A computer screen shot of a program

Description automatically generated

**Phương thức Index trong Controller:**

csharp

Copy code

[HttpPost]

[ValidateAntiForgeryToken]

public ActionResult Index(LoginModel model)

{

// Thực hiện kiểm tra đăng nhập

var result = new AccountModel().Login(model.UserName, model.Password);

if (result && ModelState.IsValid)

{

// Kiểm tra có giữ session không, nếu có thì xoá đi để tạo session mới

var existingSession = SessionHelper.GetSession();

if (existingSession != null)

{

Session.Abandon();

}

// Set session mới

SessionHelper.SetSession(new UserSession() { UserName = model.UserName });

// Chuyển hướng đến trang Index của controller Home

return RedirectToAction("Index", "Home");

}

else

{

// Logic xử lý khi đăng nhập không thành công

ModelState.AddModelError("", "Tên đăng nhập hoặc mật khẩu không đúng.");

return View(model);

}

}

HttpPost: Thuộc tính này chỉ định rằng phương thức chỉ sẽ xử lý yêu cầu HTTP POST. Điều này liên quan đến form HTML mà người dùng sẽ sử dụng để đăng nhập.

ValidateAntiForgeryToken: Một thuộc tính bảo vệ chống giả mạo để đảm bảo rằng yêu cầu POST phải chứa token chống giả mạo hợp lệ để ngăn chặn tấn công CSRF.

Phương thức nhận một tham số là LoginModel, đại diện cho thông tin đăng nhập (ví dụ: UserName, Password).

Thực hiện kiểm tra đăng nhập bằng cách gọi phương thức Login từ AccountModel. Nếu đăng nhập thành công và ModelState cũng hợp lệ, thực hiện các bước tiếp theo.

Kiểm tra xem có phiên đăng nhập nào đang tồn tại hay không. Nếu có, xoá phiên đó để tạo một phiên đăng nhập mới.

Thiết lập phiên đăng nhập mới bằng cách sử dụng SessionHelper.

Chuyển hướng đến trang Index của controller Home nếu đăng nhập thành công.

Nếu đăng nhập không thành công hoặc ModelState không hợp lệ, thêm một lỗi vào ModelState và trả về view đăng nhập với các thông tin đăng nhập hiện tại (model).

Lớp SessionHelper:

csharp

Copy code

public class SessionHelper

{

public static void SetSession(UserSession session)

{

HttpContext.Current.Session["loginSession"] = session;

}

public static UserSession GetSession()

{

var session = HttpContext.Current.Session["loginSession"];

if (session == null)

return null;

else

{

return session as UserSession;

}

}

}

SetSession: Phương thức này được sử dụng để đặt giá trị của phiên (UserSession) vào session, sử dụng HttpContext.Current.Session.

GetSession: Phương thức này được sử dụng để lấy giá trị của phiên từ session. Nếu không có phiên, nó trả về null.

Lớp UserSession:

csharp

Copy code

public class UserSession

{

public string UserName { get; set; }

}

Một lớp đơn giản chứa thông tin về phiên đăng nhập, trong trường hợp này chỉ là UserName.

Lưu ý rằng mã này sử dụng Session để lưu trữ thông tin đăng nhập. Đối với các ứng dụng lớn hơn, có thể cần xem xét các phương pháp lưu trữ phiên đăng nhập an toàn hơn như Identity Framework.

\*\*\*Lỗi khi không chịu sửa form chuyển qua BeginForm (xoá form của html)