

COMMIT : SHOW QUESTIONS IN ANSWER PAGE

ทำการสร้างหน้า home ใหม่ให้มี 2ปุ่ม คือ create quiz เพื่อเข้าไปสร้างคำถาม และ go to quiz เพื่อเข้าไปตอบคำถาม ซึ่งเมื่อกดเข้าไปที่ปุ่ม go to quiz จะเข้าไปที่หน้า answer page ซึ่งจะใช้ {% for q in show_quiz %} เพื่อเป็นการวนลูปเพื่อเข้าถึงข้อมูลใน show_quiz และ {{q.question}} เพื่อแสดงคำถาม

COMMIT : CAN SELECT CHOICE TO ANSWER QUIZ

ทำการแก้ไข model โดยแบ่งเป็น 2 model ได้แก่

1. Question ซึ่งมี 2 ส่วน คือ
 - question_text ซึ่งเอาเก็บคำถาม
 - ans เอาไว้เก็บคำตอบ
 2. Choice ซึ่งได้มีการใช้ ForeignKey เพื่อทำการเชื่อมกับ model Question และ 4 ส่วน ดังนี้
 - choice1 เอาไว้เก็บตัวเลือกที่ 1
 - choice2 เอาไว้เก็บตัวเลือกที่ 2
 - vote_T เอาไว้เก็บจำนวนคนที่ตอบคำถามถูก
 - vote_F เอาไว้เก็บจำนวนคนที่ตอบคำถามผิด
- หลังจากนั้นทำการแก้ไขไฟล์ views.py โดยใน

ฟังก์ชัน createquiz

- ได้ทำการกำหนดเงื่อนไข ถ้ามีการใช้ method POST ให้รับค่าจาก quizbox และ ans ซึ่งเป็น placeholder หลังจากนั้นใช้ q.save() ทำการเก็บข้อมูลเข้า database (ซึ่งจะอยู่ใน model Question)
- ทำการรับค่าจาก a และ b ซึ่งเป็น placeholder เช่นกัน และทำการกำหนด vote_T และ vote_F ให้เป็น 0 หลังจากนั้นกำหนด question=q ซึ่งเป็น ForeignKey และหลังจากนั้นทำการเก็บค่าเข้า database (ซึ่งจะอยู่ใน model Choice) และทำการ redirect ไปที่ <http://localhost:8000/quiz/> และกำหนด quiz_item ให้เป็นการเรียกค่าจาก object

ฟังก์ชัน goQuiz

- กำหนดให้ show_quiz เป็นตัวแปรที่เอาไว้เรียกค่าจาก object หลังจากนั้นทำการ render เทมเพลต answer

ฟังก์ชัน detail

- มีการรับค่า question_id และมีการใช้ try/except โดยใน try นั้น กำหนดให้ question เป็นตัวแปรที่เอาไว้เรียกค่าจากใน model Question ซึ่งจะเรียกค่ามาตาม question_id และ choice_item เป็นตัวแปรที่เอาไว้เรียกค่าจากใน model Choice
- ใน except จะทำการ raise และแสดงข้อความว่า Question does not exists หลังจากนั้นทำการ render เทมเพลต detail

ฟังก์ชัน vote

- มีการรับค่า question_id ซึ่งถ้ามีการใช้ method POST จะกำหนดให้ answer เก็บค่าที่มาจาก choice ซึ่งเป็น radio
- check_ans จะเรียกค่า จากใน model Question ที่เป็น คอลัมน์ ans ซึ่งได้เก็บคำตอบของคำถามเอาไว้
- vote จะทำการรับ question_id มา

- หลังจากนั้นทำการกำหนดเงื่อนไข ถ้า `answer` มีค่าตรงกับ `check_ans` จะทำการบวกค่า 1 และเก็บไว้ใน `vote_T` หลังจากนั้นจะทำการ render เทมเพลต `resultT` แต่ถ้าค่าไม่ตรงจะทำการบวกค่า 1 และเก็บไว้ใน `vote_F` หลังจากนั้นจะทำการ render เทมเพลต `resultF`

COMMIT : UPDATE FUNCTIONAL TEST

ทำการแก้ไข functional test ใหม่ในฟังก์ชัน `setUp` จะทำการเปิด firefox และ ฟังก์ชัน `test_can_start_app` จะทำการ เปิดหน้า <http://localhost:8000/quiz> และมีหน้าดังต่อไปนี้

- ทำการตรวจสอบว่ามีคำว่า Quiz อยู่ title หรือไม่
- กำหนดให้ `create_page_bt` คือปุ่มสำหรับกดไปหน้า สร้างคำถาม หลังจากนั้นทำการตรวจสอบว่า ค่า `value` ของปุ่ม คือคำว่า Create Quiz หรือไม่ หลังจากนั้นทำการ click ที่ปุ่ม
- ในหน้า Create quiz page ทำการตรวจสอบว่า มีคำว่า Create Quiz ที่ title หรือไม่และทำการตรวจสอบตัวหนังสือ โดยหาจาก tag H1 และตรวจสอบว่าเป็นคำว่า Quiz Test หรือไม่
- กำหนด `question_text` และตรวจสอบว่ามีแสดงคำว่า Create Quiz หรือไม่ หลังจากนั้นกำหนดให้ `question_box` คือช่องที่เอาไว้ใส่คำถามและทำการตรวจสอบว่าในช่องนั้นแสดงคำว่า enter you quiz หรือไม่ หลังจากนั้นทำการใส่คำว่า Fish is cat ลงไป
- กำหนดให้ `choice1_box` คือ ช่องการสร้างตัวเลือกที่ 1 และทำการตรวจสอบว่ามีคำว่า Enter your choice1 หรือไม่ หลังจากนั้นใส่ คำว่า True ลงไป
- กำหนดให้ `choice2_box` คือ ช่องการสร้างตัวเลือกที่ 2 และทำการตรวจสอบว่ามีคำว่า Enter your choice2 หรือไม่ หลังจากนั้นใส่ คำว่า False ลงไป
- กำหนด `ans_text` และตรวจสอบว่า แสดงคำว่า Create Answer หรือไม่
- กำหนด `ans_box` ให้เป็นช่องสำหรับสร้างคำตอบ และตรวจสอบว่ามีคำว่า Enter your answer หรือไม่ หลังจากนั้นใส่คำว่า False ลงไป แล้วกดปุ่ม submit
- กำหนด `go_to_quiz_bt` คือปุ่มที่มีไว้สำหรับกดไปหน้าตอบคำถาม โดยตรวจสอบค่า `value` ว่าใช่ Go to Quiz หรือไม่ หลังจากนั้นทำการกดปุ่ม
- หลังจากนั้นทำการเลือกคำถามโดยค้นหาจาก id
- เมื่อเข้ามาในหน้าตอบคำถามทำการเลือก choice หลังจากนั้นกดปุ่ม submit
- เมื่อเข้ามาในหน้า result ทำการตรวจสอบว่า มีคำว่า You answer True หรือไม่ หลังจากนั้นทำการกดปุ่ม `back_to_quiz_bt` เพื่อกลับไปหน้า home

COMMIT : REFACTOR UNIT TEST

มีการตรวจสอบว่ามีการใช้เทมเพลตหรือไม่ และใน class `itemModel` ทำการตรวจสอบว่าสามารถเก็บข้อมูลเข้า database ได้หรือไม่ โดยการสร้าง `first_item.question_text` เป็นการสร้างคำถามว่า Dog is animal และใช้ `first_item.ans` เป็นการสร้างคำตอบว่า True และทำการใช้ `first_item.save()` เพื่อเป็นการเก็บค่าเข้าไป หลังจากนั้นทำการสร้าง `second_item` และทำตามแบบที่ได้กล่าวไป หลังจากนั้นทำให้ `save_item` เป็นการเรียกค่าใน model `Question` และทำการตรวจสอบว่ามี 2 ค่าหรือไม่และทำการตรวจสอบว่าใน `question_text` มี Dog is animal กับ cat is dog หรือไม่ และตรวจสอบว่าใน `ans` มี True และ False หรือไม่ หลังจากนั้นทำการทดสอบ model `Choice` โดยการสร้าง `choice1` และ `choice2` หลังจากนั้นก็ทำการตรวจสอบว่าค่าที่เรียกมามีค่าตรงกับค่าที่ใส่ไปหรือไม่

COMMIT : UPDATE UNIT TEST AND FUNCTIONAL TEST

Functional test

- ใน functional test ทำการเพิ่มการตรวจสอบที่หน้าตอบคำถามโดยการตรวจสอบค่า choice ว่ามี value ตรงกับที่กำหนดหรือไม่

Unit test

- ในฟังก์ชัน test_can_post_question ทำการตรวจสอบว่าสามารถทำการ post คำถามได้หรือไม่และทำการตรวจสอบ object ว่ามีค่าเท่ากับ 1 หรือไม่
- ในฟังก์ชัน test_can_post_choice ทำการตรวจสอบว่าสามารถทำการ post ตัวเลือกได้หรือไม่และทำการตรวจสอบ object ว่ามีค่าเท่ากับ 1 หรือไม่
- ในฟังก์ชัน test_can_redirect_after_post จะมีการตรวจสอบการ redirect และมีการตรวจสอบสถานะว่าใช่ 302 หรือไม่