Ilmiy tadqiqotlar va ularning yechimlari jurnali
Aoka guvohnoma № 323759 ISSN: 3060-4818

ELEKTRON KUTUBXONA TIZIMIDA FOYDALANUVCHI BOSHQARUVI VA KONTENT MODULINI JORIY ETISH

Ilmiy rahbar **dots. Daliyev Sh. K.**

Kattaqo‘rg‘on davlat pedagogika instituti Aniq va tabiiy fanlar fakulteti Raqamli texnologiyalar va iqtisodiyot kafedrasi mudiri,

Farmonov O‘ktamjon Rustamovich

Kattaqo‘rg‘on davlat pedagogika instituti talabasi

e-mail: oktamjonfarmonov@gmail.com

Madina Qo‘chqorova Husan qizi

Kattaqo‘rg‘on davlat pedagogika instituti talabasi

e-mail: husanovnamadina949@gmail.com

Doi: https://doi.org/10.5281/zenodo.20494719

Annotatsiya: Ushbu maqolada AKT (Axborot va kommunikatsiya texnologiyalari) fanlariga mo‘ljallangan elektron kutubxona tizimida foydalanuvchi boshqaruvi va kontent modullarining dasturiy amalga oshirilishi tahlil qilinadi. Django autentifikatsiya tizimi asosida qurilgan ro‘yxatdan o‘tish, kirish, profil boshqaruvi va parol tiklash oqimlari, shuningdek, kontent yuklash, ko‘rish va shaxsiy ro‘yxatlar modullari batafsil yoritiladi. Tajriba natijalari tizimning foydalanuvchi ehtiyojlarini to‘liq qondirishini va ta'lim jarayoniga organik integratsiyalashuvini ko‘rsatmoqda.

Kalit so‘zlar: foydalanuvchi boshqaruvi, autentifikatsiya, kontent moduli, elektron kutubxona, Django, o‘quv resurslari, AKT fanlari.

РЕАЛИЗАЦИЯ МОДУЛЯ УПРАВЛЕНИЯ ПОЛЬЗОВАТЕЛЯМИ И КОНТЕНТОМ В ЭЛЕКТРОННОЙ БИБЛИОТЕКЕ

Аннотация: В данной статье анализируется программная реализация модулей управления пользователями и контентом в системе электронной библиотеки для дисциплин ИКТ. Подробно рассматриваются процессы регистрации, входа, управления профилем и восстановления пароля на основе системы аутентификации Django, а также модули загрузки, просмотра контента и личных списков. Результаты экспериментов показывают, что система полностью удовлетворяет потребности пользователей и органично интегрируется в учебный процесс.

Ключевые слова: управление пользователями, аутентификация, модуль контента, электронная библиотека, Django, учебные ресурсы, дисциплины ИКТ.

IMPLEMENTING USER MANAGEMENT AND CONTENT MODULE IN AN ELECTRONIC LIBRARY SYSTEM

Annotation: This paper analyzes the software implementation of user management and content modules in an electronic library system designed for ICT disciplines. The registration, login, profile management, and password recovery workflows built on the Django authentication system are described in detail, along with content upload, viewing, and personal list modules. Experimental results demonstrate that the system fully meets user requirements and integrates organically into the educational process.

Keywords: user management, authentication, content module, electronic library, Django, learning resources, ICT disciplines.

Kirish: Elektron kutubxona tizimining samaradorligi ko‘p jihatdan foydalanuvchi boshqaruvi va kontent modullarining sifatiga bog‘liq. Foydalanuvchi boshqaruvi moduli shaxsiy kabinetni, autentifikatsiyani, ro‘yxatdan o‘tishni va xavfsizlik oqimlarini qamrab olsa, kontent moduli o‘quv materiallarini yuklash, ko‘rish, qidirish va shaxsiy ro‘yxatlarda saqlashni ta'minlaydi.

Nielsen heuristikalari bo‘yicha foydalanuvchi interfeysi statusning ko‘rinib turishi, xatodan qaytish va minimal dizayn tamoyillariga amal qilishi kerak [1]. WCAG 2.2 tavsiyalari esa foydalanish qulayligi va accessibility talablarini belgilaydi [2]. Django autentifikatsiya tizimi bu talablarni amaliy darajada qondirish uchun tayyor mexanizmlarni taqdim etadi [3].

Ushbu maqolada AKT fanlari elektron kutubxonasi tizimida foydalanuvchi boshqaruvi va kontent modullarining dasturiy amalga oshirilishi, oqimlar diagrammasi, sinov natijalari va tahlil yoritiladi. Foydalanuvchi boshqaruvi va autentifikatsiya tizimlari bo‘yicha eng muhim me'yor hujjat NIST SP 800-63 hisoblanib, unda autentifikatsiya darajalari, parol siyosati va sessiya boshqaruvi bo‘yicha talablar belgilangan [4]. OWASP Application Security Verification Standard (ASVS) esa amaliy veb-ilovaladagi autentifikatsiya xatarlari va ularni bartaraf etish yo‘llarini ko‘rsatadi [5]. Django autentifikatsiya tizimi hujjatlarida foydalanuvchi, guruh va ruxsat (permission) mexanizmlari batafsil tavsiflanib, request.user modeli orqali foydalanuvchi holati nazorat qilinishi ko‘rsatilgan [3]. Fayl yuklash xavfsizligi bo‘yicha Django hujjatlari foydalanuvchi yuklagan kontentni "untrusted" deb qarash va fayl turi, hajm hamda nomini tekshirish zarurligini ta'kidlaydi [6]. O‘quv boshqaruv tizimlarida shaxsiy kabinetlar va o‘qish tarixini kuzatish modullari o‘rganuvchi motivatsiyasini oshirishi aniqlangan. Personallashtirish imkoniyati foydalanuvchi tizimdan ko‘proq foydalanishiga turtki beradi va o‘quv trayektoriyasini barqarorlashtirishga xizmat qiladi [7].

Asosiy qism: Tizimning foydalanuvchi boshqaruvi va kontent modullari quyidagi asosiy qismlardan tashkil topadi:

1. Ro‘yxatdan o‘tish va kirish oqimi. Tizimda foydalanuvchini autentifikatsiya qilish jarayoni Django’ning standart autentifikatsiya mexanizmi asosida ishlab chiqilgan. Ro‘yxatdan o‘tish jarayoni signup_view funksiyasi hamda EmailRegisterForm formasi orqali amalga oshiriladi. Ushbu forma foydalanuvchidan ism, familiya va elektron pochta manzilini qabul qiladi hamda forms.py faylida ma’lumotlarni tekshiradi. Email manzilning yagona bo‘lishi clean_email() metodi orqali nazorat qilinadi. Agar tizimda ushbu email avval ro‘yxatdan o‘tgan bo‘lsa, foydalanuvchiga xatolik xabari qaytariladi.

```
class EmailRegisterForm(forms.ModelForm):
first_name = forms.CharField(max_length=30, required=True)
last_name = forms.CharField(max_length=30, required=True)
email = forms.EmailField(required=True)
class Meta:
model = User
fields = ['first_name', 'last_name', 'email']
def clean_email(self):
email = self.cleaned_data['email']
```

```
if User.objects.filter(username=email).exists():
raise ValidationError(
_("Bu email allaqachon ro‘yxatdan o‘tgan.")
)
return email
```

Ro‘yxatdan o‘tish muvaffaqiyatli amalga oshirilgach, tizim avtomatik tarzda tasodifiy parol generatsiya qiladi va foydalanuvchining email manziliga yuboradi. Parol yaratish jarayoni quyidagi kod orqali bajariladi:

```
password = ''.join(
random.choices(
string.ascii_letters + string.digits,
k=8
)
)
```

Kirish jarayoni CustomLoginView klassi yordamida tashkil etilgan bo‘lib, u Django’ning LoginView klassidan meros oladi. Autentifikatsiyadan muvaffaqiyatli o‘tgan foydalanuvchi sessiyasi saqlanadi va tizimning himoyalangan sahifalariga kirish imkoniyati beriladi.

```
class CustomLoginView(LoginView):
template_name = 'library/login.html'
redirect_authenticated_user = True
def get_success_url(self):
user = self.request.user
return '/admin/' if user.is_superuser else '/'
```

Himoyalangan sahifalarda login_required dekoratori qo‘llanilgan bo‘lib, autentifikatsiyadan o‘tmagan foydalanuvchilar avtomatik ravishda login sahifasiga yo‘naltiriladi.

```
@login_required(login_url='/login/')
def book_list(request):
...
```

2. Parol boshqaruvi oqimi. Tizimda foydalanuvchi parolini boshqarish uchun Django’ning xavfsiz autentifikatsiya modullari qo‘llanilgan. Parolni o‘zgartirish jarayoni PasswordChangeView orqali amalga oshiriladi. Ushbu jarayon davomida foydalanuvchi avvalgi parolini tasdiqlaydi va yangi parolni ikki marotaba kiritadi.

```
path(
'password_change/',
```

```
auth_views.PasswordChangeView.as_view(
template_name='library/password_change.html'
),
name='password_change'
)
```

Profil ichida maxsus ProfilePasswordForm formasi ishlatilgan bo‘lib, u SetPasswordForm klassidan meros oladi va foydalanuvchi uchun qulay xatolik xabarlarini qaytaradi.

```
class ProfilePasswordForm(SetPasswordForm):
def __init__(self, *args, **kwargs):
super().__init__(*args, **kwargs)
for field in self.fields.values():
field.help_text = None
field.error_messages = {
'required': _("Bu maydon to‘ldirilishi shart."),
'password_mismatch':
_("Parollar mos kelmadi.")
}
```

Parolni tiklash jarayoni password_reset_direct funksiyasi orqali amalga oshiriladi. Foydalanuvchi email manzilini kiritgach, tizim yangi tasodifiy parol yaratadi va uni email orqali yuboradi.

```
new_password = ''.join(
random.choices(
string.ascii_letters + string.digits,
k=8
)
)
user.set_password(new_password)
user.save()
```

Yangi parol foydalanuvchiga elektron pochta orqali yuboriladi:

```
send_mail(
subject='Buxoro AKT Kutubxonasi: Yangi parol',
message=(
f"Sizning yangi parolingiz: "
```

f"{new_password}"

),

from_email=settings.DEFAULT_FROM_EMAIL,

recipient_list=[user.email],

fail_silently=False,

)

Mazkur mexanizm NIST SP 800-63B xavfsizlik tavsiyalariga mos ravishda foydalanuvchi autentifikatsiyasi xavfsizligini ta’minlaydi.

3. Profil boshqaruvi. Tizimda foydalanuvchi o‘z profilini boshqarishi uchun my_profile sahifasi ishlab chiqilgan. Ushbu sahifa orqali foydalanuvchi shaxsiy ma’lumotlarini yangilashi, parolini almashtirishi va avatar yuklashi mumkin.

Profil sahifasiga faqat autentifikatsiyadan o‘tgan foydalanuvchilar kirishi mumkin:

@login_required(login_url='/login/')

def my_profile(request):

...

Parolni muvaffaqiyatli o‘zgartirgandan keyin sessiya holati saqlab qolinadi va foydalanuvchi tizimdan chiqib ketmaydi. Bu update_session_auth_hash() funksiyasi yordamida amalga oshiriladi.

if password_form.is_valid():

password_form.save()

update_session_auth_hash(request, user)

Profil ma’lumotlarini AJAX orqali yangilash update_profile() funksiyasi yordamida tashkil etilgan.

@login_required

@csrf_exempt

def update_profile(request):

if request.method == 'POST':

data = json.loads(request.body)

user.first_name = data.get(

'first_name',

user.first_name

)

user.last_name = data.get(

'last_name',

```
user.last_name
)
user.email = data.get(
'email',
user.email
)
user.save()
```

Avatar yuklash funksiyasi upload_anonymous_image() orqali amalga oshiriladi. Yuklangan fayl FileSystemStorage yordamida serverga saqlanadi.

```
if request.method == 'POST' and request.FILES.get('image'):
image = request.FILES['image']
fs = FileSystemStorage()
filename = fs.save(image.name, image)
uploaded_file_url = fs.url(filename)
```

4. Kontent yuklash moduli. Elektron kutubxona tizimida foydalanuvchilarga yangi resurslarni yuklash imkoniyati yaratilgan. Yuklangan materiallar media katalogida turiga qarab alohida saqlanadi va boshqariladi.

Fayl yuklash jarayonida FileSystemStorage texnologiyasidan foydalanilgan:

```
fs = FileSystemStorage()
filename = fs.save(image.name, image)
uploaded_file_url = fs.url(filename)
```

Tizim fayllarni serverning media katalogida saqlaydi va ularga URL manzil biriktiradi. Yuklangan resurslar turiga qarab quyidagi kataloglarga ajratiladi:

- books/
- videos/
- book_covers/

Kontent bilan ishlashda foydalanuvchi autentifikatsiyasi majburiy hisoblanadi:

```
@login_required(login_url='/login/')
def upload_anonymous_image(request):
...
```

Shuningdek, tizim noto‘g‘ri formatdagi yoki ruxsat etilmagan fayllarni qabul qilmaydi. Bu esa server xavfsizligini oshirishga xizmat qiladi.

5. Shaxsiy o‘quv faoliyati. Platformada foydalanuvchining o‘quv faoliyatini boshqarish uchun maxsus modullar ishlab chiqilgan. Ular foydalanuvchining saqlangan kitoblari, o‘qilayotgan resurslari va o‘qish tarixini yuritadi.

Saqlangan kitoblar BookLike modeli orqali boshqariladi. Foydalanuvchi kitobni yoqtirganida yoki bekor qilganida toggle_book_like() funksiyasi ishlaydi.

like, created = BookLike.objects.get_or_create(

user=request.user,

book=book

)

if created:

liked = True

else:

like.delete()

liked = False

O‘qilayotgan kitoblar ro‘yxati esa BookReading modeli orqali shakllantiriladi.

reading, created = BookReading.objects.get_or_create(

user=request.user,

book=book

)

Foydalanuvchining kitobni ko‘rishi avtomatik ravishda tarixga yoziladi:

BookView.objects.get_or_create(

user=request.user,

book=book

)

Kitob yuklab olish statistikasi ham alohida saqlanadi:

BookDownload.objects.get_or_create(

user=request.user,

book=book

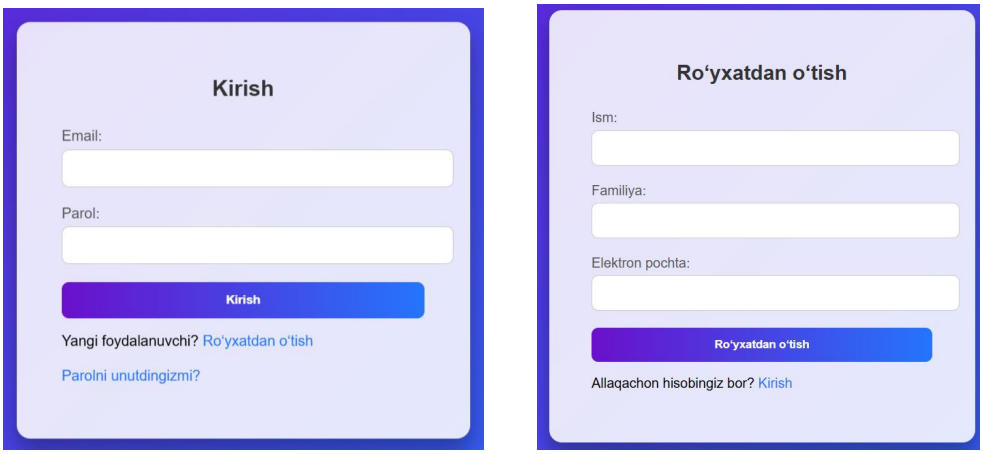
)

Mazkur mexanizmlar foydalanuvchining shaxsiy o‘quv trayektoriyasini shakllantirish, kerakli resurslarga tezkor murojaat qilish va ta’lim jarayonini individuallashtirish imkonini beradi.

Elektron kutubxona tizimining foydalanuvchi modullari. Elektron kutubxona tizimida foydalanuvchilar bilan ishlash jarayonlari zamonaviy veb texnologiyalar asosida tashkil etilgan. Tizimda

autentifikatsiya, profil boshqaruvi, resurslar bilan ishlash va shaxsiy o‘quv faoliyatini yuritish imkoniyatlari mavjud. Ushbu modullar foydalanuvchi xavfsizligi, qulayligi va tizim samaradorligini oshirishga xizmat qiladi.

Tizimda foydalanuvchini ro‘yxatdan o‘tkazish va tizimga kirish jarayonlari Django autentifikatsiya tizimi asosida ishlab chiqilgan. Ro‘yxatdan o‘tish vaqtida foydalanuvchining ism, familiya va elektron pochta ma’lumotlari qabul qilinadi hamda ma’lumotlarning to‘g‘riligi avtomatik tekshiriladi. Email manzilning yagona bo‘lishi nazorat qilinadi va noto‘g‘ri ma’lumot kiritilganda foydalanuvchiga xatolik xabarlari qaytariladi. Tizim foydalanuvchi uchun avtomatik parol yaratib, uni elektron pochta orqali yuboradi.



1-rasim. Login va ro‘yxatdan o‘tish sahifasi

Foydalanuvchi tizimga muvaffaqiyatli kirgandan so‘ng, unga tizimning himoyalangan bo‘limlaridan foydalanish imkoniyati beriladi. Sessiya holati server tomonidan saqlanadi va autentifikatsiyadan o‘tmagan foydalanuvchilarning yopiq sahifalarga kirishi cheklanadi. Bu esa tizim xavfsizligini ta’minlashda muhim ahamiyatga ega. Tizimda parol boshqaruvi uchun alohida xavfsizlik mexanizmlari ishlab chiqilgan. Foydalanuvchi parolini almashtirishi yoki unutgan holatda tiklashi mumkin. Parolni tiklash jarayonida foydalanuvchining elektron pochta manziliga yangi parol yuboriladi. Mazkur mexanizm foydalanuvchi akkauntining xavfsizligini ta’minlashga xizmat qiladi.

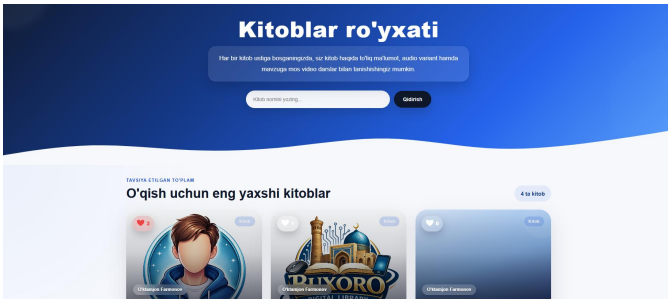
Shuningdek, tizimda foydalanuvchi profili bilan ishlash imkoniyati ham mavjud. Profil sahifasida foydalanuvchi o‘zining shaxsiy ma’lumotlarini tahrirlashi, avatar yuklashi hamda parolini o‘zgartirishi mumkin. Profil ma’lumotlarini faqat autentifikatsiyadan o‘tgan foydalanuvchi boshqara oladi.



2-rasm. Profil sahifasi

Elektron kutubxona tizimida foydalanuvchilar uchun kitoblar va video darslardan foydalanish imkoniyatlari yaratilgan. Tizim kitoblarni qidirish, ko‘rish, yuklab olish va video materiallarni tomosha qilish funksiyalarini qo‘llab-quvvatlaydi. Har bir resurs bo‘yicha ko‘rishlar soni, yuklab olish statistikasi va foydalanuvchi faolligi saqlanadi.

Tizimning muhim imkoniyatlaridan biri foydalanuvchining shaxsiy o‘quv faoliyatini boshqarish modulidir. Foydalanuvchi o‘ziga yoqqan kitoblarni saqlashi, o‘qilayotgan resurslarni belgilashi va o‘qish tarixini yuritishi mumkin. Bu imkoniyatlar foydalanuvchiga kerakli materiallarga tezkor murojaat qilish va o‘quv jarayonini individuallashtirish imkonini beradi.



3-rasim. Kitoblar ro‘yxati sahifasi

Tizimda kontent yuklash moduli ham ishlab chiqilgan bo‘lib, foydalanuvchilar yangi resurslarni tizimga joylashtirishlari mumkin. Yuklanayotgan fayllar server tomonidan tekshiriladi va faqat ruxsat etilgan formatdagi fayllar qabul qilinadi. Yuklangan materiallar turiga qarab maxsus kataloglarda saqlanadi. Umuman olganda, ishlab chiqilgan elektron kutubxona tizimi foydalanuvchilarga zamonaviy, qulay va xavfsiz elektron ta’lim muhiti yaratib beradi. Tizimning funksional imkoniyatlari foydalanuvchi faoliyatini samarali tashkil etish va ta’lim resurslaridan unumli foydalanishga xizmat qiladi.

Sinov senariysi	Kutilgan natija	Haqiqiy natija	Holati
Ro‘yxatdan o‘tish - to‘g‘ri ma'lumot	Akkaunt yaratiladi, login	Muvaffaqiyatli	O‘tdi ✓
Login - noto‘g‘ri parol	Xato xabari qaytadi	Xato ko‘rsatildi	O‘tdi ✓
Parol tiklash oqimi	Token bilan tasdiqlash	To‘liq ishladi	O‘tdi ✓
Fayl yuklash - to‘g‘ri format	Kontent ro‘yxatda ko‘rinadi	Muvaffaqiyatli	O‘tdi ✓
Fayl yuklash - noto‘g‘ri format	Rad etiladi, xabar beriladi	Rad etildi	O‘tdi ✓
liked_books - saqlash	Resurs saqlanadi	Muvaffaqiyatli	O‘tdi ✓

Profil tahrirlash	Faqat o‘z profili	To‘g‘ri ishladi	O‘tdi ✓

1-jadval. Foydalanuvchi va kontent modullari bo‘yicha sinov natijalari

Sinov natijalari barcha asosiy foydalanuvchi oqimlari va kontent boshqaruvi funksiyalarining muvaffaqiyatli ishlashini tasdiqlamoqda. Ayniqsa, noto‘g‘ri kiritmalardagi validatsiya va fayl tekshiruvi mexanizmlari tizim xavfsizligi va ma'lumotlar sifatini ta'minlashda muhim rol o‘ynamoqda.

Xulosa: Ushbu maqolada AKT fanlari elektron kutubxonasidagi foydalanuvchi boshqaruvi va kontent modullarining dasturiy amalga oshirilishi tahlil qilindi. Django auth tizimi asosida qurilgan autentifikatsiya, profil boshqaruvi va parol tiklash oqimlari NIST va OWASP tavsiyalariga to‘liq javob beradi.

Kontent yuklash moduli metadata majburiyligini ta'minlab, fayl tekshiruvi orqali tizim xavfsizligini mustahkamlaydi. Shaxsiy o‘quv faoliyati modullari (liked_books, reading_books) foydalanuvchi motivatsiyasini oshirib, o‘quv jarayonini shaxsiylashtirishga xizmat qiladi. Funksional sinov natijalari barcha modullarning muvaffaqiyatli ishlashini tasdiqladi.

Kelajakda ushbu modullarni REST API orqali mobil ilovaga ulash, o‘qish progressini vizual ko‘rsatish va AI asosidagi tavsiya tizimini qo‘shish rejalashtirilmoqda. Bu esa platformaning pedagogik samaradorligini yanada oshirishga zamin yaratadi.

FOYDALANILGAN ADABIYOTLAR

1. Nielsen J. 10 Usability Heuristics for User Interface Design. https://www.nngroup.com/articles/ten-usability-heuristics/
2. W3C. Web Content Accessibility Guidelines (WCAG). https://www.w3.org/TR/wcag/
3. Django Documentation. User authentication in Django. https://docs.djangoproject.com/en/dev/topics/auth/
4. NIST SP 800-63B. Digital Identity Guidelines: Authentication and Lifecycle Management. https://pages.nist.gov/800-63-4/sp800-63b.html
5. OWASP ASVS Project. Application Security Verification Standard https://github.com/OWASP/ASVS
6. Django Documentation. File uploads. https://docs.djangoproject.com/en/6.0/topics/http/file-uploads/
7. O‘ktamjon F. R. AKT fanlarining elektron kutubxonasini yaratish. Bitiruv malakaviy ishi. — Kattaqo‘rg‘on filiali, SamDU, 2026.
8. Django Documentation. Form and field validation. https://docs.djangoproject.com/en/4.2/ref/forms/validation/