**Anqi Li**

**20082481**

**Description：**There are four schemas, cosmetics, customers, sellers and transactions. It can access all of the cosmetics from the database and sort them by low price or high price. It can also display a special type of cosmetics by full name or substring of name and filet them by brand. Cosmetics can be added, edited and deleted by sellers, but only can be ordered by customers.

Sellers and customers have sign up, sign in and upload user logo functions. They can view and edit their personal information. Customers can add cosmetics to transaction list and the status of this transaction is unpaid. Before ordering it, customers still can edit or delete their transaction. After ordering it, the status will change to paid. Sellers can change the status of transaction which is already paid to delivering status. Similarly, customers can change delivering status to finished status. This is an activity flow mimicked real online shopping. Besides, there is a function to count the sales of each cosmetic in this website.

|  |  |  |
| --- | --- | --- |
| Resource | URIs | HTTP Request |
| List of Cosmetics | /cosmetics | Get |
| Sort Cosmetics by Low Price | /cosmetics/sortByLowPrice | Get |
| Sort Cosmetics by High Price | /cosmetics/sortByHighPrice | Get |
| List a type of Cosmetics | /cosmetics/:name | Get |
| List a type of Cosmetics by Brand | /cosmetics/:name/:brand | Get |
| List of Sellers | /sellers | Get |
| List of Transactions | /transactions | Get |
| List Sales of Cosmetics | /transactions/countSales | Get |
| Sign up a Customer | /customer/signUp | Post |
| Sign in a Customer | /customer/login | Post |
| Upload a Customer Logo | /customer/:id/uploadLogo | Post |
| Display a Customer | /customer/:id | Get |
| Edit a Customer | /customer/:id/edit | Put |
| Add a Transaction | /transaction/:buyerId/add/:cosmeId | Post |
| List of a Customer’s Transactions | /transaction/:buyerId | Get |
| Delete a Transaction | /transaction/:buyerId/:id/remove | Delete |
| Edit a Transaction | /transaction/:buyerId/:id/edit | Put |
| Summit a Transaction | /transaction/:id/order | Put |
| Confirm Receipt of a transaction | /transaction/:id/confirmReceipt | Put |
| Sign up a Seller | /seller/signUp | Post |
| Sign in a Seller | /seller/login | Post |
| Display a Seller | /seller/:id | Get |
| Edit a Seller | /seller/:id/edit | Put |
| Add a cosmetic | /cosmetics/:publisher/add | Post |
| Edit a cosmetic | /cosmetics/:publisher/:id/edit | Put |
| Delete a cosmetic | /cosmetics/:publisher/:id/delete | Delete |
| Delivery a cosmetic | /transaction/:id/delivery | Put |

**Persistence approach:** persistence in application means data still exist even though the process is finished. In this website, cosmetics are created by sellers and transactions are created by customers. Sellers and customers sign up by themselves. All information of these four objects are stored in MongoDB, a document-oriented database. When a new seller/customer/cosmetic/transaction is created, the information of it will be write into MongoDB. After, we can get the data from MongoDB by reading JSON-like documents.

**Developer experience approach:** I use a video to show what can the website do and using README can file to descript this website. Release Notes and Changelogs using git during the development and upload source code to my GitHub account. Using README file to descript this website.

**GitHub Link:** https://github.com/SMARTBIGBOSS/cosmeticweb.git

**Reference:**

<https://developer.mozilla.org/zh-CN/docs/learn/Server-side/Express_Nodejs/mongoose>

<https://mongoosejs.com/docs/schematypes.html>

<https://segmentfault.com/a/1190000008245062>

<http://www.jsdaxue.com/archives/40.html>

<https://blog.csdn.net/little_blue_ljy/article/details/78252911>

<https://www.cnblogs.com/fangyuan303687320/p/5606790.html>

<https://blog.csdn.net/qwe502763576/article/details/79659548>

<https://www.youtube.com/watch?v=srPXMt1Q0nY>

<https://www.youtube.com/watch?v=9Qzmri1WaaE>

<https://www.youtube.com/watch?v=Q-BpqyOT3a8>

<https://www.youtube.com/watch?v=9_lKMTXVk64>

<https://www.youtube.com/watch?v=7nafaH9SddU>

<https://www.youtube.com/watch?v=Zaz1IcFLd2g>

<https://www.datastax.com/dev/blog/what-persistence-and-why-does-it-matter>

<https://hackernoon.com/the-best-practices-for-a-great-developer-experience-dx-9036834382b0>

<https://mongoose.shujuwajue.com/guide/validation.html>

https://stackoverflow.com/questions/32789053/populate-aggregate-in-mongoose/32794531