KATHFORD
INTERNATIONAL COLLEGE OF ENGINEERING & MANAGEMENT | AFFILIATED TO TRIBHUVAN UNIVERSITY

Lab Report 8

User Authentication and Session Management

Course: CSC381 - E-Commerce

Submitted by:

Parakram Kharel
Roll No: 24

Kathford International College of Engineering and Management
Affiliated to Tribhuvan University

August 31, 2025

1. Objective

To develop a secure login and registration system that allows customers to register, log in, and maintain sessions during their visit to the e-commerce website.

2. Tools and Technologies Used

Technology	Purpose
HTML5	Admin pages and form structure
CSS3	Layout and visual styling
ReactJS	Inline confirmations and UI behavior
PHP(Laravel)	Server-side logic, CRUD
MySQL	Persistent storage for catalog and orders
Caddy	Web server and local hosting
Bootstrap 5	Admin UI, tables, forms, responsive grid
VS Code	Development environment
Browser DevTools	Testing and debugging

3. Theory / Background

User authentication ensures that only registered users can access protected pages (like the shopping cart or checkout). PHP sessions are used to remember the logged-in user while they browse. The main components include:

- Registration: Collects user details and stores them in a MySQL database (with password hashing).
- Login: Validates entered credentials and starts a session.
- Session Management: Keeps users logged in until they log out.

4. Page Layout Design

4.1 Registration Page

- Header - Title ”Customer Registration”
- Input Fields (Name, Email, Password,Contact No, Date of Birth)
- Email Confirmation

- Submit Button: ”Register”
- Link: Already have an account? -¿ Login Page

4.2 Login Page

- Header - Title ”Customer Login”
- Input Fields (Email, Password)
- Submit Button: ”Login”
- Link: Don’t have an account? -¿ Registration Page

4.3 Dashboard

- Header - Title ”Dashboard”
- Navigation Bar
- Content Area

5. Code Snippets

5.1 Login Controller

```
<?php
  // Handle Login
  public function store(LoginRequest $request): RedirectResponse
  {
    $request->authenticate();
    $request->session()->regenerate();
    $user = Auth::user()->loadMissing('roles');
    $userRole = $user->roles->role ?? null;
    if (in_array($userRole, ['admin', 'superadmin'])) {
      return redirect()->intended(route('admin.dashboard', absolute:
      false));
    }
    return redirect()->intended(route('dashboard', absolute: false));
  }
  // Display Login Page
  public function create(Request $request): Response
  {
    return Inertia::render('auth/login', [
      'canResetPassword' => Route::has('password.request'),
      'status' => $request->session()->get('status'),
    ]);
  }

?>
```

5.2 Register Controller

```
<?php
// Handle Register
class Customer extends Model{
    protected $table = 'customers';
    protected $primaryKey = 'id';
    protected $fillable = [
        'name',
        'contact',
        'email',
        'address',
        'dob',
        'user_id',
        'avatar_id',
    ];

    public function user(): BelongsTo    {
        return $this->belongsTo(User::class);
    }

    public function avatar(): BelongsTo    {
        return $this->belongsTo(Avatar::class);
    }
}

public function create(): Response    {
    return Inertia::render('auth/register');
}

public function store(Request $request): RedirectResponse  {
  $request->validate([
      'name' => 'required|string|max:255',
      'contact' => 'required|string|max:255',
      'email' =>
      'required|string|lowercase|email|max:255|unique:'.User::class,
      'address' => 'required|string|max:255',
      'dob' => 'required|date',
  ]);
  $customerRole = \App\Models\Roles::where('role', 'customer')->first();
  $user = User::create([
      'name' => $request->name,
      'email' => $request->email,
      'password' => Hash::make(Str::random(16)), // random fake pass
      'role_id' => $customerRole ? $customerRole->id : null,
  ]);
  $customer = Customer::create([
      'name' => $request->name,
      'contact' => $request->contact,
      'email' => $request->email,
      'address' => $request->address,
      'dob' => $request->dob,
      'user_id' => $user->id,
  ]);

  event(new Registered($user));
  Password::sendResetLink(['email' => $user->email]);
  return redirect()->route('login')->with('status',
    'Registration successful! Please check your email to set your
    password.');
}
?>
```

5.3 Login Page

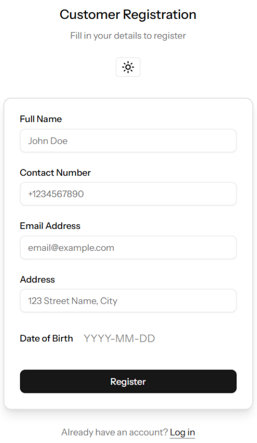
```
<form className="space-y-6" onSubmit={submit}>
  <div className="space-y-2">
      <Label htmlFor="email">
          Email address
      </Label>
      <div className="relative">
          <Mail className="absolute left-3 top-1/2 h-4 w-4 -translate-y-1/2
          text-gray-400" />
          <Input id="email" type="email" required autoFocus tabIndex={1}
          autoComplete="email" value={data.email}
          onChange={(e) => setData('email', e.target.value)}
          placeholder="email@example.com"
          />
      </div>
      <InputError message={errors.email} />
  </div>

  <div className="space-y-2">
      <div className="flex items-center justify-between">
          <Label htmlFor="password">
```

```
              Password
          </Label>
          {canResetPassword && (
              <TextLink href={route('password.request')} tabIndex={5}>
                  Forgot password?
              </TextLink>
          )}
      </div>
      <div className="relative">
          <Lock />
          <Input id="password" type={showPassword ? "text" : "password"}
              required tabIndex={2} autoComplete="current-password"
              value={data.password}
              onChange={(e) => setData('password', e.target.value)}
              placeholder="Enter your password"
          />
          <button
              type="button"
              onClick={() => setShowPassword(!showPassword)}
          >
              {showPassword ? <EyeOff className="h-4 w-4" /> : <Eye
              className="h-4 w-4" />}
          </button>
      </div>
      <InputError message={errors.password} />
  </div>

  <div className="flex items-center space-x-3">
      <Checkbox id="remember" name="remember" checked={data.remember}
          onClick={() => setData('remember', !data.remember)}
          tabIndex={3}
      />
      <Label htmlFor="remember">
          Remember me
      </Label>
  </div>

  <Button type="submit" tabIndex={4} disabled={processing}>
      {processing && <LoaderCircle />}
      {processing ? 'Signing in...' : 'Sign in'}
  </Button>
</form>
```

5.4 Register Page

```
<form onSubmit={submit}>
  <Label htmlFor="name">Full Name</Label>
  <Input id="name" type="text" required value={data.name} onChange={e => setData('name', e.target.value)} disabled={processing} placeholder="John Doe" />
  <InputError message={errors.name} />

  <Label htmlFor="contact">Contact Number</Label>
  <Input id="contact" type="text" required value={data.contact} onChange={e => setData('contact', e.target.value)} disabled={processing} placeholder="+1234567890" />
  <InputError message={errors.contact} />

  <Label htmlFor="email">Email Address</Label>
  <Input id="email" type="email" required value={data.email} onChange={e => setData('email', e.target.value)} disabled={processing} placeholder="email@example.com" />
  <InputError message={errors.email} />

  <Label htmlFor="address">Address</Label>
  <Input id="address" type="text" required value={data.address} onChange={e => setData('address', e.target.value)} disabled={processing} placeholder="123 Street Name, City" />
  <InputError message={errors.address} />

  <Label htmlFor="dob">Date of Birth</Label>
  <DatePicker
    id="dob"
    selected={data.dob ? new Date(data.dob) : null}
    onChange={d => setData('dob', d ? d.toISOString().split('T')[0] : '')}
    maxDate={new Date()}
    dateFormat="yyyy-MM-dd"
    placeholderText="YYYY-MM-DD"
    required
  />
  <InputError message={errors.dob} />

  <Button type="submit">
    Register
  </Button>

  <div>
    Already have an account? <TextLink href={route('login')}>Log in</TextLink>
  </div>
</form>
```

6. Output / Screenshots



Customer Registration

Fill in your details to register

Full Name

John Doe

Contact Number

+1234567890

Email Address

email@example.com

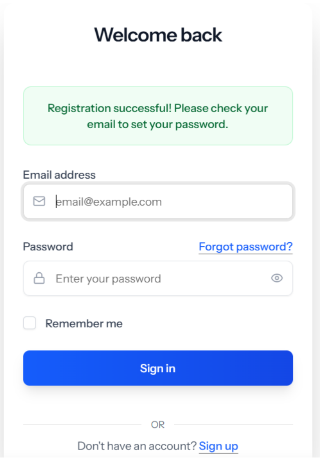
Address

123 Street Name, City

Date of Birth YYYY-MM-DD

Already have an account? Log in

Figure 1: Register Page.



Welcome back

Registration successful! Please check your email to set your password.

Email address

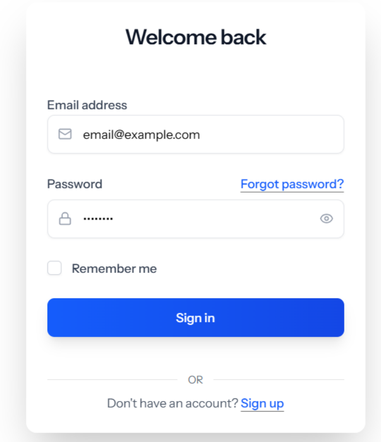
email@example.com

Password

Enter your password

Don't have an account? Sign up

Figure 2: Register Success.



Welcome back

Email address

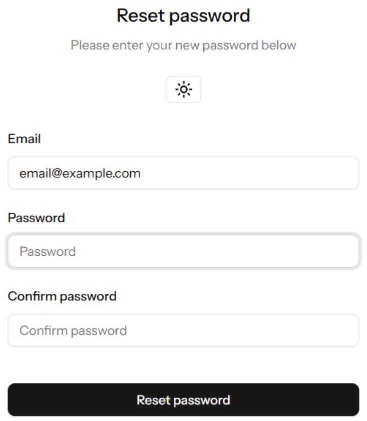
email@example.com

Password

••••••••

Don't have an account? Sign up

Figure 3: Login Page.



Reset password

Please enter your new password below

Email

email@example.com

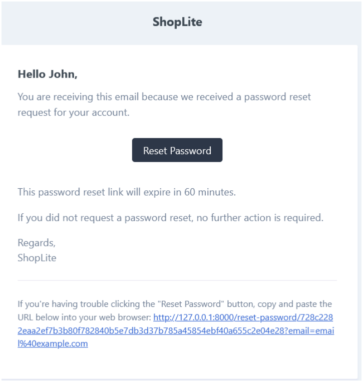
Password

Password

Confirm password

Confirm password

Figure 4: Reset Password Page.



ShopLite

Hello John,

You are receiving this email because we received a password reset request for your account.

This password reset link will expire in 60 minutes.

If you did not request a password reset, no further action is required.

Regards,
ShopLite

If you're having trouble clicking the "Reset Password" button, copy and paste the URL below into your web browser: http://127.0.0.1:8000/reset-password/728c2282eaa2ef7b3b80f782840b5e7db3d37b785a45854ebf40a655c2e04e28?email=email%40example.com

Figure 5: Password Reset Email.

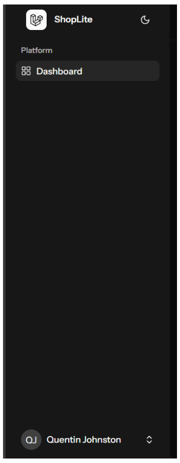


Figure 6: Dashboard after login.

7. Result

In this lab, a complete user authentication system was successfully implemented using PHP and MySQL.

8. Conclusion

The lab demonstrated the core concept of authentication and session management, which are essential in secure web applications.

9. References

- PHP mysqli Prepared Statements
- Laravel Authentication
- Bootstrap 5 Documentation
- MDN HTTP Reference
- Caddy Web Server Documentation