

INDUSTRIAL TRAINING'S FINAL REPORT
ON
STOCK MANAGEMENT SYSTEM
SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF DEGREE OF
BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING



PREPARED BY
AMIT KUMAR
41110402715
AT
DELHI METRO RAIL CORPORATION Ltd.
DMRC TRAINING INSTITUTE SHASTRI PARK, DELHI

Department of Computer Science & Engineering
Amity School of Engineering and Technology
Guru Gobind Singh Indraprastha University, New Delh

CERTIFICATE

CIN No. U74899DL 1995G0I068150

दूरभाष Tel. : 23417910/12
फैक्स Fax : 23417921



दिल्ली मेट्रो रेल कॉर्पोरेशन लि०
DELHI METRO RAIL CORPORATION LTD.

(भारत सरकार एवं दिल्ली सरकार का संयुक्त उपक्रम)
(A JOINT VENTURE OF GOVERNMENT OF INDIA AND GOVT. OF DELHI)

क्र.सं./प्रशिक्षण संस्थान/बीष्मकालीन प्रशिक्षण/प्रमाण पत्र/2018/04

दिनांक: 26.07.2018

प्रशिक्षण प्रमाण पत्र

यह प्रमाणित किया जाता है कि श्री अमित कुमार आपके संस्थान में बी टेक (कम्प्यूटर साइन्स इंजीनियरिंग) के छात्र हैं, ने दिनांक 20 जून से 25 जुलाई 2018 तक दिल्ली मेट्रो रेल कॉर्पोरेशन लिमिटेड में बीष्मकालीन प्रशिक्षण प्राप्त किया है।

दिल्ली मेट्रो की तरफ से उज्ज्वल भविष्य के लिए शुभकामनाएँ।

(महेंद्र सिंह)

प्राचार्य/प्रशिक्षण संस्थान
दिल्ली मेट्रो रेल कॉर्पोरेशन लिमिटेड

(मेट्रो भवन, फायर ब्रिगेड लेन, बाराखम्बा रोड़, नई दिल्ली-110001)
Metro Bhawan, Fire Brigade Lane, Barakhamba Road, New Delhi-110001

DECLARATION

I hereby declare that the Industrial Training Report entitled "**Stock Management System**" is an authentic record of my own work as per the requirements of Industrial Training during the period **June 20th, 2017 to July 25th, 2018** for the award of degree of Bachelor of Technology degree in Computer Science and Technology, Amity School of Engineering Technology, New Delhi, under the guidance of **Mr. MAHENDER SINGH**, Principal Training Department, DMRC.

ACKNOWLEDGEMENT

For the accomplishment of this project, I would like to take this opportunity to express gratitude to the people who have been a part of this project right from its inception and helped and supported me. It is to them I owe my deepest gratitude.

I Express my sincere gratitude to **DELHI METRO RAIL CORPORATION** (A joint Venture of Govt. of Delhi) to Participate and provide us with the technical knowledge in CSE Department of (Shastri park Depot) DELHI METRO RAIL CORPORATION.

At the outset, I would like to express my immense gratitude to my training guide, **Mr. MAHENDER SINGH**, guiding me right from the inception till the Successful completion of the Training.

I express my sincere gratitude to **Prof. Dr. Rekha Agarwal (Director)**, Amity School of Engineering and Technology, and **Prof. Pinki Nayar Head of Department (Department of CS/IT)**, for their constant encouragement and guidance. Also, I would like to thanks **Ms. Pooja Singh (Assistant Professor)**, for her constant support and guidance. Without their insights and support this project would not have been possible.

I am falling short of word for expressing my feeling of gratitude toward him for extending their valuable guidance, through critical Receives of project and the report and above all the moral support he had provided me with all stages of this Training.

Thank You

Amit Kumar

41110402715

ABSTRACT

It's incredibly hard to combat difficult stock control situations, ensuring that you can confidently order stock, sell, replenish and maintain a continual, successful, rotation of merchandise. While long-term experience in retail can really help to understand some trends, market forces and unexpected influences can make all the difference. There's barely enough time in a retail business to keep up with day to day tasks, let alone making market predictions!

With this in mind, what can be done to improve your stock management? We now have a host of Electronic Point of Sale (EPoS) solutions that are designed to do just that. A good EPoS system will meet the demands of modern retailers, providing versatile reporting and comprehensive stock level information. With these elements, it is easier to gauge the success of particular lines, seasonal trends and suggested product lines that have been popular at certain times.

Not only able to handle the stock management for your physical store, with a good EPoS system you can also successfully deal with multichannel stock levels. It's very difficult to manually keep track of sales from your eCommerce site and your physical shop, but with modern developments in technology this can now be dealt with for you.

A stock management system is the back bone of every retail concern, whether it's a bricks and mortar shop, eCommerce website, factory outlet or pop-up concession. From the smallest retail business to the biggest superstore, effective stock control is essential. Finding the right products to sell, when to sell them and not to be left with merchandise that's eating away at your profit is a challenge for any business. Not only is it frustrating to sell excess stock at a reduced to clear price, it's also corrosive to your company's resources.

TABLE OF FIGURES

Figure 1 Inventory management system.....	2
Figure 2 Bootstrap	8
Figure 3 JQuery.....	9
Figure 4 Datatables plug-in for jquery.....	10
Figure 5 JQueryUI.....	11
Figure 6 Krajee jquery plugins.....	12
Figure 7 Fullcalendar	14
Figure 8 Font awesome.....	16
Figure 9 Phpmyadmin starter.....	17
Figure 10 Mastering phpmyadmin	17
Figure 11 Phpmyadmin.....	18
Figure 12 Category addition	22
Figure 13 Users table	22
Figure 14 Structure	23
Figure 15 Manage product.....	23
Figure 16 Product addition.....	23

TABLE OF CONTENTS

Contents

CERTIFICATE.....	i
DECLARATION.....	ii
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
CHAPTER 1: INTRODUCTION	1
CHAPTER 2: SOFTWARE REQUIREMENT	7
CHAPTER 3: SOFTWARE DESCRIPTION	8
BOOTSTRAP	8
JQUERY	9
DATATABLES IS A PLUG-IN FOR THE JQUERY JAVASCRIPT	10
JQUERY UI.....	11
KRAJEE JQUERY PLUGINS.....	12
FULLCALENDAR.....	14
FONT AWESOME	16
PHPMYADMIN.....	17
CONCLUSION	20
REFERENCES	21
SCREEN OUTPUT	22
APPENDIX	24

CHAPTER 1: INTRODUCTION

Managing inventory is the ongoing process of moving parts and products into and out of a company's location(s). Companies manage their inventory on a daily basis as they place new orders for products and ship orders out to customers. It's important that business leaders gain a firm grasp of everything involved in the inventory control process. That way, they can figure out creative ways to solve inventory challenges by finding the right solutions.

When most people think of inventory management, they think of retail applications. While retail operations rely heavily on inventory management, inventory management systems are widely used in a variety of industries, from manufacturing to utilities, healthcare, education, government, and more. Inventory management systems streamline and centralize the process for controlling the flow and maintenance of inventory to ensure that the right amount of inventory is available at the right time and of the right quality.

Online Inventory Management Software with procedural PHP, MYSQLI, BOOTSTRAP, and JQUERY AJAX. This project, Online Inventory Management Software is developed with PHP, MYSQLI, BOOTSTRAP, and AJAX JQUERY as a web application. This inventory management software provides simple inventory control and sales management. This is based on the web application.

Definition of an Inventory Management System

An inventory management system is the combination of technology (hardware and software) and processes and procedures that oversee the monitoring and maintenance of stocked products, whether those products are company assets, raw materials and supplies, or finished products ready to be sent to vendors or end consumers. A complete inventory management system consists of:

- A system for identifying every inventory item and its associated information, such as barcode labels or asset tags.
- Hardware tools for reading barcode labels, such as handheld barcode scanners or smartphones with barcode scanning apps.
- Inventory management software, which provides a central database and point of reference for all inventory, coupled with the ability to analyze data, generate reports, forecast future demand, and more.
- Processes and policies for labeling, documentation, and reporting. This should include an inventory management technique such as Just in Time, ABC Analysis, First-In First-Out (FIFO), Stock Review, or another proven methodology.
- People who trained to follow these policies and processes.

SearchERP defines inventory management as “the supervision of non-capitalized assets (inventory) and stock items.” Inventory management is a component of supply chain management that oversees the flow of items (products, goods, etc.) as they move from the manufacturer to the warehouse and then to the point of sale. “A key function of inventory management is to keep a detailed record of each new or returned product as it enters or leaves a warehouse or point of sale,” SearchERP explains, which points to the importance of having a clear and established inventory management system to ensure that the process and documentation are as streamlined and efficient as possible, as well as to minimize error.



Figure 1 Inventory management system

Benefits of Inventory Management Systems

Without an inventory management system, the goods and products that flow through an organization will inevitably be in disarray. An inventory management system enables a company to maintain a centralized record of every asset and item in the control of the organization, providing a single source of truth for the location of every item, vendor and supplier information, specifications, and the total number of a particular item currently in stock.

Because inventory often consists of movable assets, inventory management systems are critical for keeping tabs on current stock levels and understanding what items move quickly and which items are more slow-moving, which in turn enables organizations to determine when it's time to reorder with greater accuracy. Overall, a comprehensive inventory management system offers countless benefits to companies including:

- Improved cash flow

- Better reporting and forecasting capabilities
- Reduction in storage costs (overhead)
- Reduced labor costs
- Reduction in dead stock
- Better organization
- Enhanced transparency
- Improved supplier, vendor, and partner relationships

INVENTORY CONTROL VS. INVENTORY MANAGEMENT

Challenges

Without a doubt, tracking inventory is an important part of doing business. It is the means by which companies remain stocked with all of the essential goods they need to sell to customers and complete daily operations. When products are sold, they must be replenished at a rate that doesn't lead to huge overstocks or frequent stock-outs.

Successfully managing inventory means hitting a moving target because supply and demand aren't constant, but they often change with the season and over the course of a product's life-cycle. There's also the problem of keeping track of inventory with different units of measure and suppliers in different countries with different tax rates and laws.

Attempts to Solve

To solve the challenge of properly managing inventory, companies often turn to software that can help them keep track of all of their inventory, orders, vendors and more. Startups frequently record inventory quantities in Microsoft Excel because it's easily accessible and they don't have to put a lot of money into obtaining it. But they find that Excel lacks the nuanced and comprehensive features they need to analyze data, spot trends in sales and supply chains, automatically update inventory records from a barcode scanner, and keep their data secure yet accessible.

Using a simple inventory solution is a step up from Excel, but its limitations soon leads growing businesses to switch to a more advanced solution over time. This leads to wasted time and money spent learning a new system every few years.

Solutions

A better solution for tracking inventory is to find software that is affordable and robust enough for small and midsize businesses to stay with for many hours. It has features that are simple enough for a young business to master, but that can also be expanded to fit that

same business's increasingly complex needs year after year. Fishbowl is a powerful inventory management system that fits into this category.

Fishbowl Manufacturing and Fishbowl Warehouse help you oversee multiple locations, track shipments, automatically reorder products, print and scan barcodes, convert units of measurement, use multi-currency features, create multilevel work orders, and much more. Fishbowl integrates with dozens of other business solutions, including Amazon, eBay, QuickBooks, Salesforce, and Xero, just to name a few. So you can share your inventory data across all of your other online and desktop platforms to ensure each one has accurate, up-to-date inventory quantities and values.

Best Practices for Inventory Management Systems

Cleary Inventory explains that a good inventory management system can help to enhance productivity, but only if you set it up with some basic essentials including:

- Location names
- Easy-to-read location labels
- Unique item identification numbers
- Units of measure
- A starting count
- A software solution that effectively monitors and tracks activity
- Clear, company-wide policies and processes
- People who know how to support these policies and processes

Implementing a comprehensive inventory management system can be complex. It's not merely a matter of purchasing a good software program; an inventory management system must address the people, processes, and technologies from end-to-end. Following best practices to set up a comprehensive, company-wide inventory management system is an investment that will pay for itself again and again through greater efficiency and a boost to the bottom line

Usefulness

Fishbowl is an ideal inventory solution for many types of businesses, including:

- Manufacturers that work with hundreds and even thousands of parts, components, and other materials.
- Wholesale distributors with multiple warehouses and suppliers both inside and outside their home country.
- Service providers that must maintain specific numbers of cleaning supplies, replacement parts, and other items to serve their customers.
- Food producers using ingredients that must be consumed before reaching certain expiration dates.

- Government agencies looking to track their assets across wide expanses and to serve a large number of people.
- And many others, ranging from SMBs to large organizations.

With Fishbowl, you can speed up your inventory management processes by using barcode scanners, save money by balancing your inventory needs, and get more out of your workers by giving them the tools to be more productive. Eliminate double data entry by integrating Fishbowl's inventory software with other business solutions, such as QuickBooks' and Xero's accounting software.

Convert multiple currencies into your home currency in purchase orders and sales orders to help with taxes and other calculations. And use production stages in multilevel work orders and bills of materials to ensure manufacturing jobs get done in the right order in the right timeframe to keep up with demand. That's a tall order, but Fishbowl is up to the task.

Fishbowl is the total automated inventory system. Start improving your company's inventory control today by starting a 14-day free trial of Fishbowl!

TL, DR SUMMARY

Inventory management is the process of tracking items as they enter and leave a warehouse. There are many challenges, such as shortages, carrying costs, and supply chain disruptions, which can be solved by using an inventory solution.

Top 6 Features to Look For In an Inventory Management System:

According to a survey conducted by Software Advice, the top six most requested inventory management features that you should be looking for in a solution include:

1. Basic Inventory Control

As simple as this sound, it tops the list of most requested features. What this means is having a systematic way to truly 'manage' and control your inventory across all of your sales channels. Very often new businesses work with spreadsheet(s) where formulas can break, and a lot of manual effort is required to keep inventory on track. For them, using a spreadsheet is the first level of business automation. These 'first-time automatons', as we refer to them, may be growing rapidly and need a better, more efficient way to manage their current business, and position themselves for future growth. They're not always aware of the systems that are out there that are tailor-made to address their growing needs.

2. Barcoding & Scanning

Increasing the number of daily orders you receive is exciting, but it can also lead to bottlenecks if you're not positioned to process and fulfill those orders efficiently.

Retailers who are processing a higher volume of orders look to streamline their processes even more by implementing barcoding and scanning at the pick/pack/ship stages.

3. Demand Forecasting

Demand forecasting is key to planning what to order so you spend resources wisely. With the proper tools in place, you can learn more than just which products are selling and which channels are performing. Having the ability to drill down into the attributes of your sales, such as size, color, scent or material, provides you essential information to help you keep more accurate levels of inventory and make better decisions about what and how much to buy.

4. Accounting Integration

While an accounting system can't do the work of a specialized order and inventory management system, it is necessary to run a successful business. Having a direct integration to your accounting package from an order and inventory management system makes all the difference. It enables you to easily post to your General Ledger account and keep track of all your other business expenses outside of inventory.

5. Lot Tracking

Retailers who sell FDA regulated products understand the importance of identifying specific stock items, or lots. If a product gets recalled for some reason, lot tracking allows you to know which of those items are in stock and need to be pulled from your inventory. You can also track which customers have already, or are about to receive, a product from that lot and notify them to ensure the safety of your customers. Tracking can be done by raw material, finished good level or both, whichever your business requires.

6. Support For Kits (aka Bundled Products; kitting)

Many retailers have added kits or bundles to their product mix, meaning the product for sale is not a single item, but a combination of products sold as a kit or bundle such as a gift basket or a collection of items in a set. Managing these types of products is more complex from an inventory standpoint. With an inventory management solution that supports kitted products, you can see how many kits can be assembled, what 'kit' items need to be re-ordered, etc.

CHAPTER 2: SOFTWARE REQUIREMENT

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed.

Requirements analysis is an important aspect of project management. Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

Software requirements can be mentioned in a SRS. A **software requirements specification** (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Supported Operating Systems

Windows 8.0

Software Requirements

Bootstrap

jQuery

DataTables is a plug-in for the jQuery Javascript

jQuery UI

Krajee JQuery Plugins

FullCalendar

FontAwesome

phpMyAdmin

CHAPTER 3: SOFTWARE DESCRIPTION

BOOTSTRAP

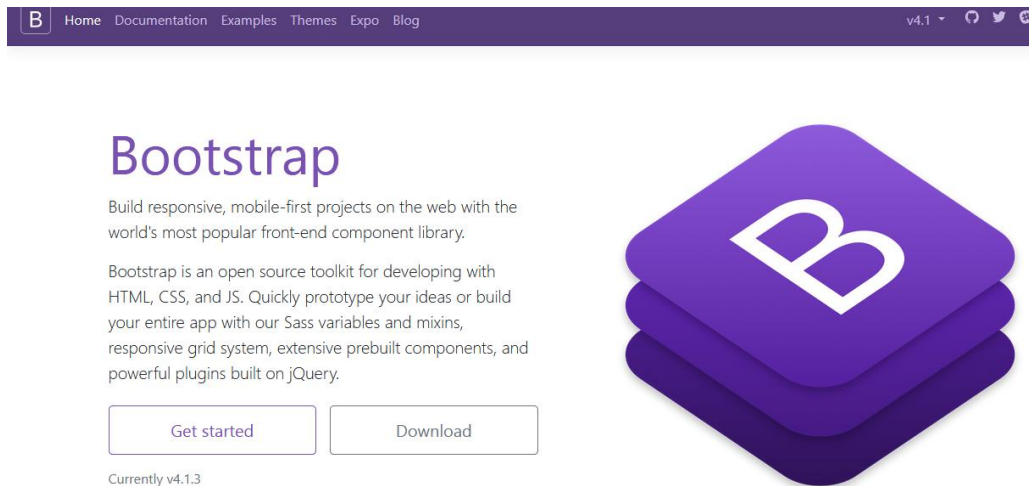


Figure 2 Bootstrap

Bootstrap is a free and open-source front-end framework for designing websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. Unlike many earlier web frameworks, it concerns itself with front-end development only. Bootstrap is the second most-starred project on GitHub, with more than 126,000 stars.

Get started with Bootstrap, the world's most popular framework for building responsive, mobile-first sites, with BootstrapCDN and a template starter page.

Bootstrap 3 supports the latest versions of the Google Chrome, Firefox, Internet Explorer, Opera, and Safari (except on Windows). It additionally supports back to IE8 and the latest Firefox Extended Support Release (ESR).

Since 2.0, Bootstrap supports responsive web design. This means the layout of web pages adjusts dynamically, taking into account the characteristics of the device used (desktop, tablet, mobile phone).

Starting with version 3.0, Bootstrap adopted a mobile-first design philosophy, emphasizing responsive design by default.

JQUERY

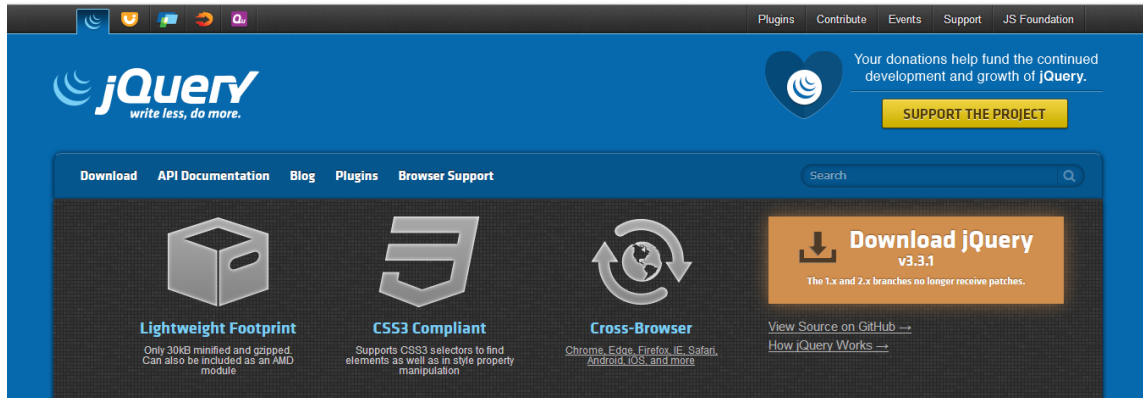


Figure 3 JQuery

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

A Brief Look

DOM Traversal and Manipulation

Get the <button> element with the class 'continue' and change its HTML to 'Next Step...'

```
1 $( "button.continue" ).html( "Next Step..." )
```

Event Handling

Show the #banner-message element that is hidden with display:none in its CSS when any button in #button-container is clicked.

```
1 var hiddenBox = $( "#banner-message" );
2 $( "#button-container button" ).on( "click", function( event ) {
3   hiddenBox.show();
4 });
```


DATATABLES IS A PLUG-IN FOR THE JQUERY JAVASCRIPT

DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, build upon the foundations of progressive enhancement, that adds all of these advanced features to any HTML table.

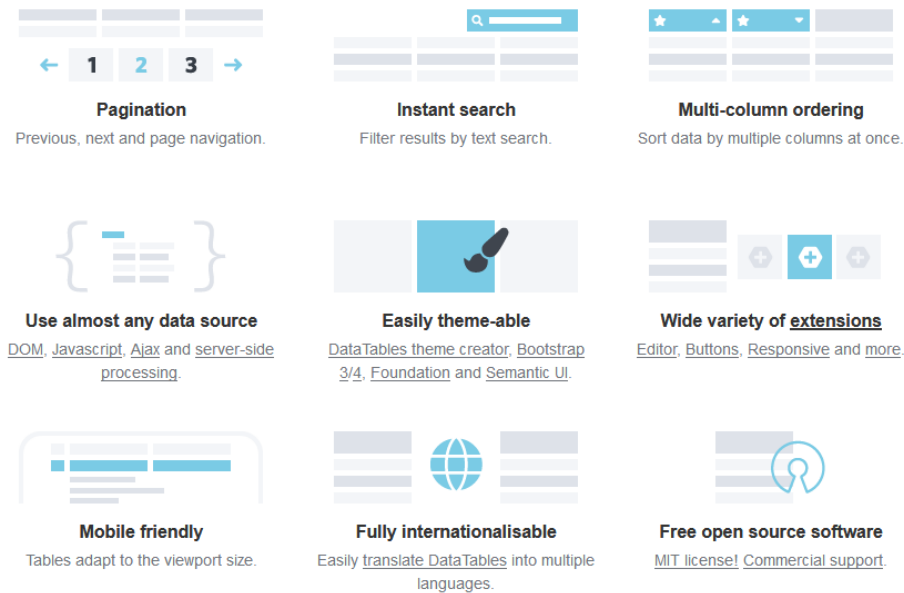


Figure 4 DataTables plug-in for jquery

The first step is to obtain the library from the DataTables website. 1.9.4 is the current stable version and 1.10 is in development now. If you don't want to download the files, you can also make use of the Microsoft CDN. As it is a jQuery plugin you should include the jQuery library too, preferably the latest version.

DataTables can work with data from a verity of sources. It can directly work on an HTML table or we can specify data as an array while initialization. Or it can work on data coming from an Ajax source.

In this article we'll create a listing of websites in the SitePoint network. We'll start with the listing of a few site names, and then we'll add more columns and features. Here we have an HTML table with one column that lists the names of just three sites. Let's see what DataTables can do with a minimal setup.

JQUERY UI

jQuery UI is a curated set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library. Whether you're building highly interactive web applications or you just need to add a date picker to a form control, jQuery UI is the perfect choice.

New theme: New theme: Our long-time default grayscale theme Smoothness has been succeeded by a modern theme with some colors, less gradients and some other improvements.

New widgets: jQuery UI 1.12 includes widgets: Controlgroup replaces buttonset and adds support for selectmenu and the new checkboxradio widget. We've added support for using jQuery UI with AMD and Bower. We also have over 50 bug fixes.

New classes option: All widgets got a new option, called classes. This gives a lot more control over visual styling and custom theming. For example, building a Bootstrap theme for jQuery UI is now a lot easier.

Interested in the full details of what changed? Check out the 1.12 upgrade guide, 1.12.0 changelog, and 1.12.1 changelog.

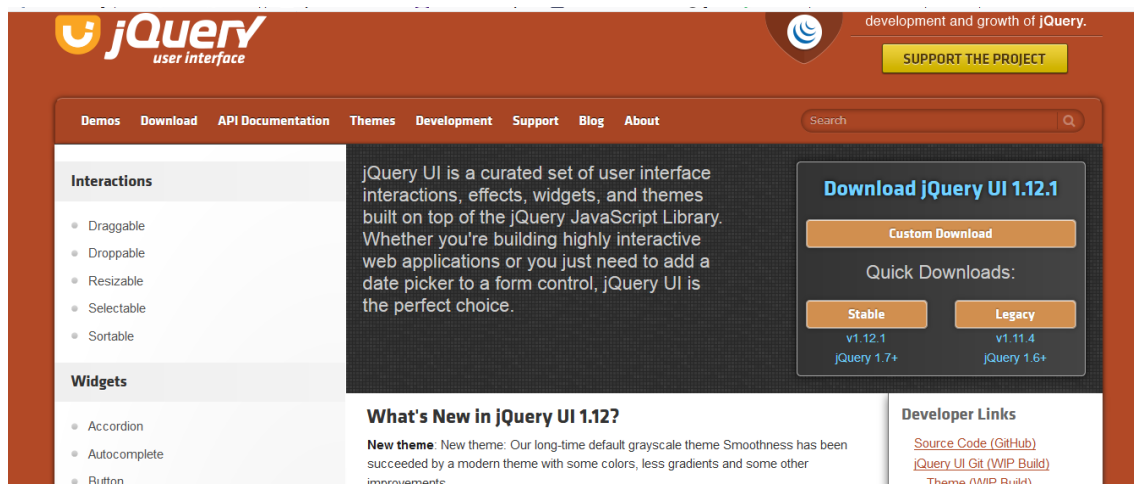


Figure 5 JQueryUI

KRAJEE JQUERY PLUGINS

A Bootstrap styled markdown editor that converts a native HTML textarea to an advanced markdown editor. The editor offers live preview, export, full screen mode, and more features. The editor includes inbuilt support for markdown parsing using markdown-it JS based parser. However, the editor can support any markdown parser via javascript library / method or even a server based parser via an ajax action. Other markdown parsers are configurable (both as a server call OR a local JS method/library). In addition, the plugin allows custom button actions and properties to be setup.



Figure 6 Krajee jquery plugins

Features

- Inbuilt support for advanced markdown syntax editing and formatting via the markdown-it JS based parser. The key features of the markdown-it parser are:
 - It follows the **CommonMark spec** and adds syntax extensions & sugar (URL autolinking, typography).
 - Configurable syntax! You can add new rules and even replace existing ones.
 - High speed.
 - Safe by default.
 - Community-written **plugins** and other packages on npm.

- In addition to supporting all common markdown syntax formats, this also includes support for various markdown-it plugins like emojis, smart arrows, checkboxes, subscript, superscript, definition list, footnote, abbreviation, marked / inserted text etc.
- Advanced additional support for emojis and emoticons to be inserted. The editor also supports using rendering emojis using twitter emojis {twemoji}.
- The editor can be configured to use any markdown based parser OR even use a server based parsing via AJAX action.
- Inbuilt live preview and conversion of the markdown text to HTML formatted text. The editor can be toggled and configured to be set in one of the three modes:
 - *Editor Mode*: Displays only the editor for editing the markdown text.
 - *Preview Mode*: Displays only the preview for viewing the HTML formatted text.
 - *Split Mode*: Shows both the editor and preview modes split side by side.
- One can configure the editor to by default to open in one of the above modes.
- Offers live preview and **synchronized scrolling** between editor and preview when typing text OR scrolling in the editor. This is especially useful in the Split Mode.
- The editor also includes inbuilt support for FULL SCREEN mode toggling and editing. One can toggle the editor to FULL screen for any of the above modes.
- One can also configure the editor to by default open in full screen mode if needed.
- Highly extensible and easily configurable toolbar buttons and actions. Supports 4 different toolbar button locations (Header Left, Header Right, Footer Left, and Footer Right).
- Inbuilt support for unlimited undo and redo of editing actions via an inbuilt UndoStack library.

FULLCALENDAR

FullCalendar is great for displaying events, but it isn't a complete solution for event content-management. Beyond dragging an event to a different time/day, you cannot change an event's name or other associated data. It is up to you to add this functionality through FullCalendar's event hooks.

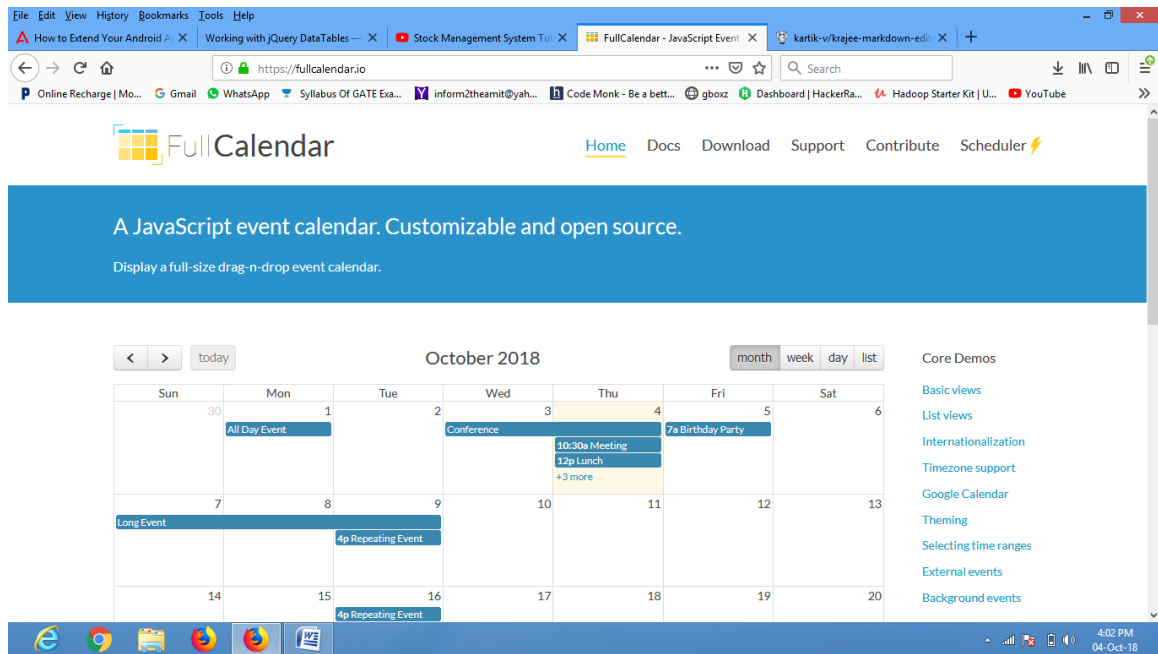


Figure 7 Fullcalendar

FullCalendar Features

Drag-and-drop events

With the fullcalendar module it is possible to drag, drop and resize your events in the calendar itself (if you have the right permissions). You can try this on the FullCalendar demo.

Note:

Only views build around fields are editable. Views build on node:created or user:login aren't editable because they have a hardcoded timestamp. Editing Google Calendar events won't work either at the moment.

Colorbox - open events in a lightbox

Fullcalendar has build in support for the colorbox module.

- Drupal 6 instructions
- Drupal 7 instructions

Google Calendar support

Starting with version 2 Fullcalendar can handle google calendars.

- Drupal 6 instructions
- Drupal 7 instructions

Event coloring

With Fullcalendar it is possible to color your events.

- Drupal 6 instructions
- Drupal 7 instructions

RTL support

Only supported in version 2

This module has build in RTL support.

The only thing you probably would like to do is changing the order of the buttons.

FONT AWESOME

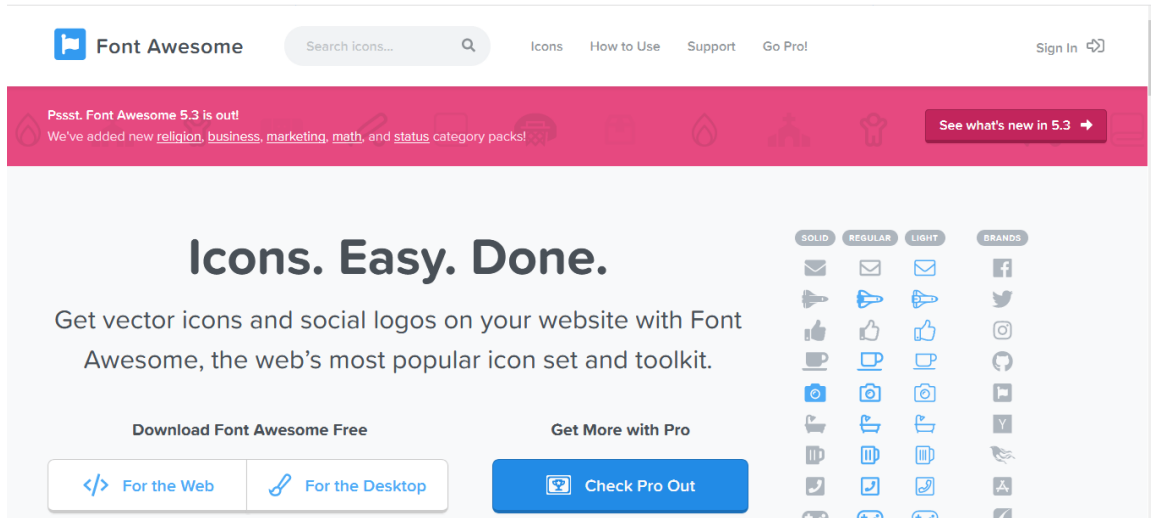


Figure 8 Font awesome

It's as easy as including a class to an `<i>` tag. Example: `<i class='icon-smile'></i>` will display . It couldn't be easier than that. With over 300 icons to choose from, and a number of great examples to show how you can use it, the Font Awesome system is the perfect solution for today's web developers.

Basic Icons

To use the Font Awesome icons, add the following line inside the `<head>` section of your HTML page:

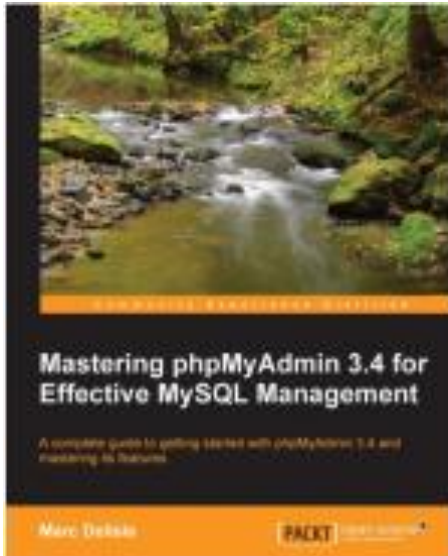
```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Note: No downloading or installation is required!

You place Font Awesome icons by using the prefix `fa` and the icon's name.

PHPMYADMIN

phpMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. phpMyAdmin supports a wide range of operations on MySQL and MariaDB. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.



**Figure 10 Mastering
phpmyadmin**

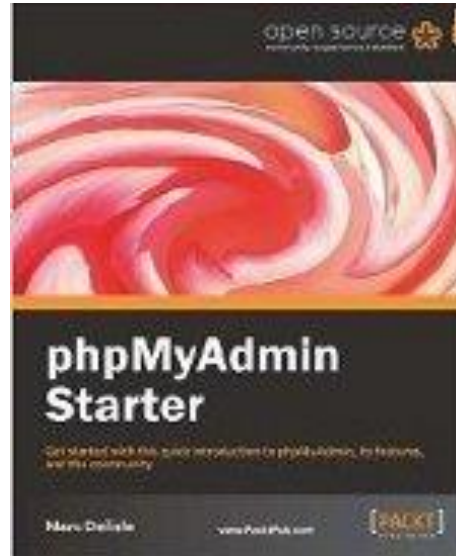


Figure 9 Phpmyadmin starter

phpMyAdmin comes with a wide range of documentation and users are welcome to update our wiki pages to share ideas and howtos for various operations. The phpMyAdmin team will try to help you if you face any problem; you can use a variety of support channels to get help.

phpMyAdmin is also very deeply documented in a book written by one of the developers – Mastering phpMyAdmin for Effective MySQL Management, which is available in English and Spanish.

To ease usage to a wide range of people, phpMyAdmin is being translated into 72 languages and supports both LTR and RTL languages.



Figure 11 Phpmyadmin

Features provided by the program include:

1. Web interface
2. MySQL and MariaDB database management
3. Import data from CSV and SQL
4. Export data to various formats: CSV, SQL, XML, PDF (via the TCPDF library), ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, LaTeX and others
5. Administering multiple servers
6. Creating PDF graphics of the database layout
7. Creating complex queries using query-by-example (QBE)
8. Searching globally in a database or a subset of it
9. Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
10. Live charts to monitor MySQL server activity like connections, processes, CPU/memory usage, etc.
11. Working with different operating systems.
12. Make complex SQL queries easier.

phpMyAdmin is a mature project with a stable and flexible code base; you can find out more about the project and its history and the awards it earned. When the project turned 15, we published a celebration page.

The phpMyAdmin project is a member of Software Freedom Conservancy. SFC is a not-for-profit organization that helps promote, improve, develop, and defend Free, Libre, and Open Source Software (FLOSS) projects.

The phpMyAdmin program is handy for performing maintenance operations on tables, backing up information, and editing things directly in the event that WordPress is not working. Occasionally, in the Support Forums, someone will post a SQL query of some benefit or other that can be run using phpMyAdmin. Although many of the same tasks can be performed on the MySQL command line, doing so is not an option for many people.

CONCLUSION

Company which doesn't have inventory management system will get problem when check whether the products known from the brand are available or not. Besides that, in the day to day activity, without inventory management system, sometimes employee needs to check to warehouse especially if the products are out of stock in all 3 shops and remember; checking to the warehouse will take longer time so it wouldn't be efficient at all.

Benefit of inventory management system for shop day to day activity is list the product that still available in database, update the database whether by reducing the number of stock available in database if the product has been sold or adding the number of stock if there were products come from the distributor or producer, and also keep the transaction history in database about what happened in retail. Besides that, inventory management system will help employee in warehouse to know whether the product in store are available or not, so can help to prevent stock minus of product in warehouse. By using inventory management system, employee will be easier to check required product based on quantity, quality, price, so the activity of employee more effective and more efficient.

104 105 Inventory management system is provided with automatic reporting system.

So each transactions happened will be archiving and organizing in the computer rather than using paper or excel. By record the transaction, it could also help the manager or boss to understand each days profit or loss so it could be said as a tool for decision support system because it will help the boss to add most sell product and no more adding Unproductive stock in warehouse.

REFERENCES

- [1] "Security fix: phpMyAdmin 4.8.3 is released". phpmyadmin.net. Retrieved 2018-08-28.
- [2]"Translations". [phpMyAdmin](http://phpmyadmin.net). Retrieved 2014-12-23.
- [3]"phpMyAdmin Review". [PCWorld](http://www.pcworld.com). 2011-04-20. Retrieved 2017-10-27.
- [4]"phpMyAdmin - About". [phpMyAdmin](http://phpmyadmin.net). Retrieved 2013-03-03.
- [5]"Global market share of Font Script technologies". [Wappalyzer](http://wappalyzer.com). Retrieved 25 January 2015.
- [6]"Font Awesome". fontawesome.com. Retrieved 2017-12-09.
- [7]"jQuery: The write less, do more, JavaScript library". The jQuery Project. Retrieved 29 April 2010.
- [8]Otto, Mark (January 17, 2012). "Bootstrap in A List Apart No. 342". Mark Otto's blog. Archived from the original on October 28, 2016. Retrieved February 23, 2017.
- [9]Otto, Mark (August 19, 2011). "Bootstrap from Twitter". Developer Blog. Twitter. Archived from the original on February 23, 2017. Retrieved February 23, 2017.

SCREEN OUTPUT

The screenshot shows the phpMyAdmin interface for the 'users' table. The table structure is defined as follows:

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index
user_id	INT		None			<input type="checkbox"/>	PRIMARY
username	VARCHAR	255	None			<input type="checkbox"/>	
password	VARCHAR	255	None			<input type="checkbox"/>	
email	INT		None			<input type="checkbox"/>	

Below the table structure, there are fields for 'Table comments:', 'Storage Engine:' (set to InnoDB), and 'Collation:'. A 'PARTITION definition:' field is also present. At the bottom right, there are 'Save' and 'Cancel' buttons.

Figure 13 Users table

The screenshot shows a web application interface with a modal window titled '+ Add Categories'. The modal contains the following fields:

- Categories Name:** A text input field with the value 'Causal'.
- Status:** A dropdown menu with the value 'Available'.

At the bottom of the modal, there are two buttons: 'Close' and 'Save changes'.

Figure 12 Category addition

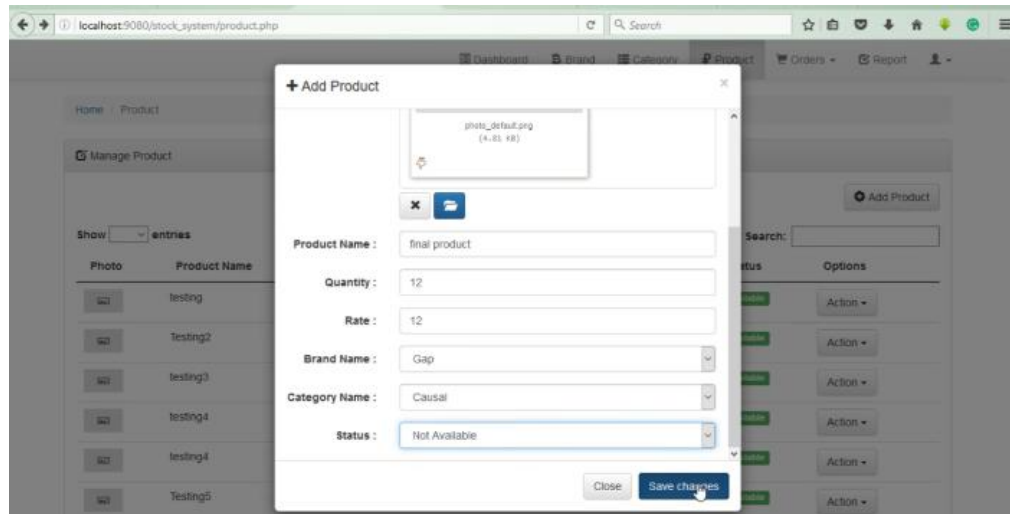


Figure 16 Product addition

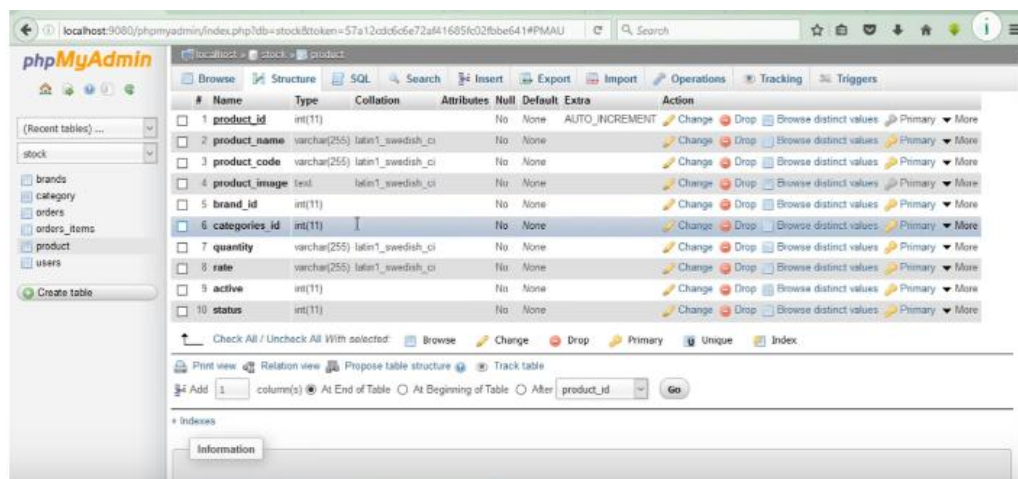


Figure 14 Structure

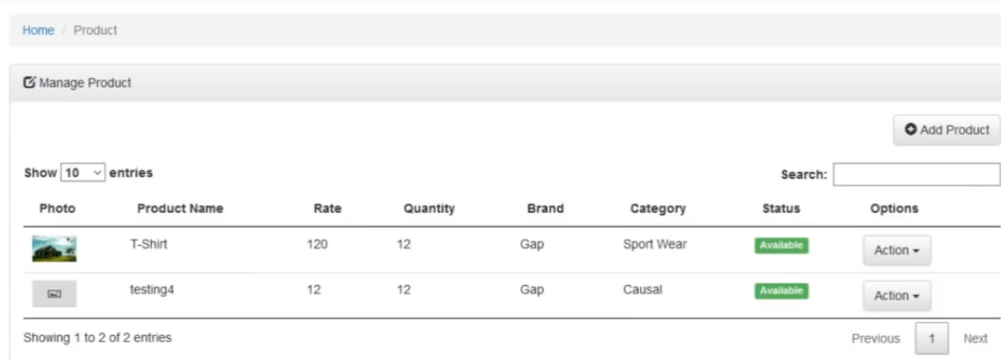


Figure 15 Manage product

APPENDIX

Index.php

```
<?php

require_once 'php_action/db_connect.php';

session_start();

if(isset($_SESSION['userId'])) {

header('location: http://localhost:9080/stock/dashboard.php');

}

$errors = array();

if($_POST) {

$username = $_POST['username'];

$password = $_POST['password'];

if(empty($username) || empty($password)) {

if($username == "") {

$errors[] = "Username is required";

}

if($password == "") {

$errors[] = "Password is required";

}

} else {

$sql = "SELECT * FROM users WHERE username = '$username'";

$result = $connect->query($sql);

if($result->num_rows == 1) {
```

```

$password = md5($password);

// exists

$mainSql = "SELECT * FROM users WHERE username = '$username' AND password
= '$password'";

$mainResult = $connect->query($mainSql);

if($mainResult->num_rows == 1) {

$value = $mainResult->fetch_assoc();

$user_id = $value['user_id'];

// set session

$_SESSION['userId'] = $user_id;

header('location: http://localhost:9080/stock/dashboard.php');

} else{

$errors[] = "Incorrect username/password combination";

} // /else

} else {

$errors[] = "Username doesnot exists";

} // /else

} // /else not empty username // password

} // /if $_POST

?>

<!DOCTYPE html>

<html>

<head>

<title>Stock Management System</title>

```



```

<!-- bootstrap -->
<link rel="stylesheet" href="assests/bootstrap/css/bootstrap.min.css">

<!-- bootstrap theme-->
<link rel="stylesheet" href="assests/bootstrap/css/bootstrap-theme.min.css">

<!-- font awesome -->
<link rel="stylesheet" href="assests/font-awesome/css/font-awesome.min.css">

<!-- custom css -->
<link rel="stylesheet" href="custom/css/custom.css">

<!-- jquery -->
<script src="assests/jquery/jquery.min.js"></script>

<!-- jquery ui -->
<link rel="stylesheet" href="assests/jquery-ui/jquery-ui.min.css">
<script src="assests/jquery-ui/jquery-ui.min.js"></script>

<!-- bootstrap js -->
<script src="assests/bootstrap/js/bootstrap.min.js"></script>

</head>

<body>

<div class="container">

<div class="row vertical">

<div class="col-md-5 col-md-offset-4">

<div class="panel panel-info">

<div class="panel-heading">

<h3 class="panel-title">Please Sign in</h3>

```

```

</div>

<div class="panel-body">

<div class="messages">

<?php if($errors) {
foreach ($errors as $key => $value) {
echo '<div class="alert alert-warning" role="alert">
<i class="glyphicon glyphicon-exclamation-sign"></i>
'.$value.'</div>';
}
} ?>

</div>

<form class="form-horizontal" action="<?php echo $_SERVER['PHP_SELF'] ?>"
method="post" id="loginForm">

<fieldset>

<div class="form-group">

<label for="username" class="col-sm-2 control-label">Username</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="username" name="username"
placeholder="Username" autocomplete="off" />

</div>

</div>

<div class="form-group">

<label for="password" class="col-sm-2 control-label">Password</label>

```

```

<div class="col-sm-10">

<input type="password" class="form-control" id="password" name="password"
placeholder="Password" autocomplete="off" />

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<button type="submit" class="btn btn-default"> <i class="glyphicon glyphicon-log-
in"></i> Sign in</button>

</div>

</div>

</fieldset>

</form>

</div>

<!-- panel-body -->

</div>

<!-- /panel -->

</div>

<!-- /col-md-4 -->

</div>

<!-- /row -->

</div>

<!-- container -->

</body>

```

```
</html>
```

Brand.php

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

```
<div class="row">
```

```
<div class="col-md-12">
```

```
<ol class="breadcrumb">
```

```
<li><a href="dashboard.php">Home</a></li>
```

```
<li class="active">Brand</li>
```

```
</ol>
```

```
<div class="panel panel-default">
```

```
<div class="panel-heading">
```

```
<div class="page-heading"> <i class="glyphicon glyphicon-edit"></i> Manage  
Brand</div>
```

```
</div> <!-- /panel-heading -->
```

```
<div class="panel-body">
```

```
<div class="remove-messages"></div>
```

```
<div class="div-action pull pull-right" style="padding-bottom:20px;">
```

```
<button class="btn btn-default button1" data-toggle="modal" data-  
target="#addBrandModel"> <i class="glyphicon glyphicon-plus-sign"></i> Add Brand  
</button>
```

```
</div> <!-- /div-action -->
```

```
<table class="table" id="manageBrandTable">
```

```
<thead>
```

```
<tr>
```

```

<th>Brand Name</th>

<th>Status</th>

<th style="width:15%;">Options</th>

</tr>

</thead>

</table>

<!-- /table -->

</div> <!-- /panel-body -->

</div> <!-- /panel -->

</div> <!-- /col-md-12 -->

</div> <!-- /row -->

<div class="modal fade" id="addBrandModel" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

<form class="form-horizontal" id="submitBrandForm"
action="php_action/createBrand.php" method="POST">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Add Brand</h4>

</div>

<div class="modal-body">

<div id="add-brand-messages"></div>

<div class="form-group">

```

```

<label for="brandName" class="col-sm-3 control-label">Brand Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="brandName" placeholder="Brand Name"
name="brandName" autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="brandStatus" class="col-sm-3 control-label">Status: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select class="form-control" id="brandStatus" name="brandStatus">

<option value="">~~SELECT~~</option>

<option value="1">Available</option>

<option value="2">Not Available</option>

</select>

</div>

</div> <!-- /form-group-->

</div> <!-- /modal-body -->

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

<button type="submit" class="btn btn-primary" id="createBrandBtn" data-loading-
text="Loading..." autocomplete="off">Save Changes</button>

</div>

```

```

<!-- /modal-footer -->

</form>

<!-- /.form -->

</div>

<!-- /modal-content -->

</div>

<!-- /modal-dailog -->

</div>

<!-- / add modal -->

<!-- edit brand -->

<div class="modal fade" id="editBrandModel" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

<form class="form-horizontal" id="editBrandForm" action="php_action/editBrand.php"
method="POST">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="fa fa-edit"></i> Edit Brand</h4>

</div>

<div class="modal-body">

<div id="edit-brand-messages"></div>

<div class="modal-loading div-hide" style="width:50px; margin:auto;padding-top:50px;
padding-bottom:50px;">

<i class="fa fa-spinner fa-pulse fa-3x fa-fw"></i>

```

```

<span class="sr-only">Loading...</span>

</div>

<div class="edit-brand-result">

<div class="form-group">

<label for="editBrandName" class="col-sm-3 control-label">Brand Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="editBrandName" placeholder="Brand
Name" name="editBrandName" autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="editBrandStatus" class="col-sm-3 control-label">Status: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select class="form-control" id="editBrandStatus" name="editBrandStatus">

<option value="">~~SELECT~~</option>

<option value="1">Available</option>

<option value="2">Not Available</option>

</select>

</div>

</div> <!-- /form-group-->

</div>

<!-- /edit brand result -->

```



```

</div> <!-- /modal-body -->

<div class="modal-footer editBrandFooter">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="submit" class="btn btn-success" id="editBrandBtn" data-loading-
text="Loading..." autocomplete="off"> <i class="glyphicon glyphicon-ok-sign"></i>
Save Changes</button>

</div>

<!-- /modal-footer -->

</form>

<!-- /.form -->

</div>

<!-- /modal-content -->

</div>

<!-- /modal-dailog -->

</div>

<!-- / add modal -->

<!-- /edit brand -->

<!-- remove brand -->

<div class="modal fade" tabindex="-1" role="dialog" id="removeMemberModal">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

```

```

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="glyphicon glyphicon-trash"></i> Remove Brand</h4>

</div>

<div class="modal-body">

<p>Do you really want to remove ?</p>

</div>

<div class="modal-footer removeBrandFooter">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="button" class="btn btn-primary" id="removeBrandBtn" data-loading-
text="Loading..."> <i class="glyphicon glyphicon-ok-sign"></i> Save changes</button>

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

<!-- /remove brand -->

<script src="custom/js/brand.js"></script>

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

```

Category.php

```

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

<div class="row">

<div class="col-md-12">

<ol class="breadcrumb">

```

```

<li><a href="dashboard.php">Home</a></li>

<li class="active">Category</li>

</ol>

<div class="panel panel-default">

<div class="panel-heading">

<div class="page-heading"> <i class="glyphicon glyphicon-edit"></i> Manage
Categories</div>

</div> <!-- /panel-heading -->

<div class="panel-body">

<div class="remove-messages"></div>

<div class="div-action pull pull-right" style="padding-bottom:20px;">

<button class="btn btn-default button1" data-toggle="modal"
id="addCategoriesModalBtn" data-target="#addCategoriesModal"> <i class="glyphicon
glyphicon-plus-sign"></i> Add Categories </button>

</div> <!-- /div-action -->

<table class="table" id="manageCategoriesTable">

<thead>

<tr>

<th>Categories Name</th>

<th>Status</th>

<th style="width:15%;">Options</th>

</tr>

</thead>

</table>

<!-- /table -->

```

```
</div> <!-- /panel-body -->
```

```
</div> <!-- /panel -->
```

```
</div> <!-- /col-md-12 -->
```

```
</div> <!-- /row -->
```

```
<!-- add categories -->
```

```
<div class="modal fade" id="addCategoriesModal" tabindex="-1" role="dialog">
```

```
<div class="modal-dialog">
```

```
<div class="modal-content">
```

```
<form class="form-horizontal" id="submitCategoriesForm"
action="php_action/createCategories.php" method="POST">
```

```
<div class="modal-header">
```

```
<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>
```

```
<h4 class="modal-title"><i class="fa fa-plus"></i> Add Categories</h4>
```

```
</div>
```

```
<div class="modal-body">
```

```
<div id="add-categories-messages"></div>
```

```
<div class="form-group">
```

```
<label for="categoriesName" class="col-sm-4 control-label">Categories Name: </label>
```

```
<label class="col-sm-1 control-label">: </label>
```

```
<div class="col-sm-7">
```

```
<input type="text" class="form-control" id="categoriesName" placeholder="Categories
Name" name="categoriesName" autocomplete="off">
```

```

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="categoriesStatus" class="col-sm-4 control-label">Status: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-7">

<select class="form-control" id="categoriesStatus" name="categoriesStatus">

<option value="">~~SELECT~~</option>

<option value="1">Available</option>

<option value="2">Not Available</option>

</select>

</div>

</div> <!-- /form-group-->

</div> <!-- /modal-body -->

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="submit" class="btn btn-primary" id="createCategoriesBtn" data-loading-
text="Loading..." autocomplete="off"> <i class="glyphicon glyphicon-ok-sign"></i>
Save Changes</button>

</div> <!-- /modal-footer -->

</form> <!-- /.form -->

</div> <!-- /modal-content -->

</div> <!-- /modal-dailog -->

</div>

```

```

<!-- /add categories -->

<!-- edit categories brand -->

<div class="modal fade" id="editCategoriesModal" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

<form class="form-horizontal" id="editCategoriesForm"
action="php_action/editCategories.php" method="POST">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="fa fa-edit"></i> Edit Brand</h4>

</div>

<div class="modal-body">

<div id="edit-categories-messages"></div>

<div class="modal-loading div-hide" style="width:50px; margin:auto;padding-top:50px;
padding-bottom:50px;">

<i class="fa fa-spinner fa-pulse fa-3x fa-fw"></i>

<span class="sr-only">Loading...</span>

</div>

<div class="edit-categories-result">

<div class="form-group">

<label for="editCategoriesName" class="col-sm-4 control-label">Categories Name:
</label>

<label class="col-sm-1 control-label">: </label>

```

```

<div class="col-sm-7">

<input type="text" class="form-control" id="editCategoriesName"
placeholder="Categories Name" name="editCategoriesName" autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="editCategoriesStatus" class="col-sm-4 control-label">Status: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-7">

<select class="form-control" id="editCategoriesStatus" name="editCategoriesStatus">

<option value="">~~SELECT~~</option>

<option value="1">Available</option>

<option value="2">Not Available</option>

</select>

</div>

</div> <!-- /form-group-->

</div>

<!-- /edit brand result -->

</div> <!-- /modal-body -->

<div class="modal-footer editCategoriesFooter">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="submit" class="btn btn-success" id="editCategoriesBtn" data-loading-
text="Loading..." autocomplete="off"> <i class="glyphicon glyphicon-ok-sign"></i>
Save Changes</button>

```

```

</div>

<!-- /modal-footer -->

</form>

<!-- /.form -->

</div>

<!-- /modal-content -->

</div>

<!-- /modal-dailog -->

</div>

<!-- /categories brand -->

<!-- categories brand -->

<div class="modal fade" tabindex="-1" role="dialog" id="removeCategoriesModal">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="glyphicon glyphicon-trash"></i> Remove Brand</h4>

</div>

<div class="modal-body">

<p>Do you really want to remove ?</p>

</div>

<div class="modal-footer removeCategoriesFooter">

```



```

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="button" class="btn btn-primary" id="removeCategoriesBtn" data-loading-
text="Loading..."> <i class="glyphicon glyphicon-ok-sign"></i> Save changes</button>

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

<!-- /categories brand -->

<script src="custom/js/categories.js"></script>

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

```

Product.php

```

<?php require_once 'php_action/db_connect.php' ?>

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

<div class="row">

<div class="col-md-12">

<ol class="breadcrumb">

<li><a href="dashboard.php">Home</a></li>

<li class="active">Product</li>

</ol>

<div class="panel panel-default">

<div class="panel-heading">

<div class="page-heading"> <i class="glyphicon glyphicon-edit"></i> Manage
Product</div>

```

```

</div> <!-- /panel-heading -->

<div class="panel-body">

<div class="remove-messages"></div>

<div class="div-action pull pull-right" style="padding-bottom:20px;">

<button class="btn btn-default button1" data-toggle="modal" id="addProductModalBtn"
data-target="#addProductModal"> <i class="glyphicon glyphicon-plus-sign"></i> Add
Product </button>

</div> <!-- /div-action -->

<table class="table" id="manageProductTable">

<thead>

<tr>

<th style="width:10%;">Photo</th>

<th>Product Name</th>

<th>Rate</th>

<th>Quantity</th>

<th>Brand</th>

<th>Category</th>

<th>Status</th>

<th style="width:15%;">Options</th>

</tr>

</thead>

</table>

<!-- /table -->

</div> <!-- /panel-body -->

```

```

</div> <!-- /panel -->

</div> <!-- /col-md-12 -->

</div> <!-- /row -->

<!-- add product -->

<div class="modal fade" id="addProductModal" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">


<form class="form-horizontal" id="submitProductForm"
action="php_action/createProduct.php" method="POST" enctype="multipart/form-
data">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="fa fa-plus"></i> Add Product</h4>

</div>

<div class="modal-body" style="max-height:450px; overflow:auto;">

<div id="add-product-messages"></div>

<div class="form-group">

<label for="productImage" class="col-sm-3 control-label">Product Image: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<!-- the avatar markup -->

<div id="kv-avatar-errors-1" class="center-block" style="display:none;"></div>

<div class="kv-avatar center-block">

```

```

<input type="file" class="form-control" id="productImage" placeholder="Product Name"
name="productImage" class="file-loading" style="width:auto;"/>

</div>

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="productName" class="col-sm-3 control-label">Product Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="productName" placeholder="Product
Name" name="productName" autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="quantity" class="col-sm-3 control-label">Quantity: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="quantity" placeholder="Quantity"
name="quantity" autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="rate" class="col-sm-3 control-label">Rate: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

```

```

<input type="text" class="form-control" id="rate" placeholder="Rate" name="rate"
autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="brandName" class="col-sm-3 control-label">Brand Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select class="form-control" id="brandName" name="brandName">

<option value="">~~SELECT~~</option>

<?php

$sql = "SELECT brand_id, brand_name, brand_active, brand_status FROM brands
WHERE brand_status = 1 AND brand_active = 1";

$result = $connect->query($sql);

while($row = $result->fetch_array()) {

echo "<option value='".$row[0]."'>".$row[1]."</option>";

} // while

?>

</select>

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="categoryName" class="col-sm-3 control-label">Category Name: </label>

<label class="col-sm-1 control-label">: </label>

```

```

<div class="col-sm-8">

<select type="text" class="form-control" id="categoryName" placeholder="Product
Name" name="categoryName" >

<option value="">~~SELECT~~</option>

<?php

$sql = "SELECT categories_id, categories_name, categories_active, categories_status
FROM categories WHERE categories_status = 1 AND categories_active = 1";

$result = $connect->query($sql);

while($row = $result->fetch_array()) {
echo "<option value='".$row[0]."'>".$row[1]."</option>";
} // while
?>

</select>

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="productStatus" class="col-sm-3 control-label">Status: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select class="form-control" id="productStatus" name="productStatus">

<option value="">~~SELECT~~</option>

<option value="1">Available</option>

<option value="2">Not Available</option>

```

```

</select>

</div>

</div> <!-- /form-group -->

</div> <!-- /modal-body -->

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="submit" class="btn btn-primary" id="createProductBtn" data-loading-
text="Loading..." autocomplete="off"> <i class="glyphicon glyphicon-ok-sign"></i>
Save Changes</button>

</div> <!-- /modal-footer -->

</form> <!-- /.form -->

</div> <!-- /modal-content -->

</div> <!-- /modal-dailog -->

</div>

<!-- /add categories -->

<!-- edit categories brand -->

<div class="modal fade" id="editProductModal" tabindex="-1" role="dialog">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="fa fa-edit"></i> Edit Product</h4>

```

```

</div>

<div class="modal-body" style="max-height:450px; overflow:auto;">

<div class="div-loading">

<i class="fa fa-spinner fa-pulse fa-3x fa-fw"></i>

<span class="sr-only">Loading...</span>

</div>

<div class="div-result">

<!-- Nav tabs -->

<ul class="nav nav-tabs" role="tablist">

<li role="presentation" class="active"><a href="#photo" aria-controls="home"
role="tab" data-toggle="tab">Photo</a></li>

<li role="presentation"><a href="#productInfo" aria-controls="profile" role="tab" data-
toggle="tab">Product Info</a></li>

</ul>

<!-- Tab panes -->

<div class="tab-content">

<div role="tabpanel" class="tab-pane active" id="photo">

<form action="php_action/editProductImage.php" method="POST"
id="updateProductImageForm" class="form-horizontal" enctype="multipart/form-data">

<br />

<div id="edit-productPhoto-messages"></div>

<div class="form-group">

<label for="editProductImage" class="col-sm-3 control-label">Product Image: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

```



```

<img src="" id="getProductImage" class="thumbnail" style="width:250px;
height:250px;" />

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="editProductImage" class="col-sm-3 control-label">Select Photo: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<!-- the avatar markup -->

<div id="kv-avatar-errors-1" class="center-block" style="display:none;"></div>

<div class="kv-avatar center-block">

<input type="file" class="form-control" id="editProductImage" placeholder="Product
Name" name="editProductImage" class="file-loading" style="width:auto;"/>

</div>

</div>

</div> <!-- /form-group-->

<div class="modal-footer editProductPhotoFooter">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<!-- <button type="submit" class="btn btn-success" id="editProductImageBtn" data-
loading-text="Loading..."> <i class="glyphicon glyphicon-ok-sign"></i> Save
Changes</button> -->

</div>

<!-- /modal-footer -->

</form>

```

```

<!-- /form -->

</div>

<!-- product image -->

<div role="tabpanel" class="tab-pane" id="productInfo">

<form class="form-horizontal" id="editProductForm"
action="php_action/editProduct.php" method="POST">

<br />

<div id="edit-product-messages"></div>

<div class="form-group">

<label for="editProductName" class="col-sm-3 control-label">Product Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="editProductName" placeholder="Product
Name" name="editProductName" autocomplete="off">

</div>

</div> <!-- /form-group -->

<div class="form-group">

<label for="editQuantity" class="col-sm-3 control-label">Quantity: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="editQuantity" placeholder="Quantity"
name="editQuantity" autocomplete="off">

</div>

</div> <!-- /form-group -->

<div class="form-group">

```

```

<label for="editRate" class="col-sm-3 control-label">Rate: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<input type="text" class="form-control" id="editRate" placeholder="Rate"
name="editRate" autocomplete="off">

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="editBrandName" class="col-sm-3 control-label">Brand Name: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select class="form-control" id="editBrandName" name="editBrandName">

<option value="">~~SELECT~~</option>

<?php

$sql = "SELECT brand_id, brand_name, brand_active, brand_status FROM brands
WHERE brand_status = 1 AND brand_active = 1";

$result = $connect->query($sql);

while($row = $result->fetch_array()) {

echo "<option value='".$row[0]."'>".$row[1]."</option>";

} // while

?>

</select>

</div>

</div> <!-- /form-group-->

```

```

<div class="form-group">

<label for="editCategoryName" class="col-sm-3 control-label">Category Name:
</label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select type="text" class="form-control" id="editCategoryName"
name="editCategoryName" >

<option value="">~~SELECT~~</option>

<?php

$sql = "SELECT categories_id, categories_name, categories_active, categories_status
FROM categories WHERE categories_status = 1 AND categories_active = 1";

$result = $connect->query($sql);

while($row = $result->fetch_array()) {

echo "<option value='".$row[0]."'>".$row[1]."</option>";

} // while

?>

</select>

</div>

</div> <!-- /form-group-->

<div class="form-group">

<label for="editProductStatus" class="col-sm-3 control-label">Status: </label>

<label class="col-sm-1 control-label">: </label>

<div class="col-sm-8">

<select class="form-control" id="editProductStatus" name="editProductStatus">

<option value="">~~SELECT~~</option>

```

```

<option value="1">Available</option>

<option value="2">Not Available</option>

</select>

</div>

</div> <!-- /form-group -->

<div class="modal-footer editProductFooter">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="submit" class="btn btn-success" id="editProductBtn" data-loading-
text="Loading..."> <i class="glyphicon glyphicon-ok-sign"></i> Save Changes</button>

</div> <!-- /modal-footer -->

</form> <!-- /.form -->

</div>

<!-- /product info -->

</div>

</div>

</div> <!-- /modal-body -->

</div>

<!-- /modal-content -->

</div>

<!-- /modal-dailog -->

</div>

<!-- /categories brand -->

```

```

<!-- categories brand -->

<div class="modal fade" tabindex="-1" role="dialog" id="removeProductModal">

<div class="modal-dialog">

<div class="modal-content">

<div class="modal-header">

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span
aria-hidden="true">&times;</span></button>

<h4 class="modal-title"><i class="glyphicon glyphicon-trash"></i> Remove
Product</h4>

</div>

<div class="modal-body">

<div class="removeProductMessages"></div>

<p>Do you really want to remove ?</p>

</div>

<div class="modal-footer removeProductFooter">

<button type="button" class="btn btn-default" data-dismiss="modal"> <i
class="glyphicon glyphicon-remove-sign"></i> Close</button>

<button type="button" class="btn btn-primary" id="removeProductBtn" data-loading-
text="Loading..."> <i class="glyphicon glyphicon-ok-sign"></i> Save changes</button>

</div>

</div><!-- /.modal-content -->

</div><!-- /.modal-dialog -->

</div><!-- /.modal -->

<!-- /categories brand -->

<script src="custom/js/product.js"></script>

```

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

Setting.php

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

```
<?php
```

```
$user_id = $_SESSION['userId'];
```

```
$sql = "SELECT * FROM users WHERE user_id = {$user_id}";
```

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

```
$result = $query->fetch_assoc();
```

```
$connect->close();
```

```
?>
```

```
<div class="row">
```

```
<div class="col-md-12">
```

```
<ol class="breadcrumb">
```

```
<li><a href="dashboard.php">Home</a></li>
```

```
<li class="active">Setting</li>
```

```
</ol>
```

```

<div class="panel panel-default">

<div class="panel-heading">

<div class="page-heading"> <i class="glyphicon glyphicon-wrench"></i> Setting</div>

</div> <!-- /panel-heading -->

<div class="panel-body">

<form action="php_action/changeUsername.php" method="post" class="form-
horizontal" id="changeUsernameForm">

<fieldset>

<legend>Change Username</legend>

<div class="changeUsenrameMessages"></div>

<div class="form-group">

<label for="username" class="col-sm-2 control-label">Username</label>

<div class="col-sm-10">

<input type="text" class="form-control" id="username" name="username"
placeholder="Username" value="<?php echo $result['username']; ?>"/>

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<input type="hidden" name="user_id" id="user_id" value="<?php echo $result['user_id']
?>" />

<button type="submit" class="btn btn-success" data-loading-text="Loading..."
id="changeUsernameBtn"> <i class="glyphicon glyphicon-ok-sign"></i> Save Changes
</button>

</div>

```



```

</div>

</fieldset>

</form>

<form action="php_action/changePassword.php" method="post" class="form-horizontal"
id="changePasswordForm">

<fieldset>

<legend>Change Password</legend>

<div class="changePasswordMessages"></div>

<div class="form-group">

<label for="password" class="col-sm-2 control-label">Current Password</label>

<div class="col-sm-10">

<input type="password" class="form-control" id="password" name="password"
placeholder="Current Password">

</div>

</div>

<div class="form-group">

<label for="npassword" class="col-sm-2 control-label">New password</label>

<div class="col-sm-10">

<input type="password" class="form-control" id="npassword" name="npassword"
placeholder="New Password">

</div>

</div>

<div class="form-group">

<label for="cpassword" class="col-sm-2 control-label">Confirm Password</label>

<div class="col-sm-10">

```

```

<input type="password" class="form-control" id="cpassword" name="cpassword"
placeholder="Confirm Password">

</div>

</div>

<div class="form-group">

<div class="col-sm-offset-2 col-sm-10">

<input type="hidden" name="user_id" id="user_id" value="<?php echo $result['user_id']
?>" />

<button type="submit" class="btn btn-primary"> <i class="glyphicon glyphicon-ok-
sign"></i> Save Changes </button>

</div>

</div>

</fieldset>

</form>

</div> <!-- /panel-body -->

</div> <!-- /panel -->

</div> <!-- /col-md-12 -->

</div> <!-- /row-->

<script src="custom/js/setting.js"></script>

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

```