Backend Documentation

Filename: app.py

Functions -

1. signup()
   Route: /signup
   Method: POST

   There are 5 parameters in the JSON sent to this API - first name (**firstName**), last name (**lastName**), email (**email**), contact (**contact**) and password (**password**).

   Response -
   a. If there already exists a row in the users table with the same email address, the message returned is “An account with this email already exists”.
   b. If there already exists a row in the users table with the same contact information, the message returned is “An account with this contact already exists”.
   c. In the case when neither of the above two hold, a new row is created in the users table with the information from the JSON and the message returned is “Added successfully”.

2. login()
   Route: /login
   Method: POST

   There are 2 parameters in the JSON sent to this API - email (**email**) and password (**password**).

   Response -
   a. If the email does not exist in the users table, the message returned is “Please create an account!”.
   b. If the email exists, but the password entered is wrong, the message returned is “Invalid credentials!”.
   c. If the email and password pair is correct, the message returned is “Logged in successfully”.

3. create_bid()
   Route: /bid/create
   Method: POST

   There are 3 parameters in the JSON sent to this API - productId (**prodId**), email (**email**) and amount (**bidAmount**).

From the product table, we find the minimum amount of the bid that the seller had mentioned.
Response -

a. If the bid amount is less than the minimum amount, the message returned is "Amount is lesser than initial price".
b. If the bid amount is greater than the minimum amount, the bid is created/updated in the bids table and the message returned is "Saved bid".

4. create_product()
Route: /create/product
Method: POST

   There are 6 parameters in the JSON sent to this API - productName (**productName**), sellerEmail (**sellerEmail**), initialPrice (**initialPrice**), increment (**increment**), photo (**photo**) and description (**description**).

   Response -
   This API creates a new product in the product table with the aforementioned details and returns the message "Added product successfully".

5. get_all_products()
Route: /product/listAll
Method: GET

   Response -
   It returns the details of all the products that are present in the product table.

6. get_product_image()
Route: /product/getImage
Method: POST

   There is 1 parameter in the JSON sent to this API - productId (**productID**).

   Response -
   It gets the image corresponding to the productId and returns it in the response.

7. get_product_details()
8. update_product_details()
9. get_landing_page()