1. INTRODUCTION

The API consists the following functionalities-

1. Authenticating User using Google Authentication
2. Determining User Location
3. Finding online Users and Sorting them according to their distance using Google Distance Matrix API
4. Suggesting nearby users according to category

2. Language and Database Details

Server Side Language Used: PHP

Database: MySQL

Server: Xampp Server (Apache, MySQL)

3. Google API Details

1. Google Authentication API (OAuth 2)
2. Google Distance matrix API (PHP)
3. Google Geocoding API
4. Google Java Script API (For address Auto completion)

4. Files

1. index.php

-> Main Page

2. add_user.php

->Adding new user to the database

3. get_location.php

->To determine User Location and returning it to main Page

4. connect.php

->Database connection

5. page_b.php

->Just another page to show how we can handle multiple pages

6. gplus-lib (Folder)

->Contains all the files used for Google API as provided by Google (With no customization)

5. Database tables

Table – user

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	sr	int(11)			No	None	AUTO_INCREMENT
2	email	varchar(250)			No	None	
3	name	varchar(250)			No	None	
4	pic_url	varchar(250)			No	None	
5	category	varchar(255)			No	None	

Table – online

#	Name	Type	Collation	Attributes	Null	Default	Extra
1	sr	int(11)			No	None	AUTO_INCREMENT
2	email	varchar(250)			No	None	
3	name	varchar(250)			No	None	
4	lat	double			No	None	
5	lon	double			No	None	
6	login	timestamp		on update CURRENT_TIMESTAMP	No	CURRENT_TIMESTAMP	ON UPDATE CURRENT_TIMESTAMP
7	category	varchar(50)			No	None	

users – Keep the record of all the users

online – Only keeps track on online users

6.Online

Process of determining the online status of users

1. User is in Online table
   - ➔ Which is automatically added once the user logs in and provide its location
   - ➔ User is removed from online table once he logs out

2. The latest activity of user is not older than 5 minutes (which can be customized)
   - ➔ Sometimes user does not log out but simply left the site and that makes user to be still present in online table but the user is not active
   - ➔ To avoid that problem, we will only take users whose latest activity is not older than 5 minutes
   - ➔ To determine the latest activity, we update '**login'** field in '**online'** table
   - ➔ And the process of keeping the '**login'** field updated is explained in the code of **page_b.php**

7.index.php

index.php is the main page of the API

The code is explained through comments in the pages. Here is some little explanation of the process

Lines	Explanation
1-3	Including Database Connection File
12	Including google API library
14-16	Creating constants of API Client ID, Client Secret, Redirect URI
18	Starting Session
28-32	Creating new client and giving Client details
34	Creating Google+ Object
51-55	If there is request for logout, log out user and remove user from Online table
60-70	If session has email set, then updating latest activity time i.e. updating 'login' to now

84-90	If there is value of ‘code’ using get method which is returned once after authenticating through google Authenticate user using that value, setting session access token value and redirecting user to the redirect URI
97-120	If session access token is present i.e. user is logged in
100-103	If session access token is expired, then creating authentication Url to re-authenticate
107-115	Obtaining user details by Google+ object
117-120	Storing User details is session to make those easily accessible
121-124	If session access token **not** present, create authentication URL
131-132	If Authentication URL is set, then show button with a link to authenticate
135-137	Checking if user is already present in users table
138-141	If old user, then store category in session
139-142	New User: Redirect user to add_user.php page
154-161	Printing all the user details and logout button
173	Checking if user’s location is available in session
181-182	Putting user in online table
187-188	Selecting all the users in online table with common same category
194-243	If there are some users matching the above criteria, then determine their distance from user using Google Distance Matrix API and sorting according to distance
247-249	If no online users found
257-260	If user’s location is not in the session, then redirecting user to the get_location.php page to get the location

8.add_user.php

Lines	Explanation
1-11	HTML form to get category input by user
20	If category value is available through post method
22-24	Initiating session
25	If session email is set
27	storing category value in session
30-21	Adding user details in the ‘user’ table
32	Redirecting user to main page

9. get_location.php

Lines	Explanation
8-22	If we have a value of ‘autocomplete’ field using post method
10-16	Storing autocomplete value in string and geocoding it with the help of Google Geocoding API (**Input by user**)
19-21	Redirecting to main page with the values of latitude and longitude in session
27-36	If we have value of latitude and longitude using post method (**Using GPS**)
29-34	Storing values of latitude and longitude in session and redirecting to main page
43-67	Two forms one for user input and other one hidden for converting values of lat. and lon. Obtained through JavaScript to POST
75-86	intitAutocomplete() – function to autocomplete address in input field
89-101	Determining the coordinates using GPS

10.connect.php

Database connection is initiated in this file

11. page_b.php

The Code here ensures that we always get the user's latest activity time in login (online table)

i.e. if there is no activity by a user for some time we will consider that user offline

So this updates to the most recent time

Thus giving us more accurate knowledge about user’s activity

If we have multiple pages, then we need to insert this code in every page

This is the only aim for this page

12.gplus-lib [Folder]

As this folder contains google API libraries without customization so for details you can refer to http://developers.google.com

13.Google Credentials used :

1. OAUTH2(Google+ API) credential used (index.php |Line:29-31)
2. API Key (Distance Matrix API) → index.php | Line: 217
3. API key (Geocode API) → get_location.php | Line: 12
4. API key (JavaScript API) → get_location.php | line : 106