หัวข้อที่6

Django Template Language(DTL)เป็นเครื่องมือสำหรับสร้างหน้าเว็บแบบDynamic โดยโครงสร้างและไวยากรณ์พื้นฐานของDLT จะเป็นการผสมระหว่าง HTML และ DTL

โดยคุณสมบัติของDTLที่ใช้ในการทำงานจะมี variables , filters , tags

Variablesจะเป็นการแทรกตัวแปรเพื่อแสดงบนหน้าเว็บ รูปแบบ{{ variable\_name }}

Filters เป็นการแก้ไขและจัดรูปแบบค่าของตัวแปรก่อนนำไปแสดงผล รูปแบบ{{ variable\_name|upper }}

Tags เป็นตัวควบคุมLogic และ Flow ของTemplates รูปแบบ {% %}

โดยTagsมีอยู่หลายแบบแต่นี้อาจเป็นTagsที่จะเห็นบ่อยๆ

{% if %}กำหนดเงื่อนไข

{% for %}ทำซ้ำ(Loop)

{% block %}เป็นการกำหนดblockของเนื้อหา

{% extends %}เป็นการระบุบtemplateที่เป็นbaseเพื่อสืบทอด (Verbina, 2025)

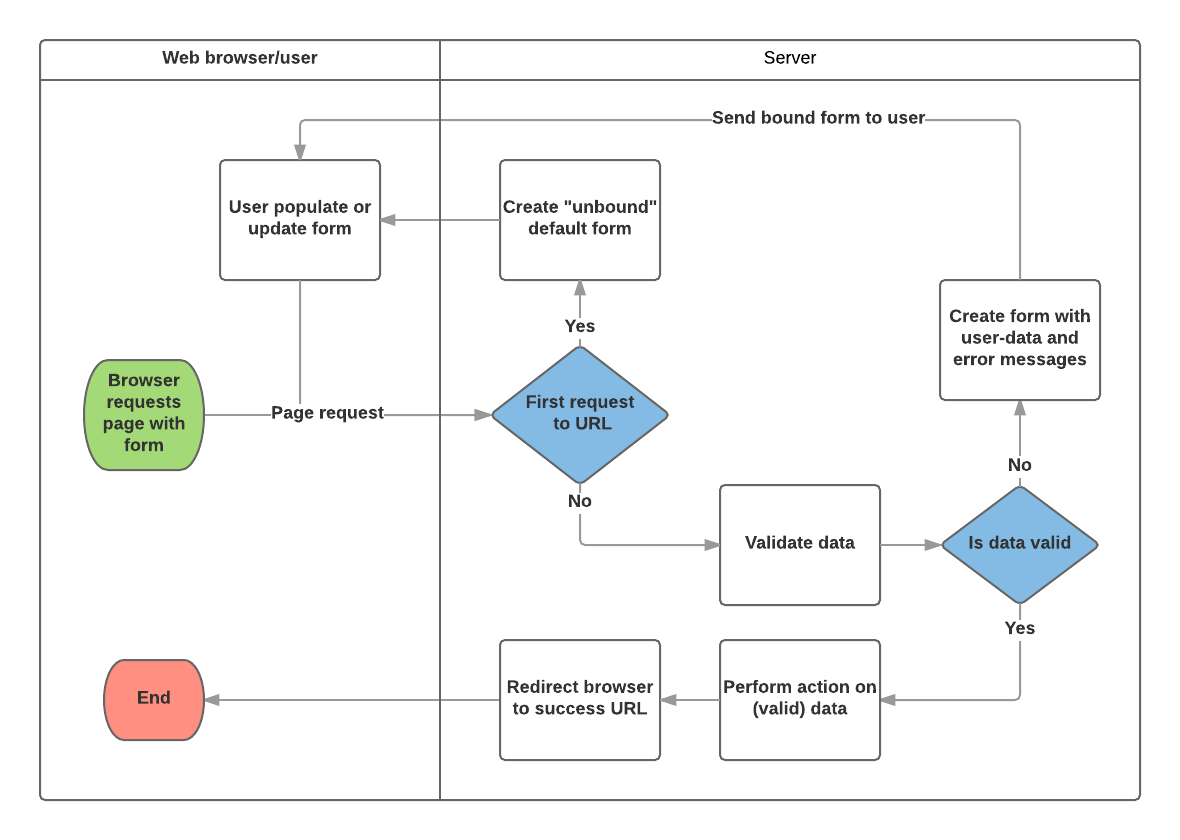
Django Formเป็นแบบฟอร์มที่ใช้สำหรับรับข้อมูลจากผู้ใช้งานในรูปแบบต่างๆ และใช้ข้อมูลนั้นดำเนินการกับฐานข้อมูล

โดยDjango แมปฟิลด์ที่กำหนดไว้ในformลงในอินพุตฟิลด์ของHTML และจัดการงาน3ส่วนที่ต่างกัน

1.จัดเตรียมและปรับโครงสร้างข้อมูลให้พร้อมสำหรับการเรนเดอร์

2.สร้างแบบฟอร์มHTML

3.รับและประมวลผลแบบฟอร์มและข้อมูลที่ส่งมาจากลูกค้า (geeksforgeeks, 2024)



ภาพจาก(https://developer.mozilla.org/en-US/docs/Learn\_web\_development/Extensions/Server-side/Django/Forms)

1.user request form ไปยังเซิฟเวอร์

2.มีการตรวจเช็คว่า request นี้เป็นครั้งแรกหรือเปล่า ถ้าใช้จากทำการส่งformเปล่ากลับไปให้user กรอกและ อัปเดตform

ถ้าไม่ใช่จะไปยังขั้นตอนถัดไป

3.เมื่อตรวจสอบว่าไม่ใช่ครั้งแรกทำการเซิฟเวอร์จะรับข้อมูลจากformและไปตรวจสอบ ถ้าข้อมูลผิดจะทำการส่งฟอร์มกลับไปให้userพร้อมกับบอกข้อความแจ้งเตือนข้อผิดพลาด

4.ในกรณีที่ข้อมูลถูกต้องจะดำเนินการตามข้อมูลที่ได้รับ และ นำผู้ใช้ไปยัง URLที่กำหนดไว้ จะเป็นการเสร็จสิ้นกระบวนการทำงาน

Django Authenticationคือระบบการยืนยันตัวตนที่ช่วยในการตรวจสอบและกำหนดสิทธิ์การเข้าถึงของผู้โดยจะตรวจสอบว่าผู้ใช้เป็นใครและกำหนดว่าผู้ใช้ที่ได้รับการตรวจสอบสิทธิ์จะได้รับอนุญาตให้ทำอะไรได้บ้าง

ในระบบนี้จะประกอบด้วย

ผู้ใช้ Users

สิทธิ์ Permissions ที่กำหนดว่าผู้ใช้สามารถทำอะไรได้บ้าง

กลุ่ม Groups การกำหนดสิทธิ์ผู้ใช้ที่มีมากกว่า1คน

การhashing password ที่สามารถกำหนดค่าได้

Form และ view สำหรับเข้าสู่ระบบหรือจำกัดเนื้อหาของuser

Authentication ของDjangoถูกออกแบบมาให้มีความยืดหยุ่นที่สูงจึงจะไม่มีฟีเจอร์บางอย่างที่มักจะพบในระบบยืนยันตัวตนของเว็บทั่วไป แต่อย่างไรก็ตาม มีการใช้แพ็คเกจจากนักพัฒนาภายนอกมาช่วยเสริมฟีเจอร์เหล่านี้เช่น

การตรวจสอบความแข็งแรงของรหัสผ่าน

การจำกัดจำนวนครั้งในการพยายามlogin

การloginผ่านบริการอื่น(OAuth)

การกำหนดสิทธิ์การเข้าถึงในระบบวัตถุ(Object-level permissions)

(Django, n.d.)

Django admin siteเป็นระบบที่จะทำการอ่านข้อมูลจากโมเดลเพื่อสร้างinterfaceที่รวดเร็วเพื่อจัดการกับเนื้อหาบนเว็บของคุณโดยจะมีการให้สร้างsuperuserเพื่อเป็นคนจัดการกับข้อมูลเหล่านั้น (Django, The Django admin site, n.d.)

การนำมาใช้งาน

ทำฟีเจอร์loginและregister

login.html

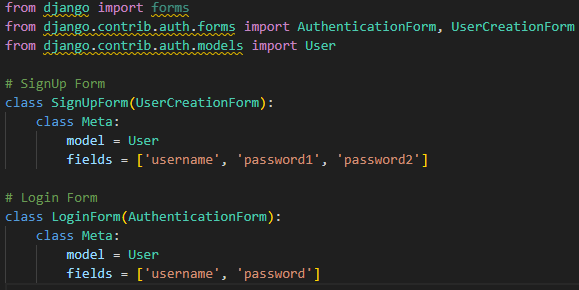


Sign\_up.html

A screen shot of a computer code

AI-generated content may be incorrect.

Forms.py

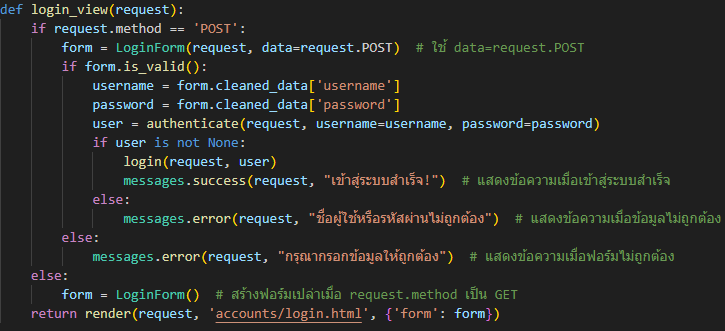


View.py(Sign\_up&Libary)

A computer screen with colorful text

AI-generated content may be incorrect.

View.py(Login)



Admin page

A screenshot of a computer

AI-generated content may be incorrect.

LoginPage

A blue and white rectangle

AI-generated content may be incorrect.

RegisterPage

A screenshot of a computer

AI-generated content may be incorrect.

# การอ้างอิง

Django. (ม.ป.ป.). *The Django admin site*. เข้าถึงได้จาก docs.djangoproject.com: https://docs.djangoproject.com/en/5.1/ref/contrib/admin/

Django. (ม.ป.ป.). *User authentication in Django*. เข้าถึงได้จาก docs.djangoproject.com: https://docs.djangoproject.com/en/5.1/topics/auth/

Evgenia Verbina. (5 2 2025). *The Ultimate Guide to Django Templates*. เข้าถึงได้จาก blog.jetbrains.com: https://blog.jetbrains.com/pycharm/2025/02/the-ultimate-guide-to-django-templates/

geeksforgeeks. (21 7 2024). *Django Forms*. เข้าถึงได้จาก geeksforgeeks.org: https://www.geeksforgeeks.org/django-forms/