TRIBHUVAN UNIVERSITY

FACULTY OF HUMANITIES AND SOCIAL SCIENCE

A Project Proposal

On

"Record Inventory: A Inventory Record Platform"

Submitted to

Department of Computer Application

National College of Computer Studies

In partial fulfillment of the requirements for Bachelor's Degree in Computer Application

Submitted By:

Rohan Maharjan(NCCS BCA 070)

Under the Supervision of

Chhetra Bahadur Chhetri

Tribhuvan University

Faculty of Humanities and Social Science

National College of Computer Studies

Supervisor's Recommendation

I hereby recommend that this project prepared under my supervision by **Rohan Maharjan** entitled **Record Inventory "A Web-based Inventory Record Platform"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

....

SIGNATURE

Chhetra Bahadur Chhetri

SUPERVISOR

Faculty Member
Department of Computer Application

National College of Computer Studies

Tribhuvan University
Faculty of Humanities and Social Sciences

National College of Computer Studies

LETTER OF APPROVAL

This is to certify that this project prepared by **Rohan Maharjan** entitled Record Inventory "A Web-based Inventory Record System" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Signature of Supervisor Chhetra Bahadur Chhetri Faculty Member Department of Computer Application National College of Computer Studies Paknajol, Kathmandu	**Signature of HOD/ Coordinator** Rajan Poudel Faculty Member Department of Computer Application National College of Computer Studies Paknajol, Kathmandu
Signature of Internal Examiner	**Signature of External Examiner**

Tribhuvan University
Faculty of Humanities and Social Sciences

National College of Computer Studies

LETTER OF APPROVAL

This is to certify that this project prepared by **Rohan Maharjan** entitled Record Inventory "A Web-based Inventory Record System" in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion, it is satisfactory in the scope and quality as a project for the required degree.

Signature of Supervisor	**Signature of HOD/ Coordinator**
Chhetra Bahadur Chhetri Faculty Member Department of Computer Application National College of Computer Studies Paknajol, Kathmandu	Rajan Poudel Faculty Member Department of Computer Application National College of Computer Studies Paknajol, Kathmandu
Signature of Internal Examiner	**Signature of External Examiner**

ABSTRACT

"Record Inventory is a user-friendly web-based system designed to help small businesses record their inventory items. The main goal of this project is to make inventory control. The main advantage of this tool is, it is free to use and open source; so it is customizable and easy to use. The main objective of Record Inventory is to provide a tool for recording data, a user-friendly interface that is easy to learn and doesn't cost any money. This tool can be used for any type of small businesses for recording their items on a database, saving a lot of time for the user.

Record Inventory "An online inventory record platform" is an effective and user-friendly site designed platform that offers small businesses to record their product details for keeping track of record. Users can view detailed information about each item in their inventory, such as product name, quantities, and price. It is an open source platform that provides a comprehensive solution to keep record for small businesses for satisfying and solving their small business problems.

Keywords: Record inventory; web application; record system; open source; secure; user-friendly.

ACKNOWLEDGEMENT

This study's completion in its present state has been a wonderful and fulfilling experience. I want to sincerely thank everyone who has helped me with this project and for their support.

I want to start by sincerely thanking the BCA department for their direction, oversight, and tremendous assistance in getting this job done. Their suggestions and support were really helpful in informing this project.

Finally, I'd like to thank the people who will hopefully get something from reading this report.

Rohan Maharjan

Table of content

List of Abbreviation

BCA	Bachelors' in Computer Applications
CASE	Computer-Aided Software Engineering
CSS	Cascading Style Sheets
HOD	Head of Department
HTML	Hyper Text Markup Language
MySQL	My Structured Query Language
PHP	Hypertext Preprocessor

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

An inventory record system is a tool for keeping transactional records. This system enables small and mid-sized businesses to effectively record their inventory, ensuring stock levels. With an inventory record system, small businesses can accurately add stocks to their inventory. The system provides detailed insights into products. Additionally, an inventory record system can integrate with other business systems like point-of-sale (POS) systems and accounting software, providing a comprehensive view of the business's financial health. Overall, implementing an inventory record system empowers small businesses to manage it arithmetically.

1.2 Problem Statement

Many small businesses struggle with manual processes, leading to a host of challenges and setbacks. These businesses often face difficulties in accurately tracking their inventory, resulting in stockouts, overstocking, and increased costs. Small businesses urgently require an inventory record system that can streamline their inventory processes, improve recording data, and provide actionable insights for recording the data and allocation.

1.3 Objectives

The main objective of Inventory Record is:

- To develop a simple way to keep data records for small B2B businesses.
- To make it an open source platform for learning and creating helpful products.

1.4 Scope of Project

The study aims to develop and implement a web-based platform for storing records while minimizing system complexity and problems. The system is intended for small and mid-sized businesses or startups for keeping record of their products.

1.5 Limitations

This project has a few limitations which have been summarized in the points below:

- Errors from Humans.
- Integration limitations.
- Technical issues and downtime
- Cost for initializing
- Unavailability of real-time updates
- Inability to track demand

1.6 Report Organization

Introduction

Chapter one introduces the concept of this project in brief. This chapter gives a general overview of the project, outlining its goals, scope, constraints, and development approach in addition to the problem that it seeks to work on.

Background study and literature review

This chapter focuses on the basic ideology of how this project will be built. It traces the study of different platforms and their workings.

System analysis and design

In this chapter, the requirements gathering, feasibility study, and designing of the project are described. It includes diagrams, functionality analysis, requirements gathering technique, and process model.

Implementation and testing

This chapter is designed to give information about how the project has been implemented, what kind of software and tools has been used, and the type of testing that the project has gone through.

Conclusion and future recommendation

This chapter includes the potential outcome of the project, and also provides recommendations for future work.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Requirement identification is the process of gathering and understanding the important things that a system needs to have. This includes figuring out the requirements by studying existing systems, talking to people through interviews, and using surveys. It's about finding out what the system should do and how it should work to meet the needs of its users.

2.1.1 Study of an existing system

The community-has already existing applications over IOS, android and web applications but they charge subscription or they aren't available in an offline environment. Inventory Now , sortly, SOS inventory, etc are some of the popular examples of inventory recor- ding applications. In the data of camcode.com , most of them are subscription based or have a limited trial period; most of them aren't available offline, either they have limited use or in app purchase. Some of them have tier editions based on free tier, pro tier or business edition, depending on these aspects a normal price can range from $25-$100 subscription.

2.1.2 Literature Review

Traditional inventory models, with a few exceptions, do not account for the existence of inventory record inaccuracy (IRI), and those that do treat IRI as random. This study explores IRI observed both within and across product categories and retail stores. Examining nearly 370,000 inventory records from 37 stores of one retailer, we find 65% to be inaccurate[2].

Most retailers suffer from substantial discrepancies between inventory quantities recorded in the system and stocks truly available to customers. Promising full inventory transparency, radio frequency identification (RFID) technology has often been suggested as a remedy to the problem[5].

Accurate inventory records are critical to the performance of retail organizations. U.S. retailers spend 1% of annual sales, approximately $30 billion per year, on automated decision support tools that use recorded inventory quantities to forecast demand, plan product assortments, and replenish store shelves (Steidtmann 1999)[2].

Rinehart (1960) is the first researcher to identify IRI as a potential obstacle to operational performance. In his work with a government supply facility, Rinehart (1960) documents substantial discrepancies between recorded and actual inventory quantities. Discrepancies exist in 0–50% of the items audited across different inventory classifications. Record inaccuracy has, he notes, a "deleterious effect" on operations because it prevents firms from achieving the benefits that result from using "theoretically optimal methods of inventory control" (p. 543)[2].

We provide evidence that the problem of record inaccuracy is substantial in magnitude in the retail con- text. And the resulting lost sales can amount to a considerable portion of firm profit[2].

Chapter 3: System Analysis and Design

3.1 System Analysis (Iterative/Incremental Model)

The iterative model has been used for the system development of this project. This model is well-suited for projects with evolving requirements and the need for continuous feedback and improvement. This model approaches initial development work is conducted based on the requirements that are clearly defined; subsequent features are added to the base software product through iterations until the final system is ready. This helps in identifying risk associated with the requirements at an early stage and mitigate them.

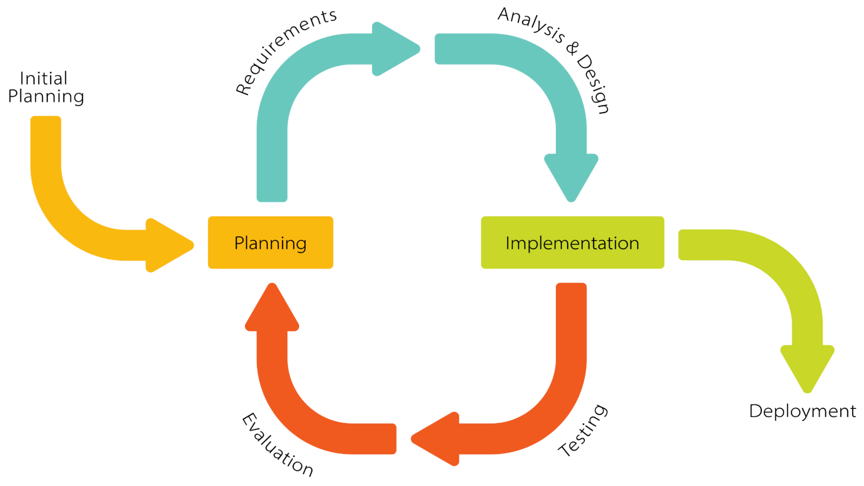


Figure 1: Iterative/incremental Method (www.wikipedia.org)

3.1.1 Requirement Identification

It is a process of gathering requirements of the products that specifies what they should do or how to do it. It is a core foundation of the entire software development project. Some of the methods to gather requirements are interviews, research, questionnaires, studying existing systems, reading and analysis of documents, etc.

3.1.1.1 Functional Requirements

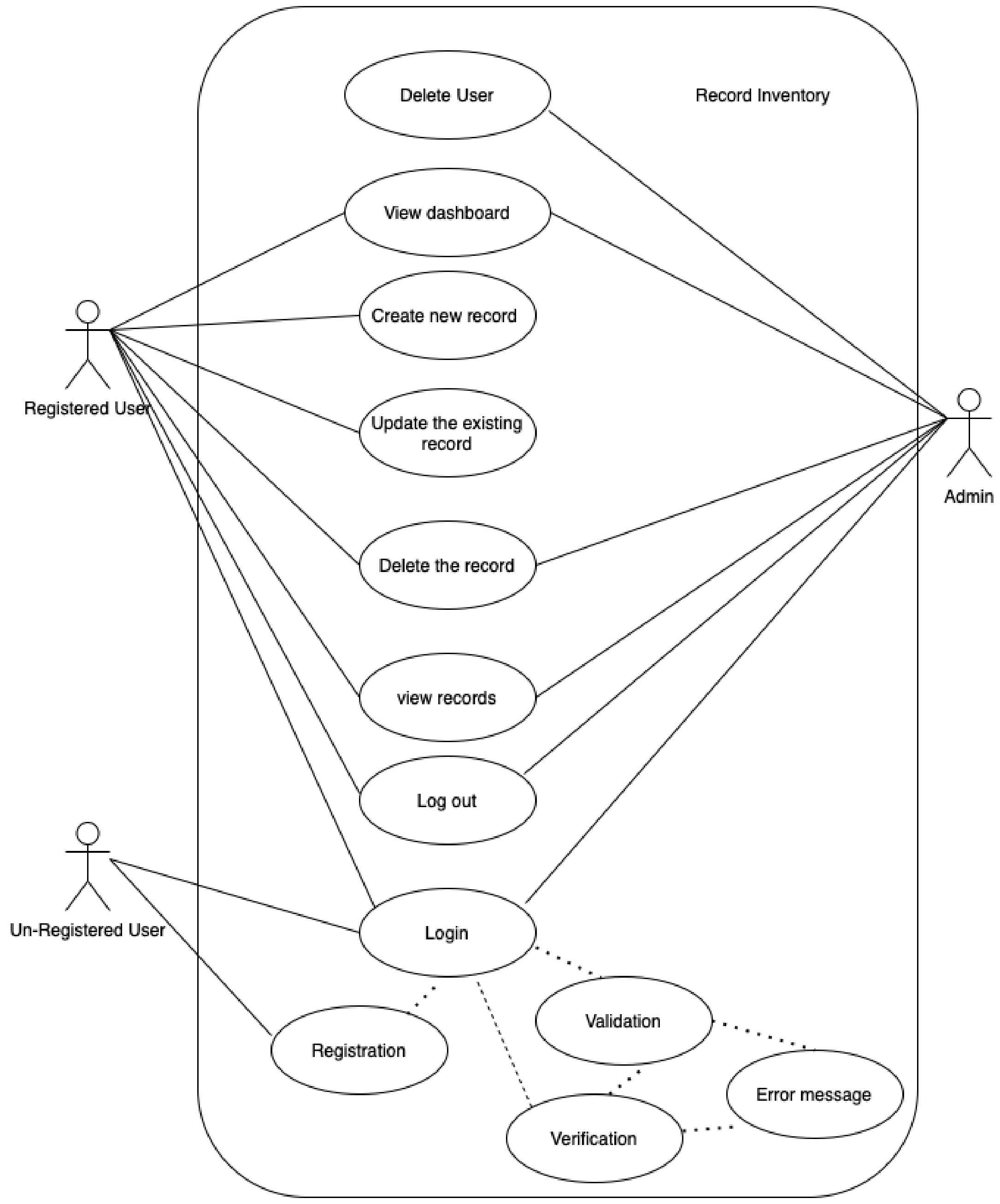


Figure 2: Use case diagram of Record Inventory.

The use case diagram specifies the basic operations a user and admin can perform in the proposed system.

- End users who have registered should be able to access the system.
- Users with the incorrect username will not be validated or allowed to login.
- To use the system for personal use, end users must log in.
- End users shouldn't be able to access the system with erroneous login information.
- Users must have access to their accounts using their username and password.
- Users with correct credentials must be able to perform CRUD (Creation, Retrieve, Update and Delete) operation on the entries on their password database.
- Without logging into the system, direct URL access should not be permitted.
- The most recent versions of modern browsers should have no trouble supporting the web application.
- The system needs to have a logout function.

3.1.1.2 Non Functional Requirements

- Once the user has logged out, they should be brought back to the login page.
- Database exporting should be implemented.
- Both server and client side validation works for login and registration.

3.1.2 Feasibility Study

The system is evaluated for future development with a set of constraints. The feasibility study is done regarding the available technologies, time constraints, area of application, cost of deployment and upkeep, and future possibilities of the project. The key consideration in feasibility analysis are:

3.1.2.1 Technical Feasibility

This system meets the technical feasibility as it will be using existing technologies like HTML, CSS, JavaScript, PHP, MySQL, etc. as well as simple hardware specifications. Since the project is web-based, users can easily access it through a web browser in the presence of the internet or a connection to the server from any hardware platform. Development of the project will be done in free software and on free platforms such as Visual Studio Code, GitHub, and Mozilla Firefox and Chromium-based browsers.

3.1.2.2 Operational Feasibility

Since the system promises to provide an easier and more understandable user interface as well as responsiveness when used on another device. Thus, the proposed system will be operationally feasible.

Users of this application must be familiar with web browsers in general. The following interactions should be possible for users to carry out:

- Launch an application by entering a URL in the browser.
- navigate a software program.
- Register for the application's primary service (such as stock recording).
- Log in to the application.

3.1.2.3 Economic Feasibility

Software used to develop this project will be mostly with open source and free software.The system will be feasible economically as the only resources needed are a laptop, internet connection, and electricity.

3.1.3 Data Modelling (E-R Diagram)

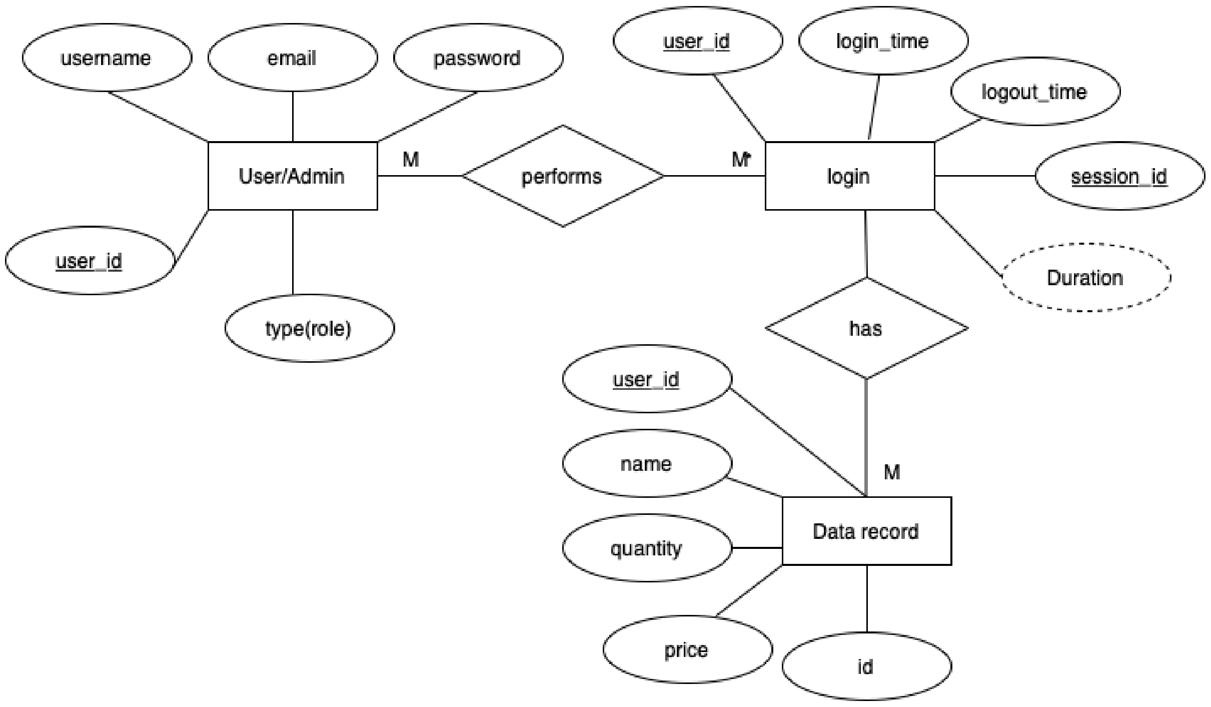


Figure 3 : E-R Diagram

The above figure shows the Entity-Relationship Diagram for Record Inventory. It helps to systematically analyze data requirements to produce a well-designed database. It shows how the data flows from the user to the product.

3.1.4 Data Modelling: Data Flow Diagram

3.1.4.1 DFD level 0 (Context-free diagram)

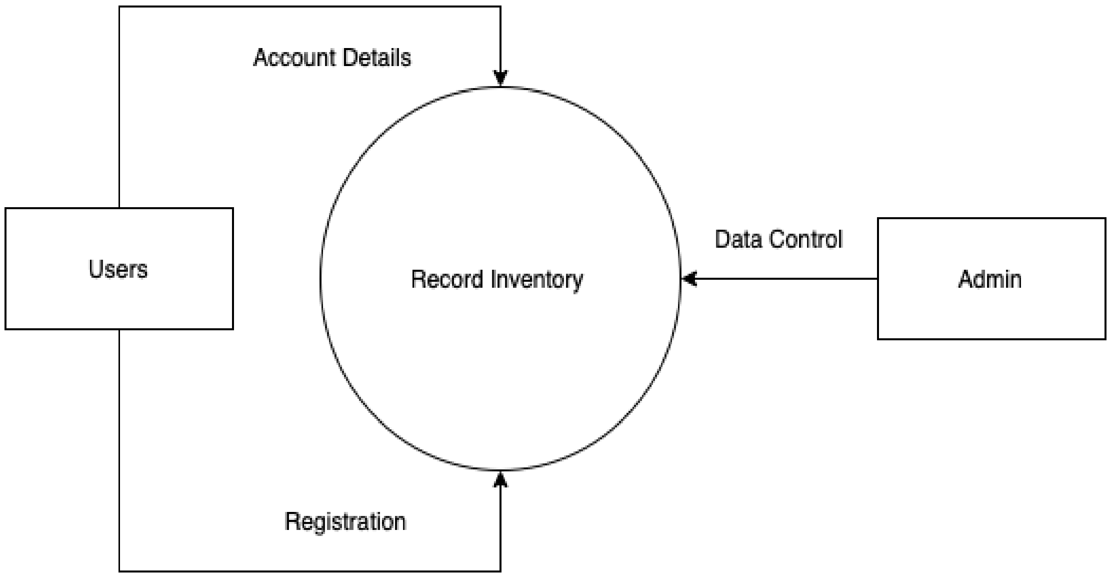


Figure 4: Context-free diagram (DFD level 0)

Context diagram shows the graphical representation of data for Record Inventory in which we can see the Users, and Admin. DFD clearly shows how the users can register and admin has data control of the user; the data which are stored in the database.

3.1.4.2 DFD level 1

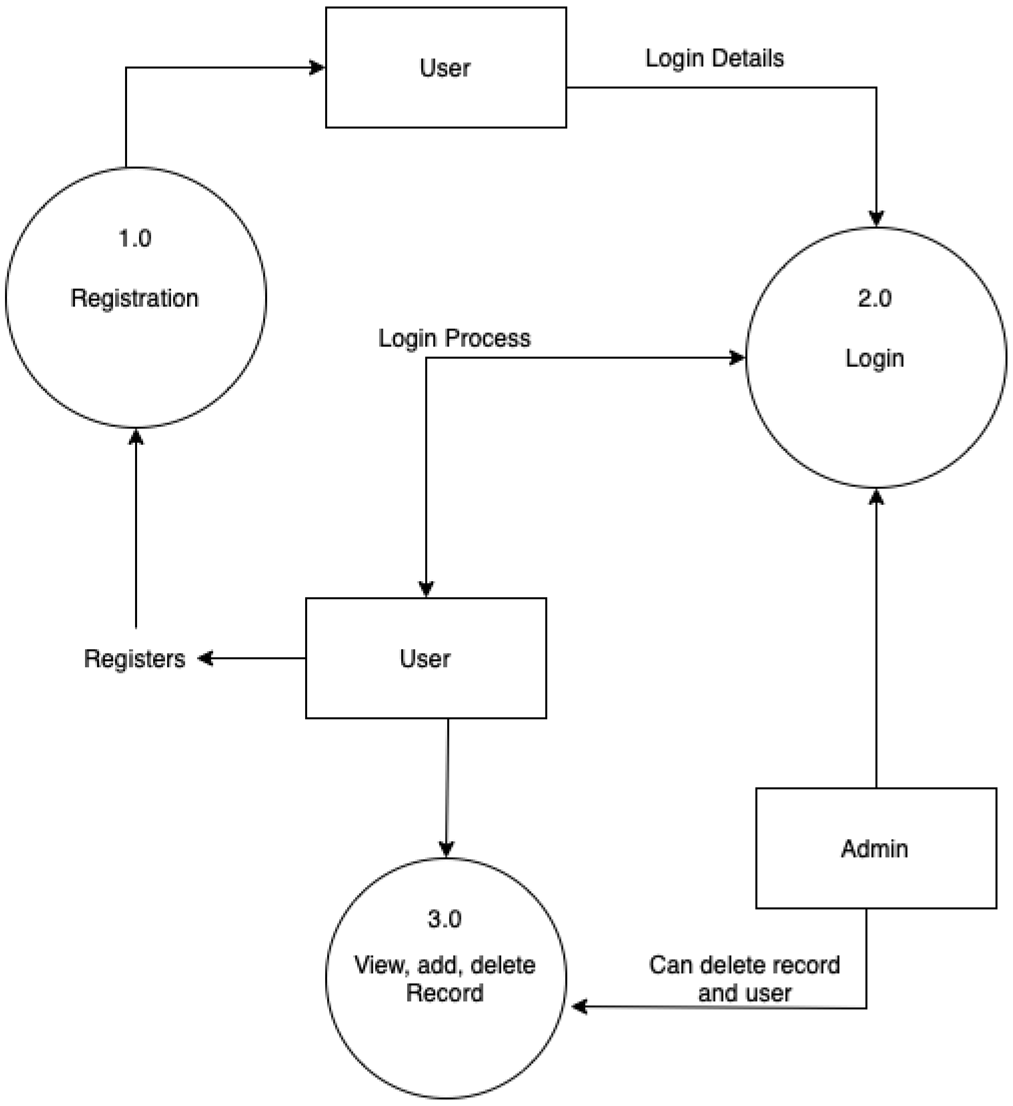


Figure 4.1: Data Flow Diagram Level 1 (User)

It shows the representation of data for Record Inventory in which we can see the flow of user data.

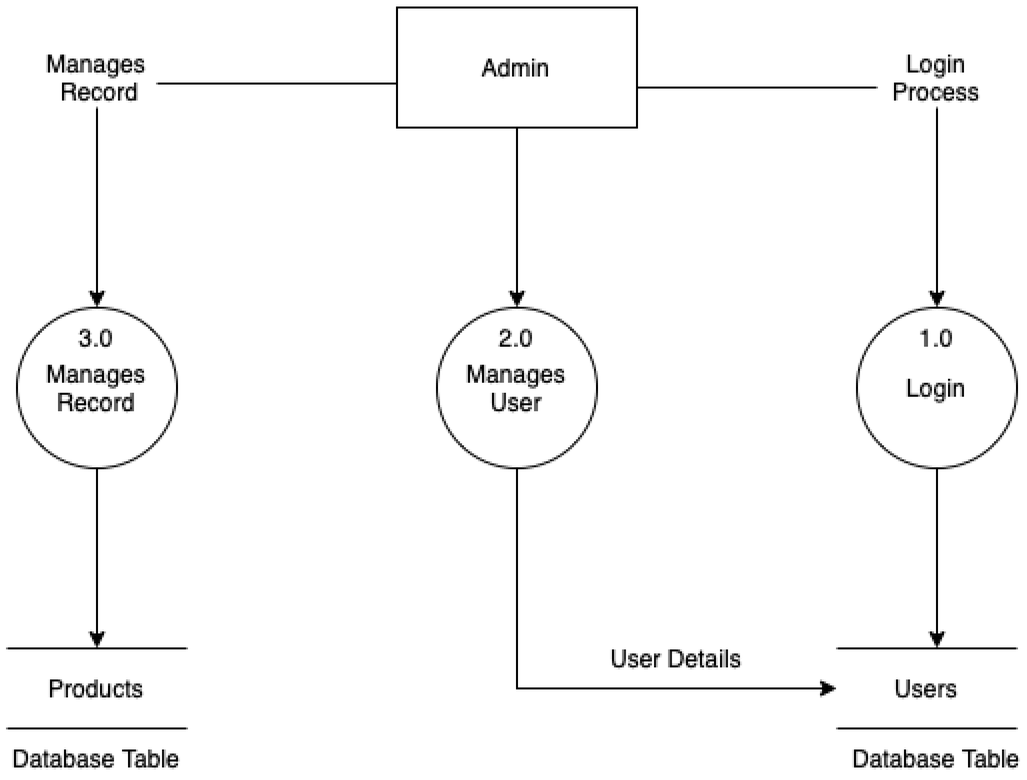


Figure 4.2: Data Flow Diagram Level 1 (Admin)

Data Flow Diagram (DFD) shows the graphical representation of data for Record Inventory in which we can see the data of record, user details, admin, and users. DFD, clearly shows how data flows between the entities and how data are stored in a database.

3.1.4.3 DFD level 2

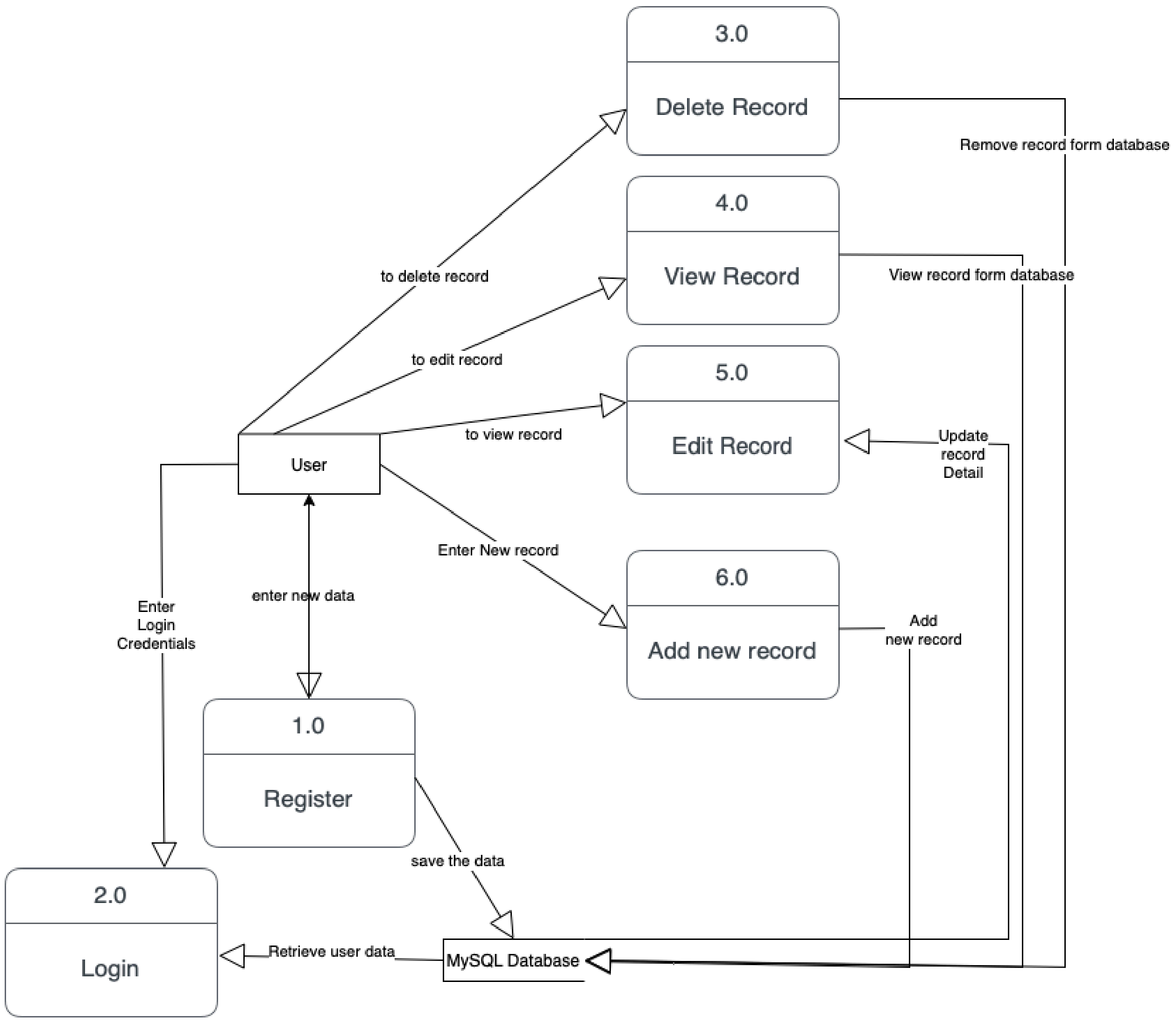


Figure 4.3: Data Flow Diagram

The data flow in the "Record Inventory" system includes registering and logging in, with their account details stored in a MySQL database. Users can view, edit, and delete product record details, view their records.

3.2. System Design

3.2.1. Architecture Design

This project makes use of a three-tier architecture. The functional process logic, data access, computer data storage, and user interface are designed and maintained as independent modules on separate modules.

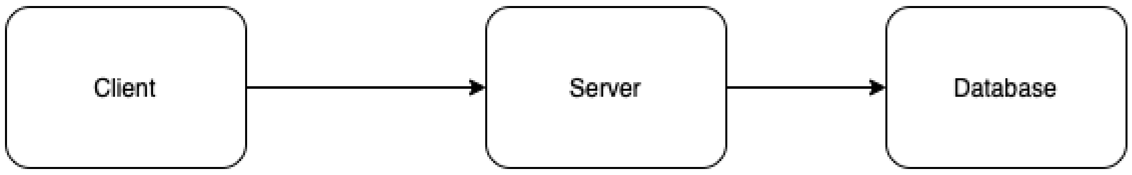


Figure 5 : Three Tier Architecture

Three-Tier Architecture has the following layers.

Client Layer: Contains the user interface that enables users to interact with the application and occupies the top level. Requests and responses are sent to and received from the other levels by this layer in order to communicate.

Server Layer: This layer, which is also known as the application layer or application server, is in charge of handling user requests and producing responses. It controls the operation of the program and interacts with the database layer to retrieve or update data.

Database Layer: The bottom layer houses the database server where data is stored and retrieved. The data is kept independent of application servers or the business logic.

Following are the reasons behind choosing three tier-architecture:

- ·Components within can be reused.
- Independent tiers are easier to maintain.
- Changing platforms only affects only one layer.
- Each layer can potentially run on a different machine.

3.2.2. Database Schema Design

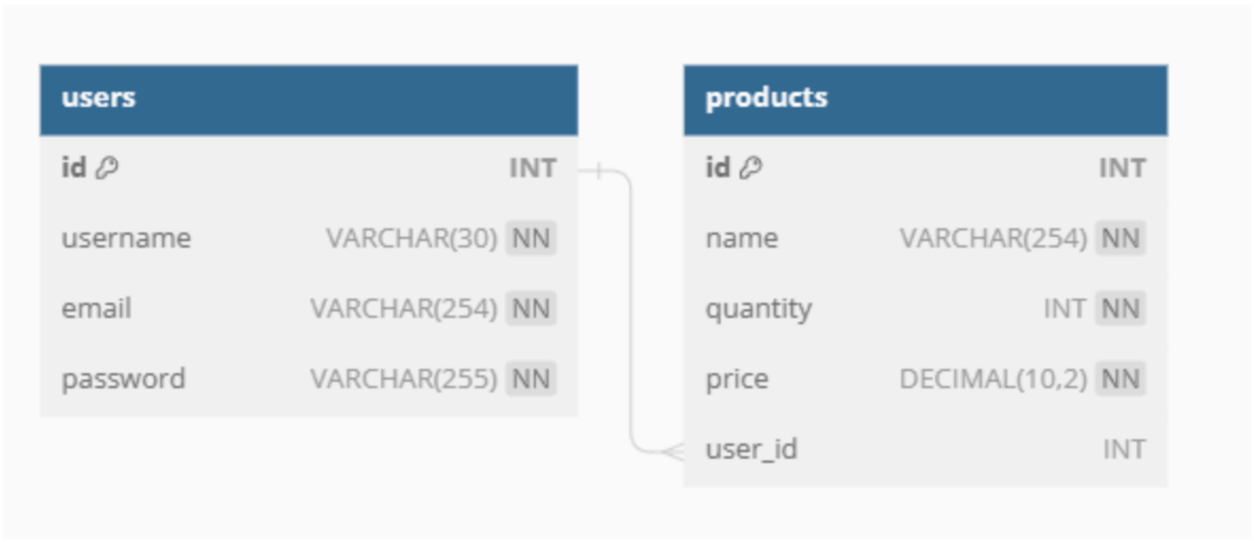


Figure 6 : Database Schema Design of Record Inventory

The database schema design for Track Inventory includes the following tables:

- users: This table stores user and admin information, including the `user_id`, `username`, `email`, `password`. The `username` and `email` fields have unique constraints. Admin and user are separated by a type in the database table.
- products: This table stores information about product i.e product name, id, quantity and price.

3.2.3 Interface Design

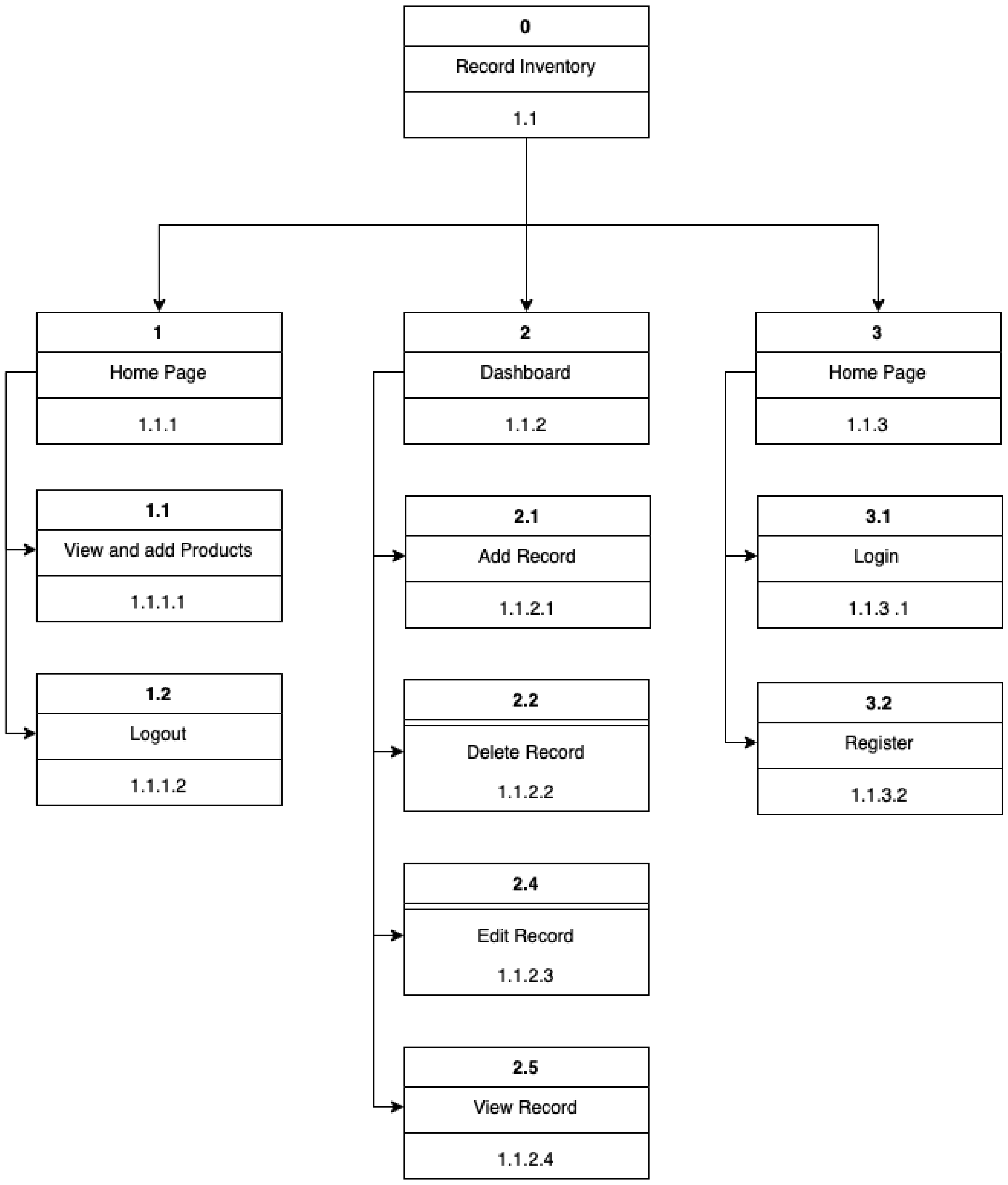


Figure 7: Interface Design

3.2.4 Physical DFD

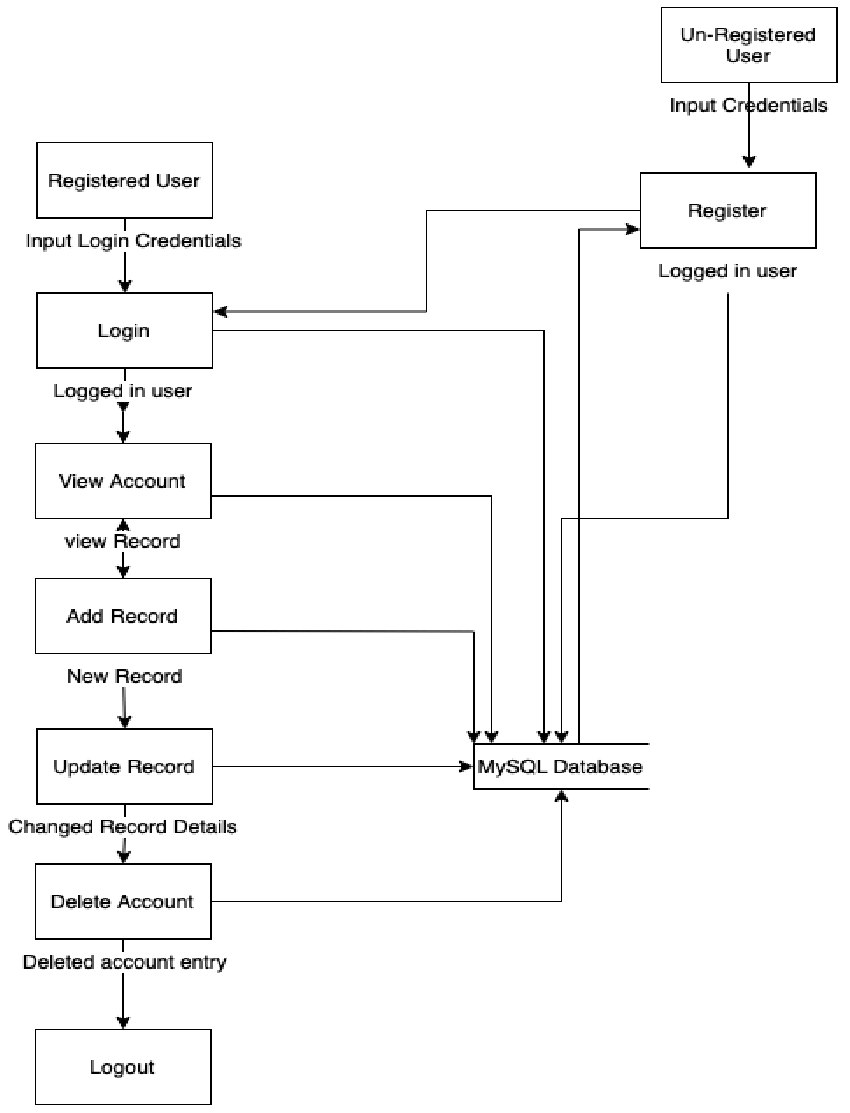


Figure 8: Physical DFD

The above figure is a Physical DFD that visualizes registered user's activity and unregistered user's activity, and the flow of the data between the user, and MySQL database. The user performs activities as logging in, viewing, adding, updating, and deleting record details.

3.2.5 Record Inventory Flowchart Diagram.

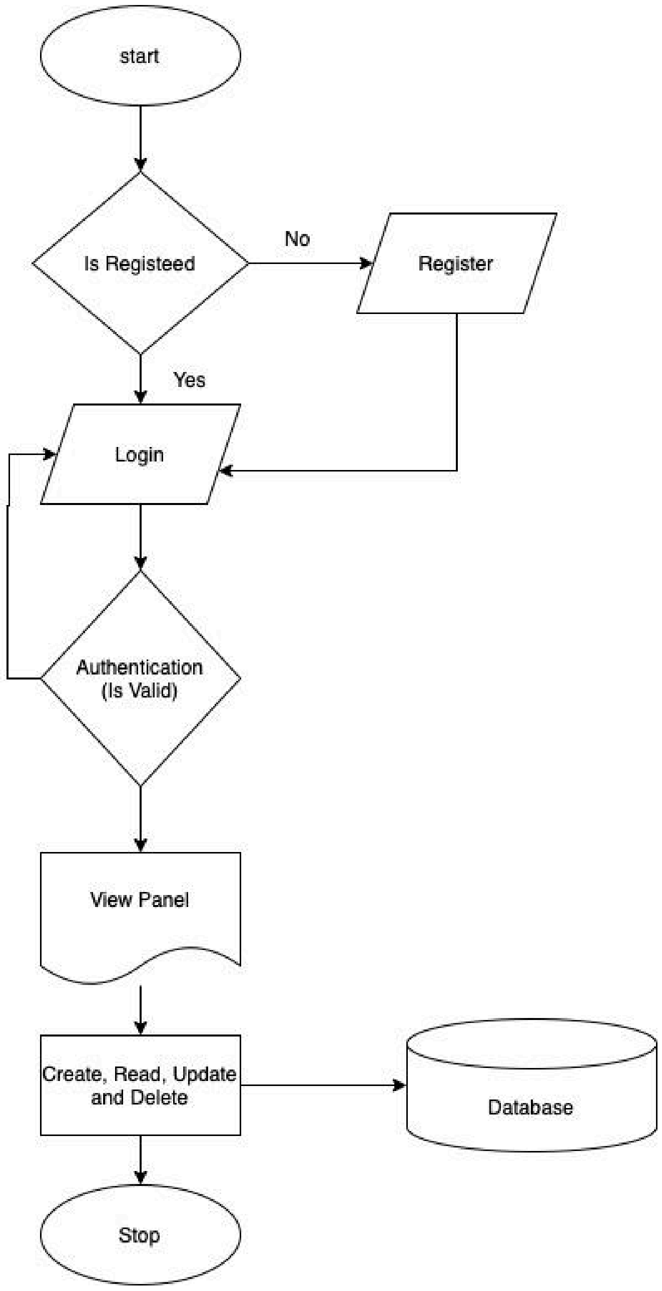


Figure 9: FlowChart registered and unregistered User

This diagram shows that the registered user can log in to get access to the system and view dashboard. Also, the user should register first, and then it gets access to login into the system. After the login is successful the users can add, update, view and delete a record.

Chapter 4: Implementation and Testing

4.1 Implementation

The system was implemented by using various types of programming language and case tools available.

4.1.1 Analysis and design tools used

Various Software tools like Canvas, docs, slides and other tools are used. The coding is done using the compiler application VSCode.

4.1.1.1 Programming tools

4.1.1.2 Front End Tools

- **HTML :-**In Record Inventory, html is used for creating different webpage and sites. It is used to create and structure sections, headings, links, paragraphs using various tags and elements.
- **CSS** :- In Record Inventory, css is used for designing different tags of html. It is also used to design different components by the help of class and id. By using css, we can control the text color, font style, coloring box and background, margin and padding between text boxes, aligning items, and many more.

4.1.1.3 Back End

- **PHP :-**In Track Inventory, PHP is used for the backend purpose and database operations of the system. It is used for server side scripting purposes to add connectivity to the database and also used to encrypt the data, validate the user data, confirm user to login. It also includes adding, updating and deleting the data from the database.

- **Database: MYSQL :-**MySQL is used for storing all the information required to the database in the system. It is used for performing CRUD operations such as creating, deleting and updating data from the database as requested by the user.

4.1.1.4 Dev tools

- Chrome Browser
  Chrome DevTools provides a comprehensive and powerful toolkit for developers to debug, test, and optimize their web applications. With the help of this tool it helped me to edit front end code in real time..

4.1.1.5 CASE Tools

- VScode

  I utilized Visual Studio Code (VS Code) as my code editor. It was my choice for development of the system because it provides easier access to more than one file and parallel development side-on-side view of your code.

- Git and github

  I utilized Git and GitHub for version control and collaborative development. It helped me to manage the source code and track changes made to it.

- Draw.io

  I used it to draw most of the diagrams of the project from scratch, like user case diagrams, entity-relationship diagrams etc.

4.1.2 Implementation detail of module

In this section, the different modules provided are:

Module 1: Register

This module allows the user to register according to the requirements.

```
<?php
include("../includes/config.inc.php");

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

$error = '';

if ($_SERVER["REQUEST_METHOD"] === "POST") {
   $username = $_POST['username'];
   $email = $_POST['email'];
   $password = $_POST['password'];
   $confirm_password = $_POST['confirm_password'];

   // Perform data validation
   if (empty($username)) {
      $error .= "Username is required<br>";
   }
   if (empty($email)) {
      $error .= "Email field is empty<br>";
   }
   if (empty($password)) {
      $error .= "Password field is empty<br>";
   }
   if ($password !== $confirm_password) {
      $error .= "Passwords do not match!<br>";
   }
   echo "<br/>";
        echo "<a href='../index.php'>Go back</a>";
   if (!empty($error)) {
      $error = "<b>There were the following error(s) in your form:</b><br>" .
$error;
   } else {
      // Check if the email already exists in the database
      $query = "SELECT id FROM users WHERE email = :email";
      $stmt = $conn->prepare($query);
      $stmt->bindParam(':email', $email);
      $stmt->execute();

      if ($stmt->rowCount() > 0) {
         $error = "Email already exists<br>";
      } else{
      $hashed_password = password_hash($password,
PASSWORD_DEFAULT);

      if (saveUser($username, $email, $hashed_password)) {
         header('Location: ../index.php');
         exit();
      } else {
         $error = "Error occurred while registering the user.";
      }
   }
}
}

function saveUser($username, $email, $hashed_password)
{
   global $conn;
   $query = "INSERT INTO users(username, email, password,type) VALUES
(:username, :email, :password,'user')";

   $stmt = $conn->prepare($query);
   $stmt->bindParam(':username', $username);
   $stmt->bindParam(':email', $email);
   $stmt->bindParam(':password', $hashed_password);

   return $stmt->execute();
}
?>
```

Module 2: Login

This module is responsible for validating the registered user.

```
<?php
$error = '';
session_start();
if (isset($_POST['login'])) {
include("../includes/config.inc.php");
if ($_SERVER["REQUEST_METHOD"] === "POST") {
   $username = $_POST['username'];
   $password = $_POST['password'];
   // Perform data filters
   if (empty($username) || empty($password)) {
      $error = "Username and password are required.";
   }
   echo "<br/>";
        echo "<a href='../index.php'>Go back</a>";
   // Check if there are any errors
   if (!empty($error)) {
      $error = "<b>There were the following error(s) in your form:</b><br>" .
$error;
   } else {
      if (authenticateUser($username, $password)) {
         $_SESSION['username'] = $username;
         $_SESSION['auth'] = true;
         $_SESSION['type'] = getUserType($username);
         $_SESSION['login_id'] = getUserId($username);
         header('Location: ../users/dashboard.php');
         exit();
      } else {
         $error = "Invalid username or password.";
         }
      }
   }
}

function authenticateUser($username, $password)
{
   global $conn;
```

```
    $query = "SELECT * FROM users WHERE username = :username OR
email = :username";
    $stmt = $conn->prepare($query);
    $stmt->bindParam(':username', $username);
    $stmt->execute();
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    if ($user && password_verify($password, $user['password'])) {
        return true;
    } else {
        return false;
    }
}

function getUserId($username)
{
    global $conn;
    $query = "SELECT id FROM users WHERE username = :username OR
email = :username";
    $stmt = $conn->prepare($query);
    $stmt->bindParam(':username', $username);
    $stmt->execute();
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    return $user['id'];
}

function getUserType($username){
    global $conn;
    $query = "SELECT type FROM users WHERE username = :username OR
email = :username";
    $stmt = $conn->prepare($query);
    $stmt->bindParam(':username', $username);
    $stmt->execute();
    $user = $stmt->fetch(PDO::FETCH_ASSOC);
    return $user['type'];
}
?>
```

Module 3: View and Add Products

This module allows the view existing record, add, update and delete the product record.

```
<?php
session_start();
include("../includes/config.inc.php");

ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);

if ($_SERVER["REQUEST_METHOD"] === "POST") {
   $name = $_POST['name'];
   $quantity = $_POST['quantity'];
        $price = $_POST['price'];
   $id = $_SESSION['login_id'];

   if (empty($name) || empty($quantity) || empty($price)) {
      $error = "All fields are required.";
   } else{
      $query = "INSERT INTO products(name, quantity, price, user_id)
VALUES (:name, :quantity, :price, :id)";

      $stmt = $conn -> prepare($query);

      $stmt -> bindParam(':name', $name);
      $stmt -> bindParam(':quantity', $quantity);
      $stmt -> bindParam(':price', $price);
      $stmt -> bindParam(':id', $id);
      $stmt -> execute();
      header("Location: index.php");
   }
}
?>
//code to delete record
<?php
session_start();
include("../includes/config.inc.php");
```

```
$id = $_GET['id'];
$query = "DELETE FROM products WHERE id= :id";
$stmt = $conn->prepare($query);
$stmt->bindParam(':id', $id);
$stmt->execute();
if($_SESSION['type'] == 'admin'){
   header("Location: Aindex.php");
}else if($_SESSION['type'] == 'user'){
   header("Location: index.php");
}
?>

//code to edit record
<?php

session_start();
include("../includes/config.inc.php");
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
if ($_SERVER["REQUEST_METHOD"] === "POST") {
   $id = $_POST['id'];
   $name = $_POST['name'];
   $quantity = $_POST['quantity'];
   $price = $_POST['price'];
   $updateQuery = "UPDATE products SET name = :name, quantity =
:quantity, price = :price WHERE id = :id";
   $updateStmt = $conn->prepare($updateQuery);
   $updateStmt->bindParam(':id', $id);
   $updateStmt->bindParam(':name', $name);
   $updateStmt->bindParam(':quantity', $quantity);
   $updateStmt->bindParam(':price', $price);

   if ($updateStmt->execute()) {
      header("Location: index.php");
      exit();
   } else {
      $error = "Failed to update product.";
   }
}
```

```
if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $query = "SELECT * FROM products WHERE id = :id";
    $stmt = $conn->prepare($query);
    $stmt->bindParam(':id', $id);
    $stmt->execute();
    $product = $stmt->fetch(PDO::FETCH_ASSOC);
    if (!$product || $stmt->rowCount() === 0) {
        $error = "Product not found.";
    }
}
?>

//Code for admin
<?php
session_start();
ini_set('display_errors', 1);
ini_set('display_startup_errors', 1);
error_reporting(E_ALL);
include("../includes/config.inc.php");
$query = "SELECT * FROM products ORDER BY id DESC";
$stmt = $conn->prepare($query);
$stmt->execute();
$products = $stmt->fetchAll(PDO::FETCH_ASSOC);
$i = 0;
?>
```

Module 4: Logout

This module allows users to logout.

```
<?php

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

4.2 Testing

4.2.1 Testing cases for unit Testing

Unit testing focuses on the functioning of the smallest unit of the software design. Unit testing is applied to each of the smallest modules so that we are ensured that every module is producing the required result. It helps to ensure that even the higher modules dependent on the smallest module function properly. Every error in the smallest modules is properly examined and tested to produce the correct output.

Unit Testing

- **Testing cases for unit testing of signup.**
- **The testing case for login verification.**
- **Test case for adding data.**
- **Test case for updating data.**
- **Test case for deleting data.**
- **Test case for logout.**
- **Test case for admin login.**
- **Test case for user management.**
- **Test case for product management.**

- **System Testing**

Unit Testing

Project Name: Track Inventory

Table 1: Test Case 1

Test Case ID : 1 **Test Title :** Verify Registration/Sign up Form **Designed Date :** 12 June, 2023

Pre-conditions : Un-registered user clicking register button. **Dependencies:**

Test Description	**Test Data**	**Expected Outcome**	**Actual Result**	**Status (Pass/ Fail)**
Navigating to register and mistakenly clicked the register button	All Empty	Data fields required	Error message Displays: Username is required, Email field is empty and password field is empty	Pass
Username entered but not entered email and password	Username: Yara	Email and password field required	Error Message Displays: Email field is empty and password is Empty.	Pass
Username and email entered but not password and confirm password	Username: Yara Email: Yara2@gmail.com	Password field required	Error Message Displays: Password field is empty	Pass
Password and confirm password doesn't match	Username: Yara Email: Yara2@gmail.com Password: yara123 Confirm Password: yara122	Confirm Password doesn't match	Error Message Displays: Passwords do not match	Pass

Table 2: Test Case 2

Test Case ID : 2 **Test Title :** Verify login with valid username and password **Test Designed Date :** 13 June, 2023

Pre-conditions : User has valid username and password Dependencies:

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
1. Navigate to login page * Provide valid username * Provide valid password * Click login button	Username: Yara Password: password123	User should be able to login	User is navigated to the dashboard with successful login	Pass
Provide invalid username and password.	Username: cat Password: cat	User login invalid	User login invalid	Pass
Clicks on login button	empty	Enter username and password	Username and password are required	Pass

Post-condition: User is validated with the database and successfully logs into the account. The account session details are logged in database

Table 3: Test Case 3

Test Case ID : 3 **Test Title :** Adding record **Test Designed Date :** 25 June, 2022

Pre-conditions : Entry should be added. **Dependencies:**

Test Description	**Test Data**	**Expected Outcome**	**Actual Result**	**Status (Pass/ Fail)**
Click the Create Products button to add a record.	Name:Mac Quantity: 2000 Price:150,000	Records should be added to the database and displayed in the view section.	Added Successfully	Pass
Click on the Create Products without providing any data	Name : Quantity: Price:	Error message saying enter other fields as well.	All fields are required	Pass

Post-condition: Records are added.

Table 4: Test Case 4

Test Case ID : 4 **Test Title :** Updating record **Test Designed Date :** 25 June, 2022

Pre-conditions : Entry should be updated. **Dependencies:**

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
Click the Edit button to edit a record.	Name:Mac Quantity: 3000 Price:550,000	Records should be updated to the database and displayed in the view section.	Updated Successfully	Pass

Post-condition: Records are updated.

Table 5: Test Case 5

Test Case ID : 5 **Test Title :** Deleting record **Test Designed Date :** 25 June, 2022

Pre-conditions : Entry should be deleted. **Dependencies:**

Test Description	**Test Data**	**Expected Outcome**	**Actual Result**	**Status (Pass/ Fail)**
Click the Delete button to delete a record.	Name:Mac Quantity: 3000 Price:550,000	Confirmation about : Are you sure you want to delete?	Deleted Successfully from database	Pass

Post-condition: Record should be deleted.

Table 6: Test Case 6

Test Case ID : 6 **Test Title :** Verify logout **Test Designed Date :** 25 June, 2022

Pre-conditions : User has valid username and password and has logged in. **Dependencies:**

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
Click on logout button	Simply click on logout button	Users should be able to logout.	User is logged out	Pass

Post-condition: User is redirected to the login page.

Table 7: Test Case 7

Test Case ID : 7 **Test Title :** Verify admin login **Test Designed Date :** 25 july, 2022	

Pre-conditions : Admin can login in **Dependencies:**	

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
Click on login button	Username: admin Password: admin	Login successful	success	Pass

Post-condition: Admin is redirected to the login page.	

Table 8: Test Case 8

Test Case ID : 8 **Test Title :** User Management **Test Designed Date :** 25 july, 2022

Pre-conditions : Admin can delete users. **Dependencies:**

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
Click on delete button	Click the delete action	Data deleted	Record deleted	Pass

Post-condition: User data is deleted.

Table 9: Test Case 9

Test Case ID : 9 **Test Title :** Products Management. **Test Designed Date :** 25 july, 2022

Pre-conditions : Admin can delete user's recorded data. **Dependencies:**

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
Click on delete button	Click the delete action	Product record deleted successfully.	Confirmation for record deletion.	Pass

Post-condition: Product data is deleted successfully.

4.2.2 Testing case for system testing

System testing was done to notify that all the modules work together without any error. Test for every module was done specifically to find out any errors in the modules while working together. Eventually, it was concluded that all the modules are functioning properly.

System Testing

Table 10: Test Case 10

Test Case ID : 10 **Test Title :** System Testing. **Test Designed Date :** 2 August, 2022

Pre-conditions : Admin can delete user's recorded data. **Dependencies:**

Test Description	Test Data	Expected Outcome	Actual Result	Status (Pass/ Fail)
Login with a valid email and password.	Username: admin Password: admin	Login success	Logs in to dashboard	Pass
Login with a valid email and password.	Username: yara2@gmail.com Password: yara123	The user should be taken to the homepage with all the set of options provided as per their role	Logins and ends up on the dashboard.	Pass

Click on view and add products & create products	Name: calculator Quantity: 100 Price: 750	The user should successfully add record data.	Data added successfully.	Pass
Click on the create button without adding any data	Name: Quantity: Price:	Show an error message.	All fields are required	Pass
Click on the delete button on users management section	Clicking the delete button.	Shows an alert to confirm for deleting user data.	Data is deleted.	Pass
Click on the delete button on product management section	Clicking on the delete button.	Shows an alert to confirm for deleting product records.	Data deleted successfully.	Pass

Post-condition:
Everything works successfully without any errors.

Chapter 5: Conclusion and Future Recommendations

5.1. Lesson Learnt / Outcome

I gained knowledge about project management, coding structure maintenance, and project programming from working on this project. I also gained knowledge on how to create the system's necessary schematics. I gained a lot of knowledge that we may use to advance any project-related task that we undertake.

5.2. Conclusion

Any device with internet connectivity can access the system's record-keeping platform, and it comes with source code that enables customization. This system will produce a data-recording web application that is useful, appealing, and simple to use.

5.3. Future Recommendations

The project will be upgraded and extended with the following features:

- Improving the design, implementation, and documentation in such a way that the page is accessible.
- The web page can be made lighter so that even with low internet speed, the page can be browsed.
- The web application will be made into Progressive Web App.
- Implementing e-commerce sites.
- Adding a report system or exporting data into excel sheets for further analysis.
- If a comprehensive solution has not yet been developed, the system should be evolved by creating numerous versions based on feedback.

References:

[1] R. (n.d.). *GitHub - RohanMaharjan123/CRUD: CRUD operation in PHP*. GitHub. https://github.com/RohanMaharjan123/CRUD

[2]]*Sci-Hub: science for the people*. (n.d.). Sci-Hub: Science for the People. https://sci-hub.se DeHoratius, N., & Raman, A. (2008). Inventory Record Inaccuracy: An Empirical Analysis. Management Science, 54(4), 627–641. doi:10.1287/mnsc.1070.0789

[3]*Google Scholar*. (n.d.). Google Scholar. https://scholar.google.com

[4] Iterative and incremental development - Wikipedia. (2017, August 9). Iterative and Incremental Development - Wikipedia. https://en.wikipedia.org/wiki/Iterative_and_incremental_development

[5]*Inventory record*. (n.d.). CEOpedia | Management Online. https://ceopedia.org/index.php/Inventory_record

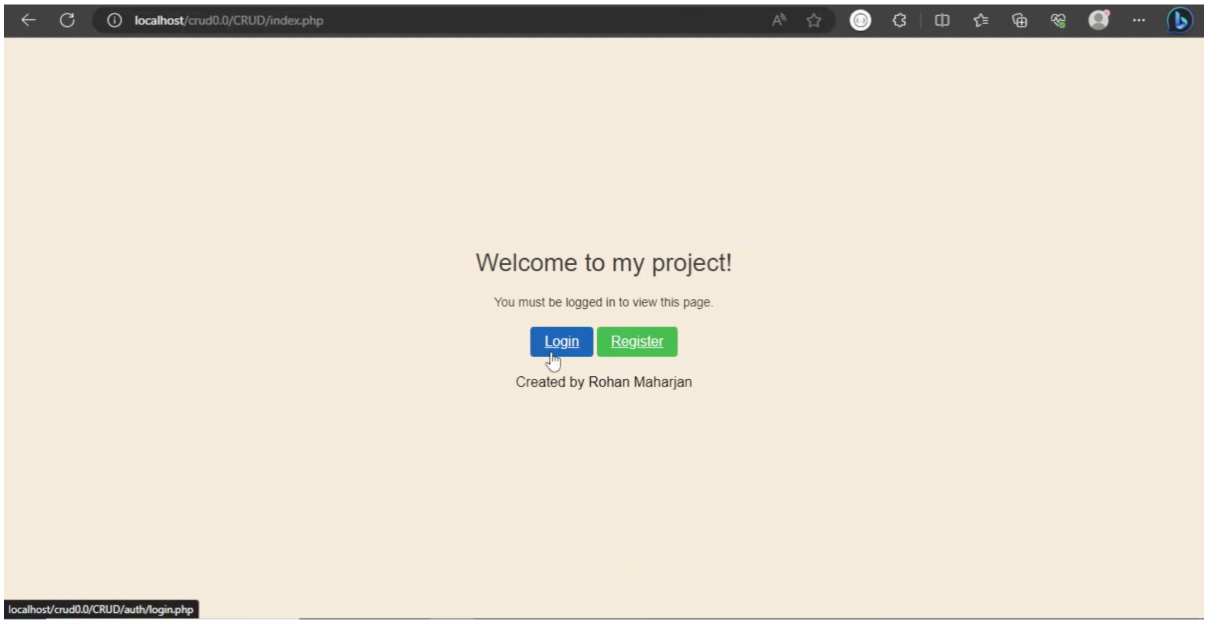
[6] JavaScript | MDN. (n.d.). JavaScript | MDN. https://developer.mozilla.org/en-US/docs/Web/JavaScript

[7] *PHP: Hypertext Preprocessor*. (2023, August 4). PHP: Hypertext Preprocessor. https://www.php.net

[8]*Database | Definition, Types, & Facts*. (n.d.). Encyclopedia Britannica. https://www.britannica.com/technology/database

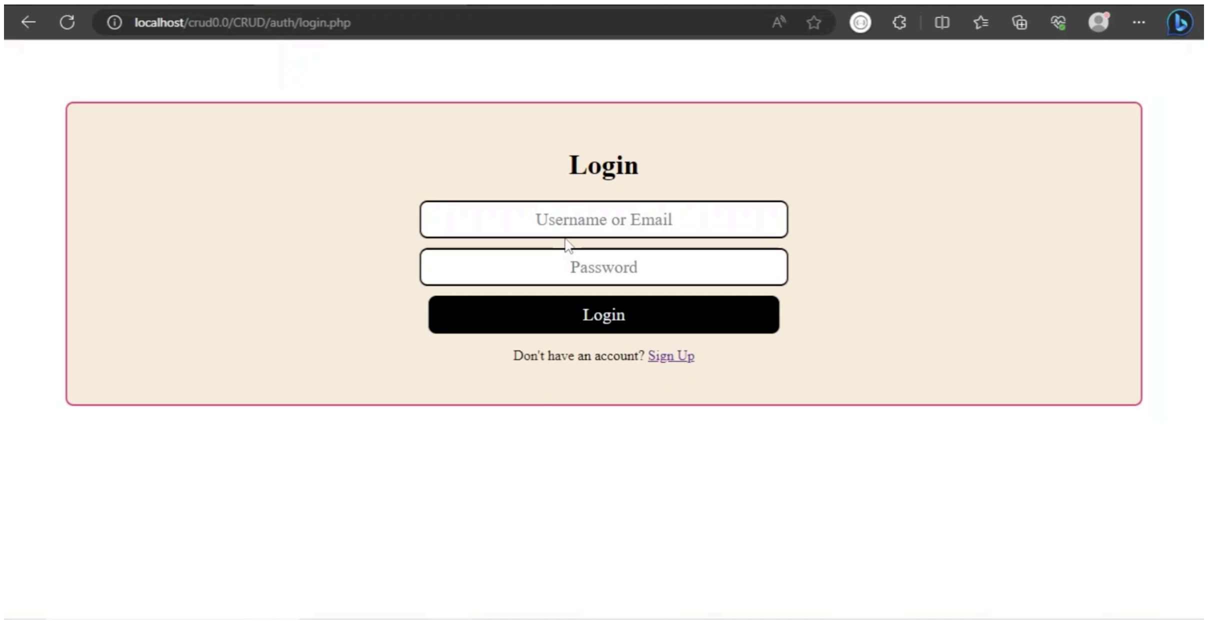
Appendix

Screenshot of locally hosted site


localhost/crud0.0/CRUD/index.php
Welcome to my project!
You must be logged in to view this page.
Login
Register
Created by Rohan Maharjan
localhost/crud0.0/CRUD/auth/login.php

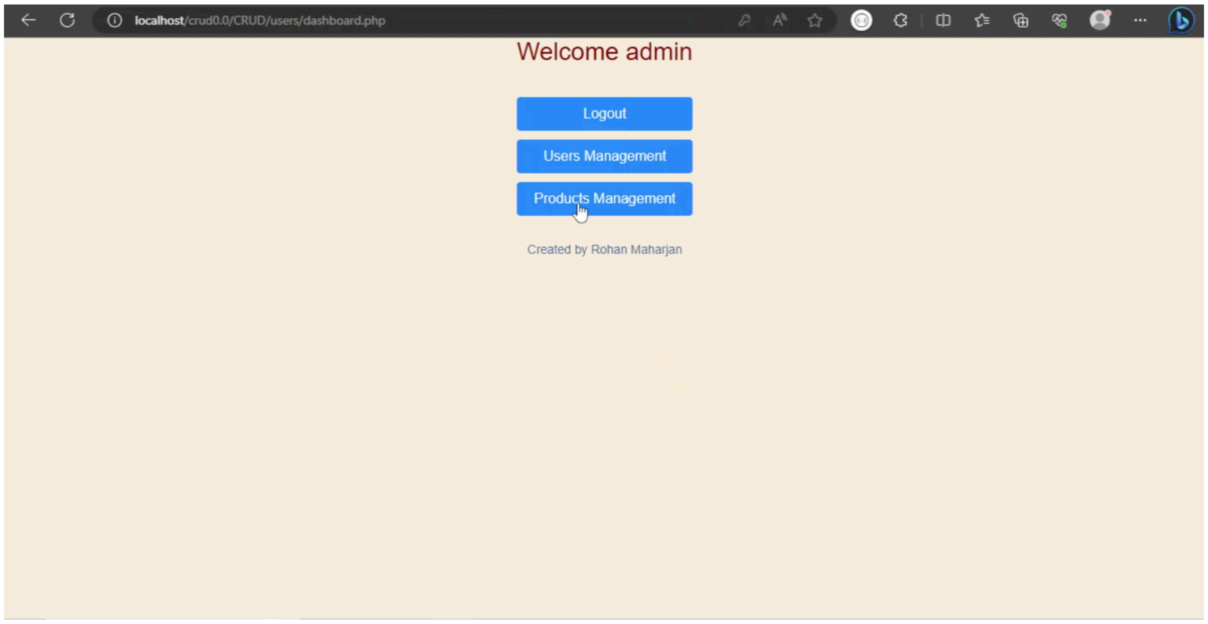
Description: Homepage of Record Inventory with Login and Register link.

Admin Login Page


localhost/crud0.0/CRUD/auth/login.php
Login
Username or Email
Password
Login
Don't have an account? Sign Up

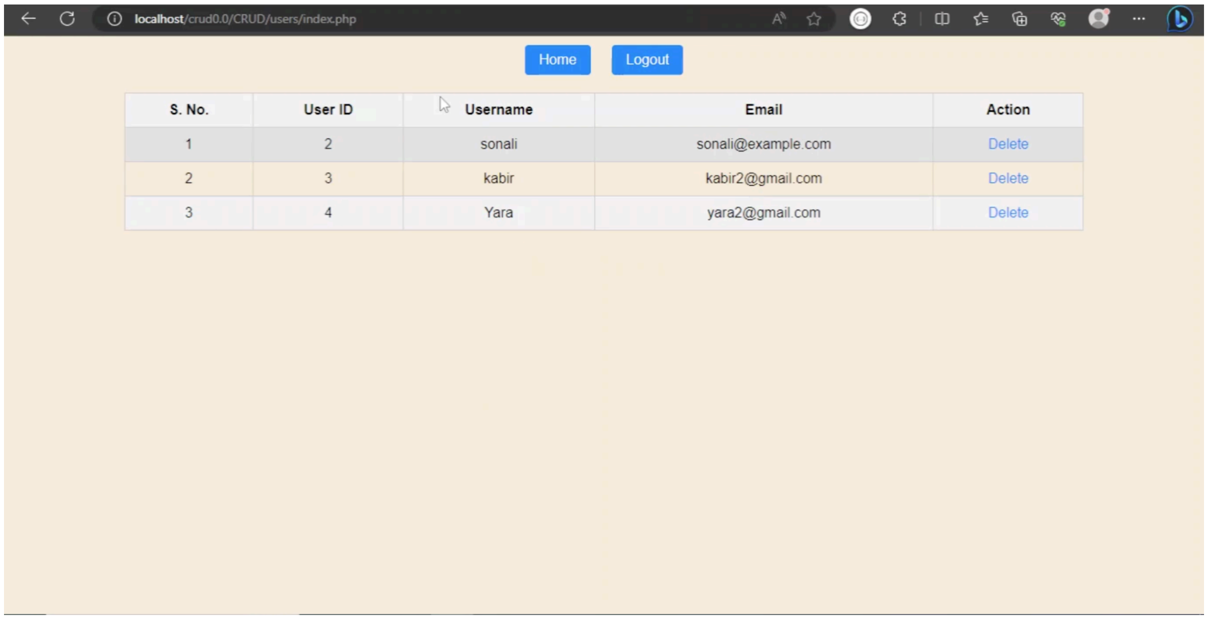
Description: Sign-in page includes a link to a sign-up page with a sign-in button and a username and password for signing in

Admin Dashboard


localhost/crud0.0/CRUD/users/dashboard.php
Welcome admin
Logout
Users Management
Products Management
Created by Rohan Maharjan

Description: Control over user management, record management(Product) and logout.

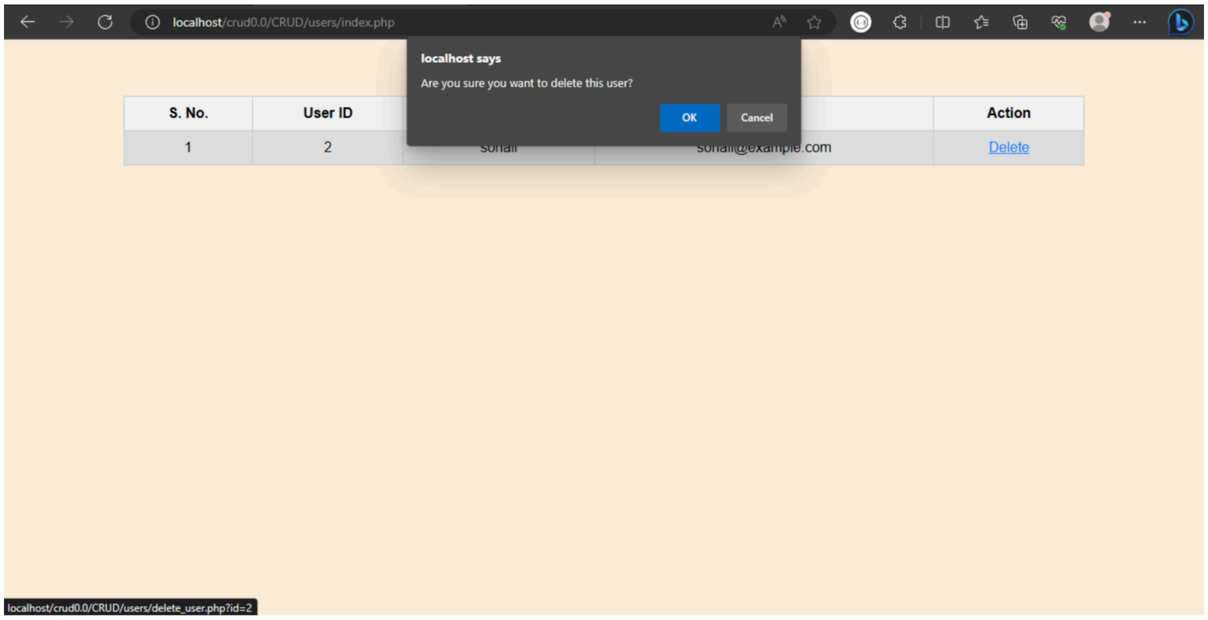
Dashboard



S. No.	User ID	Username	Email	Action
1	2	sonali	sonali@example.com	Delete
2	3	kabir	kabir2@gmail.com	Delete
3	4	Yara	yara2@gmail.com	Delete

Description: Dashboard that has data control to admin.

Deleting User


localhost/crud0.0/CRUD/users/index.php

localhost says

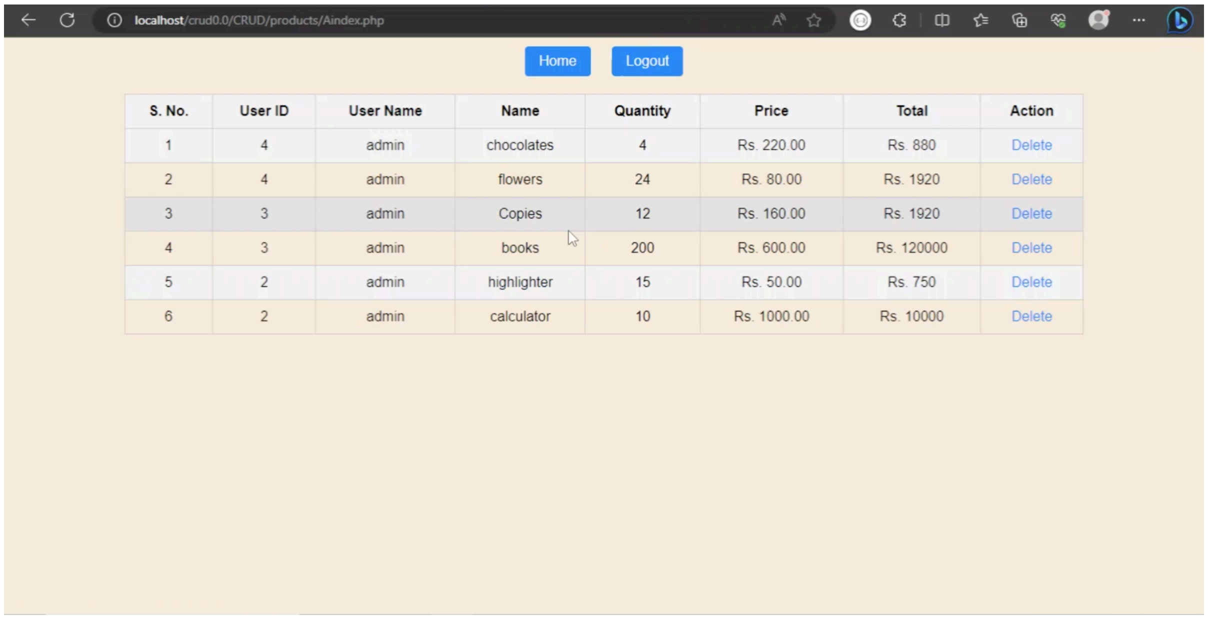
Are you sure you want to delete this user?

S. No.	User ID			Action
1	2	sonali	sonali@example.com	Delete

localhost/crud0.0/CRUD/users/delete_user.php?id=2

Description: Confirmation before deleting a user.

Products detail

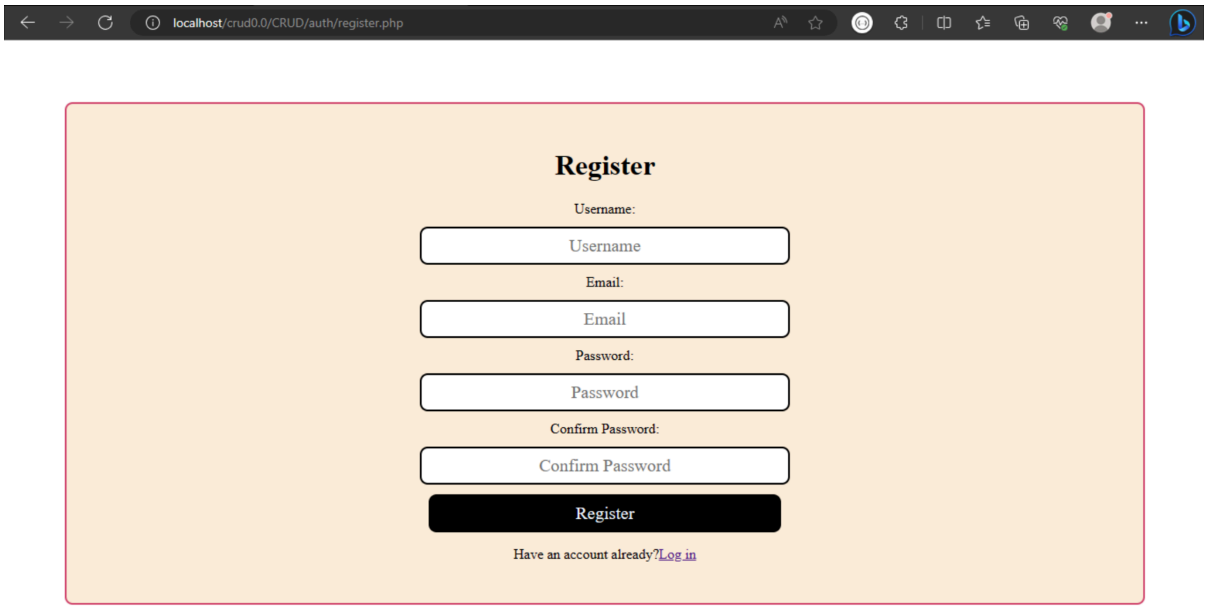


localhost/crud0.0/CRUD/products/Aindex.php

S. No.	User ID	User Name	Name	Quantity	Price	Total	Action
1	4	admin	chocolates	4	Rs. 220.00	Rs. 880	Delete
2	4	admin	flowers	24	Rs. 80.00	Rs. 1920	Delete
3	3	admin	Copies	12	Rs. 160.00	Rs. 1920	Delete
4	3	admin	books	200	Rs. 600.00	Rs. 120000	Delete
5	2	admin	highlighter	15	Rs. 50.00	Rs. 750	Delete
6	2	admin	calculator	10	Rs. 1000.00	Rs. 10000	Delete

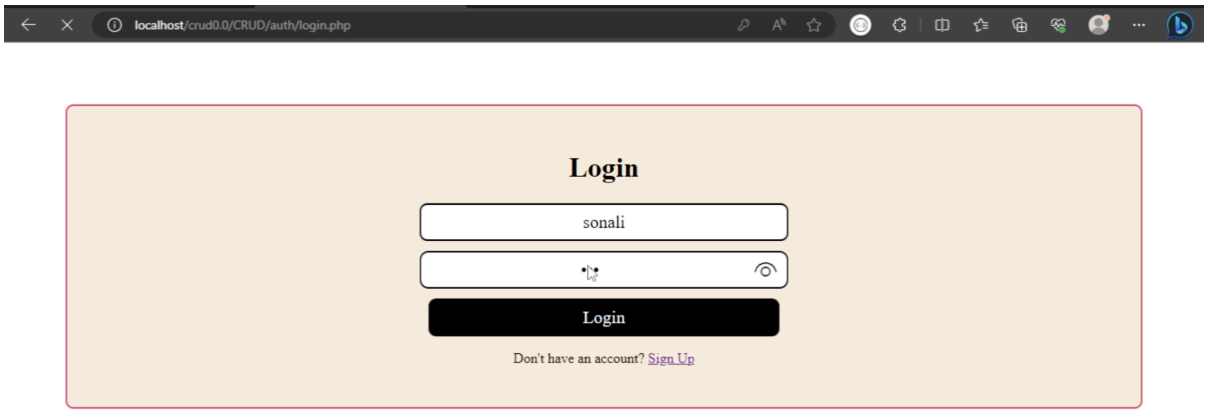
Description: Page to view entries for admin and delete record.

User Registration


localhost/crud0.0/CRUD/auth/register.php

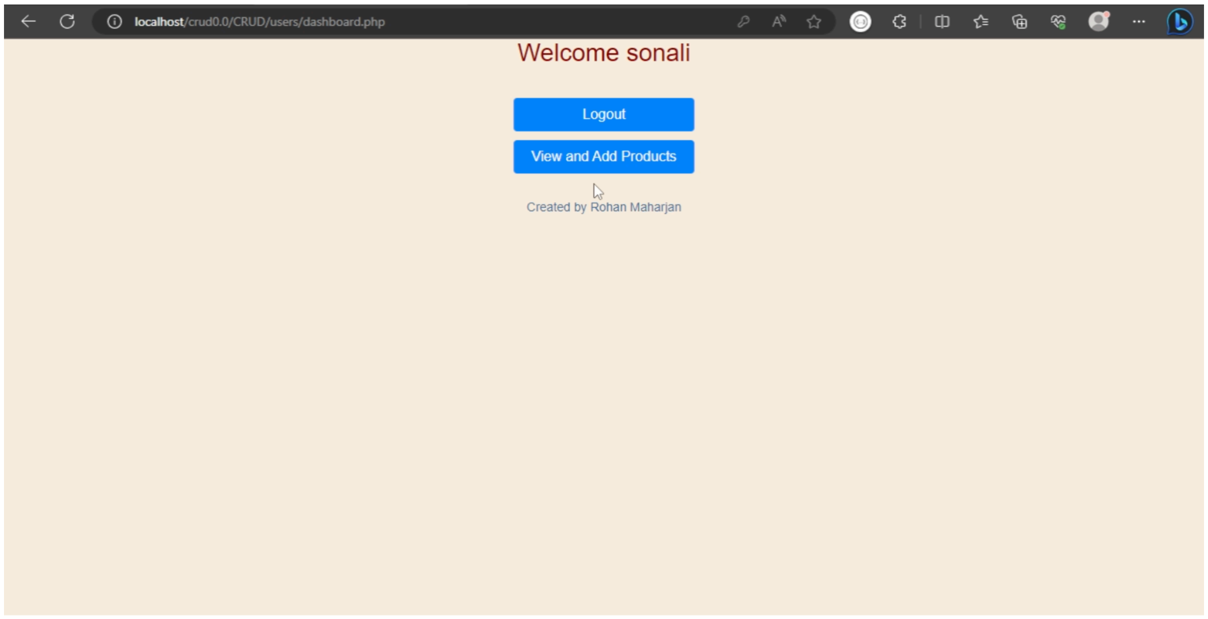
Description: Registration for users.

User login


localhost/crud0.0/CRUD/auth/login.php

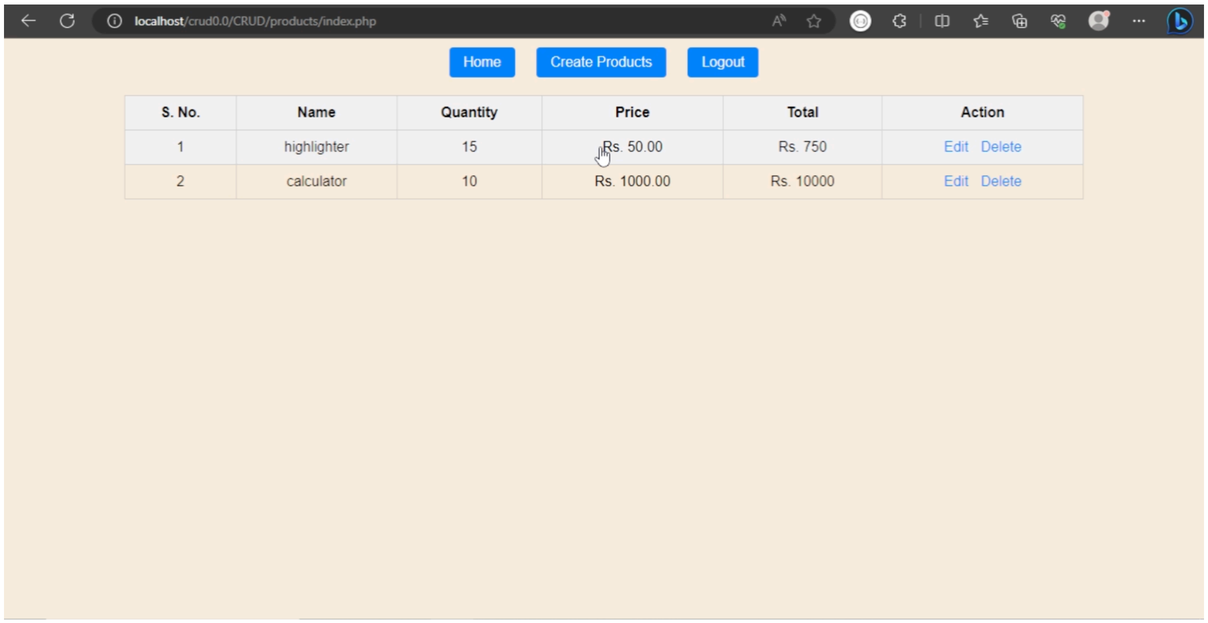
Description: Login for user after registration.

Dashboard for User


localhost/crud0.0/CRUD/users/dashboard.php
Welcome sonali
Logout
View and Add Products
Created by Rohan Maharjan

Description: Dashboard for user.

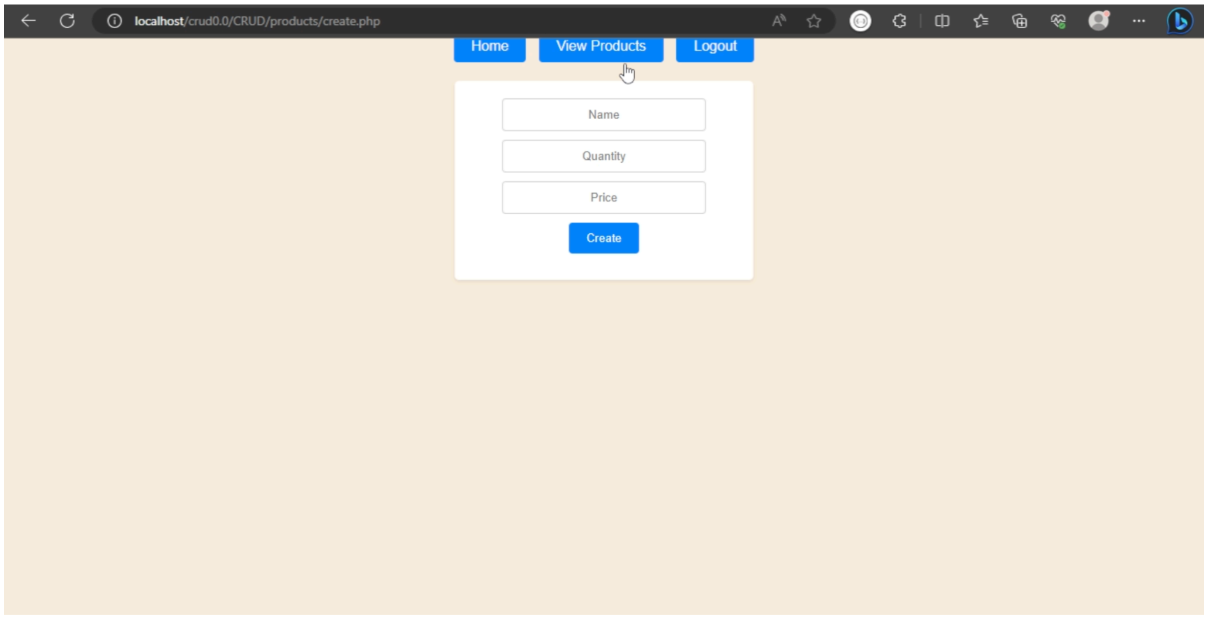
Dashboard to view record



S. No.	Name	Quantity	Price	Total	Action
1	highlighter	15	Rs. 50.00	Rs. 750	Edit Delete
2	calculator	10	Rs. 1000.00	Rs. 10000	Edit Delete

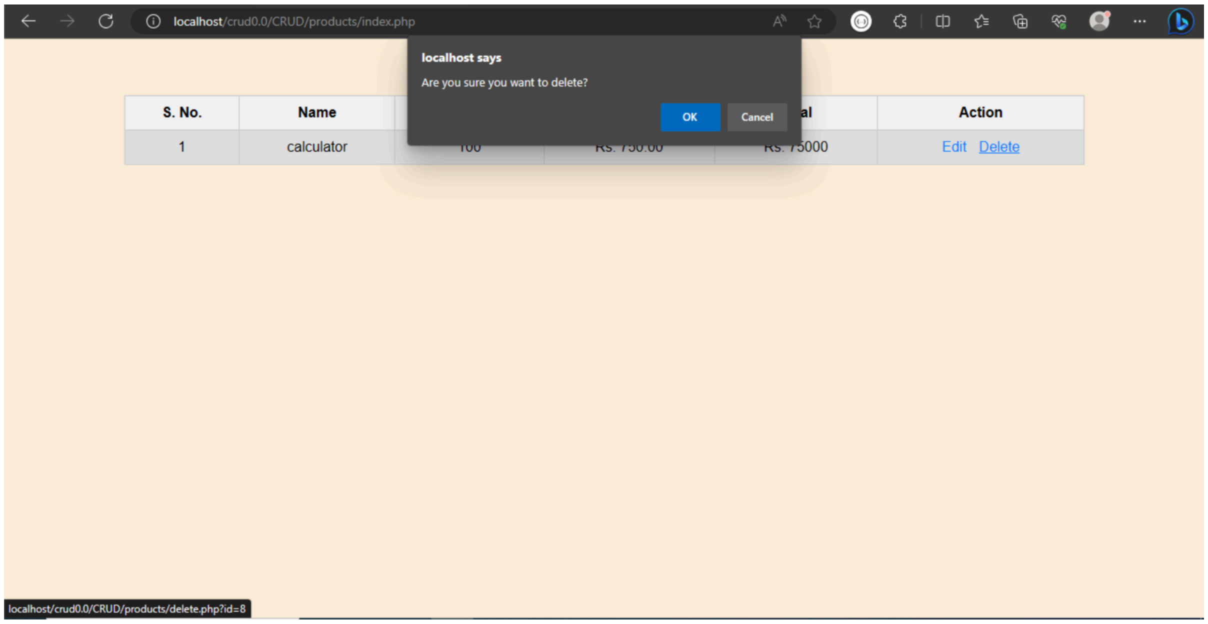
Description: View the data for editing and deleting records.

Create new detail:



Description: Record to enter new data.

Delete the record


localhost/crud0.0/CRUD/products/index.php
OK
Cancel
Edit
Delete
localhost/crud0.0/CRUD/products/delete.php?id=8

localhost says

Are you sure you want to delete?

S. No.	Name			al	Action
1	calculator	100	Rs. 750.00	Rs. 75000	Edit Delete

Description: Confirmation before deleting the record.