|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ (ИУ5)\_\_\_\_

**О Т Ч Е Т**

**по домашнему заданию**

по дисциплине: Разработка интернет-приложений\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

на тему: Создание веб-приложения на основе технологий Python, Django,\_\_\_\_\_ JavaScript, MySQL.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_­­\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_ИУ5-53Б\_\_\_\_\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Труфанов В.А.\_\_\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Гапанюк Ю.Е.\_\_\_**

(Подпись, дата) (И.О.Фамилия)

*2019 г.*

1. **Содержание задания**

В рамках домашнего задания студент разрабатывает веб-приложение на основе технологий Python, Django, JavaScript, MySQL.

Предметная область домашнего задания содержит субъект, отношение и объект. Необходимые атрибуты субъекта и объекта выбираются студентом самостоятельно. Предметная область (по вариантам) представлена в следующей таблице. По согласованию с преподавателем предметная область может быть изменена.

Разрабатываемое веб-приложение должно выполнять указанные ниже функции. Функции разбиты на группы. Для каждой функции указано количество рейтинговых баллов, которые получает студент в случае реализации данной функции.

### Регистрация

1. При входе, если не залогинен, открывается регистрация (1 балл)
2. Есть рабочая кнопка выхода (1 балл)

### Авторизация

1. Автоматически логинится, редирект на главную страницу (список сущностей). (1 балл)
2. Если логин / пароль неправильный, показана ошибка (1 балл)
3. Профиль пользователя использует стороннюю модель либо переопределена основная модель юзера (1 балл)

### Список сущностей

1. Страница открывается и данные берутся из базы. (1 балл)
2. Отображаются картинки сущностей (1 балл)
3. Есть пагинация (1 балл)
4. Пагинация или бесконечный скролл реализован на AJAX. (1 балл)

### Страница добавления сущностей

1. Реализована в виде модального окна для добавления сущностей. (1 балл)
2. Добавление сущности происходит по AJAX (1 балл)
3. Есть форма с необходимыми для сущности полями (1 балл)
4. Есть возможность загрузить картинку (1 балл)
5. В случае ошибок они отображаются (1 балл)
6. Валидация основных полей формы происходит с помощью JS (1 балл)
7. Сущность создается в базе данных и редирект на страницу с сущностью (1 балл)

### Страница с сущностью

1. Отображаются данные из базы данных (1 балл)
2. При нажатии кнопки действия происходит добавление этого действия в базу, страница перезагружается, это новое действие добавляется в список действий. (1 балл)
3. Действие выполняется без перезагрузки (AJAX) (1 балл)
4. Список действий всех пользователей над этой сущностью (1 балл)

### Страница личного кабинета пользователя

1. Возможность загрузить аватарку (1 балл)
2. Возможность изменить поля пользователя (1 балл)

### Django-админка

1. Русификация. (1 балл)
2. Работа со всеми сущностями. (1 балл)
3. Фильтрация и поиск в админке основной сущности. (1 балл)

### Deploy

1. Настроен mysql (1 балл)
2. Настроен nginx (1 балл)
3. Настроен gunicorn (1 балл)
4. Проект находится в git-репозитории (1 балл)
5. **Текст программы**

1. Шаблоны

**Базовый шаблон:**

{% load static %}

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">

<title>{% block title %}CoursEra{% endblock %}</title>

<link rel="stylesheet" href="{% static 'CoursEra/css/bootstrap.min.css' %}">

<link rel="stylesheet" href="{% static 'CoursEra/fonts/font-awesome.min.css' %}">

<link rel="stylesheet" href="{% static 'CoursEra/fonts/ionicons.min.css' %}">

<link rel="stylesheet" href="{% static 'CoursEra/css/Footer-Basic.css' %}">

<link rel="stylesheet" href="{% static 'CoursEra/css/Navigation-with-Search.css' %}">

<link rel="stylesheet" href="{% static 'CoursEra/css/ShadowAnimation.css' %}">

{% block extraCSS %}

{% endblock %}

</head>

<body>

<nav class="navbar navbar-light navbar-expand-md navigation-clean-search">

<div class="container-fluid">

<a class="navbar-brand" href="{% url 'index' %}" style="margin-right: 36px;color: #3b85a5;">КурсЭра</a>

<button data-toggle="collapse" class="navbar-toggler" data-target="#navcol-1">

<span class="sr-only">Toggle navigation</span>

<span class="navbar-toggler-icon"></span>

</button>

<div

class="collapse navbar-collapse" id="navcol-1">

{% block search %}

<form class="form-inline mr-auto" action="{% url 'index' %}">

<div class="form-group py-1 py-md-auto">

<label for="search-field">

<i class="fa fa-search" style="color: #aeaeae;"></i>

</label>

<input class="form-control search-field" type="search" id="search-field" name="search"

placeholder="Поиск курсов...">

</div>

</form>

{% endblock %}

{% if user.is\_authenticated and not request.user.is\_superuser %}

{% block auth\_buttons %}

<div class="row d-inline-flex p-0 m-0">

<div class="col">

<a href="{% url 'user-edit' %}">

<img src="{% get\_media\_prefix %}{{ request.user.avatar.name }}" style="width: 50px;height: 50px;"/>

</a>

</div>

<div class="col m-0 p-0">

<ul class="nav navbar-nav d-inline-flex">

<li role="presentation" class="nav-item d-inline-flex flex-column">

<div class="nav-item dropdown"><a data-toggle="dropdown" aria-expanded="false"

class="dropdown-toggle nav-link text-nowrap pl-1 py-0"

href="#" style="font-size: 20px;">{{ request.user.username }}</a>

<div role="menu" class="dropdown-menu dropdown-menu-right">

<a role="presentation" class="dropdown-item" href="{% url 'user-edit' %}">Профиль</a>

<a role="presentation" class="dropdown-item" href="{% url 'password\_change' %}">Пароль</a>

</div>

</div>

<div class="row no-gutters">

<div class="col px-1"><a class="text-nowrap nav-link" id="signout-link"

style="padding: 0!important;" href="#confirm-exit"

data-toggle="modal"

data-target="#confirm-exit">Выйти</a></div>

</div>

</li>

</ul>

</div>

</div>

{% endblock %}

{% else %}

{% block unauth\_buttons %}

<ul class="nav navbar-nav">

<li class="nav-item" role="presentation"><a class="nav-link" href="{% url 'login' %}">Войти</a>

</li>

<li class="nav-item" role="presentation"><a class="nav-link" href="{% url 'signup' %}">Регистрация</a>

</li>

</ul>

{% endblock %}

{% endif %}

</div>

</div>

</nav>

<div class="container">

{% if user.is\_authenticated and not request.user.is\_superuser %}

<div role="dialog" tabindex="-1" class="modal fade" id="confirm-exit">

<div class="modal-dialog modal-dialog-centered" role="document">

<div class="modal-content">

<div class="modal-header">

<h4 class="modal-title">Выход</h4>

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span

aria-hidden="true">×</span></button>

</div>

<div class="modal-body">

<p>Уверены что хотите выйти?</p>

</div>

<div class="modal-footer">

<button class="btn btn-light border-0" type="button" data-dismiss="modal">Отмена</button>

<a href='{% url 'logout' %}' class="btn btn-primary text-white border-0" role="button" style="background-color: #6698eb;">

Выйти

</a>

</div>

</div>

</div>

</div>

<script src="{% static 'CoursEra/js/modal-logout.js' %}" defer></script>

{% endif %}

{% block content %}{% endblock %}

</div>

<div class="footer-basic">

<footer>

<div class="social"><a href="#"><i class="icon ion-social-instagram"></i></a><a href="#"><i

class="icon ion-social-snapchat"></i></a><a href="#"><i class="icon ion-social-twitter"></i></a><a

href="#"><i class="icon ion-social-facebook"></i></a></div>

{% block footer\_links %}

<ul class="list-inline">

<li class="list-inline-item"></li>

<li class="list-inline-item"><a href="{% url 'about' %}">Помощь</a></li>

<li class="list-inline-item"><a href="{% url 'about' %}">Контакты</a></li>

<li class="list-inline-item"><a href="{% url 'about' %}">О нас</a></li>

<li class="list-inline-item"><a href="{% url 'about' %}">Команда</a></li>

</ul>

{% endblock %}

<div class="row d-flex align-items-center m-4">

<div class="col d-flex justify-content-end">

{% if not request.user.is\_superuser %}

<a class="text-monospace text-black-50 border rounded shadow-sm border-0 p-1 text-decoration-none"

href="{% url 'admin:index' %}" style="background-color: #b2fef7;">Админу</a>

{% endif %}

{% if request.user.is\_superuser %}

<div class="dropdown">

<button class="btn btn-info dropdown-toggle text-monospace text-black-50" data-toggle="dropdown"

aria-expanded="false" type="button" style="background-color: #b2fef7;">

Администрирование

</button>

<div role="menu" class="dropdown-menu dropdown-menu-right">

<a role="presentation" class="dropdown-item"

href="{% url 'admin:index' %}" style="background-color: #b2fef7;">Админка</a>

<a role="presentation" class="dropdown-item" href="{% url 'course-add' %}"

style="background-color: #b2fef7;" >Добавить курс</a>

</div>

</div>

{% endif %}

</div>

</div>

<p class="copyright">{% block copyright %}КурсЭра © 2019{% endblock %}</p>

</footer>

</div>

<script src="{% static 'CoursEra/js/jquery.min.js' %}"></script>

<script src="{% static 'CoursEra/js/bootstrap.min.js' %}"></script>

{% block extraJS %}

{% endblock %}

</body>

</html>

**Главная страница:**

{% extends "./base.html" %}

{% load static %}

{% block extraCSS %}

<link rel="stylesheet" href="{% static 'CoursEra/css/custom-link.css' %}">

{% endblock %}

{% block content %}

{% if user.is\_authenticated %}

<div class="row d-inline-flex my-2">

<div class="col border-right border-dark"><a class="link" href="{% url 'index' %}">Все курсы</a></div>

<div class="col"><a class="text-nowrap link" href="{% url 'index' %}?user\_courses=true">Мои курсы</a></div>

</div>

{% endif %}

{% if page\_obj.paginator.count > 0 %}

<div class="row">

{% for course in page\_obj %}

{% include 'CoursEra/course-card.html' with course=course %}

{% endfor %}

</div>

<div class="col d-flex justify-content-center mt-4">

<nav>

<ul class="pagination">

{% if page\_obj.has\_previous %}

{% comment %}

<li class="page-item"><a class="page-link" href="?page=1" aria-label="Last"><span

aria-hidden="true">первый</span></a></li>

{% endcomment %}

<li class="page-item"><a class="page-link" href="?page={{ page\_obj.previous\_page\_number }}"

aria-label="Previous"><span

aria-hidden="true">&laquo;</span></a></li>

{% endif %}

<li class="page-item"><a class="page-link" href="#">{{ page\_obj.number }} </a></li>

{% if page\_obj.has\_next %}

<li class="page-item"><a class="page-link" href="?page={{ page\_obj.next\_page\_number }}"

aria-label="Last"><span

aria-hidden="true">&raquo;</span></a></li>

{% comment %}

<li class="page-item"><a class="page-link" href="?page={{ page\_obj.paginator.num\_pages }}"

aria-label="Previous"><span

aria-hidden="true">последний</span></a></li>

{% endcomment %}

{% endif %}

</ul>

</nav>

</div>

{% else %}

{% if user.is\_authenticated %}

<h5 class="m-3">похоже, что {{user.username }} не подписан на курсы</h5>

{% else %}

<h5 class="m-3">0 результатов по запросу <i>{{ search }}</i></h5>

{% endif %}

{% endif %}

{% endblock %}

**Форма для редактирования:**

{% extends "./base.html" %}

{% load static %}

{% block extraJS %}

<script src="{% static 'CoursEra/js/file-input-name.js' %}" defer></script>

{% endblock %}

{% block content %}

<div class="row mt-4 mb-2">

<div class="col mx-2 mx-md-3 mx-xl-5">

{% block form\_tag %}

<form class="py-3 mx-xl-4 " enctype="multipart/form-data" action="{{ action\_url }}" method="post">

{% endblock %}

{% csrf\_token %}

<h5 class="font-weight-bold mb-0 mb-sm-3"><strong>{% block form\_title %}{% endblock %}</strong></h5>

{% block form\_header %}

{% include "CoursEra/form-messages.html" with form\_messages=messages %}

{% include "CoursEra/form-non-field-errors.html" with form=form %}

{% endblock %}

{% for field in form %}

{% block form\_content %}

{% include "CoursEra/form-helps.html" with field=field %}

{% include "CoursEra/form-errors.html" with field=field %}

{% endblock %}

{% endfor %}

{% block submit\_button %}

<button class="btn btn-primary d-inline-block border-0 mt-1" type="submit"

style="background-color: #6698eb;">{% block submit\_button\_title %}Сохранить данные{% endblock %}

</button>

{% endblock %}

</form>

</div>

</div>

{% endblock %}

Страница описания курса

{% extends "./base.html" %}

{% load static %}

{% block content %}

<div class="row">

<div class="col-12 col-sm-auto col-md-9 pt-3 pb-2">

<h2 class="mb-3 text-center">{{course.name}}</h2><img class="d-block mb-5 mx-auto img-fluid"

src="{% get\_media\_prefix %}{{ course.image.name }}">

<section>

<h3>Описание курса</h3>

<p class="pr-3">

{{course.description|linebreaksbr}}

</p>

<section class="pt-2">

<p class="d-inline pr-2" style="font-size: 14px;">Дата добавления курса:</p><em><span

style="font-size: 14px;">{{course.pub\_date}}</span></em></section>

</section>

</div>

<div class="col-auto col-sm-auto col-md-3 d-flex flex-grow-1 justify-content-center align-items-start align-items-xl-start sticky-top py-3 pt-md-5"

style="background-color: #56c6c6;">

<div class="sticky-top">

<h3 style="color: #ffffff;">Подписаться</h3>

<form action="{% url 'course-enroll' course.id %}" method='GET'>

<button class="btn btn-primary btn-block border-0" type="submit"

style="background-color: #6698eb;color: #ffffff;"> Подписаться</button>

</form>

</div>

</div>

</div>

{% endblock %}

**3. Подключение БД**

**Модели:**

from django.db import models

from django.contrib.auth.models import AbstractUser

from django.utils.translation import ugettext\_lazy as \_

class CustomUser(AbstractUser):

# add avatar field to user

avatar = models.ImageField(upload\_to='avatars', blank=True, default='avatars/default\_avatar.png')

# make email field required

email = models.EmailField(\_('email address'), blank=False)

class Course(models.Model):

name = models.CharField(max\_length=100)

description = models.TextField()

image = models.ImageField(upload\_to='courses')

pub\_date = models.DateTimeField('Дата добавления')

subscribers = models.ManyToManyField(CustomUser, blank=True)

class Meta:

verbose\_name = \_("Курс")

verbose\_name\_plural = \_("Курсы")

def \_\_str\_\_(self):

return self.name

**Views**

from django.contrib import messages

from django.contrib.admin.views.decorators import staff\_member\_required

from django.contrib.auth.decorators import login\_required

from django.contrib.auth.views import PasswordChangeView, LoginView

from django.shortcuts import render, get\_object\_or\_404

from django.utils.decorators import method\_decorator

from django.utils.encoding import smart\_str

from django.views.generic import DetailView, RedirectView, ListView

from CoursEra.models import Course, CustomUser

from .forms import CustomUserCreationForm, CustomUserChangeForm

from django.views import View

from django.utils.http import unquote\_plus

# Create your views here.

from .utils import FormControlsStylerMixin

class CourseList(ListView):

paginate\_by = 4

def get\_context\_data(self, \*, object\_list=None, \*\*kwargs):

context = super().get\_context\_data(\*\*kwargs)

context['search'] = self.request.GET.get('search')

return context

def get\_queryset(self):

search = self.request.GET.get('search','')

user\_courses = self.request.GET.get('user\_courses','')

if search:

search = unquote\_plus(search)

search = smart\_str(search)

query\_set = Course.objects.filter(name\_\_icontains=search)

elif user\_courses:

query\_set = self.request.user.course\_set.all()

else:

query\_set = Course.objects.all()

return query\_set

def index(request):

"""

if request.user.is\_authenticated and not request.user.is\_superuser:

current\_user = request.user

courses\_list = current\_user.course\_set.all()

else:

"""

courses\_list = Course.objects.order\_by('-pub\_date')[:5]

context = {'courses\_list': courses\_list}

return render(request, 'CoursEra/index.html', context)

class CourseUserView(RedirectView):

permanent = False

query\_string = True

current\_user = None

course = None

def setup(self, request, \*args, \*\*kwargs):

self.course = get\_object\_or\_404(Course, pk=kwargs['pk'])

self.current\_user = request.user

return super().setup(request, \*args, \*\*kwargs)

def enroll(self, \*args, \*\*kwargs):

self.course.subscribers.add(self.current\_user)

self.course.save()

def unenroll(self, \*args, \*\*kwargs):

self.course.subscribers.remove(self.current\_user)

self.course.save()

class CourseUnenrollView(CourseUserView):

pattern\_name = 'course-detailed'

def get\_redirect\_url(self, \*args, \*\*kwargs):

self.unenroll(self, \*args, \*\*kwargs)

return super().get\_redirect\_url(\*args, \*\*kwargs)

class CourseEnrollView(CourseUserView):

pattern\_name = 'course-enrolled'

def get\_redirect\_url(self, \*args, \*\*kwargs):

self.enroll(self, \*args, \*\*kwargs)

return super().get\_redirect\_url(\*args, \*\*kwargs)

class CourseRedirectView(RedirectView):

permanent = False

query\_string = True

def set\_redirect\_url(self, \*args, \*\*kwargs):

course = get\_object\_or\_404(Course, pk=kwargs['pk'])

current\_user = self.request.user

if current\_user.is\_authenticated and course.subscribers.filter(pk=current\_user.pk).exists():

self.pattern\_name = 'course-enrolled'

else:

self.pattern\_name = 'course-detailed'

def get\_redirect\_url(self, \*args, \*\*kwargs):

self.set\_redirect\_url(self, \*args, \*\*kwargs)

return super().get\_redirect\_url(\*args, \*\*kwargs)

class CourseDescView(DetailView):

model = Course

class CourseSubsView(ListView):

def get\_queryset(self):

return CustomUser.objects.filter(course\_\_pk=self.kwargs['pk'])

def get\_context\_data(self, \*\*kwargs):

# Call the base implementation first to get a context

context = super().get\_context\_data(\*\*kwargs)

# Add in a QuerySet of all the books

context['course\_id'] = self.kwargs['pk']

return context

class MyLoginView(FormControlsStylerMixin, LoginView):

def get\_form(self, form\_class=None):

form = super().get\_form(form\_class)

if form.is\_bound:

form = self.set\_validation\_attrs(form)

return form

class MyFormView(FormControlsStylerMixin, View):

template\_name = None

form\_class = None

form = None

def render\_form(self, request, \*args, \*\*kwargs):

return render(request, self.template\_name, {'form': self.form})

def get(self, request, \*args, \*\*kwargs):

self.form = self.get\_form(request, \*args, \*\*kwargs)

return self.render\_form(request, \*args, \*\*kwargs)

def post(self, request, \*args, \*\*kwargs):

self.form = self.post\_form(request, \*args, \*\*kwargs)

if self.form.is\_valid():

self.save\_form(request, \*args, \*\*kwargs)

self.form = self.get\_form(request, \*args, \*\*kwargs)

if self.form.is\_bound:

self.form = self.set\_validation\_attrs(self.form)

return self.render\_form(request, \*args, \*\*kwargs)

def get\_form(self, request, \*args, \*\*kwargs):

return self.form\_class()

def post\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(request.POST)

def save\_form(self, request, \*args, \*\*kwargs):

self.form.save()

class SignUpUserFormView(MyFormView):

def post\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(request.POST, request.FILES)

def save\_form(self, request, \*args, \*\*kwargs):

messages.success(request, 'Регистрация успешна завершена!')

super().save\_form(request, \*args, \*\*kwargs)

class AddCourseFormView(MyFormView):

def post\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(request.POST, request.FILES)

def save\_form(self, request, \*args, \*\*kwargs):

messages.success(request, 'Курс успешно добавлен!')

super().save\_form(request, \*args, \*\*kwargs)

class EditUserFormView(MyFormView):

current\_user = None

def setup(self, request, \*args, \*\*kwargs):

self.current\_user = CustomUser.objects.get(pk=request.user.pk)

return super().setup(request, \*args, \*\*kwargs)

def save\_form(self, request, \*args, \*\*kwargs):

self.form = self.form\_class(request.POST, request.FILES, instance=request.user)

super().save\_form(request, \*args, \*\*kwargs)

def get\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(instance=self.current\_user)

def post\_form(self, request, \*args, \*\*kwargs):

return self.form\_class(request.POST, request.FILES, instance=self.current\_user)

class CustomPasswordChangeView(FormControlsStylerMixin, PasswordChangeView):

def get\_form(self, form\_class=None):

form = super().get\_form(form\_class)

if form.is\_bound:

form = self.set\_validation\_attrs(form)

return form

**Forms**

from datetime import datetime

from django.contrib.auth.forms import UserCreationForm, PasswordChangeForm, AuthenticationForm

from django.forms import ModelForm

from django import forms

from CoursEra.models import CustomUser, Course

from CoursEra.utils import FormControlsStylerMixin

class CustomUserCreationForm(FormControlsStylerMixin, UserCreationForm):

class Meta(UserCreationForm.Meta):

model = CustomUser

fields = UserCreationForm.Meta.fields + ('email', 'avatar')

# add styles to inputs

def \_\_init\_\_(self, \*args, \*\*kwargs):

super().\_\_init\_\_(\*args, \*\*kwargs)

# remove helper text on password

self.fields['password1'].help\_text = None

class CustomUserChangeForm(FormControlsStylerMixin, ModelForm):

class Meta:

model = CustomUser

fields = ['username', 'email', 'avatar']

widgets = {'avatar': forms.FileInput}

labels = {

'username': 'Сменить логин',

'email': 'Сменить email',

'avatar': 'Сменить аватар'

}

class CustomPasswordChangeForm(FormControlsStylerMixin, PasswordChangeForm):

"""A view for displaying a password change form."""

class CourseAddForm(FormControlsStylerMixin, ModelForm):

class Meta:

model = Course

fields = ['name', 'description', 'image', 'pub\_date']

labels = {

'name': 'Название',

'description': 'Описание',

'image': 'Фото'

}

widgets = {'image': forms.FileInput,

'pub\_date': forms.DateTimeInput(attrs={'type': 'date'}) }

initial = {'pub\_date': datetime.now}

class MyLoginForm(FormControlsStylerMixin, AuthenticationForm):

"""login form."""

**4. Настройка Django.admin**

from django.contrib import admin

from .models import Course, CustomUser

admin.site.register(Course)

admin.site.register(CustomUser)

1. **Экранные формы с примерами работы программы**











