**A Project Report**

**on**

***“WasteNot: Serving Underserved Communities”***

Submitted in partial fulfillment of the requirements

for the award of the degree of Bachelor of Science

Information Technology (Hons.)

AKS UNIVERSITY SATNA

**BSC (IT)Hons. 4th Semester**

#### Submitted by:

##### Aman Yadav (B2392R13600006)

##### Shivam Verma (B2392R13600010)

**Under The Guidance of: - Approved by:-**

**Mrs. Shruti Gupta Prof. Akhilesh A. Waoo**

**(Assistant Professor) (Head of Department CS/IT)**

**Department of Computer Science & Engineering**

**AKS University Satna (M.P.)**

## CERTIFICATE

This certify that the project report entitled “***WasteNot***” submitted by partial fulfilment of the requirement for the degree of Bachelor of Computer Application (Hons.) **in Jan-June 2024-2025** AKS University, Satna is a bonafide project work carried out by **Aman Yadav , B2392R13600006 & Shivam Verma , B2392R13600010 ,** under my supervision. The subject of the project report has been approved by supervisor. This is also to certify that it is his/her original work and no part of this project is report has been submitted for any other degree/diploma.

All the assistance the and help received during the course of the investigation has been duly acknowledged.

1. I am satisfied that the report presented by **Aman Yadav & Shivam Verma** is worthy of consideration for award of the degree.
2. I certify:- 
   1. That he/she pursued the prescribed course of research.
   2. That he/she bears good moral character.

**Place: AKSU, Satna Supervision**

**Date:…/…./………** Prof. Akhilesh A. Waoo

(Head of department CS/IT)

……………………………. …………………………….

**Under the Guidance of : External -**

Mrs. Shruti Gupta

(Assistant Professor)

**CERTIFICATE BY THE CANDIDATE**

I certify that the project report entitled “**WasteNot*”*** is my own work conducted under the supervision of Prof. Akhilesh A. Waoo, (Supervisor), Head of Department CS/IT, AKS University, Satna (M.P.) for partial fulfilment of the requirement for the degree in Bachelor of Science in Informational Technology (**Jan-June 2024-2025)**.

I further certify that to the best of my knowledge and belief the project report does not contain any part of this work which has been submitted for the award of any degree either in this university or in any other University / Deemed University / Institutes.

…………………………..

**Place: AKSU, Satna** …………………………..

**Date:** …./…../……… Signature of Candidates

Aman Yadav

Shivam Verma

## SELF DECLARATION

I hereby declare that the work presented in this project entitled **"WasteNot"** towards the partial fulfilment of the requirement for the award of **Degree in BSC (IT)Hons.** in Department of Computer Science, **AKS University, Satna (M.P.)** is an authentic record of my own work.

I have not submitted the matter embodied in the project for the award of any other degree or diploma to any other institute or university.

Aman Yadav (B2392R13600006)

Shivam Verma (B2392R13600010)

BSC (IT) Hons.

## ACKNOWLEDGEMENT

It is a great for me in taking this opportunity to express my sincere thanks and ineptness to **Prof. Akhilesh A. Waoo**, Head of the Department of CS/IT, AKS University, Satna (M.P.)

I consider myself lucky enough to have such a great project. This project would add as an asset to my profile.

At this moment of accomplishment, first of all I pay homage to my guide, **Mrs. Shruti Gupta** (Assistant Professor) from AKS University Satna (M.P.). This work would not have been possible without his guidance, support and encouragement. Under his guidance I successfully overcame many difficulties and learned a lot.

I am deeply and forever indebted to my parents for their love, support and encouragement throughout my entire life.

.

**ABSTRACT**

1. A responsive HTML/CSS/JavaScript frontend and Node.js/Express.js backend power WasteNot, with secure APIs and deployment on Vercel/Render, showcasing full-stack skills optimized for rural accessibility.
2. JavaScript drives real-time product filtering, cart updates, and donation form validation, with a Hindi/English UI in Mint Green and Coral Peach for an engaging rural user experience.
3. WasteNot utilizes Supabase for efficient database management, storing user details, product data, agent profiles, and donation records. APIs enable seamless data retrieval and updates, supporting features like real-time donation tracking and secure checkout with Razorpay integration. The system ensures scalability and data integrity, providing a reliable backbone for e-commerce and community support.

## TABLE OF CONTENT

* Introduction……...…………………………………………………………..7
* Objective of The Project…………………………………………………....8
* Scope of The Projects…..…………………………………………………...9
* Definition of Problem………………………………………………………11
* Benefit of The Projects …………………………………………………….13
* Main Module………………………………………………………………..14
* Technical Overview………………………………………………………...15
* Html………………………………………………………………………...15
* CSS…………………………………………………………………………16
* Java Script…………… …………………………………………….………17
* System Configuration…..…………………………………………………..18
* Block Diagram…………………………………………………….……….19
* ER Diagram …..……………………………………………………………20
* Output Screen …..………………………………………………………….21
* Codding……….……………..……………………………………………..31
* Software Testing …………………………………………………………..53
* Conclusion…………………………………………………………………56
* BIBLIOGRAPHY…………………………………………………………57

**INTRODUCTION**

WasteNot **:** Millions of people in rural areas struggle to access essential products due to a lack of proper e-commerce services and delivery networks. Most online shopping platforms focus on cities, leaving remote communities without affordable goods and reliable delivery options. **WasteNot** aims to solve this problem by creating a **smart, agent-based e-commerce and logistics platform** that connects rural customers with essential products.

The platform works through **local agents** who help customers place orders and receive deliveries. These agents also earn commissions, creating **new job opportunities** and supporting **small businesses** in rural areas. This model ensures that even those with **limited internet access or digital literacy** can benefit from online shopping.

**WasteNot** uses **AI-driven technology** to manage orders, track deliveries, and optimize inventory, making the supply chain more efficient. Customers can also **track their orders in real time** and choose from various **payment methods**, including digital wallets, UPI, and cash-on-delivery (COD).

The platform helps businesses expand their reach into rural areas without the need for costly infrastructure. Additionally, **WasteNot promotes sustainability** by encouraging companies to **donate surplus goods**, reducing waste while benefiting communities. It also optimizes delivery routes to **lower environmental impact.**

By combining **technology, local entrepreneurship, and smart logistics**, WasteNot makes e-commerce **accessible, efficient, and socially impactful.** It empowers rural communities, creates economic opportunities, and ensures that essential goods reach everyone, no matter where they live

**OBJECTIVE OF THE PROJECT**

WasteNot **:**  WastNot aims to **bridge the gap in rural e-commerce** by providing an **agent-based platform** that makes essential goods accessible to underserved communities. It **empowers local entrepreneurs**, optimizes **logistics with AI-driven efficiency**, and promotes **sustainability through surplus donations**, creating a scalable and impactful business model.

### **Key Objectives of WasteNot :**

1. **Rural E-Commerce Accessibility :** WasteNot ensures that people in rural and underserved areas can easily access essential goods through a **user-friendly platform and local agent network**, overcoming digital and logistical barriers.
2. Agent-Based Entrepreneurship : The platform **empowers local agents**, allowing them to assist customers in ordering products and earning commissions, fostering **economic growth and self-employment** in rural areas.
3. Smart Logistics & Supply Chain Optimization : By using **AI-driven demand forecasting and real-time tracking**, WasteNot optimizes supply chain operations, ensuring **faster deliveries, lower costs, and reduced waste.**
4. **Sustainability & Business Growth :**  WasteNot promotes **eco-friendly logistics** and encourages businesses to donate surplus goods, benefiting communities while offering **financial incentives and sustainable business expansion**.

**SCOPE OF THE PROJECT**

Here are **6 scopes** related to Snake game project:

The WasteNot platform is a web-based e-commerce and donation system designed to provide affordable products and essential item donations to rural and underserved communities. The project scope centers on creating an intuitive, inclusive experience with local agent-based delivery, ensuring accessibility, user engagement, and societal impact. The key areas of the project’s scope are as follows:

**User Interaction and Experience :** The platform enables users to register, log in, browse products, and donate items seamlessly via a responsive interface. Features like product filters, a cart, and donation tracking enhance engagement, while a vibrant UI (Mint Green, Coral Peach, Ivory White) and Hindi/English support ensure inclusivity for rural users. A community impact section showcasing donation success stories encourages participation

**Scalability :** The platform enables users to register, log in, browse products, and donate items seamlessly via a responsive interface. Features like product filters, a cart, and donation tracking enhance engagement, while a vibrant UI (Mint Green, Coral Peach, Ivory White) and Hindi/English support ensure inclusivity for rural users. A community impact section showcasing donation success stories encourages participation.

**Cross-Platform Accessibility:** WasteNot is accessible on desktops, tablets, and low-end mobile devices, ensuring rural users with limited connectivity (e.g., 2G/3G) can engage. The mobile-first, lightweight design with compressed assets and minimal animations delivers a smooth experience across screen sizes, making e-commerce and donations widely available.

**Data Management and Security :**  User, product, agent, and donation data are securely managed using Supabase, storing details like user profiles, product stock, and donation statuses. APIs with JWT authentication ensure secure transactions and data access. While Supabase meets current needs, future iterations could adopt dedicated databases for enhanced security and larger datasets.

**Web Development:** The project demonstrates core web development skills, including HTML/CSS/JavaScript for dynamic frontend features (e.g., real-time cart updates, donation forms) and Node.js/Express.js for robust APIs. Integration with Razorpay for UPI payments and local agent coordination showcases practical e-commerce and community-driven functionality, serving as a foundation for advanced web applications.

**Future Enhancements:** Potential upgrades include gamified user rewards (e.g., badges for frequent donations), real-time agent tracking, or social media integration for sharing donation milestones. Adding features like subscription-based product deliveries, multilingual video tutorials, or community forums could further boost engagement and impact in rural areas.

**DEFINITION OF PROBLEM**

Before the development of the WasteNot platform, several challenges and limitations in rural e-commerce and community support underscored the need for this project. These issues hindered access to affordable products and donation opportunities, necessitating a modern, accessible, and socially impactful solution.

**1.** **Limited Access to E-commerce in Rural Areas :**

* aditional e-commerce platforms are urban-centric, with high costs and complex interfaces unsuitable for rural users with low-end devices and limited internet (e.g., 2G/3G).
* WasteNot addresses this by offering a lightweight, mobile-first platform with affordable products tailored for rural needs, ensuring inclusivity and ease of use.

**2.**  **Lack of Donation Platforms for Underserved Communities** **:**

* Few platforms enable donations of primary need items (e.g., food, clothing) to underserved communities, with no user-friendly tracking or confirmation process.
* WasteNot introduces a simple donation system with real-time tracking and agent-based delivery, empowering users to support rural communities effectively.

**3. Inefficient Delivery Systems for Rural Regions :**

* Conventional delivery services struggle in remote areas due to logistical challenges, leaving rural users underserved.
* WasteNot integrates local agent-based entrepreneurs to coordinate deliveries, ensuring reliable access to products and donated items in rural locations.

.

**4. Inadequate Data Management for Community Platforms** **:**

* Managing user, product, agent, and donation data without a scalable database is challenging for community-focused platforms.
* By using Supabase, WasteNot provides a secure, cloud-based solution for storing and updating data, with potential for future scalability.

**5. Lack of Multilingual and Accessible Interfaces :**

* Most e-commerce platforms lack multilingual support (e.g., Hindi) and accessibility features, alienating rural users with limited digital literacy.
* WasteNot offers a Hindi/English interface with a vibrant, high-contrast UI (Mint Green, Coral Peach) and WCAG-compliant design for readability and usability.

**6. Absence of Modern Web Technology Integration :**

* Many rural-focused platforms rely on outdated technologies, missing opportunities to showcase modern web development.
* WasteNot leverages HTML/CSS/JavaScript, Node.js/Express.js, Supabase, and Razorpay to deliver a full-stack application, demonstrating dynamic features like real-time updates and secure payments.

In conclusion, the WasteNot platform addresses these challenges by building a modern, web-based e-commerce and donation system that prioritizes rural accessibility, community support, and technological innovation. It integrates user-friendly design, secure data management, and local delivery to empower rural users and foster social good.

**BENIFIT OF THE PROJECT**

WasteNot empowers rural users with affordable e-commerce and donations, fostering community impact and developer skills through modern web technologies.

1**. Inclusive Experience :**

* Hindi/English UI (Mint Green, Coral Peach) with easy signup, shopping, and donation tracking ensures rural accessibility.

2**. Community Impact :**

* Donation system enables contributions to underserved areas with real-time tracking, boosting social good.

3**. Web Accessibility :**

* HTML/CSS/JavaScript ensures access on low-end devices and slow networks, no apps needed.

4**. Secure Data :**

* Supabase and Razorpay manage user, product, and donation data securely and cost-effectively.

5**. Developer Skills :**

* Full-stack learning with Node.js, APIs, and Supabase for e-commerce and social platforms.

6**. Scalable Base :**

* Foundation for AI recommendations, agent tracking, or forums to expand rural outreach.

7**. Local Empowerment :**

* Agent-based delivery creates income for rural entrepreneurs, enhancing economic impact.

**MAIN MODULE**

* Home Page
* Features
* Product
* Agent
* Donation
* User signup / Logout

**TECHNICAL OVERVIEW**

**HTML:-**

1. HTML **:-** HTML is a markup language that defines the structure of your content. HTML consists of a series of elements, which you use to enclose, or wrap, different parts of the content to make it appear a certain way, or act a certain way. The enclosing tags can make a word or image hyperlink to somewhere else, can italicize words, can make the font bigger or smaller, and so on.
2. A web browser can read HTML files and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses them to interpret the content of the page. HTML describes it a mark-up language rather than a programming language.
3. What is HTML used for?
4. Structuring web pages:
5. With tags and elements, we can define the headings, paragraphs, and other contents of a web page. Browsers come with a built-in stylesheet that visually differentiates these elements. Content surrounded with tags <h1>will have a large, bolded font and its own line. Surrounding text with <h2>tags will make it bold and slightly smaller than the <h1> elements, and so on. Without these tags, the text on the web page would all look the same.

**Navigating the internet :** Navigating the internet would be much harder without HTML. Imagine having to manually enter the URL of every web page you wanted to visit. This would be our reality without anchor tags. HTML's anchor tags allow us to link pages to and from each other using the href attribute.

**Embedding images and videos :** HTML also allows you to not only embed images into a webpage but also adjust their width, height, position, and even the way they're rendered. Along with embedding videos, you can use other HTML attributes to adjust video controls, timestamps, thumbnails, autoplay, and more.

**Improving client-side data storage and offline capabilities:**  HTML also improved the language's offline capabilities. Websites use cookies to store data about users that can be retrieved later when they revisit the site. Cookies work well for small data like authentication tokens and usernames, but HTML5's addition of local Storage and Index DB enabled browsers to store more complex data.

**CSS:-**

1. CSS stands for Cascading Style Sheets.
2. CSS describes how HTML elements are to be displayed on screen, paper, or in other media.
3. CSS saves a lot of work.
4. It can control the layout of multiple web pages all at once.
5. External style sheets are stored in CSS files.
6. CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

**There are three types of CSS which are given below:**

* Inline CSS: Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style

attribute.

* Internal or Embedded CSS: This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e. the CSS is embedded within the HTML file.
* External CSS: External CSS contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading, …etc). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages

**Java Script :-**

JavaScript is a high-level, interpreted programming language pivotal for web development, enabling dynamic and interactive content. It’s single-threaded, leveraging an event loop for asynchronous operations using callbacks, Promises, and async/await. Built on the ECMAScript standard, it uses prototype-based inheritance, with ES6+ introducing classes, arrow functions, destructuring, and modules (import/export). JavaScript runs in browser engines (e.g., V8 in Chrome, SpiderMonkey in Firefox) and Node.js for server-side applications. It supports dynamic typing, first-class functions, and closures, making it versatile for functional and object-oriented paradigms. Key environments include browsers (DOM, Web APIs) and Node.js (file system, HTTP). Modern features include spread/rest operators, template literals, optional chaining, and nullish coalescing. It powers web frameworks (React, Vue.js), mobile apps (React Native), and desktop apps (Electron). Performance benefits from JIT compilation, but developers must avoid memory leaks and XSS vulnerabilities. Tools like npm, Babel, Webpack, and TypeScript enhance development.

Current trends involve WebAssembly integration, edge computing (e.g., Cloudflare Workers), and AI/ML with TensorFlow.js. Security requires careful handling of CORS, CSRF, and avoiding eval(). For in-depth learning, refer to MDN Web Docs or tc39.es for ECMAScript specs.

**Framework :-**

Express.js is a minimalist, flexible Node.js web framework for building server-side applications and APIs in JavaScript. It simplifies HTTP request handling, routing, and middleware integration.

## SYSTEM CONFIGURATION

### **Hardware Requirement**

Processor: Intel Core Due 2.0 GHz or more

RAM: 1 GB or more

Hard Disk: 80 GB or more

Monitor: 15” CRT, or LCD Monitor

Keyboard: Normal or Multimedia

Mouse: Compatible Mouse

### **Software Requirement**

Front end and back end: HTML, CSS, Java Script , express.js

Window: Window7, Window 8, Window10, Window 11

Data base: Supabase

Web Browser: Mozilla Firefox, Google Chrome and etc.

Addition: S/W Notepad++,Visual Studio Code.

**BLOCK DIAGRAM**

Login

Admin

Crime Scene

Social Crime

Crimes Forms

Fraud

Robbery

Logout

Murder

Sexual Complaint

User Information

Employ Information

User

**ER DIAGRAM**

MURDER

SEXUL COMPLAINT

FRAUD

FRAUD

Submit

View by

CRIME SCENE

SOCIAL CRIME

ROBBERY

View

DASH BOARD

EMPLOY

USER

update

Add

Add/ update

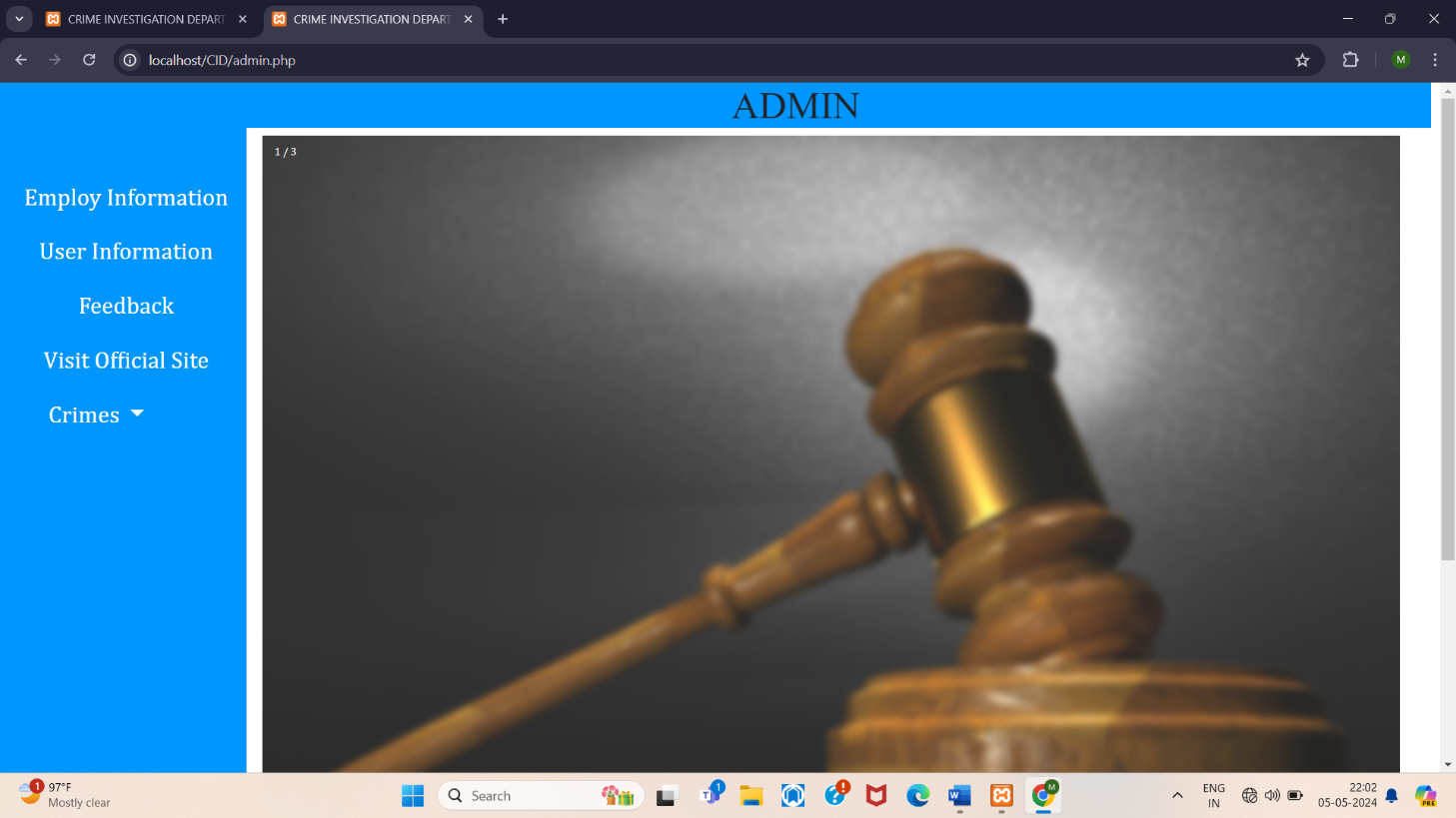
ADMIN

**OUTPUT SCREEN**

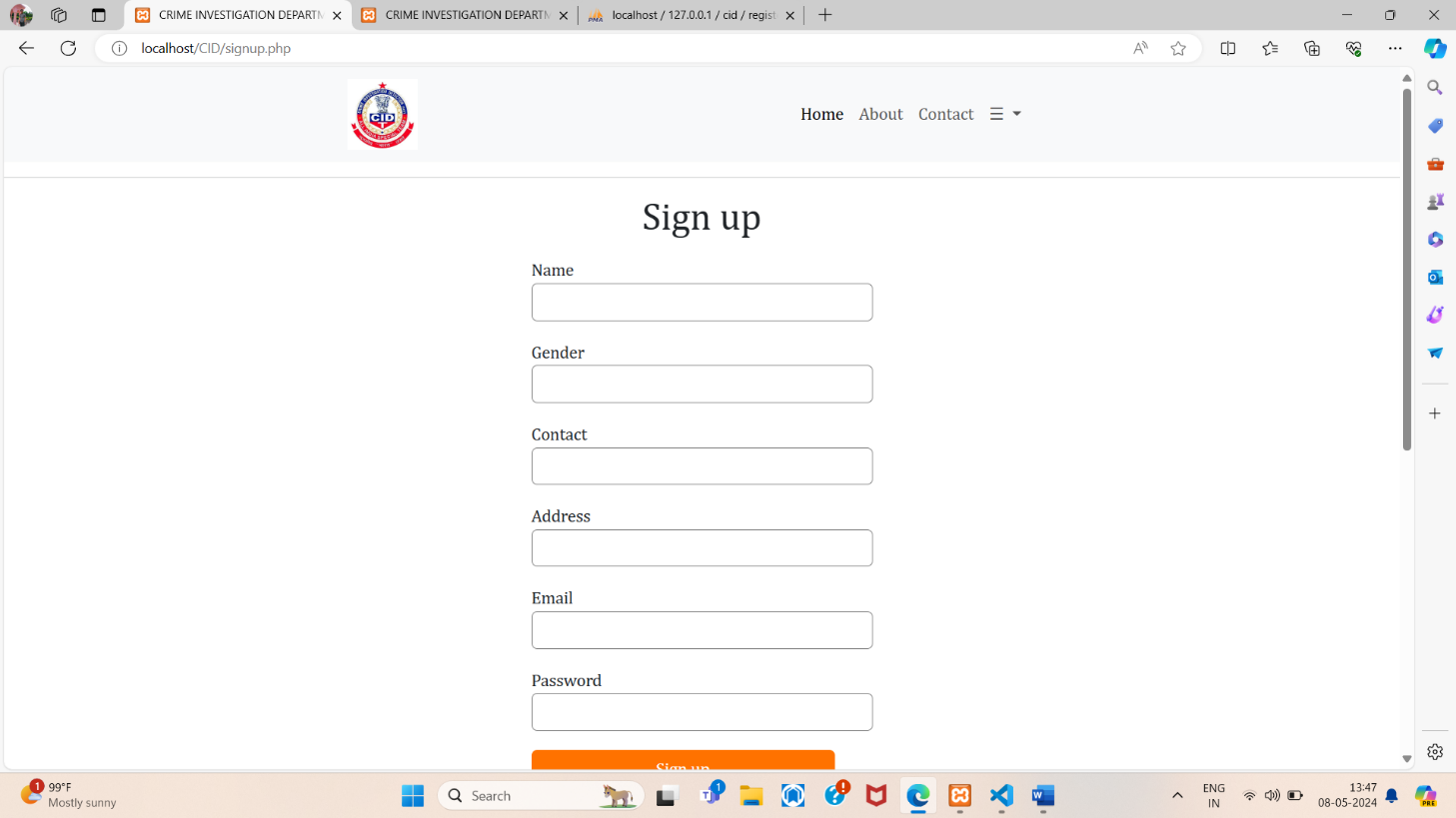
**HOME PAGE**



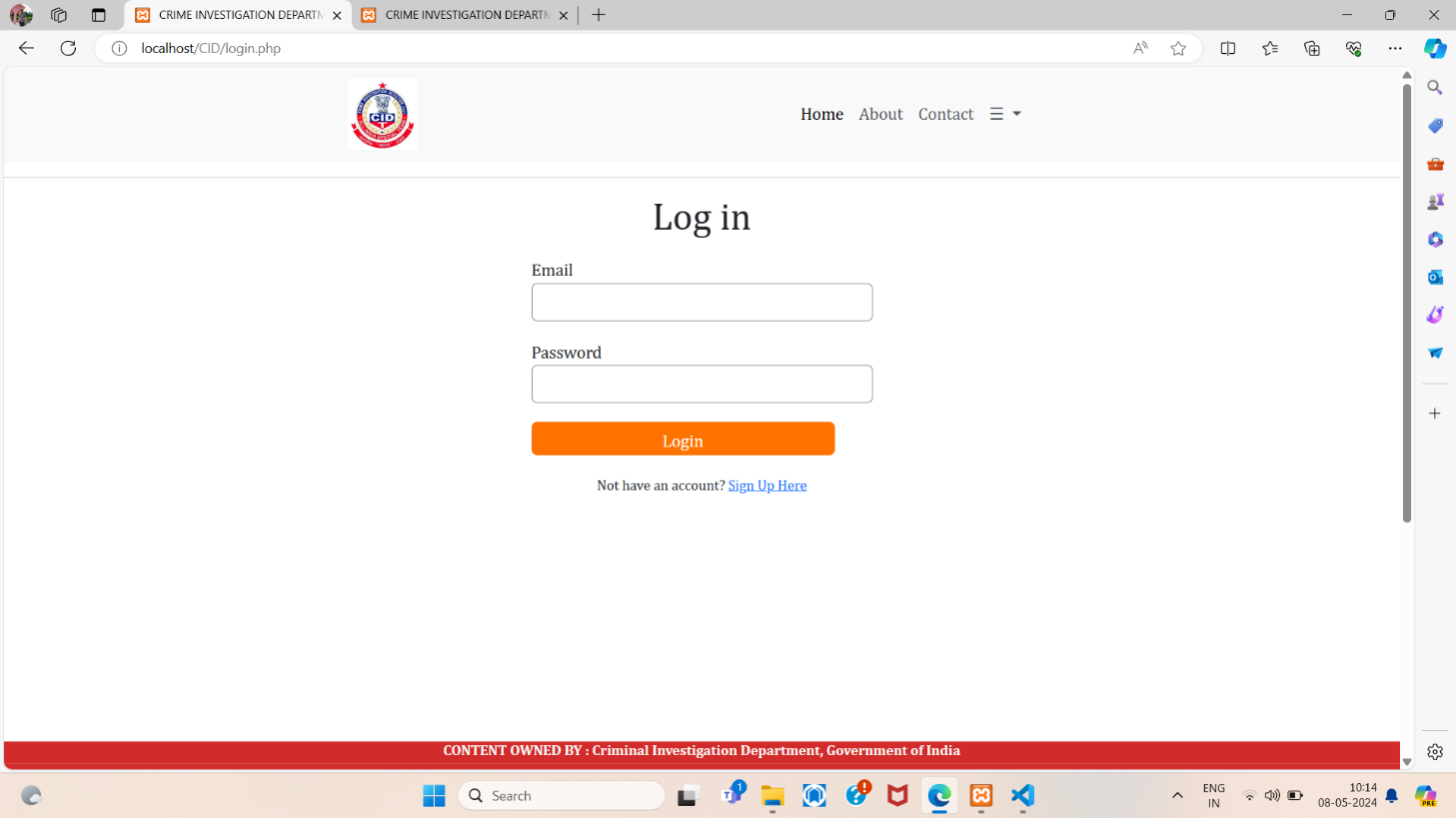
**ADMIN PAGE**



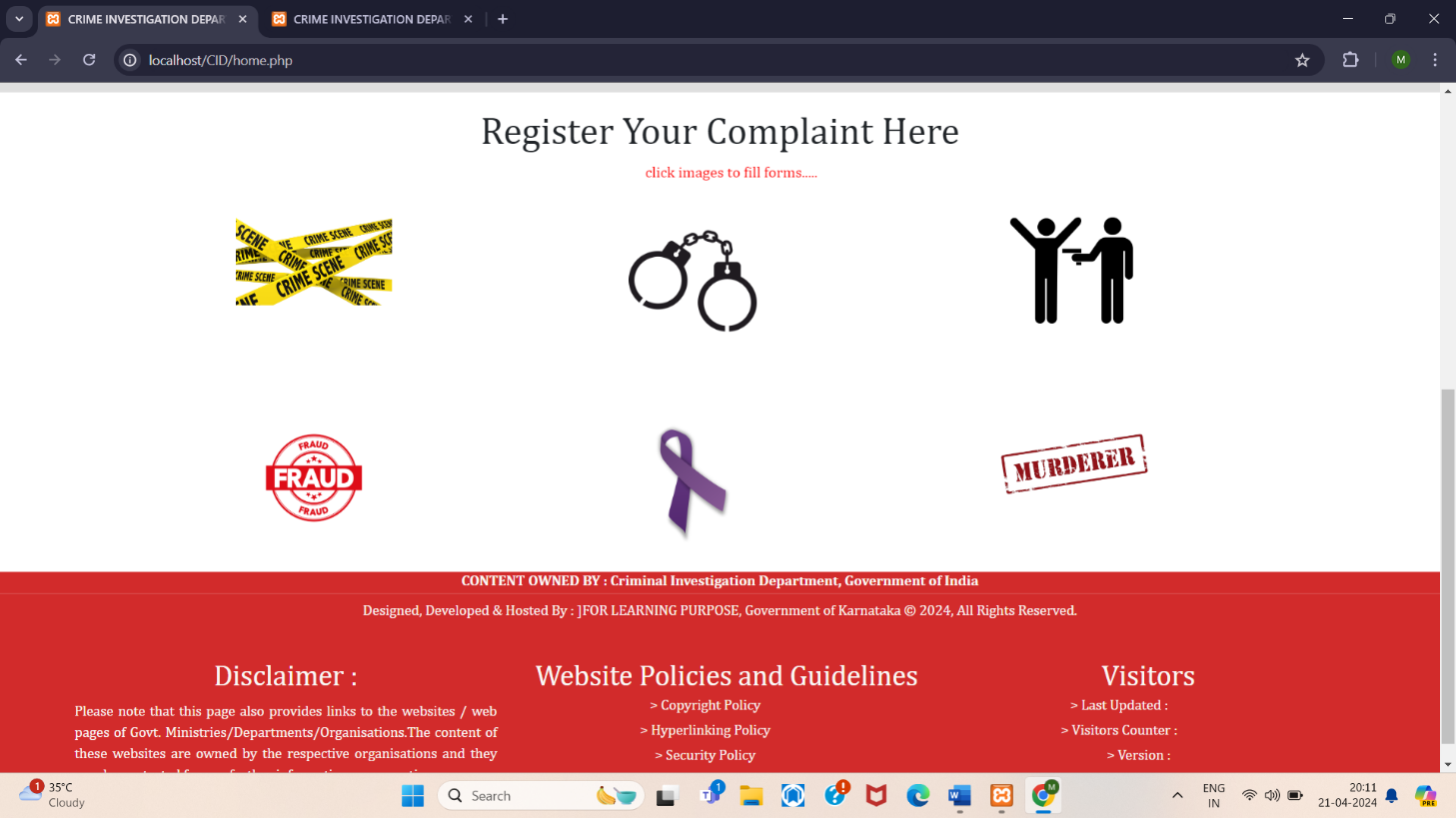
**USER SIGN UP PAGE**



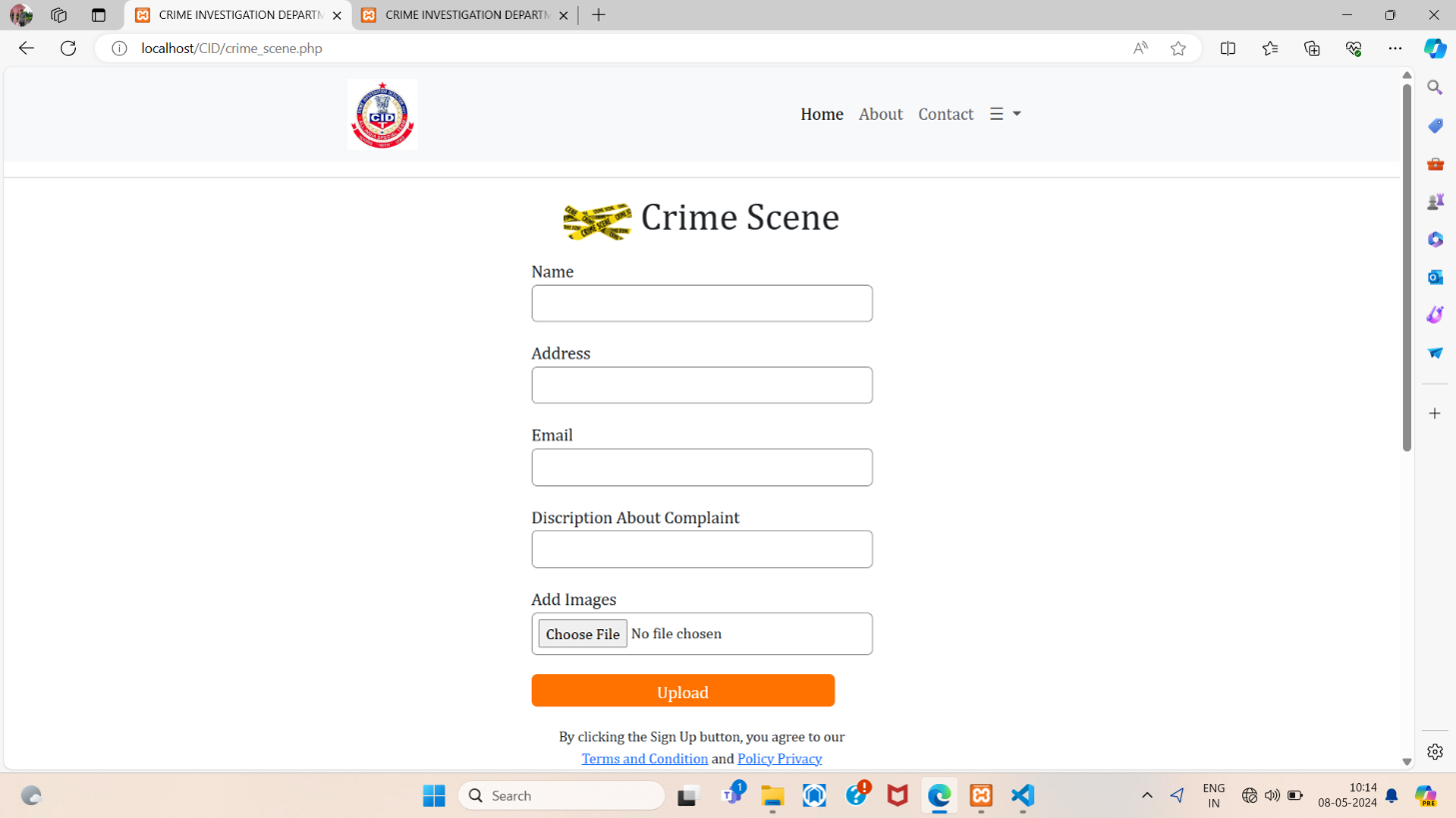
**USER LOGIN PAGE**

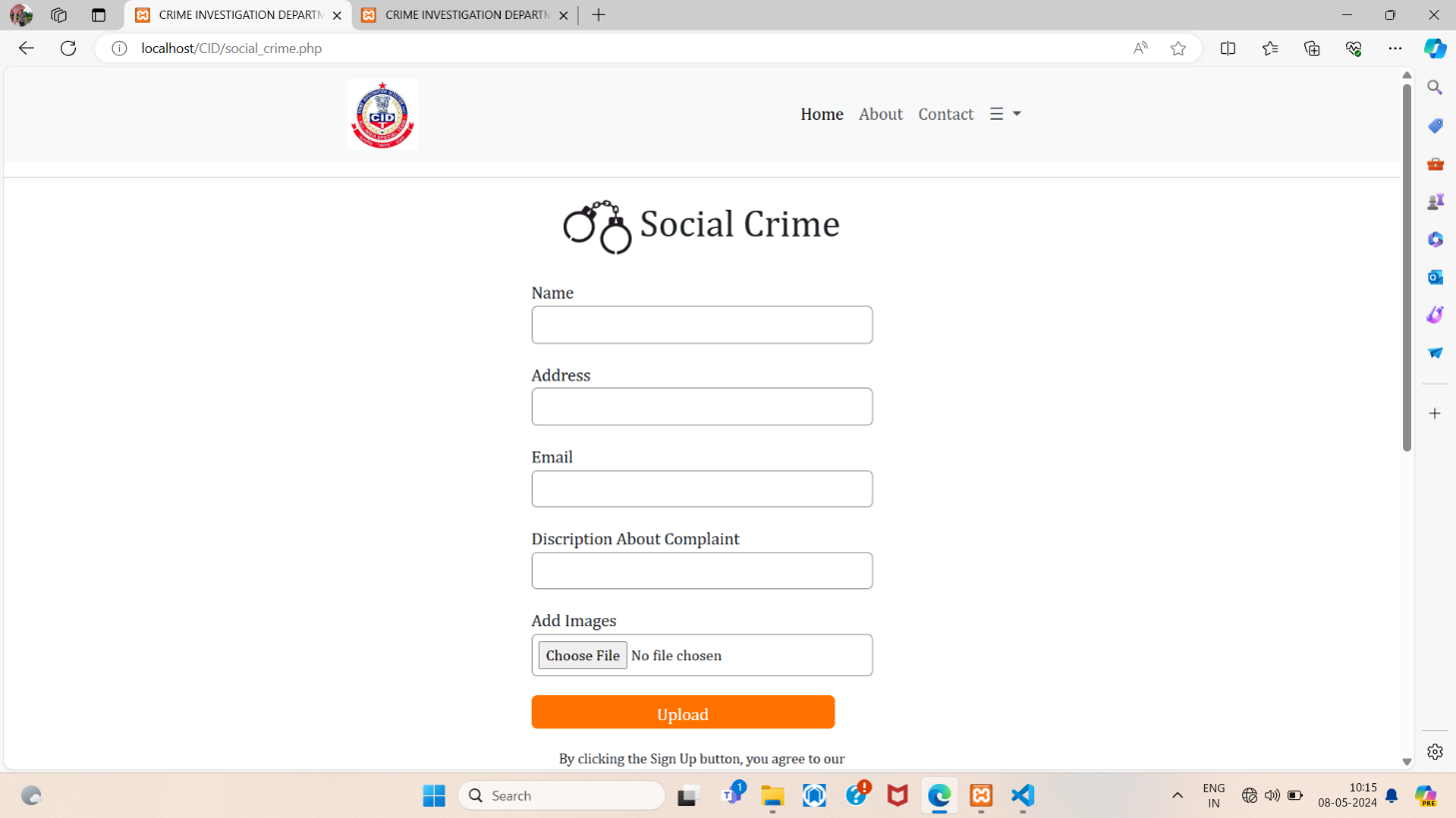


**FORMS FOR COMPLAINT**



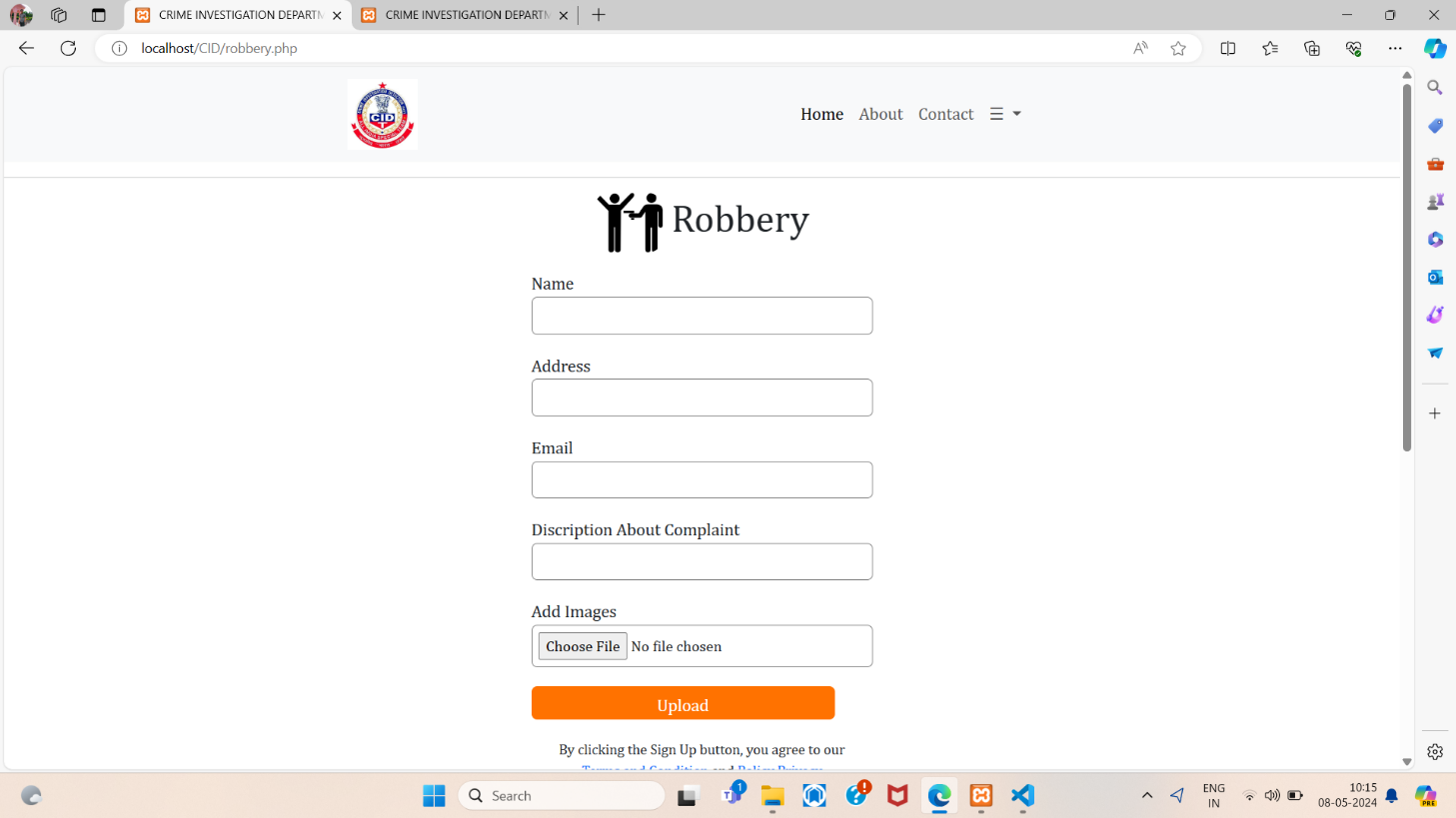
**CRIME SCENE FORM**

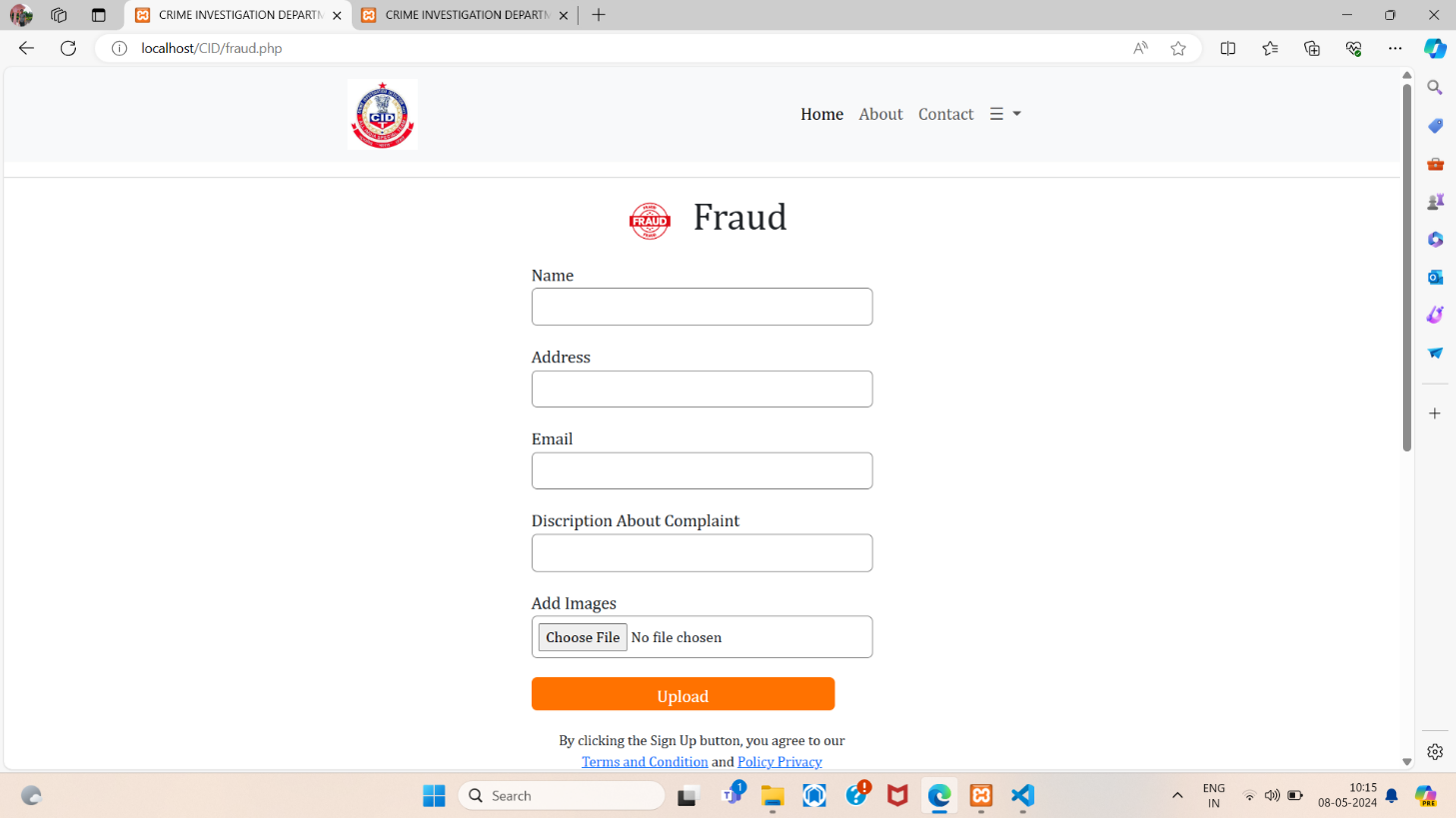




**SOCIAL CRIME**

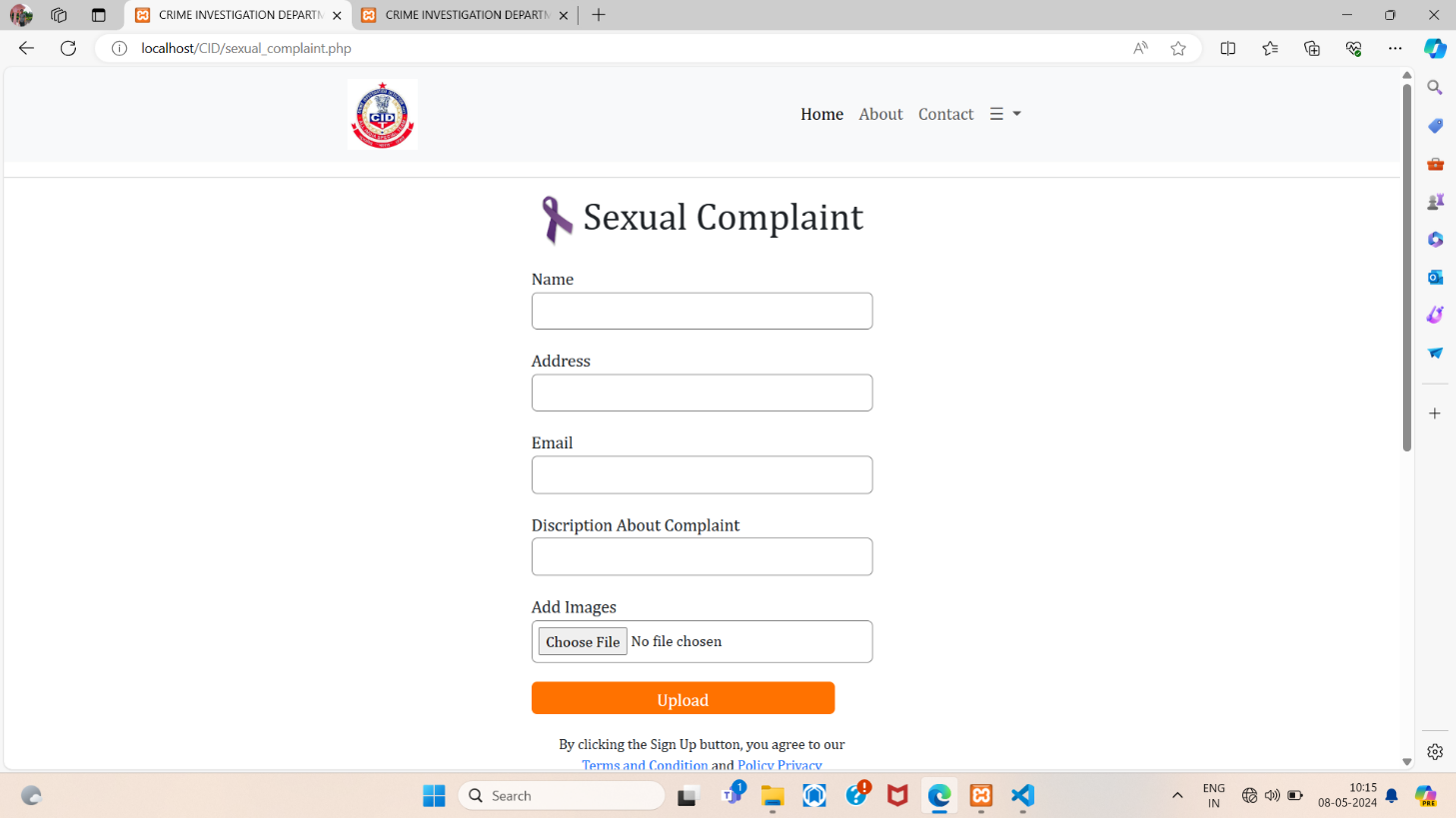
**ROBBERY**



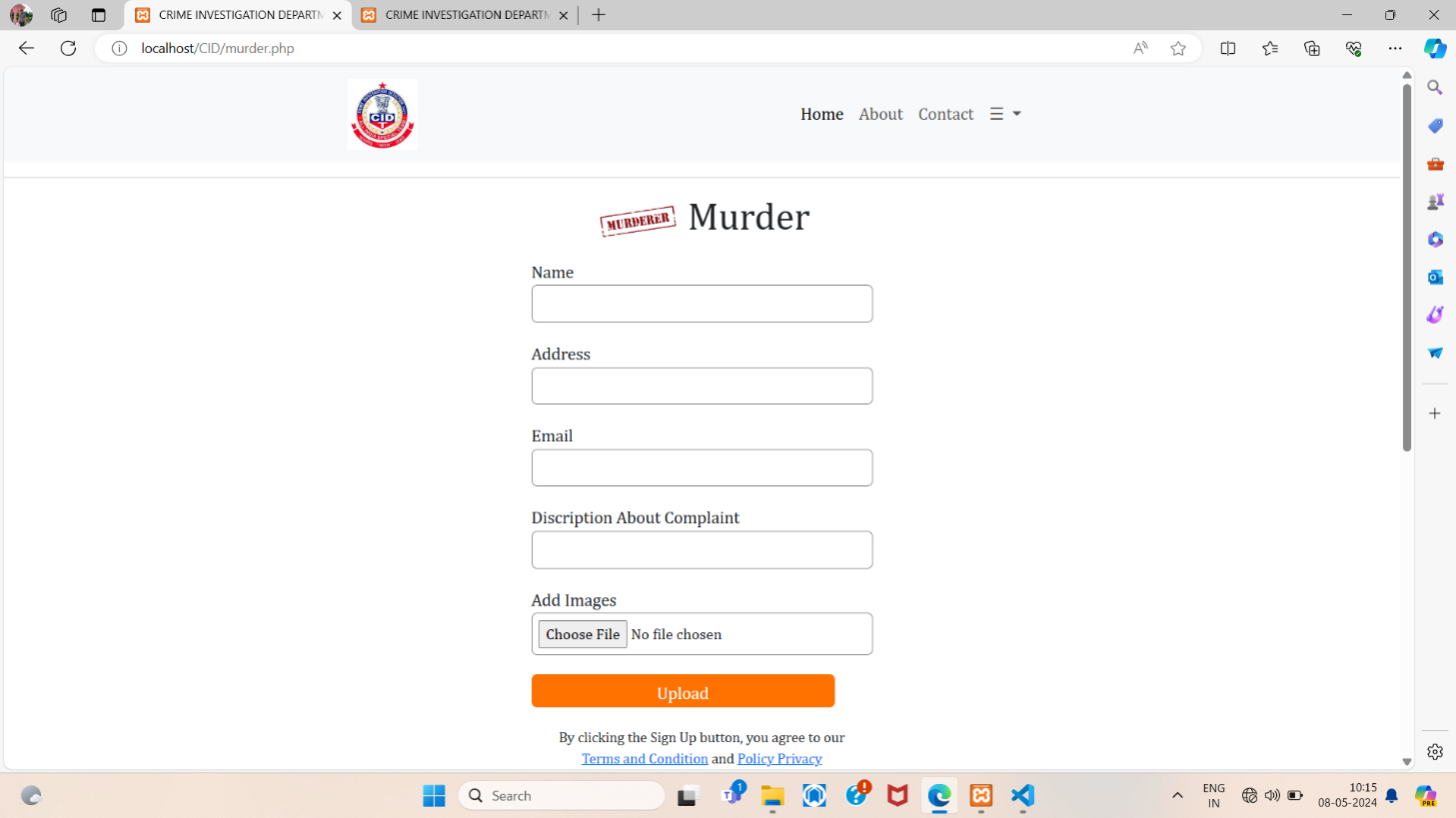


**FRAUD**

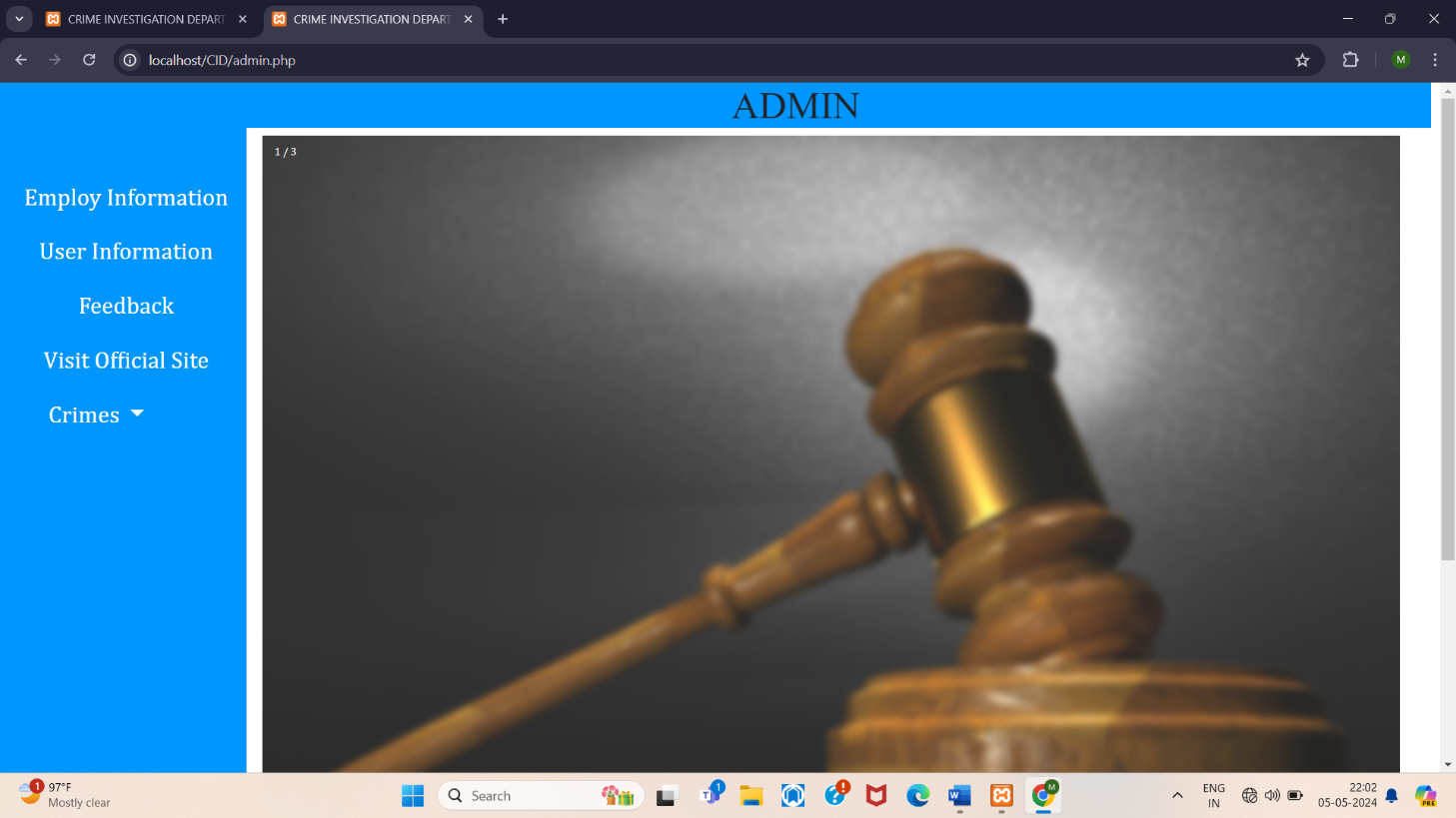
**SEXUAL COMPLAINT**



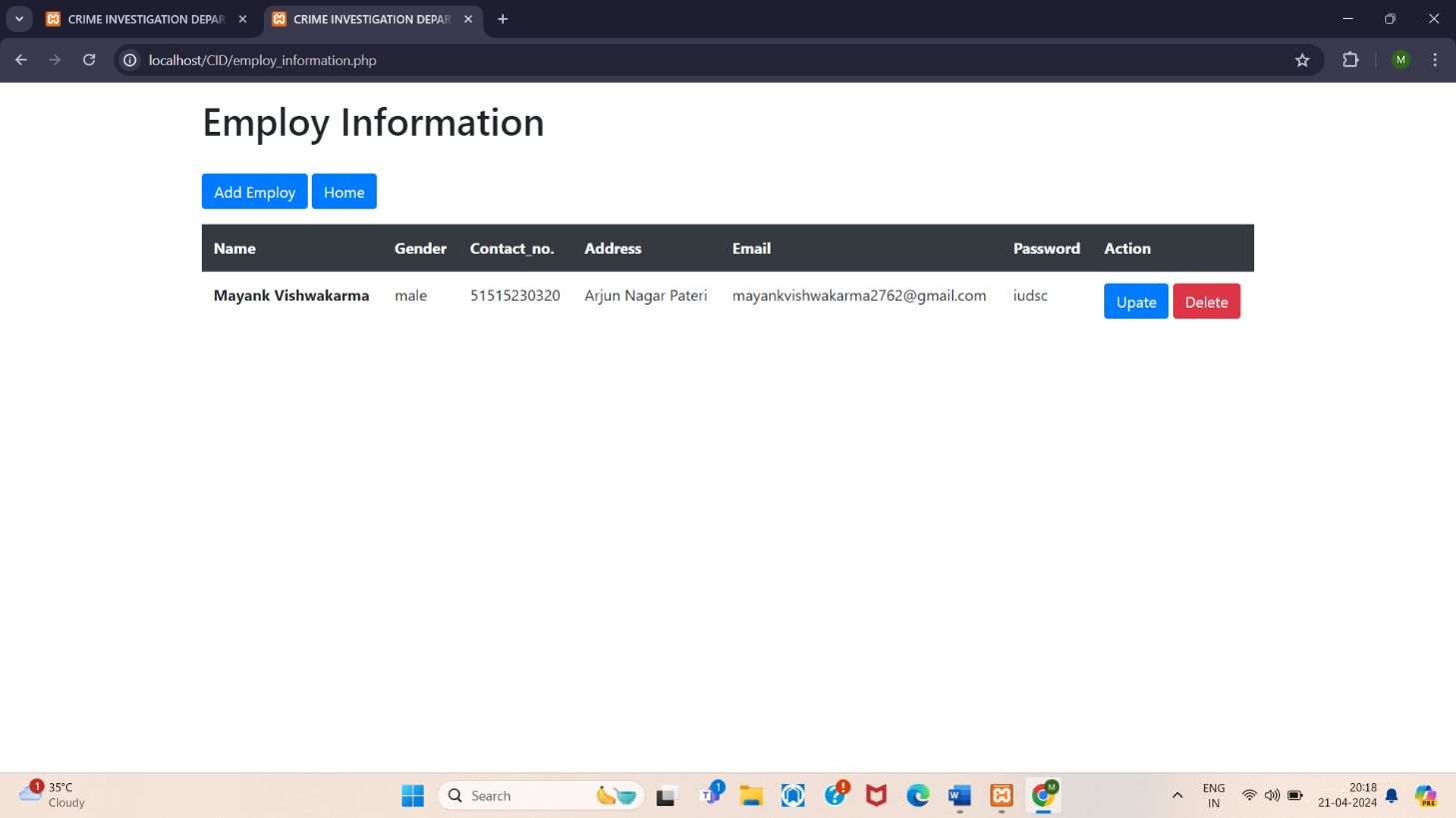
**MURDER**



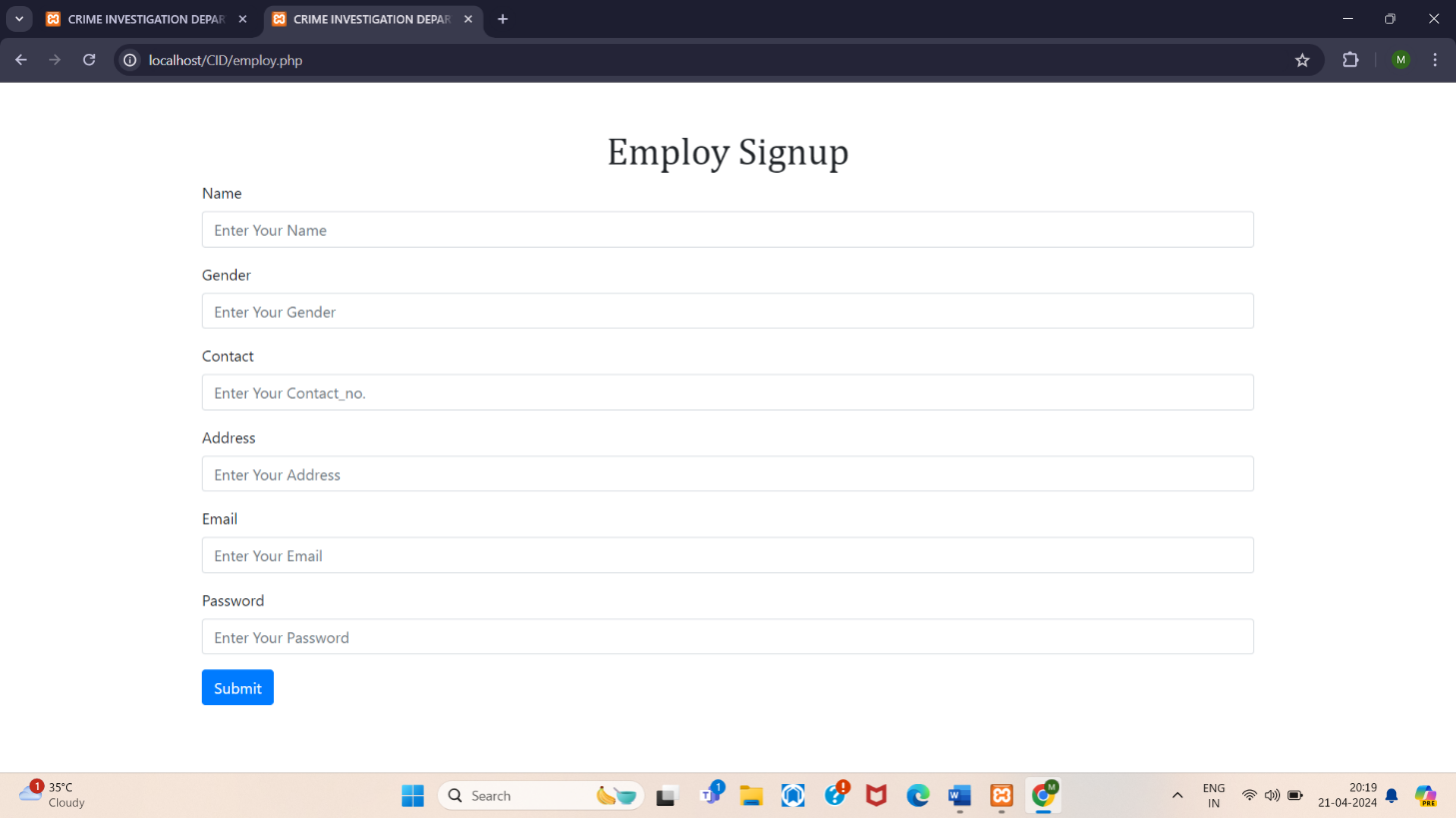
**ADMIN PAGE**



**ADMIN PAGE EMPLOY INFORMATION**



**EMPLOY FORM**



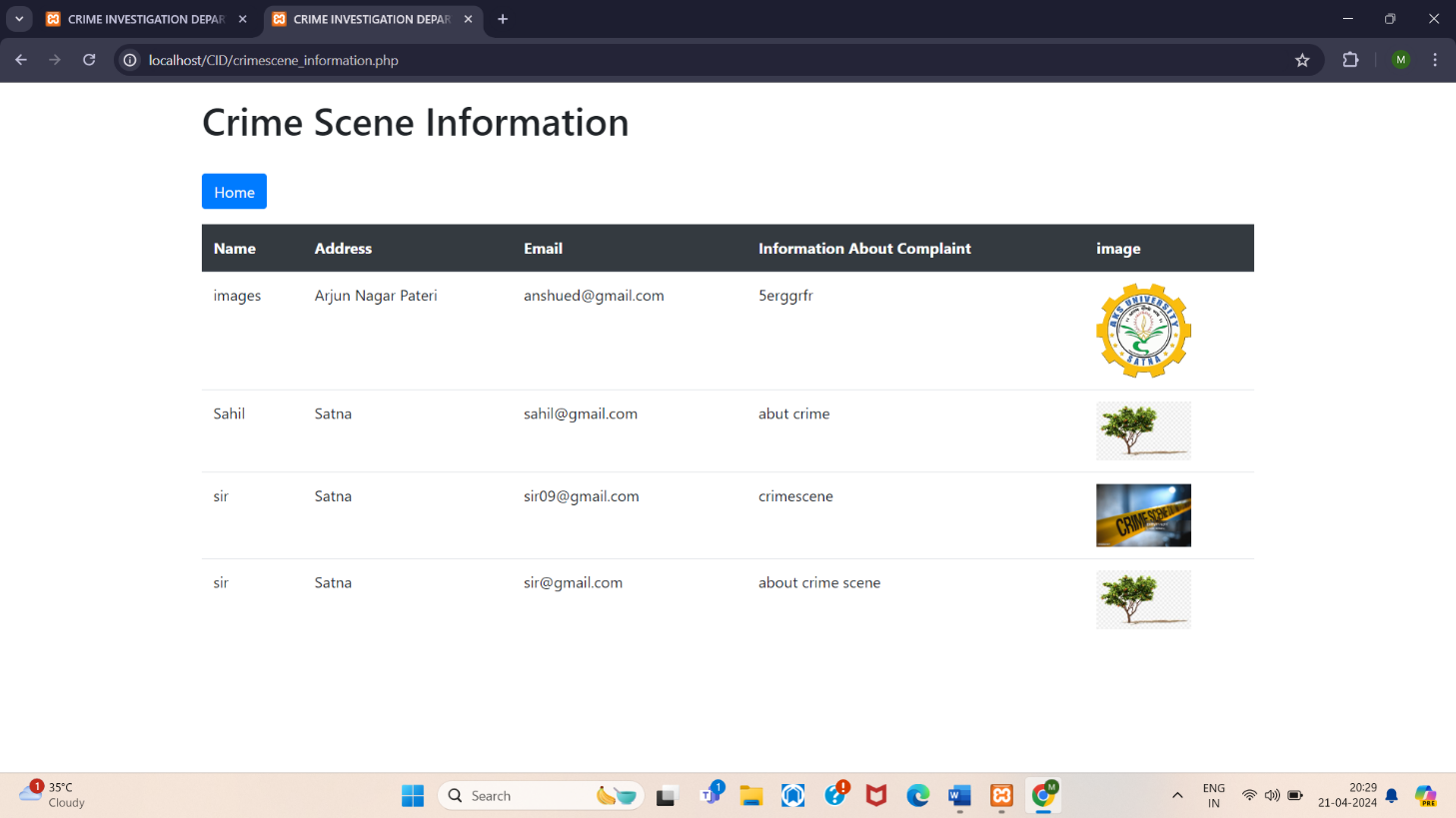
**ADMIN PAGE USER INFORMATION**



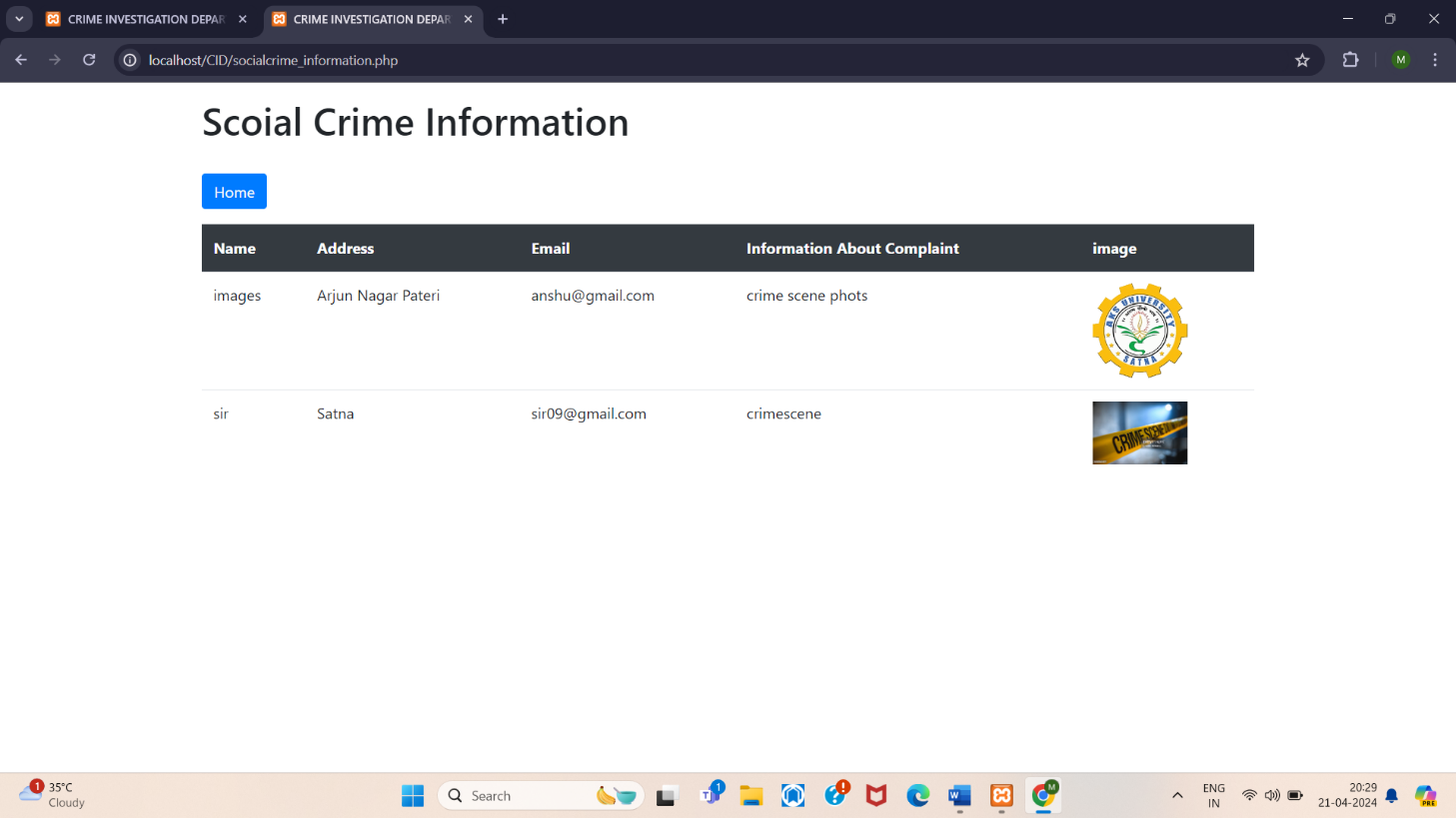
**EMPLOY PAGE CRIMES FORM**



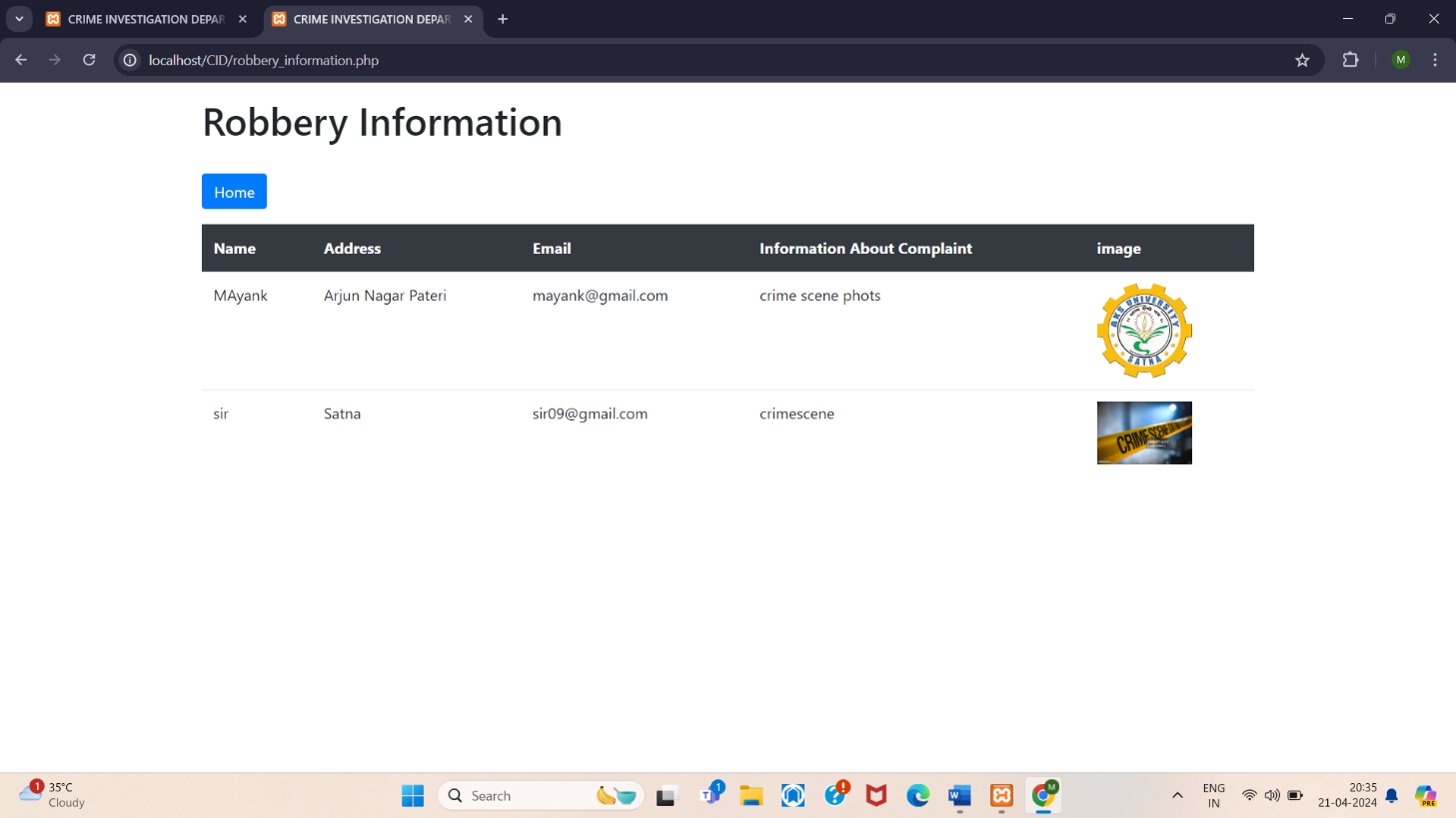
**CRIME SCENE INFORMATION**



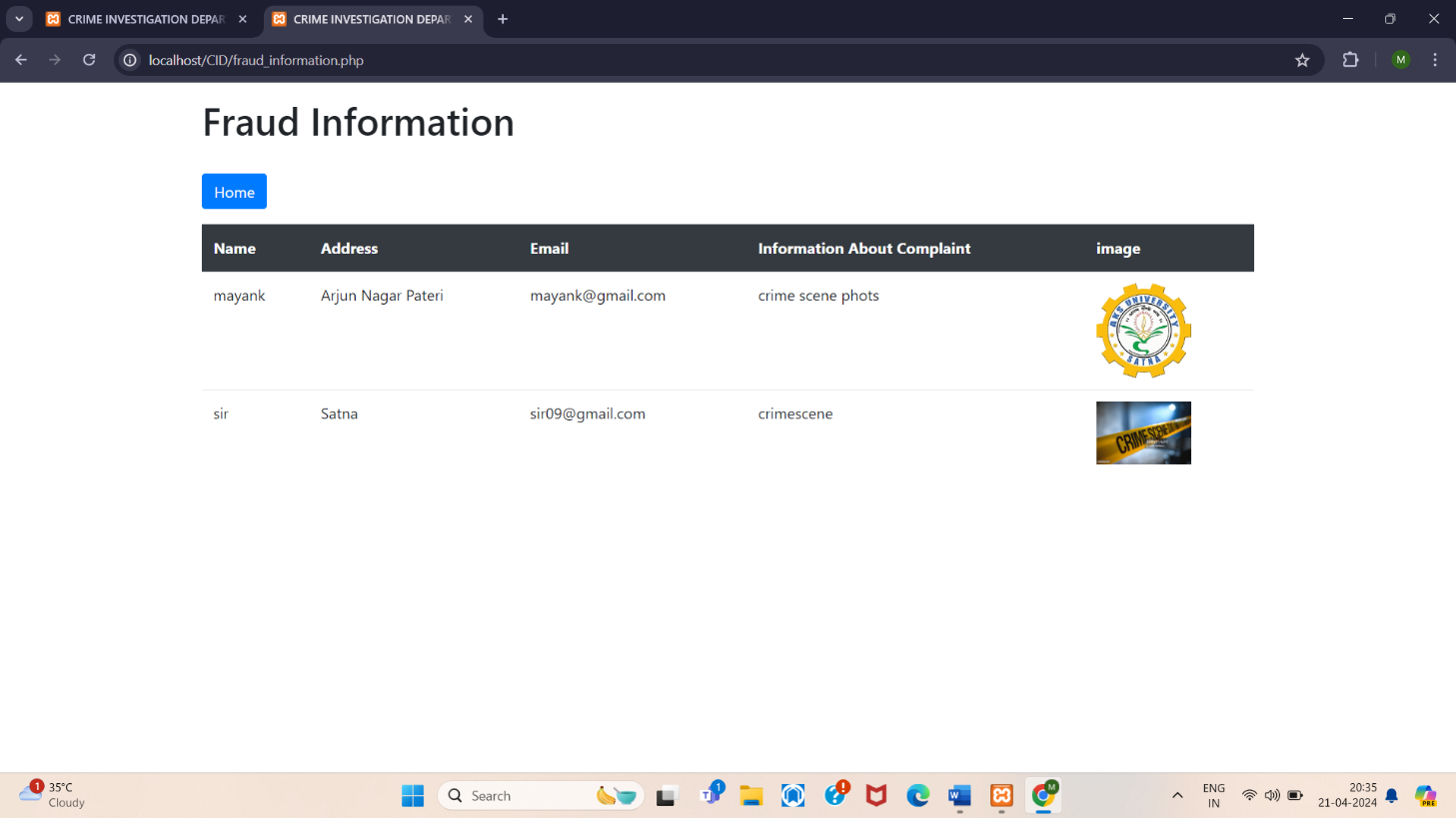
**SOCIAL CRIME INFORMATION**



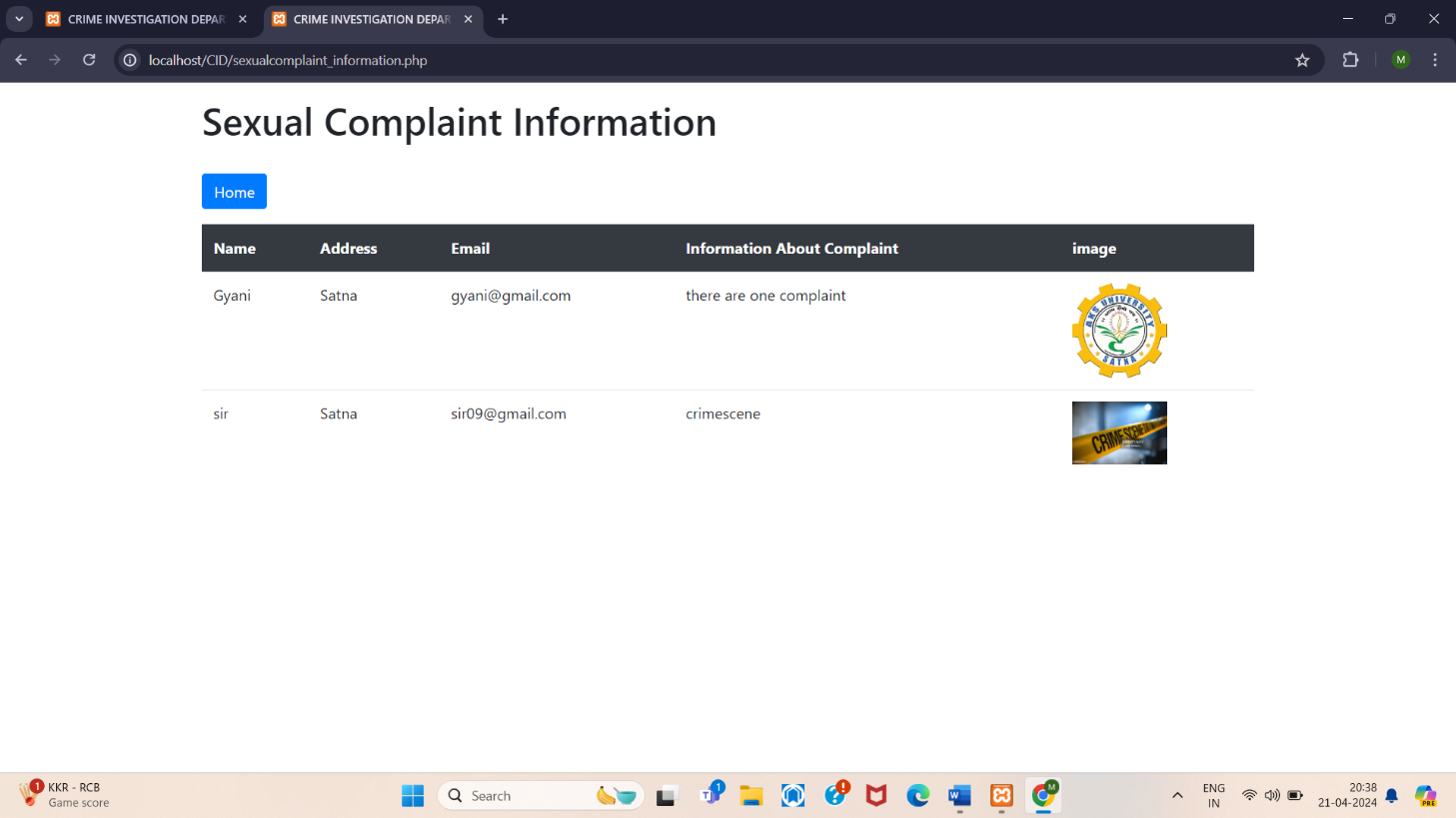
**ROBBERY INFORMATION**



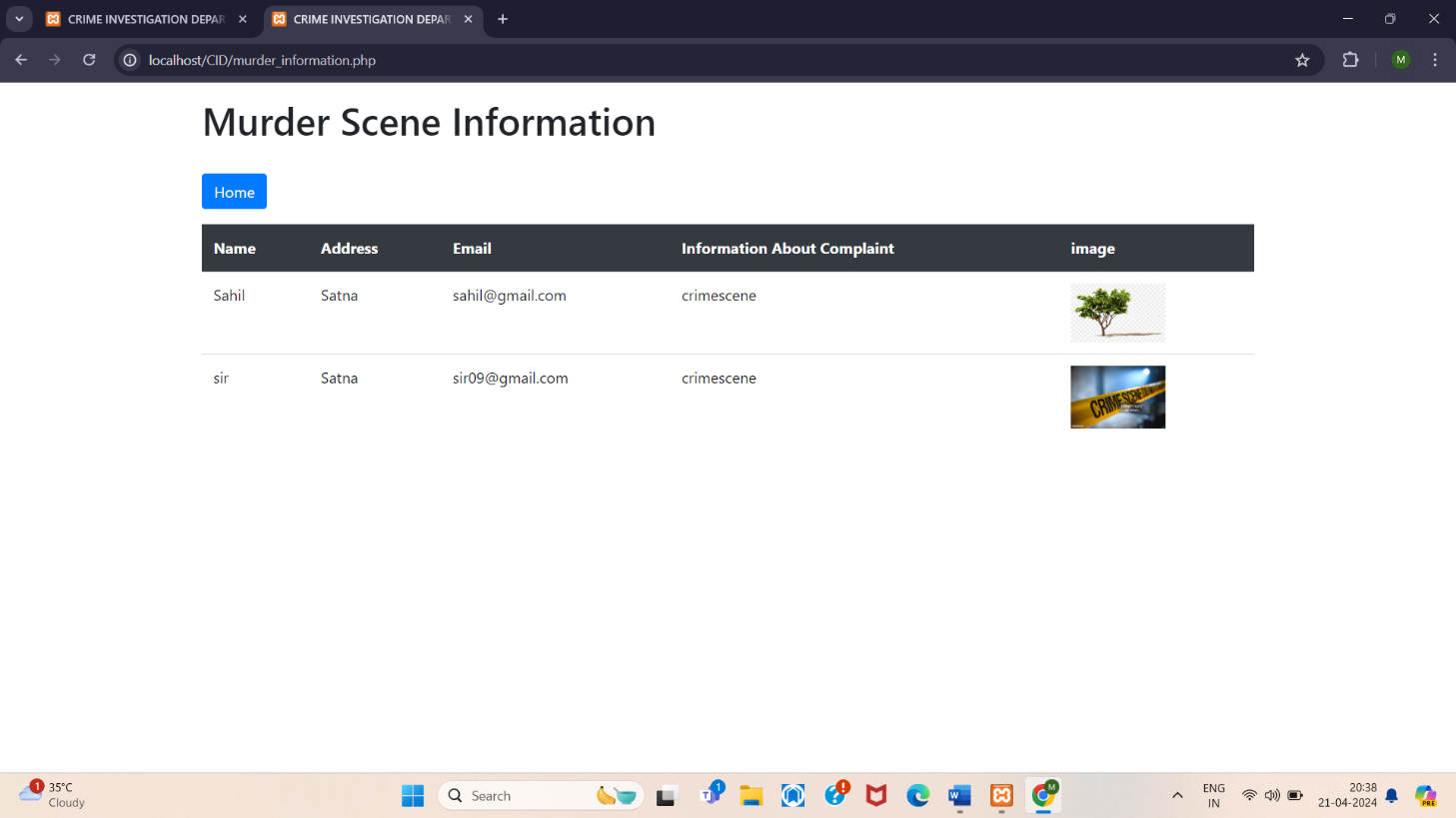
**FRAUD INFORMATION**



**SEXUAL COMPLAINT INFORMATION**



**MURDER INFORMATION**



**CODDING**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>WasteNot - Empowering Rural Commerce</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css">

<link rel="stylesheet" href="assets/css/style.css" />

</head>

<body>

<!-- Navbar -->

<nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">

<div class="container">

<a class="navbar-brand fw-bold" href="index.html">WasteNot</a>

<button class="navbar-toggler" type="button" data-bs-toggle="collapse"

data-bs-target="#navbarNav" aria-controls="navbarNav"

aria-expanded="false" aria-label="Toggle navigation">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse justify-content-end" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item"><a class="nav-link" href="index.html">Home</a></li>

          <li class="nav-item"><a class="nav-link" href="product.html">Products</a></li>

          <li class="nav-item"><a class="nav-link" href="donation.html">Donate</a></li>

          <li class="nav-item"><a class="nav-link" href="agent.html">Agents</a></li>

          <li class="nav-item"><a class="nav-link" href="feature.html">Features</a></li>

          <!-- Login/Profile Section -->

          <li class="nav-item" id="loginButton">

            <a class="btn btn-success ms-3" href="login.html">Login</a>

          </li>

          <!-- User Menu (Shown when logged in) -->

          <li class="nav-item dropdown d-none" id="userProfile">

            <a class="nav-link dropdown-toggle ms-3" href="#" id="userDropdown" role="button" data-bs-toggle="dropdown" aria-expanded="false">

              <i class="bi bi-person-circle"></i> <span id="userName">User</span>

            </a>

            <ul class="dropdown-menu dropdown-menu-end" aria-labelledby="userDropdown">

              <li><a class="dropdown-item" href="profile.html"><i class="bi bi-person"></i> My Profile</a></li>

              <li><a class="dropdown-item" href="my-orders.html"><i class="bi bi-box"></i> My Orders</a></li>

              <li><a class="dropdown-item" href="my-donations.html"><i class="bi bi-heart"></i> My Donations</a></li>

              <li><a class="dropdown-item" href="my-addresses.html"><i class="bi bi-geo-alt"></i> My Addresses</a></li>

              <li><hr class="dropdown-divider"></li>

              <li><a class="dropdown-item text-danger" href="#" id="logoutBtn"><i class="bi bi-box-arrow-right"></i> Logout</a></li>

            </ul>

          </li>

        </ul>

      </div>

    </div>

  </nav>

  <!-- Hero Section -->

  <section class="bg-success text-white text-center py-5">

    <div class="container">

      <h1 class="display-4 fw-bold mb-4">Empowering Rural Commerce</h1>

      <p class="lead mb-4">Buy essential products or donate directly to those in need — all in one platform.</p>

      <div class="d-flex justify-content-center gap-3">

        <a href="product.html" class="btn btn-light btn-lg">

          <i class="bi bi-cart"></i> Shop Now

        </a>

        <a href="donation.html" class="btn btn-outline-light btn-lg">

          <i class="bi bi-heart"></i> Donate

        </a>

      </div>

    </div>

  </section>

  <!-- Stats Section -->

  <section class="py-5 bg-white">

    <div class="container">

      <div class="row text-center g-4">

        <div class="col-md-3">

          <div class="display-4 fw-bold text-success mb-2">5000+</div>

          <p class="text-muted">Products Delivered</p>

        </div>

        <div class="col-md-3">

          <div class="display-4 fw-bold text-success mb-2">2000+</div>

          <p class="text-muted">Active Donors</p>

        </div>

        <div class="col-md-3">

          <div class="display-4 fw-bold text-success mb-2">100+</div>

          <p class="text-muted">Local Agents</p>

        </div>

        <div class="col-md-3">

          <div class="display-4 fw-bold text-success mb-2">50+</div>

          <p class="text-muted">Rural Areas Covered</p>

        </div>

      </div>

    </div>

  </section>

  <!-- Features -->

  <section class="py-5 bg-light">

    <div class="container">

  <h2 class="text-center mb-5">Why Choose WasteNot?</h2>

      <div class="row g-4">

        <div class="col-md-4">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-tags"></i>

              </div>

              <h4>Affordable Products</h4>

              <p class="text-muted">High-quality items at low prices for rural communities.</p>

            </div>

          </div>

        </div>

        <div class="col-md-4">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-heart"></i>

              </div>

              <h4>Direct Donations</h4>

              <p class="text-muted">Donate essentials directly to underserved people with transparency.</p>

            </div>

          </div>

        </div>

        <div class="col-md-4">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-truck"></i>

              </div>

              <h4>Agent-Based Delivery</h4>

              <p class="text-muted">Empowering local agents to deliver products quickly and efficiently.</p>

            </div>

          </div>

        </div>

      </div>

    </div>

  </section>

  <!-- Featured Products -->

  <section class="py-5">

    <div class="container">

      <h2 class="text-center mb-5">Featured Products</h2>

      <div class="row row-cols-1 row-cols-md-4 g-4">

        <div class="col">

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

            <div class="badge bg-success position-absolute m-2">New</div>

            <img src="./assets/images/food-package.jpg" class="card-img-top" alt="Food Package">

            <div class="card-body">

              <h5 class="card-title">Food Package</h5>

              <p class="text-success fw-bold">₹499</p>

              <a href="product.html" class="btn btn-outline-success w-100">View Details</a>

            </div>

          </div>

        </div>

        <div class="col">

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

            <div class="badge bg-danger position-absolute m-2">Sale</div>

            <img src="./assets/images/winter-clothes.jpg" class="card-img-top" alt="Winter Clothes">

            <div class="card-body">

              <h5 class="card-title">Winter Wear</h5>

              <p class="text-success fw-bold">₹999</p>

              <a href="product.html" class="btn btn-outline-success w-100">View Details</a>

            </div>

          </div>

        </div>

        <div class="col">

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

            <img src="./assets/images/study-materials.jpg" class="card-img-top" alt="Study Materials">

            <div class="card-body">

              <h5 class="card-title">Study Materials</h5>

              <p class="text-success fw-bold">₹299</p>

              <a href="product.html" class="btn btn-outline-success w-100">View Details</a>

            </div>

          </div>

        </div>

        <div class="col">

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

            <img src="./assets/images/tablet.jpg" class="card-img-top" alt="Basic Tablet">

            <div class="card-body">

              <h5 class="card-title">Basic Tablet</h5>

              <p class="text-success fw-bold">₹8,999</p>

              <a href="product.html" class="btn btn-outline-success w-100">View Details</a>

            </div>

          </div>

        </div>

      </div>

    </div>

  </section>

  <!-- CTA Section -->

  <section class="py-5 bg-success text-white text-center">

    <div class="container">

      <h2 class="mb-4">Ready to Make a Difference?</h2>

      <p class="lead mb-4">Join our community of donors and help create positive change.</p>

      <div class="d-flex justify-content-center gap-3">

        <a href="signup.html" class="btn btn-light btn-lg">

          <i class="bi bi-person-plus"></i> Join Now

        </a>

        <a href="agent.html" class="btn btn-outline-light btn-lg">

          <i class="bi bi-truck"></i> Become an Agent

        </a>

      </div>

    </div>

  </section>

  <!-- Footer -->

  <footer class="bg-dark text-white py-4">

    <div class="container">

      <div class="row g-4">

        <div class="col-md-4">

          <h5 class="mb-3">About WasteNot</h5>

          <p class="text-muted">Empowering rural communities through affordable products and direct donations.</p>

        </div>

        <div class="col-md-4">

          <h5 class="mb-3">Quick Links</h5>

          <ul class="list-unstyled">

            <li><a href="product.html" class="text-muted text-decoration-none">Products</a></li>

            <li><a href="donation.html" class="text-muted text-decoration-none">Donate</a></li>

            <li><a href="agent.html" class="text-muted text-decoration-none">Become an Agent</a></li>

            <li><a href="feature.html" class="text-muted text-decoration-none">Features</a></li>

          </ul>

        </div>

        <div class="col-md-4">

          <h5 class="mb-3">Connect With Us</h5>

          <div class="d-flex gap-3">

            <a href="#" class="text-white fs-4"><i class="bi bi-facebook"></i></a>

            <a href="#" class="text-white fs-4"><i class="bi bi-twitter"></i></a>

            <a href="#" class="text-white fs-4"><i class="bi bi-instagram"></i></a>

            <a href="#" class="text-white fs-4"><i class="bi bi-linkedin"></i></a>

          </div>

        </div>

      </div>

      <hr class="my-4">

      <div class="text-center">

        <p class="mb-0">&copy; 2025 WasteNot. All rights reserved.</p>

      </div>

    </div>

  </footer>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>

  <script src="assets/js/auth.js"></script>

</body>

</html>

**Products.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Products | WasteNot</title>

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css">

</head>

<body>

  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">

    <div class="container">

      <a class="navbar-brand fw-bold" href="index.html">WasteNot</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item"><a class="nav-link" href="feature.html">Features</a></li>

          <li class="nav-item"><a class="nav-link" href="agent.html">Agents</a></li>

          <li class="nav-item"><a class="nav-link" href="donation.html">Donate</a></li>

          <li class="nav-item"><a class="nav-link" href="product.html">Products</a></li>

          <li class="nav-item"><a class="btn btn-success ms-2" href="login.html">Login</a></li>

        </ul>

      </div>

    </div>

  </nav>

  <!-- Search and Filter Section -->

  <div class="bg-light py-4">

    <div class="container">

      <div class="row g-3">

        <div class="col-md-6">

          <div class="input-group">

            <input type="text" class="form-control" placeholder="Search products...">

            <button class="btn btn-success"><i class="bi bi-search"></i></button>

          </div>

        </div>

        <div class="col-md-3">

          <select class="form-select">

            <option value="">All Categories</option>

            <option>Food & Groceries</option>

            <option>Clothing</option>

            <option>Electronics</option>

            <option>Books & Education</option>

            <option>Furniture</option>

            <option>Household Items</option>

          </select>

        </div>

        <div class="col-md-3">

          <select class="form-select">

            <option value="">Sort By</option>

            <option>Price: Low to High</option>

            <option>Price: High to Low</option>

            <option>Newest First</option>

          </select>

        </div>

      </div>

    </div>

  </div>

  <!-- Products Section -->

  <div class="container my-5">

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

    <!-- Food Category -->

    <h4 class="mb-3 mt-5"><i class="bi bi-basket"></i> Food & Groceries</h4>

    <div class="row row-cols-1 row-cols-md-4 g-4 mb-5">

      <div class="col">

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

          <div class="badge bg-success position-absolute m-2">New</div>

          <img src="https://via.placeholder.com/400x250?text=Food" class="card-img-top" alt="Food Product">

          <div class="card-body">

            <h5 class="card-title">Pack of Dry Fruits</h5>

            <p class="card-text">Premium quality dry fruits pack.</p>

            <p class="text-success fw-bold">₹499</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Rice" class="card-img-top" alt="Rice">

          <div class="card-body">

            <h5 class="card-title">Organic Rice</h5>

            <p class="card-text">5kg pack of organic basmati rice.</p>

            <p class="text-success fw-bold">₹399</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <div class="badge bg-danger position-absolute m-2">Sale</div>

          <img src="https://via.placeholder.com/400x250?text=Pulses" class="card-img-top" alt="Pulses">

          <div class="card-body">

            <h5 class="card-title">Mixed Dal Pack</h5>

            <p class="card-text">Assorted pulses pack, 2kg.</p>

            <p class="text-success fw-bold">₹299</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

<div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Spices" class="card-img-top" alt="Spices">

          <div class="card-body">

            <h5 class="card-title">Spice Box Set</h5>

            <p class="card-text">Essential Indian spices kit.</p>

            <p class="text-success fw-bold">₹599</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

    </div>

    <!-- Clothing Category -->

    <h4 class="mb-3"><i class="bi bi-bag"></i> Clothing</h4>

    <div class="row row-cols-1 row-cols-md-4 g-4 mb-5">

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Winter" class="card-img-top" alt="Winter Jacket">

          <div class="card-body">

            <h5 class="card-title">Winter Jacket</h5>

            <p class="card-text">Warm winter jacket for cold weather.</p>

            <p class="text-success fw-bold">₹999</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <div class="badge bg-danger position-absolute m-2">Sale</div>

          <img src="https://via.placeholder.com/400x250?text=Sweater" class="card-img-top" alt="Sweater">

          <div class="card-body">

            <h5 class="card-title">Woolen Sweater</h5>

            <p class="card-text">Pure wool sweater for winter.</p>

            <p class="text-success fw-bold">₹699</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Shawl" class="card-img-top" alt="Shawl">

          <div class="card-body">

            <h5 class="card-title">Traditional Shawl</h5>

            <p class="card-text">Handwoven traditional shawl.</p>

            <p class="text-success fw-bold">₹799</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <div class="badge bg-success position-absolute m-2">New</div>

          <img src="https://via.placeholder.com/400x250?text=Blanket" class="card-img-top" alt="Blanket">

          <div class="card-body">

            <h5 class="card-title">Cotton Blanket</h5>

            <p class="card-text">Soft cotton blanket for all seasons.</p>

            <p class="text-success fw-bold">₹599</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

    </div>

    <!-- Electronics Category -->

    <h4 class="mb-3"><i class="bi bi-laptop"></i> Electronics</h4>

    <div class="row row-cols-1 row-cols-md-4 g-4">

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Laptop" class="card-img-top" alt="Laptop">

          <div class="card-body">

            <h5 class="card-title">Refurbished Laptop</h5>

            <p class="card-text">Core i5, 8GB RAM, 256GB SSD</p>

            <p class="text-success fw-bold">₹24,999</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Tablet" class="card-img-top" alt="Tablet">

          <div class="card-body">

            <h5 class="card-title">Learning Tablet</h5>

            <p class="card-text">Perfect for online education</p>

            <p class="text-success fw-bold">₹8,999</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <div class="badge bg-danger position-absolute m-2">Sale</div>

          <img src="https://via.placeholder.com/400x250?text=Phone" class="card-img-top" alt="Phone">

          <div class="card-body">

            <h5 class="card-title">Basic Smartphone</h5>

            <p class="card-text">4G enabled basic smartphone</p>

            <p class="text-success fw-bold">₹5,999</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250?text=Calculator" class="card-img-top" alt="Calculator">

          <div class="card-body">

            <h5 class="card-title">Scientific Calculator</h5>

            <p class="card-text">Essential for students</p>

            <p class="text-success fw-bold">₹899</p>

            <a href="#" class="btn btn-primary w-100">Add to Cart</a>

          </div>

        </div>

      </div>

    </div>

</div>

  <!-- Footer -->

  <footer class="bg-dark text-white text-center py-3">

    <div class="container">

      <p>&copy; 2025 WasteNot. All rights reserved.</p>

    </div>

  </footer>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**donation.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Donate | WasteNot</title>

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

</head>

<body>

  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">

    <div class="container">

      <a class="navbar-brand fw-bold" href="index.html">WasteNot</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item"><a class="nav-link" href="product.html">Products</a></li>

          <li class="nav-item"><a class="nav-link" href="donation.html">Donate</a></li>

          <li class="nav-item"><a class="nav-link" href="agent.html">Agents</a></li>

          <li class="nav-item"><a class="nav-link" href="feature.html">Features</a></li>

          <li class="nav-item"><a class="btn btn-success ms-2" href="login.html">Login</a></li>

        </ul>

      </div>

    </div>

  </nav>

  <!-- Donation Form -->

  <div class="container my-5">

    <h2 class="text-center mb-4">Donate to Support Communities</h2>

    <div class="row justify-content-center">

      <div class="col-md-8 bg-white shadow-sm p-4 rounded">

        <form>

          <div class="mb-3">

            <label class="form-label">Your Full Name</label>

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

          </div>

          <div class="mb-3">

            <label class="form-label">Email Address</label>

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

          </div>

          <div class="mb-3">

            <label class="form-label">Item to Donate</label>

            <input type="text" class="form-control" placeholder="e.g. Blankets, Food Packets" required />

          </div>

          <div class="mb-3">

            <label class="form-label">Item Category</label>

            <select class="form-select" required>

              <option value="">Select Category</option>

              <option>Food</option>

              <option>Clothing</option>

              <option>Medicine</option>

              <option>Hygiene</option>

              <option>Others</option>

            </select>

          </div>

          <div class="mb-3">

            <label class="form-label">Quantity</label>

            <input type="number" class="form-control" required />

          </div>

          <div class="mb-3">

            <label class="form-label">Location (for pickup)</label>

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

          </div>

          <div class="mb-3">

            <label class="form-label">Additional Notes</label>

            <textarea class="form-control" rows="3"></textarea>

          </div>

          <button type="submit" class="btn btn-success w-100">Submit Donation</button>

        </form>

      </div>

    </div>

  </div>

  <!-- Footer -->

  <footer class="bg-dark text-white text-center py-3 mt-5">

    <div class="container">

      <p>&copy; 2025 WasteNot. All rights reserved.</p>

    </div>

  </footer>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**agents.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Agents | WasteNot</title>

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css">

</head>

<body>

  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">

    <div class="container">

      <a class="navbar-brand fw-bold" href="index.html">WasteNot</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

      </button>

      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item"><a class="nav-link" href="product.html">Products</a></li>

          <li class="nav-item"><a class="nav-link" href="donation.html">Donate</a></li>

          <li class="nav-item"><a class="nav-link" href="agent.html">Agents</a></li>

          <li class="nav-item"><a class="nav-link" href="feature.html">Features</a></li>

          <li class="nav-item"><a class="btn btn-success ms-2" href="login.html">Login</a></li>

        </ul>

      </div>

    </div>

  </nav>

  <!-- Hero Section -->

  <section class="bg-success text-white py-5">

    <div class="container text-center">

      <h1 class="display-4 fw-bold mb-4">Become a Delivery Agent</h1>

      <p class="lead">Join our network of local agents and help connect rural communities with essential products</p>

      <a href="#apply" class="btn btn-light btn-lg mt-3">

        <i class="bi bi-person-plus"></i> Apply Now

      </a>

    </div>

  </section>

  <!-- Benefits Section -->

  <section class="py-5 bg-light">

    <div class="container">

      <h2 class="text-center mb-5">Why Become an Agent?</h2>

      <div class="row g-4">

        <div class="col-md-3">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-cash-coin"></i>

              </div>

              <h5>Earn Extra Income</h5>

              <p class="text-muted">Flexible working hours with competitive commission rates</p>

            </div>

          </div>

        </div>

        <div class="col-md-3">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-graph-up-arrow"></i>

              </div>

              <h5>Business Growth</h5>

              <p class="text-muted">Expand your network and grow your business in your area</p>

            </div>

          </div>

        </div>

        <div class="col-md-3">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-shield-check"></i>

              </div>

              <h5>Training & Support</h5>

              <p class="text-muted">Get complete training and ongoing support from our team</p>

            </div>

          </div>

        </div>

        <div class="col-md-3">

          <div class="card border-0 shadow-sm h-100">

            <div class="card-body text-center p-4">

              <div class="display-4 text-success mb-3">

                <i class="bi bi-heart"></i>

              </div>

              <h5>Social Impact</h5>

              <p class="text-muted">Make a difference in your community by helping others</p>

            </div>

          </div>

        </div>

      </div>

    </div>

  </section>

  <!-- Current Agents Section -->

  <section class="py-5">

    <div class="container">

      <h2 class="text-center mb-5">Meet Our Local Delivery Agents</h2>

      <div class="row row-cols-1 row-cols-md-3 g-4 mb-5">

        <div class="col">

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

            <img src="https://via.placeholder.com/300x200" class="card-img-top" alt="Agent 1">

            <div class="card-body">

              <h5 class="card-title">Ramesh Kumar</h5>

              <p class="card-text">📍 Jharkhand<br>✅ Available for delivery</p>

              <p class="text-muted"><i class="bi bi-star-fill text-warning"></i> 4.8/5 Rating</p>

              <a href="#" class="btn btn-outline-success w-100">Contact Agent</a>

            </div>

          </div>

        </div>

        <div class="col">

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

            <img src="https://via.placeholder.com/300x200" class="card-img-top" alt="Agent 2">

            <div class="card-body">

              <h5 class="card-title">Sunita Devi</h5>

              <p class="card-text">📍 Bihar<br>✅ Available for delivery</p>

              <p class="text-muted"><i class="bi bi-star-fill text-warning"></i> 4.9/5 Rating</p>

              <a href="#" class="btn btn-outline-success w-100">Contact Agent</a>

            </div>

          </div>

        </div>

        <div class="col">

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

            <img src="https://via.placeholder.com/300x200" class="card-img-top" alt="Agent 3">

            <div class="card-body">

              <h5 class="card-title">Manoj Yadav</h5>

              <p class="card-text">📍 Uttar Pradesh<br>❌ Currently Unavailable</p>

              <p class="text-muted"><i class="bi bi-star-fill text-warning"></i> 4.7/5 Rating</p>

              <button class="btn btn-secondary w-100" disabled>Unavailable</button>

            </div>

          </div>

        </div>

      </div>

    </div>

  </section>

  <!-- Application Form Section -->

  <section id="apply" class="py-5 bg-light">

    <div class="container">

      <div class="row justify-content-center">

        <div class="col-md-8">

          <div class="card border-0 shadow">

            <div class="card-body p-4">

              <h2 class="text-center mb-4">Agent Application Form</h2>

              <form>

                <!-- Personal Information -->

                <h5 class="mb-3">Personal Information</h5>

                <div class="row g-3 mb-4">

                  <div class="col-md-6">

                    <label class="form-label">Full Name</label>

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

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Date of Birth</label>

                    <input type="date" class="form-control" required>

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Email Address</label>

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

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Phone Number</label>

                    <input type="tel" class="form-control" required>

                  </div>

                </div>

                <!-- Address Information -->

                <h5 class="mb-3">Address Information</h5>

                <div class="row g-3 mb-4">

                  <div class="col-12">

                    <label class="form-label">Street Address</label>

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

                  </div>

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

                    <label class="form-label">City</label>

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

                  </div>

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

                    <label class="form-label">State</label>

                    <select class="form-select" required>

                      <option value="">Select State</option>

                      <option>Uttar Pradesh</option>

                      <option>Bihar</option>

                      <option>Jharkhand</option>

                      <option>West Bengal</option>

                      <option>Madhya Pradesh</option>

                    </select>

                  </div>

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

                    <label class="form-label">PIN Code</label>

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

                  </div>

                </div>

                <!-- Vehicle Information -->

                <h5 class="mb-3">Vehicle Information</h5>

                <div class="row g-3 mb-4">

                  <div class="col-md-6">

                    <label class="form-label">Vehicle Type</label>

                    <select class="form-select" required>

                      <option value="">Select Vehicle Type</option>

                      <option>Two Wheeler</option>

                      <option>Three Wheeler</option>

                      <option>Four Wheeler</option>

                      <option>Bicycle</option>

                    </select>

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Vehicle Number</label>

                    <input type="text" class="form-control">

                  </div>

                </div>

                <!-- Additional Information -->

                <h5 class="mb-3">Additional Information</h5>

                <div class="row g-3 mb-4">

                  <div class="col-12">

                    <label class="form-label">Previous Delivery Experience</label>

                    <textarea class="form-control" rows="3" placeholder="Tell us about your previous delivery experience (if any)"></textarea>

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Preferred Working Hours</label>

                    <select class="form-select" required>

                      <option value="">Select Hours</option>

                      <option>Full Time (8-10 hours)</option>

                      <option>Part Time (4-6 hours)</option>

                      <option>Weekends Only</option>

                      <option>Flexible Hours</option>

                    </select>

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Languages Known</label>

                    <input type="text" class="form-control" placeholder="e.g., Hindi, English, Bengali">

                  </div>

                </div>

                <!-- Documents Upload -->

                <h5 class="mb-3">Required Documents</h5>

                <div class="row g-3 mb-4">

                  <div class="col-md-6">

                    <label class="form-label">ID Proof (Aadhar/PAN)</label>

                    <input type="file" class="form-control" required>

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Address Proof</label>

                    <input type="file" class="form-control" required>

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Driving License</label>

                    <input type="file" class="form-control">

                  </div>

                  <div class="col-md-6">

                    <label class="form-label">Profile Photo</label>

                    <input type="file" class="form-control" required>

                  </div>

                </div>

                <!-- Terms and Submit -->

                <div class="mb-4">

                  <div class="form-check">

                    <input type="checkbox" class="form-check-input" id="terms" required>

                    <label class="form-check-label" for="terms">

                      I agree to the <a href="#">Terms and Conditions</a> and <a href="#">Privacy Policy</a>

                    </label>

                  </div>

                </div>

                <button type="submit" class="btn btn-success btn-lg w-100">Submit Application</button>

              </form>

            </div>

          </div>

        </div>

      </div>

    </div>

</section>

  <!-- Footer -->

  <footer class="bg-dark text-white py-4">

    <div class="container">

      <div class="row g-4">

        <div class="col-md-4">

          <h5 class="mb-3">About WasteNot</h5>

          <p class="text-muted">Empowering rural communities through affordable products and direct donations.</p>

        </div>

        <div class="col-md-4">

          <h5 class="mb-3">Quick Links</h5>

          <ul class="list-unstyled">

            <li><a href="product.html" class="text-muted text-decoration-none">Products</a></li>

            <li><a href="donation.html" class="text-muted text-decoration-none">Donate</a></li>

            <li><a href="agent.html" class="text-muted text-decoration-none">Become an Agent</a></li>

            <li><a href="feature.html" class="text-muted text-decoration-none">Features</a></li>

          </ul>

        </div>

        <div class="col-md-4">

          <h5 class="mb-3">Connect With Us</h5>

          <div class="d-flex gap-3">

            <a href="#" class="text-white fs-4"><i class="bi bi-facebook"></i></a>

            <a href="#" class="text-white fs-4"><i class="bi bi-twitter"></i></a>

            <a href="#" class="text-white fs-4"><i class="bi bi-instagram"></i></a>

            <a href="#" class="text-white fs-4"><i class="bi bi-linkedin"></i></a>

          </div>

        </div>

      </div>

<hr class="my-4">

      <div class="text-center">

        <p class="mb-0">&copy; 2025 WasteNot. All rights reserved.</p>

      </div>

    </div>

  </footer>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**features.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8" />

  <title>Features | WasteNot</title>

  <meta name="viewport" content="width=device-width, initial-scale=1" />

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" />

</head>

<body>

  <!-- Navbar -->

  <nav class="navbar navbar-expand-lg navbar-light bg-light shadow-sm">

    <div class="container">

      <a class="navbar-brand fw-bold" href="index.html">WasteNot</a>

      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">

        <span class="navbar-toggler-icon"></span>

</button>

      <div class="collapse navbar-collapse justify-content-end" id="navbarNav">

        <ul class="navbar-nav">

          <li class="nav-item"><a class="nav-link" href="product.html">Products</a></li>

          <li class="nav-item"><a class="nav-link" href="donation.html">Donate</a></li>

          <li class="nav-item"><a class="nav-link" href="agent.html">Agents</a></li>

<li class="nav-item"><a class="nav-link" href="feature.html">Features</a></li>

          <li class="nav-item"><a class="btn btn-success ms-2" href="login.html">Login</a></li>

        </ul>

      </div>

    </div>

  </nav>

  <!-- Features Section -->

  <div class="container my-5">

    <h2 class="text-center mb-4">Key Features of WasteNot</h2>

    <div class="row row-cols-1 row-cols-md-2 g-4">

      <!-- Feature 1 -->

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250" class="card-img-top" alt="Feature 1">

          <div class="card-body">

            <h5 class="card-title">Low-Cost Products & Services</h5>

            <p class="card-text">WasteNot ensures that you can buy essential products and services at the lowest possible prices. No matter where you live, we bring affordable solutions to everyone.</p>

          </div>

        </div>

      </div>

      <!-- Feature 2 -->

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250" class="card-img-top" alt="Feature 2">

          <div class="card-body">

            <h5 class="card-title">Rural & Remote Area Delivery</h5>

            <p class="card-text">Our local agents ensure fast and reliable delivery to even the most rural and remote areas, helping bridge the gap between urban and rural markets.</p>

          </div>

        </div>

      </div>

      <!-- Feature 3 -->

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250" class="card-img-top" alt="Feature 3">

          <div class="card-body">

  <h5 class="card-title">Donation Platform</h5>

            <p class="card-text">WasteNot also lets you give back to the community. Donate essential items to underserved communities directly from the website.</p>

          </div>

        </div>

      </div>

      <!-- Feature 4 -->

      <div class="col">

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

          <img src="https://via.placeholder.com/400x250" class="card-img-top" alt="Feature 4">

          <div class="card-body">

            <h5 class="card-title">Empowering Local Agents</h5>

            <p class="card-text">Our platform relies on local agents who not only facilitate product delivery but also contribute to the economic growth of their communities through entrepreneurial opportunities.</p>

          </div>

        </div>

      </div>

    </div>

  </div>

  <!-- Footer -->

  <footer class="bg-dark text-white text-center py-3">

    <div class="container">

      <p>&copy; 2025 WasteNot. All rights reserved.</p>

    </div>

  </footer>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**login.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Login - WasteNot</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">

  <link href="assets/css/styles.css" rel="stylesheet">

</head>

<body>

  <div class="container mt-5">

    <div class="row justify-content-center">

      <div class="col-md-6">

        <div class="card shadow">

          <div class="card-body">

            <h2 class="text-center mb-4">Login to WasteNot</h2>

            <div id="errorMessage" class="alert alert-danger" style="display: none;"></div>

            <form id="loginForm">

              <div class="mb-3">

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

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

              </div>

              <div class="mb-3">

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

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

              </div>

              <div class="d-grid gap-2">

                <button type="submit" class="btn btn-primary" id="loginButton">

                  <span class="spinner-border spinner-border-sm d-none" role="status" aria-hidden="true" id="loginSpinner"></span>

                  <span id="loginText">Login</span>

                </button>

              </div>

            </form>

            <div class="text-center mt-3">

              <p>Don't have an account? <a href="signup.html">Sign up</a></p>

              <p><a href="#" id="forgotPassword">Forgot Password?</a></p>

            </div>

          </div>

        </div>

      </div>

    </div>

  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2"></script>

  <script type="module">

    import { createClient } from 'https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2/+esm';

    const supabaseUrl = 'https://dovyylhnscqknygyhfni.supabase.co';

    const supabaseAnonKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImRvdnl5bGhuc2Nxa255Z3loZm5pIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDcwNjk5ODksImV4cCI6MjA2MjY0NTk4OX0.FQA1FM05S4Q4qIZyrl14NKedC5RtED5ZHMIYXDYcUWQ';

    const supabase = createClient(supabaseUrl, supabaseAnonKey);

    // Check if user is already logged in

    async function checkAuth() {

      const { data: { session } } = await supabase.auth.getSession();

      if (session) {

        window.location.href = 'profile.html';

      }

    }

    // Handle login form submission

    const loginForm = document.getElementById('loginForm');

    const errorMessage = document.getElementById('errorMessage');

    const loginButton = document.getElementById('loginButton');

    const loginSpinner = document.getElementById('loginSpinner');

    const loginText = document.getElementById('loginText');

    loginForm.addEventListener('submit', async (e) => {

      e.preventDefault();

      // Show loading state

      loginButton.disabled = true;

      loginSpinner.classList.remove('d-none');

      loginText.textContent = 'Logging in...';

      errorMessage.style.display = 'none';

      const email = document.getElementById('email').value.trim();

      const password = document.getElementById('password').value;

      try {

        const { data, error } = await supabase.auth.signInWithPassword({

          email,

          password

        });

        if (error) throw error;

        // Get user profile

        const { data: profile, error: profileError } = await supabase

          .from('profiles')

          .select('\*')

          .eq('id', data.user.id)

          .single();

        if (profileError && profileError.code === 'PGRST116') {

          // Create profile if it doesn't exist

          const { error: createError } = await supabase

            .from('profiles')

            .insert([{

              id: data.user.id,

              email: data.user.email,

              full\_name: data.user.user\_metadata?.full\_name || '',

              created\_at: new Date().toISOString(),

              updated\_at: new Date().toISOString()

            }]);

          if (createError) throw createError;

        } else if (profileError) {

          throw profileError;

        }

        // Redirect to profile page after successful login

        window.location.href = 'profile.html';

      } catch (error) {

        console.error('Login error:', error);

        errorMessage.textContent = error.message || 'An error occurred during login. Please try again.';

        errorMessage.style.display = 'block';

        // Reset loading state

        loginButton.disabled = false;

        loginSpinner.classList.add('d-none');

        loginText.textContent = 'Login';

      }

    });

    // Handle forgot password

    document.getElementById('forgotPassword').addEventListener('click', async (e) => {

      e.preventDefault();

      const email = document.getElementById('email').value.trim();

      if (!email) {

        errorMessage.textContent = 'Please enter your email address first.';

        errorMessage.style.display = 'block';

        return;

      }

      try {

        const { error } = await supabase.auth.resetPasswordForEmail(email, {

          redirectTo: `${window.location.origin}/reset-password.html`,

        });

        if (error) throw error;

        errorMessage.className = 'alert alert-success';

        errorMessage.textContent = 'Password reset instructions have been sent to your email.';

        errorMessage.style.display = 'block';

      } catch (error) {

        console.error('Reset password error:', error);

        errorMessage.className = 'alert alert-danger';

        errorMessage.textContent = error.message || 'Failed to send reset instructions. Please try again.';

        errorMessage.style.display = 'block';

      }

    });

    // Check auth status when page loads

    checkAuth();

  </script>

</body>

</html>

**signup.html**

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Sign Up - WasteNot</title>

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">

  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.11.3/font/bootstrap-icons.css">

  <link rel="stylesheet" href="assets/css/style.css">

</head>

<body class="bg-light">

  <div class="container mt-5">

    <div class="text-center mb-4">

      <h1 class="display-4 text-success fw-bold">WasteNot</h1>

      <p class="lead">Join our community to make a difference</p>

    </div>

    <div class="row justify-content-center">

      <div class="col-md-6 bg-white p-4 shadow rounded">

        <div class="text-end mb-3">

          <a href="index.html" class="btn btn-outline-success">

            <i class="bi bi-house-door"></i> Back to Home

          </a>

        </div>

        <h3 class="mb-4 text-center">Create an Account</h3>

        <form id="signupForm">

          <div class="mb-3">

            <label for="fullName" class="form-label">Full Name</label>

            <input type="text" class="form-control" id="fullName" required>

          </div>

          <div class="mb-3">

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

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

          </div>

          <div class="mb-3">

            <label for="phone" class="form-label">Phone Number</label>

            <input type="tel" class="form-control" id="phone" required>

          </div>

          <div class="mb-3">

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

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

          </div>

          <div class="mb-3">

            <label for="confirmPassword" class="form-label">Confirm Password</label>

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

          </div>

          <div class="mb-3">

            <label for="role" class="form-label">Account Type</label>

            <select class="form-select" id="role" required>

              <option value="user">Regular User</option>

              <option value="agent">Delivery Agent</option>

            </select>

          </div>

          <div class="d-grid">

            <button type="submit" class="btn btn-primary">Sign Up</button>

          </div>

          <div id="errorMessage" class="alert alert-danger mt-3" style="display: none;"></div>

        </form>

        <div class="text-center mt-3">

          <small>Already have an account? <a href="login.html">Login</a></small>

        </div>

      </div>

    </div>

  </div>

  <script type="module">

    import { createClient } from 'https://cdn.jsdelivr.net/npm/@supabase/supabase-js@2/+esm';

    const supabaseUrl = 'https://dovyylhnscqknygyhfni.supabase.co';

    const supabaseAnonKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImRvdnl5bGhuc2Nxa255Z3loZm5pIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDcwNjk5ODksImV4cCI6MjA2MjY0NTk4OX0.FQA1FM05S4Q4qIZyrl14NKedC5RtED5ZHMIYXDYcUWQ';

    const supabase = createClient(supabaseUrl, supabaseAnonKey);

    const auth = supabase.auth;

    const signupForm = document.getElementById('signupForm');

    const errorMessage = document.getElementById('errorMessage');

    signupForm.addEventListener('submit', async (e) => {

      e.preventDefault();

      const fullName = document.getElementById('fullName').value;

      const email = document.getElementById('email').value;

      const phone = document.getElementById('phone').value;

      const password = document.getElementById('password').value;

      const confirmPassword = document.getElementById('confirmPassword').value;

      const role = document.getElementById('role').value;

      // Validate passwords match

      if (password !== confirmPassword) {

        errorMessage.textContent = 'Passwords do not match';

        errorMessage.style.display = 'block';

        return;

      }

      try {

        // First create the auth user

        const { data: { user }, error: authError } = await auth.signUp({

          email,

          password,

          options: {

            data: {

              full\_name: fullName,

              phone: phone,

              role: role

            }

          }

        });

        if (authError) throw authError;

        // Then create the profile record

        if (user) {

          const { error: profileError } = await supabase

            .from('profiles')

            .insert([

              {

                id: user.id,

                full\_name: fullName,

                phone: phone,

                role: role,

                created\_at: new Date(),

                updated\_at: new Date()

              }

            ]);

          if (profileError) throw profileError;

        }

        // Show success message

        errorMessage.textContent = 'Registration successful! Please check your email to verify your account.';

        errorMessage.className = 'alert alert-success mt-3';

        errorMessage.style.display = 'block';

        // Clear the form

        signupForm.reset();

      } catch (error) {

        errorMessage.className = 'alert alert-danger mt-3';

        errorMessage.textContent = error.message;

        errorMessage.style.display = 'block';

      }

    });

    // Check if user is already logged in

    window.addEventListener('load', async () => {

      const { data: { session } } = await auth.getSession();

      if (session) {

        window.location.href = '/index.html';

      }

    });

  </script>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>

</html>

**style.css**

/\* Add smooth scrolling \*/

html {

  scroll-behavior: smooth;

}

/\* Custom hover effects for cards \*/

.card {

  transition: transform 0.2s ease-in-out;

}

.card:hover {

  transform: translateY(-5px);

}

/\* Custom styles for buttons \*/

.btn {

  transition: all 0.2s ease-in-out;

}

.btn:hover {

  transform: scale(1.05);

}

/\* Hero section enhancements \*/

.bg-success {

  background: linear-gradient(45deg, #198754, #20c997) !important;

}

/\* Stats section animations \*/

.display-4 {

  transition: color 0.3s ease;

}

.display-4:hover {

  color: #20c997 !important;

}

/\* Footer link hover effects \*/

footer a {

  transition: opacity 0.2s ease;

}

footer a:hover {

  opacity: 0.8;

}

/\* Responsive font sizes \*/

@media (max-width: 768px) {

  .display-4 {

    font-size: 2.5rem;

  }

  .lead {

    font-size: 1.1rem;

  }

}

**index.js**

import express from 'express';

import cors from 'cors';

import dotenv from 'dotenv';

import { createClient } from '@supabase/supabase-js';

import { requireAuth, requireRole } from './middleware/auth.js';

import path from 'path';

import { fileURLToPath } from 'url';

// Import routes

import productsRouter from './routes/products.js';

import donationsRouter from './routes/donations.js';

import agentsRouter from './routes/agents.js';

// Load environment variables

dotenv.config();

const app = express();

// Get directory name in ES module

const \_\_filename = fileURLToPath(import.meta.url);

const \_\_dirname = path.dirname(\_\_filename);

// Middleware

app.use(cors());

app.use(express.json());

app.use(express.urlencoded({ extended: true }));

// Serve static files from frontend directory

app.use(express.static(path.join(\_\_dirname, '../frontend')));

// Create Supabase client

const supabaseUrl = process.env.SUPABASE\_URL || 'https://your-project-url.supabase.co';

const supabaseKey = process.env.SUPABASE\_ANON\_KEY || 'your-anon-key';

const supabase = createClient(supabaseUrl, supabaseKey);

// Serve index.html for the root route

app.get('/', (req, res) => {

    res.sendFile(path.join(\_\_dirname, '../frontend/index.html'));

});

// API routes

app.use('/api/products', productsRouter);

app.use('/api/donations', donationsRouter);

app.use('/api/agents', agentsRouter);

// Protected routes example

app.get('/api/profile', requireAuth, async (req, res) => {

    try {

        const { data, error } = await supabase

            .from('users')

            .select('\*')

            .eq('id', req.user.id)

            .single();

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Profile error:', error);

        res.status(500).json({ message: 'Error fetching profile' });

    }

});

// Agent-only route example

app.get('/api/agent/dashboard', requireAuth, requireRole(['agent']), async (req, res) => {

    try {

        const { data, error } = await supabase

            .from('donations')

            .select('\*')

            .eq('assigned\_agent\_id', req.user.id);

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Agent dashboard error:', error);

        res.status(500).json({ message: 'Error fetching agent dashboard' });

    }

});

// Database test route

app.get('/api/test', async (req, res) => {

    try {

        const { data, error } = await supabase

            .from('users')

            .select('id')

            .limit(1);

        if (error) throw error;

        res.json({

            success: true,

            message: 'Database connection successful!',

            timestamp: new Date().toISOString()

        });

    } catch (error) {

        console.error('Database connection error:', error);

        res.status(500).json({

            success: false,

            message: 'Database connection failed',

            error: error.message

        });

    }

});

// Catch-all route to serve index.html for client-side routing

app.get('\*', (req, res) => {

    res.sendFile(path.join(\_\_dirname, '../frontend/index.html'));

});

// Error handling middleware

app.use((err, req, res, next) => {

    console.error(err.stack);

    res.status(500).json({

        message: 'Something went wrong!',

        error: process.env.NODE\_ENV === 'development' ? err.message : undefined

    });

});

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

    console.log(`Server is running on port ${PORT}`);

    console.log(`Frontend available at: http://localhost:${PORT}`);

    console.log(`API Documentation available at: http://localhost:${PORT}/api`);

});

**auth.js**

// Initialize Supabase client

const supabaseUrl = 'https://dovyylhnscqknygyhfni.supabase.co';

const supabaseAnonKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImRvdnl5bGhuc2Nxa255Z3loZm5pIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDcwNjk5ODksImV4cCI6MjA2MjY0NTk4OX0.FQA1FM05S4Q4qIZyrl14NKedC5RtED5ZHMIYXDYcUWQ';

const supabase = createClient(supabaseUrl, supabaseAnonKey);

// Check authentication status and update UI

async function checkAuthAndUpdateUI() {

  try {

    const { data: { session }, error: sessionError } = await supabase.auth.getSession();

    if (sessionError) throw sessionError;

    const { data: { user }, error: userError } = await supabase.auth.getUser();

    if (userError) throw userError;

    const loginButton = document.getElementById('loginButton');

    const userProfile = document.getElementById('userProfile');

    const userName = document.getElementById('userName');

    const userDropdown = document.getElementById('userDropdown');

    if (!loginButton || !userProfile || !userName) return; // Skip if elements don't exist

    if (user && session) {

      // User is logged in

      loginButton.classList.add('d-none');

      userProfile.classList.remove('d-none');

      if (userDropdown) userDropdown.classList.remove('d-none');

      // Get user profile data

      try {

        const { data, error: profileError } = await supabase

          .from('profiles')

          .select('full\_name, role')

          .eq('id', user.id)

          .single();

        if (profileError) throw profileError;

        if (data && data.full\_name) {

          userName.textContent = data.full\_name;

        } else {

          userName.textContent = user.email;

        }

        // Update navigation based on user role

        if (data.role === 'agent') {

          const agentNav = document.getElementById('agentNav');

          if (agentNav) agentNav.classList.remove('d-none');

        }

      } catch (err) {

        console.error('Error fetching profile:', err);

        userName.textContent = user.email;

      }

    } else {

      // User is not logged in

      loginButton.classList.remove('d-none');

      userProfile.classList.add('d-none');

      if (userDropdown) userDropdown.classList.add('d-none');

      // Hide role-specific navigation

      const agentNav = document.getElementById('agentNav');

      if (agentNav) agentNav.classList.add('d-none');

      // Redirect to login if on protected page

      const protectedPages = ['profile.html', 'my-orders.html', 'my-donations.html'];

      const currentPage = window.location.pathname.split('/').pop();

      if (protectedPages.includes(currentPage)) {

        window.location.href = 'login.html';

      }

    }

  } catch (error) {

    console.error('Auth check error:', error);

    // Handle error gracefully - show error message to user if needed

  }

}

// Handle logout

async function setupLogout() {

  const logoutBtn = document.getElementById('logoutBtn');

  if (logoutBtn) {

    logoutBtn.addEventListener('click', async () => {

      try {

        const { error } = await supabase.auth.signOut();

        if (error) throw error;

        // Clear any stored auth state

        localStorage.removeItem('supabase.auth.token');

        // Redirect to home page

        window.location.href = 'index.html';

      } catch (error) {

        console.error('Logout error:', error);

        alert('Error logging out. Please try again.');

      }

    });

  }

}

// Initialize auth checking

document.addEventListener('DOMContentLoaded', () => {

  checkAuthAndUpdateUI();

  setupLogout();

  // Listen for auth state changes

  supabase.auth.onAuthStateChange((event, session) => {

    if (event === 'SIGNED\_IN' || event === 'SIGNED\_OUT') {

      checkAuthAndUpdateUI();

    }

  });

});

// Export functions for use in other files

export { checkAuthAndUpdateUI, setupLogout };

supabaseClient.js

import { createClient } from '@supabase/supabase-js';

import dotenv from 'dotenv';

dotenv.config();

const supabaseUrl = process.env.SUPABASE\_URL || 'https://dovyylhnscqknygyhfni.supabase.co';

const supabaseKey = process.env.SUPABASE\_ANON\_KEY || 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJlZiI6ImRvdnl5bGhuc2Nxa255Z3loZm5pIiwicm9sZSI6ImFub24iLCJpYXQiOjE3NDcwNjk5ODksImV4cCI6MjA2MjY0NTk4OX0.FQA1FM05S4Q4qIZyrl14NKedC5RtED5ZHMIYXDYcUWQ';

const supabase = createClient(supabaseUrl, supabaseKey);

export { supabase };

export const auth = supabase.auth;

**agents.js**

import express from 'express';

import { requireAuth, requireRole } from '../middleware/auth.js';

import { supabase } from '../config/supabaseClient.js';

const router = express.Router();

*// Get all agents (public)*

router.get('/', async (*req*, *res*) => {

    try {

        const { data, error } = await supabase

            .from('agents')

            .select(`

                \*,

                users:user\_id (

                    full\_name,

                    email,

                    phone

                )

            `)

            .eq('verification\_status', 'verified');

        if (error) throw error;

*res*.json(data);

    } catch (error) {

        console.error('Error fetching agents:', error);

*res*.status(500).json({ message: 'Error fetching agents' });

    }

});

*// Get single agent (public)*

router.get('/:id', async (*req*, *res*) => {

    try {

        const { data, error } = await supabase

            .from('agents')

            .select(`

                \*,

                users:user\_id (

                    full\_name,

                    email,

                    phone

                )

            `)

            .eq('id', *req*.params.id)

            .single();

        if (error) throw error;

        if (!data) {

            return *res*.status(404).json({ message: 'Agent not found' });

        }

*res*.json(data);

    } catch (error) {

        console.error('Error fetching agent:', error);

*res*.status(500).json({ message: 'Error fetching agent' });

    }

});

*// Register as agent (requires authentication)*

router.post('/register', requireAuth, async (*req*, *res*) => {

    try {

        const { area\_of\_operation, vehicle\_type } = *req*.body;

*// Check if user is already an agent*

        const { data: existingAgent } = await supabase

            .from('agents')

            .select('id')

            .eq('user\_id', req.user.id)

            .single();

        if (existingAgent) {

            return res.status(400).json({ message: 'User is already registered as an agent' });

        }

        const { data, error } = await supabase

            .from('agents')

            .insert([

                {

                    user\_id: req.user.id,

                    area\_of\_operation,

                    vehicle\_type,

                    verification\_status: 'pending'

                }

            ])

            .select()

            .single();

        if (error) throw error;

        res.status(201).json(data);

    } catch (error) {

        console.error('Error registering agent:', error);

        res.status(500).json({ message: 'Error registering agent' });

    }

});

*// Update agent profile (requires authentication and agent role)*

router.put('/profile', requireAuth, requireRole(['agent']), async (*req*, *res*) => {

    try {

        const { area\_of\_operation, vehicle\_type, availability\_status } = req.body;

        const { data, error } = await supabase

            .from('agents')

            .update({

                area\_of\_operation,

                vehicle\_type,

                availability\_status,

                updated\_at: new Date()

            })

            .eq('user\_id', req.user.id)

            .select()

            .single();

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Error updating agent profile:', error);

        res.status(500).json({ message: 'Error updating agent profile' });

    }

});

*// Get agent statistics (requires authentication and agent role)*

router.get('/stats/me', requireAuth, requireRole(['agent']), async (*req*, *res*) => {

    try {

        const { data: agent, error: agentError } = await supabase

            .from('agents')

            .select('rating, total\_pickups')

            .eq('user\_id', req.user.id)

            .single();

        if (agentError) throw agentError;

        const { data: recentDonations, error: donationsError } = await supabase

            .from('donations')

            .select('\*')

            .eq('assigned\_agent\_id', req.user.id)

            .order('created\_at', { ascending: false })

            .limit(5);

        if (donationsError) throw donationsError;

        res.json({

            stats: agent,

            recent\_donations: recentDonations

        });

    } catch (error) {

        console.error('Error fetching agent statistics:', error);

        res.status(500).json({ message: 'Error fetching agent statistics' });

    }

});

*// Update agent availability (requires authentication and agent role)*

router.patch('/availability', requireAuth, requireRole(['agent']), async (*req*, *res*) => {

    try {

        const { availability\_status } = req.body;

        const validStatuses = ['available', 'busy', 'offline'];

        if (!validStatuses.includes(availability\_status)) {

            return res.status(400).json({ message: 'Invalid availability status' });

        }

        const { data, error } = await supabase

            .from('agents')

            .update({ availability\_status })

            .eq('user\_id', req.user.id)

            .select()

            .single();

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Error updating availability:', error);

        res.status(500).json({ message: 'Error updating availability' });

    }

});

export default router;

**donation.js**

import express from 'express';

import { requireAuth, requireRole } from '../middleware/auth.js';

import { supabase } from '../config/supabaseClient.js';

const router = express.Router();

*// Get all donations*

router.get('/', async (*req*, *res*) => {

    try {

        const { status, category } = *req*.query;

        let query = supabase.from('donations').select('\*');

        if (status) {

            query = query.eq('status', status);

        }

        if (category) {

            query = query.eq('category', category);

        }

        const { data, error } = await query;

        if (error) throw error;

*res*.json(data);

    } catch (error) {

        console.error('Error fetching donations:', error);

*res*.status(500).json({ message: 'Error fetching donations' });

    }

});

*// Get single donation*

router.get('/:id', async (*req*, *res*) => {

    try {

        const { data, error } = await supabase

            .from('donations')

            .select('\*')

            .eq('id', *req*.params.id)

            .single();

        if (error) throw error;

        if (!data) {

            return *res*.status(404).json({ message: 'Donation not found' });

        }

*res*.json(data);

    } catch (error) {

        console.error('Error fetching donation:', error);

*res*.status(500).json({ message: 'Error fetching donation' });

    }

});

*// Create donation (requires authentication)*

router.post('/', requireAuth, async (*req*, *res*) => {

    try {

        const {

            item\_name,

            description,

            category,

            quantity,

            pickup\_location,

            pickup\_date,

            image\_url,

            notes

        } = *req*.body;

        const { data, error } = await supabase

            .from('donations')

            .insert([

                {

                    donor\_id: req.user.id,

                    item\_name,

                    description,

                    category,

                    quantity,

                    pickup\_location,

                    pickup\_date,

                    image\_url,

                    notes,

                    status: 'pending'

                }

            ])

            .select()

            .single();

        if (error) throw error;

        res.status(201).json(data);

    } catch (error) {

        console.error('Error creating donation:', error);

        res.status(500).json({ message: 'Error creating donation' });

    }

});

*// Update donation status (requires agent role)*

router.patch('/:id/status', requireAuth, requireRole(['agent']), async (*req*, *res*) => {

    try {

        const { status } = req.body;

        const validStatuses = ['pending', 'accepted', 'picked\_up', 'completed', 'cancelled'];

        if (!validStatuses.includes(status)) {

            return res.status(400).json({ message: 'Invalid status' });

        }

        const { data, error } = await supabase

            .from('donations')

            .update({

                status,

                assigned\_agent\_id: req.user.id,

                updated\_at: new Date()

            })

            .eq('id', req.params.id)

            .select()

            .single();

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Error updating donation status:', error);

        res.status(500).json({ message: 'Error updating donation status' });

    }

});

*// Get my donations (requires authentication)*

router.get('/user/my-donations', requireAuth, async (*req*, *res*) => {

    try {

        const { data, error } = await supabase

            .from('donations')

            .select('\*')

            .eq('donor\_id', req.user.id)

            .order('created\_at', { ascending: false });

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Error fetching user donations:', error);

        res.status(500).json({ message: 'Error fetching user donations' });

    }

});

*// Get assigned donations (requires agent role)*

router.get('/agent/assigned', requireAuth, requireRole(['agent']), async (*req*, *res*) => {

    try {

        const { data, error } = await supabase

            .from('donations')

            .select('\*')

            .eq('assigned\_agent\_id', req.user.id)

            .order('created\_at', { ascending: false });

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Error fetching assigned donations:', error);

        res.status(500).json({ message: 'Error fetching assigned donations' });

    }

});

export default router;

**product.js**

import express from 'express';

import { requireAuth, requireRole } from '../middleware/auth.js';

import { supabase } from '../config/supabaseClient.js';

const router = express.Router();

*// Get all products*

router.get('/', async (*req*, *res*) => {

    try {

        const { category, search, sort } = *req*.query;

        let query = supabase.from('products').select('\*');

        if (category) {

            query = query.eq('category', category);

        }

        if (search) {

            query = query.ilike('name', `%${search}%`);

        }

        if (sort) {

            const [field, order] = sort.split(':');

            query = query.order(field, { ascending: order === 'asc' });

        }

        const { data, error } = await query;

        if (error) throw error;

*res*.json(data);

    } catch (error) {

        console.error('Error fetching products:', error);

*res*.status(500).json({ message: 'Error fetching products' });

    }

});

router.get('/:id', async (*req*, *res*) => {

    try {

        const { data, error } = await supabase

            .from('products')

            .select('\*')

            .eq('id', *req*.params.id)

            .single();

        if (error) throw error;

        if (!data) {

            return *res*.status(404).json({ message: 'Product not found' });

        }

*res*.json(data);

    } catch (error) {

        console.error('Error fetching product:', error);

*res*.status(500).json({ message: 'Error fetching product' });

    }

});

*// Create product (requires authentication)*

router.post('/', requireAuth, async (*req*, *res*) => {

    try {

        const { name, description, price, category, stock, image\_url } = *req*.body;

        const { data, error } = await supabase

            .from('products')

            .insert([

                {

                    name,

                    description,

                    price,

                    category,

                    stock,

                    image\_url,

                    seller\_id: req.user.id

                }

            ])

            .select()

            .single();

        if (error) throw error;

        res.status(201).json(data);

    } catch (error) {

        console.error('Error creating product:', error);

        res.status(500).json({ message: 'Error creating product' });

    }

});

*// Update product (requires authentication and ownership)*

router.put('/:id', requireAuth, async (*req*, *res*) => {

    try {

*// Check ownership*

        const { data: existingProduct } = await supabase

            .from('products')

            .select('seller\_id')

            .eq('id', req.params.id)

            .single();

        if (!existingProduct || existingProduct.seller\_id !== req.user.id) {

            return res.status(403).json({ message: 'Not authorized to update this product' });

        }

        const { name, description, price, category, stock, image\_url } = req.body;

        const { data, error } = await supabase

            .from('products')

            .update({

                name,

                description,

                price,

                category,

                stock,

                image\_url,

                updated\_at: new Date()

            })

            .eq('id', req.params.id)

            .select()

            .single();

        if (error) throw error;

        res.json(data);

    } catch (error) {

        console.error('Error updating product:', error);

        res.status(500).json({ message: 'Error updating product' });

    }

});

*// Delete product (requires authentication and ownership)*

router.delete('/:id', requireAuth, async (*req*, *res*) => {

    try {

*// Check ownership*

        const { data: existingProduct } = await supabase

            .from('products')

            .select('seller\_id')

            .eq('id', req.params.id)

            .single();

        if (!existingProduct || existingProduct.seller\_id !== req.user.id) {

            return res.status(403).json({ message: 'Not authorized to delete this product' });

        }

const { error } = await supabase

            .from('products')

            .delete()

            .eq('id', req.params.id);

        if (error) throw error;

        res.json({ message: 'Product deleted successfully' });

    } catch (error) {

        console.error('Error deleting product:', error);

        res.status(500).json({ message: 'Error deleting product' });

    }

});

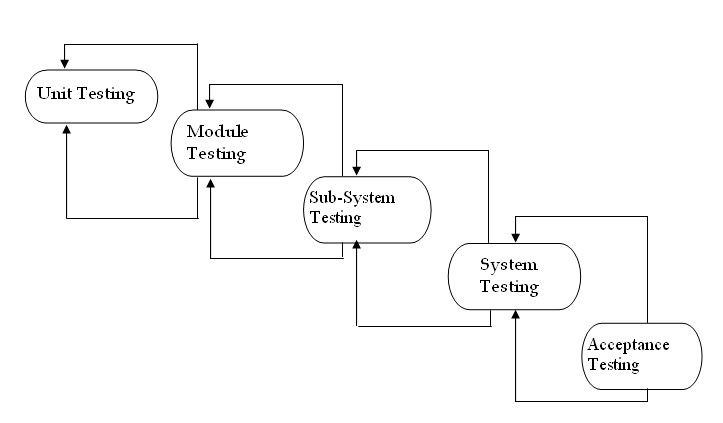
export default router;

# SOFTWARE TESTING

## 1.TESTING OBJECTIVES

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

## 2. TESTING METHODOLOGIE



### **Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

### 

### **Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre‐driven process links and integration points.

### **White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

### **Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

### **Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation

by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**CONCLUSION**

WasteNot is an innovative e-commerce and logistics platform that connects rural communities with essential goods through local agents. By using AI for demand forecasting, multi-modal logistics, and eco-friendly delivery, it ensures efficient, affordable access to products. With flexible payment options, micro-financing, and a focus on sustainability through surplus goods donation, WasteNot empowers rural entrepreneurs, strengthens local economies, and reduces environmental impact, making it a scalable and impactful solution for rural commerce.

**BIBLIOGRAPHY**

1. w3school.com

2. www.greakforgreaak .com

3. [www.youtube.com](http://www.youtube.com)

4. [www.google.com](http://www.google.com)

5.github

6. [HTML, CSS, and Javascript for Web Developers](https://www.classcentral.com/course/html-css-javascript-for-web-developers-4270?utm_source=fcc_medium&utm_medium=web&utm_campaign=cs_programming_2023) from *Johns Hopkins*

7. Learning Web Design: A Beginner’s Guide to HTML, CSS, JavaScript, and Web Graphics

8. reference by <https://cid.karnataka.gov.in/en>