**עבודה 3 - API**

JSON Client:

{

"username": string,

"FirstName": string,

"LastName": string,

"Password": string,

"Adress": string,

"City": string,

"Country": string,

"Phone": string,

"Cellular": string,

"Mail": string,

"CreditCardNumber": string,

"isADmin": int

}

JSON Garment

{

"GarmentName": string,

"PicturePath": string,

"InsertDate": Datetime,

"Price": int,

"StokAmount": int

}

JSON Order

{

"OrderID": int,

"username": string,

"OrderDate": Datetime,

"ShipmentDate": Datetime,

"Currency": string,

"TotalAmount": string

}

JSON Category

{

"CategoryID": int,

"CategoryName": string

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Explanation | Returns | Parameters | HTTP method | Method name | Id |
| Register cannot pass in GET because it is not secure.  True = success  False = fail | bool | FirstName, LastName, Adress,  City,  Country, Phone, Cellular, Mail, CreditCardNumber, isADmin, username, Password, categories: [{"CategoryID":int}] | POST | register |  |
| Login cannot pass in GET because it is not secure.  True = success  False = fail | bool | username, password | POST | login |  |
| restorePassword cannot pass in GET because it is not secure. Return password if success to restore, otherwise return "". | string | Username, Mail, City, Country, Cellular | POST | restorePassword |  |
| Garments are related to the seller not to the client. Using GET because there is no information to deliver to the server or any wish to change any information. | Return top 5 hot **Clothes** (from the last week):  JSON Garment[] | - | GET | **getTop5HotClothes** |  |
| Garment are related to the seller not to the client. Using GET because there is no information to deliver to the server or any wish to change any information. | Return new **Clothes** (from the last month):  JSON Garment[] | - | GET | **getNewClothes** |  |
| Garments are related to the seller not to the client. Using GET because there is no information to deliver to the server or any wish to change any information. | Return all clothes:  JSON Garment [] | - | GET | getAllClothes |  |
| Garments are related to the seller not to the client. Using GET because there is no details to deliver to the server to change any information. | Return recommended Garments:  [{ "GarmentName": string}] | username | GET | getRecommendedClothes |  |
| Orders are related to the seller not to the client. Using GET because there is no details to deliver to the server to change any information. | Return user orders:  JSON Orders[] | username | GET | getOrders |  |
| Order are related to the seller not to the client. Using POST because there is details to deliver to the server to insert information.  If success – returns the order details, otherwise returns empty order. | Return the order details:  JSON Order | username, ShipmentDate, Currency, TotalAmount,  clothes: :[{"GarmentName":string, "Amount":int}] | POST | buyCart |  |
| Client are related to the seller not to the client. Using GET because there is no information to deliver to the server or any wish to change any information. | Return all clients details in the system:  JSON Client[] | - | GET | getClients |  |
| Orders are related to the seller not to the client. Using GET because there is no information to deliver to the server or any wish to change any information. | Return all orders details in the system:  JSON Order[] | - | GET | getOrdersReports |  |
| Garments are related to the seller not to the client. Using POST because there is details to deliver to the server to change information.  True = success  False = fail | bool | GarmentName, PicturePath, Price, StokAmount, categoryID | POST | **addGarment** |  |
| Products are related to the seller not to the client. Using DELETE because there is details to deliver to the server to remove information.  True = success  False = fail | bool | GarmentName | DELETE | **deleteGarment** |  |
| Client cannot pass in GET because it is not secure.  True = success  False = fail | bool | FirstName, LastName, Adress,  City,  Country, Phone, Cellular, Mail, CreditCardNumber, isADmin, username, Password, categories: [{"CategoryID":int}] | POST | **addUser** |  |
| Clients are related to the seller not to the client. Using DELETE because there is details to deliver to the server to remove information.  True = success  False = fail | bool | username | DELETE | **deleteUser** |  |
| Products is related to the seller not to the client. Using GET because there is no details to deliver to change any information. | Return list of Garments with less than 3 amount in the Garment:  JSON Garment[] | - | GET | checkSupply |  |
| Garments are related to the seller not to the client. Using PUT because there is details to deliver to the server to update information.  True = success  False = fail | bool | clothes: :[{"GarmentName":string, "Amount":int}] | PUT | updateProductSupply |  |
| Categorues is related to the seller not to the client. Using GET because there is no details to deliver to change any information | Return list of Categories:  JSON Category[] | - | GET | **getAllCategories** |  |

* בביצוע ה-API התחלקנו למתכנת ולקוח. טל הייתה המתכנתת ומורן הלקוחה.

הקשיים שנתקלנו בהם היו:

* חוסר התאמה בין רצון הלקוח לרצון המתכנת, לדוגמא האם העגלה תשמר בצד הלקוח או בצד השרת.
* אי הסכמה לגבי מיקום ה-cookie. האם השרת הוא זה שמחזיר את תאריך הכניסה האחרון של המשתמש או שדבר זה נעשה בצד הלקוח.