**עבודה 3.1 – API:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Explanation | Returns | Parameters | HTTP Method | Method Name | ID |
| Using GET because there is no information to deliver to the server or any wish to change any information and the userID pass by Token | LIST of points of interest Saved in Favorites according to userID:  Favorites[] | token | GET | favPoints | 1 |
| Open modal window to rate the points of interest. The review will save in the DataBase.  Using POST because there are parameters to send **.**  True= success else  False=fail | Bool | token, pointID, review | POST | review | 2 |
| Using GET because there is no information to deliver to the server or any wish to change any information | LIST of points of interest categories : Categories[] | - | GET | categories | 3 |
| Move the user to the page of the specific point of interest with all details about.  Using GET because there is no information to deliver to the server or any wish to change any information | Specific point of interest  )include: viewsNum, ratingPrecent, description, 2LastReview( | pointID | GET | pointInfo | 4 |
| Using GET because there is no information to deliver to the server or any wish to change any information | 3 random popular point of interst above a certain rating: popularPoints[] | cartainRate | GET | popPointInLoad | 5 |
| Using POST because there are parameters to send and we don't want to send details about the user in the URL.  After submit succeed move the user to the LoginPage.  True= success  else  False=fail. | Bool | firstName, lastName, userID, password, city, country, email, category[], que1,que2,ans1,ans2 | POST | register | 6 |
| Using POST because there are parameters (private answers) to send, and we send userID and we don't want to send details about the user in the URL. | User password:  String: password | userID,  securityAns1, securityAns2 | POST | restorePassword | 7 |
| Login cannot pass in GET.  If the login succeed  True= success else False=fail  Application provides a signed token to the client, and client stores that token and sends it along with every request.we send userID and we don't want to send details about the user in the URL | token | userID, password | POST | login | 8 |
| Using GET because there is no information to deliver to the server or any wish to change any information and the userID pass by Token. | LIST of 2 points of interest according to userID, category[] that the user interested in the registration:  PopularPointsCat[] | token | GET | 2popularPoints | 9 |
| If the user saved at least 2 points of interest, we show them. Else we show message "There are no points of interest in favorites".  Using GET because there is no information to deliver to the server or any wish to change any information and the userID pass by Token. | 2 saved points of interest of the userID from the favorites:  2FavoritesPoints[] | token | GET | 2favPoints | 10 |
| Using GET because the information sended to server is not private | LIST of points of interest according category: AllPointsInCat[] | category (name) | GET | pointsAccCat | 11 |
| Using GET because there is no information to deliver to the server or any wish to change any information | LIST of All points of interest divided into categories :  AllPoints[] | - | GET | points | 12 |
| Replace the sign to mark favorite sign and add the point of interest to the user LIST in the server: Favorites[].  And then return the counter of Favorites[]  Using POST because there are parameters to send and we update the DB. | Counter of favorite point of interest of the user (Favorites[] size) | pointID, token | POST | pointAsFav | 13 |
| Replace the sign to NOT mark favorite sign and delete the point of interest from the user LIST in the server: Favorites[]and then return the counter of Favorites[]. | Counter of favorite point of interest of the user (Favorites[] size) | pointID, token | DELETE | pointAsFav | 14 |
| Using POST because we send userID and we don't want to send details about the user in the URL. | User security qestion:  securityQue[] | userID | POST | securityQues | 15 |
| Update the point rate in the database.  Using POST because there are parameters to send and we update the DB**.**  True= success else  False=fail. | Bool | rate, token | POST | ratePoint | 16 |

point

{

Id

rate (in precent)

categoryID

name  
description

viewsNum

numRate

Image

}