

DATE:	04.11.2022
TEAM ID:	PNT2022TMID02997
PROJECT NAME:	Inventory Management System for Retailers

SPRINT – II

FUNCTIONAL REQUIREMENT:

- Add products page
- Add sell page
- Sell Report
- Suppliers

DASHBOARD HTML

```
{% extends 'base.html' %}
```

```
{% block title %}
```

```
<title>Inventory Managmnet App</title>
```

```
{% endblock %}
```

```
{% block content %}
```

```
<main>
```

```
<div class="container-fluid">
```

```
<h1 class="mt-4">Dashboard</h1>
```

```
<ol class="breadcrumb mb-4">
```

```
<li class="breadcrumb-item active">Dashboard</li>
```

```
</ol>
```

```
<div class="card mb-4">
```

```
<div class="card-header">Products</div>
```

```
<div class="card-body">
```

```
<div class="card mb-4">
```

```
<div class="card-header">
```

<i class="fas fa-table mr-1"></i>

Products Table

</div>

<div class="card-body">

<div class="table-responsive">

{ % if products|length < 1 % }

<h4>There are no Products, add one above</h4>

{ % else % }

<table class="table table-bordered" id="dataTable" width="100% " cellspacing="0">

<thead>

<tr>

<th>Product Name</th>

<th>Date</th>

<th>Actions</th>

</tr>

</thead>

<tfoot>

<tr>

<th>Product Name</th>

<th>Date</th>

<th>Actions</th>

</tr>

</tfoot>

<tbody>

{ % for product in products % }

<tr>

<td>{{ product.product_id }}</td>

<td>{{ product.date_created.date() }}</td>

<td>

Delete

[illegible]

```

        <th>Date</th>
        <th>Actions</th>
    </tr>
</thead>
<tfoot>
    <tr>
        <th>Location Name</th>
        <th>Date</th>
        <th>Actions</th>
    </tr>
</tfoot>
<tbody>
    { % for location in locations % }
    <tr>
        <td>{{ location.location_id }}</td>
        <td>{{ location.date_created.date() }}</td>
        <td>
            <a href="/delete-location/{{ location.location_id }}">Delete</a>
            <br>
            <a href="/update-location/{{ location.location_id }}">Update</a>
        </td>
    </tr>
    { % endfor % }
</tbody>
</table>
    { % endif % }
</div>
</div>
</div>
</div>

```

```

</div>

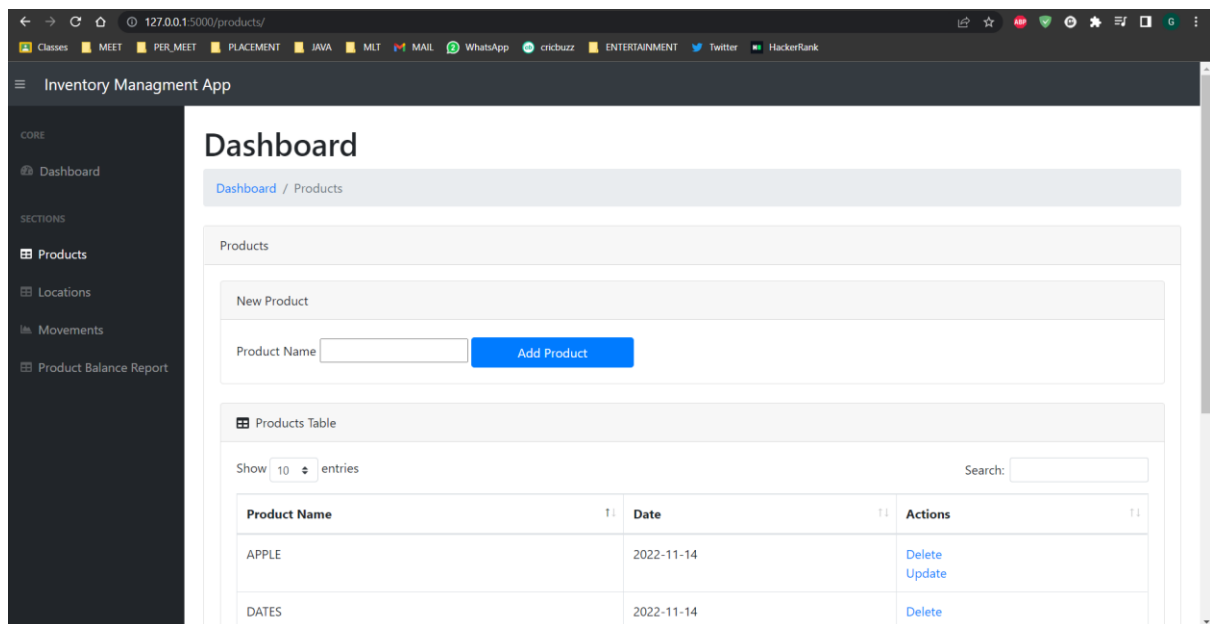
</div>

</main>

{% endblock %}

```

OUTPUT:



LOCATIONS HTML

```

{% extends 'base.html' %}

{% block title %}

<title>Inventory Managmnet App</title>

{% endblock %}

```

```

{% block content %}

<main>

  <div class="container-fluid">

    <h1 class="mt-4">Dashboard</h1>

    <ol class="breadcrumb mb-4">

```

```

    <li class="breadcrumb-item"><a href="/">Dashboard</a></li>
    <li class="breadcrumb-item active">Locations</li>
</ol>
<div class="card mb-4">
    <div class="card-header">Locations</div>
    <div class="card-body">
        <div class="card mb-4">
            <div class="card-header">New Location</div>
            <div class="card-body">
                <form action="/locations/" method="POST" id="location_form">
                    <label for="location_name" class="col-form-label">Location
Name</label>
                    <input type="text" name="location_name" id="location_name">
                    <input id="submitLocation" value="Add Location" class="btn btn-
primary">
                </form>
            </div>
        </div>
    </div>
    <div class="card mb-4">
        <div class="card-header">
            <i class="fas fa-table mr-1"></i>
            Locations Table
        </div>
        <div class="card-body">
            <div class="table-responsive">
                { % if locations|length < 1 % }
                <h4>There are no Locations, add one above</h4>
                { % else % }
                <table class="table table-bordered" id="dataTable" width="100%"
cellspacing="0">
                    <thead>

```

```

        <tr>
            <th>Location Name</th>
            <th>Date</th>
            <th>Actions</th>
        </tr>
    </thead>
    <tfoot>
        <tr>
            <th>Location Name</th>
            <th>Date</th>
            <th>Actions</th>
        </tr>
    </tfoot>
    <tbody>
        { % for location in locations % }
        <tr>
            <td>{{ location.location_id }}</td>
            <td>{{ location.date_created.date() }}</td>
            <td>
                <a href="/delete-location/{{ location.location_id }}">Delete</a>
                <br>
                <a href="/update-location/{{ location.location_id }}">Update</a>
            </td>
        </tr>
        { % endfor % }
    </tbody>
</table>
{ % endif % }
</div>
</div>

```

```

    </div>

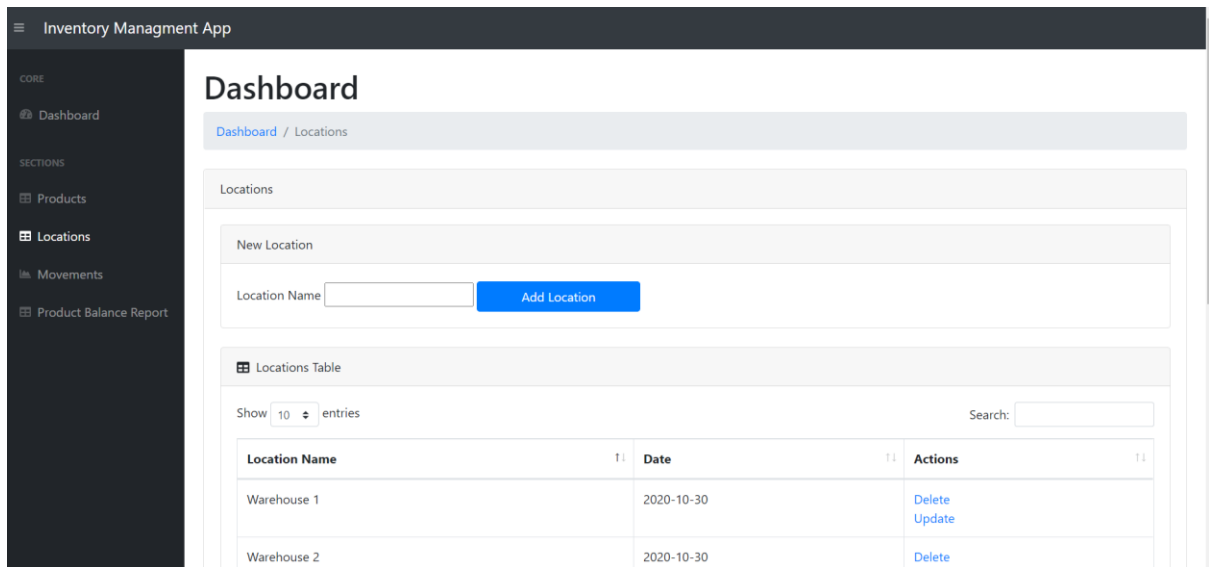
    </div>

    </div>

    </div>
</main>

{% endblock %}
```

OUTPUT:




```
app.py 2 x test_app.py # styles.css 4 inventory.db
C:\Users\saisa\Desktop> flask-inventory-app > app.py > ...
1 from flask import Flask, render_template, url_for, request, redirect
2 from flask sqlalchemy import SQLAlchemy
3 from collections import defaultdict
4 from datetime import datetime
5
6 app = Flask(__name__)
7 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///inventory.db'
8 db = SQLAlchemy(app)
9
10
11 class Product(db.Model):
12     __tablename__ = 'products'
13     product_id = db.Column(db.String(200), primary_key=True)
14     date_created = db.Column(db.DateTime, default=datetime.utcnow)
15
16     def __repr__(self):
17         return '<Product %r>' % self.product_id
18
19
20 class Location(db.Model):
21     __tablename__ = 'locations'
22     location_id = db.Column(db.String(200), primary_key=True)
23     date_created = db.Column(db.DateTime, default=datetime.utcnow)
24
25     def __repr__(self):
26         return '<Location %r>' % self.location_id
27
28
29 class ProductMovement(db.Model):
30     __tablename__ = 'productmovements'
31     movement_id = db.Column(db.Integer, primary_key=True)
32     product_id = db.Column(db.Integer, db.ForeignKey('products.product_id'))
33     qty = db.Column(db.Integer)
34     from_location = db.Column(db.Integer, db.ForeignKey('locations.location_id'))
35     to_location = db.Column(db.Integer, db.ForeignKey('locations.location_id'))
36     movement_time = db.Column(db.DateTime, default=datetime.utcnow)
```

```
app.py 2 x test_app.py # styles.css 4 ● inventory.db
C: > Users > saisa > Desktop > flask-inventory-app > app.py > ...

37
38     product      = db.relationship('Product', foreign_keys=product_id)
39     fromLoc       = db.relationship('Location', foreign_keys=from_location)
40     toLoc         = db.relationship('Location', foreign_keys=to_location)
41
42     def __repr__(self):
43         return '<ProductMovement %r>' % self.movement_id
44
45 @app.route('/', methods=["POST", "GET"])
46 def index():
47
48     if (request.method == "POST") and ('product name' in request.form):
49         product_name = request.form["product name"]
50         new_product   = Product(product_id product_name)
51
52         try:
53             db.session.add(new_product)
54             db.session.commit()
55             return redirect("/")
56
57         except:
58             return "There Was an issue while add a new Product"
59
60     if (request.method == "POST") and ('location name' in request.form):
61         location_name = request.form["location name"]
62         new_location  = Location(location_id location_name)
63
64         try:
65             db.session.add(new_location)
66             db.session.commit()
67             return redirect("/")
68
69         except:
70             return "There Was an issue while add a new location"
71     else:
72         products     = Product.query.order_by(Product.date_created).all()
```

```
app.py 2 x test_app.py # styles.css 4 inventory.db
C: > Users > saisa > Desktop > flask-inventory-app > app.py > ...
74     return render_template("index.html", products = products, locations = locations)
75
76 @app.route('/locations/', methods=["POST", "GET"])
77 def viewLocation():
78     if (request.method == "POST") and ('location_name' in request.form):
79         location_name = request.form["location_name"]
80         new_location = Location(location_id=location_name)
81
82         try:
83             db.session.add(new_location)
84             db.session.commit()
85             return redirect("/locations/")
86
87         except:
88             locations = Location.query.order_by(Location.date_created).all()
89             return "There Was an issue while add a new Location"
90     else:
91         locations = Location.query.order_by(Location.date_created).all()
92         return render_template("locations.html", locations=locations)
93
94 @app.route('/products/', methods=["POST", "GET"])
95 def viewProduct():
96     if (request.method == "POST") and ('product_name' in request.form):
97         product_name = request.form["product_name"]
98         new_product = Product(product_id=product_name)
99
100         try:
101             db.session.add(new_product)
102             db.session.commit()
103             return redirect("/products/")
104
105         except:
106             products = Product.query.order_by(Product.date_created).all()
107             return "There Was an issue while add a new Product"
108     else:
109         products = Product.query.order_by(Product.date_created).all()
```

```
app.py 2 x test_app.py # styles.css 4 inventory.db
C:\Users\saisa\Desktop> flask-inventory-app > app.py > ...
107 products = Product.query.order_by(Product.date_created).all()
110 return render_template("products.html", products=products)
111
112 @app.route("/update-product/<name>", methods=["POST", "GET"])
113 def updateProduct(name):
114     product = Product.query.get_or_404(name)
115     old_product = product.product_id
116
117     if request.method == "POST":
118         product.product_id = request.form['product_name']
119
120         try:
121             db.session.commit()
122             updateProductInMovements(old_product, request.form['product_name'])
123             return redirect("/products/")
124
125         except:
126             return "There was an issue while updating the Product"
127     else:
128         return render_template("update-product.html", product=product)
129
130 @app.route("/delete-product/<name>")
131 def deleteProduct(name):
132     product_to_delete = Product.query.get_or_404(name)
133
134     try:
135         db.session.delete(product_to_delete)
136         db.session.commit()
137         return redirect("/products/")
138     except:
139         return "There was an issue while deleting the Product"
140
141 @app.route("/update-location/<name>", methods=["POST", "GET"])
142 def updateLocation(name):
143     location = Location.query.get_or_404(name)
144     old_location = location.location_id
145
```

```
app.py 2 X test_app.py # styles.css 4 inventory.db
C: > Users > saisa > Desktop > flask-inventory-app > app.py > ...
139     return "There was an issue while deleteing the Product"
140
141 @app.route("/update-location/<name>", methods=["POST", "GET"])
142 def updateLocation(name):
143     location = Location.query.get_or_404(name)
144     old_location = location.location_id
145
146     if request.method == "POST":
147         location.location_id = request.form['location_name']
148
149         try:
150             db.session.commit()
151             updateLocationInMovements(
152                 old_location, request.form['location_name'])
153             return redirect("/locations/")
154
155         except:
156             return "There was an issue while updating the Location"
157     else:
158         return render_template("update-location.html", location=location)
159
160 @app.route("/delete-location/<name>")
161 def deleteLocation(id):
162     location_to_delete = Location.query.get_or_404(id)
163
164     try:
165         db.session.delete(location_to_delete)
166         db.session.commit()
167         return redirect("/locations/")
168     except:
169         return "There was an issue while deleteing the Location"
170
```