Книжечки-Обложечки  
  
Проект выполнила Кожемякова Елизавета, Николаева Кристина Михайловна

**Описание реализации:**

1. основной программный код

**А**. run.py

Запуск работы программы:

from shop import app  
if \_\_name\_\_ =="\_\_main\_\_":  
 app.run(debug=True)

**В.** Shop

* 1. admin
     1. Rountes.py

1. Создание страницы админа
2. Вход и регистрация (см пункт а)
   1. customers
      1. Регистрация

form = CustomerRegisterForm()  
if form.validate\_on\_submit():  
 hash\_password = bcrypt.generate\_password\_hash(form.password.data)  
 register = Register(name=form.name.data, username=form.username.data, email=form.email.data, password=hash\_password)  
 db.session.add(register)  
 flash(f'Добро пожаловать, {form.name.data}! Спасибо за регистрацию!', 'success')  
 db.session.commit()  
 return redirect(url\_for('login'))  
return render\_template('customer/register.html', form=form)

* + 1. Вход

form = CustomerLoginFrom()  
if form.validate\_on\_submit():  
 user = Register.query.filter\_by(email=form.email.data).first()  
 if user and bcrypt.check\_password\_hash(user.password, form.password.data):  
 login\_user(user)  
 flash('Вы успешно вошли!', 'success')  
 next = request.args.get('next')  
 return redirect(next or url\_for('home'))  
 flash('Неправильный логин или пароль','danger')  
 return redirect(url\_for('customerLogin'))  
   
return render\_template('customer/login.html', form=form)

* 1. Favorites
     1. Добавление книги в избранное- класс AddFav

if request.method =="POST":  
 DictItems = {product\_id:{'name':product.name,'author':product.author,'image':product.image}}  
 if 'FavoriteBooks' in session:  
 print(session['FavoriteBooks'])  
 if product\_id in session['FavoriteBooks']:  
 for key, item in session['FavoriteBooks'].items():  
 if int(key) == int(product\_id):  
 session.modified = True

* + 1. Функция detFav

Если в избранном нет книг, что мы переходим на главную страницу

if 'FavoriteBooks' not in session or len(session['FavoriteBooks']) <= 0:  
 return redirect(url\_for('home'))  
return render\_template('products/favorites.html',genres=genres())

1. Также в избранном можно удалить книгу – функция deleteitem

ession.modified = True  
for key, item in session['FavoriteBooks'].items():  
 if int(key) == id:  
 session['FavoriteBooks'].pop(key, None)  
 return redirect(url\_for('getFav'))

* 1. produce
     1. Book\_obj

Создание полей для ввода ( название книги, автор, описание, обложка)

* + 1. Создание базы данных( таблицы с книгами)

\_\_seachbale\_\_ = ['name', 'desc']  
id = db.Column(db.Integer, primary\_key=True)  
name = db.Column(db.String(80), nullable=False)  
author = db.Column(db.String(80), nullable=False)  
desc = db.Column(db.String, nullable=False)  
  
genre\_id = db.Column(db.Integer, db.ForeignKey('genre.id'),nullable=False)  
genre = db.relationship('Genre', backref=db.backref('genre', lazy=True))  
  
image = db.Column(db.String(150), nullable=False)

* + 1. Routes
       1. Функция result

Поиск по названию книги

searchword = request.args.get('q')  
products = AddBook.query.msearch(searchword, fields=['name','desc'] , limit=6)  
return render\_template('products/result.html',products=products,genres=genres())

* + - 1. Функция home- возвращение на главную страницу

page = request.args.get('page',1, type=int)  
products = AddBook.query.order\_by(AddBook.id.desc()).paginate(page=page, per\_page=8)  
return render\_template('products/index.html', products=products, genres=genres())

* + - 1. Создание и редактирование жанров и книг

Функция addgenre, updategenre, delategenre, addproduct, updateprodute, delateproduct

* 1. Statis

Состоит из 3 папок, в которых находятся файлы bystrap и картинки

* 1. Templates

В этой папке находятся все файлы html, необходимые для разметки страницы





 





 

 



