Report and a Breakdown of my code:

1. Imported modules ‘json’ and ‘datetime’
2. Class definitions:
   1. ‘Book’ : used to represent the different attributes of a book (e.g. title, author, ISBN, etc)
   2. ‘Patron’: used to store attributes like name, ID, email, phone
   3. ‘Transaction’: To handle various transactions like checkouts, returns, fines, etc.
   4. ‘Library’: Used to manage the entire library system, including books, patrons, and transactions.
3. Initialization:
   1. I created instances of the Library, Book, Patron and Transaction classes to initialize the LMS system.
4. Book Class
   1. Defined the ‘Book’ class with the given attributes:
      1. Attributes: ‘title’, ‘author’, ‘ISBN’, ‘quantity’
      2. Methods: ‘display\_details’, ‘update\_quantity’, ‘result’
5. Patron Class:
   1. Define the ‘Patron’ class with the following attributes and methods:
      1. Attributes: ‘name’, ‘id’, ‘email’, ‘phone’, ‘borrowed\_books’
      2. Methods:‘display\_details’, ‘update\_quantity’, ‘result’
6. Transaction Class
   1. Defined the ‘Transaction’ class with the following attributes and methods:
   2. Attributes: ‘book’, ‘patron’, ‘Checkout\_date’
   3. Methods: ‘late\_fees’, ‘Checkout’, Checkin’
7. Library Class:
   1. Created a ‘Library’ class to implement the management of the library system:
   2. Attribues: ‘books’, ‘patrons’, ‘transactions’
   3. Methods: ‘search\_books’, ‘add\_book’, ‘remove\_bok’, ‘add\_patrons’, ‘remove\_patron’, ‘handle\_transaction’,’ checkout’,’ checkin’’ book\_report’,’ patron\_report’,’ trans\_report’
8. Main Flow of the Program:
   1. Outlined the sequence of operations within the main program flow, including adding books and patrons, handling transactions, and generating reports
9. Data Persistence:
   1. Implemented methods to keep files using JSON
10. User Interface:
    1. Develop a simple interface to interact with the Library Management System
    2. Allow users to perform operations like adding/removing books, managing patrons, handling transactions. Etc.

Start

Check if libraries imported

False

True

Class definitions

ERROR

Function definitions

Everything works?

True

False

ERROR

Display Result

Instructions:

Here are the instructions on how to use the code

1. Initialize the library
   1. First, create an instance of the ‘Library’ class. This is your starting point for managing books, patrons, and transactions.

python

library = Library()

1. To add books:
   1. Create a ‘Book’ object with the tile, author, ISBN, and quantity.
   2. Use the ‘add\_book’ method of the ‘Library’ class to add the book to the library

Example:

library = Library()

book = Book("Title", "Author", "ISBN", 5)

library.add\_book(book)

1. Adding Patrons:
   1. create a `Patron` object with the name, ID , email, and Phone.
   2. Use the `add\_patron` method of the `Library` class to add the patron.

library = Library()

patron = Patron("Name", "ID", "Contact Info")

library.add\_patron(patron)

1. Check out a book:
   1. To check out a book to a patron, first ensure the book is available (quantity > 0). Then use the ‘checkout’ method with the book's ISBN.

library = Library()

book = library.search\_books("Title")[0] # Assuming you found the desired book

patron = library.patrons[0] # Assuming you selected a patron

transaction = Transaction(book, patron)

library.handle\_transaction(transaction)

1. Check In a book:
   1. When a patron returns a book, use the ‘checkin’ method with the book's ISBN to increase its quantity by 1.
2. Generate Reports:
   * 1. Generate reports to view the current status of books, patrons, and/or transactions.
   1. Book Report: Lists all books in the library

library.book\_report()

* 1. Patron Report: Lists all Patrons registered in the library

library.patron\_report()

* 1. Transaction Report: Lists all transactions

library.trans\_report()

Reflections on the Project:

Discoveries:

1. Organized structure; By dividing the system into different parts like books, patrons etc, the code became much easier to manage.
2. Saving Data: I wasn’t fully able to figure out how to save the data but got the idea behind it. Essentially I should use a method called JSON to save and load data from a file. This should also be possible for csv files, however I’m not fully confident with the scalability of a csv file-based system.
3. Making it Easier for User: We were told to do this but I tried my best to make the code as easy to read as possible. We could turn it into a website to make it more appealing.
4. What I did: I studied through the old notes and looked for help online(YouTube) to finish this project. I implemented basic strategies like libraries, if-then statements, etc to finish this code.

Problems I faced:

1. Dealing with mistakes: Making sure the program can handle the mistakes was very challenging. It took a lot of thinking to make sure the system wouldn’t crash. This can be seen when I use the ‘None’ and ‘is not None’ functions in the classes.
2. Keeping things organized: To keep things organized I started by writing code in different sections. For example, the book class had its own file and Patron had its own and eventually, I added everything to the same file in the end.
3. GitHub: Honestly I never used GitHub which is honestly a pain to use. It took a lot of help from my father to navigate it properly.
4. Loading and Saving: I initially tried to keep the file separate but that seemed very hard at my current skill level and because of that I wasn’t fully able to figure out loading and saving. However, I got the jest of it so I know what to do the next time.

Things I could Upgrade:

1. Easier to use: Turning the system into a website would greatly help the overall view of the project. It would make it more appealing and user-friendly
2. Making it s group project: Turning the project into a group project would greatly speed up progress and make things easier overall.
3. Checking for mistakes: I would do more testing to figure out the best possible solution for every problem it come across.

Overall, the project was good stepping stone towards becoming and expert in coding. There are chances to make it better but I think at my current skill level I will not be able to do it in a very sustainable way.