ПРИВАТНИЙ ЗАКЛАД

«МОЛОДІЖНИЙ НАВЧАЛЬНИЙ ЦЕНТР

ІМЕНІ СВЯТОГО ІВАНА БОСКО»

**ПІДСУМКОВИЙ КОНТРОЛЬ**

|  |  |
| --- | --- |
| Професія | *Оператор з обробки інформації та програмного забезпечення* |
| Код | *4113* |
| Кваліфікація | *Оператор з обробки інформації та програмного забезпечення ІІ категорії* |

Учня групи ІТ-21

Ніколаєв А.О.

Львів 2022

Зміст

**Розділ 1. Опис предметної області**

1.1.Передумови роботи.

1.2.Програми аналоги.

1.3.Функціональність системи.

**Розділ 2. Проектування та конструювання**

2.1. Проектування та конструювання бази даних.

2.2. Проектування та конструювання інтерфейсу додатку.

2.3. Конструювання парсера даних.

2.4. Конструювання телеграм бота.

**Розділ 1. Опис предметної області**

**1.1 Передумови роботи**

Важлива роль книжок в нашому житті це те що у кожної людини є певний словниковий запас слів в голові, за допомогою яких висловлюють свою думку, якщо читати багато книжок можна поповнювати свій словниковий запас, розширити кругозір на світ, дізнатись багато чого нового і цікавого, поліпшити свою пам’ять, поліпшити фантазію.

Книги в наш час це є дуже сильна і важлива річ, оскільки з книжок можна зачерпнути дуже багато знань, для цього був створений сайт і телеграм бот під назвою “Книгарня” він полегшує пошук книжок для учнів, студентів, викладачів, вчителів і всіх інших.

Книги були створенні з середині XV століття і використовуються по теперішній час, але в середині XV століття це була просто шкіра з надписами зшита якоюсь ниткою, але прогресс не стоїть на місці і всі технології поліпшуються , появляються нові і помирають старі, замість шкіри почали використовувати папір, появились обкладенки і друкарські машинки для друкування книжок, також на ринку появились електронні книги, оскільки не всі ноутбуки/телефони/планшети обладнані захисним склом щоб не погіршувався зір , було винайдено електронні-книги, розміром вона як стандартна сторінка книги, але в більшості вони працюють на чорнилах саме тому вони не портять зір, функціонал в неї обмежений, можна скачати книгу відкрити її і перегортати сторінки, плюси в тому що вам не потрібно носити з собою дуже товсті і важкі книги, ви не можете забути якусь одну книгу так як в любий момент можете скачати нову, це дуже компактний і ефективний пристрій на сьогоднішній час.

**1.2 Программи аналоги**

**Приклад системи аналога № 1:**

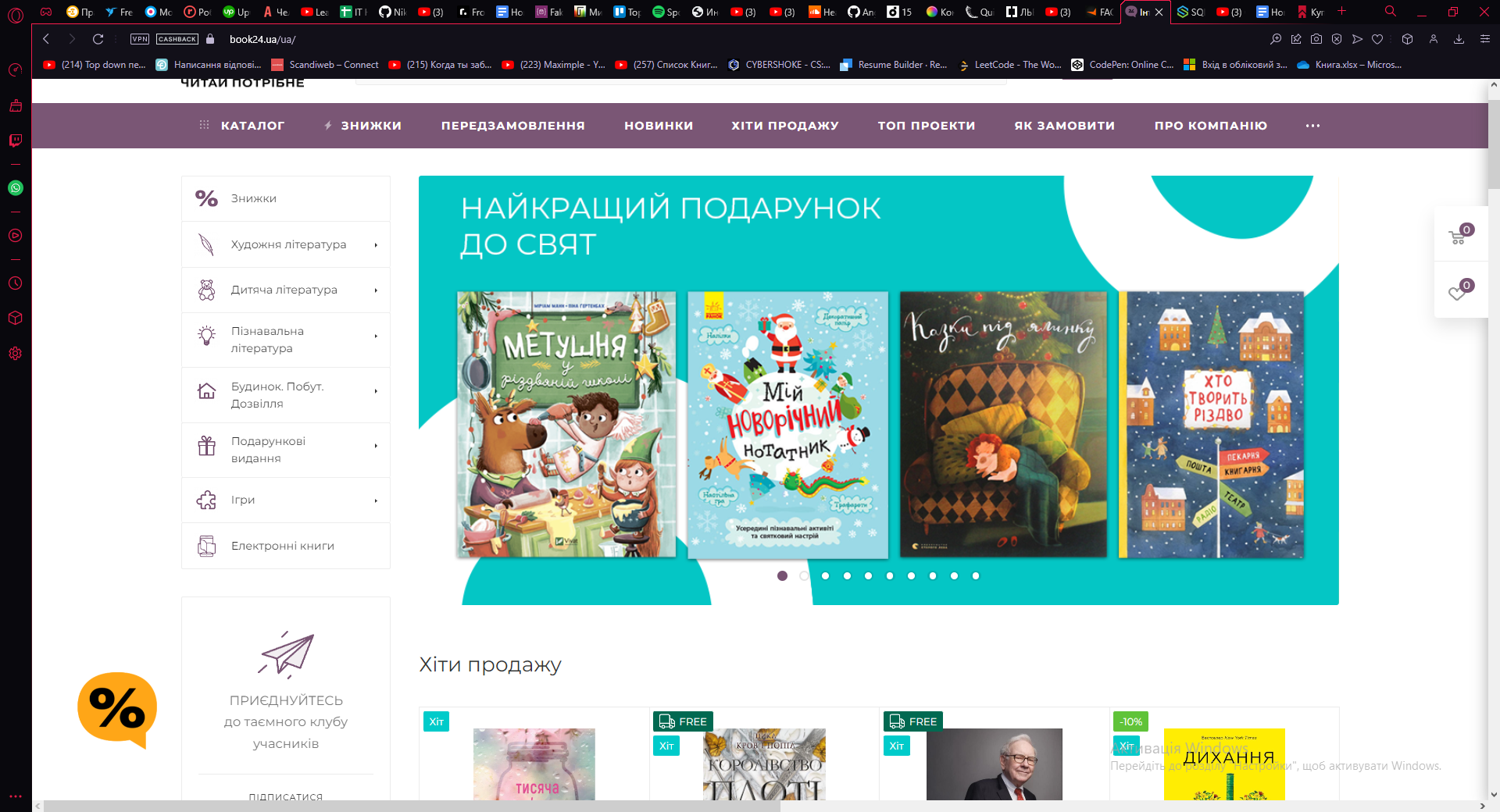
<https://book24.ua/ua/books/>

**Основне призначення:**

* Надати користувачу максимально великий обсяг книжок для пошуку і подальшої покупки за для прочитання їх.

**Основні функції** :

* Сортування по категоріям і під категоріям товарів
* Пошук книжок по їх назві



**Переваги системи:**

* Присутня служба підтримки
* Сортування книжок по категоріям і під категоріям
* Пошук книжок по назві
* Актуальні ринкові ціни
* Постійні знижки

**Недоліки системи:**

* Повільна робота системи
* Служба підтримки відповідає з затримкою
* Зазначених категорій/під категорій досить мало
* Немає системи фільтрації товару

**Приклад системи аналога № 2:**

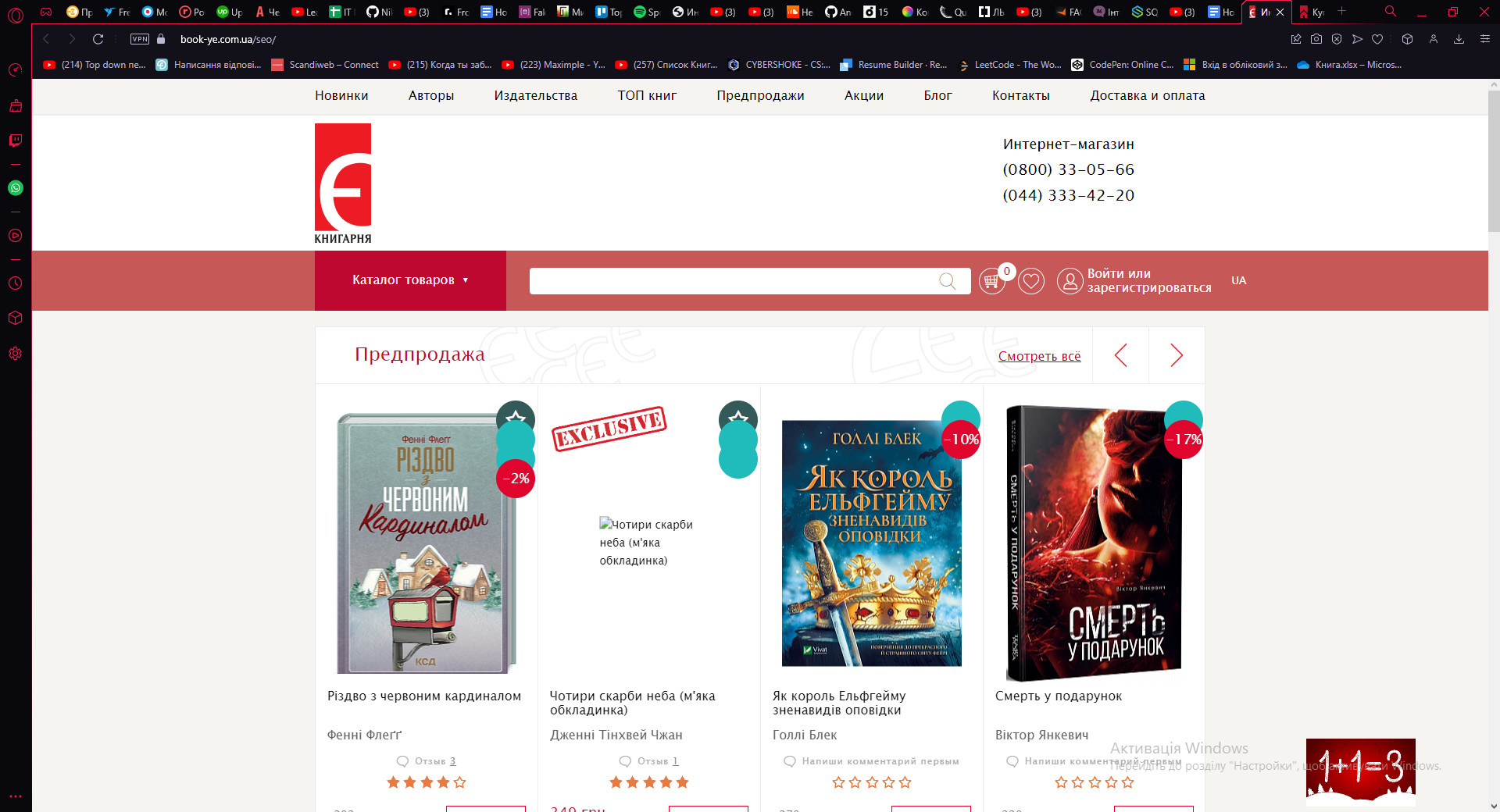
[**https://book-ye.com.ua/seo/**](https://book-ye.com.ua/seo/)

**Основне призначення:**

* Надати користувачу обсяг книжок для пошуку і подальшої покупки за для прочитання їх.

**Основні функції:**

* Сортування по категоріям і під категоріям товарів
* Пошук книжок по їх назві, категорії, ціні, автору з автозаповненням заповнювального поля
* Сортування по новинкам , топом продаж і так далі



**Переваги системи:**

* Присутня служба підтримки
* Сортування книжок по категоріям і під категоріям
* Автодоповнення поля введення для пошуку
* User Friendly Interface
* Спроможність реєстрації користувача

**Недоліки системи:**

* Велика кількість спливаючих вікон
* Дуже повільна швидкість системи
* Високі ціни на товар
* Немає системи фільтрації товару

**Приклад системи аналога № 3:**

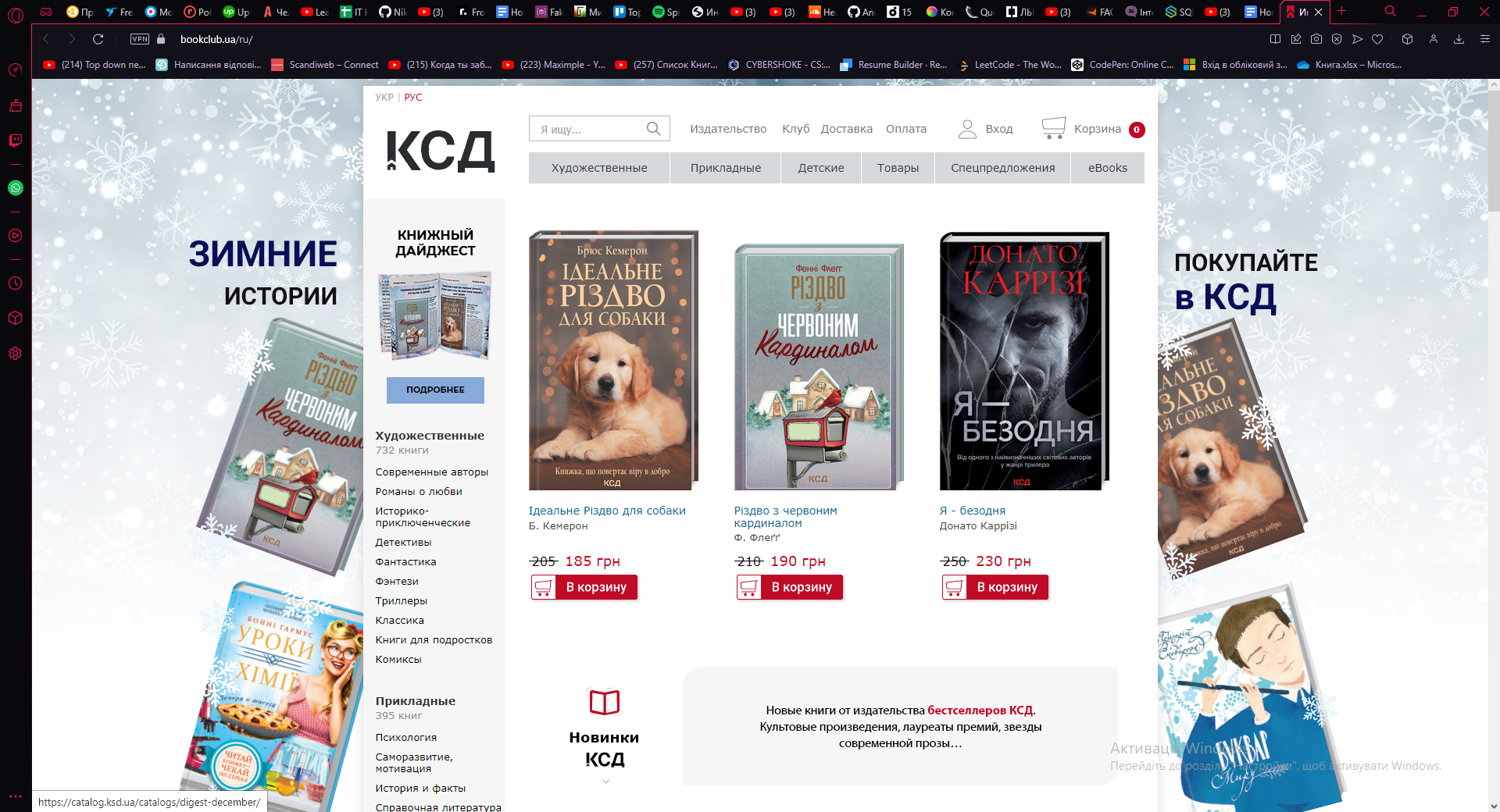
[**https://bookclub.ua/ru**](https://bookclub.ua/ru)

**Основне призначення:**

* Надати користувачу максимально великий вибір книжок для подальшої покупки

**Основні функції:**

* Пошук книжок по назві, жанру, автору
* Система реєстрації користувача
* Особистий кабінет користувача
* Корзина користувача



**Переваги системи**

* Швидка робота сайту
* Візуально приємний інтерфейс

**Недоліки системи:**

* Російськомовний сайт
* Немає служби підтримки
* Не зручний інтерфейс для використання

**1.3 Функціональність системи**

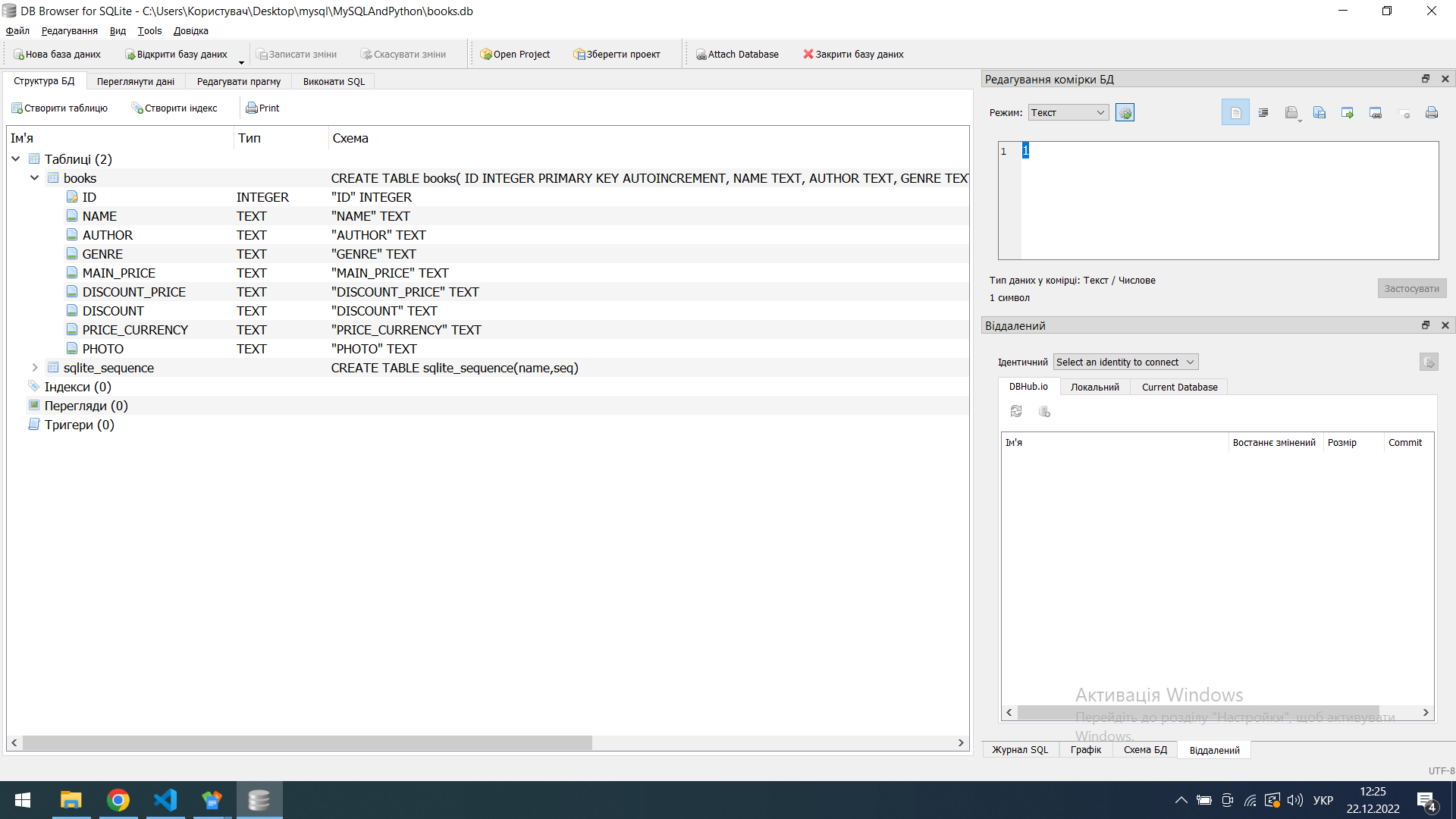
На основі проведеного аналізу систем-аналогів було визначено функції конструйованої системи:

* Парсинг бази даних для постійного поповнення товарами
* Додавання товарів з користувацької сторони
* Видалення товарів
* Виведення бази данних для користувача
* Пошук по назві

**Розділ 2. Проектування та конструювання**

**2.1 Проектування та конструювання бази даних**

Для роботи проектованої системи, було розроблено одну базу даних структура яких представлена на рис.2.1

Рисунок 2.1 Структура Бази даних books

**Створення бази данних -**

def create\_table(self):

        self.cursor.execute("""CREATE TABLE IF NOT EXISTS books(

            ID INTEGER PRIMARY KEY AUTOINCREMENT,

            NAME TEXT,

            AUTHOR TEXT,

            GENRE TEXT,

            MAIN\_PRICE TEXT,

            DISCOUNT\_PRICE TEXT,

            DISCOUNT TEXT,

            PRICE\_CURRENCY TEXT,

            PHOTO TEXT

            );

            """)

Опис полів бази данних -

* ID унікальний порядковий номер кожного запису(книги)
* NAME назва книги
* AUTHOR автор книги
* GENRE жанр книги
* MAIN\_PRICE основна ціна яка виводиться завжди
* DISCOUNT\_PRICE ціна зі знижкою, виводиться тільки тоді коли присутня
* DISCOUNT знижка показується в відсотках
* PRICE\_CURRENCY валюта
* PHOTO фотографія книги

**2.2 Проектування та конструювання інтерфейсу додатку**

Для реалізації проектованої системи потрібно розробити користувацький інтерфейс

Користувацький інтерфейс розроблений за допомогою мікро-фремворку Flask приклад коду

# importing all that we need to main program

from flask import Flask, render\_template, request

from DataBase\_Wrapper import DB

# initializing flask and dabatase

db = DB()

app = Flask(\_\_name\_\_)

# creating table if need

db.create\_table()

# making rout to main page where we will send and receive data

@app.route("/", methods = ['GET','POST'])

def hello\_world():

    # requesting from db book with 1st id

    book = db.first\_book()

    # taking data form form input and searching it in database

    if request.method == 'POST':

        name = request.form['name']

        book = db.find\_book\_for\_name(name)

#if not requested search from db, it will display first book

    books = db.return\_all\_database()

    book\_name = book[0][1]

    book\_author = book[0][2]

    book\_price = book[0][4]

    book\_currency\_price = book[0][7]

    book\_img = book[0][8]

    if book[0][6] or book[0][7] == "None":

        book\_discount\_price = ""

        book\_discount\_procent = ""

        book\_currency\_discount = ""

    else:

        book\_discount\_procent = book[0][6]

        book\_currency\_discount = book[0][7]

    return render\_template("index.html",

    book\_discount\_procent = book\_discount\_procent,

    book\_name = book\_name,

    book\_author = book\_author,

    book\_price = book\_price,

    book\_discount\_price = book\_discount\_price,

    book\_currency\_price = book\_currency\_price,

    book\_currency\_discount = book\_currency\_discount,

    book\_img = book\_img,

    books = books[0:10]

    )

# running our flask app

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**2.3 Конструювання парсера даних**

За для повного функціоналу проекту нам потрібен парсер даних щоб швидко добавляти нові поставки книг на сайт по вигідним цінам, найліпше для цього рішення це написати парсер товарів, який буде зчитувати дані і після чого записувати в базу даних з якої вже будуть виводитись на нашому сайті або телеграм боті

Код парсера даних

# importing all that we will work with

import requests

from bs4 import BeautifulSoup as BS

from DataBase\_Wrapper import DB

# initializing our db

db = DB()

# making list with our genre name and count of last page

genre\_list = [

    {"genre":"uzhasy","pages":4},

    {"genre":"fentezi","pages":26},

    {"genre":"fantastika","pages":20},

    {"genre":"klassicheskaya-ukrlit","pages":10},

    {"genre":"sovremennaya-ukrlit","pages":24},

    {"genre":"knigi\_avtorov\_nezavisimoy\_ukrainy","pages":7},

    {"genre":"klassicheskaya","pages":13},

    {"genre":"sovremennaya","pages":16},

    {"genre":"detektiv\_","pages":39},

    {"genre":"triller","pages":11},

    {"genre":"klassicheskaya\_proza","pages":68},

    {"genre":"sovremennaya\_proza\_nov","pages":175},

    {"genre":"lyubovnyy\_roman\_new","pages":26},

    {"genre":"istoricheskiy\_roman\_new","pages":23},

    {"genre":"priklyuchencheskiy\_roman","pages":7},

    {"genre":"graficheskiy\_roman","pages":17},

    {"genre":"yumor\_new","pages":3},

    {"genre":"sborniki-rasskazov","pages":3},

    {"genre":"folklor","pages":4}

]

i = 0

cycle for all genre list

while i <= len(genre\_list):

    print(f"Genre list number:{i}")

    pages = 2

# taking values such as genre name and last page count

    for key,value in genre\_list[i].items():

        if key == "genre":

            genre = value

        elif key == "pages":

            pages = value

# counting pages form 2 becouse on first page syntax is not simillar

        for page in range(2,pages+1):

            print(f"Page number:{page}")

        # making get request to our page

            r = requests.get(f"https://book24.ua/ua/catalog/{genre}/?PAGEN\_1={page}")

            # taking all html from url

            html = BS(r.content, 'html.parser')

            # make cycle to take all books in our html

            for info in html.select(".catalog\_item\_wrapp > .inner\_wrap"):

                # taking title for our book

                title = info.select(".item-title > a > span")

                title = title[0].text

                # our price currency

                price\_currency = " грн."

                # taking author of the book

                try:

                    author = info.select(".item\_info > .sa\_block > .article\_block > .font\_sxs > a")

                    author = author[0].text

                except:

                    author = "None"

                # taking discount price of the book

                try:

                    discount\_price = info.select(".item\_info > .cost > .price\_matrix\_wrapper > .prices-wrapper > .price > .values\_wrapper > .price\_value")

                    discount\_price = discount\_price[0].text

                except:

                    discount\_price = "None"

                    discount = "None"

                # taking main price of the book

                try:

                    price = info.select(".item\_info > .cost > .price\_matrix\_wrapper > .price > .values\_wrapper > .price\_value")

                    price = price[0].text

                except:

                    try:

                        price = info.select(".item\_info > .cost > .price\_matrix\_wrapper > .prices-wrapper > .discount > .values\_wrapper > .price\_value")

                        price = price[0].text

                    except:

                        try:

                            price = info.select(".item\_info > .price\_matrix\_wrapper > .price > .values\_wrapper > .price\_value")

                            price = price[0].text

                        except:

                            price = "None"

                # taking procent of discout of the book

                try:

                    discount = info.select(".image\_wrapper\_block > .stickers > div > .sticker\_sovetuem")

                    discount = discount[0].text

                except:

                    discount = "None"

                # taking photo url of the book

                try:

                    photo = info.select(".image\_wrapper\_block > .thumb > .section-gallery-wrapper > .section-gallery-wrapper\_\_item > img")

                    photo = "https://book24.ua" + photo[0].attrs['data-src']

                except:

                    photo = "None"

                # display it in console if u need it

                # print(f'Title: {title}')

                # print(f'Author: {author}')

                # print(f'Genre: {genre}')

                # print(f'Price without discount: {price} + {price\_currency}')

                # print(f'Price with discount {discount\_price} + {price\_currency}')

                # print(f'Photo {photo}')

                # print(f'Discount procent {discount}')

                # print('---------------------------------------------------------------------------------------------------------')

                # all our data in variable data

                data = (str(title),str(author),str(genre),str(price),str(discount\_price),str(discount),str(price\_currency),str(photo))

                # inserting all book data in our database using function called add

                db.add(data = data)

                page += 1

    i += 1

**2.4** Конструювання телеграм бота

Для роботи проекту також потрібен телеграм бот , як додатковий функціонал в якому будуть реалізовані такі функції як:

Вивід списка команд з поясненням що вони роблять

/help

Вивід бази данних(10 книжок за 1 запит)

/all\_books

Пошук книги по її назві

/find\_for\_name

Вихід на головну сторінку клавіатури

/exit

Перелистування сторінок вперед/назад, працює тільки для вивода бази данних

/page\_back /page\_forward