CHAPTER: 1

INTRODUCTION

1.INTRODUCTION

1.1TITLE OF THE PROJECT:

Procurement

1.2 INTRODUCTION:

"Procurement" website provides an online interface for registered users to buy building raw materials. Registered constructors add raw materials to the website list. Admin compares the newly added raw materials with theexisting raw materials for cost effectiveness and decides to approve or reject them. If the newly added raw materials prove better than existing ones then they are added. If raw materials receive negative feedback most of the time, they are removed from the website. This ensures that the customer gets the best raw material available in market for the most economical price.

1.3 Objectives of the project

- Provide an online portal for customer to buy best and cheapest building raw materials.
- Constructors can reach to larger customer base.
- Provides easier way for estimating cost effectiveness of raw materials.

1.4 PURPOSE OF THE PROJECT:

This document completely describes what the proposed system should do without describing how the system will do it. This provides the purpose as well as the complete behavior of the system. The document started with an overall view about the proposed system and analyzed all the features and functions of the same.

1.5 PROJECT CATEGORY:

Web based RDBMS

1.6 LANGUAGE TO BE USED:

1.6.1 User Interface Design: HTML,CSS,JavaScript

HTML: Hypertext Markup Language, a standardized system for tagging

Text files to achieve font,colour,graphic and hyperlink effect on word wide web pages.

Server-Side Scripting : PHP

PHP: PHP stands for PHP: Hypertext Pre-processor. PHP is server side scripting language, like ASP. Scripts are executed on server. PHP supports many database (MySQL, Informix, Oracle, ODBC, etc). PHP is open source software(OSS). This runs on the various platforms(Windows, Mac OSX, etc).

1.6.2 Back end:MYSQL:

MYSQL is a database server. MYSQL is the one of the best RDBMS being used for developing web-based software application. It is an ideal for both small and large application. MYSQL supports standard SQL. It compiles on a number of platforms.

1.7 MODULE DESCRIPTION:

1.7.1 Admin: Admin should have complete Knowledge of the website. Admin has the following control over the website.

- **Login module**: Admin logs in using username and password.
- **Manage user module:** Admin can view and delete user details.
- **Manage constructor module**: Admin can approve user registration, delete and view constructor details.
- **View raw materials module:** Admin can View raw materials after estimating their cost effectiveness.
- **Manage quotation module:** Admin can view , Approve, pay or delete quotation details.
- **Manage order module:** Admin can confirm or cancel user order details.
- **Manage payment module:** The admin manages payment module.
    - **COD module:** module to handle cash on delivery payment method.
    - **Online payment module:** Supports payment method such as card transaction, banking methods etc.
- **View feedback module:** The admin can view or delete feedback provided by the user.

1.7.2 User: User has to register the application in order access the application. User has following functions:

- **Registration module:** The user gives primary information to register to the website.

- **Login module:** users can log in to the system using username and password.
- **View raw materials module:** The user can view and order raw materials which will be filtered from low to high cost.
- **Order module:** The user can view ordered raw materials through this module.
- **Payment module**: The user can confirm service by paying an amount. They can also pay bill online or by COD method.
- **Feedback**: user can give feedback for the raw materials ordered.
- **Update Profile:** This module allows User to update their profile

1.7.3 Constructor: The constructor will have to register the application in order access the application. Constructor has following sub functions:

- **Registration module:** The constructor gives primary information to register to the website.
- **Login module:** After admin's approval constructor can log in to the system using username and password.
- **Manage raw materials:**
    - **Add raw material**: Constructor can add, raw materials from the website based on their cost effectiveness.
    - **View raw material:** Constructor can view, edit, delete and add quotation amount for raw materials.
- **Manage quotation module:** Constructor can view, edit and delete quotation details.
- **Payment module**: Admin can view raw materials then he have to credit the transaction to the constructor account through online or by COD method.
- **Update Profile:** This module allows constructor to update their profile

1.8 SOFTWARE REQUIREMENTS

- Operating system: Windows XP or above
- Text editor: Sublime Text 3
- Web server: Apache

- Database server: MySQL
- Browsers**:** Chrome, Firefox or any other browsing application

1.9 HARDWARE REQUIREMENTS

- Processor: Intel dual core or above
- Processor speed: 2GHz
- RAM – 1 GB
- Hard Disk – Minimum 40 GB

CHAPTER: 2

SOFTWARE REQUIREMENT SPECIFICATION

2. SOFTWARE REQUIREMENT SPECIFICATIONS

2.1 Introduction:

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase. Software requirements specification establishes the basis for an agreement between customers and contractors or suppliers on how the software product should function. Software requirements specification is a rigorous assessment of requirements before the more specific system design stages, and its goal is to reduce later redesign.

2.1.1 Purpose of SRS:

"**Procurement**" website provides an online interface for registered users to buy building raw materials. Registered constructors add raw materials to the website list. Admin compares the newly added raw materials with the existing raw materials for cost effectiveness and decides to approve or reject them. If the newly added raw materials prove better than existing ones then they are added. If raw materials receive negative feedback most of the time, they are removed from the website. This ensures that the customer gets the best raw material available in market for the most economical price.

2.1.2 Scope of the Project:

The SRS describes the requirement of the system. It is meant for use by the developer and will be the basis of validating the final delivered system. Any changes made to the requirement in the future will have to go through the formal change approval process. This document contains a complete description of functioning of "**Procurement"** web application.

2.1.3 Definition, Acronyms and Abbreviations:

I/O: Input/Output
OS: Operating System
SQL : Structured Query Language
PHP : Hypertext Preprocessor
DFD: Data Flow Diagram

CFD: Context Flow Diagram

HTML : Hyper Text Markup Language

ADMIN: The Administrator

CPU: Central Processing Unit

GUI: Graphical User Interface

RAM: Random Access Memory

SRS: Software Requirement Specification

IEEE: Institute of Electrical and Electronic Engineering

2.1.4 References:

- **Books**:
  - Pankaj Jalote, An Integrated approach to software engineering, ed.3, Publication: Narosa Publishing House.
  - Nixon (Robin) Learning PHP MYADMIN, CSS\Bootstrap\Javascript. Publication: Shroff publisher and Distributors Pvt. Pankaj Jalote Ltd.

- **Websites**:
  - www.google.com
  - www.w3schools.com
  - www.tutorialspoint.com

2.2 Overall Description:

2.2.1 Product Perspective:

Product perspective is essentially the relationship of the product to another products defining if the product is independent or a part of a large product. This web application is self-contained and works relatively as efficient as other packages related to the subject. It provides simple database rather than complex one for high requirements and it provides good and easy graphical user interface to both beginners as well as experienced users of computers.

2.2.2 Product Functions:

- Admin manages the entire system.
- Admin approves constructor Registration or delete and view the details.

- Admin is also responsible for managing raw material detail based on their cost effectiveness, amount is credited to the constructor.
- User can register to the website then he can directly login to the system.
- User can view raw material based on cost effectiveness & order the raw material.
- User can pay money by ONLINE or COD method.
- Constructor can Update their profile.
- Constructor can add raw materials to the website based on cost effectiveness.
- Constructor can add quotation to the raw material.

2.2.3 User characteristics:

The system is designed with the intension to provide easy to use and simple system so that no special training is needed.

"Procurement" web application has 3 levels of users:

Admin: Manages the whole system. Admin can approves constructor registration. but the user can directly log into the system. Admin manages both accounts.

User: User can view raw materials based on quality and cost effectiveness .then orders the item. The user make payment through online or cod method. user can also give feedback for the item.

Constructor: Constructor can update their profile. Constructor add their item to the website after admins approvation. Constructor can add quotations to the rawmaterials. constructor also receive payment from the admin for their item.

2.3 Operating Environment

2.3.1 Design and implementation constraints:

- It has a relational database
- This application requires internet connection
- It gives error message to the user
- Only Administrator can control user addition and deletion

2.3.2 General Constraints:

- The user should be familiar with primary usage of web application and internet.

- The developed system can run on any platform (windows XP or above) but only with the latest web browser installed.

2.3.3 Assumption and Dependencies:

- User should be familiar with the usage of internet and websites.
- The developed web application should run on any platform (UNIX, Linux, Mac, Windows etc) that contains the latest JavaScript enabled web browser.
- The information provided by the user is assumed to be genuine.
- The web application is dependent on internet connection.

2.4 Specific Requirements:

This section describes all the details that the system developer needs to know for designing and developing this system. These requirements can be organized by the modes of operation, features and functional hierarchies.

2.4.1 External Interface Requirements:

All the interactions of the software with people,hardware and other software are specified “Procurement“ software should be simple and easy to understand as well as to use.

2.4.1.1 User Interface:

- System provides a user friendly GUI to the users.
- Appropriate error message are generated when a user performs an operation which is invalid.

2.4.1.2 Hardware Interface:

- Processor: 133-MHz Intel Pentium-class processor or higher
- RAM: 2GB and above
- Hard disk Utilization: 80GB and above
- Input Devices: Mouse, Keyboard
- Output Devices: Monitor, Printer

2.4.1.3 Software Interface:

- Browser: Internet Explorer, Google chrome, Mozilla Firefox.

- Application Server: XAMP Server.
- Other Software: Sublime Text Editor.
- Language: PHP 5.4
- Front End:HTML5, CSS3,JavaScript
- Operating system: Windows XP or above
- Database (Back End): My SQL

2.4.1.4 Communication Interface:

The system works with the help of internet connection. The application is linked to the database maintained in the server.

2.4.2 Functional Requirements:

The module can be accessed by the admin only, user or constructor cannot access this module. Sub modules are:

2.4.2.1 Admin: Admin should have complete Knowledge of the website. Admin has the following control over the website.

- **Login module**: Admin logs in using username and password.
- **Manage user module:** Admin can approve user registration, delete, edit and view user details.
- **Manage constructor module**: Admin can approve user registration, delete, edit, and view constructor details.
- **Manage quotation module:** Admin can manage quotation details.
- **Approve raw materials module:** Admin can manage raw materials after estimating their cost effectiveness.
- **Manage payment module:** The admin manages payment module.
- **COD module:** module to handle cash on delivery payment method.
- **Online payment module:** Supports payment method such as card transaction, banking methods etc.
- **View feedback module:** The admin can view feedback provided by the user.

2.4.2.2 User: User will have an account. Different planners being registered and their services.

- **Registration module:** The user gives primary information to register to the website.
- **Login module:** After admin's approval user can log in to the system using username and password.
- **View raw materials module:** The user can view raw materials which will be filtered from low to high cost.
- **Order module:** The user can order for raw materials through this module.
- **Payment module**: The user can confirm service by paying optional advance amount. They can also pay bill online or by COD method.
- **Feedback**: user can give feedback for the raw materials ordered.

2.4.2.3 Constructor:

- **Registration module:** The constructor gives primary information to register to the website.
- **Login module:** After admin's approval constructor can log in to the system using username and password.
- **Update Profile:** This module allows constructor to update their profile
- **Manage raw materials:** Constructor can add or delete raw materials from the website based on their cost effectiveness.
- **Payment module**: Admin have to credit the transaction to the constructor account through online or by COD method.

2.4.3 Performance Requirements:

The system will run efficiently on any modern JavaScript enabled web browser. The server will require 4GB RAM or above, Intel Pentium-class processor or higher, 80GB or above hard disk utilization.

2.4.4 Design Constraint:

The system must be designed to allow web usability. That is, the system must be designed in such a way that will be easy to use and visible on most of the browsers.

2.4.5 Other Requirements:

2.4.5.1 Safety Requirements:

This system is safe to use. Data is stored in the central database and it is made sure that there is minimum data loss in case of any errors. In case the user forgets the password, the user can set a new password .Authentication helps to check the permission level of the user accessing the site.

2.4.5.2 Security Requirements:

The proposed website is a secured website. There are different categories of users. they are Admin, Constructor and User. Depending upon the category of constructor the access rights are decided by the admin, it means if constructor can add rawmaterials. then he can be able to modify,delete and view data.

2.4.5.3 Software Quality Attributes:

- Efficiency: The system is built using pre-defined guided process and the data from all system are stored in one central database.
- Extensibility: The system supports future modifications without any side effects.
- Portability: The system can run in any operating system with latest version of the browser installed.
- Flexibility: The system keeps on updating the data according to the changes that take place.
- Reliability: As the system is centered on the database, the reliability depends on the maintenance of the database.
- Usability: The interface is user friendly; it is easy to access and understand.

CHAPTER: 3

SYSTEM DESIGN

3. SYSTEM DESIGN

3.1 Introduction:

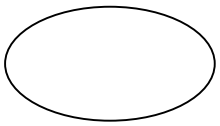
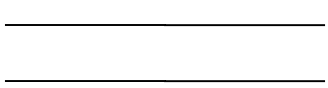
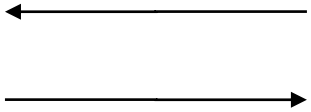
Software design is a process through which requirements are translated in to a representation of the software. In detail design we specify how the modules in the system interact with each other. Detailed specification is given by explaining in natural language for the modules is supposed to do.

Detail design specification describes the features of the system. Detail design is the refinement of the system design that essentially expands system design to contain more detail description of the processing logic of the components and data structure such that design can be easily implemented.

3.2 Context Flow Diagram (CFD):

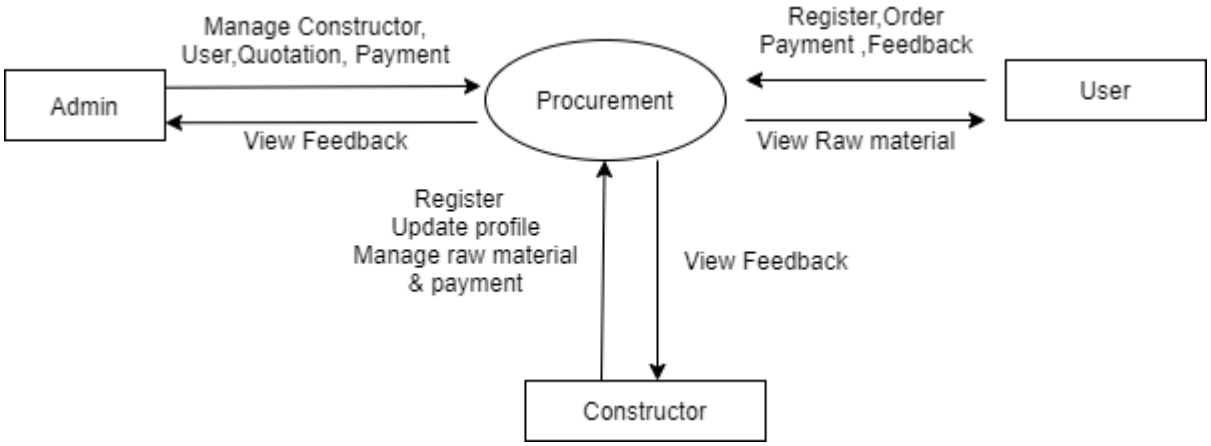
Context flow diagram is a top level data flow diagram. It only contains one process node that generalizes the function of the entire system in relationship to external entities. In context diagram the entire system is treated as a single process and all its inputs, outputs, sinks and sources are identified and shown.

3.2.1 Notation used in Data Flow Diagram:

Name	Notation	Description
Process		A process transforms incoming data flow into outgoing data flow. The processes are shown by named circles.
Datastore		Data stores are repositories of data in the system. They are sometimes also referred to as files.
Dataflows		Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.

External Entity		External entities are objects outside the system with which the system communicates. External Entities are sources and destinations of the system's inputs and outputs

Context Flow Diagram (CFD):



3.3 Data Flow Diagram (DFD):

Data Flow Diagram is a graphical representation of a system or a portion of the system. It consists of data flows, process, sources and sink and stores all the description through the use of easily understandable symbols.

DFD is one of the most important modelling tools. It is used to model the system, components that interact with the system, uses the data and information flows in the system.

DFD shows the information moves through the and how it is modified by a series of transformations. It is a graphical technique that depicts information moves from input or output.

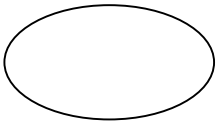
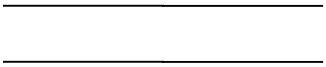
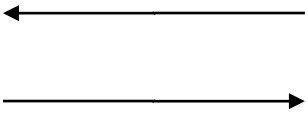
DFD is also knows as bubble chart or Data Flow Graphs. DFD may be used to represent the system at any level of abstraction. DFD's may partition into a level that represents increasing information flows and functional details.

Rules Regarding DFD Construction:

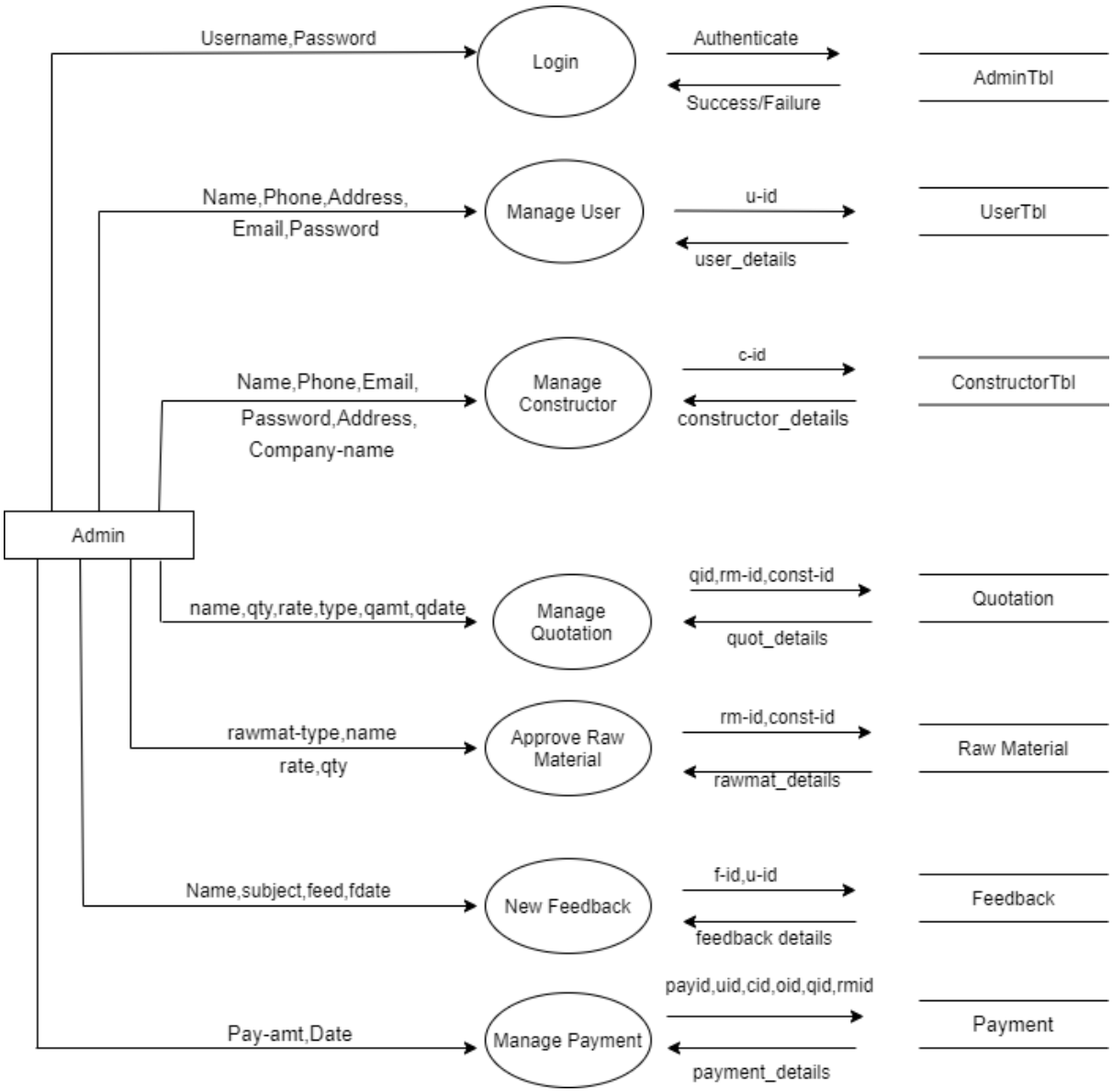
- A process cannot have only outputs.

- A process cannot have only inputs.
- The inputs to a process must be sufficient to produce the outputs from the process.
- All data stores must be connected to at least one process.
- All data stores must be connected to a source or sink.
- A data flow can have only one direction of flow. Multiple data flows to and/or from the same process and data store must be shown by separate arrows.
- If the exact same data flows to two separate arrows, it should be represented by a forked arrow.
- Data cannot flow directly back into the process it has just left. All data flows must be named using a noun phrase.

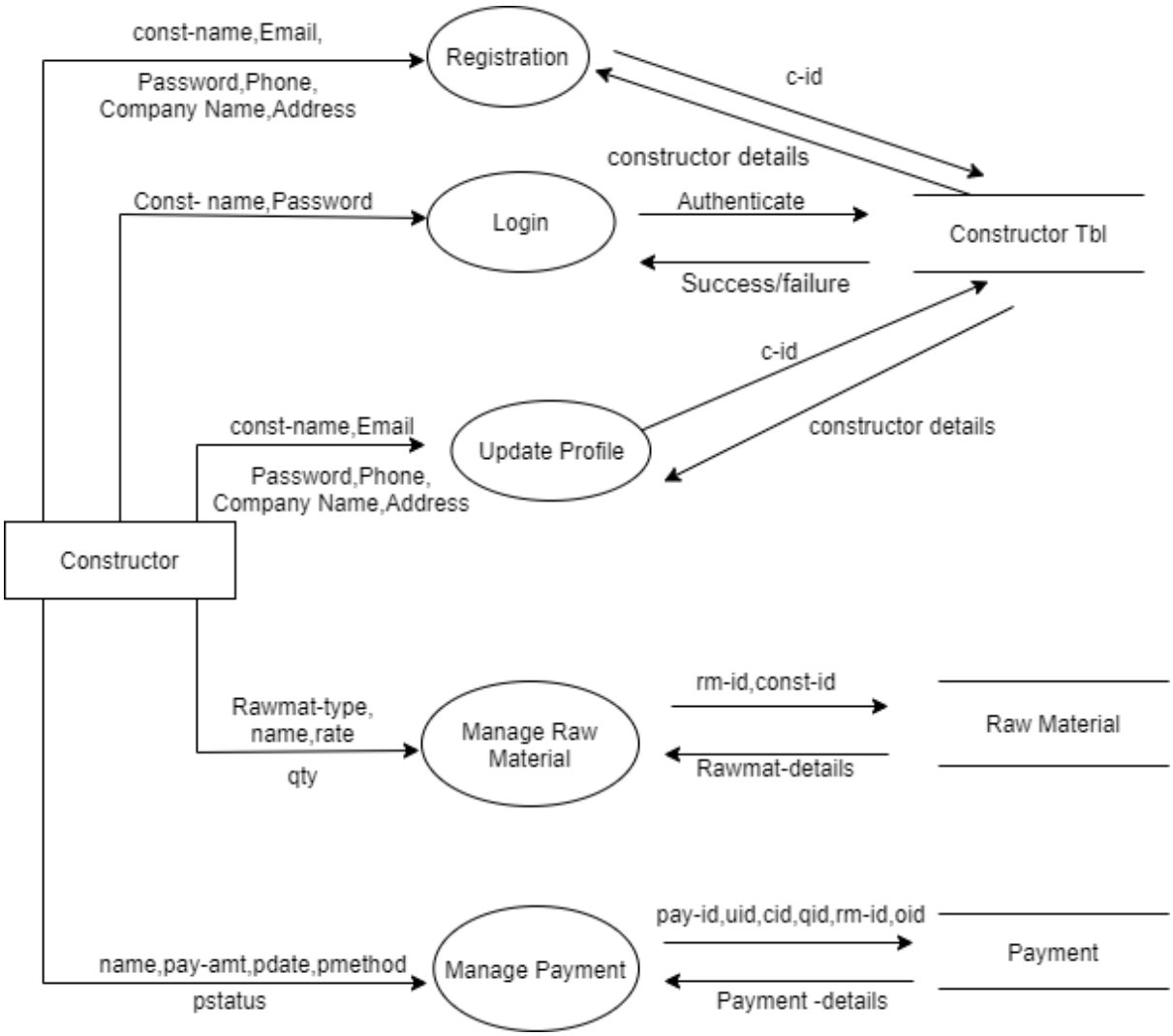
3.3.1Notation used in Data Flow Diagram:

Name	Notation	Description
Process		A process transforms incoming data flow into outgoing data flow. The processes are shown by named circles.
Datastore		Data stores are repositories of data in the system. They are sometimes also referred to as files.
Dataflows		Data flows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.
External Entity		External entities are objects outside the system with which the system communicates. External Entities are sources and destinations of the system's inputs and outputs

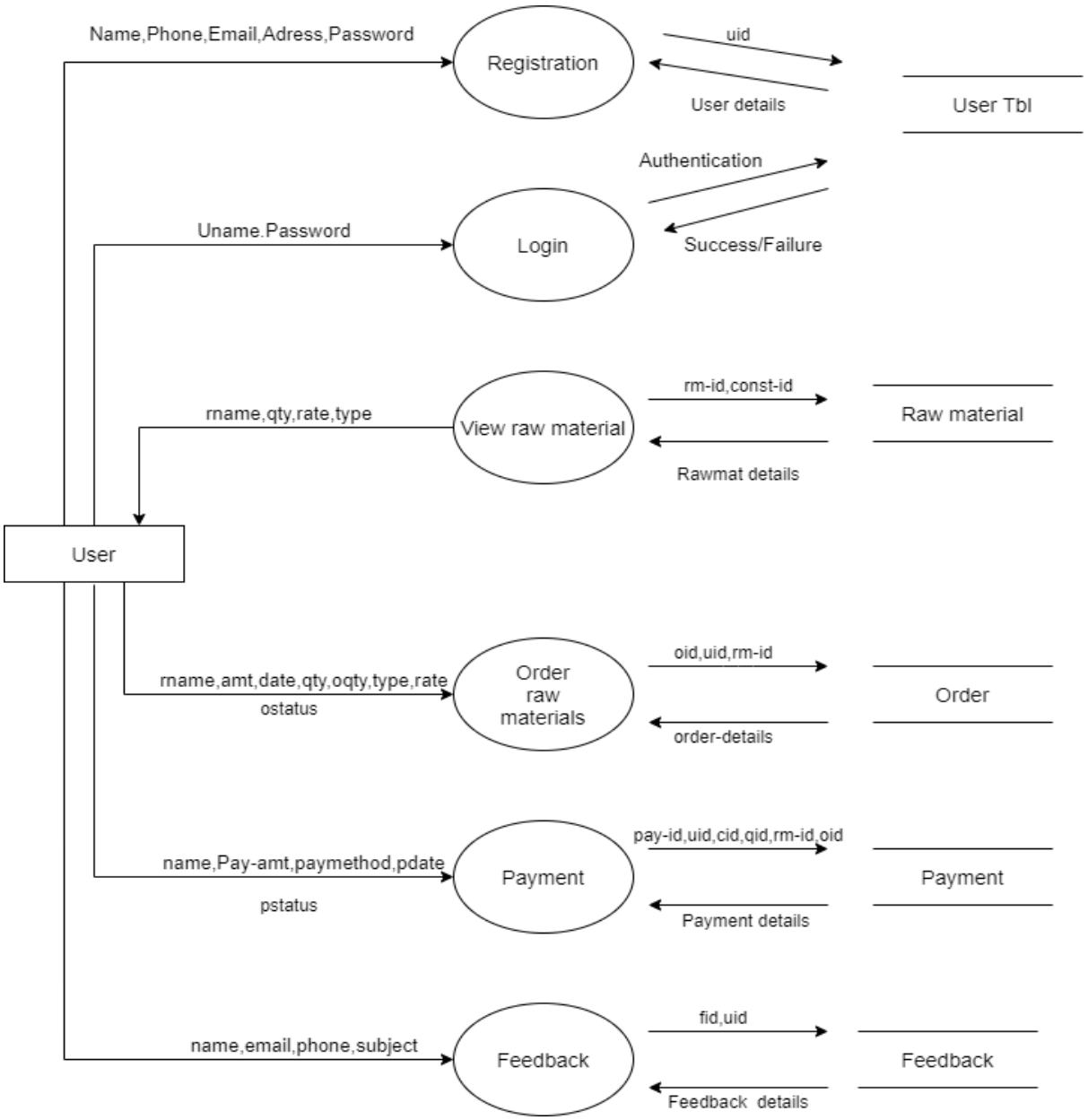
3.3.2 DFD level 1 for Admin



3.3.3 DFD level 1 for Constructor



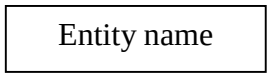
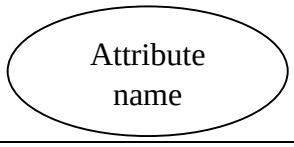
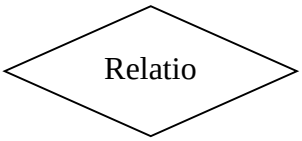
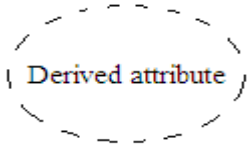
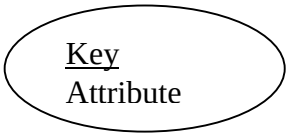
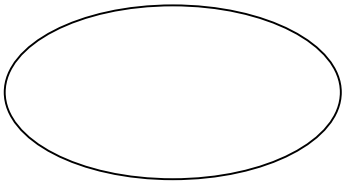
3.3.4 DFD level 1 for User



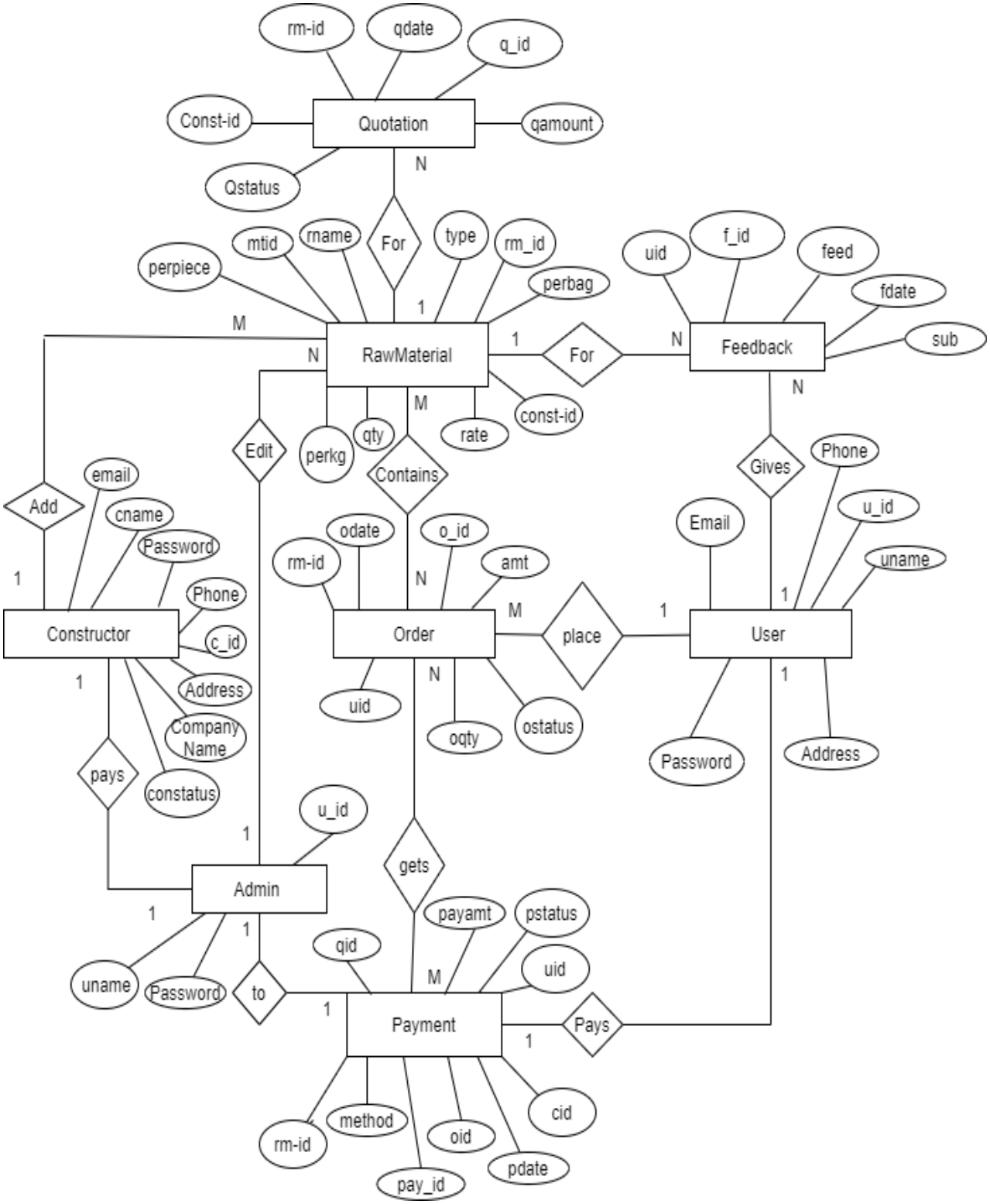
3.4 Entity-Relationship Diagram:

The basic objective of the ER model representation is an entity which is a “thing” in a real world with an independent existence. Entities are physical items or aggregations of data items that are important to the business we analyse or to the system; we intend to build. An entity represents an object defined within the information system about which you want to store information. Entities are named as singular nouns and are shown in rectangles in an ER-Diagram. Each entity is described by several attributes; individual instances of an entity will have different attribute values.

3. 4.1 ER-Diagram Symbols:

Name	Notation	Description
Entity		It may be an object with the physical existence or conceptual existence. It is represented by a Rectangle.
Attribute		The properties of the entity can be a attribute. It is represented by a Ellipse.
Relationship		Whenever an attribute of one entity refers to another entity, some relationship exists. It is represented by a Diamond.
Link		Lines link attributes to entity sets and entity sets to relation.
Derived Attribute		Dashed ellipse denotes derived attributes.
Key Attribute		An entity type usually has an attribute whose values are distinct for each individual entry in the entity set. It is represented by a Underlined word in ellipse.
Multivalued Attribute		Attributes that have different numbers of values for a particular attribute. It is represented by a Double ellipse represents multi-valued attributes.
Cardinality Ratio	1) 1:1 2) 1:M 3) M:1 4) M:N	It specifies the maximum number of relationships instances that an entity can participate in. There are four cardinality ratios.

3.4.2 Entity-Relationship Diagram(ER):



CHAPTER: 4

DETAILED DESIGN

4. DETAIL DESIGN

4.1 INTRODUCTION:

Detailed design is the second level of the design process. During detailed design, we specify how the module in the system interacts with each other and the internal logic of each of the modules specified during system design is decided, hence it is also called as logic design.

Detailed design essentially expands the system design and database design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

4.2 APPLICABLE DOCUMENTS:

Software Requirement Specification(SRS)

System Design Document(SDD)

Database Design Document(DBDD)

The detailed design refines the system design document. Hence the first applicable document here is system design. Also we are referring the data structure. Hence the second applicable document here is database design.

4.3 STRUCTURE OF SOFTWARE PACKAGE :

The software package contains of following functional components

4.3.1 Admin

4.3.2 Constructor

4.3.3 User

4.4 MODULAR DECOMPOSITION OF COMPONENTS

4.4.1 ADMIN LOGIN MODULE

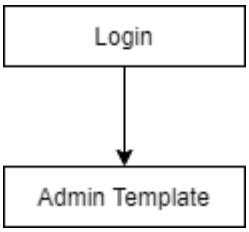
4.4.1.1 DESIGN ASSUMPTION

This module is designed with an intension to provide security to the admin.

4.4.1.2 IDENTIFICATION OF THE MODULE:

Login module does not contain sub module

4.4.1.3 STRUCTURE CHART:



4.4.1.4 MODULE DESIGN FOR LOGIN:

INPUTS:

Admin inputs username and password.

OUTPUT:

On successful login admin enters into the template.

4.4.2 ADMIN MODULE

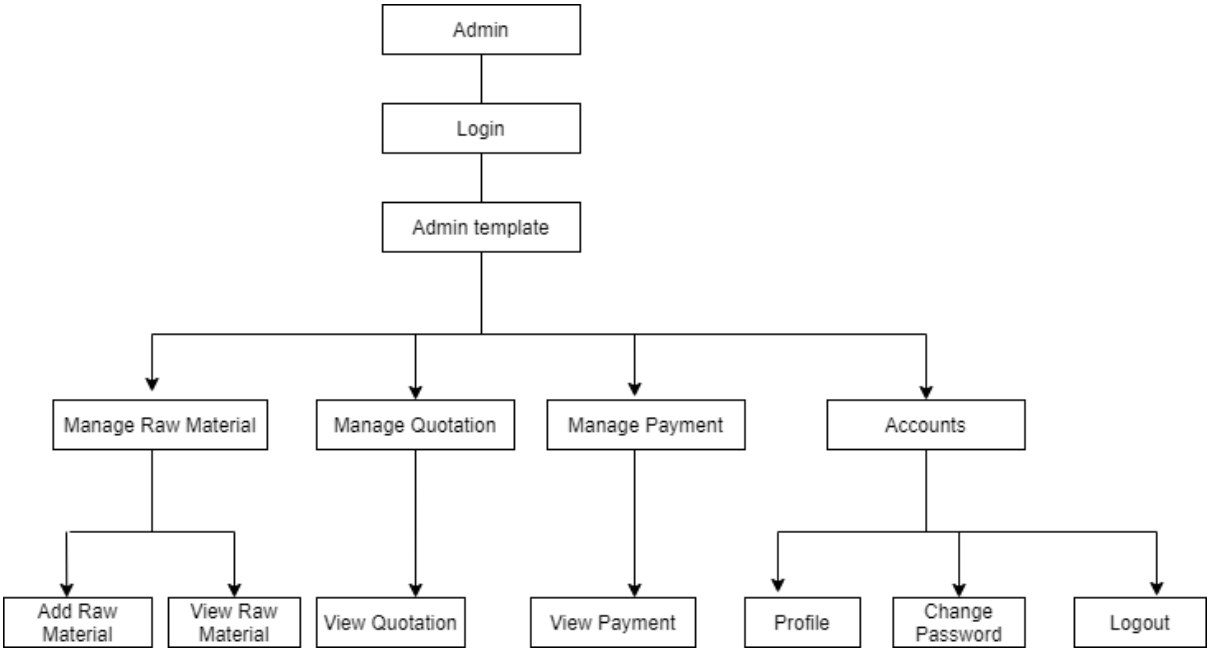
4.4.2.1 DESIGN ASSUMPTION

This module is designed with an intention to allow the admin to perform the specified activities that are available in template.

4.4.2.2 IDENTIFICATION OF THE MODULE:

- Manage Users
  - View User
  - View Constructor
- Manage service
  - View Raw Material
  - View Quotation
  - View Payment
- Feedback
  - View Feedback
- Profile
  - Change Password
  - Logout

4.4.2.3 STRUCTURE CHART



4.4.2.4 Module Design for Admin:

Inputs:

User:uid,uname,password,email,address,phone.

Constructor:cid,cname,Password,Email,CompanyName,Address,Phone

Rawmaterial: rm_id, rname, qty,rate,type,mtid,perpiece,perbag,perkg.

Quotation:qid,rm_id,const_id,qdate,qamount.

Payment: Payid,Pdate,payamount,oid,uid,rm_id.

Feedback:fid,uid,fdate,feed.

Output:

The entered details will be stored in the respective databases and will be displayed in the application.

4.4.3 CONSTRUCTOR MODULE

4.4.3.1 DESIGN ASSUMPTION

This module is designed with an intention to allow the constructor to use any of the options available in the template.

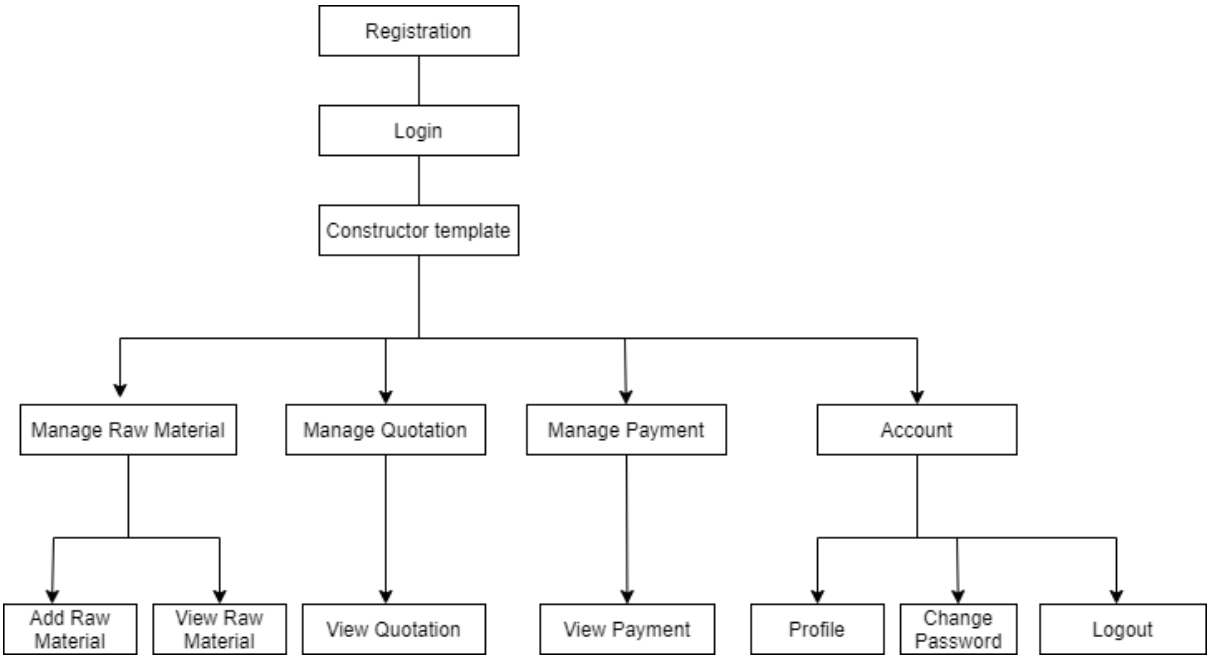
4.4.3.2 IDENTIFICATION OF THE MODULE:

- **Manage Raw Material**

- Add raw material
- View raw material

- **Manage Quotation**
  - View Quotation
- **Manage Payment**
  - View Payment
- **Account**
  - Profile
  - Change Password
  - Logout

4.4.3.3 STRUCTURE CHART



4.4.3.4 Module Design for Constructor

Inputs:

Rawmaterial: rm_id, rname, qty,rate,type,mtid,perpiece,perbag,perkg.

Quotation: qid,rm_id,const_id,qdate,qamount**.**

Payment: Payid,Pdate,payamount,oid,uid,rm_id.

Feedback: fid,uid,fdate,feed.

Output:
The entered details will be stored in the respective databases and will be displayed in the application.

4.4.4 USER MODULE

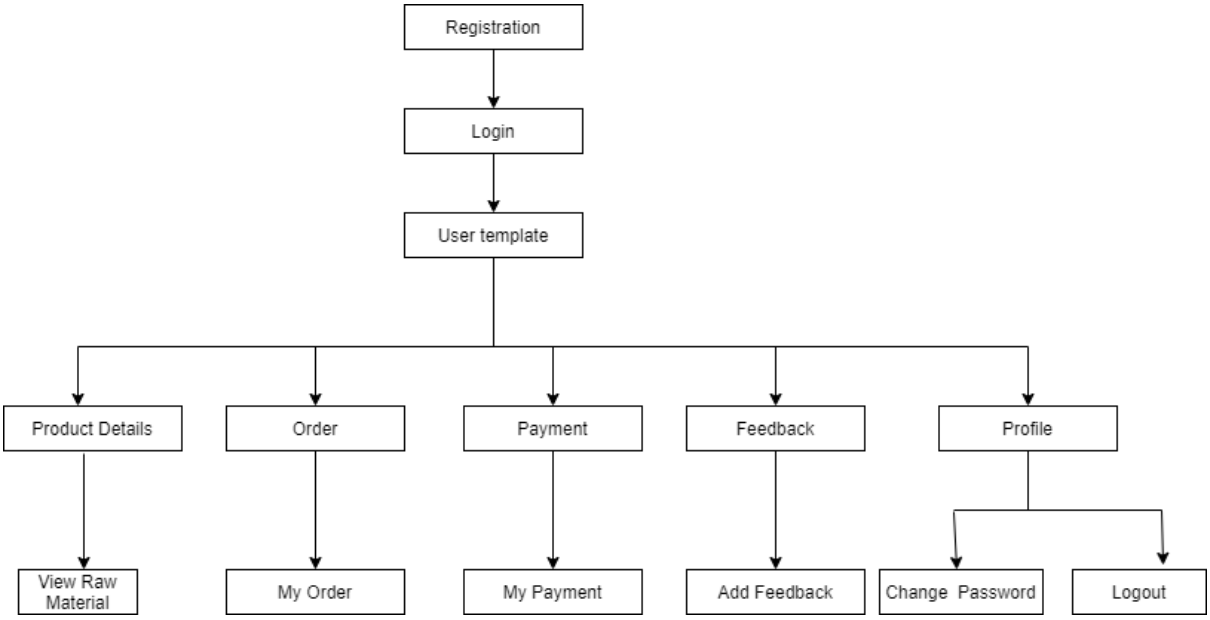
4.4.4.1 DESIGN ASSUMPTION

This module is designed with an intention to allow the user to access the application for retrieval of event information.

4.4.4.2 IDENTIFICATION OF THE MODULE:

- **Product details**
  - View raw material
- **Order**
  - My Order
- **Payment**
  - My Payment
- **Feedback**
  - Add Feedback
- **Profile**
  - Change Password
  - Logout

4.4.4.3 STRUCTURE CHART



4.4.4.4 Module Design for User

Inputs:

Rawmaterial: rm_id, rname, qty,rate,type,mtid,perpiece,perbag,perkg.

Order: oid, odate,oqty,uid,rm_id.

Payment: Payid,Pdate,payamount,oid,uid,rm_id.

Feedback:fid,uid,fdate,feed.

Output:
The entered details will be stored in the respective databases and will be displayed in the application.

CHAPTER: 5

DATABASE DESIGN

5. DATABASE DESIGN

5.1 Introduction:

The word “database”, used to describe everything from a single set of data, to a complex set of tools, such as SQL server, and a whole lot in between. The term data model to mean the conceptual description of the problem space. This includes the definition of entities, their attributes, and the entity constraints. The data model also includes a description of the relationships between entities and any constraints on those relationships. It is the translation of the conceptual model into a physical representation, which shall be implemented using a database management system. The main advantage of this application is to reduce the manual work. The system also does the required calculations, maintains error free reports and validates the final report within a short period of time.

5.2 Purpose :

The ultimate purpose of a database management system is to store and transform data into information to support making decisions. The physical database: the collection of files that contain the data. The database engine: the software that makes it possible to access and modify the contents of the database**.**

5.3 Table Structure:

The database Procurement is organized into following tables:

- Admin
- Constructor
- User
- Rawmaterial
- Quotation
- Order
- Payment
- Feedback

5.3.1 Table Name: Admin Login Table

This table is used to store the login details of Admin.

Admin Login Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	U_id	Int	11	Primary Key	User id
2	Uname	Varchar	123	NotNull	Name of the user
3	Password	Varchar	123	NotNull	Password of the admin

5.3.2 Table Name: UserLogin Table

This table is used to store the UserLogin details

UserLogin Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	U_id	Int	11	Primary Key	User id
2	Uname	Varchar	123	NotNull	Name of the user
3	Password	Varchar	123	NotNull	Password of the User
4	Email	Varchar	123	NotNull	Email id of the User
5	Address	Varchar	123	NotNull	Address of the User
6	Phone	Bigint	11	NotNull	Phone number of the user

5.3.3 Table Name: ConstructorLogin Table

This table is used to store details of ConstructorLogin Table

ConstructorLogin Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	C_id	Int	11	Primary Key	Constructor id
2	Cname	varchar	123	NotNull	Name of the Constructor
3	Password	varchar	121	NotNull	Password of the Constructor
4	Email	Varchar	123	NotNull	Email id of the constructor
5	Companyname	Varchar	123	NotNull	Name of the Constructor Company
6	Address	Varchar	123	NotNull	Address of the Constructor
7	Phone	Int	11	NotNull	Phone number of the Constructor

5.3.4 Table Name: Payment Table

This table is used to store details of Payment Table

Payment Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	Pay_id	Int	11	Primary Key	Payment id
2	Pdate	Date	15	NotNull	Payment date
3	Payamount	Bigint	20	NotNull	Pay amount
4	Oid	Int	11	NotNull	Order id
5	Uid	Int	11	NotNull	User id
6	rm_id	Int	11	NotNull	Raw material id

5.3.5 Table Name: Order Table

This table is used to store details of Order Table

Order Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	O_id	Int	11	Primary Key	Order id
2	Odate	Date	15	NotNull	Order date
3	Oqty	Int	11	NotNull	Order quantity
4	Amount	Int	11	NotNull	Order amount
5	Uid	Int	11	NotNull	User id
6	rm_id	Int	11	NotNull	Raw material id

5.3.6 Table Name: Raw material Table

This table is used to store details of Raw material Table

Raw material Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	rm_id	Int	11	Primary Key	Raw material id
2	rname	Varchar	123	NotNull	Raw material name
3	Qty	Int	11	NotNull	Raw material quantity
4	Rate	Int	11	NotNull	Raw material rate
5	Type	varchar	123	NotNull	Raw material type
6	mtid	Int	11	NotNull	Material id
7	Perpiece	Bigint	20	NotNull	Material per piece
8	Perbag	Bigint	20	NotNull	Material per bag
9	perkg	bigint	20	NotNull	Material per kg

5.3.7 Table Name: Quotation Table

This table is used to store details of Quotation Table

Quotation Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	Q_id	Int	11	Primary Key	Quotation id
2	rm_id	int	11	NotNull	Raw material id
3	const_id	int	11	NotNull	Constructor id
4	qdate	date	15	NotNull	Quotation date
5	qamount	bigint	20	NotNull	Quotation amount

5.3.8 Table Name: Feedback Table

This table is used to store details of Feedback

Feedback Table					
SL.No	**Field Name**	**DataType**	**Size**	**Constraints**	**Description**
1	F_id	int	11	Primary Key	Feedback id
2	Uid	int	11	NotNull	User id
3	Fdate	Date	15	NotNull	Feedback date
4	Feed	Varchar	123	NotNull	Feedback

CHAPTER: 6

CODING

6.CODING

Coding for connection

```
<?php
$con=mysqli_connect("localhost","root","","procurement");
?>
```

Coding for Admin login

```
<?php
include "config.php";
if (isset($_POST['login']))
 {
$uname=$_POST['uname'];
$password=$_POST['password'];
$qry=mysqli_query($con,"SELECT * FROM `admin` WHERE uname='$uname' and password='$password'");
$row=mysqli_fetch_array($qry);
$uid=$row['uid'];
$num=mysqli_num_rows($qry);
if ($num>0)
{
$_SESSION['user']=$uname;
$_SESSION['uid']=$uid;
echo "<script>alert('Success');window.location='../Admin/index.php';</script>";
}
Else
{
```

```
echo "<script>alert('Invalid user name and password');</script>";

}

}

?>
```

Coding for User Registration

```
<div class="limiter">
<div class="container-login100" style="background-image: url('images/bg-01.jpg');">
<div class="wrap-login100 p-t-30 p-b-50">
<span class="login100-form-title p-b-41">
User Registration
</span>
<form class="login100-form validate-form p-b-33 p-t-5" action="" method="POST">

<div class="wrap-input100 validate-input" data-validate = "Enter username">
<input class="input100" type="text" name="uname" pattern="[A-Z a-z]+" title="Only alphabets" placeholder="User name" >
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Enter password">
<input class="input100" type="password" name="password" pattern=".{6,}" title="six or more characters" placeholder="Password" required="">
<span class="focus-input100" data-placeholder="&#xe80f;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter Email">
<input class="input100" type="Email" name="email" placeholder="Email"  required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter Address">
<input class="input100" type="text" name="address" placeholder="Address" required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter Contact">
```

```
<input class="input100" type="text" name="phone" pattern="^\d{10}$" title="Please enter 10 digits contact number" placeholder="Contact" maxlength="10"  required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="container-login100-form-btn m-t-32">
<button class="login100-form-btn" type="submit" name="register">
Register
</button>
<a href="userlogin.php" class="login100-form-btn">Have an Account</a>
</div>
</form>

<?php
include "config.php";
if (isset($_POST['register'])) {
$uname=$_POST['uname'];
$password=$_POST['password'];
$email=$_POST['email'];
$address=$_POST['address'];
$phone=$_POST['phone'];
$qry=mysqli_query($con,"INSERT INTO `user`(`uname`, `password`, `email`, `address`, `phone`) VALUES ('".$uname."','".$password."','".$email."','".$address."','".$phone."')") or die(mysqli_error($con));
echo "<script>alert('Saved');window.location='userlogin.php';</script>";
}
?>
</div>
</div>
</div>
```

Coding for User login

```
<?php
include "config.php";
if (isset($_POST['login'])) {
```

```
$uname=$_POST['uname'];
$password=$_POST['password'];
$qry=mysqli_query($con,"SELECT * FROM `user` WHERE uname='$uname' and password='$password'");
$row=mysqli_fetch_array($qry);
$cid=$row['uid'];
$num=mysqli_num_rows($qry);
if ($num>0)
{
$_SESSION['user']=$uname;
$_SESSION['uid']=$cid;
echo "<script>alert('Success');window.location='../user/userhome.php';</script>";
}
else
{
echo "<script>alert('Invalid user name and password');</script>";
}
}
?>
```

Coding for change User profile

```
<?php
include "config.php";
if (isset($_POST['register'])) {
$cid=$_SESSION['cid'];
$cname=$_POST['cname'];
$password=$_POST['password'];
$email=$_POST['email'];
$CompanyName=$_POST['CompanyName'];
$address=$_POST['address'];
$phone=$_POST['phone'];
$qry=mysqli_query($con,"UPDATE`constructor`SET`cname`='".$cname."',`password`='".$password."',`email`='".$email."',`companyname`='".$CompanyName."',`address`='".$address."
```

```
','`phone`='".$phone."','`constatus`='Approoved'WHEREcid=".$cid)ordie(mysqli_error($con))
;
echo "<script>alert('Saved');window.location='constructorlogin.php';</script>";
}
?>
</div>
</div>
</div>
```

Coding for Constructor Registration

```
<div class="limiter">
<div class="container-login100" style="background-image: url('images/bg-01.jpg');">
<div class="wrap-login100 p-t-30 p-b-50">
<span class="login100-form-title p-b-41">
Construction Registration
</span>
<form class="login100-form validate-form p-b-33 p-t-5" action="" method="POST">

<div class="wrap-input100 validate-input" data-validate = "Enter Constructor name">
<input class="input100" type="text" name="cname" pattern="[A-Z a-z]+" title="Only alphabets" placeholder="Constructor name" required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate="Enter password">
<input class="input100" type="password" name="password" pattern=".{6,}" title="six or more digits"  placeholder="Password" required="">
<span class="focus-input100" data-placeholder="&#xe80f;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter Email">
<input class="input100" type="email" name="email" placeholder="Email" required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter CompanyName">
```

```
<input class="input100" type="text" name="CompanyName" placeholder="CompanyName" required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter Address">
<input class="input100" type="text" name="address" placeholder="Address" required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="wrap-input100 validate-input" data-validate = "Enter Contact">
<input class="input100" type="text" name="phone" pattern="^\d{10}$" title="Please enter 10 digits contact number" placeholder="Contact" maxlength="10" required="">
<span class="focus-input100" data-placeholder="&#xe82a;"></span>
</div>
<div class="container-login100-form-btn m-t-32">
<button class="login100-form-btn" type="submit" name="register">
Register
</button>
<a href="constructorlogin.php" class="login100-form-btn">Have an Account</a>
</div>
</form>
<?php
include "config.php";
if (isset($_POST['register'])) {
$cname=$_POST['cname'];
$password=$_POST['password'];
$email=$_POST['email'];
$CompanyName=$_POST['CompanyName'];
$address=$_POST['address'];
$phone=$_POST['phone'];
$qry=mysqli_query($con,"INSERT INTO `constructor`(`cname`, `password`, `email`, `companyname`,`address`,`phone`)VALUES ('".$cname."','".$password."','".$email."','".$CompanyName."','".$address."','".$phone."')") or die(mysqli_error($con));
```

```
echo"<script>alert('wait untill admin
approovation');window.location='constructorlogin.php';</script>";
}
?>
</div>
</div>
</div>
```

Coding for Constructor login

```
<?php
include "config.php";
if (isset($_POST['login']))
{
$cname=$_POST['cname'];
$password=$_POST['password'];
$qry=mysqli_query($con,"SELECT * FROM `constructor` WHERE cname='$cname' and
password='$password' and constatus='Approoved'");
$row=mysqli_fetch_array($qry);
$cid=$row['cid'];
$num=mysqli_num_rows($qry);
if ($num>0)
{
$_SESSION['user']=$cname;
$_SESSION['cid']=$cid;
echo
"<script>alert('Success');window.location='../constructor/constructorhome.php';</script>";
}
else
{
echo "<script>alert('Invalid user name and password');</script>";
}
}
?>
```

Coding for change Constructor profile

```
<?php
include "config.php";
if (isset($_POST['register'])) {
$cid=$_SESSION['cid'];
$cname=$_POST['cname'];
$password=$_POST['password'];
$email=$_POST['email'];
$CompanyName=$_POST['CompanyName'];
$address=$_POST['address'];
$phone=$_POST['phone'];
$qry=mysqli_query($con,"UPDATE`constructor`SET`cname`='".$cname."',`password`='".$p
assword."',`email`='".$email."',`companyname`='".$CompanyName."',`address`='".$address."
',`phone`='".$phone."',`constatus`='Approoved'WHEREcid=".$cid)ordie(mysqli_error($con))
;
echo "<script>alert('Saved');window.location='constructorlogin.php';</script>";
}
?>
</div>
</div>
</div>
```

Admin:

Coding for View User

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM user");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['uname']; ?></td>
```

```
<td><?php echo $row['email']; ?></td>
<td><?php echo $row['address']; ?></td>
<td><?php echo $row['phone']; ?></td>
<td>
<a href="deluser.php?id=<?php echo $row['uid']; ?>" class="btn btn-danger" title="Delete" onclick="return confirm('Are you sure to delete');"><i class="fa fa-trash"></i></a></td>
</tr>
<?php }
?>
```

Coding for View constructor

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM constructor");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['cname']; ?></td>
<td><?php echo $row['email']; ?></td>
<td><?php echo $row['address']; ?></td>
<td><?php echo $row['phone']; ?></td>
<td><?php echo $row['companyname']; ?></td>
<?php
$st=$row['constatus'];
if ($st=='NotYet') { ?>
<td>
<a href="upcon.php?id=<?php echo $row['cid']; ?>" class="btn btn-primary" title="Update"><i class="fa fa-edit"></i></a>
<a href="delcon.php?id=<?php echo $row['cid']; ?>" class="btn btn-danger" title="Delete" onclick="return confirm('Are you sure to delete');"><i class="fa fa-trash"></i></a></td>
<?php   }
```

```
elseif ($st=='Approoved') { ?>
<td><span class="btn btn-primary">Approoved</span></td>
<?php  }
?>
</tr>
<?php }
 ?>
```

Coding for View raw material

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM raw_mat");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
</td>
</tr>
<?php } ?>
```

Coding for view Quotation

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM raw_mat INNER JOIN quotation on raw_mat.rm_id=quotation.rm_id inner join constructor on quotation.const_id=constructor.cid");
while($row=mysqli_fetch_array($sel))
```

```
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
<td><?php echo $row['qamount']; ?></td>
<td><?php echo $row['qdate']; ?></td>
<?php
$qst=$row['Qstatus'];
if ($qst=='Notyet') { ?>
<td><a href="Apquotation.php?id=<?php echo $row['qid']; ?>" class="btn btn-primary" title="Update"><i class="fa fa-check"></i></a>
<a href="Requotation.php?id=<?php echo $row['qid']; ?>" class="btn btn-primary" title="Update"><i class="fa fa-close"></i></a>
</td>
<?php }
elseif ($qst=='Approoved') { ?>
<td colspan="2"><span class="btn btn-primary">Approoved</span>
<a href="Makepayment.php?qid=<?php echo $row['qid']; ?>&cid=<?php echo $row['const_id']; ?>&qamount=<?php echo $row['qamount']; ?>&rmid=<?php echo $row['rm_id']; ?>" class="btn btn-primary" title="Update">Pay</a>
</td>

<?php }
elseif ($qst=='Rijected') { ?>
<td><span class="btn btn-primary">Rijected</span></td>
<?php  }
elseif ($qst=='Paid') { ?>
<td><span class="btn btn-primary">Payment Done</span></td>
<?php     }
?>
```

```
</tr>
<?php }
?>
```

Coding for Order

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM `orders` INNER JOIN raw_mat on orders.rm_id=raw_mat.rm_id INNER JOIN user on orders.uid=user.uid");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['uname']; ?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
<td><?php echo $row['oqty']; ?></td>
<td><?php echo $row['odate']; ?></td>
<td><?php echo $row['Ostatus']; ?></td>
<?php
$st=$row['Ostatus'];
if ($st=='Ordered') { ?>
<td><a href="confirmorder.php?id=<?php echo $row['oid']; ?>" class="btn btn-primary" title="Update">Confirm</a>
<a href="cancelorder.php?id=<?php echo $row['oid']; ?>" class="btn btn-primary" title="Update">Cancel</a></td>
<?php  }
elseif ($st=='Confirmed') { ?>
<td> <span class="btn btn-primary">Confirmed</span></td>
<?php  }
```

```
elseif ($st=='Cancelled') { ?>
<td> <span class="btn btn-primary">Cancelled</span></td>
<?php  }
elseif ($st=='Paid') { ?>
<td>  <span class="btn btn-primary">Paid</span></td>
<?php  }
?>
</td>
</tr>
<?php }
 ?>
```

Coding for Payment

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM `payment` INNER JOIN user on payment.uid=user.uid  ");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['uname']; ?></td>
<td><?php echo $row['payamount']; ?></td>
<td><?php echo $row['method']; ?></td>
<td><?php echo $row['pdate']; ?></td>
<td><?php echo $row['pstatus']; ?></td>
<?php
$st=$row['pstatus'];
if ($st=='Paid') { ?>
<td><a href="confirmpay.php?id=<?php echo $row['payid']; ?>" class="btn btn-primary" title="Update">Confirm</a>
```

```
<?php  }
elseif ($st=='Paid') { ?>
<td> <span class="btn btn-primary">Paid</span></td>
<?php  }
elseif ($st=='None') { ?>
<td>  <span class="btn btn-primary">None</span></td>
<?php  }
elseif ($st=='Confirmed') { ?>
<td>
<span class="btn btn-primary">Confirmed</span>
</td>
<?php  }
?>
</td>
</tr>
<?php }
 ?>
```

Coding for view Feedback

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM `feedback` INNER JOIN user on feedback.uid=user.uid  ");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['uname']; ?></td>
<td><?php echo $row['sub']; ?></td>
<td><?php echo $row['feed']; ?></td>
<td><?php echo $row['fdate']; ?></td>
```

```
<td><a href="delfeed.php?id=<?php echo $row['fid']; ?>" class="btn btn-primary" title="Update">Delete</a>
</td>
</tr>
<?php }
 ?>
```

Coding for Add Quotation

```
<?php include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"SELECT * FROM raw_mat where rm_id=".$id);
$row=mysqli_fetch_array($qry);
?>
<form action="" method="POST" id="xs-contact-form" class="xs-form">
<div class="row">
<div class="col-md-6">
<input type="text" class="form-control" name="rname" value="<?php echo $row['rname'];?>" placeholder="Your raw material name" id="xs_contact_name" readonly="">
<input type="hidden" name="id" value="<?php echo $id;?>">
</div>
<div class="col-md-6">
<input type="text" class="form-control invaild" name="qdate" placeholder="Quantity" id="xs_contact_email" value="<?php echo date('y-m-d');?>">
</div>
</div>
<div class="row">
<div class="col-md-6">
<input type="text" class="form-control" name="qamount" placeholder="Amount" id="xs_contact_phone">
</div>
</div>
<div class="readmore text-center">
<button class="main-btn btn-1 btn-1e" type="submit" name="add">ADD</button>
```

```
</div>
</form>
<?php
include "config.php";
if (isset($_POST['add'])) {
$id=$_POST['id'];
$qdate=$_POST['qdate'];
$qamount=$_POST['qamount'];
$qry=mysqli_query($con,"INSERT INTO `quotation`(`rm_id`, `const_id`, `qdate`, `qamount`) VALUES ('".$id."','".$qdate."','".$qamount."')") or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');window.location='index.php';</script>";
}
?>
</div>
```

Coding for approovation

```
<?php
include "config.php";
if (isset($_POST['add'])) {
$cid=$_SESSION['cid'];
$rmid=$_POST['rmid'];
$id=$_POST['id'];
$qdate=$_POST['qdate'];
$qamount=$_POST['qamount'];
$qry=mysqli_query($con,"UPDATE`quotation`SET`rm_id`='".$rmid."',`const_id`='".$cid."',`qdate`='".$qdate."',`qamount`='".$qamount."' WHERE qid=".$id) or die(mysqli_error($con));
echo"<script>alert('Approved Successfully');window.location='viewquotation.php';</script>";
}
?>
```

Coding for Approove quotation

```
<?php
include "config.php";
```

```
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE `quotation` SET `Qstatus`='Approoved' WHERE qid=".$id) or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');window.location='viewquotation.php';</script>";
?>
```

Coding for Cancel Order

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE `orders` SET `Ostatus`='Cancelled' WHERE oid=".$id);
echo "<script>alert('Cancelled Successfully');window.location='vieworder.php';</script>";
?>
```

Coding for Confirm Order

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE `orders` SET `Ostatus`='Confirmed' WHERE oid=".$id);
echo "<script>alert('Confirmed Successfully');window.location='vieworder.php';</script>";
?>
```

Coding for Confirm Payment

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE `payment` SET `pstatus`='Confirmed' WHERE payid=".$id);
echo "<script>alert('Confirmed Successfully');window.location='viewpayment.php';</script>";
?>
```

Coding for Deletion:

Coding for Delete Constructor

```
<?php
include "config.php";
```

```
$id=$_GET['id'];
$qry=mysqli_query($con,"DELETE FROM constructor where cid=".$id);
echo "<script>alert('Deleted
Successfully');window.location='viewconstructor.php';</script>";
?>
```

Coding for Delete User

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"DELETE FROM user where uid=".$id);
echo "<script>alert('Deleted Successfully');window.location='viewuser.php';</script>";
?>
```

Coding for Delete feedback

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"DELETE FROM feedback where fid=".$id);
echo "<script>alert('Deleted Successfully');window.location='viewfeedback.php';</script>";
?>
```

Coding for MakePayment

```
<?php
include "config.php";
$qid=$_GET['qid'];
$qamount=$_GET['qamount'];
$cid=$_GET['cid'];
$rmid=$_GET['rmid'];
?>
<form action="" method="POST" style="width:100%; ">
<div class="card" >
<div class="card-header">
<div class="card-title">Form to add Payment</div>
</div>
<div class="card-body" >
```

```
<div class="row">
<div class="col-md-12">
<div class="form-group has-success">
<label for="successInput">Quotationid</label>
<input type="text" id="successInput" class="form-control" name="qid" value="<?php echo $qid;?>" readonly="TRUE">
</div>
<div class="form-group has-success">
<label for="successInput">ConstId</label>
<input type="text" name="cid" class="form-control" value="<?php echo $cid;?>">
</div>
<div class="form-group has-success">
<label for="successInput">Raw Mat Id</label>
<input type="text" name="rmid" class="form-control" value="<?php echo $rmid;?>">
</div>
<div class="form-group has-success">
<label for="successInput">PaymentMethod</label>
<label for="chkYes">
<input type="radio" id="chkYes" value="card" name="chkPassPort"  required="" />
Card
</label>
<label for="chkNo">
<input type="radio" id="chkNo" value="cash" name="chkPassPort" required="" />
Cash
</label>
<hr />
<div id="dvPassport" style="display: none">
CardHolderName:
<input type="text" id="txtPassportNumber" pattern="[A-Za-z\s]+" title="characters ad space only" class="form-control" />
CardNumber:
<input type="number" id="txtPassportNumber" pattern="^\d{16}$" title="16 digit numbers only" placeholder="Enter 16 digit Card Number" class="form-control" />
```

```
CVV:
<input type="number" id="txtPassportNumber"  pattern="^\d{3}$" title="3 digits only" class="form-control" placeholder="Enter cvv" class="form-control" />
ExpDate:
<input type="date" id="txtPassportNumber" class="form-control" />
</div>
</div>
<div class="form-group has-success">
<label for="successInput">Amount</label>
<input type="number" id="successInput" class="form-control" name="amount"value="<?php echo $qamount;?>" readonly="TRUE">
</div>
</div>
</div>
</div>
<div class="card-action">
<button class="btn btn-success" name="add" type="submit">Save</button>
<button class="btn btn-danger" type="reset">Cancel</button>
</div>
</div>
</form>
<?php
include "config.php";
if (isset($_POST['add'])) {
$cid=$_POST['cid'];
$rmid=$_POST['rmid'];
$qid=$_POST['qid'];
$amount=$_POST['amount'];
$chkPassPort=$_POST['chkPassPort'];
$date=date('y-m-d');
$qry=mysqli_query($con,"INSERT INTO `payment`(`cid`, `rm_id`, `qid`, `pdate`,`method`, `payamount`,`pstatus`)VALUES('".$cid."','".$rmid."','".$qid."','".$date."','".$chkPassPort."','".$amount."','Paid')");
$qry=mysqli_query($con,"UPDATE `quotation` SET `qstatus`='Paid' WHERE qid=".$qid);
```

```
echo "<script>alert('Saved');window.location='index.php';</script>";
}
?>
</div>
</div>
</div>
```

Coding for ReQuotation

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE `quotation` SET `Qstatus`='Rejected' WHERE qid=".$id) or die(mysqli_error($con));
echo "<script>alert('Rejected Successfully');window.location='viewquotation.php';</script>";
?>
```

Coding for Reject Order

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE `orders` SET `Ostatus`='Cancelled' WHERE oid=".$id);
echo "<script>alert('Confirmed Successfully');window.location='vieworder.php';</script>";
?>
```

Coding for Update Constructor

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"UPDATE constructor set `constatus`='Approoved' where cid=".$id);
echo"<script>alert('Approved Successfully');window.location='viewconstructor.php';</script>";
?>
```

User:

Coding for View raw material

```
<?php
```

```
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM raw_mat");
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
<td><a href="orders.php?id=<?php echo $row['rm_id']; ?>" class="btn btn-primary" title="Update">Ordernow</a>
</td>
</tr>
<?php }
 ?>
```

Coding for Order

```
<?php include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"SELECT * FROM raw_mat where rm_id=".$id);
$row=mysqli_fetch_array($qry);
?>
<form action="" method="POST" id="xs-contact-form" class="xs-form">
<div class="row">
<div class="col-md-6">
<input type="text" class="form-control" name="rname" placeholder="Your raw material name" id="xs_contact_name" value="<?php echo $row['rname'];?>" >
<input type="hidden" name="id" value="<?php echo $id;?>">
<input type="hidden" name="rate" value="<?php echo $row['rate'];?>">
</div>
<div class="col-md-6">
```

```
<input type="number" class="form-control invaild" name="oqty" placeholder="Quantity" id="xs_contact_email">
</div>
</div>
<div class="readmore text-center">
<button class="main-btn btn-1 btn-1e" type="submit" name="add">ADD</button>
</div>
</form>
<?php
include "config.php";
if (isset($_POST['add'])) {
$uid=$_SESSION['uid'];
$id=$_POST['id'];
$qty=$_POST['oqty'];
$rate=$_POST['rate'];
$amt=$qty*$rate;
$date=date('y-m-d');
$qry=mysqli_query($con,"INSERT INTO `orders`( `uid`, `rm_id`, `oqty`, `amount`, `odate`,`Ostatus`) VALUES ('".$uid."','".$id."','".$qty."','".$amt."','".$date."','Ordered')") or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');window.location='ViewRM.php';</script>";
}
?>
</div>
```

Coding for Paynow

```
<?php
include "config.php";
$uid=$_SESSION['uid'];
$id1=$_GET['oid'];
$id2=$_GET['rmid'];
$id=$_GET['amount'];
$qry=mysqli_query($con,"SELECT * FROM `raw_mat` WHERE rm_id=".$id2);
$row=mysqli_fetch_array($qry);
```

```
?>
<form action="" method="POST" style="width:100%; ">
<div class="card" >
<div class="card-header">
<div class="card-title">Form to add Payment</div>
</div>
<div class="card-body" >
<div class="row">
<div class="col-md-12">
<div class="form-group has-success">
<label for="successInput">Orderid</label>
<input type="text" id="successInput" class="form-control" name="id1" value="<?php echo $id1;?>" readonly="TRUE">
</div>
<div class="form-group has-success">
<label for="successInput">Material</label>
<input type="text" id="successInput" value="<?php echo $row['rname'];?>" class="form-control" name="title"readonly="TRUE">
<input type="hidden" name="uid" value="<?php echo $uid;?>">
<input type="hidden" name="rmid" value="<?php echo $id2;?>">
</div>
<div class="form-group has-success">
<label for="successInput">PaymentMethod</label>
<label for="chkYes">
<input type="radio" id="chkYes" value="card" name="chkPassPort" required="" />
Card
</label>
<label for="chkNo">
<input type="radio" id="chkNo" value="cash" name="chkPassPort" required="" />
Cash
</label>
<hr />
<div id="dvPassport" style="display: none">
CardHolderName:
```

```
<input type="text" id="txtPassportNumber" pattern="[A-Za-z\s]+" title="characters ad space only" class="form-control" />
CardNumber:
<input type="number" id="txtPassportNumber" pattern="^\d{16}$" title="16 digit numbers only" placeholder="Enter 16 digit Card Number" class="form-control" />
CVV:
<input type="number" id="txtPassportNumber"  pattern="^\d{3}$" title="3 digits only" class="form-control" placeholder="Enter cvv" class="form-control" />
ExpDate:
<input type="date" id="txtPassportNumber" class="form-control" />
</div>
</div>
<div class="form-group has-success">
<label for="successInput">Amount</label>
<input type="number" id="successInput" class="form-control" name="amount"value="<?php echo $id;?>" readonly="TRUE">
</div>
</div>
</div>
</div>
<div class="card-action">
<button class="btn btn-success" name="add" type="submit">Save</button>
<button class="btn btn-danger" type="reset">Cancel</button>
</div>
</div>
</form>
<?php
include "config.php";
if (isset($_POST['add'])) {
$uid=$_POST['uid'];
$rmid=$_POST['rmid'];
$id1=$_POST['id1'];
$amount=$_POST['amount'];
$chkPassPort=$_POST['chkPassPort'];
```

```
$date=date('y-m-d');
$qry=mysqli_query($con,"INSERT INTO `payment`(`uid`, `rm_id`, `oid`, `pdate`,`method`, `payamount`,`pstatus`)VALUES ('".$uid."','".$rmid."','".$id1."','".$date."','".$chkPassPort."','".$amount."','Paid')");
$qry=mysqli_query($con,"UPDATE `orders` SET `Ostatus`='Paid' WHERE oid=".$id1);
echo "<script>alert('Saved');window.location='Userhome.php';</script>";
}
?>
</div>
```

Coding for MyPayment

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM `payment` INNER JOIN raw_mat on payment.rm_id=raw_mat.rm_id WHERE payment.uid=".$_SESSION['uid']);
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['payamount']; ?></td>
<td><?php echo $row['method']; ?></td>
<td><?php echo $row['pdate']; ?></td>
<td><?php echo $row['pstatus']; ?></td>
<?php
$st=$row['pstatus'];
if ($st=='Confirmed') { ?>
<td><a href="bill.php?rmid=<?php echo $row['rm_id']; ?>&payid=<?php echo $row['payid'];?>&amount=<?php echo $row['payamount'];?>" class="btn btn-primary" title="Update">Bill</a>
<?php  }
elseif ($st=='Paid') { ?>
```

```
<td> <span class="btn btn-primary">Paid</span></td>
<?php  }
elseif ($st=='None') { ?>
<td>  <span class="btn btn-primary">None</span></td>
<?php  }
?>
</td>
</tr>
<?php } ?>
```

Coding for MyOrder

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM `orders` INNER JOIN raw_mat on orders.rm_id=raw_mat.rm_id WHERE orders.uid=".$_SESSION['uid']);
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
<td><?php echo $row['oqty']; ?></td>
<td><?php echo $row['amount']; ?></td>
<td><?php echo $row['odate']; ?></td>
<td><?php echo $row['Ostatus']; ?></td>
<?php
$st=$row['Ostatus'];
if ($st=='Confirmed') { ?>
```

```
<td><a href="Paynow.php?rmid=<?php echo $row['rm_id']; ?>&oid=<?php echo $row['oid'];?>&amount=<?php echo $row['amount'];?>" class="btn btn-primary" title="Update">Paynow</a>
<?php  }
elseif ($st=='Ordered') { ?>
<td> <span class="btn btn-primary">Ordered</span></td>
<?php  }
elseif ($st=='Paid') { ?>
<td>  <span class="btn btn-primary">Paid</span></td>
<?php  }
?>
</td>
</tr>
<?php } ?>
```

Coding for Feedback

```
<?php
include "config.php";
if (isset($_POST['send'])) {
$uid=$_SESSION['uid'];
$sub=$_POST['Subject'];
$feed=$_POST['message'];
$date=date('y-m-d');
$qry=mysqli_query($con,"INSERT INTO `feedback`(`uid`, `fdate`, `sub`, `feed`) VALUES ('".$uid."','".$date."','".$sub."','".$feed."')") or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');</script>";
}
?>
```

Coding for Bill

```
<?php include "config.php";
$qry=mysqli_query($con,"SELECT * FROM  payment ,  user  where payment.uid=".$_SESSION['uid']);
$rw=mysqli_fetch_array($qry);
```

```
?>
<tr>
<th>UserName:<p><?php echo $_SESSION['user'];?></p></th>
<th>Phone:<?php echo $rw['phone'];?></th>
</tr>
<tr>
<th>SL.No</th>
<th>Material</th>
<th style="text-align: right;">Rate</th>
<th style="text-align: right;">OrderedQUANTITY</th>
<th style="text-align: right;">AMOUNT</th>
</tr>
<?php include("config.php");
$id=$_GET['payid'];
$query = mysqli_query($con,"SELECT * FROM `orders`,`raw_mat`,`payment` where `payment`.oid=`orders`.oid and `payment`.rm_id=`raw_mat`.rm_id and `payment`.payid=".$id);
$i=1;
while($row = mysqli_fetch_array($query)){
?>
<tr style=" line-height: 90%; font-weight: bold;">
<td> <?php echo $i++; ?> </td>
<td style="text-align: right;"><?php echo $row['rname']; ?>  </td>
<td style="text-align: right;"> <?php echo $row['rate']; ?> </td>
<td style="text-align: right;"> <?php echo number_format($row['oqty'],2); ?> </td>
<td style="text-align: right;"> <?php echo number_format($row['payamount'],2); ?> </td>
</tr>
<?php }
?>
```

Constructor:

Coding for Add Raw Material

```
<?php
include "config.php";
if (isset($_POST['add'])) {
```

```
$cid=$_SESSION['cid'];
$rname=$_POST['rname'];
$qty=$_POST['qty'];
$type=$_POST['type'];
$rate=$_POST['rate'];
$qry=mysqli_query($con,"INSERT INTO `raw_mat`(`const_id`,`rname`, `qty`, `rate`, `type`)
VALUES ('".$cid."','".$rname."','".$qty."','".$rate."','".$type."')") or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');</script>";
}
?>
```

Coding for Add Quotation

```
<?php
include "config.php";
if (isset($_POST['add'])) {
$cid=$_SESSION['cid'];
$id=$_POST['id'];
$qdate=$_POST['qdate'];
$qamount=$_POST['qamount'];
$qry=mysqli_query($con,"INSERT INTO `quotation`(`rm_id`, `const_id`, `qdate`,
`qamount`) VALUES ('".$id."','".$cid."','".$qdate."','".$qamount."')") or
die(mysqli_error($con));
echo "<script>alert('Saved Successfully');window.location='viewquotation.php';</script>";
}
?>
```

Coding for View Raw Material

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM raw_mat where const_id=".$_SESSION['cid']);
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
```

```
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
<td><a href="upRM.php?id=<?php echo $row['rm_id']; ?>" class="btn btn-primary" title="Update"><i class="fa fa-edit"></i></a> <a href="delRM.php?id=<?php echo $row['rm_id']; ?>" class="btn btn-danger" title="Delete" onclick="return confirm('Are you sure to delete');"><i class="fa fa-trash"></i></a>
<a href="addquotation.php?id=<?php echo $row['rm_id']; ?>" class="btn btn-primary" title="Update"><i class="fa fa-plus"></i></a>
</td>
</tr>
<?php }
 ?>
```

Coding for View Quotation

```
<?php
include "config.php";
$i=1;
$cid=$_SESSION['cid'];
$sel=mysqli_query($con,"SELECT * FROM raw_mat INNER JOIN quotation on raw_mat.rm_id=quotation.rm_id WHERE quotation.const_id=".$cid);
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['qty']; ?></td>
<td><?php echo $row['rate']; ?></td>
<td><?php echo $row['type']; ?></td>
<td><?php echo $row['qamount']; ?></td>
<td><?php echo $row['qdate']; ?></td>
```

```
<td><a href="Upquotation.php?id=<?php echo $row['qid']; ?>" class="btn btn-primary" title="Update"><i class="fa fa-edit"></i></a> <a href="delQt.php?id=<?php echo $row['qid']; ?>" class="btn btn-danger" title="Delete" onclick="return confirm('Are you sure to delete');"><i class="fa fa-trash"></i></a>
</td>
</tr>
<?php }
?>
```

Coding for Update Raw Material

```
<?php include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"SELECT * FROM raw_mat where rm_id=".$id);
$row=mysqli_fetch_array($qry);
?>
<form action="" method="POST" id="xs-contact-form" class="xs-form">
<div class="row">
<div class="col-md-6">
<input type="text" class="form-control" name="rname" value="<?php echo $row['rname'];?>" placeholder="Your raw material name" id="xs_contact_name">
<input type="hidden" name="id" value="<?php echo $id;?>">
</div>
<div class="col-md-6">
<input type="number" class="form-control invaild" name="qty"value="<?php echo $row['qty'];?>" placeholder="Quantity" id="xs_contact_email">
</div>
</div>
<div class="row">
<div class="col-md-6">
<input type="text" class="form-control" name="type" placeholder="Type" value="<?php echo $row['type'];?>" id="xs_contact_phone">
</div>
<div class="col-md-6">
```

```
<input type="number" class="form-control" name="rate"value="<?php echo $row['rate'];?>"
placeholder="Rate" id="xs_contact_subject">
</div>
</div>
<div class="readmore text-center">
<button class="main-btn btn-1 btn-1e" type="submit" name="add">ADD</button>
</div>
</form>
<?php
include "config.php";
if (isset($_POST['add'])) {
$id=$_POST['id'];
$rname=$_POST['rname'];
$qty=$_POST['qty'];
$type=$_POST['type'];
$rate=$_POST['rate'];
$qry=mysqli_query($con,"UPDATE`raw_mat`SET`rname`='".$rname."',`qty`='".$qty."',`rate
`='".$rate."',`type`='".$type."' WHERE rm_id=".$id) or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');window.location='ViewRM.php';</script>";
}
?>
</div>
```

Coding for Update Quotation

```
<?php include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"SELECT * FROM raw_mat INNER JOIN quotation on
raw_mat.rm_id=quotation.rm_id WHERE quotation.qid=".$id);
$row=mysqli_fetch_array($qry);
?>
<form action="" method="POST" id="xs-contact-form" class="xs-form">
<div class="row">
<div class="col-md-6">
```

```
<input type="text" class="form-control" name="rname" value="<?php echo $row['rname'];?>" placeholder="Your raw material name" id="xs_contact_name" readonly="">
<input type="hidden" name="rmid" value="<?php echo $row['rm_id'];?>">
<input type="hidden" name="id" value="<?php echo $id;?>">
</div>
<div class="col-md-6">
<input type="text" class="form-control invaild" name="qdate" placeholder="Quantity" id="xs_contact_email" value="<?php echo date('y-m-d');?>">
</div>
</div>
<div class="row">
<div class="col-md-6">
<input type="text" class="form-control" name="qamount" value="<?php echo $row['qamount'];?>"  placeholder="Amount" id="xs_contact_phone">
</div>
</div>
<div class="readmore text-center">
<button class="main-btn btn-1 btn-1e" type="submit" name="add">Edit</button>
</div>
</form>

<?php
include "config.php";
if (isset($_POST['add'])) {
$cid=$_SESSION['cid'];
$rmid=$_POST['rmid'];
$id=$_POST['id'];
$qdate=$_POST['qdate'];
$qamount=$_POST['qamount'];
$qry=mysqli_query($con,"UPDATE`quotation`SET`rm_id`='".$rmid."',`const_id`='".$cid."',`qdate`='".$qdate."',`qamount`='".$qamount."' WHERE qid=".$id) or die(mysqli_error($con));
echo "<script>alert('Saved Successfully');window.location='viewquotation.php';</script>";
}
```

```
?>
</div>
```

Coding for Deletion:

Coding for Delete Raw Material

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"DELETE FROM raw_mat where rm_id=".$id);
echo "<script>alert('Deled Successfully');window.location='ViewRM.php';</script>";
?>
```

Coding for Delete Quotation

```
<?php
include "config.php";
$id=$_GET['id'];
$qry=mysqli_query($con,"DELETE FROM quotation where qid=".$id);
echo "<script>alert('Deled Successfully');window.location='Viewquotation.php';</script>";
?>
```

Coding for Payment

```
<?php
include "config.php";
$i=1;
$sel=mysqli_query($con,"SELECT * FROM `payment` INNER JOIN raw_mat on payment.rm_id=raw_mat.rm_id  WHERE payment.cid=".$_SESSION['cid']);
while($row=mysqli_fetch_array($sel))
{
?>
<tr>
<td><?php echo $i++;?></td>
<td><?php echo $row['rname']; ?></td>
<td><?php echo $row['payamount']; ?></td>
<td><?php echo $row['method']; ?></td>
<td><?php echo $row['pdate']; ?></td>
<td><?php echo $row['pstatus']; ?></td>
<?php
```

```
$st=$row['pstatus'];
if ($st=='Paid') { ?>
<td><a href="bill.php?rmid=<?php echo $row['rm_id']; ?>&payid=<?php echo $row['payid'];?>&amount=<?php echo $row['payamount'];?>" class="btn btn-primary" title="Update">Bill</a>
<?php  }
elseif ($st=='None') { ?>
<td>  <span class="btn btn-primary">None</span></td>
<?php  }
?>
</td>
</tr>
<?php }
 ?>
```

Coding for Bill

```
<?php include "config.php";
$qry=mysqli_query($con,"SELECT * FROM payment , user where payment.cid=".$_SESSION['cid']);
$rw=mysqli_fetch_array($qry);
?>
<tr>
<th>UserName:<p><?php echo $_SESSION['user'];?></p></th>
<th>Phone:<?php echo $rw['phone'];?></th>
</tr>
<tr>
<th>SL.No</th>
<th>Material</th>
<th style="text-align: right;">Rate</th>
<th style="text-align: right;">QuotationAmount</th>
<th style="text-align: right;">PaidAMOUNT</th>
</tr>
<?php include("config.php");
$id=$_GET['payid'];
```

```
$query = mysqli_query($con,"SELECT * FROM `quotation`,`raw_mat`,`payment` where `payment`.qid=`quotation`.qid and `payment`.rm_id=`raw_mat`.rm_id and `payment`.`payid`=".$id);
$i=1;
while($row = mysqli_fetch_array($query)){
?>
<tr style=" line-height: 90%; font-weight: bold;">
<td> <?php echo $i++; ?> </td>
<td style="text-align: right;"><?php echo $row['rname']; ?>  </td>
<td style="text-align: right;"> <?php echo $row['rate']; ?> </td>
<td style="text-align: right;"> <?php echo number_format($row['qamount'],2); ?> </td>
<td style="text-align: right;"> <?php echo number_format($row['payamount'],2); ?> </td>
</tr>
<?php }
?>
```

Coding for Change Password

```
<?php
session_start();
include "config.php";
if (isset($_POST['change'])) {
$uid=$_SESSION['cid'];
$opwd=$_POST['opwd'];
$npwd=$_POST['npwd'];
$cpwd=$_POST['cpwd'];
$qry=mysqli_query($con,"SELECT * from constructor where cid='$uid' and password='$opwd'") or die(mysqli_error($con));
$rows=mysqli_num_rows($qry);
if($rows==1)
{
if($npwd==$cpwd)
{
$sql=mysqli_query($con,"UPDATE constructor set password='$npwd' where cid='$uid'") or die(mysqli_error($con));
if($sql)
```

```
{
echo "<script>alert('Password is successfully changed');
window.location='constructorhome.php';</script>";
}
else{
echo "<script>alert('Password couldn't be changed'); </script>";
}
}
else{
echo "<script>alert('Password mismatch'); </script>";
}
}
}
?>
```

Coding for Logout

```
<?php
session_start();
session_destroy();
header("location:../index.php");
?>
```

CHAPTER: 7

TESTING

7.TESTING

7.1 Introduction:

Software testing is an investigation conducted to provide stack holders with information about the quality of the product or service under test. Testing has been defined as the process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item. Software testing is the process used to assess the quality of computer software.

It involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions should include both normal and abnormal conditions. Testing should intentionally attempt to make things go wrong to determine if things happen when they should. It is oriented to 'detection'.

Software testing has three main purposes:

- The verification process confirms that the software meet its technical specifications. A "specification" is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
- The validation process confirms that the software meets the business requirements.
- A defect is a variance between the expected and actual result. The defect's ultimate source may be traced to a fault introduced in the specification, design, or development phases. Not all the defects will necessarily result in failures.

There are two types of software testing:

- Black box testing-internal system design is not considered in this type of testing. test are based on requirements and functionality.
- White box testing-this testing is based on knowledge of thaw internal logic of an application's code. Also known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing document, which consists of event, action, input, output, expected result and actual result. Clinically defined a test case is an input and an expected result. This can be pragmatic as 'for condition x your derived result is y'; where as other test cases described in more detail the input scenario and what results

might be expected. It can occasionally be a series of steps but one with expected results or expected outcome. A test case should also contain a place for the actual result.

White box testing is applicable at the unit, integration and system levels of the software testing process.

7.2 Testing objectives:

- Finding defects which may get created by the programmer while developing the software
- Gaining confidence in and providing information about the level of quality
- To prevent defects
- To make sure that the end results meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specification.

7.3 Testing methods:

System testing is the stage of implementation. This is to check whether system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is subject to a variety of tests: online response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

7.4 Testing steps

7.4.1 Unit Testing:

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

7.4.2 Integration Testing:

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting test to uncover errors associated within the interface.

The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

7.4.3 Validation:

At the culmination of the integration testing, Software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test begin in validation testing. Validation testing can be defined in many ways, but a simple definition is that the validation succeeds when the software functions in a manner that is expected by the customer. After validation test has been conducted, one of the three possible conditions exists.

- The function or performance characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered and a deficiency lists is created.
- Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

7.4.4 Output Testing:

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to the user needs. For the hard copy also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

7.4.5 User acceptance Testing:

User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regard to the following point:

- Input screen design.
- Output screen design.
- Online message should be guide to the user.
- Format of reports and other outputs.

7.5 Test cases

7.5.1 Admin Login form:

Sl. No	Test Condition	Expected Result	Result
1	If admin clicks on login button without entering username and password.	Please fill out this field.	Successful
2	If username is entered but password is blank in admin page.	Please fill out this field.	Successful
3	If password is entered but username is blank in admin page.	Please fill out this field.	Successful
4	If the username or password is incorrect in admin page.	Invalid username and password.	Successful
5	If the valid username and valid password is entered in admin page.	System displays main page.	Successful

7.5.2 User Login form :

Sl. No	Test Condition	Expected Result	Result
1	If User clicks on login button without entering username and password.	Please fill out this field.	Successful
2	If username is entered but password is blank.	Please fill out this field.	Successful
3	If password is entered but username is blank.	Please fill out this field.	Successful
4	If the username or password is incorrect.	Invalid username and password.	Successful
5	If the valid username and valid password is entered	System displays main page.	Successful

7.5.2.1 Registration Form:

SL NO	TEST CONDITION	EXPECTED OUTPUT	TEST RESULT
1.	When the user clicks “create an Account”	It displays the User Registration page	Successful

2.	If the name field is Invalid number and click "Register" button.	Please match the Requested format only alphabet.	Successful
3.	If the user enters Invalid Password and click "Register" button	Please match the Requested format Six or more digits	Successful
4.	If the email field is Invalid and click "Register" button	Please include an '@' in the email address.	Successful
5.	If the company name field is blank and click "Register" button	Please fill out this field	
6.	If the address field is empty.	Please fill out this field	Successful
7.	If you enter Invalid Phone number with Characters.	Please enter 10 digits contact number	Successful

7.5.3 Constructor Login form:

Sl. No	Test Condition	Expected Result	Result
1	If Constructor clicks on login button without entering username and password.	Please fill out this field.	Successful
2	If Constructor name is entered but password is blank.	Please fill out this field.	Successful
3	If password is entered but Constructor name is blank.	Please fill out this field.	Successful
4	If the Constructor name or password is incorrect.	Invalid username and password.	Successful
5	If the valid Constructor name and valid password is entered	System displays main page.	Successful

7.5.3.1 Registration Form:

SL NO	TEST CONDITION	EXPECTED OUTPUT	TEST RESULT
1.	When the Constructor clicks “create an Account”	It displays the Constructor Registration page	Successful
2.	If the name field is Invalid and click “Register” button.	Please match the Requested format only alphabet.	Successful
3.	If the Constructor enters Invalid Password and click “Register” button	Please match the Requested format Six or more characters.	Successful
4.	If the email field is Invalid and click “Register” button	Please include an ‘@’ in the email address.	Successful
5.	If the address field is empty.	Please fill out this field	Successful
6.	If you enter Invalid Phone number with Characters.	Please enter 10 digits contact number	Successful

7.5.4 Main Form:

SL NO	TEST CONDITION	EXPECTED OUTPUT	TEST RESULT
1.	When the user clicks the “Home” in the menu.	It displays the Home page	Successful
2.	When the user clicks the “About us” in the menu.	It displays the About Us page	Successful
3.	When the user clicks the “Our Service” in the menu.	It displays the Our Service page	Successful

4.	When the user clicks the "Gallery" in the menu.	It displays the Gallery page	Successful
5.	When the user clicks the "Accounts" in the menu.	It displays the Accounts page	Successful

7.5.5 Implementation:

Implementation is the stage of the Project when the theoretical is turned into a working system. At this stage workload and the latest upheaval shifts to the user department. If the Implementation stage is not clearly planned and controlled it can cause choice. The term Implementation has different meaning, ranging from the conversion of the basic application to a compatible replacement of a computer system. Implementation is used here to mean the process converting a new or a revised

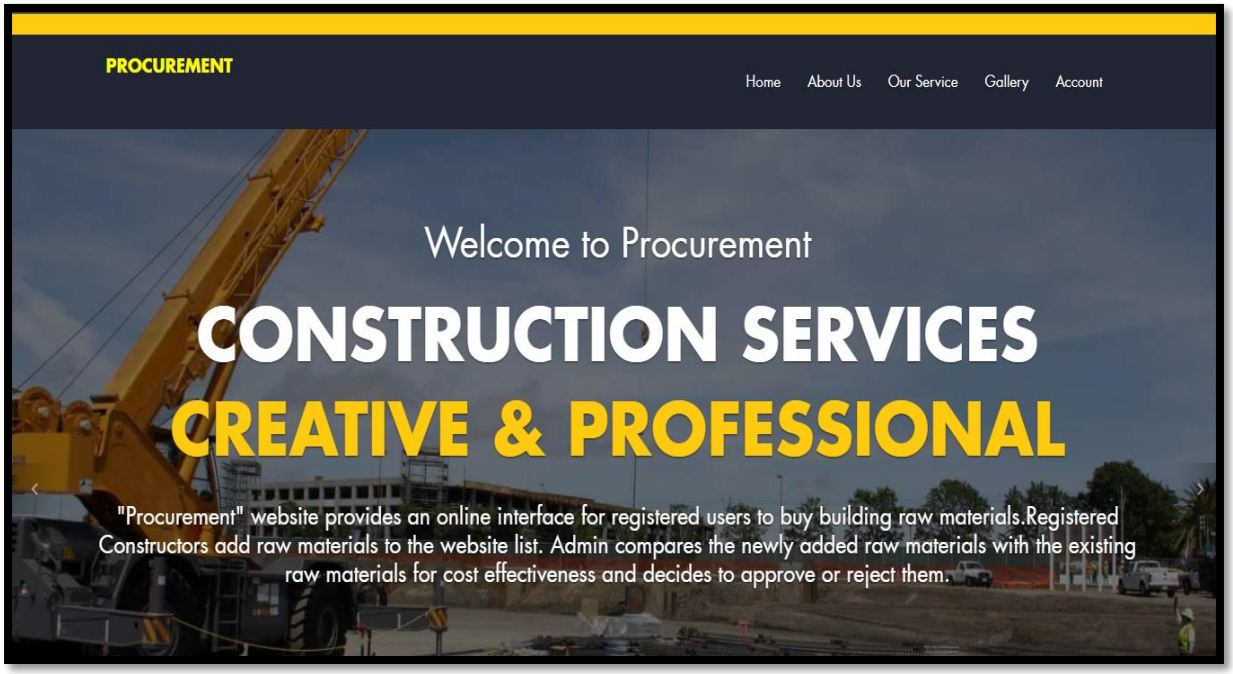
System design into an operational one. During the implementation stage we convert the detail code in a programming language.

CHAPTER: 8

USER INTERFACE

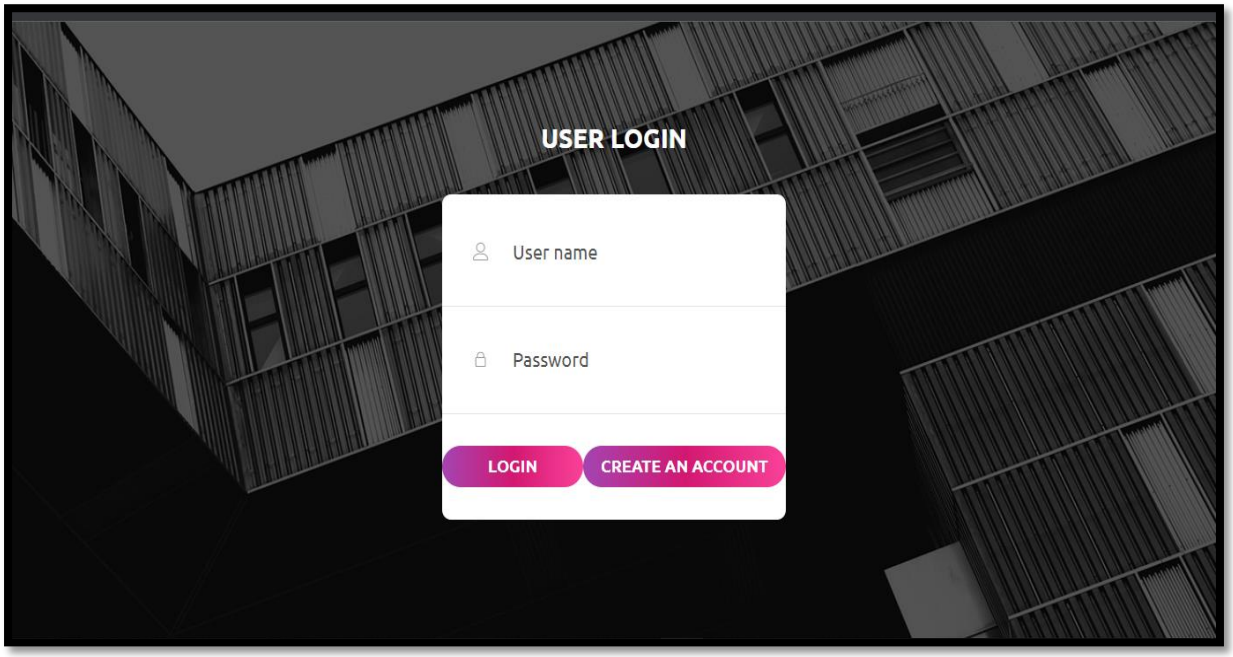
8.USER INTERFACE

Home Page:

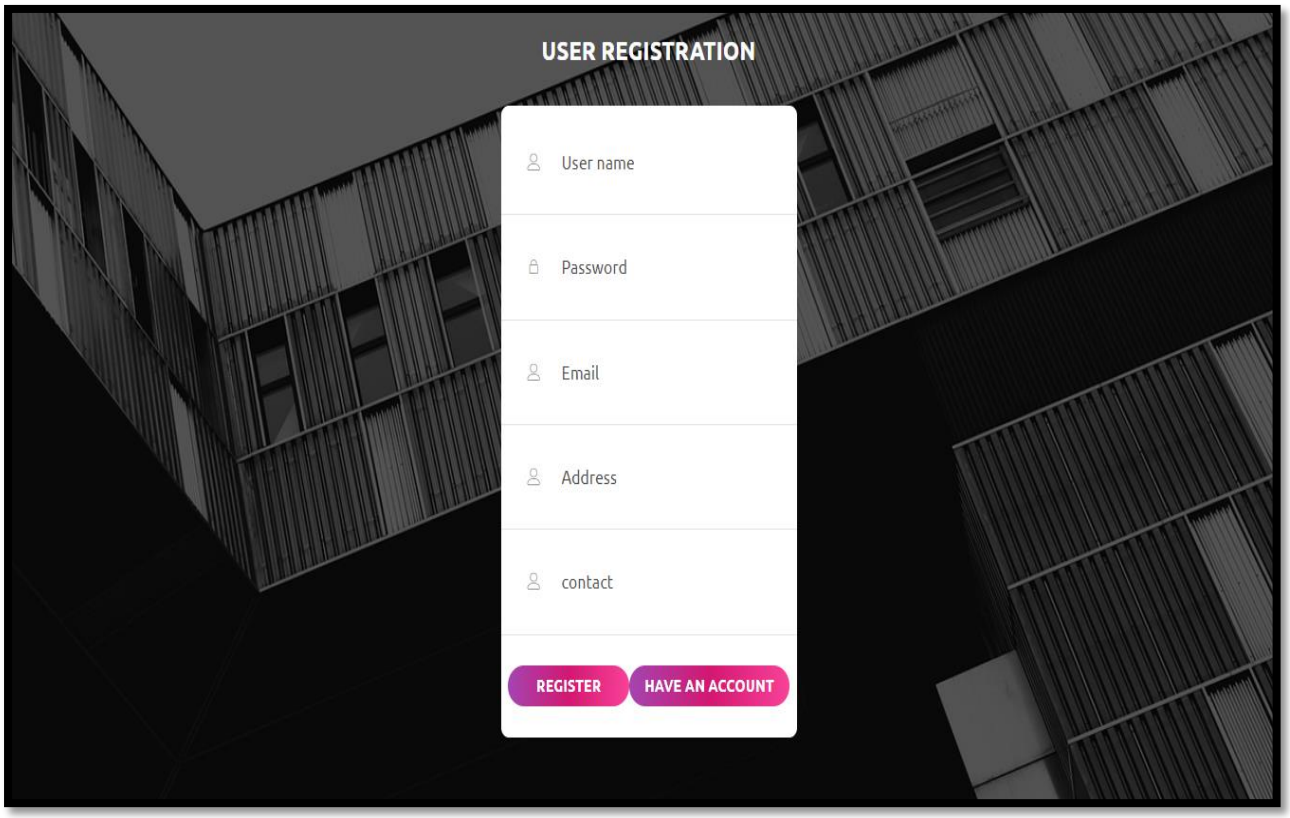


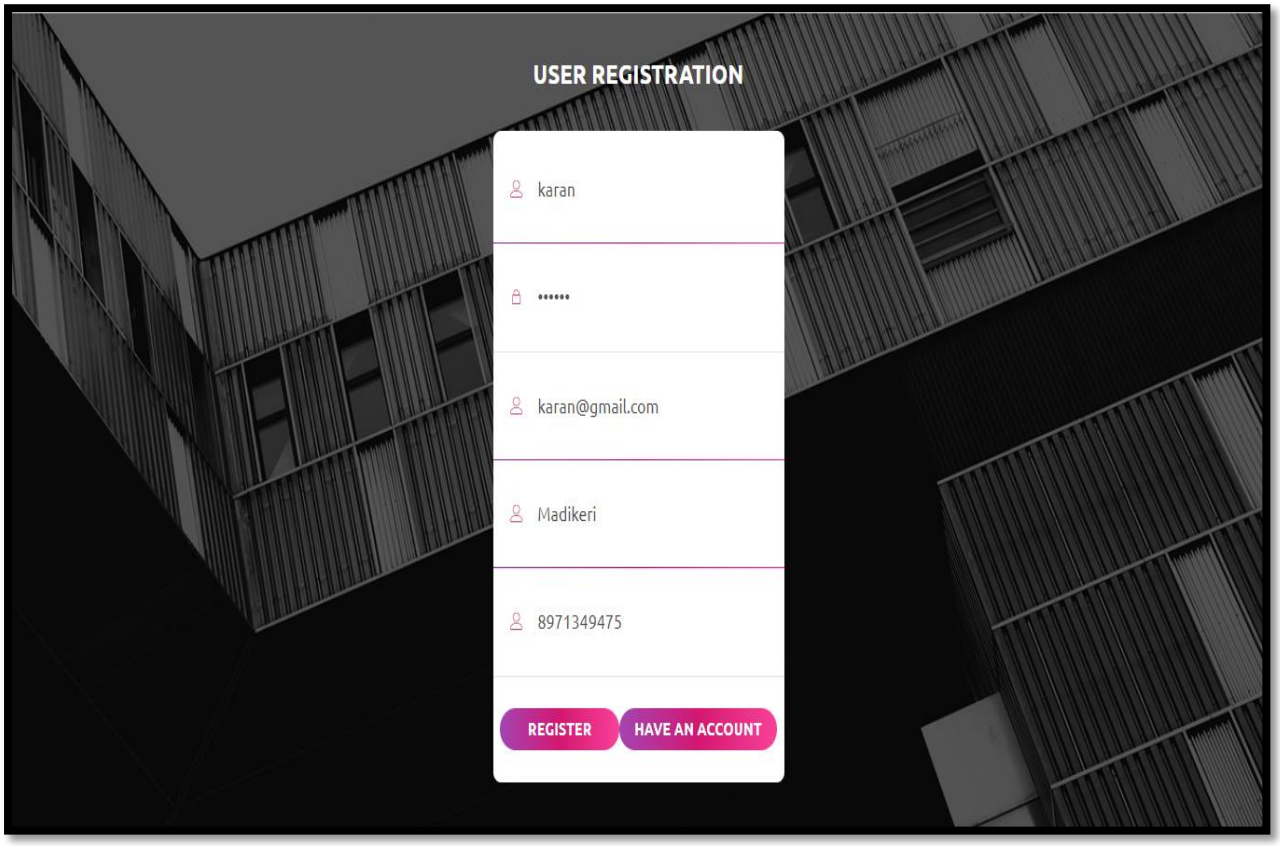
About Page:

User Login Page:

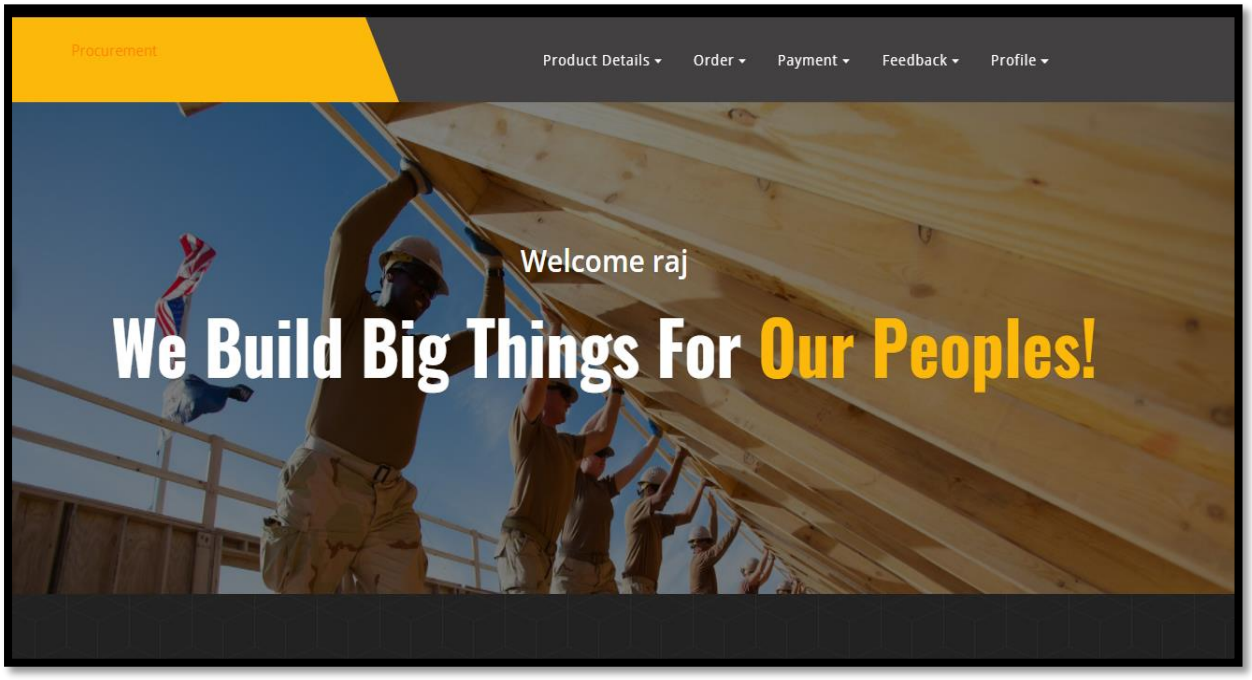


User Registration Page:

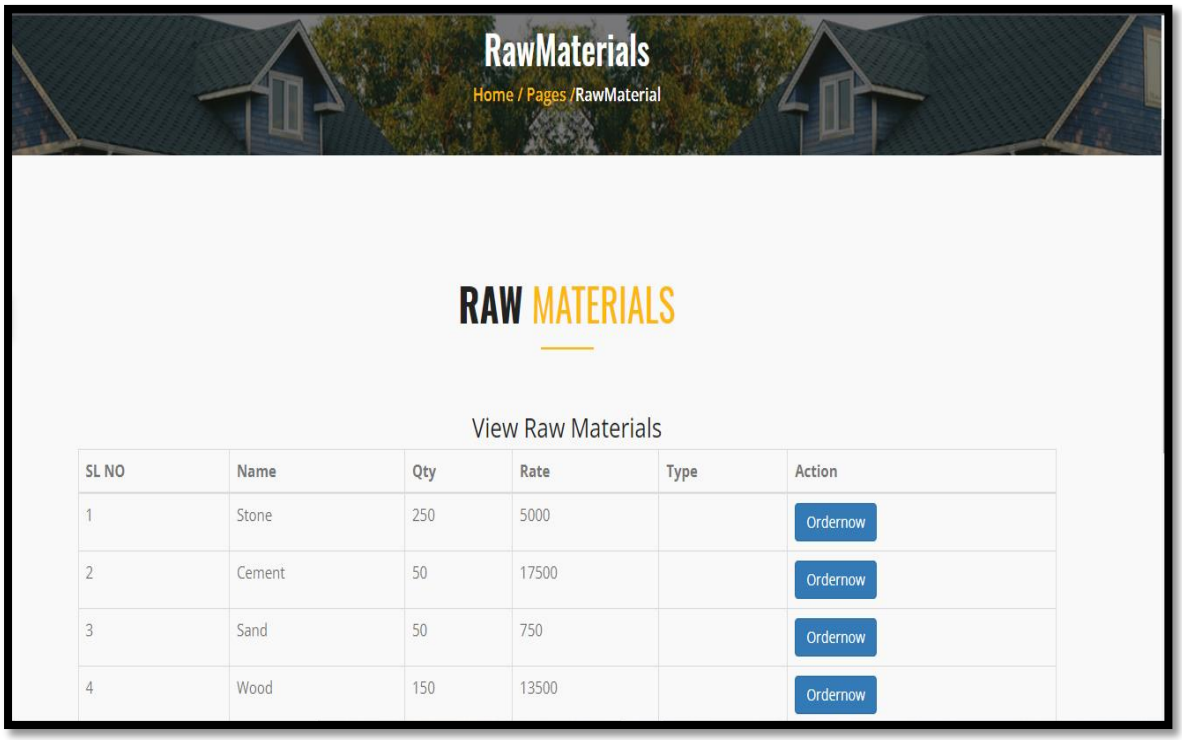




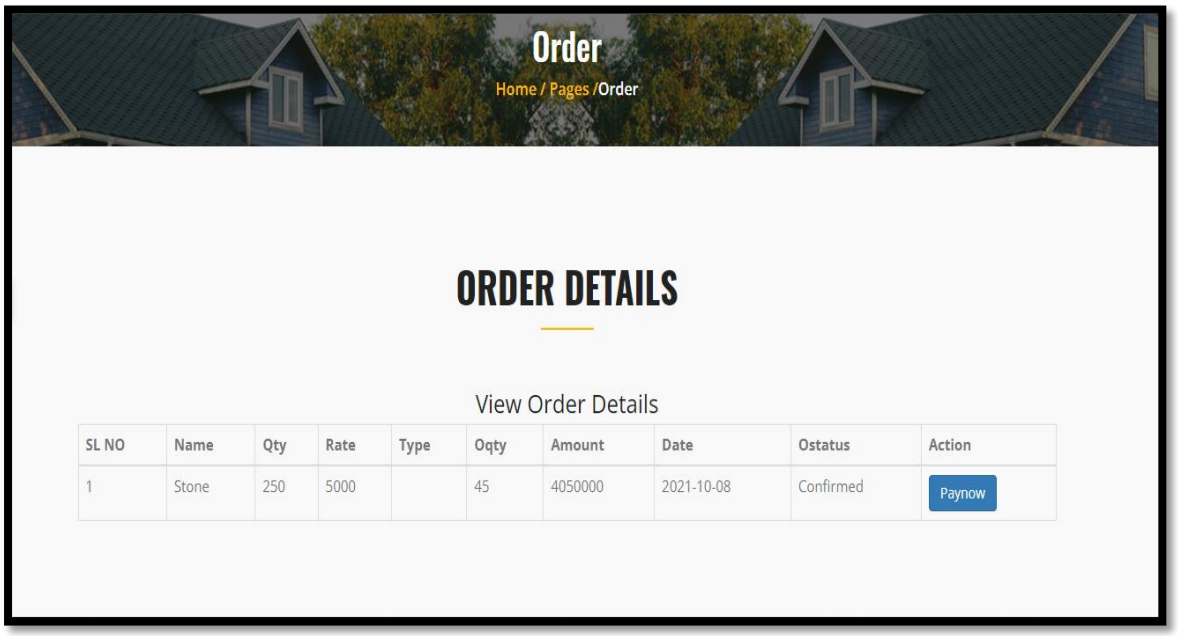
User Home Page:



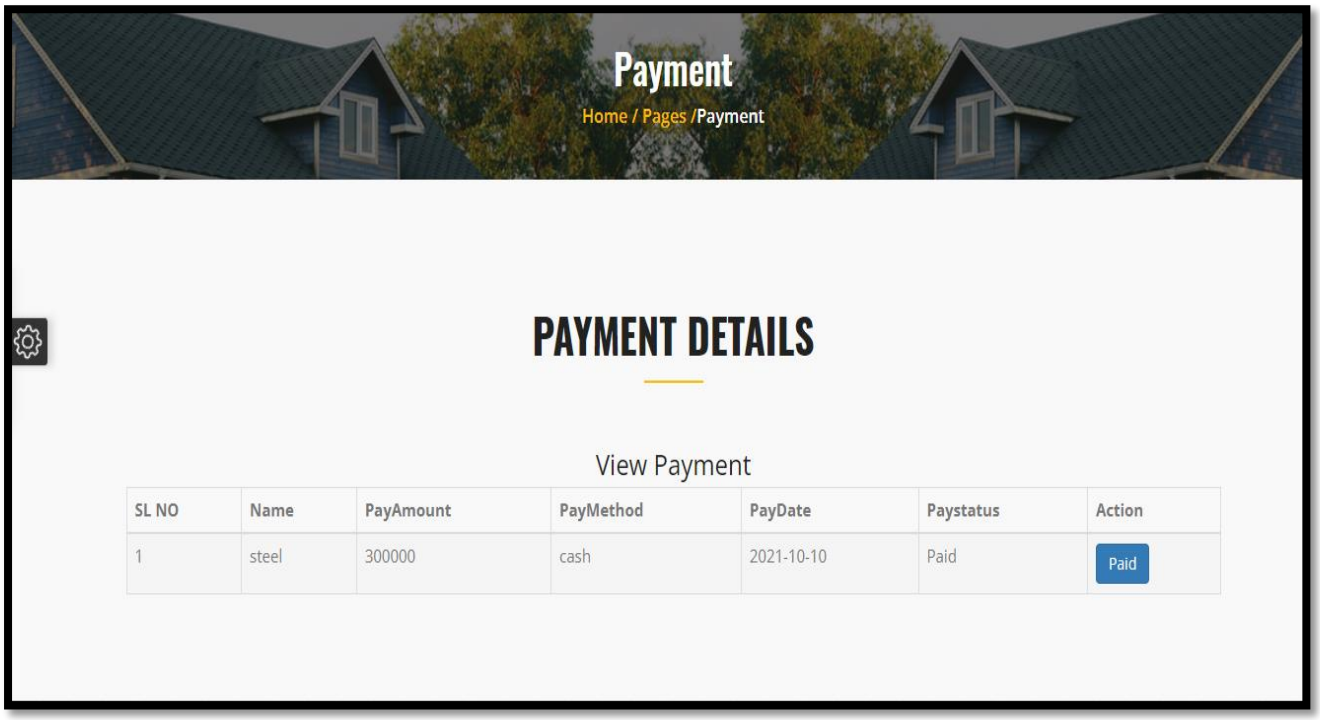
View raw material:



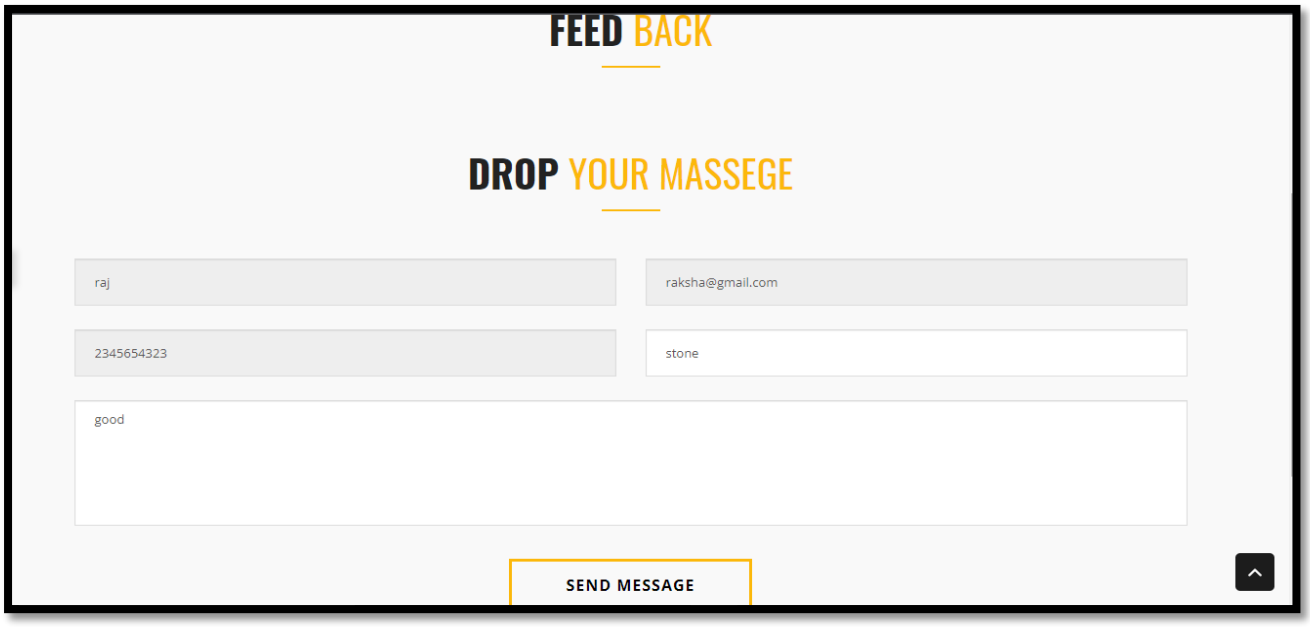
Order Details:



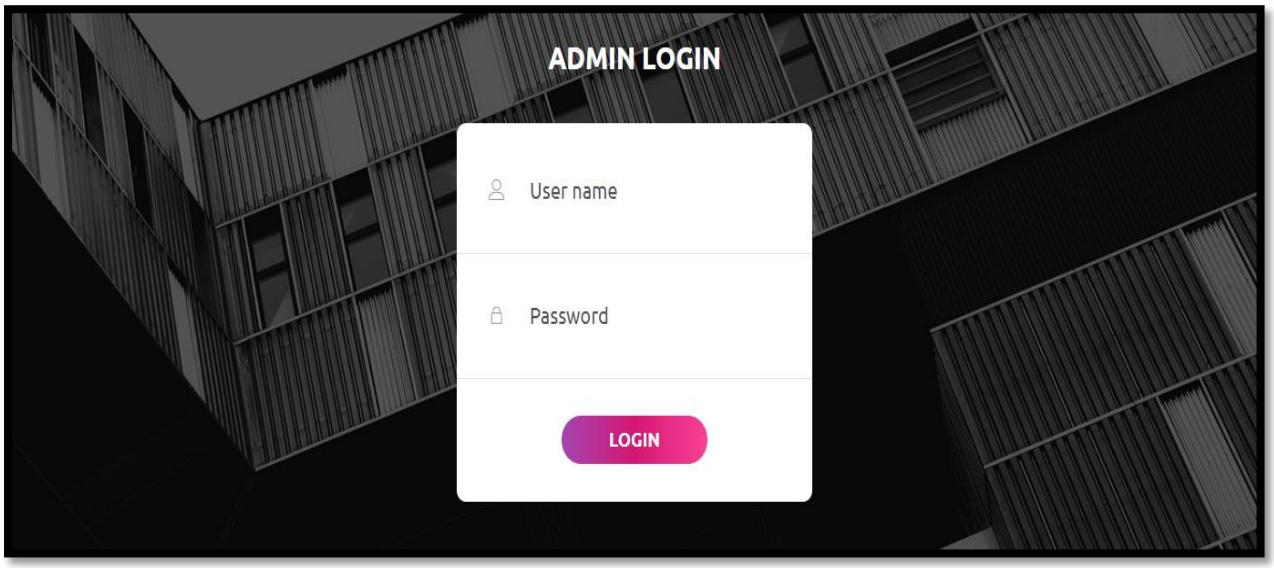
View Payment Page:



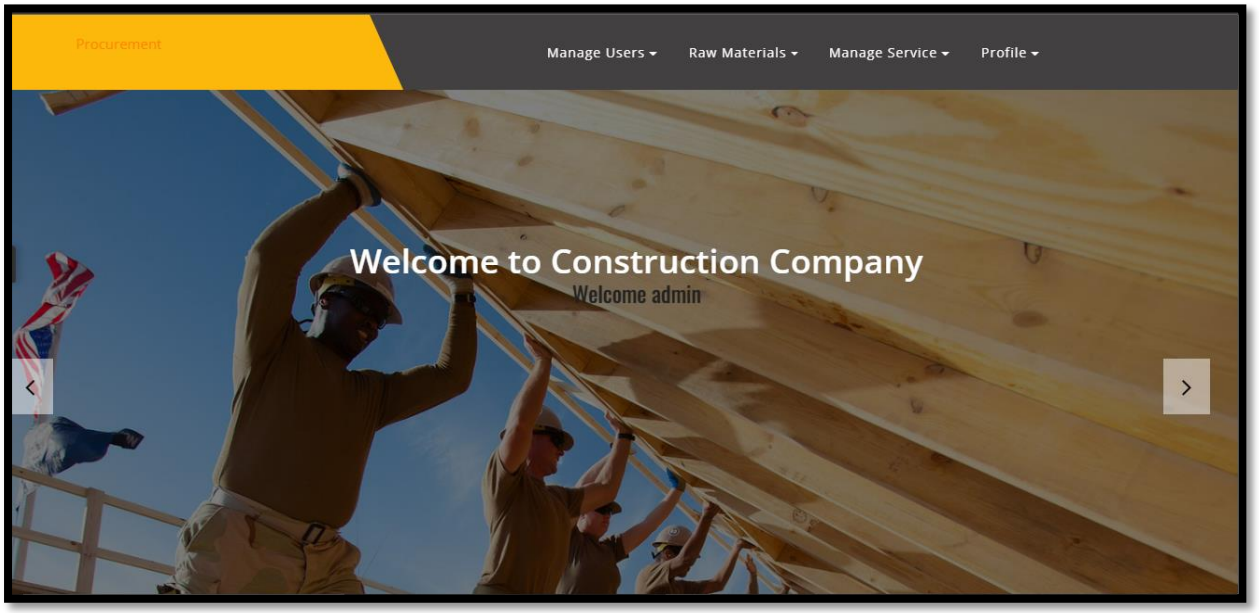
View Feedback :



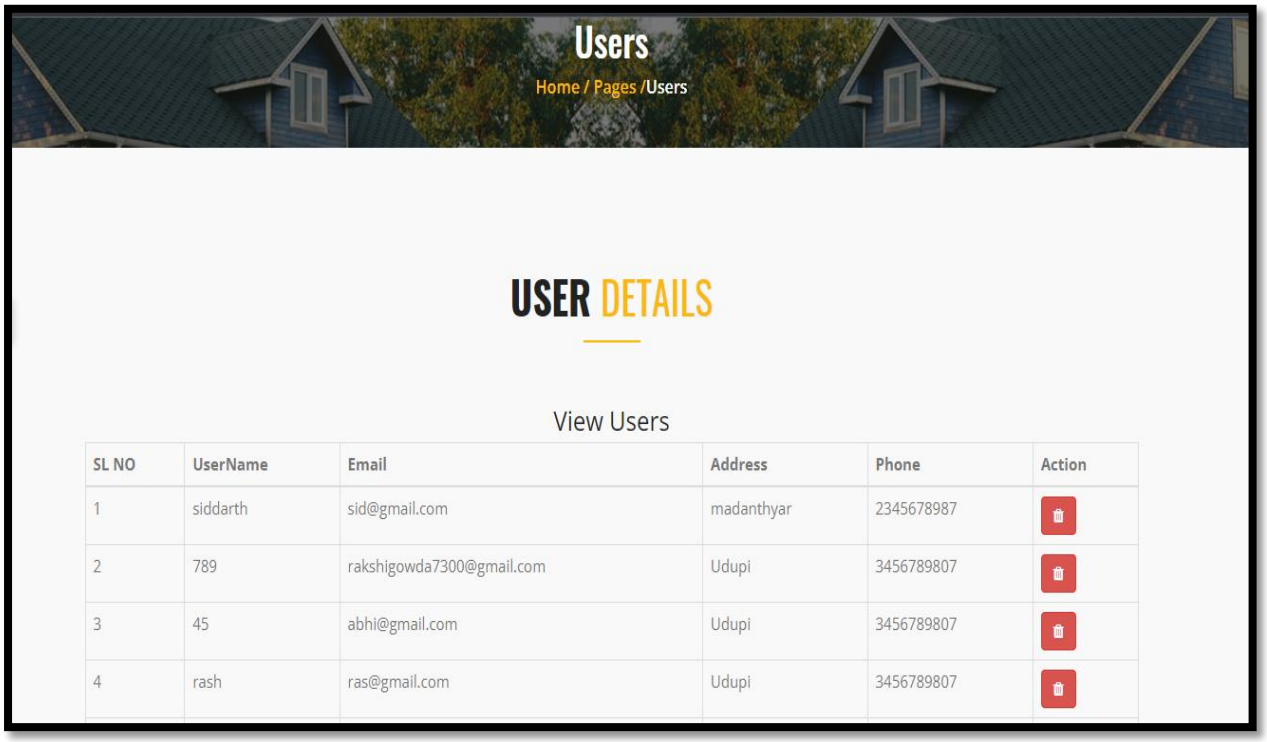
Admin login page:



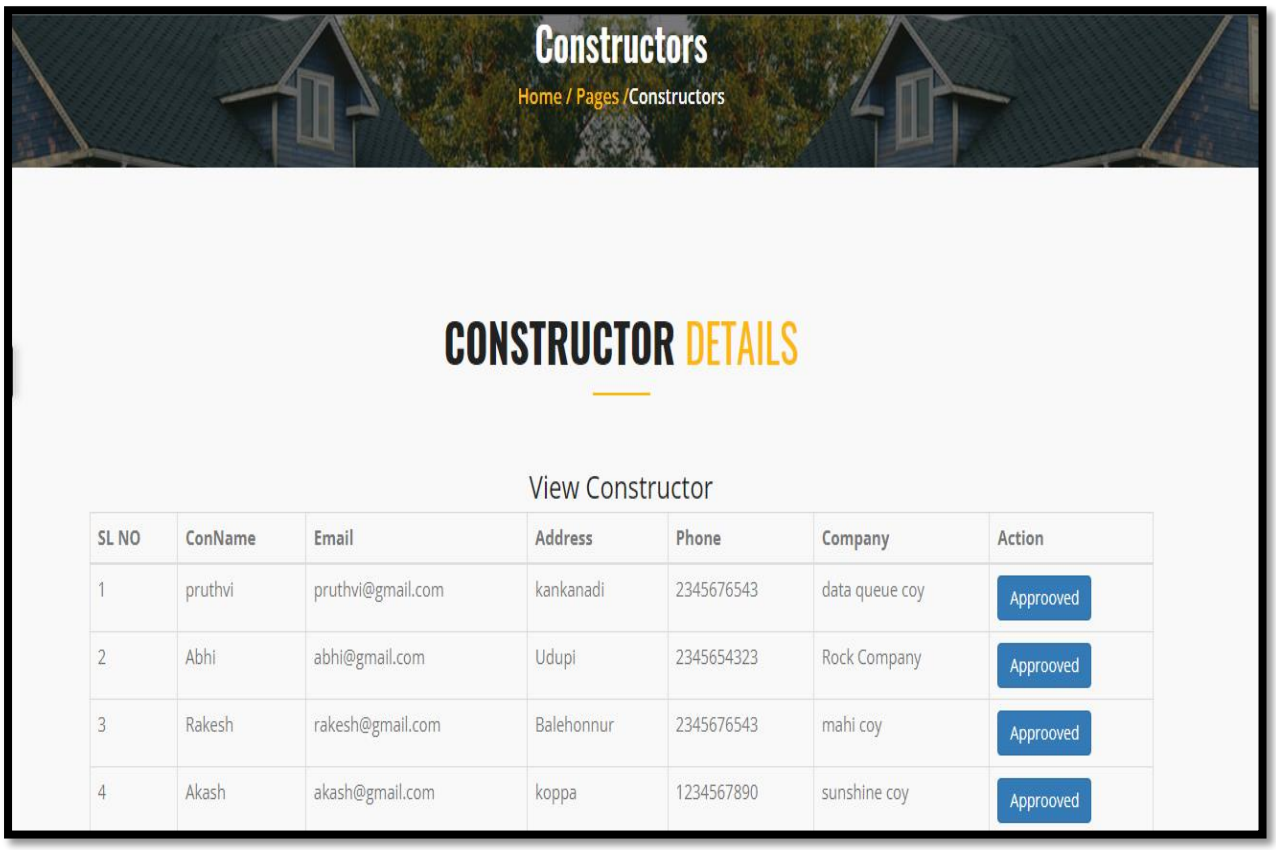
Admin Home Page:



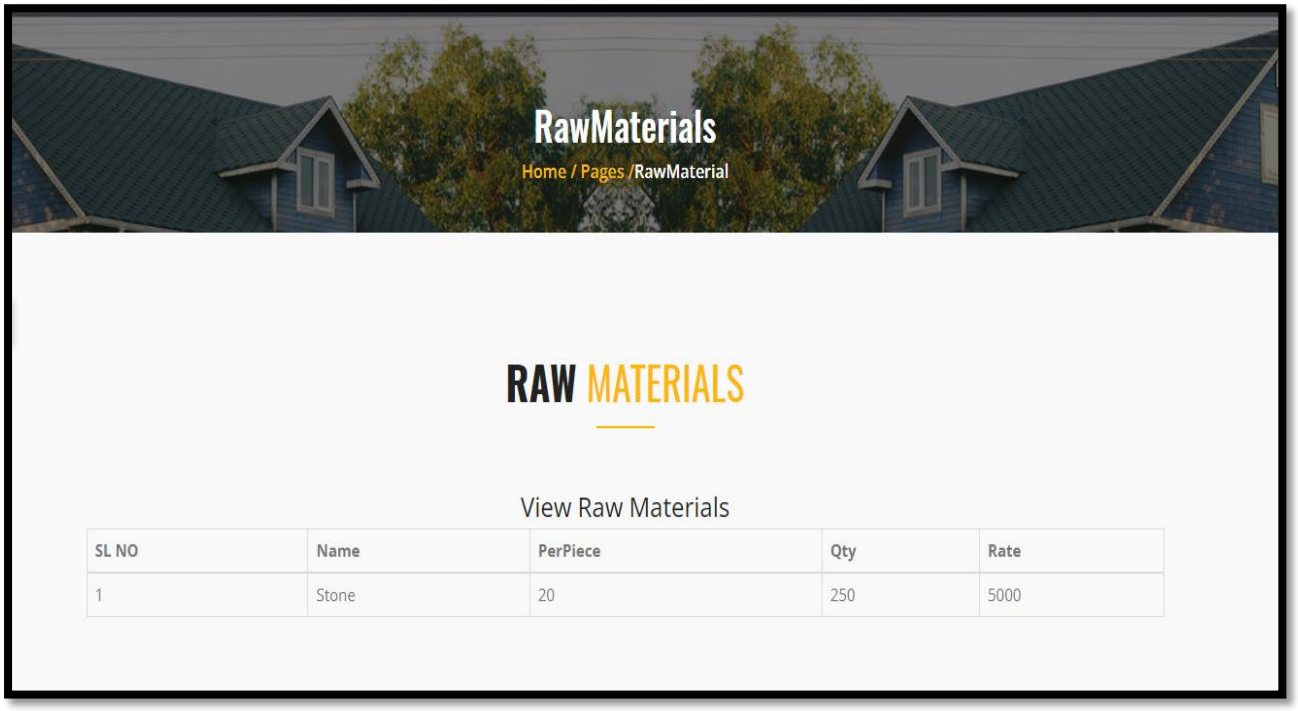
View User Page:



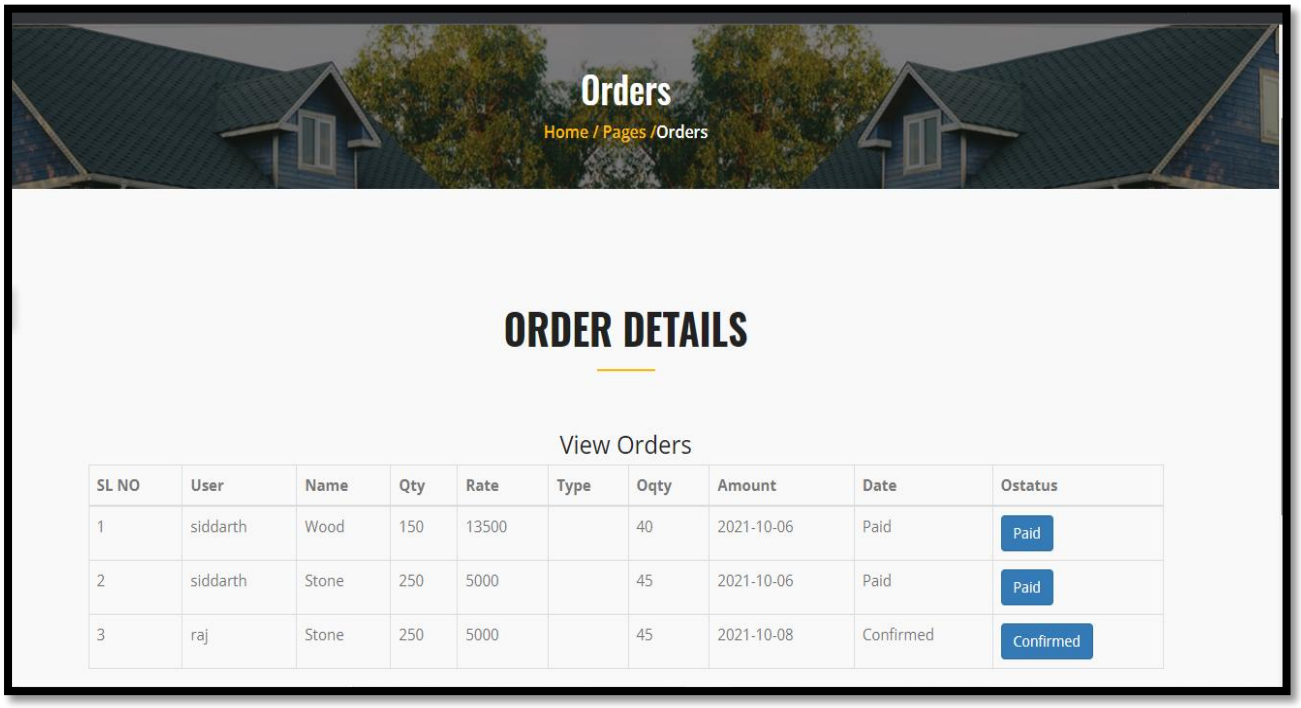
View Constructor:



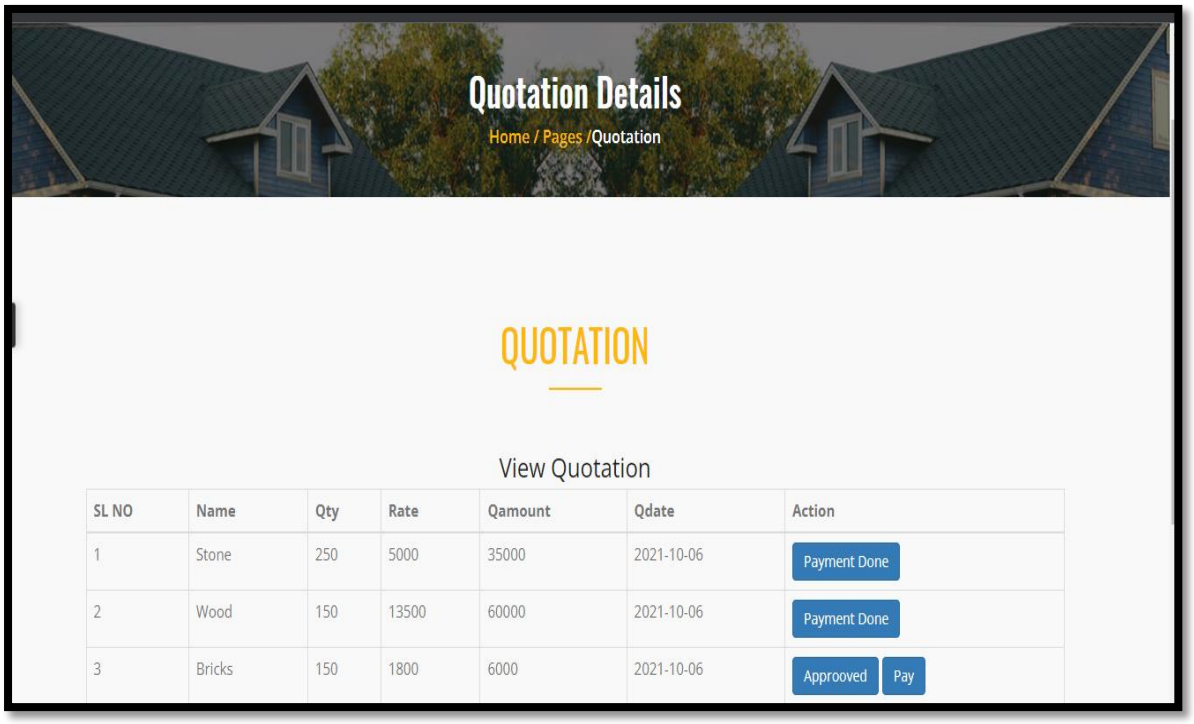
View Raw Material:



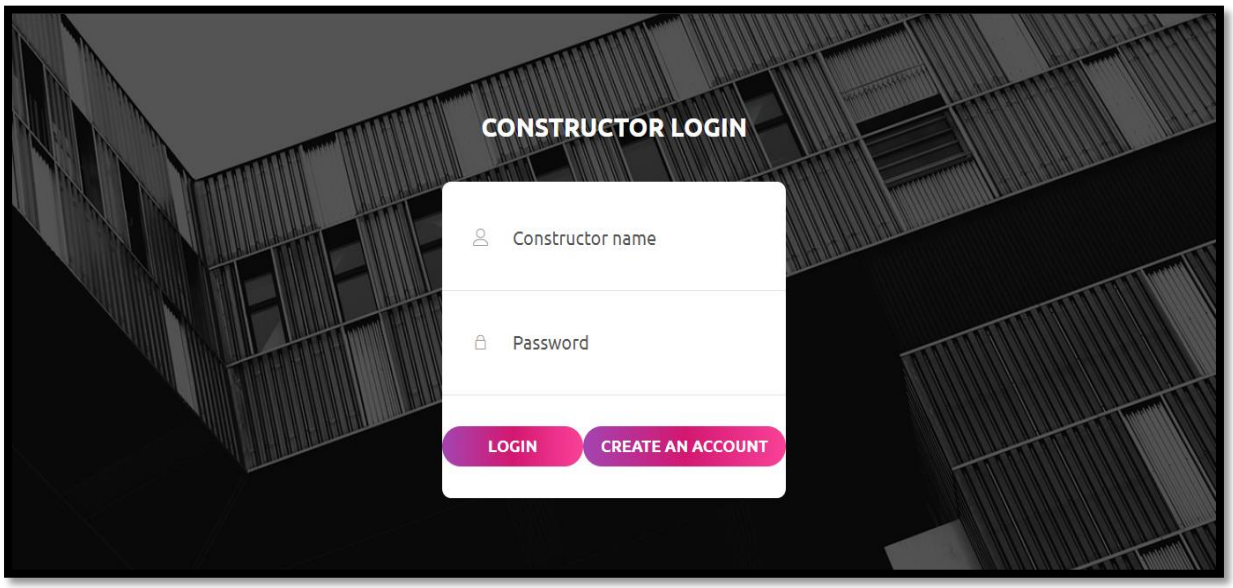
View Order details:



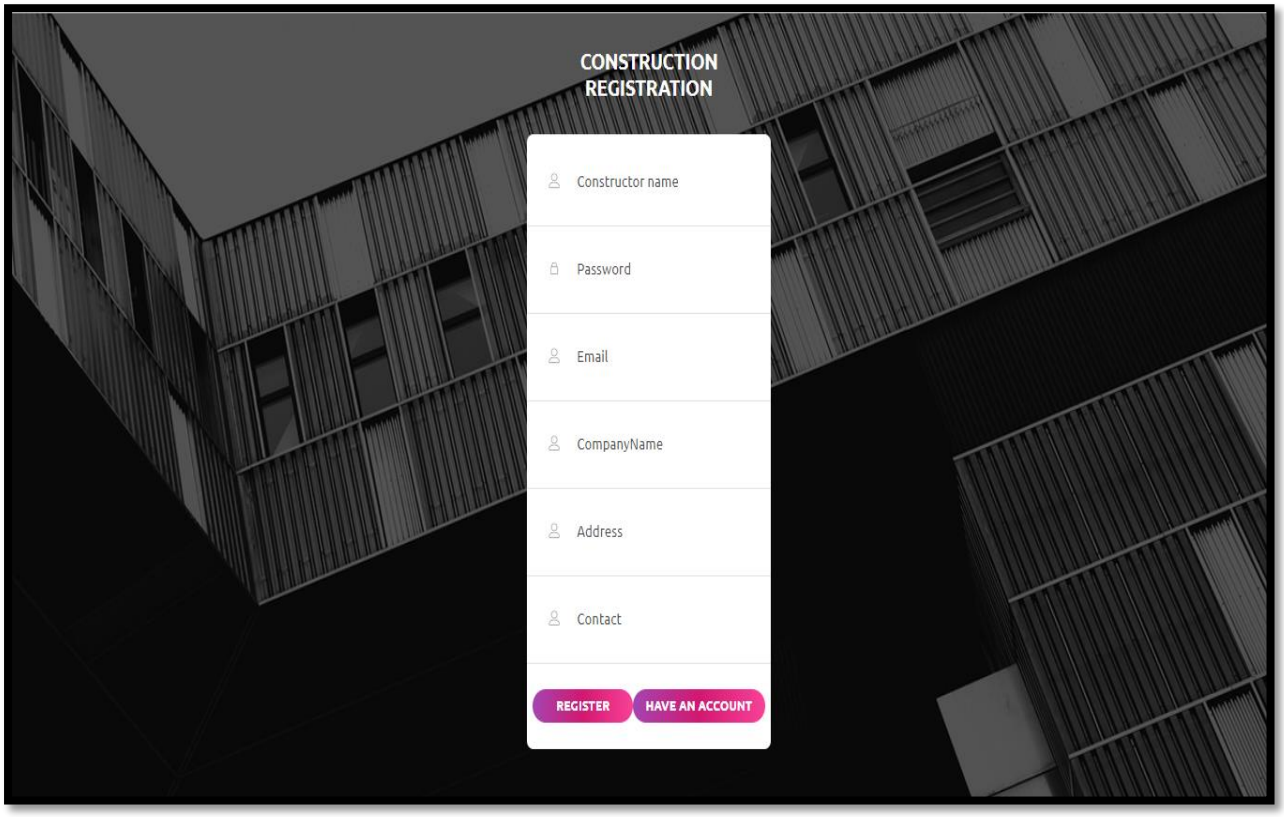
View Quotation details:

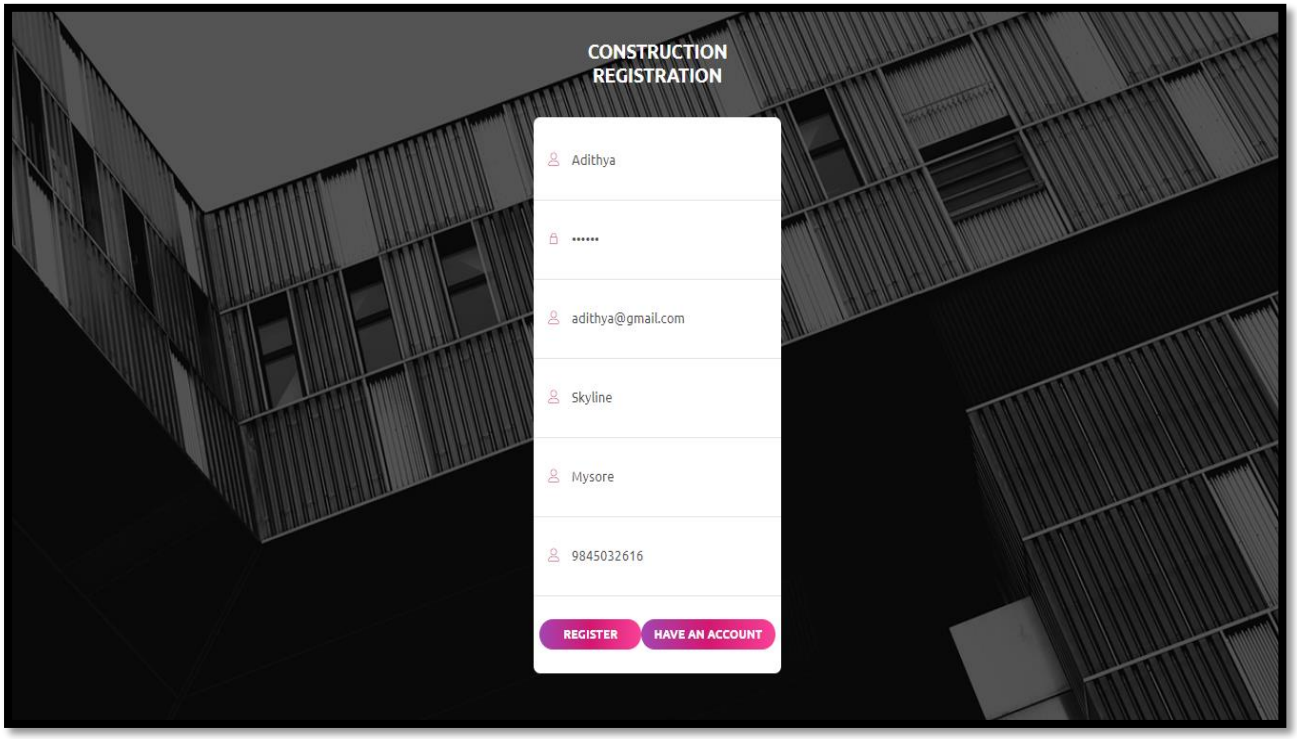


Constructor login Page:

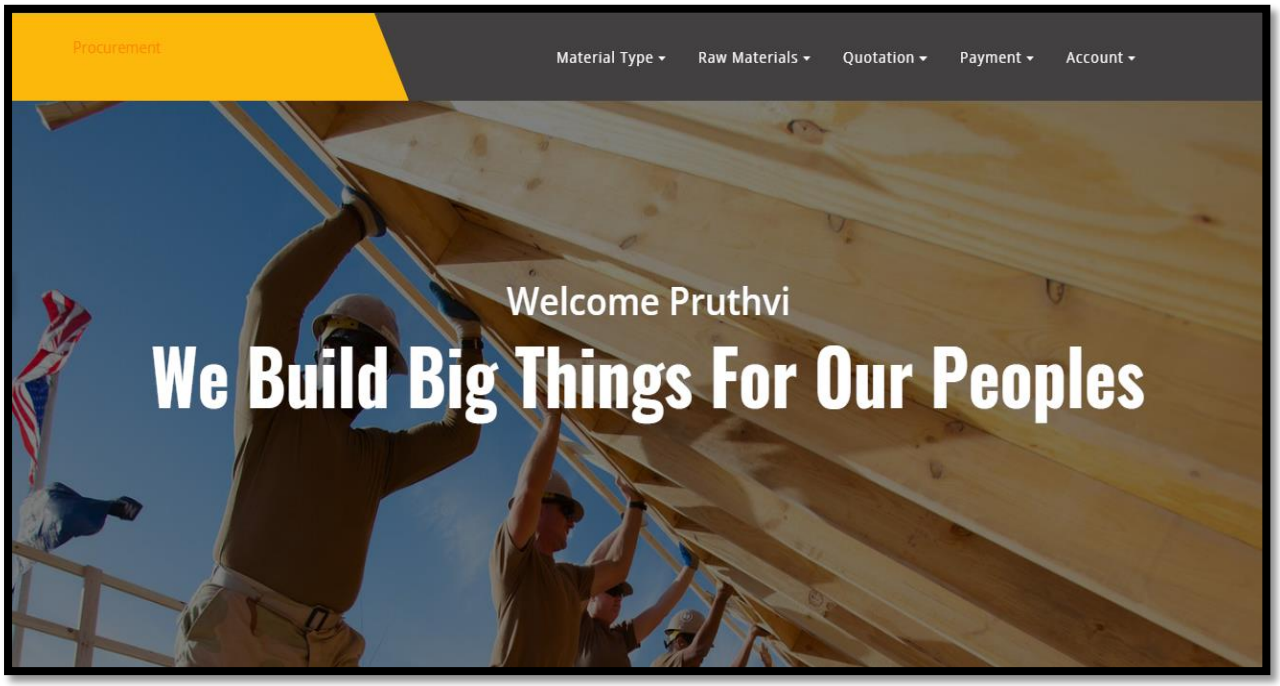


Constructor Registration form:

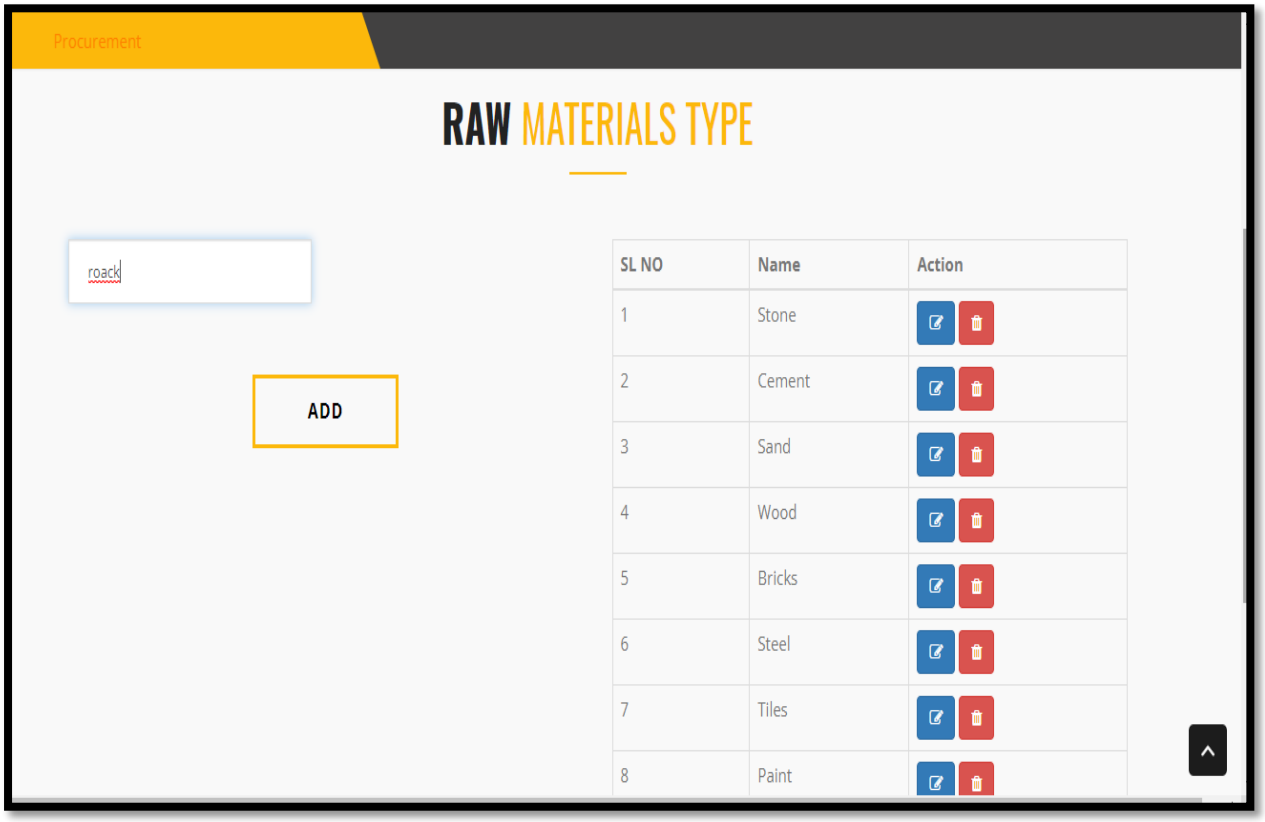




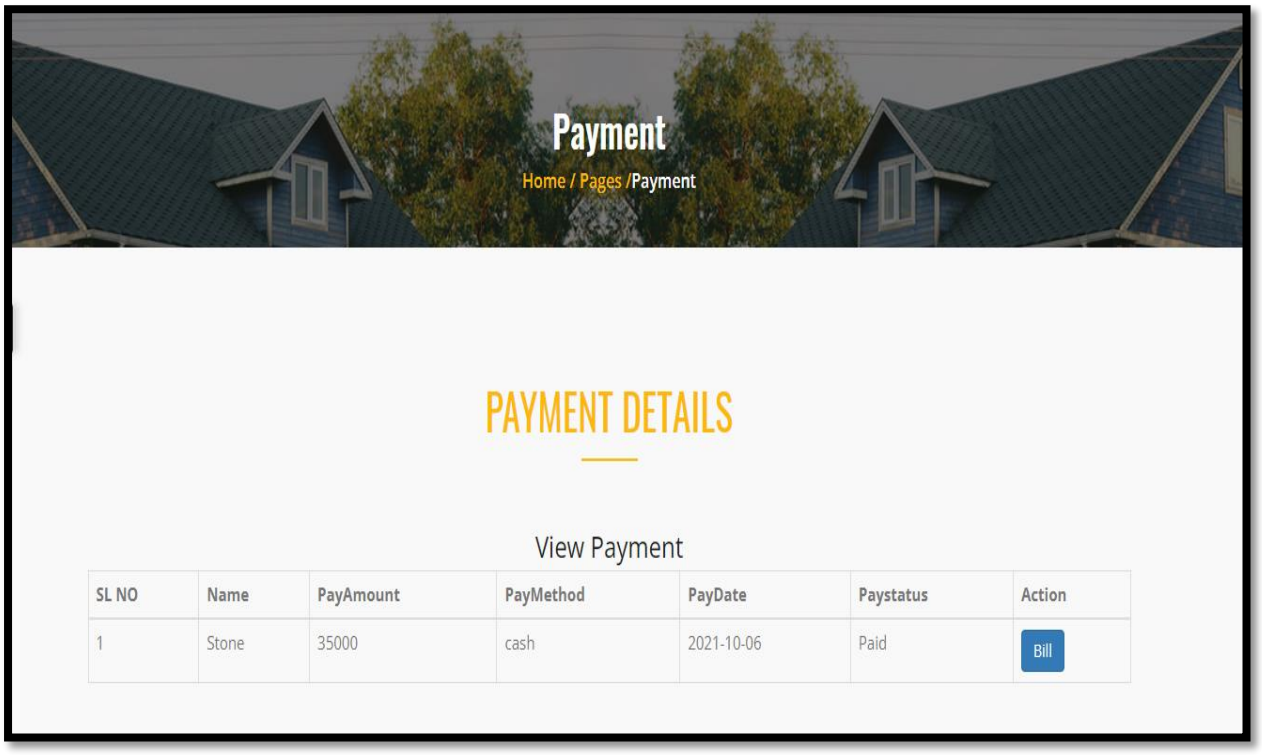
Constructor Home Page:



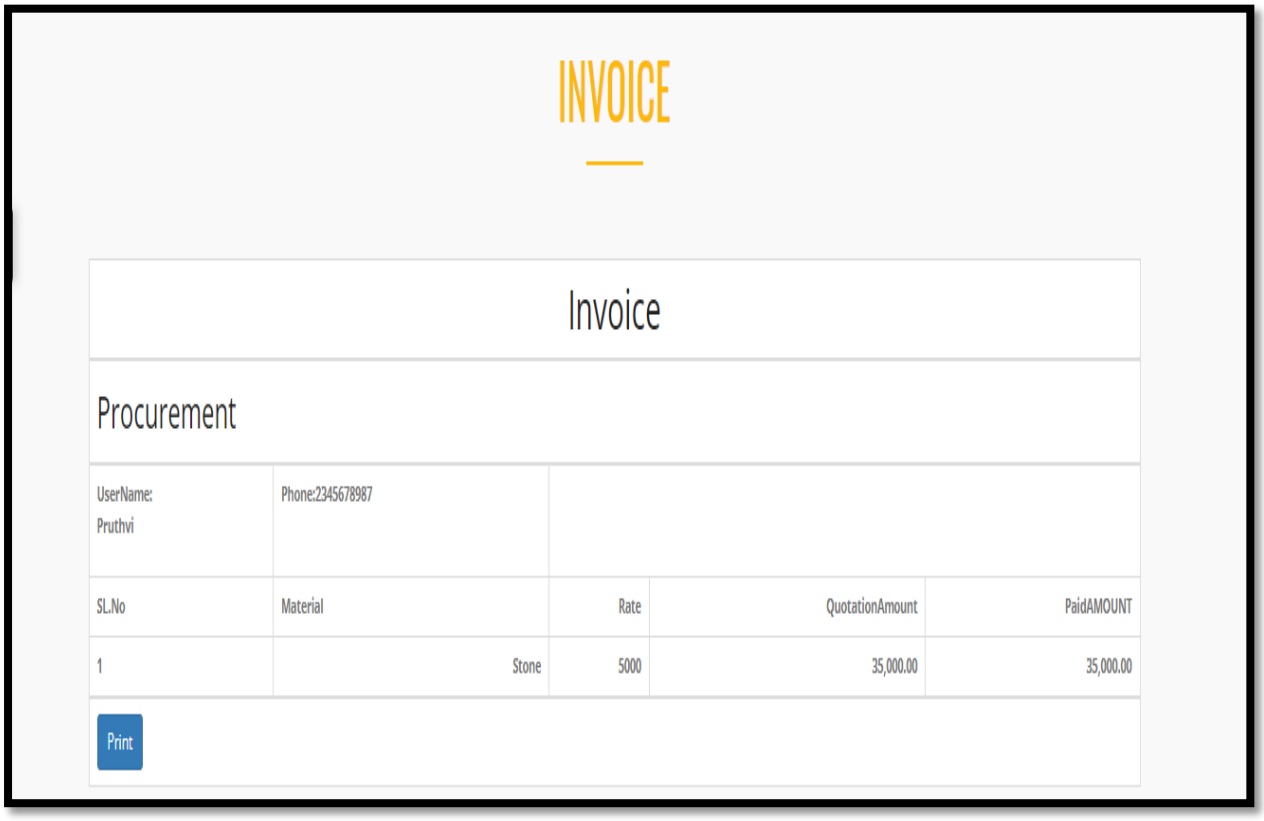
Add Raw Material Type Page:



View Payment Page:



View Bill Details:



CHAPTER: 9

LIMITATIONS

9. LIMITATIONS

- The system should be connected to the Internet
- User should have basic Knowledge in using the website
- User has only retrieving option.
- Xamp server should be installed
- PHPMyAdmin server need to be installed to store the database.
- Browser is must to run the application

CHAPTER: 10

FUTURE SCOPE

10. FUTURE SCOPE

- Develop an android application for the website.
- Provide chat service between user and constructor.
- Subscribe to our Website.
- In future we can add features such as filling applications through online.

CHAPTER: 11

BIBLIYOGRAPHY

11. BIBLIOGRAPHY

- **Books:**

  - Pankaj Jalote, An integrated approach to software engineering, ed.3, Publication: Narosa Publishing House.
  - Nixon (Robin) Learning PHP MYADMIN, CSS\Bootstrap\Javascript. Publication:Shroff publisher and Distributors Pvt Pankaj Jalote Ltd.

- **Websites**:

  - www.google.com
  - www.w3schools.com
  - www.tutorialspoint.com