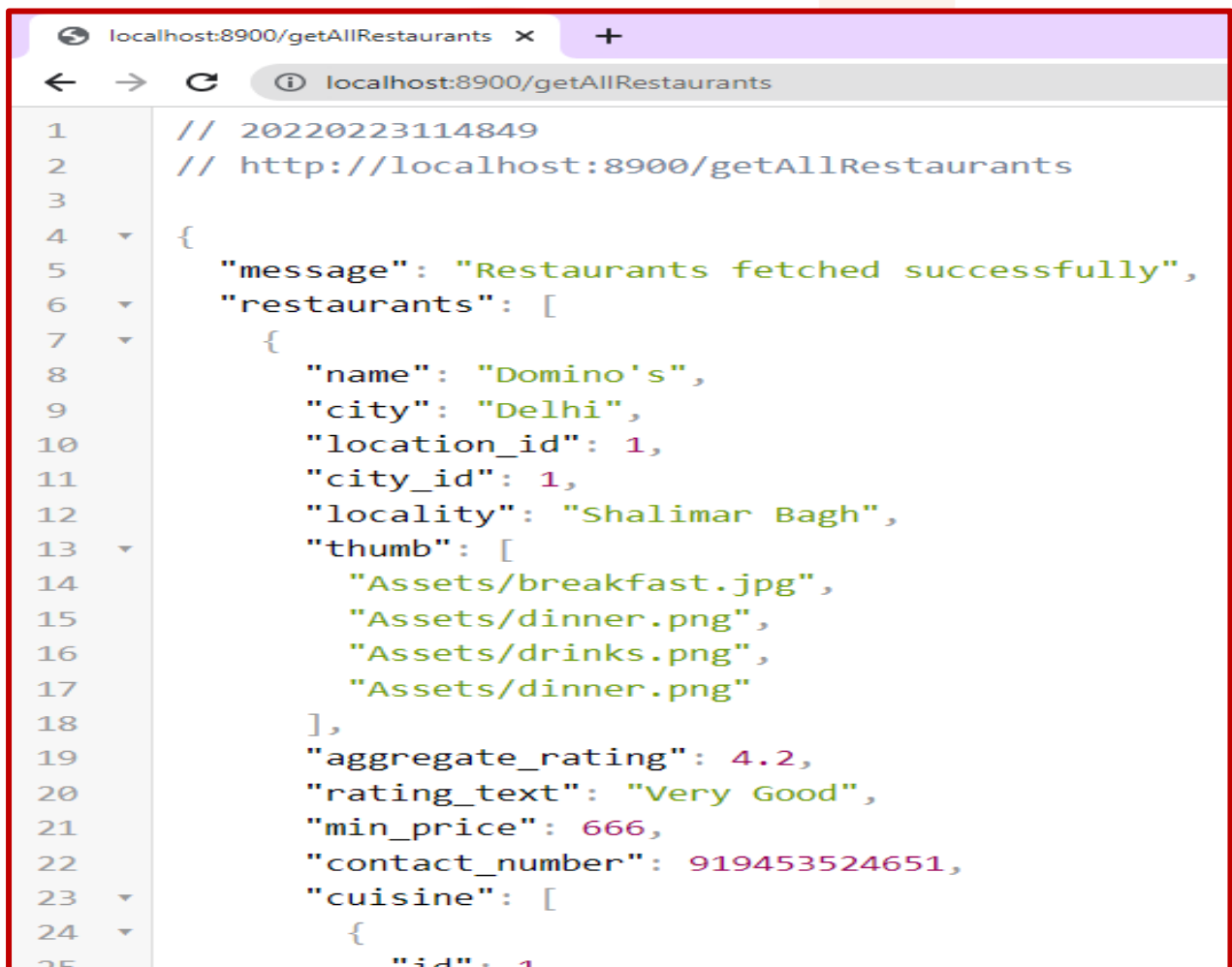


Day 62 Task

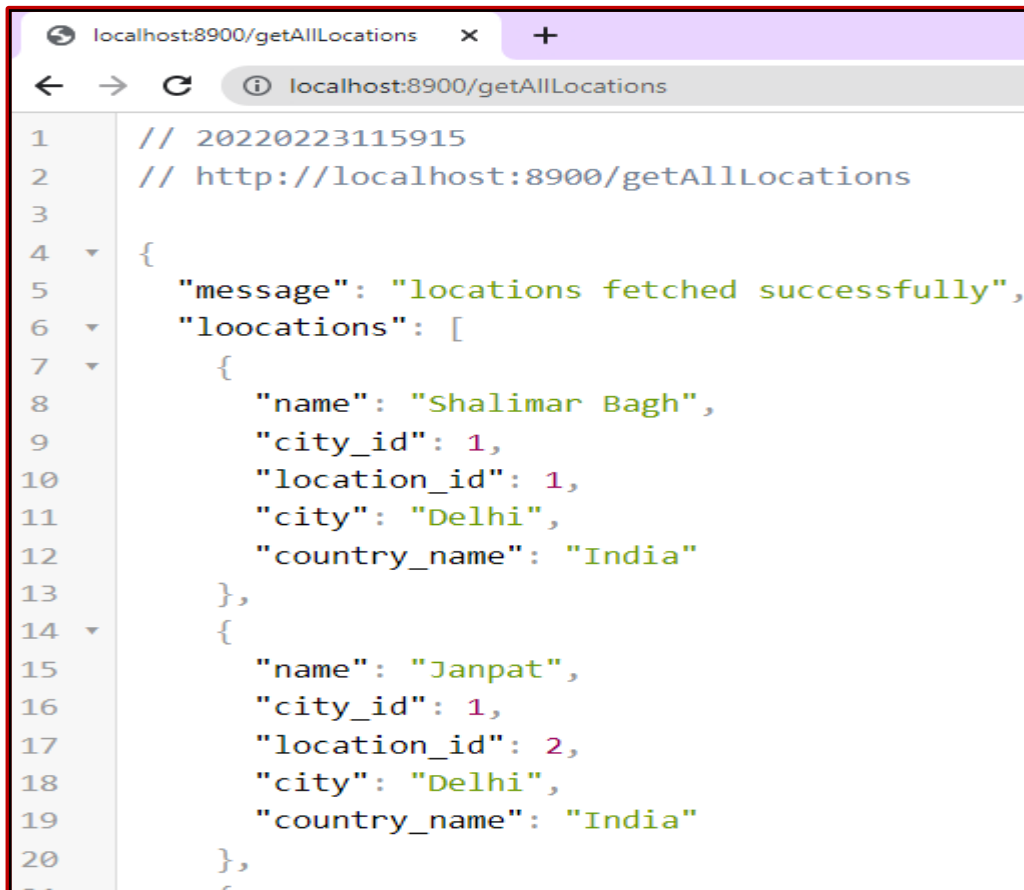
APIs for Home page and Filter page

1. Fetch data of all restaurants from MongoDB cloud database using Express and implement it in MVC architecture.
The expected output is shown below.



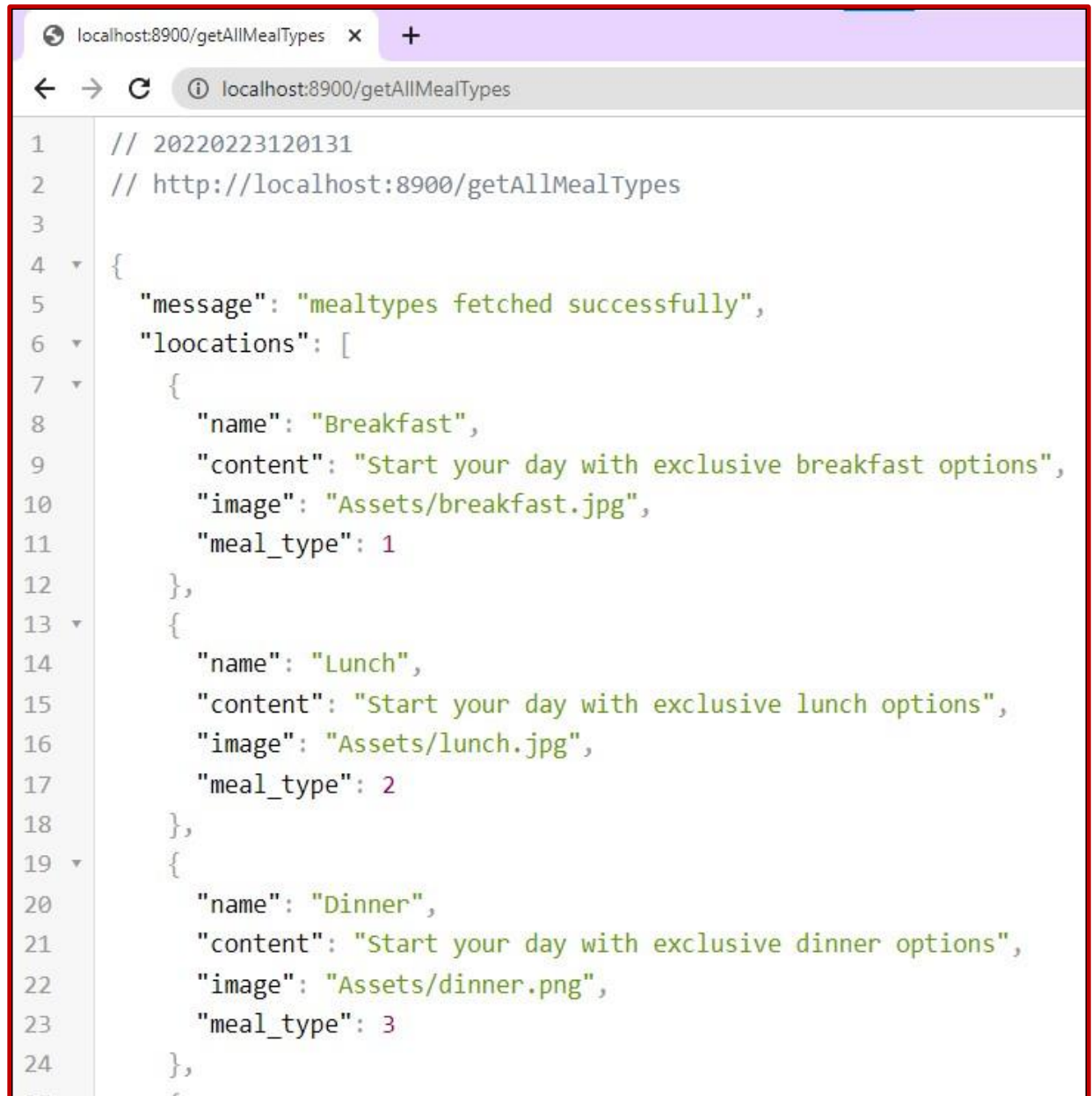
```
1 // 20220223114849
2 // http://localhost:8900/getAllRestaurants
3
4 {
5   "message": "Restaurants fetched successfully",
6   "restaurants": [
7     {
8       "name": "Domino's",
9       "city": "Delhi",
10      "location_id": 1,
11      "city_id": 1,
12      "locality": "Shalimar Bagh",
13      "thumb": [
14        "Assets/breakfast.jpg",
15        "Assets/dinner.png",
16        "Assets/drinks.png",
17        "Assets/dinner.png"
18      ],
19      "aggregate_rating": 4.2,
20      "rating_text": "Very Good",
21      "min_price": 666,
22      "contact_number": 919453524651,
23      "cuisine": [
24        {
25          "id": 1
```

2. Fetch data of all locations from MongoDB cloud database using Express and implement it in MVC architecture.
The expected output is shown below.



```
1 // 20220223115915
2 // http://localhost:8900/getAllLocations
3
4 {
5   "message": "locations fetched successfully",
6   "locations": [
7     {
8       "name": "Shalimar Bagh",
9       "city_id": 1,
10      "location_id": 1,
11      "city": "Delhi",
12      "country_name": "India"
13    },
14    {
15      "name": "Janpat",
16      "city_id": 1,
17      "location_id": 2,
18      "city": "Delhi",
19      "country_name": "India"
20    },
21  ]
22 }
```

3. Fetch data of all meal types from MongoDB Cloud instance using Express and implement it in MVC architecture.
The expected output is shown below.



```
1 // 20220223120131
2 // http://localhost:8900/getAllMealTypes
3
4 {
5   "message": "mealtypes fetched successfully",
6   "loocations": [
7     {
8       "name": "Breakfast",
9       "content": "Start your day with exclusive breakfast options",
10      "image": "Assets/breakfast.jpg",
11      "meal_type": 1
12    },
13    {
14      "name": "Lunch",
15      "content": "Start your day with exclusive lunch options",
16      "image": "Assets/lunch.jpg",
17      "meal_type": 2
18    },
19    {
20      "name": "Dinner",
21      "content": "Start your day with exclusive dinner options",
22      "image": "Assets/dinner.png",
23      "meal_type": 3
24    },
25  ]
26 }
```

4. Parameterized APIs: Create an API to filter the restaurants on city param and restaurant id param.

The expected output is shown below



```
1 // 20220223121034
2 // http://localhost:8900/getRestaurantsByLocation/Delhi
3
4 {
5   "message": "restaurants found",
6   "restaurants": [
7     {
8       "name": "Domino's",
9       "city": "Delhi",
10      "location_id": 1,
11      "city_id": 1,
12      "locality": "Shalimar Bagh",
13      "thumb": [
14        "Assets/breakfast.jpg",
15        "Assets/dinner.png",
16        "Assets/drinks.png",
17        "Assets/dinner.png"
18      ],
19      "aggregate_rating": 4.2,
20      "rating_text": "Very Good",
21      "min_price": 666,
22      "contact_number": 919453524651,
23      "cuisine": [
24        {
25          "id": 1,
```

5. Create an API to fetch the restaurants based on meal type, location, cuisine, low cost, high cost, sort, and the page number.

6. Create an API for user Log on and Sign in functionalities?