

**CP: Peijia Lu (peijialu@usc.edu)**

**Team Members:**

Andrew Hariri (hariria@usc.edu)

Arthur Krut (akrut@usc.edu)

Bhavin Shah (bhavints@usc.edu)

Majed Jendi (jendi@usc.edu)

Sung Bin Kim (sungbink@usc.edu)

Zach Izzard (izzard@usc.edu)

## **Testing Document**

- Initializations made
  - Populating the items database with 5 items
  - Populating the users database with 6 users

### **Browse Page**

(accessible to both logged out and logged in users):

- Clicking on the heart icon will store the item's information into saved items(if signed in)
- Clicking on the item will redirect the user to the specific product page(if signed in)
- Clicking on the saved items will open a side table showing all the saved items(if signed in)
- Clicking the delete button will remove the specific item from the saved items
- Clicking on the sign in button will prompt the google api and request information. After signing in, the button will change to account
- If the user is not signed in, requesting any functionality will prompt the user to sign in

### **Product Page**

(accessible to both logged out and logged in users):

- Clicking on the heart icon will store the item's information into saved items(if signed in)
- Clicking on the saved items will open a side table showing all the saved items(if signed in)
- Clicking the delete button will remove the specific item from the saved items
- Clicking on the message button will open a simple form to send to the seller(if signed in)
- If some information is missing, prompts the user to fill out missing information
- Clicking on the account button will redirect to the account page
- Clicking on the icon on the upper left will redirect to the browse page
- If the user is not signed in, requesting any functionality will prompt the user to sign in

### **Account Page**

(only accessible to signed in users):

- Clicking on the icon on the upper left will redirect to the browse page

- Clicking on the add listing button will redirect to the add listing button
- Clicking on the delete button on the listed item allows the users to delete the item.
- Clicking on the reset saving button will delete all the saved items after a confirmation message

## Add Listing Page

- If the user tries to submit a new listing with missing information, program will alert user to fill out missing information
- Successful submission will redirect to the account page

```

1  import json
2  import requests
3
4
5  def test_correct_login(self):
6
7      #Adding all the random users to the database
8      data = {}
9      data['Users'] = []
10     data['User'].append({
11         'objectId': '1',
12         'firstName': 'Majed',
13         'lastName': 'Jendi',
14         'phoneNumber': '2035550000',
15         'zipCode': '90007',
16         'googleUserId': '1111111111',
17         'userRating': '3',
18         'savedItems': ''
19     })
20     data['User'].append({
21         'objectId': '2',
22         'firstName': 'Andrew',
23         'lastName': 'Hariri',
24         'phoneNumber': '2035550000',
25         'zipCode': '90007',
26         'googleUserId': '7777777777',
27         'userRating': '3',
28         'savedItems': ''
29     })
30     data['User'].append({
31         'objectId': '3',
32         'firstName': 'Zach',
33         'lastName': 'Izzard',
34         'phoneNumber': '2035550000',
35         'zipCode': '90007',
36         'googleUserId': '5555555555',
37         'userRating': '',
38         'savedItems': ''
39     })
40     data['User'].append({

```

```

38     })
39     data['User'].append({
40         'objectId': '4',
41         'firstName': 'Bhavin',
42         'lastName': 'Shah',
43         'phoneNumber': '2035550000',
44         'zipCode': '90007',
45         'googleUserId': 'AAAA44444444',
46         'userRating': '',
47         'savedItems': ''
48     })
49     data['User'].append({
50         'objectId': '5',
51         'firstName': 'Arthur Krut',
52         'phoneNumber': '2035550000',
53         'zipCode': '90007',
54         'googleUserId': '3333333333',
55         'userRating': '',
56         'savedItems': ''
57     })
58     data['User'].append({
59         'objectId': '6',
60         'firstName': 'Brain',
61         'lastName': 'Kin',
62         'phoneNumber': '2035550000',
63         'zipCode': '90007',
64         'googleUserId': '2222222222',
65         'userRating': '',
66         'savedItems': ''
67     })
68 })
69
70 json_data = json.dumps(data)
71 # /...
72 #Testing to see that all users are added to the DB
73 for user in json_data:
74     url = "http://localhost:9000/Users/"
75     # Specify the url: url
76

```

```

69     json_data = json.dumps(data)
70
71     # /....
72     #Testing to see that all users are added to the DB
73     for user in json_data:
74         url = "http://localhost:9000/Users/"
75         # Specify the url: url
76
77         # Packages the request, send the request and catch the response: r
78         r = requests.post(url + ObjectId)
79
80         # Extract the response: text
81         text = r.text
82
83         # Print the html
84         print(text)
85
86     for user in json_data:
87         url = "http://localhost:9000/Users/"
88         # Specify the url: url
89
90         # Packages the request, send the request and catch the response: r
91         r = requests.get(url + ObjectId)
92
93         # Extract the response: text
94         text = r.ObjectId
95
96         # Print the html
97         print("Expected output is: " + user.ObjectId)
98         print("Actual output is: " + ObjectId)
99
100     #Testing if an item is being added to the database
101     url = "http://localhost:9000/Items/"
102     data = {}
103     item['item'] = []
104     item['item'].append({
105         'ObjectId': '1'
106         'title': 'Honda',
107         'description': 'Car',

```

```

106         'title': 'Honda',
107         'description': 'Car',
108         'condition': 'Good',
109         'forSale': 'true',
110         'userSellingItem': '1',
111         'highestBidder': '',
112         'endForSaleDate': '',
113         'startForSaleDate': '',
114         'maxBid': '',
115         'image': '',
116         'location': ''
117     })
118
119     r = requests.post(url + data)
120
121     #Testing if the output will match the extracted item
122     r = requests.get(url + ObjectId)
123     # Extract the response: text
124     text = r.ObjectId
125     # Print the html
126     print("Expected output is: " + item.ObjectId)
127     print("Actual output is: " + text)
128
129     text = r.title
130     print("Expected output is: " + item.title)
131     print("Actual output is: " + text)
132
133     text = r.condition
134     print("Expected output is: " + item.condition)
135     print("Actual output is: " + text)
136
137     text = r.description
138     print("Expected output is: " + item.description)
139     print("Actual output is: " + text)
140
141     text = r.forSale
142     print("Expected output is: " + item.forSale)
143     print("Actual output is: " + text)
144

```

```

145     text = r.userSellingItem
146     print("Expected output is: " + item.userSellingItem)
147     print("Actual output is: " + text)
148
149     #Testing if an item is saved to favorites properly for user objectid 1
150     url = "http://localhost:9000/Users/addSavedItem/"
151     SavedItem savedItem = new SavedItem(1)
152     requests.post(url + 1 + "/" + savedItem)
153     requests.post(url + 2 + "/" + savedItem)
154     requests.post(url + 3 + "/" + savedItem)
155
156     url = "http://localhost:9000/Users/"
157     #Testing if the saved item exists in the User's saved item's list
158     List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
159     if not 1 in savedItems:
160         print("ERRORR: saved item is not found")
161
162     List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
163     if not 1 in savedItems:
164         print("ERRORR: saved item is not found")
165
166     List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
167     if not 1 in savedItems:
168         print("ERRORR: saved item is not found")
169
170     List<Item> savedItems = requests.get(url + 2 + "/SavedItems")
171     if not 1 in savedItems:
172         print("ERRORR: saved item is not found")
173
174     List<Item> savedItems = requests.get(url + 2 + "/SavedItems")
175     if not 1 in savedItems:
176         print("ERRORR: saved item is not found")
177     else:
178         print("PASSED THE ITEM LIKING TEST*****")
179
180     #Deleting Items test cases -----><-----><
181
182     #Deleting all the database

```

```

182     #Emptying all the database
183     url = "http://localhost:9000/Users/"
184     i = 0
185     for user in json_data:
186         i++
187         requests.delete(url + i)
188
189     #Checking if the Users table is empty
190     r = requests.post(url)
191
192     if not r is null:
193         print("Users were not deleted properly")
194     else:
195         print("Users deleted properly")
196
197     #Disliking an item
198     List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
199     requests.post(url + 1 + "/SavedItems/remove")
200
201     savedItems = requests.get(url + 1 + "/SavedItems")
202
203     if savedItems.size() != 0:
204         print("Users deleted properly")
205     else:
206         print("Passed disliking and item test")
207
208
209     #Deleting all items from the list
210     url = "http://localhost:9000/Users/"
211     i = 0
212     for user in json_data:
213         i++
214         requests.delete(url + i)
215
216     #Checking if the Users table is empty
217     r = requests.post(url)
218
219     if not r is null:
220         print("Users were not deleted properly")

```