**STOCKPILE**

**ABSTRACT**

Stockpile is a software which is widely used by retailers, shopkeepers, manufacturing units and other merchants across different businesses. It is used for managing stock of products in their warehouse or in the shops. It is a software which is helpful for the businesses operate hardware stores, where storeowner keeps the records of sales and purchase.

Stockpile will have the ability to track sales and available inventory, tells a storeowner when it's time to reorder and how much to purchase. It is a web application developed for all operating systems which focused in the area of inventory control and generates the various required reports. This system can be used to store the details of the inventory, update the inventory based on the sale details, produce receipts for sales, generate sales, and inventory reports. It uses a client-server model with a connected database to allow multiple stores and warehouses to be connected.

It is too hard to maintain the record about the daily intake of the raw material and view in the detail as a whole. The improvement in the production cannot be viewed easily when done manually. A record about the production when once created is too hard to delete.

  This is mainly designed for the manufacturing companies. It improves its business by facility of maintaining the record. Its password setting of the administrator helps to improve security. It helps the user to view the details of the stock of various categories and the sales of their requirement.

Employees can plan, enter and document warehouse and internal stock movements by managing goods receipts, goods issues, storage, picking and packing, physical stock transfers and transfer postings.

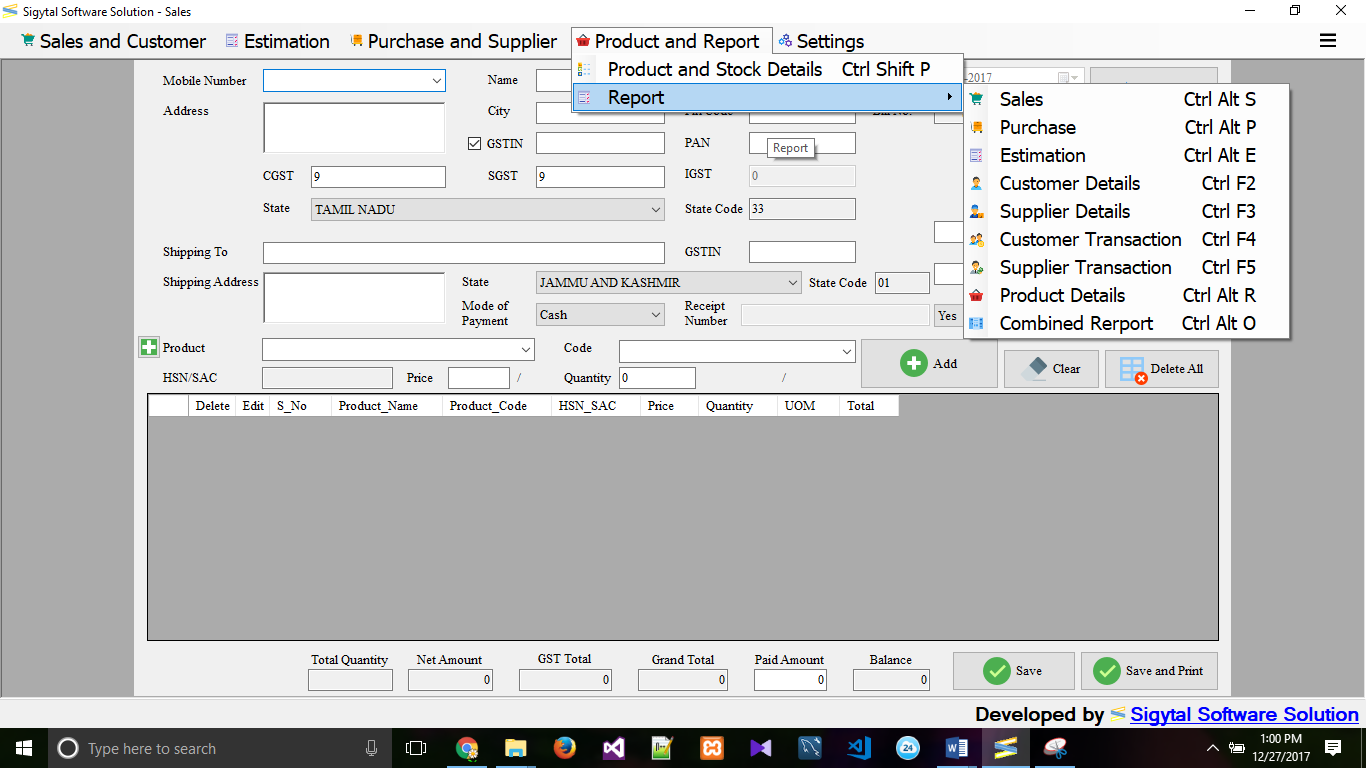
A stock management system is a software that can be integrated across your business, and with existing backend technology. It collates and maintains all inventory related data, and becomes the benchmark for comparison.

**Existing System:**

In the existing system the project was developed by desktop application. It cannot be exchange the stock between the two branches in same company.

**Drawbacks:**

* Difficult to keep old records.
* Time Consuming.
* We cannot view the other branches details.

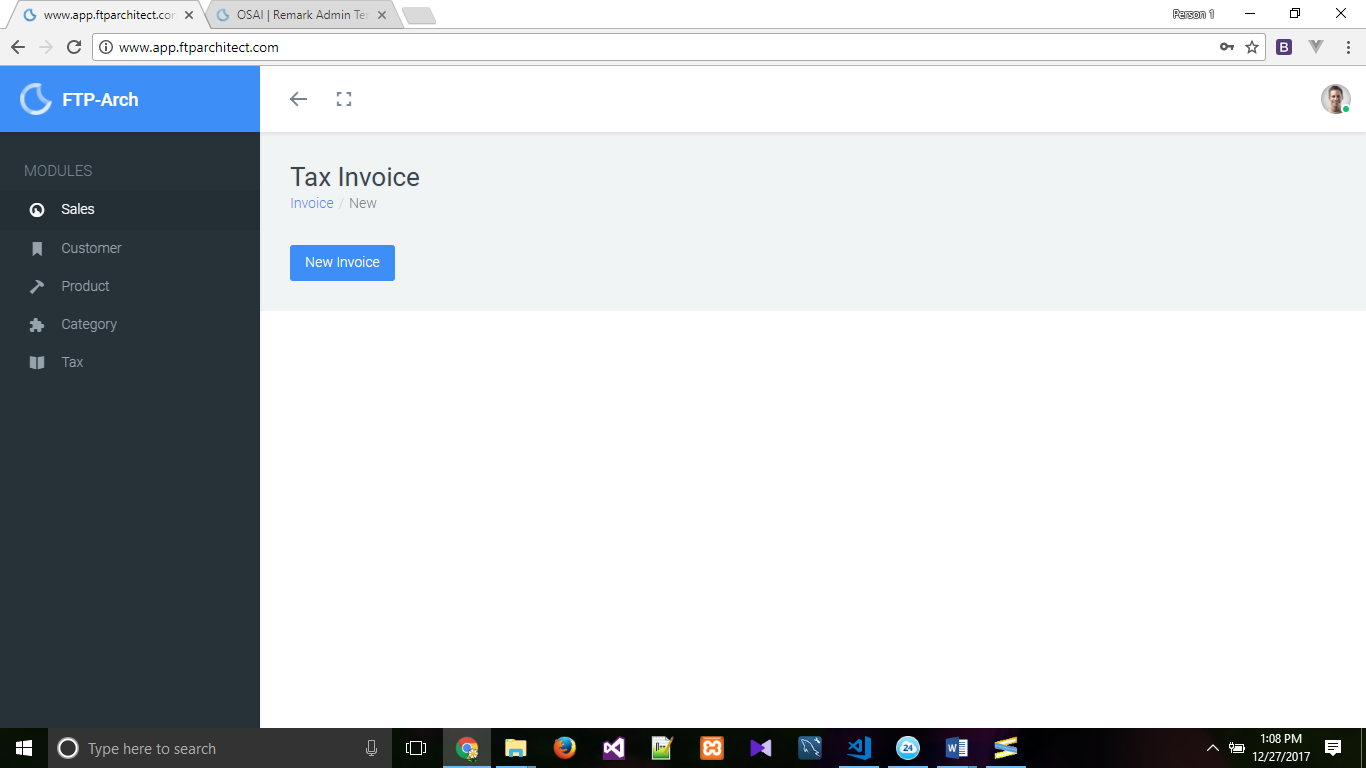


**Proposed System:**

The main aim of this project is to developed online application. This system can be used to store the details of the inventory, update the inventory based on the sale details, produce receipts for sales, generate sales and inventory reports periodically etc. This is one integrated system that contains both the user component (used by salespersons, sales managers, inventory managers) and the admin component (used by the administrators for performing admin level functions such as adding new items to the inventory, changing the price of an item).

**Advantages:**

* Users can view this application from anywhere if user has internet facility.
* We can exchange the inventory between two or more branches.
* Fully automated.
* Speed and accuracy is increased.
* More Security
* Provide more Reports based on the stock and sales.



**Modules**

* + Super Admin
  + Company Details
* Branches
* Employees
* Customers
  + Sales
  + Purchase
  + Supplier
  + Stock and Products
  + User Management

**Description of Module**

**Super Admin:**

Admin is basically the master controller of the Stockpile. He has the rights to manage all the modules of the system. He can add users, delete users, check the total sales in a particular month, check the pending orders, cancel an order and all the other functionalities present in the Stockpile system. Generally there is only one admin, but admin has the right to give any other user the admin rights. Apart from admin all other users will have limited access to the system.

**Company Details:**

It includes Company details such as branches details, employee’s details and Customer details

**Employees:**

Employees, who works under the supervision of company. It includes each employee personal details and role of the company

**Customer:**

It includes customer personal details and transaction details.

**Sales:**

**Sell Items:**

This option is available to the sales persons who are at the point of sale. Point of sale is the place where billing and other transactions are done.

**List Sales:**

This option, as the name suggests shows all the sales made by the retailer. It shows details like date, time, total bill, payment status, view bill.

**Purchase:**

This module is used to manage all purchases done by the retailer. It helps the retailer in maintaining the records of all its purchases from different suppliers.

**Add Order:**

It is used by the manager to add an order into the system. It require details like items ordered and their quantities, date of ordering, suppliers details, total cost. It will also have a feature of current status of the order which will have options like Ordered, Pending and Received.

**List Orders:**

This feature is used to display all the orders made by the merchant. It displays all the details like date of ordering, supplier details, payment status, order status.

**Supplier:**

**Add Suppliers:**

This option will help the company keep a record of suppliers by adding their details to the system. The admin will have to enter details like Name, Address, contact number and email Id.

**List Suppliers:**

It will list all the suppliers whose record is available in the database of the Stockpile System. It will also have an option to view all the dealings with that particular supplier. Similarly there can be options like Add Customers and List customers which help companies to maintain a record of their customers.

**Stock and Product:**

This module is used to manage the items being stocked in warehouse.

**Add Product:**

This sub-module is used for adding new products to the system. It will require some basic details like Product Category, Product Name, cost price, selling price, its quantity.

**List Product:**

This sub-module lists all the items present in the database of the Stockpile. It will have options to edit the details of each item or delete a particular item from the list.

**User Management:**

The users are Admin, Manager, Sales staff, Purchasing Staff. This module is only available to the Admin or Owner of the company.

**Add Users:**

This option is used by the Admin to add new Users into the system. It will ask for all the details of the user like his/her name, email id, phone number, gender. The admin will have the option to set the type of User.

The Admin have to select that whether the new user created will be an Admin, Manager, Sales staff or purchasing staff. The admin also have to set login credentials of the new user. For that a unique username and password will be set.

**List Users:**

This option will show the admin all the users using the system. This will show all the basic details of the user with its date of creation and last login time. It will also have an option to view all login details of each user. Other important feature in this module is to Activate or deactivate users account. So the admin has option to deactivate the account of any user. After which that particular user won’t be able to login into his account.

**SYSTEM REQUIREMENT**

**APPLICATION SPECIFICATION:**

**CLIENT:** Any browser that supports with ES6, BS4, CSS3, HTML5

**SERVER:** PHP 7.0, Laraval 5.5.14, VueJS v2.5.13, Webpack v3.10.0, Apache 2.2.31, MariaDB 10.1.24, CPanel 11.62.0.

**HARDWARE SPECIFICATIONS**

The components of the computer such as electrical, electronic and mechanical units are known as the hardware of computer. The input, output unit and central processing unit (CPU) are called as hardware. Thus hardware is the equipment involved in the functioning of a computer.

**REQUIREMENT**

Processor : Intel i3 2nd gen (2.20 GHz).

RAM : 2 GB or above.

Hard disk : 10GB or More.

Monitor : 15”CRT or LCD monitor.

Keyboard : Normal or multimedia.

Mouse : Compatible mouse.

**SOFTWARE SPECIFICATIONS**

It deals with defining software resource requirements and prerequisites that needed to be installed on a computer to provide optimal functioning of an application. These requirements or prerequisites are generally not included in the software installation of package and needed to be installed separately before the software is installed.

**REQUIREMENT**

Front End : Laraval, Vue JS

Back End : Maria DB.

Server Used : Web Server XAMPP, Any browser.

Operating System : X86\_64 bit, Linux

**SOFTWARE FEATURES**

**FRONT END**

**VUEJS**

Vue is a progressive framework for building user interfaces. Unlike other huge frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with [modern tooling](https://vuejs.org/v2/guide/single-file-components.html) and [supporting libraries](https://github.com/vuejs/awesome-vue#components--libraries).

**LARAVEL**

**Laravel 5.5** is the latest stable version of Laravel framework, released on August. It is an enhancement in 5.4 and comes with new features like Collection Dumping, Package Auto discovery, queued job chaining, React front-end presets, renderable, mailable and many more that make development more enjoyable and easier.

**LARAVEL DIRECTORY STRUCTURE:**

**THE ROOT DIRECTORY**

The root directory of a fresh Laravel installation contains a variety of folders:

* The app directory, as you might expect, contains the core code of your application.
* The bootstrap folder contains a few files that bootstrap the framework and configure autoloading.
* The config directory, as the name implies, contains all of your application’s configuration files.
* The database folder contains your database migration and seeds.
* The public directory contains the front controller and your assets (images, JavaScript, CSS, etc.).
* The resources directory contains your views, raw assets (LESS, SASS, CoffeeScript), and “language” files.
* The storage directory contains compiled Blade templates, file based sessions, file caches, and other files generated by the framework.

**FEATURES OF LAVAREL FRAMEWORK:**

* **Routing system:**

This framework takes an incredibly simple and easy-to-use approach to routing. Most beginner PHP developers aren’t familiar with anything other than the most natural of route systems. It brings more flexibility and control over which route is triggered on the application. A directory is created to match any desired URL.

* **Unit-Testing:**  It is an important part of Laravel framework. It runs hundreds of tests to ensure that new changes don’t unexpectedly break anything. One can then run tests with the “Artisan” command-line utility.
* **View Composers:**  These are blocks of code that can be run when a view is loaded. This keeps from having to make sure that controllers load a bunch of data from models for views that are unrelated to that method’s page content.
* **Application Logic:**  It can be implemented within any application either using controllers or directly into route declarations using syntax similar to the Sinatra framework. Laravel is designed with privileges giving a developer the flexibility that they need to create everything from very small sites to massive enterprise applications.
* **Automatic Pagination:**  It prevents application logic from being cluttered up with a bunch of pagination configuration. It gets the count of DB records and selected data using a limit/offset called ‘paginate’ and interacts Laravel where to output the paging links in view. Laravel’s pagination system was designed for programmers to make their work easier to implement and change. Laravel can handle these things automatically and makes tasks easier.

**ADVANTAGES OF LARAVEL:**

* Large community behind it, providing support and components that can be used with it.
* Frameworks like Laravel that are open source and have large communities are usually written well, and quite secure as a result. Basically, you alone will probably not write code as well as the hundreds of people that have contributed to a framework like Laravel.
* Code is kept organized, and using a framework like Laravel helps enforce some best practices.
* If you’re using a framework you don’t have to write all of that code for the standard features yourself every time.
* They’re usually well optimized too, otherwise they wouldn’t be used!
* Frameworks come with a lot of features built in that help you out, like template engines, dependency injection containers, service layers, and well-structured code.

**BACK END**

**MARIADB:**

MariaDB is a fork of [MySQL](https://en.wikibooks.org/wiki/MySQL). MySQL is the world's most popular RDBMS,In 2008 Sun Microsystems bought MySQL. After the acquisition, the development process has changed. The team has started to release new MySQL versions less frequently, so the new code is less tested. There were also less contributions from the community.

When Oracle announced the acquisition of Sun Microsystems (and thus MySQL software), most of the MySQL developers left Sun to join its forks: MariaDB and Drizzle.

The scopes of MariaDB are:

* Import all the new code that will be added to the main MySQL branch, but enhancing it to make it more stable;
* Clean the MySQL code;
* Add contributions from the community (new plugins, new features);
* Develop the Maria storage engine;
* Adding new features to the server.

Many of the improvements to the code have been written by third parties. The Monty Programs wants to keep open the development process. Its resources are not competitive with other big companies, but it benefits from the community's work. Most of the improvements are imported from Percona's patches, which are included in OurDelta MySQL builds.

**COST ESTIMATION AND SCHEDULING**

Software cost is related to many variables such as Human, Technical, Environment and Effort applied to develop it. The estimates of cost depend, on our ability to estimate and evaluate several factors, given below.

|  |  |
| --- | --- |
| **DESCRIPTION OF TASK** | **NO OF DAYS** |
| Abstract | 1 |
| Problem Statement | 1 |
| System Requirements | 1 |
| Design | 15 |
| Coding | 20 |
| Implementation | 4 |
| Testing | 5 |
| Reports | 10 |
| Deployment | 2 |
| Scope | 1 |
| **Total** | 60 |

Software cost is related to many variables such as Human, Technical, Environment and Effort applied to develop it. To estimate the effort needed for the software project, Function Point Analysis (FPA) and COCOMO model are used to predict the size and cost of developing the system. Function points are derived using an empirical relationship based on countable measures of software’s information domain and assessments of software complexity. COCOMO, **Co**nstructive **Co**st **Mo**del, is a good measure for estimating the number of person-months required to develop software. COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes (Cost Drivers).

The COCOMO cost estimation formula is

E = c\*sizek

Where, E = effort in person-months. The effort measure helps to make estimates like the number of person months that will take for the project to execute. The size estimate is converted in to effort estimates.

c = 3.0 for semidetached mode

k = 1.12 in semidetached mode

Size= (SLOC)/1000=3.0

Thus the effort for making LPG online organizer is 10.31 pm.

D = a\*Eh

Where

D = Development time in chronological months

a = 2.5 in semi-detached mode

h = 0.38 in semi-detached mode

No of days worked = 60 days

1 day work = 8 Hours

Total no of hours = 480 Hours

Cost for 1 hour = *₹* 100

Total number of cost =Total number of hours\*cost for 1 hour

Cost = 480\*100 = 48,000/-

Therefore total cost of the product is *₹* 48,000/-

s