### Synopsis

**Abstract:**

The Cotton Commission Agent System is a web-based platform designed to streamline the cotton trading process between farmers, commission agents, and buyers such as textile mills and wholesalers. The system ensures efficient, transparent, and secure transactions by digitizing cotton listings, order management, and payment handling. While commission agents negotiate trades and oversee quality control, the system requires all payments to be manually approved by the admin before order processing. Shipment tracking is handled manually by the farmer. This approach maintains security and transparency, while supporting rural digital inclusion.

**Introduction:**

India is one of the largest cotton producers, yet cotton farmers often face challenges in connecting with the right buyers and getting fair prices. Middlemen, or commission agents, play a significant role in facilitating this trade, but the process is traditionally manual, time-consuming, and prone to miscommunication. The Cotton Commission Agent System digitizes this workflow, enabling real-time cotton listing, negotiation, admin-verified payments, and order tracking — creating a seamless experience for all stakeholders, especially in rural and agricultural sectors.

**Objectives:**

· Digitize the cotton trading process between farmers, commission agents, and buyers.

· Enable farmers to list cotton offers with quality, price, and quantity details.

· Facilitate commission agents in purchasing, listing, and reselling cotton to buyers.

· Provide a platform for buyers to search, place orders, and upload payment proof.

· Ensure admin-verified payment approval for order confirmation.

· Allow farmers to manually update order tracking.

· Generate sales, payment, and commission reports for tracking and transparency.

**Problem Statement:**   
Cotton trading in traditional markets suffers from inefficiencies such as lack of transparency in pricing, delayed payments, poor inventory tracking, and logistic challenges. A unified digital solution is needed where farmers can list their produce, agents can manage procurement and resale, and buyers can place and track orders — all with secure, admin-controlled payments and a basic but transparent logistics update system.

**Block Diagram/Project Flowchart with Short Explanation:** **Flowchart Overview:**

**[Farmer Registers & Lists Cotton]**  
The farmer registers on the platform and lists the available cotton with details like quality, quantity, and price.

**[Agent Buys Cotton → Pays Farmer]**  
The commission agent reviews the cotton listings, negotiates the price with the farmer, and then buys the cotton, paying the farmer.

**[Agent Lists Cotton for Buyers]**  
The agent lists the purchased cotton on the platform for buyers (such as textile mills or wholesalers) to browse and place orders.

**[Buyer Searches & Places Order]**  
The buyer searches for cotton based on the available listings, selects the type, and places an order specifying the quantity and quality needed.

**[Buyer Uploads Payment Proof]**  
Once the buyer places an order, they upload proof of payment for the cotton they have purchased.

**[Admin Verifies & Approves Payment]**  
The admin verifies the payment proof uploaded by the buyer and approves it for further processing.

**[Agent Ships Cotton to Buyer]**  
After the payment is confirmed by the admin, the agent arranges for the shipment of the cotton from the agent's warehouse or the farmer's location to the buyer.

**[Buyer Confirms Delivery]**  
Once the buyer receives the cotton, they confirm the delivery by ensuring the quantity and quality match the order.

**[Agent Gets Commission]**  
The commission agent receives their commission on the cotton sale for facilitating the transaction, including the shipment.

**[Reports & Records Generated]**  
The system generates reports on the sale, payment, commission, and delivery status for transparency and record-keeping.

**Benefits to Society:**

· **Empowers Farmers:** Direct platform access improves profitability and visibility.

· **Secure Payments:** Admin-approved payment flow reduces fraud and disputes.

· **Digital Inclusion:** Introduces farmers and agents to digital platforms.

· **Efficient Transactions:** Speeds up negotiation, order placement, and communication.

· **Transparency:** Tracks every trade and payment with audit trails and reports.

**Project requirements:**

## ****Hardware Requirements****

* **Processor :** Intel i5 or higher
* **RAM :** 4GB or more
* **Storage :** 20GB minimum

## ****Software Requirements****

* **Operating System :**  Windows/Linux/MacOS
* **Backend :**  PHP
* **Frontend :**  HTML, CSS, JavaScript, Bootstrap
* **Database :**  MySQL
* **Web Server :** XAMPP

**5. References:**

**Client Consultation:**

* Contacted **Shri Siddaroodha Traders** to understand the cotton trading process, market challenges, and pricing factors.

Web resouce:

* <https://chatgpt.com/c/67e6a5d7-b970-8012-89a3-dc1eb420af84>
* chrome-extension://kdpelmjpfafjppnhbloffcjpeomlnpah/https://www.cottonworks.com/wp-content/uploads/2017/11/Using-the-Cotton-Management-System.pdf
* <https://indiantextilejournal.com/cotton-management-system-process-control-a-case-study/>

### 2. INTRODUCTION

The **Cotton Management System** is a web-based application built to streamline the cotton trading process between farmers, commission agents, and buyers like textile mills and wholesalers. This system ensures transparency, digital inclusion, and security through manual admin approval of payments and real-time listing of cotton stocks. It facilitates secure negotiations, order management, and reporting while enabling rural farmers to participate in the digital economy.

India, being one of the largest producers of cotton, faces challenges in terms of pricing irregularities, lack of access to markets for farmers, and an inefficient distribution network. Traditional cotton trading systems rely on physical marketplaces and intermediaries, which often lead to unfair trade practices and delayed transactions. The Cotton Management System seeks to replace this outdated mechanism with a structured, digitized process that empowers all stakeholders.

### 3. SYSTEM ANALYSIS

#### Existing System

Manual trading process.

Lack of transparency in pricing.

Delayed or fraudulent payments.

Poor communication and logistics tracking.

Limited data storage or reporting mechanisms.

No real-time updates or order statuses.

#### Proposed System

Online cotton listing by agents with detailed input fields for quality, price, and quantity.

A user-friendly interface for buyers to search and filter listings.

Buyer-side ordering with secure upload of payment proof.

Admin-side manual payment verification and status update.

Shipment tracking manually updated by farmers with order confirmation options.

Auto-generation of reports for all stakeholders.

#### Feasibility Study

**Technical:** The system is built using widely adopted open-source technologies (PHP, MySQL, Bootstrap) ensuring scalability and robustness.

**Operational:** Agents and buyers can adapt to the web interface easily, and minimal training can help farmers understand shipment updates.

**Economic:** Cost-effective deployment using a local server (XAMPP) ensures minimal investment for rural centers.

#### Requirement Gathering

Interviews with farmers and agents.

Review of existing cotton trading software.

Client consultation with Shri Siddaroodha Traders.

Online sources and case studies on cotton logistics and process control.

### 4. SYSTEM DESIGN

#### Flow Diagram Explanation

**Farmer Registration:** Farmers share their produce details with agents.

**Agent Listing:** Agents create listings with price and cotton details.

**Buyer Search:** Buyers browse available stock.

**Order Placement:** Buyers order cotton with preferred quantity.

**Payment Upload:** Proof of payment is uploaded.

**Admin Approval:** Admin verifies and approves payment.

**Shipment Processing:** Agents handle logistics.

**Buyer Acknowledgment:** Buyers confirm receipt.

**Report Generation:** Admin and agent dashboards show records.

#### ER Diagram

**Entities:** Farmer, Agent, Buyer, Admin, Cotton, Payment, Order, Shipment, Commission

**Relationships:**

Agent buys from Farmer

Agent lists Cotton

Buyer places Orders

Payments are linked to Orders

Admin verifies Payments

Shipment is arranged by Agent

Commission is calculated post-delivery

#### Data Flow Diagram (DFD)

**Level 0:** High-level process flow among major entities.

**Level 1:** Detailed transaction and payment workflows.

**Level 2:** Input and output mapping for each module.

#### UML Diagrams

**Use Case Diagrams** for Agent, Buyer, Admin.

**Sequence Diagrams** for Order Processing.

**Activity Diagrams** for Listing, Payment, and Delivery Workflows.

### 5. SYSTEM DEVELOPMENT

#### Frontend Technologies

**HTML5:** Page structure and forms.

**CSS3:** Layout styling and responsive design.

**Bootstrap 5:** For responsive grids, components, and modals.

**JavaScript:** Form validation and dynamic interactions.

#### Backend Technologies

**PHP 8:** Handles all business logic and dynamic interactions.

**MySQL 8:** Database engine to store all data entities.

**XAMPP:** Combines Apache, MySQL, and PHP for local testing.

#### Database Schema Overview

**Users Table:** Stores login data and roles.

**Cotton Table:** Contains details of available cotton.

**Orders Table:** Logs all orders placed.

**Payments Table:** Records payment proofs and statuses.

**Shipments Table:** Contains shipment tracking and delivery confirmations.

**Commissions Table:** Automatically calculates commissions.

#### Security Measures

Input sanitization to prevent SQL injection.

Password hashing using bcrypt.

Session validation for secure logins.

### 6. SYSTEM IMPLEMENTATION

#### Installation Guide

Install XAMPP and start Apache & MySQL.

Copy project folder to htdocs directory.

Create database cotton\_system in phpMyAdmin.

Import cotton\_system.sql file.

Open localhost/cotton\_system in browser.

#### System Modules

**Authentication Module:** Login and registration for all user types.

**Cotton Listing Module:** Used by agents to list cotton.

**Order Management Module:** Allows buyers to place and track orders.

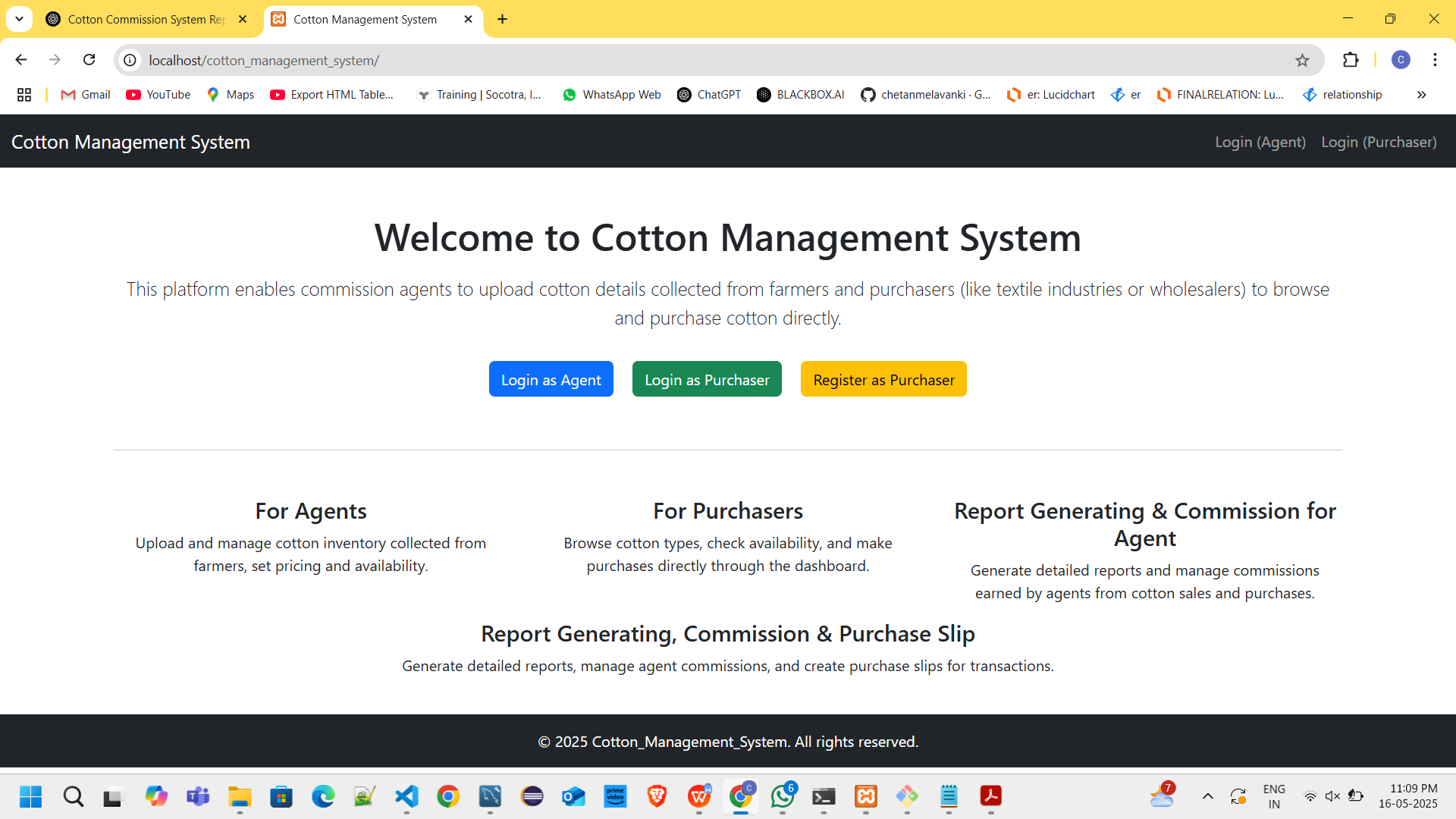
**Payment Verification Module:** Admin-side interface to approve or reject payments.

**Report Generation Module:** Auto-generated monthly sales and commission reports.

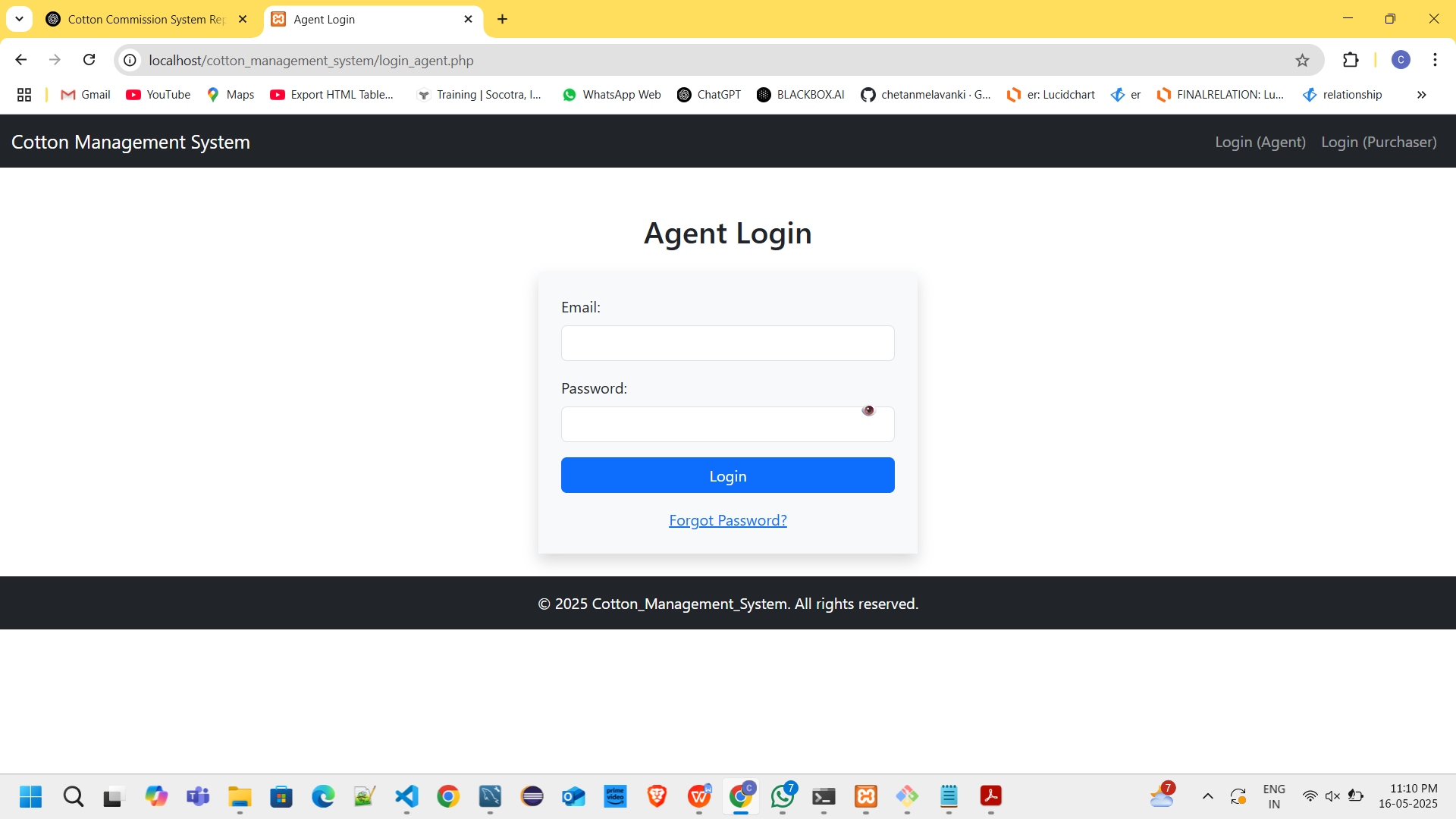
**Profile Management Module:** Update user details and contact info.

### SCREENSHOTS

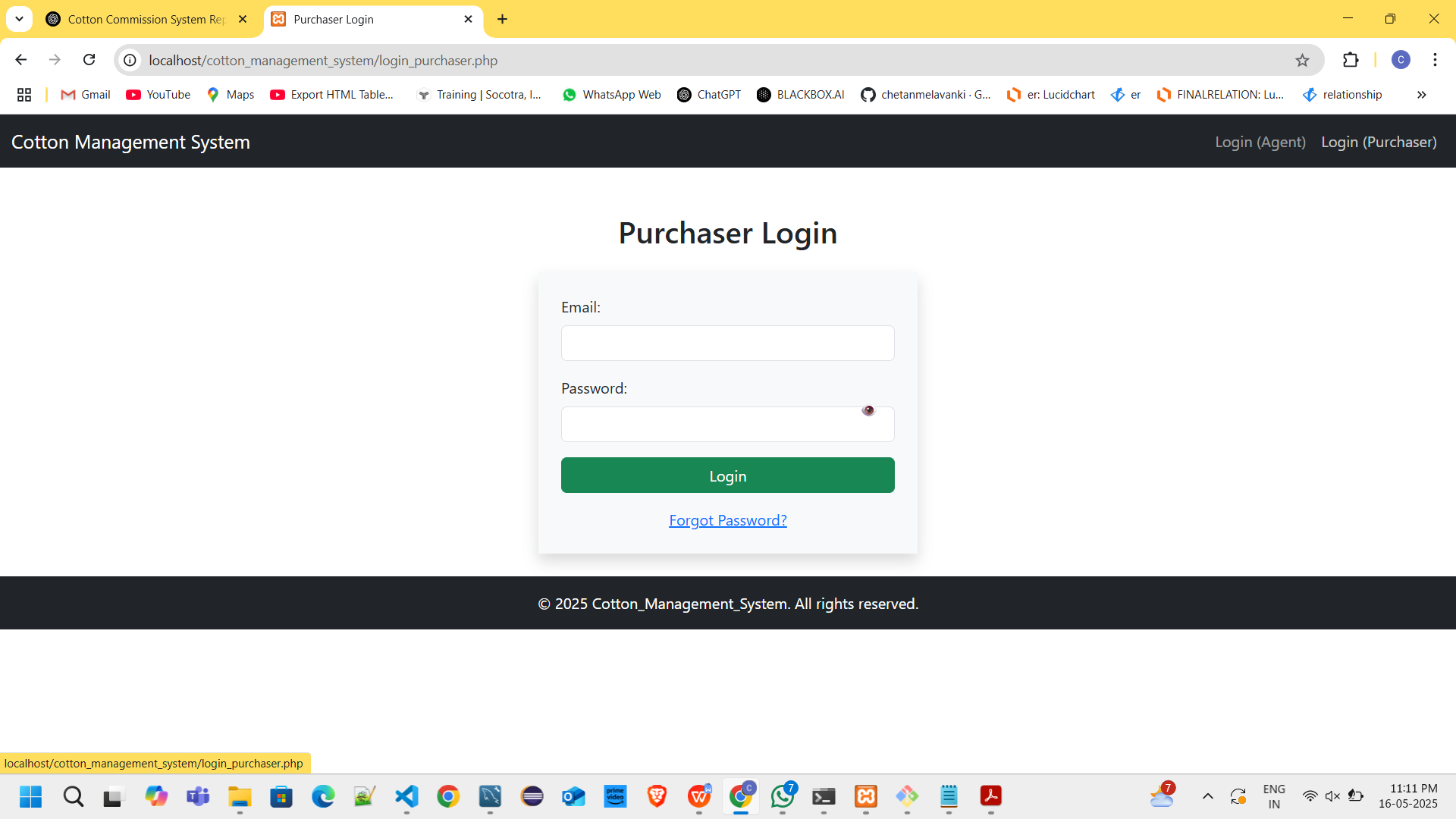
1. Index Page



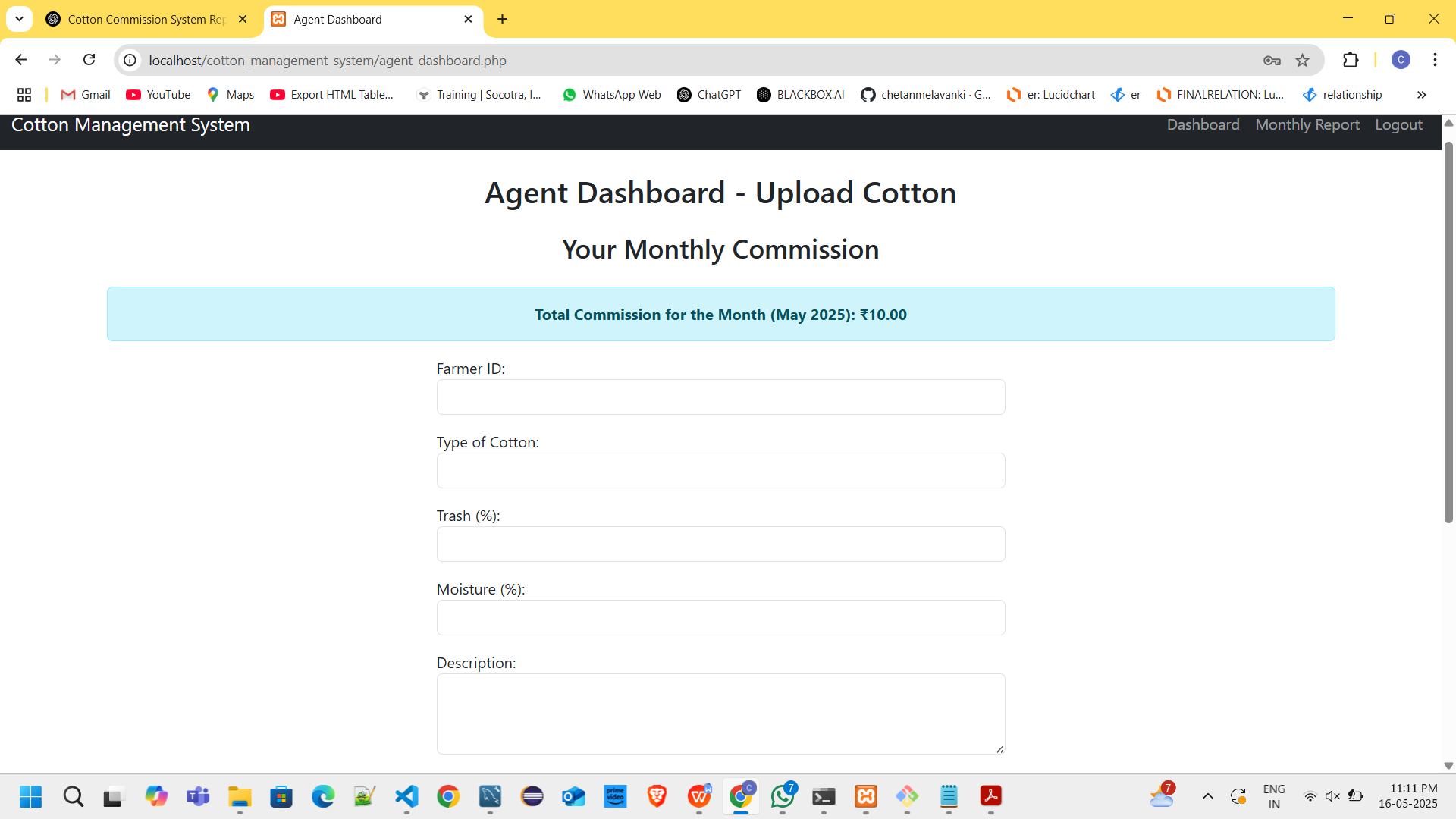
1. Agent Login



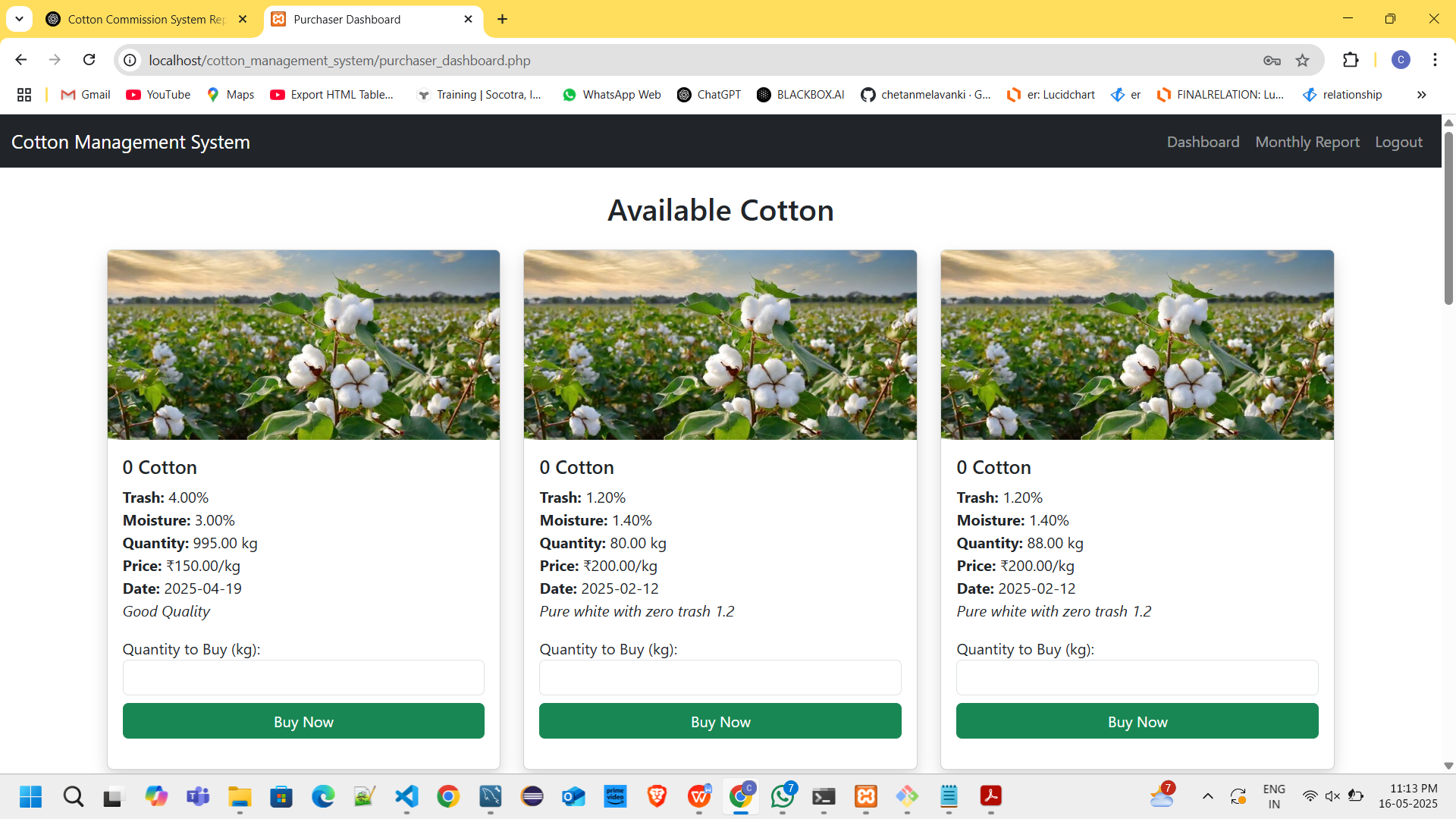
1. Purchaser Login



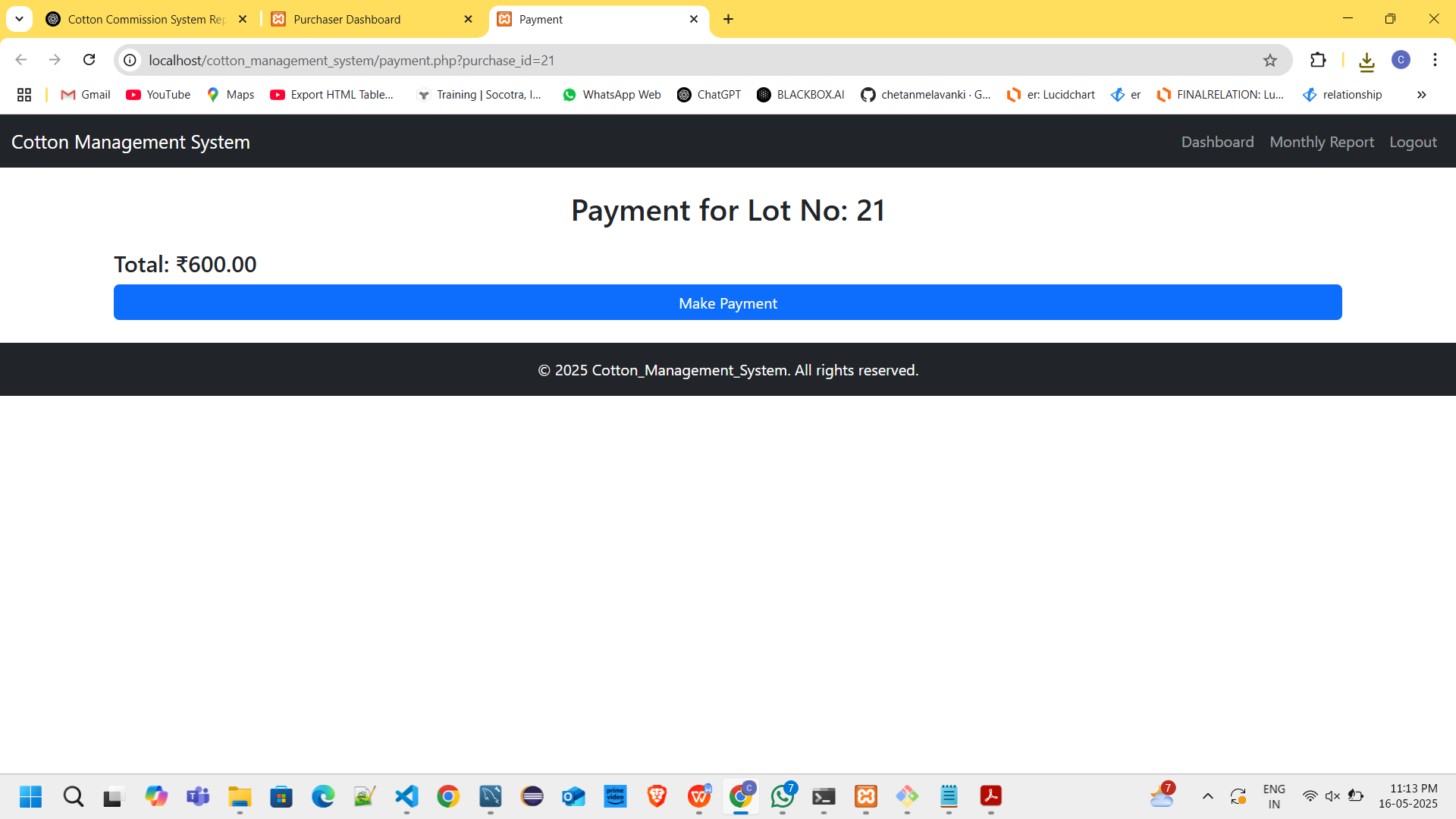
1. Agent Dashboard



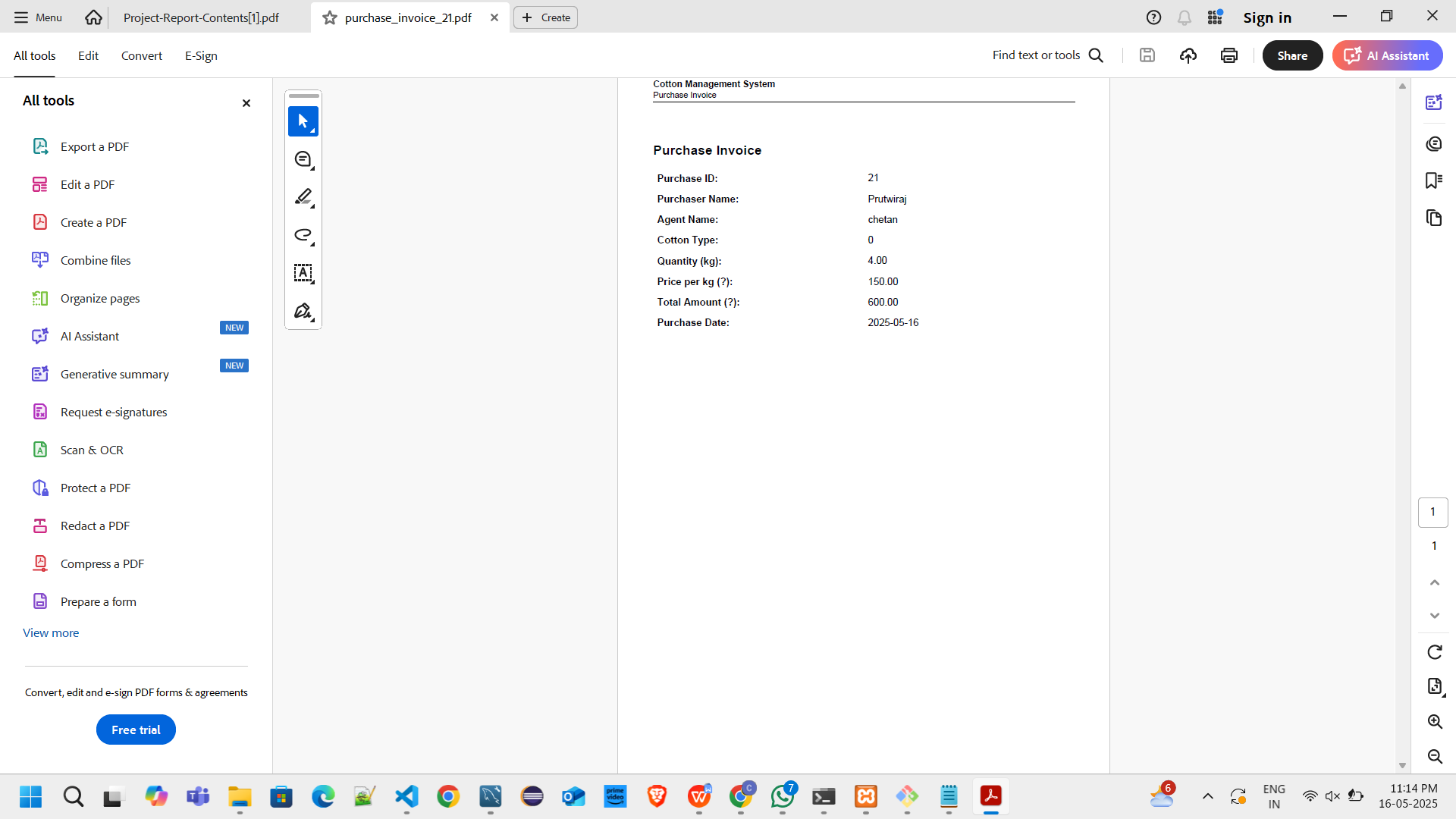
1. Purchaser Dashboard



1. Order Placement Screen



1. Report Download Screen



### 8. TESTING

#### Types of Testing

**Unit Testing:** Tested each PHP module independently.

**Integration Testing:** End-to-end payment flow and order fulfillment.

**System Testing:** Ensured the entire system performs according to specification.

**Regression Testing:** Ensured no existing module was affected after adding new features.

**User Acceptance Testing (UAT):** Conducted with a group of agents and buyers.

#### Sample Test Cases

TC01: Successful login with valid credentials.

TC02: Agent lists new cotton successfully.

TC03: Buyer places an order and uploads payment.

TC04: Admin verifies payment.

TC05: Agent marks shipment status.

TC06: Buyer confirms delivery.

### 9. LIMITATIONS

* Admin verification introduces delay.
* Shipment tracking is dependent on manual updates.
* Limited to cotton trade; not expandable to other crops without redesign.
* No automated fraud detection for fake payment uploads.\

### 10. FUTURE SCOPE & ENHANCEMENT

* Add support for real-time GPS shipment tracking.
* Integrate SMS/WhatsApp-based alerts for rural users.
* Expand platform to handle multiple crops.
* Introduce automatic invoice generation and GST calculation.
* Mobile App version for Android/iOS.
* Multilingual support for regional users.
* Role-based access control for agents with sub-users.
* Feedback/rating system for buyers and agents.
* Integration with government cotton procurement schemes.

### 11. SOURCE CODE STRUCTURE

#### PHP Files

* db.php: Database connection handler.

<?php

$servername = "localhost";  *// Your database host, typically 'localhost'*

$username = "root";         *// Your database username, typically 'root' for local*

$password = "ROOT";             *// Your database password, typically empty for local*

$dbname = "cotton\_management\_system";  *// The name of your database*

*// Create connection*

$conn = new **mysqli**($servername, $username, $password, $dbname);

*// Check connection*

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

?>

* login\_agent.php: Agent login page.

<?php

**session\_start**();

include('db.php');

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    $stmt = $conn->**prepare**("SELECT \* FROM agent WHERE email = ?");

    $stmt->**bind\_param**("s", $email);

    $stmt->**execute**();

    $result = $stmt->**get\_result**();

    $agent = $result->**fetch\_assoc**();

    if ($agent && $password === $agent['password']) {

        $\_SESSION['agent\_id'] = $agent['agent\_id'];

        $\_SESSION['agent\_name'] = $agent['name'];

        $\_SESSION['user\_id'] = $agent['agent\_id'];

        $\_SESSION['user\_type'] = 'Agent';

**header**("Location: agent\_dashboard.php");

        exit;

    } else {

        $error = "Invalid email or password.";

    }

}

?>

<?php include('includes/navbar.php'); ?>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

  <meta *charset*="UTF-8">

  <title>Agent Login</title>

  <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" *rel*="stylesheet">

</head>

<body>

<div *class*="container mt-5">

  <h2 *class*="text-center mb-4">Agent Login</h2>

  <?php if (!**empty**($error)): ?>

    <div *class*="alert alert-danger text-center"><?= $error ?></div>

  <?php endif; ?>

  <form *method*="POST" *class*="mx-auto shadow p-4 bg-light" *style*="max-width: 400px;">

    <div *class*="mb-3">

      <label *for*="email" *class*="form-label">Email:</label>

      <input *type*="email" *name*="email" *id*="email" *class*="form-control" *required*>

    </div>

    <div *class*="mb-3 position-relative">

      <label *for*="password" *class*="form-label">Password:</label>

      <input *type*="password" *name*="password" *id*="password" *class*="form-control" *required*>

      <span *class*="position-absolute top-50 end-0 translate-middle-y me-3" *style*="cursor: pointer;" *onclick*="**togglePassword**()">👁️</span>

    </div>

    <button *class*="btn btn-primary w-100" *type*="submit">Login</button>

    <div *class*="text-center mt-3">

      <a *href*="forgot\_password.php">Forgot Password?</a>

    </div>

  </form>

</div>

<script>

function **togglePassword**() {

  const field = document.**getElementById**("password");

  field.type = field.type === "password" ? "text" : "password";

}

</script>

<?php include('includes/footer.php'); ?>

</body>

</html>

* login\_purchaser.php: Purchaser login page.

<?php

**session\_start**();

include('db.php');

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $email = $\_POST['email'];

    $password = $\_POST['password'];

    $stmt = $conn->**prepare**("SELECT \* FROM purchaser WHERE email = ?");

    $stmt->**bind\_param**("s", $email);

    $stmt->**execute**();

    $result = $stmt->**get\_result**();

    $purchaser = $result->**fetch\_assoc**();

    if ($purchaser && **password\_verify**($password, $purchaser['password'])) {

        $\_SESSION['purchaser\_id'] = $purchaser['purchaser\_id'];

        $\_SESSION['purchaser\_name'] = $purchaser['name'];

        $\_SESSION['user\_id'] = $purchaser['purchaser\_id'];

        $\_SESSION['user\_type'] = 'Purchaser';

**header**("Location: purchaser\_dashboard.php");

        exit;

    } else {

        $error = "Invalid email or password.";

    }

}

?>

<?php include('includes/navbar.php'); ?>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

  <meta *charset*="UTF-8">

  <title>Purchaser Login</title>

  <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" *rel*="stylesheet">

</head>

<body>

<div *class*="container mt-5">

  <h2 *class*="text-center mb-4">Purchaser Login</h2>

  <?php if (!**empty**($error)): ?>

    <div *class*="alert alert-danger text-center"><?= $error ?></div>

  <?php endif; ?>

  <form *method*="POST" *class*="mx-auto shadow p-4 bg-light" *style*="max-width: 400px;">

    <div *class*="mb-3">

      <label *for*="email" *class*="form-label">Email:</label>

      <input *type*="email" *name*="email" *id*="email" *class*="form-control" *required*>

    </div>

    <div *class*="mb-3 position-relative">

      <label *for*="password" *class*="form-label">Password:</label>

      <input *type*="password" *name*="password" *id*="password" *class*="form-control" *required*>

      <span *class*="position-absolute top-50 end-0 translate-middle-y me-3" *style*="cursor: pointer;" *onclick*="**togglePassword**()">👁️</span>

    </div>

    <button *class*="btn btn-success w-100" *type*="submit">Login</button>

    <div *class*="text-center mt-3">

      <a *href*="forgot\_password\_purchaser.php">Forgot Password?</a>

    </div>

  </form>

</div>

<script>

function **togglePassword**() {

  const field = document.**getElementById**("password");

  field.type = field.type === "password" ? "text" : "password";

}

</script>

<?php include('includes/footer.php'); ?>

</body>

</html>

* register\_purchaser.php: Purchaser registration.

<?php

**session\_start**();

include('db.php');

$success = "";

$error = "";

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $name = $\_POST['name'];

    $location = $\_POST['location'];

    $contact\_number = $\_POST['contact\_number'];

    $email = $\_POST['email'];

    $password = **password\_hash**($\_POST['password'], PASSWORD\_DEFAULT);

    $registered\_date = **date**('Y-m-d');

*// Check if email already exists*

    $check = $conn->**prepare**("SELECT \* FROM purchaser WHERE email = ?");

    $check->**bind\_param**("s", $email);

    $check->**execute**();

    $res = $check->**get\_result**();

    if ($res->num\_rows > 0) {

        $error = "Email already registered. Please login.";

    } else {

        $stmt = $conn->**prepare**("INSERT INTO purchaser (name, location, contact\_number, email, password, registered\_date) VALUES (?, ?, ?, ?, ?, ?)");

        $stmt->**bind\_param**("ssssss", $name, $location, $contact\_number, $email, $password, $registered\_date);

        if ($stmt->**execute**()) {

            $success = "Registration successful. You can now login.";

        } else {

            $error = "Registration failed. Try again.";

        }

    }

}

?>

<?php include('includes/navbar.php'); ?>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <title>Register Purchaser</title>

    <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" *rel*="stylesheet">

</head>

<body>

<div *class*="container mt-5">

    <h2 *class*="text-center">Purchaser Registration</h2>

    <?php if ($success): ?>

        <div *class*="alert alert-success text-center"><?= $success ?></div>

    <?php elseif ($error): ?>

        <div *class*="alert alert-danger text-center"><?= $error ?></div>

    <?php endif; ?>

    <form *method*="POST" *class*="mx-auto" *style*="max-width: 500px;">

        <div *class*="mb-3">

            <label>Name:</label>

            <input *type*="text" *name*="name" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Location:</label>

            <input *type*="text" *name*="location" *class*="form-control">

        </div>

        <div *class*="mb-3">

            <label>Contact Number:</label>

            <input *type*="text" *name*="contact\_number" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Email:</label>

            <input *type*="email" *name*="email" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Password:</label>

            <input *type*="password" *name*="password" *class*="form-control" *required*>

        </div>

        <button *type*="submit" *class*="btn btn-primary w-100">Register</button>

    </form>

</div>

<?php include('includes/footer.php'); ?>

</body>

</html>

* agent\_dashboard.php: Cotton listing and management.

<?php

**session\_start**();

include('db.php');

*// Redirect if not logged in as agent*

if (!**isset**($\_SESSION['agent\_id'])) {

**header**("Location: login\_agent.php");

    exit();

}

$agent\_id = $\_SESSION['agent\_id'];

$success = "";

$error = "";

*// Fetching total commission for the current month*

$current\_month = **date**('Y-m');  *// e.g., '2025-05'*

$commission\_query = $conn->**prepare**("SELECT **SUM**(agent\_commission) AS total\_commission

                                    FROM purchase

                                    WHERE agent\_id = ? AND DATE\_FORMAT(purchase\_date, '%Y-%m') = ?");

$commission\_query->**bind\_param**("is", $agent\_id, $current\_month);

$commission\_query->**execute**();

$commission\_result = $commission\_query->**get\_result**();

$commission\_data = $commission\_result->**fetch\_assoc**();

$total\_commission = $commission\_data['total\_commission'] ?? 0;

*// Handle cotton upload*

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

    $farmer\_id = $\_POST['farmer\_id'];

    $type = **htmlspecialchars**(**trim**($\_POST['type']));

    $trash = $\_POST['trash'];

    $moisture = $\_POST['moisture'];

    $description = **mysqli\_real\_escape\_string**($conn, $\_POST['description']);

    $quantity = $\_POST['quantity'];

    $price\_per\_kg = $\_POST['price\_per\_kg'];

    $produce\_date = $\_POST['produce\_date'];

    if (!**DateTime**::**createFromFormat**('Y-m-d', $produce\_date)) {

        $error = "Invalid date format. Please use YYYY-MM-DD.";

    } else {

        if (**isset**($\_FILES['image']) && $\_FILES['image']['error'] === 0) {

            $img\_name = **basename**($\_FILES['image']['name']);

            $target = "assets/images/" . $img\_name;

            if (**move\_uploaded\_file**($\_FILES['image']['tmp\_name'], $target)) {

                $stmt = $conn->**prepare**("INSERT INTO cotton (

                    farmer\_id, agent\_id, type, trash, moisture, description,

                    quantity, price\_per\_kg, produce\_date, image\_url

                ) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

                $stmt->**bind\_param**(

                    "iissdssdss",

                    $farmer\_id,

                    $agent\_id,

                    $type,

                    $trash,

                    $moisture,

                    $description,

                    $quantity,

                    $price\_per\_kg,

                    $produce\_date,

                    $target

                );

                if ($stmt->**execute**()) {

                    $success = "Cotton details uploaded successfully.";

                } else {

                    $error = "Failed to upload cotton details. Error: " . $stmt->error;

                }

                $stmt->**close**();

            } else {

                $error = "Failed to upload image.";

            }

        } else {

            $error = "Image file is required.";

        }

    }

}

?>

<?php include('includes/navbar.php'); ?>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <title>Agent Dashboard</title>

    <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" *rel*="stylesheet">

</head>

<body>

<div *class*="container mt-4">

    <h2 *class*="text-center mb-4">Agent Dashboard - Upload Cotton</h2>

    <?php if ($success): ?>

        <div *class*="alert alert-success text-center"><?= $success ?></div>

    <?php elseif ($error): ?>

        <div *class*="alert alert-danger text-center"><?= $error ?></div>

    <?php endif; ?>

    <h3 *class*="text-center mb-4">Your Monthly Commission</h3>

    <div *class*="alert alert-info text-center">

        <strong>Total Commission for the Month (<?= **date**('F Y') ?>): ₹<?= **number\_format**($total\_commission, 2) ?></strong>

    </div>

    <form *method*="POST" *enctype*="multipart/form-data" *class*="mx-auto" *style*="max-width: 600px;">

        <div *class*="mb-3">

            <label>Farmer ID:</label>

            <input *type*="number" *name*="farmer\_id" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Type of Cotton:</label>

            <input *type*="text" *name*="type" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Trash (%):</label>

            <input *type*="number" *step*="0.01" *name*="trash" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Moisture (%):</label>

            <input *type*="number" *step*="0.01" *name*="moisture" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Description:</label>

            <textarea *name*="description" *class*="form-control" *rows*="3" *required*></textarea>

        </div>

        <div *class*="mb-3">

            <label>Quantity (kg):</label>

            <input *type*="number" *step*="0.01" *name*="quantity" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Price per kg (₹):</label>

            <input *type*="number" *step*="0.01" *name*="price\_per\_kg" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Produce Date:</label>

            <input *type*="date" *name*="produce\_date" *class*="form-control" *required*>

        </div>

        <div *class*="mb-3">

            <label>Cotton Image:</label>

            <input *type*="file" *name*="image" *class*="form-control" *accept*="image/\*" *required*>

        </div>

        <button *type*="submit" *class*="btn btn-primary w-100">Upload Cotton</button>

    </form>

</div>

<?php include('includes/footer.php'); ?>

</body>

</html>

* purchaser\_dashboard.php: Order and payment interface.

<?php

**session\_start**();

include('db.php');

if (!**isset**($\_SESSION['purchaser\_id'])) {

**header**("Location: login\_purchaser.php");

    exit();

}

$purchaser\_id = $\_SESSION['purchaser\_id'];

$success = "";

$error = "";

*// Handle Purchase*

if ($\_SERVER['REQUEST\_METHOD'] === 'POST' && **isset**($\_POST['cotton\_id'])) {

    $cotton\_id = $\_POST['cotton\_id'];

    $agent\_id = $\_POST['agent\_id'];

    $quantity = $\_POST['quantity'];

    $price = $\_POST['price\_per\_kg'];

    $total = $quantity \* $price;

    $lot\_number = **uniqid**('LOT');

    $purchase\_date = **date**('Y-m-d');

*// Calculate commission (5% agent commission as an example)*

    $agent\_commission = $total \* 0.05; *// 5% of the total amount*

*// Get current quantity*

    $q = $conn->**prepare**("SELECT quantity FROM cotton WHERE cotton\_id = ?");

    $q->**bind\_param**("i", $cotton\_id);

    $q->**execute**();

    $qResult = $q->**get\_result**();

    $currentCotton = $qResult->**fetch\_assoc**();

    if (!$currentCotton) {

        $error = "Cotton record not found.";

    } elseif ($quantity > $currentCotton['quantity']) {

        $error = "Requested quantity exceeds available stock.";

    } else {

        $current\_quantity = $currentCotton['quantity'];

        $new\_quantity = $current\_quantity - $quantity;

*// Insert into purchase table including agent commission*

        $stmt = $conn->**prepare**("INSERT INTO purchase (cotton\_id, purchaser\_id, agent\_id, lot\_number, purchase\_date, quantity, price, total\_amount, payment\_status, agent\_commission)

            VALUES (?, ?, ?, ?, ?, ?, ?, ?, 'Pending', ?)");

        $stmt->**bind\_param**("iiissdddi", $cotton\_id, $purchaser\_id, $agent\_id, $lot\_number, $purchase\_date, $quantity, $price, $total, $agent\_commission);

        if ($stmt->**execute**()) {

            $purchase\_id = $stmt->insert\_id;

*// Update cotton quantity and status*

            if ($new\_quantity <= 0) {

                $update = $conn->**prepare**("UPDATE cotton SET quantity = 0, status = 'Sold' WHERE cotton\_id = ?");

                $update->**bind\_param**("i", $cotton\_id);

            } else {

                $update = $conn->**prepare**("UPDATE cotton SET quantity = ? WHERE cotton\_id = ?");

                $update->**bind\_param**("ii", $new\_quantity, $cotton\_id);

            }

            $update->**execute**();

            $success = "Purchase successful! Lot No: $lot\_number.

            <a href='payment.php?purchase\_id=$purchase\_id' class='btn btn-primary btn-sm ms-2' target='\_blank'>Proceed to Payment</a>";

        } else {

            $error = "Purchase failed.";

        }

    }

}

*// Get available cotton*

$result = $conn->**query**("SELECT \* FROM cotton WHERE status = 'Available' OR quantity > 0 ORDER BY produce\_date DESC");

?>

<?php include('includes/navbar.php'); ?>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <title>Purchaser Dashboard</title>

    <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" *rel*="stylesheet">

</head>

<body>

<div *class*="container mt-4">

    <h2 *class*="text-center mb-4">Available Cotton</h2>

    <?php if ($success): ?>

        <div *class*="alert alert-success"><?= $success ?></div>

    <?php elseif ($error): ?>

        <div *class*="alert alert-danger"><?= $error ?></div>

    <?php endif; ?>

    <div *class*="row">

        <?php while ($row = $result->**fetch\_assoc**()): ?>

            <?php if ($row['quantity'] > 0): ?>

                <div *class*="col-md-4 mb-4">

                    <div *class*="card h-100 shadow">

                        <?php if ($row['image\_url']): ?>

                            <img *src*="<?= **htmlspecialchars**($row['image\_url']) ?>" *class*="card-img-top" *style*="height: 200px; object-fit: cover;">

                        <?php endif; ?>

                        <div *class*="card-body">

                            <h5 *class*="card-title"><?= **htmlspecialchars**($row['type']) ?> Cotton</h5>

                            <p *class*="card-text">

                                <strong>Trash:</strong> <?= $row['trash'] ?>%<br>

                                <strong>Moisture:</strong> <?= $row['moisture'] ?>%<br>

                                <strong>Quantity:</strong> <?= $row['quantity'] ?> kg<br>

                                <strong>Price:</strong> ₹<?= $row['price\_per\_kg'] ?>/kg<br>

                                <strong>Date:</strong> <?= $row['produce\_date'] ?><br>

                                <em><?= **htmlspecialchars**($row['description']) ?></em>

                            </p>

                            <form *method*="POST">

                                <input *type*="hidden" *name*="cotton\_id" *value*="<?= $row['cotton\_id'] ?>">

                                <input *type*="hidden" *name*="agent\_id" *value*="<?= $row['agent\_id'] ?>">

                                <input *type*="hidden" *name*="price\_per\_kg" *value*="<?= $row['price\_per\_kg'] ?>">

                                <div *class*="mb-2">

                                    <label>Quantity to Buy (kg):</label>

                                    <input *type*="number" *name*="quantity" *class*="form-control" *min*="1" *max*="<?= $row['quantity'] ?>" *required*>

                                </div>

                                <button *type*="submit" *class*="btn btn-success w-100">Buy Now</button>

                            </form>

                        </div>

                    </div>

                </div>

            <?php endif; ?>

        <?php endwhile; ?>

    </div>

</div>

<?php include('includes/footer.php'); ?>

</body>

</html>

* logout.php: Session termination script.

<?php

**session\_start**();

**session\_unset**();

**session\_destroy**();

**header**("Location: index.php");

exit;

* report.php: Monthly sales and commission data.

<?php

**session\_start**();

include('db.php');

*// Simulated login check*

*// Replace this logic with your actual login system*

if (!**isset**($\_SESSION['user\_type']) || !**isset**($\_SESSION['user\_id'])) {

**echo** "Access denied.";

    exit();

}

$userType = $\_SESSION['user\_type']; *// 'purchaser' or 'agent'*

$userId = $\_SESSION['user\_id'];

$month = **date**('m');

$year = **date**('Y');

if ($userType === 'purchaser') {

    $stmt = $conn->**prepare**("SELECT p.purchase\_id, c.type, p.quantity, p.price, p.total\_amount, p.purchase\_date, a.name AS agent\_name

                            FROM purchase p

                            JOIN cotton c ON p.cotton\_id = c.cotton\_id

                            JOIN agent a ON p.agent\_id = a.agent\_id

                            WHERE p.purchaser\_id = ? AND **MONTH**(p.purchase\_date) = ? AND **YEAR**(p.purchase\_date) = ?");

} else {

    $stmt = $conn->**prepare**("SELECT p.purchase\_id, c.type, p.quantity, p.price, p.total\_amount, p.purchase\_date, pu.name AS purchaser\_name

                            FROM purchase p

                            JOIN cotton c ON p.cotton\_id = c.cotton\_id

                            JOIN purchaser pu ON p.purchaser\_id = pu.purchaser\_id

                            WHERE p.agent\_id = ? AND **MONTH**(p.purchase\_date) = ? AND **YEAR**(p.purchase\_date) = ?");

}

$stmt->**bind\_param**("iii", $userId, $month, $year);

$stmt->**execute**();

$result = $stmt->**get\_result**();

?>

<?php include('includes/navbar.php'); ?>

<!DOCTYPE *html*>

<html *lang*="en">

<head>

    <meta *charset*="UTF-8">

    <title>Monthly Report</title>

    <link *href*="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" *rel*="stylesheet">

</head>

<body>

<div *class*="container mt-4">

    <h2 *class*="mb-4 text-center"><?= **ucfirst**($userType) ?> - Monthly Report (<?= **date**('F Y') ?>)</h2>

    <table *class*="table table-bordered table-striped">

        <thead *class*="table-dark">

            <tr>

                <th>Purchase ID</th>

                <th>Cotton Type</th>

                <th>Quantity (kg)</th>

                <th>Price per Kg (₹)</th>

                <th>Total Amount (₹)</th>

                <th>Purchase Date</th>

                <th><?= $userType === 'purchaser' ? 'Agent' : 'Purchaser' ?></th>

            </tr>

        </thead>

        <tbody>

            <?php if ($result->num\_rows > 0): ?>

                <?php while ($row = $result->**fetch\_assoc**()): ?>

                <tr>

                    <td><?= $row['purchase\_id'] ?></td>

                    <td><?= **htmlspecialchars**($row['type']) ?></td>

                    <td><?= $row['quantity'] ?></td>

                    <td><?= $row['price'] ?></td>

                    <td><?= $row['total\_amount'] ?></td>

                    <td><?= $row['purchase\_date'] ?></td>

                    <td><?= $userType === 'purchaser' ? $row['agent\_name'] : $row['purchaser\_name'] ?></td>

                </tr>

                <?php endwhile; ?>

            <?php else: ?>

                <tr><td *colspan*="7" *class*="text-center">No transactions found for this month.</td></tr>

            <?php endif; ?>

        </tbody>

    </table>

</div>

<?php include('includes/footer.php'); ?>

</body>

</html>

#### Folder Structure

* /css/ – Stylesheets
* /js/ – JavaScript functions
* /uploads/ – Payment proofs
* /includes/ – Shared PHP files
* /reports/ – PDF/CSV report downloads

### 12. BIBLIOGRAPHY

* Shri Siddaroodha Traders (Client Interaction)
* Indian Textile Journal
* CottonWorks – Process Control & System Study
* ChatGPT-generated flow, structure, and analysis: <https://chatgpt.com/c/67e6a5d7-b970-8012-89a3-dc1eb420af84>
* Chrome Extension: cottonworks.com CMS PDF