**סביבות פיתוח באינטרנט – מטלה 3 חלק ב'**

**Json Notation:**

**Product =**

[productID] ,[productName] ,[entranceDate] ,[quantity],[price],[brand],[color]

**User** =

[userName],[firstName],[lastName],[country],[city],[address],[birthDate],[password]

,[mail],[phone],[isAdmin]

**Purchase =** [purchaseID],[userName],[purchaseDate] ,[deliveryDate] ,[price]

**PurchasedProduct =** [purchaseID],[productID],[quantity]

**cartProduct =** [productID],[quantity]

**Category =** [categoryID],[categoryName]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method Name** | **HTTP Method** | **Parameters** | **Returns** | **Explanation** |
| getTop5Products | GET | - | Product[5] | The server should return the top 5 products (most popular) in the shop. |
| Login | POST | {Username, password} | Success: {user}  Fail: 'fail' | POST is the safest method for login.  If the connection is successful returns true.  If there are products in the cart returns the cart Products. |
| getRestorePasswordQuestions | GET | {Username} | {question1,  question2} | Used for initiating the password retrieval process of a user.  The method returns the questions to the user. |
| restorePassword | POST | Content:[  {username},  {Answer1,Answer2}  ] | Success:Password  Fail:'fail' | Used for answering the questions for password restore. |
| register | POST | Content:[  {User},  {Q1,A1,Q2,A2},  {Cat1,Cat2 }  ] | Bool | Creates a new user. Sends the user details, questions and answers for password retrieval and user's preferred categories.  Questions sent by their ID because we chose to limit the user with a certain questions. |
| getCategories | GET | - | [AllCategories] | Return List of the categories. |
| get5NewestProducts | GET | - | Product[5] | Returns a list of the 5 newest products in the shop. |
| getRecommendedProductsByUsers | GET | userName | Products[3] | Returns list of the recommended products as calculated by similar users up to 3 products. |
| getRecommendedProductsByCategories | GET | userName | Products[3] | Returns list of the recommended products by the user preffered categories. |
| getUserPurchases | GET | userName | Purchases[] | Return list of the purchases made by the user |
| getCurrency | GET | - | float | Returns the current currency  For NIS<-->Dollar.  Made with usage of external API. |
| Purchase | POST | Content:[  {username, deliveryDate,  Price},  cartProducts[]  ] | Success:  {purchaseID }  Fail:  'fail' | Creates a new purchase object on the server. If the purchase fails return NULL. Else return the ID of the new purchase.  Updates all of the relevant tables with updated values. |
| gerPurchasedProducts | GET | purchaseID | purchasedProduct[] | Returns the products purchased on the purchase with 'purchaseID'. |
| updateCart | POST | Content:[  {userName},  cartProducts[]  ] | - | Updates the saved cart with the updated list of cart products (quantity for each product ID). |
| getUserCart | GET | userName | cartProducts[] | Returns the current cart of the user with the specified userName |
| findProduct | GET | productName | Success:product  Fail:[] | Returns product by specific model name. |
| getProductByID | GET | productID | Success:product  Fail:[] | Returns the product with the attached ID. |
| getProductsByCategory | GET | categoryID | Product[] | Returns the products with the specified category. |
| getProductsByColor | GET | color | Product[] | Returns the products with the specified color. |
| getProductsByBrand | GET | brand | Product[] | Returns the products with the specified brand. |
| getAllProducts | GET | productID[] | Product[] | Returns all the products in the DB. |
| removeProduct | POST | productID | Success:'success'  Fail:'fail' | Deletes the product from with the specified product. |
| addProduct | POST | Content:[  {product},  Category[]  ] | Success:'success'  Fail:'fail' | POST is used for the creation of the new entity. productID is set by the server so POST fits better to this case then PUT. |
| removeUser | POST | userName | Success:'success'  Fail:'fail' | Removes the user with the specified username. |

**הנחות:**

כל עניין ה-authentication ובדיקות הקלט מתבצע ב-client כפי שהוסבר על ידי ארז בכיתה. (למרות שלדעתנו ה-cookie ב-server).

הפונקציונליות של הוספה והורדה מהעגלה מתבצעות באמצעות updateCart. בכל Session התחברות ה-client יחזיק את העגלה המעודכנת לאורך כל ה-Session.

בזמן התחברות ייקבל המשתמש את המצב העדכני של העגלה מאחר והחלטנו לשמור אותה בצורה מקומית אצל הלקוח.

גם הטיפול בנכונות הכמות הרצויה בהזמנה לעומת המלאי הקיים יקרה ב-client שיקבל את המידע לגבי המלאי הקיים כחלק מהמוצר. בכל פעם שיעבור לדף העגלה נעדכן את ה-client עם המלאי המעודכן ב- DB.

כרגע עבור כל המחיקות ב-DB ביצענו Cascade. (עבור foreign key כמובן) זאת מאחר ואנו מניחים שכאשר admin מוחק רשומה מסויימת הוא לוקח בחשבון שכל האינפורמציה הרלוונטית עבור יישות זו.