

Book Store

Management System

Best In The Market

Ankush Roy

Umakanta Academy (English)

ankush341111@gmail.com

Chandrapur, Agartala

Shibam Roy

Umakanta Academy (English)

shibamisgay@gmail.com

Santipara, Agartala

Uday Barman

Umakanta Academy (English)

uday@gmail.com

India

Souharda Saha

Umakanta Academy (English)

souharda@gmail.com

India

Subrangshu

Umakanta Academy (English)

subrangshu@gmail.com

India

Preface

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliquam quaerat voluptatem. Ut enim aequo doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut.

Contents

Preface	ii
Contents	iii
1. Introduction	2
2. Methodology	4
3. Source Code	8
A. Table of Properties	16
Bibliography	17

Introduction

Etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitibus aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae sine metu degendae praesidia firmissima. – Filium morte

multavit. – Si sine causa, nollem me ab eo delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglexerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluisset quam illum etiam ipsum dedocere. Sol Democrito magnus videtur, quippe homini erudito in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omnino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatissimi. Mihi quidem videtur, inermis ac nudus est. Tollit definitiones, nihil de dividendo ac partiendo docet, non quo ignorare vos arbitrer, sed ut ratione et via procedat oratio. Quaerimus igitur, quid sit extremum et ultimum bonorum, quod omnium philosophorum sententia tale debet esse, ut eius magnitudinem celeritas, diurnitatem allevatio consoletur. Ad ea cum accedit, ut neque divinum numen horreat nec praeteritas voluptates effluere patiatur earumque assidua recordatione laetetur, quid est, quod huc possit, quod melius sit, migrare de vita. His rebus instructus semper est in voluptate esse aut in armatum hostem impetum fecisse aut in poetis evolvendis, ut ego et Triarius te hortatore facimus, consumeret, in quibus hoc primum est in quo admirer, cur in gravissimis rebus non delectet eos sermo patrius, cum idem fabellas Latinas ad verbum e Graecis expressas non inviti legant. Quis enim tam inimicus paene nomini Romano est, qui Ennii Medeam aut Antiopam Pacuvii spernat aut reiciat, quod se isdem Euripidis fabulis delectari dicat, Latinas litteras oderit? Synephebos ego, inquit, potius Caecilii aut Andriam Terentii quam utramque Menandri legam? A quibus tantum dissentio, ut, cum Sophocles vel optime scripserit.

$$e = mc^2 \quad (1)$$

Methodology

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliquam quaerat voluptatem. Ut enim aequo doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitibus aut rerum necessitatibus saepe eveniet, ut et voluptates repudianda sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae sine metu degendae praesidia firmissima. – Filium morte mul-

tavit. – Si sine causa, nollem me ab eo delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglexerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluisset quam illum etiam ipsum dedocere. Sol Democrito magnus videtur, quippe homini eruditio in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omnino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatissimi. Mihi quidem videtur, inermis ac nudus est. Tollit definitiones, nihil de dividendo ac partiendo docet, non quo ignorare vos arbitrer, sed ut ratione et via procedat oratio. Quaerimus igitur, quid sit extremum et ultimum bonorum, quod omnium philosophorum sententia tale debet esse, ut eius magnitudinem celeritas, diurnitatem allevatio consoletur. Ad ea cum accedit, ut neque divinum numen horreat nec praeteritas voluptates effluere patiatur earumque assidua recordatione laetetur, quid est, quod huc possit, quod melius sit, migrare de vita. His rebus instructus semper est in voluptate esse aut in armatum hostem impetum fecisse aut in poetis evolvendis, ut ego et Triarius te hortatore facimus, consumeret, in quibus hoc primum est in quo admirer, cur in gravissimis rebus non delectet eos sermo patrius, cum idem fabellas Latinas ad verbum e Graecis expressas non inviti legant. Quis enim tam inimicus paene nomini Romano est, qui Ennii Medeam aut Antiopam Pacuvii spernat aut reiciat, quod se isdem Euripidis fabulis delectari dicat, Latinas litteras oderit? Synephebos ego, inquit, potius Caecilii aut Andriam Terentii quam utramque Menandi legam? A quibus tantum dissentio, ut, cum Sophocles vel optime scripserit Electram, tamen male conversam Atilii mihi legendam putem, de quo Lucilius: 'ferreum scriptorem', verum, opinor, scriptorem tamen, ut legendus sit. Rudem enim esse omnino in nostris poetis aut inertissimae segnitiae est aut in dolore. Omnis autem privatione doloris putat Epicurus terminari summam voluptatem, ut postea variari voluptas distingue possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam.

2.1. Sub-Section

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliquam quaerat voluptatem. Ut enim aequo doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitibus aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae sine metu degendae praesidia firmissima. – Filium morte multavit. – Si sine causa, nollem me ab eo delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglexerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluisset quam illum etiam ipsum dedocere. Sol Democrito magnus videtur, quippe homini erudito in geometriaque perfecto, huic pedalis fortasse; tantum enim esse omnino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatissimi. Mihi quidem videtur, inermis ac nudus est. Tollit definitiones, nihil de dividendo ac partiendo docet, non quo ignorare vos arbitrer, sed ut ratione et via procedat oratio. Quaer-

imus igitur, quid sit extremum et ultimum bonorum, quod omnium philosophorum sententia tale debet esse, ut eius magnitudinem celeritas, diurnitatem allevatio consoletur. Ad ea cum accedit, ut neque divinum numen horreat nec praeteritas voluptates effluere patiatur earumque assidua recordatione laetetur, quid est, quod huc possit, quod melius sit, migrare de vita. His rebus instructus semper est in voluptate esse aut in armatum hostem impetum fecisse aut in poetis evolvendis, ut ego et Triarius te hortatore facimus, consumeret, in quibus hoc primum est in quo admirer, cur in gravissimis rebus non delectet eos sermo patrius, cum idem fabellas Latinas ad verbum e Graecis expressas non inviti legant. *Quis enim tam inimicus paene nomini Romano est, qui Ennii Medeam aut Antiopam Pacuvii spernat aut reiciat, quod se isdem Euripidis fabulis delectari dicat, Latinas litteras oderit?* Synephebos ego, inquit, potius Caecilii aut Andriam Terentii quam utramque Menandri legam? A quibus tantum dissentio, ut, cum Sophocles vel optime scripserit Electram, tamen male conversam Atilii mihi legendam putem, de quo Lucilius: 'ferreum scriptorem', verum, opinor, scriptorem tamen, ut legendus sit. Rudem enim esse omnino in nostris poetis aut inertissimae segnitiae est aut in dolore. Omnis autem privatione doloris putat Epicurus terminari summam voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam.

Source Code

```
main.py

import sqlite3
import os
import sys
import datetime
from rich.console import Console
from rich.table import Table

db_was_missing = not os.path.exists("./data.db")

con = sqlite3.connect("data.db")
cur = con.cursor()
c=Console()

# Functions
def create_table():
    """
    Creates required tables, namely books and authors.
    We kept them in 2 different tables so that we can store multiple authors for the
    same book.
    """

    cur.execute("""
        create table if not exists
            books(
                book_id integer primary key autoincrement,
                title text,
                genre text,
                language text,
                price float,
                publish_year date,
                added_date date,
                stocks integer
            )
    """)


```

```

cur.execute("""
    create table if not exists
        authors(
            book_id integer,
            name text,
            foreign key (book_id) references books(book_id)
        )
""")
"""

def add_data():
    """
        Adds some data so that we can test these functions. NOT TO BE INCLUDED IN THE
        FINAL PROJECT,
        ONLY FOR TESTING PURPOSES
        And the prices are in USD, shouldn't be a problem though, cause its only for
        testing anyways
    """

    # the data is AI generated, so it could be wrong, but who cares
    """Insert 10 sample books with authors for testing."""
    sample_books = [
        # (title, genre, language, price, publish_year, added_date, stocks, [authors])
        ("To Kill a Mockingbird", "Fiction", "English", 12.99, "1960-07-11", "2025-11-01",
         10, ["Harper Lee"]),
        ("1984", "Dystopian", "English", 10.50, "1949-06-08", "2025-11-01", 8, ["George
Orwell"]),
        ("The Great Gatsby", "Classic", "English", 11.20, "1925-04-10", "2025-11-01",
         6, ["F. Scott Fitzgerald"]),
        ("One Hundred Years of Solitude", "Magical Realism", "Spanish", 14.80,
         "1967-06-05", "2025-11-01", 5, ["Gabriel García Márquez"]),
        ("Pride and Prejudice", "Romance", "English", 8.50, "1813-01-28", "2025-11-01",
         12, ["Jane Austen"]),
        ("The Hobbit", "Fantasy", "English", 13.75, "1937-09-21", "2025-11-01", 15,
         ["J.R.R. Tolkien"]),
        ("The Alchemist", "Adventure", "Portuguese", 10.00, "1988-04-15", "2025-11-01",
         7, ["Paulo Coelho"]),
        ("The Da Vinci Code", "Thriller", "English", 16.20, "2003-03-18", "2025-11-01",
         9, ["Dan Brown"]),
        ("The Book Thief", "Historical Fiction", "English", 12.50, "2005-03-14",
         "2025-11-01", 8, ["Markus Zusak"]),
        ("Good Omens", "Fantasy", "English", 13.99, "1990-05-01", "2025-11-01", 10,
         ["Neil Gaiman", "Terry Pratchett"]),
    ]

    for (title, genre, language, price, publish_year, added_date, stocks, authors)
in sample_books:
    data = [title, genre, language, price, publish_year, added_date, stocks]
    add_book(data, authors)

con.commit()

def print_table(data:list,cols=[]):
    new_data=[]
    for row in data:
        new_data.append(list(map(str,row)))

```

```

table=Table(title="Books",style="cyan")

if(len(cols)==0):
    for col in cur.description:
        table.add_column(col[0])
else:
    for col in cols:
        table.add_column(col)

for point in new_data:
    table.add_row(*point)
c.print(table)

def add_book(data:list,authors:list):
    """
    This function adds a new book to the "books" table.
    It takes all the data for a new row as a list, and adds that to the "books" table.
    Additionally, it also takes an authors list, and for each author it creates a new
    entry in the "authors" table which use the book_id column of "books" as a foreign key.
    """
    cur.execute("""
        insert into books (title, genre, language, price, publish_year, added_date,
stocks)
        values
        (?, ?, ?, ?, ?, ?, ?)
    """,data)

book_id=cur.lastrowid

for author in authors:
    cur.execute("""
        insert into authors (book_id, name)
        values
        (? ,?)
    """,(book_id,author))
con.commit()

c.print(f"[green] Book added with id {book_id}. [/green]")

def delete_book(bookId):
    """
    Deletes a particular book entry in the "books" table along with its respective
    "authors" entry
    """
    cur.execute("""
        delete from books
        where book_id=?
    """,(bookId,))
    cur.execute("""
        delete from authors
        where book_id=?
    """,(bookId,))

    con.commit()

```

```

def increment_stock(book_id):
    """
    Increments the stock of a particular book by 1
    """
    cur.execute("""
        update books
        set stocks=stocks+1
        where book_id=?
    """, (book_id,))

def decrement_stock(book_id):
    """
    Decrements the stock of a particular book by 1
    """
    cur.execute("""
        update books
        set stocks=stocks-1
        where book_id=?
    """, (book_id,))

def set_stock(book_id,value):
    """
    Sets the stock of a particular book to a certain value
    """
    cur.execute("""
        update books
        set stocks=? 
        where book_id=?
    """, (value,book_id))

def get_author(book_id):
    """
    Returns the author for a particular book
    """
    cur.execute("""
        select name from authors
        where book_id=?
    """, (book_id,))
    return cur.fetchall()

# USAGE

def title(s):
    c.print(f"[bold underline cyan] {s} [/bold underline cyan]")

def pause(msg="[gray]\nPress Enter to continue...[/gray]"):
    c.input(msg)

# helper to take inputs
def custom_input(prompt, cast=str, default=None, allow_empty=False):
    v = input(prompt).strip()
    if v == "" and default is not None:
        return default
    if v == "" and allow_empty:

```

```

    return ""

# type checking
if cast == str:
    return v
elif cast == int:
    if v.isdigit():
        return int(v)
    c.print("[red]Invalid input! enter a integer!![/red]")
elif cast == float:
    # ai generated ***
    if v.replace('.', '', 1).lstrip('-').isdigit():
        return float(v)
    c.print("[red]Invalid input: expected float.[/red]")
    # ***
else:
    c.print("[red]Invalid input[/red]")


# show details of a particular book using its id
def show_book_details(book_id):
    cur.execute("select * from books where book_id=?", (book_id,))
    book = cur.fetchone()
    if not book:
        c.print("[red]Book not found.[/red]")
        return

    table=Table(title="Book record",style="yellow")

    for col in cur.description:
        table.add_column(col[0])

    to_str=list(map(str,book))
    table.add_row(*to_str)
    c.print(table)

    authors = get_author(book_id)

    if not authors:
        c.print("[red]No Authors found![/red]")
    else:
        c.print("[yellow]\nAuthors:[/yellow]")
        for a in authors:
            c.print(" -", a[0])

# take data, use addBook func to add book to the db
def ui_add_book():
    # TODO: add guardrails for inputting
    title("Add new book")
    t = input("Title: ").strip()
    g = input("Genre: ").strip()
    lang = input("Language: ").strip()
    price = custom_input("Price (e.g. 12.99): ", cast=float, default=0.0)
    publish_year = input("Publish date (YYYY-MM-DD): ").strip() or
    datetime.date.today().isoformat()
    added_date = input("Added date (YYYY-MM-DD) [default today]: ").strip() or

```

```

datetime.date.today().isoformat()
stocks = custom_input("Initial stocks (integer): ", cast=int, default=0)
authors = input("Authors (comma separated): ").strip()
authors_list = [a.strip() for a in authors.split(",") if a.strip()]

data = [t, g, lang, price, publish_year, added_date, stocks]
add_book(data, authors_list)

# take a guess. commenting is tough
def ui_search_by_title():
    q = input("Search title substring: ").strip()
    output = cur.execute("select * from books where title like ?", (f"%{q}%",))
    if not output:
        c.print("[red]No book with the provided title found![/red]")
    print_table(output)

# .....
def ui_search_by_author():
    q = input("Search author substring: ").strip()
    output = cur.execute("""
        select b.*
        from books b, authors a
        where a.book_id = b.book_id
        and a.name like ?
        group by b.book_id
    """, (f"%{q}%",))
    if not output:
        c.print("[red]No book with the provided author found![/red]")
    print_table(output)

# this function returns the meaning of life
def ui_search_by_genre():
    q = input("Genre substring: ").strip()
    output = cur.execute("select * from books where genre like ?", (f"%{q}%",))
    if not output:
        c.print("[red]No book with the provided genre found![/red]")
    print_table(output)

# this function solves the problems of life
def ui_delete_book():
    ID = custom_input("Book id to delete: ", cast=int)
    cur.execute("select title from books where book_id=?", (ID,))
    row = cur.fetchone()
    if not row:
        c.print("[red]No such book.[/red]")
        return
    confirm = input(f"Delete '{row[0]}' (id {ID})? [y/n]: ").strip().lower()
    if confirm == 'y':
        delete_book(ID)
        c.print("[green]Deleted.[/green]")
    else:
        c.print("[yellow]Aborted.[/yellow]")

def ui_increment_stock():
    ID = custom_input("Book id: ", cast=int)

```

```

increment_stock(ID)
con.commit()
c.print("[green]Stock incremented.[/green]")

def ui_decrement_stock():
    ID = custom_input("Book id: ", cast=int)
    decrement_stock(ID)
    con.commit()
    c.print("[green]Stock decremented.[/green]")

def ui_set_stock():
    ID = custom_input("Book id: ", cast=int)
    val = custom_input("Set stocks to: ", cast=int)
    set_stock(ID, val)
    con.commit()
    c.print("[green]Stock set.[/green]")

# lists all the books from the db
def ui_list_all():
    output = cur.execute("select * from books")
    if not output:
        c.print("[red]No books present!![/red]")
    print_table(output)

def main_menu():
    while True:
        c.clear()
        title("BookStore – Text UI")
        print("""
        1) List all books
        2) View book details
        3) Add a book
        4) Delete a book
        5) Increment stock
        6) Decrement stock
        7) Set stock value
        8) Search by title
        9) Search by author
        10) Search by genre
        0) Quit
        """)
        choice = input("Choose an option: ").strip()
        match choice:
            case '1':
                ui_list_all()
                pause()
            case '2':
                book_id = custom_input("Book id: ", cast=int)
                show_book_details(book_id)
                pause()
            case '3':
                ui_add_book()
                pause()
            case '4':

```

```

        ui_delete_book()
        pause()
    case '5':
        ui_increment_stock()
        pause()
    case '6':
        ui_decrement_stock()
        pause()
    case '7':
        ui_set_stock()
        pause()
    case '8':
        ui_search_by_title()
        pause()
    case '9':
        ui_search_by_author()
        pause()
    case '10':
        ui_search_by_genre()
        pause()
    case '0':
        c.print("[yellow]Bye![/yellow]")
        break
    case _:
        c.print("[red]Invalid choice.[/red]")
        pause()

create_table()
if db_was_missing:
    add_data()
main_menu()

```

Table of Properties

- PNC - Permutation and combination
- CNC - Chennai National Champions
- PNC - Permutation and combination
- CNC - Chennai National Champions
- PNC - Permutation and combination
- CNC - Chennai National Champions
- PNC - Permutation and combination
- CNC - Chennai National Champions

Bibliography

- [1] A. Roy, S. Roy, S. Saha, S. Roy, and U. Barman, *Book Store Management System*. 2025. [Online]. Available: <https://github.com/ShibamRoy9826/BookStoreManagementSystem>
- [2] C. Naaktgeboren, “Lyceum: Academic Book Template in Typst.” [Online]. Available: <https://github.com/cnaak/lyceum.typ>
- [3] S. Arora, *Computer Science with Python Class XI*. Dhanpat Rai & Co., 2021. [Online]. Available: <https://www.amazon.in/Computer-Science-Python-Class-XI/dp/>
- [4] S. Arora, *Computer Science with Python Class XII*. Dhanpat Rai & Co., 2021. [Online]. Available: <https://www.amazon.in/Progress-Computer-Science-Python-XII/dp/>
- [5] Python Software Foundation, “Download Python | Python.org.” [Online]. Available: <https://www.python.org/downloads/>
- [6] Oracle Corporation, “MySQL Downloads.” [Online]. Available: <https://dev.mysql.com/downloads/>
- [7] Python Software Foundation, “Python Documentation.” [Online]. Available: <https://docs.python.org/3/>
- [8] Will McGugan, “Rich Documentation.” [Online]. Available: <https://rich.readthedocs.io/>