

**School of Computer Science and Engineering**

VIT Chennai

Vandalur - Kelambakkam Road, Chennai - 600 127

**J - Final Review Report**

**Programme:** SCOPE

**Course Code:** CSE3002

**Course Name: Internet and Web Programming**

**Slot:** A2

**Faculty:** Dr. Premalatha M

Title: Book Selling Store with Email Verification and Receipts

Name : Advait Deochakke

Reg No : 20BCE1143

**Abstract**

**Keywords**

PHP, CSS, PHPmailer, DBMS, HTML, Random String Generation, Composer

This project shows how we can connect multiple pages and libraries to a backend database, and develop a verification system using user submitted E-mail id and OTP generation, transfer data between php webpages, and more

**Introduction**

1. **Introduction:**

In this project, we have a Book Selling website. The website features a Home Page, a Login Page, Registration Page, User Home page, Buying Page, and more.

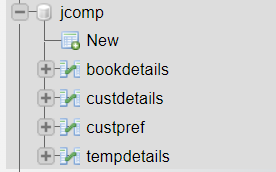
Users can register with email, and get a verification code in the email along with temporary ID. After successfully completing the registration, user data gets added to user database and user is able to log in.

User can generate a preference matrix, by which they get shown specifically the books with the tags that they like, if in stock.

On the buy page, user can enter the ID of the book displayed, and buy it. If balance is inefficient, they are prompted to add balance to the account.

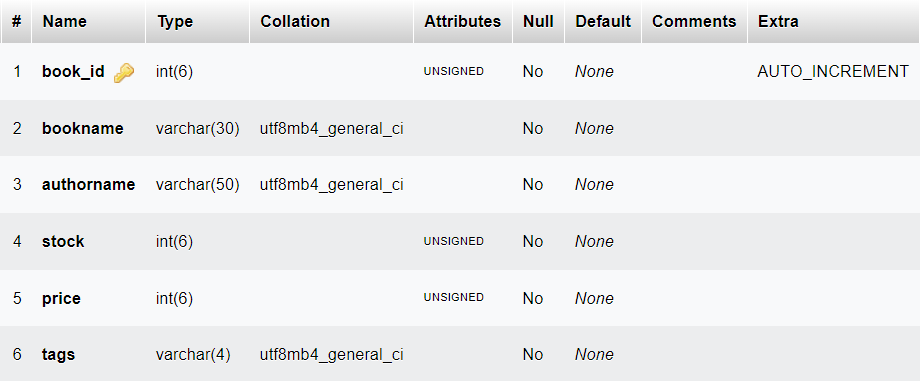
After purchase, an email receipt is sent to them and the book is deducted from stock.

1. **Data Set Description**:



In your database, we have 4 tables

→ Book Details



All row names are self explanatory

The row “tags” holds the tags given to the book in the following form:

s for sci-fi; t for thriller; l for literature; b for biography.

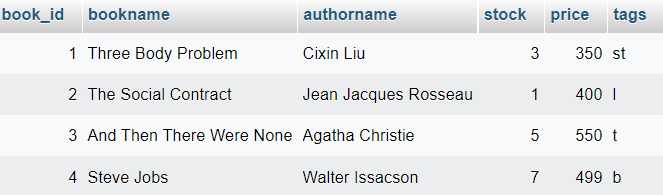
The book is assigned all the tags that it follows.

Eg, if a book is sci-fi, its ‘tags’ row is ‘s’

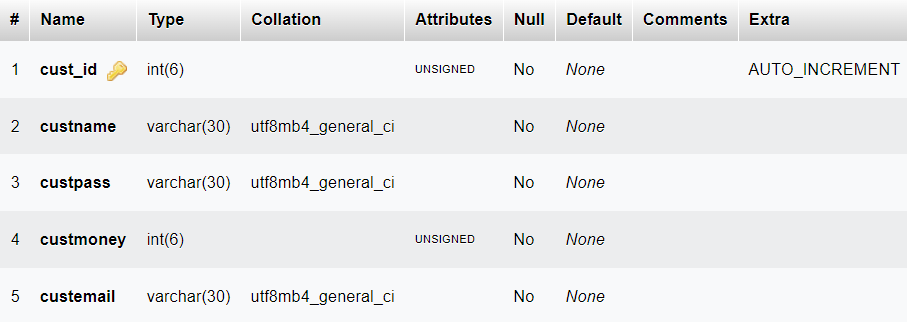
Eg, if a book is both thriller and biography, its ‘tags’ row is ‘tb’

We can search these tags using regex

Eg data for Book Details

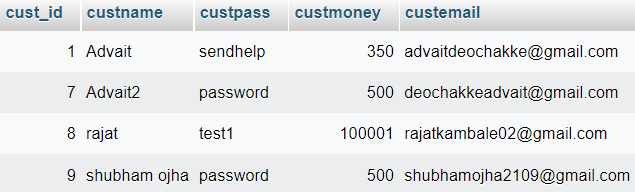


→ Cust Details

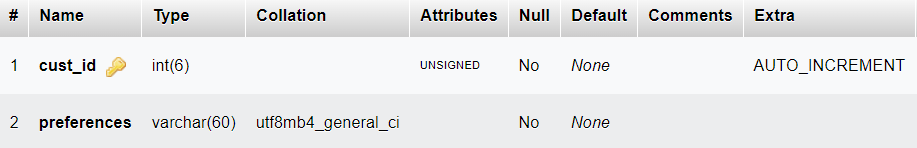


All row names are self explanatory.

Eg, data for Cust Details

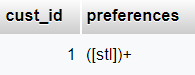


→ Cust Pref (table)



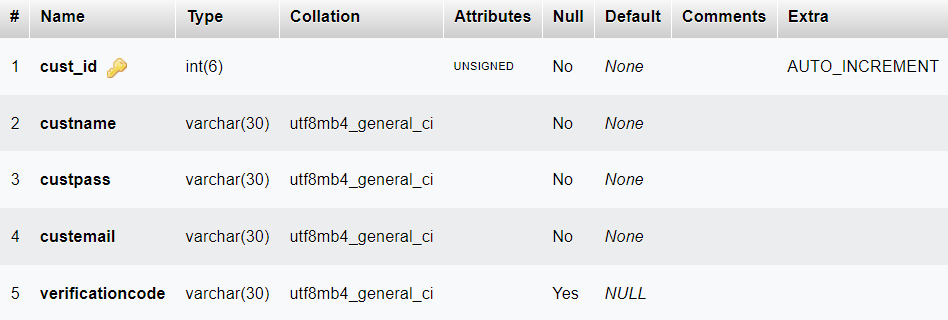
The ‘preferences’ row is filled with the Regex text which contains the customers preferences

Eg, data for Customer Preference



The preferences tag above ‘([stl]+)’ matches any tag which has at least one of the tags of sci-fi, thriller, or literature. It will not trigger any book which only has the tag Biography.

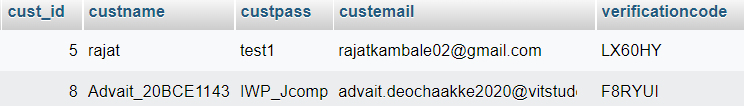
→ TempDetails table (contains the verification data)



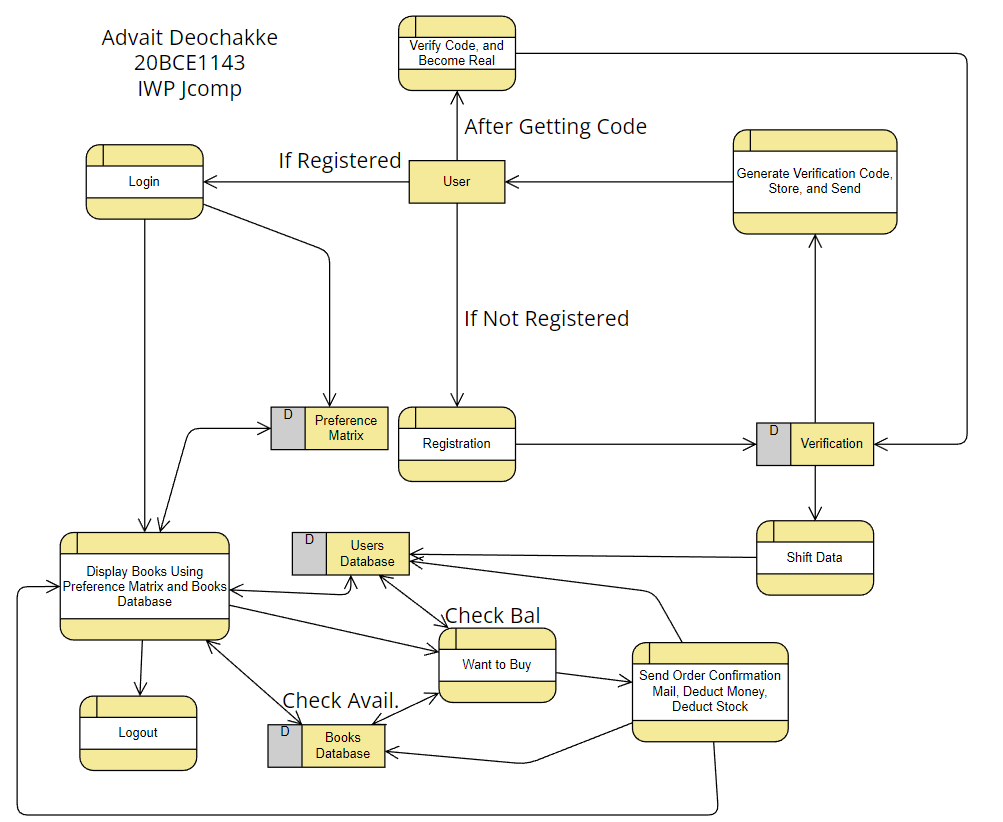
To prevent a customer filling up customer data with random data, we first save it to this table, then after verification is complete, transfer it to the main dataset.

We can routinely delete data from this table without any problems, so too much data should not build up and use up excess space.

Eg data for TempDetails



1. **Data Flow Diagram:**

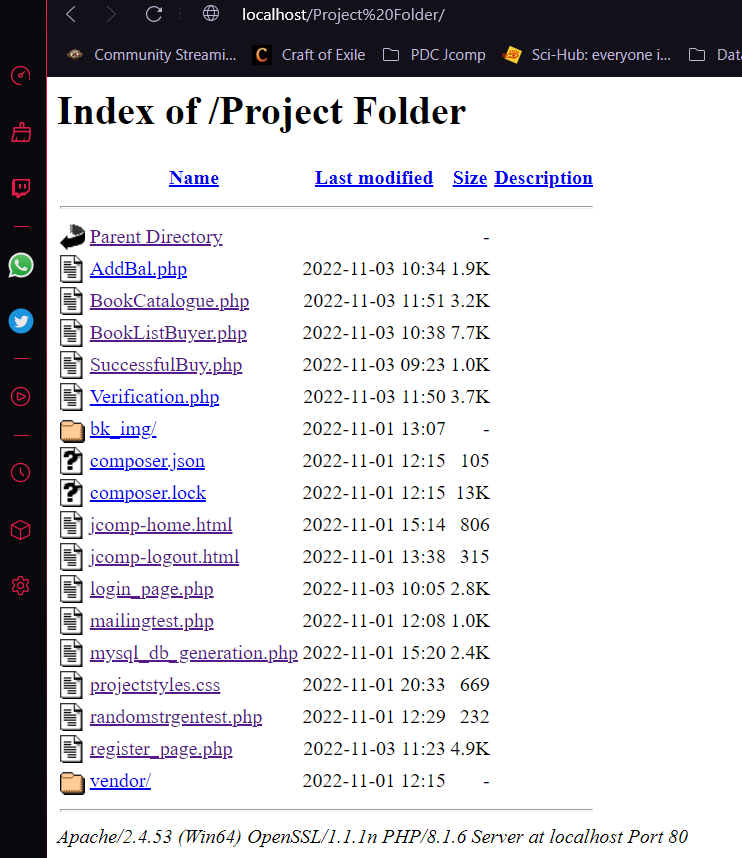


**Implementation**

1. **System Requirements:**

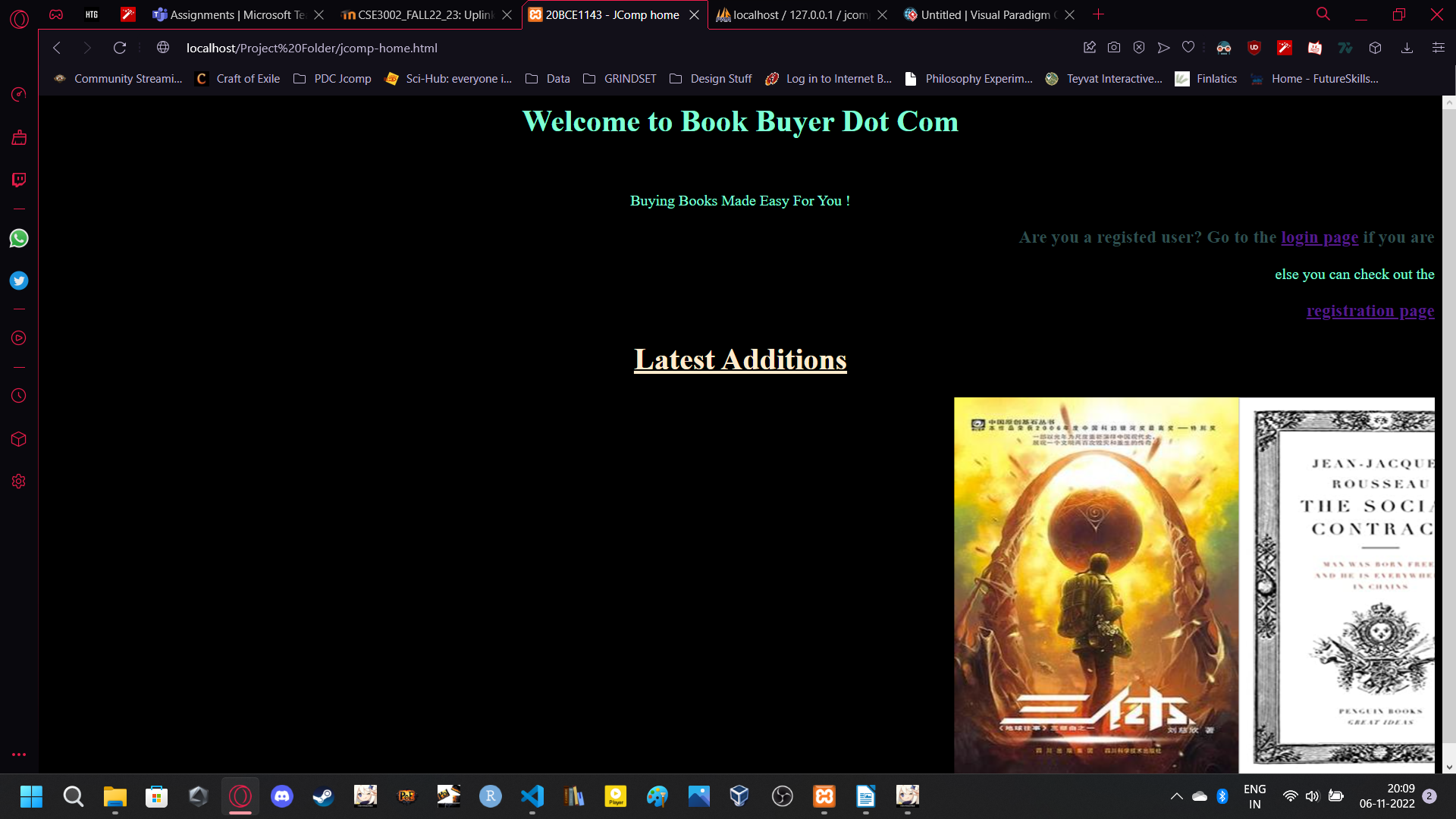
* Any browser capable of running PHP and HTML
* Stable Internet connection
* Display and Keyboard Input, Mouse
* Any Dual Core processor, Intel Pentium, AMD Athlon
* Minimum 1Gb System Ram
* Any OS will do, as long as web browser is available

1. **Implementation (To not overly clutter the document, code is attached separate)**



Project Files and Folder

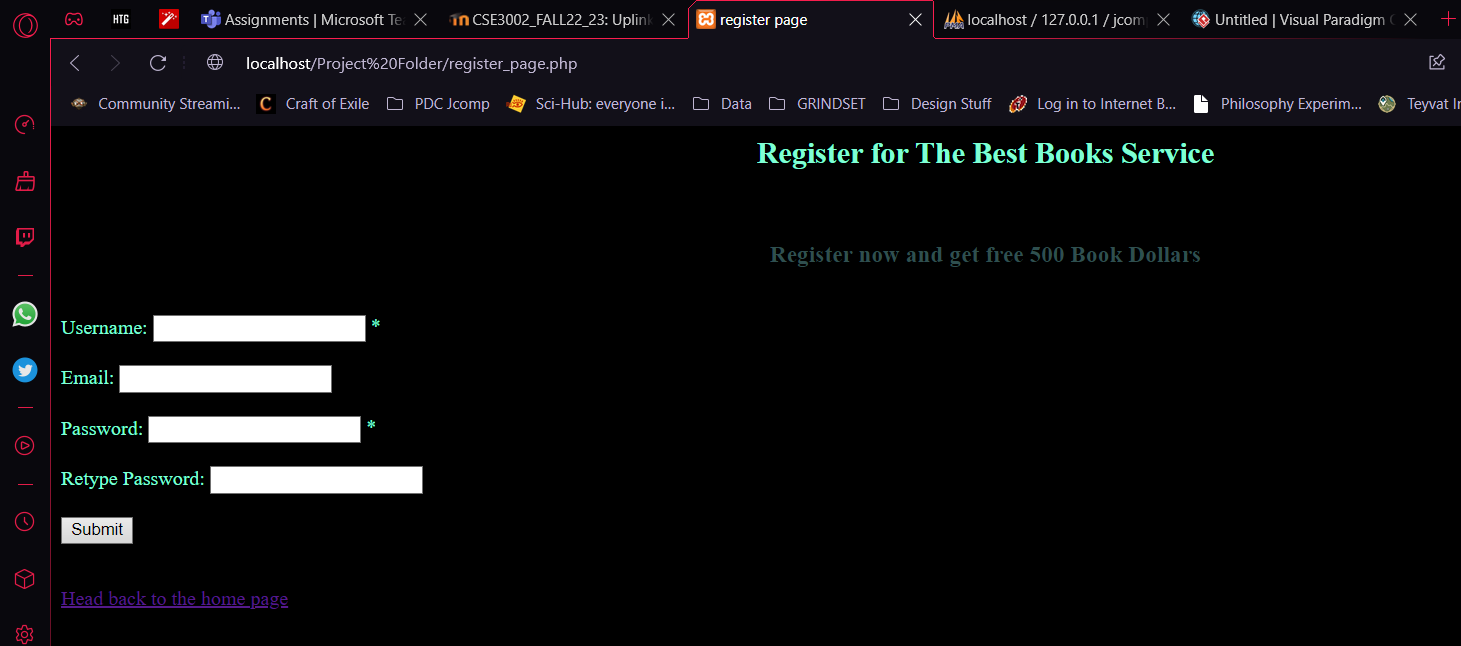
Start at Home Page:



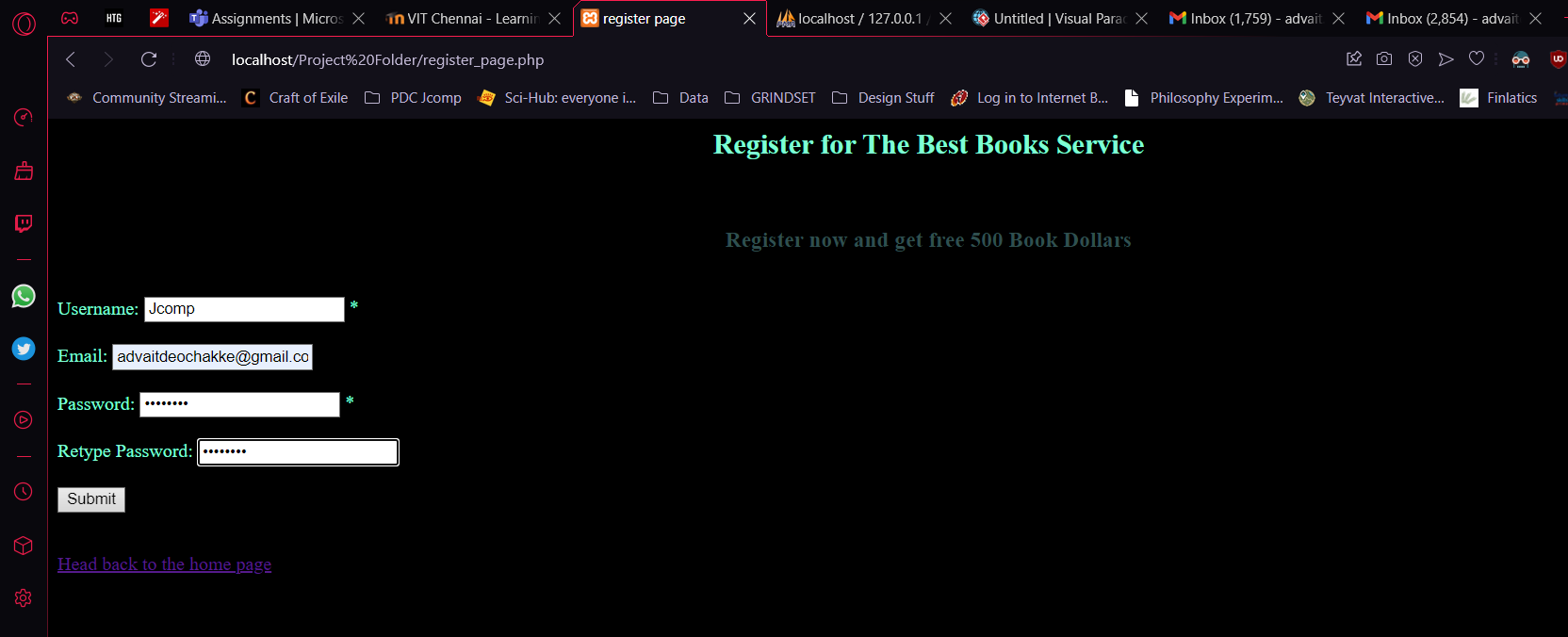
Notice the Login and Registration Page:

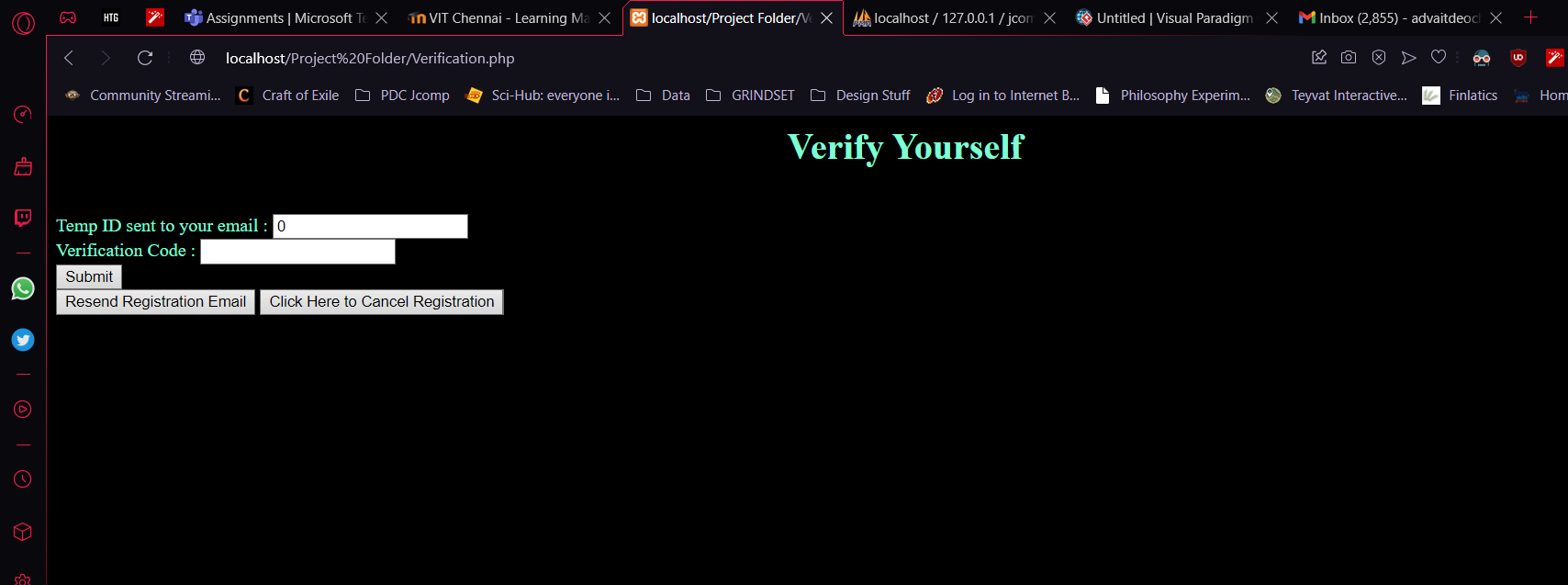
First, lets try for registration

→

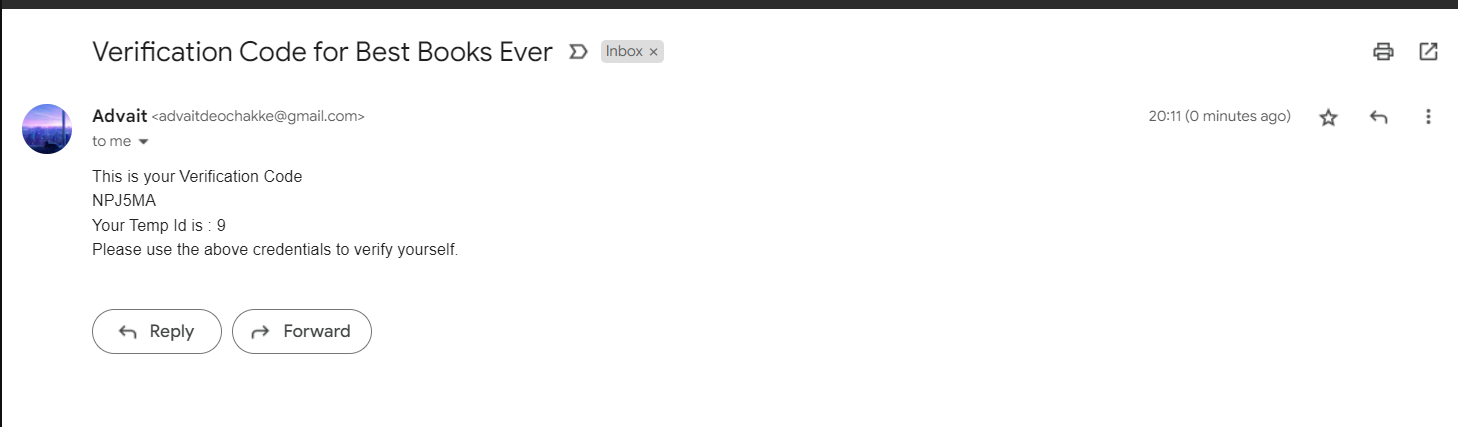


Appropriately fill in details, form is checked for validation with PHP





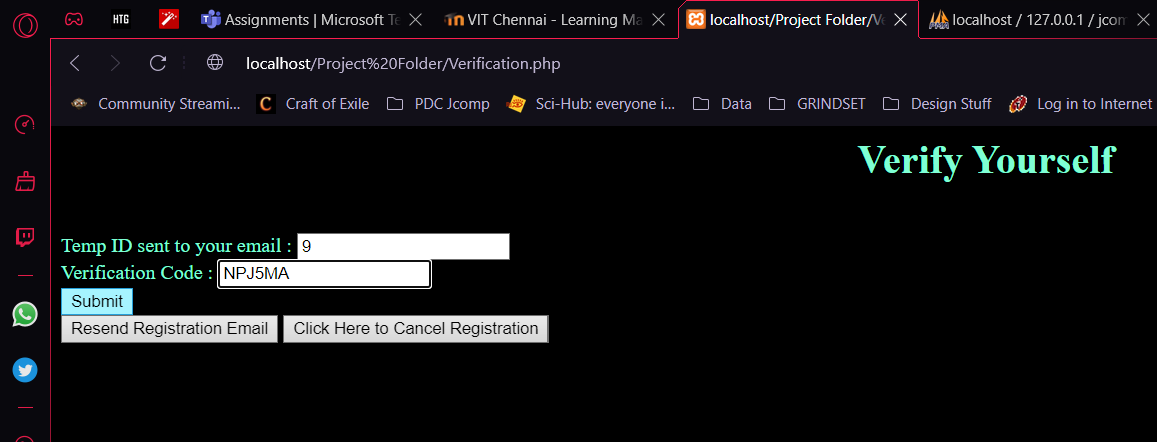
Check Email for code : (Note: May be in Spam folder)



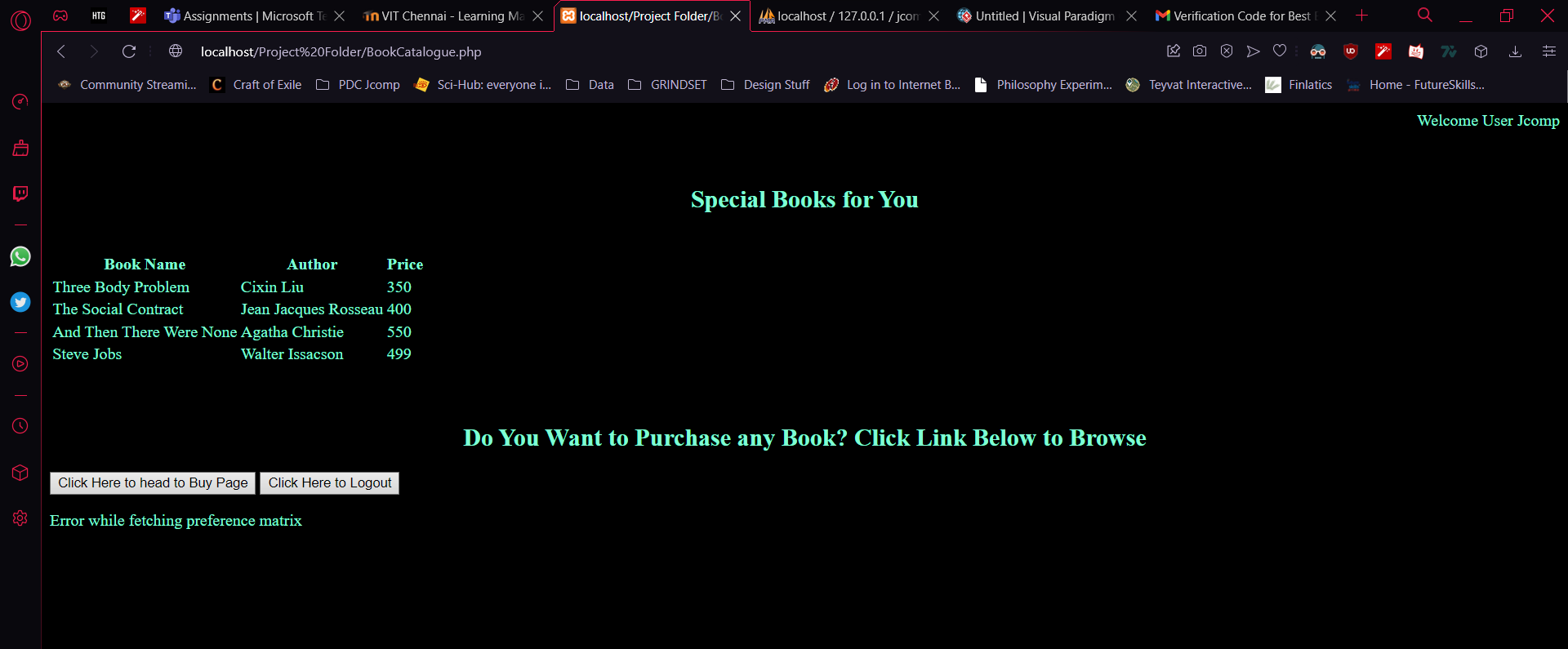
Enter the Above Code and ID

(Sent using gmail SMTP API, automation enabled with specific password)

(Mail generated using PHPmailer)

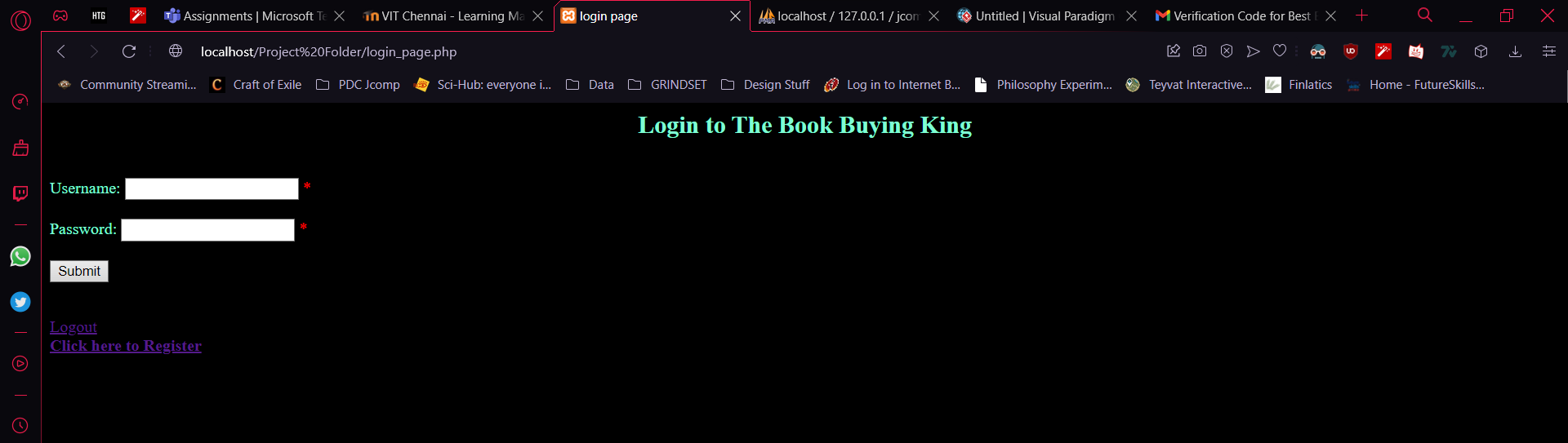


Can resend registration code if required. If cancelled, data will be deleted from verification database

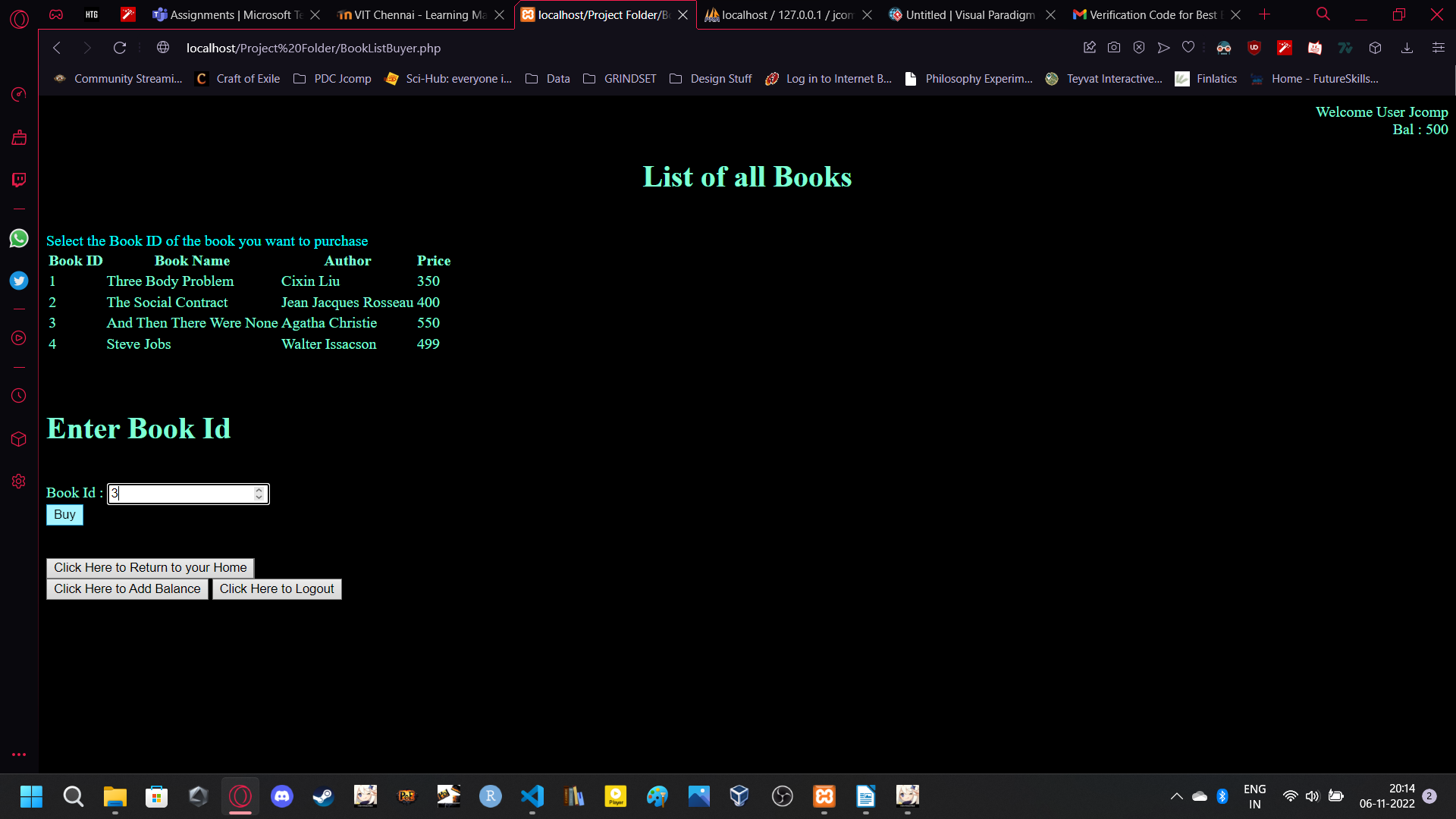


After verification, automatic login

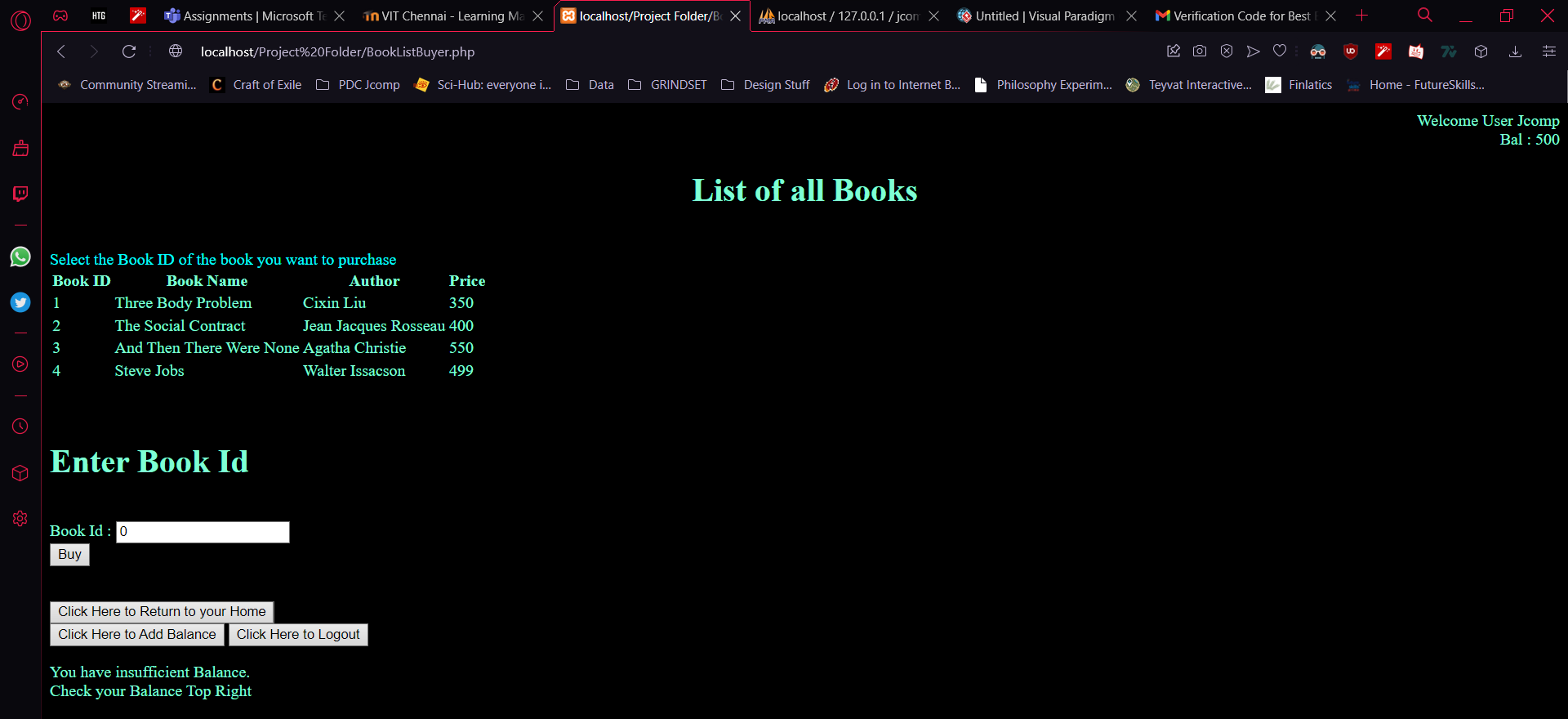
Showing Login Page →



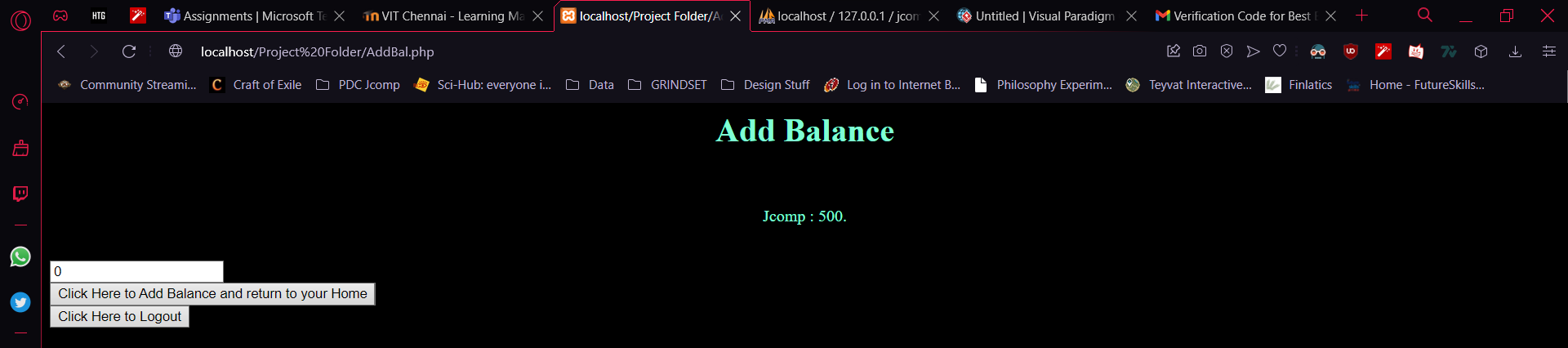
After logging in, head to book buying page



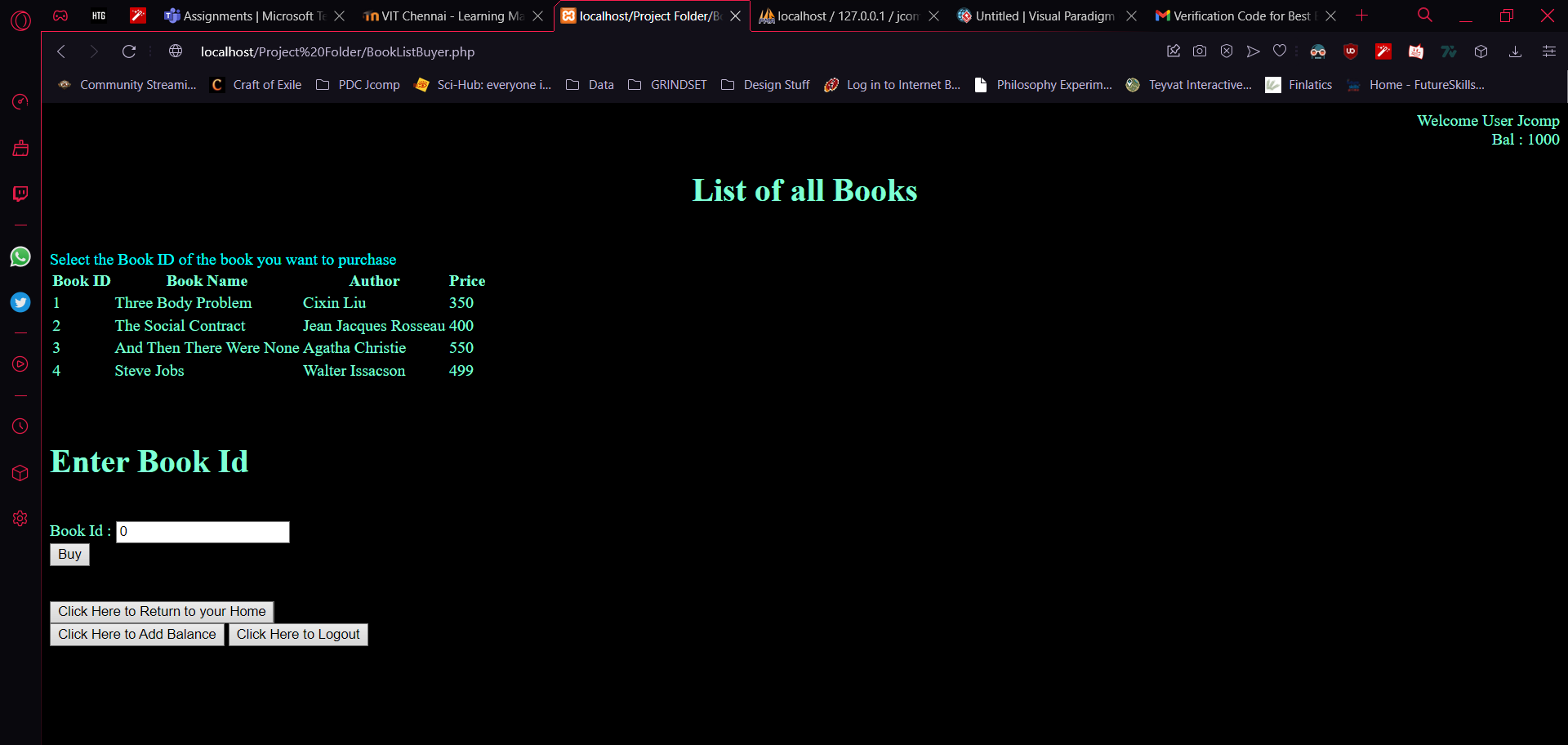
Enter book id and Press buy



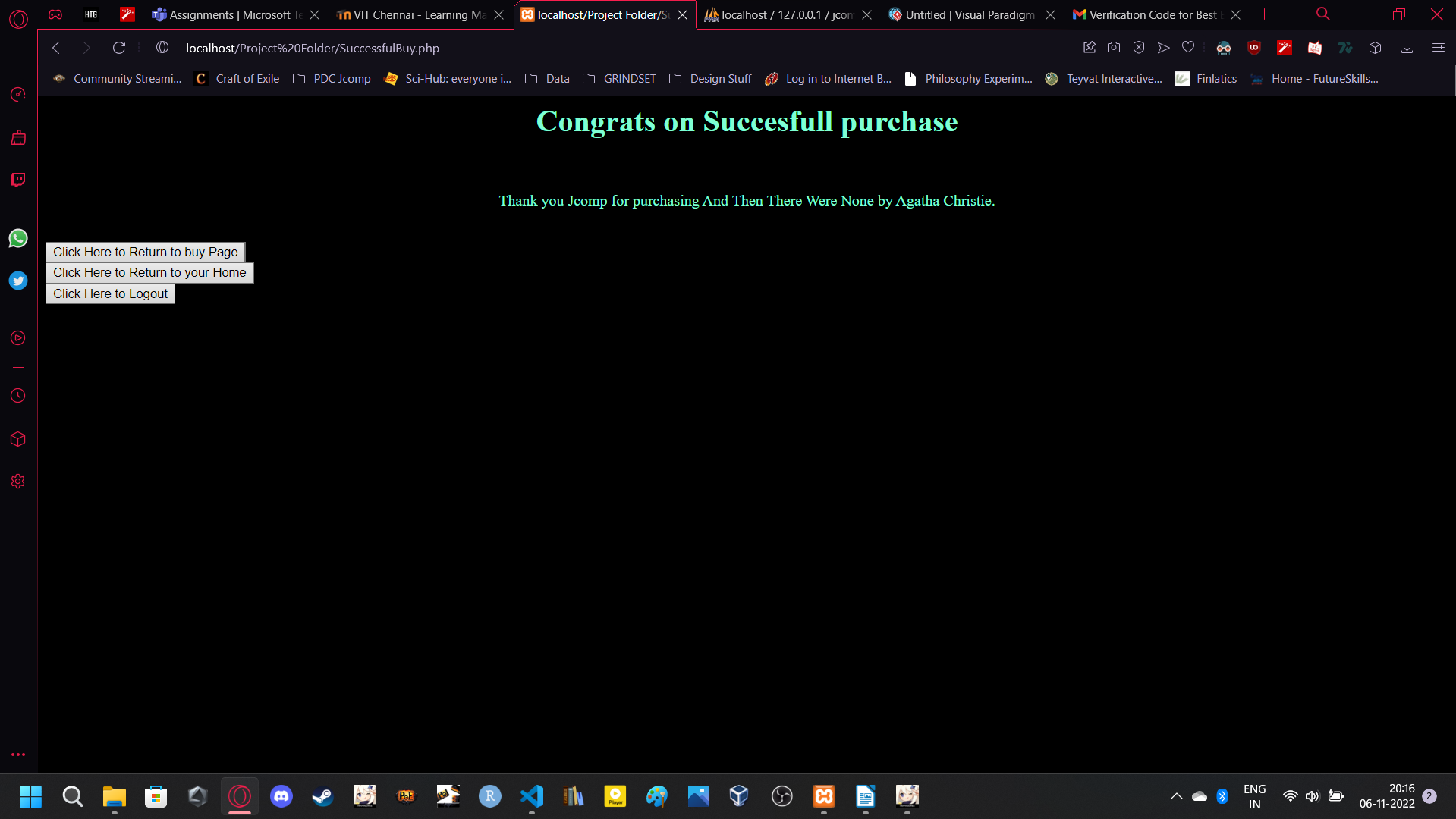
When insufficient Balance, can add. Only when Logged In



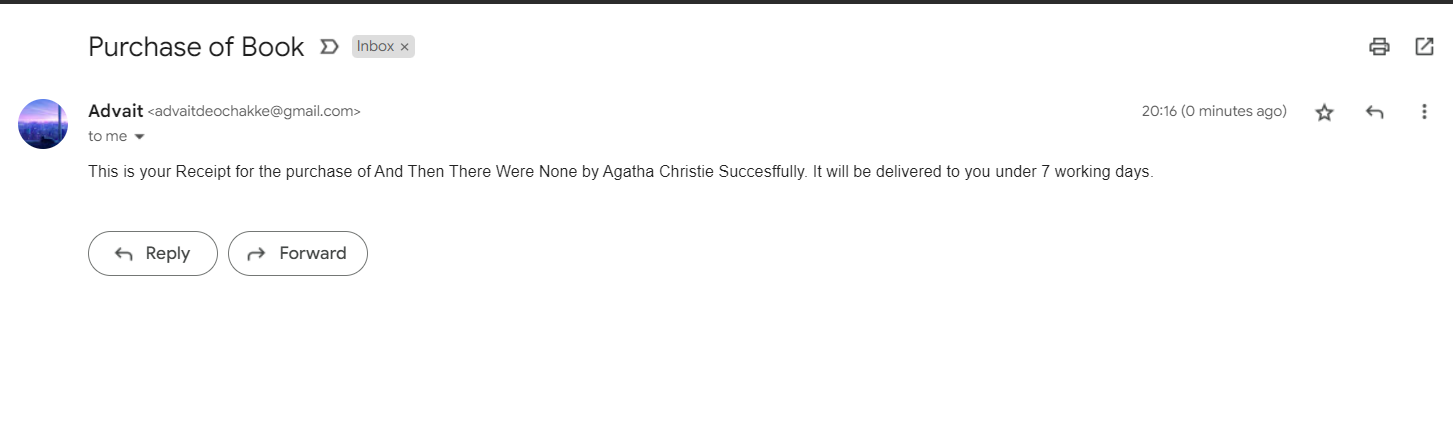
After putting in 500 and pressing Enter : back to Home page → To buying page



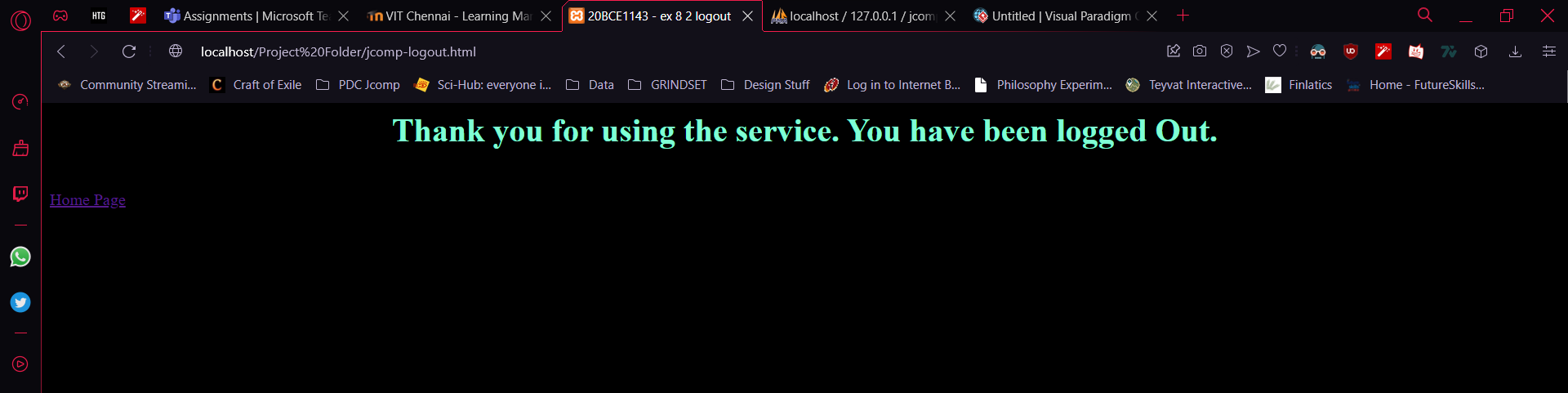
Balance updated, now buy book



Check mail for receipt



Logout →



**Results and Discussion**

1. **Results and Discussion**

We were able to generate, maintain, use, and develop a well functioning website, with the key feature of sending emails to real user accounts with accuracy. We were also able to use SQL Queries such as advanced search with REGEXP, and techniques such as temporarily storing user data until validation to ensure a more efficient workload.

The Project code, along with Data in a .csv format, is available at the following Link :

1. **Conclusion**

The project was clearly developed, and debugged and tested. A functional website was created which combined PHP, HTML, CSS, MySQL, PHPMailer.

**References**

1. **References**

[1] Sudana, I. M., Qudus, N., & Prasetyo, S. E. (2019, October). Implementation of PHPMailer with SMTP protocol in the development of web-based e-learning prototype. In Journal of Physics: Conference Series (Vol. 1321, No. 3, p. 032027). IOP Publishing.

[2] Goodman, D. (2002). Dynamic HTML: The definitive reference: A comprehensive resource for HTML, CSS, DOM & JavaScript. " O'Reilly Media, Inc.".

[3] Myers, J. (1999). RFC2554: SMTP Service Extension for Authentication.