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

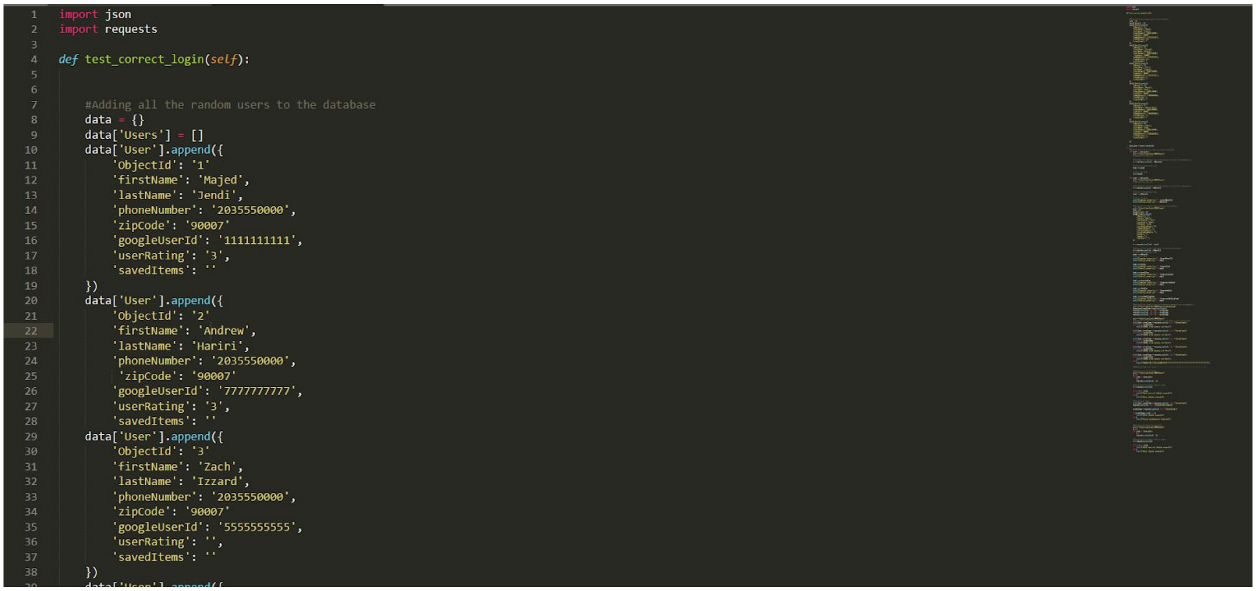
Account Page
(only accessible to signed in users):

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

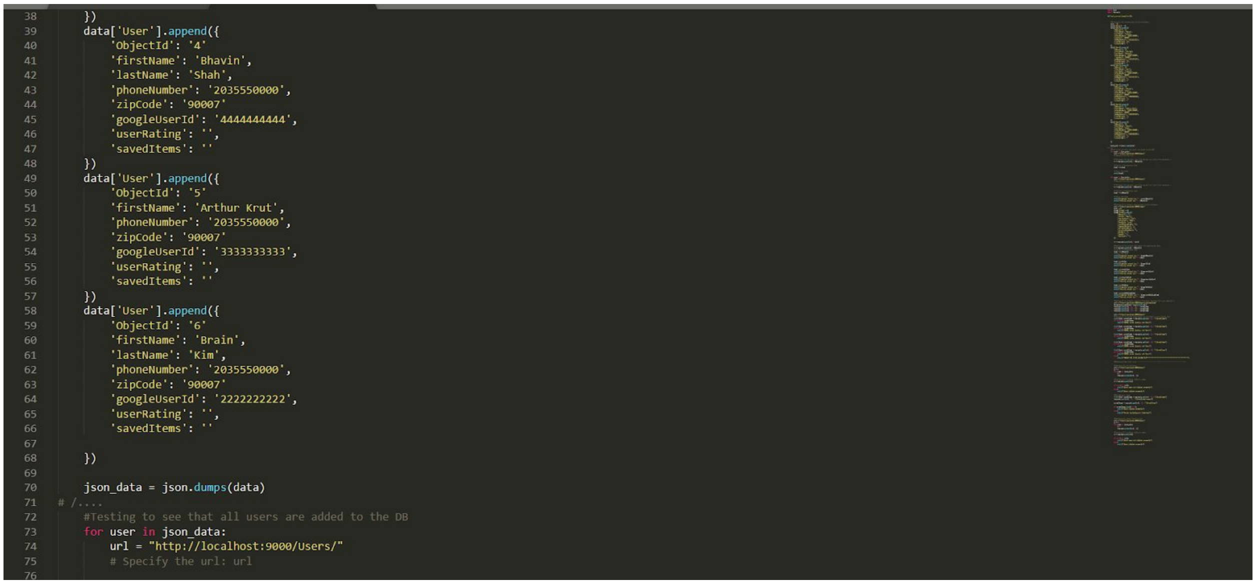
- Clicking on the add listing button will redirect to the add listing button
- Clicking on the delete button on the listed item allows the users to delete the item.
- Clicking on the reset saving button will delete all the saved items after a confirmation message

Add Listing Page

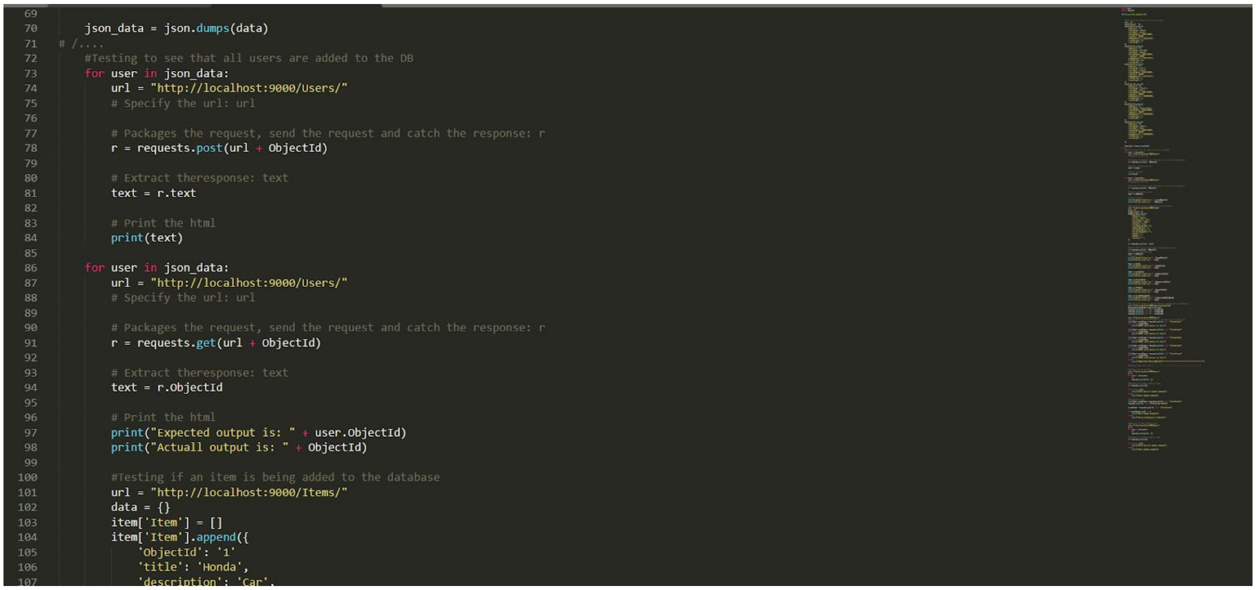
- If the user tries to submit a new listing with missing information, program will alert user to fill out missing information
- Successful submission will redirect to the account page



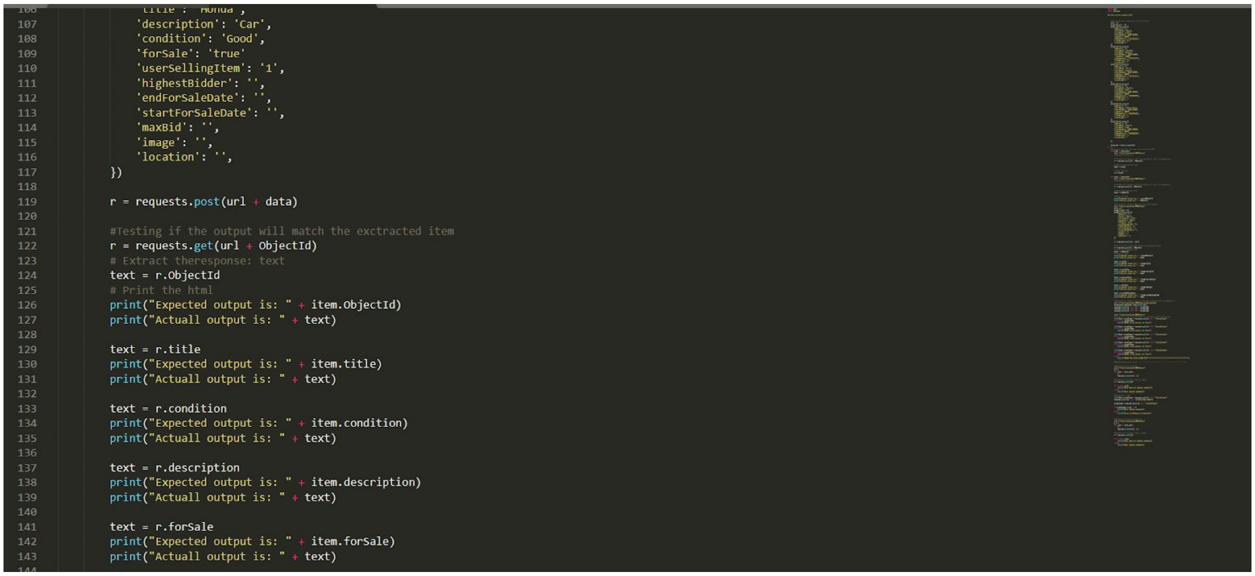
```python
import json
import requests

def test_correct_login(self):


    #Adding all the random users to the database
    data = {}
    data['Users'] = []
    data['User'].append({
        'ObjectId': '1'
        'firstName': 'Majed',
        'lastName': 'Jendi',
        'phoneNumber': '2035550000',
        'zipCode': '90007'
        'googleUserId': '1111111111',
        'userRating': '3',
        'savedItems': ''
    })
    data['User'].append({
        'ObjectId': '2'
        'firstName': 'Andrew',
        'lastName': 'Hariri',
        'phoneNumber': '2035550000',
         'zipCode': '90007'
        'googleUserId': '7777777777',
        'userRating': '3',
        'savedItems': ''
    data['User'].append({
        'ObjectId': '3'
        'firstName': 'Zach',
        'lastName': 'Izzard',
        'phoneNumber': '2035550000',
        'zipCode': '90007'
        'googleUserId': '5555555555',
        'userRating': '',
        'savedItems': ''
    })
    data['User'].append({
```



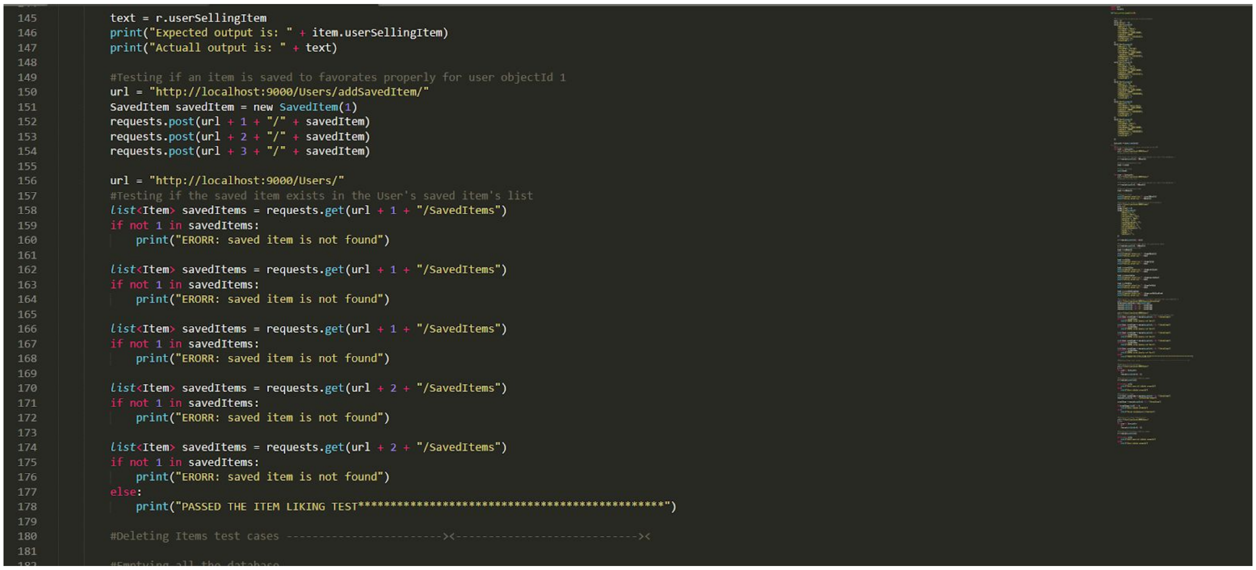
```python
    })
    data['User'].append({
        'ObjectId': '4'
        'firstName': 'Bhavin',
        'lastName': 'Shah',
        'phoneNumber': '2035550000',
        'zipCode': '90007'
        'googleUserId': '4444444444',
        'userRating': '',
        'savedItems': ''
    })
    data['User'].append({
        'ObjectId': '5'
        'firstName': 'Arthur Krut',
        'phoneNumber': '2035550000',
        'zipCode': '90007'
        'googleUserId': '3333333333',
        'userRating': '',
        'savedItems': ''
    })
    data['User'].append({
        'ObjectId': '6'
        'firstName': 'Brain',
        'lastName': 'Kim',
        'phoneNumber': '2035550000',
        'zipCode': '90007'
        'googleUserId': '2222222222',
        'userRating': '',
        'savedItems': ''

    })

    json_data = json.dumps(data)
# /....
    #Testing to see that all users are added to the DB
    for user in json_data:
        url = "http://localhost:9000/Users/"
        # Specify the url: url

```



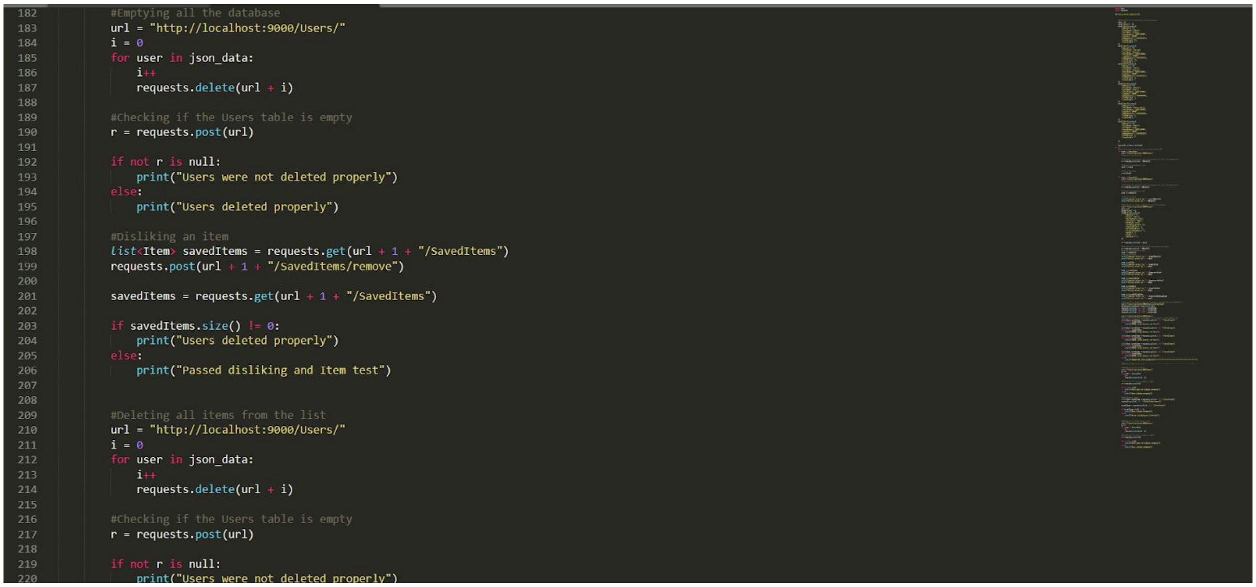
```python

    json_data = json.dumps(data)
# /....
    #Testing to see that all users are added to the DB
    for user in json_data:
        url = "http://localhost:9000/Users/"
        # Specify the url: url

        # Packages the request, send the request and catch the response: r
        r = requests.post(url + ObjectId)

        # Extract theresponse: text
        text = r.text

        # Print the html
        print(text)

    for user in json_data:
        url = "http://localhost:9000/Users/"
        # Specify the url: url

        # Packages the request, send the request and catch the response: r
        r = requests.get(url + ObjectId)

        # Extract theresponse: text
        text = r.ObjectId

        # Print the html
        print("Expected output is: " + user.ObjectId)
        print("Actuall output is: " + ObjectId)

        #Testing if an item is being added to the database
        url = "http://localhost:9000/Items/"
        data = {}
        item['Item'] = []
        item['Item'].append({
            'ObjectId': '1'
            'title': 'Honda',
            'description': 'Car',
```



```python
            'title': 'Honda',
            'description': 'Car',
            'condition': 'Good',
            'forSale': 'true'
            'userSellingItem': '1',
            'highestBidder': '',
            'endForSaleDate': '',
            'startForSaleDate': '',
            'maxBid': '',
            'image': '',
            'location': '',
        })

        r = requests.post(url + data)

        #Testing if the output will match the exctracted item
        r = requests.get(url + ObjectId)
        # Extract theresponse: text
        text = r.ObjectId
        # Print the html
        print("Expected output is: " + item.ObjectId)
        print("Actuall output is: " + text)

        text = r.title
        print("Expected output is: " + item.title)
        print("Actuall output is: " + text)

        text = r.condition
        print("Expected output is: " + item.condition)
        print("Actuall output is: " + text)

        text = r.description
        print("Expected output is: " + item.description)
        print("Actuall output is: " + text)

        text = r.forSale
        print("Expected output is: " + item.forSale)
        print("Actuall output is: " + text)
```



```
        text = r.userSellingItem
        print("Expected output is: " + item.userSellingItem)
        print("Actuall output is: " + text)

        #Testing if an item is saved to favorates properly for user objectId 1
        url = "http://localhost:9000/Users/addSavedItem/"
        SavedItem savedItem = new SavedItem(1)
        requests.post(url + 1 + "/" + savedItem)
        requests.post(url + 2 + "/" + savedItem)
        requests.post(url + 3 + "/" + savedItem)

        url = "http://localhost:9000/Users/"
        #Testing if the saved item exists in the User's saved item's list
        List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
        if not 1 in savedItems:
            print("ERORR: saved item is not found")

        List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
        if not 1 in savedItems:
            print("ERORR: saved item is not found")

        List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
        if not 1 in savedItems:
            print("ERORR: saved item is not found")

        List<Item> savedItems = requests.get(url + 2 + "/SavedItems")
        if not 1 in savedItems:
            print("ERORR: saved item is not found")

        List<Item> savedItems = requests.get(url + 2 + "/SavedItems")
        if not 1 in savedItems:
            print("ERORR: saved item is not found")
        else:
            print("PASSED THE ITEM LIKING TEST**********************************************")

        #Deleting Items test cases ------------------------><--------------------------><

        #Emptying all the database
```



```
        #Emptying all the database
        url = "http://localhost:9000/Users/"
        i = 0
        for user in json_data:
            i++
            requests.delete(url + i)

        #Checking if the Users table is empty
        r = requests.post(url)

        if not r is null:
            print("Users were not deleted properly")
        else:
            print("Users deleted properly")

        #Disliking an item
        List<Item> savedItems = requests.get(url + 1 + "/SavedItems")
        requests.post(url + 1 + "/SavedItems/remove")

        savedItems = requests.get(url + 1 + "/SavedItems")

        if savedItems.size() != 0:
            print("Users deleted properly")
        else:
            print("Passed disliking and Item test")


        #Deleting all items from the list
        url = "http://localhost:9000/Users/"
        i = 0
        for user in json_data:
            i++
            requests.delete(url + i)

        #Checking if the Users table is empty
        r = requests.post(url)

        if not r is null:
            print("Users were not deleted properly")
```