**INTRODUCTION**

**2.1 INTRODUCTION & OBJECTIVE**

Construction management system is Web Application to manage all day to day operations related to events for an organization. There are mainly 4 modules in this website Dashboard, Status, admin, Super admin. In this website Super admin can register the members and assign roles to each member.

* 1. **EXISTING SYSTEM**

Before developing this application all activities done manually, then all the activities take more time and also take more manpower. Collecting the information about Projects, assigning work to Engineers and suggestions done manually.

The current system is taking more time to give the suggestions for the Engineers which will be a delay process and not accurate also. With the current system user is facing so many problems for entering the data regarding ConstructionWorks. It is more difficult to verify all the information and making reports for each and every project and difficult to generate the suggestions.

* 1. **PROPOSED SYSTEM**

The proposed system is to replace the existing manual system with a software solution. Using this system we can overcome the drawbacks of the current system. Using this system we can enter the master data such as Member details, Vendor details, Projects details and current status of the projects. These samples can be viewed by the super admin and also the respective members also, after seeing the project status the super admin can suggest the admin(member) to make changes.

This system involves following activities:

Functionality:

* Super Admin:

Create users.

Create branches.

View Dashboards of all Users.

View the status of the Projects.

* Admin

Create Projects.

Create Vendors.

Create Items.

Generate Purchase orders.

Generate the payment Requests.

Create the branches.

**SYSTEM ANALYSIS**

**3.1 STUDY OF THE SYSTEM**

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI’S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The ‘administrative user interface’ concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. These interfaces help the administrators with all the transactional states like Data insertion, Data deletion and Date updation along with the extensive data search capabilities.

The ‘operational or generic user interface’ helps the end users of the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information in a customized manner as per the included flexibilities

**3.2 PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Spiral Model):**

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

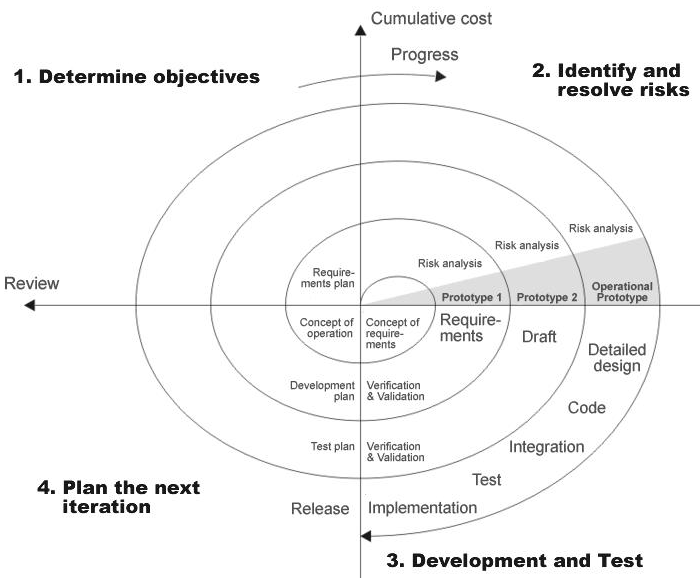


FIG1 :SOFTWARE DEVELOPMENT CYCLE.

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**3.3 SYSTEM ARCHITECTURE**

**Architecture flow:**

Below architecture diagram represents mainly flow of requests from users to database through servers. In this scenario overall system is designed in three tires separately using three layers called presentation layer, business logic layer and data link layer. This project was developed using 3-tire architecture.

**Business Logic Layer**

**Presentation Layer**

**Request**

**Response**

**Data Link**

**Layer**

**Data Base**

**REQUIREMENT SPECIFICATION**

**4.1 FUNCTIONAL REQUIREMENTS SPECIFICATION**

**FUNCTIONAL REQUIREMENTS:**

* The details of the Construction management
  + First Name
  + Last Name
  + User Name
  + Password
  + Gender
  + Email
  + Address
  + Postal Code
  + Phno

Controlled solely by the (owner) Super Admin alone.

* The details of the Construction available are to be maintained and can be

updated regularly.

Controlled solely by the (owner) Super Admin alone.

**4.2 PERFORMANCEREQUIREMENTS:**

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

**4.3 SOFTWARE REQUIREMENTS**:

Operating System : Windows xp,windows7

Technology : PHP/MYSQL

Web Technologies : Html, JavaScript, CSS,

Web Server : Apache

Database : MYSQL

**4.4 MINIMUM HARDWARE REQUIREMENTS**:

Pentium processor : P4 233 MHz

Hard disk : 20GB

RAM : 1GB

Monitor : LCD

**4.3.1. INTRODUCTION TO PHP**

PHP is stand for hypertext preprocessor. PHP is a powerful server-side scripting language for creating dynamic and interactive websites. PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. PHP is perfectly suited for Web development and can be embedded directly into the HTML code. The PHP syntax is very similar to C language. PHP is often used together with Apache (web server) on various operating systems. It also supports ISAPI (Internet Server Application Programming Interface.) and can be used with Microsoft's IIS on Windows.

In that all the concepts are from OOPS (Object Oriented Programming System), so if anyone knows about that concept then he/she can learn php easily. PHP scripts are executed on the server side. PHP supports many databases like (MySQL, Oracle, ODBC etc…). PHP files have a file extension of ".php" or ".tpl" ".phtml" Importance of Java to the Internet

**4.3.2 MYSQL**

* MySQL is the most popular Open source Database System.  MySQL Database Management System. The main goal of MySQL are speed and robustness.  MySQL, the most popular open source SQL (Structured Query Language) Database Management system, is developers, distributed, and supported by MySQL AB.
* MySQL AB is a commercial company, founded by the MySQL developers. It is a second generation open Source company that unites open source values and Methodology with a successful business model.  The MySQL website (http:www.mysql.com/) provides the latest information about MySQL software and MySQL AB.

**4.3.3 HTML**

Hypertext Markup Language(HTML), the languages of the world wide web(WWW), allows users to produces web pages that included text, graphics and pointer to other web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879,SGML(Standard Generalized Markup Language),but

Specialized to hypertext and adapted to the Web. The idea behind Hypertext one point to another point. We can navigate through the information based on out interest and preference. A markup language is simply a series of items enclosed within the elements should be displayed.

Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

Html can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags(special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics,fonts,different sizes, color, etc.. can enhance the presentation of the document. Anything

That is not a tag is part of the document it self.

**Basic Html Tags**:

<!-- --> Specific Comments.

<A>………</A> Creates Hypertext links.

<B>………</B> Creates hypertext links.

<Big>……..</Big> Formats text in large-font

<Body>…….</Body> contains all tags and text in the Html-document

<Center>……</Center> Creates Text

<DD>………..</DD> Definition of a term.

<TABLE>……</TABLE> creates table

<Td>………..</Td> indicates table data in a table.

<Tr>………..</Tr> designates a table row

<Th>……….</Th> creates a heading in a table.

##### ADVANTAGES:-

* + A HTML document is small and hence easy to send over the net.It is small because it does not include formatted information.
  + HTML is platform independent

HTML tags are not case-sensitive.

* + 1. **JAVA SCRIPT**

The Java Script LanguageJavaScript is a compact , object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. and Livewire enables you to create server-based applications similar to common gateway interface(cgi) programs.

In a client application for Navigator, JavaScript statements embedded in an HTML Page can recognize and respond to user events such as mouse clicks form

Input, and page navigation.

**SYSTEM DESIGN**

**5.1 INTRODUCTION**

Systems design

**Introduction: Systems design** is the process or art of defining the architecture,

components, modules, interfaces, and data for a system to satisfy specified requirements.

Onecould see it as the application of systems theory to product development. There is some

overlap and synergy with the disciplines of systems analysis, systems architecture and

systems engineering.

**5.2 E-R DIAGRAM:**



**DATA FLOWDIAGRAMS**

**6.1 BASIC DATA FLOW DIAGRAM**

**6.2 LEVEL 0.1**

**6.3 LEVEL 0.2**

**6.4 Level 1**

**6.5 Level 2**

**IMPLEMENTATION**

**7.1 CODING**

Home1.php

<html>

<head>

<title>CONSTRUCTION MANAGEMENT</title>

<style>

ul{

padding: -4px;

overflow: hidden;

background-color: #8A2BE2;

width: 1120px;

margin-left: 60px;

}

li {

float: left;

display: inline-table;

}

li a, .dropbtn {

display: inline-block;

color: #E1CBCB;

text-align: center;

padding: 12px 66px;

text-decoration: none;

}

li a:hover, .dropdown:hover .dropbtn {

background-color: #111;

}

.dropdown {

display: inline-block;

}

.dropdown-content {

display: none;

position: absolute;

background-color: #F9F9F9;

min-width: 189px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

}

.dropdown-content a {

color: black;

padding: 12px 16px;

text-decoration: none;

display: block;

}

.dropdown-content a:hover {background-color: #f1f1f1}

.dropdown:hover .dropdown-content {

display: block;

}

</style>

<link href="css/log.css" rel="stylesheet" type="text/css">

</head>

<body class="bgimage"><imgsrc="images/logo.jpeg" height=250 width=250 style="margin-left: 63px;"/><font class="header" style="margin-left: 63px;">CONSTRUCTION MANAGEMENT</font><br><br><br><br><br>

<ul>

<li><a class="active" href="#home">Home</a></li>

<li><a href="index.php">Login</a></li>

<li>

<div class="dropdown">

<a href="#" class="dropbtn">Create</a>

<div class="dropdown-content">

<a href="cmember.php">Create Member</a>

<a href="cbranch.php">Create Branch</a>

<a href="cproject.php">Create Project</a>

<a href="cvendor.php">Create Vendor</a>

<a href="citem.php">Create Item</a>

<a href="cpurchaseorder.php">Create Purchase Order</a>

<a href="cvprequest.php">Create Payment Request</a>

</div>

</div></li>

</body>

</html>

Connect.php

<?php

$con = mysql\_connect("localhost","root","");

$db = mysql\_select\_db("construction",$con);

?>

Create Member.php

<?php

include\_once "connect.php";

include "header.php";

if(($\_SERVER["REQUEST\_METHOD"]=="POST")||($\_SERVER["REQUEST\_METHOD"]=="post"))

{

$first\_name=$\_POST["first\_name"];

$last\_name=$\_POST["last\_name"];

$user\_name=$\_POST["user\_name"];

$email=$\_POST["email"];

$password=$\_POST["password"];

$confirm\_password=$\_POST["confirm\_password"];

$phno=$\_POST["phno"];

$gender=$\_POST["gender"];

$address=$\_POST["address"];

$postal\_code=$\_POST["postal\_code"];

if($password==$confirm\_password)

{

$qry="insert into cmember (`first\_name`, `last\_name`, `user\_name`, `email`, `password`, `confirm\_password`,

`phno`, `gender`, `address`, `postal\_code`)values('$first\_name','$last\_name','$user\_name','$email','$password','$confirm\_password',

'$phno','$gender','$address','$postal\_code')";

mysql\_query($qry);

header("Location:dmember.php");

}

else

{

echo "<html><font color='red'>entered password missmatach</font></html>";

}

<body class="bgimage">

<center>

<fieldset style="margin-top: -27px;

width: 1041px;

margin-left: 70px;"><legend><b>Create Member</b></legend>

<form style="margin-top:-165px" name="form" action="cmember.php" method="post">

<table>

<tr><td>FirstName:</td><td><input type="text" id="first\_name" name="first\_name" required/></td></tr><br>

<tr><td>LstName:</td><td><input type="text" id="last\_name" name="last\_name" required/></td></tr><br>

<tr><td>UserName:</td><td><input type="text" id="user\_name" name="user\_name" required/></td></tr><br>

<tr><td>Email:</td><td><input type="email" id="email" name="email" required/></td></tr><br>

<tr><td>Password:</td><td><input type="password" id="password" name="password" required/></td></tr><br>

<tr><td>ConfirmPassword:</td><td><input type="password" id="confirm\_password" name="confirm\_password" onChange="validate()";required/></td></tr><br>

<tr><td>Phno:</td><td><input type="text" id="phno" name="phno" pattern="[7-9]{1}[0-9]{9}" required/></td></tr><br>

<tr><td>Gender:</td><td><select name="gender" required id="gender"><br>

<option>---------</option>

<option>Male</option>

<option>Female</option>

</select></td></tr><br>

<tr><td>Address:</td><td><input type="text" id="address" name="address" required/></td></tr><br>

<tr><td>PostalCode:</td><td><input type="text" id="postal\_code" name="postal\_code" required/></td></tr><br><br>

<tr><td><input type="submit" name="clear" value="Create Member"/></td>

<td><input type="Reset" name="clear" value="clear"/><br></td>

</table></form></fieldset></center>

</body>

Display.php

<html>

<head>

<style>

body{

color:white;

}

#bod{

position : absolute;

top : 500px0px;

left : 50px;

}

#bod1{

position : absolute;

top : 20%;

left :03%;

}

.input

{

backgroundcolor : yellow;

}

.button {

background-color: orange;

border: none;

color: black;

padding: 15px 32px;

text-align: center;

text-decoration: none;

display: inline-block;

font-size: 12px;

margin: 4px 2px;

cursor: pointer;

}

</style>

</head>

<body bgcolor="black">

<div id="bod">

<h1>EMPLOYEE DETAILS</h1>

<form method="POST" action="details.php">

ENAME:<input type="text" class="input" name="ename"><font color='red'><?php echo $msg; ?></font><br>

<input type="submit" value="submit"/>

</form>

</div>

<div id="bod1">

<a href="man.html" class="button">BACK</a>

</div>

</body>

</html>

**7.2 DATABASE TABLES**

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD\_NAME | DATA\_TYPE | FIELD\_SIZE | CONSTRAINTS |
| Cbranch\_id | Integer | 25 | Primanry\_key |
| Admin | Varchar | 25 |  |
| Branch\_name | Varchar | 25 |  |
| Branch\_code | Varchar | 25 |  |
| baddress | Varchar | 25 |  |
| Bcontact\_person | Varchar | 25 |  |
| bphno | Integer | 25 |  |
| Branch\_mail | Varchar | 25 |  |
| Bservice\_tax | Integer | 25 |  |
| bnumber | Integer | 25 |  |
| Tin\_number | Integer | 25 |  |
| Pan\_number | Integer | 25 |  |

CITEM TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD\_NAME | DATA\_TYPE | FIELD\_SIZE | CONSTRAINTS |
| Item\_id | Integer | 20 | Primary\_key |
| Admin | Varchar | 20 |  |
| Item\_name | Varchar | 20 |  |
| Units | Integer | 20 |  |
| Part\_number | Integer | 20 |  |
| Sizes | Integer | 20 |  |

CMEMBER TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| FIELD\_NAME | DATA\_TYPE | FIELD\_SIZE | CONSTRAINTS |
| Member\_id | Integer | 30 | Primary\_key |
| First\_name | Varchar | 30 |  |
| Last\_name | Varchar | 30 |  |
| User\_name | Varchar | 30 |  |
| password | Varchar | 30 |  |
| Confirm\_password | Varchar | 30 |  |
| phno | Integer | 30 |  |
| gender | Varchar | 30 |  |
| address | Varchar | 30 |  |
| Postal\_code | Integer | 30 |  |

CPURCHASE\_ORDER TABLE

|  |  |  |  |
| --- | --- | --- | --- |
| Purchase\_order\_no | Integer | 25 | Primary\_key |
| admin | Varchar | 25 |  |
| Branch\_name | Varchar | 25 |  |
| Project\_name | Varchar | 25 |  |
| Vendor\_name | Varchar | 25 |  |
| Items\_to\_purchase | Integer | 25 |  |
| Amount\_per\_item | Integer | 25 |  |
| Units | Integer | 25 |  |
| Total\_amount | Integer | 25 |  |

**SYSTEMTESTING**

**8.1 TESTING TO INTRODUCTION**

**Introduction to Testing:**

Testing is a process, which reveals errors in the program. It is the major quality measure employed during software development. During software development. During testing, the program is executed with a set of test cases and the output of the program for the test cases is evaluated to determine if the program is performing as it is expected to perform.

**8.2TESTINGINSTRATEGIES**

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

**Unit Testing:**

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements.

**Each module can be tested using the following two Strategies:**

**Black Box Testing:**

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

* Incorrect or missing functions
* Interface errors
* Errors in data structure or external database access
* Performance errors
* Initialization and termination errors.

In this testing only the output is checked for correctness. The logical flow of the data is not checked.

**White Box testing:**

In this the 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 uses to generate the test cases in the following cases:

* Guarantee that all independent paths have been Executed.
* Execute all logical decisions on their true and false Sides.
* Execute all loops at their boundaries and within their operational bounds
* Execute internal data structures to ensure their validity.

#### Test Approach :

**Testing can be done in two ways:**

* Bottom up approach
* Top down approach

**Bottom up Approach:**

Testing can be performed starting from smallest and lowest level modules and proceeding one at a time. For each module in bottom up testing a short program executes the module and provides the needed data so that the module is asked to perform the way it will when embedded with in the larger system. When bottom level modules are tested attention turns to those on the next level that use the lower level ones they are tested individually and then linked with the previously examined lower level modules.

**Top down approach:**

This type of testing starts from upper level modules. Since the detailed activities usually performed in the lower level routines are not provided stubs are written. A stub is a module shell called by upper level module and that when reached properly will return a message to the calling module indicating that proper interaction occurred. No attempt is made to verify the correctness of the lower level module.

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

**SYSTEMSECURITY**

Setting Up Authentication for Web Applications

9.1 Introduction:

To configure authentication for a Web Application, use the <login-config> element of the web.xml deployment descriptor. In this element you define the security realm containing the user credentials, the method of authentication, and the location of resources for authentication.

**9.2 SECURITY IN SOFTWARE**

To set up authentication for Web Applications:

1. Open the web.xml deployment descriptor in a text editor or use the Administration Console. Specify the authentication method using the <auth-method> element. The available options are:

**BASIC**:Basic authentication uses the Web Browser to display a username/password dialog box. This username and password is authenticated against the realm.

**FORM**

Form-based authentication requires that you return an HTML form containing the username and password. The fields returned from the form elements must be: j\_username and j\_password, and the action attribute must be j\_security\_check. Here is an example of the HTML coding for using FORM authentication:

<form method="POST" action="j\_security\_check">

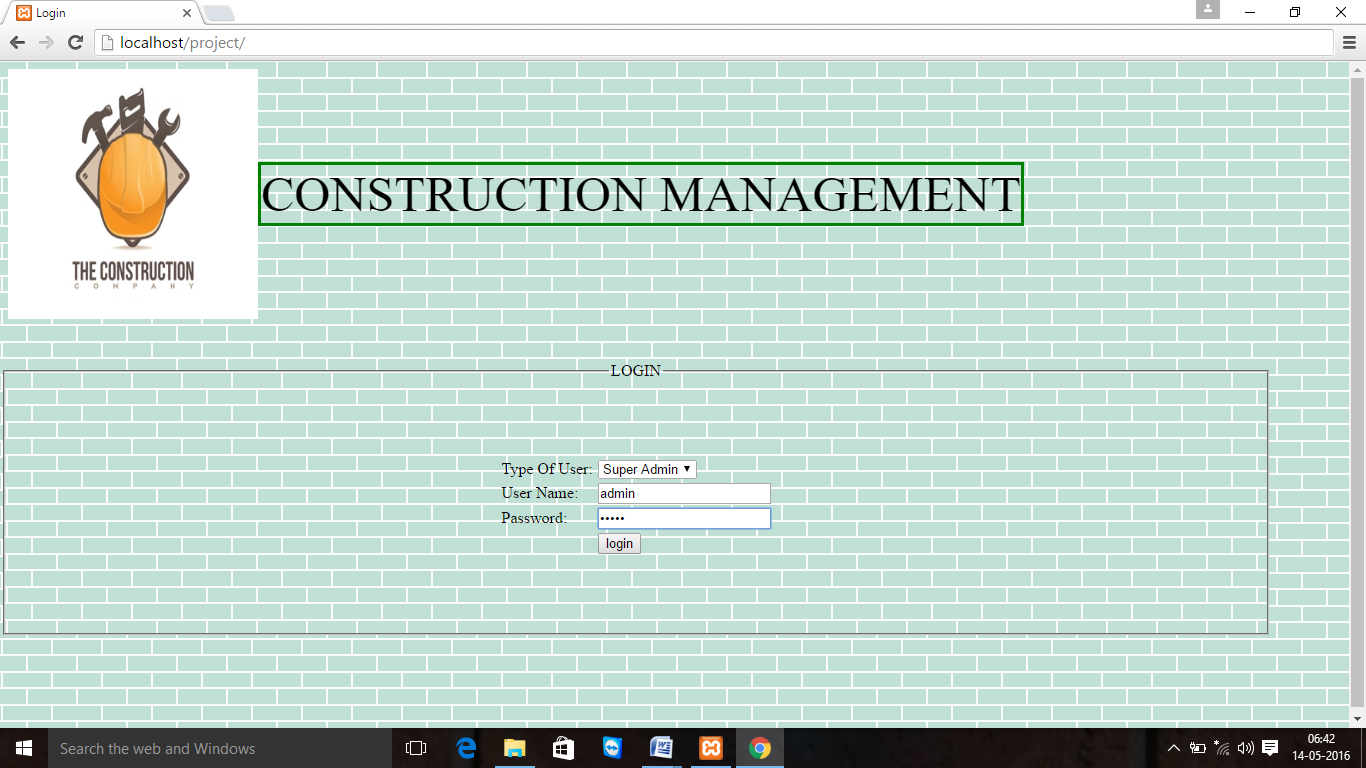
   <input type="text" name="j\_username">  
   <input type="password" name="j\_password">

</form>

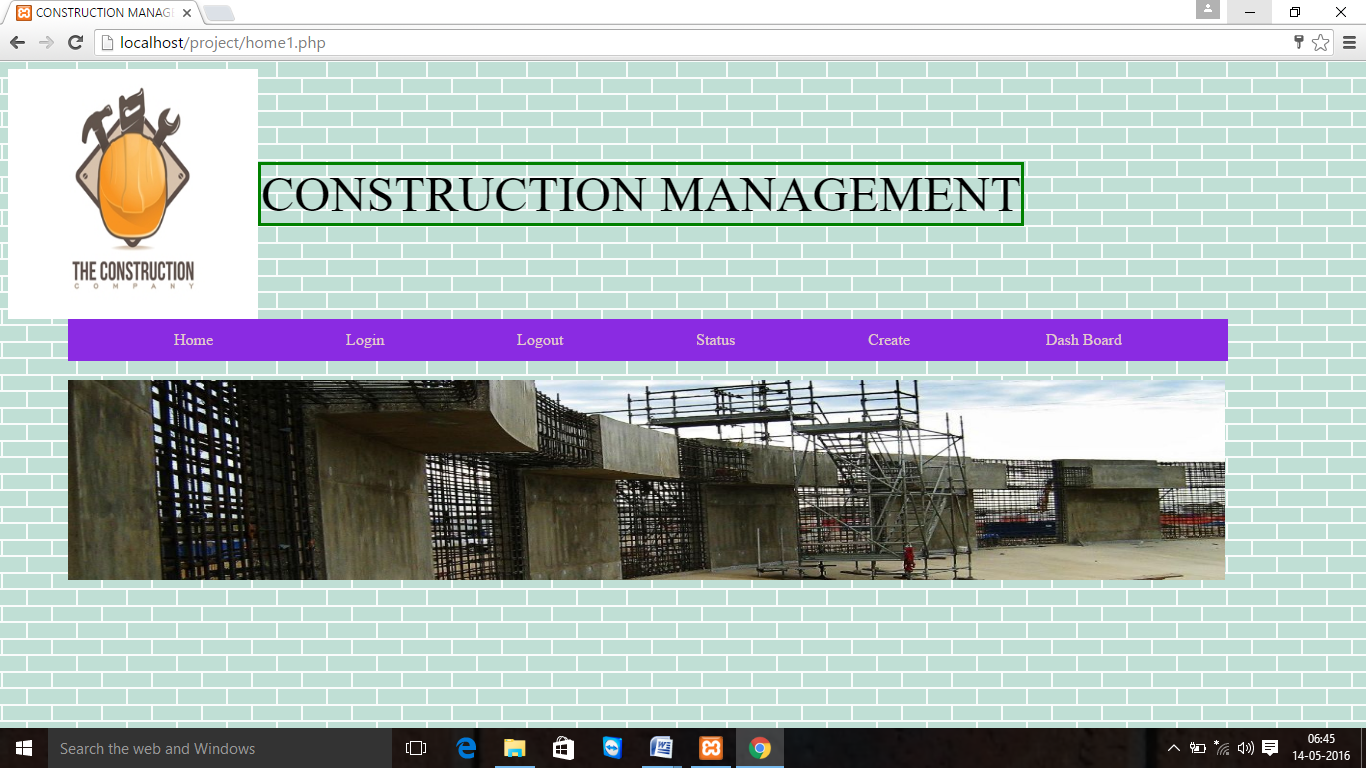
The resource used to generate the HTML form may be an HTML page, a JSP, or a servlet. You define this resource with the <form-login-page> element.

**SNAP SHOTS**

Selecting Type Of User

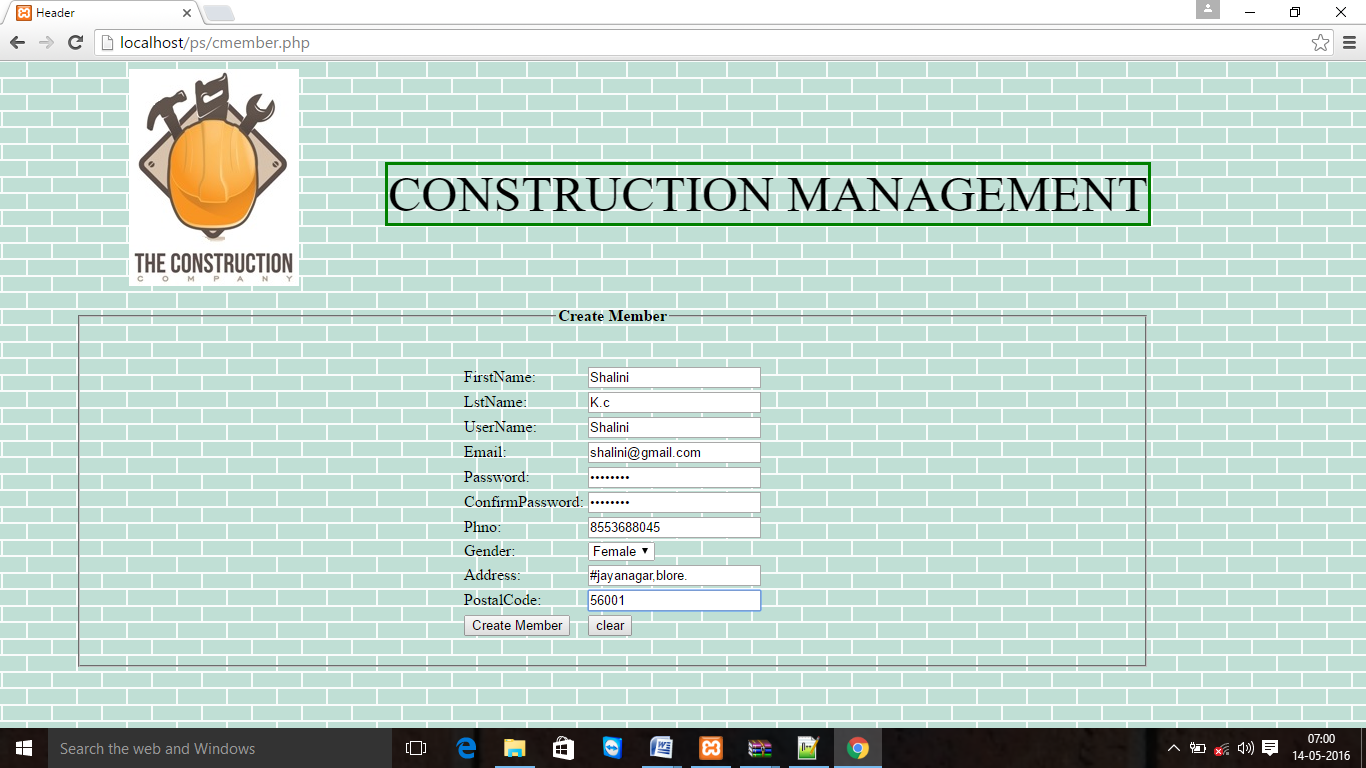


Home1.php

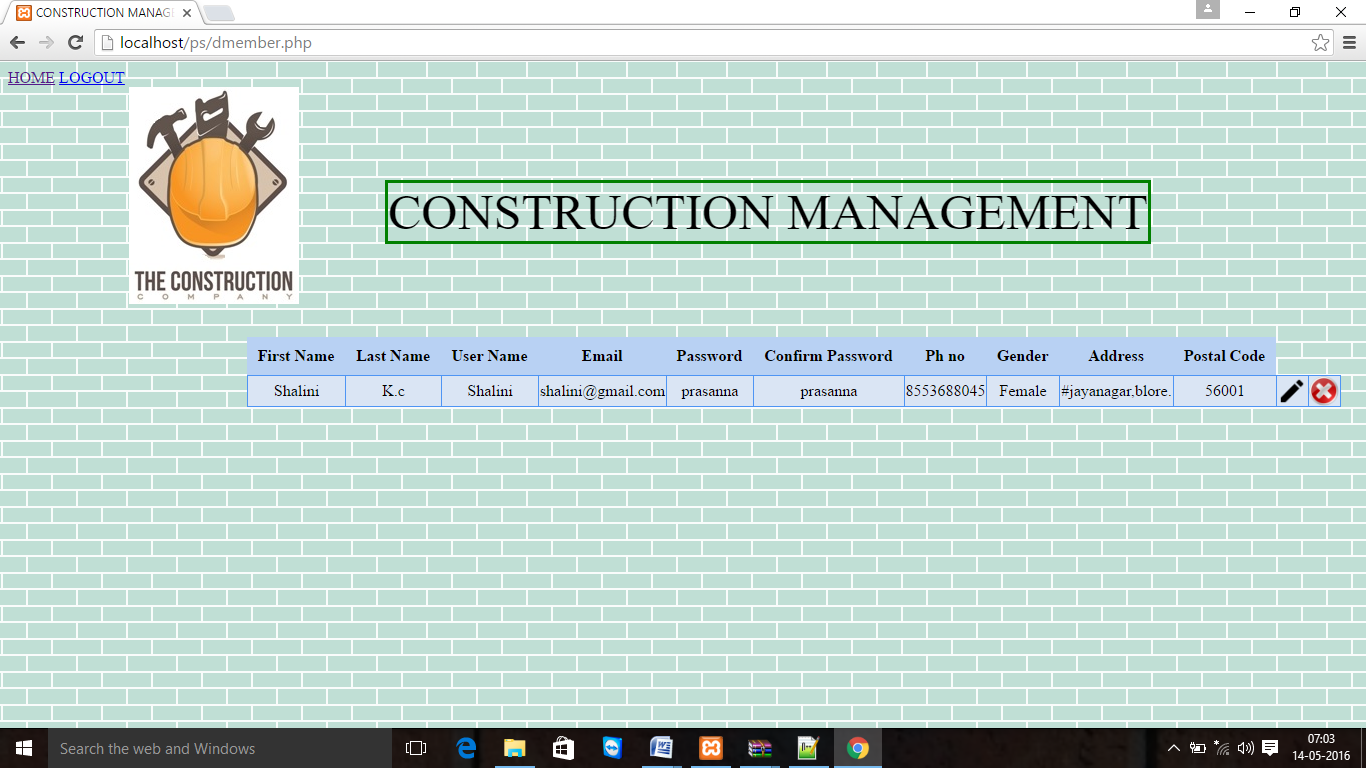


Creating Member.

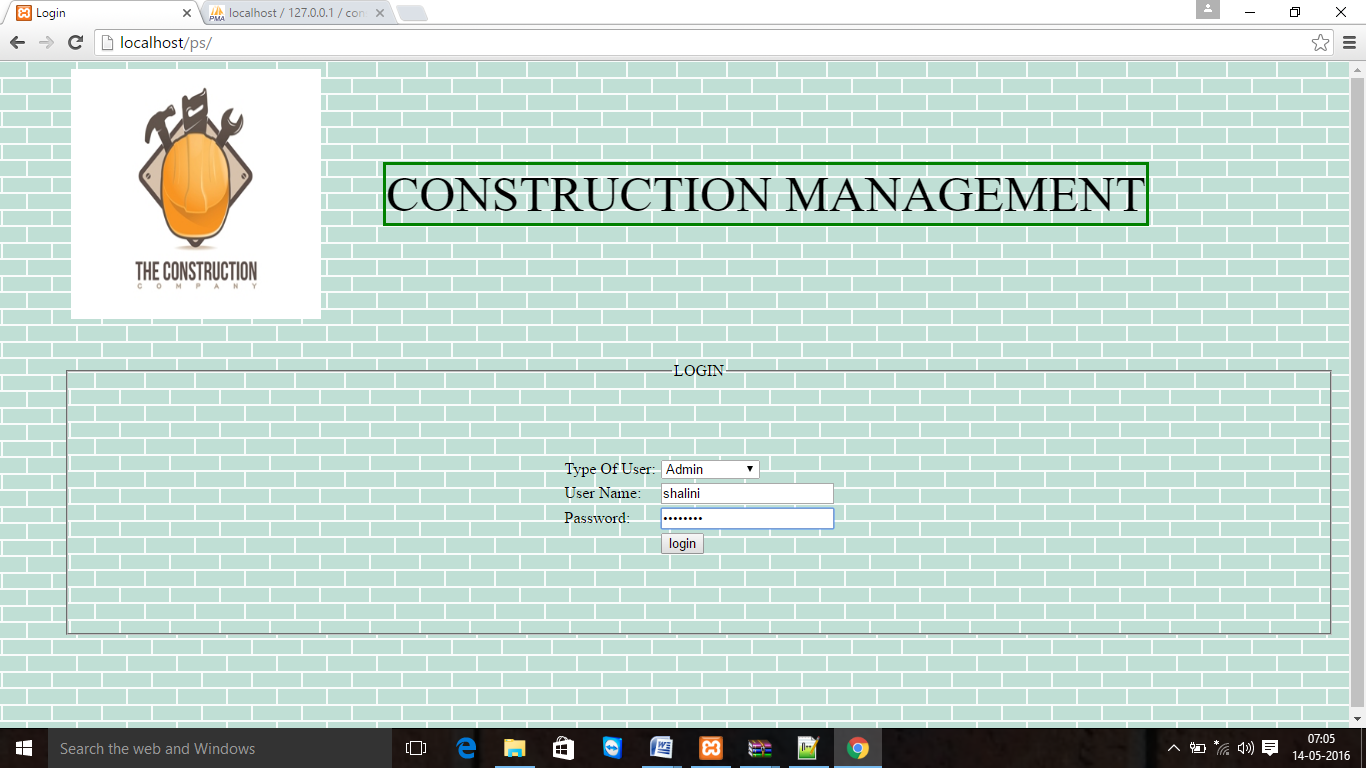




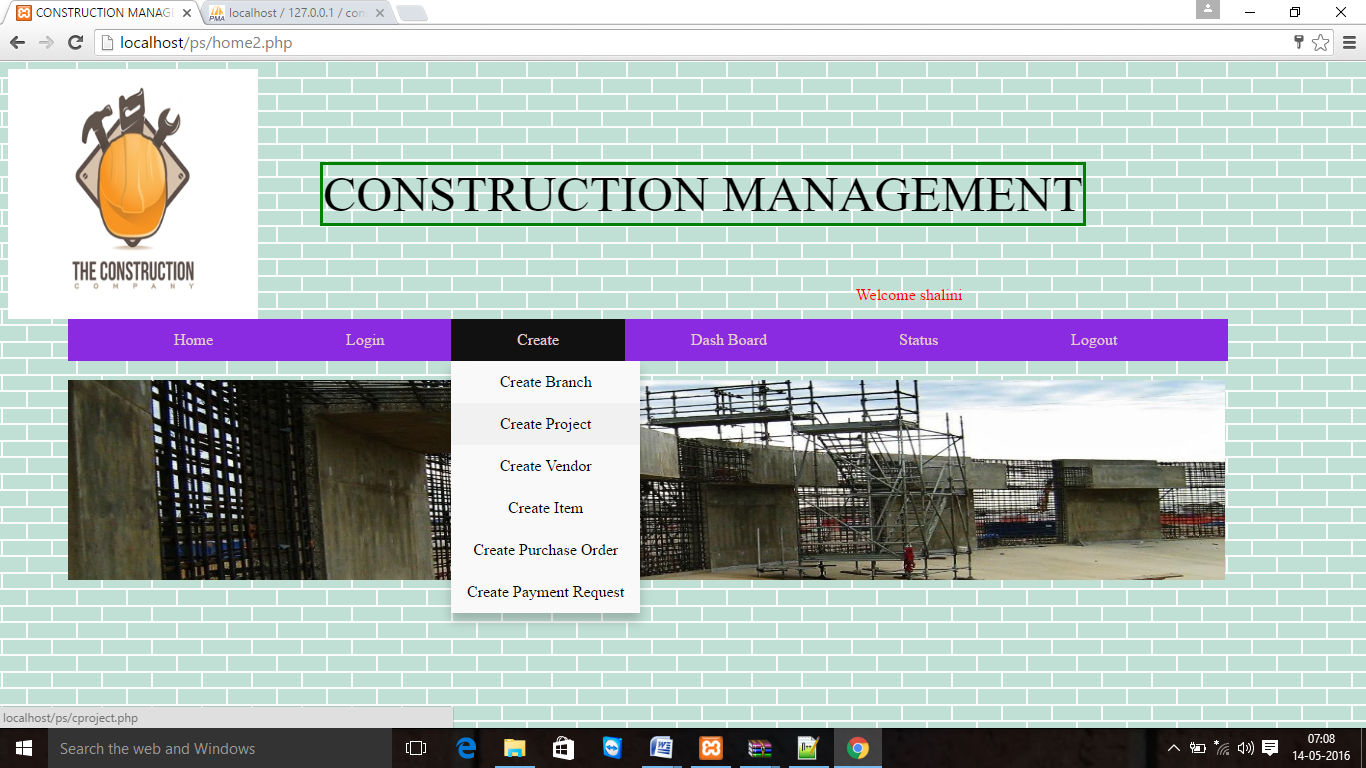
Dmember.



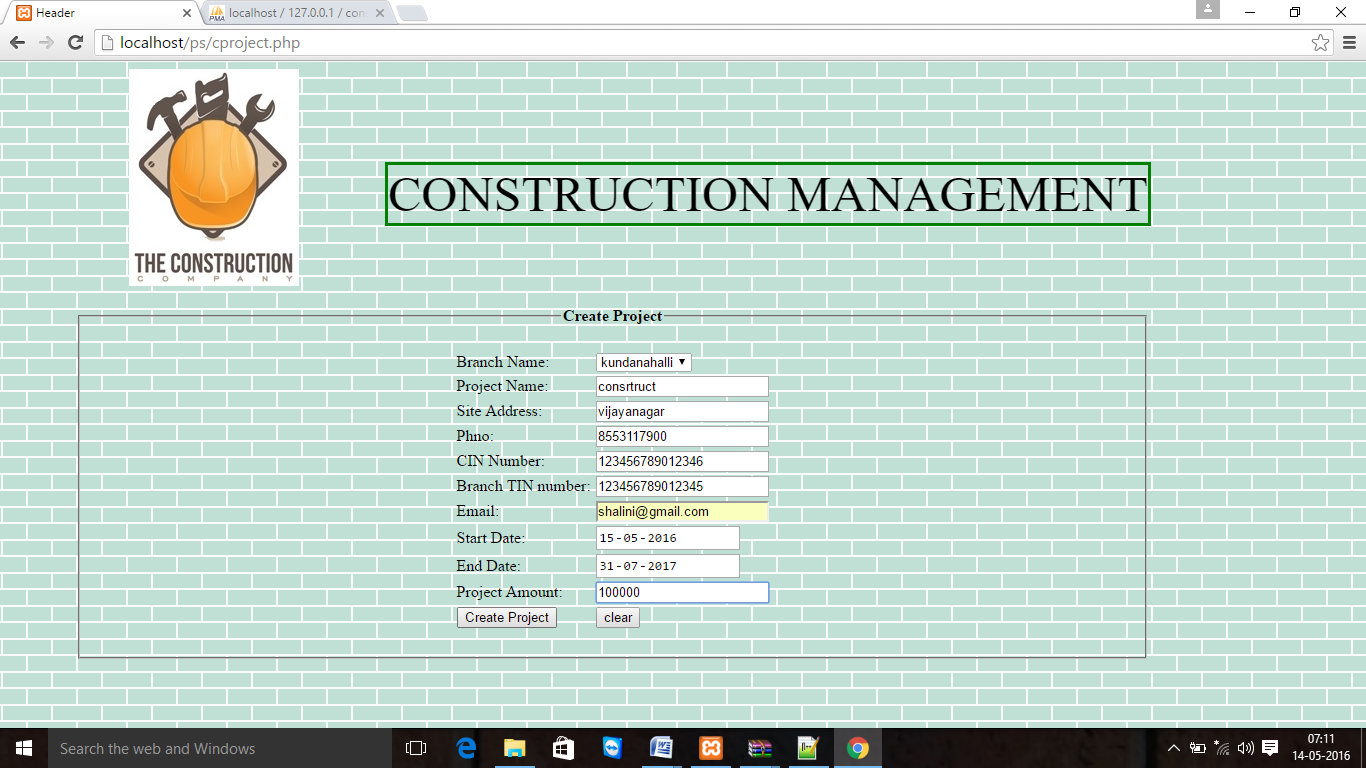
Admin Page



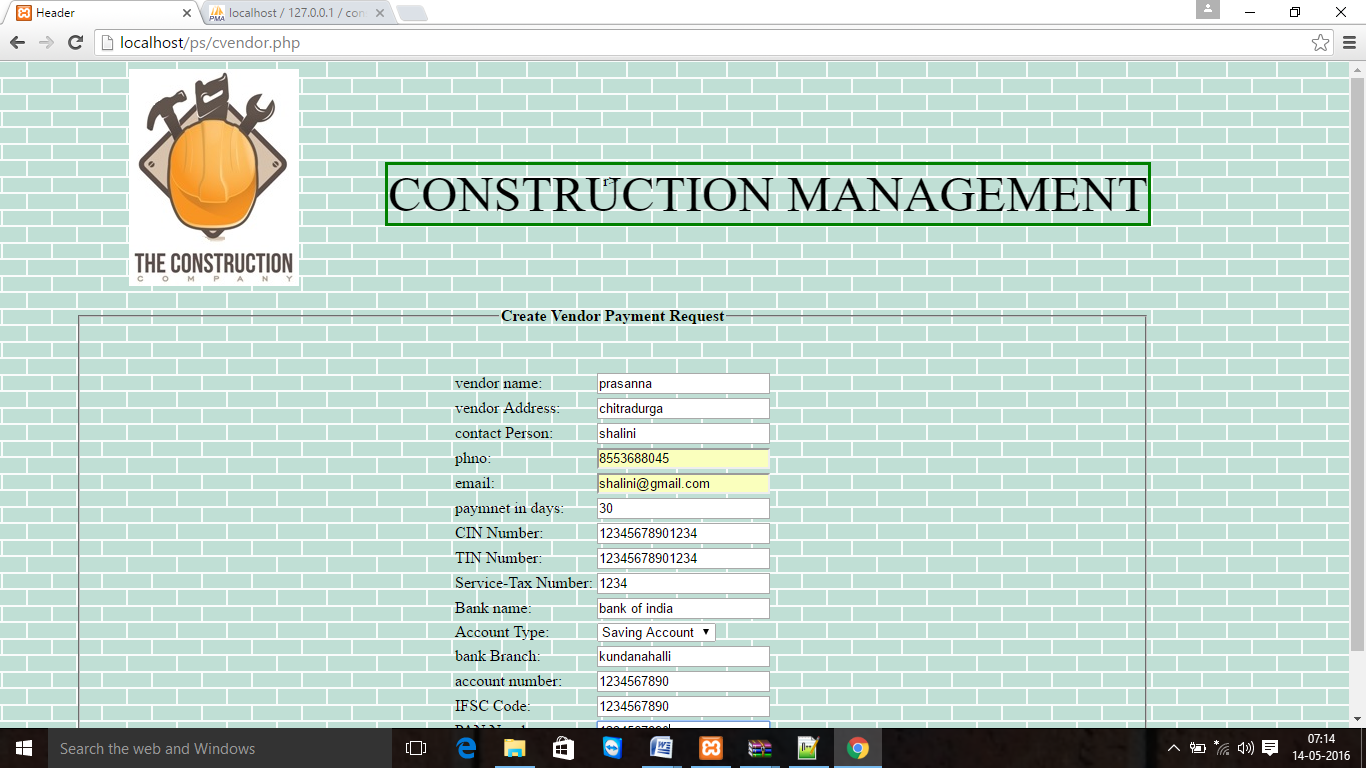
Admin Home Page



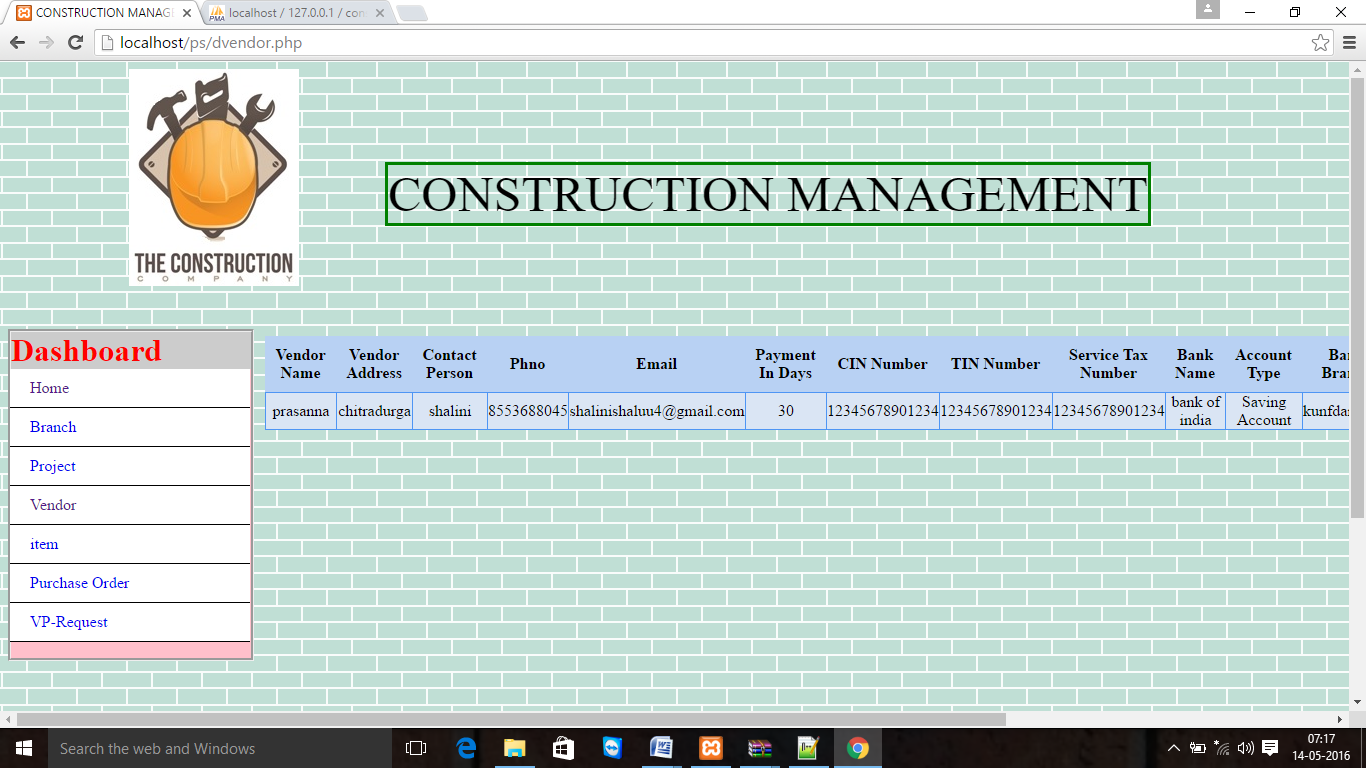
Create Project



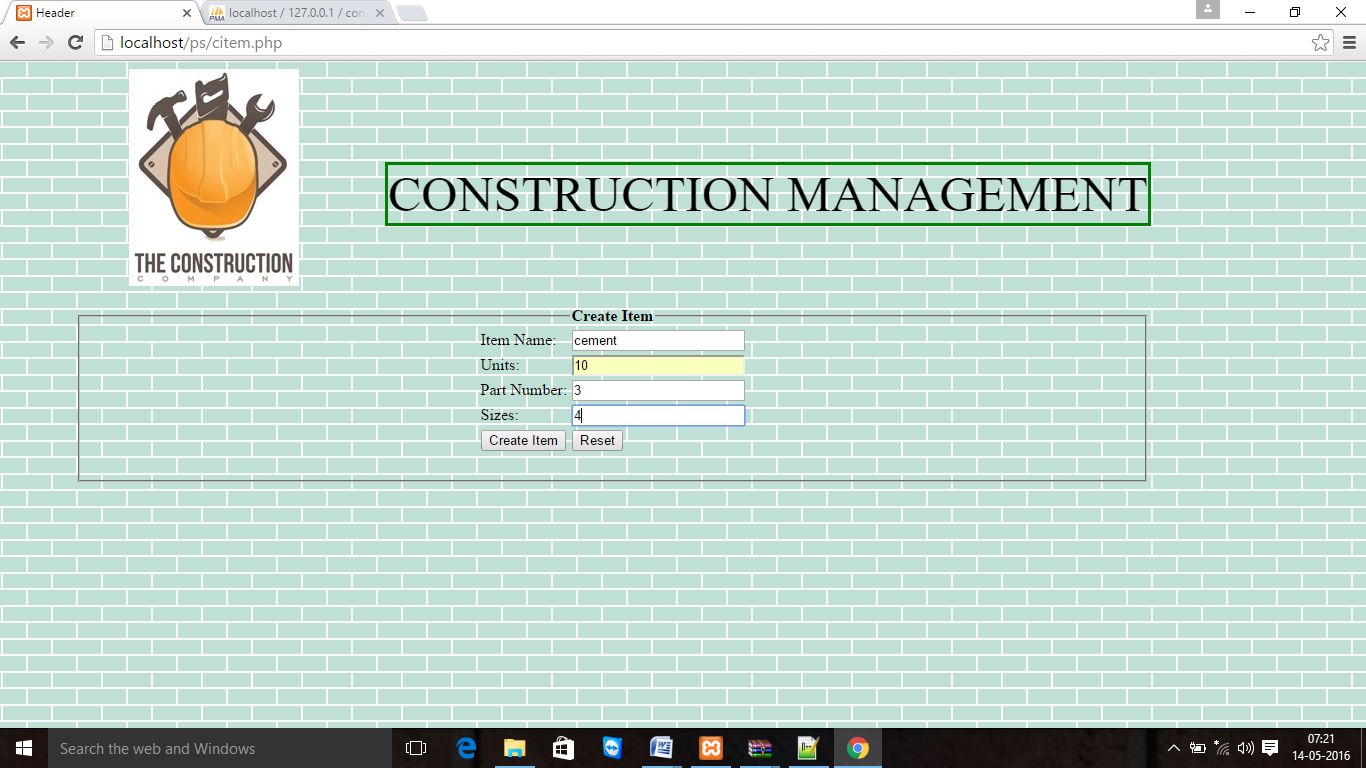
Create Vendor



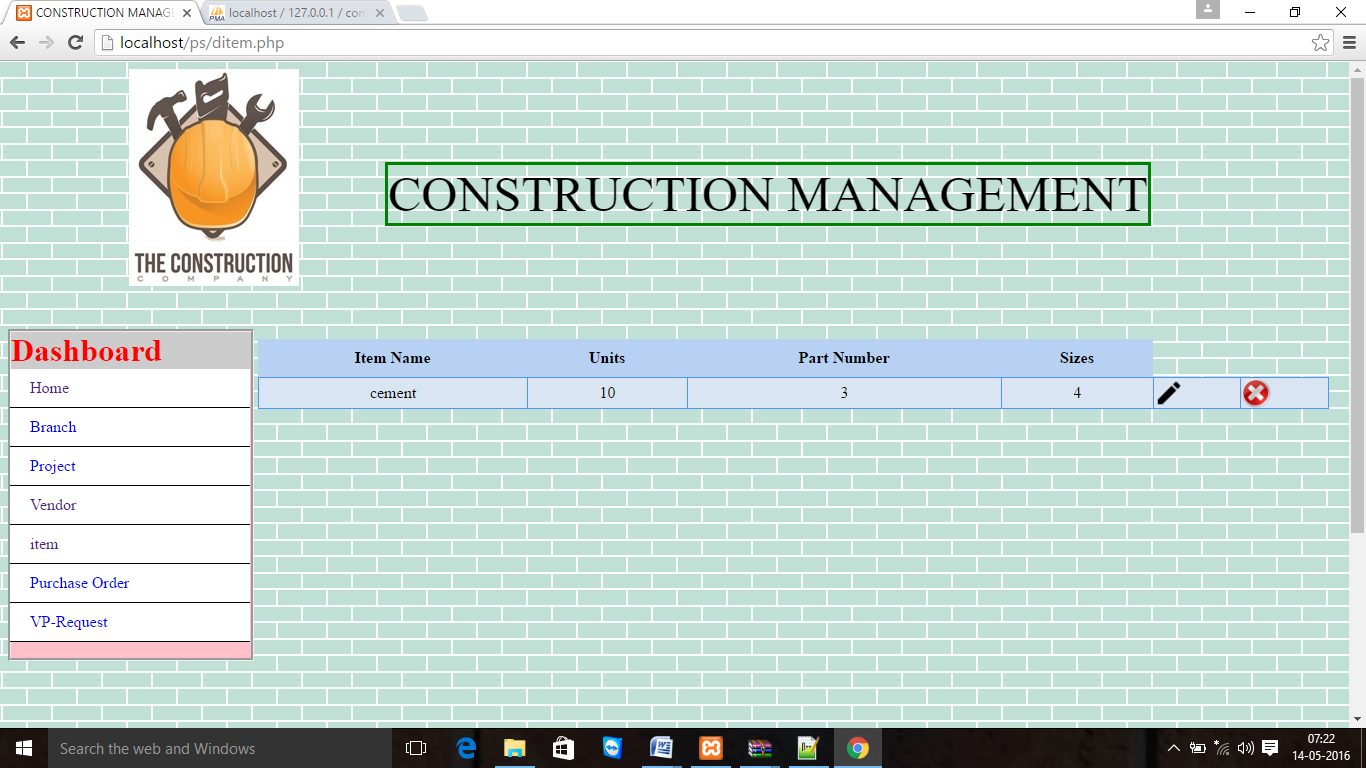
Display Vendor



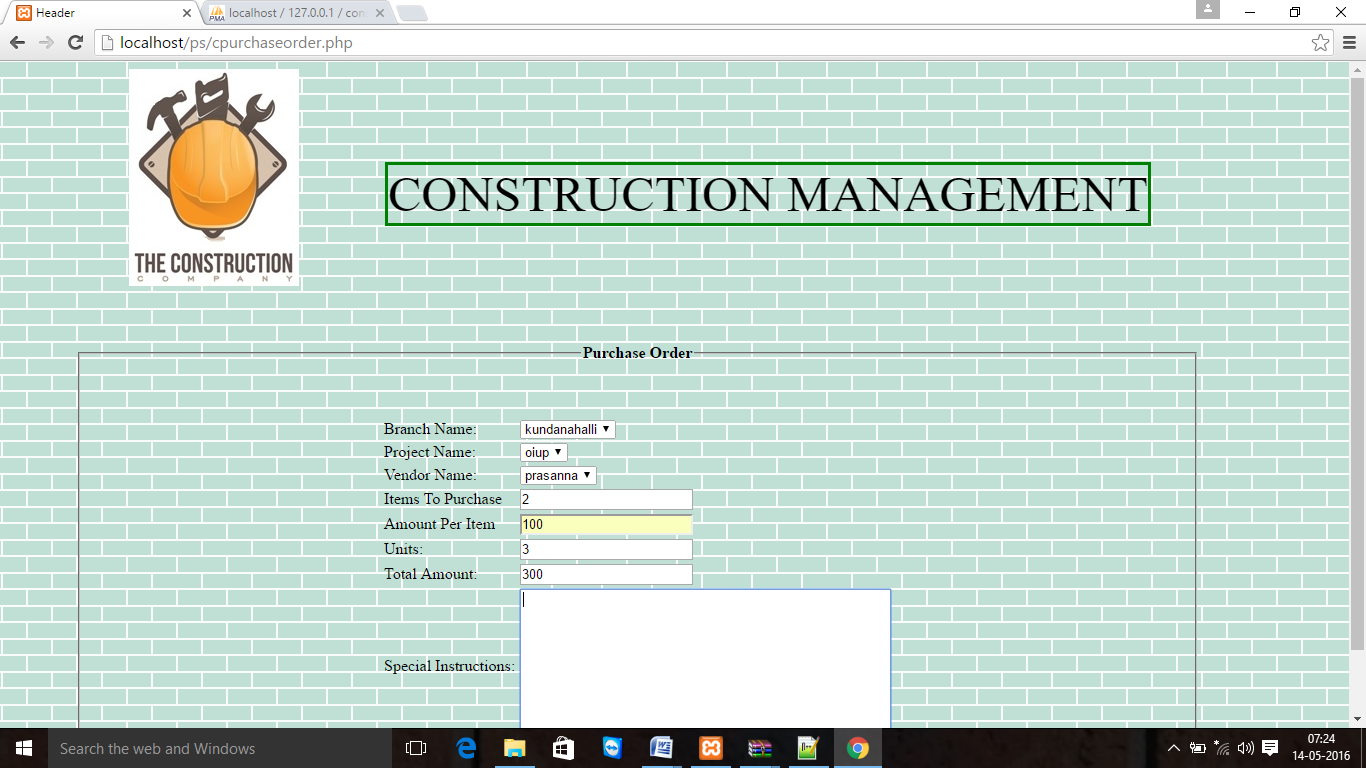
Create Item



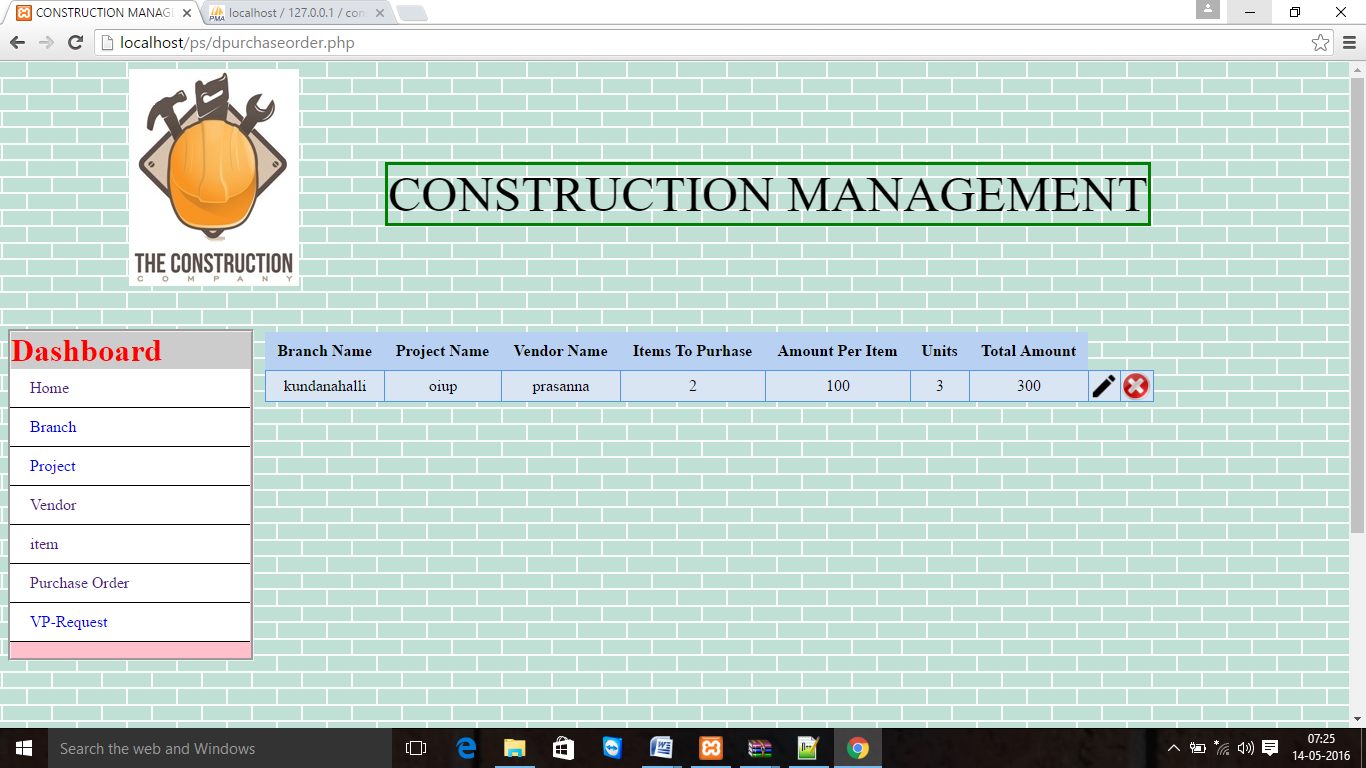
Display Item



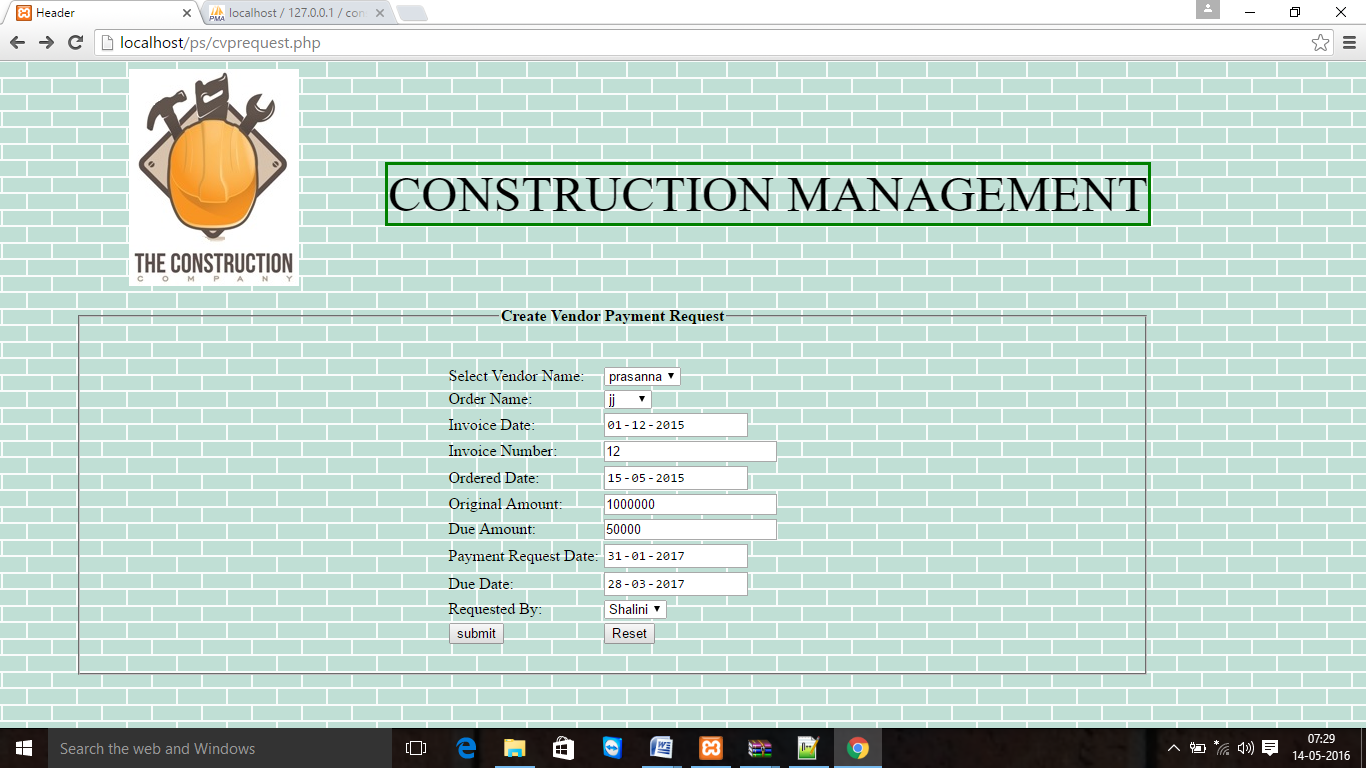
Create Purchase Order



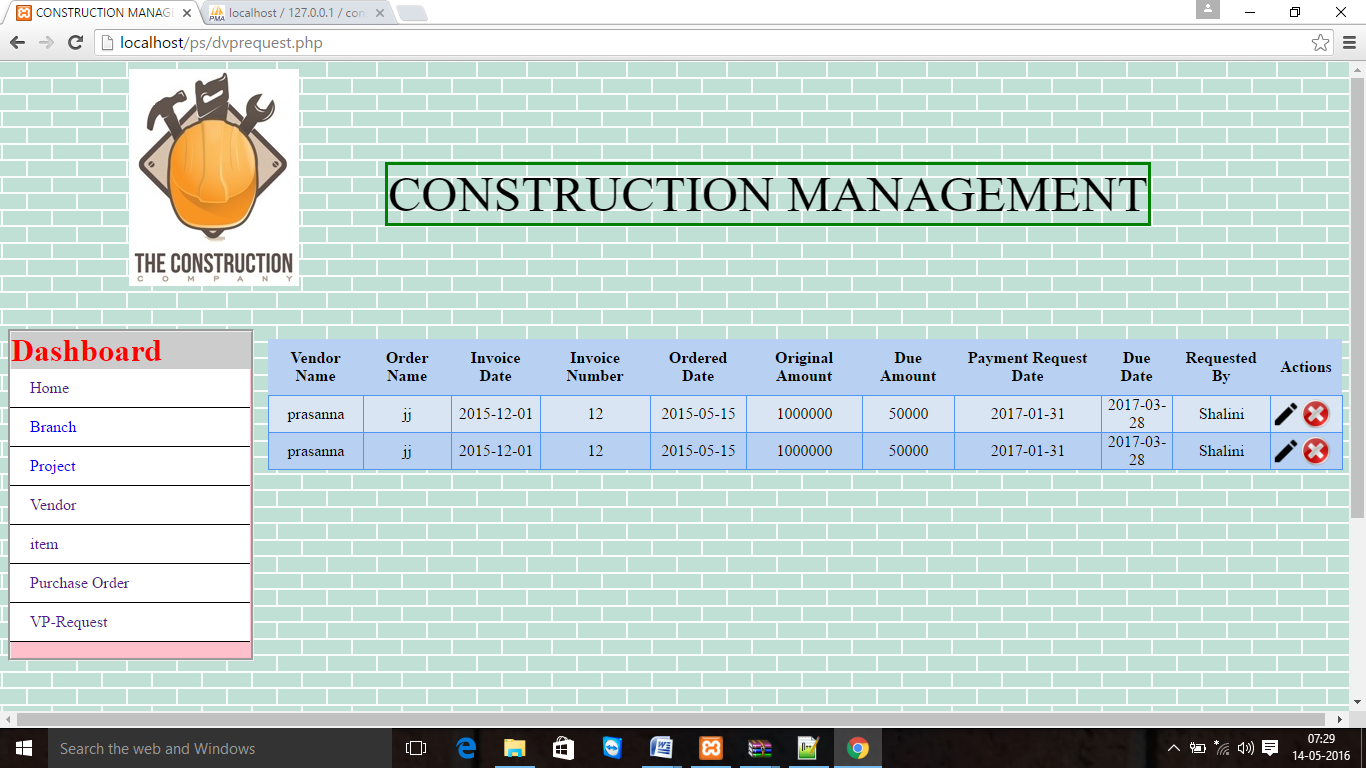
Display Purchase Order



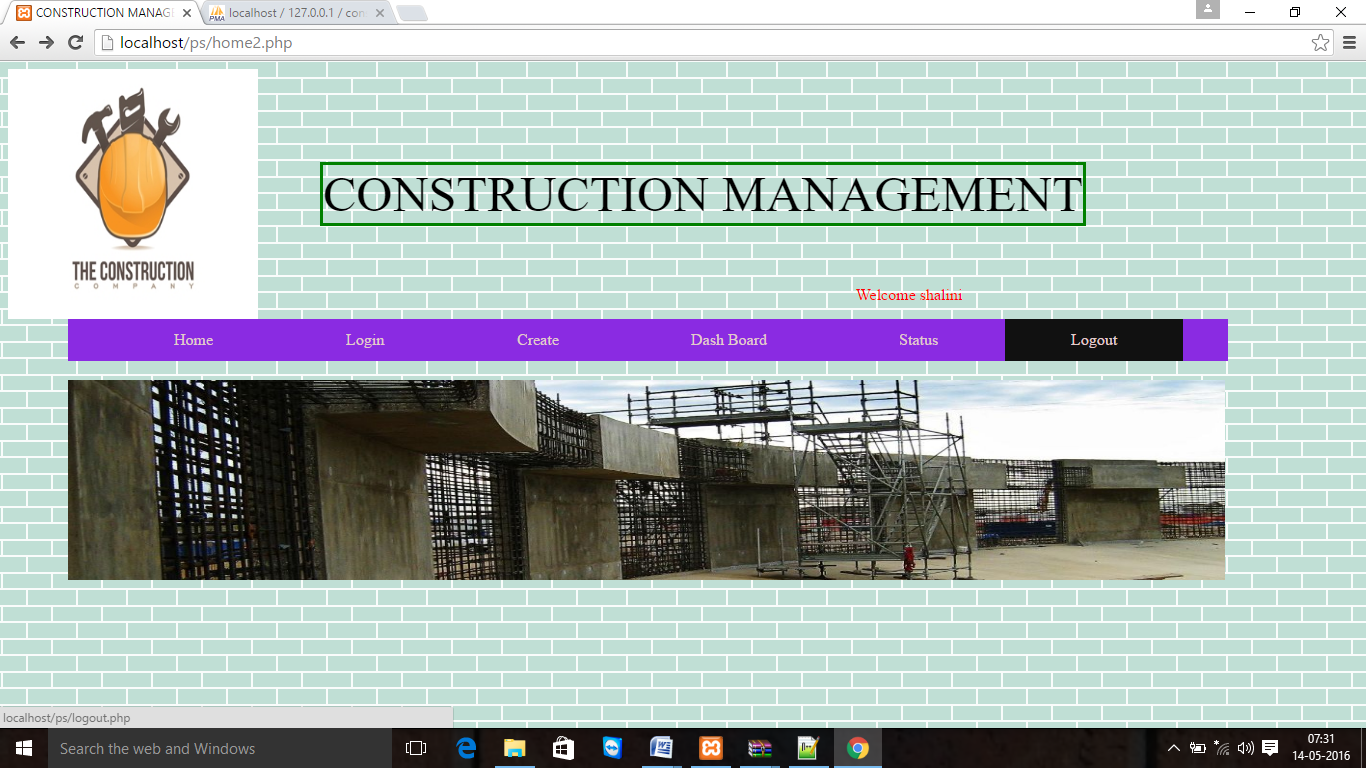
Create Vendor Request



Display Vendor Request



Logout



**CONCLUSION**

This website is basically used for keep data on related construction. This website provides features that member can create Projects, Branches, Vendors, items, etc. of construction. And also they can change the created records. This website is fast and easy retrieval and to reduce the work and time. Use of this project would make the website interactive.

Limitation

* Only admin can create a new Member.
* User cannot print whole data only admin can.
* Members can change or create the records when access is granted by admin

**11.1. FUTURE ENHANCEMENT:**

This system is developed based upon the basics requirement so may there is some limitation in development. But as per global requirement it may be enhanced to following points in next release.

* Member can give orders online.
* Member can do online bidding.
* Data security can be enhanced.
* Reminder System can be enhanced.
* User Tracking.

**12. BIBLIOGRAPHY**

Websites

* [www.google.com](http://www.google.com)
* [www.en.wikipedia.com](http://www.en.wikipedia.com)
* [www.journal.au.edu/rjcim/sep99/02-drouen.pdf](http://www.journal.au.edu/rjcim/sep99/02-drouen.pdf)
* [www.filebox.vt.edu/users/wfan/paper/hicss99.pdf](http://www.filebox.vt.edu/users/wfan/paper/hicss99.pdf)

**References**

* [**www.w3cschools.com**](http://www.w3cschools.com)
* [**www.php.org**](http://www.php.org)
* [**www.phptutorial.com**](http://www.phptutorial.com)

**Text Books**

* **Introduction to advanced web- Robert W. Sebesta**