**CUI**

**Comsats University Islamabad (Vehari Campus)**



**Academic Year 22-27**

**Department: CS**

**Assignment No:**

# "04"

**Student Name:**

**Hanzla Shahzad (FA22-BSE-009)**

**Subject:**

**Web Technology**

## Batch: “19”

**Teacher Name:**

**Ma’am Yasmeen Jana**

|  |
| --- |
| **AuthController.php**  <?php  namespace App\Http\Controllers;  use App\Models\Student;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Auth;  use Illuminate\Support\Facades\Hash;  use Illuminate\Support\Facades\Session;  class AuthController extends Controller  {      public function showRegister()      {          return view('auth.register');      }      public function register(Request $request)      {          $request->validate([              'name' => 'required|string|max:255',              'email' => 'required|email|unique:students,email',              'department' => 'required|string|max:255',              'password' => 'required|string|min:8|confirmed',          ]);          $student = Student::create([              'name' => $request->name,              'email' => $request->email,              'department' => $request->department,              'password' => Hash::make($request->password),          ]);          Auth::login($student);          Session::flash('message', 'Registration successful!');          return redirect()->route('dashboard');      }      public function showLogin()      {          return view('auth.login');      }      public function login(Request $request)      {          $request->validate([              'email' => 'required|email',              'password' => 'required|string',          ]);          if (Auth::attempt(['email' => $request->email, 'password' => $request->password], $request->has('remember'))) {              Session::regenerate();              Session::flash('message', 'Login successful!');              return redirect()->route('dashboard');          }          return back()->withErrors(['email' => 'Invalid credentials']);      }      public function logout()      {          Auth::logout();          Session::flush();          cookie()->queue(cookie()->forget('auth\_token'));          Session::flash('message', 'Logged out successfully!');          return redirect()->route('login');      }  }  **Controller.php**  <?php  namespace App\Http\Controllers;  use Illuminate\Routing\Controller as BaseController;  class Controller extends BaseController  {      // Base controller for all other controllers  }  **CourseController.php**  <?php  namespace App\Http\Controllers;  use App\Models\Course;  use App\Models\Student;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Auth;  class CourseController extends Controller  {      public function index(Request $request)      {          $department = $request->query('department');          $instructor = $request->query('instructor');          $query = Course::query();          if ($department) {              $query->where('department', 'like', '%' . $department . '%');          }          if ($instructor) {              $query->where('instructor', 'like', '%' . $instructor . '%');          }          $courses = $query->get();          return $request->ajax() ? response()->json($courses) : view('courses.index', compact('courses'));      }      public function create()      {          return view('courses.add');      }      public function store(Request $request)      {          $validated = $request->validate([              'title' => 'required|string|max:255',              'code' => 'required|string|max:20|unique:courses',              'credit\_hours' => 'required|integer|min:1|max:6',              'department' => 'required|string|max:255',              'instructor' => 'required|string|max:255',              'description' => 'nullable|string'          ]);          $course = Course::create($validated);          if ($request->ajax() || $request->wantsJson()) {              return response()->json([                  'message' => 'Course added successfully!',                  'course' => $course              ]);          }          return redirect()->route('courses.index')              ->with('message', 'Course added successfully!');      }      public function register(Request $request)      {          $student = Auth::user();          $courseId = $request->course\_id;          if (!$student->courses()->where('course\_id', $courseId)->exists()) {              $student->courses()->attach($courseId);              return response()->json(['message' => 'Course registered successfully!']);          }          return response()->json(['message' => 'You are already registered for this course.'], 400);      }      public function unregister(Request $request)      {          $student = Auth::user();          $courseId = $request->course\_id;          if ($student->courses()->where('course\_id', $courseId)->exists()) {              $student->courses()->detach($courseId);              if ($request->ajax() || $request->wantsJson()) {                  return response()->json(['message' => 'Course unregistered successfully!']);              } else {                  return redirect()->route('profile.show')->with('message', 'Course unregistered successfully!');              }          }          if ($request->ajax() || $request->wantsJson()) {              return response()->json(['message' => 'You are not registered for this course.'], 400);          } else {              return redirect()->route('profile.show')->with('error', 'You are not registered for this course.');          }      }  }  **ProfileController.php**  <?php  namespace App\Http\Controllers;  use App\Models\Student;  use Illuminate\Http\Request;  use Illuminate\Support\Facades\Auth;  use Illuminate\Support\Facades\Session;  class ProfileController extends Controller  {      public function show()      {          $student = Auth::user();          return view('profile.show', compact('student'));      }      public function edit()      {          $student = Auth::user();          return view('profile.edit', compact('student'));      }      public function update(Request $request)      {          $student = Auth::user();          $request->validate([              'name' => 'required|string|max:255',              'email' => 'required|email|unique:students,email,' . $student->id,              'department' => 'required|string|max:255',          ]);          $student->update([              'name' => $request->name,              'email' => $request->email,              'department' => $request->department,          ]);          Session::flash('message', 'Profile updated successfully!');          return redirect()->route('profile.show');      }      public function exportJson()      {          $student = Auth::user();          $courses = $student->courses()->get(['title', 'credit\_hours', 'instructor', 'department']);          return response()->json($courses)->header('Content-Disposition', 'attachment; filename=courses.json');      }      public function exportXml()      {          $student = Auth::user();          $courses = $student->courses()->get(['title', 'credit\_hours', 'instructor', 'department']);          $xml = new \SimpleXMLElement('<?xml version="1.0" encoding="UTF-8"?><courses></courses>');          foreach ($courses as $course) {              $courseXml = $xml->addChild('course');              $courseXml->addChild('title', htmlspecialchars($course->title));              $courseXml->addChild('credit\_hours', $course->credit\_hours);              $courseXml->addChild('instructor', htmlspecialchars($course->instructor));              $courseXml->addChild('department', htmlspecialchars($course->department));          }          return response($xml->asXML(), 200, [              'Content-Type' => 'application/xml',              'Content-Disposition' => 'attachment; filename=courses.xml',          ]);      }  }  **EnsureAuthenticated**  <?php  namespace App\Http\Middleware;  use Closure;  use Illuminate\Support\Facades\Auth;  class EnsureAuthenticated  {      public function handle($request, Closure $next)      {          if (!Auth::check()) {              return redirect()->route('login')->withErrors(['message' => 'Please log in to access this page.']);          }          return $next($request);      }  }  **Course.php**  <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Model;  class Course extends Model  {      protected $fillable = [          'title',          'code',          'credit\_hours',          'instructor',          'department',          'description'      ];      public function students()      {          return $this->belongsToMany(Student::class, 'course\_student')              ->withTimestamps();      }  }  **Student.php**  <?php  namespace App\Models;  use Illuminate\Foundation\Auth\User as Authenticatable;  use Illuminate\Notifications\Notifiable;  class Student extends Authenticatable  {      use Notifiable;      protected $fillable = ['name', 'email', 'department', 'password'];      protected $hidden = ['password', 'remember\_token'];      public function courses()      {          return $this->belongsToMany(Course::class, 'student\_courses');      }  }  **StudentCourse.php**  <?php  namespace App\Models;  use Illuminate\Database\Eloquent\Model;  class StudentCourse extends Model  {      protected $fillable = ['student\_id', 'course\_id'];  }    **AppServiceProvider.php**  <?php  namespace App\Providers;  use Illuminate\Support\ServiceProvider;  class AppServiceProvider extends ServiceProvider  {      /\*\*       \* Register any application services.       \*/      public function register(): void      {          //      }      /\*\*       \* Bootstrap any application services.       \*/      public function boot(): void      {          //      }  }  **Migration**  <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {      /\*\*       \* Run the migrations.       \*/      public function up(): void      {          Schema::create('sessions', function (Blueprint $table) {              $table->string('id')->primary();              $table->foreignId('user\_id')->nullable()->index();              $table->string('ip\_address', 45)->nullable();              $table->text('user\_agent')->nullable();              $table->longText('payload');              $table->integer('last\_activity')->index();          });      }      /\*\*       \* Reverse the migrations.       \*/      public function down(): void      {          Schema::dropIfExists('sessions');      }  };  <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration  {      /\*\*       \* Run the migrations.       \*/      public function up(): void      {          Schema::table('courses', function (Blueprint $table) {              $table->string('code')->unique()->after('title');              $table->text('description')->nullable()->after('instructor');          });      }      /\*\*       \* Reverse the migrations.       \*/      public function down(): void      {          Schema::table('courses', function (Blueprint $table) {              $table->dropColumn(['code', 'description']);          });      }  };  <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration {      public function up(): void      {          Schema::create('student\_courses', function (Blueprint $table) {              $table->id();              $table->foreignId('student\_id')->constrained()->onDelete('cascade');              $table->foreignId('course\_id')->constrained()->onDelete('cascade');              $table->timestamps();          });      }      public function down(): void      {          Schema::dropIfExists('student\_courses');      }  };  <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration {      public function up(): void      {          Schema::create('students', function (Blueprint $table) {              $table->id();              $table->string('name');              $table->string('email')->unique();              $table->string('department');              $table->string('password');              $table->rememberToken();              $table->timestamps();          });      }      public function down(): void      {          Schema::dropIfExists('students');      }  };  <?php  use Illuminate\Database\Migrations\Migration;  use Illuminate\Database\Schema\Blueprint;  use Illuminate\Support\Facades\Schema;  return new class extends Migration {      public function up(): void      {          Schema::create('courses', function (Blueprint $table) {              $table->id();              $table->string('title');              $table->integer('credit\_hours');              $table->string('instructor');              $table->string('department');              $table->timestamps();          });      }      public function down(): void      {          Schema::dropIfExists('courses');      }  };  **Web.php**  <?php  use App\Http\Controllers\AuthController;  use App\Http\Controllers\CourseController;  use App\Http\Controllers\ProfileController;  use Illuminate\Support\Facades\Route;  use Illuminate\Support\Facades\Auth;  Route::get('/', function () {      if (Auth::check()) {          return redirect()->route('dashboard');      } else {          return redirect()->route('login');      }  });  Route::get('/register', [AuthController::class, 'showRegister'])->name('register');  Route::post('/register', [AuthController::class, 'register']);  Route::get('/login', [AuthController::class, 'showLogin'])->name('login');  Route::post('/login', [AuthController::class, 'login']);  Route::post('/logout', [AuthController::class, 'logout'])->name('logout');  Route::middleware('auth')->group(function () {      Route::get('/dashboard', fn() => view('dashboard'))->name('dashboard');        // Course routes      Route::get('/courses', [CourseController::class, 'index'])->name('courses.index');      Route::get('/courses/add', [CourseController::class, 'create'])->name('courses.create');      Route::post('/courses', [CourseController::class, 'store'])->name('courses.store');      Route::post('/courses/register', [CourseController::class, 'register'])->name('courses.register');      Route::post('/courses/unregister', [CourseController::class, 'unregister'])->name('courses.unregister');        // Profile routes      Route::get('/profile', [ProfileController::class, 'show'])->name('profile.show');      Route::get('/profile/edit', [ProfileController::class, 'edit'])->name('profile.edit');      Route::put('/profile', [ProfileController::class, 'update'])->name('profile.update');      Route::get('/profile/export/json', [ProfileController::class, 'exportJson'])->name('profile.export.json');      Route::get('/profile/export/xml', [ProfileController::class, 'exportXml'])->name('profile.export.xml');  });  **Login.blade.php**  @extends('layouts.app')  @section('content')  <div class="auth-container">      <div class="auth-card">          <div class="auth-header">              <div class="logo-container">                  <i class="fas fa-graduation-cap"></i>              </div>              <h2>Welcome Back!</h2>              <p>Sign in to continue to your student portal</p>          </div>          <form method="POST" action="{{ route('login') }}" class="auth-form">              @csrf                <div class="form-group">                  <label for="email">                      <i class="fas fa-envelope"></i>                      Email Address                  </label>                  <div class="input-group">                      <input type="email" id="email" name="email" value="{{ old('email') }}" required                             placeholder="Enter your email">                  </div>                  @error('email')                      <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                  @enderror              </div>              <div class="form-group">                  <label for="password">                      <i class="fas fa-lock"></i>                      Password                  </label>                  <div class="input-group password-input">                      <input type="password" id="password" name="password" required                             placeholder="Enter your password">                      <button type="button" class="toggle-password">                          <i class="fas fa-eye"></i>                      </button>                  </div>                  @error('password')                      <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                  @enderror              </div>              <div class="form-options">                  <label class="checkbox-label">                      <input type="checkbox" name="remember" {{ old('remember') ? 'checked' : '' }}>                      <span>Remember me</span>                  </label>                  @if (Route::has('password.request'))                      <a href="{{ route('password.request') }}" class="forgot-password">                          Forgot Password?                      </a>                  @endif              </div>              <button type="submit" class="btn-auth">                  <i class="fas fa-sign-in-alt"></i>                  Sign In              </button>          </form>          <div class="auth-footer">              <p>Don't have an account? <a href="{{ route('register') }}">Register Now</a></p>          </div>      </div>  </div>  <style>  :root {      --primary-color: #1a237e;      --secondary-color: #0d47a1;      --accent-color: #2962ff;      --gray: #546e7a;      --gray-light: #eceff1;      --white: #ffffff;      --error: #d32f2f;      --success: #2e7d32;      --transition: all 0.3s ease;  }  \* {      margin: 0;      padding: 0;      box-sizing: border-box;  }  html, body {      height: 100%;      margin: 0;      padding: 0;  }  body {      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;      line-height: 1.6;      color: var(--primary-color);      background: var(--gray-light);      overflow-y: auto;  }  .auth-container {      min-height: 100vh;      width: 100%;      display: flex;      align-items: center;      justify-content: center;      background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));      padding: 2rem 1rem;  }  .auth-card {      background: var(--white);      padding: 2.5rem;      border-radius: 24px;      box-shadow: 0 20px 40px rgba(0, 0, 0, 0.2);      width: 100%;      max-width: 480px;      animation: slideUp 0.5s ease;      margin: 2rem 0;  }  .logo-container {      width: 85px;      height: 85px;      background: linear-gradient(135deg, var(--primary-color), var(--accent-color));      border-radius: 50%;      display: flex;      align-items: center;      justify-content: center;      margin: 0 auto 1.5rem;      box-shadow: 0 8px 20px rgba(0, 0, 0, 0.15);      transition: var(--transition);  }  .logo-container:hover {      transform: scale(1.05);      box-shadow: 0 12px 25px rgba(0, 0, 0, 0.2);  }  .logo-container i {      font-size: 2.5rem;      color: var(--white);  }  .auth-header {      text-align: center;      margin-bottom: 2rem;  }  .auth-header h2 {      color: var(--primary-color);      font-size: 1.8rem;      margin-bottom: 0.5rem;      font-weight: 600;  }  .auth-header p {      color: var(--gray);      font-size: 1rem;  }  .auth-form {      display: grid;      gap: 1.25rem;  }  .form-group {      display: flex;      flex-direction: column;      gap: 0.5rem;  }  .form-group label {      color: var(--primary-color);      font-weight: 500;      display: flex;      align-items: center;      gap: 0.5rem;      font-size: 1rem;  }  .form-group label i {      color: var(--accent-color);  }  .input-group {      position: relative;  }  .form-group input {      width: 100%;      padding: 0.9rem 1.1rem;      border: 2px solid var(--gray-light);      border-radius: 12px;      font-size: 1rem;      transition: var(--transition);      background: var(--white);  }  .form-group input:focus {      border-color: var(--accent-color);      box-shadow: 0 0 0 4px rgba(41, 98, 255, 0.1);      outline: none;  }  .form-group input::placeholder {      color: var(--gray);      opacity: 0.7;  }  .password-input {      position: relative;  }  .toggle-password {      position: absolute;      right: 1.25rem;      top: 50%;      transform: translateY(-50%);      background: none;      border: none;      color: var(--gray);      cursor: pointer;      padding: 0;      transition: var(--transition);      font-size: 1.1rem;  }  .toggle-password:hover {      color: var(--accent-color);  }  .form-options {      display: flex;      justify-content: space-between;      align-items: center;      font-size: 0.95rem;      margin: 0.25rem 0;  }  .checkbox-label {      display: flex;      align-items: center;      gap: 0.5rem;      color: var(--gray);      cursor: pointer;  }  .checkbox-label input[type="checkbox"] {      width: 18px;      height: 18px;      accent-color: var(--accent-color);      cursor: pointer;  }  .forgot-password {      color: var(--accent-color);      text-decoration: none;      transition: var(--transition);      font-weight: 500;  }  .forgot-password:hover {      color: var(--primary-color);      text-decoration: underline;  }  .btn-auth {      background: linear-gradient(135deg, var(--primary-color), var(--accent-color));      color: var(--white);      padding: 0.9rem;      border: none;      border-radius: 12px;      font-size: 1rem;      font-weight: 500;      cursor: pointer;      display: flex;      align-items: center;      justify-content: center;      gap: 0.5rem;      transition: var(--transition);      margin-top: 1rem;      box-shadow: 0 4px 15px rgba(41, 98, 255, 0.2);  }  .btn-auth:hover {      transform: translateY(-2px);      box-shadow: 0 8px 20px rgba(41, 98, 255, 0.3);  }  .btn-auth:active {      transform: translateY(0);  }  .auth-footer {      text-align: center;      margin-top: 2rem;      color: var(--gray);      font-size: 0.95rem;  }  .auth-footer a {      color: var(--accent-color);      text-decoration: none;      font-weight: 500;      transition: var(--transition);  }  .auth-footer a:hover {      color: var(--primary-color);      text-decoration: underline;  }  .error-message {      color: var(--error);      font-size: 0.9rem;      display: flex;      align-items: center;      gap: 0.4rem;      margin-top: 0.25rem;  }  @keyframes slideUp {      from {          opacity: 0;          transform: translateY(30px);      }      to {          opacity: 1;          transform: translateY(0);      }  }  @media (max-width: 768px) {      .auth-container {          padding: 1rem;          align-items: flex-start;      }      .auth-card {          padding: 2rem;          margin: 1rem 0;      }      .logo-container {          width: 70px;          height: 70px;          margin-bottom: 1rem;      }      .logo-container i {          font-size: 2rem;      }      .auth-header {          margin-bottom: 1.5rem;      }      .auth-header h2 {          font-size: 1.6rem;          margin-bottom: 0.25rem;      }      .auth-form {          gap: 1rem;      }      .form-group {          gap: 0.35rem;      }      .form-group input {          padding: 0.8rem 1rem;      }      .btn-auth {          padding: 0.8rem;          margin-top: 0.75rem;      }      .auth-footer {          margin-top: 1.5rem;      }  }  @media (max-height: 700px) {      .auth-container {          align-items: flex-start;      }      .auth-card {          margin: 1rem 0;      }      .logo-container {          width: 70px;          height: 70px;          margin-bottom: 1rem;      }      .logo-container i {          font-size: 2rem;      }      .auth-header {          margin-bottom: 1.5rem;      }      .auth-header h2 {          font-size: 1.6rem;          margin-bottom: 0.25rem;      }      .auth-form {          gap: 1rem;      }      .form-group {          gap: 0.35rem;      }      .form-group input {          padding: 0.8rem 1rem;      }      .btn-auth {          padding: 0.8rem;          margin-top: 0.75rem;      }      .auth-footer {          margin-top: 1.5rem;      }  }  </style>  <script>  document.querySelector('.toggle-password').addEventListener('click', function() {      const passwordInput = this.parentElement.querySelector('input');      const icon = this.querySelector('i');        if (passwordInput.type === 'password') {          passwordInput.type = 'text';          icon.classList.remove('fa-eye');          icon.classList.add('fa-eye-slash');      } else {          passwordInput.type = 'password';          icon.classList.remove('fa-eye-slash');          icon.classList.add('fa-eye');      }  });  </script>  @endsection  **Register.blade.php**  @extends('layouts.app')  @section('content')  <div class="auth-container">      <div class="auth-card">          <div class="auth-header">              <i class="fas fa-user-plus"></i>              <h2>Create Account</h2>              <p>Join our student community today</p>          </div>          <form method="POST" action="{{ route('register') }}" class="auth-form">              @csrf                <div class="form-row">                  <div class="form-group">                      <label for="name">                          <i class="fas fa-user"></i>                          Full Name                      </label>                      <input type="text" id="name" name="name" value="{{ old('name') }}" required                             placeholder="Enter your full name">                      @error('name')                          <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                      @enderror                  </div>                  <div class="form-group">                      <label for="student\_id">                          <i class="fas fa-id-card"></i>                          Student ID                      </label>                      <input type="text" id="student\_id" name="student\_id" value="{{ old('student\_id') }}" required                             placeholder="Enter your student ID">                      @error('student\_id')                          <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                      @enderror                  </div>              </div>              <div class="form-group">                  <label for="email">                      <i class="fas fa-envelope"></i>                      Email Address                  </label>                  <input type="email" id="email" name="email" value="{{ old('email') }}" required                         placeholder="Enter your email">                  @error('email')                      <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                  @enderror              </div>              <div class="form-row">                  <div class="form-group">                      <label for="password">                          <i class="fas fa-lock"></i>                          Password                      </label>                      <div class="password-input">                          <input type="password" id="password" name="password" required                                 placeholder="Create password">                          <button type="button" class="toggle-password">                              <i class="fas fa-eye"></i>                          </button>                      </div>                      @error('password')                          <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                      @enderror                  </div>                  <div class="form-group">                      <label for="password\_confirmation">                          <i class="fas fa-lock"></i>                          Confirm Password                      </label>                      <div class="password-input">                          <input type="password" id="password\_confirmation" name="password\_confirmation" required                                 placeholder="Confirm password">                          <button type="button" class="toggle-password">                              <i class="fas fa-eye"></i>                          </button>                      </div>                  </div>              </div>              <div class="form-group">                  <label for="department">                      <i class="fas fa-building"></i>                      Department                  </label>                  <select id="department" name="department" required>                      <option value="">Select Department</option>                      <option value="Computer Science" {{ old('department') == 'Computer Science' ? 'selected' : '' }}>Computer Science</option>                      <option value="Electrical Engineering" {{ old('department') == 'Electrical Engineering' ? 'selected' : '' }}>Electrical Engineering</option>                      <option value="Mechanical Engineering" {{ old('department') == 'Mechanical Engineering' ? 'selected' : '' }}>Mechanical Engineering</option>                      <option value="Civil Engineering" {{ old('department') == 'Civil Engineering' ? 'selected' : '' }}>Civil Engineering</option>                      <option value="Business Administration" {{ old('department') == 'Business Administration' ? 'selected' : '' }}>Business Administration</option>                  </select>                  @error('department')                      <span class="error-message"><i class="fas fa-exclamation-circle"></i> {{ $message }}</span>                  @enderror              </div>              <div class="terms-group">                  <label class="checkbox-label">                      <input type="checkbox" name="terms" required>                      <span>I agree to the <a href="#">Terms of Service</a> and <a href="#">Privacy Policy</a></span>                  </label>              </div>              <button type="submit" class="btn-auth">                  <i class="fas fa-user-plus"></i>                  Create Account              </button>          </form>          <div class="auth-footer">              <p>Already have an account? <a href="{{ route('login') }}">Sign In</a></p>          </div>      </div>  </div>  <style>  .auth-container {      min-height: calc(100vh - 80px);      display: flex;      align-items: center;      justify-content: center;      padding: 2rem;      background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));  }  .auth-card {      background: var(--white);      padding: 2.5rem;      border-radius: 20px;      box-shadow: 0 10px 30px rgba(0, 0, 0, 0.1);      width: 100%;      max-width: 600px;      animation: slideUp 0.5s ease;  }  .auth-header {      text-align: center;      margin-bottom: 2rem;  }  .auth-header i {      font-size: 3rem;      color: var(--primary-color);      margin-bottom: 1rem;  }  .auth-header h2 {      color: var(--primary-color);      font-size: 1.8rem;      margin-bottom: 0.5rem;  }  .auth-header p {      color: var(--gray);      font-size: 1rem;  }  .auth-form {      display: grid;      gap: 1.5rem;  }  .form-row {      display: grid;      grid-template-columns: 1fr 1fr;      gap: 1rem;  }  .form-group {      display: flex;      flex-direction: column;      gap: 0.5rem;  }  .form-group label {      color: var(--primary-color);      font-weight: 500;      display: flex;      align-items: center;      gap: 0.5rem;  }  .form-group label i {      color: var(--accent-color);  }  .form-group input,  .form-group select {      padding: 0.8rem 1rem;      border: 2px solid var(--gray-light);      border-radius: 10px;      font-size: 1rem;      transition: var(--transition);  }  .form-group input:focus,  .form-group select:focus {      border-color: var(--accent-color);      box-shadow: 0 0 0 3px rgba(112, 119, 161, 0.1);      outline: none;  }  .password-input {      position: relative;  }  .toggle-password {      position: absolute;      right: 1rem;      top: 50%;      transform: translateY(-50%);      background: none;      border: none;      color: var(--gray);      cursor: pointer;      padding: 0;      transition: var(--transition);  }  .toggle-password:hover {      color: var(--accent-color);  }  .terms-group {      margin-top: 0.5rem;  }  .checkbox-label {      display: flex;      align-items: flex-start;      gap: 0.5rem;      color: var(--gray);      font-size: 0.9rem;  }  .checkbox-label input[type="checkbox"] {      width: 16px;      height: 16px;      margin-top: 0.2rem;      accent-color: var(--accent-color);  }  .checkbox-label a {      color: var(--accent-color);      text-decoration: none;      transition: var(--transition);  }  .checkbox-label a:hover {      color: var(--primary-color);      text-decoration: underline;  }  .btn-auth {      background: var(--primary-color);      color: var(--white);      padding: 1rem;      border: none;      border-radius: 10px;      font-size: 1rem;      font-weight: 500;      cursor: pointer;      display: flex;      align-items: center;      justify-content: center;      gap: 0.5rem;      transition: var(--transition);  }  .btn-auth:hover {      background: var(--secondary-color);      transform: translateY(-2px);      box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);  }  .auth-footer {      text-align: center;      margin-top: 2rem;      color: var(--gray);  }  .auth-footer a {      color: var(--accent-color);      text-decoration: none;      font-weight: 500;      transition: var(--transition);  }  .auth-footer a:hover {      color: var(--primary-color);      text-decoration: underline;  }  .error-message {      color: var(--error);      font-size: 0.9rem;      display: flex;      align-items: center;      gap: 0.3rem;  }  @keyframes slideUp {      from {          opacity: 0;          transform: translateY(20px);      }      to {          opacity: 1;          transform: translateY(0);      }  }  @media (max-width: 768px) {      .auth-container {          padding: 1rem;      }      .auth-card {          padding: 2rem;      }      .form-row {          grid-template-columns: 1fr;      }  }  </style>  <script>  document.querySelectorAll('.toggle-password').forEach(button => {      button.addEventListener('click', function() {          const passwordInput = this.parentElement.querySelector('input');          const icon = this.querySelector('i');            if (passwordInput.type === 'password') {              passwordInput.type = 'text';              icon.classList.remove('fa-eye');              icon.classList.add('fa-eye-slash');          } else {              passwordInput.type = 'password';              icon.classList.remove('fa-eye-slash');              icon.classList.add('fa-eye');          }      });  });  </script>  @endsection |

|  |
| --- |
| Outputs  123456789101112131415 |