**Controller  
  
CustomerController.cs**

using BankLoanProject.Data;

using BankLoanProject.Models;

using BankLoanProject.Models.Entities;

using BankLoanProject.Services.Interface;

using Microsoft.AspNetCore.Mvc;

using Microsoft.AspNetCore.Mvc.Rendering;

using Microsoft.EntityFrameworkCore;

using Microsoft.Data.SqlClient; // For SqlException

using Microsoft.EntityFrameworkCore;

//using BankLoanProject.Filters;

namespace BankLoanProject.Controllers

{

public class CustomerController : Controller

{

private readonly BankLoanDbContext \_context;

private readonly IWebHostEnvironment \_env;

private readonly ICustomerService \_customerService;

public CustomerController(BankLoanDbContext context, ICustomerService customerService, IWebHostEnvironment env)

{

\_context = context;

\_customerService = customerService;

\_env = env;

}

private void PopulateIdentityTypes()

{

ViewBag.IdentityTypes = new SelectList(new[]

{

"PAN Card",

"Passport",

"Driving License",

"Aadhar Card"

});

}

public IActionResult CustomerRegistration()

{

PopulateIdentityTypes();

return View();

}

public IActionResult CustomerProfile()

{

return View();

}

[HttpPost]

public async Task<IActionResult> CustomerRegistration(CustomerViewModel model)

{

PopulateIdentityTypes();

if (ModelState.IsValid)

{

// ✅ Check for duplicate phone number BEFORE saving

var existingPhone = await \_context.Customers.AnyAsync(c => c.Phone == model.Phone);

if (existingPhone)

{

ViewBag.Error = "Phone number already exists. Please use a different number.";

return View(model);

}

try

{

var folderPath = Path.Combine(\_env.WebRootPath, "uploads", model.Email);

Console.WriteLine($"Creating directory at: {folderPath}");

Directory.CreateDirectory(folderPath);

var filePath = Path.Combine(folderPath, model.IdentityProof.FileName);

Console.WriteLine($"Saving file to: {filePath}");

using (var stream = new FileStream(filePath, FileMode.Create))

{

await model.IdentityProof.CopyToAsync(stream);

Console.WriteLine("File uploaded successfully.");

}

var customer = new Customer

{

Name = model.Name,

Email = model.Email,

Phone = model.Phone,

Address = model.Address,

KycStatus = "PENDING",

Password = model.Password,

Role = "Customer",

DateOfBirth = model.DateOfBirth,

DocumentPath = $"/uploads/{model.Email}/{model.IdentityProof.FileName}",

IdentityType = model.IdentityType,

};

\_context.Customers.Add(customer);

Console.WriteLine("Customer added to context.");

await \_context.SaveChangesAsync();

Console.WriteLine("Changes saved to database.");

ModelState.Clear();

ViewBag.Success = "Customer registered successfully!";

return RedirectToAction("Hero", "Account");

}

catch (DbUpdateException dbEx)

{

Console.WriteLine($"Database update error: {dbEx.InnerException?.Message ?? dbEx.Message}");

}

catch (SqlException sqlEx)

{

Console.WriteLine($"SQL error: {sqlEx.Message}");

ViewBag.Error = "A SQL error occurred during registration.";

}

catch (Exception ex)

{

Console.WriteLine($"General exception: {ex.Message}");

ViewBag.Error = "An unexpected error occurred during registration.";

}

}

else

{

foreach (var state in ModelState)

{

foreach (var error in state.Value.Errors)

{

Console.WriteLine($"Key: {state.Key}, Error: {error.ErrorMessage}");

}

}

}

Console.WriteLine("Model state is invalid.");

ViewBag.Error = "Please fill all fields and upload a document.";

return View(model);

}

public async Task<IActionResult> CustomerTable()

{

var customers = await \_context.Customers.ToListAsync();

return View(customers); // Pass as model, not ViewBag

}

//[SessionAuthorize("Admin")]

[HttpGet]

public async Task<IActionResult> Edit(int id)

{

var customer = await \_customerService.GetByIdAsync(id);

if (customer == null)

{

return NotFound();

}

var model = new EditCustomerViewModel

{

CustomerId = customer.CustomerId,

Name = customer.Name,

Email = customer.Email,

Phone = customer.Phone,

DateOfBirth = customer.DateOfBirth,

Address = customer.Address,

IdentityType = customer.IdentityType,

KycStatus = customer.KycStatus

};

return View(model);

}

//[SessionAuthorize("Admin")]

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Edit(EditCustomerViewModel model)

{

if (ModelState.IsValid)

{

var existingCustomer = await \_context.Customers.FindAsync(model.CustomerId);

if (existingCustomer == null)

{

return NotFound();

}

// ✅ Now update fields

existingCustomer.Name = model.Name;

existingCustomer.Email = model.Email;

existingCustomer.Phone = model.Phone;

existingCustomer.DateOfBirth = model.DateOfBirth;

existingCustomer.Address = model.Address;

existingCustomer.IdentityType = model.IdentityType;

existingCustomer.KycStatus = model.KycStatus;

await \_context.SaveChangesAsync();

return RedirectToAction("CustomerTable");

}

else

{

foreach (var state in ModelState)

{

foreach (var error in state.Value.Errors)

{

Console.WriteLine($"Key: {state.Key}, Error: {error.ErrorMessage}");

}

}

}

return View(model);

}

//[SessionAuthorize("Admin")]

[HttpPost]

[ValidateAntiForgeryToken]

public async Task<IActionResult> Delete(int id)

{

await \_customerService.DeleteAsync(id);

return RedirectToAction("CustomerTable");

}

//[SessionAuthorize("Admin")]

[HttpPost]

public async Task<IActionResult> Verify(int id)

{

var customer = await \_context.Customers.FindAsync(id);

if (customer == null)

{

return NotFound();

}

customer.KycStatus = "Approved";

await \_context.SaveChangesAsync();

return RedirectToAction("CustomerTable");

}

}

}

**Models**

**CustomerViewModels.cs**using System.ComponentModel.DataAnnotations;

using Microsoft.AspNetCore.Mvc.ModelBinding;

namespace BankLoanProject.Models

{

public class CustomerViewModel

{

[Required]

public string Name { get; set; }

[Required, EmailAddress]

public string Email { get; set; }

[Required]

public string Phone { get; set; }

[Required]

public DateTime DateOfBirth { get; set; }

[Required]

public string Address { get; set; }

[Required]

public string IdentityType { get; set; }

[Required]

[DataType(DataType.Password)]

public string Password { get; set; }

[Required]

[Display(Name = "Identity Proof")]

public IFormFile IdentityProof { get; set; }

}

}

**EditCustomerViewModel.cs**using System.ComponentModel.DataAnnotations;

namespace BankLoanProject.Models

{

public class EditCustomerViewModel

{

public int CustomerId { get; set; }

[Required]

public string Name { get; set; }

[Required, EmailAddress]

public string Email { get; set; }

[Required]

public string Phone { get; set; }

[Required]

public DateTime DateOfBirth { get; set; }

[Required]

public string Address { get; set; }

[Required]

public string IdentityType { get; set; }

[Required]

public string KycStatus { get; set; }

}

}

**Services   
  
Interface   
  
ICustomerService.cs**using BankLoanProject.Models.Entities;

namespace BankLoanProject.Services.Interface

{

public interface ICustomerService

{

Task<List<Customer>> GetAllAsync();

Task<Customer?> GetByIdAsync(int id);

Task AddAsync(Customer customer);

Task UpdateAsync(Customer customer);

Task DeleteAsync(int id);

}

}

**Module Services**

**CustomerService.cs**

using BankLoanProject.Data;

using BankLoanProject.Models.Entities;

using BankLoanProject.Services.Interface;

using Microsoft.EntityFrameworkCore;

namespace BankLoanProject.Services.ModuleService

{

public class CustomerService:ICustomerService

{

private readonly BankLoanDbContext \_context;

public CustomerService(BankLoanDbContext context)

{

\_context = context;

}

public async Task<List<Customer>> GetAllAsync() => await \_context.Customers.ToListAsync();

public async Task<Customer?> GetByIdAsync(int id) => await \_context.Customers.FindAsync(id);

public async Task AddAsync(Customer customer)

{

\_context.Customers.Add(customer);

await \_context.SaveChangesAsync();

}

public async Task UpdateAsync(Customer customer)

{

\_context.Customers.Update(customer);

await \_context.SaveChangesAsync();

}

public async Task DeleteAsync(int id)

{

var customer = await \_context.Customers.FindAsync(id);

if (customer != null)

{

\_context.Customers.Remove(customer);

await \_context.SaveChangesAsync();

}

}

}

}

**Views   
  
Customer  
  
CustomerRegistration.cshtml**

@model BankLoanProject.Models.CustomerViewModel

@{

ViewBag.Title = "Customer Registration";

Layout = null;

var maxDate = DateTime.Today.AddYears(-18).ToString("yyyy-MM-dd");

}

<div class="main-container">

@if (ViewBag.Success != null)

{

<div class="alert alert-success" id="successAlert">

<i class="fas fa-check-circle"></i>

@ViewBag.Success

</div>

}

@if (ViewBag.Error != null)

{

<div class="alert alert-danger" id="errorAlert">

<i class="fas fa-exclamation-triangle"></i>

@ViewBag.Error

</div>

}

<link rel="stylesheet" href="~/css/register.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css" />

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

<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

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

<!-- Floating background shapes -->

<div class="bg-shape"></div>

<div class="bg-shape"></div>

<div class="bg-shape"></div>

<div class="bg-shape"></div>

@\* <div class="d-flex justify-content-start mb-auto"> \*@

@\* <button class="btn btn-outline-secondary" onclick="goBack()"> \*@

@\* ← Back \*@

@\* </button> \*@

@\* </div> \*@

<div class="registration-card">

<div class="form-header">

<h1 class="form-title">

<i class="fas fa-user-plus me-3"></i>Customer Registration

</h1>

<p class="form-subtitle">Join our platform and start your journey with us</p>

</div>

<form asp-action="CustomerRegistration" asp-controller="Customer" method="post" enctype="multipart/form-data" id="registrationForm" novalidate>

<!-- Personal Information Section -->

<div class="section-title">

<i class="fas fa-user"></i>

Personal Information

</div>

<div class="row">

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

<div class="form-group">

<label asp-for="Name" class="form-label">

<i class="fas fa-user"></i>

Full Name

</label>

<input asp-for="Name" class="form-control" placeholder="Enter your full name" required />

<span asp-validation-for="Name" class="text-danger"></span>

<div class="error-message" id="fullNameError">

<i class="fas fa-exclamation-triangle"></i>

Please enter a valid full name (only letters and spaces allowed)!

</div>

</div>

</div>

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

<div class="form-group">

<label asp-for="Email" class="form-label">

<i class="fas fa-envelope"></i>

Email Address

</label>

<input asp-for="Email" class="form-control" placeholder="Enter your email address" required />

<span asp-validation-for="Email" class="text-danger"></span>

<div class="error-message" id="emailError">

<i class="fas fa-exclamation-triangle"></i>

Please enter a valid email address!

</div>

</div>

</div>

</div>

<div class="row">

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

<div class="form-group">

<label asp-for="Phone" class="form-label">

<i class="fas fa-phone"></i>

Phone Number

</label>

<input asp-for="Phone" class="form-control" placeholder="Enter your 10-digit phone number" required />

<span asp-validation-for="Phone" class="text-danger"></span>

<div class="error-message" id="phoneError">

<i class="fas fa-exclamation-triangle"></i>

Please enter exactly 10 digits!

</div>

</div>

</div>

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

<div class="form-group">

<label asp-for="DateOfBirth" class="form-label">

<i class="fas fa-calendar"></i>

Date of Birth

</label>

<input asp-for="DateOfBirth" class="form-control" type="date" required onkeydown="return false;" max="@maxDate" />

<span asp-validation-for="DateOfBirth" class="text-danger"></span>

<div class="error-message" id="dobError">

<i class="fas fa-exclamation-triangle"></i>

Please select your date of birth!

</div>

</div>

</div>

</div>

<div class="form-group">

<label asp-for="Address" class="form-label">

<i class="fas fa-map-marker-alt"></i>

Complete Address

</label>

<textarea asp-for="Address" class="form-control" rows="3" placeholder="Enter your complete address" required></textarea>

<span asp-validation-for="Address" class="text-danger"></span>

<div class="error-message" id="addressError">

<i class="fas fa-exclamation-triangle"></i>

Please enter your complete address!

</div>

</div>

<!-- Document Upload Section -->

<div class="section-title">

<i class="fas fa-file-upload"></i>

Document Upload

</div>

<div class="row">

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

<div class="form-group">

<label asp-for="IdentityType" class="form-label">

<i class="fas fa-id-card"></i>

Identity Document Type

</label>

<select asp-for="IdentityType" class="form-select" asp-items="ViewBag.IdentityTypes" required>

<option value="">Select Identity Document</option>

</select>

<span asp-validation-for="IdentityType" class="text-danger"></span>

<div class="error-message" id="identityTypeError">

<i class="fas fa-exclamation-triangle"></i>

Please select an identity document type!

</div>

</div>

</div>

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

<div class="form-group">

<label asp-for="IdentityProof" class="form-label">

<i class="fas fa-upload"></i>

Upload Identity Proof

</label>

<div class="file-upload-wrapper">

<label class="file-upload-btn" for="IdentityProof">

<i class="fas fa-cloud-upload-alt"></i>

<span id="identityProofText">Choose Identity Document</span>

</label>

<input asp-for="IdentityProof" type="file" id="IdentityProof" accept=".pdf,.jpg,.jpeg,.png" required style="display: none;" />

</div>

<div class="file-name" id="identityFileName"></div>

<span asp-validation-for="IdentityProof" class="text-danger"></span>

<div class="error-message" id="fileError">

<i class="fas fa-exclamation-triangle"></i>

Please upload a valid document (.pdf, .png, .jpeg only)!

</div>

</div>

</div>

</div>

<!-- Account Security Section -->

<div class="section-title">

<i class="fas fa-shield-alt"></i>

Account Security

</div>

<div class="row">

<div class="col-md-12">

<div class="form-group">

<label asp-for="Password" class="form-label">

<i class="fas fa-lock"></i> Create Password

</label>

<input asp-for="Password" id="Password" class="form-control" type="password" placeholder="Create a strong password (8-10 characters)" required />

<span asp-validation-for="Password" class="text-danger"></span>

<div class="error-message" id="passwordError">

<i class="fas fa-exclamation-triangle"></i> Password must be 8 to 10 characters long.

</div>

</div>

</div>

</div>

<div class="row">

<div class="col-md-12">

<div class="form-group">

<label for="ConfirmPassword" class="form-label">

<i class="fas fa-lock"></i> Confirm Password

</label>

<input id="ConfirmPassword" name="ConfirmPassword" class="form-control" type="password" placeholder="Re-enter your password" required />

<div class="error-message" id="confirmPasswordError">

<i class="fas fa-exclamation-triangle"></i> Passwords do not match!

</div>

</div>

</div>

</div>

<button type="submit" class="submit-btn" id="submitBtn">

<div class="loading-spinner" id="loadingSpinner"></div>

<i class="fas fa-user-plus"></i>

<span id="submitText">Create Account</span>

</button>

<div class="success-message" id="confirmationMessage">

<i class="fas fa-check-circle"></i>

Registration Successful! Welcome aboard!

</div>

</form>

</div>

</div>

<script>

$(document).ready(function() {

let formSubmitted = false;

// Auto-hide success/error alerts after 5 seconds

setTimeout(function() {

$('#successAlert, #errorAlert').fadeOut('slow');

}, 5000);

// Prevent form resubmission on page reload

if (window.history.replaceState) {

window.history.replaceState(null, null, window.location.href);

}

// File upload handling with validation

$('#IdentityProof').on('change', function() {

const file = this.files[0];

const fileName = file ? file.name : '';

if (fileName) {

// Check file extension

const allowedExtensions = ['.pdf', '.png', '.jpeg', '.jpg'];

const fileExtension = fileName.toLowerCase().substring(fileName.lastIndexOf('.'));

if (allowedExtensions.includes(fileExtension)) {

$('#identityProofText').text('Document Selected');

$('#identityFileName').text(fileName.length > 40 ? fileName.substring(0, 40) + '...' : fileName).show();

$('#fileError').hide();

$(this).removeClass('is-invalid');

} else {

$('#identityProofText').text('Choose Identity Document');

$('#identityFileName').hide();

$('#fileError').show();

$(this).addClass('is-invalid');

this.value = ''; // Clear the invalid file

}

} else {

$('#identityProofText').text('Choose Identity Document');

$('#identityFileName').hide();

}

});

// Real-time validation for all fields

$('#registrationForm input, #registrationForm select, #registrationForm textarea').on('input change blur', function() {

validateField($(this));

});

// Real-time password confirmation validation

$('#ConfirmPassword').on('input', function() {

const password = $('#Password').val();

const confirmPassword = $(this).val();

if (confirmPassword && password !== confirmPassword) {

$(this).addClass('is-invalid');

$('#confirmPasswordError').show();

} else {

$(this).removeClass('is-invalid');

$('#confirmPasswordError').hide();

}

});

// Form submission handling

$('#registrationForm').on('submit', function(e) {

// Prevent double submission

if (formSubmitted) {

e.preventDefault();

return false;

}

// Validate form

if (!validateForm()) {

e.preventDefault();

return false;

}

// Mark form as submitted

formSubmitted = true;

// Show loading state

const submitBtn = $('#submitBtn');

const spinner = $('#loadingSpinner');

const submitText = $('#submitText');

submitBtn.prop('disabled', true);

spinner.show();

submitText.text('Creating Account...');

// Let the form submit normally to maintain backend functionality

return true;

});

// Individual field validation function

function validateField($field) {

const fieldId = $field.attr('id');

const fieldValue = $field.val().trim();

let isValid = true;

// Clear previous error state

$field.removeClass('is-invalid');

$field.closest('.form-group').find('.error-message').hide();

switch (fieldId) {

case 'Name':

// Only letters and spaces allowed

const nameRegex = /^[a-zA-Z\s]+$/;

if (!fieldValue || !nameRegex.test(fieldValue)) {

$field.addClass('is-invalid');

$('#fullNameError').show();

isValid = false;

}

break;

case 'Email':

// Standard email validation

const emailRegex = /^[^\s]+[^\s]+\.[^\s]+$/;

if (!fieldValue || !emailRegex.test(fieldValue)) {

$field.addClass('is-invalid');

$('#emailError').show();

isValid = false;

}

break;

case 'Phone':

// Exactly 10 digits

const phoneRegex = /^\d{10}$/;

if (!fieldValue || !phoneRegex.test(fieldValue)) {

$field.addClass('is-invalid');

$('#phoneError').show();

isValid = false;

}

break;

case 'Password':

// 8-10 characters

if (!fieldValue || fieldValue.length < 8 || fieldValue.length > 10) {

$field.addClass('is-invalid');

$('#passwordError').show();

isValid = false;

}

break;

}

return isValid;

}

// Complete form validation function

function validateForm() {

let isValid = true;

// Clear previous errors

$('.error-message').hide();

$('.form-control, .form-select').removeClass('is-invalid');

// Validate Full Name - only letters and spaces

const fullName = $('[name="Name"]').val().trim();

const nameRegex = /^[a-zA-Z\s]+$/;

if (!fullName || !nameRegex.test(fullName)) {

$('[name="Name"]').addClass('is-invalid');

$('#fullNameError').show();

isValid = false;

}

// Validate Email - proper email format

const email = $('[name="Email"]').val().trim();

const emailRegex = /^[^\s]+[^\s]+\.[^\s]+$/;

if (!email || !emailRegex.test(email)) {

$('[name="Email"]').addClass('is-invalid');

$('#emailError').show();

isValid = false;

}

// Validate Phone - exactly 10 digits

const phone = $('[name="Phone"]').val().trim();

const phoneRegex = /^\d{10}$/;

if (!phone || !phoneRegex.test(phone)) {

$('[name="Phone"]').addClass('is-invalid');

$('#phoneError').show();

isValid = false;

}

// Validate Date of Birth

const dob = $('[name="DateOfBirth"]').val();

if (!dob) {

$('[name="DateOfBirth"]').addClass('is-invalid');

$('#dobError').show();

isValid = false;

}

// Validate Address

const address = $('[name="Address"]').val().trim();

if (!address) {

$('[name="Address"]').addClass('is-invalid');

$('#addressError').show();

isValid = false;

}

// Validate Identity Type

const identityType = $('[name="IdentityType"]').val();

if (!identityType) {

$('[name="IdentityType"]').addClass('is-invalid');

$('#identityTypeError').show();

isValid = false;

}

// Validate File Upload - check if file exists and has correct extension

const fileInput = $('#IdentityProof')[0];

if (!fileInput.files || fileInput.files.length === 0) {

$('#IdentityProof').addClass('is-invalid');

$('#fileError').show();

isValid = false;

} else {

const fileName = fileInput.files[0].name.toLowerCase();

const allowedExtensions = ['.pdf', '.png', '.jpeg', '.jpg'];

const fileExtension = fileName.substring(fileName.lastIndexOf('.'));

if (!allowedExtensions.includes(fileExtension)) {

$('#IdentityProof').addClass('is-invalid');

$('#fileError').show();

isValid = false;

}

}

// Validate Password - 8 to 10 characters

const password = $('#Password').val();

if (!password || password.length < 8 || password.length > 10) {

$('#Password').addClass('is-invalid');

$('#passwordError').show();

isValid = false;

}

// Validate Confirm Password - must match password

const confirmPassword = $('#ConfirmPassword').val();

if (!confirmPassword || password !== confirmPassword) {

$('#ConfirmPassword').addClass('is-invalid');

$('#confirmPasswordError').show();

isValid = false;

}

// Scroll to first error

if (!isValid) {

const firstError = $('.is-invalid').first();

if (firstError.length) {

$('html, body').animate({

scrollTop: firstError.offset().top - 100

}, 500);

}

}

return isValid;

}

// Prevent back button resubmission

window.addEventListener('pageshow', function(event) {

if (event.persisted || (window.performance && window.performance.navigation.type === 2)) {

// Reset form state

formSubmitted = false;

$('#submitBtn').prop('disabled', false);

$('#loadingSpinner').hide();

$('#submitText').text('Create Account');

}

});

// Handle server-side error messages

$(document).ready(function () {

var errorMessage = '@ViewBag.Error';

if (errorMessage) {

// Error message will be displayed in the alert div, no need for additional popup

}

});

});

function goBack() {

window.history.back();

}

</script>

**CustomerTable.cshtml**

@model IEnumerable<BankLoanProject.Models.Entities.Customer>

<link rel="stylesheet" href="~/css/admindashboard.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />

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

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

<!-- Dashboard Header -->

<div class="dashboard-header">

<h1>

<i class="fas fa-user-shield me-3"></i>

Customer Management Dashboard

</h1>

</div>

<!-- Table Controls -->

<div class="table-controls">

<div class="rows-control">

<label for="rowsPerPage">

<i class="fas fa-list-ol"></i>

Rows per page:

</label>

<select id="rowsPerPage" class="form-select" onchange="changeRowsPerPage()">

<option value="5" selected>5</option>

<option value="10">10</option>

<option value="15">15</option>

</select>

</div>

<div>

<label for="statusFilter" class="form-label"><i class="fas fa-filter me-1"></i>Status</label>

<select id="statusFilter" class="form-select filter-dropdown" onchange="applyFilters()">

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

<option value="Pending">Pending</option>

<option value="Verified">Verified</option>

</select>

</div>

<div class="controls-right">

<div class="input-group search-input-group">

<span class="input-group-text">

<i class="fas fa-search"></i>

</span>

<input type="text" class="form-control" id="searchInput" placeholder="Search customers..." oninput="searchCustomers()">

</div>

</div>

</div>

<!-- Table Container -->

<div class="table-container">

<div class="table-wrapper">

<table class="table">

<thead>

<tr>

<th>Customer ID</th>

<th>Name</th>

<th>Email</th>

<th>Phone</th>

<th>Address</th>

<th>Documents</th>

<th>KYC Status</th>

<th class="actions-column">Actions</th>

</tr>

</thead>

<tbody id="customerTableBody">

@foreach (var customer in Model)

{

<tr class="customer-row" data-customer-id="@customer.CustomerId" data-original-id="@customer.CustomerId" data-kyc-status="@customer.KycStatus.ToLower()">

<td><span class="customer-id">@customer.CustomerId</span></td>

<td><strong>@customer.Name</strong></td>

<td>@customer.Email</td>

<td>@customer.Phone</td>

<td>@customer.Address</td>

<td>

<a href="@customer.DocumentPath" target="\_blank" class="document-link">

documents/@(customer.Name.ToLower().Replace(" ", "\_"))\_kyc.pdf

</a>

</td>

<td>

@if (customer.KycStatus == "Approved" || customer.KycStatus == "Verified")

{

<span class="status-badge status-verified">

<i class="fas fa-check-circle"></i> Verified

</span>

}

else

{

<span class="status-badge status-pending">

<i class="fas fa-clock"></i> @customer.KycStatus

</span>

}

</td>

<td class="actions-cell">

<div class="action-buttons">

<button class="btn-action btn-edit" onclick="openEditModal(@customer.CustomerId, '@customer.KycStatus')">

<i class="fas fa-edit"></i>

<span>Edit</span>

</button>

<form asp-action="Delete" method="post" style="display:inline;" onsubmit="handleFormSubmit(event)">

<input type="hidden" name="id" value="@customer.CustomerId" />

<button type="submit" class="btn-action btn-delete" onclick="return confirm('Are you sure you want to delete this customer?');">

<i class="fas fa-trash-alt"></i>

<span>Delete</span>

</button>

</form>

<form asp-action="Verify" method="post" style="display:inline;" onsubmit="handleFormSubmit(event)">

<input type="hidden" name="id" value="@customer.CustomerId" />

<button type="submit" class="btn-action btn-verify">

<i class="fas fa-user-check"></i>

<span>Verify</span>

</button>

</form>

</div>

</td>

</tr>

}

</tbody>

</table>

</div>

<!-- Pagination Section -->

<div class="pagination-section">

<div class="pagination-info" id="paginationInfo">

<!-- Pagination info will be generated by JavaScript -->

</div>

<nav aria-label="Disbursement pagination">

<ul class="pagination" id="paginationContainer">

<!-- Pagination buttons will be generated by JavaScript -->

</ul>

</nav>

</div>

</div>

<!-- Edit Modal -->

<div class="modal fade" id="editModal" tabindex="-1" aria-labelledby="editModalLabel" aria-hidden="true">

<div class="modal-dialog modal-lg">

<div class="modal-content">

<div class="modal-header">

<h5 class="modal-title" id="editModalLabel">

<i class="fas fa-edit"></i> Edit Customer

</h5>

<button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>

</div>

<div class="modal-body" id="editModalBody">

<!-- Form will be loaded here -->

</div>

</div>

</div>

</div>

@section Scripts {

<script>

// Global variables for pagination and search

let currentPage = 1;

let rowsPerPage = parseInt(document.getElementById('rowsPerPage')?.value || 10); // Default

let allRows = [];

let filteredRows = [];

let currentScrollPosition = 0;

// Initialize on page load

document.addEventListener('DOMContentLoaded', function () {

restoreSavedState();

initializeTable();

});

function restoreSavedState() {

const savedRowsPerPage = sessionStorage.getItem('rowsPerPage') || localStorage.getItem('rowsPerPage');

const savedCurrentPage = sessionStorage.getItem('currentPage') || localStorage.getItem('currentPage');

if (savedRowsPerPage) {

rowsPerPage = parseInt(savedRowsPerPage);

document.getElementById('rowsPerPage').value = rowsPerPage;

}

if (savedCurrentPage) {

currentPage = parseInt(savedCurrentPage);

}

}

function saveCurrentState() {

sessionStorage.setItem('rowsPerPage', rowsPerPage);

sessionStorage.setItem('currentPage', currentPage);

localStorage.setItem('rowsPerPage', rowsPerPage);

localStorage.setItem('currentPage', currentPage);

}

function initializeTable() {

allRows = Array.from(document.querySelectorAll('.customer-row'));

applyFilters();

updateTable();

}

function applyFilters() {

const searchTerm = document.getElementById('searchInput')?.value.toLowerCase();

const statusFilter = document.getElementById('statusFilter')?.value.toLowerCase();

filteredRows = allRows.filter(row => {

const cells = row.getElementsByTagName('td');

if (cells.length === 0) return false;

if (cells.length !== 8) return false;

const statusTypeIndex = 6;

const rowText = row.textContent.toLowerCase();

const statusTypeText = cells[statusTypeIndex]?.textContent.trim().toLowerCase() || '';

const matchesSearch = !searchTerm || rowText.includes(searchTerm);

const matchesStatus = !statusFilter || statusTypeText.includes(statusFilter);

return matchesSearch && matchesStatus;

});

currentPage = 1;

updateTable();

}

function updateTable() {

currentScrollPosition = window.pageYOffset || document.documentElement.scrollTop;

allRows.forEach(row => row.style.display = 'none');

const totalRows = filteredRows.length;

const totalPages = Math.ceil(totalRows / rowsPerPage) || 1;

const startIndex = (currentPage - 1) \* rowsPerPage;

const endIndex = Math.min(startIndex + rowsPerPage, totalRows);

for (let i = startIndex; i < endIndex; i++) {

if (filteredRows[i]) {

filteredRows[i].style.display = '';

}

}

updatePagination(totalPages, totalRows, startIndex, endIndex);

saveCurrentState();

setTimeout(() => {

window.scrollTo(0, currentScrollPosition);

}, 10);

}

function updatePagination(totalPages, totalRows, startIndex, endIndex) {

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

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

paginationContainer.innerHTML = '';

if (totalPages <= 1) {

const singleBtn = createBootstrapPaginationButton('1', 1, false, true);

paginationContainer.appendChild(singleBtn);

} else {

const firstBtn = createBootstrapPaginationButton('«', 1, currentPage === 1);

paginationContainer.appendChild(firstBtn);

const prevBtn = createBootstrapPaginationButton('‹', currentPage - 1, currentPage === 1);

paginationContainer.appendChild(prevBtn);

const maxVisiblePages = 5;

let startPage = Math.max(1, currentPage - Math.floor(maxVisiblePages / 2));

let endPage = Math.min(totalPages, startPage + maxVisiblePages - 1);

if (endPage - startPage + 1 < maxVisiblePages) {

startPage = Math.max(1, endPage - maxVisiblePages + 1);

}

for (let i = startPage; i <= endPage; i++) {

const pageBtn = createBootstrapPaginationButton(i, i, false, i === currentPage);

paginationContainer.appendChild(pageBtn);

}

const nextBtn = createBootstrapPaginationButton('›', currentPage + 1, currentPage === totalPages);

paginationContainer.appendChild(nextBtn);

const lastBtn = createBootstrapPaginationButton('»', totalPages, currentPage === totalPages);

paginationContainer.appendChild(lastBtn);

}

if (totalRows > 0) {

paginationInfo.innerHTML = `<i class="fas fa-info-circle"></i> Showing ${startIndex + 1} to ${endIndex} of ${totalRows} entries`;

} else {

paginationInfo.innerHTML = `<i class="fas fa-info-circle"></i> No matching records found`;

}

}

function createBootstrapPaginationButton(text, page, disabled, active = false) {

const li = document.createElement('li');

li.className = `page-item ${active ? 'active' : ''} ${disabled ? 'disabled' : ''}`;

const button = document.createElement('a');

button.className = 'page-link';

button.innerHTML = text;

button.href = '#';

if (!disabled) {

button.onclick = (e) => {

e.preventDefault();

goToPage(page);

};

}

li.appendChild(button);

return li;

}

function goToPage(page) {

currentPage = page;

updateTable();

}

function changeRowsPerPage() {

rowsPerPage = parseInt(document.getElementById('rowsPerPage').value);

currentPage = 1; // Reset to first page

updateTable();

}

function searchCustomers() {

currentPage = 1; // Reset to first page

applyFilters();

updateTable();

}

function refreshData() {

const refreshBtn = document.querySelector('.btn-refresh i');

refreshBtn.classList.add('fa-spin');

document.getElementById('searchInput').value = '';

currentPage = 1; // Reset to first page

applyFilters();

setTimeout(() => {

updateTable();

refreshBtn.classList.remove('fa-spin');

}, 800);

}

function openEditModal(id, kycStatus) {

$.get('/Customer/Edit/' + id, function (data) {

$('#editModalBody').html(data);

setTimeout(() => {

const kycSelect = document.querySelector('#editForm select[name="KycStatus"]');

if (kycSelect) {

const normalizedStatus = (kycStatus === "Approved") ? "Verified" : kycStatus;

kycSelect.value = normalizedStatus;

}

}, 100);

$('#editModal').modal('show');

});

}

function handleFormSubmit(event) {

currentScrollPosition = window.pageYOffset || document.documentElement.scrollTop;

saveCurrentState();

sessionStorage.setItem('scrollPosition', currentScrollPosition);

}

window.addEventListener('load', function () {

const savedScrollPosition = sessionStorage.getItem('scrollPosition');

if (savedScrollPosition) {

setTimeout(() => {

window.scrollTo(0, parseInt(savedScrollPosition));

sessionStorage.removeItem('scrollPosition');

}, 100);

}

});

</script>

}

**Edit.cshtml**

@model BankLoanProject.Models.EditCustomerViewModel

<!-- Remove any layout or navigation references -->

@{

Layout = null;

var maxDate = DateTime.Today.AddYears(-18).ToString("yyyy-MM-dd");

}

<form asp-controller="Customer" asp-action="Edit" method="post" id="editForm">

@Html.AntiForgeryToken()

<input type="hidden" asp-for="CustomerId" />

<link rel="stylesheet" href="~/css/admindashboard.css" />

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css" />

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

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

<div class="row">

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

<div class="form-group">

<label asp-for="Name">

<i class="fas fa-user"></i> Name

</label>

<input asp-for="Name" class="form-control" placeholder="Enter full name" />

<span asp-validation-for="Name" class="text-danger"></span>

</div>

</div>

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

<div class="form-group">

<label asp-for="Email">

<i class="fas fa-envelope"></i> Email

</label>

<input asp-for="Email" class="form-control" placeholder="Enter email address" />

<span asp-validation-for="Email" class="text-danger"></span>

</div>

</div>

</div>

<div class="row">

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

<div class="form-group">

<label asp-for="Phone">

<i class="fas fa-phone"></i> Phone

</label>

<input asp-for="Phone" class="form-control" placeholder="Enter phone number" />

<span asp-validation-for="Phone" class="text-danger"></span>

</div>

</div>

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

<div class="form-group">

<label asp-for="DateOfBirth">

<i class="fas fa-calendar"></i> Date of Birth

</label>

<input asp-for="DateOfBirth" class="form-control" type="date" onkeydown="return false;" max="@maxDate" />

<span asp-validation-for="DateOfBirth" class="text-danger"></span>

</div>

</div>

</div>

<div class="form-group">

<label asp-for="Address">

<i class="fas fa-map-marker-alt"></i> Address

</label>

<textarea asp-for="Address" class="form-control" rows="3" placeholder="Enter complete address"></textarea>

<span asp-validation-for="Address" class="text-danger"></span>

</div>

<div class="row">

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

<div class="form-group">

<label asp-for="IdentityType">

<i class="fas fa-id-card"></i> Identity Type

</label>

<input asp-for="IdentityType" class="form-control" placeholder="Enter identity type" readonly/>

<span asp-validation-for="IdentityType" class="text-danger"></span>

</div>

</div>

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

<div class="form-group">

<label asp-for="KycStatus">

<i class="fas fa-check-circle"></i> KYC Status

</label>

<select asp-for="KycStatus" class="form-control">

<option value="Pending">Pending</option>

<option value="Verified">Verified</option>

</select>

<span asp-validation-for="KycStatus" class="text-danger"></span>

</div>

</div>

</div>

<div class="d-flex justify-content-end gap-2 mt-4">

<button type="button" class="btn-secondary" data-bs-dismiss="modal">

<i class="fas fa-times"></i> Cancel

</button>

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

<i class="fas fa-save"></i> Save Changes

</button>

</div>

</form>

<script>

$(document).ready(function () {

$('#editForm').submit(function (e) {

e.preventDefault();

let isValid = true;

// Clear previous errors

$('.form-control').removeClass('is-invalid');

// Validate Name

const name = $('[name="Name"]').val().trim();

const nameRegex = /^[a-zA-Z\s]+$/;

if (!name || !nameRegex.test(name)) {

$('[name="Name"]').addClass('is-invalid');

isValid = false;

}

// Validate Email

const email = $('[name="Email"]').val().trim();

const emailRegex = /^[^\s]+[^\s]+\.[^\s]+$/;

if (!email || !emailRegex.test(email)) {

$('[name="Email"]').addClass('is-invalid');

isValid = false;

}

// Validate Phone

const phone = $('[name="Phone"]').val().trim();

const phoneRegex = /^\d{10}$/;

if (!phone || !phoneRegex.test(phone)) {

$('[name="Phone"]').addClass('is-invalid');

isValid = false;

}

// Validate Address

const address = $('[name="Address"]').val().trim();

if (!address) {

$('[name="Address"]').addClass('is-invalid');

isValid = false;

}

// Validate Identity Type

const identityType = $('[name="IdentityType"]').val().trim();

if (!identityType) {

$('[name="IdentityType"]').addClass('is-invalid');

isValid = false;

}

if (!isValid) {

$('html, body').animate({

scrollTop: $('.is-invalid').first().offset().top - 100

}, 500);

return;

}

const submitBtn = $(this).find('.btn-primary');

const originalText = submitBtn.html();

submitBtn.html('<i class="fas fa-spinner fa-spin"></i> Saving...');

submitBtn.prop('disabled', true);

$.ajax({

type: 'POST',

url: '/Customer/Edit',

data: $(this).serialize(),

success: function () {

$('#editModal').modal('hide');

location.reload();

},

error: function () {

alert('Error updating customer. Please try again.');

submitBtn.html(originalText);

submitBtn.prop('disabled', false);

}

});

});

// File validation

$('#IdentityProof').on('change', function () {

const file = this.files[0];

const allowedExtensions = ['pdf', 'jpg', 'jpeg', 'png'];

if (file) {

const fileName = file.name;

const fileExtension = fileName.split('.').pop().toLowerCase();

if (!allowedExtensions.includes(fileExtension)) {

alert("Only PDF, JPEG, and PNG files are allowed.");

$(this).val('');

$('#identityProofText').text('Choose Identity Document');

$('#identityFileName').hide();

return;

}

$('#identityProofText').text('Document Selected');

$('#identityFileName').text(fileName.length > 40 ? fileName.substring(0, 40) + '...' : fileName).show();

$('#fileError').hide();

$(this).removeClass('is-invalid');

} else {

$('#identityProofText').text('Choose Identity Document');

$('#identityFileName').hide();

}

});

// Show server-side error if present

var errorMessage = '@(ViewBag.Error ?? "")';

if (errorMessage) {

alert(errorMessage);

}

});

</script>

**Wwwroot**

**Css**

**Register.css**

/\* Modern Customer Registration Form Styling \*/

:root {

--primary: #2c3e50;

--secondary: #3498db;

--accent: #16a085;

--light: #f8f9fa;

--dark: #343a40;

--success: #27ae60;

--warning: #f39c12;

--danger: #e74c3c;

--info: #1abc9c;

}

body {

background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

padding: 20px 0;

min-height: 100vh;

position: relative;

overflow-x: hidden;

margin: 0;

}

/\* Animated Background Elements \*/

body::before {

content: "";

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background: radial-gradient(circle at 20% 80%, rgba(52, 152, 219, 0.1) 0%, transparent 50%), radial-gradient(circle at 80% 20%, rgba(22, 160, 133, 0.1) 0%, transparent 50%), radial-gradient(circle at 40% 40%, rgba(44, 62, 80, 0.05) 0%, transparent 50%);

z-index: -2;

animation: backgroundShift 20s ease-in-out infinite;

}

body::after {

content: "";

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

background-image: linear-gradient(45deg, rgba(52, 152, 219, 0.03) 25%, transparent 25%), linear-gradient(-45deg, rgba(22, 160, 133, 0.03) 25%, transparent 25%), linear-gradient(45deg, transparent 75%, rgba(44, 62, 80, 0.02) 75%), linear-gradient(-45deg, transparent 75%, rgba(52, 152, 219, 0.02) 75%);

background-size: 60px 60px;

background-position: 0 0, 0 30px, 30px -30px, -30px 0px;

z-index: -1;

animation: patternMove 30s linear infinite;

}

/\* Floating geometric shapes \*/

.bg-shape {

position: fixed;

border-radius: 50%;

background: linear-gradient(135deg, rgba(52, 152, 219, 0.1), rgba(22, 160, 133, 0.1));

z-index: -1;

animation: float 15s ease-in-out infinite;

}

.bg-shape:nth-child(1) {

width: 200px;

height: 200px;

top: 10%;

left: 10%;

animation-delay: 0s;

}

.bg-shape:nth-child(2) {

width: 150px;

height: 150px;

top: 60%;

right: 15%;

animation-delay: 5s;

}

.bg-shape:nth-child(3) {

width: 100px;

height: 100px;

bottom: 20%;

left: 20%;

animation-delay: 10s;

}

.bg-shape:nth-child(4) {

width: 80px;

height: 80px;

top: 30%;

right: 30%;

animation-delay: 7s;

}

@keyframes backgroundShift {

0%, 100% {

transform: translateX(0) translateY(0);

}

25% {

transform: translateX(-20px) translateY(-10px);

}

50% {

transform: translateX(20px) translateY(10px);

}

75% {

transform: translateX(-10px) translateY(20px);

}

}

@keyframes patternMove {

0% {

background-position: 0 0, 0 30px, 30px -30px, -30px 0px;

}

100% {

background-position: 60px 60px, 60px 90px, 90px 30px, 30px 60px;

}

}

@keyframes float {

0%, 100% {

transform: translateY(0px) rotate(0deg);

opacity: 0.7;

}

25% {

transform: translateY(-20px) rotate(90deg);

opacity: 0.4;

}

50% {

transform: translateY(-10px) rotate(180deg);

opacity: 0.8;

}

75% {

transform: translateY(-30px) rotate(270deg);

opacity: 0.3;

}

}

.main-container {

max-width: 1000px;

margin: 0 auto;

padding: 0 15px;

position: relative;

z-index: 1;

}

.registration-card {

background: rgba(255, 255, 255, 0.95);

backdrop-filter: blur(10px);

border-radius: 20px;

padding: 30px;

box-shadow: 0 20px 40px rgba(0, 0, 0, 0.1), 0 0 0 1px rgba(255, 255, 255, 0.2);

margin: 20px 0;

border: 1px solid rgba(255, 255, 255, 0.3);

position: relative;

overflow: hidden;

}

.registration-card::before {

content: "";

position: absolute;

top: 0;

left: 0;

right: 0;

height: 4px;

background: linear-gradient(90deg, var(--secondary), var(--accent), var(--primary));

border-radius: 20px 20px 0 0;

}

.form-header {

background: linear-gradient(135deg, var(--primary), var(--secondary));

color: white;

border-radius: 15px;

padding: 25px;

margin-bottom: 30px;

text-align: center;

box-shadow: 0 10px 30px rgba(44, 62, 80, 0.3), inset 0 1px 0 rgba(255, 255, 255, 0.2);

position: relative;

overflow: hidden;

}

.form-header::before {

content: "";

position: absolute;

top: 0;

left: 0;

right: 0;

bottom: 0;

background: url("data:image/svg+xml,%3Csvg width='60' height='60' viewBox='0 0 60 60' xmlns='http://www.w3.org/2000/svg'%3E%3Cg fill='none' fillrule='evenodd'%3E%3Cg fill='rgba(255,255,255,0.1)' fill-opacity='0.1'%3E%3Ccircle cx='30' cy='30' r='2'/%3E%3C/g%3E%3C/g%3E%3C/svg%3E");

opacity: 0.3;

}

.form-title {

font-size: 2rem;

font-weight: 700;

margin-bottom: 8px;

text-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);

position: relative;

z-index: 1;

}

.form-subtitle {

font-size: 1rem;

opacity: 0.9;

margin: 0;

position: relative;

z-index: 1;

}

.section-title {

color: var(--primary);

font-weight: 600;

font-size: 1.1rem;

margin-bottom: 20px;

padding-bottom: 8px;

border-bottom: 2px solid var(--secondary);

display: flex;

align-items: center;

gap: 10px;

position: relative;

}

.section-title::after {

content: "";

position: absolute;

bottom: -2px;

left: 0;

width: 50px;

height: 2px;

border-radius: 1px;

}

.form-group {

margin-bottom: 20px;

}

.form-label {

color: var(--primary);

font-weight: 600;

margin-bottom: 8px;

display: flex;

align-items: center;

gap: 8px;

font-size: 0.9rem;

}

.form-label i {

color: var(--secondary);

width: 16px;

}

.form-control,

.form-select {

border: 2px solid rgba(224, 224, 224, 0.8);

border-radius: 10px;

padding: 12px 15px;

font-size: 0.95rem;

transition: all 0.3s;

background: rgba(255, 255, 255, 0.9);

backdrop-filter: blur(5px);

width: 100%;

box-sizing: border-box;

}

.form-control:focus,

.form-select:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

outline: none;

background: rgba(255, 255, 255, 1);

}

.form-control.is-invalid,

.form-select.is-invalid {

border-color: var(--danger) !important;

box-shadow: 0 0 0 0.25rem rgba(231, 76, 60, 0.25) !important;

}

.file-upload-wrapper {

position: relative;

overflow: hidden;

display: inline-block;

width: 100%;

}

.file-upload-btn {

background: linear-gradient(135deg, var(--info), var(--secondary));

color: white;

padding: 12px 20px;

border-radius: 10px;

border: none;

cursor: pointer;

font-weight: 600;

font-size: 0.9rem;

transition: all 0.3s;

width: 100%;

text-align: center;

display: flex;

align-items: center;

justify-content: center;

gap: 8px;

box-shadow: 0 6px 20px rgba(26, 188, 156, 0.3);

position: relative;

overflow: hidden;

}

.file-upload-btn::before {

content: "";

position: absolute;

top: 0;

left: -100%;

width: 100%;

height: 100%;

background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);

transition: left 0.5s;

}

.file-upload-btn:hover::before {

left: 100%;

}

.file-upload-btn:hover {

transform: translateY(-2px);

box-shadow: 0 8px 25px rgba(26, 188, 156, 0.4);

}

.file-name {

font-size: 0.85rem;

color: var(--success);

margin-top: 5px;

font-weight: 500;

display: none;

}

.submit-btn {

background: linear-gradient(135deg, var(--secondary), var(--primary));

border: none;

border-radius: 12px;

padding: 15px 30px;

font-size: 1.1rem;

font-weight: 700;

color: white;

width: 100%;

cursor: pointer;

transition: all 0.3s;

text-transform: uppercase;

letter-spacing: 1px;

box-shadow: 0 10px 30px rgba(52, 152, 219, 0.4);

margin-top: 20px;

position: relative;

overflow: hidden;

display: flex;

align-items: center;

justify-content: center;

gap: 10px;

}

.submit-btn::before {

content: "";

position: absolute;

top: 0;

left: -100%;

width: 100%;

height: 100%;

background: linear-gradient(90deg, transparent, rgba(255, 255, 255, 0.2), transparent);

transition: left 0.5s;

}

.submit-btn:hover::before {

left: 100%;

}

.submit-btn:hover {

transform: translateY(-3px);

box-shadow: 0 15px 40px rgba(52, 152, 219, 0.5);

}

.submit-btn:disabled {

opacity: 0.7;

cursor: not-allowed;

transform: none;

}

.loading-spinner {

display: none;

width: 20px;

height: 20px;

border: 2px solid rgba(255, 255, 255, 0.3);

border-radius: 50%;

border-top-color: white;

animation: spin 1s ease-in-out infinite;

margin-right: 10px;

}

/\* Alert messages \*/

.alert {

padding: 15px 20px;

margin-bottom: 20px;

border-radius: 12px;

font-weight: 500;

display: flex;

align-items: center;

gap: 10px;

backdrop-filter: blur(10px);

}

.alert-success {

background: linear-gradient(135deg, rgba(39, 174, 96, 0.9), rgba(26, 188, 156, 0.9));

color: white;

box-shadow: 0 10px 30px rgba(39, 174, 96, 0.3);

}

.alert-danger {

background: linear-gradient(135deg, rgba(231, 76, 60, 0.9), rgba(192, 57, 43, 0.9));

color: white;

box-shadow: 0 10px 30px rgba(231, 76, 60, 0.3);

}

/\* Error messages \*/

.error-message {

background: linear-gradient(135deg, var(--danger), #c0392b);

color: white;

padding: 8px 12px;

border-radius: 8px;

margin-top: 8px;

font-size: 0.85rem;

font-weight: 500;

display: none;

animation: slideInUp 0.3s ease-out;

box-shadow: 0 4px 15px rgba(231, 76, 60, 0.3);

}

.success-message {

background: linear-gradient(135deg, var(--success), var(--info));

color: white;

padding: 20px;

border-radius: 12px;

text-align: center;

font-weight: 600;

font-size: 1.1rem;

margin-top: 20px;

display: none;

animation: bounceIn 0.6s ease-out;

box-shadow: 0 10px 30px rgba(39, 174, 96, 0.3);

}

/\* Validation errors \*/

.text-danger {

color: var(--danger);

font-size: 0.85rem;

margin-top: 5px;

display: block;

font-weight: 500;

}

@keyframes spin {

to {

transform: rotate(360deg);

}

}

@keyframes slideInUp {

from {

opacity: 0;

transform: translateY(20px);

}

to {

opacity: 1;

transform: translateY(0);

}

}

@keyframes bounceIn {

0% {

opacity: 0;

transform: scale(0.3);

}

50% {

opacity: 1;

transform: scale(1.05);

}

70% {

transform: scale(0.9);

}

100% {

opacity: 1;

transform: scale(1);

}

}

@media (max-width: 768px) {

.main-container {

padding: 0 10px;

}

.registration-card {

padding: 20px;

margin: 10px 0;

border-radius: 15px;

}

.form-title {

font-size: 1.5rem;

}

.form-control,

.form-select {

padding: 10px 12px;

font-size: 0.9rem;

}

.bg-shape {

display: none;

}

}

/\* Custom styling for better alignment \*/

.row {

margin-left: -10px;

margin-right: -10px;

}

.row > \* {

padding-left: 10px;

padding-right: 10px;

}

/\* Utility classes \*/

.me-3 {

margin-right: 1rem;

}

**Admindashboard.css**

/\* Admin Dashboard Styling \*/

:root {

--primary: #2c3e50;

--secondary: #3498db;

--accent: #16a085;

--light: #f8f9fa;

--dark: #343a40;

--success: #27ae60;

--warning: #f39c12;

--danger: #e74c3c;

--info: #1abc9c;

--table-header-bg: linear-gradient(to right, #2c3e50, #3498db);

--pagination-border: #495057;

--pagination-bg: #f8f9fa;

--pagination-active-bg: #3498db;

}

body {

background-color: #f5f7fa;

font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;

padding: 20px 0;

margin: 0;

}

/\* Header Container \*/

.dashboard-header {

background: linear-gradient(135deg, var(--primary), var(--secondary));

color: white;

border-radius: 15px;

padding: 10px 25px;

margin: 70px auto 30px auto;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);

text-align: center;

max-width: 92%;

}

.dashboard-header h1 {

margin: 0;

font-size: 1.6rem;

font-weight: 700;

text-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);

display: flex;

align-items: center;

justify-content: center;

gap: 15px;

}

.dashboard-header h2 {

margin: 0;

font-size: 1.5rem;

font-weight: 700;

text-shadow: 0 2px 4px rgba(0, 0, 0, 0.3);

display: flex;

align-items: center;

justify-content: center;

gap: 15px;

}

.dashboard-header p {

margin: 8px 0 0 0;

opacity: 0.9;

font-size: 1rem;

}

/\* Table Controls \*/

.table-controls {

background-color: white;

border-radius: 12px;

padding: 20px;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);

margin: 20px auto;

max-width: 96%;

display: flex;

justify-content: space-between;

align-items: center;

flex-wrap: wrap;

gap: 15px;

min-height: 70px;

}

.rows-control {

display: flex;

align-items: center;

gap: 10px;

}

.rows-control label {

color: var(--primary);

font-weight: 600;

font-size: 0.95rem;

display: flex;

align-items: center;

gap: 8px;

white-space: nowrap;

}

.rows-control.form-select {

min-width: 80px;

border: 2px solid #e0e0e0;

border-radius: 6px;

font-size: 0.9rem;

}

.rows-control.form-select:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

}

/\* Controls Right Side \*/

.controls-right {

display: flex;

align-items: center;

gap: 15px;

}

/\* Bootstrap Search Input Group \*/

.search-input-group {

width: 300px;

}

.search-input-group.input-group-text {

background-color: var(--secondary);

color: white;

border: 2px solid var(--secondary);

}

.search-input-group.form-control {

border: 2px solid #e0e0e0;

border-left: none;

}

.search-input-group.form-control:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

}

/\* Table Container \*/

.table-container {

background-color: white;

border-radius: 12px;

padding: 0;

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.05);

margin: 20px auto;

max-width: 96%;

overflow: hidden;

border: 4px solid var(--primary);

min-height: auto;

display: flex;

flex-direction: column;

}

/\* Table wrapper to ensure consistent height \*/

.table-wrapper {

flex: 1;

min-height: auto;

display: flex;

flex-direction: column;

}

/\* Table Styling \*/

.table {

margin-bottom: 0;

border-collapse: separate;

border-spacing: 0;

width: 100%;

/\* flex: 1;\*/

}

/\* Table Header \*/

.table thead {

background: var(--table-header-bg);

}

.table thead th {

background: var(--table-header-bg);

color: white;

font-weight: 600;

border: 1px solid rgba(255, 255, 255, 0.2);

border-bottom: 2px solid rgba(255, 255, 255, 0.3);

padding: 8px 6px;

text-transform: uppercase;

font-size: 0.7rem;

letter-spacing: 1px;

vertical-align: middle;

position: relative;

text-align: center; /\* Changed from left to center \*/

white-space: nowrap; /\* Prevent header text wrapping \*/

}

.table thead th:first-child {

border-left: none;

}

.table thead th:last-child {

border-right: none;

}

.table thead th i {

margin-right: 8px;

font-size: 1rem;

}

.table thead th.actions-column {

width: 220px;

min-width: 220px;

}

.table tbody td {

padding: 8px 6px;

border: 1px solid #e0e0e0;

border-top: none;

vertical-align: middle;

word-break: break-word; /\* Allow text to wrap within cells \*/

font-size: 0.8rem;

text-align: center; /\* Added to center table body content \*/

}

.table tbody td:first-child {

border-left: none;

}

.table tbody td:last-child {

border-right: none;

}

.table tbody tr:hover {

background-color: rgba(52, 152, 219, 0.05);

transition: background-color 0.2s ease;

}

.table tbody tr:last-child td {

border-bottom: none;

}

/\* Customer ID Styling \*/

.customer-id {

font-weight: 700;

color: var(--primary);

font-size: 1rem;

}

/\* Status Badge \*/

.status-badge {

padding: 6px 12px;

border-radius: 20px;

font-weight: 500;

display: inline-flex;

align-items: center;

gap: 5px;

font-size: 0.8rem;

text-align: center;

min-width: 90px;

justify-content: center;

}

.status-verified {

background-color: rgba(39, 174, 96, 0.2);

color: var(--success);

border: 1px solid var(--success);

}

.status-pending {

background-color: rgba(243, 156, 18, 0.2);

color: var(--warning);

border: 1px solid var(--warning);

}

.status-approved {

background-color: rgba(39, 174, 96, 0.2);

color: var(--success);

border: 1px solid var(--success);

}

/\* Document Link \*/

.document-link {

color: var(--secondary);

text-decoration: none;

font-weight: 500;

transition: all 0.3s ease;

}

.document-link:hover {

color: var(--primary);

text-decoration: underline;

}

/\* Action Buttons - Modern Card Style \*/

.actions-cell {

width: 220px;

min-width: 220px;

padding: 12px 8px;

}

.action-buttons {

display: flex;

gap: 4px;

flex-wrap: nowrap;

justify-content: center;

align-items: center;

}

.btn-action {

border-radius: 8px;

padding: 4px 6px;

font-size: 0.6rem;

font-weight: 600;

transition: all 0.3s;

border: none;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

min-width: 35px;

height: 35px;

cursor: pointer;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

gap: 2px;

}

.btn-action i {

font-size: 0.7rem;

margin-bottom: 2px;

}

.btn-action span {

font-size: 0.55rem;

font-weight: 600;

text-transform: uppercase;

letter-spacing: 0.5px;

}

.btn-edit {

background-color: #ff9500;

color: white;

}

.btn-edit:hover {

background-color: #e6850e;

color: white;

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(255, 149, 0, 0.3);

}

.btn-delete {

background-color: #ff4757;

color: white;

}

.btn-delete:hover {

background-color: #ff3742;

color: white;

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(255, 71, 87, 0.3);

}

.btn-verify {

background-color: #2ed573;

color: white;

}

.btn-verify:hover {

background-color: #26d065;

color: white;

transform: translateY(-2px);

box-shadow: 0 4\_px 8px rgba(46, 213, 115, 0.3);

}

/\* Pagination section - Minimal Gap \*/

.pagination-section {

background-color: #f8f9fa;

border-top: 1px solid #e0e0e0;

padding: 8px 15px;

margin-top: 0;

}

/\* Enhanced Bootstrap Pagination Styling \*/

.pagination {

margin: 4px 0 0 0;

gap: 6px;

justify-content: center;

}

/\* Pagination Info \*/

.pagination-info {

color: var(--primary);

font-weight: 500;

font-size: 0.85rem;

display: flex;

align-items: center;

justify-content: center;

gap: 6px;

margin-bottom: 4px;

}

.pagination .page-item .page-link {

color: #2c3e50;

background-color: var(--pagination-bg);

border: 3px solid #2c3e50;

padding: 8px 12px;

font-weight: 700;

font-size: 0.9rem;

min-width: 40px;

min-height: 40px;

text-align: center;

display: flex;

align-items: center;

justify-content: center;

border-radius: 6px;

transition: all 0.3s ease;

text-decoration: none;

box-shadow: 0 1px 3px rgba(0, 0, 0, 0.1);

}

.pagination .page-item.disabled .page-link {

opacity: 0.5;

cursor: not-allowed;

background-color: #e9ecef;

border-color: #6c757d;

color: #495057;

transform: none;

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.05);

}

.pagination .page-item.disabled .page-link:hover {

background-color: #e9ecef;

border-color: #6c757d;

color: #495057;

transform: none;

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.05);

}

.pagination .page-item.active .page-link {

background-color: var(--pagination-active-bg);

color: white;

border-color: var(--secondary);

transform: translateY(-2px);

box-shadow: 0 3px 6px rgba(52, 152, 219, 0.4);

font-weight: 800;

}

.pagination .page-item.active .page-link:hover {

background-color: var(--secondary);

color: white;

border-color: var(--primary);

}

/\* Modal Styling \*/

.modal-content {

border-radius: 15px;

border: none;

box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);

overflow: hidden;

}

.modal-header {

background: linear-gradient(to right, var(--primary), var(--secondary));

color: white;

border-bottom: none;

padding: 20px 25px;

}

.modal-title {

font-weight: 600;

letter-spacing: 0.5px;

font-size: 1.3rem;

display: flex;

align-items: center;

gap: 10px;

}

.modal-body {

padding: 25px;

background-color: #fafafa;

}

/\* Form Styling \*/

.form-group {

margin-bottom: 20px;

position: relative;

}

.form-group label {

font-weight: 600;

color: var(--primary);

margin-bottom: 8px;

display: flex;

align-items: center;

gap: 8px;

font-size: 0.95rem;

}

.form-control {

border: 2px solid #e0e0e0;

border-radius: 8px;

padding: 12px 15px;

transition: all 0.3s;

font-size: 0.95rem;

width: 100%;

box-sizing: border-box;

}

.form-control:focus {

border-color: var(--secondary);

box-shadow: 0 0 0 0.25rem rgba(52, 152, 219, 0.25);

outline: none;

}

/\* Form Buttons \*/

.btn-primary {

background: linear-gradient(135deg, var(--secondary), var(--primary));

color: white;

border: none;

border-radius: 8px;

padding: 12px 25px;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: flex;

align-items: center;

gap: 8px;

}

.btn-primary:hover {

background: linear-gradient(135deg, var(--primary), var(--secondary));

color: white;

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3);

}

.btn-secondary {

background-color: #6c757d;

color: white;

border: none;

border-radius: 8px;

padding: 12px 25px;

font-weight: 600;

transition: all 0.3s;

cursor: pointer;

display: flex;

align-items: center;

gap: 8px;

}

.btn-secondary:hover {

background-color: #5a6268;

color: white;

transform: translateY(-2px);

}

/\* Close button styling \*/

.btn-close {

background: none;

border: none;

font-size: 1.5rem;

color: white;

opacity: 0.8;

}

.btn-close:hover {

opacity: 1;

}

/\* Row styling \*/

.row {

display: flex;

margin: 0 -10px;

flex-wrap: wrap;

}

.col-md-6 {

flex: 0 0 50%;

max-width: 50%;

padding: 0 10px;

box-sizing: border-box;

}

/\* Responsive Design \*/

@media (max-width: 768px) {

.table-controls {

flex-direction: column;

align-items: stretch;

gap: 15px;

min-height: 140px;

}

.controls-right {

flex-direction: column;

gap: 10px;

}

.search-input-group {

width: 100%;

}

.table-container {

overflow-x: auto;

max-width: 98%;

}

.table {

min-width: 900px;

}

.dashboard-header {

margin: 15px auto;

padding: 20px;

max-width: 98%;

}

.dashboard-header h1 {

font-size: 1.5rem;

}

.action-buttons {

flex-direction: column;

gap: 2px;

}

.btn-action {

font-size: 0.55rem;

padding: 2px 4px;

min-width: 30px;

height: 30px;

}

.col-md-6 {

flex: 0 0 100%;

max-width: 100%;

}

.refresh-container {

max-width: 98%;

}

.pagination {

gap: 4px;

}

.pagination.page-item.page-link {

padding: 6px 10px;

min-width: 35px;

min-height: 35px;

font-size: 0.8rem;

}

}

@media (max-width: 576px) {

.dashboard-header h1 {

font-size: 1.3rem;

flex-direction: column;

gap: 10px;

}

.dashboard-header p {

font-size: 1rem;

}

.table thead th {

padding: 4px 2px;

font-size: 0.6rem;

}

.table tbody td {

padding: 4px 2px;

font-size: 0.7rem;

}

.actions-cell {

width: 120px;

min-width: 120px;

}

.btn-action {

font-size: 0.5rem;

padding: 2px 3px;

min-width: 25px;

height: 25px;

}

.pagination.page-item.page-link {

padding: 4px 8px;

min-width: 30px;

min-height: 30px;

font-size: 0.75rem;

}

}

/\* Loading states \*/

.loading {

opacity: 0.6;

pointer-events: none;

}

/\* Smooth transitions \*/

\* {

box-sizing: border-box;

}

/\* Re-enabled smooth transitions \*/

.table-container,

.table-controls {

transition: all 0.3s ease;

}

/\* Column width adjustments to prevent overflow \*/

.table th:nth-child(1) {

width: 8%;

}

/\* Customer ID \*/

.table th:nth-child(2) {

width: 15%;

}

/\* Name \*/

.table th:nth-child(3) {

width: 15%;

}

/\* Email \*/

.table th:nth-child(4) {

width: 17%;

}

/\* Address \*/

.table th:nth-child(5) {

width: 15%;

}

/\* Documents \*/

.table th:nth-child(6) {

width: 10%;

}

/\* KYC Status \*/

.table th:nth-child(7) {

width: 20%;

}

/\* Actions \*/

/\* Ensure table cells don't overflow \*/

.table td {

max-width: 0;

overflow: hidden;

text-overflow: ellipsis;

}

.pagination.page-item.page-link {

color: #2c3e50; /\* Darker text color \*/

background-color: var(--pagination-bg);

border: 3px solid #2c3e50; /\* Darker border \*/

padding: 8px 12px;

font-weight: 700;

font-size: 0.9rem;

min-width: 40px;

min-height: 40px;

text-align: center;

display: flex;

align-items: center;

justify-content: center;

border-radius: 8px;

transition: all 0.3s ease;

text-decoration: none;

box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);

}

.pagination.page-item.disabled.page-link {

opacity: 0.5;

cursor: not-allowed;

background-color: #e9ecef;

border-color: #6c757d; /\* Darker border for disabled state \*/

color: #495057; /\* Darker text for disabled state \*/

transform: none;

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.05);

}

.pagination.page-item.disabled.page-link:hover {

background-color: #e9ecef;

border-color: #6c757d; /\* Darker border for disabled hover state \*/

color: #495057; /\* Darker text for disabled hover state \*/

transform: none;

box-shadow: 0 1px 2px rgba(0, 0, 0, 0.05);

}

/\* Add this CSS rule after the existing pagination styles to highlight the active page

button \*/

.pagination.page-item.active.page-link {

background-color: var(--pagination-active-bg);

color: white;

border-color: var(--secondary);

transform: translateY(-2px);

box-shadow: 0 4px 8px rgba(52, 152, 219, 0.4);

font-weight: 800;

}

.pagination.page-item.active.page-link:hover {

background-color: var(--secondary);

color: white;

border-color: var(--primary);

}