

# MAD1 Project Report

## Author:

Name: Atkal Shubham Sheshnarayan  
Roll number: 23f1002838  
Student email: [23f1002838@ds.study.iitm.ac.in](mailto:23f1002838@ds.study.iitm.ac.in)  
About me: Passionate about Computer Science and fascinated by its uses in various fields, currently pursuing Online BS degree from IITM.

## Description:

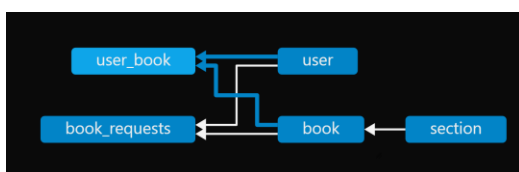
In this project one should create a website which is capable of Hosting E-library and can moderate various aspects of this E-library at ease, and also students can register themselves on this website and can read various books provided by this Library and can also download the books by paying small amount. Basically, one has to create Library Management System.

## Technologies used:

- Python: The programming language used for developing the project.
- Flask: A micro web framework for Python used for building this web applications.
- render\_template: Flask function for rendering HTML templates.
- request: Flask module for handling HTTP requests.
- flash: Flask module for flashing messages to the user.
- redirect: Flask module for redirecting requests to another endpoint.
- url\_for: Flask function for generating URLs for endpoint functions.
- session: Flask module for managing user sessions.
- generate\_password\_hash: Function from Werkzeug library for securely hashing passwords.
- check\_password\_hash: Function from Werkzeug library for checking password hashes.
- os: Python module for interacting with the operating system
- SQLAlchemy: Python SQL toolkit and Object-Relational Mapper (ORM) used for database interactions.
- date: Python module for working with dates.
- HTML: Markup language used for structuring the content of web pages.
- CSS: Styling language used for defining the presentation of HTML elements.
- Bootstrap: Front-end framework used for building responsive and mobile-first website.

## DB Schema Design:

Name	Type	Schema
book	CREATE TABLE book ( id INTEGER NOT NULL,	
id	INTEGER	"id" INTEGER NOT NULL
section_id	INTEGER	"section_id" INTEGER NOT NULL
book_title	VARCHAR(80)	"book_title" VARCHAR(80) NOT NULL
author	VARCHAR(80)	"author" VARCHAR(80) NOT NULL
description	VARCHAR(500)	"description" VARCHAR(500)
link	VARCHAR(300)	"link" VARCHAR(300) NOT NULL
date_created	DATETIME	"date_created" DATETIME NOT NULL
image	VARCHAR(300)	"image" VARCHAR(300) NOT NULL
book_requests	CREATE TABLE book_requests ( id INTEGER NOT NULL,	
id	INTEGER	"id" INTEGER NOT NULL
book_id	INTEGER	"book_id" INTEGER NOT NULL
user_id	INTEGER	"user_id" INTEGER NOT NULL
date	DATETIME	"date" DATETIME NOT NULL
days_requested	INTEGER	"days_requested" INTEGER NOT NULL
librarian	CREATE TABLE librarian ( id INTEGER NOT NULL,	
id	INTEGER	"id" INTEGER NOT NULL
username	VARCHAR(80)	"username" VARCHAR(80) NOT NULL
password	VARCHAR(300)	"password" VARCHAR(300) NOT NULL
section	CREATE TABLE section ( id INTEGER NOT NULL,	
id	INTEGER	"id" INTEGER NOT NULL
section_title	VARCHAR(80)	"section_title" VARCHAR(80) NOT NULL
date_created	DATETIME	"date_created" DATETIME NOT NULL
image	VARCHAR(300)	"image" VARCHAR(300) NOT NULL
description	VARCHAR(500)	"description" VARCHAR(500) NOT NULL
user	CREATE TABLE user ( id INTEGER NOT NULL,	
id	INTEGER	"id" INTEGER NOT NULL
username	VARCHAR(80)	"username" VARCHAR(80) NOT NULL
password	VARCHAR(300)	"password" VARCHAR(300) NOT NULL
first_name	VARCHAR(80)	"first_name" VARCHAR(80) NOT NULL
last_name	VARCHAR(80)	"last_name" VARCHAR(80) NOT NULL
user_book	CREATE TABLE user_book ( id INTEGER NOT NULL,	
id	INTEGER	"id" INTEGER NOT NULL
user_id	INTEGER	"user_id" INTEGER NOT NULL
book_id	INTEGER	"book_id" INTEGER NOT NULL
status	VARCHAR(80)	"status" VARCHAR(80) NOT NULL
rating	INTEGER	"rating" INTEGER
review	VARCHAR(500)	"review" VARCHAR(500)
paid	BOOLEAN	"paid" BOOLEAN NOT NULL
date_borrowed	DATETIME	"date_borrowed" DATETIME
date_returned	DATETIME	"date_returned" DATETIME
days_requested	INTEGER	"days_requested" INTEGER NOT NULL
times_read	INTEGER	"times_read" INTEGER NOT NULL



## API Design:

This project contains a RESTful API designed for a library management system using Flask, a Python web framework.

- API Design

The API follows RESTful principles with clear endpoints for each functionality:

User Management: Register, login, and logout for users.

Librarian Management: Register, login, and logout for librarians.

Book Management: View available books, request books, rate books, and view book statistics.

Section Management: Manage book sections, create, delete sections

Request Management: Handle book borrowing requests.

Statistics: Provide statistics for users and librarians.

- Endpoints:

- User Endpoints:

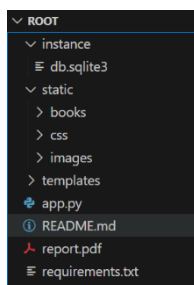
`/user/register, /user/login, /user/logout, /user/requestbook, /user/books, /user/return_book, /user/ratebook, /user/stats, /user/cancelbook, /user/viewbook, /user/downloadbook`

- Librarian Endpoints:

`/library/register, /library/login, /library/logout, /library/home, /library/add_section, /library/view_book_details, /library/deletebook, /library/grantbooks, /library/rejectbooks, /library/delete_section, /library/revoke_access, /library/stats, /library/viewbooks, /library/current_books`

- Other Endpoints: `/, /forgot_password, /about`

## Architecture and Features:



As can be seen in image, project is inside the root folder, in which all the controllers and other code is in app.py itself, also there is requirements.txt file which has all the required libraries names for this project.

Also all html templates are in templates folder and pdf books and css is in static/books and static/css respectively. Instance is prebuilt and is in instance folder

Some additional features:

1. Students can rate the book and avg. rating is calculated, useful to select the book for others
2. Readers Index is calculated (0-1), which encourages students to read more books (index= (total books / read books) / 10 \* avg number of days books read). [Max 1]

## Video:

Link: <https://drive.google.com/file/d/1rGOoKoYhCc0iMvZ3OSaW4JjrjawR6clf/view?usp=sharing>