

INVENTORY MANAGEMENT SYSTEM

A PROJECT REPORT

Submitted for

CSE3009 – NoSQL Databases

by

Amit Kumar Sahu (18MIS7250)
Rushik kumar Avula (18BCN7008)



VIT-AP
UNIVERSITY

SCHOOL OF COMPUTER SCIENCE ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

May 2021

ABSTRACT

This report is based on the project we have done, which is called Inventory Management Software. This is developed mostly based on the trending NoSQL solution MongoDB. The main use of the above mentioned software is for the retailers who want to manage their product stocks among their various branches. The project is with a well understandable user interface with well formed security. The total code for this project can be found [here](#). Since the NoSQL solution is MongoDB we can say that our software is able to manage heavy data efficiently. Presently for the demo purpose we have initiated the software with 1k rows of data. With our software the stock manager can check for the products in the company's godown as well as insert, delete, update the products. We also have employed a chat feature for the go down manager and branch manager. Branch office has a separate module which is used for checking the product data present in the branch. This module also has the ability to insert, update the stock present in the branch.

TABLE OF CONTENTS

Chapter	Title	Page No.
1	Introduction	4
1.1	Objectives	4
1.2	Background and Literature Survey	5
1.3	Organization of the Report	5
2	Inventory Management System	6
2.1	Proposed System	6
2.2	Working Methodology	7
2.3	Software Details	7
3	Database Analysis	8
3.1	Dataset used and attributes	8
4	Results and Discussion	9
5	Conclusion & Future Works	15
6	Appendix	16
7	References	22

CHAPTER 1

INTRODUCTION

We know how the food and grocery market in India is rising day-by-day. Analysis says that by 2024, this market will reach \$790. In a gap of 7 years, from 2013 to 2019 this market doubled from \$300 to \$600. There are a lot of companies running behind this like D-Mart, Reliance Fresh, Big Bazar etc. and there is no need to talk about the online market because you see that every half yearly an app is getting released.

You can't imagine the amount of data released every day due to this. Each day, millions of transactions happen and companies need to record everything. For suppose consider a company like D-Mart, every day it gets its stocks from its go down, records it every time. While, go down, update it every time and insert product data when it gets the specific product from the company. They also maintain separate databases for the products in the branch because they alone can count to thousands. This huge data cannot be handled by traditional relational databases and this is why we need to shift to a more optimal solution called a NoSQL database.

1.1 Objectives

The following are the objectives of this project:

- Creating a web application to provide effective assistance to product stock managers of grocery or market companies.
- Effective solution for handling tons of data using NoSQL database MongoDB.
- Wide variety of features like inserting, deleting, updating and searching the available data.
- Easy to use and interactive UI.
- A communication portal for interaction between the company and branch managers.

1.2 Background and Literature Survey

The India online grocery market size was valued at USD 2.9 billion in 2020 and is expected to expand at a compound annual growth rate (CAGR) of 37.1% from 2021 to 2028. The market has gained immense traction over the past months on account of the changing lifestyle of the consumers, growing urbanization, and the tech-savvy generation who prefers buying products online.

This project formed on the basis of problems like the ever increasing data in grocery and market industry and traditional databases are not well in managing this and this can cause huge computational costs. We need a simple and effective solution for this problem and the solution is NoSQL database. This can manage the tonnes of data easily. Analysis can also be done with very less computational cost.

1.3 Organization of the Report

The remaining chapters of the project report are described as follows:

- Chapter 2 contains the proposed system, methodology and software details.
- Chapter 3 gives the cost involved in the implementation of the project.
- Chapter 4 discusses the results obtained after the project was implemented.
- Chapter 5 concludes the report.
- Chapter 6 consists of codes.
- Chapter 7 gives references.

CHAPTER 2

INVENTORY MANAGEMENT SYSTEM

This Chapter describes the proposed system, working methodology and software details.

2.1 Proposed System

The following block diagram shows the system architecture of this project.

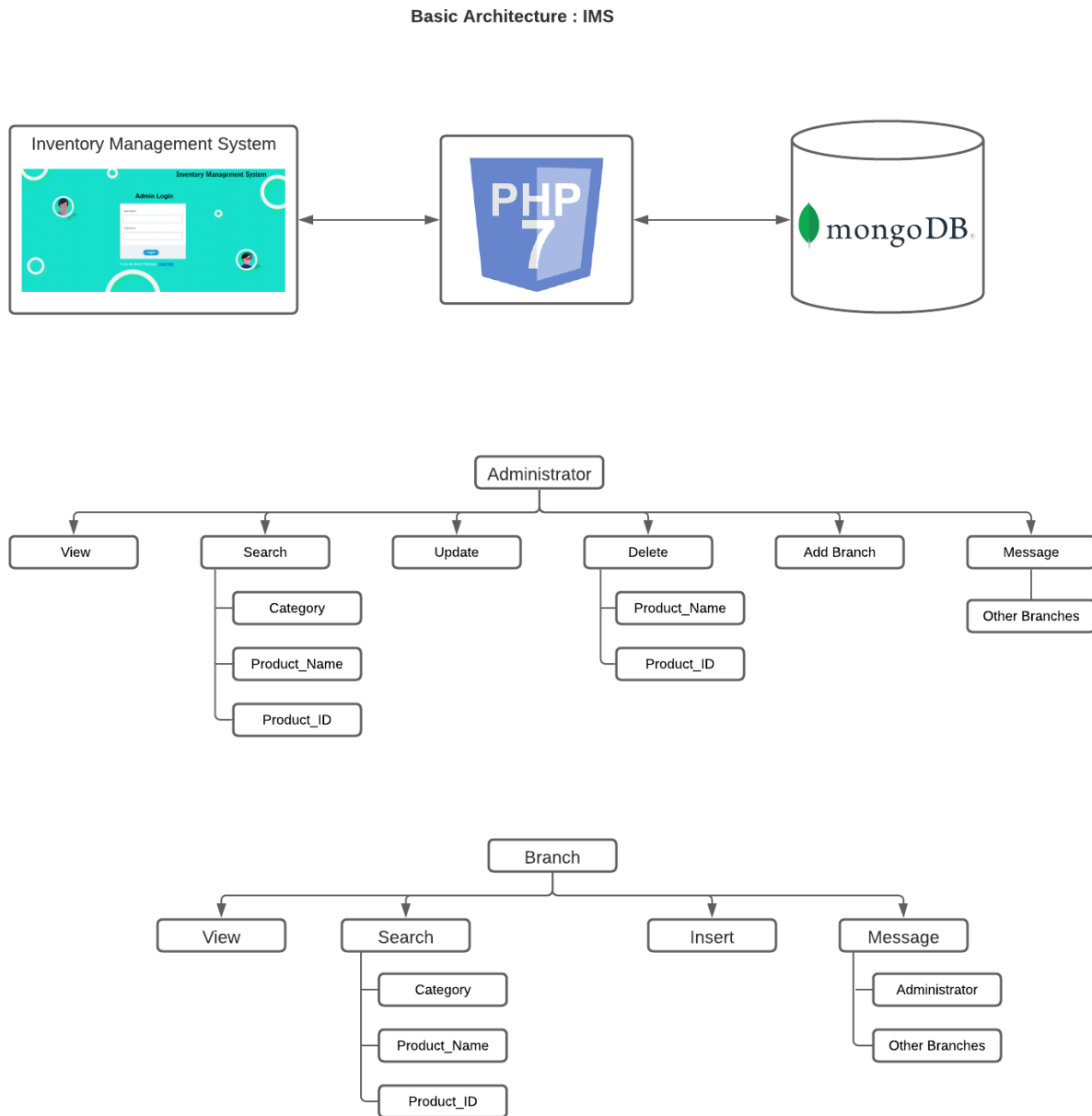


Figure 1. System Architecture Diagram

2.2 Working Methodology

The project is completely based on software with two sections, front-end and back-end. The front-end consists of a Bootstrap4 file. It has been hosted on Heroku (powered by Salesforce). The back-end is built using PHP-Composer. With a database as MongoDB atlas.

This application is used to show the stock details. It gives the details with search options by Category, by Product ID, or by Product Name. The details components are described below:

Admin Login page: As application starts the login page appears. Admin login is determined by the username and password that has all the authority to send message, search, add, update and delete the stock of the organization as per the requirement.

Create Branch: Admin can create a branch if he/she needs to extend or manage their product and shops easily. He can create the account of the branch along with Branch Name, Branch ID, Password and Address. Later he can share credentials to the Branch Manager to handover.

Branch Login page: This page is exclusively for branch manager login. The branch manager can send messages to other branches or the admin about the stocks and update them about his/her branch status, also the branch manager can search or insert products to the database.

2.3 Software Details

Front End	Bootstrap4
Back End	PHP - Composer (PHP v7)
Database	MongoDB Atlas
Server	Heroku

CHAPTER 3

DATABASE ANALYSIS

3.1 Datasets and attributes

We have used a cloud database - MongoDB Atlas free tier cluster.

In the document model of MongoDB Atlas, data becomes like code in the form of JSON documents. Developers, not DBAs, shape the data and easily map it to application objects, accelerating time to market.

The data consists of 8000 Product Details. With the following attributes.

```
_id: ObjectId("60af9c1bd0e6f1f0dbb6be85")
Order ID: "CA-2017-152156"
Ship Date: "11/11/2017"
Ship Mode: "Second Class"
Quantity: "42420"
Product ID: "FUR-BO-10001798"
Category: "Furniture"
Sub-Category: "Bookcases"
Product Name: "Bush Somerset Collection Bookcase"
Price: "261.96"
```

Other Schema

Product_data

DATABASE SIZE: 280.9KB INDEX SIZE: 80KB TOTAL COLLECTIONS: 2

CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
Branch_details	3	443B	148B	1	36KB	36KB
Godown_stock	1000	280.47KB	288B	1	44KB	44KB

Company_Chat

DATABASE SIZE: 363B INDEX SIZE: 76KB TOTAL COLLECTIONS: 3

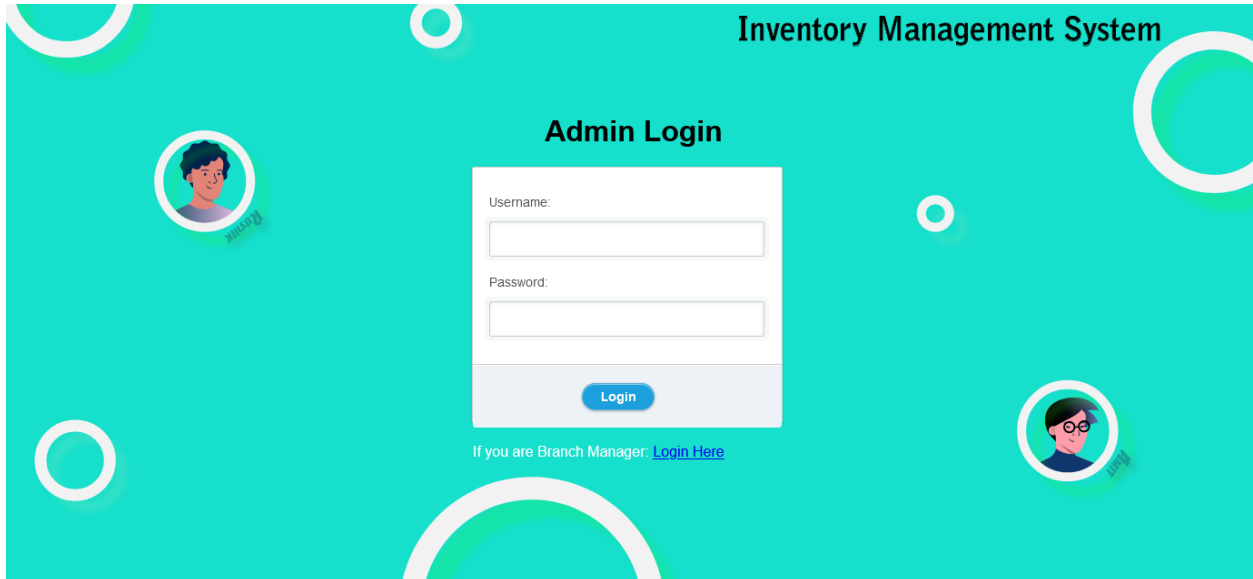
CREATE COLLECTION

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
Branch1	1	93B	93B	1	20KB	20KB
HYD98	1	112B	112B	1	20KB	20KB
admin	2	158B	79B	1	36KB	36KB

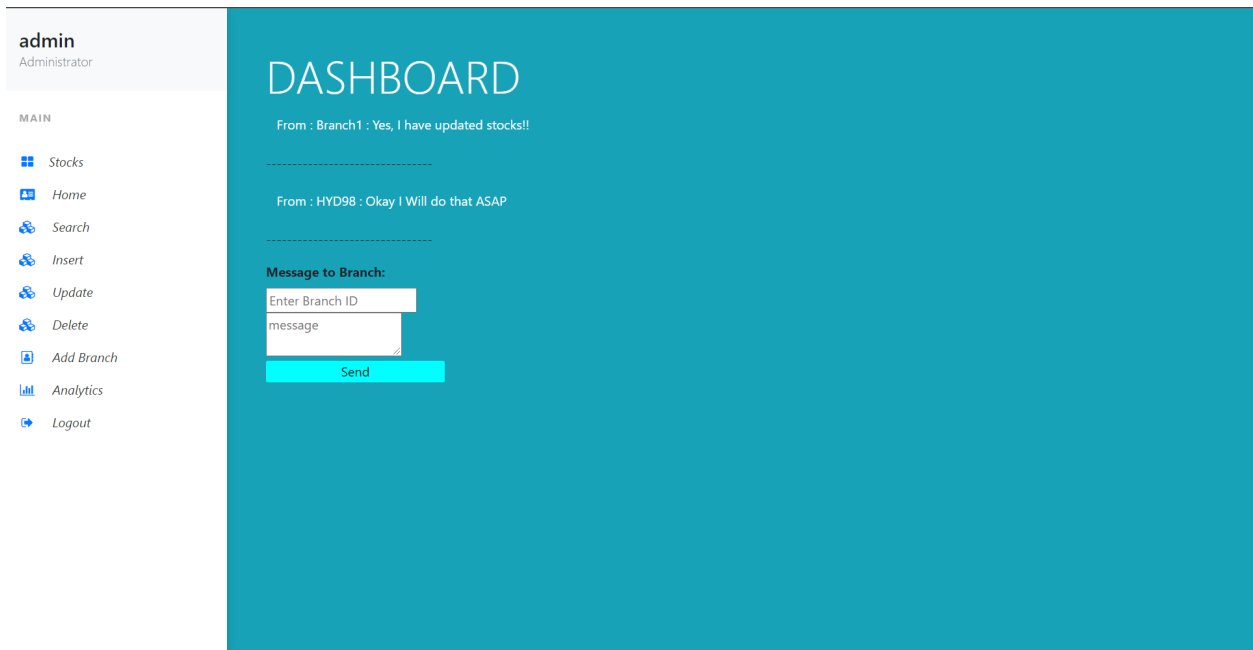
CHAPTER 4

RESULTS AND DISCUSSIONS

The application consists of two roles - admin and branch manager respectively. Both the roles have different access properties.



Admin DASHBOARD



View Stocks

admin
Administrator

MAIN

Stocks

Home

Search

Insert

Update

Delete

Add Branch

Analytics

Logout

STOCK DATA

Product ID	Product Name	Ship Date	Category	Sub-Category	Price	Quantity
FUR-BO-10001798	Bush Somerset Collection Bookcase	11/11/2017	Furniture	Bookcases	261.96	42420
FUR-CH-10000454	Hon Deluxe Fabric Upholstered Stacking Chairs, Rounded Back	11/11/2017	Furniture	Chairs	731.94	42420
OFF-LA-10000240	Self-Adhesive Address Labels for Typewriters by Universal	06/16/2017	Office Supplies	Labels	14.62	90036
FUR-TA-10000577	Bretford CR4500 Series Slim Rectangular Table	10/18/2016	Furniture	Tables	957.5775	33311
OFF-ST-10000577	Eldon Fold 'N Roll Cart System	10/18/2016	Office Supplies	Storage	22.368	33311

Search Stocks

admin
Administrator

MAIN

Stocks

Home

Search

Insert

Update

Delete

Add Branch

Analytics

Logout

SEARCH

Which product you want to check?

Choose a Category:

Furniture

Choose a Sub-Category:

Accessories

Search

Enter Product Name

Search

Enter Product ID

Search

Product ID	Product Name	Ship Date	Category	Sub-Category	Price	Quantity
TEC-AC-10003027	Imation 8GB Mini TravelDrive USB 2.0 Flash Drive	01/20/2017	Technology	Accessories	90.57	90049
TEC-AC-10000171	Verbatim 25 GB 6x Blu-ray Single Layer Recordable Disc, 25/Pack	03/13/2017	Technology	Accessories	45.98	55122
TEC-AC-10002167	Imation 8gb Micro Traveldrive Usb 2.0 Flash Drive	06/25/2017	Technology	Accessories	45	19901
TEC-AC-10002167	Imation 8gb Micro Traveldrive Usb 2.0 Flash Drive	06/18/2017	Technology	Accessories	30	12180

Add Product

admin
Administrator

MAIN

- Stocks
- Home
- Search
- Insert
- Update
- Delete
- Add Branch
- Analytics
- Logout

Insert

Insert a new product to the inventory?

Order ID: Ship Date: Choose a Ship Mode:

Quantity: Product ID: Choose a Category:

Choose a Sub-Category:

Product Name: Price:

Add Product with new category

admin
Administrator

MAIN

- Stocks
- Home
- Search
- Insert
- Update
- Delete
- Add Branch
- Analytics
- Logout

Insert

Insert a new product to the inventory?

Order ID: Ship Date: Choose a Ship Mode:

Quantity: Product ID: Category:

Sub-Category: Product Name: Price:

Update Product

The screenshot shows a web browser window with the URL `https://ims-nosql.herokuapp.com/update_p.php`. The page has a teal background and a sidebar on the left. The sidebar contains the user name 'admin' (Administrator) and a 'MAIN' menu with links: Stocks, Home, Search, Insert, Update, Delete, Add Branch, Analytics, and Logout. The main content area is titled 'Update' and asks 'Which product you want to update?'. It features a 'Product ID' input field. Below it is a dropdown menu for 'Ship Date' with a list of options: Ship Date, Ship Mode, Quantity, Category, Sub-Category, Product Name, and Price. The 'Ship Mode' option is selected, and an 'Update' button is visible next to it. The browser's taskbar at the bottom shows various application icons and the system clock indicating 11:51 on 11-06-2021.

admin
Administrator

MAIN

- Stocks
- Home
- Search
- Insert
- Update
- Delete
- Add Branch
- Analytics
- Logout

Update

Which product you want to update?

Product ID

Ship Date Enter Value

Ship Date

Ship Mode Update

Quantity

Category

Sub-Category

Product Name

Price

Delete Product

The screenshot shows a web browser window with the URL `https://ims-nosql.herokuapp.com/delete_p.php`. The page has a teal background and a sidebar on the left. The sidebar contains the user name 'admin' (Administrator) and a 'MAIN' menu with links: Stocks, Home, Search, Insert, Update, Delete, Add Branch, Analytics, and Logout. The main content area is titled 'DELETE' and asks 'Which product you want to delete?'. It features two input fields: 'Enter Product Name' and 'Enter Product ID', each with a 'Delete' button next to it. Below the input fields is a table with the following columns: Product ID, Product Name, Ship Date, Category, Sub-Category, Price, and Quantity. The browser's taskbar at the bottom shows various application icons and the system clock indicating 11:51 on 11-06-2021.

admin
Administrator

MAIN

- Stocks
- Home
- Search
- Insert
- Update
- Delete
- Add Branch
- Analytics
- Logout

DELETE

Which product you want to delete?

Enter Product Name Delete

Enter Product ID Delete

Product ID	Product Name	Ship Date	Category	Sub-Category	Price	Quantity
------------	--------------	-----------	----------	--------------	-------	----------

Add Branch

admin
Administrator

MAIN

Stocks

Home

Search

Insert

Update

Delete

Add Branch

Analytics

Logout

Add Branch

Branch Name:

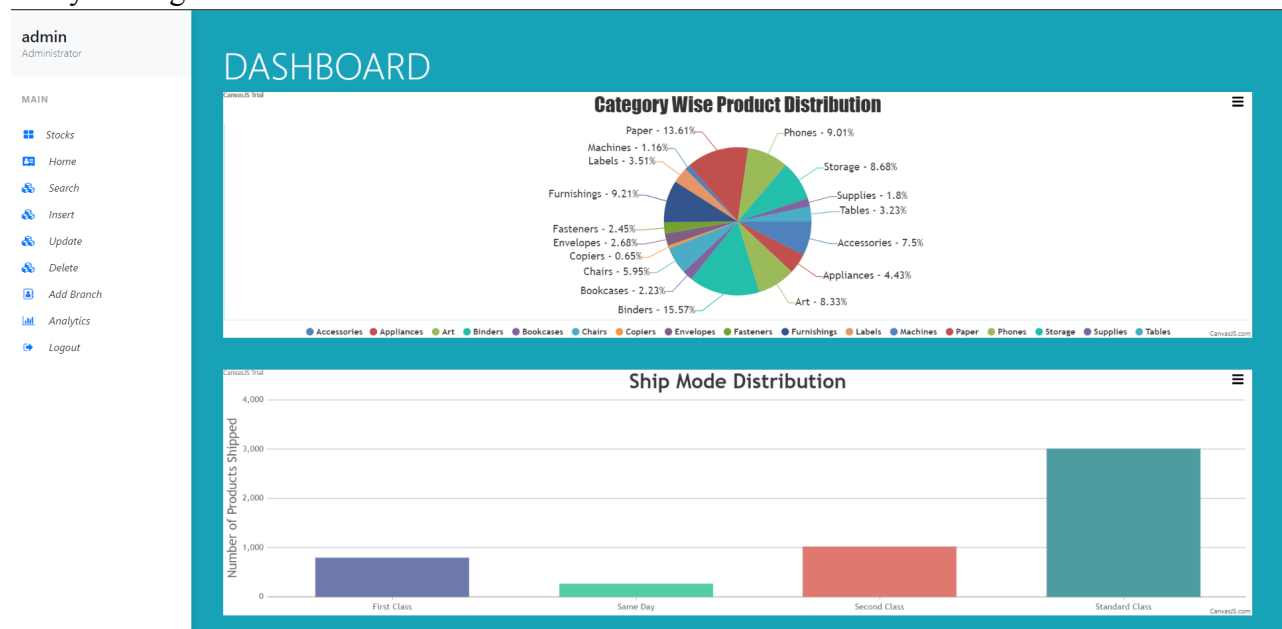
Branch Username:

Branch Password:

Address :

Add Branch

Analytics Page



Analytics Continue

admin
Administrator

MAIN

Stocks

Home

Search

Insert

Update

Delete

Add Branch

Analytics

Logout

Analysis

Price ▼ Greater than the Value ▼ Provide Some value: Analyze

Product ID	Product Name	Ship Date	Category	Sub-Category	Price	Quantity
10001798	Nexus	13/05/2021	Technology	Accessories	18999	23
TEC-MA-10000822	Lexmark MX611dhe Monochrome Laser Printer	09/12/2015	Technology	Machines	8159	78207
TEC-MA-10004125	Cubify CubeX 3D Printer Triple Head Print	11/04/2018	Technology	Machines	7999	27217
OFF-BI-10003527	Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind	03/22/2016	Office Supplies	Binders	6354	30318
PID-12346	RAM	13/05/2021	Computer Tools	Computer Accessories	5000	123
OFF-BI-10004995	GBC DocuBind P400 Electric Binding System	09/03/2017	Office Supplies	Binders	4355	10009
TEC-MA-10002927	Canon imageCLASS MF7460 Monochrome Digital Laser Multifunction Copier	11/09/2015	Technology	Machines	3991	10024
TEC-AC-10003832	Logitech P710e Mobile Speakerphone	09/17/2017	Technology	Accessories	3347	92037

Branch Manager Login

Inventory Management System

Branch Login

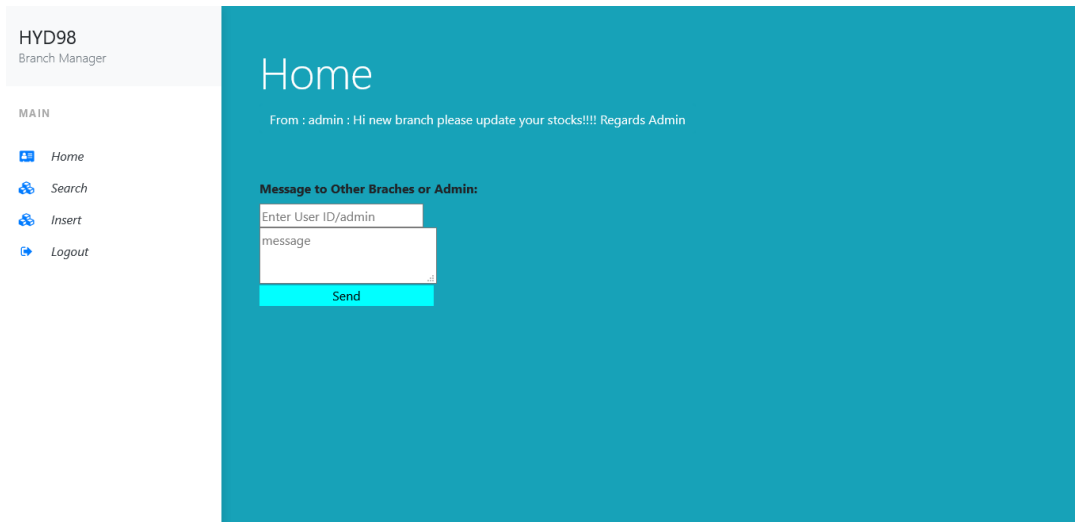
Username:

Password:

Login

If you are Admin [Login Here](#)

Branch DASHBOARD [Search and insert are same as admin page]



Project Link: <https://ims-nosql.herokuapp.com/index.php>

CHAPTER 5

CONCLUSION AND FUTURE WORK

This project helps people to record huge amounts of data easily and manage them perfectly. Each and every feature helps the manager to its maximum extent. Compared to traditional databases, NoSQL databases are easy to implement and operate.

This field has many future developments and it can include many features. Some of them are listed below

- Creating a module which is a direct implementation of sending stocks to the branches. For suppose, Branch requested for a certain amount of stocks then this module takes input of that data from admin and then does updation from Main Go_down, insertion in branch.
- Branch can have another module for recording the customer transactions.
- Analytics module which can show the transactions in the form of graphs.

CHAPTER 6

APPENDIX

Connection to MongoDB

```
<?php
require_once __DIR__ . '/mongo/vendor/autoload.php';
$con = new MongoClient("mongodb://localhost:27017");
?>
```

Inserting a product - insert_p.php

```
<?php
//Connecting MongoDB and to database and collection
include 'dbconnect.php';
$db = $con->Product_data;
$collection = $db->Godown_stock;

if(!empty($_POST['orderid'])) {
    $orderid = $_POST['orderid'];
    $shipdate = $_POST['shipdate'];
    $shipmode = $_POST['shipmode'];
    $quantity = $_POST['quantity'];
    $productid = $_POST['productid'];
    $categories = $_POST['categories'];
    $SubCategory = $_POST['SubCategory'];
    $pname = $_POST['pname'];
    $price = $_POST['price'];

    $insertOneResult = $collection->insertOne([
        'Order ID' => $orderid,
        'Ship Date' => $shipdate,
        'Ship Mode' => $shipmode,
        'Quantity' => $quantity,
        'Product ID' => $productid,
        'Ship Mode' => $shipmode,
        'Category' => $categories,
        'Sub-Category' => $SubCategory,
        'Product Name' => $pname,
        'Price' => $price,
    ]);
}
```



```

        echo "<script>alert('Inserted Data');</script>";
        printf("Inserted %d document(s)\n", $insertOneResult->getInsertedCount());
    }

    if(empty($_POST['pid'])){
        $pid="";
    }

?>

```

Update stock - update_p.php

```

<?php
    include 'dbconnect.php';
    $db = $con->Product_data;
    $collection = $db->Godown_stock;

    if(!empty($_POST['pid'])){

        $pid=$_POST["pid"];
        $pname=$_POST["pname"];
        $cat=$_POST["cat"];
        // $scat=$_POST["scat"];
        // $price=$_POST["price"];
        // $qnt=$_POST["qnt"];

        $cursor = $collection->updateOne(['Product ID'=> $pid,['$set'=>['Product
Name'=>$pname,'Category'=>$cat]]);

        foreach($cursor as $document){
            echo"<tr>";
            echo"<td>".$document["Product ID"]."</td>";
            echo"<td>".$document["Product Name"]."</td>";
            echo"<td>".$document["Ship Date"]."</td>";
            echo"<td>".$document["Category"]."</td>";
            echo"<td>".$document["Sub-Category"]."</td>";
            echo"<td>".$document["Price"]."</td>";

```

```

        echo"<td>".$document["Quantity"]."</td>";
        echo"</tr>";
    }
}

if(empty($_POST['pid'])){
    $pid="";
}

// $pass=$_POST["tf2"];
// $Query = array('Product ID' => $pid);

//iterator

?>

```

Search - search_p.php

```

<?php
//Connecting MongoDB and to database and collection
include 'dbconnect.php';
$db = $con->Product_data;
$collection = $db->Godown_stock;

//iterator category
$cursor = $collection->distinct('Category');
echo"<label for='categories'>Choose a Category:</label>";
echo"<select name='categories' id='category'>";
foreach($cursor as $document){
    echo"<option value='$document'>$document</option>";
}
echo"</select><br><br>";

//iterator sub - category
$cursor = $collection->distinct('Sub-Category');
echo"<label for='SubCategory'>Choose a Sub-Category:</label>";
echo"<select name='SubCategory' id='subcategory'>";
foreach($cursor as $document){
    echo"<option value='$document'>$document</option>";
}

```

```

    }
    echo"</select><br><br>";

?>
<input type="submit" value="Search" class="search-p1"><br><br>
</form>

</form>
<form action="search_p.php" method="post">
<input type="text" name="pname" placeholder="Enter Product Name">
<input type="submit" value="Search" class="search-p1"><br><br>
</form>
<form action="search_p.php" method="post">
<input type="text" name="pid" placeholder="Enter Product ID">
<input type="submit" value="Search" class="search-p1">
</form>
    </div> <br><br>

    <table>
    <tr>
    <th>Product ID</th>
    <th>Product Name</th>
    <th>Ship Date</th>
    <th>Category</th>
    <th>Sub-Category</th>
    <th>Price</th>
    <th>Quantity</th>
    </tr>

<?php

    if(!empty($_POST['categories']) or !empty($_POST['SubCategory'])) {
    $cat = $_POST["categories"];
    $sub = $_POST["SubCategory"];

    //iterator
    $cursor = $collection->find(['Category' => $cat, 'Sub-Category' => $sub,]);
    foreach($cursor as $document) {
    echo"<tr>";
    echo"<td>".$document["Product ID"]."</td>";

```

```

        echo"<td>".$document["Product Name"]."</td>";
        echo"<td>".$document["Ship Date"]."</td>";
        echo"<td>".$document["Category"]."</td>";
        echo"<td>".$document["Sub-Category"]."</td>";
        echo"<td>".$document["Price"]."</td>";
        echo"<td>".$document["Quantity"]."</td>";
        echo"</tr>";
    }
}

if(!empty($_POST['pname'])){
    $pname = $_POST["pname"];

    //iterator
    $cursor = $collection->find(['Product Name' => $pname,]);
    foreach($cursor as $document){
        echo"<tr>";
        echo"<td>".$document["Product ID"]."</td>";
        echo"<td>".$document["Product Name"]."</td>";
        echo"<td>".$document["Ship Date"]."</td>";
        echo"<td>".$document["Category"]."</td>";
        echo"<td>".$document["Sub-Category"]."</td>";
        echo"<td>".$document["Price"]."</td>";
        echo"<td>".$document["Quantity"]."</td>";
        echo"</tr>";
    }
}

if(!empty($_POST['pid'])){
    $pid = $_POST["pid"];

    //iterator
    $cursor = $collection->find(['Product ID' => $pid,]);
    foreach($cursor as $document){
        echo"<tr>";
        echo"<td>".$document["Product ID"]."</td>";
        echo"<td>".$document["Product Name"]."</td>";
        echo"<td>".$document["Ship Date"]."</td>";
        echo"<td>".$document["Category"]."</td>";
        echo"<td>".$document["Sub-Category"]."</td>";
        echo"<td>".$document["Price"]."</td>";
    }
}

```

```
echo"<td>".$document["Quantity"]."</td>";
echo"</tr>";
}
}

?>
```

Delete - delete_p.php

```
<?php
    include 'dbconnect.php';
    $db = $con->Product_data;
    $collection = $db->Godown_stock;

    if(!empty($_POST['pname'])){
        $pname = $_POST["pname"];

        //iterator
        $cursor = $collection->deleteOne(['Product Name' => $pname,]);

echo "<script>alert('Deleted Data');</script>";
printf("Inserted %d document(s)\n", $cursor->getDeletedCount());
    }

    else if(!empty($_POST['pid'])){
        $pid = $_POST["pid"];

        //iterator
        $cursor = $collection->deleteOne(['Product ID' => $pid,]);

echo "<script>alert('Deleted Data');</script>";
printf("Deleted %d document(s)\n", $cursor->getDeletedCount());
    }

?>
```

CHAPTER 7

REFERENCES

1. <https://docs.mongodb.com/drivers/php/>
2. https://www.tutorialspoint.com/mongodb/mongodb_php.htm
3. [how to connect mongo db with php using composer PART-1. - YouTube](#)
4. [How to connect mongodb with php part-2 - YouTube](#)
5. <https://docs.mongodb.com/php-library/v1.2/reference/method/MongoDBCollection-find/>
6. <https://www.consultancy.in/news/3408/indias-food-grocery-market-to-reach-790-billion-by-2024>