***‘Shop Stop’ Application Documentation****Coursework for the JavaScript class at   
University of Economics – Varna.*

C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0285750.wmf***C:\Program Files (x86)\Microsoft Office\MEDIA\CAGCAT10\j0292020.wmf***

**Academics**:

Prof. Dr. Snezhana Sulova

Chief Assist. Dr. Boris Bankov

**Prepared by:**

*Yordan Yordanov  
group 39  
F.№114452*

“*Shop Stop*” is very simple **product** **catalog** website (like [OLX](https://www.olx.bg/), [Amazon](https://www.amazon.com/) – but simpler). The application will consist of **users**, **products** and **categories**. Each **user** can **register**, **login** and **logout**. **Users** also can **create**, **buy**, **edit** or **delete** a **product**. Each **product** has **a** **category** in which it is specified. Site will implement of **searching** for a **product** by the product’s **name** or **category**.

### Project Specification

Designed and implemented a web application using HTML5, CSS3, JavaScript, Node.js, Express.js, Pug.js, Mongo DB and Mongoose.

Source code of the project [HERE](https://github.com/profjordanov/ShopStop" \t "_blank)!

### Functionality

* **User Login**
  + Login in current application using username and password of already registered user.
* **User Register**
  + Register a new user by providing username and password.
* **User Logout**
  + Logouts from the application.
* **Product Create**
  + Creates a new product, makes currently logged in user as it’s publisher.
* **Product Buy**
  + Products may be bought by any user.
  + One product could not be bought more than one time.
* **Product Edit**
  + Edits product's information.
* **Product Delete**
  + Deletes specific product from database.
* **Category Create**
  + Create a category which later on can be picked when a new product is created (uploaded).
* **Products by Name**
  + List all products which name contains given text.
* **Products by Category**
  + List all products are in particular category (use the name of the category).

### Models

Here will be described the mandatory information for each model:

**User**

* **Username** – **string** which is **required** and **unique**
* **Password** – **string** which is **required** and represents the **hashed** value of the user's **password**
* **Created Products** – **collection** of **products** created by given user

**Product**

* **Name** – **string** which is **required** and **unique**
* **Description** – **string** containing some additional information about the product
* **Price** – **decimal** number which is **non-negative**
* **Creator** – **reference** to the **user** who created the product
* **Image Url** – **string** containing **reference** to an **image** which displays the given product
* **Is Bought** – **bool** flag which is set to true whenever the user buys the product
* **Category** – **reference** to the **category** where the current product is placed in.

**Category**

* **Name** – **string** which is **required** and **unique**
* **Products** – **collection** of all **products** in given category

### This is how app looks like (before user login):

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | |
|  |  | |
|  |  | |

### This is how the app looks like (after user login):

|  |  |  |
| --- | --- | --- |
|  | | |
|  | | |
|  | |  |
|  | | |
|  | | |
|  | | |
|  | | |
|  |  | |
|  |  | |
|  |  | |