



**CSE471: System Analysis and Design Project Report**  
**Project Title: Online Auction System**

<b>Group No: 02, CSE471 Lab Section: 03, Spring 2024</b>	
<b>ID</b>	<b>Name</b>
<b>20101232</b>	<b>Sopnil Roy Niloy</b>
<b>21301425</b>	<b>Afif Rayhan Pranto</b>
<b>23241108</b>	<b>Rejwan Shafi</b>
<b>24141198</b>	<b>Homairah Ferdousia</b>

## Table of Contents

Section No	Content	Page No
1	Introduction	2
2	Functional Requirement	4
3	User Manual	5
4	Diagrams	6
4	Frontend Development	10
5	Backend Development	44
6	Database	69
7	Technology (Framework, Language)	75
8	GitHub Repo Link	75
9	Individual Contribution	76

## Introduction

**"Nilam: An Online Auction System"** is a website developed using the Laravel framework that provides a platform for users to engage in online auctions. This system is designed to facilitate the smooth buying and selling of goods through a simple and user-friendly interface, enabling users to participate in auctions, place bids, and manage their profiles with ease. **Nilam** is equipped with robust features aimed at providing a secure, efficient, and transparent auctioning experience for both buyers and sellers. The project is divided into four core modules: User Management, Auction Management, Bidding and Search, and Administration. Each module plays a crucial role in ensuring the smooth functioning of the system. The **User Management** module handles all aspects of user authentication, including Google Login integration, profile management, password recovery, and multi-authentication capabilities, to ensure secure access to the platform. The **Auction Management** module enables users to create, edit, and delete auctions, while also managing the auction's approval process and timer functionality, providing a streamlined auction setup for sellers. The **Bidding and Search** module focuses on the core auction activities, allowing users to place bids on available items, view bid histories, and search for specific auctions or items, enhancing user engagement and experience. Lastly, the **Administration** module offers a powerful dashboard for administrators to manage users, oversee auction activities, and handle seller approvals, maintaining overall platform integrity and ensuring fair practices. Utilizing the Laravel framework, our website delivers an easy and dynamic solution for online auctions, useful to casual users and professional sellers in a secure and controlled environment. This report provides an in-depth overview of the system's design and implementation, highlighting the functionalities and technical aspects of each module along with the necessary diagrams.

## **Functional Requirements**

### **Module 1: *User Management***

1. User Authentication
2. Social Media Authentication
3. User Profile Management
4. Password Recovery
5. Multi Authentication

### **Module 2: *Auction Management***

1. Create Auction
2. Edit Auction
3. Delete Auction
4. Auction Timer
5. Auction Approval

### **Module 3: *Bidding and Search***

1. Place Bid
2. Bid History
3. Search Bar

### **Module 4: *Administration***

1. Admin Dashboard
2. User management for Admins
3. Auction Management for Admins
4. Seller Approval

### **Module 5: *Payment and Transaction***

1. Payment Gateway Integration
2. Transaction Confirmation
3. Transaction History
4. Payment Status Notification

## User Manual

First, make sure that Composer is installed on your device. After installing Composer write the following commands one by one to set up the project to run on your PC.

*composer global require laravel/installer*

*laravel new project*

*cd project*

*composer require laravel/breeze --dev*

*php artisan breeze:install*

After running these commands it will ask some questions, the answers are as follows,

choose blade

dark mode support no

phpunit

*npm install*

*npm run build*

*composer require laravel/socialite*

*composer require laravel/ui*

*php artisan ui bootstrap*

*php artisan ui bootstrap --auth*

*npm install bootstrap*

*npm install*

*npm run build*

*git clone <https://github.com/RejwanShafi/nilam-an-online-auction-system.git>*

*cd nilam-an-online-auction-system*

*git fetch*

*php artisan migrate*

Diagrams

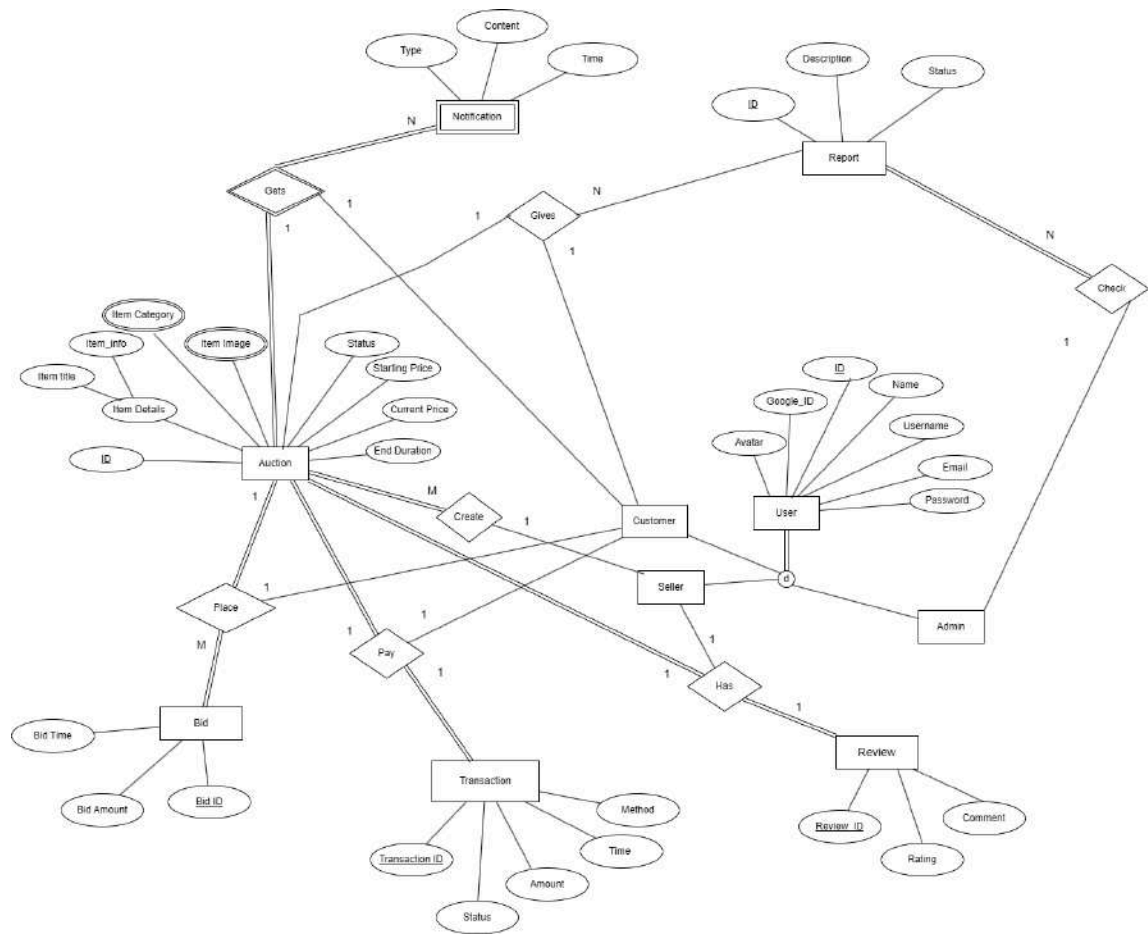


Figure: EER Diagram

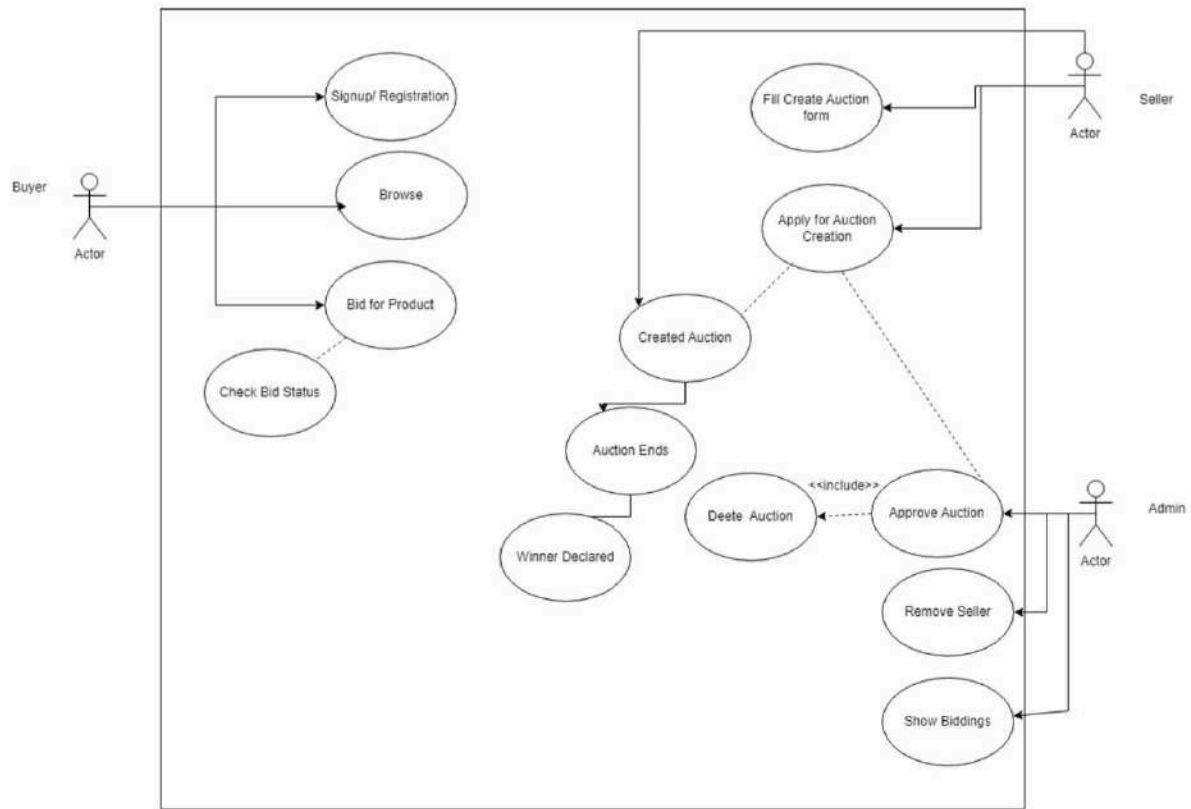


Figure: Use Case Diagram

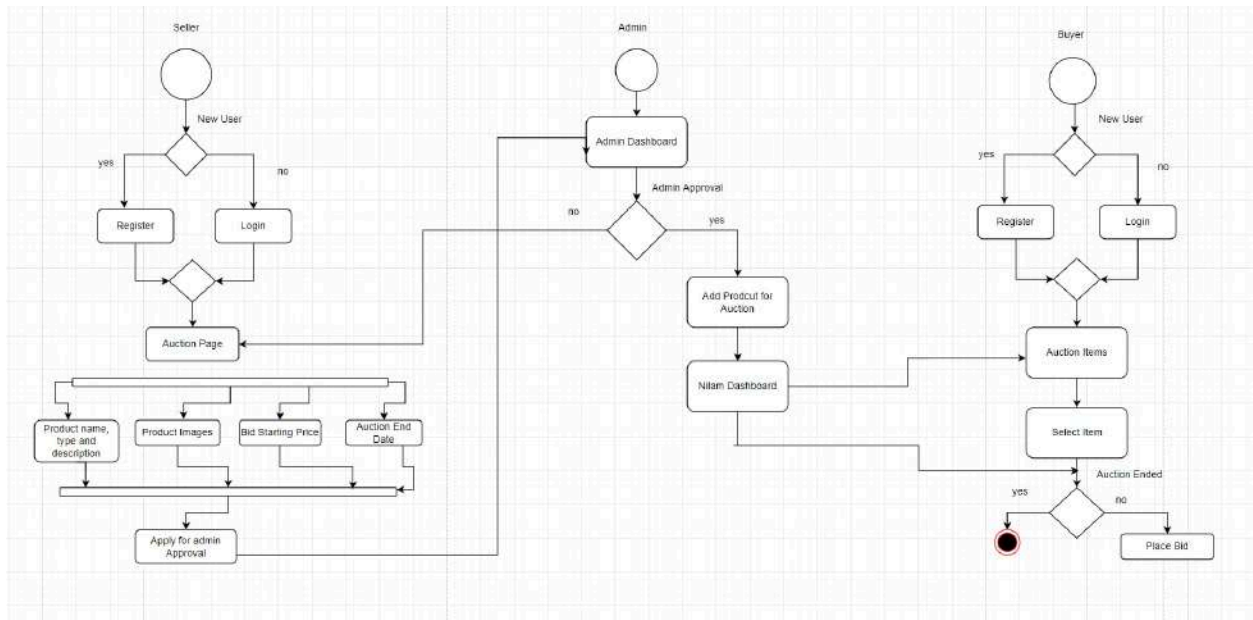


Figure: Activity Diagram

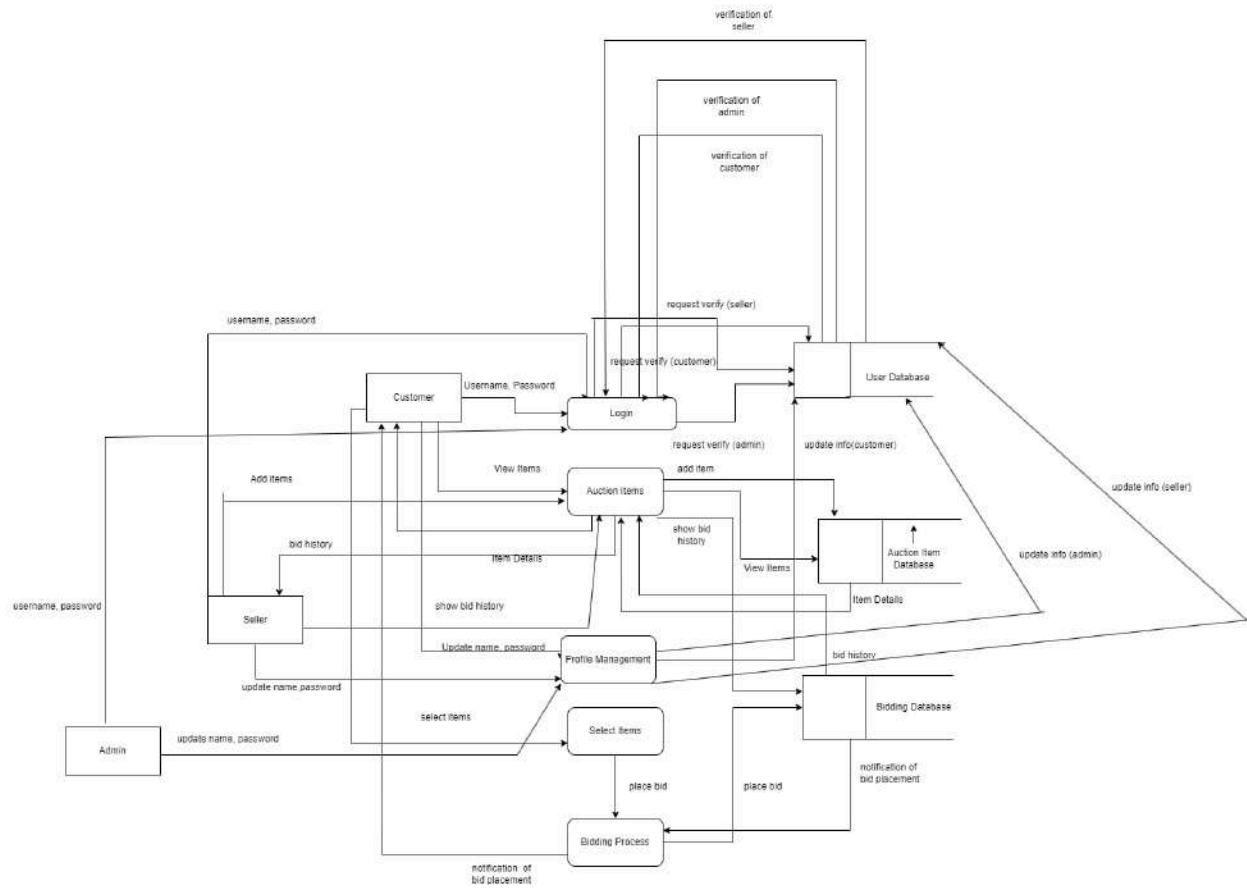


Figure: Dataflow Diagram



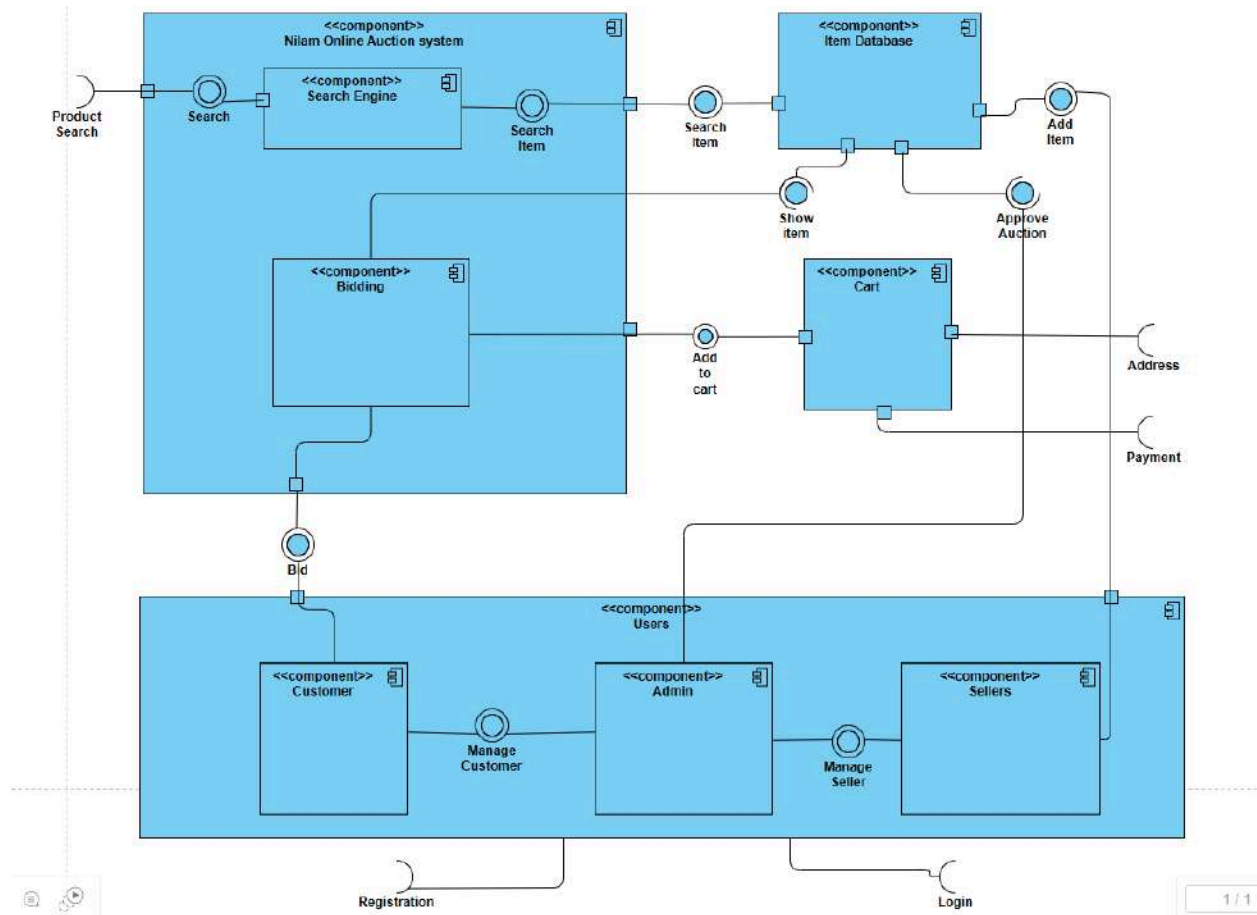
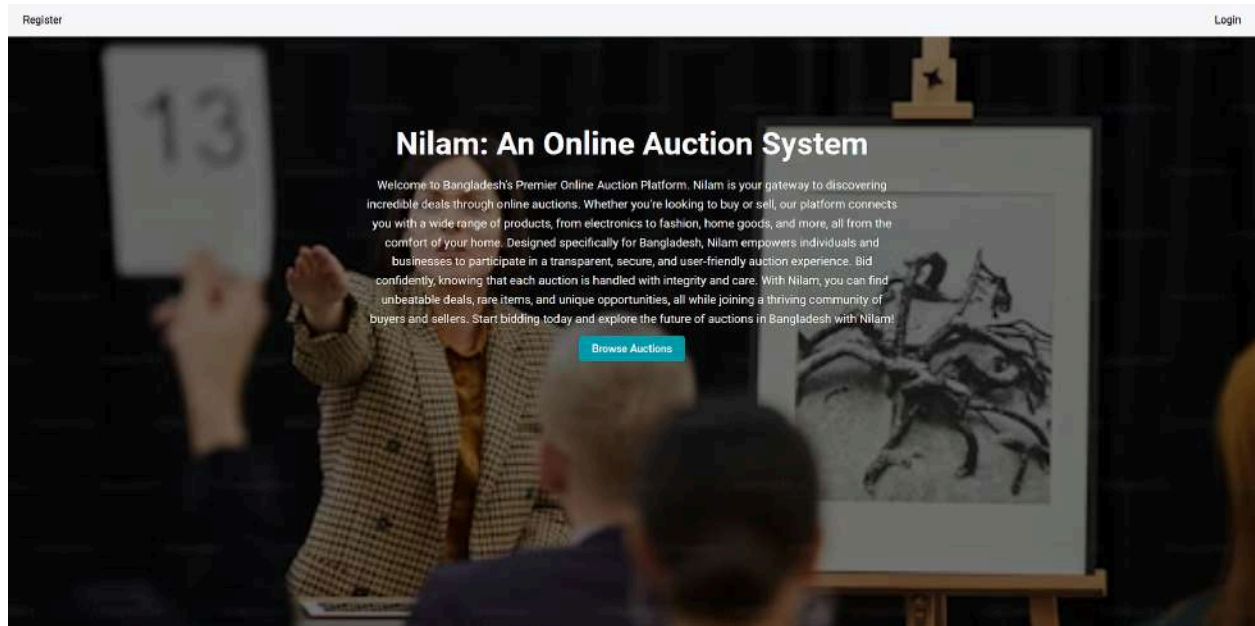


Figure: Component Diagram

## Frontend Development

**Welcome Page:** This is the welcome page view

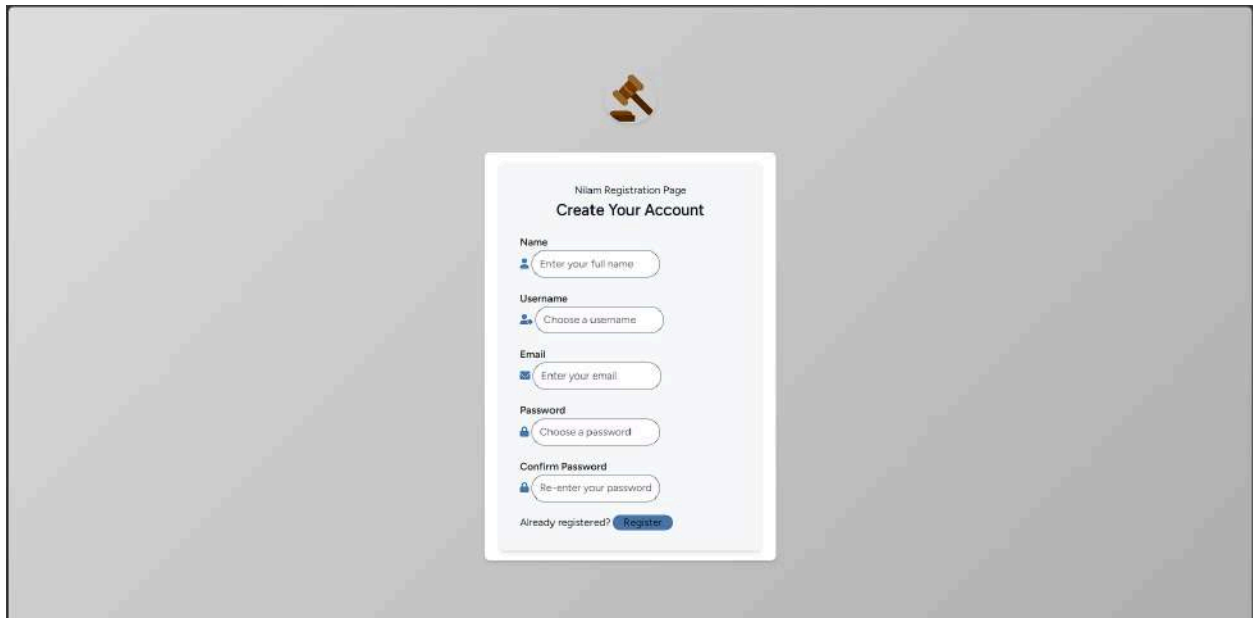


Following is the code for the welcome page

```
<body>
  <!-- Navbar -->
  <div class="navbar">
    @if(auth()->check())
      <div class="menu">
        <a href="{{ route(name: 'logout') }}"
          onclick="event.preventDefault(); document.getElementById('logout-form').submit();"
          Logout
        </a>
        <form id="logout-form" action="{{ route(name: 'logout') }}" method="POST" style="display: none;"
          @csrf
        </form>
      </div>
    @else
      <div class="logo">
        <a href="/register">Register</a>
      </div>
      <div class="menu">
        <a href="/login">Login</a>
      </div>
    @endif
  </div>

  <!-- Hero Section -->
  <div class="hero">
    <h1>Nilam: An Online Auction System</h1>
    <p>Welcome to Bangladesh's Premier Online Auction Platform. Nilam is your gateway to discovering incredible deals through online auctions. Whether you're looking
    <a href="/dashboard" class="btn">Browse Auctions</a>
  </div>
```

**Register Page:** This is the Register Page view



The screenshot shows a registration form titled "Niam Registration Page" and "Create Your Account". The form is centered on a gray background with a subtle geometric pattern. Above the form is a small icon of a gavel. The form contains the following fields and elements:

- Name:** A text input field with the placeholder "Enter your full name".
- Username:** A text input field with the placeholder "Choose a username".
- Email:** A text input field with the placeholder "Enter your email".
- Password:** A text input field with the placeholder "Choose a password".
- Confirm Password:** A text input field with the placeholder "Re-enter your password".
- Already registered?** A link labeled "Register" in a blue button.

Following are the code for register page

```

</x-guest-layout>
<div class="container mt-5 d-flex justify-content-center">
  <div class="col-lg-6 col-md-8 col-sm-12 register-container">
    <div style="text-align: center;"><span>Nilam Registration Page</span></div>

    <h2 class="form-title">Create Your Account</h2>
    <form method="POST" action="{ { route(name: 'register') } }">
      @csrf

      <!-- Name -->
      <div class="form-group">
        <label for="name" class="form-label">Name</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-user"></i></span>
          <input id="name" class="form-control @error('name') is-invalid @enderror" type="text" name="name" value="{ { old(key: 'name') } }" required autofocus autocomplete="name">
        </div>
        @error('name')
          <small class="text-danger">{{ $message }}</small>
        @enderror
      </div>

      <!-- Username -->
      <div class="form-group">
        <label for="username" class="form-label">Username</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-user-tag"></i></span>
          <input id="username" class="form-control @error('username') is-invalid @enderror" type="text" name="username" value="{ { old(key: 'username') } }" required placeholder="Username">
        </div>
        @error('username')
          <small class="text-danger">{{ $message }}</small>
        @enderror
      </div>

      <!-- Email Address -->
      <div class="form-group">
        <label for="email" class="form-label">Email</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-envelope"></i></span>
          <input id="email" class="form-control @error('email') is-invalid @enderror" type="email" name="email" value="{ { old(key: 'email') } }" required autocomplete="email">
        </div>
        @error('email')
          <small class="text-danger">{{ $message }}</small>
        @enderror
      </div>

      <!-- Password -->
      <div class="form-group">
        <label for="password" class="form-label">Password</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-lock"></i></span>
          <input id="password" class="form-control @error('password') is-invalid @enderror" type="password" name="password" required autocomplete="new-password" placeholder="Password">
        </div>
        @error('password')
          <small class="text-danger">{{ $message }}</small>
        @enderror
      </div>

      <!-- Confirm Password -->
      <div class="form-group">
        <label for="password_confirmation" class="form-label">Confirm Password</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-lock"></i></span>
          <input id="password_confirmation" class="form-control" type="password" name="password_confirmation" required autocomplete="new-password" placeholder="Re-enter your password">
        </div>
      </div>

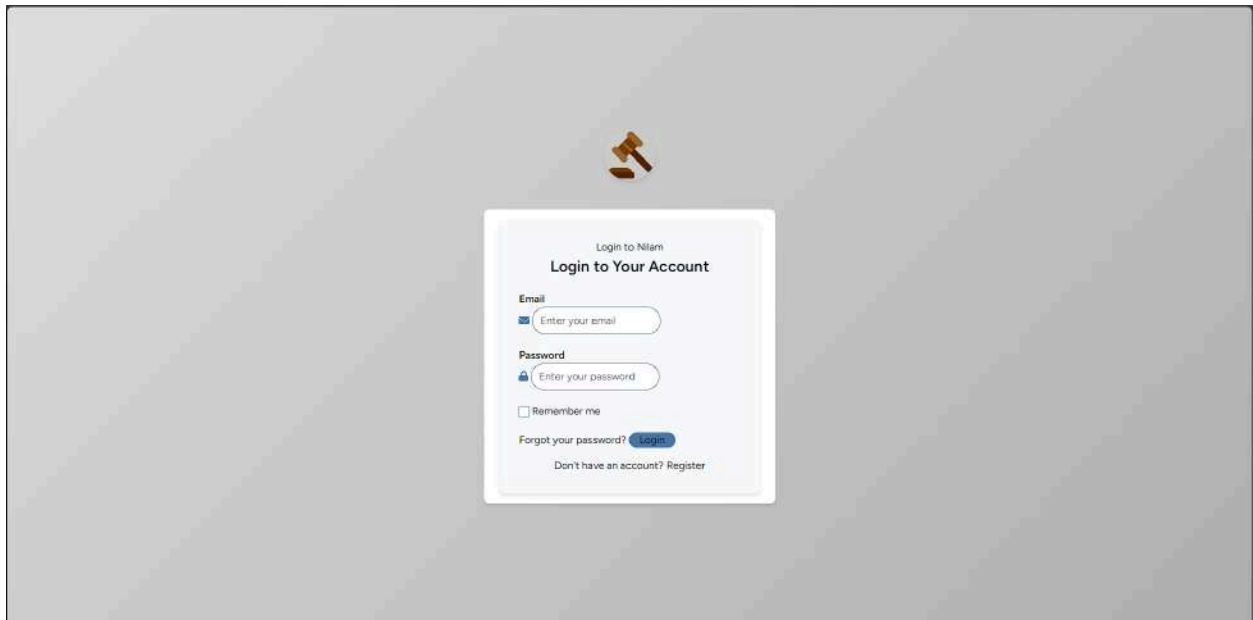
      <input type="hidden" name="role" value="3">

      <div class="d-flex justify-content-between mt-4">
        <a class="text-decoration-none underline-link" href="{ { route(name: 'login') } }">Already registered?</a>
        <button type="submit" class="btn btn-primary px-4">Register</button>
      </div>
    </form>
  </div>
</div>

<!-- Include FontAwesome for icons -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.8.0-beta3/js/all.min.js"></script>
</x-guest-layout>

```

**Login Page:** This is the Login Page view

The image shows a login page with a light gray background featuring a subtle diagonal line pattern. At the top center, there is a small icon of a wooden gavel. Below this, a white login form is centered. The form has a title 'Login to Nilam' and a subtitle 'Login to Your Account'. It contains two input fields: 'Email' with a blue envelope icon and 'Password' with a blue lock icon. Below the password field is a 'Remember me' checkbox. At the bottom of the form, there is a link 'Forgot your password?' and a blue 'Login' button. Below the login button, there is a link 'Don't have an account? Register'.

Following is the code for the login page

```

<div class="container mt-5 d-flex justify-content-center">
  <div class="col-lg-6 col-md-8 col-sm-12 login-container">
    <div style="text-align: center;"><span>Login to Nilam</span></div>

    <h2 class="form-title">Login to Your Account</h2>
    <form method="POST" action="{ { route(name: 'login') } }">
      @csrf

      <!-- Email Address -->
      <div class="form-group">
        <label for="email" class="form-label">Email</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-envelope"></i></span>
          <input id="email" class="form-control @error('email') is-invalid @enderror" type="email" name="email" value="{ { old(key: 'email') } }" required autofocus autocomplete="email">
        </div>
        @error('email')
          <small class="text-danger">{{ $message }}</small>
        @enderror
      </div>

      <!-- Password -->
      <div class="form-group">
        <label for="password" class="form-label">Password</label>
        <div class="input-group">
          <span class="input-group-text form-icon"><i class="fas fa-lock"></i></span>
          <input id="password" class="form-control @error('password') is-invalid @enderror" type="password" name="password" required autocomplete="current-password" placeholder="Password">
        </div>
        @error('password')
          <small class="text-danger">{{ $message }}</small>
        @enderror
      </div>

```

```

      <div class="form-group mt-4">
        <div class="form-check">
          <input class="form-check-input" type="checkbox" name="remember" id="remember" {{ old(key: 'remember') ? 'checked' : '' }}>
          <label class="form-check-label" for="remember">
            Remember me
          </label>
        </div>
      </div>

      <div class="d-flex justify-content-between mt-4">
        @if (Route::has(name: 'password.request'))
          <a class="text-decoration-none underline-link" href="{ { route(name: 'password.request') } }">
            Forgot your password?
          </a>
        @endif

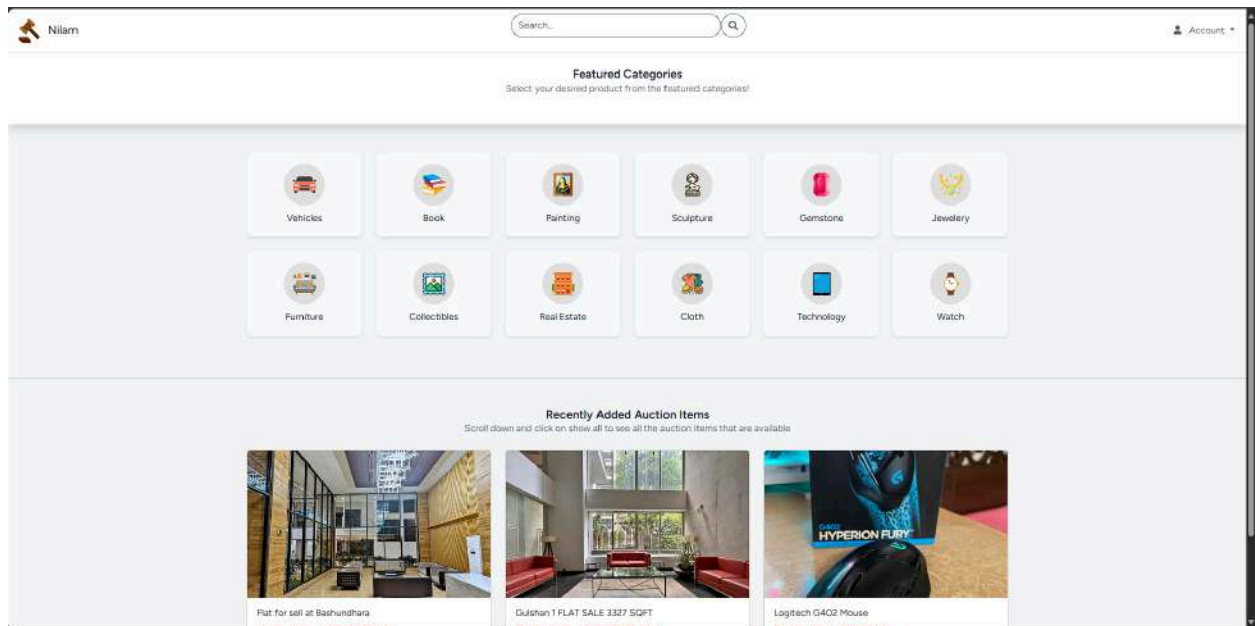
        <button type="submit" class="btn btn-primary px-4">Login</button>
      </div>
    </form>

    <div class="text-center mt-4">
      <p>Don't have an account? <a href="{ { route(name: 'register') } }" class="text-decoration-none underline-link">Register</a></p>
    </div>
  </div>
</div>

<!-- Include FontAwesome for Icons -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/js/all.min.js"></script>
</x-guest-layout>

```

## Home Page: This is the Home Page view



Following are the code for home page

```

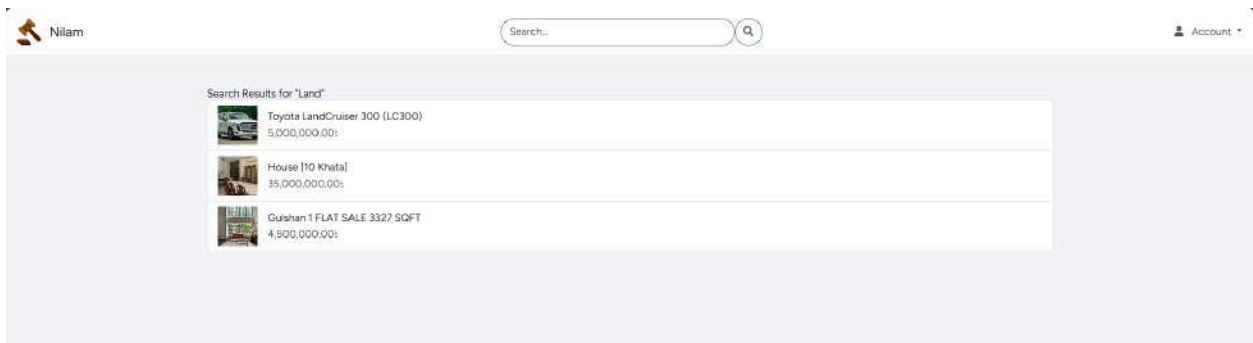
<x-app-layout>
  <x-slot name="header">
    <h2 class="font-semibold text-xl text-gray-800 leading-tight text-center">
      {{ __key__: 'Featured Categories' }}
    </h2>
    <p class="text-center text-gray-500 mb-4">Select your desired product from the featured categories!</p>
  </x-slot>

  <div class="main-content">
    <div class="container my-5">
      <div class="row">
        <!-- Loop through categories and display each category in a box -->
        @foreach($categories as $category)
          <div class="col-md-2 text-center mb-4">
            <a href="{{ route(name: 'categories.show', parameters: ['category' => $category->name]) }}" class="text-decoration-none">
              <div class="featured-category">
                <div class="icon-circle">
                  name }}" />
                </div>
                <p{{ $category->name }}</p>
              </div>
            </a>
          </div>
        @endforeach
      </div>
    </div>
    <div class="container my-5">
      <h2 class="font-semibold text-xl text-gray-800 leading-tight text-center">
        {{ __key__: 'Recently Added Auction Items' }}
      </h2>
      <p class="text-center text-gray-500 mb-4">Scroll down and click on show all to see all the auction items that are available!</p>
      <div class="row">
        @foreach ($auctionItems as $item)
          <div class="col-md-4">
            <a href="{{ url(path: '/item_details/' . $item->id) }}" class="text-decoration-none">
              <div class="card h-100"> <!-- Ensure uniform height for the card -->
                @if($item->images->isNotEmpty())
                  title }}" />
                @else
                  
                @endif
                <div class="card-body d-flex flex-column">
                  <h5 class="card-title">{{ $item->title }}</h5>
                  <p class="card-text">Starting Price: {{ number_format($item->starting_price, decimals: 2) }} taka</p>
                  <p class="mt-auto">
                    @foreach($item->categories as $category)
                      <span class="badge bg-primary">{{ $category->name }}</span>
                    @endforeach
                  </p>
                </div>
              </div>
            </a>
          </div>
        @endforeach

        <div class="col-md-25 text-center mt-4">
          <a href="{{ route(name: 'all_items') }}" class="btn btn-primary" style="right: 20px; bottom: 20px; background-color: blue; color: white;">
            Show All
          </a>
        </div>
      </div>
    </div>
  </x-app-layout>

```

**Search Functionality:** This is the view for the search functionality



Following is the code for search functionality



```

@extends('layouts.app2')

@section('section','content')


## Search Results for "{{ $query }}"</h2> @if($items->isEmpty()) @foreach($items as $item) <a href="{{ url(path:' /item_details/' . $item->id) }}" class="list-group-item list-group-item-action d-flex align-items-center"> @if($item->images->isEmpty())  @else  @endif <{{ $item->title }}</h5> <{{ number_format(num: $item->starting_price, decimals: 2) }}</p> </div> </a> @endif </div> @else Show All Items Functionality: This is the view to show all the available auction items 17


```

Following are the code for show all items functionality

```
@extends('layouts.app0')

@section('section','content')


## {{ __key: 'All Available Auction Items' }}



@foreach($auctionItems as $item)


@if\(\$item->images->isNotEmpty\(\)\)
@php
\$imagePath = \$item->images->first\(\)->url;
\$fullImagePath = asset\(path: '/storage/' . \$imagePath\);
@endphp
!\[{{ \$item->title }}\]\({{ \$fullImagePath }}\)
@else
!\[Placeholder\]\(https://via.placeholder.com/400x200\)
@endif


{{ \$item->title }}



Starting Price: {{ number\_format\(num: \$item->starting\_price, decimals: 2\) }} Taka



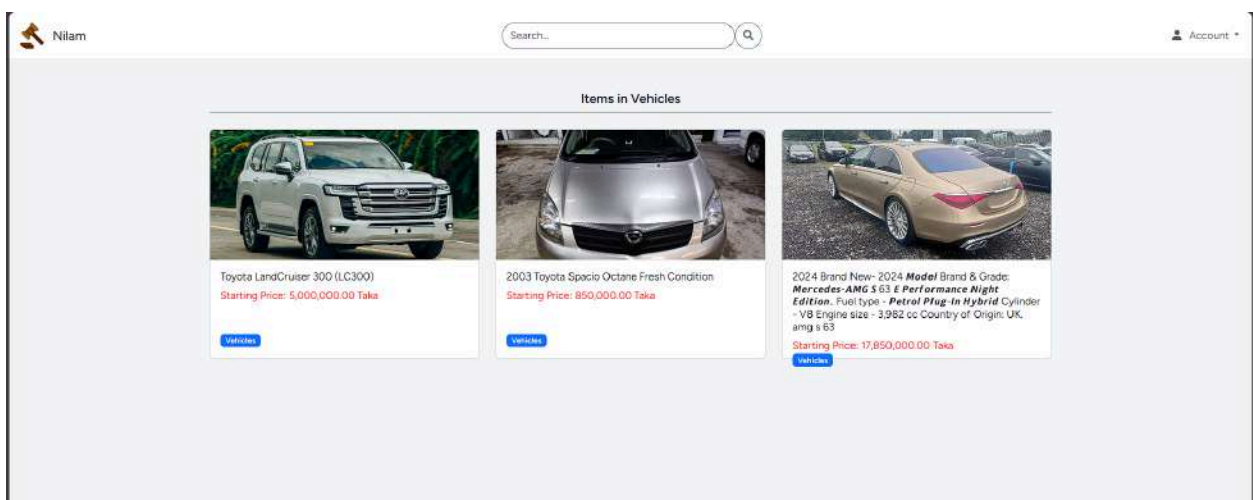
@foreach\(\$item->categories as \$category\)
{{ \$category->name }}
@endforeach



{{ $auctionItem->links() }}


```

**Category-based Filtering Functionality:** This is the view to show category-based filtering



Following are the codes for category-based filtering view

```

@extends(view: 'layouts.app2')

@section('section', 'content')


<!-- Category Title -->
    <h2 class="font-roboto text-xl text-gray-800 leading-tight text-center">Items in
    {{ __(key: $category->name) }}
    </h2>

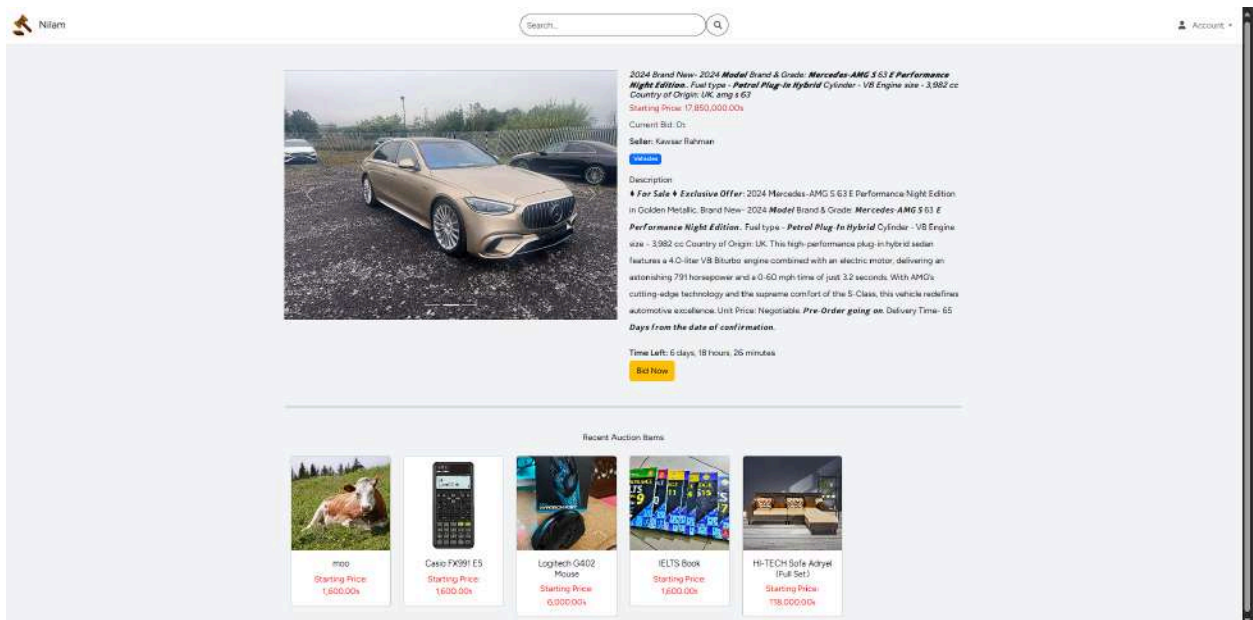
    <!-- Horizontal Line -->
    <div class="d-flex justify-content-center mt-2 mb-4">
        <hr style="width: 100%; border-top: 2px solid #000;">
    </div>

    <div class="row">
        @foreach ($auctionItems as $item)
            <div class="col-md-4 mb-4">
                <a href="{{ url(path: '/item_details/' . $item->id) }}" class="text-decoration-none">
                    <div class="card h-100">
                        @if($item->images->isEmpty())
                            @php
                                $imagePath = $item->images->first()->url;
                                $fullImagePath = asset(path: 'storage/' . $imagePath);
                            @endphp
                            title }}" style="height: 200px; object-fit: cover;">
                        @else
                            
                        @endif
                        <div class="card-body d-flex flex-column" style="height: 150px;">
                            <h5 class="card-title">{{ $item->title }}</h5>
                            <p class="card-text">Starting Price: {{ number_format($item->starting_price, decimals: 2) }} Taka</p>
                            <div class="mt-auto">
                                <div class="d-flex flex-wrap">
                                    @foreach($item->categories as $category)
                                        <span class="badge bg-primary me-1">{{ $category->name }}</span>
                                    @endforeach
                                </div>
                            </div>
                        </div>
                    </div>
                </a>
            </div>
        @endforeach
    </div>


    @if($auctionItems->isEmpty())
        <div class="text-center mt-5">
            <p>No Items found under this category.</p>
        </div>
    @endif
</div>


```

**Show Item Details Functionality:** This is the view for the selected item



This is the view of the auction item details if the item auction end duration has passed and the current logged in user is not the highest bidder



moo

Starting Price: 1,600.00₹

Current Bid: 2,500₹

Seller: Sopnil Roy Niloy

[Bidding](#)

Description


cow

Time Left: 0 days, 0 hours, 0 minutes (Auction Ended)

This auction has been won by someone else.


---

Recent Auction Items




moo

Starting Price: 1,600.00₹




Casio FX991 ES

Starting Price: 1,600.00₹




Logitech G402 Mouse

Starting Price: 6,000.00₹



IELTS Book

Starting Price: 1,600.00₹



HI-TECH Sofa Adryel (Full Set)

Starting Price: 118,000.00₹

The following is the view of the auction item details if the item auction end duration has passed and the current logged in user is the highest bidder





```

        <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="prev">
          <span class="carousel-control-prev-icon" aria-hidden="true"></span>
          <span class="visually-hidden">Previous</span>
        </button>
        <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators" data-bs-slide="next">
          <span class="carousel-control-next-icon" aria-hidden="true"></span>
          <span class="visually-hidden">Next</span>
        </button>
      </div>
    </div>

    <!-- Right Section: Product Details -->
    <div class="col-md-6" style="line-height: 2;>
      <h2 style="font-weight: bold; font-style: italic;>{{ $auctionItem->title }}</h2>
      <div class="d-flex align-items-center">
        <div class="me-3 text-danger" style="line-height: 2;>Starting Price: {{ number_format(num: $auctionItem->starting_price, decimals: 2) }}$</div>
      </div>
      <div class="d-flex align-items-center">
        <div class="me-3 text-muted" style="line-height: 2;>Current Bid: {{ number_format(num: $auctionItem->current_bid ?? $auctionItem->current_bid ) }}$</div>
      </div>

      <!-- Seller Info -->
      <p><strong>Seller:</strong> {{ $auctionItem->seller->name }}</p>

      <!-- Categories -->
      <div class="mb-3">
        @foreach($auctionItem->categories as $category)
          <span class="badge bg-primary">{{ $category->name }}</span>
        @endforeach
      </div>
    </div>
  </div>

```

```

    </div>

    <!-- Description -->
    <div class="mb-3" style="line-height: 2;>
      <h3>Description</h3>
      <p>{{ $auctionItem->description }}</p>
    </div>

    <!-- Time Left -->
    <p><strong>Time Left:</strong>
      @if($auctionItem->isAuctionEnded())
        0 days, 0 hours, 0 minutes (Auction Ended)
      @else
        {{ $timeleft->days }} days, {{ $timeleft->hours }} hours, {{ $timeleft->minutes }} minutes
      @endif
    </p>

    <!-- Buy Now / Bid Now -->
    <div class="d-flex">
      @if($auctionItem->isAuctionEnded())
        @if($highestBidder)
          <button class="btn btn-primary me-2" data-bs-toggle="modal" data-bs-target="#buyNowModal">Buy Now</button>
        @else
          <div class="text-danger">This auction has been won by someone else.</div>
        @endif
      @else
        <button class="btn btn-warning" data-bs-toggle="modal" data-bs-target="#bidModal">Bid Now</button>
      @endif
    </div>
  </div>
</div>

```

```

<!-- Modal for placing a bid -->
<div class="modal fade" id="bidModal" tabindex="-1" aria-labelledby="bidModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <form action="{{ route('name', 'placeBid', parameters: $auctionItem->id) }}" method="POST">
        @csrf
        <div class="modal-header">
          <h3 class="modal-title" id="bidModalLabel">Place Your Bid </h3>
          <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
        </div>
        <div class="modal-body">
          <div class="form-group">
            <label for="auctionItem">Auction Item </label>
            <input type="text" class="form-control" value="{{ $auctionItem->title }}" disabled>
          </div>
          <div class="form-group mt-1">
            <label for="bidAmount">Bid Amount </label>
            <input type="number" class="form-control" id="bidAmount" min="0" bid_amount required placeholder="Enter your bid amount" />
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Close</button>
            <button type="submit" class="btn btn-primary">Place Bid</button>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

<div class="modal fade" id="buyNowModal" tabindex="-1" aria-labelledby="buyNowModalLabel" aria-hidden="true">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h3 class="modal-title" id="buyNowModalLabel">Buy Now</h3>
        <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
      </div>
      <div class="modal-body">
        <form action="{{ route('name', 'buyNow' ) }}" method="POST">
          @csrf

```

```

<form action="{ route(name: 'buyNow') }}" method="POST">
  @csrf
  <div class="modal-body">
    <input type="hidden" name="auction_item_id" value="{ {{ $auctionItem->id }}" />

    <!-- User Details -->
    <div class="mb-3">
      <label for="name" class="form-label">Name</label>
      <input type="text" class="form-control" id="name" value="{ {{ Auth::user()->name }}" disabled="
    </div>

    <div class="mb-3">
      <label for="email" class="form-label">Email</label>
      <input type="email" class="form-control" id="email" value="{ {{ Auth::user()->email }}" disabled="
    </div>

    <div class="mb-3">
      <label for="address" class="form-label">Address</label>
      <input type="text" class="form-control" id="address" value="{ {{ Auth::user()->address }}" disabled="
    </div>

    <!-- Current Bid -->
    <div class="mb-3">
      <label for="current_bid" class="form-label">Current Bid</label>
      <input type="text" class="form-control" id="current_bid" value="{ {{ number_format($auctionItem->current_bid, decimals: 2) }}" disabled="
    </div>

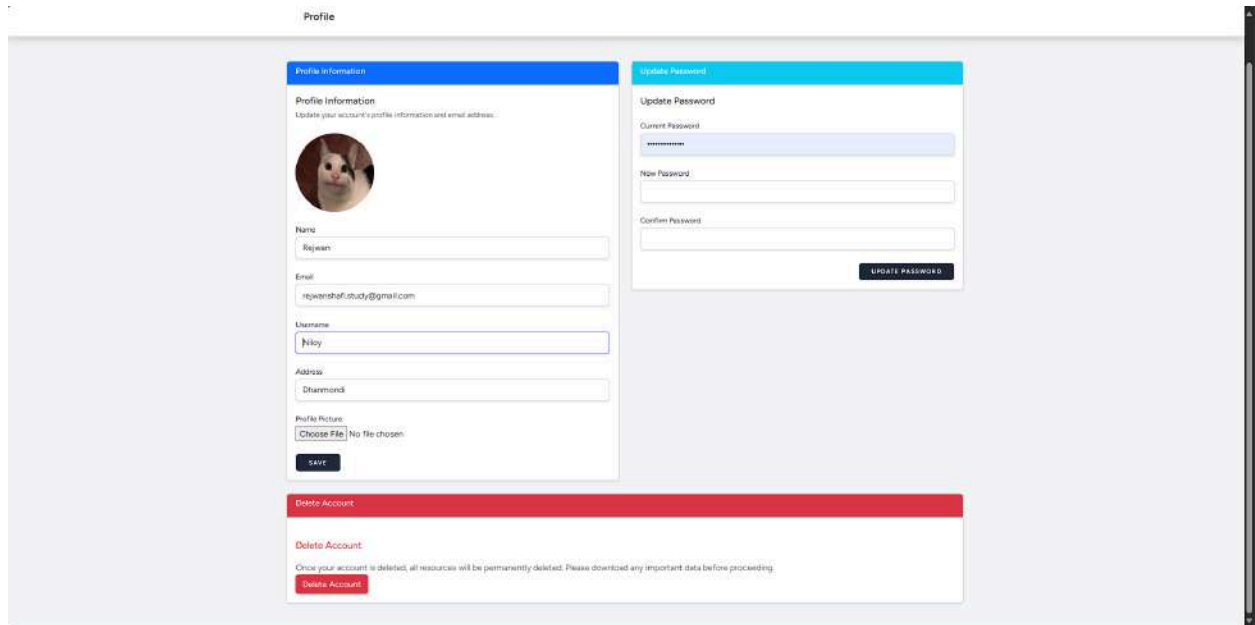
    <!-- Payment Method -->
    <div class="mb-3">
      <label for="payment_method" class="form-label">Payment Method</label>
      <select name="payment_method" class="form-select" id="payment_method" required="
        <!-- option value="bank">Bank</option> -->
        <option value="cash-on-delivery">Cash on Delivery</option>
      </select>
    </div>

    <div class="modal-footer">
      <button type="button" class="btn btn-secondary" data-bs-dismiss="modal">Cancel</button>
      <button type="submit" class="btn btn-primary">Confirm Purchase</button>
    </div>
  </form>
</div>
</div>
<!-- Styling horizontal line -->
<hr class="my-4 style="border-top: 3px solid #ccc;"/>

<!-- Recent Auction Items -->
<div class="container mt-5">
  <div class="text-center mb-4">Recent Auction Items</div>
  <div class="row">
    @foreach($recentItems as $item)
      <div class="col-md-4">
        <a href="{ {{ url($item->item_details) . $item->id }}" class="text-decoration: none">
          <div class="card">
            @if($item->image->isNotEmpty())
              title }}" style="height: 100px; object-fit: cover;"/>
            @else
              
            </div>
            <div class="card-body text-center">
              <div class="card-title">{{ $item->title }}</div>
              <p class="card-text" style="color: red;>Starting Price: {{ number_format($item->starting_price, decimals: 2) }}$</p>
            </div>
          </div>
        </a>
      </div>
    @endforeach
  </div>
</div>
</div>
</section>

```

**Update Profile Functionality:** This is the view for update profile



Following is the code for the profile update functionality

```

1  <x-app-layout>
2    <x-slot name="header">
3      <h2 class="font-semibold text-xl text-gray-800 leading-tight">
4        {{ __ (key: 'Profile') }}
5      </h2>
6    </x-slot>
7
8    <div class="py-12">
9      <div class="container mx-auto">
10
11        <div class="row g-4">
12          <!-- Profile Information Section -->
13          <div class="col-lg-6 col-md-12">
14            <div class="card shadow-sm">
15              <div class="card-header bg-primary text-white">
16                <h3 class="card-title">{{ __ (key: 'Profile Information') }}</h3>
17              </div>
18              <div class="card-body">
19                @include(view: 'profile.partials.update-profile-information-form')
20              </div>
21            </div>
22          </div>
23
24          <!-- Update Password Section -->
25          <div class="col-lg-6 col-md-12">
26            <div class="card shadow-sm">
27              <div class="card-header bg-info text-white">
28                <h3 class="card-title">{{ __ (key: 'Update Password') }}</h3>
29              </div>
30              <div class="card-body">
31                @include(view: 'profile.partials.update-password-form')
32              </div>
33            </div>
34          </div>
35        </div>

```



```

    </div>

    <!-- Delete Account Section -->
    <div class="col-12">
      <div class="card shadow-sm">
        <div class="card-header bg-danger text-white">
          <h3 class="card-title">{{ __ (key: 'Delete Account') }}</h3>
        </div>
        <div class="card-body">
          @include(view: 'profile.partials.delete-user-form')
        </div>
      </div>
    </div>
  </div>
</div>
</x-app-layout>

```

```

<section>
  <header>
    <h2 class="text-lg font-medium text-gray-900 mb-3">
      {{ __ (key: 'Update Password') }}
    </h2>
  </header>

  <form method="POST" action="{{ route(name: 'user.updatePassword') }}">
    @csrf
    <!-- Current Password -->
    <div class="mt-4">
      <x-input-label for="current_password" :value="__ (key: 'Current Password') " />
      <x-text-input id="current_password" class="block mt-1 w-full" type="password" name="current_password" required />
      <x-input-error :messages="$errors->get('current_password')" class="mt-2" />
    </div>

    <!-- New Password -->
    <div class="mt-4">
      <x-input-label for="password" :value="__ (key: 'New Password') " />
      <x-text-input id="password" class="block mt-1 w-full" type="password" name="password" required />
      <x-input-error :messages="$errors->get('password')" class="mt-2" />
    </div>

    <!-- Confirm New Password -->
    <div class="mt-4">
      <x-input-label for="password_confirmation" :value="__ (key: 'Confirm Password') " />
      <x-text-input id="password_confirmation" class="block mt-1 w-full" type="password" name="password_confirmation" required />
      <x-input-error :messages="$errors->get('password_confirmation')" class="mt-2" />
    </div>

    <div class="flex items-center justify-end mt-4">
      <x-primary-button class="ml-4">
        {{ __ (key: 'Update Password') }}
      </x-primary-button>
    </div>
  </form>
</section>


```

```

1 <section>
2 <header>
3   <h2 class="text-lg font-medium text-gray-600">
4     {{ __key__ 'Profile Information' }}
5   </h2>
6   <p class="mt-1 text-sm text-gray-600">
7     {{ __key__ 'Update your account's profile information and email address.' }}
8   </p>
9 </header>
10
11 <form method="post" action="{{ route(name) 'profile.update' }}" enctype="multipart/form-data" class="mt-6 space-y-6">
12   @csrf
13   @method('patch')
14
15   <!-- Avatar Section -->
16   <div class="text-center mb-4">
17     @if (auth()->user()->avatar)
18       
19     @else
20       
21     @endif
22   </div>
23
24   <!-- Disabled Name Field -->
25   <div>
26     <input-label for="name" value="{{ __key__ 'Name' }}" />
27     <input id="name" name="name" type="text" class="mt-1 block w-full" value="old(key) 'name', default({user->name})" disabled />
28     <input-error class="mt-2" messages="errors" get('name') />
29   </div>
30
31   <!-- Disabled Email Field -->
32   <div>
33     <input-label for="email" value="{{ __key__ 'Email' }}" />
34     <input id="email" name="email" type="email" class="mt-1 block w-full" value="old(key) 'email', default({user->email})" disabled />
35     <input-error class="mt-2" messages="errors" get('email') />
36   </div>
37
38   <!-- Username Field -->
39   <div>
40     <input-label for="username" value="{{ __key__ 'Username' }}" />
41     <input id="username" name="username" type="text" class="mt-1 block w-full" value="old(key) 'username', default({user->username})" required autofocus autocomplete="username" />
42     <input-error class="mt-2" messages="errors" get('username') />
43   </div>
44
45   <!-- Address Field -->
46   <div>
47     <input-label for="address" value="{{ __key__ 'Address' }}" />
48     <input id="address" name="address" type="text" class="mt-1 block w-full" value="old(key) 'address', default({user->address})" />
49     <input-error class="mt-2" messages="errors" get('address') />
50   </div>
51
52   <!-- Avatar Upload Field -->
53   <div>
54     <input-label for="avatar" value="{{ __key__ 'Profile Picture' }}" />
55     <input id="avatar" type="file" name="avatar" class="mt-1 block w-full" />
56     <input-error class="mt-2" messages="errors" get('avatar') />
57   </div>
58
59   <!-- Submit Button -->
60   <div class="flex items-center gap-4">
61     <x-primary-button {{ __key__ 'Save' }} /></x-primary-button>
62
63     @if (session(key) 'status') === 'profile-updated'
64       <div>
65         <x-data="{ show: true }">
66           <x-show="show">
67             <x-transition>
68               <x-init="setTimeout(() => show = false, 2000)" />
69               <div class="text-sm text-gray-600">
70                 {{ __key__ 'Saved.' }}
71               </div>
72             </x-transition>
73           </x-show>
74         </x-data>
75       </div>
76     @endif
77   </div>
78 </form>
79 </section>

```

**Transaction History Functionality:** In this view, the user can see the auction items that they have won

 Nilam
 

Account

Your Transactions				
Auction Item Title	Seller Name	Starting Price	Current Price	Payment Status
moo	Sopnil Roy Niloy	1,600.00t	2,500.00t	<span>Completed</span>

```

1  @extends('layouts.app')
2
3  @section('content')
4  <div class="container mt-5">
5      <h2>Your Transactions</h2>
6
7      @if($wonItems->isEmpty())
8          <p class="text-center">You haven't won any auctions yet.</p>
9      @else
10         <table class="table table-hover table-striped table-bordered" style="border-radius: 15px;">
11             <thead class="table-dark">
12                 <tr>
13                     <th>Auction Item Title</th>
14                     <th>Seller Name</th>
15                     <th>Starting Price</th>
16                     <th>Current Price</th>
17                     <th>Payment Status</th>
18                 </tr>
19             </thead>
20             <tbody>
21                 @foreach($wonItems as $item)
22                     <tr>
23                         <td>{{ $item->title }}</td>
24                         <td>{{ $item->seller->name }}</td>
25                         <td>{{ number_format($item->starting_price, decimals: 2) }}$</td>
26                         <td>{{ number_format($item->current_bid, decimals: 2) }}$</td>
27                         <td>
28                             @if($item->payment_status === 'paid')
29                                 <span class="badge bg-success">Paid</span>
30                             @else
31                                 <span class="badge bg-warning">Pending</span>
32                             @endif
33                         </td>
34                     </tr>
35                 @endforeach

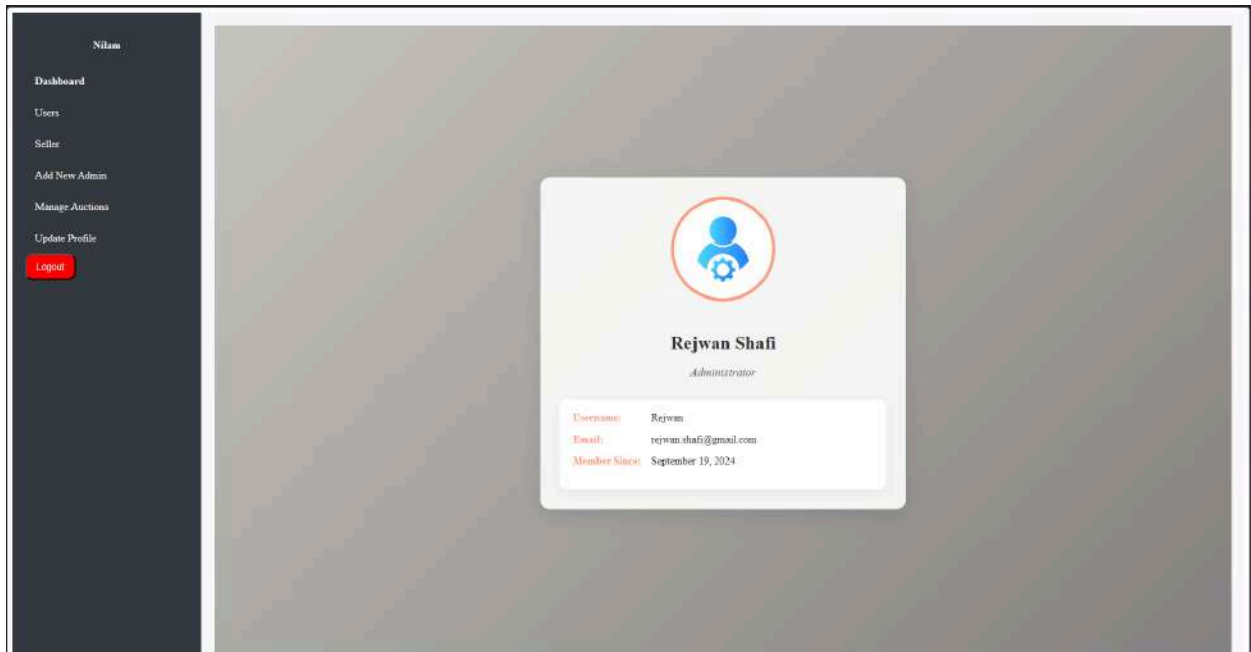
```

```

35         @endforeach
36     </tbody>
37 </table>
38 @endif
39 </div>
40 @endsection
41

```

**Admin Dashboard Functionality:** This is the view of Admin Dashboard



Here is the code for the admin dashboard

```
@extends('layouts.admin')

@section('title', 'Admin Profile')

5 {

    @section('content')
        <div class="profile-container">
            <div class="profile-card">
                <div class="profile-header">
                    
                    <h2 class="profile-name">{{ $admin->name }}</h2>
                    <p class="profile-role">Administrator</p>
                </div>
                <div class="profile-info">
                    <div class="info-item">
                        <span class="info-label">Username:</span>
                        <span class="info-value">{{ $admin->username }}</span>
                    </div>
                    <div class="info-item">
                        <span class="info-label">Email:</span>
                        <span class="info-value">{{ $admin->email }}</span>
                    </div>
                    <div class="info-item">
                        <span class="info-label">Member Since:</span>
                        <span class="info-value">{{ $admin->created_at->format('F d, Y') }}</span>
                    </div>
                </div>
            </div>
        </div>
    @endsection
}
```

**User Management Facility for admin:** In this view the admin can remove any logged in users from the database

Nizam

Dashboard

Users

Seller

Add New Admin

Manage Actions

Update Profile

Logout

All Users

ID	Name	Email	Action
2	Rajwan	rajwanshafi.stady@gmail.com	Delete
4	Homarah Ferdousia	homara@gmail.com	Delete
5	Afif Rayhan Pranto	pranto@gmail.com	Delete

Here is the code for the view

```

@extends('layouts.admin')

@section('content')

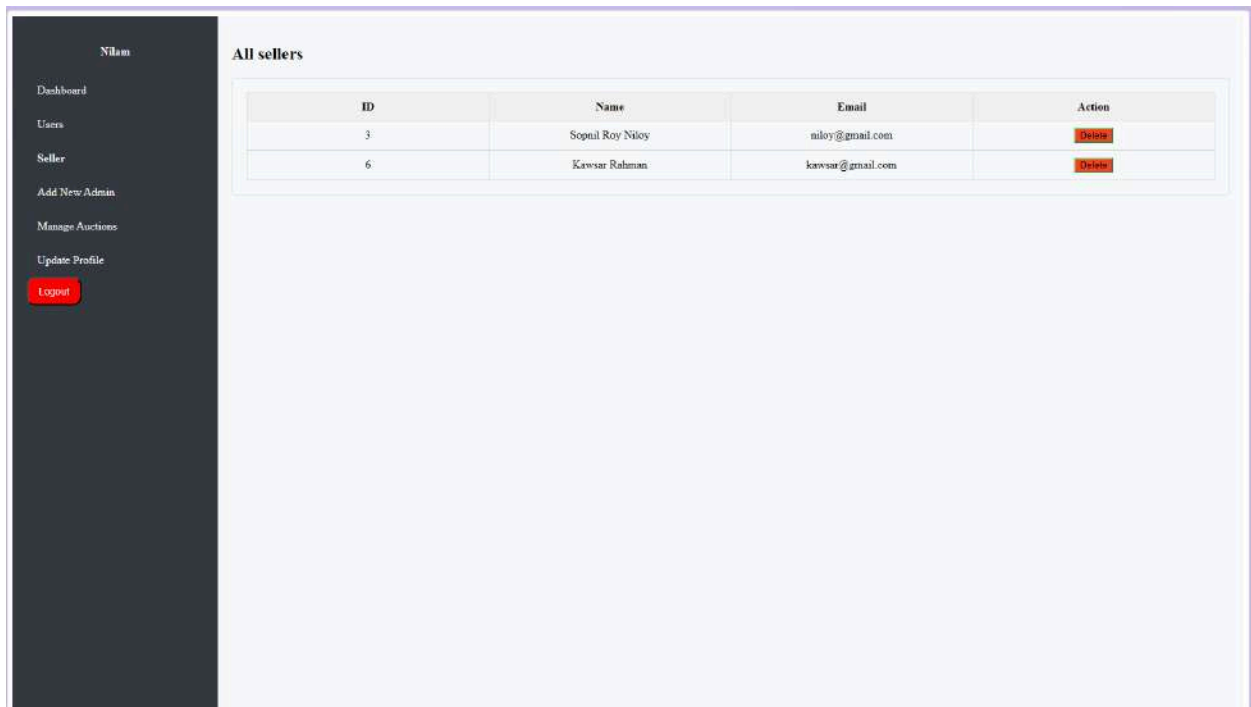

## All Users



| ID               | Name               | Email               | Action                      |
|------------------|--------------------|---------------------|-----------------------------|
| {{ \$user->id }} | {{ \$user->name }} | {{ \$user->email }} | <div> Delete Form -- </div> |


```

**Seller Management Facility for admin:** In this view the admin can remove any logged-in seller from the database



Here is the code for the view

```

extends('base', layout: 'main')
section(section: 'title', content: 'Seller Management')
section(section: 'content')


<h2>All Sellers</h2>
    <div class="table">
        <table class="table">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Name</th>
                    <th>Email</th>
                    <th>Action</th>
                </tr>
            </thead>
            <tbody>
                <div (Sellers.isEmpty)>
                    <div (Sellers as Seller)>
                        <tr>
                            <td ({{ Seller.id }})>
                            <td ({{ Seller.name }})>
                            <td ({{ Seller.email }})>
                            <td>
                                <form action="{{ route('admin.users.destroy', parameters: {seller: id}) }}" method="POST">
                                    <input type="button" value="Delete" />
                                </form>
                            </td>
                        </tr>
                    </div>
                </tbody>
            </table>
            <div (Sellers.isEmpty)>
                <div colspan="4">No users found.</div>
            </div>
        </div>
    </div>
</section>


```

**Auction Approval Facility:** This is the view for the auction approval facility for the admin

Nilam

Dashboard

Users

Seller

Add New Admin

Manage Auctions

Auction Approval

Auction Status

Biddings

Update Profile

Logout

Manage Auction Items

ID	Title	End Time	Current Bid	Seller Name	Actions
37	xxx	2024-09-26 03:16:00		Kavasar Rahman	<div>Delete</div> <div>Approve</div>

Here is the code for the view

```

1 @extends('layouts.admin')
2
3 @section('content')
4     <div class="container">
5         <h2 Manage Auction Items /h2>
6         <div class="row">
7             <table class="table">
8                 <thead>
9                     <tr>
10                         <th>ID</th>
11                         <th>Title</th>
12                         <th>End Time</th>
13                         <th>Current Bid</th>
14                         <th>Seller Name</th>
15                         <th>Actions</th>
16                     </tr>
17                 </thead>
18                 <tbody>
19                     @if ($auctionItems->isEmpty())
20                         @foreach ($auctionItems as $auctionItem)
21                             <tr>
22                                 <td>{{ $auctionItem->id }}</td>
23                                 <td>{{ $auctionItem->title }}</td>
24                                 <td>{{ $auctionItem->end_time }}</td>
25                                 <td>{{ $auctionItem->current_bid }}</td>
26                                 <td>{{ $auctionItem->seller_name }}</td>
27                                 <td>
28                                     <form action="{{ route('admin.auction.delete', $auctionItem->id) }}" method="POST" style="display:inline;">
29                                         @csrf
30                                         @method('DELETE')
31                                         <button type="submit" class="btn btn-danger" style="background-color: #fa5a5d; color: #000; border-color: #fa5a5d; onclick:"return confirm('Are you sure you want to delete this item?');">Delete</button>
32                                     </form>
33                                     <form action="{{ route('admin.auction.approve', $auctionItem->id) }}" method="POST" style="display:inline;">
34                                         @csrf
35                                         @method('PUT')
36                                         <button type="submit" class="btn btn-success" style="background-color: #28a745; color: #000; border-color: #28a745;">Approve</button>
37                                     </form>
38                                 </td>
39                             </tr>
40                         @endforeach
41                     @else
42                         <tr>
43                             <td colspan="6">No auction items found.</td>
44                         </tr>
45                     @endif
46                 </tbody>
47             </table>
48         </div>
49     </div>
50 @endsection

```



**Auction Status Facility:** This is the view where can see all the auctioned item's status

Nilam

Dashboard

Users

Seller

Add New Admin

Manage Auctions

Update Profile

Logout

All Auction Items

ID	Title	End Time	Current Bid	Seller Name	Status
19	Toyota LandCruiser 300 (LC300)	2024-09-21 16:30:00	5000001.00	Sopnil Roy Niley	Approved
20	House [10 Khata]	2024-09-21 15:40:00		Sopnil Roy Niley	Approved
21	Sea painting	2024-09-26 19:49:00		Sopnil Roy Niley	Approved
22	2003 Toyota Spacio Octane Fresh Condition	2024-09-26 19:49:00		Sopnil Roy Niley	Approved
24	IPad Air 5	2024-09-27 20:14:00		Kavwar Rahman	Approved
25	2024 Brand New- 2024 <b>Model</b> Brand & Grade: <b>Mercedes-AMG S 63 E Performance Night Edition</b> . Fuel type - <b>Petrol Plug-in Hybrid</b> Cylinder - V8 Engine size - 3.982 cc Country of Origin: UK. <b>amg s 63</b>	2024-09-27 20:19:00		Kavwar Rahman	Approved
27	Gulshan 1 FLAT SALE 3327 SQFT	2024-09-23 20:23:00		Kavwar Rahman	Approved
28	Flat for sell at Bashundhara	2024-09-27 20:31:00	1100003.00	Kavwar Rahman	Approved
29	HI-TECH Sofa Adryel (Full Set)	2024-09-25 20:31:00		Kavwar Rahman	Approved
31	IELTS Book	2024-09-27 23:03:00		Kavwar Rahman	Approved
32	Logitech G402 Mouse	2024-09-23 23:04:00		Kavwar Rahman	Approved
33	Casio FX991 ES	2024-09-20 23:08:00		Kavwar Rahman	Approved
35	moo	2024-09-20 23:45:00	2500.00	Sopnil Roy Niley	Sold
36	Ceramic Vase Pack 2, White Modern Bad Vase, Ceramic Modern Vase Decor, Sculpture Decor, Fire Place Decoration, Mid Century Modern Drip	2024-09-21 02:52:00	10020.00	Sopnil Roy Niley	Sold

Here is the code for the view

```

@permissions([roles: 'isAdmin'])
@section(section: 'content')


<h3>All Auction Items</h3>


| ID | Title | End Time | Current Bid | Seller Name | Status |
| --- | --- | --- | --- | --- | --- |

@if (auctionItems.isEmpty())
@for (auctionItem as AuctionItem)
| {{ auctionItem.id }} | {{ auctionItem.title }} | {{ auctionItem.end_time }} | {{ auctionItem.current_bid }} | {{ auctionItem.seller_name }} | @if (auctionItem.status == 1) button class="btn" disabled style="background-color: #FFD700; color: #000; border-color: #FFD700;" Waiting /button @elseif (auctionItem.status == 2) button class="btn" disabled style="background-color: #4CAF50; color: #000; border-color: #4CAF50;" Approved /button @elseif (auctionItem.status == 3) button class="btn" disabled style="background-color: #F44336; color: #000; border-color: #F44336;" Sold /button @else button class="btn btn-secondary" disabled Unknown Status /button @endif |


```

No auction items found.					

**Bidding Status Facility:** In this view, Admin can see all the bidding history of the available items that are not sold yet

Auction Bids					
Item ID	Item Title	Starting Price	Current Price	Bidder Name	Seller Name
19	Toyota LandCruiser 300 (LC300)	5,000,000.00₹	5,000,001.00₹	Adif Rayhan Pranto	Soumi Roy Nidoy
28	Flat for sell at Bashundhara	1,100,000.00₹	1,100,003.00₹	Adif Rayhan Pranto	Kausar Rahman

Here is the code for the view

```

1  @extends('layouts.admin')
2
3  @section('section:content')
4  <div class="container">
5      <h2>Auction Bids</h2>
6      <div class="box">
7          <table class="table">
8              <thead>
9                  <tr>
10                     <th>Item ID</th>
11                     <th>Item Title</th>
12                     <th>Starting Price</th>
13                     <th>Current Price</th>
14                     <th>Bidder Name</th>
15                     <th>Seller Name</th>
16                 </tr>
17             </thead>
18             <tbody>
19                 @if ($items->isEmpty())
20                     @foreach ($items as $item)
21                         <tr>
22                             <td>{{ $item['item_id'] }}</td>
23                             <td>{{ $item['item_title'] }}</td>
24                             <td>{{ number_format(num: $item['starting_price'], decimals: 2) }}₹</td>
25                             <td>{{ number_format(num: $item['current_price'], decimals: 2) }}₹</td>
26                             <td>{{ $item['bidder_name'] }}</td>
27                             <td>{{ $item['seller_name'] }}</td>
28                         </tr>
29                     @endforeach
30                 @else
31                     <tr>
32                         <td colspan="6">No bids found for the auction items.</td>
33                     </tr>
34                 @endif
35             </tbody>
36         </table>
37     </div>
38 </div>
39 @endsection
40

```

## Payment Approval Functionality: This is the view for the payment approval page

Nilam

Dashboard

Users

Seller

Add New Admin

Manage Auctions

Update Profile

Logout

Pending Payments

Payment ID	Auction Item	User	Amount	Payment Method	Status	Action
4	Casio MTF-814SD-9A/DF Quartz Analog Dial Stainless Steel Unisex	Alif Rayhan Pranto	50000.00	Cash on delivery	Pending	<div>ApproveReject</div>

Here is the code

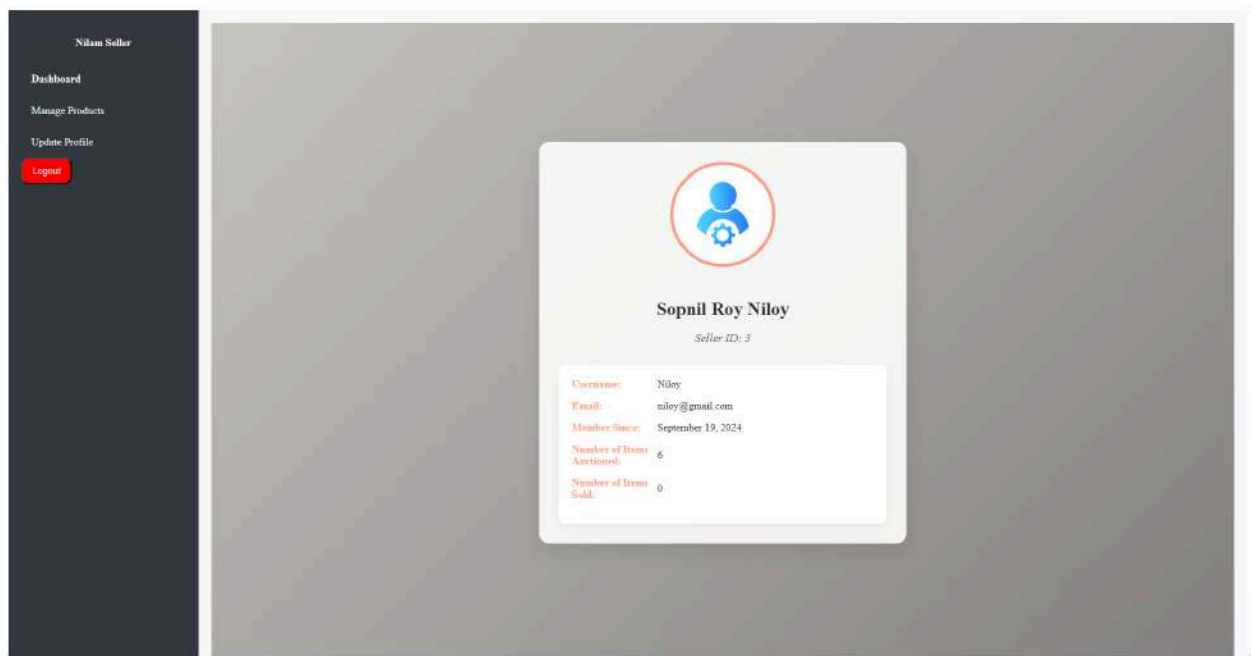
```
@extends('layouts.admin')

@section('section','content')


<h2>Pending Payments</h2>
    @if (session(key: 'success'))
        <div class="alert alert-success">
            {{ session(key: 'success') }}
        </div>
    @endif
    <table class="table table-bordered">
        <thead>
            <tr>
                <th>Payment ID</th>
                <th>Auction Item</th>
                <th>User</th>
                <th>Amount</th>
                <th>Payment Method</th>
                <th>Status</th>
                <th>Action</th>
            </tr>
        </thead>
        <tbody>
            @foreach($payments as $payment)
                <tr>
                    <td>{{ $payment->id }}</td>
                    <td>{{ $payment->auctionItem->title }}</td>
                    <td>{{ $payment->user->name }}</td>
                    <td>{{ $payment->amount }}</td>
                    <td>{{ ucfirst(string: $payment->payment_method) }}</td>
                    <td>{{ ucfirst(string: $payment->status) }}</td>
                    <td>
                        <!-- Approve Button -->
                        <a href="{{ route(name: 'admin.payment-approve', parameters: $payment->id) }}" class="btn btn-success btn-sm">
                            Approve
                        </a>
                        <!-- Reject Button -->
                        <a href="{{ route(name: 'admin.payment-reject', parameters: $payment->id) }}" class="btn btn-danger btn-sm">
                            Reject
                        </a>
                    </td>
                </tr>
            @endforeach
        </tbody>
    </table>
</div>
@endsection


```

**Seller Dashboard Functionality:** Here is the view of the seller dashboard



Here is the code for the view

```
@extends('layouts.seller')

@section('section', 'title', content: 'Seller Profile')

5 {

    @section('section', 'content')
    <div class="profile-container">
        <div class="profile-card">
            <div class="profile-header">
                
                <h2 class="profile-name">{{ $seller->Name }}</h2>
                <p class="profile-role">Seller ID: {{ $seller->id }}</p>
            </div>
            <div class="profile-info">
                <div class="info-item">
                    <span class="info-label">Username:</span>
                    <span class="info-value">{{ $seller->username }}</span>
                </div>
                <div class="info-item">
                    <span class="info-label">Email:</span>
                    <span class="info-value">{{ $seller->email }}</span>
                </div>
                <div class="info-item">
                    <span class="info-label">Member Since:</span>
                    <span class="info-value">{{ $seller->created_at->format("F d, Y") }}</span>
                </div>
                <div class="info-item">
                    <span class="info-label">Number of Items Auctioned:</span>
                    <span class="info-value">{{ $auctionedItemsCount }}</span>
                </div>
                <div class="info-item">
                    <span class="info-label">Number of Items Sold:</span>
                    <span class="info-value">{{ $soldItemsCount }}</span>
                </div>
            </div>
        </div>
    </div>
    @endsection
}
```

**Add new Auction Item Functionality:** Here is the view while an seller wants to create an item for auction

Nilam Seller

Dashboard

Manage Products

Update Profile

Logout

Add New Auction Item

Title:

Item Title

Description:

Description

Starting Price:

Starting Price

Current Bid (Optional):

Current Bid

Auction End Date:

dd/mm/yyyy --:-- --

Select Category:

Toggle Categories

Upload Images:

Choose Files

No file chosen

You can upload up to 3 images.

Add Item

Here is the code for the view

```

1 @extends('views.layouts.seller')
2
3 @section(section: 'content')
4 <div class="form-container">
5   <div class="form-box">
6     <!-- Admin icon at the top -->
7     <div class="text-center mb-4">
8       
9       <h4 class="mt-2">Add New Auction Item</h4>
10    </div>
11
12    <form method="POST" action="{{ route('names.seller.storeAuctionItem') }}" enctype="multipart/form-data" id="auctionForm">
13      @csrf
14
15      <!-- Item Title -->
16      <div class="form-group mb-3">
17        <label for="title" class="form-label">Title:</label>
18        <input type="text" class="form-control" id="title" name="title" placeholder="Item title" required>
19      </div>
20
21      <!-- Description -->
22      <div class="form-group mb-3">
23        <label for="description" class="form-label">Description:</label>
24        <input type="text" class="form-control" id="description" name="description" placeholder="Description" required>
25      </div>
26
27      <!-- Starting Price -->
28      <div class="form-group mb-3">
29        <label for="starting_price" class="form-label">Starting Price:</label>
30        <input type="number" class="form-control" id="starting_price" name="starting_price" placeholder="Starting Price" required step="0.01">
31      </div>
32
33      <!-- Current Bid (Optional) -->
34      <div class="form-group mb-3">
35        <label for="current_bid" class="form-label">Current Bid (Optional):</label>
36        <input type="number" class="form-control" id="current_bid" name="current_bid" placeholder="Current Bid" step="0.01">
37      </div>
38
39      <!-- End Time -->
40      <div class="form-group mb-3">
41        <label for="end_time" class="form-label">Auction End Date:</label>
42        <input type="datetime-local" class="form-control" id="end_time" name="end_time" required>
43      </div>
44
45      <!-- Categories -->
46      <div class="form-group mb-3">
47        <label class="form-label">Select Category:</label>
48      </div>
49      <!-- Toggle button -->

```

```

105    <!-- Toggle button -->
106    <div class="btn btn-outline-primary w-100" id="category-toggle">Toggle Categories</div>
107
108    <!-- Dropdown (Hidden Initially) -->
109    <div class="category-dropdown mt-2">
110      <select class="form-control" id="categories" name="categories[]" multiple>
111        @foreach($categories as $category)
112          <option value="{{ $category->id }}">{{ $category->name }}</option>
113        @endforeach
114      </select>
115    </div>
116  </div>
117
118  <!-- Image Uploads -->
119  <div class="form-group mb-3">
120    <label for="images" class="form-label">Upload Images:</label>
121    <input type="file" class="form-control" id="images" name="images[]" multiple accept="image/*">
122    <small class="form-text text-muted">You can upload up to 3 images.</small>
123  </div>
124
125
126  <!-- Submit Button -->
127  <div class="form-group mb-3">
128    <button type="submit" class="btn btn-primary">Add Item</button>
129  </div>
130 </form>
131 </div>
132 </div>
133 @endsection
134
135 @section(section: 'scripts')
136 <script>
137   // Get the elements
138   const toggleButton = document.getElementById('category-toggle');
139   const dropdown = document.querySelector('.category-dropdown');
140
141   // Toggle the dropdown when the button is clicked
142   toggleButton.addEventListener('click', function() {
143     dropdown.style.display = dropdown.style.display === 'block' ? 'none' : 'block';
144   });
145
146   // Close dropdown if clicked outside
147   document.addEventListener('click', function(event) {
148     const isClickInside = toggleButton.contains(event.target) || dropdown.contains(event.target);

```

```

@section(section: 'scripts')
<script>
  // Get the elements
  const toggleButton = document.getElementById('category-toggle');
  const dropdown = document.querySelector('.category-dropdown');

  // Toggle the dropdown when the button is clicked
  toggleButton.addEventListener('click', function() {
    dropdown.style.display = dropdown.style.display === 'block' ? 'none' : 'block';
  });

  // Close dropdown if clicked outside
  document.addEventListener('click', function(event) {
    const isClickInside = toggleButton.contains(event.target) || dropdown.contains(event.target);
    if (!isClickInside) {
      dropdown.style.display = 'none';
    }
  });
</script>
@endsection

```

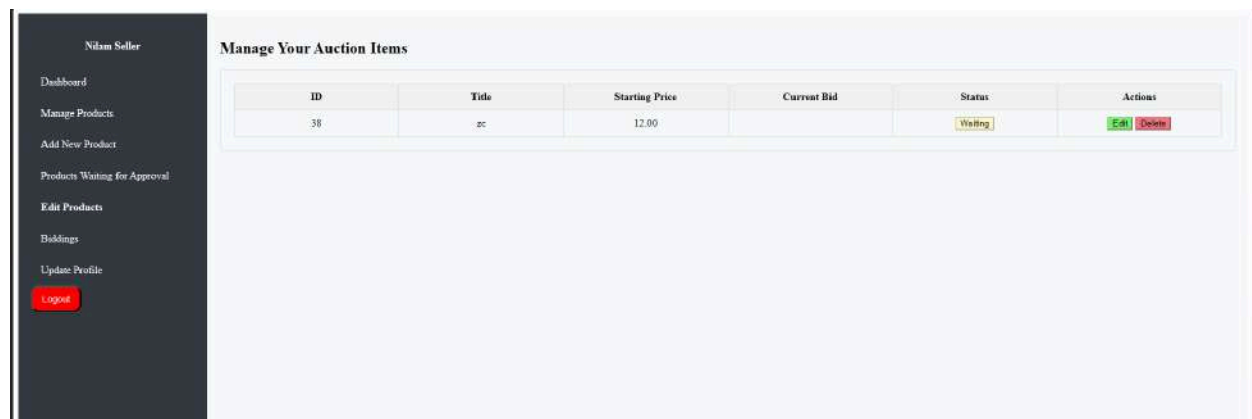
**Products Approval waiting Page functionality:** In this view the seller can see the items that are waiting for admin approval

Nilam Seller Dashboard Manage Products Update Profile Logout		Product Approval		
ID	Title	Starting Price	Current Bid	Status
38	ze	12.00		Waiting

Here is the code for the view

```
3 @section(section: 'content')
4 <div class="container">
5 <h2>Product Approval</h2>
6 <div class="box">
7 <table class="table">
8 <thead>
9 <tr>
10 <th>ID</th>
11 <th>Title</th>
12 <th>Starting Price</th>
13 <th>Current Bid</th>
14 <th>Status</th>
15 </tr>
16 </thead>
17 <tbody>
18 @if ($auctionwaititem->isNotEmpty())
19 @foreach ($auctionwaititem as $auctionwaititem)
20 <tr>
21 <td>{{ $auctionwaititem->id }}</td>
22 <td>{{ $auctionwaititem->title }}</td>
23 <td>{{ $auctionwaititem->starting_price }}</td>
24 <td>{{ $auctionwaititem->current_bid }}</td>
25 <td>
26 @if ($auctionwaititem->status == 3)
27 <button class="btn" disabled
28 style="background-color: #FFFACD; color: #000; border-color: #FFFACD;">Waiting</button>
29 @elseif ($auctionwaititem->status == 2)
30 <button class="btn" disabled
31 style="background-color: #74b2d2; color: #000; border-color: #74b2d2;">Approved</button>
32 @elseif ($auctionwaititem->status == 1)
33 <button class="btn" disabled
34 style="background-color: #7cd274; color: #000; border-color: #7cd274;">Sold</button>
35 @else
36 <button class="btn btn-secondary" disabled>Unknown Status</button>
37 @endif
38 </td>
39 </tr>
40 @endforeach
41 @else
42 <tr>
43 <td colspan="5">No auction items found.</td>
44 </tr>
45 @endif
46 </tbody>
47 </table>
48 </div>
49 </div>
50 @endsection
```

**Edit Product Detail Functionality:** In this view, the seller can see the products that are still in waiting status and edit their title, description and starting price. Also if the item is still in waiting status then the seller himself can delete the product





Here is the code for the view

```
@section(section:"content")


## Manage Your Auction Items/h2> | ID | Title | Starting Price | Current Bid | Status | Action | |----------|-------------|---------------------|------------------|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------| | @item.ID | @item.Title | @item.StartingPrice | @item.CurrentBid | @item.Status | <div> <div> @if (item.Status == 1) { button class="btn btn-primary disabled" style="background-color: #f1f3f4; color: #000;"Waiting/button&gt; } } <div> <div>Edit Button</div> <div>Delete Button</div> </div> </div> </div> | } else { No auction items found. }


```



**Sold Items Bidding Functionality:** In this view, the seller will be able to show the items that he auctioned for sale and the user bought them

<div> <div>Nilam Seller</div> <div> Dashboard Manage Products Add New Product Products Waiting for Approval Edit Products Show Shopped Items Update Profile Logout </div> </div>		<div>Sold Auction Items</div> <table> <tr> <th>Item ID</th><th>Item Title</th><th>Starting Price</th><th>Current Bid</th><th>Bidder Name</th><th>Seller Name</th><th>Payment Status</th></tr> <tr> <td>35</td><td>mao</td><td>1,600.00₹</td><td>1,800.00₹</td><td>Homasirah Fendousia</td><td>Sopnil Roy Niloy</td><td>pending</td></tr> <tr> <td>35</td><td>mao</td><td>1,600.00₹</td><td>2,500.00₹</td><td>Arif Rayhan Pranto</td><td>Sopnil Roy Niloy</td><td>pending</td></tr> <tr> <td>36</td><td>Ceramic Vase Pack 2, White Modern Bud Vase, Ceramic Modern Vase Decor, Sculpture Decor, Fire Place Decoration, Mid Century Modern Drig</td><td>2,000.00₹</td><td>10,020.00₹</td><td>Arif Rayhan Pranto</td><td>Sopnil Roy Niloy</td><td>pending</td></tr> </table>					Item ID	Item Title	Starting Price	Current Bid	Bidder Name	Seller Name	Payment Status	35	mao	1,600.00₹	1,800.00₹	Homasirah Fendousia	Sopnil Roy Niloy	pending	35	mao	1,600.00₹	2,500.00₹	Arif Rayhan Pranto	Sopnil Roy Niloy	pending	36	Ceramic Vase Pack 2, White Modern Bud Vase, Ceramic Modern Vase Decor, Sculpture Decor, Fire Place Decoration, Mid Century Modern Drig	2,000.00₹	10,020.00₹	Arif Rayhan Pranto	Sopnil Roy Niloy	pending
Item ID	Item Title	Starting Price	Current Bid	Bidder Name	Seller Name	Payment Status																												
35	mao	1,600.00₹	1,800.00₹	Homasirah Fendousia	Sopnil Roy Niloy	pending																												
35	mao	1,600.00₹	2,500.00₹	Arif Rayhan Pranto	Sopnil Roy Niloy	pending																												
36	Ceramic Vase Pack 2, White Modern Bud Vase, Ceramic Modern Vase Decor, Sculpture Decor, Fire Place Decoration, Mid Century Modern Drig	2,000.00₹	10,020.00₹	Arif Rayhan Pranto	Sopnil Roy Niloy	pending																												

Here is the code for the view

```

@extends(view: 'layouts.seller')

@section(section: 'content')


## Sold Auction Items



| Item ID                                 | Item Title                                 | Starting Price                                | Current Bid                                                                                                                                                                                                                                                                                                                                                                                                                                        | Bidder Name             | Seller Name                | Payment Status                                        |                                       |                                                   |                                                                   |                                                                           |  |  |  |  |  |
|-----------------------------------------|--------------------------------------------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|----------------------------|-------------------------------------------------------|---------------------------------------|---------------------------------------------------|-------------------------------------------------------------------|---------------------------------------------------------------------------|--|--|--|--|--|
| @foreach (\$auctionItems->isNotEmpty()) | @foreach (\$auctionItems as \$auctionItem) | @foreach (\$auctionItem->bidRecords as \$bid) | <td>{{ \$auctionItem-&gt;id }}</td> <td>{{ \$auctionItem-&gt;title }}</td> <td>{{ number_format(\$auctionItem-&gt;starting_price, 2) }}</td> <td>{{ number_format(\$bid-&gt;amount, 2) }}</td> <td>{{ \$bid-&gt;bidder ? \$bid-&gt;bidder-&gt;name : 'N/A' }}</td> <td>{{ \$auctionItem-&gt;seller ? \$auctionItem-&gt;seller-&gt;name : 'N/A' }}</td> <td>{{ \$auctionItem-&gt;payment ? \$auctionItem-&gt;payment-&gt;status : 'Pending' }}</td> | {{ \$auctionItem->id }} | {{ \$auctionItem->title }} | {{ number_format(\$auctionItem->starting_price, 2) }} | {{ number_format(\$bid->amount, 2) }} | {{ \$bid->bidder ? \$bid->bidder->name : 'N/A' }} | {{ \$auctionItem->seller ? \$auctionItem->seller->name : 'N/A' }} | {{ \$auctionItem->payment ? \$auctionItem->payment->status : 'Pending' }} |  |  |  |  |  |


```

**Show Active Bidding Functionaity:** In this view user will be able to view the active biddings

<div> <div> Nilam Seller </div> <div> Dashboard </div> <div> Manage Products </div> <div> Update Profile </div> <div> Logout </div> </div>																				
<div> <div>Items with Active Biddings</div> <div> Total Items with Active Biddings: 4 </div> <table> <tr> <th>Item ID</th><th>Item Title</th><th>Starting Price</th><th>Current Bid</th><th>Bidder Name</th><th>Seller Name</th><th>Status</th></tr> <tr> <td>19</td><td>Toyota LandCruiser 300 (LC300)</td><td>5,000,000.00b</td><td>5,000,001.00b</td><td>Ahif Rayhan Piarso</td><td>Sopnil Roy Niloy</td><td>Active Bidding</td></tr> </table> </div>							Item ID	Item Title	Starting Price	Current Bid	Bidder Name	Seller Name	Status	19	Toyota LandCruiser 300 (LC300)	5,000,000.00b	5,000,001.00b	Ahif Rayhan Piarso	Sopnil Roy Niloy	Active Bidding
Item ID	Item Title	Starting Price	Current Bid	Bidder Name	Seller Name	Status														
19	Toyota LandCruiser 300 (LC300)	5,000,000.00b	5,000,001.00b	Ahif Rayhan Piarso	Sopnil Roy Niloy	Active Bidding														

```

@extends('view: 'layouts.seller')

@section(section: 'content')


## Items with Active Biddings



Total Items with Active Biddings: [[ $itemsCount ]]



| Item ID                                   | Item Title | Starting Price | Current Bid | Bidder Name | Seller Name | Status |
|-------------------------------------------|------------|----------------|-------------|-------------|-------------|--------|
| No active biddings found for this seller. |            |                |             |             |             |        |


```

```


```

## Backend Development

```

use App\Http\Controllers\ProfileController;
use Illuminate\Support\Facades\Route;
use Laravel\Socialite\Facades\Socialite;
use App\Models\User;
use Illuminate\Support\Facades\Auth;
use App\Http\Controllers\GoogleController;
use App\Http\Controllers\ActionApprovalController;
use App\Http\Controllers\AdminDashboardController;
use App\Http\Controllers\SellerDashboardController;
use App\Http\Controllers\TransactionManagementController;
use App\Http\Controllers\Auth\PasswordController;
use App\Http\Controllers\AuctionController;
use App\Http\Controllers\CategoryController;
use App\Http\Controllers\SearchController;
use App\Http\Controllers\AccountController;
use App\Http\Controllers\TransactionController;
use App\Http\Controllers\BidController;
use App\Http\Controllers\PaymentApprovalController;

Route::get('/' , function () {return view('welcome');});

// Google login
Route::get('/auth/google', action: [App\Http\Controllers\GoogleController::class, 'redirectToGoogle'])->name('auth.google');
Route::get('/auth/google/callback', action: [App\Http\Controllers\GoogleController::class, 'handleGoogleCallback'])->name('auth.google.callback');

// Normal user dashboard
Route::get('/dashboard', action: [UserController::class, 'dashboard'])->middleware([middleware: ['auth', 'verified', 'user']])>name('user.dashboard');
Route::post('/user/update-password', action: [PasswordController::class, 'updatePassword'])->name('user.updatePassword');
Route::get('/transactions', action: [TransactionController::class, 'index'])->name('user.transactions');
Route::post('/action/confirm-purchase/{id}', action: [TransactionController::class, 'confirmPurchase'])->name('user.action.confirmPurchase');

// Auction item fetch logic
Route::get('/all-items', action: [AuctionController::class, 'getAllItems'])->middleware([middleware: ['auth', 'verified', 'user']])>name('user.all-items');
Route::get('/item-details/{id}', action: [AuctionController::class, 'show'])->middleware([middleware: ['auth', 'verified', 'user']])>name('user.auction.show');
Route::get('/categories', action: [CategoryController::class, 'showCategory'])->middleware([middleware: ['auth', 'verified', 'user']])>name('user.categories.show');
Route::get('/search', action: [SearchController::class, 'search'])->middleware([middleware: ['auth', 'verified', 'user']])>name('user.search');
Route::post('/action/{id}/place-bid', action: [AuctionController::class, 'placeBid'])->name('user.placeBid');
Route::post('/buy-now', action: [AuctionController::class, 'buyNow'])->name('user.buyNow');

// Seller dashboard
Route::get('/seller/dashboard', action: [SellerDashboardController::class, 'index'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.dashboard');
Route::get('/seller/auction-items/create', action: [SellerDashboardController::class, 'createAuctionItem'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.createAuctionItem');
Route::post('/seller/auction-item/store', action: [SellerDashboardController::class, 'storeAuctionItem'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.storeAuctionItem');
Route::get('/seller/auction-item/status', action: [SellerDashboardController::class, 'showAuctionItem'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.auction-item-status');
Route::get('/seller/sold-items-records', action: [SellerDashboardController::class, 'soldItems'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.sold-items');
Route::get('/seller/edit-item-requests', action: [SellerDashboardController::class, 'editItems'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.edit-items');
Route::get('/seller/bidding-items', action: [SellerDashboardController::class, 'bidding'])->middleware([middleware: ['auth', 'verified', 'seller']])>name('seller.bidding-items');

// Delete Auction Item
Route::delete('/seller/auction/{id}/delete', action: [SellerDashboardController::class, 'delete'])->name('seller.auction.delete');

// Update Auction Item
Route::put('/seller/auction/{id}/update', action: [SellerDashboardController::class, 'update'])->name('seller.auction.update');

// Admin dashboard
Route::get('/admin/dashboard', action: [AdminDashboardController::class, 'index'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.dashboard');
Route::get('/admin/users', action: [AdminDashboardController::class, 'users'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.users');
Route::delete('/admin/users/{id}', action: [AdminDashboardController::class, 'destroy'])->name('admin.users.destroy');

Route::get('/admin/sellers', action: [AdminDashboardController::class, 'sellers'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.sellers');
Route::get('/admin/add', action: [AdminDashboardController::class, 'showAddSellerForm'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.add');
Route::post('/admin/register', action: [AdminDashboardController::class, 'registerSeller'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.register');
Route::get('/admin/auction-status', action: [AuctionApprovalController::class, 'index'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.auction-status');
Route::get('/admin/auction-approve', action: [AuctionApprovalController::class, 'index'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.auction-approve');
Route::get('/admin/bids', action: [AuctionManagementController::class, 'bids'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.bids');

Route::get('/admin/payment-approval', action: [PaymentApprovalController::class, 'index'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.payment-approval');
Route::get('/admin/payment-approve/{id}', action: [PaymentApprovalController::class, 'approve'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.payment-approve');
Route::get('/admin/payment-reject/{id}', action: [PaymentApprovalController::class, 'reject'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.payment-reject');

// Route to delete auction item
Route::delete('/admin/auction/{id}/delete', action: [AuctionManagementController::class, 'delete'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.auction.delete');

// Route to approve auction item
Route::put('/admin/auction/{id}/approve', action: [AuctionManagementController::class, 'approve'])->middleware([middleware: ['auth', 'verified', 'admin']])>name('admin.auction.approve');

Route::middleware([middleware: ['auth']])>group(function () {
    Route::get('/profile', action: [ProfileController::class, 'edit'])->name('user.profile.edit');
    Route::patch('/profile', action: [ProfileController::class, 'update'])->name('user.profile.update');
    Route::delete('/profile', action: [ProfileController::class, 'destroy'])->name('user.profile.destroy');

    Route::put('/password', action: [PasswordController::class, 'update'])->name('user.password.update');
});

```

```
require __DIR__ . '/auth.php';
```

Figure: web.php

```

</php>

use App\Http\Controllers\Auth\AuthenticatedSessionController;
use App\Http\Controllers\Auth\ConfirmablePasswordController;
use App\Http\Controllers\Auth\EmailVerificationNotificationController;
use App\Http\Controllers\Auth\EmailVerificationPromptController;
use App\Http\Controllers\Auth\NewPasswordController;
use App\Http\Controllers\Auth>PasswordController;
use App\Http\Controllers\Auth>PasswordResetLinkController;
use App\Http\Controllers\Auth\RegisteredUserController;
use App\Http\Controllers\Auth\VerifyEmailController;
use Illuminate\Support\Facades\Route;

Route::middleware('guest')->group(callback: function () { void {
    Route::get(uri: 'register', action: [RegisteredUserController::class, 'create'])
        ->name(name: 'register');

    Route::post(uri: 'register', action: [RegisteredUserController::class, 'store']);

    Route::get(uri: 'login', action: [AuthenticatedSessionController::class, 'create'])
        ->name(name: 'login');

    Route::post(uri: 'login', action: [AuthenticatedSessionController::class, 'store']);

    Route::get(uri: 'forgot-password', action: [PasswordResetLinkController::class, 'create'])
        ->name(name: 'password.request');

    Route::post(uri: 'forgot-password', action: [PasswordResetLinkController::class, 'store'])
        ->name(name: 'password.email');

    Route::get(uri: 'reset-password/{token}', action: [NewPasswordController::class, 'create'])
        ->name(name: 'password.reset');

    Route::post(uri: 'reset-password', action: [NewPasswordController::class, 'store'])
        ->name(name: 'password.store');
});

Route::middleware('auth')->group(callback: function () { void {
    Route::get(uri: 'verify-email', action: [EmailVerificationPromptController::class])
        ->name(name: 'verification.notice');

    Route::get(uri: 'verify-email/{id}/{hash}', action: [VerifyEmailController::class])
        ->middleware(middleware: ['signed', 'throttle:6,1'])
        ->name(name: 'verification.verify');

    Route::post(uri: 'email/verification-notification', action: [EmailVerificationNotificationController::class, 'store'])
        ->middleware(middleware: 'throttle:6,1')
        ->name(name: 'verification.send');
});

Route::middleware('auth')->group(callback: function () { void {
39 Route::get(uri: 'verify-email', action: [EmailVerificationPromptController::class])
40     ->name(name: 'verification.notice');
41
42     Route::get(uri: 'verify-email/{id}/{hash}', action: [VerifyEmailController::class])
43         ->middleware(middleware: ['signed', 'throttle:6,1'])
44         ->name(name: 'verification.verify');
45
46     Route::post(uri: 'email/verification-notification', action: [EmailVerificationNotificationController::class, 'store'])
47         ->middleware(middleware: 'throttle:6,1')
48         ->name(name: 'verification.send');
49
50     Route::get(uri: 'confirm-password', action: [ConfirmablePasswordController::class, 'show'])
51         ->name(name: 'password.confirm');
52
53     Route::post(uri: 'confirm-password', action: [ConfirmablePasswordController::class, 'store']);
54
55     Route::put(uri: 'password', action: [PasswordController::class, 'update'])->name(name: 'password.update');
56
57     Route::post(uri: 'logout', action: [AuthenticatedSessionController::class, 'destroy'])
58         ->name(name: 'logout');
59
60 });

```

Figure: auth.php



```

1 <?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use App\Http\Controllers\Controller;
6 use Illuminate\Http\RedirectResponse;
7 use Illuminate\Http\Request;
8 use Illuminate\Support\Facades\Hash;
9 use Illuminate\Validation\Rules\Password;
10 use Illuminate\Support\Facades\Log;
11
12 5 references | 0 implementations
13 class PasswordController extends Controller
14 {
15     /**
16      * Update the user's password.
17      */
18
19     1 reference | 0 overrides
20     public function updatePassword(Request $request, RedirectResponse
21     {
22         $request->validate([
23             'current_password' => ['required', 'string', 'min:8'],
24             'password' => ['required', 'string', 'min:8', 'confirmed'],
25         ]);
26
27         // Check if current password matches the user's actual password
28         if (!Hash::check($request->current_password, Hash::make(auth()->user()->password))) {
29             return back()->withErrors(provider: ['current_password' => 'Current password is incorrect.']);
30         }
31
32         // Update the new password (hash it first)
33         $user = auth()->user();
34         $user->password = Hash::make($request->password);
35         $user->save();
36
37         return back()->with('status', value: 'Password updated successfully!');
38     }
39 }

```

Figure: Password controller

```

<?php

namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use App\Http\Requests\Auth\LoginRequest;
use Illuminate\Http\RedirectResponse;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Illuminate\View\View;

/**
 * @reference | 0 implementations
 */
class AuthenticatedSessionController extends Controller
{
    /**
     * Display the login view.
     */
    /**
     * @reference | 0 overrides
     */
    public function create(): View
    {
        return view('auth.login');
    }

    /**
     * Handle an incoming authentication request.
     */
    /**
     * @reference | 0 overrides
     */
    public function store(LoginRequest $request): RedirectResponse
    {
        $request->authenticate();

        $request->session()->regenerate();

        // Now will be redirecting based on the user role
        // dd($request->all(), $request->user());
        $loggedInUserRole = $request->user()->role;

        // Seller
        if($loggedInUserRole==1){
            return redirect()->intended(default: route(name: 'seller.dashboard', absolute: false));
        }
        // Admin
        elseif($loggedInUserRole==2){
            return redirect()->intended(default: route(name: 'admin.dashboard', absolute: false));
        }

        return redirect()->intended(default: route(name: 'dashboard', absolute: false));
    }
}

```

Figure: Authenticated session User



```

<?php

namespace App\Http\Controllers;

use App\Models\AuctionItem;
use App\Models\BidRecord;
use Illuminate\Http\Request;

/**
 * @reference: 0 implementations
 */
class AccountController extends Controller
{
    /**
     * @reference: 0 overrides
     */
    public function index(): Factory|View
    {
        return view('views.account');
    }

    /**
     * @reference: 0 overrides
     */
    public function editProfile(): Factory|View
    {
        // Return view to update profile
        return view('views.account.edit-profile');
    }

    /**
     * @reference: 0 overrides
     */
    public function editPassword(): Factory|View
    {
        // Return view to update password
        return view('views.account.edit-password');
    }

    /**
     * @reference: 0 overrides
     */
    public function wonHistory(): Factory|View
    {
        $wonAuctions = AuctionItem::whereHas('bidRecords', callback: function ($query): void {
            $query->where('customer_id', auth()->id())
                ->whereColumn('bid_records.amount', 'auction_items.current_bid')
                ->where('auction_items.end_time', '<', now());
        })->get();

        return view('views.account.won-history', data: compact('var_name', 'wonAuctions'));
    }

    // Fetch the user's transactions
    /**
     * @reference: 0 overrides
     */
    public function transactions(): Factory|View
    {
        $transactions = BidRecord::where('customer_id', auth()->id())->whereHas('auctionitem', callback: function($query): void {
            $query->where('end_time', '<', now());
        })->get();

        return view('views.account.transactions', data: compact('var_name', 'transactions'));
    }
}

```

Figure: AccountController

```

<?php
namespace App\Http\Controllers;

use App\Models\User;
use App\Models\AuctionItem;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Illuminate\Support\Facades\Auth;

/**
 * @property $request
 */
class AdminDashboardController extends Controller
{
    //
    /** reference: @overwrites */
    public function index(): FactoryView
    {
        // Dashboard content
        // return view('admin.dashboard');
        $admin = Auth::user();

        return view('admin.dashboard', data: compact('var_name', 'admin'));
    }

    /** reference: @overwrites */
    public function sellerDashboard(): FactoryView
    {
        $seller = Auth::user(); // Assuming the authenticated user is the seller

        return view('admin.seller.dashboard', data: compact('var_name', 'seller'));
    }

    /** reference: @overwrites */
    public function users(): FactoryView
    {
        // Fetch normal users
        $normalUsers = User::where('role', '=', 1)
            >get(['id', 'name', 'email']);
        return view('admin.users', data: compact('var_name', 'normalUsers'));
    }

    /** reference: @overwrites */
    public function sellers(): FactoryView
    {
        // Fetch sellers (assuming role 2 is for sellers)
        $sellers = User::where('role', '=', 2)
            >get(['id', 'name', 'email']);
        return view('admin.sellers', data: compact('var_name', 'sellers'));
    }

    /** reference: @overwrites */
    public function aucApprove(): FactoryView
    {
        // Fetch auction items with seller details
        $auctionItems = AuctionItem::with('seller')
            >get(); // eager load seller relationship

        return view('admin.auc-status', data: compact('var_name', 'auctionItems'));
    }

    /** reference: @overwrites */
    public function showAddAdminForm(): FactoryView
    {
        return view('admin.add-admin');
    }

    // Handle the registration of the new admin
    /** reference: @overwrites */
    public function registerAdmin(Request $request): RedirectResponse
    {
        $request->validate([
            'name' => 'required|string|max:255',
            'username' => 'required|string|max:355',
            'email' => 'required|string|email|max:255|unique:users',
            'password' => 'required|string|min:8|confirmed',
        ]);

        User::create([
            'name' => $request->name,
            'username' => $request->username,
            'email' => $request->email,
            'password' => Hash::make($request->password),
            'role' => 2, // set role to 2 for admin
        ]);

        return redirect()->route('admin.dashboard')->with('key', 'success', 'value', 'New admin registered successfully!');
    }

    /** reference: @overwrites */
    public function destroy($id): RedirectResponse
    {
        $user = User::findOrFail($id);
        $user->delete();

        return redirect()->route('admin.dashboard')->with('key', 'success', 'value', 'User deleted successfully. ');
    }
}

```

Figure: Admin Dashboard Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\AuctionItem;
use Illuminate\Http\Request;

2 references | 0 implementations
class AuctionApprovalController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory|View
    {
        // Fetch auction items with seller details
        $auctionItems = AuctionItem::with(relations: 'seller')->get(); // Eager load seller relationship

        return view(view: 'admin.auc-status', data: compact(var_name: 'auctionItems'));
    }
}

```

Figure: Auction Approval Controller

```

<?php

namespace App\Http\Controllers;

use App\Models\AuctionItem;
use App\Models\BidRecord;
use App\Models\Category;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Auth;
use Carbon\Carbon;
use Illuminate\Support\Str;
use App\Models\Payment;

/**
 * @property-read \Illuminate\Pagination\LengthAwarePaginator $auctionItems
 */
class AuctionController extends Controller
{
    /**
     * @return \Illuminate\Contracts\View\Factory|\View
     */
    public function getallitem()
    {
        $auctionItems = AuctionItem::where('column', 'status', 'operation')
            ->with(['categories', 'images' => function ($query) {
                $query->take(1); // Only fetch the first image for each auction item
            }])
            ->paginate(perPage: 9); // 9 items per page (3 rows x 3 columns)

        return view('view.all_items', ['data' => compact('var_name', 'auctionItems')]);
    }

    /**
     * @return \Illuminate\Contracts\View\Factory|\View
     */
    public function dashboard()
    {
        // Fetch only the 3 most recent auction items
        $categories = Category::all();

        $auctionItems = AuctionItem::where('column', 'status', 'operation')
            ->orderBy('created_at', 'desc') // Order by creation date
            ->with(['categories', 'images'])
            ->take(3) // Limit to 3 items
            ->get();

        return view('view.dashboard', ['data' => compact('var_name', 'categories', 'var_name', 'auctionItems')]);
    }

    /**
     * @return \Illuminate\Contracts\View\Factory|\View
     */
    public function show($id)
    {
        $auctionItem = AuctionItem::with(['images', 'categories', 'seller'])->findOrFail($id);
        $recentItems = AuctionItem::latest()->take(5)->get();
    }
}

```

```

12  /**
13   * @return \Illuminate\Contracts\View\Factory|\View
14   */
15  public function show($id)
16  {
17      $auctionItem = AuctionItem::with(['images', 'categories', 'seller'])->findOrFail($id);
18      $recentItems = AuctionItem::latest()->take(5)->get();
19
20      $timeLeft = $this->calculateTimeLeft($auctionItem->end_time);
21      $highestBidder = $this->highestBidder($auctionItem->end_time);
22
23      return view('view.auction.show', ['data' => compact('var_name', 'auctionItem', 'var_name', 'recentItems', 'timeLeft', 'highestBidder')]);
24  }
25
26  /**
27   * @return \Illuminate\Contracts\View\Factory|\View
28   */
29  private function calculateTimeLeft($endTime)
30  {
31      $now = Carbon::now();
32      $end = Carbon::parse($endTime);
33
34      if ($now->gt($end)) {
35          return [object] {
36              's' => 0,
37              'm' => 0,
38              's' => 0
39          };
40      }
41
42      $diff = $now->diff($end);
43
44      return [object] {
45          's' => $diff->s,
46          'm' => $diff->m,
47          'h' => $diff->h
48      };
49  }
50
51  /**
52   * @return \Illuminate\Contracts\View\Factory|\View
53   */
54  private function highestBidder($auctionItem)
55  {
56      if (!$this->check()) {
57          return false;
58      }
59
60      $highestBid = $auctionItem->bidRecords()->orderBy('amount', 'desc')->first();
61      return $highestBid && $highestBid->customer_id == Auth::id();
62  }
63  }

```

```

1 public function placeBid(Request $request) { RedirectResponse
2
3     $request->validate(rules:[
4         'bid_amount' => 'required|numeric|min:0',
5     ]);
6
7     $auctionItem = AuctionItem::find($request->auction_item_id);
8
9     // Check if auction has ended
10    if ($auctionItem->isAuctionEnded()) {
11        return redirect()->back()->with(key:'error', value:'The auction has already ended. You cannot place a bid.');
```

```

12    }
13
14    // Check if bid amount is less than starting price
15    if ($request->bid_amount < $auctionItem->starting_price) {
16        return redirect()->back()->with(key:'error', value:'Bid amount must be higher than the starting price.');
```

```

17    }
18
19    // Check if bid amount is less than or equal to the current highest bid
20    $highestBid = $auctionItem->bidRecords()->max('amount');
21    if ($highestBid && $request->bid_amount <= $highestBid) {
22        return redirect()->back()->with(key:'error', value:'Bid amount must be higher than the current highest bid.');
```

```

23    }
24
25    // Save the bid in bid records
26    $bidRecord::create(attributes:[
27        'auction_id' => $auctionItem->id,
28        'customer_id' => Auth::id(),
29        'amount' => $request->bid_amount,
30    ]);
31
32    // Update current bid in auction item
33    $auctionItem->current_bid = $request->bid_amount;
34    $auctionItem->save();
35
36    return redirect()->back()->with(key:'success', value:'bid placed successfully!');
```

```

37    }
38    // reference: 0 overrides
39    public function buyNow(Request $request) { RedirectResponse
40    {
41        $request->validate(rules:[
42            'payment_method' => 'required',
43            'auction_item_id' => 'required|exists:auction_items,id',
44        ]);
45
46        $auctionItem = AuctionItem::findOrFail($request->auction_item_id);
47
48        // handle payment processing logic
49
50        // generate random token
51        $token = Str::random(15);
52
53        // Other payment methods can be handled here (e.g., credit card)
54        $token = null;
55
56        // Save payment record
57        $payment::create(attributes:[
58            'user_id' => Auth::id(),
59            'auction_id' => $auctionItem->id,
60            'amount' => $auctionItem->current_bid,
61            'payment_method' => $request->payment_method,
62            'token' => $token,
63            'status' => 'pending', // item is pending initially
64        ]);
65
66        $auctionItem->update(['status' => 'sold']);
67
68        return redirect()->back()->with(key:'success', value:'Purchase successful! Check your transactions for details.');
```

```

69    }
70
71    // handle payment processing logic
72    if ($request->payment_method == 'cash-on-delivery') {
73        $token = Str::random(15); // generate random token
74    } else {
75        // Other payment methods can be handled here (e.g., credit card)
76        $token = null;
77    }
78
79    // Save payment record
80    $payment::create(attributes:[
81        'user_id' => Auth::id(),
82        'auction_id' => $auctionItem->id,
83        'amount' => $auctionItem->current_bid,
84        'payment_method' => $request->payment_method,
85        'token' => $token,
86        'status' => 'pending', // item is pending initially
87    ]);
88
89    $auctionItem->update(['status' => 'sold']);
90
91    return redirect()->back()->with(key:'success', value:'Purchase successful! Check your transactions for details.');
```

Figure:AuctionController

```

1 //php
2
3
4 namespace App\Http\Controllers;
5
6 use App\Models\AuctionItem;
7 use App\Models\BidRecord;
8 use App\Models\User;
9 use Illuminate\Http\Request;
10
11 //reference: 10 implements
12 class AuctionManagementController extends Controller
13 {
14     //
15
16     //reference: 10 overrides
17     public function index() { Factory|View
18     {
19         // Fetch auction items with seller details
20         $auctionItems = AuctionItem::where('status', 'operator: >')->with('relations: seller')->get();
21
22         return view('admin.auc-approve', ['data' => compact('var_name', 'auctionItem')]);
23     }
24
25     //reference: 10 overrides
26     public function delete($id) { Redirect|Response
27     {
28         $auctionItem = AuctionItem::find($id);
29
30         if ($auctionItem) {
31             $auctionItem->delete(); // Delete the auction item
32             return redirect()->back()->with('key: success', 'value: Auction item deleted successfully!');
33         }
34
35         return redirect()->back()->with('key: error', 'value: Auction item not found!');
36     }
37
38     // Function to approve an auction item
39     //reference: 10 overrides
40     public function approve($id) { Redirect|Response
41     {
42         $auctionItem = AuctionItem::find($id);
43
44         if ($auctionItem) {
45             $auctionItem->status = 2; // Set status to 2 (Approved)
46             $auctionItem->save(); // Save the update
47             return redirect()->back()->with('key: success', 'value: Auction item approved successfully!');
48         }
49
50         return redirect()->back()->with('key: error', 'value: Auction item not found!');
51     }
52
53     //reference: 10 overrides
54     public function bids() { Factory|View
55     {
56         // Get all auction items where the status is 2, and the current bid is not null or 0
57         $auctionItems = AuctionItem::where('status', 'operator: 2')
58             ->whereNotNull('current_bid')
59             ->where('current_bid', 'operator: >', 'value: 0')
60             ->with('relations: [bidRecords, seller]')
61             ->get();
62
63         // Join with the bid records and users to get the required data
64         $items = $auctionItems->map(callback: function (AuctionItem $auctionItem): array {
65             // Get the highest bid record for this auction item
66             $highestBid = $auctionItem->bidRecords()->orderBy('amount', 'direction: desc')->first();
67             $bidder = $highestBid ? User::find($highestBid->customer_id) : null;
68
69             return [
70                 'item_id' => $auctionItem->id,
71                 'item_title' => $auctionItem->title,
72                 'starting_price' => $auctionItem->starting_price,
73                 'current_price' => $auctionItem->current_bid,
74                 'bidder_name' => $bidder ? $bidder->name : 'N/A',
75                 'seller_name' => $auctionItem->seller ? $auctionItem->seller->name : 'N/A',
76             ];
77         });
78
79         return view('admin.bids', ['data' => compact('var_name', 'items')]);
80     }
81 }
82 }
83 }

```

Figure: AuctionManagementController

```

<?php

namespace App\Http\Controllers;

use App\Models\AuctionItem;
use App\Models\Category;
use Illuminate\Http\Request;

2 references | 0 implementations
class CategoryController extends Controller
{
    1 reference | 0 overrides
    public function showByCategory(Request $request): Factory|View
    {
        $categoryName = $request->query(key: 'category');

        // Fetch the category
        $category = Category::where(column: 'name', operator: $categoryName)->firstOrFail();

        // Get the auction items related to this category
        $auctionItems = AuctionItem::whereHas(relation: 'categories', callback: function ($query) use ($category): void {
            $query->where('category_id', $category->id);
        })->with(relations: 'images', 'categories')->get();

        // Pass the auction items and category to the view
        return view(view: 'auction.filtered_items', data: compact(var_name: 'auctionItems', var_names: 'category'));
    }
}

```

Figure: Category Controller

```

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Laravel\Socialite\Facades\Socialite;
use App\Models\User;
use Illuminate\Support\Facades\Auth;

3 references | 0 implementations
class GoogleController extends Controller
{
    1 reference | 0 overrides
    public function redirectToGoogle(): RedirectResponse
    {
        return Socialite::driver(driver: 'google')->redirect();
    }

    1 reference | 0 overrides
    public function handleGoogleCallback(): mixed|Redirector|RedirectResponse
    {
        $user = Socialite::driver(driver: 'google')->user();
        $findUser = User::where(column: 'google_id', operator: $user->id)->first();
        if ($findUser) {
            Auth::login(user: $findUser);
            $findUser->update([
                'avatar' => $user->getAvatar()
            ]);
            return redirect()->intended(default: 'dashboard');
        } else {
            $user = User::updateOrCreate(attributes: [
                'email' => $user->email,
            ], values: [
                'name' => $user->name,
                'google_id' => $user->token,
                'password' => encrypt(value: '12345678'),
                'avatar' => $user->avatar, // Store Google profile picture
            ]);

            Auth::login(user: $user);
        }
        return redirect(to: '/dashboard');
    }
}

```

Figure: GoogleController

```

<?php
namespace App\Http\Controllers;

use App\Models\Payment;
use Illuminate\Http\Request;
use App\Models\AuctionItem;
use App\Models\BidRecord;
use App\Models\User;

4 references | 0 implementations
class PaymentApprovalController extends Controller
{
    1 reference | 0 overrides
    public function index(): Factory\View
    {
        // Fetch all payments with status pending
        $payments = Payment::where('status', 'pending')->with('relations: auctionItem, user')->get();

        return view('admin.payment-approval', data: compact('var_name', 'payments'));
    }

    1 reference | 0 overrides
    public function approve($id): RedirectResponse
    {
        $payment = Payment::findOrFail(id: $id);
        $payment->status = 'approved';
        $payment->save();

        return redirect()->back()->with(key: 'success', value: 'Payment approved successfully.');

```

Figure: Payment Approval Controller



```

1 <?php
2 namespace App\Http\Controllers;
3
4 use App\Http\Requests\ProfileUpdateRequest;
5 use Illuminate\Http\RedirectResponse;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Auth;
8 use Illuminate\Support\Facades\Redirect;
9 use Illuminate\View\View;
10
11
12 class ProfileController extends Controller
13 {
14     /**
15      * Display the user's profile form.
16      */
17     /**reference|Overides
18      * public function edit(Request $request): View
19      * {
20      *     return view('profile.edit', data:[
21      *         'user' => $request->user(),
22      *     ]);
23      * }
24
25     /**reference|Overides
26     * public function update(Request $request): RedirectResponse
27     * {
28     *     $request->validate(rules:[
29     *         'username' => ['required','string','max:255'],
30     *         'avatar' => ['nullable','image','mimes:jpeg,png,jpg,gif,svg','max:2048'],
31     *         'address' => ['nullable','string','max:255'],
32     *     ]);
33
34     *     $user = Auth::user();
35
36     *     // Update username and address
37     *     $user->username = $request->input(key: 'username');
38     *     $user->address = $request->input(key: 'address'); // Add this
39
40     *     // Handle avatar upload
41     *     if ($request->hasFile(key: 'avatar')) {
42     *         $avatarName = time() . '.' . $request->avatar->extension();
43     *         $request->avatar->storeAs('avatars', $avatarName, 'public');
44     *         $user->avatar = $avatarName;
45     *     }
46
47     *     $user->save();
48     * }
49
50 }

```

```

51
52 public function updateProfileInformation(Request $request): RedirectResponse
53 {
54     $request->validate(rules:[
55         'username' => ['required', 'string', 'max:255', 'unique:users,username,' . auth()->id()],
56         'avatar' => ['nullable', 'image', 'mimes:jpeg,png,jpg,gif,svg', 'max:5120'], // Validation for image
57     ]);
58
59     $user = auth()->user();
60
61     // Handle avatar upload
62     if ($request->hasFile(key: 'avatar')) {
63         $avatar = $request->file(key: 'avatar');
64         $avatarName = time() . '.' . $avatar->getClientOriginalExtension();
65         $avatar->storeAs(path: 'public/avatars', name: $avatarName); // Store in "public/avatars"
66
67         // Update avatar in the user table
68         $user->avatar = $avatarName;
69     }
70
71     // update username
72     $user->username = $request->username;
73     $user->save();
74
75     return redirect()->back()->with(key: 'status', value: 'Profile updated successfully!');
76 }
77
78 /**
79  * Delete the user's account.
80  */
81 /**reference|Overides
82  * public function destroy(Request $request): RedirectResponse
83  * {
84  *     $request->validateWithBag('accountDeletion', rules:[
85  *         'password' => ['required', 'current_password'],
86  *     ]);
87
88  *     $user = $request->user();
89
90     *     Auth::logout();
91
92     *     $user->delete();
93     * }
94
95 }

```

```
1 reference | 0 overrides
public function destroy(Request $request): RedirectResponse
{
    $request->validateWithBag(errorBag: 'userDeletion', rules: [
        'password' => ['required', 'current_password'],
    ]);

    $user = $request->user();

    Auth::logout();

    $user->delete();

    $request->session()->invalidate();
    $request->session()->regenerateToken();

    return Redirect::to(path: '/');
}
}
```

Figure: Profile Controller

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\Models\AuctionItem;

2 references | 0 implementations
class SearchController extends Controller
{
    1 reference | 0 overrides
    public function search(Request $request): Factory\View
    {
        $query = $request->input(key: 'query');

        // Search in auction items' titles and descriptions
        $items = AuctionItem::where(column: 'title', operator: 'LIKE', value: "%{$query}%")
            ->orWhere(column: 'description', operator: 'LIKE', value: "%{$query}%")
            ->with(relations: 'images')
            ->get();

        return view(view: 'search_results', data: compact(var_name: 'items', var_names: 'query'));
    }
}
```

Figure: SearchController

```

2. // C#

namespace App\Http\Controllers
{
    use App\Models\User;
    use App\Models\AuctionItem;
    use App\Models\Category;
    use App\Models\Image;
    use Illuminate\Http\Request;
    use Illuminate\Support\Facades\Auth;

    /**
     * @reference: [Singletons]
     */
    class SellerDashboardController extends Controller
    {
        /**
         * @reference: [Singletons]
         */
        public function index() { Factory\view }
        {
            $seller = Auth::user(); // Assuming the authenticated user is the seller

            // Get the count of auctioned item for this seller
            $auctionItemsCount = $seller->auctionItems()->count();

            // Assuming 'status' or 'is_sold' indicates whether an item is sold
            $soldItemsCount = $seller->auctionItems()->where('column', 'status', 'operator', 'sold')->count();

            return view('seller.dashboard', data: compact('var_name', 'seller', 'var_name', 'auctionItemsCount', 'soldItemsCount'));
        }

        /**
         * @reference: [Singletons]
         */
        public function showWaiting() { Factory\view }
        {
            // Fetch auction items where status is 3 (waiting for approval)
            $seller = Auth::user();

            // Fetch auction items where status is 3 and seller_id is the current seller
            $auctionItems = AuctionItem::where('column', 'status', 'operator', 3)->where('column', 'seller_id', 'operator', $seller->id)->get();

            return view('seller.items-waiting', data: compact('var_name', 'auctionWaiting'));
        }

        /**
         * @reference: [Singletons]
         */
        public function editItems() { Factory\view }
        {
            $seller = Auth::user(); // Get the current seller
            $auctionItems = AuctionItem::where('column', 'status', 'operator', 1)->where('column', 'seller_id', 'operator', $seller->id)->get(); // Fetch only status 1 items

            return view('seller.edit-items', data: compact('var_name', 'auctionItems'));
        }

        /**
         * @reference: [Singletons]
         */
        public function soldItems() { Factory\view }
        {
            // Get the currently logged-in seller's ID
            $sellerId = Auth::id();

            // Fetch auction items where status is 2 and seller_id is the current seller's ID
            $auctionItems = AuctionItem::where('column', 'status', 'operator', 2)
                ->where('column', 'seller_id', 'operator', $sellerId)
                ->with('relations' => ['bidRecords.bidder', 'seller', 'payment'])
                ->get();

            $itemsCount = $auctionItems->count();

            // Return both the auction items and the count to the view
            return view('seller.sold-items', data: compact('var_name', 'auctionItems', 'var_name', 'itemsCount'));
        }

        /**
         * @reference: [Singletons]
         */
        public function biddings() { Factory\view }
        {
            // Get the currently logged-in seller's ID
            $sellerId = Auth::id();

            // Fetch auction items where status is 1 or 2 and seller_id is the current seller's ID
            $auctionItems = AuctionItem::whereIn('column', 'status', 'values', [1, 2])
                ->where('column', 'seller_id', 'operator', $sellerId)
                ->with('relations' => ['bidRecords.bidder', 'seller', 'payment']) // Ignore last relationships
                ->get();

            // Get the count of auction items
            $itemsCount = $auctionItems->count();

            // Return both the auction items and the count to the view
            return view('seller.biddings', data: compact('var_name', 'auctionItems', 'var_name', 'itemsCount'));
        }

        /**
         * @reference: [Singletons]
         */
        public function delete($id) { Request\delete }
        {
            $auctionItem = AuctionItem::find($id);

            if ($auctionItem && $auctionItem->seller_id == Auth::id()) {
                $auctionItem->delete();
                return redirect()->back()->with('key', 'success', 'value', 'Auction item deleted successfully!');
            }

            return redirect()->back()->with('key', 'error', 'value', 'Auction item not found or not authorized!');
        }
    }
}

```

```

104 public function update(Request $request): RedirectResponse
105 {
106     $auctionItem = AuctionItem::find(id: $request->auction_id);
107
108     if ($auctionItem && $auctionItem->seller_id == Auth::id()) {
109         $auctionItem->title = $request->input(key: 'title');
110         $auctionItem->description = $request->input(key: 'description');
111         $auctionItem->starting_price = $request->input(key: 'starting_price');
112         $auctionItem->save();
113
114         return redirect()->back()->with(key: 'success', value: 'Auction item updated successfully!');
115     }
116
117     return redirect()->back()->with(key: 'error', value: 'Auction item not found or not authorized!');
118 }
119
120 1 reference | Overrides
121 public function createAuctionItem(): Factory\View
122 {
123     $categories = Category::all(); // Fetch all available categories
124     return view(view: 'seller.add-item', data: compact(var_name: 'categories'));
125 }
126
127 1 reference | Overrides
128 public function storeAuctionItem(Request $request): RedirectResponse
129 {
130     // Validate the form inputs
131     $request->validate(rules: [
132         'title' => 'required|string',
133         'description' => 'required|string',
134         'starting_price' => 'required|numeric',
135         'current_bid' => 'nullable|numeric',
136         'end_time' => 'required|date',
137         'categories' => 'required|array',
138         'image_paths.*' => 'nullable|image|max:5120', // Ensure image paths are valid strings
139     ]);
140
141     // Create the auction item
142     $auctionItem = AuctionItem::create(attributes: [
143         'title' => $request->title,
144         'description' => $request->description,
145         'starting_price' => $request->starting_price,
146         'current_bid' => $request->current_bid,
147         'end_time' => $request->end_time,
148         'seller_id' => auth()->id(), // Assign the current logged-in seller's ID
149     ]);

```

```

150     // Attach the categories to the auction item
151     $auctionItem->categories()->attach($request->categories);
152
153     // Handling image paths
154     if ($request->hasFile(key: 'images')) {
155         foreach ($request->file(key: 'images') as $image) {
156             if ($image) {
157                 // Store the image in the 'auction_images' folder in the 'public' disk
158                 $imagePath = $image->store(path: 'auction_images', options: public);
159
160                 // Save the image path to the database
161                 $image->create(attributes: [
162                     'url' => $imagePath, // Correct: use the path as returned by the store() method
163                     'auction_item_id' => $auctionItem->id,
164                 ]);
165             }
166         }
167     }
168
169     // Redirect the seller back to the dashboard with a success message
170     return redirect()->route(route: 'seller.dashboard')->with(key: 'success', value: 'Auction item added successfully!');
171 }

```

Figure: SellerDashboardController

```

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Models\AuctionItem;
6 use Illuminate\Support\Facades\Auth;
7
8 3 references | 0 implementations
9 class TransactionController extends Controller
10 {
11     1 reference | 0 overrides
12     public function index(): Factory|View
13     {
14         // Get the current logged-in user's ID
15         $userId = Auth::id();
16
17         // Get all auction items where the user was the highest bidder and the auction has ended
18         $wonItems = AuctionItem::whereHas('bidRecords', callback: function ($query) use ($userId): void {
19             $query->where('customer_id', $userId);
20         })
21         ->where(column: 'end time', operator: '<', value: now()) // Auction has ended
22         ->whereHas(relation: 'bidRecords', callback: function ($query): void {
23             $query->orderBy('amount', 'desc')->limit(1); // Ensure it fetches the highest bid
24         })
25         ->with(relations: 'seller', 'payment') // Fetch seller and payment status
26         ->get();
27
28         return view('transactions', data: compact('wonItems'));
29     }
30     1 reference | 0 overrides
31     public function confirmPurchase(Request $request, $id): RedirectResponse
32     {
33         // Find the auction item by its ID
34         $auctionItem = AuctionItem::findOrFail($id);
35
36         // Check if the user is the highest bidder
37         if (auth()->id() != $auctionItem->bidRecords()->orderBy('amount', 'desc')->first()->customer_id) {
38             return redirect()->back()->with(key: 'error', value: 'You are not authorized to confirm this purchase.');

```

Figure: TransactionController

```

1 <?php
2
3 namespace App\Http\Controllers\Admin;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\User;
7 use Illuminate\Http\Request;
8
9 0 references | 0 implementations
10 class UserController extends Controller
11 {
12     // Other methods...
13
14     0 references | 0 overrides
15     public function destroy($id): RedirectResponse
16     {
17         $user = User::findOrFail($id);
18         $user->delete();
19
20         return redirect()->route(route: 'admin.users')->with(key: 'success', value: 'User deleted successfully.');

```

Figure: UserController

```

1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Symfony\Component\HttpFoundation\Response;
8  use Illuminate\Support\Facades\Auth;
9
10  2 references | 0 implementations
11  class Admin
12  {
13      /**
14       * Handle an incoming request.
15       *
16       * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
17       */
18      0 references | 0 overrides
19      public function handle(Request $request, Closure $next): Response
20      {
21          // if user is not logged in then it will redirect to login page
22          if(!Auth::check()){
23              return redirect()->route(route: 'login');
24          }
25
26          $userRole=Auth::user()->role;
27
28          // Seller
29          if($userRole==1){
30              return redirect()->route(route: 'seller.dashboard');
31          }
32
33          // Admin
34          if($userRole==2){
35              return $next($request);
36          }
37
38          // Normal User
39          if($userRole==3){
40              return redirect()->route(route: 'dashboard');
41          }
42      }
43  }

```

Figure: Admin Middleware

```

<?php

namespace App\Http\Middleware;

use Closure;
use Illuminate\Http\Request;
use Symfony\Component\HttpFoundation\Response;
use Illuminate\Support\Facades\Auth;

/*
 * references: 0 implementations
 */
class NormalUser
{
    /**
     * Handle an incoming request.
     *
     * @param  \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response)  $next
     */
    public function handle(Request $request, Closure $next): Response
    {
        // if user is not logged in then it will redirect to login page
        if(!Auth::check()){
            return redirect()->route(route: 'login');
        }

        $userRole=Auth::user()->role;

        // Seller
        if($userRole==1){
            return redirect()->route(route: 'seller_dashboard');
        }

        // Admin
        if($userRole==2){
            return redirect()->route(route: 'admin_dashboard');
        }

        // Normal User
        if($userRole==3){
            return $next($request);
        }
    }
}

```

Figure: NormalUser Middleware

```

1  <?php
2
3  namespace App\Http\Middleware;
4
5  use Closure;
6  use Illuminate\Http\Request;
7  use Symfony\Component\HttpFoundation\Response;
8  use Illuminate\Support\Facades\Auth;
9
10  2 references | 0 implementations
11  class Seller
12  {
13      /**
14       * Handle an incoming request.
15       *
16       * @param \Closure(\Illuminate\Http\Request): (\Symfony\Component\HttpFoundation\Response) $next
17       */
18      0 references | 0 overrides
19      public function handle(Request $request, Closure $next): Response
20      {
21          // if user is not logged in then it will redirect to login page
22          if(!Auth::check()){
23              return redirect()->route(route: 'login');
24          }
25
26          $userRole=Auth::user()->role;
27
28          // Seller
29          if($userRole==1){
30              return $next($request);
31          }
32
33          // Admin
34          if($userRole==2){
35              return redirect()->route(route: 'admin.dashboard');
36          }
37
38          // Normal User
39          if($userRole==3){
40              return redirect()->route(route: 'dashboard');
41          }
42      }
43  }

```

Figure: Seller Middleware



```

1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 40 references | 0 implementations
9 class AuctionItem extends Model
10 {
11     0 references
12     protected $fillable = ['title', 'description', 'starting_price', 'current_bid', 'end_time', 'seller_id', 'status'];
13
14     0 references | 0 overrides
15     public function categories(): BelongsToMany
16     {
17         return $this->belongsToMany(related: Category::class, table: 'auction_item_category');
18     }
19
20     0 references | 0 overrides
21     public function seller(): BelongsTo
22     {
23         return $this->belongsTo(related: User::class, foreignKey: 'seller_id');
24     }
25
26     0 references | 0 overrides
27     public function images(): HasMany
28     {
29         return $this->hasMany(related: Image::class);
30     }
31
32     3 references | 0 overrides
33     public function bidRecords(): HasMany
34     {
35         return $this->hasMany(related: BidRecord::class, foreignKey: 'auction_id');
36     }
37
38     0 references | 0 overrides
39     public function getHighestBidAttribute(): mixed
40     {
41         return $this->bidRecords()->max(column: 'amount');
42     }
43
44     0 references | 0 overrides
45     public function isAuctionEnded(): bool
46     {
47         return now()->greaterThanOrEqualTo(date: $this->end_time);
48     }
49
50     0 references | 0 overrides
51     public function isHighestBidder(): bool
52     {
53         $highestBid = $this->bidRecords()->orderBy(column: 'amount', direction: 'desc')->first();
54         return $highestBid && $highestBid->user_id === auth()->id();
55     }
56
57     0 references | 0 overrides
58     public function payment(): HasOne
59     {
60         return $this->hasOne(related: Payment::class, foreignKey: 'auction_id');
61     }
62 }

```

Figure: AuctionItem Model

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class BidRecord extends Model
9  {
10     use HasFactory;
11
12     0 references
13     protected $fillable = ['auction_id', 'customer_id', 'amount'];
14
15     0 references | 0 overrides
16     public function bidder(): BelongsTo
17     {
18         return $this->belongsTo(related: User::class, foreignKey: 'customer_id');
19     }
20
21     0 references | 0 overrides
22     public function auctionItem(): BelongsTo
23     {
24         return $this->belongsTo(related: AuctionItem::class, foreignKey: 'auction_id');
25     }
26
27     0 references | 0 overrides
28     public function user(): BelongsTo
29     {
30         return $this->belongsTo(related: User::class);
31     }
32
33     // App/Models/BidRecord.php
34 }

```

Figure: BidRecord Model

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class Category extends Model
9  {
10     0 references
11     protected $fillable = ['name'];
12
13     0 references | 0 overrides
14     public function auctionItems(): BelongsToMany
15     {
16         return $this->belongsToMany(related: AuctionItem::class, table: 'auction_item_category');
17     }
18 }

```

Figure: Category Model

```

1  <?php
2
3  namespace App\Models;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  3 references | 0 implementations
9  class Image extends Model
10 {
11     0 references
12     protected $fillable = ['url', 'auction_item_id'];
13
14     0 references | 0 overrides
15     public function auctionItem(): BelongsTo
16     {
17         return $this->belongsTo(related: AuctionItem::class);
18     }
19 }

```

Figure: Image Model

```

<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

7 references | 0 implementations
class Payment extends Model
{
    use HasFactory;

    0 references
    protected $fillable = ['user_id', 'auction_id', 'amount', 'payment_method', 'token', 'status'];

    // In Payment.php model

    0 references | 0 overrides
    public function auctionItem(): BelongsTo
    {
        return $this->belongsTo(related: AuctionItem::class, foreignKey: 'auction_id');
    }

    0 references | 0 overrides
    public function user(): BelongsTo
    {
        return $this->belongsTo(related: User::class, foreignKey: 'user_id');
    }
}

```

Figure: Payment Model

```

1  <?php
2
3  namespace App\Models;
4
5  // use illuminate\Contracts\Auth\MustVerifyEmail;
6  use Illuminate\Database\Eloquent\Factories\HasFactory;
7  use Illuminate\Foundation\Auth\User as Authenticatable;
8  use Illuminate\Notifications\Notifiable;
9  use App\Models\AuctionItem;
10
11  53 references|0 implementations
12  class User extends Authenticatable
13  {
14      use HasFactory, Notifiable;
15
16      /**
17       * The attributes that are mass assignable.
18       *
19       * @var array<int, string>
20       */
21      0 references
22      protected $fillable = [
23          'name',
24          'username',
25          'email',
26          'password',
27          'role',
28          'google_id',
29          'avatar',
30      ];
31
32      /**
33       * The attributes that should be hidden for serialization.
34       *
35       * @var array<int, string>
36       */
37      0 references
38      protected $hidden = [
39          'password',
40          'remember_token',
41      ];
42
43      /**
44       * Get the attributes that should be cast.
45       *
46       * @return array<string, string>
47       */
48      0 references|0 overrides
49      protected function casts(): array
50      {
51          return [
52              'email_verified_at' => 'datetime',
53              'password' => 'hashed',
54          ];
55      }
56
57      0 references|0 overrides
58      public function auctionItems(): \Illuminate\Database\Eloquent\Relations\HasMany
59      {
60          return $this->hasMany(AuctionItem::class, 'seller_id');
61      }
62  }

```

Figure : User Model

## Database

phpMyAdmin

Recent

Favorites

New

information\_schema

mysql

ntam

ntam2

New

audion\_items

audion\_item\_category

bid\_records

cache

cache\_locks

categories

failed\_jobs

images

jobs

job\_batches

migrations

password\_reset\_tokens

payments

sessions

users

performance\_schema

phpmyadmin

test

Search: 122410.0.0.0 Database: web2020 Table: auction\_items

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Tracking

Triggers

Table structure

Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT	<div>Change Drop More</div>
2	title	varchar(255)	utf8mb4_unicode_ci		No	None			<div>Change Drop More</div>
3	description	text	utf8mb4_unicode_ci		No	None			<div>Change Drop More</div>
4	starting_price	decimal(14,2)			No	None			<div>Change Drop More</div>
5	current_bid	decimal(14,2)		Yes	NULL				<div>Change Drop More</div>
6	end_time	datetime		No	None				<div>Change Drop More</div>
7	seller_id	bigint(20)		UNSIGNED	No	None			<div>Change Drop More</div>
8	created_at	timestamp		Yes	NULL				<div>Change Drop More</div>
9	updated_at	timestamp		Yes	NULL				<div>Change Drop More</div>
10	status	tinyint(3)		UNSIGNED	No	3	1: Sold   2: Approved   3: Waiting		<div>Change Drop More</div>

☐ Check all

With selected:

Browse

Change

Drop

Primary

Unique

Index

Spatial

Fulltext

Add to central columns

Remove from central columns

Print

Propose table structure

Track table

Move columns

Normalize

Add

1

column(s)

after status

Go

Indexes

Action	Keyname	Type	Unique	Packed	Column	Cardinality	Collation	Null	Comment
<div>Edit Rename Drop</div>	PRIMARY	BTREE	Yes	No	id	15	A	No	
<div>Edit Rename Drop</div>	audion_items_seller_id_foreign	BTREE	No	No	seller_id	5	A	No	

Save options

id

title

description

starting\_price

current\_bid

end\_time

seller\_id

created\_at

updated\_at

status

Edit

Copy

Delete

19

Toyota LandCruiser 300 (LC300)

The 2022 Toyota Land Cruiser 300 armored all-wheel...

5000000.00

5000000.00

2024-09-21 15:30:00

3

2024-09-20 13:45:47

2024-09-20 21:32:38

2

Edit

Copy

Delete

20

House (10 Photos)

House with 10,000 Sqft 4 bedroom 4 bathroom All...

25000000.00

2024-09-21 15:40:00

3

2024-09-20 13:46:05

2024-09-20 13:53:45

2

Edit

Copy

Delete

21

See painting

Acrylic painting Canvas 30x40

6130.00

2024-09-20 19:49:00

3

2024-09-20 13:49:34

2024-09-20 13:53:45

2

Edit

Copy

Delete

22

2003 Toyota Spacia Oldtime Fresh Condition

The Spacia comes with the cost of reliability that...

820000.00

2024-09-20 19:49:00

3

2024-09-20 13:50:29

2024-09-20 13:53:47

2

Edit

Copy

Delete

24

iPad Air 5

iPad Air 5th Generation M1 64 GB iPad OS 17 0.2 ...

55500.00

2024-09-27 20:14:00

6

2024-09-20 14:16:50

2024-09-20 14:21:33

2

Edit

Copy

Delete

25

2024 Brand New- 2024 Model Brand A Grade Mercedes

★ For Sale ★ Exclusive Offer- 2024 Mercedes-AMG ...

17820000.00

2024-09-27 20:19:00

6

2024-09-20 14:19:20

2024-09-20 14:21:35

2

Edit

Copy

Delete

27

Guthman 1 FLAT SALE 3327 SQFT

Luxurious Home Auction in Guthman, Dhaka - Your Dr...

4500000.00

2024-09-22 20:23:00

6

2024-09-20 14:30:44

2024-09-20 14:32:15

2

Edit

Copy

Delete

28

Flat for sell at Bashundhara

Golden Opportunity! Get a discount in this crisis in...

1100000.00

1100000.00

2024-09-27 20:31:00

6

2024-09-20 14:31:31

2024-09-20 15:10:47

2

Edit

Copy

Delete

29

H-TECH Spidee Adrenal (Full Set)

SIZE & FIT L-1420 x W- 910 x H- 980 Color Antiqu...

118500.00

2024-09-25 20:31:00

6

2024-09-20 14:32:00

2024-09-20 14:32:18

2

Edit

Copy

Delete

31

IELTS Book

IELTS Academic Book List 1 Cambridge Ielts Academ...

1600.00

2024-09-27 23:03:00

6

2024-09-20 23:04:11

2024-09-20 23:06:29

2

Edit

Copy

Delete

32

Logitech G902 Mouse

An exclusive gaming mouse

6000.00

6001.00

2024-09-25 23:04:00

6

2024-09-20 23:04:53

2024-09-21 07:10:29

2

Edit

Copy

Delete

33

Casio FX991EX

Casio Calculator

1600.00

2024-09-20 23:06:00

6

2024-09-20 23:06:05

2024-09-20 23:40:12

2

Edit

Copy

Delete

35

neo

cow

1600.00

2500.00

2024-09-20 23:45:00

3

2024-09-20 23:42:55

2024-09-20 23:43:35

1

Edit

Copy

Delete

36

Cosmos Vase Pack 3, White Modern flat Vase, Ceram...

Vase

2000.00

1980.00

2024-09-21 02:52:00

3

2024-09-21 02:36:14

2024-09-21 02:52:10

1

Edit

Copy

Delete

38

Casio MTP-B1450-SAVDF Quartz Analog Dial Stainless...

Let's meet up with our brand new and latest collec...

8000.00

89000.00

2024-09-21 07:28:00

3

2024-09-21 07:26:39

2024-09-21 07:29:40

1

Check all

With selected:

Edit

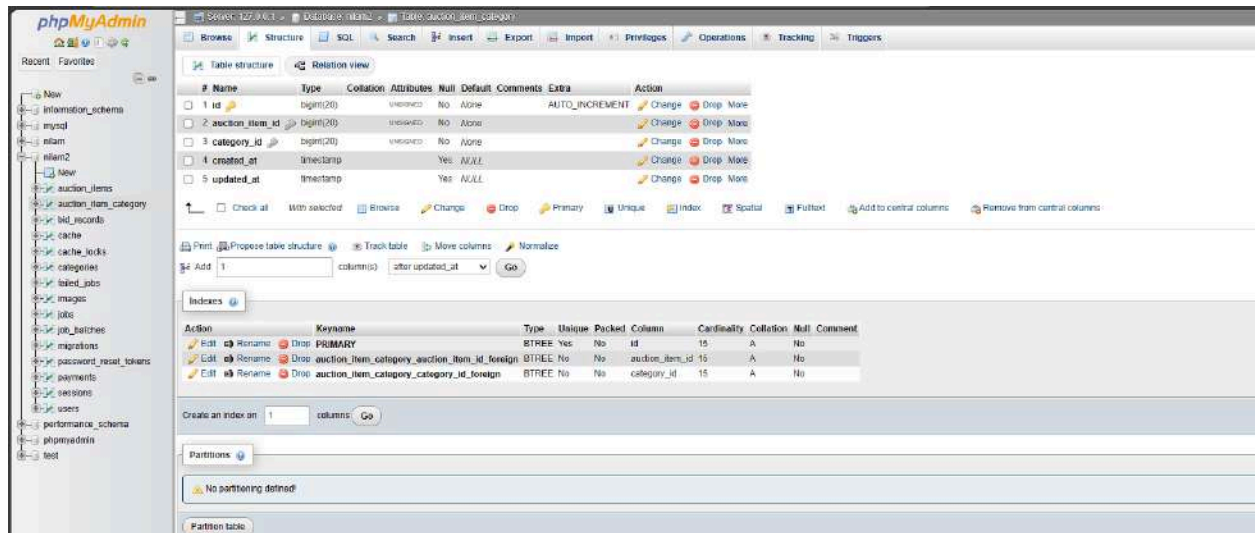
Copy

Delete

Export

Figure: Auction Items Table



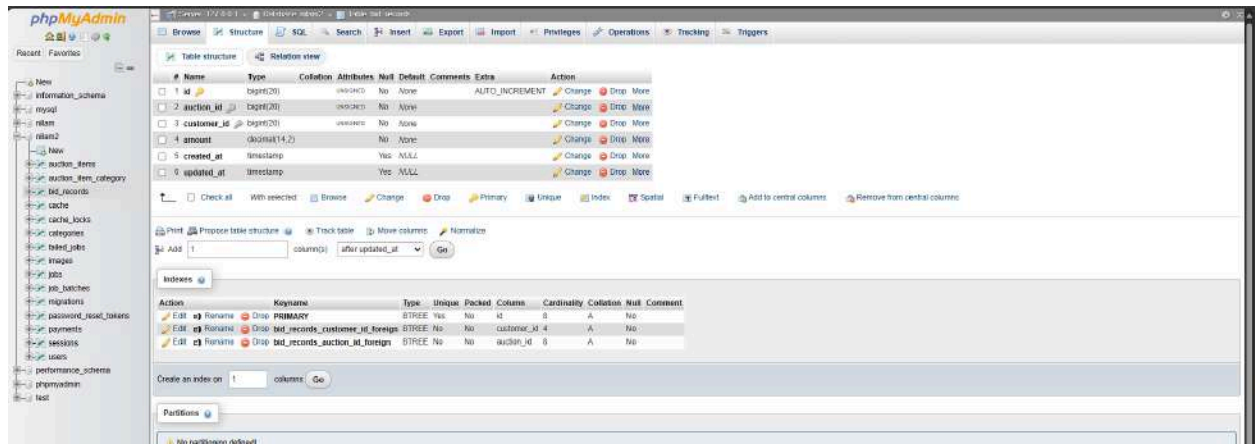


Extra options

					id	auction_item_id	category_id	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete		19	19	1	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		20	20	9	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		21	21	3	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		22	22	1	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		24	24	11	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		25	25	1	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		27	27	9	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		28	28	9	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		29	29	7	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		31	31	2	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		32	32	11	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		33	33	11	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		35	35	3	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		36	36	4	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete		39	39	12	NULL	NULL

☐ Check all    With selected:    Edit    Copy    Delete    Export

Figure: Auction Item Category Table

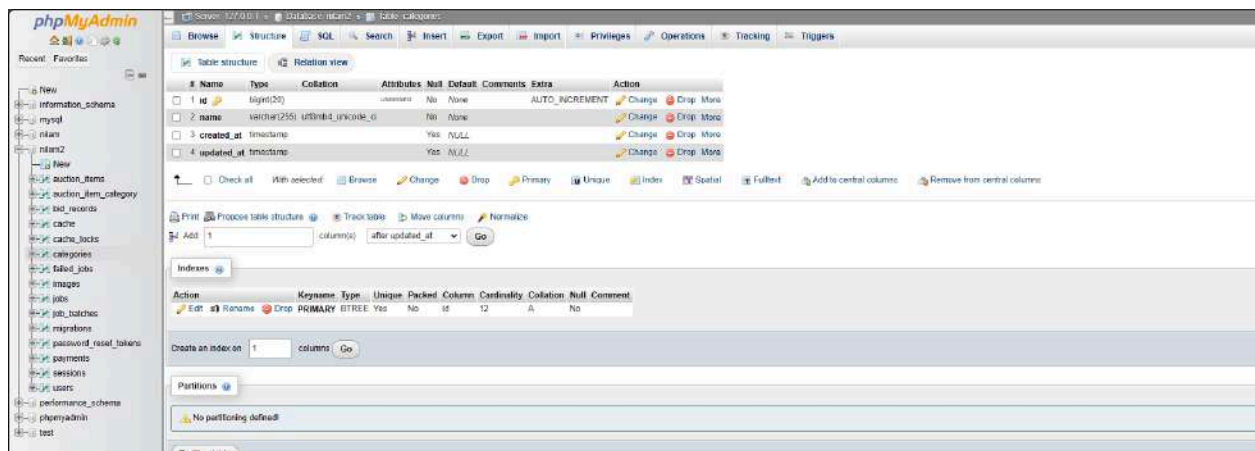


Extra options

	id	auction_id	customer_id	amount	created_at	updated_at
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	28	5	1100002.00	2024-09-20 15:01:44	2024-09-20 15:01:44
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	2	28	5	1100003.00	2024-09-20 15:18:47	2024-09-20 15:18:47
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	3	19	5	5000001.00	2024-09-20 21:32:38	2024-09-20 21:32:38
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	6	35	4	1800.00	2024-09-20 23:43:18	2024-09-20 23:43:18
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	7	35	5	2500.00	2024-09-20 23:43:35	2024-09-20 23:43:35
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	8	36	5	10020.00	2024-09-21 02:50:22	2024-09-21 02:50:22
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	9	32	5	6001.00	2024-09-21 07:16:29	2024-09-21 07:16:29
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	10	39	5	90000.00	2024-09-21 07:27:08	2024-09-21 07:27:08

☐ Check all
 With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

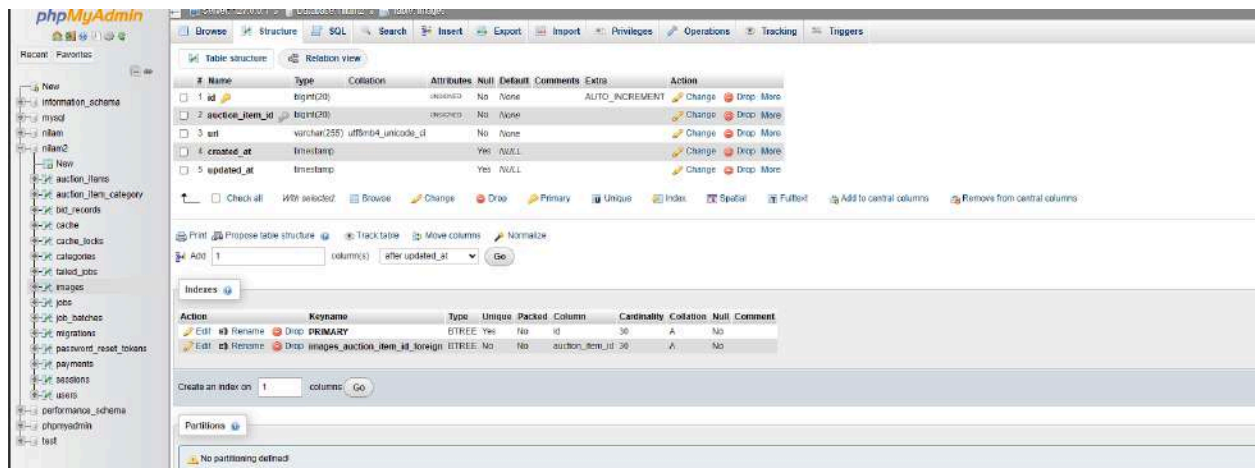
Figure: Bid Records Table



				id	name	created_at	updated_at
<input type="checkbox"/>	Edit	Copy	Delete	1	Vehicles	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	2	Book	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	3	Painting	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	4	Sculpture	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	5	Gemstone	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	6	Jewelery	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	7	Furniture	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	8	Collectibles	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	9	Real Estate	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	10	Cloth	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	11	Technology	NULL	NULL
<input type="checkbox"/>	Edit	Copy	Delete	12	Watch	NULL	NULL

Figure: Category Table





Extra options

				id	auction_item_id	url	created_at	updated_at
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	38	19	2024-09-20 13:45:47
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	39	19	2024-09-20 13:45:47
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	40	19	2024-09-20 13:45:47
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	41	20	2024-09-20 13:49:05
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	42	20	2024-09-20 13:49:05
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	43	20	2024-09-20 13:49:05
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	44	21	2024-09-20 13:49:34
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	45	22	2024-09-20 13:50:29
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	47	24	2024-09-20 14:18:50
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	48	24	2024-09-20 14:18:50
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	49	24	2024-09-20 14:18:50
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	50	25	2024-09-20 14:19:20
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	51	25	2024-09-20 14:19:20
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	52	25	2024-09-20 14:19:20
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	56	27	2024-09-20 14:30:44
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	57	27	2024-09-20 14:30:44
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	58	27	2024-09-20 14:30:44
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	59	20	2024-09-20 14:31:31
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	60	28	2024-09-20 14:31:31
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	61	28	2024-09-20 14:31:31
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	62	29	2024-09-20 14:32:09
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	63	29	2024-09-20 14:32:09
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	66	31	2024-09-20 23:04:12
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	67	32	2024-09-20 23:04:53
<input type="checkbox"/>	Edit	<input type="checkbox"/>	Copy	<input type="checkbox"/>	Delete	68	32	2024-09-20 23:04:53

Figure: Images Table



## **Technology (Framework, Language)**

**Framework:** Laravel, Bootstrap

**Language:** PHP

**Package:** Socialite, Laravel breeze

## **GitHub Repository**

Link: [RejwanShafi/nilam-an-online-auction-system \(github.com\)](https://github.com/RejwanShafi/nilam-an-online-auction-system)

### Individual Contribution

<b>ID</b>	<b>Name</b>	<b>Contribution</b>
<b>20101232</b>	<b>Sopnil Roy Niloy</b>	<ol style="list-style-type: none"> <li>1. Social Media Authentication</li> <li>2. Bidding and Buy option</li> <li>3. Edit the product details that are waiting for approval [Seller]</li> <li>4. Activity Diagram, Use Case Diagram</li> </ol>
<b>21301425</b>	<b>Afif Rayhan Pranto</b>	<ol style="list-style-type: none"> <li>1. Admin Dashboard Add Image of Product</li> <li>2. Search</li> <li>3. User Management Admin</li> <li>4. Component Diagram</li> </ol>
<b>23241108</b>	<b>Rejwan Shafi</b>	<ol style="list-style-type: none"> <li>1. Database Integration,</li> <li>2. Home Page [Frontend]</li> <li>3. Category Based Filtering</li> <li>4. Payment Approval</li> <li>5. Approve Listed auction item</li> <li>6. User profile picture and address update</li> <li>7. Show transaction history for users</li> <li>8. Show Active bidding status</li> <li>9. Sold Auction Items view [Seller]</li> <li>10. EER Diagram</li> </ol>
<b>24141198</b>	<b>Homairah Ferdousia</b>	<ol style="list-style-type: none"> <li>1. Login and Register</li> <li>2. Seller Dashboard</li> <li>3. Seller Add Auction Items [Frontend, Backend]</li> <li>4. Show all Available Auction Items</li> <li>5. Show products that are waiting for admin approval [seller]</li> <li>6. Auction Item Details</li> <li>7. User Password Update</li> <li>8. Data Flow Diagram</li> </ol>

