Warehouse Management

Mohammad Hasnain Rajan

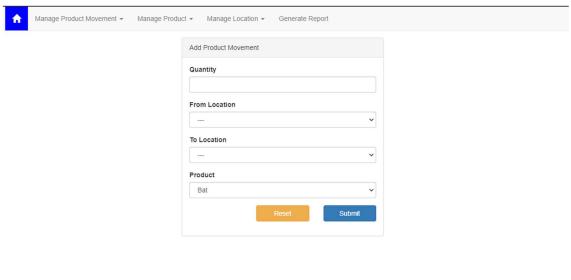
Following are the steps to run the application

- 1. git clone https://github.com/hasnain2808/Warehouse-Management
- 2. cd Warehouse-Management
- 3. python -m venv venv
- 4. activate the venv depending on the OS
 - a. Windows → venv\Scripts\activate
 - b. Ubuntu → source venv\bin\activate
- 5. python -m pip install -r requirements.txt
- 6. set variables depending upon os
 - a. Windows
 - i. set FLASK_APP=warehouse_management
 - ii. set FLASK_ENV=development
 - b. Ubuntu
 - i. export FLASK_APP=warehouse_management
 - ii. export FLASK_ENV=development
- 7. #flask init-db Run this command only if you want a fresh sqlite instance
- 8. flask run

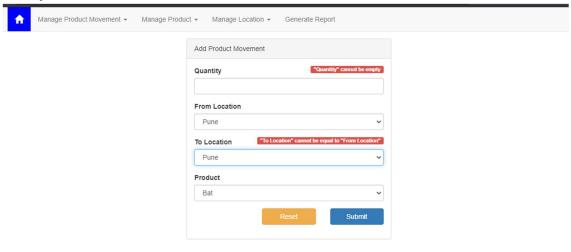
Go to http://localhost:5000/ to use the App

Updated UI Screenshots:

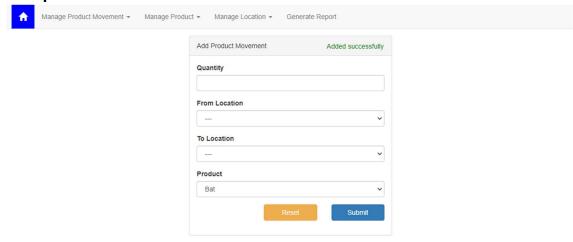
• Add Product Movement



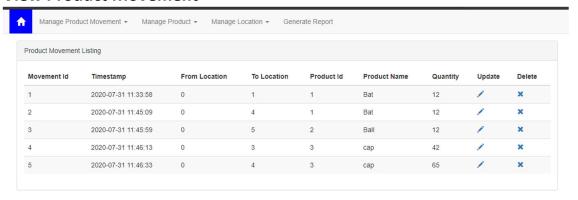
Add product movement validations



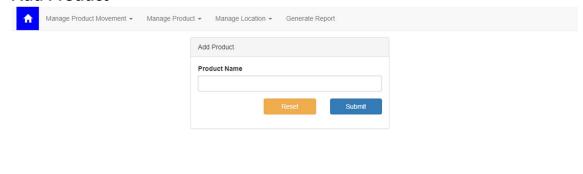
• Add product movement success



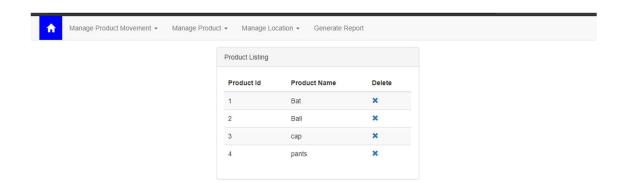
• View Product Movement



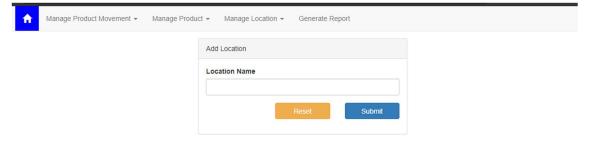
• Add Product



• View Product



• Add Location

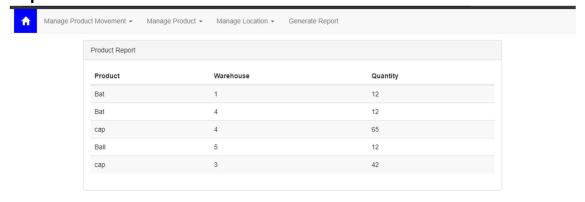


ocation/add_location#

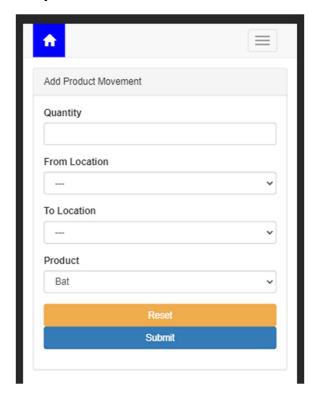
View Location



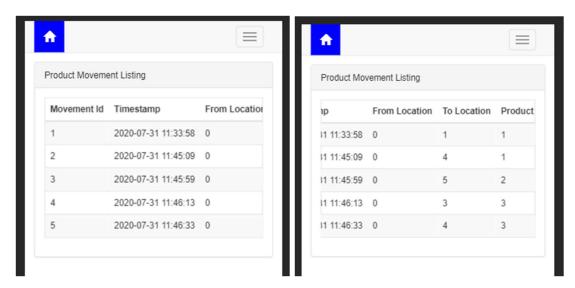
Report



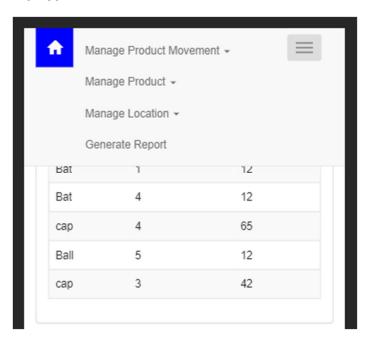
• Responsive



The tables can be scrolled



Navbar



This report summarizes the key features of the MVP

Database

SQLite was used so that evaluation is easier

The requirements issued directed creation of the following Tables

- 1. Product (product_id)
- 2. Location (location_id)
- 3. ProductMovement (movement_id, timestamp, from_location, to_location, product_id, qty)

Script used

```
DROP TABLE IF EXISTS Product;
DROP TABLE IF EXISTS Location;
DROP TABLE IF EXISTS ProductMovement;
CREATE TABLE Product (
 product_id INTEGER PRIMARY KEY AUTOINCREMENT
  );
CREATE TABLE Location (
 location_id INTEGER PRIMARY KEY AUTOINCREMENT
  );
CREATE TABLE ProductMovement (
 movement_id INTEGER PRIMARY KEY AUTOINCREMENT,
 timestamp TIMESTAMP DEFAULT CURRENT TIMESTAMP NOT NULL,
 from location INTEGER ,
 to_location INTEGER ,
 product_id INTEGER NOT NULL,
 qty INTEGER,
 FOREIGN KEY (movement_id) REFERENCES Product (product_id),
 FOREIGN KEY (from_location) REFERENCES Location (location_id),
 FOREIGN KEY (to_location) REFERENCES Location (location_id)
```

Views:

The views were created as directed in the requirements. Edit for Location and Product is not created because as they have only one column and that is the primary key. Editing it is not correct. Although Delete is added for all Entities

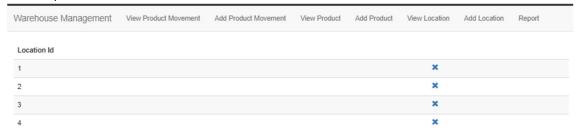
Requirement	Endpoint
Add Product	/product/add_product
View Product	/product/view_product
Delete Product	/product/delete_product/id
Add Location	/location/add_location
View Location	/location/view_location
Delete Location	/location/delete_location/id
Add ProductMovement	/product_movement/add_product_movement
Edit ProductMovement	/product_movement/update_product_movement/id
View ProductMovement	/product_movement/view_product_movement
Delete ProductMovement	/product_movement/delete_product_movement/1
Report	/product_movement/get_report

Use Cases

1. Create 3/4 Products

Product Id	Delete
1	×
2	×
3	×
4	×

2. Create 3/4 Locations



3. Make ProductMovements

a. After 8 Movements

View

Movement Id	Timestamp	From Location	To Location	Product Id	Quantity	Update	Delete
1	2020-07-21 01:44:44	None	1	1	5	1	×
2	2020-07-21 01:44:51	None	1	2	5	1	×
3	2020-07-21 01:45:01	None	2	3	5	1	×
4	2020-07-21 01:45:08	None	2	4	5	1	×
5	2020-07-21 01:45:42	None	3	1	5	1	×
6	2020-07-21 01:45:50	None	3	2	5	1	×
7	2020-07-21 01:46:01	None	4	3	5	1	×
8	2020-07-21 01:46:08	None	4	4	5	1	×

Report

Product	Warehouse	Quantity
1	1	5
2	1	5
3	2	5
4	2	5
1	3	5
2	3	5
3	4	5
4	4	5

b. After 12 movements

View

Warehouse Mar	nagement View Product Move	ment Add Product Movem	nent View Product	Add Product	View Location	Add Location	Report
Movement Id	Timestamp	From Location	To Location	Product Id	Quantity	Update	Delete
1	2020-07-21 01:44:44	None	1	1	5	1	×
2	2020-07-21 01:44:51	None	1	2	5	1	×
3	2020-07-21 01:45:01	None	2	3	5	1	×
4	2020-07-21 01:45:08	None	2	4	5	1	×
5	2020-07-21 01:45:42	None	3	1	5	1	×
6	2020-07-21 01:45:50	None	3	2	5	1	×
7	2020-07-21 01:46:01	None	4	3	5	1	×
8	2020-07-21 01:46:08	None	4	4	5	1	×
9	2020-07-21 01:48:49	1	2	1	2	1	×
10	2020-07-21 01:50:22	1	3	1	1	1	×
11	2020-07-21 01:51:14	4	2	4	2	1	×
13	2020-07-21 01:59:06	3	1	2	1	1	×

Report

Warehouse Management	View Product Movement	Add Product Movement	View Product	Add Product	View Location	Add Location	Report
Product	Wareho	use			Quantity		
1	1				2		
2	1				6		
3	2				5		
4	2				7		
1	2				2		
1	3				6		
2	3				4		
3	4				5		
4	4				3		

c. After 20 movements

Warehouse Man	nagement View Product Movemen	t Add Product Movemen	nt View Product	Add Product	View Location	Add Location	Report
Movement Id	Timestamp	From Location	To Location	Product Id	Quantity	Update	Delete
1	2020-07-21 01:44:44	None	1	1	5	1	×
2	2020-07-21 01:44:51	None	1	2	5	1	×
3	2020-07-21 01:45:01	None	2	3	5	1	×
4	2020-07-21 01:45:08	None	2	4	5	1	×
5	2020-07-21 01:45:42	None	3	1	5	1	×
6	2020-07-21 01:45:50	None	3	2	5	1	×
7	2020-07-21 01:46:01	None	4	3	5	1	×
8	2020-07-21 01:46:08	None	4	4	5	1	×
9	2020-07-21 01:48:49	1	2	1	2	1	×
10	2020-07-21 01:50:22	1	3	1	1	1	×
11	2020-07-21 01:51:14	4	2	4	2	1	×
13	2020-07-21 01:59:06	3	1	2	1	1	×
14	2020-07-21 02:01:32	4	1	3	4	1	×
15	2020-07-21 02:02:40	2	1	4	3	1	×
16	2020-07-21 02:03:48	3	4	1	1	1	×
17	2020-07-21 02:05:05	2	None	3	5	1	×
18	2020-07-21 02:05:46	1	None	4	3	1	×
19	2020-07-21 02:06:15	1	None	3	2	1	×
20	2020-07-21 02:07:11	3	None	1	3	1	×
21	2020-07-21 02:07:53	3	1	2	4	1	×

4. Final Report

Warehouse Management	View Product Movement Add Product Movemen	t View Product Add Product View Location Add Location Report
Product	Warehouse	Quantity
1	1	2
2	1	10
3	1	2
4	1	0
3	2	0
4	2	4
1	2	2
1	3	2
2	3	0
3	4	1
4	4	3
1	4	1

Key Features

- 1. Responsive UI using Bootstrap
- 2. Add, view, Delete Location
- 3. Add, view, Delete Product
- 4. Add, view, Edit, Delete ProductMovements
- 5. Final Report handles all types of product movements. If the user moves out something that is not available that is easily recognizable in the report by a negative quantity
- 6. Error handling using try/ catch so that appropriate message is displayed on the console and a error page to the user
- 7. Form Validations
- 8. Used SQLite to facilitate Evaluation
- 9. Code with comments and docstrings
- 10. Used Reusable templates for navbar