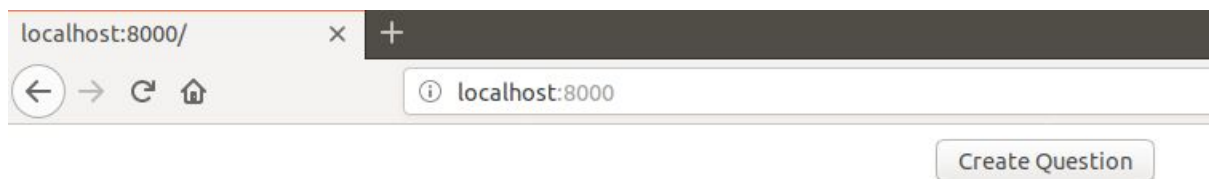


## commit 1

```
commit 0dbc12ab7fa597436e6935c094af50152c7d8646
Author: GreenT <treorey@gmail.com>
Date:   Fri Jun 15 14:49:23 2018 +0700
```

สร้าง application myapp ขึ้นมา สร้าง template ขึ้นมา 4 template ได้แก่

1.index.html เป็นหน้าแรกของ application ทำหน้าที่แสดงรายการคำถามทั้งหมดที่มี และมีปุ่มสร้างคำถาม



2.detail.html ทำหน้าที่แสดงคำถามและมีช่องใส่คำตอบ

3.results.html แสดงผลว่าคำตอบที่ได้ตอบไปถูกหรือผิด

4.create.html มีไว้เพื่อสร้างคำถามและคำตอบ

## commit 2

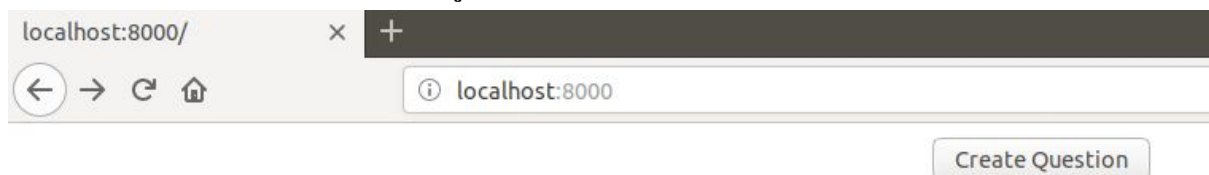
```
commit dd04b1a785021a9a5c757e9acb6d508aa115604e
Author: GreenT <treorey@gmail.com>
Date:   Fri Jun 15 15:11:41 2018 +0700

    edit template, so index template can show question and can click in to each
    question
```

แก้ไข template ให้แสดงผลตามที่ต้องการ โดยแต่ละ template แสดงผลดังนี้

## index.html

สามารถแสดงรายการกระทู้ได้

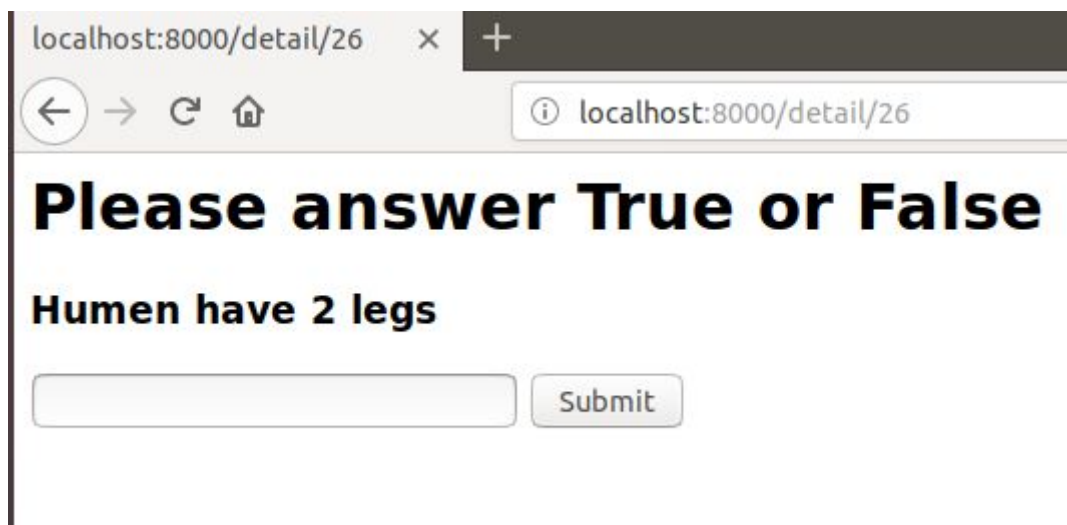


## Question

- Humen have 2 legs

## detail.html

มีช่องให้ตอบคำถาม สามารถส่งคำตอบไปได้



แยก template results.html เป็น results\_t.html กับ results\_f.html

results\_t.html

เมื่อตอบคำถามถูกจะ render มาที่ template นี้

results\_f.html

จะ render มาที่ template นี้เมื่อตอบคำถามผิด

ซึ่งทั้งสอง template ยังไม่สามารถแสดงผล vote ได้เนื่องจากติดปัญหาตรง model Vote ที่ไม่สามารถสร้างพร้อมกับการสร้างคำถามได้

```
from django.db import models

class Question(models.Model):
    question_text = models.CharField(max_length=200)
    choice_text = models.CharField(max_length=200)

class Vote(models.Model):
    connect = models.ForeignKey(Question, default=None, on_delete=models.CASCADE)
    vote_true = models.IntegerField(default=0)
    vote_false = models.IntegerField(default=0)
```

myapp/models.py

commit 3

```
commit 8ed6486309fefb3a210f9cf5560787ef8c1ebbfd
Author: GreenT <treorey@gmail.com>
Date: Fri Jun 15 19:35:01 2018 +0700

    Can create record of vote
```

สามารถสร้าง record vote ที่มี id ตรงกับคำถาม พร้อมกันกับตอนสร้างคำถามได้ โดยมีโค้ดดังนี้

```
def create(request):
    if request.method == 'POST':
        q = Question(question_text=request.POST.get('question', ''), choice_text=request.POST.get('answer', ''))
        q.save()

        v = Vote(vote_true=0, vote_false=0, connect=q)
        v.save()
        return redirect('/')
    items = Question.objects.all()
    return render(request, 'index.html', {'items': items})
```

## Question table

Table: myapp\_questio New Re

	id	question_text	choice_text
	Filter	Filter	Filter
1	26	Humen hav...	True

## Vote table

	id	vote_true	vote_false	connect_id
	Filter	Filter	Filter	Filter
1	19	0	0	26

## commit 4

```
commit 1142a4b93c9f4692f19f623251fa1e69c4e80496
Author: GreenT <treorey@gmail.com>
Date:   Fri Jun 15 20:02:07 2018 +0700

    Application can count vote
```

สามารถเก็บจำนวนคนตอบคำถามถูกและผิดได้ โดยจะแสดงในหน้า results\_t.html และ results\_f.html



## commit 5

```
commit dce0af661aa74dd404f1f2475a8ce0200d7e4d5c
Author: GreenT <treorey@gmail.com>
Date:   Fri Jun 15 23:44:04 2018 +0700

    unit test can test save item to Question table and Vote table
```

สร้าง unit test เพื่อทดสอบว่าสามารถเพิ่มคำถามและคะแนนโหวตได้จริงหรือไม่

## ตัวอย่างโค้ด unit test

```
class ItemModelTest(TestCase):
    def test_saving_and_retrieving_items(self):

        frist_question = Question()
        frist_question.question_text = 'Human have 2 eyes'
        frist_question.save()
        frist_answer = Question()

        second_question = Question()
        second_question.question_text = 'Human have 3 ears'
        second_question.save()
        second_answer = Question()

        save_item = Question.objects.all()
        self.assertEqual(save_item.count(),2)

        first_saved_item = save_item[0]
        second_saved_item =save_item[1]
        self.assertEqual(first_saved_item.question_text, 'Human have 2 eyes')
        self.assertEqual(second_saved_item.question_text, 'Human have 3 ears')
```

## ผลจากการ run unit test

```
greent@greent-X450JB:~/Documents/test_2$ python3 manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
..
-----
Ran 2 tests in 0.004s

OK
Destroying test database for alias 'default'...
```

## commit 6

```
commit e6347b6a08f2ef7a6d309090a3db2fb6318a5ab1 (HEAD -> master)
Author: GreenT <treorey@gmail.com>
Date: Sun Jun 17 12:29:55 2018 +0700

    finish functional test
```

สร้าง functional test เพื่อทดสอบว่าสามารถสร้างคำถาม ตอบคำถามและ application สามารถแสดงผลอย่างที่ต้องการได้หรือไม่



## ตัวอย่างโค้ด functional test

```
def test_can_start_a_list_and_retrieve_it_later(self):
    self.browser.get('http://localhost:8000')

    check_button_create = self.browser.find_element_by_name('create')
    self.assertEqual(check_button_create.get_attribute('value'), 'Create Question')
    time.sleep(3)

    check_button_create.click()
    check_url_create= self.browser.current_url
    self.assertRegex(check_url_create, '/createpage')
    time.sleep(2)

    input_question = self.browser.find_element_by_id('question')
    input_answer = self.browser.find_element_by_id('answer')

    self.assertEqual(input_question.get_attribute('placeholder'),'Enter your question')
    input_question.send_keys('Dog have 4 legs')

    self.assertEqual(input_answer.get_attribute('placeholder'),'Enter your answer')
    input_answer.send_keys('True')
    time.sleep(3)

    submit_button = self.browser.find_element_by_name('create_button')
    submit_button.click()
    time.sleep(2)

    find_text = self.browser.find_element_by_id('q2')

    self.assertEqual(find_text.get_attribute('name'), 'Dog have 4 legs')
    find_text.click()
    time.sleep(2)

    find_ans = self.browser.find_element_by_id('choice')
    find_ans.send_keys('True')
    submit_button2 = self.browser.find_element_by_id('create_ans')
    submit_button2.click()

    check_url_results= self.browser.current_url
    self.assertRegex(check_url_results, '/keepresults/*')
    time.sleep(2)
```

## ผลจากการ run functional test

```
greent@greent-X450JB:~/Documents/test_2$ python3 functional_tests.py
F
=====
FAIL: test_can_start_a_list_and_retrieve_it_later (__main__.NewVisitorTest)
-----
Traceback (most recent call last):
  File "functional_tests.py", line 59, in test_can_start_a_list_and_retrieve_it_later
    self.fail('Finish the test!')
AssertionError: Finish the test!
-----
Ran 1 test in 28.972s

FAILED (failures=1)
```