****

**REPORT**

**TEAM DETAILS:**

Member 1:

Name : Sanjay Chari

USN : PES1201700278

Section : G

Member 2:

Name : Athul Sandosh

USN : PES1201701110

Section : G

Member 3:

Name : Aditya Sankaran

USN : PES1201700710

Section : G

**ABSTRACT:**

**OVERVIEW:**The project will be focused on handling the library system of a university, across the various branches which it may have. To phrase it briefly, it is a database management system for a multi-branch university

**GENERAL FEATURES:**

The system will have a login feature for the borrower (student/faculty). On successful login, a list of books borrowed by the person will be displayed, along with any overdue fees that have to be paid by them.

There will also be an admin/librarian login option which is meant only for librarians. After logging in, the librarian can add/remove books from the books database and can also issue books to borrowers.

The admin/librarian can also view the list of requested books from various borrowers that are unavailable at that moment. In case the book is available in another branch of the university, the system will send a notification to the admin, based on which the admin can make an appropriate decision.

The system will browse through the list of books previously borrowed by the borrower and will give suggestions for books based on their genre for non-educational/fiction books.

Based on the department and semester of the borrower, the system will provide suggestions for educational books to the borrower

**UNIQUE FEATURES:**

* Displaying overdue fees
* If a book is unavailable in one branch, availability in other branches is checked and displayed.
* Fictional books recommendation based on genres of previously borrowed books
* Educational books recommendation based on department and semester of the borrower

**TOOLS USED :**

* PostgreSQL
* HTML
* CSS
* PHP
* JavaScript

**TOOLS THAT COULD PROVIDE BETTER PERFORMANCE :**

* **Flask :**

**Flask** is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries.[[3]](https://en.wikipedia.org/wiki/Flask_(web_framework)#cite_note-3) It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

By using Flask, a more secure authentication system could have been incorporated in our project. This means that, data abstraction of user data would be more efficient using Flask. Flask also has a larger library of extensions that would be beneficial in the development of more interactive webpages. There is also the advantage that Python, the language in which the Flask framework is written, is relatively easier to code in than PHP.

* **Django :**

Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability) and "pluggability" of components, less code, low coupling, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself).[[8]](https://en.wikipedia.org/wiki/Django_(web_framework)#cite_note-8) Python is used throughout, even for settings files and data models.

Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete) interface.

Django would help in our project as it would reduce the number of lines of code that are written and would improve the readabiity of our code. Also, the create, read, update and delete interface would be very relevant to our project.