Django

Presenters:

Emad Aghayi

Anne Martine Augustine

What is Django?

- A web development framework for Python
- Is for perfectionists with deadline
- Suitable for
  - Dynamic and database driven websites
  - Content based websites
- Ex: Craigslist, Google AppEngine
- Named after famous Guitarist “Django Reinhardt”
- Open sourced and first version released 2008

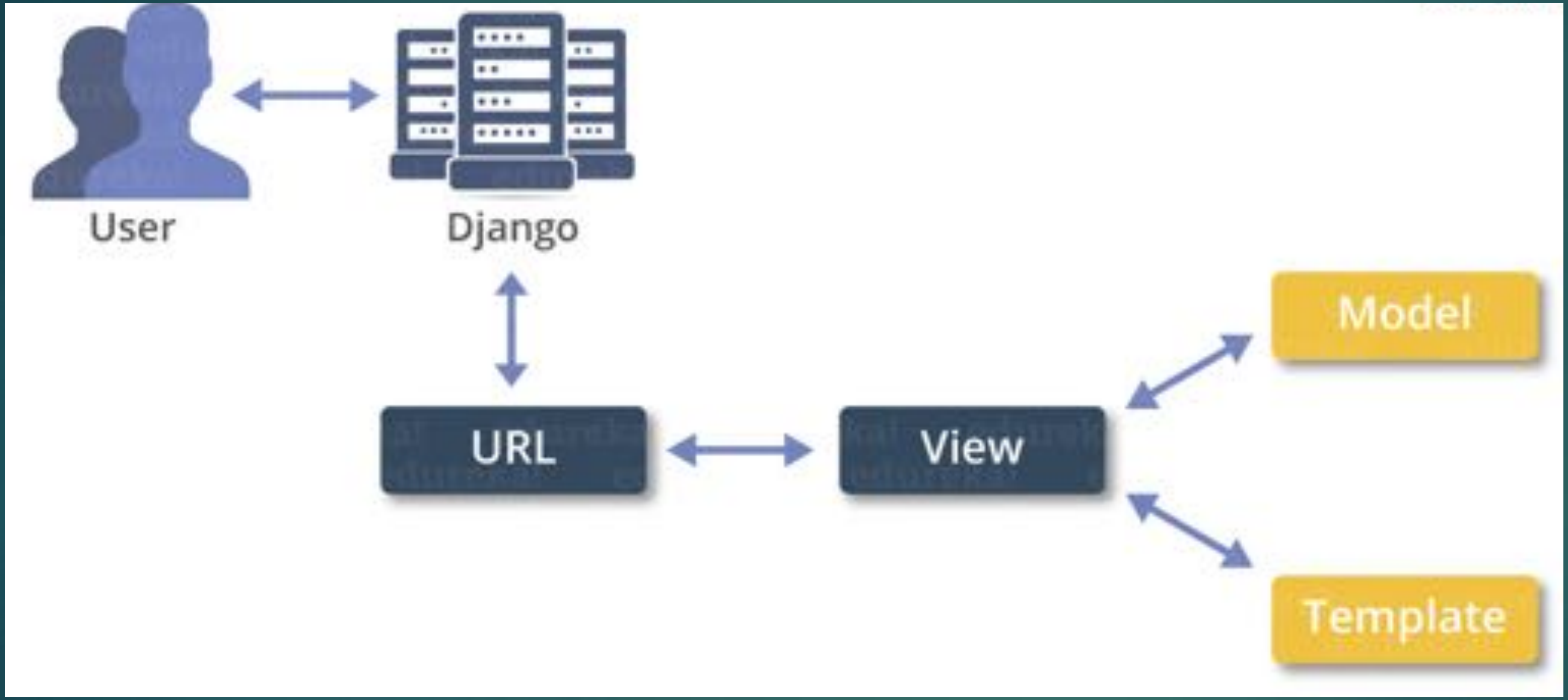
Architecture

- MVT Architecture and MVC Design Patern
  - Model: describes data strucure
  - Views: controls what a user sees
  - Template: How a user sees it
  - Controller: Django framework, URL parsing

Important Modules

- CRUD interface
- Authentication
- Form handler
- Session manager
- localiziation

Django Architecture



[1]: https://www.edureka.co/blog/django-tutorial/

Why Django? Advantages

- MVC design pattern: modulate the code
- Auto web admin to ease the website administration
- Avoid code duplication by HTML template (DRY Principle)
- Schema and setting is in python

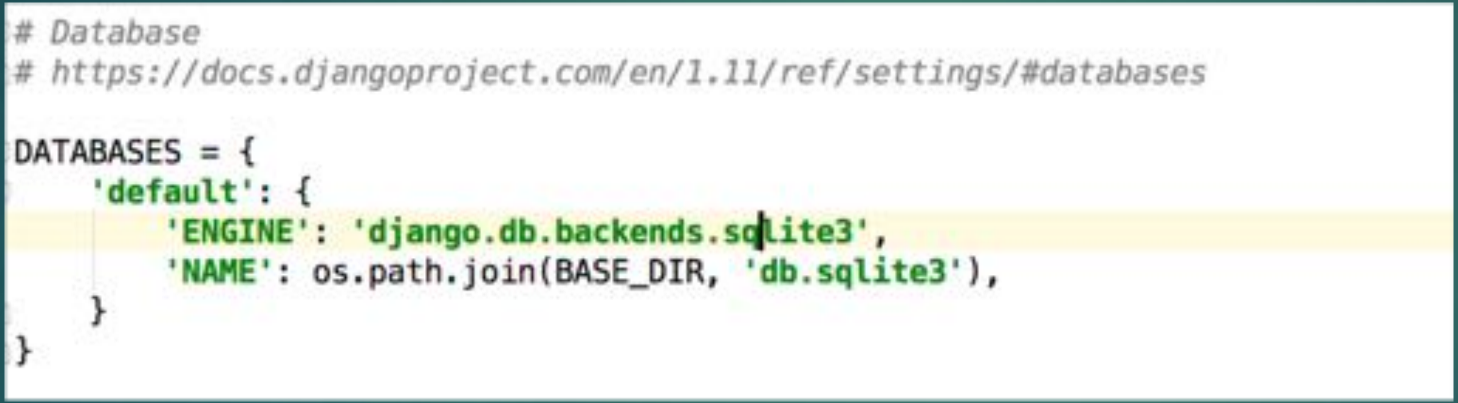
Demo

Steps:

1. Create a project
2. Start the application
3. Create the database
4. Define setting in Setting.py
5. Define models, templates, views
6. Create URL mapping
7. Deployment

Demo …

- You can do steps 1, 2 very easily vie PyCharm IDE
- Step3: Database configuration is done in Setting.py file

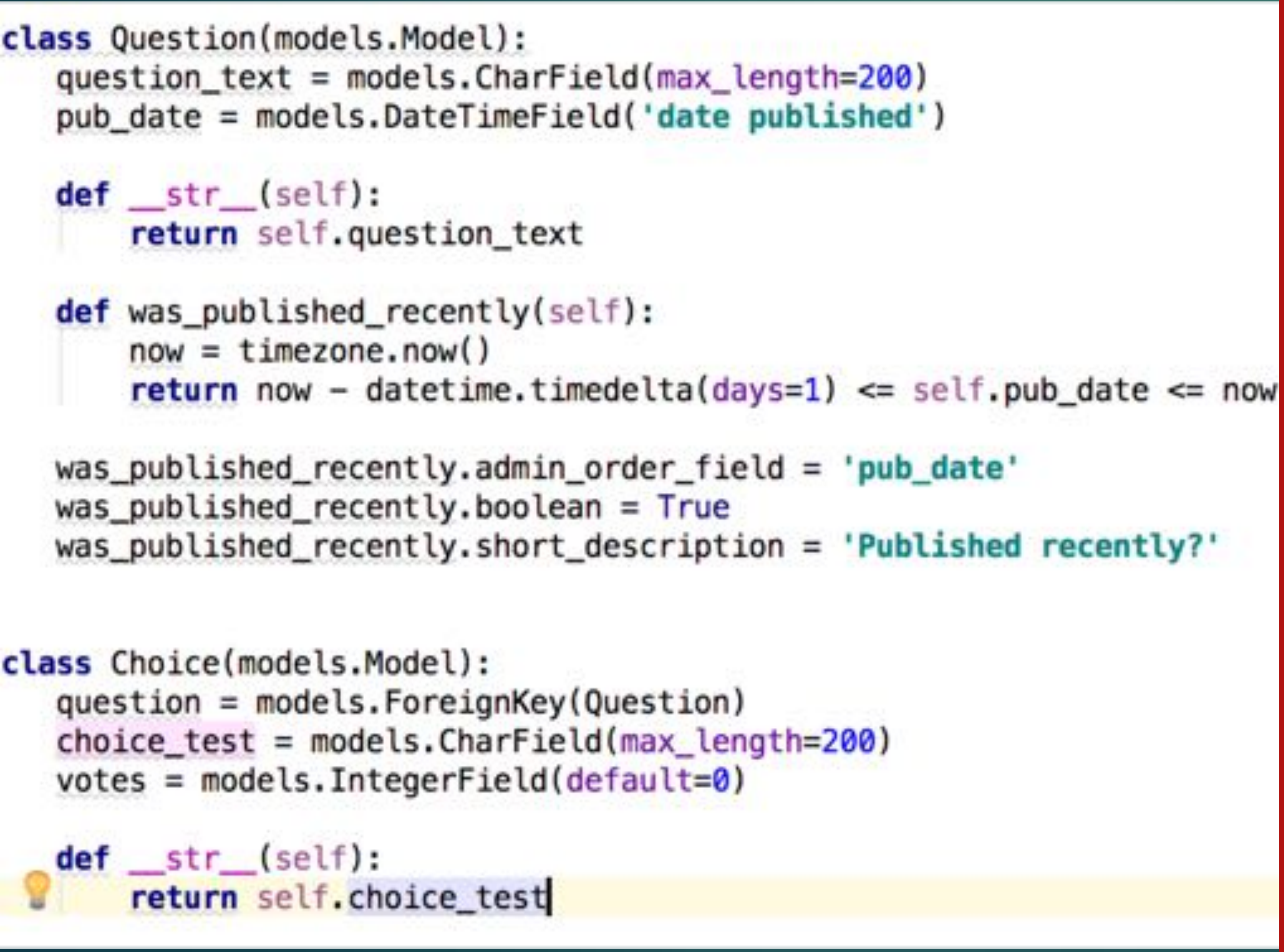


```
# Database
# https://docs.djangoproject.com/en/1.11/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

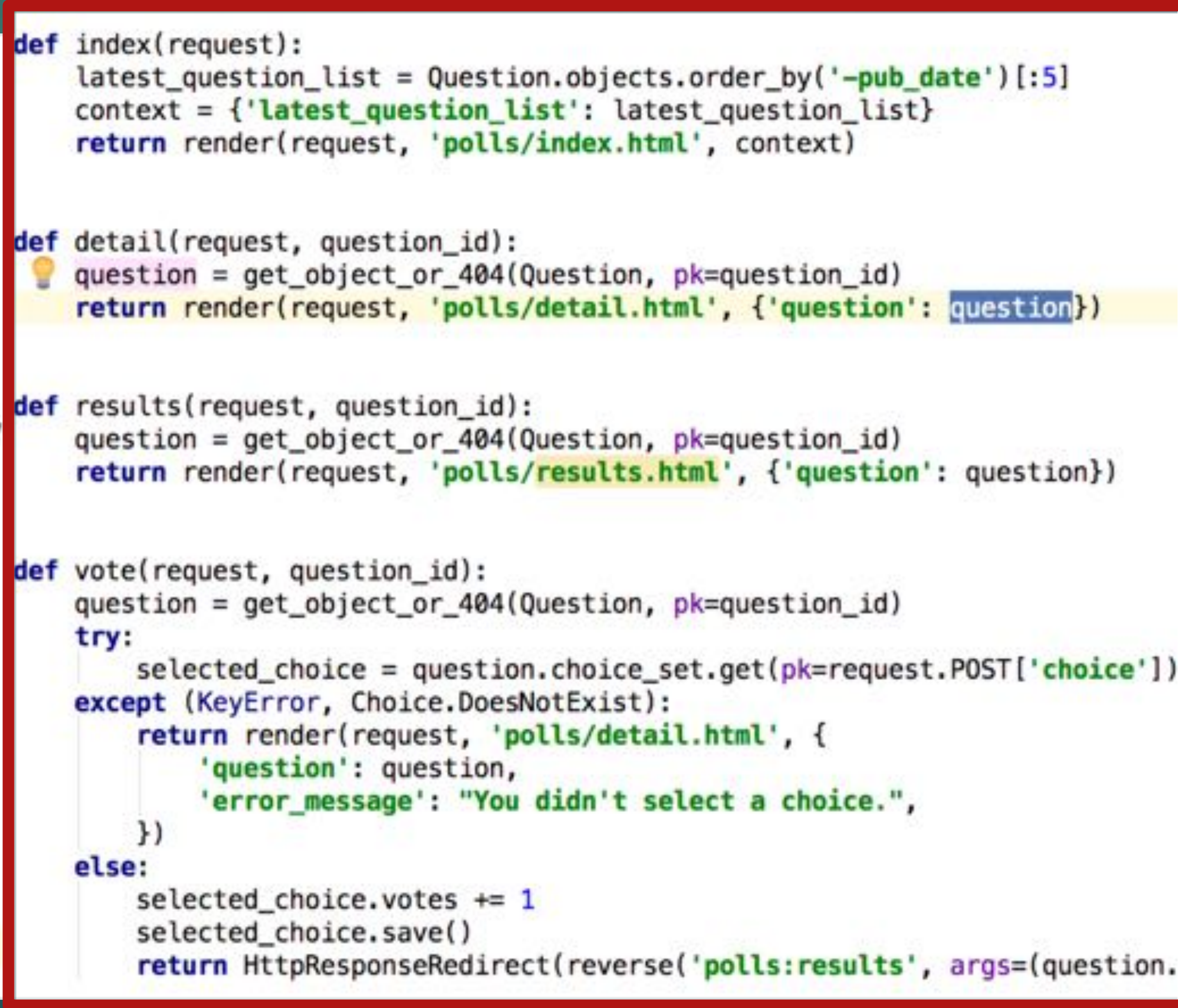
Demo…

- Step5: Define Model

Step5: Define View



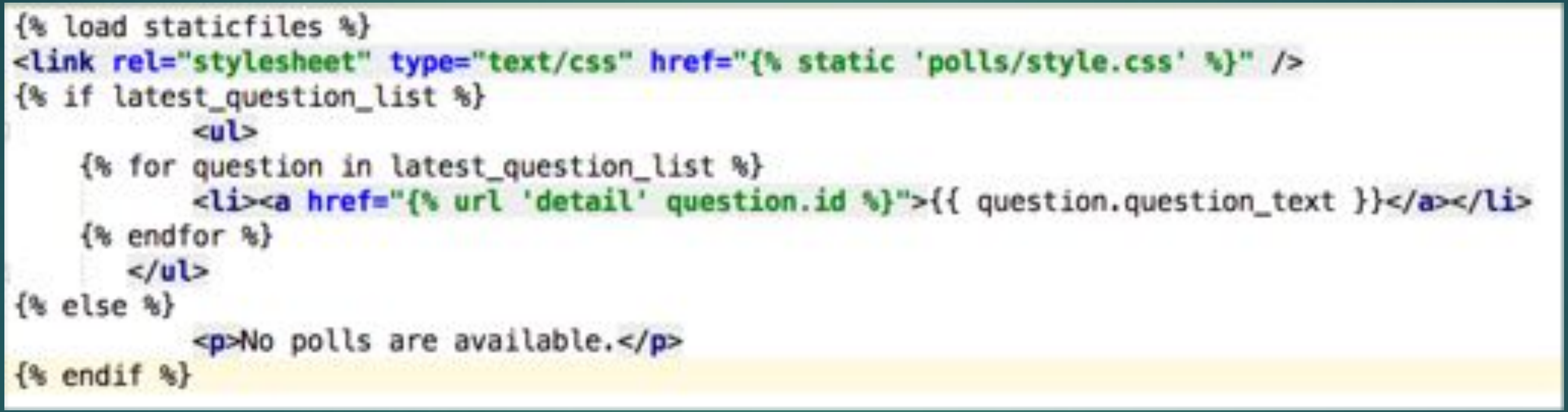
```python
class Question(models.Model):
    question_text = models.CharField(max_length=200)
    pub_date = models.DateTimeField('date published')

    def __str__(self):
        return self.question_text

    def was_published_recently(self):
        now = timezone.now()
        return now - datetime.timedelta(days=1) <= self.pub_date <= now

    was_published_recently.admin_order_field = 'pub_date'
    was_published_recently.boolean = True
    was_published_recently.short_description = 'Published recently?'


class Choice(models.Model):
    question = models.ForeignKey(Question)
    choice_test = models.CharField(max_length=200)
    votes = models.IntegerField(default=0)

    def __str__(self):
        return self.choice_test
```



```python
def index(request):
    latest_question_list = Question.objects.order_by('-pub_date')[:5]
    context = {'latest_question_list': latest_question_list}
    return render(request, 'polls/index.html', context)


def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/detail.html', {'question': question})


def results(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/results.html', {'question': question})


def vote(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    try:
        selected_choice = question.choice_set.get(pk=request.POST['choice'])
    except (KeyError, Choice.DoesNotExist):
        return render(request, 'polls/detail.html', {
            'question': question,
            'error_message': "You didn't select a choice.",
        })
    else:
        selected_choice.votes += 1
        selected_choice.save()
        return HttpResponseRedirect(reverse('polls:results', args=(question.
```

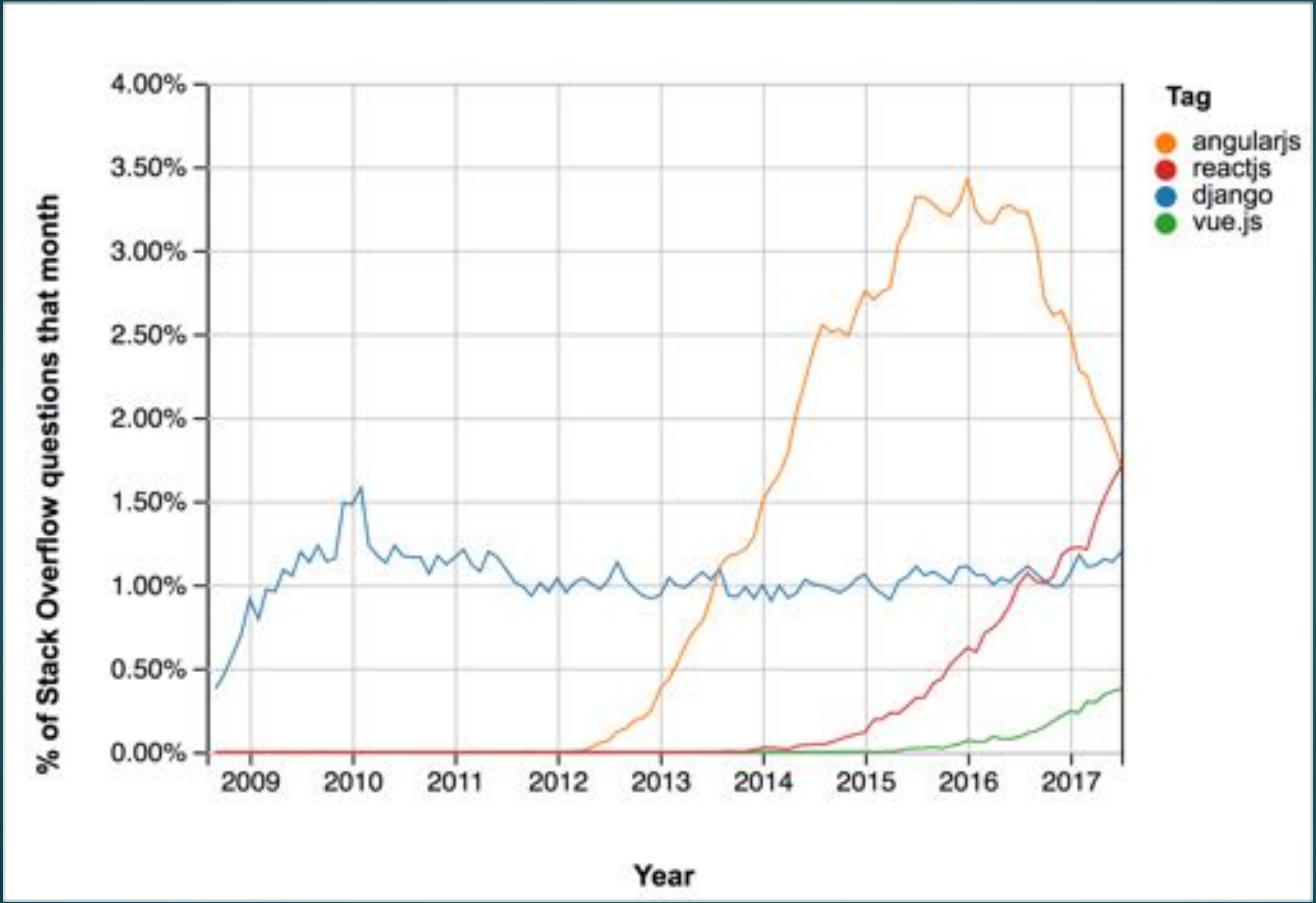
Demo …

- Step 5: Define Template



```
{% load staticfiles %}
<link rel="stylesheet" type="text/css" href="{% static 'polls/style.css' %}" />
{% if latest_question_list %}
            <ul>
    {% for question in latest_question_list %}
            <li><a href="{% url 'detail' question.id %}">{{ question.question_text }}</a></li>
    {% endfor %}
        </ul>
{% else %}
            <p>No polls are available.</p>
{% endif %}
```

Stack Overflow Trends



Sum-up

- Advantages (+)
  - Fast: Thus it becomes an ideal solution for developers having a primary focus on **deadlines**.
  - Fully Loaded: It works in a way that includes dozens of extras to help with user authentication, site maps, content administration, RSS feeds and much more such things.
  - Scalable
- Disadvantages(-)
  - Uses routing pattern specify its URL
  - Django is too monolithic
  - Everything is based on Django ORM.
  - Knowledge of full system is required to work.

Question

- Information for learning Django:
  - A website for finding more information about Django
    - https://www.djangoproject.com
  - A simple web app on the GitHub
    - https://github.com/eaghayi/Polls

- Refrences
  - https://www.jetbrains.com/help/pycharm/step-4-creating-and-running-your-first-django-project.html