

FILE UP

2022 Dec 22

Rohit S Mathews

Swetha Rajeev

Vignesh S Kumar

INTRODUCTION

Fileup is a storage service that enables users to store and access files online. It stores documents and photos. The project is built with PHP, Bootstrap, HTML, and CSS.

TECHNOLOGY STACK

HTML

CSS



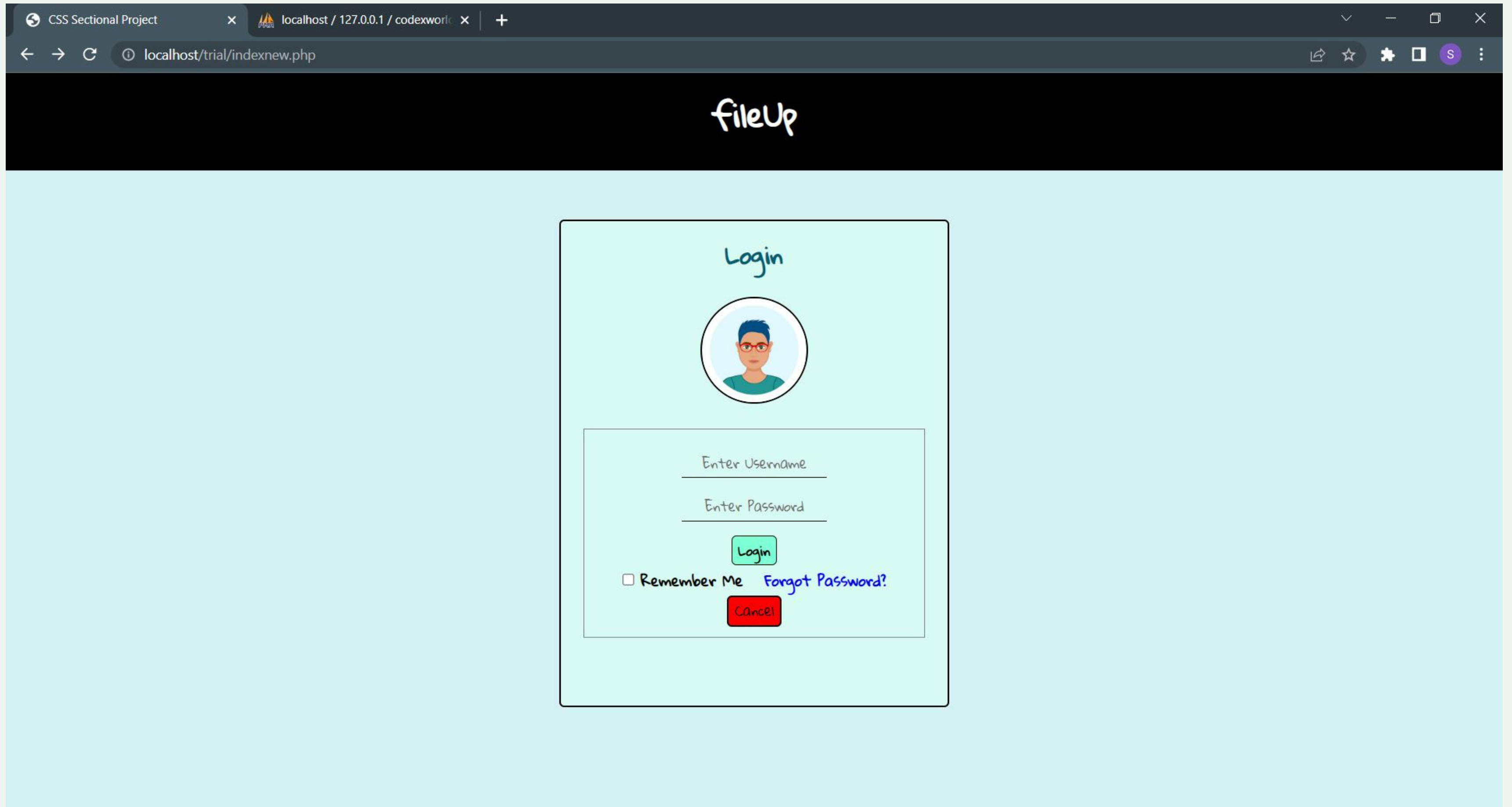
Frontend



Backend



Database



Select Image File to Upload:

Choose File

No file chosen

Upload

Uploaded Images

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPG

Delete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

1 Modifier: onlyOwner

2 Supplier registers in the Registration smart contract.

3 Supplier deploys the Inventory smart contract.

4 Supplier enters item details to add in the Inventory smart contract.

5 if item number exists already then

6 Increase the available quantity for that item.

7 else

8 Set the quantity and price for the new item and add it to the smart contract.

9 end

10 if price is changed then

11 Adjust the price of the item to the new price.

12 end

13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPG

Delete

Algorithm 3: Updating Reputation Scores of Suppliers

Input: supplier Ethereum address, status of transaction

1 Validate that the Ethereum address of the given supplier was added to the Reputation smart contract.

2 if retailer has already provided its feedback about this supplier then

3 Revert transaction.

4 else

5 Accept feedback.

6 if supplier provides acceptable service then

7 The reputation score of the supplier is increased.

8 else

9 The reputation score of the supplier is decreased.

10 end

11 .

12 end

algo3.JPG

Delete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

1 Modifier: onlyOwner

2 Supplier registers in the Registration smart contract.

3 Supplier deploys the Inventory smart contract.

4 Supplier enters item details to add in the Inventory smart contract.

5 if item number exists already then

6 Increase the available quantity for that item.

7 else

8 Set the quantity and price for the new item and add it to the smart contract.

9 end

10 if price is changed then

11 Adjust the price of the item to the new price.

12 end

13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPG

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPG

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPG

Delete

Document

localhost / 127.0.0.1 / codexworld

localhost/trial/home.php

Search

Search

logOut

fileUp

Select Image File to Upload:

Choose FileNo file chosenUpload

Uploaded Images

Algorithm 2: Purchase Orders
Input: inventory address, item number, quantity ordered
1 Modifier: onlyRetailer
2 The retailer places a purchase order in the Order Management smart contract.
3 **if** *Inventory smart contract Ethereum address is valid* **then**
4 Get the available number of items for sale for this product from the Inventory smart contract.
5 **if** *quantity ordered is available in stock* **then**
6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.
7 Inform the supplier of the new purchase order via events.
8 **end**
9 **else**
10 Revert transaction.
11 **end**

algo2.JPGDelete

Algorithm 1: Adding Items to the Supplier Inventory
Input: item Number, quantity to be added, item price
1 Modifier: onlyOwner
2 Supplier registers in the Registration smart contract.
3 Supplier deploys the Inventory smart contract.
4 Supplier enters item details to add in the Inventory smart contract.
5 **if** *item number exists already* **then**
6 Increase the available quantity for that item.
7 **else**
8 Set the quantity and price for the new item and add it to the smart contract.
9 **end**
10 **if** *price is changed* **then**
11 Adjust the price of the item to the new price.
12 **end**
13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPGDelete

Open

This PC > New Volume (D:) > Main Project

Search Main Project

OrganizeNew folder

New Volume (D:)

- Captures
- Certificates
- coding project
- fileupload
- Main Project
- minipro
- minipro1
- New folder
- oppol
- phn
- php

sample

SC

Seminar Report Sam

Supply_Chain_Inventory_Sharing_Using_Ethereum_Blockchain_and_S...

Swetha Rajeev_59

swetha web

swetha

System Architecture

Tidal

tutorial 2 AI

File name: sc

All Files

Open

Cancel

Delete

Select Image File to Upload:

Choose Filesc.pngUpload

Uploaded Images

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPGDelete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

1 Modifier: onlyOwner

2 Supplier registers in the Registration smart contract.

3 Supplier deploys the Inventory smart contract.

4 Supplier enters item details to add in the Inventory smart contract.

5 if item number exists already then

6 Increase the available quantity for that item.

7 else

8 Set the quantity and price for the new item and add it to the smart contract.

9 end

10 if price is changed then

11 Adjust the price of the item to the new price.

12 end

13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPGDelete

Algorithm 3: Updating Reputation Scores of Suppliers

Input: supplier Ethereum address, status of transaction

1 Validate that the Ethereum address of the given supplier was added to the Reputation smart contract.

2 if retailer has already provided its feedback about this supplier then

3 Revert transaction.

4 else

5 Accept feedback.

6 if supplier provides acceptable service then

7 The reputation score of the supplier is increased.

8 else

9 The reputation score of the supplier is decreased.

10 end

11 .

12 end

algo3.JPGDelete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

1 Modifier: onlyOwner

2 Supplier registers in the Registration smart contract.

3 Supplier deploys the Inventory smart contract.

4 Supplier enters item details to add in the Inventory smart contract.

5 if item number exists already then

6 Increase the available quantity for that item.

7 else

8 Set the quantity and price for the new item and add it to the smart contract.

9 end

10 if price is changed then

11 Adjust the price of the item to the new price.

12 end

13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPGDelete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPGDelete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

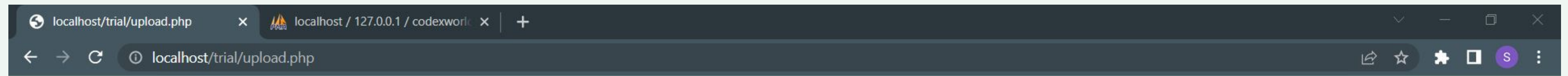
9 else

10 Revert transaction.

11 end

algo2.JPGDelete

7



The file sc.png has been uploaded successfully.

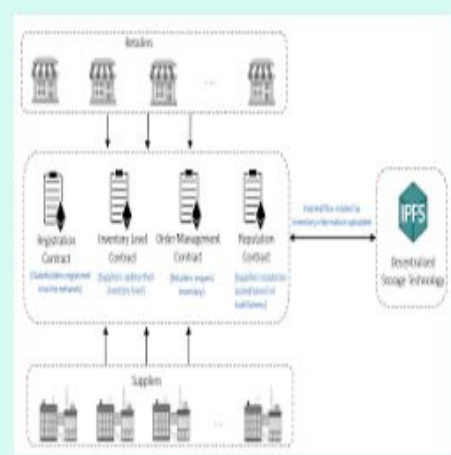
Select Image File to Upload:

Choose File

sc.png

Upload

Uploaded Images



sc.png

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

- 1 Modifier: onlyRetailer
- 2 The retailer places a purchase order in the Order Management smart contract.
- 3 **if** Inventory smart contract Ethereum address is valid **then**
- 4 Get the available number of items for sale for this product from the Inventory smart contract.
- 5 **if** quantity ordered is available in stock **then**
- 6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.
- 7 Inform the supplier of the new purchase order via events.
- 8 **end**
- 9 **else**
- 10 Revert transaction.
- 11 **end**

algo2.JPG

Delete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

- 1 Modifier: onlyOwner
- 2 Supplier registers in the Registration smart contract.
- 3 Supplier deploys the Inventory smart contract.
- 4 Supplier enters item details to add in the Inventory smart contract.
- 5 **if** item number exists already **then**
- 6 Increase the available quantity for that item.
- 7 **else**
- 8 Set the quantity and price for the new item and add it to the smart contract.
- 9 **end**
- 10 **if** price is changed **then**
- 11 Adjust the price of the item to the new price.
- 12 **end**
- 13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPG

Delete

Algorithm 3: Updating Reputation Scores of Suppliers

Input: supplier Ethereum address, status of transaction

- 1 Validate that the Ethereum address of the given supplier was added to the Reputation smart contract.
- 2 **if** retailer has already provided its feedback about this supplier **then**
- 3 Revert transaction.
- 4 **else**
- 5 Accept feedback.
- 6 **if** supplier provides acceptable service **then**
- 7 The reputation score of the supplier is increased.
- 8 **else**
- 9 The reputation score of the supplier is decreased.
- 10 **end**
- 11
- 12 **end**

algo3.JPG

Delete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

- 1 Modifier: onlyOwner
- 2 Supplier registers in the Registration smart contract.
- 3 Supplier deploys the Inventory smart contract.
- 4 Supplier enters item details to add in the Inventory smart contract.
- 5 **if** item number exists already **then**
- 6 Increase the available quantity for that item.
- 7 **else**
- 8 Set the quantity and price for the new item and add it to the smart contract.
- 9 **end**
- 10 **if** price is changed **then**
- 11 Adjust the price of the item to the new price.
- 12 **end**
- 13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPG

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

- 1 Modifier: onlyRetailer
- 2 The retailer places a purchase order in the Order Management smart contract.
- 3 **if** Inventory smart contract Ethereum address is valid **then**
- 4 Get the available number of items for sale for this product from the Inventory smart contract.
- 5 **if** quantity ordered is available in stock **then**
- 6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.
- 7 Inform the supplier of the new purchase order via events.
- 8 **end**
- 9 **else**
- 10 Revert transaction.
- 11 **end**

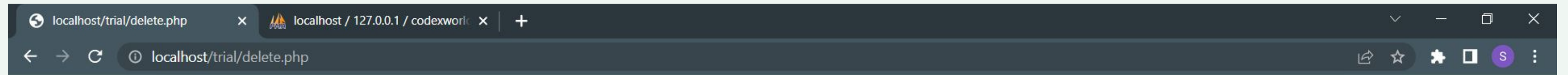
algo2.JPG

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

- 1 Modifier: onlyRetailer
- 2 The retailer places a purchase order in the Order Management smart contract.
- 3 **if** Inventory smart contract Ethereum address is valid **then**
- 4 Get the available number of items for sale for this product from the Inventory smart contract.
- 5 **if** quantity ordered is available in stock **then**
- 6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.
- 7 Inform the supplier of the new purchase order via events.
- 8 **end**
- 9 **else**
- 10 Revert transaction.
- 11 **end**



Deleted Successfully

Select Image File to Upload:

Choose File

No file chosen

Upload

Uploaded Images

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPG

Delete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

1 Modifier: onlyOwner

2 Supplier registers in the Registration smart contract.

3 Supplier deploys the Inventory smart contract.

4 Supplier enters item details to add in the Inventory smart contract.

5 if item number exists already then

6 Increase the available quantity for that item.

7 else

8 Set the quantity and price for the new item and add it to the smart contract.

9 end

10 if price is changed then

11 Adjust the price of the item to the new price.

12 end

13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPG

Delete

Algorithm 3: Updating Reputation Scores of Suppliers

Input: supplier Ethereum address, status of transaction

1 Validate that the Ethereum address of the given supplier was added to the Reputation smart contract.

2 if retailer has already provided its feedback about this supplier then

3 Revert transaction.

4 else

5 Accept feedback.

6 if supplier provides acceptable service then

7 The reputation score of the supplier is increased.

8 else

9 The reputation score of the supplier is decreased.

10 end

11 .

12 end

algo3.JPG

Delete

Algorithm 1: Adding Items to the Supplier Inventory

Input: item Number, quantity to be added, item price

1 Modifier: onlyOwner

2 Supplier registers in the Registration smart contract.

3 Supplier deploys the Inventory smart contract.

4 Supplier enters item details to add in the Inventory smart contract.

5 if item number exists already then

6 Increase the available quantity for that item.

7 else

8 Set the quantity and price for the new item and add it to the smart contract.

9 end

10 if price is changed then

11 Adjust the price of the item to the new price.

12 end

13 Broadcast the quantity update by triggering the appropriate event.

algo1.JPG

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPG

Delete

Algorithm 2: Purchase Orders

Input: inventory address,item number, quantity ordered

1 Modifier: onlyRetailer

2 The retailer places a purchase order in the Order Management smart contract.

3 if Inventory smart contract Ethereum address is valid then

4 Get the available number of items for sale for this product from the Inventory smart contract.

5 if quantity ordered is available in stock then

6 Deduct the quantity ordered from the available stock in the supplier inventory smart contract.

7 Inform the supplier of the new purchase order via events.

8 end

9 else

10 Revert transaction.

11 end

algo2.JPG

Delete

THANK YOU

