

Project Synopsis

Introduction:

The College Inventory Management System is a web-based application developed using PHP and MySQL. This system is designed to efficiently manage and track the inventory items of various departments within a college, including lab equipment, furniture, stationery, and other assets. It helps reduce manual work, enhances accuracy, and improves overall inventory monitoring.

Objective:

The primary objective of this project is to automate the process of managing inventory items in a college. The system will allow administrators to add, update, delete, and track items easily and generate reports for analysis.

Scope of the Project:

- Maintain records of all inventory items across departments.
- Monitor stock levels and item movements.
- Assign items to specific rooms/labs/staff.
- Generate real-time reports for management.
- User roles such as Admin and Department Users.

Modules:

- Login/Authentication Module

- Admin and user login
- Password management
- Inventory Management Module
 - Add/edit/delete/view inventory items
 - Categorization by department and type
- Stock Management Module
 - Track quantity, availability, and issued/returned items
- User Management Module
 - Manage department users and access levels

Reports Module

Generate reports (daily, weekly, monthly)

Export to PDF/Excel

Technology Used:

- Front-end: HTML, CSS, JavaScript, Bootstrap

- Back-end: PHP
- Database: MySQL
- Server: Apache (XAMPP/WAMP)

Advantages:

- Streamlined inventory tracking
- Real-time monitoring and updates
- Reduced chances of manual errors
- Improved resource utilization

Future Enhancements:

- Barcode/RFID scanning for faster processing
- Mobile app integration
- Alert system for low stock
- Asset depreciation tracking

Conclusion:

The College Inventory Management System will serve as a comprehensive solution for managing college inventory efficiently. It will bring transparency, accuracy, and ease to the process, thereby supporting administrative staff in better resource handling.

PHP FRAMEWORK

2.1 PHP OVERVIEW

PHP is an embedded scripting language that is excellent for creating dynamic Web sites based on database content or different characteristics of browsers. It is available when you have a Departmental (Web Central) Publishing account, a Faculty Publishing account, Student Organization Publishing account, or if you use Custom Web Publishing.

What is PHP?

1. PHP stands for **PHP: Hypertext Pre-processor**
2. PHP is a server-side scripting language, like ASP
3. PHP scripts are executed on the server
4. PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
5. PHP is open source software
6. PHP is free to download and use

What is a PHP File?

1. PHP files can contain text, HTML tags and scripts
2. PHP files are returned to the browser as plain HTML
3. PHP files have a file extension of ".php", ".php3", or ".phtml"

2.2 PHP Features

1. Allows you to build templates to ease site maintenance
2. Serve different content to users based on their browser, IP address, date and time, or numerous other characteristics
3. Enables connections with databases such as MySQL
4. Build discussion forums or Web-based email programs
5. Read and process XML

2.3 WHY WE ARE USING PHP?

1. FREE OF COST

PHP is open source and is developed and updated by a community of developers from around the globe. Therefore, all its components are free to use and distribute.

2. CAPABLE

It can be used to design any type of website and can handle websites with a lot of traffic. Face book, Twitter, Wikipedia and many other very widely visited websites use it as their framework. And because it is server-side scripting, it can do anything that other CGI programs can do.

3. EASY

It has a readable and easily understandable syntax. Its code is embedded in the HTML source code and it is based on C/C++. Therefore, it is very familiar and programmers are very comfortable coding with it.

4. PLATFORM INDEPENDENT

It can be run on all major operating systems like Linux, UNIX, Mac OS and Windows.

5. SUPPORTS ALL MAJOR WEB SERVERS

It supports all major web servers like Apache, Microsoft IIS, Netscape, personal webserver, I Planet server, etc.

6. SUPPORTS ALL MAJOR DATABASES

IT supports all major databases including MySQL, dBase, IBM DB2, Inter Base, Front Base, ODBC, PostgreSQL, SQLite, etc.

7. FASTER DEVELOPMENTS

It uses its own memory space and thus decreases the loading time and workload from the server. The processing speed is fast and web applications like Ecommerce, CRM, CMS and Forums are also developed faster by it.

8. SECURE

It has multiple layers of security to prevent threats and malicious attacks.

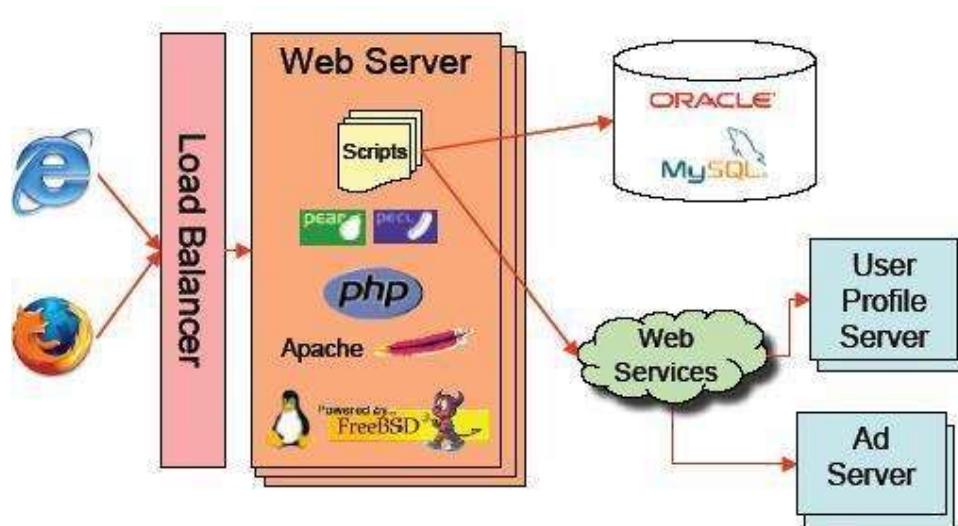
9. LARGE COMMUNITIES

It has a large community of developers who regular and timely updates tutorials, documentation, online help and FAQs.

10. PROVEN AND TRUSTED

It is being used since close to two decades now since its inception in 1995. It is trusted by thousands of websites and developers and the list is increasing day by day. It has also proven its capability and versatility by developing and maintaining some of the most highly visited and popular websites.

PHP ARCHITECTURE



The query cache plugin is implemented as a PHP extension. It is written in C and operates under the hood of PHP. During the startup of the PHP interpreter, it gets registered as [mysqlnd](#) plugin to replace selected mysqlnd C methods. Hereby, it can change the behaviour of any PHP MySQL extension ([mysqli](#), [PDO_MYSQL](#), [MySQL](#)) compiled to use the mysqlnd library without changing the extensions API. This makes the plugin be compatible with each and every PHP MySQL application. Because existing APIs are not changed, it is almost

transparent to use. Please, see the mysqlnd plugin API description for a discussion of the advantages of the plugin architecture and a comparison with proxy based solutions.

2.5 WHAT IS MYSQL DATABASE?

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is most noted for its quick processing, proven reliability, ease and flexibility of use.

2.6 FEATURES OF MYSQL

- 1. Relational Database System:** Like almost all other database systems on the market, MySQL is a relational database system.
- 2. Client/Server Architecture:** MySQL is a client/server system. There is a database server (MySQL) and arbitrarily many clients (application programs), which communicate with the server; that is, they query data, save changes, etc. The clients can run on the same computer as the server or on another computer (communication via a local network or the Internet).

2.7 ADAVANTAGES OF MYSQL

- 1. It is easy to use:** While a basic knowledge of SQL is required—and most relational databases require the same knowledge—MySQL is very easy to use. With only a few simple SQL statements, you can build and interact with MySQL.
- 2. It is secure:** MySQL includes solid data security layers that protect sensitive data from intruders. Rights can be set to allow some or all privileges to individuals. Passwords are encrypted.

3. It is inexpensive: MySQL is included for free with NetWare® 6.5 and available by free download from [MySQL Web site](#).

4. It is fast: In the interest of speed, MySQL designers made the decision to offer fewer features than other major database competitors, such as Sybase and Oracle. However, despite having fewer features than the other commercial database products, MySQL still offers all of the features required by most database developers.

5. It is scalable: MySQL can handle almost any amount of data, up to as much as 50 million rows or more. The default file size limit is about 4 GB. However, you can increase this number to a theoretical limit of 8 TB of data.

6. It runs on many operating systems: MySQL runs on many operating systems, including Novell NetWare, Windows* Linux*, many varieties of UNIX* (such as Sun* Solaris*, AIX, and DEC* UNIX), OS/2, FreeBSD*, and others.

7. It manages memory very well: MySQL server has been thoroughly tested to prevent memory leaks.

2.8 About JavaScript

JavaScript ('dʒa:va:, skript/; JS) is a dynamic computer programming language.

It is most commonly used as part of Web browsers, whose implementations allow client-side **scripts** to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.

2.9 HTML:

HTML stands for hypertext markup Language. It is very useful to make web pages and very easy to learn. Hypertext Markup file is a text file containing small markup tags. These marks up tags tell the browser how to display a web page. It has two types of extensions one is .html and second is .htm but both are used for html web pages. For hypertext markup

language you can use the simple text editor for example; use notepad for writing your HTML code in the windows. If you are using Mac you can use simple text editor.

HTML uses approach of what you see is what you get. You can also use to write tags other software that is FrontPage and Dreamweaver. In HTML character are surrounded by the tags. HTML tags come in pair. The beauty of this language is that it is not case sensitive. Every web page need HTML with it you cannot make the good web pages. And it is the base for every web page and used to display the text in the web pages there are some other latest version of HTML like DHTML which stands for dynamic html and is used to make the web pages more interactive.

Features of HTML:

- It is simple to understand and implement.
- HTML constructs are very easy to comprehend, and can be used effectively by anybody.
- HTML syntax is a worldwide standard.
- The methodology used by HTML to markup information is independent of its representation on a particular hardware or software architecture.
- It is not a programming language.
- And it is also not a description language.

User characteristics:

1. User must have computer knowledge
2. To view the details of the all the orders and all he/she should be registered as admin.

Principal actors

Actors on the screen

1. Admin

End Users

1. Website visitors
2. Registered Users

General Constraints

1. For full working of our software requires Internet connection.
2. Admin module is only for the use of Admin.
3. Customer module is for only Registered Users.

Assumptions and Dependencies

1. Full working of website is dependent on the availability of Internet connection.
2. It is multi user website.

Product prospective:

The website is aimed at providing web application to shop. Registered users can easily place the orders to the Shop; they can purchase the products displayed in the Website.

SYSTEM REQUIREMENTS AND ANALYSIS

4.1 Specification

Specification is used to specify the requirement for the initial implementation of system and update in the future.

Tools and Platforms used:

- Back end-MySQL5.6.12
- Front end-PHP5.4.16

Software requirements:

- Platforms-PHP
- Database- My SQL (Back end)
- Tools- Java scrip, HTML, AJAX
- Operating System-Windows 8
- Designing Tool - Dreamweaver

Hardware requirement:

- Processor-p-VI and onwards
- RAM- 128 MB onwards
- Hard disk space-minimum

4.2 System Analysis

Database Storage

Proposed Database is intended to store, retrieve, update, and manipulate details of User details database which include tables.

- Admin Account.
- User Account.

Design Constraints

- Complete validation has been done no mandatory fields was to be left unfilled, if left unfilled then appropriate alert message will be displayed.
- Data in the database cannot be modified until and unless you are the proper authenticator.
- Efficient error handling capacities have been provided.
- The Admin can view the records whenever and wherever required as the system is easily available on the net.

Data Flow Diagram

Graphical description of system's data & how the processes transform the data is known as data flow diagram. It is also known as Bubble chart. DFD modules, system using external entities from which data flow to a process, which transforms the data, and create output data transforms which go to. Other processes or external entities like files. The main metric of DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used and where the result flows.

5.1 Notations Used In Data Flow Diagram:

Functional Processing: It is represented by an oval. The processing or main transactions are specified by this notation.



Data Flow: It is represented by an arrow line and name of the data is specified by the side of the line as label. This is used for data movement.



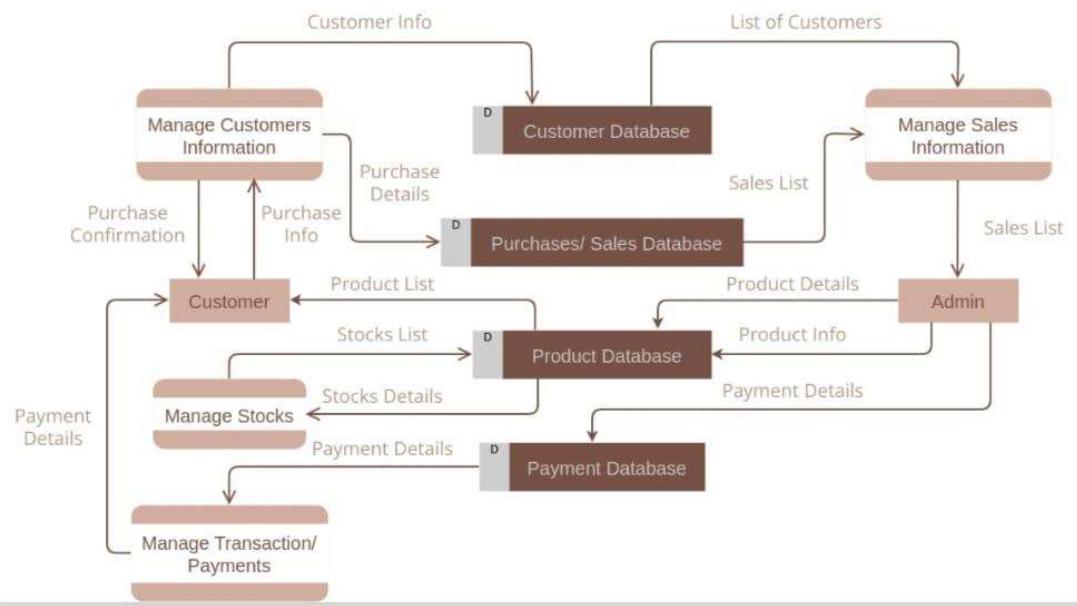
Data Store: It is represented by one open-end rectangle. The databases used in the system are specified by this notation.



Source or sink: It is represented by one open-end rectangle. It is used for specifying from where data comes and where it reaches.



DATA FLOW MODEL DAIGRAMA



ER MODEL

The E-R Model:

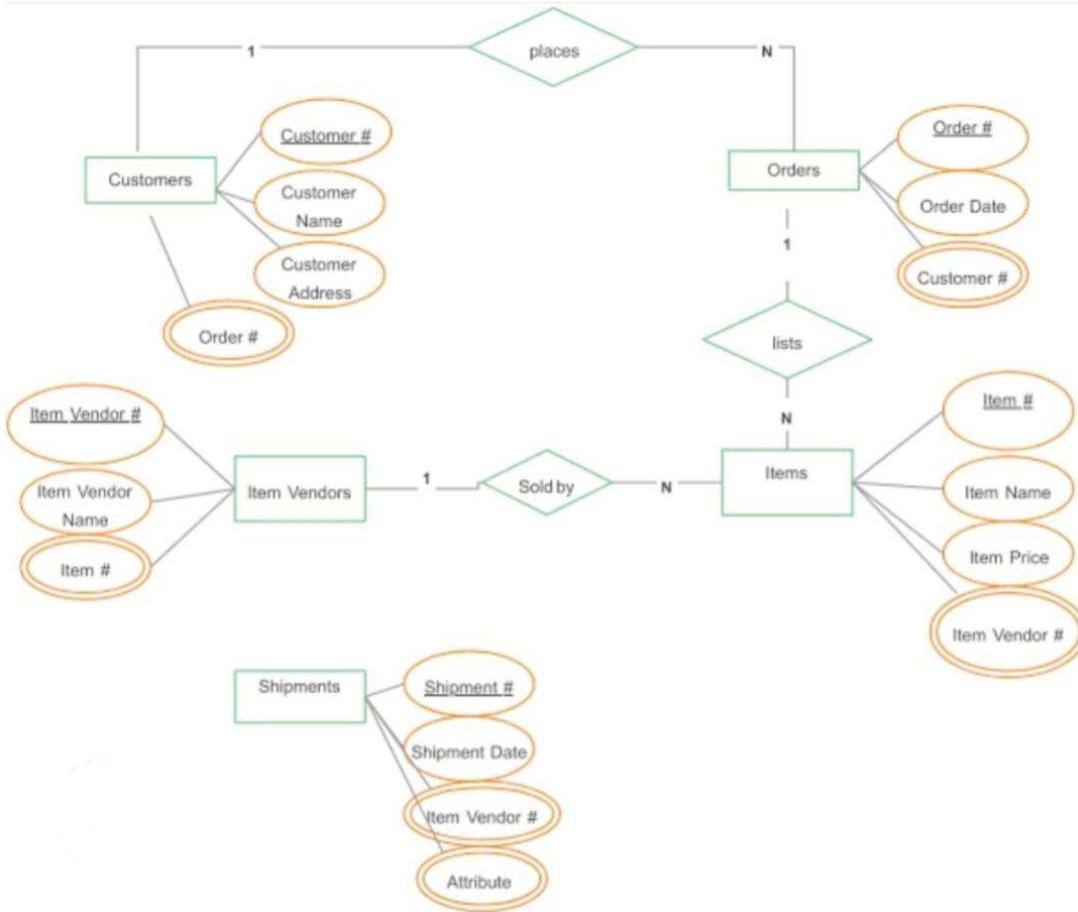
The entity-relationship model is based on a perception of the world as consisting of a collection of basic objects (entities) and relationships among these objects.

- An entity is a distinguishable object that exists.
- Each entity has associated with it a set of attributes describing it.
- E.g. number and balance for an account entity.
- A relationship is an association among several entities.
- E.g. a Cust-act relationship associates a customer with each account he or she has.
- The set of all entities or relationships of the same type is called the entity set or relationship set.
- Another essential element of the E-R diagram is the mapping cardinalities, which express the number of entities to which another entity can be associated via a relationship set.

The overall logical structure of a database can be expressed graphically by an E-R diagram:

- Rectangles: represent entity sets.
- Ellipses: represent attributes.
- Diamonds: represent relationships among entity sets.
- Lines: link attributes to entity sets and entity sets to relationships.

ER DIAGRAM:



TABLES USED IN THE DATABASE

DEPARTMENTS:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 dept_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 dept_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3 password	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4 role	int(11)			No	0			Change Drop More
<input type="checkbox"/>	5 dept_details	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	6 added_at	date			Yes	NULL			Change Drop More

ITEMS:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 item_id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 item_name	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3 item_cat	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4 item_detail	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	5 bill_no	varchar(30)	latin1_swedish_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	6 supplier_id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	7 dept_id	int(11)			No	1			Change Drop More
<input type="checkbox"/>	8 supplied_at	date			No	None			Change Drop More

SUPPLIERS:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 supplier_id 🚪	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2 supplier_name 💬	varchar(30)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	3 supplier_details	varchar(255)	latin1_swedish_ci		No	None			Change Drop More
<input type="checkbox"/>	4 added_at	date			No	None			Change Drop More

SAMPLE OUTPUT or SCREENSHOTS

LOGIN PAGE:



ADMIN HOME PAGE:

College Stock Inventory

ADMIN Home Items Departments Suppliers Reports Settings Welcome ADMIN 02:44:51pm Logout

Admin Dashboard Overview

Total Departments: 3

Available Suppliers: 9

Items In Stock: 8

Bar Chart - Resource Counts

Pie Chart - Resource Distribution

The screenshot displays the 'College Stock Inventory' admin dashboard. At the top, there's a navigation bar with links for 'ADMIN Home', 'Items', 'Departments', 'Suppliers', 'Reports', 'Settings', 'Welcome ADMIN' (showing the time as 02:44:51pm), and 'Logout'. Below the navigation is a title 'College Stock Inventory'. The main area features a section titled 'Admin Dashboard Overview' with three summary boxes: 'Total Departments' (3), 'Available Suppliers' (9), and 'Items In Stock' (8). Below this are two charts: a bar chart titled 'Bar Chart - Resource Counts' showing two bars (green and red) against a scale from 5 to 9, and a pie chart titled 'Pie Chart - Resource Distribution' divided into three segments (purple, teal, and yellow).



ITEMS LIST:

College Stock Inventory

ADMIN Home Items Departments Suppliers Reports Settings Welcome ADMIN 11:07:59am Logout

+ New PDF

ALL ITEMS

Show 10 entries Search:

#	ITEM	CATEGORY	DETAIL	SUPPLIER	DEPARTMENT	SUPPLIED AT	ACTION
1	Item Five	Second Category	Demo Demo Demo	Redsupplies	Stock Administrator	2021-08-01	<button>Edit</button> <button>Delete</button>
2	Item Six	Third Category	Demo Demo	Redsupplies	Stock Administrator	2021-08-01	<button>Edit</button> <button>Delete</button>
3	Item Seven	First Category	Demo Demo	MG Foods	Stock Administrator	2021-08-01	<button>Edit</button> <button>Delete</button>
4	Item Eight	First Category	Demo Details	Vista Suppliers	Stock Administrator	2021-08-01	<button>Edit</button> <button>Delete</button>
5	Item Nine	Fourth Category	Demo Details N	MegaGoods Supplies	Stock Administrator	2021-08-01	<button>Edit</button> <button>Delete</button>
6	Item Ten	Fifth Category	This is a demo detail	AEC Components	Stock Administrator	2021-08-01	<button>Edit</button> <button>Delete</button>

ITEMS ALLOCATE

College Stock Inventory

ADMIN Home Items Departments Suppliers Reports Settings Welcome ADMIN 11:10:30am Logout

Show 10 entries Search:

#	ITEM	CATEGORY	DETAIL	SUPPLIED AT	ACTION
1	Item Five	Second Category	Demo Demo Demo	2021-08-01	
2	Item Six	Third Category	Demo Demo	2021-08-01	
3	Item Seven	First Category	Demo Demo	2021-08-01	
4	Item Eight	First Category	Demo Details	2021-08-01	
5	Item Nine	Fourth Category	Demo Details N	2021-08-01	
6	Item Ten	Fifth Category	This is a demo detail	2021-08-01	
7	Item Eleven	Fourth Category	Demo Details..	2021-08-01	

DEAPARTMENTS:

College Stock Inventory

ADMIN Home Items Departments Suppliers Reports Settings Welcome ADMIN 11:03:21am Logout

[+ New](#) [PDF](#)

Show 10 entries Search:

#	ID	DEPARTMENT	DETAIL	ADDED ON	ACTION
1	2	Department 2	Dept 2	2018-04-05	Edit Delete
2	10	Department 3	Dept 3	2018-04-04	Edit Delete
3	16	Department 4	Dept 4	2018-04-19	Edit Delete

Showing 1 to 3 of 3 entries [Previous](#) [Next](#)

© Copyright 2024-2025 | All Rights Reserved
Developed By VNC COLLEGE HOSPET STUDENTS

SUPPLIERS:

College Stock Inventory

ADMIN Home Items ▾ Departments ▾ Suppliers ▾ Reports ▾ Settings ▾ Welcome ADMIN 11:12:15am Logout

+ New PDF

Show 10 entries Search:

#	NAME	DETAIL	ADDED AT	ACTION
1	Fozzby	984 Woodstock Drive	2019-04-12	
2	Redsupplies	4407 Jerry Toth Drive	2019-02-01	
3	MegaGoods Supplies	1908 Leo Street	2020-01-17	
4	Mooszer Electronics	1491 Shinn Avenue	2019-12-06	
5	AEC Components	1743 Washburn Street	2019-12-13	
6	MG Foods	2617 Happy Hollow Road	2019-10-18	
7	Vista Suppliers	2820 Sunset Drive	2019-02-07	

STOCK DETAILS:

College Stock Inventory

ADMIN Home Items Departments Suppliers Reports Settings Welcome ADMIN 11:13:25am Logout

Item Distribution Overview

Department-wise Item Count		Supplier-wise Item Count	
Department	Total Items	Supplier	Total Items
Stock Administrator	8	MegaGoods Supplies	3
Department 2	3	MG Foods	2
Department 4	1	Redsupplies	2
Department 3	0	AEC Components	1
		Fozzby	1
		Vista Suppliers	1
		USPharma	1
		Mooszer Electronics	1

ADMIN PROFILE:

College Stock Inventory

ADMIN Home Items ▾ Departments ▾ Suppliers ▾ Reports ▾ Settings ▾ Welcome ADMIN 11:14:42am Logout

ADMIN PROFILE

DEPARTMENT ID :1	
DEPARTMENT NAME :Stock Administrator	
DETAIL :Default Stock Department	
ADDED AT :2018-03-27	

CHANGE PASSWORD:

College Stock Inventory

ADMIN Home Items Departments Suppliers Reports Settings Welcome ADMIN 11:15:44am Logout

CHANGE PASSWORD

* Fields Are Required

CURRENT PASSWORD: *

NEW PASSWORD: *

CONFIRM PASSWORD: *

Change

RESET DEPARTMENT PASSWORD:

College Stock Inventory

ADMIN Home Items ▾ Departments ▾ Suppliers ▾ Reports ▾ Settings ▾ Welcome ADMIN 11:17:04am Logout

Reset Password For Departments

Email: *

DEPARTMENT: *

SOURCE CODE

INDEX.PHP

```
<?php
require_once 'includes/header.php';
require_once 'bootstrap.php';
?>

<style>
body {
    background: url('images/your-background.jpg') no-repeat center
    center fixed;
    background-size: cover;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

.login-wrapper {
    margin-top: 100px;
    max-width: 400px;
    padding: 30px;
```

```
background: rgba(255, 255, 255, 0.9);
border-radius: 10px;
box-shadow: 0px 0px 10px 2px rgba(0,0,0,0.2);

}

h1.text-center {
    color: #fff;
    margin-top: 50px;
    font-weight: bold;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
}

label {
    font-weight: 500;
}

.btn-success {
    width: 100%;
}

</style>

<h1 class="text-center">COLLEGE INVENTORY SYSTEM</h1>

<div class="container d-flex justify-content-center">
```

```
<div class="login-wrapper mt-5">
    <h3 class="text-center mb-4">Login Panel</h3>

    <form action="<?php echo URLROOT; ?>/actions/__login.php"
method="post">

        <div>
            <?php flash(); ?>
        </div>

        <div class="form-group">
            <label for="id">ID: <sup>*</sup></label>
            <input type="text" name="id" class="form-control"
placeholder="Enter Your ID" required>
        </div>

        <div class="form-group">
            <label for="password">Password: <sup>*</sup></label>
            <input type="password" name="password" class="form-
control" placeholder="Password" required>
        </div>

        <input type="hidden" name="token" value="<?php echo
Token::generate(); ?>">
```

```
<div class="form-group mt-3">  
    <input type="submit" name="submit" value="LOGIN"  
    class="btn btn-success">  
</div>  
  
</form>  
</div>  
</div>  
  
<?php require_once './includes/footer.php'; ?>
```

ABOUT.PHP

```
<?php require_once './includes/header.php';?>  
<?php LogInCheck(); ?>  
<h1>Stock Management System</h1>  
  
<p>Version: <strong><?php echo APPVERSION; ?></strong></p>  
<?php require_once './includes/footer.php';?>
```

ADMIN HOME PAGE:

```
<?php
require_once './includes/header.php';
LogInCheck();
require_once './includes/admin_nav.php';
flash();
require_once 'db.php';

// --- Fetch data once and store in variables ---
$sql = "SELECT COUNT(`dept_id`) AS dept_available FROM
`department` WHERE `dept_id` <> '1'";
$result0 = $conn->query($sql);
$row0 = $result0->fetch_assoc();
$deptCount = $row0['dept_available'];

$sql = "SELECT COUNT(`supplier_id`) AS supplier_available FROM
`supplier`";
$result1 = $conn->query($sql);
$row1 = $result1->fetch_assoc();
$supplierCount = $row1['supplier_available'];

$sql = "SELECT COUNT(`item_id`) AS item_available FROM `item`"
WHERE `dept_id` = 1";
```

```
$result2 = $conn->query($sql);
$row2 = $result2->fetch_assoc();
$itemCount = $row2['item_available'];
?>

<!-- Include Chart.js -->
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>

<style>
.stat-card {
    background: #fffffcc;
    border-radius: 12px;
    padding: 20px;
    box-shadow: 0 4px 15px rgba(0,0,0,0.1);
    margin-bottom: 30px;
    transition: 0.3s ease-in-out;
}

.stat-card:hover {
    transform: scale(1.02);
}

.dashboard-title {
    text-align: center;
```

```
        margin-top: 30px;  
        font-weight: bold;  
        font-size: 28px;  
    }  
  
.chart-container {  
    background: #fff;  
    border-radius: 12px;  
    padding: 20px;  
    margin-top: 20px;  
    box-shadow: 0 4px 15px rgba(0,0,0,0.1);  
}  
  
.chart-container canvas {  
    width: 100% !important;  
    height: 300px !important;  
}  
</style>  
  
<div class="container">  
    <h2 class="dashboard-title">Admin Dashboard Overview</h2>  
    <div class="row">  
        <?php if ($_SESSION['role'] == 'admin') { ?>
```

```
<!-- Department -->

<div class="col-md-4">

    <div class="stat-card text-center">

        <h4>Total Departments</h4>

        <h2 class='text-primary'><?= $deptCount ?></h2>

    </div>

</div>

<!-- Suppliers -->

<div class="col-md-4">

    <div class="stat-card text-center">

        <h4>Available Suppliers</h4>

        <h2 class='text-success'><?= $supplierCount ?></h2>

    </div>

</div>

<!-- Items -->

<div class="col-md-4">

    <div class="stat-card text-center">

        <h4>Items In Stock</h4>

        <h2 class='text-danger'><?= $itemCount ?></h2>

    </div>

</div>

<?php } ?>
```

```
</div>

<!-- Charts Row -->
<div class="row">
    <!-- Bar Chart -->
    <div class="col-md-6">
        <div class="chart-container">
            <h4 class="text-center">Bar Chart - Resource Counts</h4>
            <canvas id="adminChart"></canvas>
        </div>
    </div>

    <!-- Pie Chart -->
    <div class="col-md-6">
        <div class="chart-container">
            <h4 class="text-center">Pie Chart - Resource Distribution</h4>
            <canvas id="pieChart"></canvas>
        </div>
    </div>
</div>

<!-- Chart.js Scripts -->
```

```
<script>

    // Bar Chart

    const ctx =
document.getElementById('adminChart').getContext('2d');

    const adminChart = new Chart(ctx, {

        type: 'bar',

        data: {

            labels: ['Departments', 'Suppliers', 'Items'],

            datasets: [{

                label: 'Total Count',

                data: [<?= $deptCount ?>, <?= $supplierCount ?>, <?=
$itemCount ?>],


                backgroundColor: ['#007bff', '#28a745', '#dc3545'],

                borderColor: ['#0056b3', '#1c7c33', '#a71d2a'],

                borderWidth: 1

            }]

        },

        options: {

            responsive: true,

            scales: {

                y: {

                    beginAtZero: true,

                    ticks: {

                        precision: 0

                    }

                }

            }

        }

    });


```

```

        }
    }
}

});

// Pie Chart

const pieCtx =
document.getElementById('pieChart').getContext('2d');

const pieChart = new Chart(pieCtx, {
    type: 'pie',
    data: {
        labels: ['Departments', 'Suppliers', 'Items'],
        datasets: [{
            data: [<?= $deptCount ?>, <?= $supplierCount ?>, <?= $itemCount ?>],
            backgroundColor: ['#17a2b8', '#ffc107', '#6f42c1'],
            borderColor: '#fff',
            borderWidth: 2
        }]
    },
    options: {
        responsive: true,
        plugins: {
            legend: {
                position: 'bottom'
            }
        }
    }
});

```

```
        }
    }
}

});

</script>

<?php
// Optional: backup confirmation
if (isset($bak) && $bak == 1) {
    echo '<script>alert("Backup taken");</script>';
}
?>

<?php require_once './includes/footer.php'; ?>
```

ALLOCATE.PHP

```
<?php
require_once './includes/header.php';
//if not logged in return him to login page
LogInCheck();
require_once './includes/admin_nav.php';
//var_dump($_SESSION);
```

```

?>

<!--exp1-->

<div class="col-sm-10 col-sm-offset-1">
<div class="row">
    <!--place for error message flashing-->
    <?php
        //this will display any kind of error message as
        flash();
    ?>
</div>
<div class="" style="height: 10px;">
</div>
<div class="row">
    <table id="myTable" class="table table-bordered table-hover table-striped table-responsive">
        <?php
            include_once('db.php');
            //sql according to role
            $sql = "SELECT * FROM item WHERE `dept_id` = '1'" ;
            echo '<thead>' ;
            <tr>

```

```

<th>#</th>
<th>ITEM</th>
<th>CATEGORY</th>
<th>DETAIL</th>
<th>SUPPLIED AT</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>';

$result = $conn->query($sql);
$i = 1;
while($row = $result->fetch_assoc())
{
echo"<tr>
<td>".$i."</td>
<td>".$row['item_name']."</td>
<td>".$row['item_cat']."</td>
<td>".$row['item_detail']."</td>
<td>".$row['supplied_at']."</td>
<td><a href='#allocate_". $row['item_id']."' class='btn btn-success btn-sm' data-toggle='modal'><span class='glyphicon glyphicon-edit'></span>Allocate</a>
</td>
</tr>";
```

```
$i++;

//require_once 'models/allocate_itemModel.php';

include('models/allocate_itemModel.php') ;

}

//$conn->close();

?>

</tbody>

</table>

<hr>

</div>

</div>
```

BACKUP.PHP

```
<?php

/** 
 * this is used to create
 * back up of the system
 */

require_once 'bootstrap.php';
```

```
LogInCheck();

//if not admin redirect to home page
if($_SESSION['role'] != 'admin')
{
    header('location: admin_home.php');
}

//take backup
$result = backDb('localhost','root','','stock');

if( $result == 1)
{
    // $_SESSION['success'] = 'back up taken successfully';
    echo '<script>location.href = "admin_home.php";</script>';

}

else{
    $_SESSION['error'] = 'something went wrong ';
    header('location: admin_home.php');
}
```

BOOSTRAP.PHP

```
<?php
```

```
/**  
 * this is used to  
 * load all required  
 * libraries  
 */  
  
//set error reporting to zero  
error_reporting(0);  
  
  
  
// Load Config  
require_once 'config/config.php';  
  
  
// Load Helpers if any  
require_once 'helpers/session_helper.php';  
require_once 'helpers/url_helper.php';  
require_once 'helpers/otp_helper.php';  
require_once 'helpers/back_up_helper.php';  
require_once 'helpers/csrf_helper.php';
```

CHANGE PASS.PHP

```
<?php
```

```
require_once './includes/header.php';
LogInCheck();
require_once './includes/admin_nav.php';
?>
<!--pwd change begin-->
<div class="container">
<div class="row">
<div class="col-md-offset-2">
<div class="panel panel-primary">
<div class="panel-body">
<div class="text-center text-info">CHANGE
PASSWORD</div>
</div>
<div class="panel-footer">
<div class="row">
<div class="well well-lg">
<form action=<?php echo URLROOT;
?>/actions/__change_pass.php" method="post">
<!--place for error message flashing-->
<div>
<?php
//this will display any kind of error/success
message
flash();
?>
```

```
</div>

<div class="alert-info text-center">
    <em>* Fields Are Required </em>
</div>

<div class="form-group">
    <label for="pass">CURRENT PASSWORD:<br/>
<sup>*</sup></label>
    <input type="password" name="pass" class="form-control form-control-lg" placeholder="enter current password" required>
</div>

<div class="form-group">
    <label for="new_pass">NEW PASSWORD:<br/>
<sup>*</sup></label>
    <input type="password" name="new_pass" id="new_pass" class="form-control form-control-lg" placeholder="enter new password" required>
</div>

<div class="form-group">
    <label for="con_pass">CONFIRM PASSWORD:<br/>
<sup>*</sup></label>
    <input type="password" name="con_pass" id="con_pass" class="form-control form-control-lg" placeholder="confirm new password" required>
</div>
```

```
<div class="form-group">
    <input type="submit" value="Change"
name="Change" class="btn btn-success btn-sm">
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<!--end pwd chnge-->

<hr>
<br>
<script>

</script>
<?php require_once './includes/footer.php'; ?>
```

DEPARTMENTS.PHP

```
<?php
require_once './includes/header.php';
LogInCheck();
require_once './includes/admin_nav.php';
//var_dump($_SESSION);
?>
<!--exp1-->

<div class="col-sm-10 col-sm-offset-1">
    <div class="row">
        <!--place for error message flashing-->
        <?php
            //this will display any kind of error message as
            flash();
        ?>
    </div>
    <div class="row">
        <a href="#addnew" data-toggle="modal" class="btn btn-primary"><span class="glyphicon glyphicon-plus"></span> New</a>
        <a href="../reports/all_departments.php" target="_new"
            class="btn btn-success pull-right"><span class="glyphicon glyphicon-print"></span> PDF</a>
    </div>
</div>
```

```
</div>

<div class="" style="height: 10px;">
</div>

<div class="row">
    <table id="myTable" class="table table-hover table-bordered
table-striped table-responsive">
        <thead>
            <th>#</th>
            <th>ID</th>
            <th>DEPARTMENT</th>
            <th>DETAIL</th>
            <th>ADDED ON</th>
            <th>ACTION</th>
        </thead>
        <tbody>
            <?php
                include_once('db.php');
                $sql = "SELECT * FROM `department` WHERE `role` <> '1'";
                $query = $conn->query($sql);
                $i=1;
                while($row = $query->fetch_assoc())

```

```

{
    echo"<tr>
<td>".$i."</td>
<td>".$row['dept_id']."'</td>
<td>".$row['dept_name']."'</td>
<td>".$row['dept_details']."'</td>
<td>".$row['added_at']."'</td>
<td><a href='".$row['dept_id']."' class='btn
btn-success btn-sm' data-toggle='modal'><span class='glyphicon
glyphicon-edit'></span> Edit</a>
<a href='".$row['dept_id']."' class='btn btn-
danger btn-sm' data-toggle='modal'><span class='glyphicon
glyphicon-trash'></span> Delete</a>
</td>
</tr>";
$i++;
include('models/edit_delete_departmentModel.php') ;
}

?>
</tbody>
</table>
<hr>
<?php
//add required models
require_once 'models/add_departmentModel.php';

```

```
?>

</div>

</div>

<!--end exp1-->

<?php require_once './includes/footer.php'; ?>
```

FORGOT_PASS.PHP

```
<?php

require_once 'includes/header.php';
require_once 'includes/admin_nav.php';
LogInCheck();
if(!($_SESSION['role']=='admin'))
{
    header('location: admin_home.php');
}
?>

<nav class="nav">
    <div class="container-fluid">
        <h1 class="text-center"></h1>
    </div>
</nav><hr>
```

```
<div class="row">&nbsp;<br></div>

<div class="row">
    <div class="col-md-6 col-sm-12 mx-auto col-md-offset-3">
        <div class="jumbotron card card-body bg-light mt-5">
            <h2 clas="text-center">Reset Password For Departments</h2>
            <p></p>
            <form action="php echo URLROOT;
?&gt;/actions/_forgot_pass.php" method="post"&gt;

                &lt;div&gt;
                    &lt;?php
                    //this will display any kind of error/success message
                    flash();
                    ?&gt;
                &lt;/div&gt;

                &lt;div class="form-group"&gt;
                    &lt;label for="email"&gt;Email: &lt;sup&gt;*&lt;/sup&gt;&lt;/label&gt;
                    &lt;input type="email" name="email" class="form-control form-control-lg" placeholder="Enter Department Email" required&gt;
                &lt;/div&gt;

                &lt;div class="form-group"&gt;
                    &lt;div class=""&gt;
                        &lt;label for="dept_id"&gt;DEPARTMENT:
                        &lt;sup&gt;*&lt;/sup&gt;&lt;/label&gt;
                    &lt;/div&gt;
                &lt;/div&gt;
            &lt;/form&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre
```

```

<!--begin option-->

<?php

//code for fetching the suppliers' information

require_once 'db.php';

$sql = "SELECT * FROM `department` WHERE `dept_id` <> '1'";

$query = $conn->query($sql);

echo "<select class='form-control id='dept_id' name='dept_id' required>";

echo "<option value="">Select Department</option>";

while($row = $query->fetch_assoc()){

    echo "<option value=\"".$row['dept_id']."'>".$row['dept_name']."</option>";

}

echo "</select>";

$conn->close();

?>

<!--end-->

</div>

</div>

<div class="form-group">

    <input type="submit" value="RESET" name="submit" class="btn btn-success">

</div>

```

```
</form>

</div>

</div>

</div>

<?php require_once './includes/footer.php'; ?>
```

ITEM_COUT.PHP

```
<?php

require_once './includes/header.php';

LogInCheck();

require_once './includes/admin_nav.php';

require_once 'db.php';

?>

<!-- Include Chart.js --&gt;
&lt;script src="https://cdn.jsdelivr.net/npm/chart.js"&gt;&lt;/script&gt;
<!-- Bootstrap Modal support --&gt;
&lt;link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css"&gt;</pre>
```

```

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap
.min.js"></script>

<div class="container mt-4">
    <h3 class="text-center">Item Distribution Overview</h3>

    <div class="row mt-4">
        <!-- Department-Wise -->
        <div class="col-md-6">
            <h4>Department-wise Item Count</h4>
            <table class="table table-bordered table-striped">
                <thead>
                    <tr>
                        <th>Department</th>
                        <th>Total Items</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                        $deptData = [];
                        $deptSql = "SELECT d.dept_id, d.dept_name,
                        COUNT(i.item_id) AS item_count
                        FROM department d
                        LEFT JOIN item i ON d.dept_id = i.dept_id
                    ";
                
```

```

        GROUP BY d.dept_id
        ORDER BY item_count DESC";

$deptResult = $conn->query($deptSql);

while ($row = $deptResult->fetch_assoc()) {
    $deptData[] = $row;
    echo "<tr>
        <td>{$row['dept_name']}</td>
        <td><a href='#deptModal{$row['dept_id']}' data-
        toggle='modal'>{$row['item_count']}</a></td>
    </tr>";
}
?>
</tbody>
</table>
</div>

<!-- Supplier-Wise -->
<div class="col-md-6">
    <h4>Supplier-wise Item Count</h4>
    <table class="table table-bordered table-striped">
        <thead>
            <tr>
                <th>Supplier</th>
                <th>Total Items</th>

```

```

        </tr>
    </thead>
    <tbody>
        <?php
            $supplierData = [];
            $supplierSql = "SELECT s.supplier_id, s.supplier_name,
            COUNT(i.item_id) AS item_count
                FROM supplier s
                LEFT JOIN item i ON s.supplier_id = i.supplier_id
                GROUP BY s.supplier_id
                ORDER BY item_count DESC";
            $supplierResult = $conn->query($supplierSql);
            while ($row = $supplierResult->fetch_assoc()) {
                $supplierData[] = $row;
                echo "<tr>
                    <td>{$row['supplier_name']}</td>
                    <td><a href='#supplierModal{$row['supplier_id']}'>{$row['item_count']}

```

```

<!-- Charts -->

<div class="row mt-4">
    <div class="col-md-6">
        <h5 class="text-center">Department-wise Bar Chart</h5>
        <canvas id="deptChart"></canvas>
    </div>
    <div class="col-md-6">
        <h5 class="text-center">Supplier-wise Bar Chart</h5>
        <canvas id="supplierChart"></canvas>
    </div>
</div>

<!-- Department Modals -->
<?php foreach ($deptData as $dept): ?>
<div class="modal fade" id="deptModal<?= $dept['dept_id'] ?>" tabindex="-1" role="dialog">
    <div class="modal-dialog modal-lg">
        <div class="modal-content">
            <div class="modal-header">
                <h4 class="modal-title">Items in <?= $dept['dept_name'] ?></h4>
                <button type="button" class="close" data-dismiss="modal">&times;</button>
            </div>

```

```

        </div>

        <div class="modal-body">
            <?php
                $deptId = $dept['dept_id'];

                $itemsSql = "SELECT item_name, item_cat, item_detail FROM
                item WHERE dept_id = $deptId";

                $items = $conn->query($itemsSql);

                if ($items->num_rows > 0) {

                    echo "<table class='table table-
                    bordered'><tr><th>Item</th><th>Category</th><th>Detail</th></tr
                    >";
                    while ($item = $items->fetch_assoc()) {
                        echo "<tr>
                            <td>{$item['item_name']}</td>
                            <td>{$item['item_cat']}</td>
                            <td>{$item['item_detail']}</td>
                        </tr>";
                    }
                    echo "</table>";
                } else {
                    echo "<p>No items found for this department.</p>";
                }
            ?>
        </div>
    </div>

```

```

        </div>
    </div>
    <?php endforeach; ?>

    <!-- Supplier Modals -->
    <?php foreach ($supplierData as $sup): ?>
        <div class="modal fade" id="supplierModal<?= $sup['supplier_id'] ?>" tabindex="-1" role="dialog">
            <div class="modal-dialog modal-lg">
                <div class="modal-content">
                    <div class="modal-header">
                        <h4 class="modal-title">Items from <?= $sup['supplier_name'] ?></h4>
                        <button type="button" class="close" data-dismiss="modal">&times;</button>
                    </div>
                    <div class="modal-body">
                        <?php
                            $supId = $sup['supplier_id'];
                            $itemsSql = "SELECT item_name, item_cat, item_detail FROM item WHERE supplier_id = $supId";
                            $items = $conn->query($itemsSql);
                            if ($items->num_rows > 0) {
                                echo "<table class='table table-bordered'><tr><th>Item</th><th>Category</th><th>Detail</th></tr>";
                            }
                        </?php
                    </div>
                </div>
            </div>
        </div>
    <?php endforeach; ?>

```

```

while ($item = $items->fetch_assoc()) {
    echo "<tr>
        <td>{$item['item_name']}</td>
        <td>{$item['item_cat']}</td>
        <td>{$item['item_detail']}</td>
    </tr>";
}

echo "</table>";

} else {
    echo "<p>No items found for this supplier.</p>";
}

?>

</div>
</div>
</div>
</div>
<?php endforeach; ?>

<!-- Chart Script --&gt;
&lt;script&gt;
    const deptChart = new
    Chart(document.getElementById('deptChart'), {
        type: 'bar',
        data: {
</pre>

```

```

        labels: <?= json_encode(array_column($deptData,
'dept_name')) ?>,
        datasets: [{

            label: 'Total Items',
            data: <?= json_encode(array_column($deptData,
'item_count')) ?>,
            backgroundColor: '#007bff'
        }]
    },
    options: {
        responsive: true,
        scales: {
            y: { beginAtZero: true }
        }
    }
});


```

```

const supplierChart = new
Chart(document.getElementById('supplierChart'), {
    type: 'bar',
    data: {

        labels: <?= json_encode(array_column($supplierData,
'supplier_name')) ?>,
        datasets: [{

            label: 'Total Items',

```

```
        data: <?= json_encode(array_column($supplierData,  
'item_count')) ?>,  
        backgroundColor: '#28a745'  
    ]]  
,  
options: {  
    responsive: true,  
    scales: {  
        y: { beginAtZero: true }  
    }  
}  
});  
</script>
```

```
<?php require_once './includes/footer.php'; ?>
```

ITEM.PHP

```
<?php  
require_once './includes/header.php';  
LogInCheck();  
require_once './includes/admin_nav.php';  
require_once 'db.php';  
?>
```

```
<div class="container">

    <!-- Flash message area -->

    <div class="row">
        <?php flash(); ?>
    </div>

    <!-- Top buttons -->
    <div class="row mb-3">
        <?php if ($_SESSION['role'] == 'admin') { ?>
            <a href="#addnew" data-toggle="modal" class="btn btn-primary">
                <span class="glyphicon glyphicon-plus"></span> New
            </a>
        <?php } ?>
            <a href=".//reports/all_items.php" target="_blank" class="btn btn-success pull-right">
                <span class="glyphicon glyphicon-print"></span> PDF
            </a>
    </div>

    <!-- Title -->
    <div class="row">
        <h3 class="text-muted text-center">ALL ITEMS</h3>
    </div>
```

```

<!-- Item Table -->

<div class="row">
    <div class="table-responsive">
        <table id="myTable" class="table table-hover table-bordered table-striped">
            <?php
                $item_current_dept_id = $_SESSION['dept_id'];

                // Query based on user role
                $sql = ($_SESSION['role'] == 'admin') ?
                    "SELECT * FROM item
                     JOIN department ON item.dept_id = department.dept_id
                     JOIN supplier ON item.supplier_id = supplier.supplier_id"
                    :
                    "SELECT * FROM item
                     JOIN department ON item.dept_id = department.dept_id
                     WHERE item.dept_id = '$item_current_dept_id'";

                $query = $conn->query($sql);

                // Admin Table
                if ($_SESSION['role'] == 'admin') {
                    echo '
                        <thead>

```

```

<tr>
    <th>#</th>
    <th>ITEM</th>
    <th>CATEGORY</th>
    <th>DETAIL</th>
    <th>SUPPLIER</th>
    <th>DEPARTMENT</th>
    <th>SUPPLIED AT</th>
    <th>ACTION</th>
</tr>
</thead>
<tbody>';

```

```

$ i = 1;
while ($row = $query->fetch_assoc()) {
    echo "
<tr>
    <td>{$i}</td>
    <td>{$row['item_name']}</td>
    <td>{$row['item_cat']}</td>
    <td>{$row['item_detail']}</td>
    <td>{$row['supplier_name']}</td>
    <td>{$row['dept_name']}</td>
    <td>{$row['supplied_at']}</td>

```

```

<td>
    <a href='#{edit_{$row['item_id']}}' class='btn btn-success btn-sm' data-toggle='modal'>
        <span class='glyphicon glyphicon-edit'></span>
    Edit
    </a>
    <a href='#{delete_{$row['item_id']}}' class='btn btn-danger btn-sm' data-toggle='modal'>
        <span class='glyphicon glyphicon-trash'></span>
    Delete
    </a>
</td>
</tr>";
$i++;
}

// Include modal templates (must exist)
include('models/edit_delete_itemModel.php');

}

echo '</tbody>';
// Add Item Modal
require_once 'models/add_itemModel.php';

} else {
    // Department Table

```

```

echo '

<thead>

<tr>

    <th>SL NO</th>
    <th>ID</th>
    <th>ITEM</th>
    <th>CATEGORY</th>
    <th>DETAIL</th>

</tr>

</thead>

<tbody>';




$ i = 1;

while ($row = $query->fetch_assoc()) {

    echo "

<tr>

    <td>{$i}</td>
    <td>{$row['item_id']}</td>
    <td>{$row['item_name']}</td>
    <td>{$row['item_cat']}</td>
    <td>{$row['item_detail']}</td>

</tr>";

    $i++;

}

```

```
        echo '</tbody>';
    }
?>
</table>
</div>
</div>
</div>

<?php require_once './includes/footer.php'; ?>
```

LOGOUT.PHP

```
<?php
require_once 'bootstrap.php';
LogInCheck();
//unset session
unset($_SESSION['dept_name']);
unset($_SESSION['dept_id']);
unset($_SESSION['role']);

//redirect to log in page
$_SESSION['success'] = 'you have successfully logged out ';
header('location: index.php');
```

MAIN PAGE.PHP

```
<?php require_once './includes/header.php'; ?>
<?php LogInCheck(); ?>
<?php
    require_once './includes/navbar.php';
    //flash message
    flash();
?>
<?php require_once './includes/footer.php'; ?>
```

REGISTER.PHP

```
<?php require_once './includes/header.php';?>
<?php LogInCheck(); ?>
this is registration page
<?php require_once './includes/footer.php';?>
```

REPORT.PHP

```
<?php
require_once './includes/header.php';
LogInCheck();
require_once './includes/admin_nav.php';
?>

<form class="navbar-form navbar-left form-inline" role="search">
    <div class="form-group">
        <input type="date" class="form-control"
placeholder="Search">
    </div>
    <div class="form-group">
        <input type="text" value="to" disabled>
    </div>
    <div class="form-group">
        <input type="date" class="form-control"
placeholder="Search">
    </div>
    <button type="submit" class="btn btn-default">Submit</button>
</form>

<?php require_once './includes/footer.php'; ?>
```

SEND SMS.PHP

```
<?php  
// Authorisation details.  
$username = "";  
$hash = "";  
  
// Config variables. Consult http://api.textlocal.in/docs for more info.  
$test = "0";  
  
// Data for text message. This is the text message data.  
$sender = "TXTLCL"; // This is who the message appears to be from.  
$numbers = ""; // A single number or a comma-separated list of numbers  
$message = "";  
// 612 chars or less  
// A single number or a comma-separated list of numbers  
$message = urlencode($message);  
$data =  
"username=".$username."&hash=".$hash."&message=".$message."  
&sender=".$sender."&numbers=".$numbers."&test=".$test;  
$ch = curl_init('http://api.textlocal.in/send/?');
```

```
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, $data);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
$result = curl_exec($ch); // This is the result from the API
//print_r($result);
curl_close($ch);
?>
```

SUPPLIERS.PHP

```
<?php
require_once 'includes/header.php';
LogInCheck();
require_once 'includes/admin_nav.php';
?>
<!--exp1-->

<div class="col-sm-10 col-sm-offset-1">
    <div class="row">
        <!--place for error message flashing-->
        <?php
            //this will display any kind of error message as
            flash();
        ?>
```

```

        </div>

        <div class="row">
            <a href="#addnew" data-toggle="modal" class="btn btn-primary"><span class="glyphicon glyphicon-plus"></span> New</a>
            <a href="../reports/all_suppliers.php" target="_new" class="btn btn-success pull-right"><span class="glyphicon glyphicon-print"></span> PDF</a>
        </div>

        <div class="" style="height: 10px;">
        </div>

        <div class="row">
            <table id="myTable" class="table table-hover table-bordered table-striped table-responsive">
                <thead>
                    <th>#</th>
                    <th>NAME</th>
                    <th>DETAIL</th>
                    <th>ADDED AT</th>
                    <th>ACTION</th>
                </thead>
                <tbody>
                    <?php
                    include_once('db.php');

                    $sql = "SELECT * FROM `supplier`";

```

```

$query = $conn->query($sql);

$i=1;

while($row = $query->fetch_assoc())
{
echo"<tr>
<td>".$i."</td>
<td>".$row['supplier_name']."</td>
<td>".$row['supplier_details']."</td>
<td>".$row['added_at']."</td>
<td><a href='".$row['supplier_id']."' class='btn
btn-success btn-sm' data-toggle='modal'><span class='glyphicon
glyphicon-edit'></span> Edit</a>
</td>";

$i++;

include('models/edit_delete_supplierModel.php') ;

}

?>

</tbody>
</table>
<hr>
<?php
//add required models
require_once 'models/add_supplierModel.php';

```

```
?>

</div>

</div>

</div>

<!--end exp1-->

<?php require_once './includes/footer.php'; ?>
```

VIEW PROFILE.PHP

```
<?php

require_once 'includes/header.php';

LogInCheck();

require_once 'includes/admin_nav.php';

//var_dump($_SESSION);

?>

<?php

//connect db

require_once 'db.php';

$sql = "SELECT * FROM `department` WHERE `dept_id`="" .

$_SESSION['dept_id'] . """;

//print_r($sql);

$query = $conn->query($sql);
```

```

$row = $query->fetch_assoc();
//var_dump($row);
$conn->close();
?>
<!--profile view code begin-->


<?php echo
$_SESSION['dept_name'] . ' PROFILE'; ?></h3>



DEPARTMENT ID :</b><em><?php echo $row['dept_id']
?></em>&nbsp;&nbsp;&nbsp;<a href="#" class='btn btn-success btn-sm'><span class='glyphicon glyphicon-edit'></span></a>



DEPARTMENT NAME :</b><em><?php echo
$row['dept_name'] ;?></em>&nbsp;&nbsp;&nbsp;<a href="#" class='btn btn-success btn-sm'><span class='glyphicon glyphicon-edit'></span></a>


```

```
</div>

<div class="well well-sm">

    <b>DETAIL :</b><em><?php echo $row['dept_details']
;?></em>&nbsp;&nbsp;&nbsp;<a href="#" class='btn btn-success btn-
sm'><span class='glyphicon glyphicon-edit'></span></a>

</div>

<div class="well well-sm">

    <b>ADDED AT :</b><em><?php echo $row['added_at']
;?></em>&nbsp;&nbsp;&nbsp;<a href="#" class='btn btn-success btn-
sm'><span class='glyphicon glyphicon-edit'></span></a>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!--end profile-->

<hr>

<?php require_once 'includes/footer.php'; ?>
```

TESTING

Introduction:

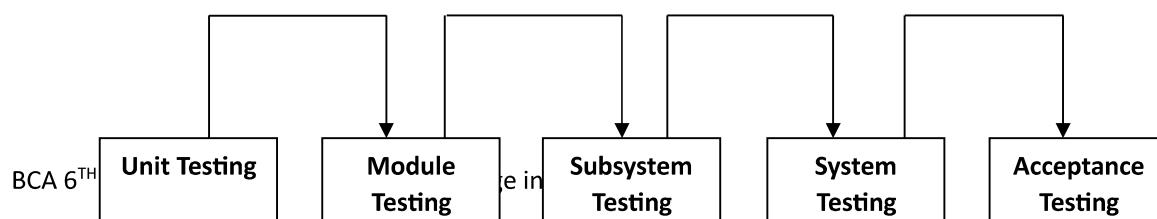
Software Testing is a process of executing program with an intent of finding error. Testing is vital to success of the system. Testing demonstrates that the software functions appear to be working according to the specifications and performance requirements appeared to have been met. If a test is conducted successfully, it will discover errors in the software.

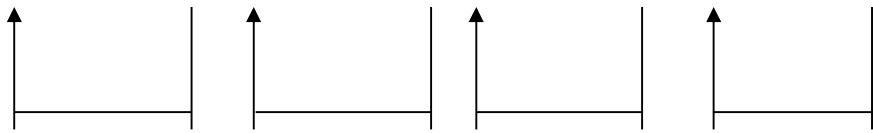
Software Testing consists of all test life cycle activities like static and dynamic testing concerned with planning, preparation and evaluation of software products to determine that the software products satisfy customers requirements and are fit for customer use.

The various strategies that were used in testing this software are as follows:

- Unit Testing
- Integration Testing
- System Testing
 - Validation Testing
 - Black Box Testing
 - White Box Testing

Acceptance Testing





UNIT TESTING:

Unit testing is done on individual modules as they are completed and become executable. This system was tested with the set of proper test data for each module and results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module.

This is also known as MODULE TESTING. This testing is carried out during phases, each module is founded to be working satisfactory as regards to the expected result from the module.

Unit testing involves the design of the test cases that validate the internal program logic is functioning properly, and that program input produces valid output. All decision branches and internal code flow should be validated.

INTEGRATION TESTING:

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied as shown by the Unit testing, the combination of the components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. Integration Testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests to aggregates and delivers as its output. The Integration Testing verifies functional, performance, and reliability requirements placed on a major design items.

FUNCTIONAL TESTING:

Functional tests provide a systematic demonstration of the functions tested that are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional Testing is centred on the following items:

Valid Input: Identified classes of valid input must be accepted.

Invalid Input: Identified classes of invalid input must be accepted.

Functions: Identified functions must be exercised.

Output: Identified classes of application outputs must be exercised.

Systems/Procedures: Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of the current test is determined.

SYSTEM TESTING:

In this the entire software system is tested. The reference document for this process is the requirement document. Here the entire software is tested and the performance of the system was observed to see that it satisfies the requirement specification.

System testing tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System

testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

System testing involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user. The system meets all requirements of the client's specifications.

The following are the types of system tests that were carried out for the system:

- **Validation Testing:**

The system has been tested and implemented successfully and thus ensured that all requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

- **Black Box Testing:**

This method focuses on the functional requirements of the software. This testing enables to derive set input conditions that will fully exercise all functional requirements of the program. Black Box Testing attempts to find errors in the following category.

- Incorrect or missing functions.
- Interface errors.
- Error in external database access.
- Performance errors.
- Initialization and Termination errors.

- **White Box Testing:**

This is performed early in the testing process, while Black Box testing is applied during the last stage of testing. In this test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases.

It has been used to generate the test case in the following test cases:

- Guarantee that all independent paths have been executed.
- Execute all logical decisions from their True and False side.
- Execute all loops at their boundaries and within their operational bounds.
- Execute internal data structures to ensure their validity.

- Ensure whether all the possible validity checks and validity lookups have been provided to validate data entry.

Performance Testing:

Performance Testing can serve different purpose. It can demonstrate that the system meets the performance criteria. It can compare two systems to find which performs better, or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tools such as profilers to measure what parts of a device or software contributes most to the poor performance.

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

Acceptance Testing:

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors. It deals with successful satisfaction of user needs. This project is approved and accepted by the clients. The process flow and execution is 99% working with respect to system testing procedure.

• Test Objectives:

- The system is tested with variety of inputs. The System is tested for accuracy and correctness of the results obtained. Finally the system is tested for interoperability.
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested:

- Verify that the entries are of the correct format.

- No duplicate entries should be allowed.
- All links should take the user to the correct page.

Test cases:

Example:

Login or signup

<u>Test Case</u>	<u>Input</u>	<u>Expected O/P</u>	<u>Actual O/P</u>	<u>Result</u>
1	Valid Username and Password and details	It should display respective page according to user type.	Respective Homepage is displayed .	Passed
2	Invalid Username and Password	It should give appropriate error message saying “Enter proper User-name and Password”	Error message Displayed	Passed
3	User enters valid Username and password.	Respective Homepage is displayed	Respective Home is displayed .	Passed

Example 2:

Products

<u>Test Case</u>	<u>Input</u>	<u>Expected O/P</u>	<u>Actual O/P</u>	<u>Result</u>
1	Valid product name and Product details	The product should stored in database	The product has been stored in database	Passed
2	Add/Update/delete Member details	Add/Update/delete action is taken.	Add/Update/delete Product Had taken place	Passed
3	Valid product name and details but no price.	It should show error message "please enter the required field"	Error message "please enter the required field"	Passed

Example 3:

Category

<u>Test Case</u>	<u>Input</u>	<u>Expected O/P</u>	<u>Actual O/P</u>	<u>Result</u>
1	Valid category name and category description	The category should stored in database	The category has been stored in database	Passed
2	Add description without category name.	Should show error "please enter category name"	Showing error " please enter category name"	Passed

CONCLUSION:

The website reduces manual work. It also increases the reliability. Website provides information about shop. Website helps in maintaining purchase information. Website will help in online business.

FUTURE SCOPE FOR THE SYSTEM:

Change is part of a project .So we provide way to make some changes and enhance the project for future. We have included few modules, some other modules can be included in future. Some modules are like

1. Instant Chatting.
2. Building an android Mobile Application.
3. Providing NEFT Facility.

BIBLIOGRAPHY

References

1. Fundamentals of database system-B.Navathe
2. Ian Sommerville “Software Engineering” Pearson Education Ltd 6th edition 2004
3. Ali Bahrami Object Oriented Systems Development, McGraw hill, 1999.
4. The main reference for this SRS is the information which is provided by IEEE. According the format they have provided, this SRS is been created.
5. ‘Guide to the Website engineering. Body of knowledge’ book by the SWEBOK is been referred.
6. [http://ieeexplore.ieee.org /document/6146379/](http://ieeexplore.ieee.org/document/6146379/)
The reference has been provided by the above link.
7. [https:// www.w3schools.com.](https://www.w3schools.com)