Library Management System

A Python-based Object-Oriented Programming (OOP) project that simulates a library management system with book inventory tracking, checkout, and return functionality.

Features

- **Book Management**: Create and manage book objects with title and author information
- **Availability Tracking**: Real-time tracking of book availability status
- **Checkout System**: Check out books with automatic availability updates
- **Return System**: Process book returns and update availability
- **Library Catalog**: Centralized library system to manage multiple books
- **Search Functionality**: Find books by title (case-insensitive)
- **Display Interface**: View all books in the library collection

Project Structure

```
library-management-system/
│
├── Building a Library Project.ipynb    # Main Jupyter Notebook
├── library_system.py                  # Converted Python script (optional)
├── README.md                          # This file
├── requirements.txt                   # Python dependencies (empty - no external deps)
└── LICENSE                            # License file
```

Classes and Methods

Class: `Book`

Represents an individual book in the library system.

Attributes:

- `title` (str): The title of the book
- `author` (str): The author of the book
- `available` (bool): Availability status (default: True)

Methods:

Method	Parameters	Returns	Description
`__init__()`	`title`, `author`	None	Initializes a new book object
`checkout()`	None	`bool`	Checks out the book if available; returns True on success, False if unavailable
`return_book()`	None	None	Returns the book and sets availability to True
`display_info()`	None	None	Prints book title and author information

Class: `Library`

Manages a collection of books and provides library operations.

Attributes:

- `books` (list): A list containing all Book objects in the library

Methods:

Method	Parameters	Returns	Description
`__init__()`	None	None	Initializes an empty library
`add_book()`	`book` (Book)	None	Adds a book to the library collection
`display_books()`	None	None	Displays information for all books in the library
`get_book_by_title()`	`title` (str)	`Book` or `None`	Searches for and returns a book by title (case-insensitive)

Key OOP Concepts Demonstrated

This project showcases several important Object-Oriented Programming principles:

1. **Encapsulation**: Data (attributes) and methods are bundled within classes
2. **Abstraction**: Complex operations are hidden behind simple method calls
3. **State Management**: Objects maintain their own state (e.g., book availability)
4. **Composition**: Library class contains and manages Book objects
5. **Method Design**: Each method has a single, clear responsibility

Contributing

Contributions are welcome! Here's how you can help:

1. Fork the repository
2. Create a feature branch (`git checkout -b feature/AmazingFeature`)
3. Commit your changes (`git commit -m 'Add some AmazingFeature'`)

4. Push to the branch (`git push origin feature/AmazingFeature`)
5. Open a Pull Request

Contribution Ideas:

- Fix the `get_book_by_title()` bug
- Add unit tests
- Implement suggested enhancements
- Improve documentation
- Add error handling

Learning Outcomes

This project is perfect for learning:

- Python Object-Oriented Programming (OOP)
- Class design and implementation
- State management in objects
- Collection management (lists)
- String manipulation and comparison
- Boolean logic and control flow

License

Acknowledgments

- Built as a learning project for Python OOP concepts
- Inspired by real-world library management systems
- Perfect for beginners learning object-oriented programming

Contact

For questions, suggestions, or feedback, please open an issue on GitHub.