SOFTWARE ENGINEERING (CEN5035)

GatorX

(A web application to view and review places)

Backend API documentation

Developers:

APIs:

- 1. User Management (Register, login, etc.)
- 2. View/Add places
- 3. View/Post/Edit Reviews

User Management API:

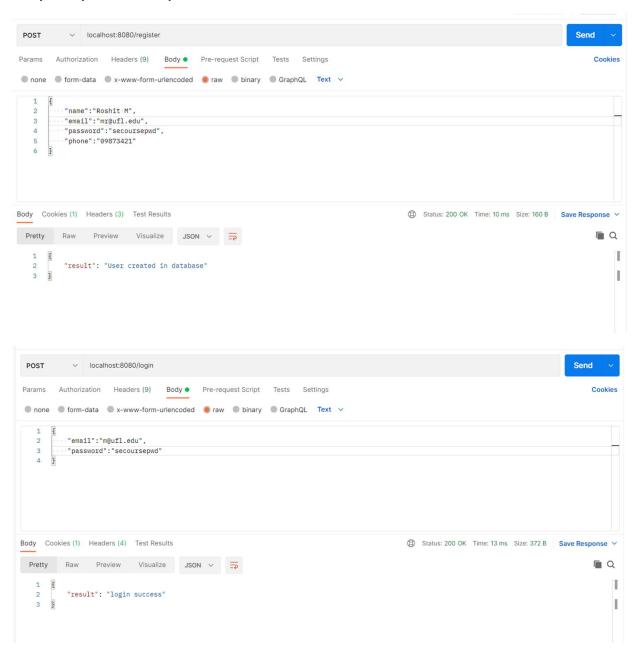
Description:

- This API allows the user to register by entering their credentials and getting added to the database.
- Once the user is registered, they can login into the website whenever, using their email address and password
- Registering and logging in is mandatory for any user trying to post any new reviews for the visible locations on the site

Acceptance Criteria:

- For registering, the following fields are required to be entered by the user: *Name, Email, Phone and Password*.
- If any/all of these fields weren't entered, the API throws an error message.
- Once all the inputs are properly entered by the user, the API throws a confirmation message:
 "User added to the database"
- If the same user tries registering again using the same credentials, the API throws the message: "User already exists"
- Similarly for login, the API should throw an error message, if the user entered either a wrong email or password.

Sample Requests and Responses:



API for Viewing/Adding places:

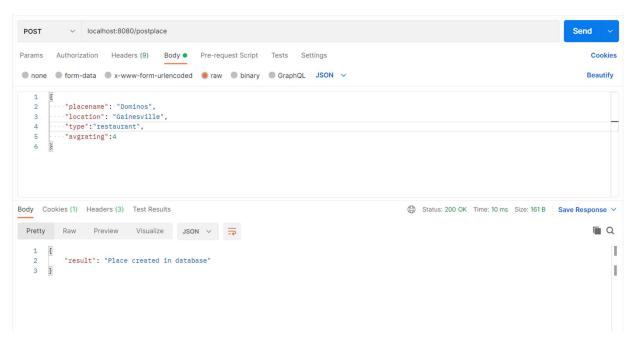
Description:

- This API sends a get request whenever the user wishes to view all the places available to review.
- A post request is sent whenever the user wishes to add a new place to the database. The
 following are the required parameters for sending a post request to add a new place:
 placename, location, type, avgrating.

Acceptance Criteria:

- The API should throw an error message whenever any of the above mentioned required parameters aren't specified.
- Whenever a get request is sent the API should send a response with the list of places along with the values of all the other place parameters.

Sample Requests and Responses:



```
∨ localhost:8080/getallplaces
       Authorization Headers (6) Body Pre-request Script Tests Settings
Body Cookies Headers (3) Test Results
                                                                                                       Status: 200 OK Time: 3 ms Size: 724 B Save Response V
  Pretty Raw Preview Visualize JSON V
            "result": [
                     "ID": 1,
                    "CreatedAt": "2022-02-27T12:06:15.2494023-05:00",
                    "UpdatedAt": "2022-02-27T12:06:15.2494023-05:00",
                    "DeletedAt": null,
                    "placename": "Chikfila",
"location": "The Hub, UF",
                    "type": "food",
   10
                    "avgrating": 3
   13
   14
                    "ID": 2.
                     "CreatedAt": "2022-03-03T12:31:47.3128113-05:00",
  15
                     "UpdatedAt": "2022-03-03T12:31:47.3128113-05:00",
                    "DeletedAt": null,
                    "placename": "McD",
"location": "Gainesville",
   18
  19
                     "type": "food",
  20
                     "avgrating": 4
  21
   22
   24
                    "ID": 3,
```

API for Viewing, Posting or Editing reviews:

Description:

- Logged in users can add a review to a corresponding place. The API receives a POST request with the place id of the corresponding place.
- Reviews from other users for all the places can be viewed
- Previously posted reviews can be edited.

Acceptance Criteria:

- Whenever the user is not logged in and tries to write a review, the API should throw an error message saying: "User not logged in"
- The API should throw an error message when any of the following required fields are not filled, when trying to post a review: review: review, rating, placeid, reviewerid
- The review id should be included in the URL when trying to edit an old review.
- Whenever a get request is sent, the API should send data having all the previously posted reviews that are stored on the database.
- When a delete review request is sent to the API along with the review id, the API should remove the corresponding review from the database.

Sample Requests and Responses:

