethods > test9
1 import time
2 import unittest
3
4 from selenium import webdriver
5 from selenium.common.exceptions import NoSuchElementException
6
7 from selenium.webdriver.common.by import By
8
9 # from pytime import time
10
11
12 browser = webdriver.Safari()
13 browser.get("file:///Users/iuvuu/Desktop/library_1/login.html")
14 class TestStringMethods(unittest.TestCase):
15     maxDiff = None
16
17     #Login page
18     #test1- Checking image location and size
19     def test1(self):
20         try:
21             l = browser.find_element(By.ID, "lib_login_img")
22             s = l.size
23             location = l.location
24             size = l.size
25             w, h = size["width"], size["height"]
26             tw = 328
27             th = 193
28             xcor = 236
29             ycor = 115
30             x = location.get("x")
31             y = location.get("y")
32             #print("width - ", w)
33             #print("height - ", h)
34             #print("xcor - ", x)
35             #print("ycor - ", y)
36             self.assertEqual((w, h), (tw, th))
37             self.assertEqual((x, y), (xcor, ycor))
38             print("Testcase 1 Passed")
39
40         except NoSuchElementException:
41             print("Test case 1 failed")
42
43     #test2- Username label size and location and text
44     def test2(self):
45         try:
46             l = browser.find_element(By.ID, "login_user_1")
```

This screenshot shows a code editor interface with multiple tabs open. The active tab is `test1.py`, which contains Python code for testing a login page. The code includes several test cases using the `selenium` library to interact with elements like `login_user_l` and `login_pass_l`. It uses assertions to check element sizes and locations.

```
43     #test2- Username label size and location and text
44     def test2(self):
45         try:
46             l = browser.find_element(By.ID, "login_user_l")
47             s = l.text
48             location = l.location
49             size = l.size
50             w, h = size["width"], size["height"]
51             tw = 79
52             th = 25
53             xcor = 236
54             ycor = 323
55             x = location.get("x")
56             y = location.get("y")
57             #print("width - ", w)
58             #print("height - ", h)
59             #print("xcor - ", x)
60             #print("ycor - ", y)
61             self.assertEqual((w, h), (tw, th))
62             self.assertEqual((x, y), (xcor, ycor))
63             value = "Username:"
64             self.assertEqual(s.strip(), value.strip())
65             print("Testcase 2 Passed")
66
67         except NoSuchElementException:
68             print("Test Case 2 failed")
69
70     #test3- Password label size and location and text
71     def test3(self):
72         try:
73             l = browser.find_element(By.ID, "login_pass_l")
74             s = l.text
75             location = l.location
76             size = l.size
77             w, h = size["width"], size["height"]
78             tw = 75
79             th = 25
80             xcor = 236
81             ycor = 409
82             x = location.get("x")
83             y = location.get("y")
84             #print("width - ", w)
85             #print("height - ", h)
86             #print("xcor - ", x)
87             #print("ycor - ", y)
88             self.assertEqual((w, h), (tw, th))
89             self.assertEqual((x, y), (xcor, ycor))
90             print("Testcase 3 Passed")
91
92         except NoSuchElementException:
93             print("Test Case 3 failed")
94
95     #test4- Username input box size and location
96     def test4(self):
97         try:
98             l = browser.find_element(By.ID, "username")
99             s = l.text
100            location = l.location
101            size = l.size
102            w, h = size["width"], size["height"]
103            tw = 328
104            th = 39
105            xcor = 236
106            ycor = 355
107            x = location.get("x")
108            y = location.get("y")
109            #print("width - ", w)
110            #print("height - ", h)
111            #print("xcor - ", x)
112            #print("ycor - ", y)
113            self.assertEqual((w, h), (tw, th))
114            self.assertEqual((x, y), (xcor, ycor))
115            print("Testcase 4 Passed")
116
117        except NoSuchElementException:
118            print("Test Case 4 failed")
119
120    #test5- Password input box size and location
121    def test5(self):
122        try:
123            l = browser.find_element(By.ID, "password")
124            s = l.text
125            location = l.location
126            size = l.size
127            w, h = size["width"], size["height"]
128            tw = 328
129            th = 39
130            xcor = 236
131            ycor = 441
132            x = location.get("x")
133            y = location.get("y")
134            #print("width - ", w)
135            #print("height - ", h)
```

This screenshot shows a second instance of a code editor with the same `test1.py` file open. The code is identical to the one in the first screenshot, containing five test cases for a login page using the `selenium` library.

```
43     #test2- Username label size and location and text
44     def test2(self):
45         try:
46             l = browser.find_element(By.ID, "login_user_l")
47             s = l.text
48             location = l.location
49             size = l.size
50             w, h = size["width"], size["height"]
51             tw = 79
52             th = 25
53             xcor = 236
54             ycor = 323
55             x = location.get("x")
56             y = location.get("y")
57             #print("width - ", w)
58             #print("height - ", h)
59             #print("xcor - ", x)
60             #print("ycor - ", y)
61             self.assertEqual((w, h), (tw, th))
62             self.assertEqual((x, y), (xcor, ycor))
63             value = "Username:"
64             self.assertEqual(s.strip(), value.strip())
65             print("Testcase 2 Passed")
66
67         except NoSuchElementException:
68             print("Test Case 2 failed")
69
70     #test3- Password label size and location and text
71     def test3(self):
72         try:
73             l = browser.find_element(By.ID, "login_pass_l")
74             s = l.text
75             location = l.location
76             size = l.size
77             w, h = size["width"], size["height"]
78             tw = 75
79             th = 25
80             xcor = 236
81             ycor = 409
82             x = location.get("x")
83             y = location.get("y")
84             #print("width - ", w)
85             #print("height - ", h)
86             #print("xcor - ", x)
87             #print("ycor - ", y)
88             self.assertEqual((w, h), (tw, th))
89             self.assertEqual((x, y), (xcor, ycor))
90             print("Testcase 3 Passed")
91
92         except NoSuchElementException:
93             print("Test Case 3 failed")
94
95     #test4- Username input box size and location
96     def test4(self):
97         try:
98             l = browser.find_element(By.ID, "username")
99             s = l.text
100            location = l.location
101            size = l.size
102            w, h = size["width"], size["height"]
103            tw = 328
104            th = 39
105            xcor = 236
106            ycor = 355
107            x = location.get("x")
108            y = location.get("y")
109            #print("width - ", w)
110            #print("height - ", h)
111            #print("xcor - ", x)
112            #print("ycor - ", y)
113            self.assertEqual((w, h), (tw, th))
114            self.assertEqual((x, y), (xcor, ycor))
115            print("Testcase 4 Passed")
116
117        except NoSuchElementException:
118            print("Test Case 4 failed")
119
120    #test5- Password input box size and location
121    def test5(self):
122        try:
123            l = browser.find_element(By.ID, "password")
124            s = l.text
125            location = l.location
126            size = l.size
127            w, h = size["width"], size["height"]
128            tw = 328
129            th = 39
130            xcor = 236
131            ycor = 441
132            x = location.get("x")
133            y = location.get("y")
134            #print("width - ", w)
135            #print("height - ", h)
```

```
login.html test1.py explore.html explore.js from selenium import webdriver Untitled-1

#test9- Login button size and location
def test9(self):
    try:
        l = browser.find_element(By.ID, "login_btn")
        s = l.text
        location = l.location
        size = l.size
        w, h = size["width"], size["height"]
        tw = 66
        th = 39
        xcor = 498
        ycor = 639
        x = location.get("x")
        y = location.get("y")
        #print("width - ", w)
        #print("height - ", h)
        #print("xcor - ", x)
        #print("ycor - ", y)
        self.assertEqual(w, h), (tw, th)
        self.assertEqual(x, y), (xcor, ycor)
        print("Testcase 5 Passed")
    except NoSuchElementException:
        print("Test Case 5 failed")
#test6- Login button size and location
def test6(self):
    try:
        l = browser.find_element(By.ID, "login_btn")
        s = l.text
        location = l.location
        size = l.size
        w, h = size["width"], size["height"]
        tw = 66
        th = 39
        xcor = 498
        ycor = 639
        x = location.get("x")
        y = location.get("y")
        #print("width - ", w)
        #print("height - ", h)
        #print("xcor - ", x)
        #print("ycor - ", y)
        self.assertEqual(w, h), (tw, th)
        self.assertEqual(x, y), (xcor, ycor)
        print("Testcase 6 Passed")
    except NoSuchElementException:
        print("Test Case 6 failed")
#test7- Keep me signed in checkbox size and location
def test7(self):
    try:
        l = browser.find_element(By.ID, "keepSignedIn")
        s = l.text
        location = l.location
        size = l.size
        w, h = size["width"], size["height"]
        tw = 12
        th = 13
        xcor = 236
        ycor = 644
        x = location.get("x")
    except NoSuchElementException:
        print("Test Case 7 failed")
Ln 237, Col 40 Spaces: 4 UTF-8 LF Python 3.9.13 ('base': conda) ⌂ ⌂
```

```
login.html test1.py explore.html explore.js from selenium import webdriver Untitled-1

#test9- Login button size and location
def test9(self):
    try:
        l = browser.find_element(By.ID, "login_btn")
        s = l.text
        location = l.location
        size = l.size
        w, h = size["width"], size["height"]
        tw = 66
        th = 39
        xcor = 498
        ycor = 639
        x = location.get("x")
        y = location.get("y")
        #print("width - ", w)
        #print("height - ", h)
        #print("xcor - ", x)
        #print("ycor - ", y)
        self.assertEqual(w, h), (tw, th)
        self.assertEqual(x, y), (xcor, ycor)
        print("Testcase 7 Passed")
    except NoSuchElementException:
        print("Test Case 7 failed")
#test8- Entering the credentials and logging in. Checking the flow
def test8(self):
    try:
        username = browser.find_element(By.ID, "username")
        username.clear()
        username.send_keys("Junaita")

        password = browser.find_element(By.ID, "password")
        password.clear()
        password.send_keys("Library123")

        browser.find_element(By.ID, "login_btn").click()

        i = 0
        while i < 20:
            i = i + 1

        l = browser.find_element(By.ID, "profile-pic")
    except:
        print("Testcase 8 Failed")
#test9(self):
def test9(self):
    try:
        l = browser.find_element(By.ID, "add-genre")
        s = l.text
        location = l.location
        size = l.size
        w, h = size["width"], size["height"]
        tw = 139
        th = 24
        xcor = 415
        ycor = 412
        x = location.get("x")
    except NoSuchElementException:
        print("Test Case 9 failed")
Ln 237, Col 40 Spaces: 4 UTF-8 LF Python 3.9.13 ('base': conda) ⌂ ⌂
```

The screenshot shows a Jupyter Notebook interface with several tabs at the top: 'login.html', 'test1.py', 'explore.html', 'explore.js', and 'from selenium import webdriver Untitled-1'. The main area contains Python test code for a login page:

```
209
210
211
212
213     l = browser.find_element(By.ID, "profile-pic")
214     print("Testcase 8 Passed")
215
216 except:
217     print("Testcase 8 Failed")
218
219 def test9(self):
220     try:
221         l = browser.find_element(By.ID, "add-genre")
222         s = l.text
223         location = l.location
224         size = l.size
225         w, h = size["width"], size["height"]
226         tw = 139
227         th = 24
228         xcor = 415
229         ycor = 412
230         x = location.get("x")
231         y = location.get("y")
232         #print("width - ", w)
233         #print("height - ", h)
234         #print("xcor - ", x)
235         #print("ycor - ", y)
236         self.assertEqual((w, h), (tw, th))
237         self.assertEqual((x, y), (xcor, ycor))
238         print("Testcase 9 Passed")
239
240
241 if __name__ == "__main__":
242     unittest.main()
```

The status bar at the bottom indicates: Ln 237, Col 40, Spaces: 4, UTF-8, LF, Python 3.9.13 ('base': conda).

## Test case 1

Test Case 1 is used to check the image location and size of the image in the login page. I first ran the test case and fetched the location and size and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

## Test Case2

Test Case 2 is used to check the username label location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

## Test Case 3

Test Case 3 is used to check the password label location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

#### **Test Case 4**

Test Case 4 is used to check the username input box location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

#### **Test Case 5**

Test Case 5 is used to check the password input box location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

#### **Test Case 6**

Test Case 6 is used to check the login button location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

#### **Test Case 7**

Test Case 7 is used to check the keep me signed in checkbox location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

#### **Test Case 8**

Entering the credentials in the login page and checking if the flow is correctly going to the personal page.

#### **Test Case 9**

Test Case 9 is used to check the add genre input box location and size in the login page. I first ran the test case and fetched the location and then assigned it to the variables in version 1. The same assigned variables were used to test version 2.

#### **Output-**

Version 1- All the test cases are passed.

```
(base) juuuu@Junaitas-MacBook-Air library_1 % python test1.py
Testcase 1 Passed
.Testcase 2 Passed
.Testcase 3 Passed
.Testcase 4 Passed
.Testcase 5 Passed
.Testcase 6 Passed
.Testcase 7 Passed
.Testcase 8 Passed
.Testcase 9 Passed
.

-----
Ran 9 tests in 0.388s

OK
```

## Version 2-

All the testcases failed as all the orientation that is being checked is changed in Version 2. The last test case that checks the location and size of the add genre input box in the personal page is not reached as the flow is changed from personal page to the explore page in version 2. The comparison between the location in version 1 and version 2 is also displayed in the output.

```
(base) juuuu@Junaitas-MacBook-Air library_2 % python test1.py
FFFFFFFFFFTestcase 8 Failed
.Test Case 9 failed
.
=====
FAIL: test1 (_main_.TestStringMethods)
-----
Traceback (most recent call last):
  File "/Users/juuuu/Desktop/library_2/test1.py", line 37, in test1
    self.assertEqual(x, y), (xcor, ycor))
AssertionError: Tuples differ: (236, 493) != (236, 115)

First differing element 1:
493
115
-
(236, 493)
+
(236, 115)

=====
FAIL: test2 (_main_.TestStringMethods)
-----
Traceback (most recent call last):
  File "/Users/juuuu/Desktop/library_2/test1.py", line 61, in test2
    self.assertEqual((w, h), (tw, th))
AssertionError: Tuples differ: (79, 24) != (79, 25)

First differing element 1:
24
25
-
(79, 24)
?
+
(79, 25)
?

=====
FAIL: test3 (_main_.TestStringMethods)
-----
Traceback (most recent call last):
  File "/Users/juuuu/Desktop/library_2/test1.py", line 87, in test3
    self.assertEqual((w, h), (tw, th))
AssertionError: Tuples differ: (75, 24) != (75, 25)

First differing element 1:
24
25
-
(75, 24)
?
+
(75, 25)
```

```
FAIL: test4 (_main_.TestStringMethods)
=====
Traceback (most recent call last):
  File "/Users/jiuhuu/Desktop/library_2/test1.py", line 113, in test4
    self.assertEqual(w, h), (tw, th)
AssertionError: Tuples differ: (328, 38) != (328, 39)

First differing element 1:
38
39
-
  (328, 38)
?
+
  (328, 39)
?

=====
FAIL: test5 (_main_.TestStringMethods)
=====
Traceback (most recent call last):
  File "/Users/jiuhuu/Desktop/library_2/test1.py", line 137, in test5
    self.assertEqual(w, h), (tw, th)
AssertionError: Tuples differ: (328, 38) != (328, 39)

First differing element 1:
38
39
-
  (328, 38)
?
+
  (328, 39)
?

=====
FAIL: test6 (_main_.TestStringMethods)
=====
Traceback (most recent call last):
  File "/Users/jiuhuu/Desktop/library_2/test1.py", line 161, in test6
    self.assertEqual(w, h), (tw, th)
AssertionError: Tuples differ: (66, 38) != (66, 39)

First differing element 1:
38
39
-
  (66, 38)
?
+
  (66, 39)
```

```
test1.py          login.html      JS login.js      personal.html

JS login.js > $() callback > on('submit') callback
15
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER
zsh + □ 會 ×

File "/Users/juuuu/Desktop/library_2/test1.py", line 137, in test5
  self.assertEqual(w, h), (tw, th)
AssertionError: Tuples differ: (328, 38) != (328, 39)

First differing element 1:
38
39
- (328, 38)
?
+
+ (328, 39)
?

=====
FAIL: test6 (_main_.TestStringMethods)

Traceback (most recent call last):
  File "/Users/juuuu/Desktop/library_2/test1.py", line 161, in test6
    self.assertEqual((w, h), (tw, th))
AssertionError: Tuples differ: (66, 38) != (66, 39)

First differing element 1:
38
39
- (66, 38)
?
+
+ (66, 39)
?

=====
FAIL: test7 (_main_.TestStringMethods)

Traceback (most recent call last):
  File "/Users/juuuu/Desktop/library_2/test1.py", line 186, in test7
    self.assertEqual((x, y), (xcor, ycor))
AssertionError: Tuples differ: (236, 473) != (236, 644)

First differing element 0:
216
236
-
- (216, 473)
+
+ (236, 644)

Ran 9 tests in 0.334s
FAILED (failures=7)
(base) juuuu@Junaitas-MacBook-Air library_2 %
```

## **Assessment of the tool-**

### **Features and Functionality-**

Developers frequently choose the Selenium Python library for testing because it offers a complete set of tools and features for GUI testing of web applications. A web application can be evaluated with Selenium in multiple windows, tabs, and frames. Mouse clicks, scrolling, and keyboard input can all be mimicked by Selenium. Selenium can simulate navigational activities including refreshing the page, clicking links, and heading back and forth. The dropdowns, checkboxes, radio buttons, and alerts on a web page can all be interacted with by Selenium. A variety of ways, such by ID, class, name, or XPath, are provided by Selenium to locate elements on a web page. Selenium is compatible with an extensive variety of web browsers, including Chrome, Firefox, Safari, and Edge. [1]

### **Type of Coverage-**

For testing web applications' GUIs, the Selenium Python library offers comprehensive coverage.

Some of the most important coverage options offered by Selenium include the following,

**Functional coverage:** Selenium can verify the accessibility of web applications by simulating user actions like button clicks and form submissions, as well as observing how various page elements respond.

**Regression coverage:** By automating the testing of specific functionality across diverse application versions, Selenium can help in the identification of regression issues.

**Compatibility coverage:** Selenium can test a web application's compatibility with a variety of browsers and mobile devices to ensure it works as anticipated across various platforms.

**Performance coverage:** Selenium can evaluate the functionality of web applications by simulating various user traffic volumes and monitoring the rate at which various pages render.

### **Reuse of Test Cases-**

Reusing test cases across numerous platforms and environments is one of the primary benefits of using the Selenium Python library for GUI testing of web applications. Cross-browser testing enables Selenium to execute tests on a variety of browsers, like Chrome, Firefox, Safari, and Edge. This allows developers to reuse test cases across various browsers. Selenium enables testing with data, which enables developers to reuse test logic across multiple test cases by executing the same test case using different input data. Selenium interfaces with well-known frameworks for test automation, like Pytest, Unittest, and Robot Framework, which offer tools for maintaining and reusing test cases in various settings.

### **Test Results Provided-**

The test results provided by the Selenium python library are very descriptive when the test cases fail. For example, when an element was tested for its location, a comparison between the required coordinates and the coordinates fetched by running the program is provided. This makes rectification of errors convenient. There is no report provided, the output after running the test cases are displayed in the terminal where the program is being executed.

### **Ease of Usage-**

Reusing test cases across numerous platforms and environments is one of the primary benefits of using the Selenium Python library for GUI testing of web applications. Cross-browser testing enables Selenium to execute tests on a variety of browsers, like Chrome, Firefox, Safari, and Edge. This allows developers to reuse test cases across various browsers. Selenium enables testing with data, which enables developers to reuse test logic across multiple test cases by executing the same test case using different input data. Selenium interfaces with well-known frameworks for test automation, like Pytest, Unittest, and Robot Framework, which offer tools for maintaining and reusing test cases in various settings.

### **Types of GUI elements that can be tested-**

Buttons: Selenium may test buttons by clicking them and confirming that they carry out the anticipated action, like submitting a form or going to a different page.

Text fields: Selenium can test text fields by entering text and ensuring that the input is received and displayed properly.

Dropdown lists: Selenium may test dropdown lists by picking various options and confirming that the selected option is displayed correctly.

Images and multimedia components: Selenium may evaluate images and multimedia components by making sure that they are displayed correctly and that they follow out the expected actions, such playing a video or showing a tooltip.

Radio buttons and checkboxes: Selenium can test radio buttons and checkboxes by selecting and deselecting them while guaranteeing the right substitutes are selected.

Tables and grids: Selenium can test tables and grids by making sure that data is shown correctly and that the intended sorting and filtering operations have been carried out.

Pop-ups and dialog boxes: Selenium can test pop-ups and dialog boxes by guaranteeing that they are displayed correctly and that consumers can dismiss or close them as expected.

Navigation elements: Selenium can test navigational elements like menus and links by clicking them and verifying that the desired page or a component is displayed. [2]

## **References-**

[1] <https://www.geeksforgeeks.org/selenium-python-tutorial/#>

[2] <https://www.simplilearn.com/tutorials/python-tutorial/selenium-with-python>