LIST OF FIGURES

LIST OF ACRONYMS

Sl.no	Acronyms	Expansion
1.	DBMS	Database Management System
2.	SQL	Structure Query Language
3.	PHP	Hypertext Preprocessor
4.	HTML	Hypertext Markup Language
5.	CSS	Cascading Style Sheet
6.	ER Diagram	Entity Relationship Diagram

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The Inventory Management System has been developed to override the problemsprevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by this existing system. Moreover this system is designed for the particular need of the company to carry out operations in a smooth and effective manner.

The application is reduced as much as possible to avoid the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it proves it is user-friendly. It can assist the user to concentrate ontheir other activities rather to concentrate on the record keeping. Thus it will help organizationin better utilization of resources.

Every organization whether big or small, has challenges to overcome and managing theinformation of Vendor, Inventory, Cost, Delivery, Order. Every Inventory Management System has different Inventory needs, therefore we design exclusive employee management systems that are adapted to your managerial requirements. This is designed to assist in strategic planning, and help you to ensure that your organization is equipped with the right level of information and for those busy executive who are always on the go, our systems comewith remote access features. These systems ultimately allow you to better manage resources.

1.2 DBMS (DATABASE MANAGEMENT SYSTEM)

Database is a collection of related data and data is a collection of facts and figures that can be processed to produce information. Mostly data represents recordable facts. Data aids in producing information, which is based on facts. For example, if we have data about marks obtained by all students, we can then conclude about toppers and average marks.

A database management system (DBMS) is a software package designed to define, manipulate, retrieve and manage data in a database. A DBMS generally manipulates the data itself, the data format, field names, record structure and file structure. It also defines rules to validate and manipulate this data.

A DBMS relieves users of framing programs for data maintenance. Fourth- generation query languages, such as SQL, are used along with the DBMS package to interact with a database. Some other DBMS examples include:

1.2.1 MySQL

1.2.2 SQL Server

1.2.3 Oracle

1.2.4 dBASE

1.2.5 FoxPro

1.3 PHP (HYPERTEXT PREPROCESSOR)

PHP is the most popular and widely used server-side scripting language for web development. It is used to make the Dynamic pages in websites. Rasmus LerdorfT was the creator of PHP in 1995. PHP codes are embedding in HTML source codes for making the page dynamic. PHP can deal with most of the requirements in web development like Database, File handling, String operations, Arrays, Graphics, File Uploads, Data processing etc. PHP can be used in any operating system with a web server Supports PHP. Apache web server is one of the popular web servers dealing with PHP + MySQL. Moreover, PHP is absolutely free to use.

1.4 PROBLEM STATEMENT

The Problem faced by the company is they do not have any systematic system to record and keep their inventory data . It is difficult for the admin to record the inventory data quickly andsafely because they only keep it in the logbook and not properly organized.

1.5 OBJECTIVES

The objective of the project is to deliver an efficient inventory management system whose main functionality apart from calculating the inventory include predicting the requirement forthe next demand and if there is a “Special Occasion” then accordingly the manager selects theparticular occasion and extra requirements are added to the next issuing order to the vendors which needs to be approved by the manager.

The product also aims to keep track of the shelf life of resources. If any resource nears the end of its shelf life, it would acknowledge to the manager (admin) the details of the quantity that isnear its expiration date.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 HARWARE REQUIREMENTS

The hardware required for the development of this project is:

• Processor	: Intel Core i7
• Processor speed	: 2.4 GHz
• RAM	: 8 GB RAM
• System Type	: 64-Bit Operating System

2.2 SOFTWARE REQUIREMENTS

The software required for the development of this project is:

• Software	: MySQL and xampp.
• OS	: Windows 7 and above
• Front End	: HTML, CSS , jQuery and JavaScript.
• Programming Language	: Php and SQL
• Database	: MySQL

CHAPTER 3

DESIGNS

3.1 ENTITY RELATIONSHIP DIAGRAM

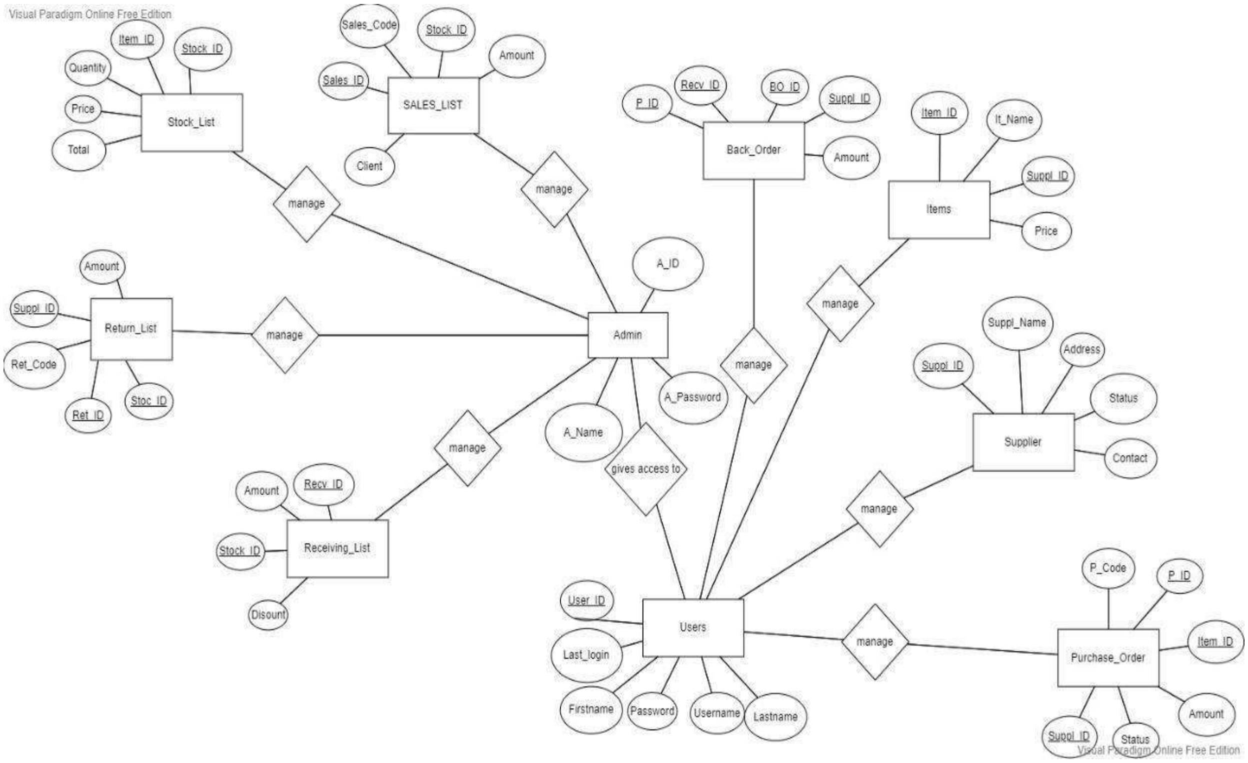


Fig 3.1 Entity Relation Diagram for Inventory Management System

3.2 DESCRIPTION

The ER Model figure shows conceptual view of the database. It works around real-world entities and the associations among them. At view level, the ER model is considered a good option for designing databases. So, let's see each entity:

ADMIN TABLE

This entity stores the information about admin who logs in using his username and password .And can add users.

USER TABLE

This entity stores the information about users who are added by the admin. Attributes are user_id, firstname , lastname, username ,password, last_login.

PURCHASE_ORDER TABLE

This entity stores the information about the purchase made by the Admin or Staff. Attributes are p_id, p_code, suppl_id, status , amount and item_id .

SUPPLIER TABLE

This entity stores information about suppliers who supplies bulk goods to the Inventory. Attributes are suppl_id, suppl_name, contact, status and address.

RECEIVING_LIST TABLE

This entity stores records for the bulk goods received by the Suppliers. Attributes are recv_id , amount, discount, stock_id.

ITEMS TABLE

This entity stores the records and details about the items . Attributes are item_id , it_name , suppl_id , price .

BACK_ORDER TABLE

This entity records the details about goods which are partially received from the suppliers.Attributes are bo_id, recv_id, p_id, bo_code, suppl_id, amount and status.

RETURN_LIST TABLE

This entity stores the details about the goods which are returned back to the supplier due tovarious reasons. Attributes are ret_id, ret_code, suppl_id, stock_id, and amount.

STOCK_LIST TABLE

This entity contains the records about the goods currently in Inventory . Attributes are stock_id, item_id, quantity, units, cost and total amount.

SALES_LIST TABLE

This table records the goods of the Inventory which are available for Sale for clients . Attributes are sales_id, sales_code, stock_id, clients and amount.

3.3 SEVEN STEPS FOR ER TO SCHEMA CONVERSION

Step 1: Mapping of Regular Entity Types.
For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E. Include only the simple component attributes of a composite attribute. Choose one of the key attributes of E as the primary key for R. If the chosen key of E is a composite, then the set of simple attributes that form it will together form the primary key of R. If multiple keys were identified for E during the conceptual design, the information describing the attributes that form each additional key is kept in order to specify secondary (unique) keys of relation R. Knowledge about keys is also kept for indexing purposes andother types of analyses.

Step 2: Mapping of Weak Entity Types.
For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes (or simple components of composite attributes) of was attributes of R. In addition, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s); this takes care of mapping the identifying relationship type of W. The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any. If there is a weak entity type E2 whose owner is also a weak entity type E1, then E1 should be mapped before E2 to determine its primary key first.

Step 3: Mapping of Binary1:1 Relationship Types.
For each binary 1:1 relationship type R in the ER schema, identify the relations S and T thatcorrespond to the entity types participating in R. There are three possible approaches:

1. The foreign key approach.

2. The merged relationship approach, and

The first approach is the most useful and should be followed unless special conditions exist, aswe discuss below.

1. **Foreign key approach*:***
Choose one of the relations—S, say—and include as a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S. Include all the simple attributes (or simple components of composite attributes) of the 1:1 relationship type R as attributes of S.

2. **Merged relation approach:**
An alternative mapping of a 1:1 relationship type is to merge the two entity types and the relationship into a single relation. This is possible when both participations are total, as this would indicate that the two tables will have the exact same number of tuples at all times.

3. **Cross-reference or relationship relation approach:**
The third option is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types. As we will see, this approach is required for binary M: N relationships. The relation R is called a relationship relation (or sometimes a lookup table), because each tuple in R represents a relationship instance that relates one tuple from S with one tuple from T. The relation R will include the primary key attributes of S and T as foreignkeys to S and T. The primary key of R will be one of the two foreign keys, and the other foreign key will be a unique key of R. The drawback is having an extra relation, and requiring an extra join operation when combining related tuples from the tables.

Step 4: Mapping of Binary 1: N Relationship Types.
For each regular binary 1: N relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R; we do this because each entity instance on the N-side is related to at most one entity instance on the 1-side of the relationship type. Include any simple attributes (or simple components of composite attributes) of the 1: N relationship type as attributes of S.

Step 5: Mapping of Binary M: N Relationship Types.

For each binary M: N relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S. Also include any simple attributes of the M: N relationship type (or simple components of composite attributes) as attributes of S.

Notice that we cannot represent an M: N relationship type by a single foreign key attribute in one of the participating relations (as we did for 1:1 or 1: N relationship types) because of the M: N cardinality ratio; we must create a separate relationship relation S.

Step 6: Mapping of Multivalued Attributes.

For each multivalued attribute A, create a new relation R. This relation R will include an attribute corresponding to A, plus the primary key attribute K—as a foreign key in R—of the relation that represents the entity type or relationship type that has A as a multivalued attribute. The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Step 7: Mapping of N-array Relationship Types.

For each n-array relationship type R, where n > 2, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types . Also include any simple attributes of the n-array relationship type (or simple components of composite attributes) as attributes of S. The primary key of S is usually a combination of all the foreign keys that reference the relations representing the participating entity types. However, if the cardinality constraints on any of the entity types E participating in R is 1, then the primary key of should not include the foreign key attribute that references therelation E ‘corresponding to E.

3.4 Schema Diagram

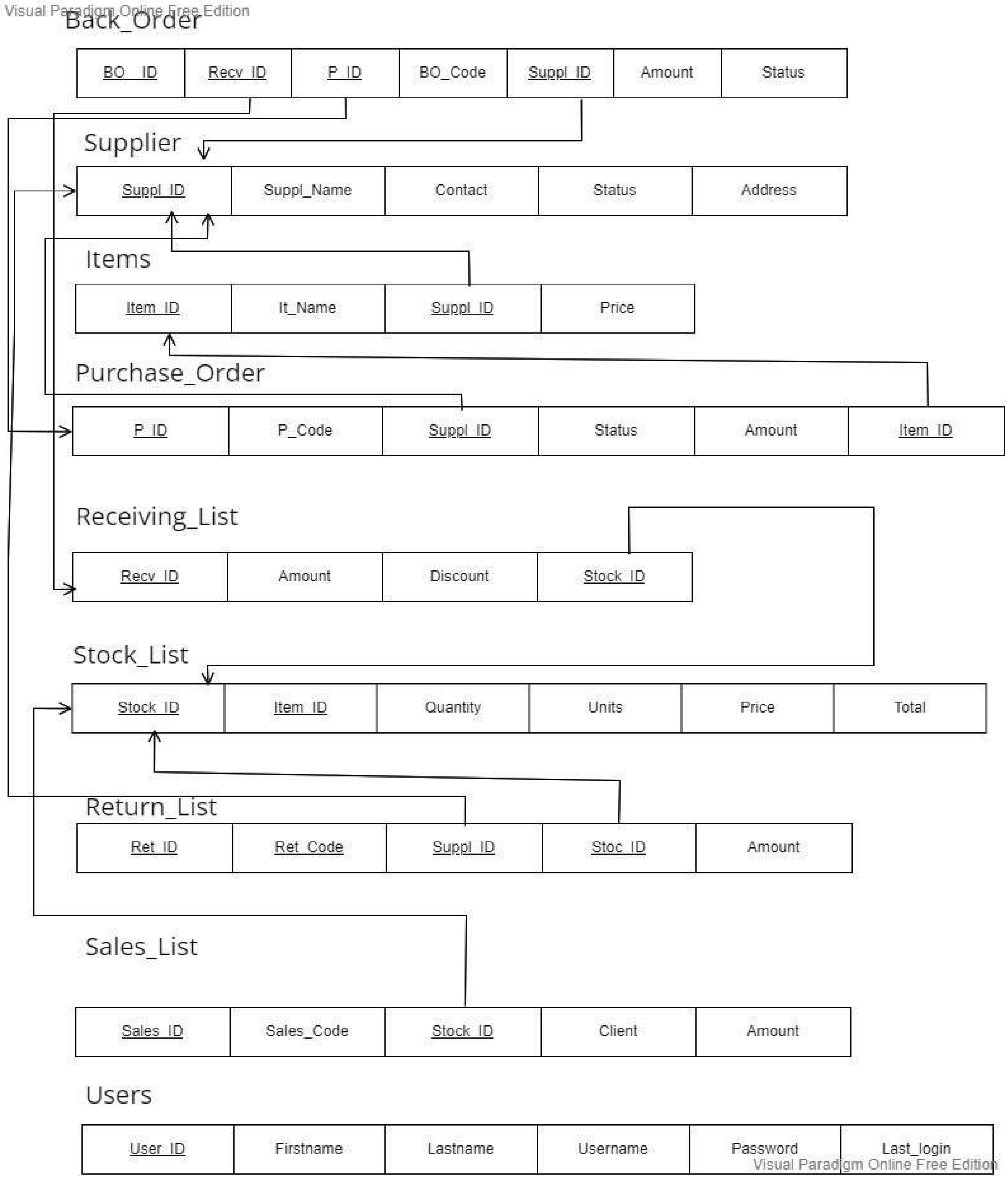


Fig 3.4 Schema Diagram

3.5 Database Description

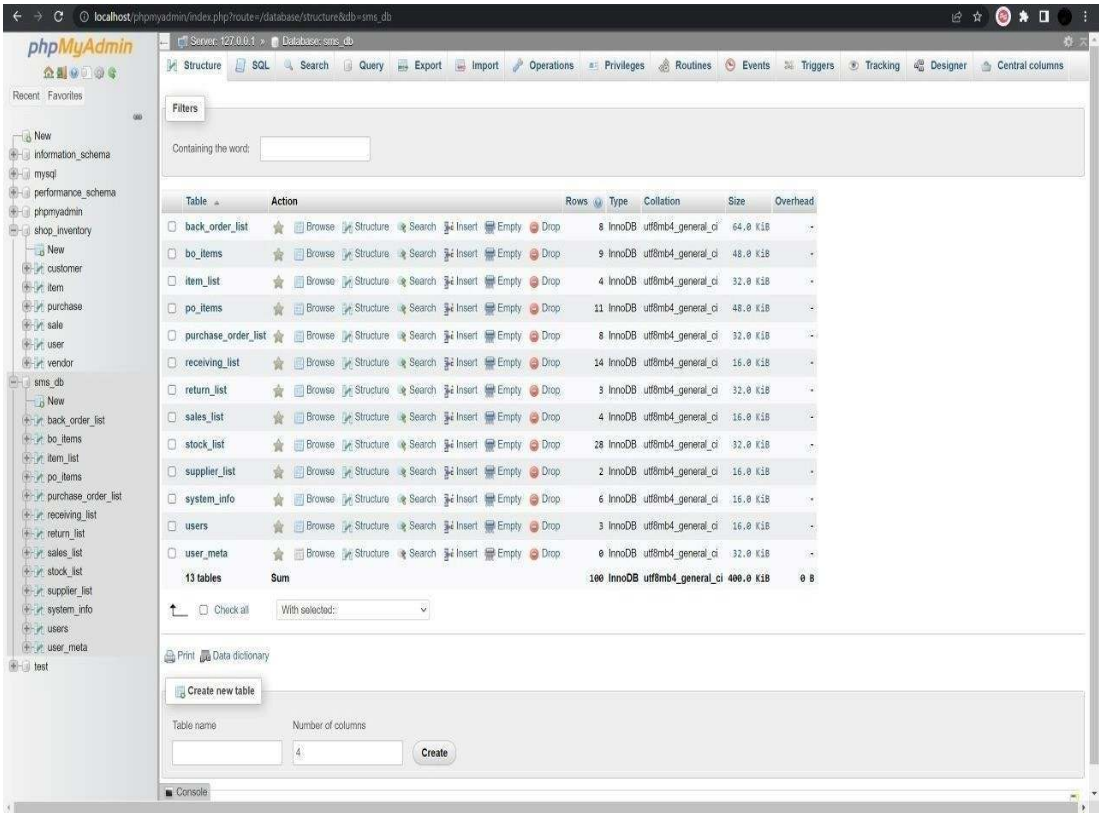


Table	Action	Rows	Type	Collation	Size	Overhead
back_order_list	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	64.0 KiB	-
bo_items	Browse Structure Search Insert Empty Drop	9	InnoDB	utf8mb4_general_ci	48.0 KiB	-
item_list	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	32.0 KiB	-
po_items	Browse Structure Search Insert Empty Drop	11	InnoDB	utf8mb4_general_ci	48.0 KiB	-
purchase_order_list	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8mb4_general_ci	32.0 KiB	-
receiving_list	Browse Structure Search Insert Empty Drop	14	InnoDB	utf8mb4_general_ci	16.0 KiB	-
return_list	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	32.0 KiB	-
sales_list	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8mb4_general_ci	16.0 KiB	-
stock_list	Browse Structure Search Insert Empty Drop	28	InnoDB	utf8mb4_general_ci	32.0 KiB	-
supplier_list	Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
system_info	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8mb4_general_ci	16.0 KiB	-
users	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
user_meta	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	32.0 KiB	-
13 tables	Sum	100	InnoDB	utf8mb4_general_ci	400.0 KiB	0 B

Table 3.5: Description of inventory database

CHAPTER 4

IMPLEMENTATION CODE

4.1 Connection code for front end and back end:

```
<?php define('DB_SERVER','localhost');define('DB_USER','root');define('DB_PASS'
,'');
define('DB_NAME',’sms');
$con = mysqli_connect(DB_SERVER,DB_USER,DB_PASS,DB_NAME);
// Check connection
if (mysqli_connect_errno())
{
echo "Failed to connect to MySQL: " . mysqli_connect_error();
}
?>
```

4.2 SQL Statements: Create commands:

```
CREATE TABLE `back_order_list` ( `id` int(30) NOT NULL,`receiving_id` int(30)
NOT NULL, `po_id` int(30) NOT NULL,`bo_code` varchar(50) NOT NULL,
`supplier_id` int(30)NOT NULL,`amount` float NOT NULL, `discount_perc` float NOT
NULL DEFAULT 0,`discount` float NOT NULL DEFAULT 0,`tax_perc` float NOT
NULL DEFAULT 0,`tax` float NOT NULL DEFAULT 0,`remarks` text DEFAULT
NULL,`status` tinyint(4) NOT NULL DEFAULT 0 COMMENT '0 = pending, 1 =
partially received, 2 =received', `date_created` datetime NOT NULL DEFAULT
current_timestamp(),`date_updated` datetime NOT NULL DEFAULT
current_timestamp() ONUPDATEcurrent_timestamp()) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4; CREATE TABLE `bo_items` (`bo_id` int(30) NOT
NULL,`item_id` int(30) NOT NULL,`quantity` int(30) NOT NULL,`price` float NOT
NULL DEFAULT 0, `unit` varchar(50) NOT NULL,`total` float NOT NULL
DEFAULT 0) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `item_list`, `id` int(30) NOT NULL,`name` text NOT
NULL,`description` text NOT NULL,`supplier_id` int(30) NOT NULL,`cost` float NOT
NULL DEFAULT 0`status` tinyint(1) NOT NULL DEFAULT 1,`date_created` datetime
NOT NULL DEFAULTcurrent_timestamp(),`date_updated` datetime NOT NULL
DEFAULT current_timestamp() ON UPDATE current_timestamp() ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4; CREATE TABLE `po_items` ( `po_id` int(30) NOT
NULL,`item_id` int(30) NOT NULL,`quantity` int(30) NOT NULL,`price` float NOT
NULL DEFAULT 0,`unit` varchar(50) NOT NULL,`total` float NOT NULL DEFAULT
0) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
CREATE TABLE `purchase_order_list` (`id` int(30) NOT NULL,`po_code` varchar(50)
NOTNULL,`supplier_id` int(30) NOT NULL,`amount` float NOT
NULL,`discount_perc` float NOT NULL DEFAULT 0,`discount` float NOT NULL
DEFAULT 0,`tax_perc` float NOT NULL DEFAULT 0,`tax` float NOT NULL
DEFAULT 0,`remarks` textNOT NULL, `status` tinyint(4) NOT NULL DEFAULT 0
COMMENT '0 = pending, 1 = partially received, 2 =received',`date_created` datetime
NOT NULL DEFAULT current_timestamp(),`date_updated` datetime NOT NULL
DEFAULT current_timestamp() ON UPDATEcurrent_timestamp()) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4; CREATE TABLE `receiving_list` (`id` int(30) NOT
NULL,`form_id` int(30) NOT NULL,`from_order` tinyint(1) NOT NULL DEFAULT 1
COMMENT '1=PO ,2 = BO',`amount` float NOT NULL DEFAULT 0, `discount_perc`
float NOT NULL DEFAULT 0,`discount` float NOT NULL DEFAULT 0,`tax_perc`
float NOT NULL DEFAULT 0,`tax` float NOT NULL DEFAULT 0,`stock_ids` text
DEFAULT NULL,`remarks` text DEFAULTNULL,`date_created` datetime NOT NULL
DEFAULT current_timestamp(), `date_updated`datetime NOT NULL DEFAULT
current_timestamp() ON UPDATE current_timestamp()) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4;
CREATE TABLE `return_list` (`id` int(30) NOT NULL,`return_code` varchar(50) NOT
NULL,`supplier_id` int(30) NOT NULL `amount` float NOT NULL DEFAULT
0,`remarks`text DEFAULT NULL,`stock_ids` text NOT NULL, `date_created` datetime
NOT NULL DEFAULT current_timestamp(),`date_updated` datetime NOT NULL
DEFAULT current_timestamp() ON UPDATE current_timestamp()) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4;
```

```
CREATE TABLE `sales_list` (`id` int(30) NOT NULL,`sales_code` varchar(50) NOT
NULL,`client` text DEFAULT NULL,`amount` float NOT NULL DEFAULT
0,`remarks` textDEFAULT NULL,`stock_ids` text NOT NULL,`date_created` datetime
NOT NULL DEFAULT current_timestamp(),`date_updated` datetime NOT NULL
DEFAULT current_timestamp() ON UPDATE current_timestamp()) ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4;
CREATE TABLE `stock_list` (`id` int(30) NOT NULL,`item_id` int(30) NOT NULL,
`quantity` int(30) NOT NULL,`unit` varchar(250) DEFAULT NULL,`price` float NOT
NULLDEFAULT 0,`total` float NOT NULL DEFAULT current_timestamp(),`type`
tinyint(1) NOT NULL DEFAULT 1 COMMENT '1=IN , 2=OUT',`date_created`
datetime NOT NULL DEFAULT current_timestamp()) ENGINE=InnoDB DEFAULT
CHARSET=utf8mb4;
 CREATE TABLE `supplier_list` (`id` int(30) NOT NULL,`name` text NOT
NULL,`address` text NOT NULL,`cperson` text NOT NULL,`contact` text NOT
NULL,`status` tinyint(1) NOT NULL        DEFAULT 1,`date_created`datetim
NOT  NULL DEFAULTcurrent_timestamp(),`date_updated` datetime NOT NULL
DEFAULT current_timestamp() ON UPDATE current_timestamp())ENGINE=InnoDB
DEFAULT CHARSET=utf8mb4; CREATETABLE `system_info`(`id`  int(30)   NOT
NULL,`meta_field`text NOTNULL,`meta_value` text NOT
NULL)ENGINE=InnoDBDEFAULTCHARSET=utf8mb4;
CREATETABLE`users`(`id`int(50) NOT NULL,`firstname`varchar(250) NOT
NULL,`middlename` text DEFAULT NULL,`lastname` varchar(250) Not
NULL,`username`text NOT NULL,`password` text NOT NULL,`avatar` text DEFAULT
NULL,`last_login` datetime DEFAULT NULL,`type` tinyint(1) NOT NULL DEFAULT
0,`date_added` datetimeNOT NULL DEFAULT current_timestamp(),`date_updated`
datetime DEFAULT NULL ON
UPDATEcurrent_timestamp())ENGINE=InnoDBDEFAULTCHARSET=utf8mb4;
CREATE TABLE `user_meta` (`user_id` int(30) NOT NULL,`meta_field` text
NOT NULL,`meta_value` text NOT NULL,`date_created` datetime NOT NULL
DEFAULT current_timestamp())ENGINE=InnoDBDEFAULTCHARSET=utf8mb;
```

Insert commands:

```
INSERT INTO `back_order_list` (`id`, `receiving_id`, `po_id`, `bo_code`,`supplier_id`,
`amount`, `discount_perc`, `discount`, `tax_perc`, `tax`, `remarks`, `status`date_created`,
`date_updated`) VALUES(1, 1, 1, 'BO-0001', 1, 40740, 3, 1125, 12, 4365, NULL, 1,
'2021-11-03 11:20:38', '2021-11-03 11:20:51'),(2, 2, 1, 'BO-0002', 1, 20370, 3, 562.5, 12,
2182.5,NULL, 2, '2021-11-03 11:20:51', '2021-11-03 11:21:00'),(3, 4, 2, 'BO-0003', 2,
42826, 5,2012.5, 12, 4588.5, NULL, 1, '2021-11-03 11:51:41', '2021-11-03 11:52:02'),(4,
5, 2, 'BO-0004', 2, 10640, 5, 500, 12, 1140, NULL, 2, '2021-11-03 11:52:02', '2021-11-
03 11:52:15');
INSERT INTO `bo_items` (`bo_id`, `item_id`, `quantity`, `price`, `unit`, `total`)
VALUES(1,1, 250, 150, 'pcs', 37500),(2, 1, 125, 150, 'pcs', 18750),(3, 2, 150, 200,
'Boxes', 30000),(3, 4,50, 205, 'pcs', 10250),(4, 2, 50, 200, 'Boxes', 10000);
INSERT INTO `item_list` (`id`, `name`, `description`, `supplier_id`, `cost`, `status`,
`date_created`, `date_updated`) VALUES(1, 'Item 101', 'Sample Only', 1, 150, 1, '2021-
11-0210:01:55', '2021-11-02 10:01:55'),(2, 'Item 102', 'Sample only', 2, 200, 1, '2021-11-
0210:02:12', '2021-11-02 10:02:12'),(3, 'Item 103', 'Sample', 1, 185, 1, '2021-11-02
10:02:27','2021-11-02 10:02:27'),(4, 'Item 104', 'Sample only', 2, 205, 1, '2021-11-02
10:02:47', '2021-11-02 10:02:47');
INSERT INTO `po_items` (`po_id`, `item_id`, `quantity`, `price`, `unit`, `total`)
VALUES(1,1, 500, 150, 'pcs', 75000),(2, 2, 300, 200, 'Boxes', 60000),(2, 4, 200, 205,
'pcs', 41000);
INSERT INTO `purchase_order_list` (`id`, `po_code`, `supplier_id`, `amount`,
`discount_perc`, `discount`, `tax_perc`, `tax`, `remarks`, `status`, `date_created`,
`date_updated`) VALUES(1, 'PO-0001', 1, 81480, 3, 2250, 12, 8730, 'Sample', 2, '2021-
11-0311:20:22', '2021-11-03 11:21:00'),(2, 'PO-0002', 2, 107464, 5, 5050, 12, 11514,
'Sample PO Only', 2, '2021-11-03 11:50:50', '2021-11-03 11:52:15');
```

```
INSERT INTO `receiving_list` (`id`, `form_id`, `from_order`, `amount`, `discount_perc`,
`discount`, `tax_perc`, `tax`, `stock_ids`, `remarks`, `date_created`, `date_updated`)
VALUES(1, 1, 1, 40740, 3, 1125, 12, 4365, '1', 'Sample', '202111-03 11:20:38', '2021-11-
0311:20:38'),(2, 1, 2, 20370, 3, 562.5, 12, 2182.5, '2', '', '2021-11-03 11:20:51', '2021-11-
0311:20:51'),(3, 2, 2, 20370, 3, 562.5, 12, 2182.5, '3', 'Success', '2021-11-03 11:21:00',
'2021-11-03 11:21:00'),(4, 2, 1, 64638, 5, 3037.5, 12, 6925.5, '4,5', 'Sample Receiving
(Partial)', '2021-11-03 11:51:41', '2021-11-03 11:51:41'),(5, 3, 2, 32186, 5, 1512.5, 12,
3448.5, '6,7', 'BOReceive (Partial)', '2021-11-03 11:52:02', '2021-11-03 11:52:02'),
(6, 4, 2, 10640, 5, 500, 12, 1140, '8', 'Sample Success', '2021-11-03 11:52:15', '2021-11-
0311:52:15');
INSERT INTO `return_list` (`id`, `return_code`, `supplier_id`, `amount`, `remarks`,
`stock_ids`, `date_created`,`date_updated`) VALUES(1, 'R-0001', 2, 3025, 'Sample
Issue','16,17', '2021-11-03 13:45:53', '2021-11-03 13:45:53');
INSERT INTO `sales_list` (`id`, `sales_code`, `client`, `amount`, `remarks`, `stock_ids`,
`date_created`,`date_updated`) VALUES(1, 'SALE-0001', 'John Smith', 7625, 'Sample
Remarks', '24,25,26', '2021-11-03 14:03:30', '2021-11-03 14:08:27');
INSERT INTO `stock_list` (`id`, `item_id`, `quantity`, `unit`, `price`, `total`, `type`,
`date_created`) VALUES(1, 1, 250, 'pcs', 150, 37500, 1, '2021-11-03 11:20:38'),(2, 1,
125,'pcs', 150, 18750, 1, '2021-11-03 11:20:51'),(3, 1, 125, 'pcs', 150, 18750, 1, '2021-11-
0311:21:00'),(4, 2, 150, 'Boxes', 200, 30000, 1, '2021-11-03 11:51:41'),(5, 4, 150, 'pcs',
205,30750, 1, '2021-11-03 11:51:41'),(6, 2, 100, 'Boxes', 200, 20000, 1, '2021-11-03
11:52:02'),(7, 4, 50, 'pcs', 205, 10250, 1, '2021-11-03 11:52:02'),(8, 2, 50, 'Boxes', 200,
10000, 1, '2021-11-03 11:52:15'),(16, 2, 10, 'pcs', 200, 2000, 2, '2021-11-03
13:45:53'),(17, 4, 5, 'boxes', 205,1025, 2, '2021-11-03 13:45:53'),(24, 1, 10, 'pcs', 150,
1500, 2, '2021-11-03 14:08:27'),(25, 2, 5,'pcs', 200, 1000, 2, '2021-11-03 14:08:27'),(26,
4, 25, 'boxes', 205, 5125, 2, '2021-11-03 14:08:27');
INSERT INTO `supplier_list` (`id`, `name`, `address`, `cperson`, `contact`, `status`,
`date_created`, `date_updated`) VALUES(1, 'Supplier 101', 'Sample Supplier Address
101','Supplier Staff 101', '09123456789', 1, '2021-11-02 09:36:19', '2021-11-02
09:36:19'),(2,'Supplier 102', 'Sample Address 102', 'Supplier Staff 102', '0987654332', 1,
'2021-11-02 09:36:54', '2021-11-02 09:36:54');
```

```
INSERT INTO `system_info` (`id`, `meta_field`, `meta_value`) VALUES(1, 'name',
'StockManagement System - PHP'),(6, 'short_name', 'SMS- PHP'),(11, 'logo',
'uploads/logo1635816671.png'),(13, 'user_avatar','uploads/user_avatar.jpg'),(14, 'cover',
'uploads/cover-1635816671.png'),(15, 'content', 'Array');
INSERT INTO `users` (`id`, `firstname`, `middlename`, `lastname`, `username`,
`password`,`avatar`, `last_login`, `type`, `date_added`, `date_updated`) VALUES1,
'Adminstrator', NULL,'Admin', 'admin', '0192023a7bbd73250516f069df18b500',
'uploads/avatar- 1.png?v=1635556826', NULL, 1, '2021-01-20 14:02:37','2021-10-30
09:20:26'),(10, 'John',NULL, 'Smith', 'jsmith', '39ce7e2a8573b41ce73b5ba41617f8f7',
'uploads/avatar- 10.png?v=1635920488', NULL, 2, '2021-11-03 14:21:28', '2021-11-03
14:21:28'),(11, 'Claire', NULL, 'Blake', 'cblake', 'cd74fae0a3adf459f73bbf187607ccea',
'uploads/avatar-11.png?v=1635920566', NULL, 1, '2021-11-03 14:22:46', '2021-11-03
14:22:46');
```

Alter commands:

```
ALTER TABLE `back_order_list`ADD PRIMARY KEY (`id`),ADD KEY `supplier_id`
(`supplier_id`),ADD KEY `po_id` (`po_id`),ADD KEY `receiving_id` (`receiving_id`);
ALTER TABLE `bo_items`ADD KEY `item_id` (`item_id`),ADD KEY `bo_id`
(`bo_id`);ALTER TABLE `item_list`ADD PRIMARY KEY (`id`),ADD KEY
`supplier_id` (`supplier_id`);
ALTER TABLE `po_items`ADD KEY `po_id` (`po_id`),ADD KEY `item_id`
(`item_id`); ALTER TABLE `purchase_order_list`ADD PRIMARY KEY (`id`),ADD
KEY `supplier_id`(`supplier_id`);
ALTER TABLE `receiving_list`ADD PRIMARY KEY (`id`);
ALTER TABLE `return_list`ADD PRIMARY KEY (`id`),ADD KEY `supplier_id`
(`supplier_id`);
ALTER TABLE `sales_list`ADD PRIMARY KEY (`id`);
ALTER TABLE `stock_list`ADD PRIMARY KEY (`id`),ADD KEY `item_id`
(`item_id`);ALTER TABLE `supplier_list`ADD PRIMARY KEY (`id`);
ALTER TABLE `system_info`ADD PRIMARY KEY(`id`);
ALTER TABLE `users`ADD PRIMARY KEY (`id`);
ALTER TABLE `user_meta`ADD KEY `user_id` (`user_id`);
ALTER TABLE `back_order_list`MODIFY `id` int(30) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=5;
ALTER TABLE `item_list`MODIFY `id` int(30) NOT NULLAUTO_INCREMENT,
AUTO_INCREMENT=6;
ALTER TABLE `purchase_order_list`MODIFY `id` int(30) NOT NULL
AUTO_INCREMENT, AUTO_INCREMENT=3;
ALTER TABLE `receiving_list`MODIFY `id` int(30) NOT
NULLAUTO_INCREMENT,AUTO_INCREMENT=7;
ALTER TABLE `return_list`MODIFY `id` int(30) NOT NULLAUTO_INCREMENT,
AUTO_INCREMENT=2;
ALTER TABLE `sales_list`MODIFY `id` int(30) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=2;
ALTER TABLE `stock_list`MODIFY `id` int(30) NOT NULL AUTO_INCREMENT,
AUTO_INCREMENT=27;
ALTER TABLE `supplier_list`MODIFY `id` int(30) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=4;
ALTER TABLE `system_info`MODIFY `id` int(30) NOT NULL
AUTO_INCREMENT,AUTO_INCREMENT=16;
ALTER TABLE `users`MODIFY `id` int(50) NOT NULLAUTO_INCREMENT,
AUTO_INCREMENT=12;
ALTER TABLE `back_order_list`ADD CONSTRAINT `back_order_list_ibfk_1`
FOREIGNKEY (`supplier_id`) REFERENCES `supplier_list` (`id`) ON DELETE
CASCADE,ADD CONSTRAINT `back_order_list_ibfk_2` FOREIGN KEY (`po_id`)
REFERENCES `purchase_order_list` (`id`) ON DELETE CASCADE,ADD
CONSTRAINT `back_order_list_ibfk_3` FOREIGN KEY (`receiving_id`)
REFERENCES `receiving_list`(`id`) ON DELETE CASCADE;
```

```
ALTER TABLE `bo_items`ADD CONSTRAINT `bo_items_ibfk_1` FOREIGN KEY
(`item_id`) REFERENCES `item_list` (`id`) ON DELETE CASCADE,ADD
CONSTRAINT `bo_items_ibfk_2` FOREIGN KEY (`bo_id`) REFERENCES
`back_order_list` (`id`) ONDELETE CASCADE;
ALTER TABLE `item_list`ADD CONSTRAINT `item_list_ibfk_1` FOREIGN KEY
(`supplier_id`) REFERENCES `supplier_list` (`id`) ON DELETE CASCADE; ALTER
TABLE `po_items`ADD CONSTRAINT `po_items_ibfk_1` FOREIGN KEY(`po_id`)
REFERENCES `purchase_order_list` (`id`) ON DELETE CASCADE,ADD
CONSTRAINT `po_items_ibfk_2` FOREIGN KEY (`item_id`) REFERENCES
`item_list`(`id`) ON DELETE CASCADE;
ALTER TABLE `purchase_order_list`ADD CONSTRAINT
`purchase_order_list_ibfk_1`FOREIGN KEY (`supplier_id`) REFERENCES
`supplier_list` (`id`) ON DELETE CASCADE;
ALTER TABLE `return_list`ADD CONSTRAINT `return_list_ibfk_1` FOREIGN KEY
(`supplier_id`) REFERENCES `supplier_list` (`id`) ON DELETE CASCADE;
ALTER TABLE `stock_list`ADD CONSTRAINT `stock_list_ibfk_1` FOREIGN KEY
(`item_id`) REFERENCES `item_list` (`id`) ON DELETE CASCADE;
```

CHAPTER 5

SNAPSHOTS

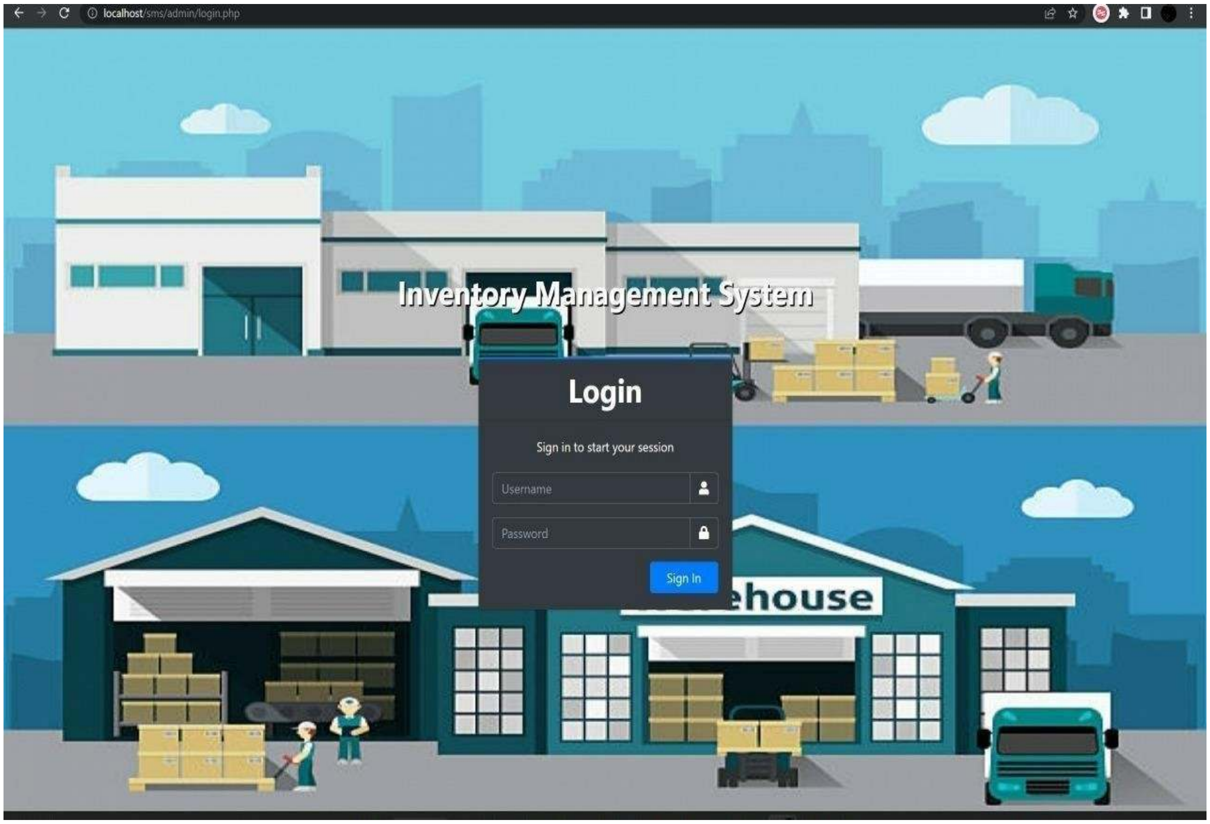
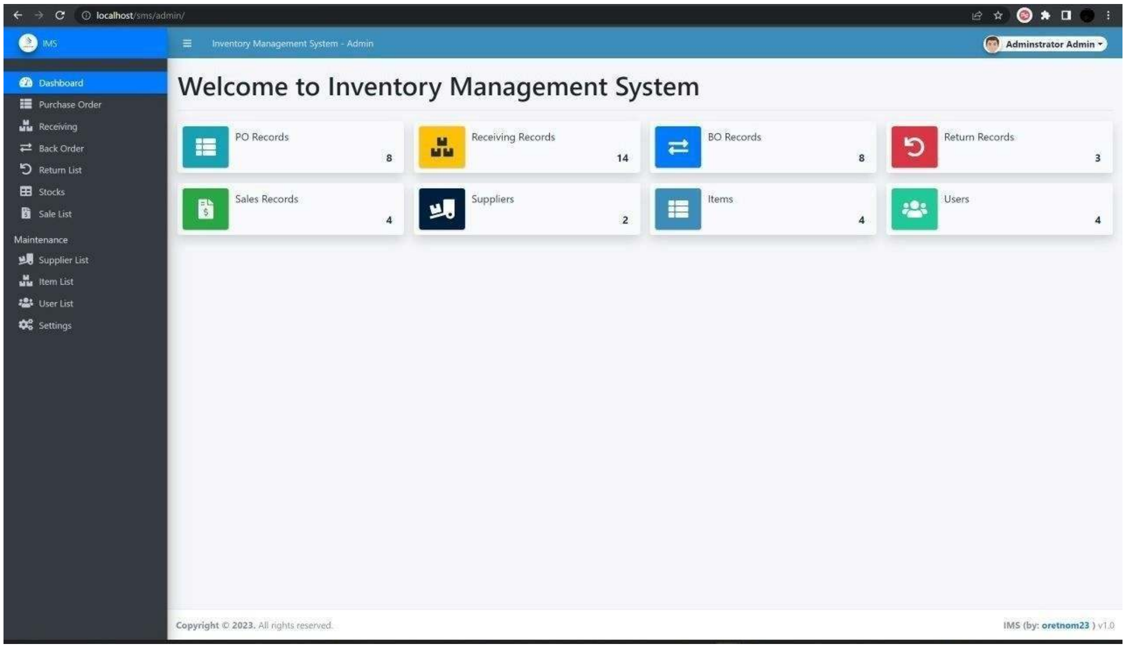

localhost/sms/admin/login.php
Inventory Management System
Login
Sign in to start your session
Username
Password
Sign In
ehouse

Fig 5.1: Login page



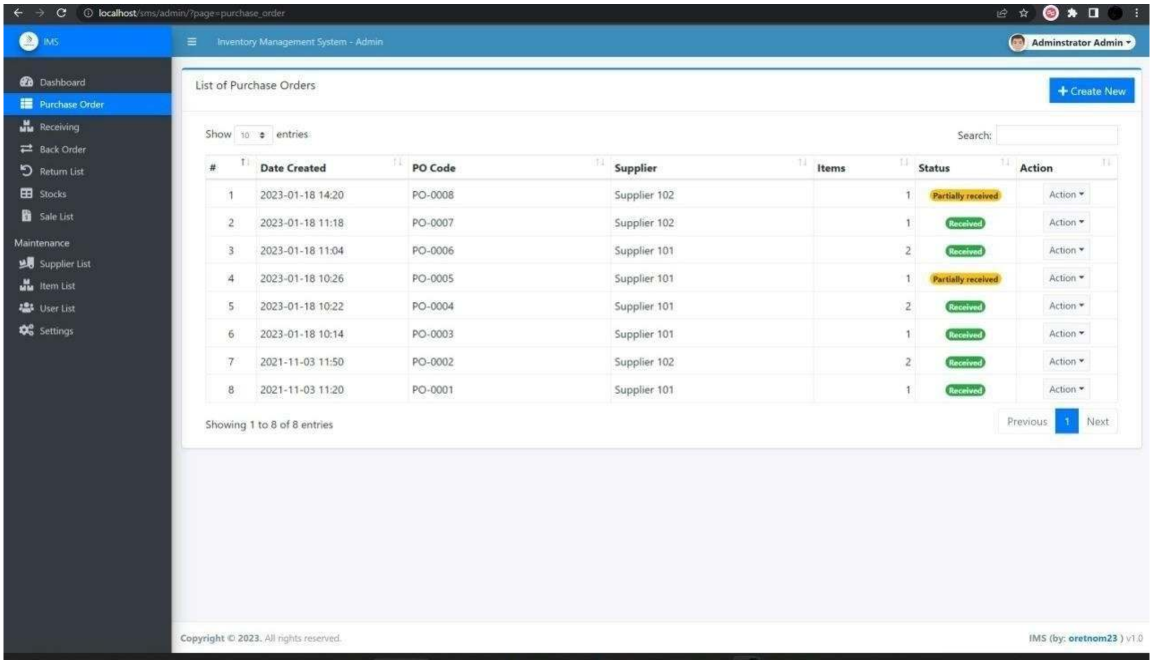
Welcome to Inventory Management System

PO Records	8
Receiving Records	14
BO Records	8
Return Records	3
Sales Records	4
Suppliers	2
Items	4
Users	4

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.2: Dashboard Page



List of Purchase Orders

Show 10 entries

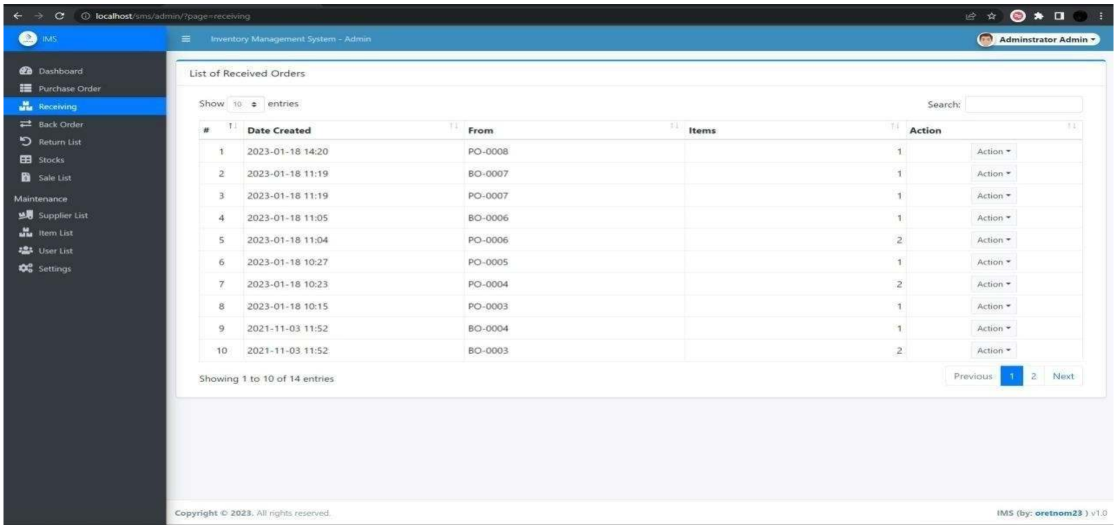
#	Date Created	PO Code	Supplier	Items	Status	Action
1	2023-01-18 14:20	PO-0008	Supplier 102	1	Partially received	Action
2	2023-01-18 11:18	PO-0007	Supplier 102	1	Received	Action
3	2023-01-18 11:04	PO-0006	Supplier 101	2	Received	Action
4	2023-01-18 10:26	PO-0005	Supplier 101	1	Partially received	Action
5	2023-01-18 10:22	PO-0004	Supplier 101	2	Received	Action
6	2023-01-18 10:14	PO-0003	Supplier 101	1	Received	Action
7	2021-11-03 11:50	PO-0002	Supplier 102	2	Received	Action
8	2021-11-03 11:20	PO-0001	Supplier 101	1	Received	Action

Showing 1 to 8 of 8 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.3: Purchase Order Page



List of Received Orders

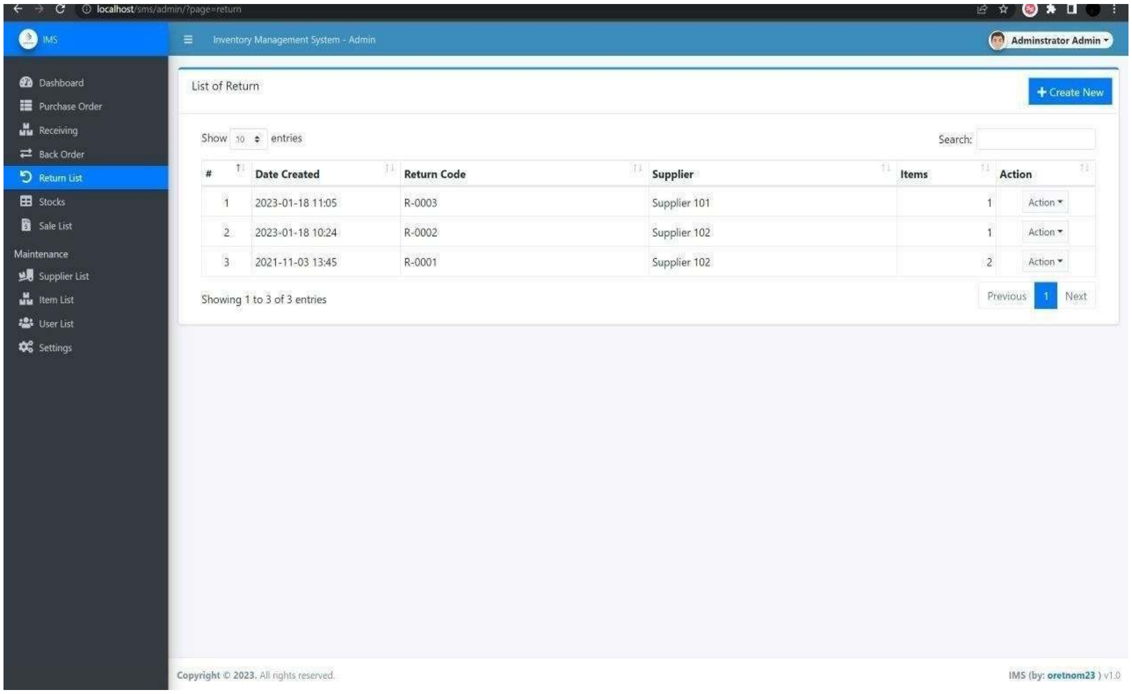
#	Date Created	From	Items	Action
1	2023-01-18 14:20	PO-0008	1	Action
2	2023-01-18 11:19	BO-0007	1	Action
3	2023-01-18 11:19	PO-0007	1	Action
4	2023-01-18 11:05	BO-0006	1	Action
5	2023-01-18 11:04	PO-0006	2	Action
6	2023-01-18 10:27	PO-0005	1	Action
7	2023-01-18 10:23	PO-0004	2	Action
8	2023-01-18 10:15	PO-0003	1	Action
9	2021-11-03 11:52	BO-0004	1	Action
10	2021-11-03 11:52	BO-0003	2	Action

Showing 1 to 10 of 14 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.4: Receiving List Page



List of Return

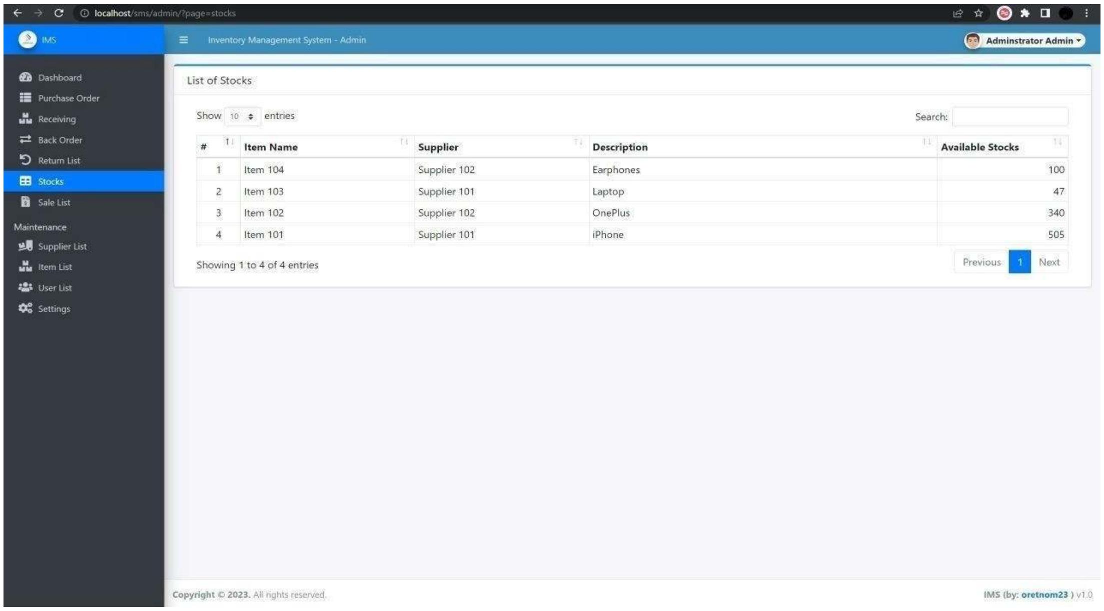
#	Date Created	Return Code	Supplier	Items	Action
1	2023-01-18 11:05	R-0003	Supplier 101	1	Action
2	2023-01-18 10:24	R-0002	Supplier 102	1	Action
3	2021-11-03 13:45	R-0001	Supplier 102	2	Action

Showing 1 to 3 of 3 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.5: Return List Page

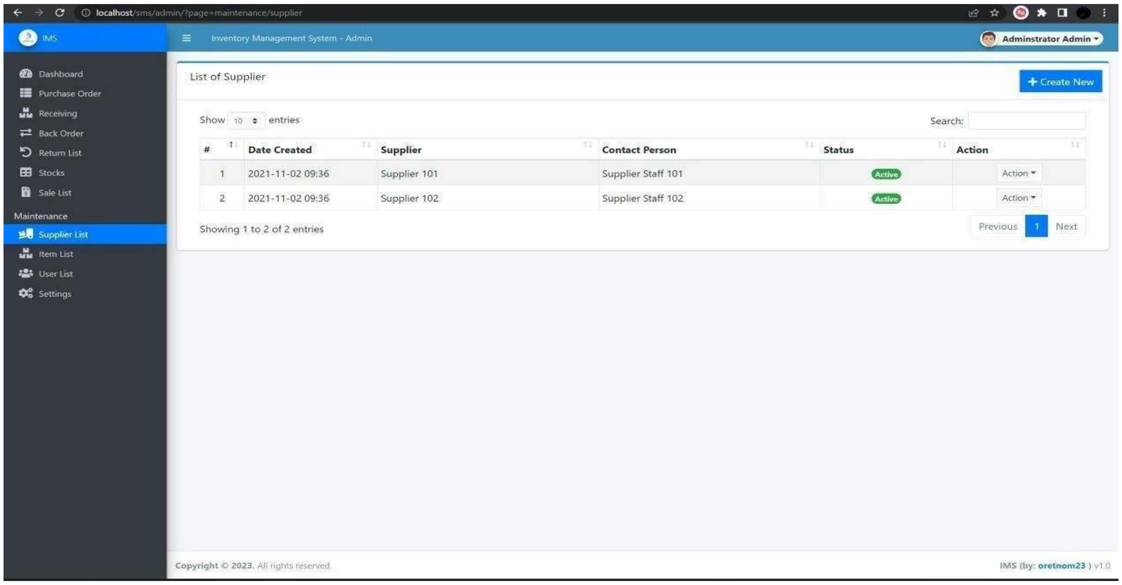


List of Stocks

#	Item Name	Supplier	Description	Available Stocks
1	Item 104	Supplier 102	Earphones	100
2	Item 103	Supplier 101	Laptop	47
3	Item 102	Supplier 102	OnePlus	340
4	Item 101	Supplier 101	iPhone	505

Showing 1 to 4 of 4 entries

Fig 5.6: Stock List Page

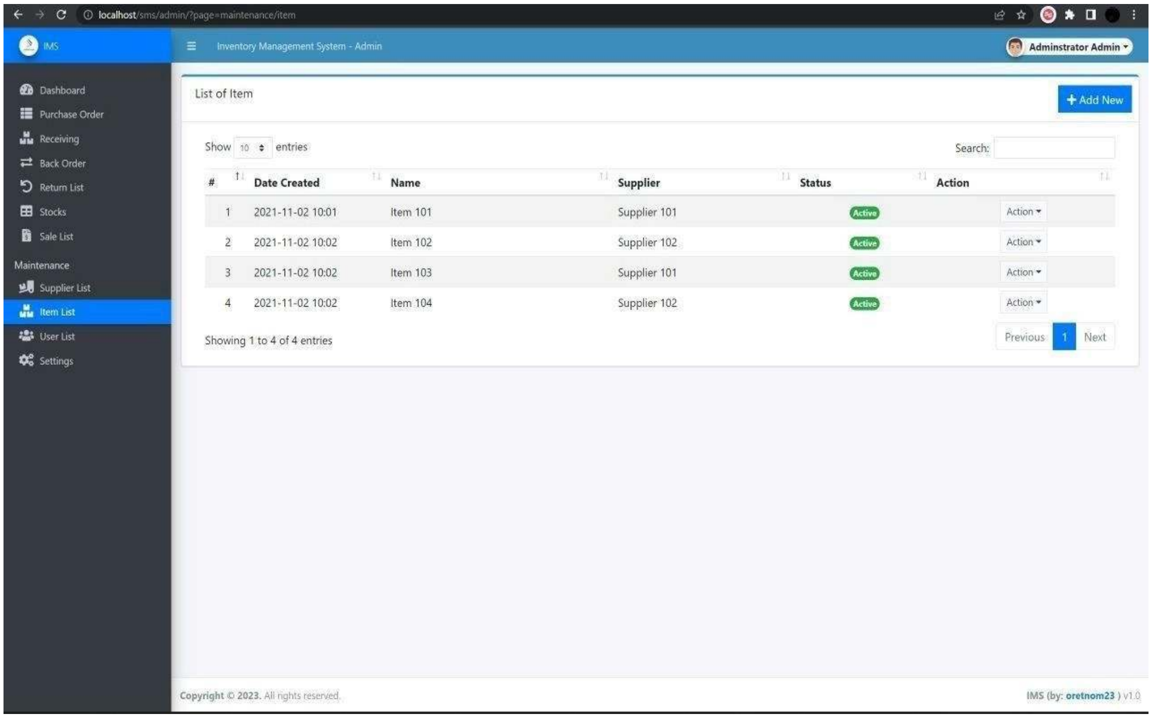


List of Supplier

#	Date Created	Supplier	Contact Person	Status	Action
1	2021-11-02 09:36	Supplier 101	Supplier Staff 101	Active	Action
2	2021-11-02 09:36	Supplier 102	Supplier Staff 102	Active	Action

Showing 1 to 2 of 2 entries

Fig 5.7: Supplier List Page



List of Item

Show 10 entries

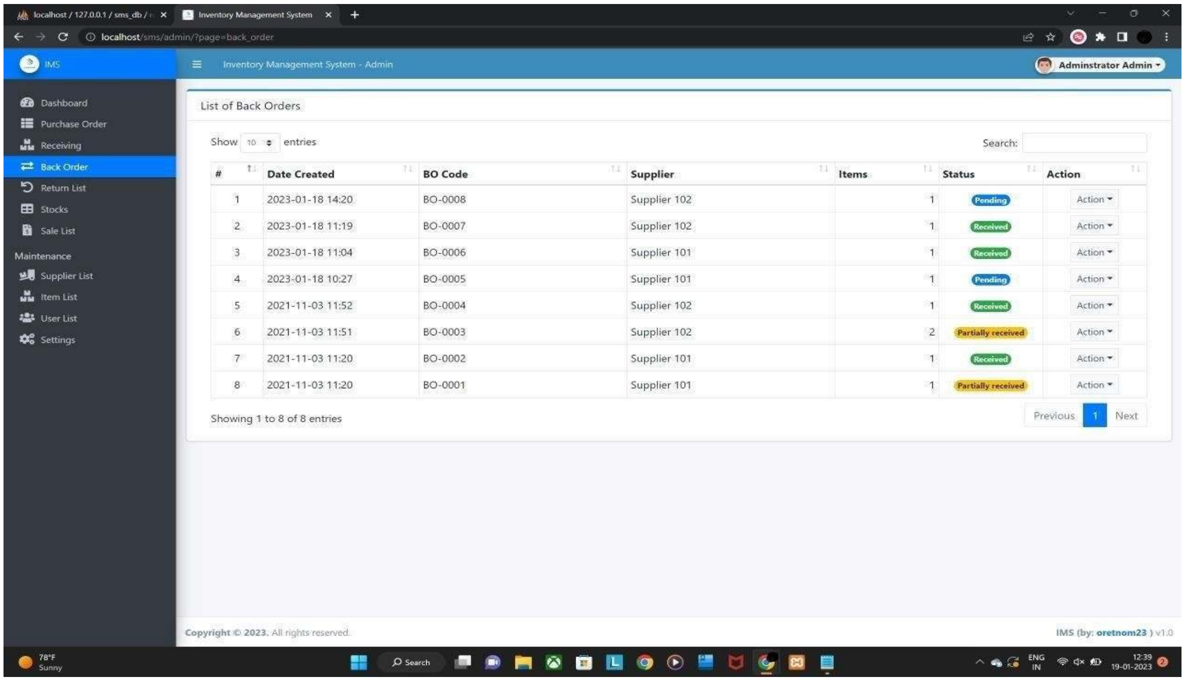
#	Date Created	Name	Supplier	Status	Action
1	2021-11-02 10:01	Item 101	Supplier 101	Active	Action
2	2021-11-02 10:02	Item 102	Supplier 102	Active	Action
3	2021-11-02 10:02	Item 103	Supplier 101	Active	Action
4	2021-11-02 10:02	Item 104	Supplier 102	Active	Action

Showing 1 to 4 of 4 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.8: Item List Page



List of Back Orders

Show 10 entries

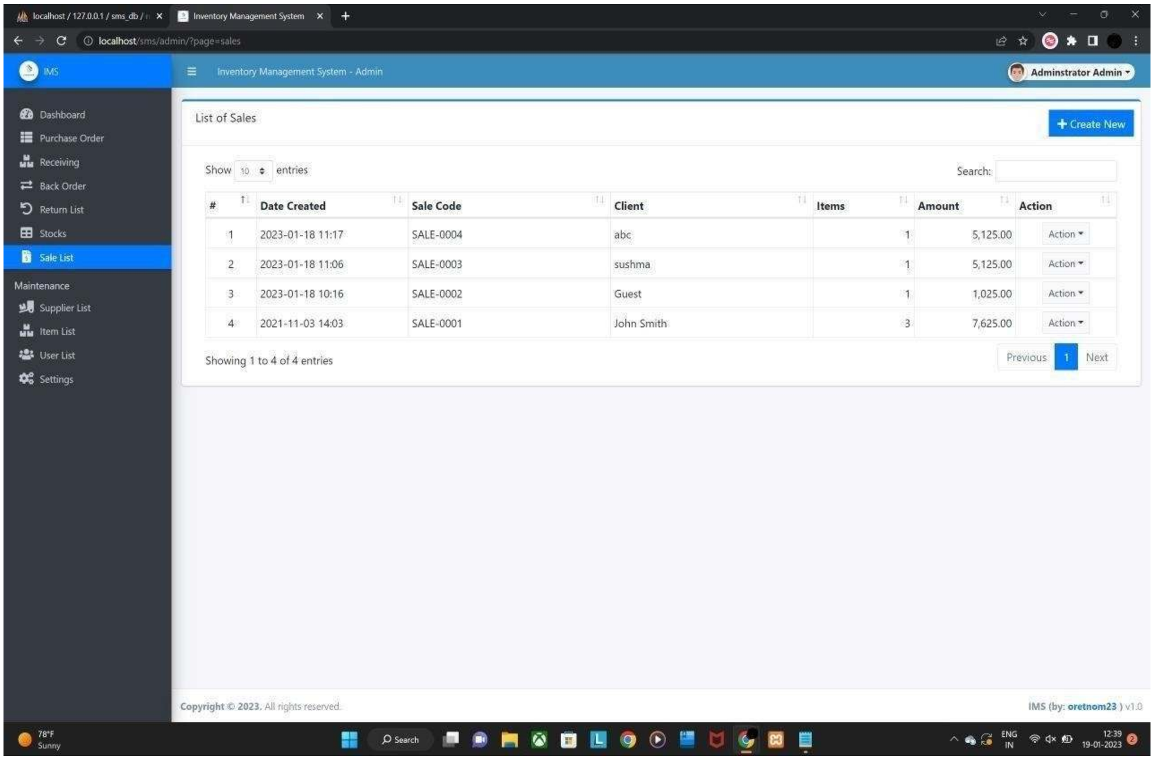
#	Date Created	BO Code	Supplier	Items	Status	Action
1	2023-01-18 14:20	BO-0008	Supplier 102	1	Pending	Action
2	2023-01-18 11:19	BO-0007	Supplier 102	1	Received	Action
3	2023-01-18 11:04	BO-0006	Supplier 101	1	Received	Action
4	2023-01-18 10:27	BO-0005	Supplier 101	1	Pending	Action
5	2021-11-03 11:52	BO-0004	Supplier 102	1	Received	Action
6	2021-11-03 11:51	BO-0003	Supplier 102	2	Partially received	Action
7	2021-11-03 11:20	BO-0002	Supplier 101	1	Received	Action
8	2021-11-03 11:20	BO-0001	Supplier 101	1	Partially received	Action

Showing 1 to 8 of 8 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.9: BackOrder Page



List of Sales

Show 10 entries

Search:

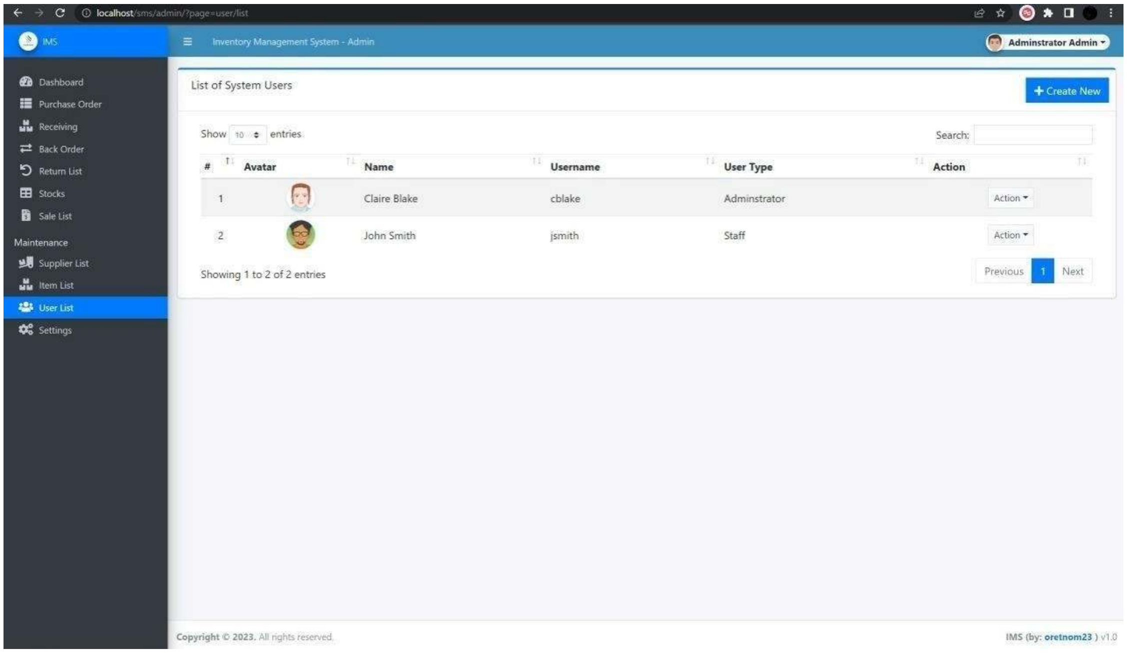
#	Date Created	Sale Code	Client	Items	Amount	Action
1	2023-01-18 11:17	SALE-0004	abc	1	5,125.00	Action
2	2023-01-18 11:06	SALE-0003	sushma	1	5,125.00	Action
3	2023-01-18 10:16	SALE-0002	Guest	1	1,025.00	Action
4	2021-11-03 14:03	SALE-0001	John Smith	3	7,625.00	Action

Showing 1 to 4 of 4 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.10: Sales List Page



List of System Users

Show 10 entries

Search:

#	Avatar	Name	Username	User Type	Action
1		Claire Blake	cblake	Adminstrator	Action
2		John Smith	jsmith	Staff	Action

Showing 1 to 2 of 2 entries

Copyright © 2023. All rights reserved.

IMS (by: oretnom23) v1.0

Fig 5.11: User List Page

CONCLUSION

To conclude, Inventory Management System is a simple desktop based application basically suitable for small organization. It has every basic items which are used for the small organization. Our team is successful in making the application where we can update, insert and delete the item as per the requirement. This application also provides a simple report on daily basis to know the daily sales and purchase details. This application matches for small organization where there small limited if godown.

Through it has some limitations, our team strongly believes that the implementation of this system will surely benefit the organization.

REFERENCES

1. Fundamentals of database systems, Ramez Elmasri and S B Navathe, 7th Edition, 2017,Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGrawHill.
3. Coronel, Morris, and Rob, Database Principles Fundamentals of Design, ImplementationandManagement, Cengage Learning 2012.\
4. Silberschatz Korth and Sudharshan, Database System Concepts, 6th Edition, McGraw Hill,2013.
5. https://www.w3schools.com/
6. https://www.google.com/
7. https://www.tutorialspoint.com/dbms/index.htm