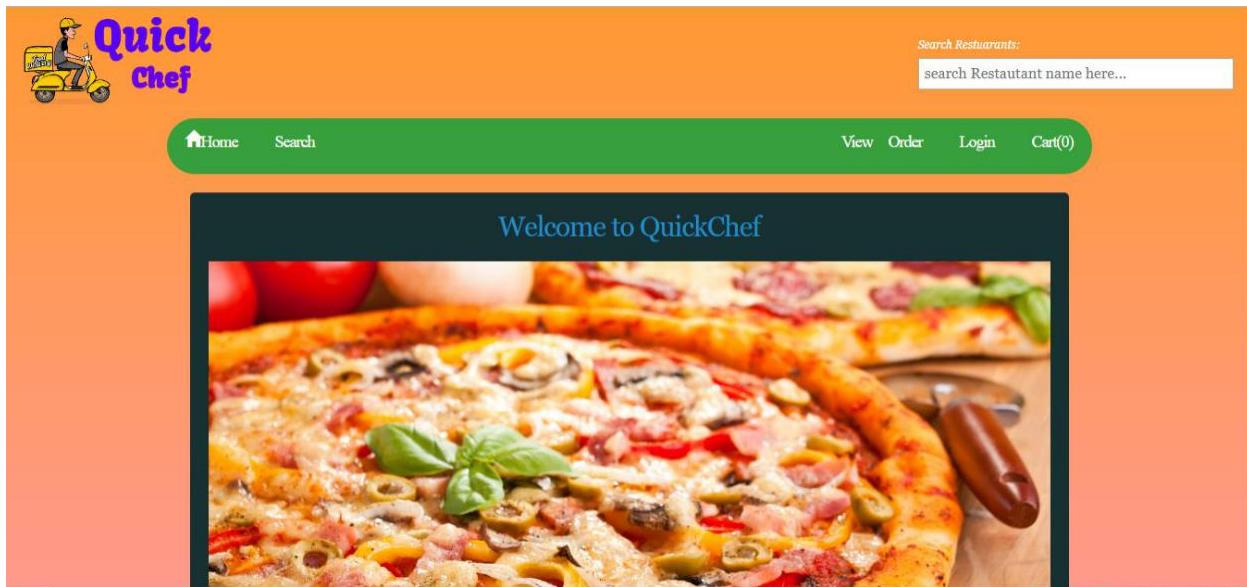
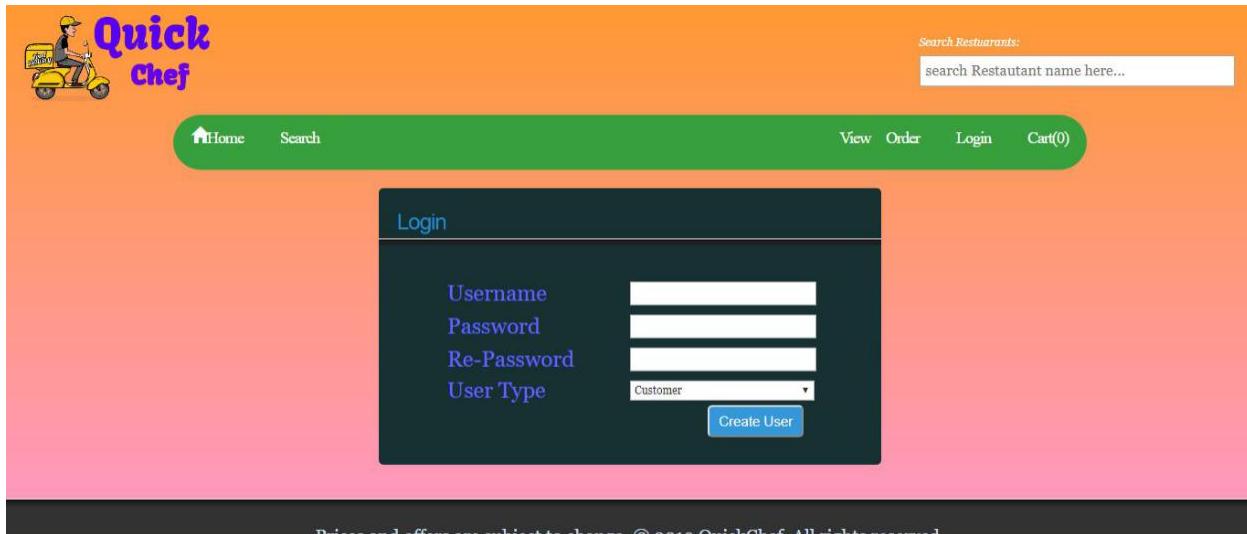


Final Project Snapshots



Home Page with a search tab(with autocomplete feature) with which we can search restaurants by its name. Clicking the search button on the menu tab, it redirects to the list of restaurants. View button is to view the orders for a particular user. Cart displays the items that the user has added for ordering.



Login and Registration Screen. Existing users can login to the application and new user can register using this portal.



**Quick
Chef**

[Home](#)

[Search](#)

[ViewOrder](#)

Hello,Jai

Account

Address Search

[Submit](#)

Search Restaurants:

A

Au Cheval

Aloha Wagon Cart(0)

ArePA George

Ali Baba Doner

Aloha Eats

Resturants

Seoul Taco
738 N Clark St Chicago, IL
60654\$

4.0



[Buy From this Restaurant](#)

Farm & Sea Turkish Cuisine
2833 N Broadway Chicago, IL
60657\$

5.0



[Buy From this Restaurant](#)

Papa's Cache Sabroso
2517 W Division St Chicago, IL
60622\$

4.0



[Buy From this Restaurant](#)

SoJu BBQ
36 S Ashland Chicago, IL
60607\$

4.5



[Buy From this Restaurant](#)

Au Cheval
800 W Randolph St Chicago, IL
60607\$

4.5



[Buy From this Restaurant](#)

Cracked on Milwaukee
1359 N Milwaukee Ave Chicago,
IL 60622\$

4.5



[Buy From this Restaurant](#)



Landbirds
2532 N California Ave Chicago,
IL 60647\$

4.5



Taco Loco Of Pilsen
2022 S Leavitt St Chicago, IL
60608\$

5.0



Small Cheval
1732 N Milwaukee Ave Chicago,
IL 60647\$

4.5

Search Page. On click of the search button, this page is displayed which shows the list of the restaurants. There is also another search bar where user can search the restaurants based on the address of the restaurant. All the restaurants, their images, names, addresses and their ratings are being displayed. Users have to select the restaurant that they need to order by clicking on 'Buy From this Restaurant' button.



Search Restaurants:

search Restautant name here...

Home

Search

View Order

Login

Cart(0)

Seoul Taco Menu

Fried Chicken

20.99\$



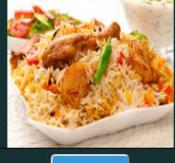
North Indian Curry

19.99\$



Biryani

299.99\$



Buy Now

WriteReview

ViewReview

Buy Now

WriteReview

ViewReview

Buy Now

WriteReview

ViewReview

Steak

25.0\$



Chicago Hot Dog

9.99\$



Italian beef

27.99\$



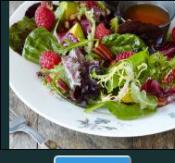
Dosa

299.99\$



Salada

29.99\$



Milk Shakes

15.0\$



Buy Now

WriteReview

ViewReview

Buy Now

WriteReview

ViewReview

Buy Now

WriteReview

ViewReview

Buy Now

WriteReview

ViewReview

Menu Page. On click of ‘Buy From this Restaurant’ button, it is redirected to the menu page where in it displays all the menu available in that restaurant. Users can add the food from the list and can proceed to check out. Users can write reviews and view the reviews for a particular dish.

Cart(1)

1.	Fried Chicken	: 20.99
Total		20.99
		CheckOut

Cart displays all the items that are added by the user along with their total price.

Order

Customer Name:	jai
Restaurant name:	Seoul Tacos
Restaurant Address:	738 N Clark St Chicago, IL 60654
Product Purchased:	Fried Chicken
Product Price:	20.99
Total Order Cost:	20.99

Submit

738 N Clark St
738 N Clark St, Chicago, IL 60654
Gretna Kane ansehen
Routenplanen

738 North Clark Street
738 North Clark Street, Chicago, IL 60654
Route zu diesem Ziel
Greystone Global
Midcity Parking Garage
A Plus Chicago Movers
Merchant Enterprises
Harris Beach
33 W Chicago Ave Garage
Brazilian Steakhouse
ZEDAST

Google Kartendaten © 2019 Google. Nutzungsbedingungen. Partner bei Google Maps mitarbeiten

CheckOut. Order Summary Page displays the order details and along with it, it displays the restaurant's location on the map it use the *google maps API*. Users need to enter their valid card details and address to place an order.



QuickChef

Search Restaurants:
search Restaurant name here...

Home Search ViewOrder HelloJai Account Logout Cart(1)

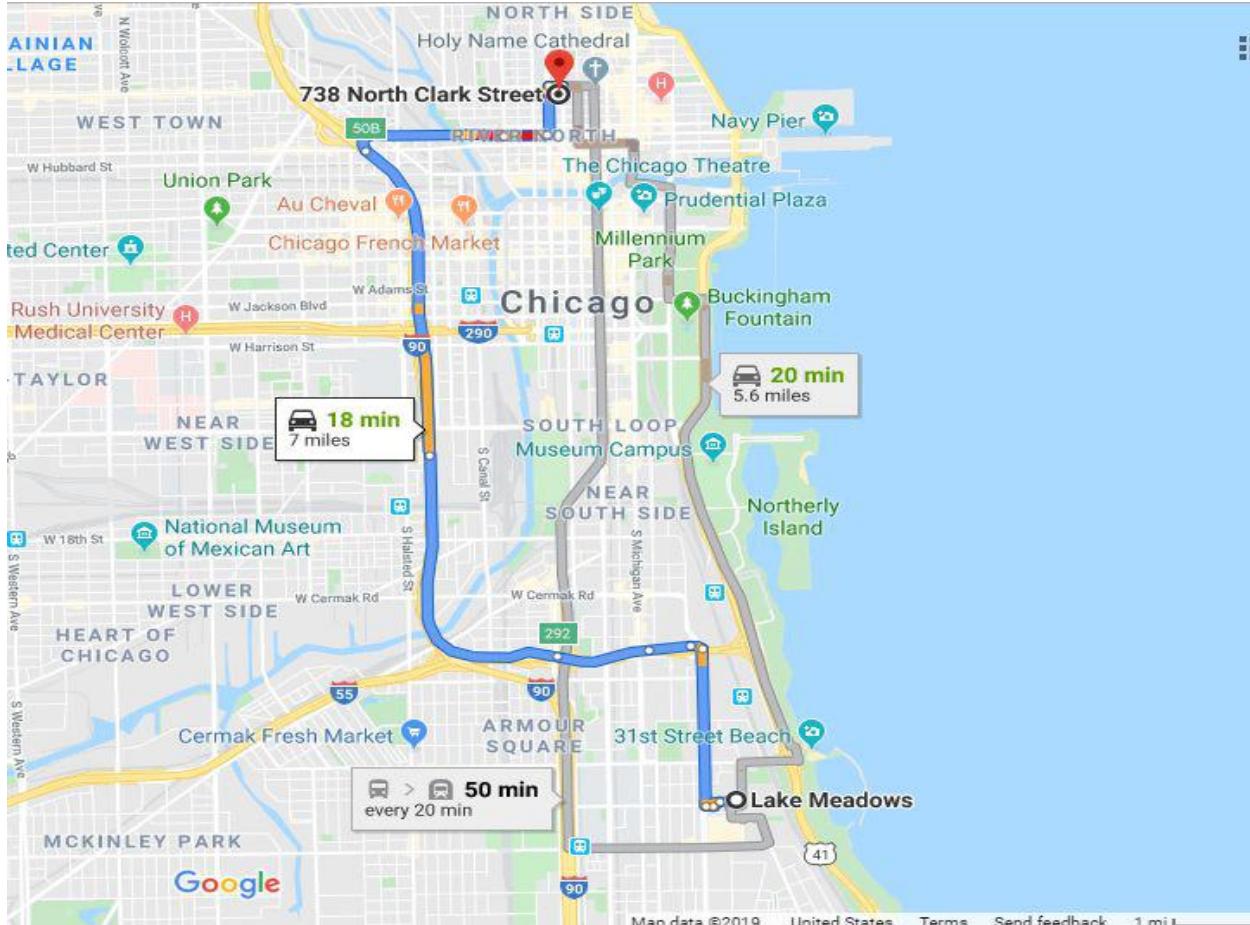
Order

Your Order is stored
Your Order No is 6
Your order will be delivered in **20 mins**

Locate your delivery on Maps

Prices and offers are subject to change. © 2019 QuickChef. All rights reserved.

Users will be displayed with their order number and along with that, users can know the approximate delivery time of their order in this window. It uses **google distance matrix API for prediction**. Users can locate the distance and approximate time of arrival in a map view by clicking on ‘Locate your delivery on Maps’ button.



A map view of the direction, distance and approximate time of arrival from the restaurant to the user's location.

Account

User Name:	jai			
User Type:	customer			
OrderId:	UserName:	productOrdered:	productPrice:	
1.	jai	Fish Fry	Price: 99.99	CancelOrder
2.	jai	North Indian Curry	Price: 19.99	CancelOrder
3.	jai	Biryani	Price: 299.99	CancelOrder
4.	jai	Dosa	Price: 299.99	CancelOrder
5.	jai	North Indian Curry	Price: 19.99	CancelOrder

Users can cancel the order if they wish to, and can place a new order.

Welcome to QuickChef

This window is for the **store manager**. He can add, delete, and update order upon user's request

Order

OrderId:	UserName:	productOrdered:	productPrice:	
2.	jai	North Indian Curry	Price: 19.99	CancelOrder

Admin can cancel the order placed by the user.



Quick Chef

Search Restaurants:
search Restautant name here...

Home Search ViewOrder Hello,Jai Account Logout Cart(0)

Review

Product Name:	chicken
Restaurants	Seoul Taco
Product Price:	20.99
Review Rating:	4 <input type="button" value="▼"/>
Retailer Zip Code:	60616
Retailer City:	Chicago
Review Date:	24-Nov-2019
Review Text:	Good!!

Users can rate the food in that restaurant and **write reviews** about it .



Quick Chef

Search Restaurants:
search Restautant name here...

Home Search ViewOrder Hello,Jai Account Logout Cart(0)

Review

Product Name:	chicken
userNmae:	jai
price:	20
Retailer City:	Chicago
Review Rating:	4
Review Date:	2019-11-24
Review Text:	Good!!

Prices and offers are subject to change. © 2019 QuickChef. All rights reserved.

Review written by the user is stored in MongoDB(Nosql database)

```
{
  "_id": ObjectId("5dd88289b8c19f280080a1669"),
  "title": "myReviews",
  "userName": "jk",
  "retailername": "Jai",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "Yes",
  "userid": "3454634",
  "userage": "26",
  "usergender": "Male",
  "useroccupation": "Master",
  "productName": "iPhone7",
  "producttype": "games",
  "productMaker": "apple",
  "reviewRating": 5,
  "reviewDate": "2019-11-24",
  "reviewText": "Very good!",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 42
},
{
  "_id": ObjectId("5dd8832cb8c19f280080a166c"),
  "title": "myReviews",
  "userName": "kj",
  "retailername": "bhata",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "Yes",
  "userid": "1234",
  "userage": "24",
  "usergender": "Male",
  "useroccupation": "Pro",
  "productName": "LG2",
  "producttype": "televisions",
  "productMaker": "lg",
  "reviewRating": 5,
  "reviewDate": "2019-11-21",
  "reviewText": "GOOD",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 139
},
{
  "_id": ObjectId("5dd89da7b8c19f280080a1670"),
  "title": "myReviews",
  "userName": "jai",
  "retailername": "Jaikeerthi",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "Yes",
  "userid": "13898",
  "userage": "29",
  "usergender": "Male",
  "useroccupation": "Govt Employee",
  "productName": "att",
  "producttype": "wirelessplans",
  "productMaker": "Basic",
  "reviewRating": 5,
  "reviewDate": "2019-11-22",
  "reviewText": "Good",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 50
},
{
  "_id": ObjectId("5dd8a004ab8c19f280080a1674"),
  "title": "myReviews",
  "userName": "jai",
  "retailername": "Jaikeerthi",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "Yes",
  "userid": "1234",
  "userage": "23",
  "usergender": "Male",
  "useroccupation": "Business Man",
  "productName": "Microsoft",
  "productType": "soundsystems",
  "productMaker": "Microsoft",
  "reviewRating": 5,
  "reviewDate": "2019-11-22",
  "reviewText": "Good",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 238
},
{
  "_id": ObjectId("5dd8a395b8c19f280080a1679"),
  "title": "myReviews",
  "userName": "jai",
  "retailername": "Jaikeerthi",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "Yes",
  "userid": "3454634",
  "userage": "23",
  "usergender": "Male",
  "useroccupation": "Business Man",
  "productName": "att",
  "producttype": "wirelessplans",
  "productMaker": "Basic",
  "reviewRating": 2,
  "reviewDate": "2019-11-13",
  "reviewText": "Good",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 50
},
{
  "_id": ObjectId("5dd8a4a0ab8c19f280080a167d"),
  "title": "myReviews",
  "userName": "john",
  "retailername": "Jaikeerthi",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "yes",
  "userid": "23432543",
  "userage": "22",
  "usergender": "Male",
  "useroccupation": "Bussines",
  "productName": "Sharp3",
  "producttype": "televisions",
  "productMaker": "sharp",
  "reviewRating": 4,
  "reviewDate": "2019-11-06",
  "reviewText": "Hi",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 169
},
{
  "_id": ObjectId("5dd8a4a0ab8c19f280080a1680"),
  "title": "myReviews",
  "userName": "john",
  "retailername": "Jaikeerthi",
  "retailerstate": "IL",
  "productsale": "Yes",
  "manufacturerebate": "Yes",
  "userid": "1234",
  "userage": "23",
  "usergender": "Male",
  "useroccupation": "Business Man",
  "productName": "HuaweiP20",
  "productType": "games",
  "productMaker": "huawei",
  "reviewRating": 4,
  "reviewDate": "2019-11-22",
  "reviewText": "Hi",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 50
},
{
  "_id": ObjectId("5dd9edb6b8c19f0cb4c4e4b8"),
  "title": "myReviews",
  "userName": "jai",
  "productName": "northindiancurry1",
  "productType": "tablets",
  "productMaker": null,
  "reviewRating": 2,
  "reviewDate": "2019-11-16",
  "reviewText": "Good",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 19
},
{
  "_id": ObjectId("5db1dc4d8c19f04189003f1"),
  "title": "myReviews",
  "userName": "jai",
  "productName": "chicken",
  "producttype": "tablets",
  "productMaker": null,
  "reviewRating": 4,
  "reviewDate": "2019-11-24",
  "reviewText": "Good!!",
  "retailerpin": "60616",
  "retailercity": "Chicago",
  "price": 20
}
>
}
```

MongoDB console which is used for storing reviews.

API's Used:

The screenshot shows the Yelp Fusion API documentation for the `/businesses/search` endpoint. The main content area includes the endpoint URL, parameters (term, location, latitude, longitude), and a note about returning businesses with reviews. A sidebar on the right shows a user profile for "Jaikeerthi S." located in Chicago, IL, with options for About Me, Find Friends, Cash Back NEW, and Account Settings. The bottom left sidebar lists other API endpoints like General, Fusion API, GraphQL, Manage App, and specific sections for Yelp Fusion such as Introduction, Business Endpoints, Business Search, Phone Search, Transaction Search, Business Details, and Business Match.

Yelp API developers account and Yelp business search API.

```

1 import requests
2 import json
3 import mysql.connector
4 mydb = mysql.connector.connect(
5   host="localhost",
6   user="root",
7   passwd="root",
8   database="Quickchef"
9 )
10 mycursor = mydb.cursor()
11
12 business_id = "Bu_MSebhXavCbdYUk6oasQ"
13
14 API_KEY = "c4MwC9ckw-B9U9K7g6LR92e38015_1NQx5cNDFcmDqFGfqgUMSo/eaB8dipIe7RL616efHnqXHcEJPf5Rx7K5M9nesISy@35qIF0jX0801EMgYaNjEX00y"
15 ENDPOINT = "https://api.yelp.com/v3/businesses/search"
16 HEADERS = {"Authorization": "bearer " + API_KEY}
17
18 PARAMETERS = {"term": "Food",
19               "location": "USA"}
20
21 response = requests.get(url = ENDPOINT,
22                         params = PARAMETERS,
23                         headers = HEADERS)
24
25 business_data = response.json()
26 #print(business_data)
27 for i in business_data["businesses"]:
28     address = " ".join(i["location"]["display_address"])
29     sql = "INSERT INTO restaurants (id, name, image_url, rating, latitude, longitude, address) VALUES (%s, %s, %s, %s, %s, %s, %s)"
30     val = (i["id"], i["name"], i["image_url"], i["rating"], i["coordinates"]["latitude"], i["coordinates"]["longitude"], i["address"])
31     mycursor.execute(sql, val)
32     mydb.commit()
33     print(mycursor.rowcount, "Record inserted.")
34
35
36 # print ("Name: ", i["name"])
37 # print ("Image URL: ", i["image_url"])
38 # print ("Rating: ", i["rating"])
39 # print ("Coordinates-latitude: ", i["coordinates"]["latitude"])
40

```

Python script that fires the Yelp API to fetch the restaurant details and store it into the MySQL database

	id	name	image_url	rating	latitude	longitude	address
1	0eMo-MeawM-tmbwpzv3rDg	Carritas Uruapan	BL08	4.5	41.857612801225	-87.669703625051	1725 W 18th St Chicago, IL 60608
2	4oATsYP0OF0cdHz-_c_lw	1492 Cuban Fusion	BL08	4.5	41.909976	-87.692779	2615 West North Ave Chicago, IL 60647
3	t2nXebM7aqTj_qq6saZ89g	Eataly - Chicago	BL08	4.0	41.892335432963	-87.6262607945148	43 E Ohio St Chicago, IL 60611
4	IPRSekE7gBLoGo6NMhOlLQ	The Crack Shack - Little ...	BL08	4.0	32.72776	-117.1713	2266 Kettner Blvd San Diego, CA 92101
5	OfjuGieDXyPB8jWa0pvehA	Homestyle Hawaiian	BL08	4.0	32.801671968633	-117.159088104705	7524 Mesa College Dr San Diego, CA 92111

View of MySQL database that stores the restaurants data that is fetched from the Yelp API.

The screenshot shows the Google Cloud Platform API & Services dashboard under the 'Credentials' tab. A single API key named 'API key 1' was created on November 18, 2019, with no restrictions and the key value AlzaSyC8PidiVrq-TVf9GMvnPhj1u7vD-D-NUw.

The screenshot shows the Google Maps Platform Distance Matrix API documentation. It includes sections for 'Sample request and response', 'Try it!', and a code snippet for Java demonstrating how to make a request to the API.

```

try{
    address = address.replaceAll("\s", "+");
    String url = "https://maps.googleapis.com/maps/api/distancematrix/json?origins=" + myaddress + "&destinations=" + add;
    URL obj = new URL(url);
    HttpURLConnection con = (HttpURLConnection) obj.openConnection();
    con.setRequestMethod("GET");

    con.setRequestProperty("User-Agent", "Mozilla/5.0");
    int responseCode = con.getResponseCode();

    BufferedReader in = new BufferedReader(
        new InputStreamReader(con.getInputStream()));
    String inputLine;
    StringBuffer response1 = new StringBuffer();
    while ((inputLine = in.readLine()) != null) {
        response1.append(inputLine);
    }
    in.close();

    JSONObject myResponse = new JSONObject(response1.toString());
}

```

Google Distance Matrix API is used to predict the time of arrival from the restaurant to the user's location

The screenshot shows the Google Maps Platform Documentation page for the Directions API. The top navigation bar includes links for Overview, Products, Pricing, Documentation (selected), and Blog. A search bar and language selection are also present. On the left, a sidebar menu has 'Get Started' selected, showing links for Developer Guide, Get an API Key, Web Services, Best Practices, Client Libraries, Policies and Terms, Usage and Billing, Policies, Terms of Service, and Other Web Service APIs. The main content area discusses the Directions API's HTTP interface and provides a sample request URL: <https://maps.googleapis.com/maps/api/directions/json?origin=Disneyland&destination=Universal+Studio>. It also includes a 'Try it!' section, developer notes, and a sample response JSON. A right sidebar contains links for Contents, Sample request and response, Start coding with our client libraries, Authentication, quotas, pricing, and policies, Activate the API and get an API key, Quotas and pricing, Policies, and Learn more.

Google Maps API is used to provide map view of users and restaurant's location, distance and approximate time of arrival