I. Auth

1. Định nghĩa

- Là hệ thống xác thực người dùng của Django. Nó hỗ trợ quản lý tài khoản người dùng, nhóm, quyền và phiên người dùng dựa trên cookie

2. Ứng dụng

- Hệ thống này có 2 chức năng chính là xác minh người dung và phân quyền (các tác vụ mà mỗi người dùng có thể làm). Cấu trúc của hệ thống xác thực bao gồm:

+ Người dùng: Tạo tài khoản, lưu trữ và chỉnh sửa thông tin được nhận

+ Quyền: (có/không) chỉ định liệu người dùng có thể thực hiện một tác vụ nhất định hay không.

+ Nhóm: Một cách chung để cung cấp quyền cho nhiều người dung thông qua phương pháp phân nhóm.

+ Hệ thống mật khẩu được bảo vệ thông qua thuật toán băm (biến văn bản gốc thành một chuỗi số và chữ cái khó hiểu)

3. Syntax

- Cú pháp xác thực cơ bản:



+ Trong chế độ đăng ký, cần lấy thông tin xác thực làm đối số:



+ Trong chế độ đăng nhập, cần có thông tin xác thực yêu cầu (tên người dùng, mật khẩu) để so sánh với tên người dùng và mật khẩu được lưu trữ.

- Cú pháp đăng nhập:



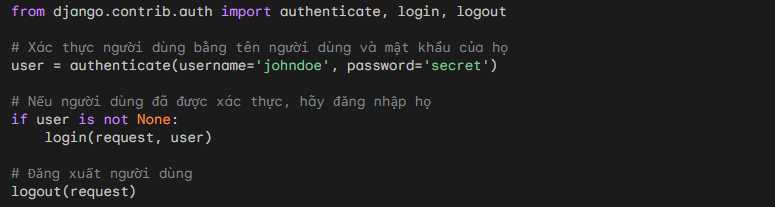


- Cú pháp đăng xuất

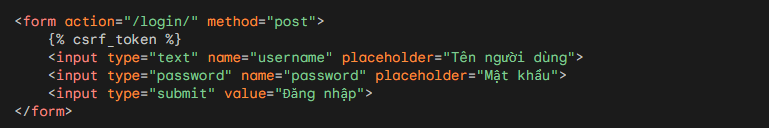


4. Ví dụ:

- Ví dụ về xác minh người dùng:



- Xác minh người dung được chèn vào biểu mẫu:



II. Validation

1. Định nghĩa

- Là chức năng xác thực trường và biểu mẫu được nhập vào của Django. Nó so sánh dữ liệu được nhận với các điều kiện cho trước và trả về **ValidationError** nếu dữ liệu nhập vào sai định dạng. Điều này rất quan trọng để ngăn ngừa lỗi và lỗ hổng bảo mật.

2. Ứng dụng

- Django cung cấp một số cách để xác thực dữ liệu, bao gồm:

+ Xác thực biểu mẫu: Các biểu mẫu Django cung cấp xác thực tích hợp cho các trường phổ biến, chẳng hạn như các trường bắt buộc, địa chỉ email và URL.

+ Xác thực mô hình: Các mô hình Django cung cấp xác thực tích hợp cho các loại trường phổ biến, chẳng hạn như CharField, IntegerField và DateTimeField.

+ Trình phân loại: Trình phân loại Django REST framework cung cấp xác thực tích hợp cho các kiểu dữ liệu phổ biến, chẳng hạn như chuỗi, số và ngày.

+ Chúng ta cũng có thể thêm xác thực tùy chỉnh của riêng mình vào biểu mẫu.

- Tạo định dạng mẫu cho các trường trong biểu mẫu

- Kiểm tra xem người dùng đã nhập đủ các trường bắt buộc và nhập đúng định dạng chưa

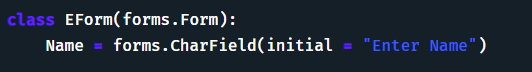
3. Syntax

- Cú pháp cơ bản với 1 trường:

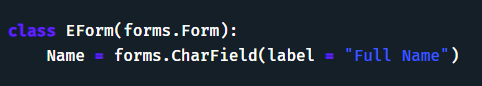


- 

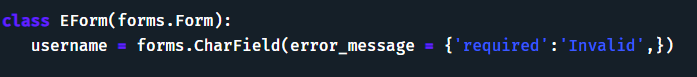
- Cú pháp cung cấp văn bản được nhập trước vào các trường:



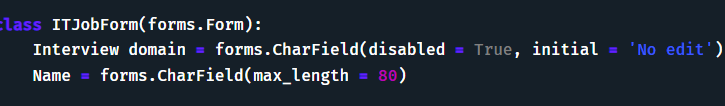
- Cú pháp thay đổi tên trường (mặc định là tên được chỉ định trong form.py):



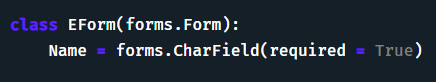
- Cú pháp điều chỉnh thông báo lỗi:



- Cú pháp tạo biểu mẫu không được chỉnh sửa:

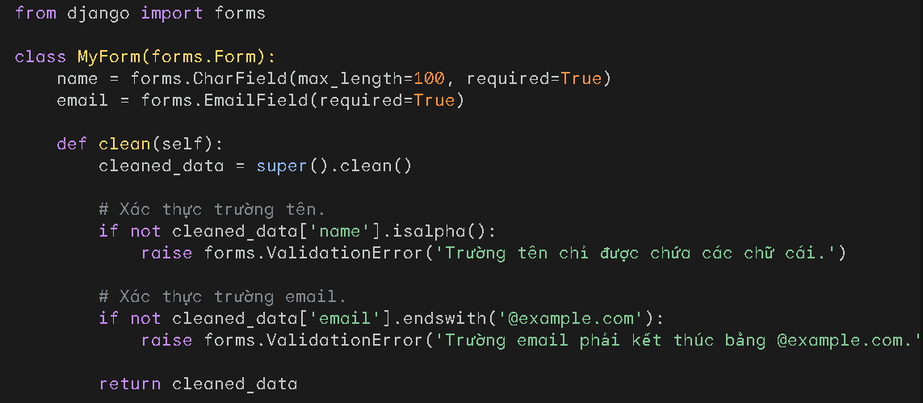


- Cú pháp bắt buộc điền nội dung vào trường:



4. Ví dụ

- Ví dụ về cách xác thực dữ liệu trong biểu mẫu Django (tên và email):



III. Session

1. Định nghĩa

- Session trong Django là một cơ chế lưu trữ thông tin trên phía máy chủ trong quá trình tương tác của người dùng với một ứng dụng web. Sessions được sử dụng để theo dõi trạng thái của người dùng, chẳng hạn như liệu họ đã đăng nhập hay chưa, họ đã thực hiện những tác vụ gì trong trang web.

- Sessions được triển khai bằng cookie. Khi một người dùng truy cập trang web Django lần đầu tiên, một cookie session sẽ được tạo và gửi đến trình duyệt của người dùng. Cookie chứa một ID session duy nhất, là một chuỗi ký tự ngẫu nhiên. Trình duyệt sau đó sẽ lưu cookie và gửi lại cho trang web Django trên mỗi yêu cầu tiếp theo.

- Để sử dụng sessions trong Django, bạn cần thêm lớp SessionMiddleware vào cài đặt MIDDLEWARE của mình. Bộ trung gian này sẽ xử lý việc tạo và quản lý cookie session.

2. Ứng dụng

- Theo dõi liệu người dùng đã đăng nhập hay chưa.

- Theo dõi các mặt hàng mà người dùng đã thêm vào giỏ hàng của họ.

- Theo dõi vị trí hiện tại của người dùng trong một biểu mẫu nhiều bước.

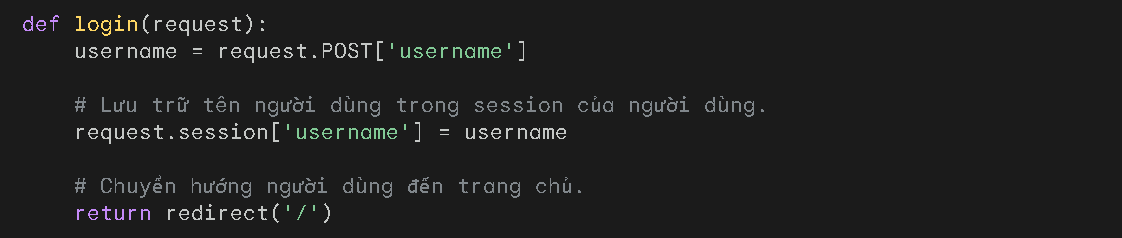
- Ghi nhớ sở thích ngôn ngữ của người dùng.

- Ghi nhớ lịch sử tìm kiếm của người dùng.

3. Syntax

4. Ví dụ

- Ví dụ lưu trữ phiên người dùng:



- Ví dụ về truy xuất người dung trong phiên hoạt động

