# SOFTWARE ENGINEERING

# SAMEED SHAH

## Problem Statement

In this task of software engineering, we have been assigned a task of web development. Where the core of our project resides in the development of backend logics using python and its tool Flask. Out of 3 different situations, we decided to go for the development of a Book Store. Where a specific number of features are assigned in order to develop this online store. All five features are mandatory.

## Development

The development section is mainly divided into five different core steps. Each and every step adheres to the requirements of the project. In other words, the features. Before getting started to understand the logics behind the project, let us acknowledge ourself with all the tools and platforms used for this project. Below is the list of all the tools used:

1. Backend Development
   1. Flask (python)
   2. MySQL (Database: localhost)
   3. Pymysql
   4. OpenCv (Image Processing)
2. Simple Frontend Development
   1. HTML
   2. CSS
   3. Bootstrap

Now we know what kinds of tools we used for this project. So, now let’s jump into understanding about the development of the features.

### FEATURE 1

In this section, we are required to create an admin panel. The duty of an admin is to add different sort of books into the system. Having said that, it is also important for the system to valid that the stocks shouldn’t contain more than duplicate entry of the any book and every book should have different ISBN-13 numbers, as the ISBN-13 number should be unique. Therefore, to tackle this situation, we added the ISBN-13 number as a primary key. Due to which, we were able to tackle the bug of duplicity. After logging in through the admin panel by entering his/her username and password, the person can enter books into the stocks by adding relevant information about the book, which would eventually help everyone to look-up for any book with only required information. The input fields for this form includes:

1. ISBN Number
2. Book Title
3. Book Author
4. Book Cover Photo
5. Book Description
6. Book Adding Date
7. Book Trade Price
8. Book Retail Price
9. Book Quantity

With these kinds of relevant information it is possible for any person to look-up for any kind of book.

We created a database in MySQL with the name Booksite. And then created a table names as Booksite.stocks. The stock table holds the information of the book stocks. To add any data, we using the python’s pymysql library to execute any sort of query against the system to run any query. Here we used the main INSERT query to add the data into the stocks.

Graphical user interface, text, application, email

Description automatically generated

### FEATURE 2

This next feature uses the SELECT query to view the data outside of the admin panel. After logging-in into the system as a user – not as an admin. We will view all the items available into the stocks so that we can add them into the cart.

A table is created, in this table we iterate the tuple to get all the data using the SELECT query and added a button that says “add to cart” against all the books displayed. The job of this button is to add the data into the cart for the specific logged-in user. So that the user can view or buy this item by going into the view cart section.

### FEATURE 3

This third feature is about to add the book(s) into the cart. If we understand the logic, we are creating a POST request against all the “add to cart” button, where this post request contains all the required information which is required to add any specific book into the cart. The post request contains the fields such as:

1. ID (primary key, auto-increment)
2. Username
3. ISBN
4. Quantity

The above information is adequate to fill up the cart and to check-out when important.

### FEATURE 4

Just like every other feature, this one is also one of the most important features of any E-Commerce/B2B/B2C websites. The “VIEW CART” feature. In the view cart section, we are passing the username of the person through a tag. When clicking on the button, a SELECT query will run, whose job is to display all the information saved in the CART table against the input username. And also, the display other information from the stocks table such as that cover photo or the retail price of any book

Graphical user interface

Description automatically generated

### FEATURE 5

Last but not the least. The final step is to implement the check-out button. The job of this button is to perform some operations against the stocks of the relevant book. Which is to update the quantity of the stocks when clicking on the stocks. In other words, the check-out button is actually a type of confirmation from the user that he/she is ready to buy those items. Which them upon, the system will take the required action and updates the quantity of those stocks. And to also add the static Payment method in order to finalize the shopping process

Graphical user interface, application

Description automatically generated

## CONCLUSION

It is important for any learning developer to understand the flow of any given project. Because, without drawing the map or flow of your final goal. You can get stuck in the middle or may just end up with spaghetti. Just like in this project, we tried to make the backend as smooth as possible and tried to create APIs for every action performed. Which also helps us to test our system and to figure out the bugs.