ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΕΚΜΑΘΗΣΗΣ ΤΕΧΝΙΚΩΝ ΚΑΙ ΕΝΝΟΙΩΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΜΕ ΤΗΝ PYTHON

Γιαγκούδης Ιωάννης, Πανεπιστήμιο Μακεδονίας, 2019

Περιεχόμενα

- Στόχοι και αναμενόμενη συνεισφορά
- Παιχνιδοποίηση
- Django
- Εκπαιδευτικό υλικό
- Συμπεράσματα
- Live demo

Στόχος

- Ανάπτυξη εφαρμογής για διδασκαλία προγραμματισμού με Python
  - Βασικές έννοιες
  - Βασικες τεχνικες

Συνεισφορά

- Εξοικείωση του με βασικές έννοιες προγραμματισμού
- Εισαγωγή του χρήστη σε βασικές έννοιες μέσω παραδειγμάτων κώδικα
- Έλεγχος κατανόησης μέσω κουίζ ερωτήσεων κλειστού τύπου
- Εξάσκηση μέσω προγραμματιστικών ασκήσεων
- Χρήση του κώδικα της εφαρμογής ως μελέτη περίπτωσης

Παιχνιδοποίηση - Ορισμός

Η παιχνιδοποίηση αναφέρεται στη χρήση στοιχείων ανάπτυξης παιχνιδιών σε καταστάσεις που δεν έχουν παραδοσιακά σχέση με παιχνίδια με στόχο την αύξηση της δραστηριότητας του χρήστη.

Παιχνιδοποίηση - Δομικά Συστατικά

Οι Robinson και Bellotti (2013) έχουν χωρίσει τα δομικά συστατικά παιχνιδοποίησης σε έξι υποκατηγορίες:

- Γενικό πλαίσιο
- Γενικοί κανόνες
- Κοινωνικά στοιχεία
- Κίνητρα
- Πόροι και Περιορισμοί
- Ανατροφοδότηση

Παιχνιδοποίηση - Στοιχεία

Τα στοιχεία παιχνιδοποίησης που ενσωματώθηκαν στην εφαρμογή είναι:

- Πόντοι
- Πίνακας κατάταξης
- Επιτευγματα

Παιχνιδοποίηση - Πλεονεκτήματα

- Κεντρίζει το ενδιαφέρον των μαθητών
- Ενθαρρύνει την αλληλεπίδραση μεταξύ των μαθητών
- Δημιουργεί αίσθηση υπερηφάνειας
- Ωθεί στην ενασχόληση με το διδακτικό υλικό εκτός του χώρου διδασκαλίας

Django

- framework για την ανάπτυξη εφαρμογών ιστού με Python
- σχεδιασμένο για να βοηθάει στη γρήγορη δημιουργία εφαρμογών
- περιέχει έτοιμα εργαλεία (π.χ. σύστημα δρομολόγησης διευθύνσεων)
- παρέχει γραφικό περιβάλλον διαχείρισης της βάσης δεδομένων
- μπορεί να κλιμακωθεί οριζόντια
- έχει καλή τεκμηρίωση

Django - Βασικά Στοιχεία

Κάθε εφαρμογή Django (project) αποτελείται από υποεφαρμογές (apps) που περιέχουν:

- Μοντέλα (Models)
- Προβολές (Views)
- Πρότυπα (Templates)
- Διευθύνσεις URL

Μοντέλα (Models)

Το Django διαχειρίζεται τη βάση δεδομένων με το Object Relational Mapping (O.R.M.).

Τα μοντέλα ορίζουν τη δομή της βάσης δεδομένων μας. Κάθε κλάση είναι ένας πίνακας στη βάση δεδομένων και κάθε ιδιότητα αντιστοιχεί σε μία στήλη.

Εκτός από ιδιότητες μπορεί να περιλαμβάνει και μεθόδους

Είναι η μοναδική πηγή δεδομένων μέσα στην εφαρμογή.

Μοντέλα (Models)

```python
from django.db import models


# Create your models here.

class Chapter(models.Model):
    name = models.TextField(max_length=100)

    def __str__(self):
        return self.name

    class Meta:
        ordering = ['id']


class Exercise(models.Model):
    chapter = models.ForeignKey(Chapter, on_delete=models.CASCADE, blank=True, null=True)
    name = models.CharField(max_length=100, null=False, blank=False, )
    instructions = models.TextField(null=False, blank=False)
    test_imports = models.TextField(blank=True)
    test = models.TextField(null=False, blank=False)
    text = models.TextField(null=True, blank=True)

    def __str__(self):
        return self.name

    class Meta:
        ordering = ['id']
```

Προβολές (Views)

Το επίπεδο των προβολών είναι υπεύθυνο για τη διαχείριση των HTTP αιτημάτων.

Υπάρχουν δύο είδη προβολών:

- Βασισμένες σε Μεθόδους
- Βασισμένες σε Κλάσεις

Οι προβολές βρίσκονται στο αρχείο views.py κάθε υποεφαρμογής.

Προβολή Βασισμένη σε Μέθοδο

```
from django.http import HttpResponse


def index(request):
    return HttpResponse("Hello, world!")
```

Προβολή Βασισμένη σε Κλάση

```
class CodingExerciseListView(LoginRequiredMixin, ListView):
    model = Chapter
    template_name = 'codingExercise/exercise_list.html'

    def get_context_data(self, **kwargs):
        context = super(CodingExerciseListView, self).get_context_data(**kwargs)
        profileObject = Profile.objects.get(user__username=self.request.user)
        completed_exercises = profileObject.getCompletedExercises(True)
        context['completed_exercises'] = completed_exercises
        return context
```

Πρότυπα (Templates)

Τα πρότυπα είναι υπεύθυνα για την προβολή της εφαρμογής.

Επιτρέπουν τη χρήση μεταβλητών.

Επιτρέπουν τη χρήση δομών επιλογής (if) και δομών επανάληψης (for).

Είναι αρχεία HTML.

Πρότυπα (Templates)

```
{% extends 'base.html' %}

{% block content %}


    <div class="container">
        {% for chapter in object_list %}
            <div class="chapter-container">
                <h1 class="card-header bg-dark text-white text-center "> {{ chapter }}</h1>
                <ul class="list-group">

                    {% for exercise in chapter.exercise_set.values %}
                        <li class="list-group-item">
                            {% if exercise.name in completed_exercises %}
                                <i class="fa fa-check-circle-o fa-2x" aria-hidden="true" style="...">></i>
                            {% else %}

                                <i class="fa fa-question-circle-o fa-2x" aria-hidden="true" style="...">></i>
                            {% endif %}

                            <a href="{% url 'codingExercise:exercise' pk=exercise.id %}">{{ exercise.name }} </a>

                        </li>




                    {% empty %}
                        <li>No exercises yet.</li>
                    {% endfor %}

                </ul>
            </div>
        {% endfor %}

    </div>

{% endblock %}
```

Διευθύνσεις URL

Κάθε προβολή σχετίζεται με μια διεύθυνση.

Για λόγους οργάνωσης κάθε υποεφαρμογή έχει έχει το δικό της αρχείο με διευθύνσεις URL και όλα αυτά τα αρχεία συμπεριλαμβάνονται σε ένα κεντικό αρχείο.

Οι διευθύνσεις ορίζονται με κανονικές εκφράσεις.

```
from django.conf.urls import url


from .views import CodingExerciseListView, CodingExerciseDetailView, Finished


app_name = 'codingExercise'
urlpatterns = [
    url(r'^finished$', Finished.as_view(), name='finished'),
    url(r'^$', CodingExerciseListView.as_view(), name='exercise_list'),
    url(r'^(?P<pk>[0-9]+)$', CodingExerciseDetailView.as_view(), name='exercise'),

]
```

```
from django.conf.urls import url, include
from django.contrib import admin
from django.views.generic import TemplateView

urlpatterns = [
    url(r'^admin/', admin.site.urls),
    url(r'^ce/', include('codingExercise.urls')),
    url(r'^account/', include('account.urls')),
    url(r'^quiz/', include('quiz.urls')),
    url(r'^example/', include('examples.urls')),
    url(r'^leaderboard/', include('leaderboard.urls')),
    url(r'^', TemplateView.as_view(template_name='home.html'), name='home'),
    url(r'^badges/', include('badges.urls')),
```

Εκπαιδευτικό Υλικό

Η εφαρμογή περιέχει:

- Παραδείγματα Κώδικα
- Προγραμματιστικές Ασκήσεις
- Τεστ Ερωτήσεων Πολλαπλής Επιλογής

Εκπαιδευτικό Υλικό - Θεματικές Ενότητες

- Βασικά
- Δομές Δεδομένων
- Μέθοδοι
- Δομές Επιλογής
- Δομές Επανάληψης
- Αλφαριθμητικά
- Αντικειμενοστρεφής Προγραμματισμός

Συμπεράσματα

Περιορισμοί

- Έλλειψη δυνατότητας ελέγχου ορθότητας πιο δύσκολων θεμάτων

Επεκτάσεις

- Ενσωμάτωση περισσότερων στοιχείων παιχνιδοποίησης
- Προσθήκη περιεχομένου χρηστών
- Αξιολόγηση της εφαρμογής από χρήστες

Τέλος

- Live Demo

Ευχαριστώ για τον χρόνο σας!