Library Management System API Documentation.

1. Admin Signup:

a) Insert admin details in tables.

b) An admin records should be unique based on email, which means

duplicates records should not be able to enter in DB.

```
import requests

url = "http://127.0.0.1:8000/admin/signup"

payload={'admin_email': 'Bob@gmail.com',
'admin_password': 'qwerty1234',
'admin_name': 'Bob'}
files=[

]
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files
)

print(response.text)
```

Output:

```
{
  "id": 8,
  "admin_name": "Bob",
  "admin_email": "Bob@gmail.com",
  "admin_password": "qwerty1234",
  "Status": "Already Present"
}
```

Note: If admin doesn’t present then “Status”: admin added successfully.

2. Admin Login:

a) The Admin can log in based on email and password.

```
import requests

url = "http://127.0.0.1:8000/admin/login"

payload={'admin_email': 'Bob@gmail.com',
'admin_password': 'qwerty1234'}
files=[

]
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files
)

print(response.text)
```

OutPut:

```
{
    "id": 8,
    "admin_name": "Bob",
    "admin_email": "Bob@gmail.com",
    "admin_password": "qwerty1234",
    "Status": "Present"
}
```

Note: If password didn’t match then “Status”: Password didn’t match and if email didn’t find then “Status”: Admin doesn’t found.

3. Create an entry for Books.

a) The admin can create a new entry of a book.

```python
import requests

url = "http://127.0.0.1:8000/book/newentry"

payload={'book_id': 'R22',
'book_name': 'Rich Dad Poor Dad',
'book_price': '250',
'book_author': 'Robert T',
'book_dateOfPublication': '2022-09-09',
'book_inlanguage': 'English'}
files=[

]
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files
)

print(response.text)
```

Output:

```
{
   "id": 40,
   "book_id": "R22",
   "book_name": "Rich Dad Poor Dad",
   "book_price": "Rs 250/",
   "book_author": "Robert T",
   "book_dateOfPublication": "2022-09-09",
   "book_inlanguage": "English",
   "status": "Not Added, Book-Id Already Present"
}
```

Note: If book-id not already present then "status": "Added "

4. Retrieve all the books.

a) Retrieve all the books.

```
import requests

url = "http://127.0.0.1:8000/book/retrieveall"

payload={}
files={}
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files
)

print(response.text)
```

Output:

```
[
  {
    "id": 5,
    "book_id": "A21",
    "book_name": "Atomic Habits",
    "book_price": "Rs 300.0/",
    "book_author": "James Clear",
    "book_dateOfPublication": "2019-06-16",
    "book_inlanguage": "English"
  },
  {
    "id": 37,
    "book_id": "M12",
    "book_name": "The Psychology of Money",
    "book_price": "Rs 669.0/",
    "book_author": "Morgan House",
    "book_dateOfPublication": "2020-07-15",
    "book_inlanguage": "English"
  },
]
```

5. Update a book.

a) The book records should be able to update.

```
import requests

url = "http://127.0.0.1:8000/book/update"

payload={'book_id': 'R22dd32',
'book_name': 'Rich Dad Poor Dad',
'book_price': '150',
'book_author': 'Robert Kiyosaki',
'book_dateOfPublication': '3/1/2002',
'book_inlanguage': 'English'}
files=[

]
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files
)

print(response.text)
```

Output:

```
{
   "id": 40,
   "book_id": "R22",
   "book_name": "Rich Dad Poor Dad",
   "book_price": "Rs 150/",
   "book_author": "Robert Kiyosaki",
   "book_dateOfPublication": "3/1/2002",
   "book_inlanguage": "English",
   "status": "Updated"
}
```

Note: If book id did not find then Response: `Book Not found`

6. Delete a book.

a) The book record should be deleted

```
import requests

url = "http://127.0.0.1:8000/book/delete"

payload={'book_id': 'R2233'}
files=[

]
headers = {}

response = requests.request("POST", url, headers=headers, data=payload, files=files
)

print(response.text)
```

Output:

```
{
  "id": 40,
  "book_id": "R22",
  "book_name": "Rich Dad Poor Dad",
  "book_price": "Rs 150/",
  "book_author": "Robert Kiyosaki",
  "book_dateOfPublication": "3/1/2002",
  "book_inlanguage": "English",
  "status": "Deleted"
}
```

Note: If book id did not find then Response: `Book Not found`

Student View: 1.

View all the records of book

```
import requests

url = "http://127.0.0.1:8000/book/singlebookInfo?book_id=R22"

payload={}
files={}
headers = {}

response = requests.request("GET", url, headers=headers, data=payload, files=files)

print(response.text)
```

Output:

```
{
   "id": 43,
   "book_id": "R22",
   "book_name": "Rich Dad Poor Dad",
   "book_price": "Rs 250/",
   "book_author": "Robert T",
   "book_dateOfPublication": "2022-09-09",
   "book_inlanguage": "English"
}
```

Note: If book id did not find then Response: `Book Not found`

Note: Import json file created by postman to run directly the endpoints.