API Endpoints For Books-API

- Test request
    - Request type : GET
    - Description : a test request to check if the backend is running
    - Request endpoint : “/info/test”
    - Request format : None
    - Request output : the backend should return a string with “working!” as the content

- Add book request
    - Request type : POST
    - Description : a request to add info about books to the backend
    - Request endpoint : “/info/addBookInfo”
    - Request format (form-data) :

      ```
      {
        "title": "string",
        "author": "string",
        "isbn": "string",
        "publisher": "string",
        "subject": "string"
      }
      ```

      The book and cover can be included with the multipart form data if required for upload as :

      ```
      {
       “book” : “file”,
       “cover” : “file”
      }
      ```

- Request output : a response holding the info of the book should be given back from the backend. A unique book id will also be generated from the backend and be given in the response.
- Errors would include :
  "ISBN is required",
  "An error occurred while uploading the book"
  "An error occurred while uploading the cover"

- Get book request
  - Request type : POST
  - Description : a request to get info about books from the backend
  - Request endpoint : "/info/getBookInfo"
  - Request format :
    ```
    {
      "searchType" : " ",        (string)
      "keyword" : " "            (string)
    }
    ```
  - Search Types : inTitle , inAuthor , inPublisher , inIsbn , inSubject

  - Request output : a response holding an array of the books with the keyword should be given back from the backend

- Edit request
  - Request type : POST
  - Description : a request to edit info about books to the backend
  - Request endpoint : "/info/editBookInfo"
  - Request format :
    ```
    {
      "title" : " ",                (string)(required)(Old name)
      "isbn" : " ",                 (string)(required)(Old Isbn)
    ```

```
"newTitle": " ",                (string) (optional)
"newAuthor": [ ],               (array of strings)  (optional)
"newIsbn": " ",                 (string)  (optional)
"newPublisher": " ",            (string) (optional)
"newPublishingDate": " ",       (date) (optional)
"newEdition" : " ",             (string) (optional)
"newSubject": " "               (string) (optional)
}
```

- Request output : a response telling the book has been updated will be given back from the backend

- User register:
  - Request type : POST
  - Description : a request to edit info about books to the backend
  - Request endpoint : “/login/register”
  - Request format :
    ```
    {
      "username": "string",
      "password": "string",
      “role” : “string” (only 2 options for this,  “user” and “admin”) (required)
    }
    ```
  - Request output :
    “User has been registered”

- User login :
  - Request type : POST
  - Description : a request to edit info about books to the backend
  - Request endpoint : “/login/”
  - Request format :
    ```
    {
      "username": "string",
    ```

```
  "password": "string"
}
```

- Request output :
  ```
  {
    "token": "string"
  }
  ```

- Submit cover page:
  - Request type : POST
  - Description : a request to submit cover pages for books
  - Request endpoint : “/info/submitCoverPage”
  - Request format :
    ```
    {
      "title": "string",
      "isbn": “string”,
      “cover” : “file”
    }
    ```
  - Request output :
    ```
    {
      "message": "string",
      "book": {
        "_id": "string",
        "title": "string",
        "author": ["string"],
        "bookID": "string",
        "isbn": "string",
        "publisher": "string",
        "subject": "string",
        "coverPageURL": "string",
        "__v": 0
    ```

```
  }
}
```

- Request cover page:
    - Request type : GET
    - Description : a request to get cover pages for books
    - Request endpoint : “/info/getCoverPage”
    - Request format :

      ```
      {
        "title": "string",
        "isbn": “string”,
      }
      ```

    - Request output :
      Cover page file (if both the book and cover page exist)

- Request Terms and conditions :
    - Request type : GET
    - Description : a request to get TnC for the website
    - Request endpoint : “/info/TnC”
    - Request format :
      { }

    - Request output :
      Terms and conditions

- Request for book upload:
    - Request type: POST
    - Description: A request to upload a book
    - Request endpoint: "/info/uploadBook"
    - Request format:

```
{
  "isbn": "string", (required)
  "overwrite": "string",
 (optional , set to “true” for storage overwrite)
  "book": "file" (mulit-part-form-data) (optional)
}
```

  - Request output:
    “Book uploaded successfully”

  - Possible responses:
    200: Book uploaded successfully
    400: Book already exists in the storage (and overwrite is false or not provided)
    404: Book not found in the database
    500: An error occurred while uploading the book

- Request for downloading book
  - Request type: GET
  - Description: A request to download a book
  - Request endpoint: "/info/downloadBook"
  - Request format:
    ```
    {
      "isbn": "string"
    }
    ```

  - Request output: The book file if it exists

  - Possible responses:
    - 200: Book download initiated successfully
    - 404: Book does not exist in the storage
    - 500: An error occurred while downloading the book

- Top reads request
  - Request type: GET
  - Description: A request to retrieve the top reads
  - Request endpoint: "/info/topReads"
  - Request format: None

  - Request output:
    Array of 10 books sorted with the most number of downloads

  - Possible responses:
    - 200: Top reads retrieved successfully
    - 500: An error occurred while retrieving the top reads

- least reads request
  - Request type: GET
  - Description: A request to retrieve the least reads
  - Request endpoint: "/info/leastReads"
  - Request format: None

  - Request output:
    Array of 10 books sorted with the least number of downloads

  - Possible responses:
    - 200: Top reads retrieved successfully
    - 500: An error occurred while retrieving the top reads

- All books request
  - Request type: GET
  - Description: A request to retrieve a list of all books
  - Request endpoint: "/info/allBooks"
  - Request format: None

    - Request output:
      Array of all the books present in the database

    - Possible responses:
      - 200: Top reads retrieved successfully
      - 500: An error occurred while retrieving the top reads

- Delete book request
    - Request type: DELETE
    - Description: A request to delete a certain book
    - Request endpoint: "/info/deleteBook"
    - Request format: None

    - Request output:
      Array of all the books present in the database

    - Possible responses:
      - 200: Top reads retrieved successfully
      - 500: An error occurred while retrieving the top reads