***IIIT, Bhubaneswar***

INTERNATIONAL INSTITUTE

OF

INFORMATION TECHNOLOGY

***BOOKSHOP AUTOMATION***

***System Requirement & Specification***

**Submitted By: Members of Group 6**

**Aman Kumar Ray (B419014) – Documentation & Coding**

**Nirmalya Ray (B419036) – Coding & Unit Testing**

**Samip Tripathy (B419050) – Diagrams & Report**

**Satwik Sahoo (B419051) – Testing & Report**

**Guided By:**

**Jayashree Piri Ma’am**

**Abstract**

**The project report consists of 2 phases. In the first phase, the problem assigned to us has been mentioned. Second phase provides the solution for the same problem along with an insight into the development process involved.**

***INTRODUCTION* gives the information regarding Bookshop Automation System. It also gives a brief introduction about the project under the topic *PROBLEM DEFINITION*. It gives a short introduction about Bookshop and its operations.**

**The next part is the *REQUIREMENT ANALYSIS* which is mainly concerned with study of various requirements such as user, System & Performance.**

**The next part is on *SYSTEM DESIGN*, which includes interface design, detailed design. It also includes Structure Chart, DFD, Use Case Diagram & Database Design.**

**The next part is on implementing the design to *CODE* and *TESTING* the cases for ease of access.**

**Last part is regarding *CONCLUSION* about the above mentioned problem.**

\_\_

\_

*Table Of* *CONTENT*

**Introduction 4**

**Approach to problem solving 5**

**Problem Definition 6**

**Life Cycle Model 7**

**Waterfall Model 8**

**Detail Data Flow Diagram 10**

**Structured Chart 13**

**Use Case Diagram for Entire System 13**

**Unit Testing 14**

**Integration Testing 18**

**Integration Testing Output 19**

**Conclusion 36**

**Reference and Bibliography 39**

***INTRODUCTION:***

**The Bookshop Automation System is to automate all operations in a bookshop. Generally it includes Order Processing, Stock Management and Accounts Management.**

**Before automating a bookshop we have to understand the concept of automation. In automation of any operation we make a system which does work automatically as the respective events occur, for which it is meant.**

**There are some common examples of automation like the autopilot system in the aircraft, automatic home systems (electric system, water system, fire alarm system, doors system etc). These are the best examples of the automation systems.**

**Here we are trying to develop such a system which provides automation on any type of bookshop. This means a shop with a system which provides the facility to the customers of the shop to purchase the books from the shop without any complexity.**

**For example, if any customer wants to purchase any book from the shop then the customer can forage the database by its Title or Author’s Name. And then purchase it by paying the price on the bookshop cash counter and receiving its invoice. And if the required book is unavailable then he/she can put in a requisition.**

***APPROACH TO PROBLEM SOLVING***:

**IN BRIEF LIFE CYCLE:**

| **S.No.** | **Stage** | **Key Question** | **Results** |
| --- | --- | --- | --- |
| **1** | **Need Recognition**   * Preliminary survey/initial investigation | * What is the problem or opportunity? | * Statement of scope and objectives. |
| **2** | **Analysis**  • Detailed evaluation of present system  • Data Collection | * What are the facts? | * Logical Model of the System. |
| **3** | **Design**   * Design Specifications * Programmed Construction * Testing | * Specifically, how must the problem be solved? * How ready are programs for acceptance tests? | * Alternative design * Test plans * Operating procedures. |
| **4** | **Implementation**   * User training * System Conversion | * What is an actual operation? | * Programmed. * User Friendly documentation |
| **5** | **Post Implementation**   * Evaluation * Maintenance * Enhancements | * Is the key system running? * Should the system be   Modified? | * User Requirements met. * User standards met. * Satisfied user. |

***PROBLEM DEFINITION:***

**Almost every activity in the world today is controlled by computer driven software programs.**

**This trend was first accommodated by engineering applications in the past. However, as the lifestyle became more and more complex, every area of human interactions was invaded by various software systems, such as real time, business, simulation, embedded, web based, personal and more recently, artificial intelligence software etc.**

**According to the above facts, managing and maintaining a book shop could also be controlled by efficient software. This project focuses on designing such efficient and reliable software which controls the transactions of a bookshop. In the real world, it tends to be associated with automated systems as they provide many benefits than doing the same thing manually.**

**As mentioned above, here we have introduced a system which can be used to maintain a bookshop. When we are concerning the manual process of a bookshop, the major problem is the waste of time. A customer has to waste his/her valuable time when he needs to buy a book as all the events such as searching, purchasing are done by members of the staff .This makes the manual process very slow.**

**But automation will reduce this time. In a bookshop we should deal with a large store. Then the person has to maintain it with documents which are recorded by him. Therefore, there may be defective reports.**

**As we are familiar with this type of system at instance we will be able to have the results that we want. Communication with suppliers, customers and other related organizations will be more successful as the system is so fast.**

**When the bookshop issues an item to a customer, all the stages of the transaction procedure will be facilitated by the system & it will be more accurate. Also it will drastically reduce the effort of the Accounts and Sales people as they don’t have to maintain any handwritten records.**

***LIFE CYCLE MODEL***

**We have followed the Waterfall Development Model for the Development of Bookshop Automation System Software.**

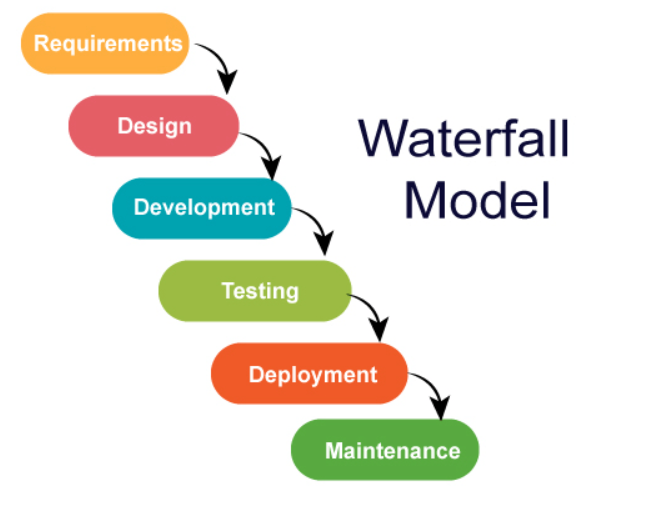
**Let us see what it is and how it is implemented.**

**The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.**

**The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. In this waterfall model, the phases do not overlap.**

**Waterfall approach was the first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.**

**The following illustration is a representation of the different phases of the Waterfall Model.**

****

**Requirements**

**Consider if there is a client and he wants to develop software, then the client reaches out to the company. Suppose he reaches out to the service-based company and asks the company to build the software. Then Company will collect all the requirements, the knowledge that the customer has or the client wants to have on his software, the company will collect all the information from the client and prepare the documentation. Once this activity is performed, then the design phase gets started.**

**Design**

**In this phase, we prepare the high-level and low-level designs. Before developing the software, the design of the software is required. Suppose the customer wants a specific design, then UI (User Interface) of the website will be made by the designer and dataflow are also designed in this phase that how data will flow. After this, the structure charts and Use case Diagrams are also made in this phase. Once this phase is completed, the development phase will get started.**

**Development**

**In the Development phase, the software development team starts coding and developing the software. This is the longest phase of the waterfall model as developers need more time to build the software. Once the development of the software is completed, then the project is handed over to the testers.**

**Testing**

**The testing team will test the software, and if any bug is found, then they inform the developers about the issue they found and make sure that the bug is fixed. They ensure that the end-to-end software is completed.**

**Deployment**

**Once the project is tested, the project is deployed so that it becomes live to the real-time users.**

**Maintenance**

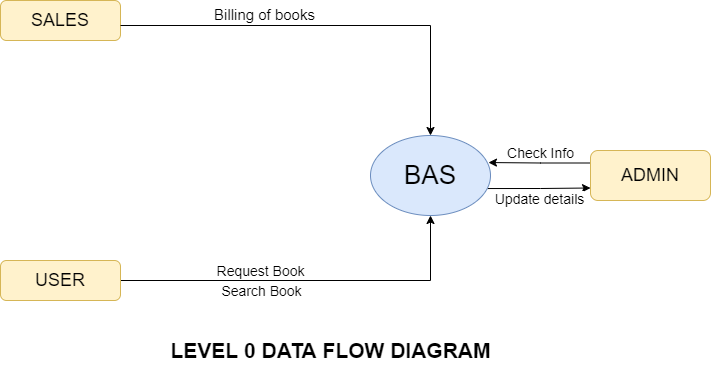
**Finally, the project is deployed and available to the clients. Clients want the maintenance period for one or two years because if any bug is found or want a slightly enhanced feature in the project, they need some team to handle such stuff. Due to this reason, the software has to go through the maintenance period.**

***DATA FLOW DIAGRAM (DFD):***

**A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. They can be used to analyze an existing system or model a new one.**

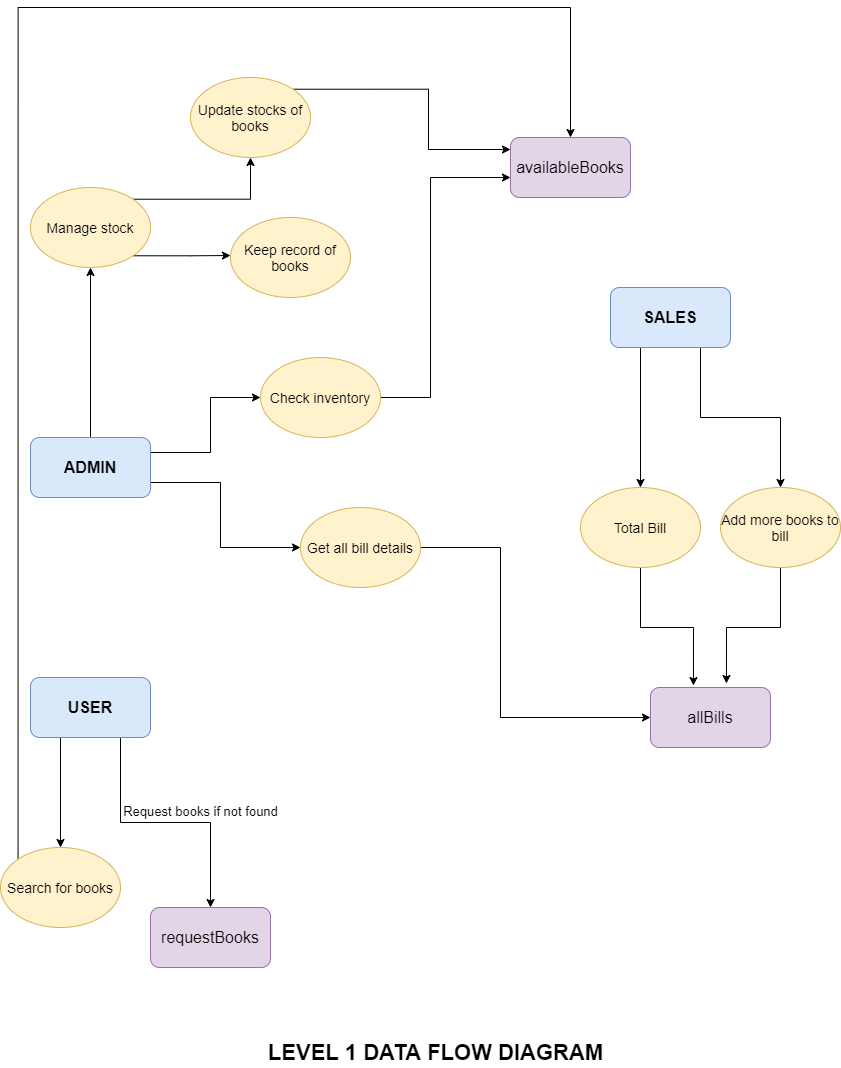
***LEVEL 0 DFD***

**DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows. Below there is level 0 DFD of BAS which gives basic overview of the system.**



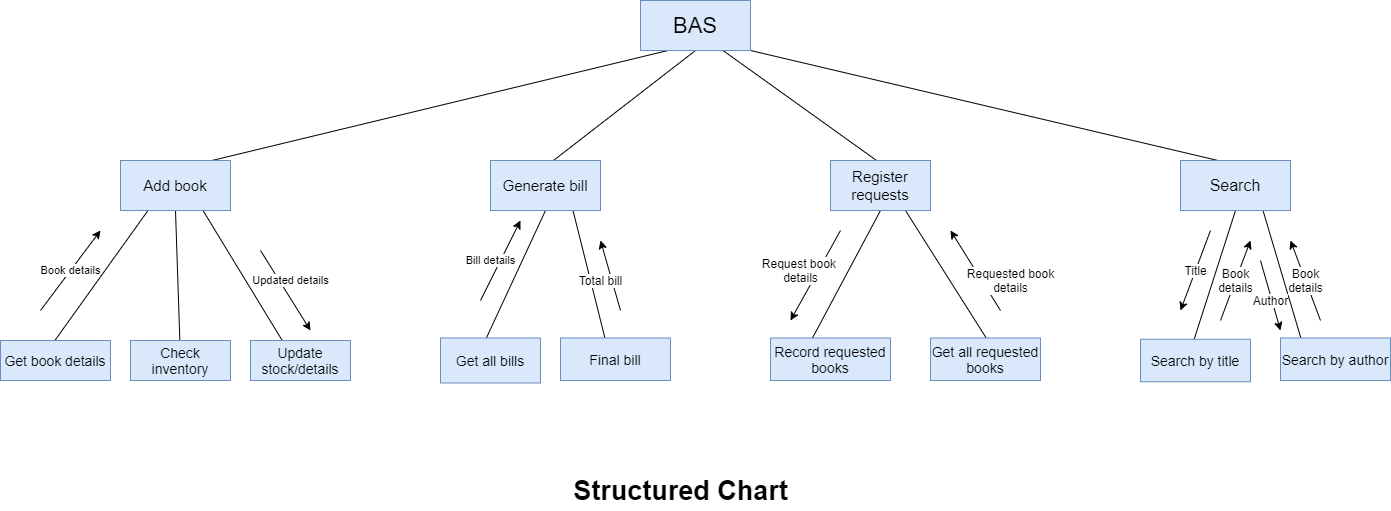
***LEVEL 1 DFD***

**In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and break down the high-level process of 0-level DFD into sub processes. A level 1 DFD notates each of the main sub-processes that together form the complete system.**

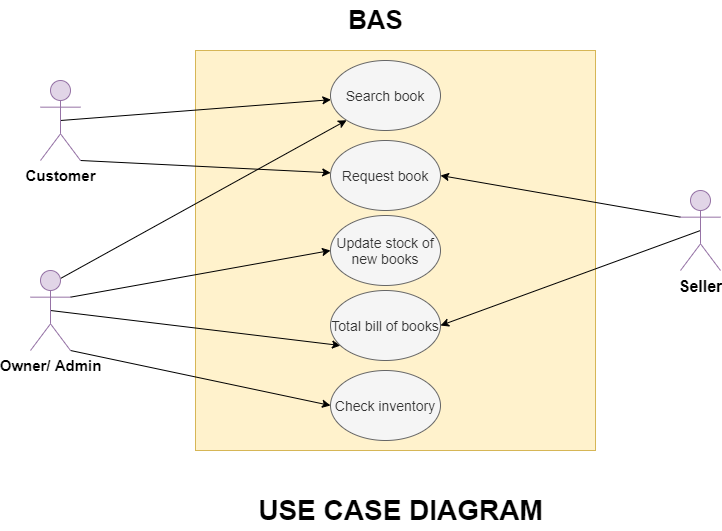


***STRUCTURED CHART:***

**Structure Chart represents hierarchical structure of modules. It breaks down the entire system into lowest functional modules, describing functions and sub-functions of each module of a system to a greater detail. The lines represent the connection and or ownership between activities and sub activities as they are used in organization charts.**

***USE CASE DIAGRAM:***

**A use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior, and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation. In short, a use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses. The actors are often shown as stick figures. It only summarizes some of the relationships between use cases, actors, and systems.**

****

***WHITE BOX TESTING:***

**Unit testing involves the testing of each unit of an individual component of the software application. It is the first level of testing. The aim behind unit testing is to validate unit components with its performance.**

**A unit is a single testable part of a software system and tested during the development phase of the application software.**

**The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.**

**Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as Unit testing or Components Testing.**

**In a testing level hierarchy, unit testing is the first level of testing done before integration and other remaining levels of the testing.**

* **Unit testing helps testers and developers to understand the base of code that makes them able to change defect causing code quickly.**
* **Unit testing helps in the documentation.**
* **Unit testing fixes defects very early in the development phase that's why there is a possibility to occur a smaller number of defects in upcoming testing levels.**
* **It helps with code reusability by migrating code and test cases.**

**1.1 addBookFun()**

INPUT: Details Of The Book: ISBN-12213 , Book Title-”Agatha”, Author Name=”ABC”, Stock=15, Price=399, Rack=5

CONDITION: Book Should Not Be Present In the Available Books Database

EXPECTED OUTPUT: Inserts Correctly Into Database.

RESULT: PASS

**1.2 getAllBills()**

INPUT: No Input

EXPECTED OUTPUT: All Bills Present in All Bills Database

OUTPUT: All Bills Received

RESULT: PASS

**1.3 getAllBooks()**

INPUT: No Input

EXPECTED OUTPUT: All Books Details Present in Available Database

OUTPUT: All Books Details Received

RESULT: PASS

**1.4 getAllRequestedBooks()**

INPUT: No Input

EXPECTED OUTPUT: All Books Present in Requested Books Database

OUTPUT: All Books Received

RESULT: PASS

**1.5 checkInventory()**

INPUT: No Input

CONDITION: The Stock Of The Book Must Be Less Than Threshold

EXPECTED OUTPUT: All Books Present in Available Books Database Whose Stock is Less Than Threshold

RESULT: PASS

**1.6 requestBook()**

INPUT: A new request for a book: ISBN-12213 , Book Title -”Agatha”, Author name=”ABC”, Publisher Details = ”Marvel”

EXPECTED OUTPUT: Inserts correctly into database.

OUTPUT: Inserts correctly

RESULT: PASS

**1.7 addBookToBill()**

INPUT: A request for adding a book to bill: Bill Number-10, ISBN-12213 , No. Of Copies-4, Price-399

EXPECTED OUTPUT: Inserts correctly into database.

OUTPUT: Inserts correctly

RESULT: PASS

**1.8 searchBookByTitle()**

INPUT: Book Title

CONDITION: Book Must Be Present in The Available Books Database

EXPECTED OUTPUT: Details Of Book Having The Title as Given Book Title

RESULT: PASS

**1.9 searchBookByAuthor()**

INPUT: Author Name

CONDITION: Book Must Be Present in The Available Books Database

EXPECTED OUTPUT: Details Of Book Having The Author as Given Author Name

RESULT: PASS

**1.10 printBill()**

INPUT: Bill Number

CONDITION: Bill Number Must Be Present in The All Bills Database

EXPECTED OUTPUT: Details Of Books with Number Of Copies And Price With The Provided Bill Number To Be Received

RESULT: PASS

**1.11 updateBook()**

INPUT: Updated Details Of The Book: ISBN-12213 , Book Title-”Agatha”, Author Name=”ABC”, Stock=25, Price=399, Rack=5

CONDITION: Book Should Already Be Present In the Available Books Database

EXPECTED OUTPUT: Edits Book Details Correctly In The Database.

RESULT: PASS

# ***BLACK BOX TESTING:***

**Integration testing is the second level of the software testing process after unit testing. In this testing, units or individual components of the software are tested in a group. The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.**

**Unit testing uses modules for testing purposes, and these modules are combined and tested in integration testing. The Software is developed with a number of software modules that are coded by different coders or programmers. The goal of integration testing is to check the correctness of communication among all the modules.**

**Once all the components or modules are working independently, then we need to check the data flow between the dependent modules is known as integration testing.**

**Let us see one sample example of a banking application, as we can see in the below image of amount transfer.**

## ***Why Do We Need Integration Testing***

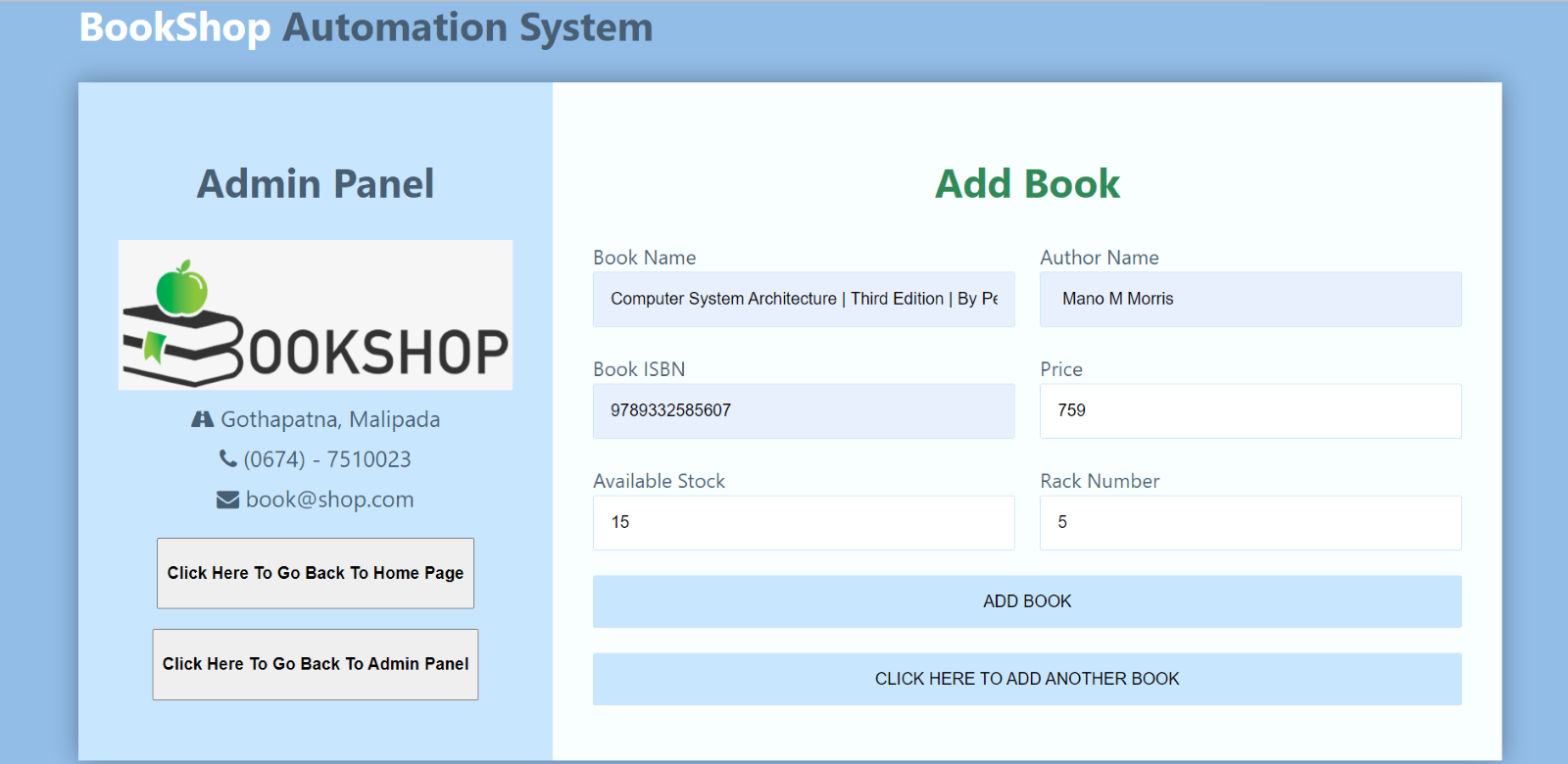
**Although all modules of software application already tested in unit testing, errors still exist due to the following reasons:**

1. **Each module is designed by individual software developers whose programming logic may differ from developers of other modules, so integration testing becomes essential to determine the working of software modules.**
2. **To check the interaction of software modules with the database whether it is erroneous or not.**
3. **Requirements can be changed or enhanced at the time of module development. These new requirements may not be tested at the level of unit testing hence integration testing becomes mandatory.**
4. **Incompatibility between modules of software could create errors.**
5. **If exception handling is inadequate between modules, it can create bugs.**

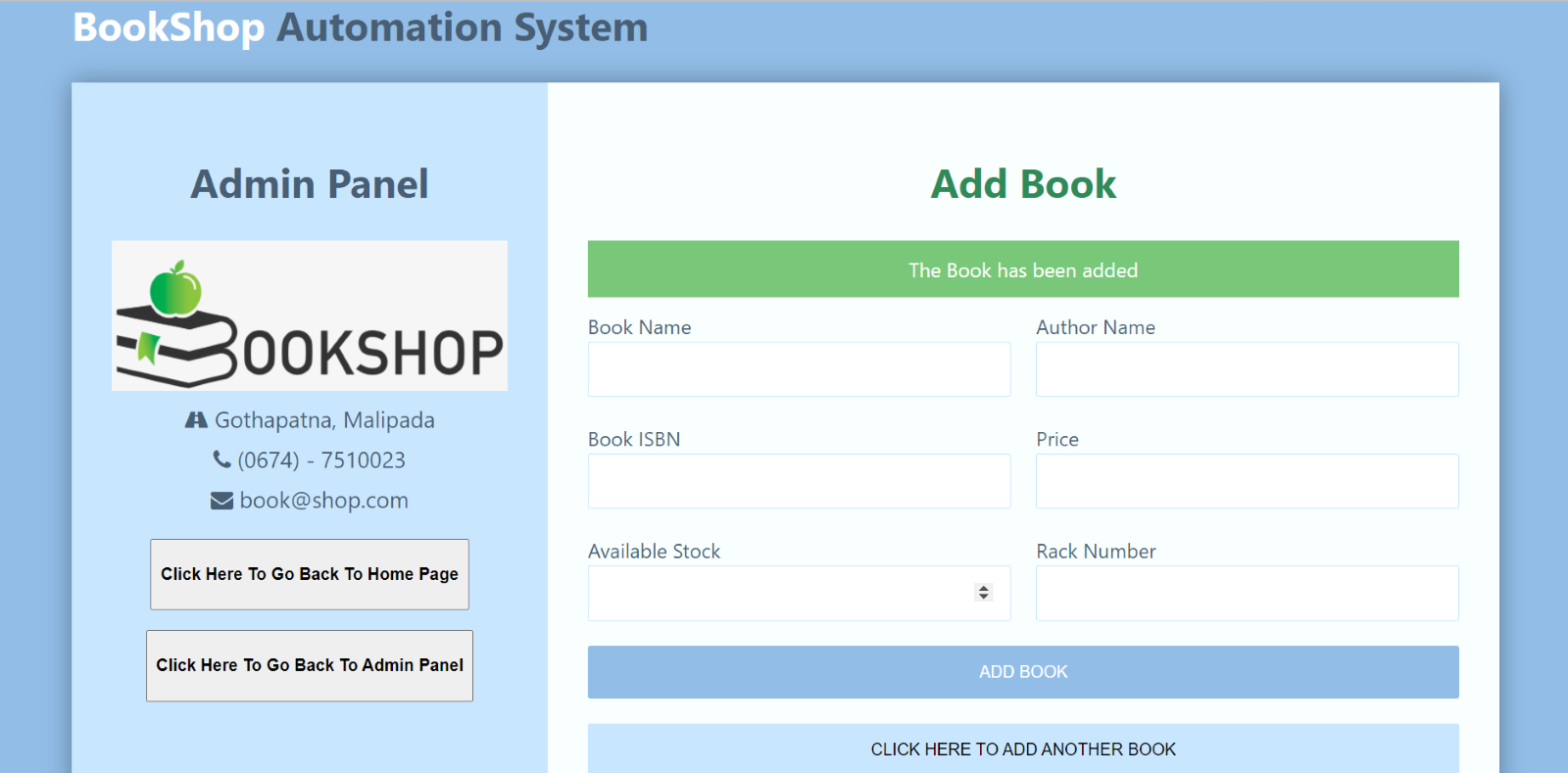
INTEGRATION TESTING: -

1. Add book-

1.1 **Case when all form entries are entered successfully -**

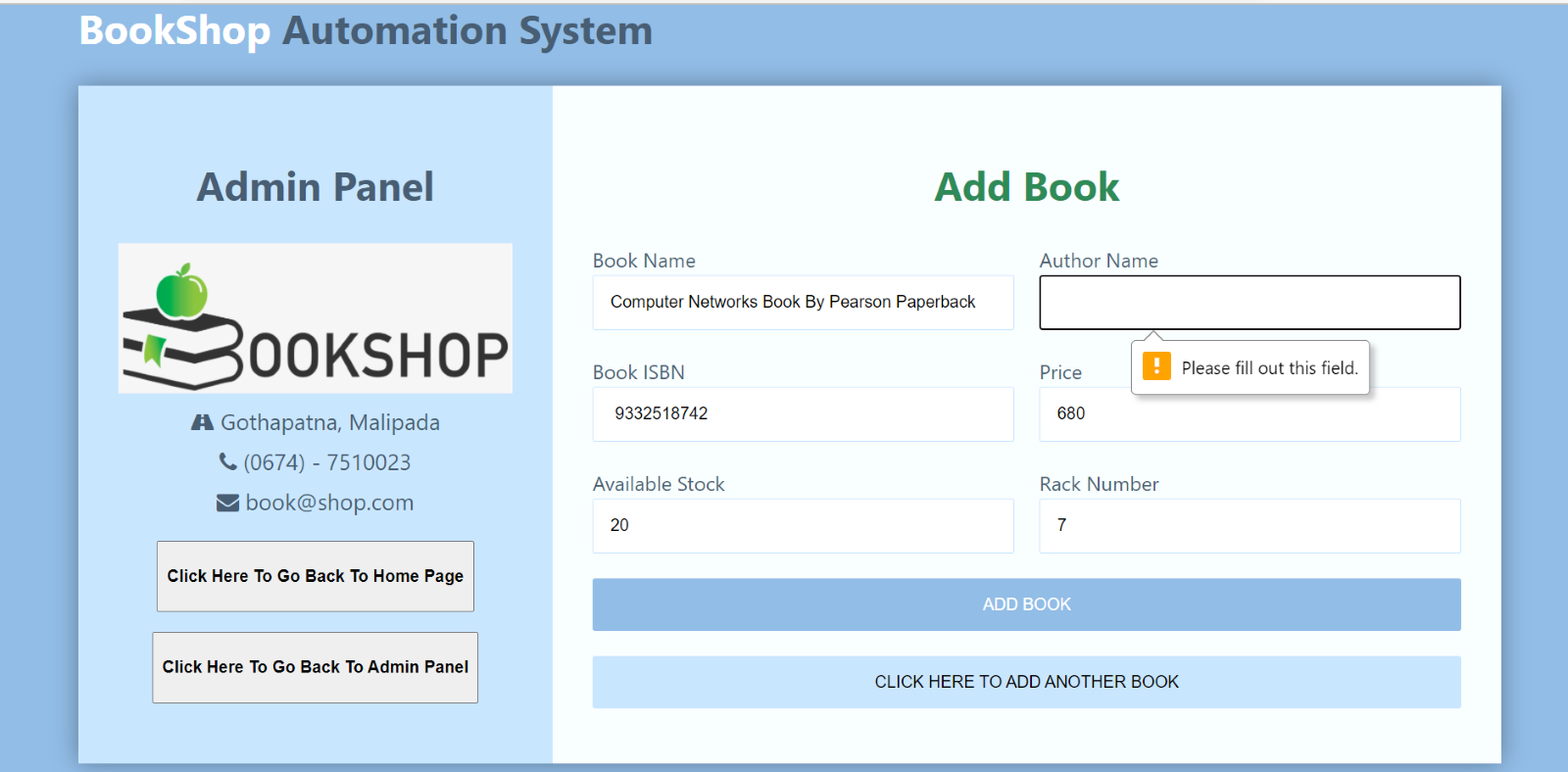


Expected output: - book added successfully

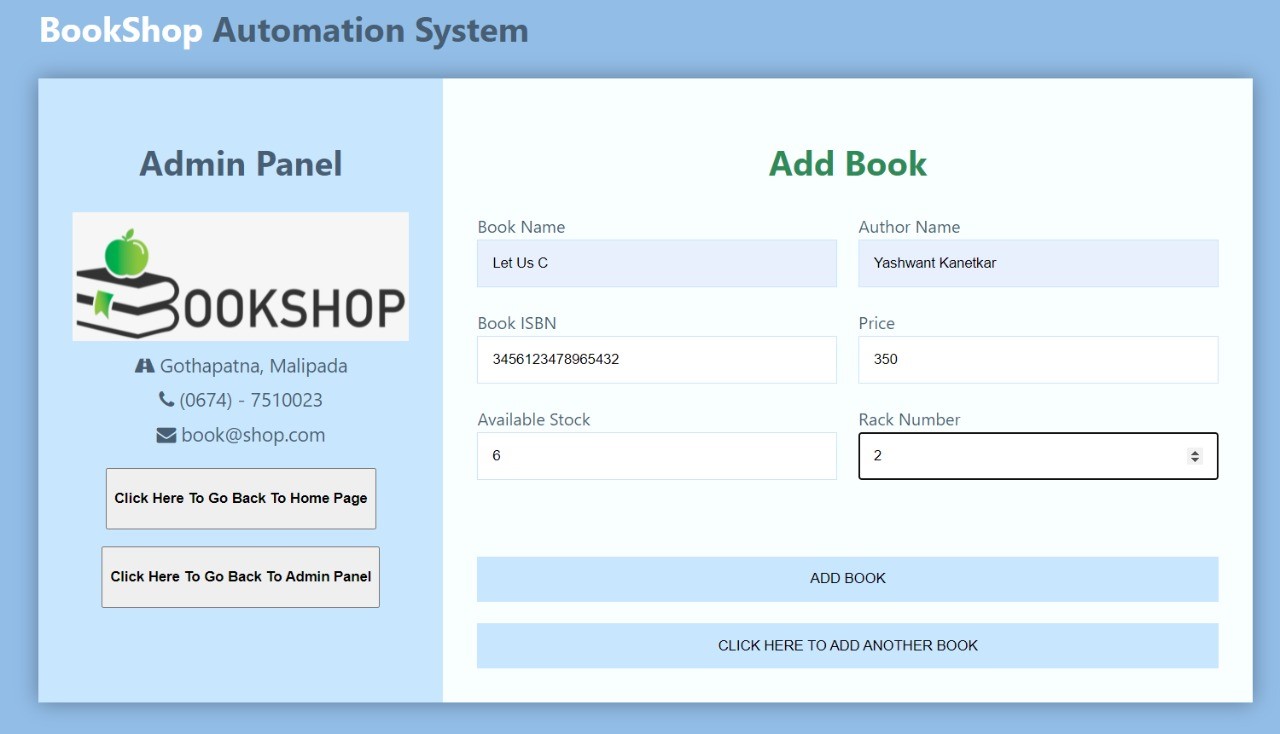


1.2 **Case when entry is not entered correctly -**

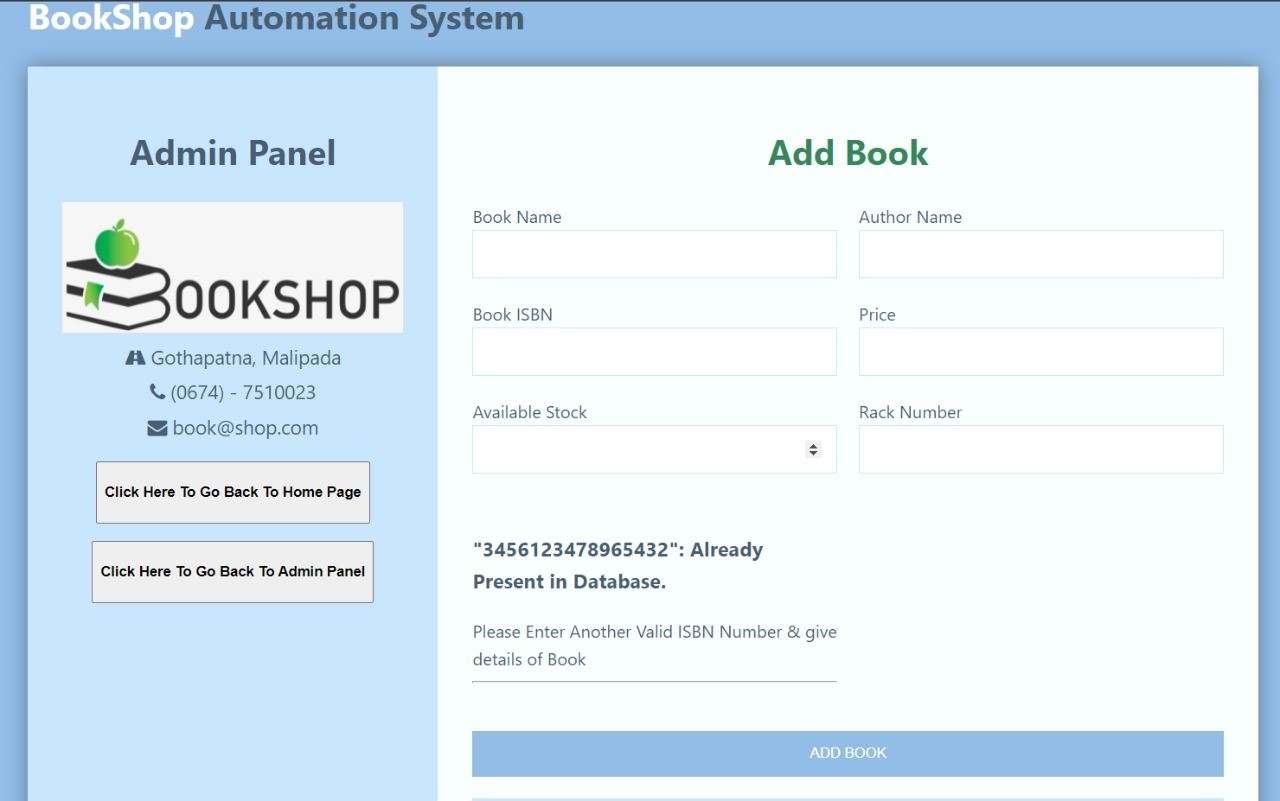
Expected output: - dialog box pops up



1.3 When ISBN of an already existing book is entered

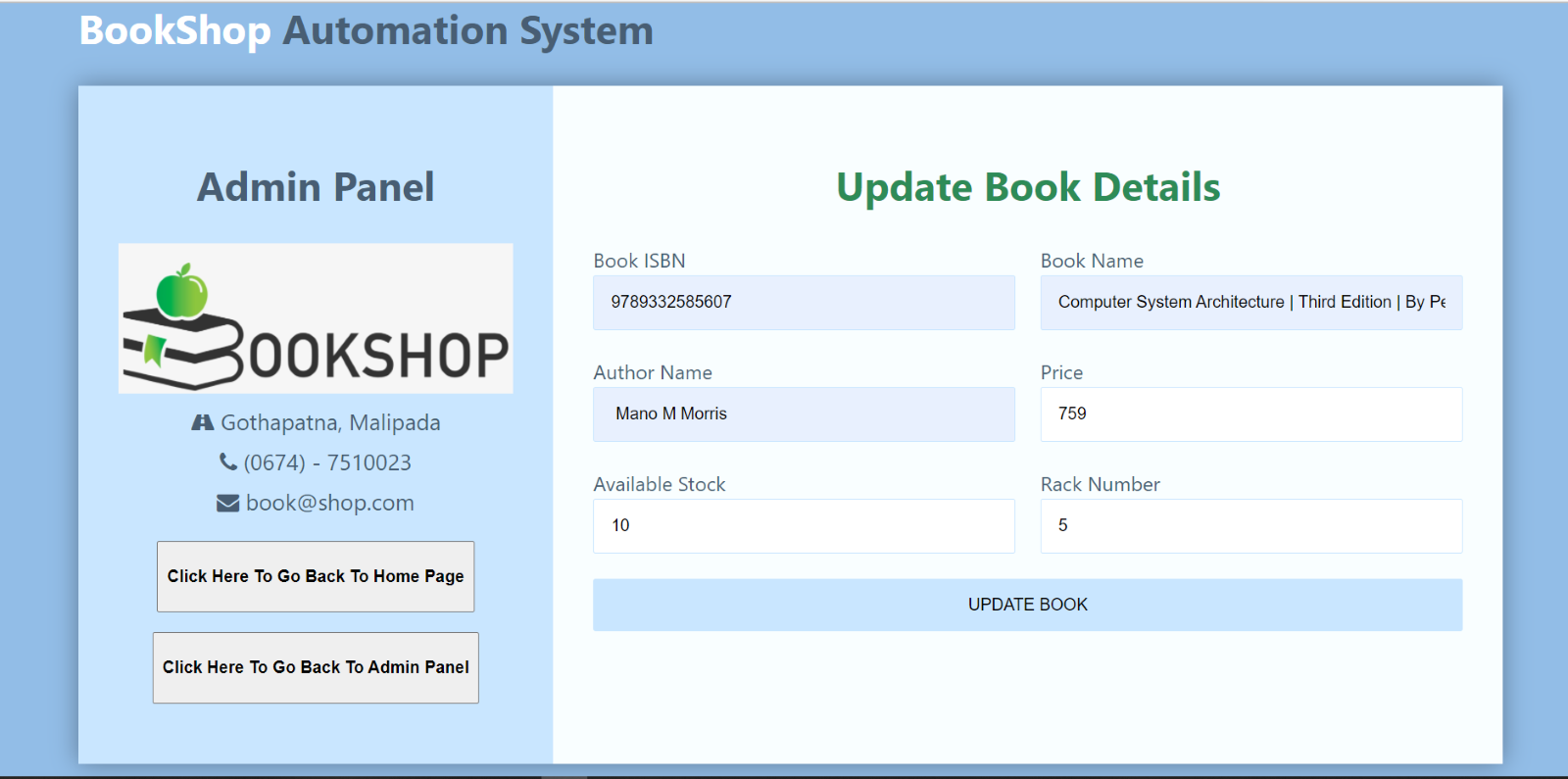


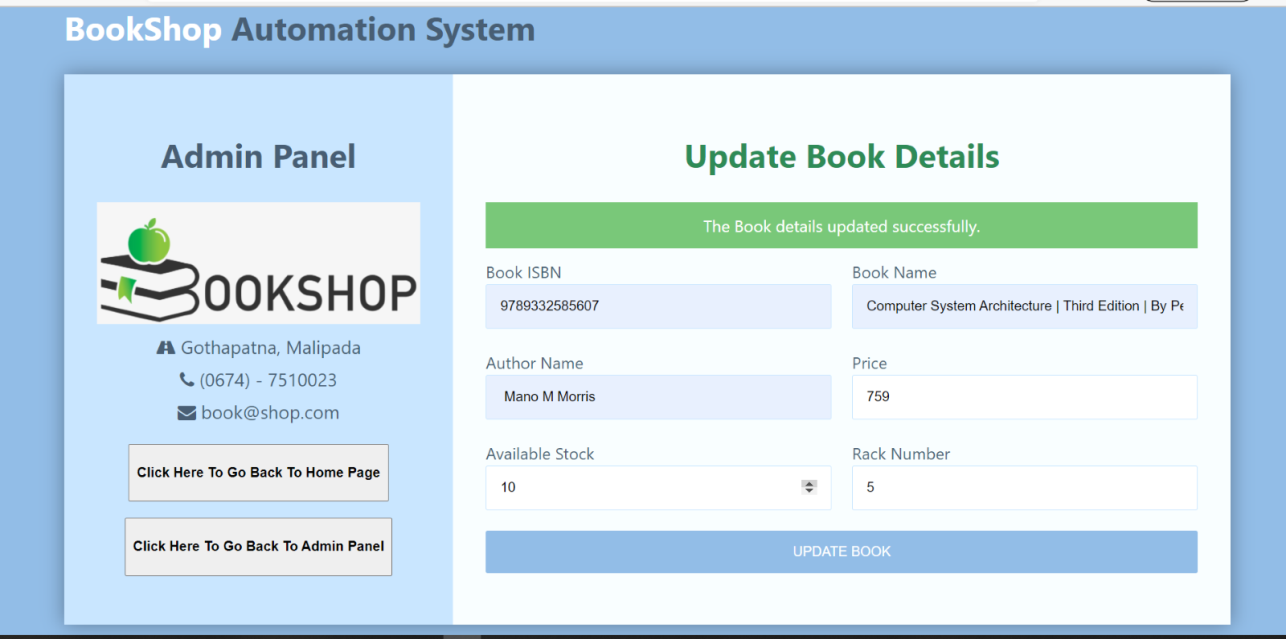
Expected output: - error message appears



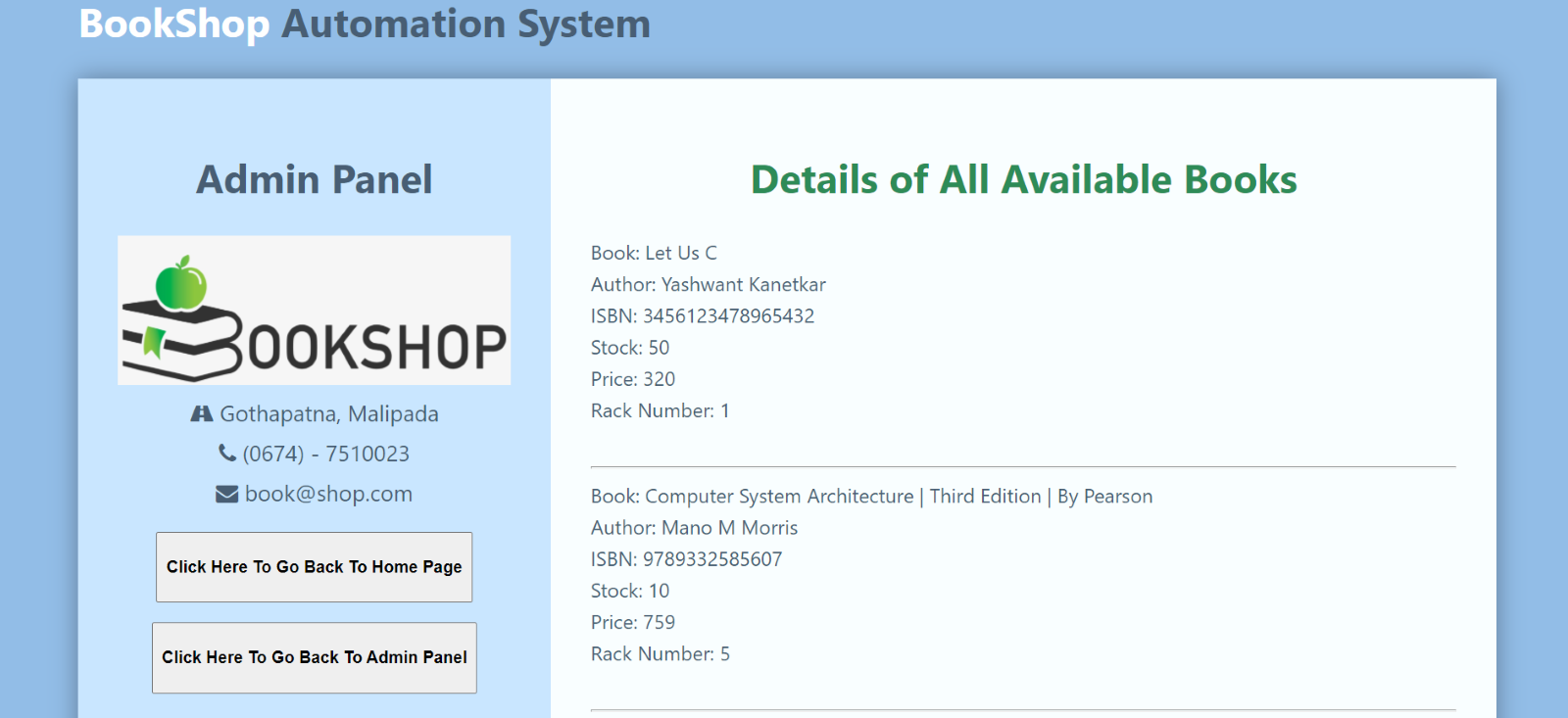
2. Update stock/details of book -

2.1 **Case when all form entries are entered correctly-**

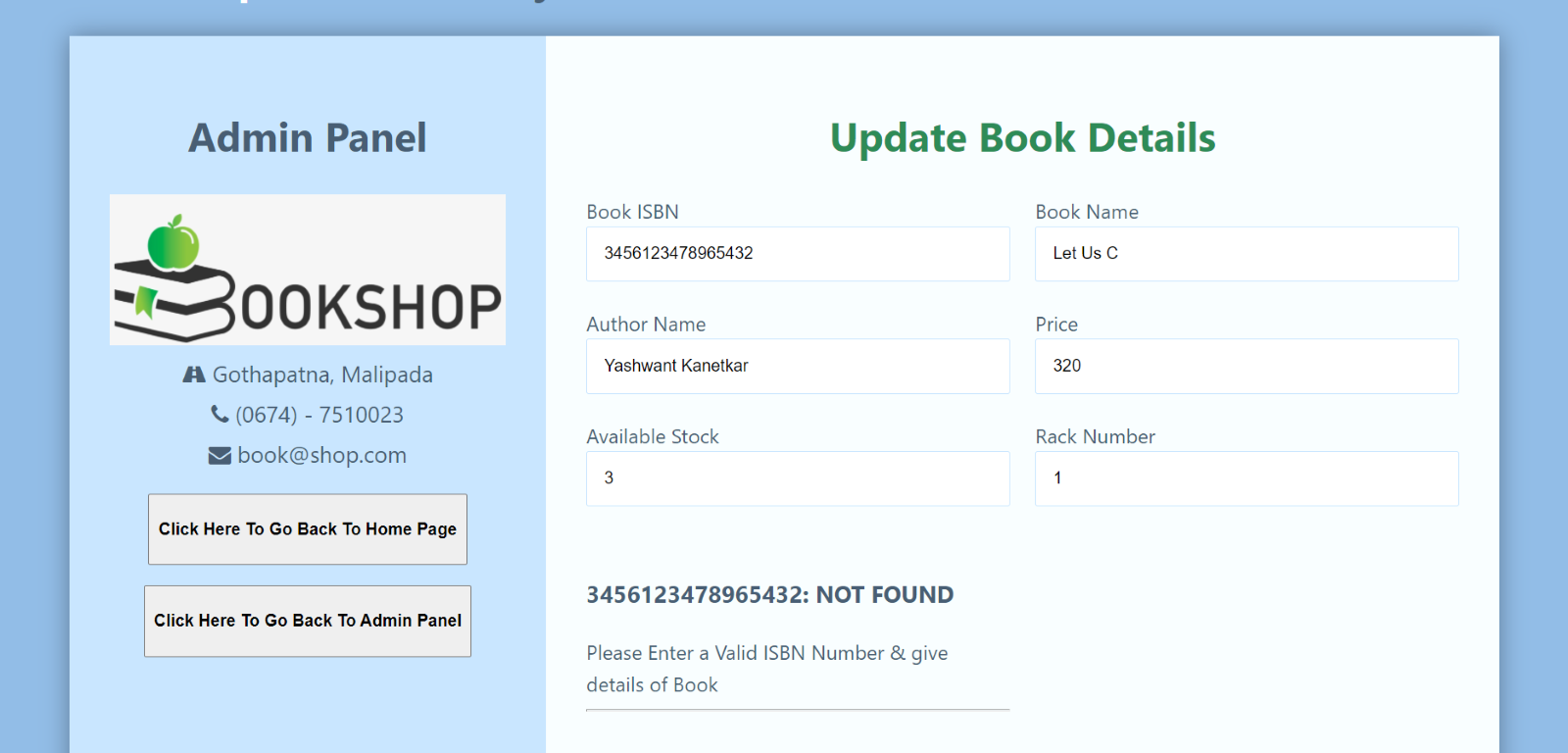


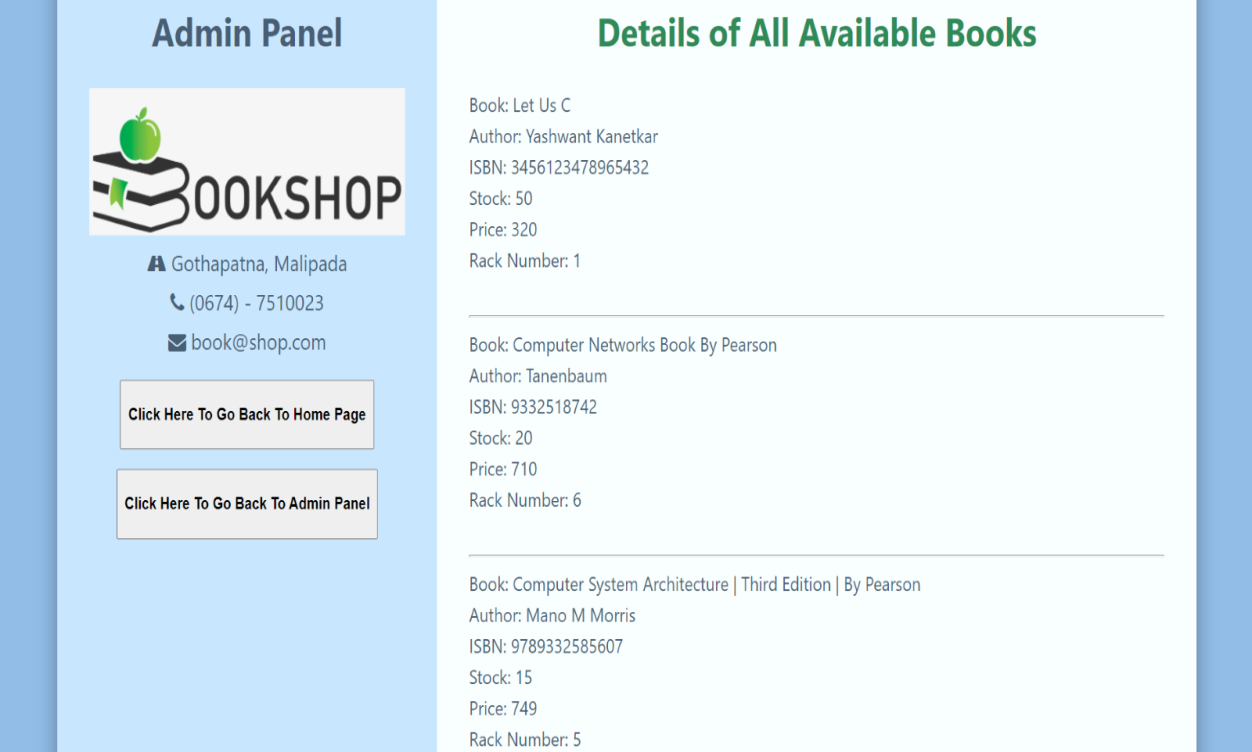


**Expected output: Return back to the update database panel-**

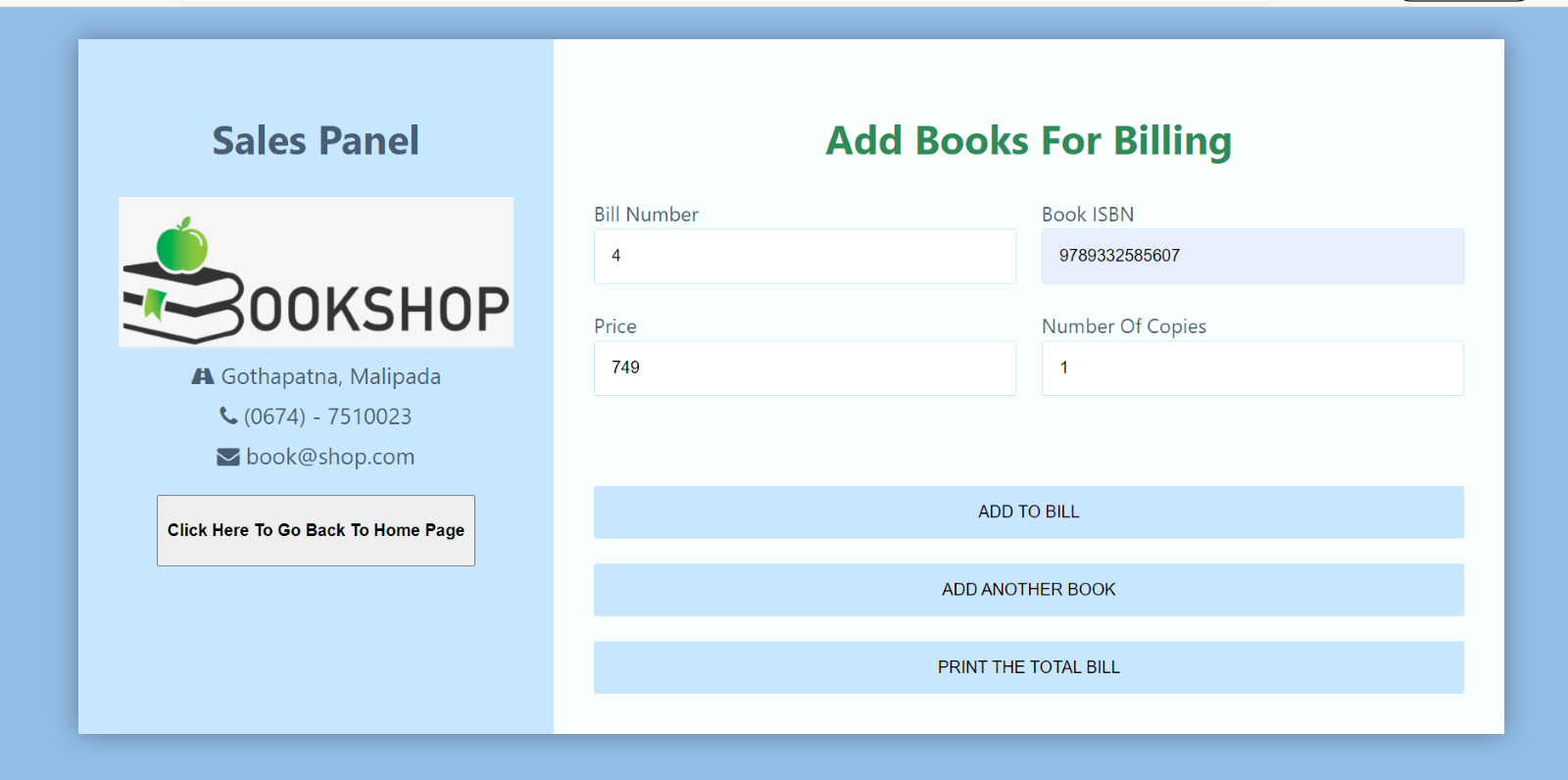


2.2 **Case when entry is not entered correctly -**

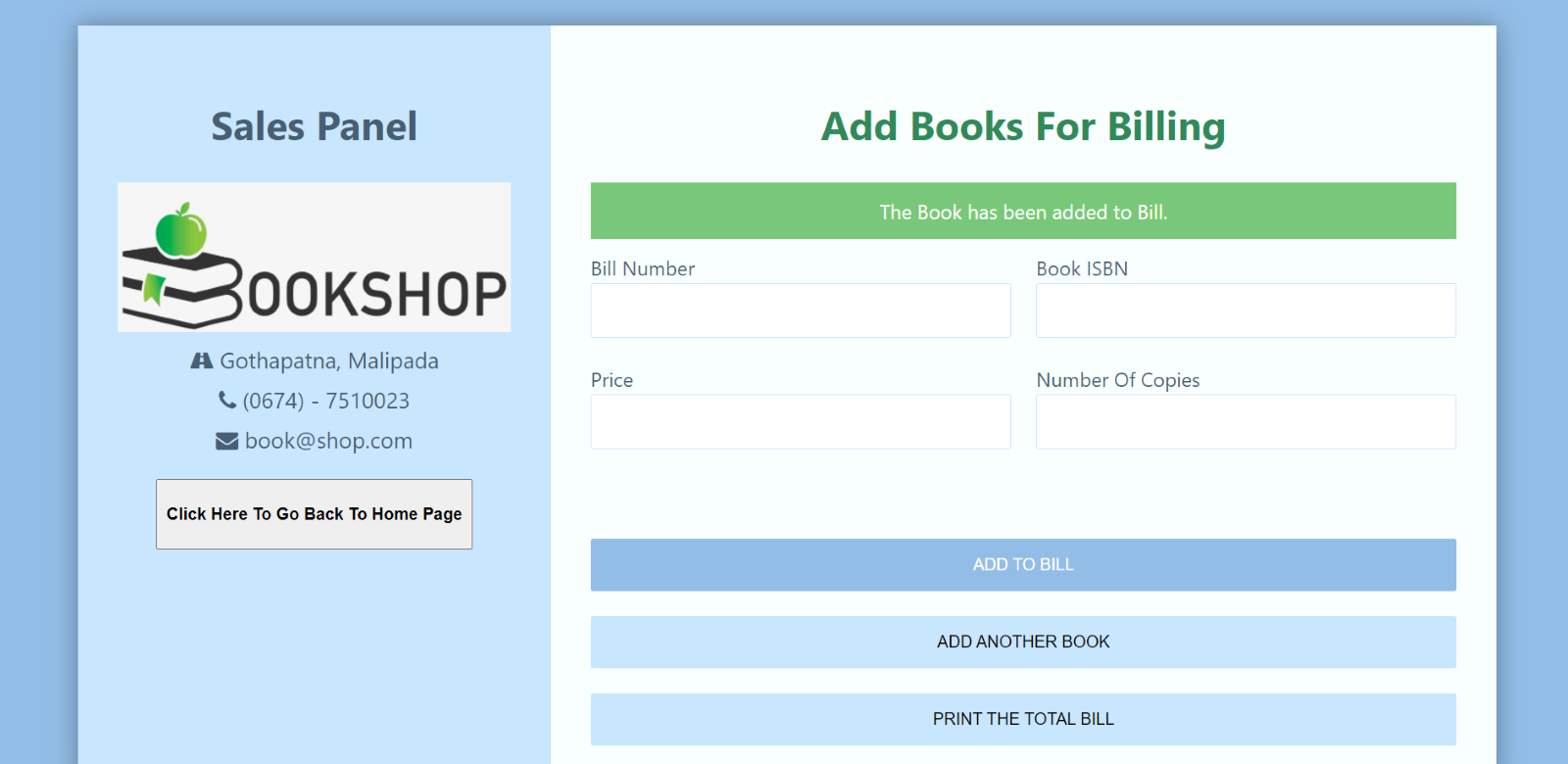
Expected output: - error message appears **

3. Details of All book Available - 

4. Add Book for Billing: -

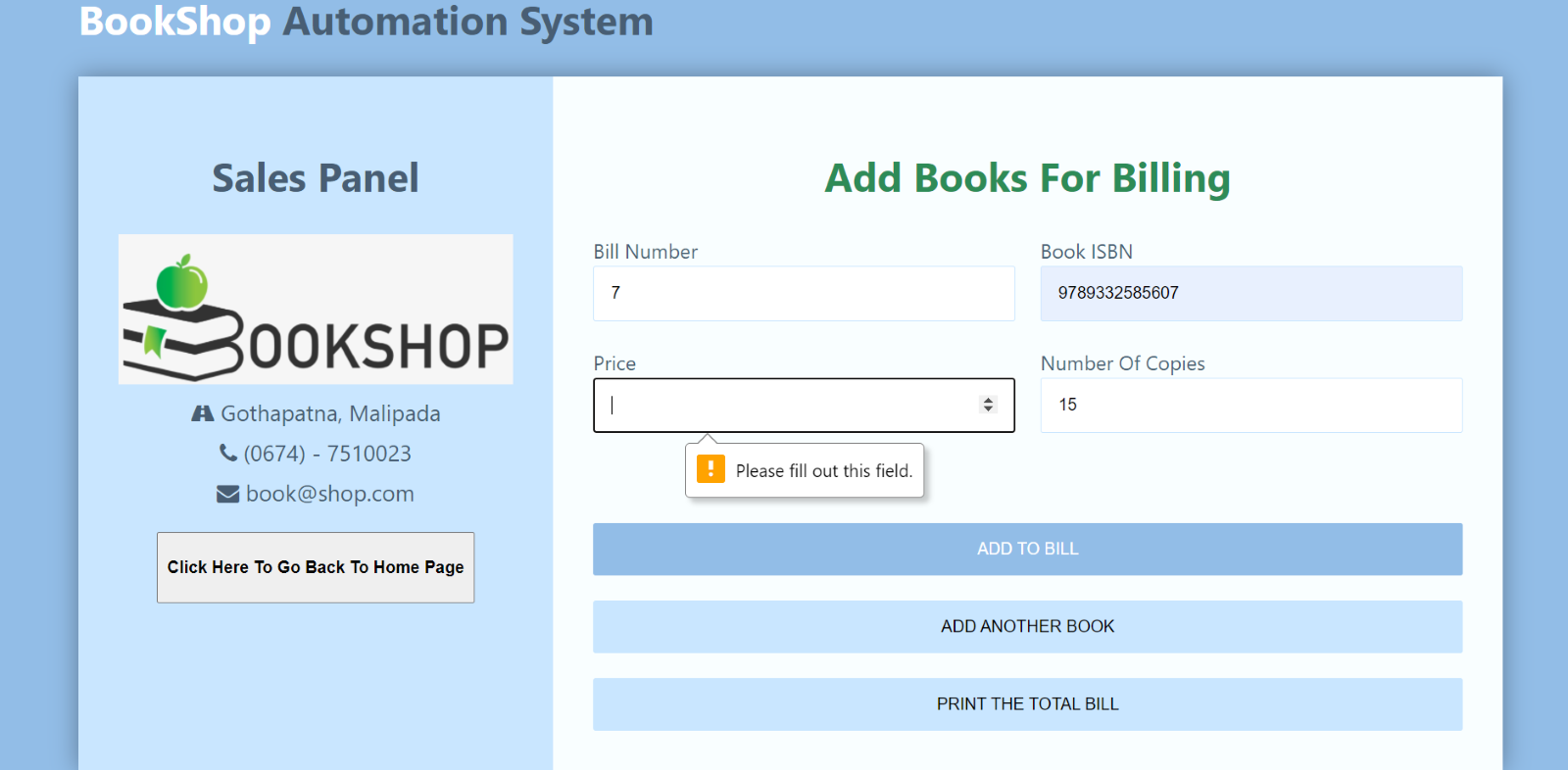
4.1 **Case when all form entries are entered successfully -** 

**Expected output: book successfully added to billing:**

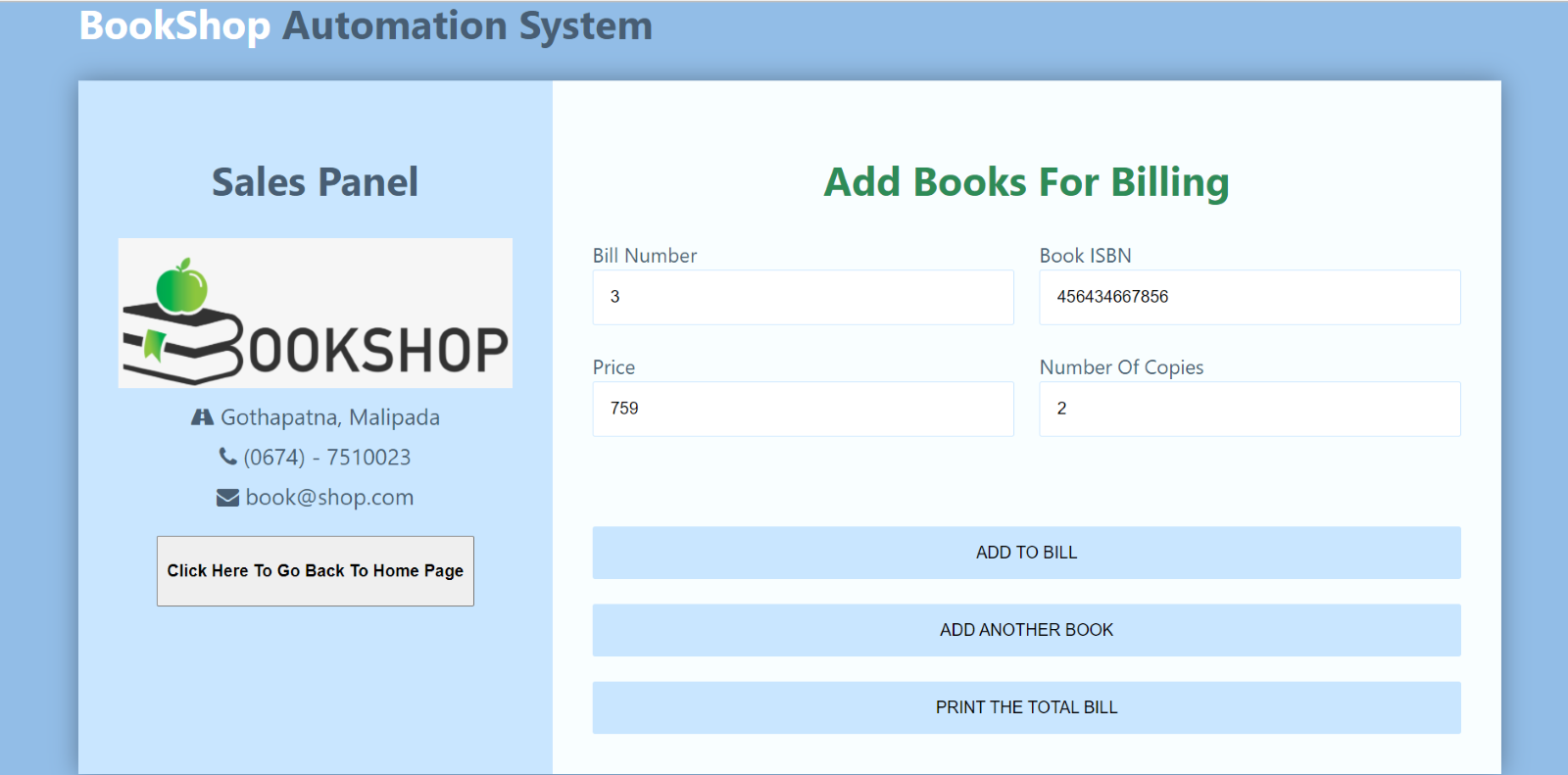


4.2 **Case when entry is not entered correctly -**

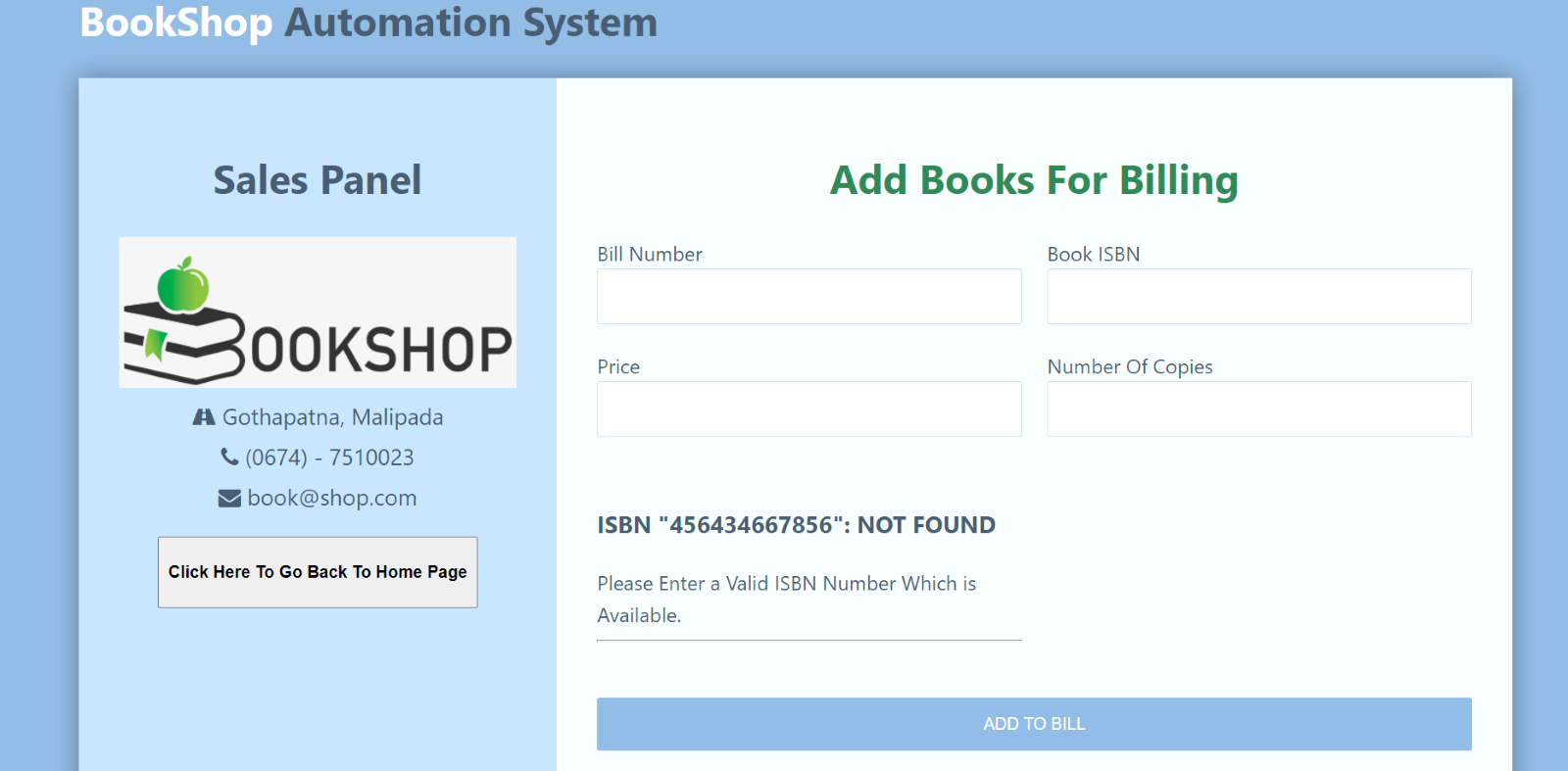
Expected output: - dialog box pops up



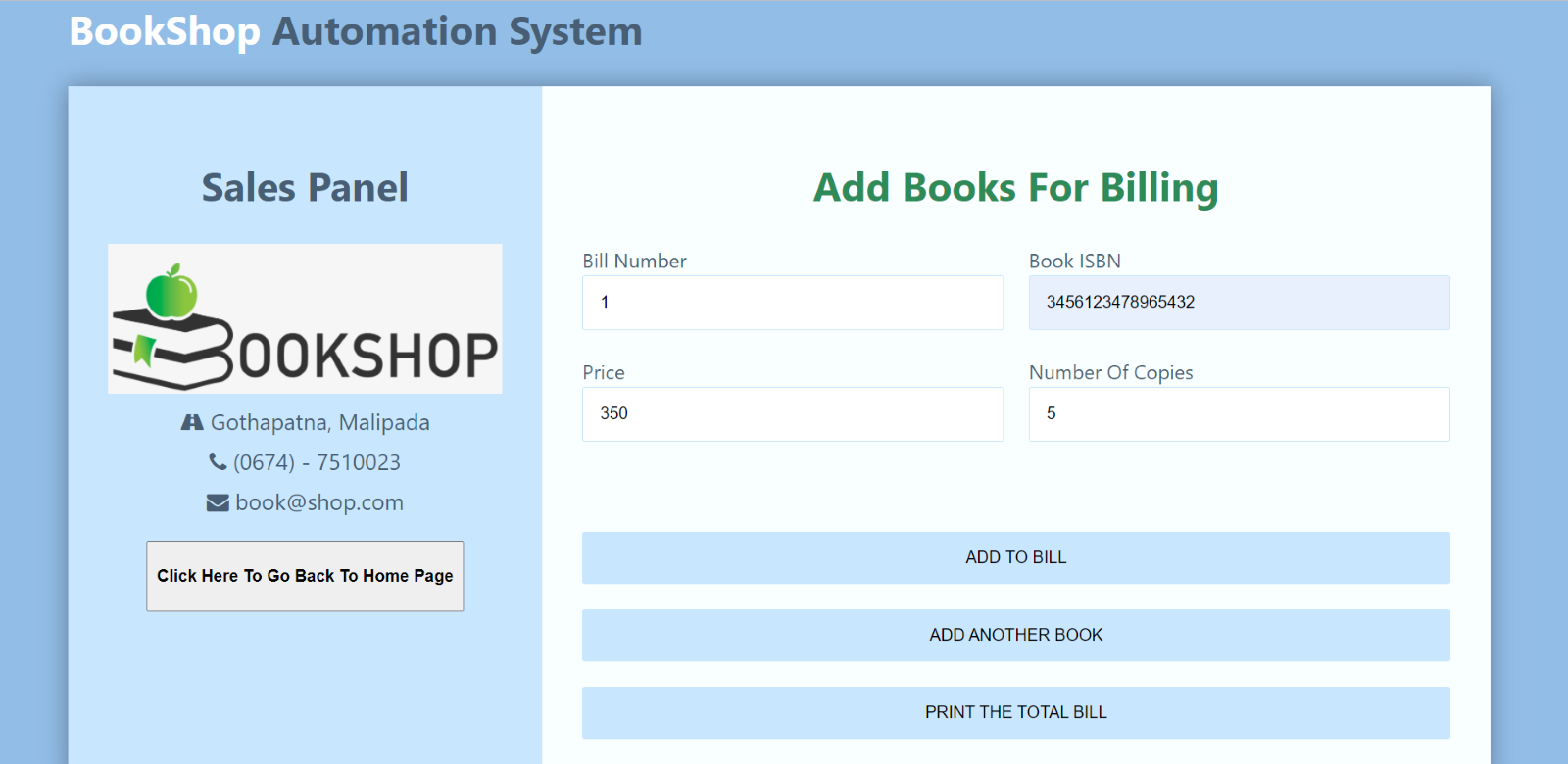
4.3 When ISBN is Invalid: -



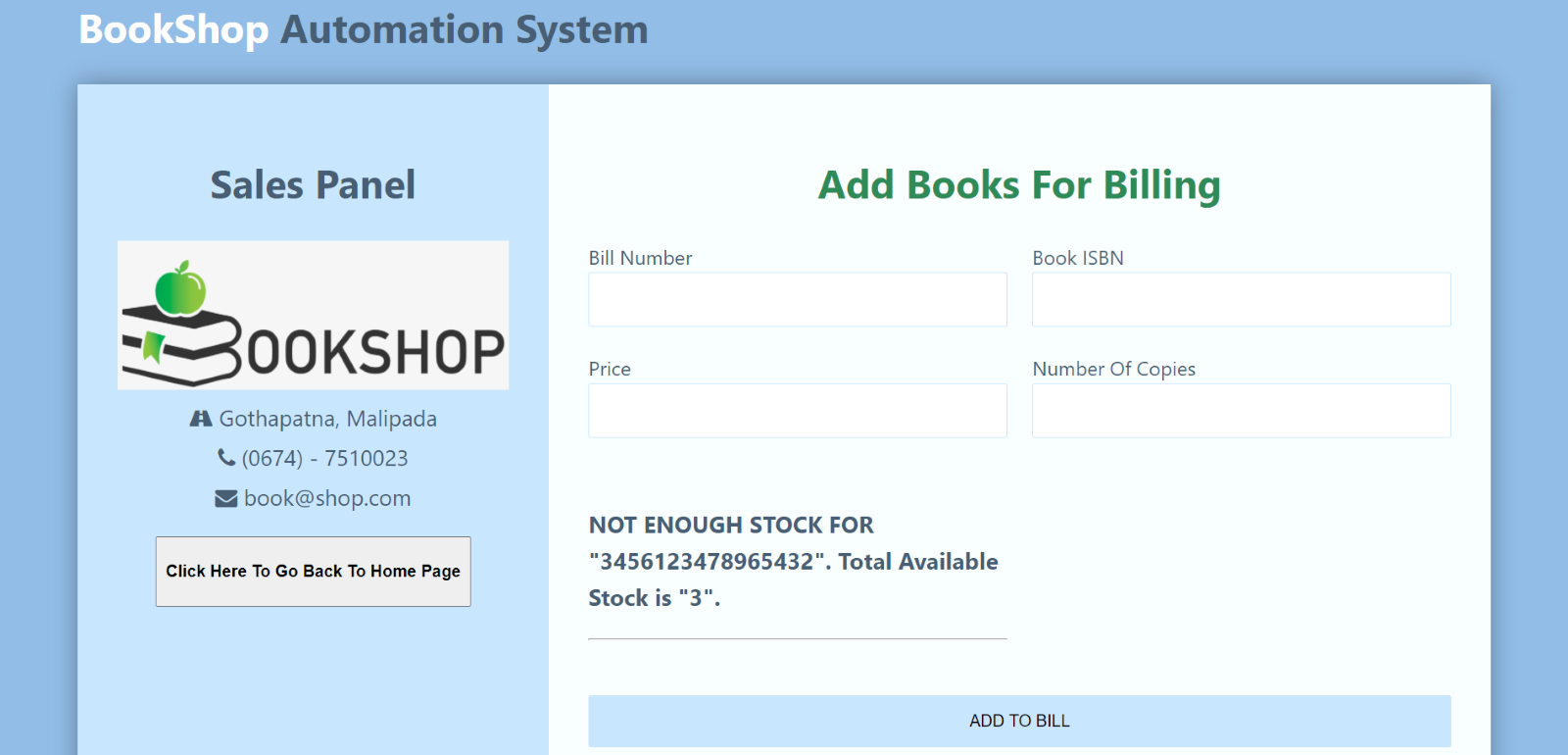
Expected output – error message appears



4.4 **Case when entered no. of copies exceeds total available in stock -**

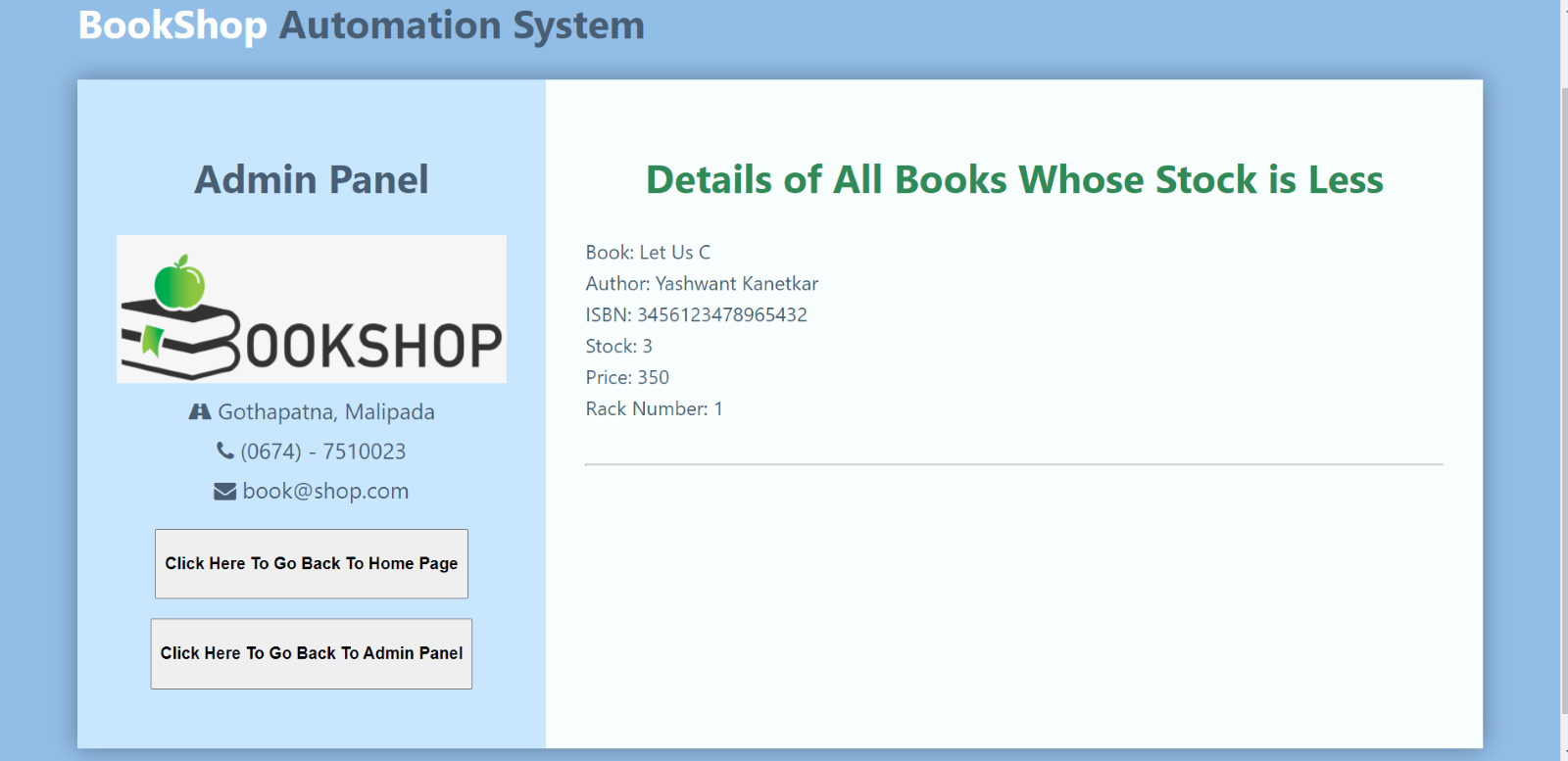


Expected output: - error message appears



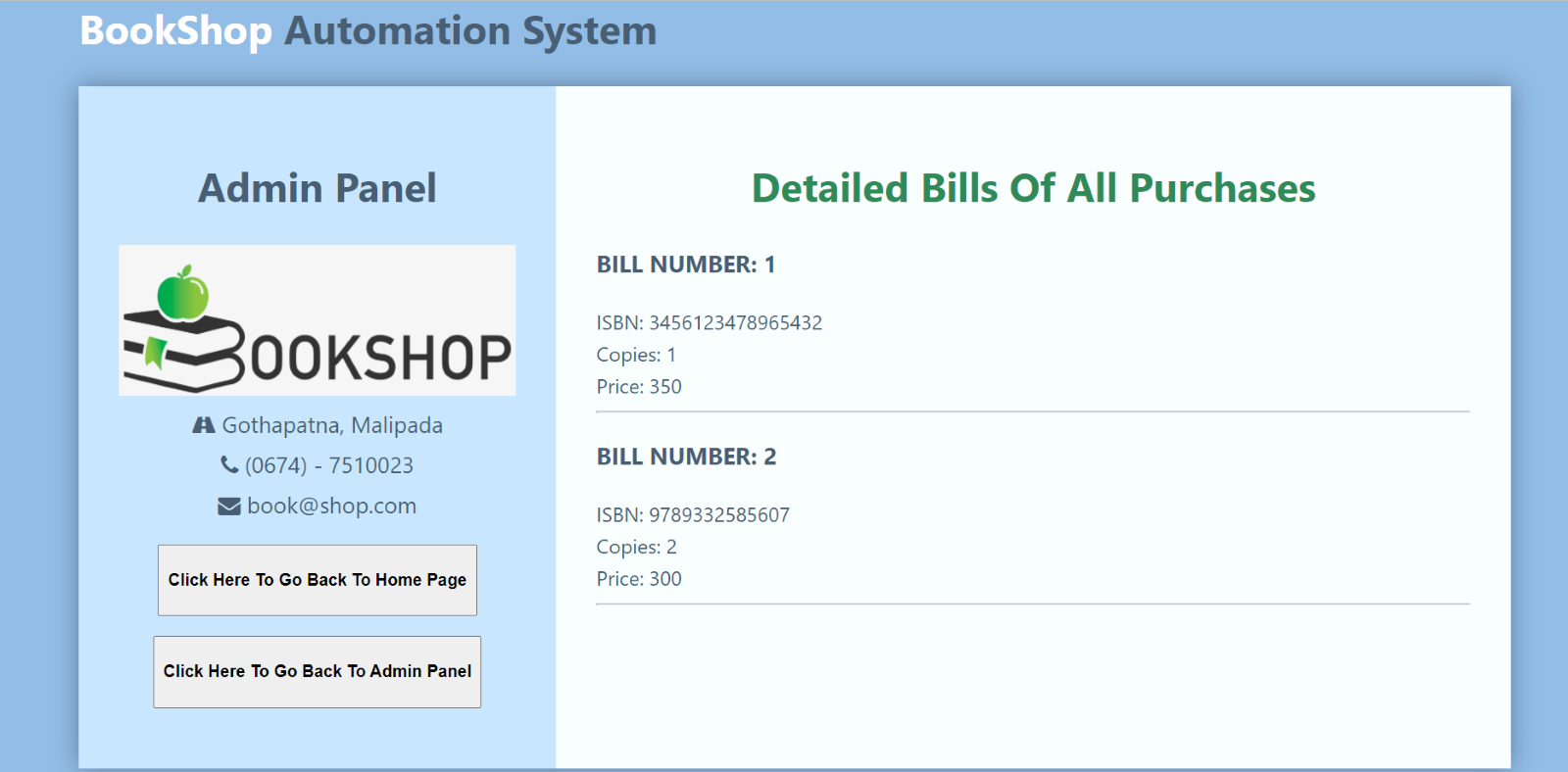
5. Check Inventory for books: -

5.1 If book have less than 5 copies, it shows up in inventory section -

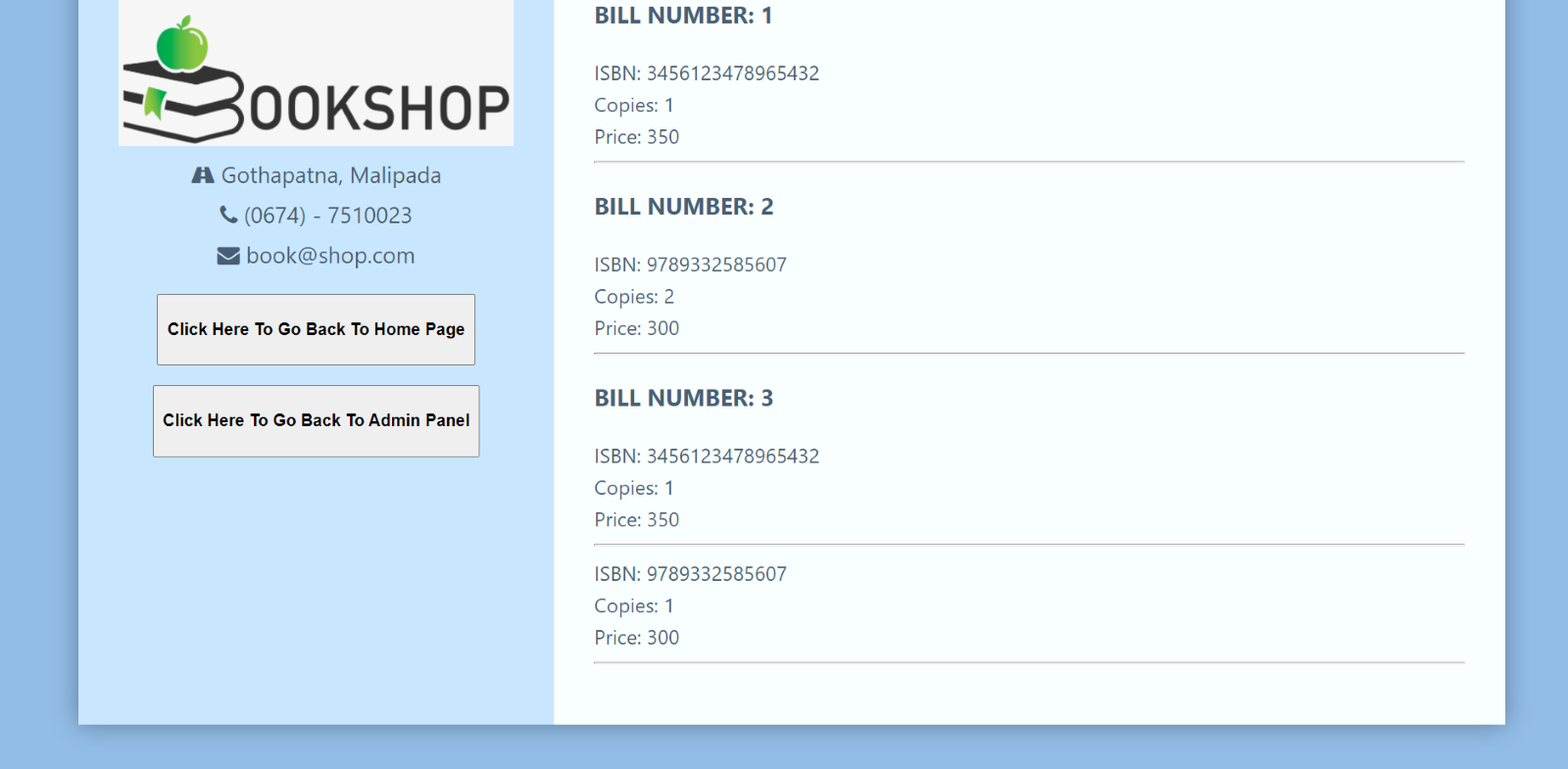


6. Total number of purchases made: -

6.1 When 1 book is purchased-

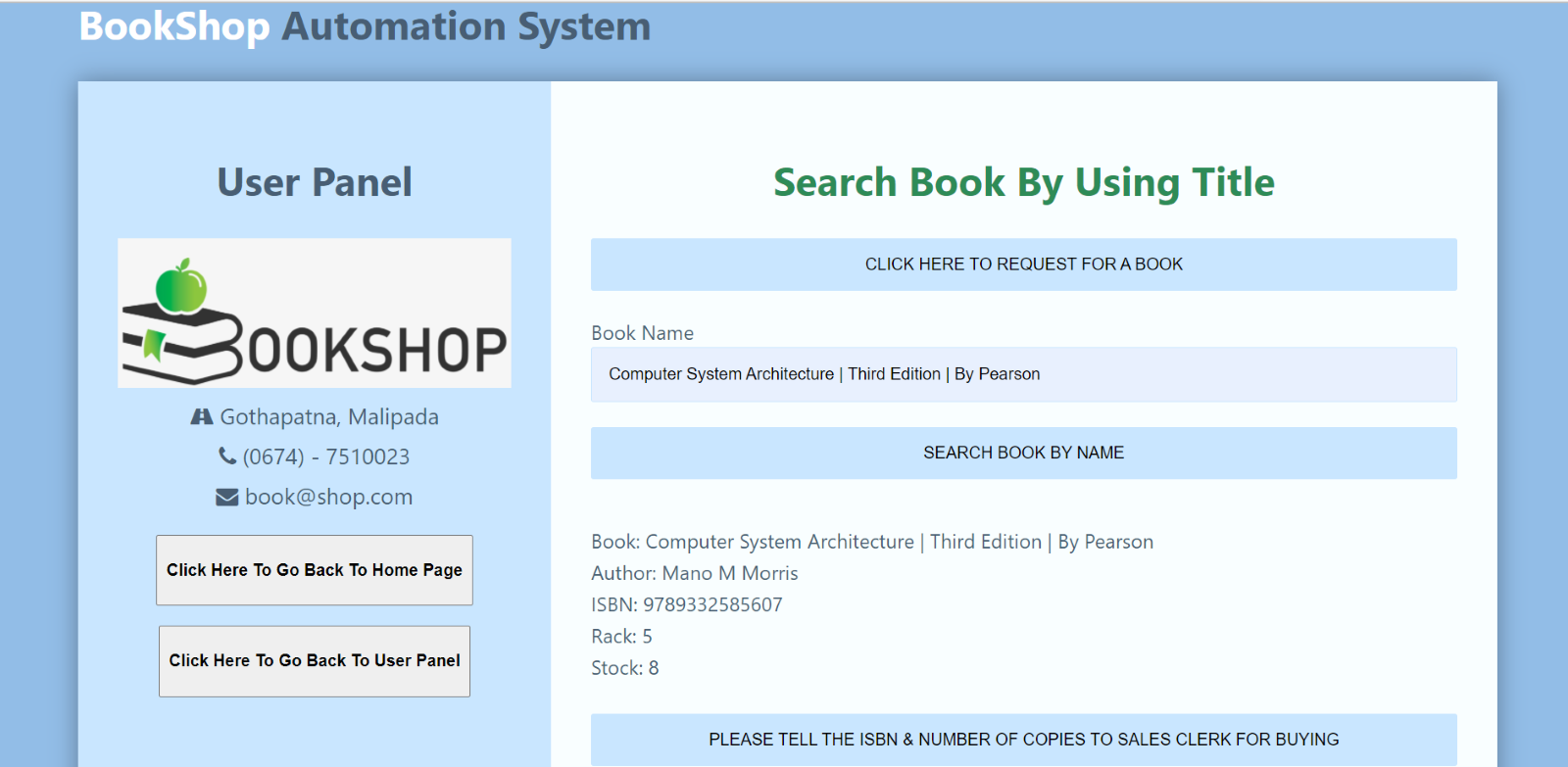


6.2 When more than one book is purchased (bill number 3) -



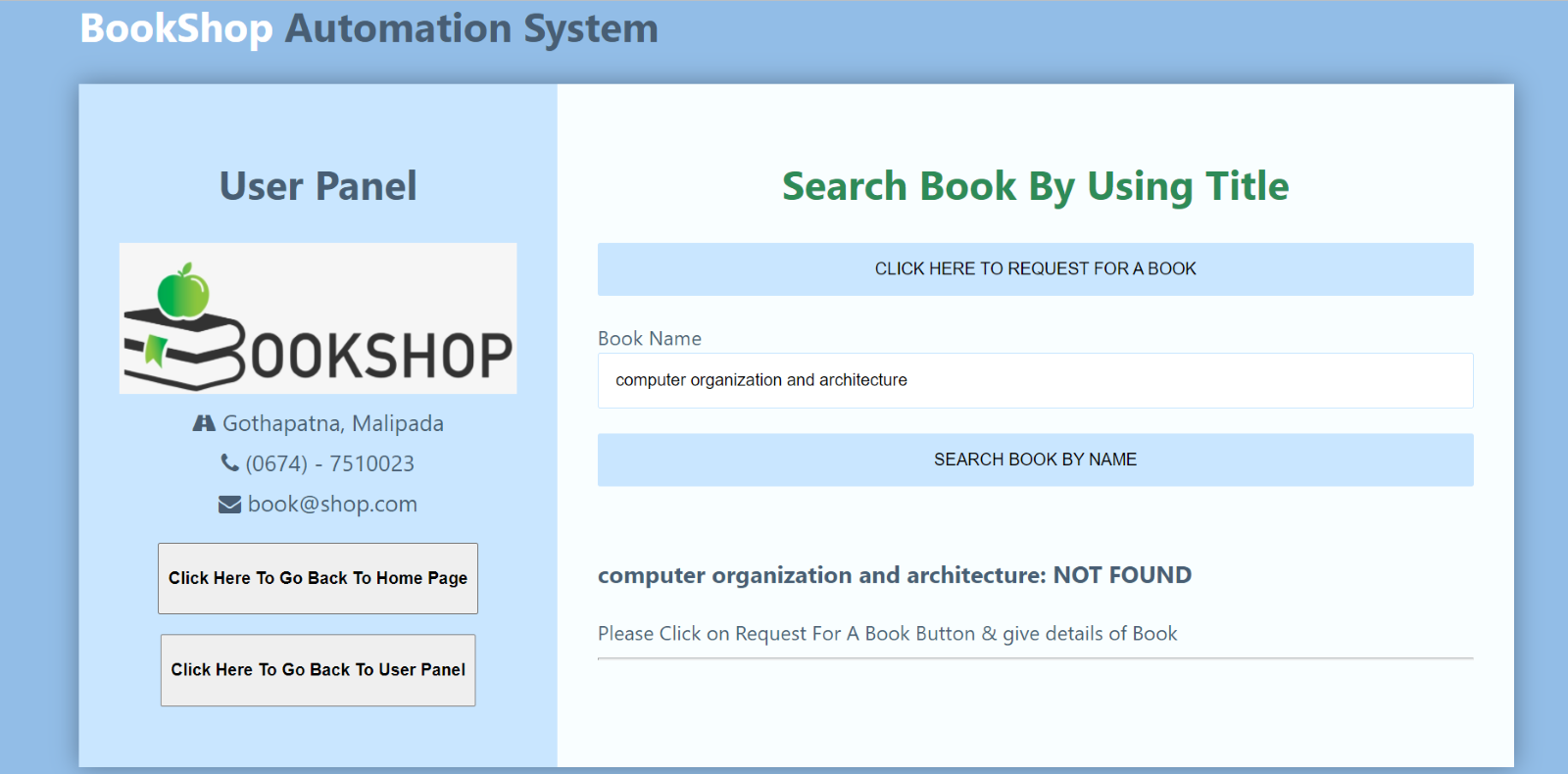
7. Search Book by Using Title: -

7.1 When book is present and is in stock: -



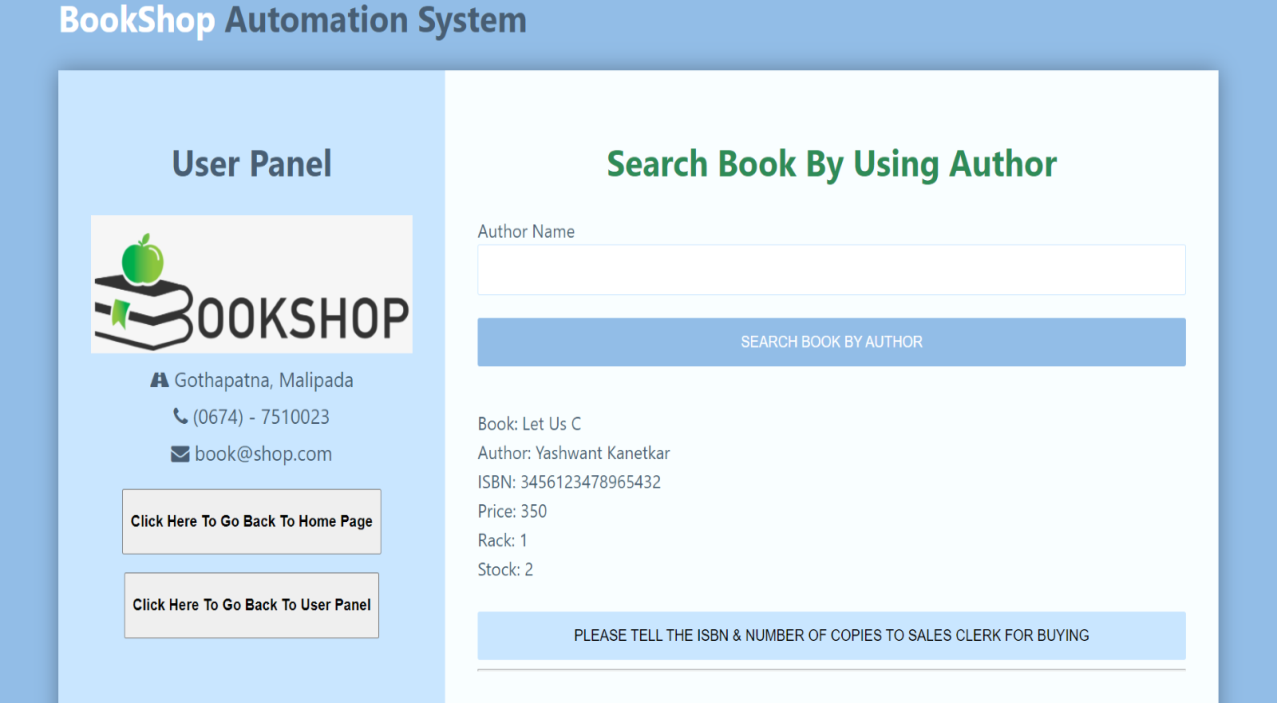
7.2 errors - when book is not present: -

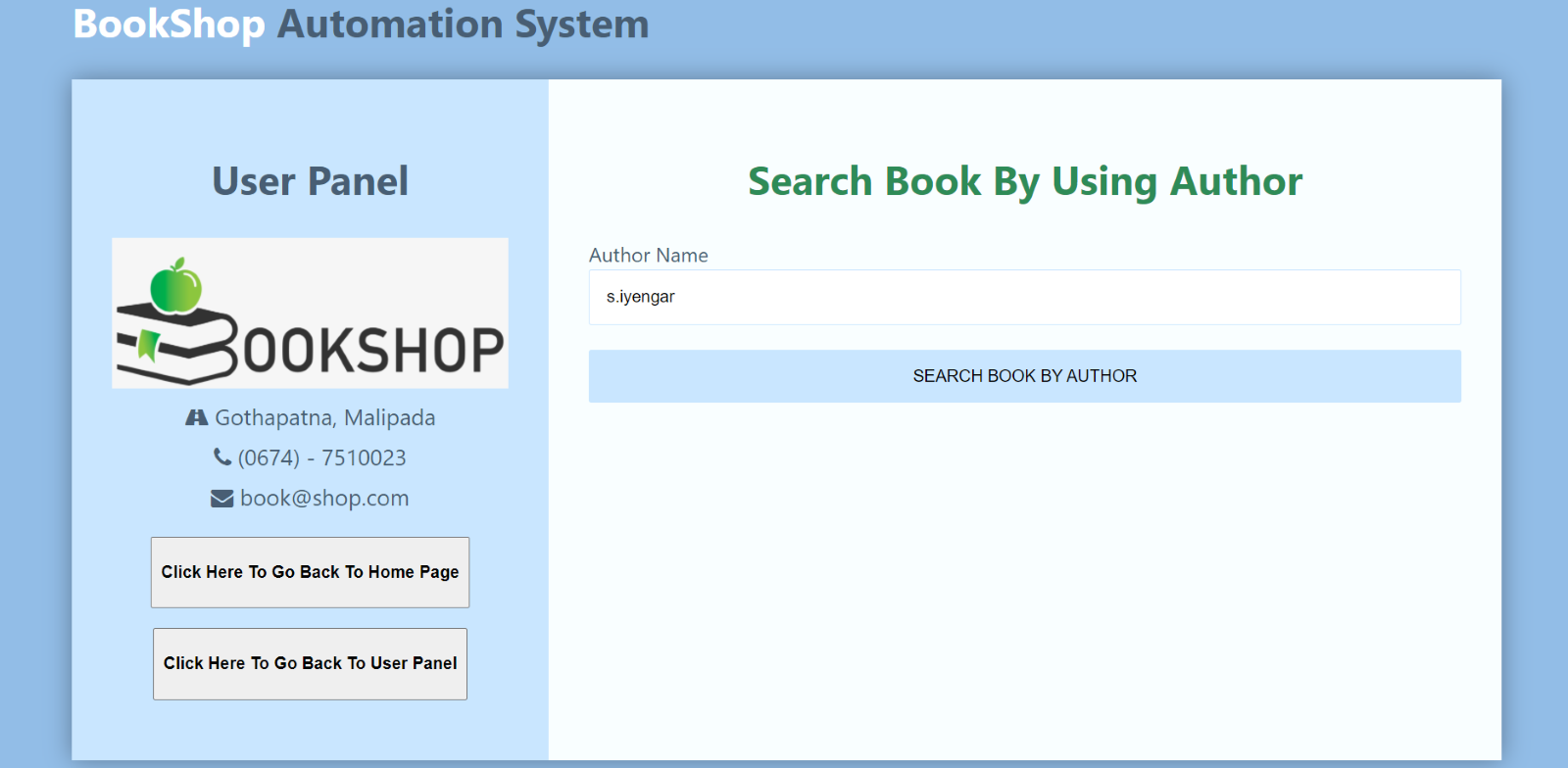
Expected output – error message appears



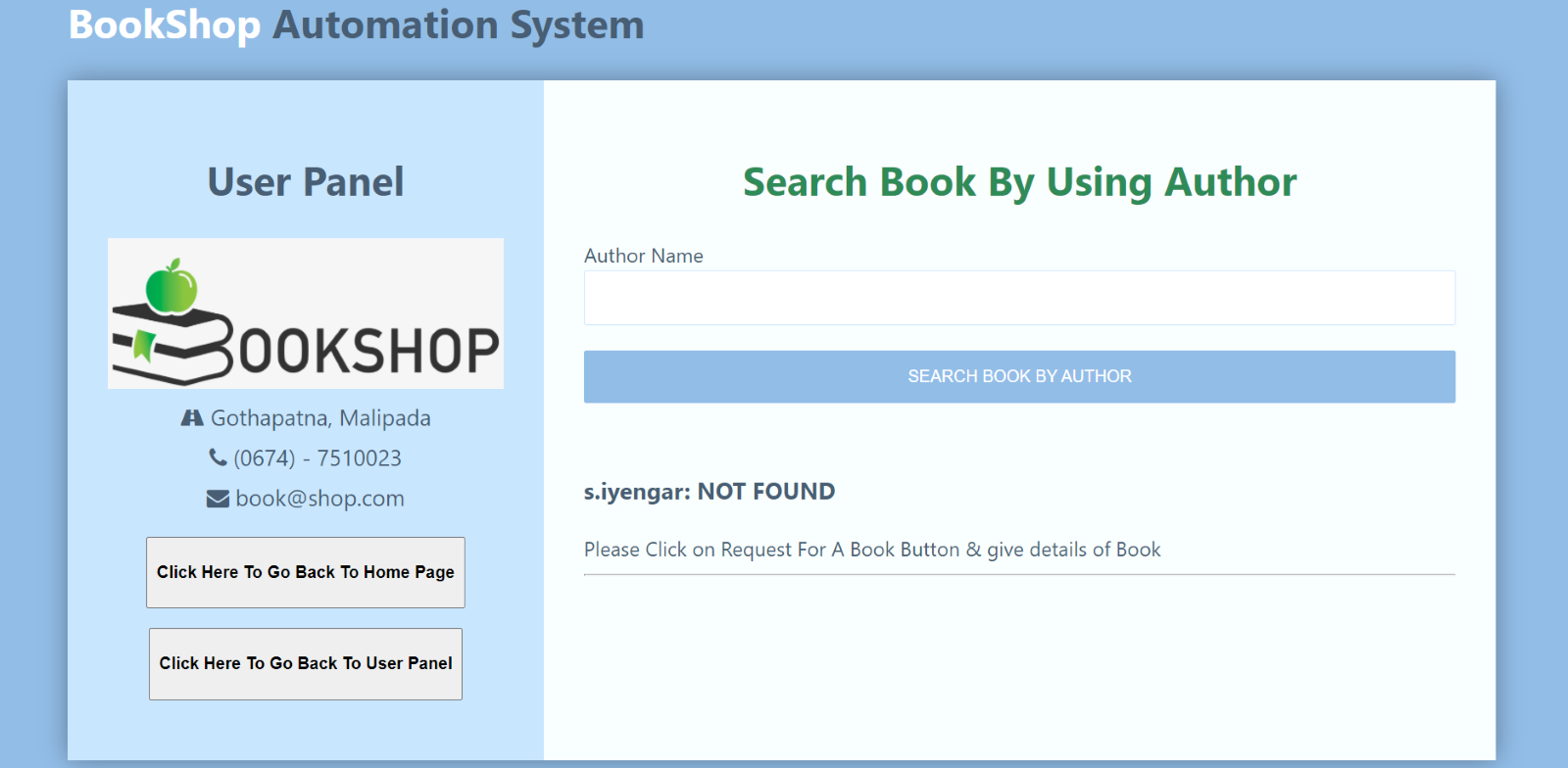
8. Search Book by Using Author Name

Expected output: - details of book and its stock is displayed

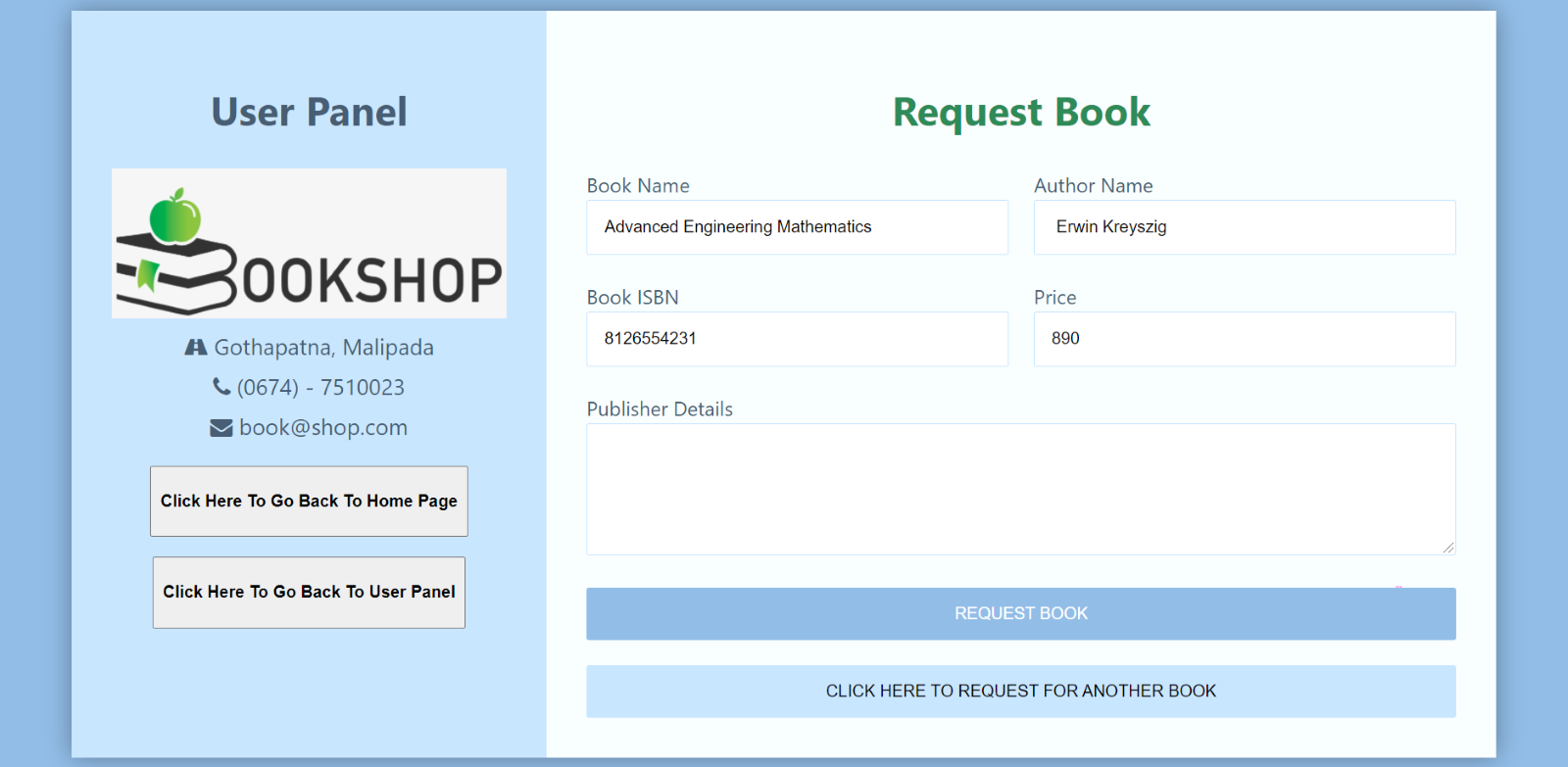


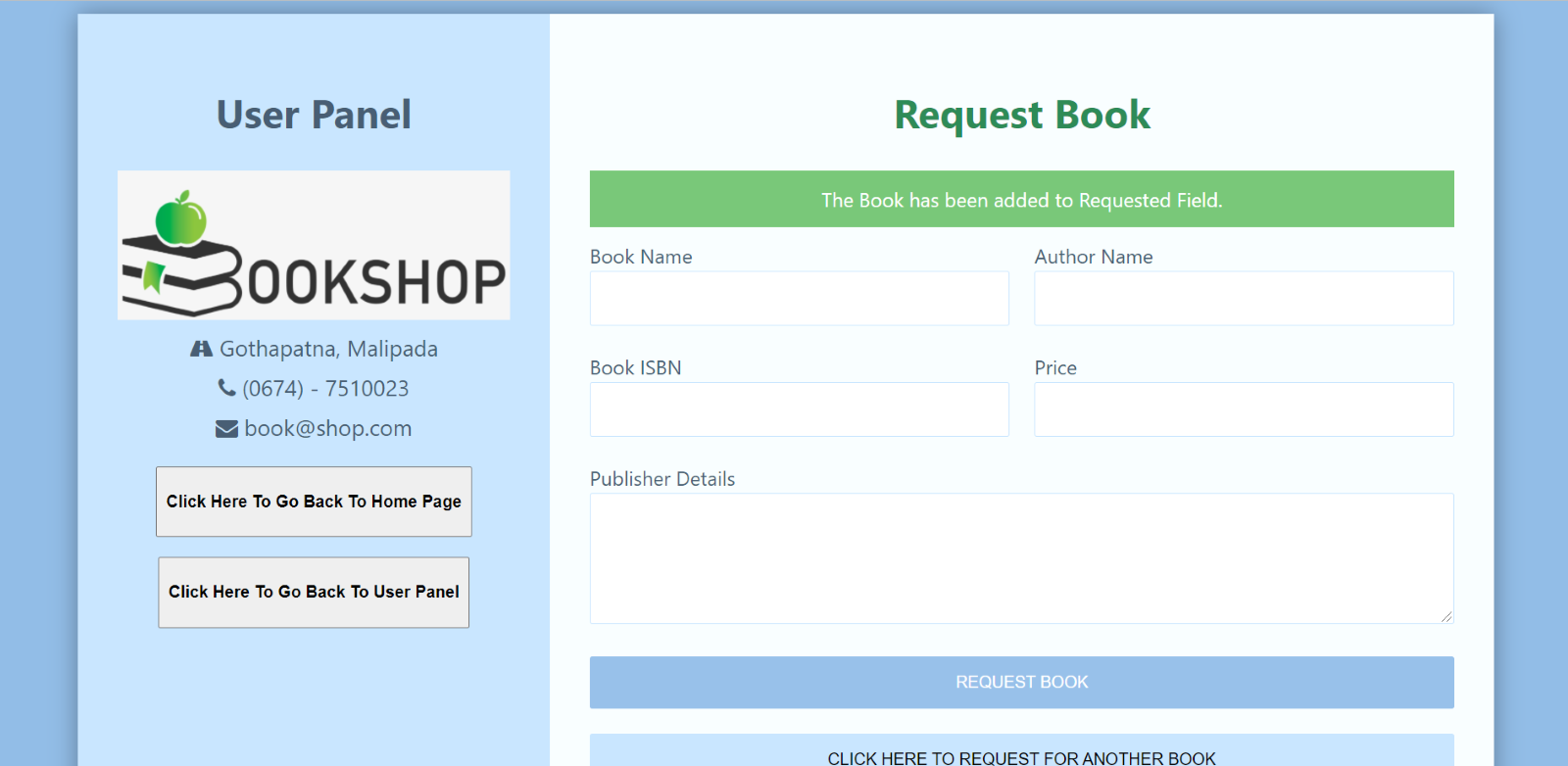
8.2 when book is not present - 

Expected output – error message appears

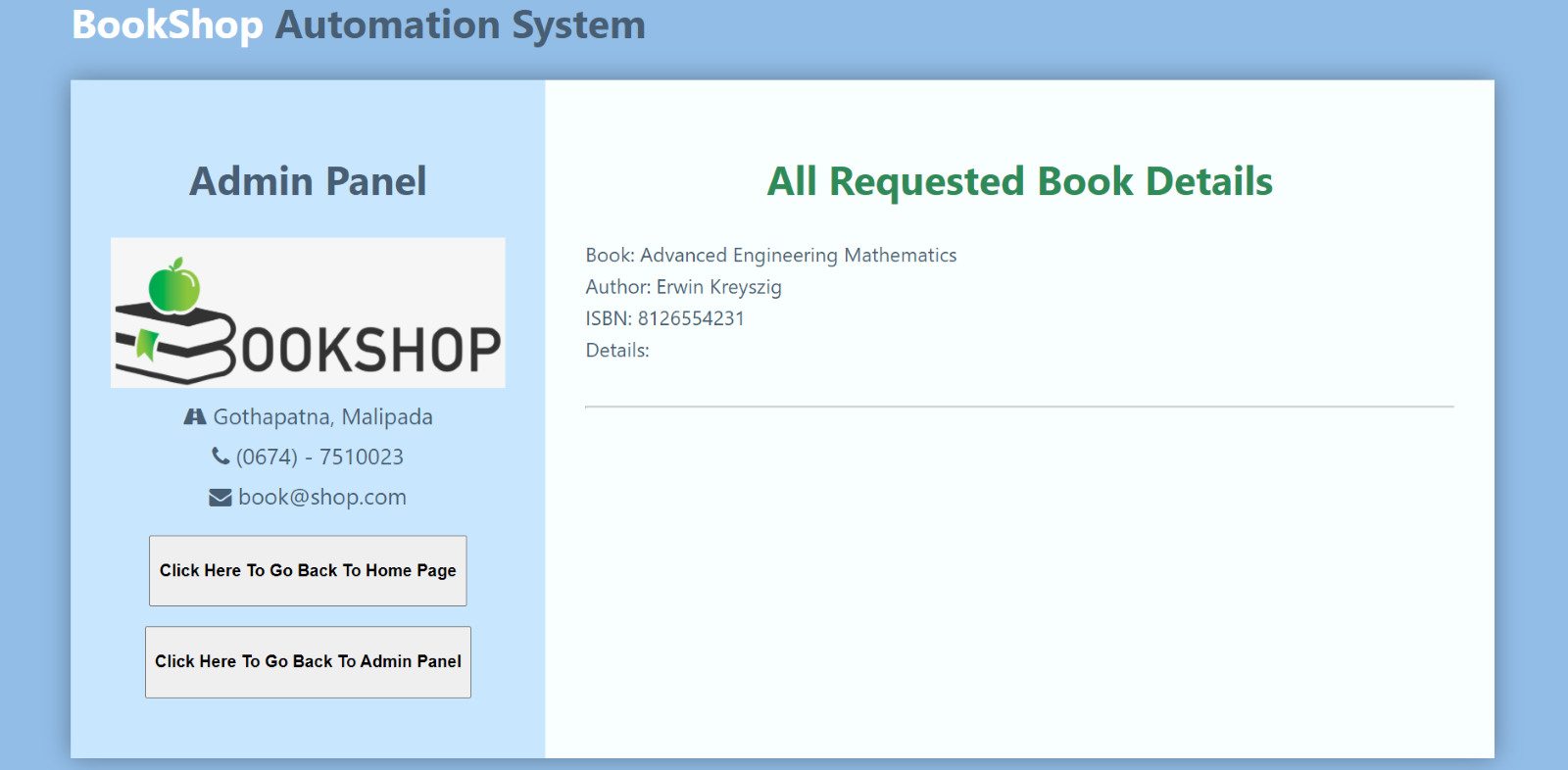


9. Request book: -

9.1 when all fields are entered: - 

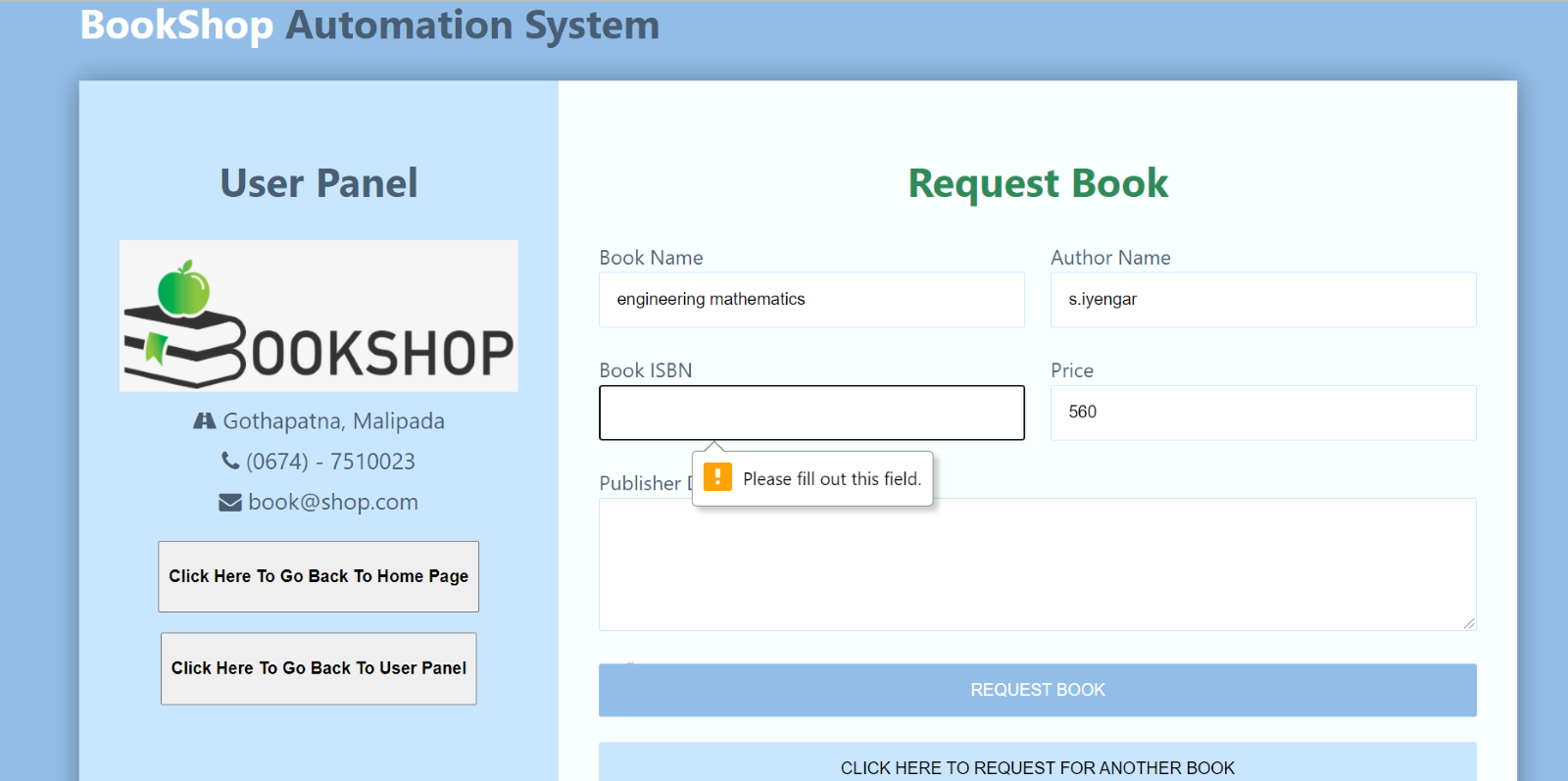


Expected output: - successfully added to request list



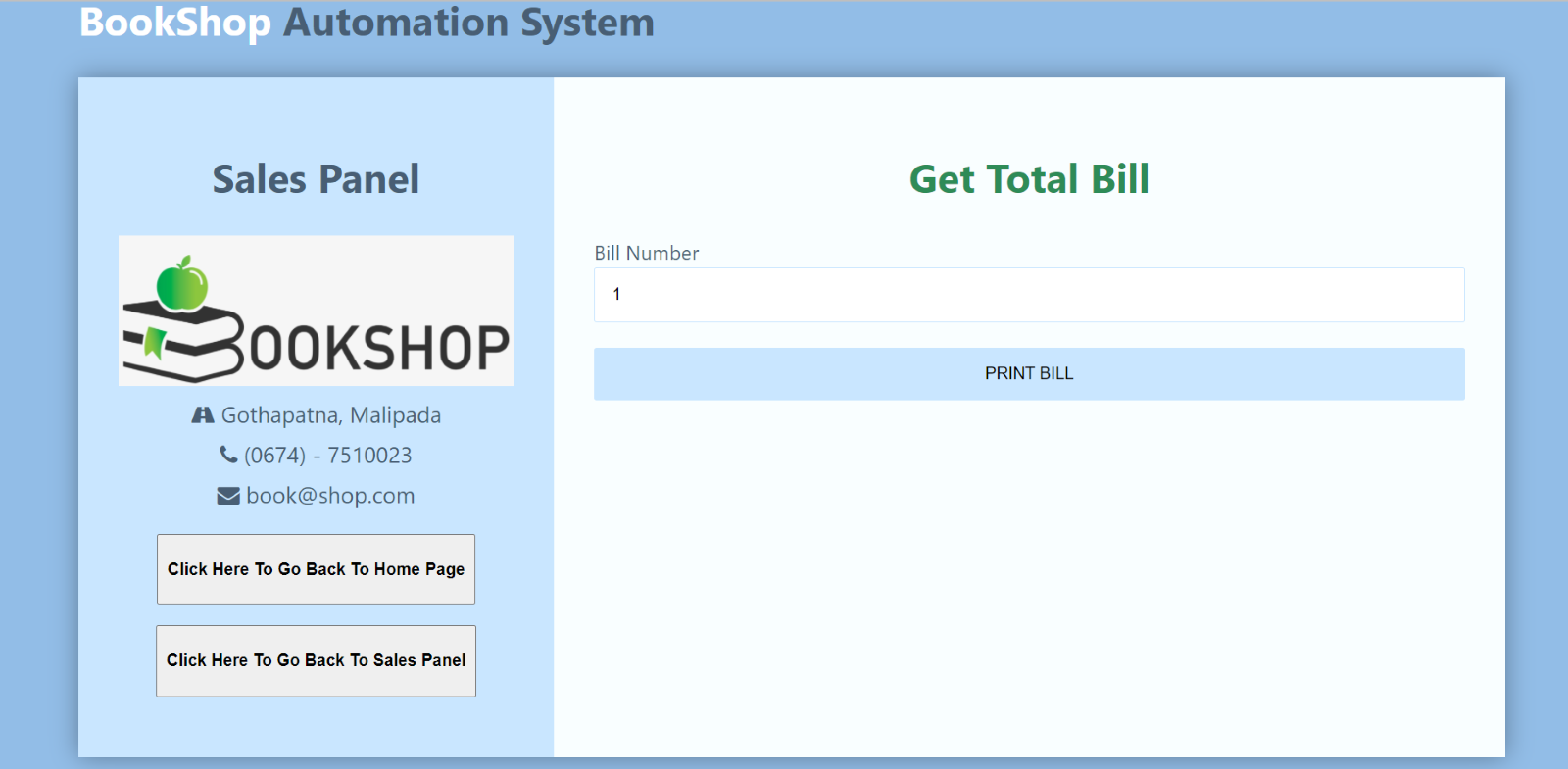
9.2 case when input is not filled correctly: -

Expected output - dialog box pops up

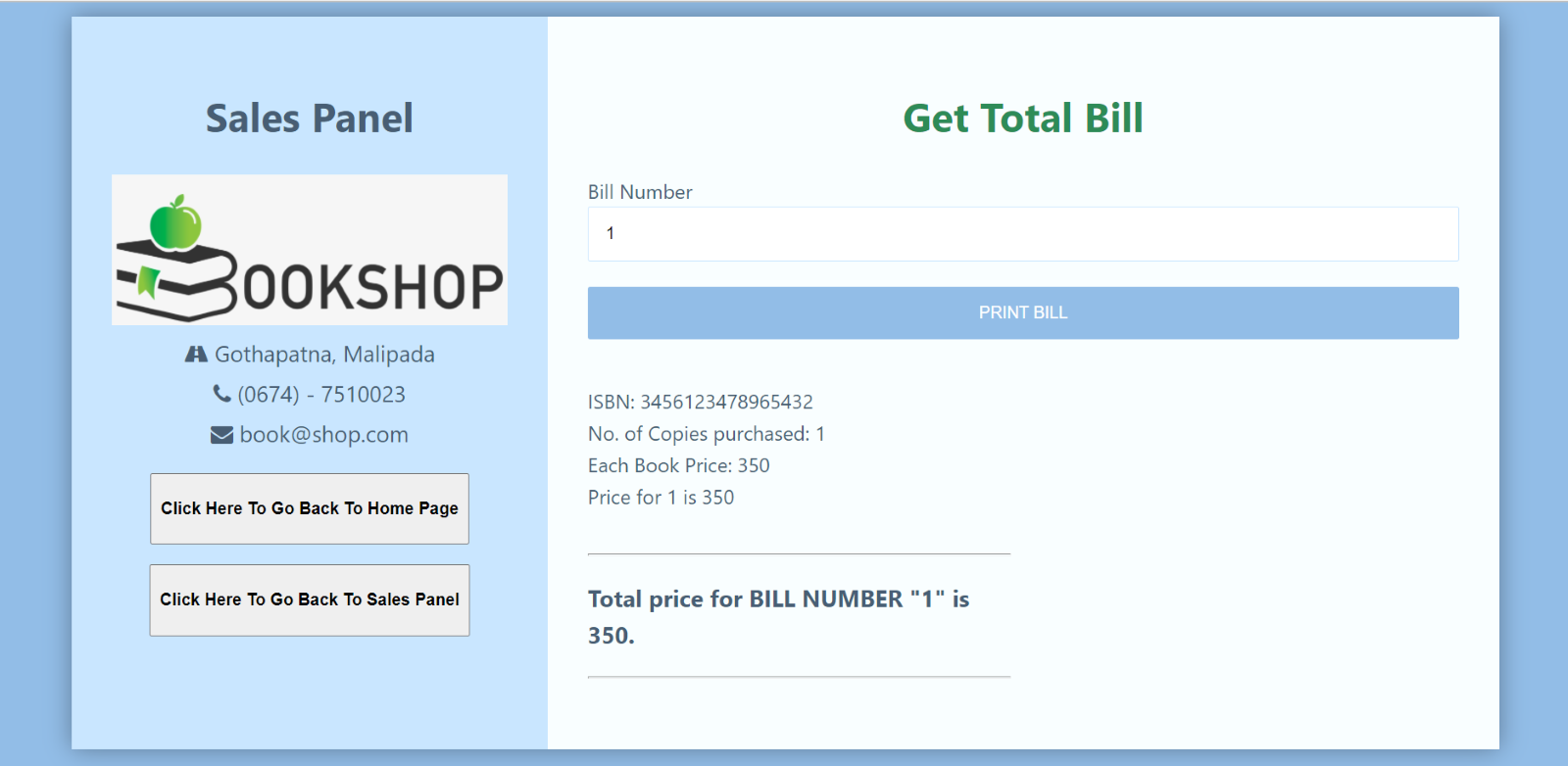


10. Printing of Bill: -

10.1 Input the bill number you want to print: -

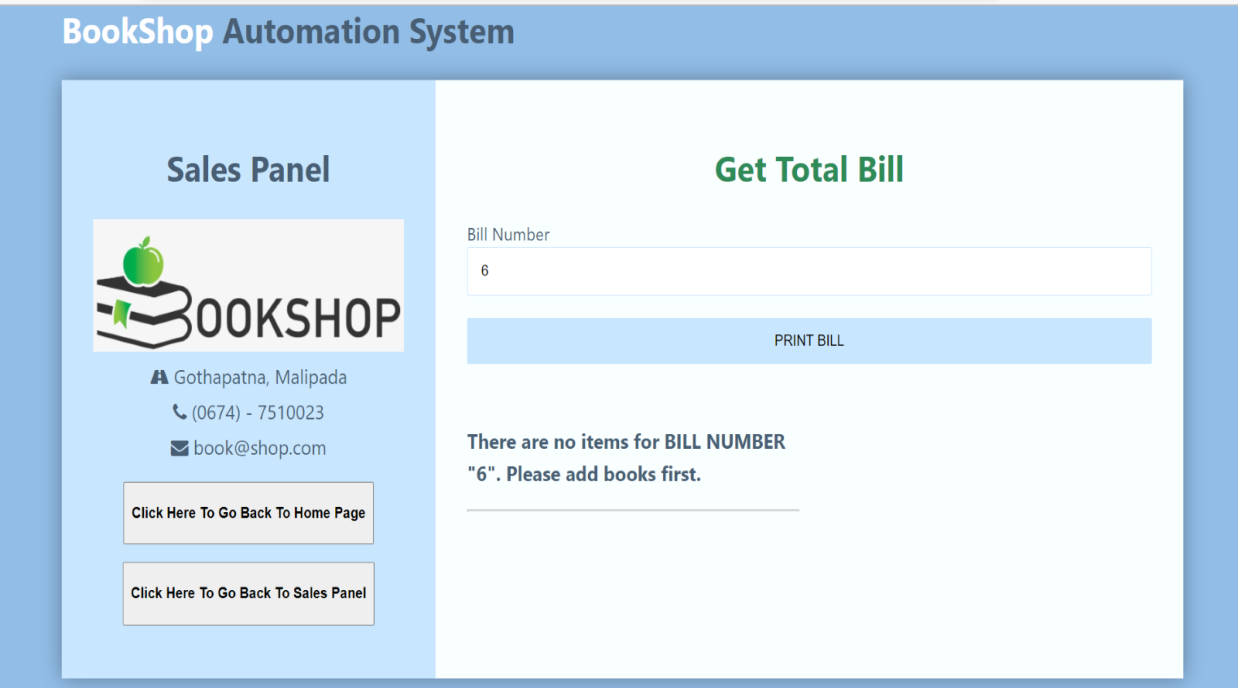


Output- bill



10.2 error - if input bill number doesn’t exist: -

Expected output – error message appears



***CONCLUSION:-***

This Bookshop Automation System is an attempt to overcome the present inefficient and time consuming process of locating, reserving and purchasing quality reading materials available in the store. Currently, clients have to go through a time consuming process to perform aforementioned tasks which cause waste of labor and firms resources. Through our automated book store solution, we provide an easy way of searching, requisitioning and purchasing of books.

All the newly coined processes will address time consuming, ineffective and inefficient areas of the existing system which has being wasting a lot of firms resources such as, labor, electricity, equipment, products and services, while discouraging customers to make purchases and repelling clients from the book store.

Customer satisfaction plays the most vital role in any form of product and service rendering store as the existence of any firm solely depends on its customer-base. Therefore, every system should facilitate the customer satisfaction up to a certain extent which is feasible from the company perspective.

The aforementioned facts ensure customer satisfaction to a greater extent benefiting the store in:

* Retaining current customers
* Tempting current customers to attract their friends to the store
* Attracting new customers
* Identifying profitable customers Identifying different categories of customers
* Making necessary alterations and plans to address broader range of customers
* Identifying key areas of the inventory which need to be maintained at a healthy stock limit
* More effective management decisions and development of new strategies to increase profit

These particulars will make sure the broadening the customer base of the store which will have good impact on the sales and revenue of the store.

Employee satisfaction also plays an influential role in healthy revenue levels of a firm. Due to the proposed system, employees will have to handle minimum amount of workload than that of the existing system which will help the employees to provide optimal service to the firm while maintaining healthy physical and mental levels.

Even though these advantages prevail, due to lack of IT literacy and fluency of clients and lack of distribution of internet facility will have a negative impact and it will take some time to cover up the capital investment made on implementing the new system. Since the technical facilities are expanding in great heaps, the proposed system will facilitate enhancing productivity immensely.

***REFERENCE AND BIBLIOGRAPHY*:**

* [*https://www.javatpoint.com/software-engineering-tutorial*](https://www.javatpoint.com/software-engineering-tutorial)
* [*https://www.tutorialspoint.com/software\_engineering/index.htm*](https://www.tutorialspoint.com/software_engineering/index.htm)
* [*https://en.wikipedia.org/wiki/Software\_engineering*](https://en.wikipedia.org/wiki/Software_engineering)
* [*http://software.isixsigma.com/library/content/c040915b.asp*](http://software.isixsigma.com/library/content/c040915b.asp)
* *Australian Computer Society, 2003, ASC Code of Ethics. Retrieved March 15,2007,c from* [*http://www.acs.org.au.htm*](http://www.acs.org.au.htm)
* *Out Source 2 India n.d.: Why Do Software Projects Fail? Retrieved 22 March 2007 from* [*http://www.outsource2india.com/software/SoftwareProjectFailure.asp*](http://www.outsource2india.com/software/SoftwareProjectFailure.asp)