

Test Script	
Create user & Login Database testing	Date: 07/05/2020

Test Name	Browse Menu Database testing			
Use Case Tested:	Browse Menu			
Test Description:	Test for if the client can receive items from the database when called.			
Pre-conditions	An established Database			
Post-conditions	We either get an error that shows it failed or we get back items from the database.			
Notes:	Used postman and the app itself to check if they both receive the same things			
Result (Pass/Fail/Warning/Incomplete)	PASS			
	TEST STEP	EXPECTED TEST RESULTS	P	F
1.	Getting database information to display in postman (Used line "localhost/BeverageApi/public/getitems" to get the information).	The Api grabs the items and their respective information (id gets title, price, short description).	P	
2.	Getting database information to display in a live environment (the emulated Pixel 2).	The contents should display on screen, same as above.	P	

Added items into the MySql database

Test Script	
Create user & Login Database testing	Date: 07/05/2020

The results in picture form.

The image shows a REST client interface with a GET request to `localhost/BeverageApi/public/getitems`. The response body is a JSON array of three items:

```

1  {
2    "id": "1",
3    "name": "Branch",
4    "description": "A stick",
5    "price": "100"
6  },
7  {
8    "id": "2",
9    "name": "Coffee",
10   "description": "A morning brew",
11   "price": "5.5"
12 },
13 {
14   "id": "3",
15   "name": "Choclate Chip Cooky",
16   "description": "Good with coffee",
17   "price": "15.5"
18 }
19
20

```

Overlaid on the right is a mobile app interface titled "Customer_App". It displays the same data in a list format:

	Branch	100.0
1	A stick	
	Coffee	5.5
2	A morning brew	
	Choclate Chip Cooky	15.5
3	Good with coffee	

Test Script	
Create user & Login Database testing	Date: 07/05/2020

Test Data

id	title	shortdesc	price	image
1	Branch	A stick	100	BranchPicture
2	Coffee	A morning brew	5.5	PictureOfCoffee
3	Choclate Chip Cooky	Good with coffee	15.5	CookyFunny