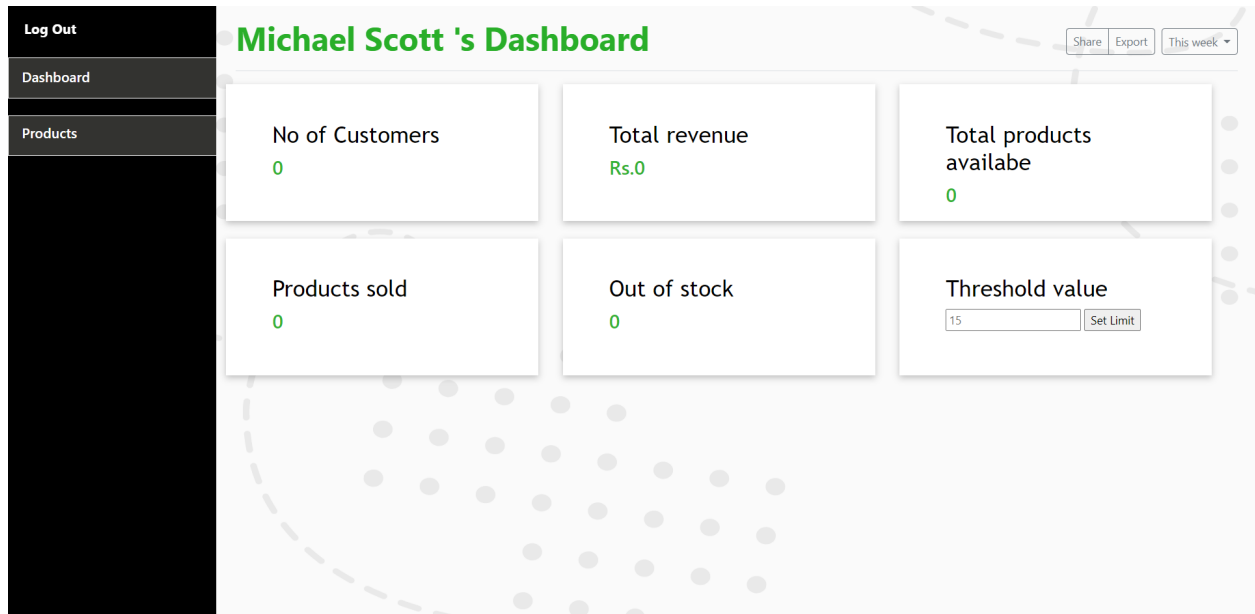


## Sprint 2

Team ID	PNT2022TMID03825
Project Name	Inventory Management System For Retailers

### Dashboard:



### Products Page:

The "Michael Scott's Product Details" page includes a top navigation bar with "Dashboard" and "Log Out" links, and an "Add a Product" button. The central table lists the following products:

Product ID	Product Name	Available Stock	Add Stock	Remove Stock	Delete Stock
4	Flower	17	<input type="text"/> +	<input type="text"/> -	
10	Corn	20	<input type="text"/> +	<input type="text"/> -	

The footer of the page is decorated with a series of icons representing inventory management tasks: boxes, a warehouse, a forklift, a checklist, and gears.

**CODE:**

## Dashboard.html

```

1 <!doctype html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>IMSFR</title>
8   <link rel="canonical" href="https://getbootstrap.com/docs/5.2/examples/dashboard/">
9   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
10     integrity="sha384-Zenh87qX5JnK2210vW8A8CKrkdKQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTR1" crossorigin="anonymous">
11     <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
12 </style>
13 html,
14 body {
15   height: 100%;
16   background-image: url("\\static\\images\\dashboardbg.png");
17   background-position: center;
18   background-repeat: no-repeat;
19   background-size: 100% 100%;
20 }
21
22 .bd-placeholder-img {
23   font-size: 1.125rem;
24   text-align: center;
25   -webkit-user-select: none;
26   -moz-user-select: none;
27   user-select: none;
28 }
29
30 @media (min-width: 768px) {
31   .bd-placeholder-img-lg {
32     font-size: 3.5rem;

```

```
<?php<br><!--<br>templates > dashboard.html > html<br><br>133 <body><br>134 <header><br>135 <div><br>136 <nav class="navbar navbar-expand-md fixed-top navbar-light bg-transparent mb-4"><br>137 <div class="container-fluid"><br>138 <button class="navbar-toggler type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse"><br>139 aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation"><br>140 <span class="navbar-toggler-icon"></span><br>141 </button><br>142 <div class="collapse navbar-collapse" id="navbarCollapse"><br>143 <ul class="navbar-nav me-auto mb-2 mb-md-0"><br>144 <li class="nav-item"><br>145 <a style="color: white; font-size: 17px; font-weight: 700; font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;"><br>146 href="{{ url_for('.logout') }}" class="nav-link">Log Out</a><br>147 </li><br>148 </ul><br>149 </div><br>150 </div><br>151 </nav><br>152 </div><br>153 </header><br><br>154 <div class="container-fluid"><br>155 <div class="row"><br>156 <div class="col-md-3 col-lg-2 d-md-block sidebar collapse"><br>157 <div class="position-sticky pt-3 sidebar-sticky"><br>158 <ul class="nav flex-column"><br>159 <li class="nav-item"><br>160 <div class="navborder"><br>161 <a style="color: white; font-size: 17px;" class="nav-link aria-current="page" href="#"><br>162 <span data-feather="home" class="align-text-bottom"></span><br>163 Dashboard<br>164
```

## Products.html

```
productsM.html X
templates > productsM.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css" rel="stylesheet"
9     integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzeps5IDxhbcnCeU0xjzrPF/et3URY9Bv1WTRI" crossorigin="anonymous">
10   <style>
11     input[type="number"] {
12       padding: 2px;
13       width: 80px;
14     }
15
16     input[type="image"] {
17       border: none;
18       padding: 0;
19       margin: 0;
20     }
21
22     .page-head {
23       padding-top: 40px;
24       text-align: center;
25       color: rgb(41, 174, 41);
26     }
27   </style>
28   <link href="/static/css/signin.css" rel="stylesheet">
29 </head>
30
31 <body>
32   <nav class="navbar navbar-expand-md fixed-top navbar-light bg-transparent mb-4">
33     <div class="container-fluid">
34       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarCollapse"
35         aria-controls="navbarCollapse" aria-expanded="false" aria-label="Toggle navigation">
36         <span class="navbar-toggler-icon"></span>
37       </button>
38       <div class="collapse navbar-collapse" id="navbarCollapse">
39         <ul class="navbar-nav me-auto mb-2 mb-md-0">
40           <li class="nav-item">
41             <a style="color: black; font-size: 17px; font-weight: 700; font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;"
42               href="{{ url_for('.dashboard', username=username) }}" class="nav-link">Dashboard</a>
43           </li>
44           <li class="nav-item">
45             <a style="color: black; font-size: 17px; font-weight: 700; font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;"
46               href="{{ url_for('.logout') }}" class="nav-link">Log Out</a>
47           </li>
48         </ul>
49       </div>
50     </div>
51   </nav>
52   <div class="heading">
53     <div class="page-head">
54       <h1>{{username}}'s Product Details</h1>
55     </div>
56   </div>
57   <div style="margin-left:50px;">
58     <br><br><br>
59     <a style="margin-bottom: 20px; color: black; font-size: 17px; font-weight: 700; font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;"
```

```

productsM.html X
templates > productsM.html > ...
64 <div style="margin-left:50px;">
65 <br>
66 <table style="background-color: # white;" class="table table-hover table-responsive">
67 <tr>
68 <th><label class="info head">Product ID</label></th>
69 <th><label class="info head">Product Name</label></th>
70 <th><label class="info head">Available Stock</label></th>
71 <th><label class="info head"></label></th>
72 <th><label style="margin-left:-195px;" class="info head">Add Stock</label></th>
73 <!-- <th><label class="info head"></label></th> -->
74 <th><label class="info head">Remove Stock</label></th>
75 <th><label class="info head">Delete Stock</label></th>
76 </tr>
77 {% for product in products %}
78 <tr>
79 <td><label class="info">{{product['PRODUCTID']}}</label></td>
80 <td><label class="info">{{product['PRODUCTNAME']}}</label></td>
81 <td><label class="info">{{product['AVAILABLESTOCK']}}</label></td>
82 <form method='POST'
83 action="{{url_for('.editProduct', username=username, action='add', pid=product['PRODUCTID'])}}">
84 <td><label class="info">
85 <input type="number" name="newstock" id="newstock" style="color:# black;" required>
86 </label></td>
87 <td>
88 <label style="margin-left: -90px;" class="info">
89 <input type="image" src="../../static/add.png" width="30" height="30">
90 </label>
91 </td>
92 </form>
93 </td>
94 <form method='POST'
95 action="{{url_for('.editProduct', username=username, action='remove', pid=product['PRODUCTID'])}}">
96 <td><label class="info">

```

## App.py

```

app.py 4 X
app.py > ...
64 @app.route("/<username>/dashboard", methods=('POST', 'GET'))
65 @app.route("/<username>", methods=('POST', 'GET'))
66 def dashboard(username):
67     fetchPrices = "SELECT p.productname,up.unitprice,up.availablestock FROM products p,userproducts up WHERE p.productid=up.productid AND up.user"
68     stmt = ibm_db.exec_immediate(conn, fetchPrices)
69     productInfo = ibm_db.fetch_assoc(stmt)
70     products=[]
71     prices=[]
72     while productInfo != False:
73         products.append(productInfo['PRODUCTNAME'])
74         prices.append(productInfo['UNITPRICE'] * productInfo['AVAILABLESTOCK'])
75         productInfo = ibm_db.fetch_assoc(stmt)
76
77     if request.method=='POST':
78         th_value = request.form['threshold']
79         sql = "INSERT INTO threshold_value (email, th_value) VALUES ('{}', '{}');".format(session['email'], th_value)
80         ibm_db.exec_immediate(conn, sql)
81         return redirect(url_for('.dashboard', username=username))
82     return render_template("dashboard.html", username=username, success=True, overallValue=sum(prices))
83
84
85 @app.route("/<username>/manageProducts", methods=('POST', 'GET'))
86 def manageProducts(username):
87     sql = "SELECT up.productid,p.productname,up.availablestock FROM products p, users u, userproducts up WHERE u.username=up.username AND p.produ"
88     stmt = ibm_db.exec_immediate(conn, sql)
89     getProducts = ibm_db.fetch_assoc(stmt)
90     products = []
91     while getProducts != False:
92         products.append(getProducts)
93         getProducts = ibm_db.fetch_assoc(stmt)
94     return render_template("productsM.html", username=username, success=True, products=products)
95

```

```

app.py 4 X
app.py > ...

96
97 @app.route("/<username>/manageProducts/edit=<pid>,action=<action>", methods=('POST', 'GET'))
98 def editProduct(username, pid, action):
99     if request.method == 'POST':
100         stock = int(request.form['newstock'])
101         checkAvailable = "SELECT * FROM userproducts WHERE productid='{}' AND username='{}';".format(pid, username)
102         statement = ibm_db.exec_immediate(conn, checkAvailable)
103         productDetails = ibm_db.fetch_assoc(statement)
104         available = productDetails['AVAILABLESTOCK']
105         if action == "add":
106             stock = available + stock
107         else:
108             stock = available - stock
109             checkstatus(conn, session['email'], username)
110             updateQuery = "UPDATE userproducts SET availablestock='{}' WHERE productid='{}' AND username='{}';".format(stock, pid, username)
111             ibm_db.exec_immediate(conn, updateQuery)
112     return redirect(url_for('.manageProducts', username=username))
113
114
115 @app.route("/<username>/manageProducts/delete=<pid>", methods=('POST', 'GET'))
116 def deleteProduct(username, pid):
117     deleteQuery = "DELETE FROM userproducts WHERE productid='{}' AND username='{}';".format(pid, username)
118     ibm_db.exec_immediate(conn, deleteQuery)
119     return redirect(url_for('.manageProducts', username=username))
120
121
122 @app.route("/<username>/addProduct", methods=('POST', 'GET'))
123 def addProducts(username):
124     error = None
125     if request.method == 'POST':
126         pid = request.form['pid']
127         pname = request.form['pname']

```