ONLINE BOOK STORE

PROJECT REPORT

Submitted in partial fulfillment of requirement for the award of the Degree

Bachelor of Computer Science

In the faculty of Computer Science of Bharathiar University, Coimbatore

Submitted by

H.JUFRIYA HASHLIN

(Reg.No.2022K0132)

Under the guidance of

Dr.A.FINNY BELWIN, MCA., M.Sc., PhD.

Guest lecturer, Department of Computer Science

Department of Computer Science

L.R.G GOVERNMENT ARTS COLLEGE FOR WOMEN

(Affiliated To Bharathiar University)

TIRUPUR-4

APRIL-2023

CERTIFICATE

CERTIFICATE

This is to certify that the project work entitled **“ONLINE BOOK STORE”** Submitted to Bharathiar University in partial fulfilled of the requirement for the award of the Degree of Bachelor of computer science is a record of the original work done by **Ms.H.JUFRIYA HASHLIN (Reg.No.2022K0132)** Under my supervisor and that project work has not formed the basis for the any Degree /Diploma /Association /Fellowship or similar title to any candidate of any university.

Internal Guide	**Head of the Department**
Dr.A.FINNY BELWIN,MCA., M.Sc., Ph.D	**Dr.R.PARIMAL,MSc.,M.Phil.,Ph.D**

Viva-voce examination is held on ___________________L.R.G Government Arts College for Women, Tirupur-641604.

Internal Examiner	**External Examiner**

DECLARATION

DECLARATION

I hereby declare that the project work submitted to the **Department of the Computer Science, L.R.G. Government Arts College for Women, Tirupur** , affiliated to Bharathiar University, Coimbatore in the partial fulfillment of the required for the award of Bachelor of Computer Science is an original work done by me during the Sixth semester.

Place: **Signature of the Candidate**

Date:

(H.JUFRIYA HASHLIN)

(Reg.No:2022K0132)

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

As first and foremost I would like to external my thankfulness to the almighty for blessing the work of my hands, I am grateful to my parents for continued encouragement that they had given to me.

I would like to external my profound gratitude and sincere thanks to **Dr.M.R.YEZHILI, MA.,M.Phil.,Ph.D** Principal, L.R.G Government Arts College For Women , for the encouragement rendered to me during this project.

It's my privilege to thank **DR.R.PARIMALA, MSc.,M.Phil.,Ph.D** Incharge Head of the department of computer science for her valuable guidance and support throughout the project development work.

I express my deep sense of gratitude and sincere thanks to my guide **Dr.A.FINNY BELWIN, MCA.,M.Sc.,Ph.D** Guest Lecturer, Department of computer science, for providing all sorts of facilities to complete my project work successfully.

I also extend my sincere thanks to all the other faculty member of the department and technical assistance for their co-operation and valuable guidance.

I thank our college library for providing me with many informative books that help me to enrich my knowledge to bring out the project successfully.

SYNOPSIS

SYNOPSIS

The main goal of this project is that we can purchase and read books in online, we can use this project anywhere whether it's an download or purchase. When we need to reading books in online we just search it on Google and purchase it.

But this software actually working on download the books as free, A administrator login the software and upload the book details and upload the book as well, then user create an user account and login and download book details whether they need, it's an actually user purpose.

The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system.

Thus by this all it proves it is user-friendly. Online Book Store, as described above, can lead to error free, secure, reliable and fast management system.

It can assist the user to concentrate on their other activities rather to concentrate on the record keeping. Thus it will help organization in better utilization of resources.

It aims at improving the efficiency in the Issue of books or magazines and reduces the complexities involved in it to the maximum possible extent.

If the entire process of 'Issue of Books or Magazines' is done in a manual manner then it would take several months for the books or magazines to reach the applicant.

CONTENTS

CONTENT

PAGE.NO

1. INTRODUCTION

1.1 ABSTRACT

The main goal of this project is that we can purchase and read a books in online, we can use this project anywhere whether it’s an download or purchase. When we need to reading books in online we just search it on Google and purchase it. But this software actually working on download the books as free, A administrator login the software and upload the book details and upload the book as well, hen user create an user account and login and download book details whether they need, it’s an actually user purpose. The main objective of the project is to create an online book store that allows users to search and purchase a book online based on title, author and subject. The selected books are displayed in a tabular format and the user can download their books online through online payment. Using this Website the user can purchase a book online instead of going out to a book store and wasting time.

1.2 SYSTEM SPECIFICATION

System Requirements Specification also known as Software Requirements Specification, is a document or set of documentation that describes the features and behavior of a software application

WINDOWS OS

Windows is a graphical operating system developed by Microsoft. It allows users to view and store files, run the software, play games, watch videos, and provides a way to connect to the internet. It was released for both home computing and professional works.

Microsoft introduced the first version as 1.0

It was released for both home computing and professional functions of Windows on 10 November 1983. Later, it was released on many versions of Windows as well as the current version, Windows 10.

In 1993, the first business-oriented version of Windows was released, which is known as Windows NT 3.1. Then it introduced the next versions, Windows 3.5, 4/0, and Windows 2000. When the XP Windows was released by Microsoft in 2001, the company designed its various versions for a personal and business environment. It was designed based on standard x86 hardware, like Intel and AMD processor. Accordingly, it can run on different brands of hardware, such as HP, Dell, and Sony computers, including home-built PCs.
Play Video

Editions of Windows
Microsoft has produced several editions of Windows, starting with Windows
XP. These versions have the same core operating system, but some versions included advance features with an additional cost. There are two most common editions of Windows:

- Windows Home
- Windows Professional

Windows Home is basic edition of Windows. It offers all the fundamental functions of

Windows, such as browsing the web, connecting to the Internet, playing video games, using office software, watching videos. Furthermore, it is less expensive and comes pre-installed with many new computers.

JAVA

Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.

The Java syntax is similar to C++, but is strictly an object-oriented programming language. For example, most Java programs contain classes, which are used to define objects, and methods, which are assigned to individual classes. Java is also known for being stricter than C++, meaning variables and functions must be explicitly defined. This means Java source code may produce errors or "exceptions" more easily than other languages, but it also limits other types of errors that may be caused by undefined variables or unassigned types.

Unlike Windows executable (.EXE files) or Macintosh applications (.APP files), Java programs are not run directly by the operating system. Instead, Java programs are interpreted by the Java Virtual Machine, or JVM, which runs on multiple platforms. This means all Java programs are multiplatform and can run on different platforms, including Macintosh, Windows, and Unix computers. However, the JVM must be installed for Java applications or applets to run at all. Fortunately, the JVM is included as part of the Java Runtime Environment (JRE),

ORACLE

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information. A database server is the key to solving the problems of information management. In general, a server reliably manages a large amount of data in a multiuser environment so that many users can concurrently access the same data. All this is accomplished while delivering high performance. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.

Oracle Database is the first database designed for enterprise grid computing, the most flexible and cost effective way to manage information and applications. Enterprise grid computing creates large pools of industry-standard, modular storage and servers. With this architecture, each new system can be rapidly provisioned from the pool of components. There is no need for peak workloads, because capacity can be easily added or reallocated from the resource pools as needed.

The database has logical structures and physical structures. Because the physical and logical structures are separate, the physical storage of data can be managed without affecting the access to logical storage structures.

The Oracle grid architecture pools large numbers of servers, storage, and networks into a flexible, on-demand computing resource for enterprise computing needs. The grid computing infrastructure continually analyzes demand for resources and adjusts supply accordingly.

For example, you could run different applications on a grid of several linked database servers. When reports are due at the end of the month, the database administrator could automatically provision more servers to that application to handle the increased demand.

1.2.1 HARDWARE CONFIGURATION

➢ Processor	: Intel Core i3
➢ RAM	: 4 GB RAM
➢ Hard Disk Drive	: 500 GB
➢ Keyboard	: Multimedia Keyboard
➢ Mouse	: Optical

1.2.2 SOFTWARE SPECIFICATION

➢ Operating System	: Windows 10
➢ Front End	: JAVA
➢ Back End	: ORACLE

2. SYSTEM STUDY

2.1 EXISTING SYSTEM

Previously when user need to reading a booking normally go to refer some website and update their knowledge. This is an not a standard way for reading a book reader. It doesn't give a perfect searching algorithm. Usually the book reader searching a book then the referring site giving a lot of option to the user. It may the can get confusing when searching a book. It took take too much time to finding a single book. So this software may help to resolving an issue.

2.1.1 DRAWBACKS

- Take too much time for searching a book
- It doesn't provide pay books
- It may not giving an specific author book details
- It's not an user friendly

2.2 PROPOSED SYSTEM

In this system can be fulfilled the previous system disadvantages. In this system user can easily identify they searched the book by author wise as well. User go to the book store web page and searching a books. Before that an administrator save the book details and upload the book. It may use an lot of benefits.

2.2.1 FEATURES

- User can easily identify the book
- Download the books
- Find author profile
- Very user friendly

3. SYSTEM DESIGN AND DEVELOPMENT

3.1 FILE DESIGN

The selection of the file system design approach is done according to the needs of the developers what are the needed requirements and specifications for the new design. It allowed us to identify where our proposal fitted in with relation to current and past file system development. Our experience with file system development is limited so the research served to identify the different techniques that can be used. The variety of file systems encountered show what an active area of research file system development is. The file systems may be from one of the two fundamental categories. In one category, the file system is developed in user space and runs as a user process. Another file system may be developed in the kernel space and runs as a privileged process. Another one is the mixed approach in which we can take the advantages of both aforesaid approaches. Each development option has its own pros and cons. In this article, these design approaches are discussed.

A file system is the data structure designed to support the abstraction of the data blocks as an archive and collection of files. This data structure is unique because it is stored on secondary storage (usually the disk), which is a very slow device.

The file system structure is the most basic level of organization in an operating system. Almost all of the ways an operating system interacts with its users, applications, and security model are dependent upon the way it organizes files on storage devices.

File Design Information systems in business are file and database oriented. Data are accumulated into files that are processed or maintained by the system. The systems analyst is responsible for designing files, determining their contents and selecting a method for organizing the data.

The most important purpose of a file system is to manage user data. This includes storing, retrieving and updating data. Some file systems accept data for storage as a stream of bytes which are collected and stored in a manner efficient for the media.

3.2 INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:'

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user
- will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

3.3 OUTPUT DESIGN

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

External Outputs

Manufacturers create and design external outputs for printers. External outputs enable the system to leave the trigger actions on the part of their recipients or confirm actions to their recipients.

Some of the external outputs are designed as turnaround outputs, which are implemented as a form and re-enter the system as an input.

Internal outputs

Internal outputs are present inside the system, and used by end-users and managers. They support the management in decision making and reporting.

Output Integrity Controls

Output integrity controls include routing codes to identify the receiving system, and verification messages to confirm successful receipt of messages that are handled by network protocol.

Printed or screen-format reports should include a date/time for report printing and the data. Multipage reports contain report title or description, and pagination. Pre-printed forms usually include a version number and effective date.

3.4 DATABASE DESIGN

Today's businesses depend on their databases to provide information essential for day-to-day operations, especially in case of electronic commerce businesses who has a definite advantage with up-to-date database access. Good design forms the foundation of any database, and experienced hands are required in the automation process to design for optimum and stable performance.

Software Solutions have been constantly working on these platforms and have attained a level of expertise. We apply proven methodologies to design, develop, integrate and implement database systems to attain its optimum level of performance and maximize security to meet the client's business model.

Business needs addressed:

- Determine the basic objects about which the information is stored
- Determine the relationships between these groups of information and the objects
- Effectively manage data and create intelligent information
- Remote database administration or on site administrative support
- Database creation, management, and maintenance
- Information retrieval efficiency, remove data redundancy and ensure data security

The most important consideration in designing the database is how the information will be used. The main objective of designing a database is Data Integration, Data Integrity and Data Independence.

Data Integration

In a database, information from several files is coordinated, accessed and operated upon as through it is in a single file. Logically, the information is centralized, physically; the data may be located on different devices, connected through data communication facilities.

Data Integrity

Data integrity means storing all data in one place only and how each application accesses it. This approach results in more consistent information, one update being sufficient to achieve record status for all applications. This lead stoles data redundancy that is data items need not be duplicated.

Data Independence

Data independence is the insulation of application programs from changing aspects of physical data organization. This objective seeks to allow changes in the content and organization of physical data without reprogramming of application and allow modifications to application programs without reorganizing the physical data.

3.5 SYSTEM DEVELOPMENT

Systems development is the process of defining, designing, testing, and implementing a new software application or program. It could include the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software.

Systems development life cycle phases include planning, system analysis, system design, development, implementation, integration and testing, and operations and maintenance.

3.5.1 DESCRIPTION OF MOODULES

1. User Registration
2. Book Registration and view module
3. Purchase Book
4. Register Author Details

MODULES DESCRIPTION

1. User Registration

This module is used before the user has to be login. This will be collect the user information like name, mobile number, email id etc.. The admin can view the all users profile who are all interest to register the account.

2.Book Registration and view module

A book registration module can be main module of this project, an admin can enter the book details and view the book details. If the admin create the book inside the book registration module then the user can find the book and download it.

3.Purchase Book

Some books are need to should buy, so when the user need to buy a book they can purchase it. Admin also can easily identify who are all purchase the book. After the purchasing only they can download it..

4. Register Author Details

Before creating a book author details are most important by an user, so the admin has to register the author details then view the author details both user and admin.

4. TESTING AND IMPLEMENTATION

TESTING METHODOLOGIES

System testing is state of implementation, which is aimed at ensuring that the system works accurately and efficiently as expect before live operation commences. It certifies that the whole set of programs hang together.

System testing requires a test plan that consists of several key activities and step for run program, string, system and user acceptance testing. The implementation of newly designed package is important in adopting a successful new system

Testing is the important stage in software development. the system test in implementation stage in software development process. The system testing implementation should be confirmation that all is correct and an opportunity to show the users that the system works as expected. It accounts the largest percentage of technical effort in the software development process.

Testing phase in the development cycle validates the code against the functional specification testing is vital to achievement of the system goals. The objective of the testing is to discover errors to fulfill this objective a series of test step unit, integration. Validation and system tests were planned and executed the test steps are:

SYSTEM TESTING

Testing is an integral part of any system development life cycle. Insufficient and untested applications may tend to crash and the result is loss of economic and manpower investment besides user's dissatisfaction and downfall of reputation. Software testing can be looked upon as one among many processes, an organization performs, and that provides the lost opportunity to correct any flaws in the developed system. Software testing includes selecting test data that have more probability of giving errors.

The first step in system testing is to develop a plan that tests all aspects of the system. Completeness, correctness, reliability and maintainability of the software are to be tested for the best quality assurance that the system meets the specification and requirements for its intended use and performance. System testing is the most useful practical process of executing a program with the implicit intention of finding errors that make the program fails.

System testing is done in three phases.

- Unit Testing
- Integration Testing
- Validation Testing

UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software the module. Using the detailed design and the process specification testing is done to registration by the user with in the boundary of the Login module. The login form receives the username and password details and validates the value with the database. If valid, the home page is displayed.

INTEGRATION TESTING

Integration Testing is the process of this activity can be considered as testing the design and hence module interaction. The primary objective of integration testing is to discover the interfaces between the components. Login for registration form are integrated and tested together. If the user is newly registered, the received details will be stored in the registration table. While logging in, the application will check for valid user name and password in the registration table and if valid the user is prompted for submitting complaints.

Data can be lost across an interface, one module can have adverse effect on another sub function when combined it may not produce the desired major functions. Integration testing is a systematic testing for constructing test to uncover errors associated within an interface.

The objectives taken from unit tested modules and a program structure is built for integrated testing. All the modules are combined and the test is made.

A correction made in this testing is difficult because the vast expenses of the entire program complicated the isolation of causes. In this integration testing step, all the errors are corrected for next testing process.

VALIDATION TESTING

Validation are independent procedures that are used together for checking that a product, service, or system meets requirements and specifications and that it fulfills its in purpose the actual result from the expected result for the complaint process.

Select the complaint category of the complaint by user. The input given to various forms fields are validated effectively. Each module is tested independently. It is tested that the complaint module fields receive the correct input for the necessary details such as complaint category, complaint id, reference name, complaint description, and email for further process.

After the completion of the integrated testing, software is completely assembled as a package; interfacing error has been uncovered and corrected and a final series of software test validation begins.

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software function in a manner that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists.

OUTPUT TESTING

The next process of validation testing, is output testing of the proposed system, since no system could be successful if it does not produce the required output in the specified format. Asking the user about the format required, list the output to be generated or displayed by the system under considerations.

Output testing is a different test whose primary purpose is to fully exercise the computer based system although each test has a different purpose all the work should verify that all system elements have been properly integrated and perform allocated functions.

The output format on the screen is found to be corrected as the format was designed in the system design phase according to the user needs for the hard copy also; the output testing has not resulted in any correction in the system.

SYSTEM IMPLEMENTATION

When the initial design was done for the system, the client was consulted for the acceptance of the design so that further proceedings of the system development can be carried on. After the development of the system a demonstration was given to them about the working of the system. The aim of the system illustration was to identify any malfunction of the system.

After the management of the system was approved the system implemented in the concern, initially the system was run parallel with existing manual system. The system has been tested with live data and has proved to be error free and user friendly.

Implementation is the process of converting a new or revised system design into an operational one when the initial design was done by the system; a demonstration was given to the end user about the working system.

This process is uses to verify and identify any logical mess working of the system by feeding various combinations of test data. After the approval of the system by both end user and management the system was implemented.

System implementation is made up of many activities. The six major activities are as follows.

CODING

Coding is the process of whereby the physical design specifications created by the analysis team turned into working computer code by the programming team. A design code may be a tool which helps ensure that the aspiration for quality and quantity for customers and their requirements, particularly for large scale projects, sought by the water agency Design pattern are documented tried and tested solutions for recurring problems in a given context. So basically you have a problem context and the proposed solution for the same.

INSTALLATION

Installation is the process during which the current system is replaced by the new system. This includes conversion of existing data, software, and documentation and work procedures to those consistent with the new system.

DOCUMENTATION

Documentation is descriptive information that describes the use and operation of the system. The user guide is provided to the end user as the student and administrator. The documentation part contains the details as follows,

User requirement and water agency details administration has been made online. Any customer can request their water requirement details through online and also use of documentation, they can view the purpose of each purpose, The admin could verify the authentication of the users, users requirements and need to take delivery process, thus the documentation is made of full view of project thus it gives the guideline to study the project and how to execute also.

USER TRAINING AND SUPPORT

The software is installed at the deployment environment, the developer will give training to the end user of the regional transport officer and police admin officer in that software. The goal of an end user training program is to produce a motivated user who has the skills needed to apply what has been to apply what has been learned to perform the job related task. The following are the instruction which is specified the handling and un-handling events in the application,

- The authenticated user of admin and office workers only login in the application with authorized username and password.
- Don't make user waste their time to come straight to the water agency or make a phone call.
- It can easily track through online by the user.
- Very user friendliness software

IMPLEMENTATION PROCEDURES

Implementation includes all the activities that take place to convert the old system to the new one. Proper implementation is essential to provide a reliable system to meet the organization requirements. Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

IMPLEMENTATION PROCEDURES

PILOT RUNNING

Processing the current data by only one user at a time called the pilot running process. When one user is accessing the data at one system, the system is sets to be engaged and connected in network. This process is useful only in system where more than one user is restricted.

PARALLEL RUNNING:

Processing the current data by more than one user at a time simultaneously is said to be parallel running process. This same system can be viewed and accessed by more than one user at the time. Hence the implementation method used in the system is a pilot type of implementation.

Implementation is the stage in the project where the theoretical design is turned into a working system. The most crucial stage is achieving a successful new system & giving the user confidence in that the new system will work efficiently & effectively in the implementation state.

The stage consists of,

- Testing the developed program with sample data.
- Detection's and correction of error.
- Creating whether the system meets user requirements.
- Making necessary changes as desired by the user.
- Training user personnel.

USER TRAINING

User Training is designed to prepare the user for testing &consenting the system. .

- User Manual.
- Help Screens.
- Training Demonstration.

USER MANUAL

The summary of important functions about the system and software can be provided as a document to the user.

HELP SCREENS

This features now available in every software package, especially when it is used with a menu. The user selects the "Help" option from the menu. The system accesses the necessary description or information for user reference.

TRAINING DEMONSTRATION:

Another User Training element is a Training Demonstration. Live demonstrations with personal contact are extremely effective for Training Users.

SYSTEM MAINTENANCE

Maintenance is actually the implementation of the review plan. As important as it is, many programmers and analysts are to perform or identify themselves with the maintenance effort. There are psychological, personality and professional reasons for this. Analysts and programmers spend far more time maintaining programs than they do writing them. Maintenance accounts for 50-80 percent of total system development

Maintenance is expensive. One way to reduce the maintenance costs are through maintenance management and software modification audits**.**

- Maintenance is not as rewarding as exciting as developing systems. It is perceived as requiring neither skill not experience.
- Users are not fully cognizant of the maintenance problem or its high cost.
- Few tools and techniques are available for maintenance.
- A good test plan is lacking.
- Standards, procedures, and guidelines are poorly defined and enforced.
- Programs are often maintained without care for structure and documentation.
- There are minimal standards for maintenance.
- Programmers expect that they will not be in their current commitment by time their programs go into the maintenance cycle.

Corrective Maintenance

It means repairing, processing or performance failure or making changes because of previously uncovered problems or false assumptions. Task performed to identify, isolate, and rectify a fault so that the failed equipment, machine, or system can be restored to an operational condition within the tolerances or limits established for in-service operations.

Corrective maintenance can be subdivided into "immediate corrective maintenance" (in which work starts immediately after a failure) and "deferred corrective maintenance" (in which work is delayed in conformance to a given set of maintenance rules).

Perfective Maintenance

It means changes made to a system to add new features or to improve performance. Preventive maintenance is predetermined work performed to a schedule with the aim of preventing the wear and tear or sudden failure of equipment components. process or control equipment failure can have adverse results in both human and economic terms. In addition to down time and the costs involved to repair and/or replace equipment parts or components, there is the risk of injury to operators, and of acute exposures to chemical and/or physical agents.

Time-based or run-based Periodically inspecting, servicing, cleaning, or replacing parts to prevent sudden failure .On-line monitoring of equipment in order to use important/expensive parts to the limit of their serviceable life. Preventive maintenance involves changes made to a system to reduce the chance of future system failure.

An example of preventive maintenance might be to increase the number of records that a system can process far beyond what is currently needed or to generalize how a system sends report information to a printer so that so that the system can adapt to changes in printer technology.

Preventive Maintenance

Changes made to a system to avoid possible future problems Perfective maintenance involves making enhancements to improve processing performance, interface usability, or to add desired, but not necessarily required, system features. The objective of perfective maintenance is to improve response time, system efficiency, reliability, or maintainability.

During system operation, changes in user activity or data pattern can cause a decline in efficiency, and perfective maintenance might be needed to restore performance. Usually, the perfective maintenance work is initiated by the IT department, while the corrective and adaptive maintenance work is normally requested by users.

5. CONCLUSION

Bookstores play a critical role in promoting literacy, education, and intellectual discourse in communities. They are essential platforms for authors, publishers, and readers to connect and exchange ideas. Bookstores offer several benefits, including access to a wide range of books, knowledgeable staff, and a vibrant community of book lovers.

Bookstores provide access to a wide range of books, including bestsellers, classics, and niche titles. This helps to promote a culture of reading and learning, enabling individuals to explore diverse perspectives and ideas.

Moreover, bookstores provide knowledgeable staff who can offer recommendations, answer questions, and provide insights into different genres and authors. This enhances the overall customer experience, leading to improved customer loyalty and satisfaction.

Bookstores also provide a vibrant community of book lovers, offering opportunities for author events, book clubs, and other literary gatherings. This helps to promote intellectual discourse, exchange of ideas, and a sense of community among book lovers.

In conclusion, bookstores are essential platforms for promoting literacy, education, and intellectual discourse in communities. They offer several benefits, including access to a wide range of books, knowledgeable staff, and a vibrant community of book lovers. By leveraging these benefits, bookstores can play a critical role in fostering a culture of reading and learning, enhancing the overall wellbeing of individuals and communities.

BIBLIOGRAPHY

Book Reference:

1. Bloch, Joshua. Effective Java: Programming Language Guide. Addison-Wesley, 2017.
2. Eckel, Bruce. Thinking in Java. Prentice Hall, 2006.
3. Freeman, Eric, and Elisabeth Robson. Head First Java. O'Reilly Media, 2005.
4. Horstman, Cay S. Core Java Volume I—Fundamentals. Prentice Hall, 2018.
5. Lea, Doug. Concurrent Programming in Java: Design Principles and Patterns. Addison-Wesley, 2000.

Website:

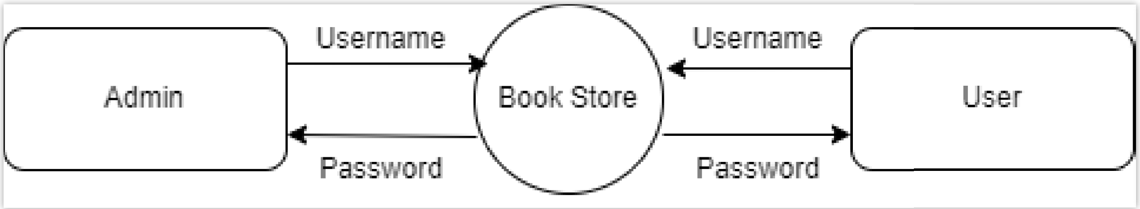
1. Baeldung. "Java Tutorials and Articles." Baeldung, 2023, https://www.baeldung.com/java-tutorials.
2. GeeksforGeeks. "Java Programming Language." GeeksforGeeks, 2023, https://www.geeksforgeeks.org/java/.
3. Stack Overflow. "Questions tagged [java]." Stack Overflow, https://stackoverflow.com/questions/tagged/java.
4. Tutorials Point. "Java Tutorial." Tutorials Point, 2023, https://www.tutorialspoint.com/java/index.htm.
5. Vogella. "Java Tutorials." Vogella, 2023, https://www.vogella.com/tutorials/java.html

APPENDICES

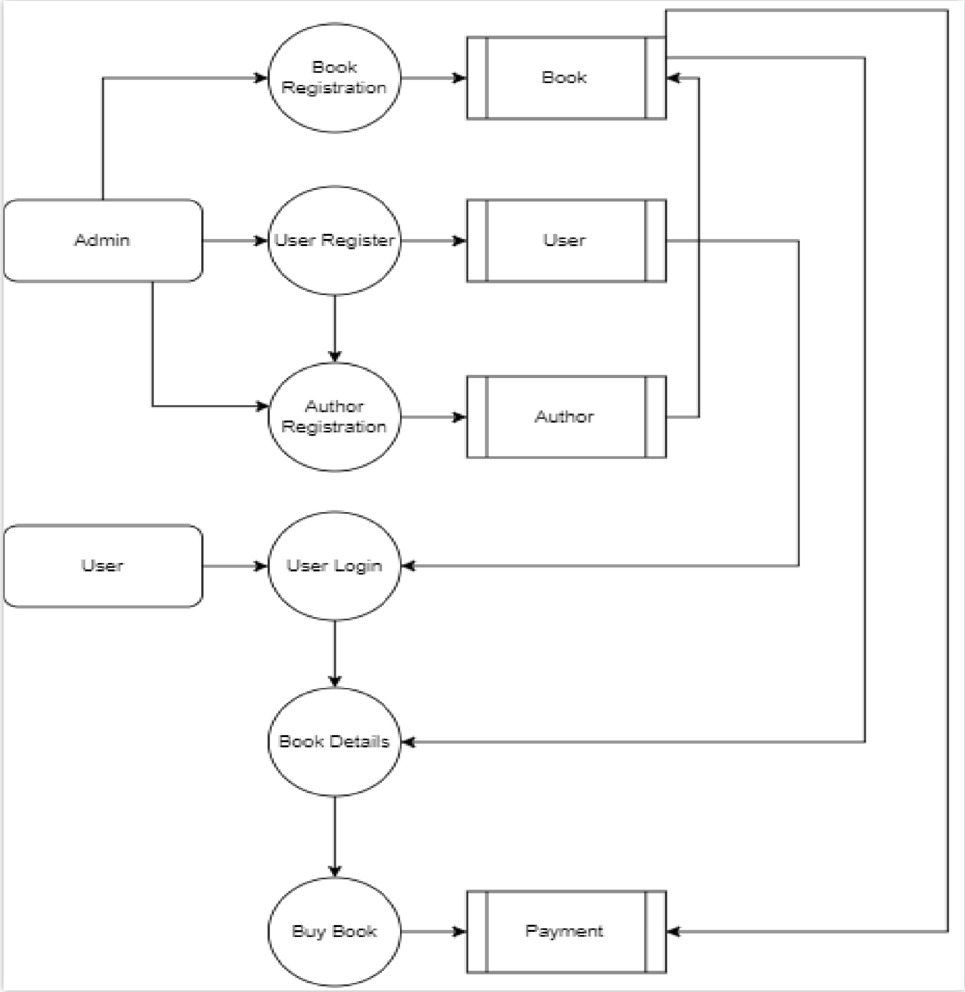
A. DATA FLOW DIAGRAM

A data-flow diagram (DFD)is a way of representing a flow of a data of a process or system. The DFD also provides information about the outputs and inputs of each entity and process itself. A data-flow diagram is a part of structured-analysis modeling tools.

LEVEL 0:



LEVEL 1:


Book Registration
Book
Admin
User Register
User
Author Registration
Author
User
User Login
Book Details
Buy Book
Payment

B. TABLE STRUCTURE

The table needed for each module was designed and the specification of each and every column was given based on the records and details collected during record specification of the system study.

TABLE NAME: ADMIN

FIELD	DATA TYPE	SIZE	CONSTRAINT
Adminid	INT	10	Primary key
Username	Varchar	20	Not null
Password	Varchar	20	Not null

TABLE NAME: AUTHOR

FIELD	DATA TYPE	SIZE	CONSTRAINT
Author id	Int	10	Primary key
Authorname	Varchar	30	Not null
Description	Varchar	30	Not null
Qualification	Varchar	30	Not null

TABLE NAME: BOOK

FIELD	DATA TYPE	SIZE	CONSTRAINT
Book id	Int	10	Primary key
Author id	Int	10	Foreign key
Published	Date	10	Not null
Book name	Varchar	30	Not null

Category	Varchar	30	Not null
Subject	Varchar	30	Not null
Price	Int	10	Not null

TABLE NAME: PURCHASE

FIELD	DATA TYPE	SIZE	CONSTRAINT
Purchase id	Int	10	Primary key
Bookid	Int	10	Foreign key
User Id	Int	10	Not null
purchase Id	Int	10	Not null

TABLE NAME: USER

FIELD	DATA TYPE	SIZE	CONSTRAINT
User id	Int	10	Primary key
Firstname	Varchar	30	Notnull
Lastname	Varchar	30	Not null
Mobile	Varchar	10	Not null
Email	Varchar	30	Not null
Username	Varchar	30	Not null
Password	Varchar	30	Not null

C.SAMPLE CODING

```
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

@RestController
public class LoginController {

    @GetMapping("/login")
    public LoginResponse login(@RequestParam String username, @RequestParam String password)
{
        try {
            // Connect to the database
            Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydatabase",
"username", "password");

            // Check if the user is an admin
PreparedStatementstmt = conn.prepareStatement("SELECT id FROM admin WHERE username = ?
AND password = ?");
stmt.setString(1, username);
stmt.setString(2, password);
ResultSetrs = stmt.executeQuery();
            if (rs.next()) {
int id = rs.getInt("id");
                return new LoginResponse(id, 1, "Success");
            }

            // Check if the user is a regular user
stmt = conn.prepareStatement("SELECT id FROM user WHERE username = ? AND password = ?");
stmt.setString(1, username);
stmt.setString(2, password);
rs = stmt.executeQuery();
```

```
            if (rs.next()) {
int id = rs.getInt("id");
                return new LoginResponse(id, 2, "Success");
            }

            // Invalid username or password
            return new LoginResponse(0, 0, "Invalid username or password");
        } catch (SQLException e) {
e.printStackTrace();
            return new LoginResponse(0, 0, "Database error");
        }
    }

    private static class LoginResponse {
        private final int id;
        private final int status;
        private final String message;

        public LoginResponse(int id, int status, String message) {
this.id = id;
this.status = status;
this.message = message;
        }

        // getters and setters
    }
}
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import java.sql.*;

@RestController
public class LoginController {

    @GetMapping("/login")
    public String login(@RequestParam String username, @RequestParam String password) {

        String url = "jdbc:mysql://localhost:3306/mydatabase"; // replace with your database URL
        String dbUsername = "myuser"; // replace with your database username
    String dbPassword = "mypassword"; // replace with your database password

    try {
Class.forName("com.mysql.jdbc.Driver");
      Connection conn = DriverManager.getConnection(url, dbUsername, dbPassword);

      String query = "SELECT id FROM admin WHERE username = ? AND password = ?";
PreparedStatementstmt = conn.prepareStatement(query);
stmt.setString(1, username);
stmt.setString(2, password);
ResultSetrs = stmt.executeQuery();

      if (rs.next()) {
int id = rs.getInt("id");
        return "{\"id\":" + id + ",\"status\":1,\"message\":\"Success\"}";
      } else {
        query = "SELECT id FROM user WHERE username = ? AND password = ?";
stmt = conn.prepareStatement(query);
stmt.setString(1, username);
stmt.setString(2, password);
rs = stmt.executeQuery();

        if (rs.next()) {
int id = rs.getInt("id");
          return "{\"id\":" + id + ",\"status\":2,\"message\":\"Success\"}";
        } else {
          return "{\"status\":0,\"message\":\"Invalid username or password\"}";
        }
      }

    } catch (Exception e) {
e.printStackTrace();
      return "{\"status\":0,\"message\":\"Error\"}";
    }
  }
}
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.RestController;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.jdbc.core.JdbcTemplate;

import java.util.ArrayList;
import java.util.List;
import java.util.Map;

@RestController
public class BookController {

    @Autowired
    private JdbcTemplatejdbcTemplate;

    @GetMapping("/books")
    public List<Book>getBooks(@RequestParam("userid") intuserId) {

        String sql = "select book.id, book.bookname, author.authorname, book.publish, book.category,
book.subject, book.price, book.ext, " +
                "case when payment.purchasestatus is null then 'Purchase' else 'Download' end as
isdownload " +
                "from book left join payment on(payment.bookid=book.id) left join author
on(author.id=book.authorid) where payment.userid=? " +
                "UNION " +
                "select book.id, book.bookname, author.authorname, book.publish, book.category,
book.subject, book.price, book.ext, " +
                "case when payment.purchasestatus is null then 'Purchase' else 'Download' end as
isdownload " +
                "from book left join payment on(payment.bookid=book.id) left join author
on(author.id=book.authorid) where payment.userid is null";

        List<Book> books = new ArrayList<>();
        List<Map<String, Object>> rows = jdbcTemplate.queryForList(sql, userId);

        for (Map<String, Object> row : rows) {
            Book book = new Book();
book.setId((Integer) row.get("id"));
book.setBookName((String) row.get("bookname"));
book.setAuthorName((String) row.get("authorname"));
book.setPublish((String) row.get("publish"));
book.setCategory((String) row.get("category"));
book.setSubject((String) row.get("subject"));
book.setPrice((Double) row.get("price"));
book.setExt((String) row.get("ext"));
book.setIsDownload((String) row.get("isdownload"));
books.add(book);
        }

        return books;
    }

}@RestController
public class BookController {
    @Autowired
    private BookRepositorybookRepository;

    @PostMapping("/books")
    public ResponseEntity<String>createBook(
            @RequestParam Long authorId,
            @RequestParam String bookName,
            @RequestParam String publish,
            @RequestParam String category,
            @RequestParam String subject,
            @RequestParam double price,
            @RequestParam("file") MultipartFile file) throws IOException {

        Book book = new Book();
book.setAuthorId(authorId);
book.setBookName(bookName);
book.setPublish(publish);
book.setCategory(category);
book.setSubject(subject);
book.setPrice(price);
        book.setExt(FilenameUtils.getExtension(file.getOriginalFilename()));

        book = bookRepository.save(book);

        Path path = Paths.get("./upload/" + book.getId() + "." + book.getExt());
Files.write(path, file.getBytes());

        return ResponseEntity.ok("Book created successfully");
    }
```

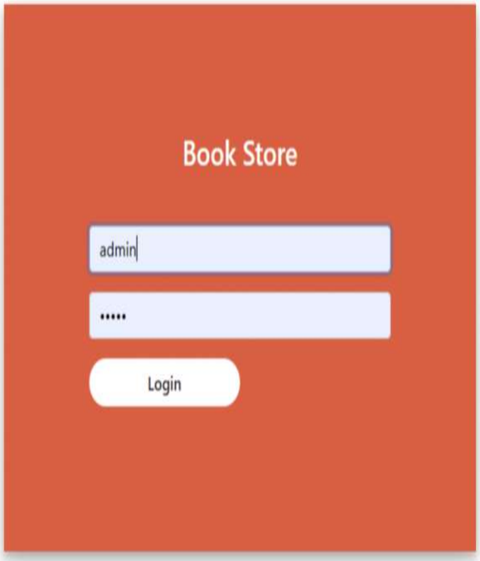
```
}

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;

import javax.transaction.Transactional;

@RestController
public class PaymentController {

    @Autowired
    private PaymentRepositorypaymentRepository;

    @PostMapping("/payment")
    @Transactional
    public ResponseEntity<String>createPayment(
            @RequestParam Long bookid,
            @RequestParam Long userid
    ) {
        Payment payment = new Payment();
payment.setBookId(bookid);
payment.setUserId(userid);
payment.setPurchaseStatus(true);

paymentRepository.save(payment);

        return new ResponseEntity<>("Payment successfully completed", HttpStatus.OK);
    }
}

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.web.bind.annotation.*;

import javax.transaction.Transactional;
```

```
@RestController
public class UserController {

    @Autowired
    private UserRepositoryuserRepository;

    @Autowired
    private PasswordEncoderpasswordEncoder;

    @PostMapping("/user")
    @Transactional
    public ResponseEntity<String>createUser(
            @RequestParam String firstname,
            @RequestParam String lastname,
            @RequestParam String mobile,
            @RequestParam String email,
            @RequestParam String username,
            @RequestParam String password
    ) {
        User user = new User();
user.setFirstName(firstname);
user.setLastName(lastname);
user.setMobile(mobile);
user.setEmail(email);
user.setUsername(username);
user.setPassword(passwordEncoder.encode(password));

userRepository.save(user);

        return new ResponseEntity<>("User created successfully", HttpStatus.OK);
    }
}

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.web.bind.annotation.*;

import javax.transaction.Transactional;
```

```
@RestController
public class UserController {

  @Autowired
  private UserRepositoryuserRepository;

  @Autowired
  private PasswordEncoderpasswordEncoder;

  @PostMapping("/user")
  @Transactional
  public ResponseEntity<String>createUser(
      @RequestParam String firstname,
      @RequestParam String lastname,
      @RequestParam String mobile,
      @RequestParam String email,
      @RequestParam String username,
      @RequestParam String password
  ) {
    User user = new User();
user.setFirstName(firstname);
user.setLastName(lastname);
user.setMobile(mobile);
user.setEmail(email);
user.setUsername(username);
user.setPassword(passwordEncoder.encode(password));

userRepository.save(user);

    return new ResponseEntity<>("User created successfully", HttpStatus.OK);
  }
}
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RestController;

import java.util.List;
import java.util.stream.Collectors;

@RestController
public class AuthorController {

    @Autowired
    private AuthorRepositoryauthorRepository;

    @GetMapping("/authors")
    public ResponseEntity<List<AuthorDto>>getAuthors() {
        List<AuthorDto> authors = authorRepository.findAll()
                .stream()
                .map(AuthorDto::fromEntity)
                .collect(Collectors.toList());

        return new ResponseEntity<>(authors, HttpStatus.OK);
    }

    public static class AuthorDto {
        private Long id;
        private String authorname;
        private String description;
        private String qualification;

        public Long getId() {
            return id;
        }

        public void setId(Long id) {
this.id = id;
        }

        public String getAuthorname() {
            return authorname;
        }

        public void setAuthorname(String authorname) {
this.authorname = authorname;
        }
        public String getDescription() {
            return description;
        }

        public void setDescription(String description) {
this.description = description;
        }

        public String getQualification() {
            return qualification;
        }

        public void setQualification(String qualification) {
this.qualification = qualification;
        }

        public static AuthorDtofromEntity(Author author) {
AuthorDtodto = new AuthorDto();
dto.setId(author.getId());
dto.setAuthorname(author.getAuthorname());
dto.setDescription(author.getDescription());
dto.setQualification(author.getQualification());
            return dto;
        }
    }
}
```

D. SAMPLE INPUT & OUTPUT DESIGN

LOGIN PAGE:

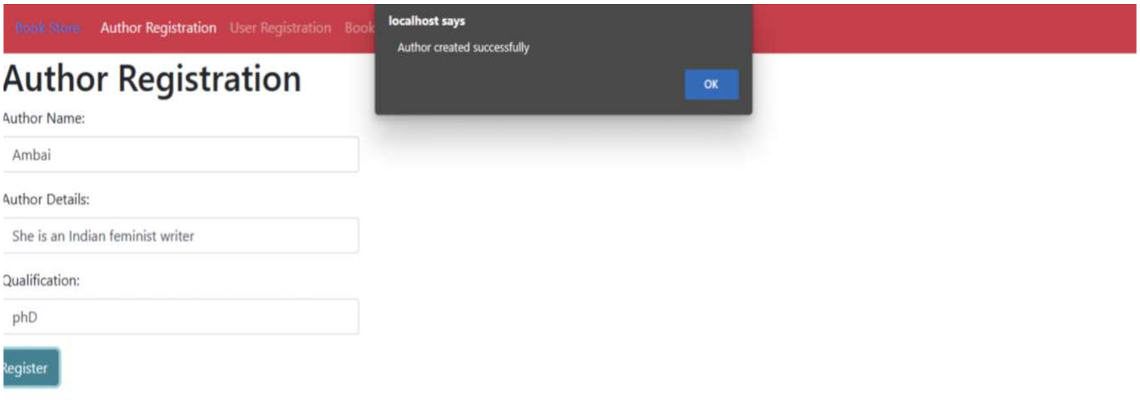


Book Store

admin

•••••

AUTHOR REGISTRATION:



localhost says

Author created successfully

Author Registration

uthor Name:

Ambai

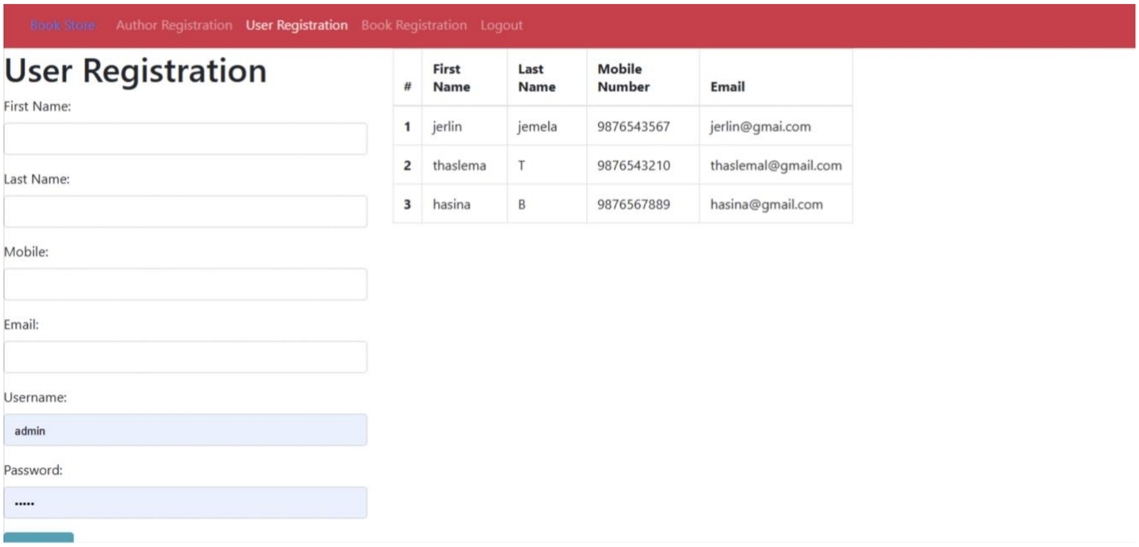
uthor Details:

She is an Indian feminist writer

Qualification:

phD

USER REGISTRATION:



User Registration

First Name:

Last Name:

Mobile:

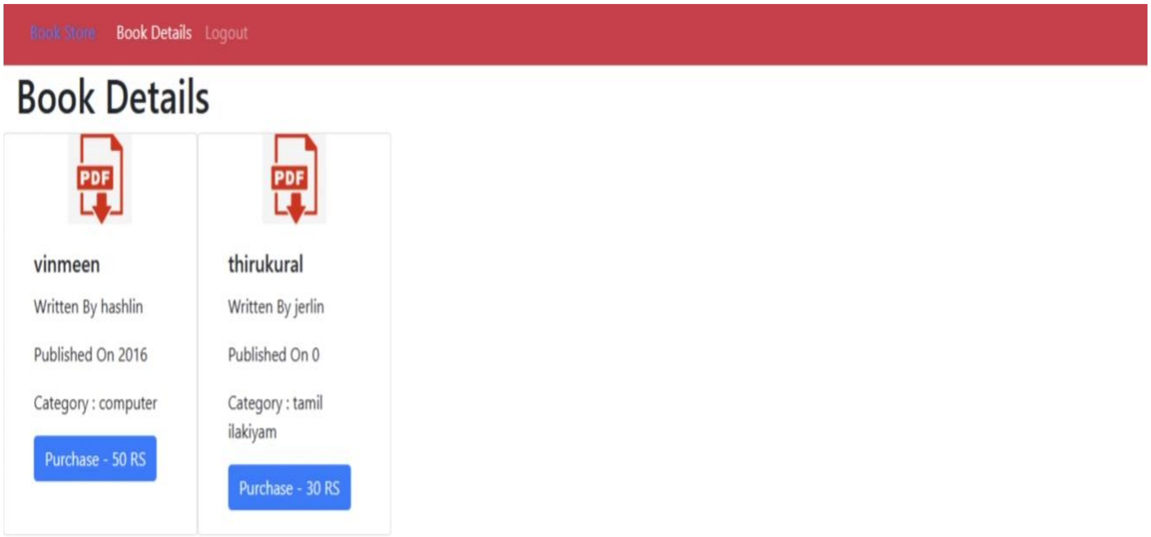
Email:

Username: admin

Password: •••••

#	First Name	Last Name	Mobile Number	Email
1	jerlin	jemela	9876543567	jerlin@gmai.com
2	thaslema	T	9876543210	thaslemal@gmail.com
3	hasina	B	9876567889	hasina@gmail.com

BOOK DETAILS:



Book Details

vinmeen

Written By hashlin

Published On 2016

Category : computer

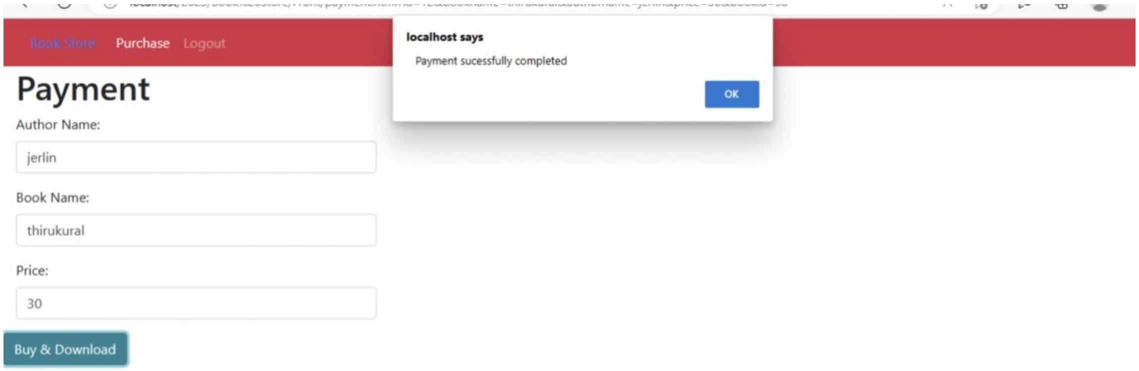
thirukural

Written By jerlin

Published On 0

Category : tamil ilakiyam

PAYMENT:



localhost says

Payment sucessfully completed

Payment

Author Name:

jerlin

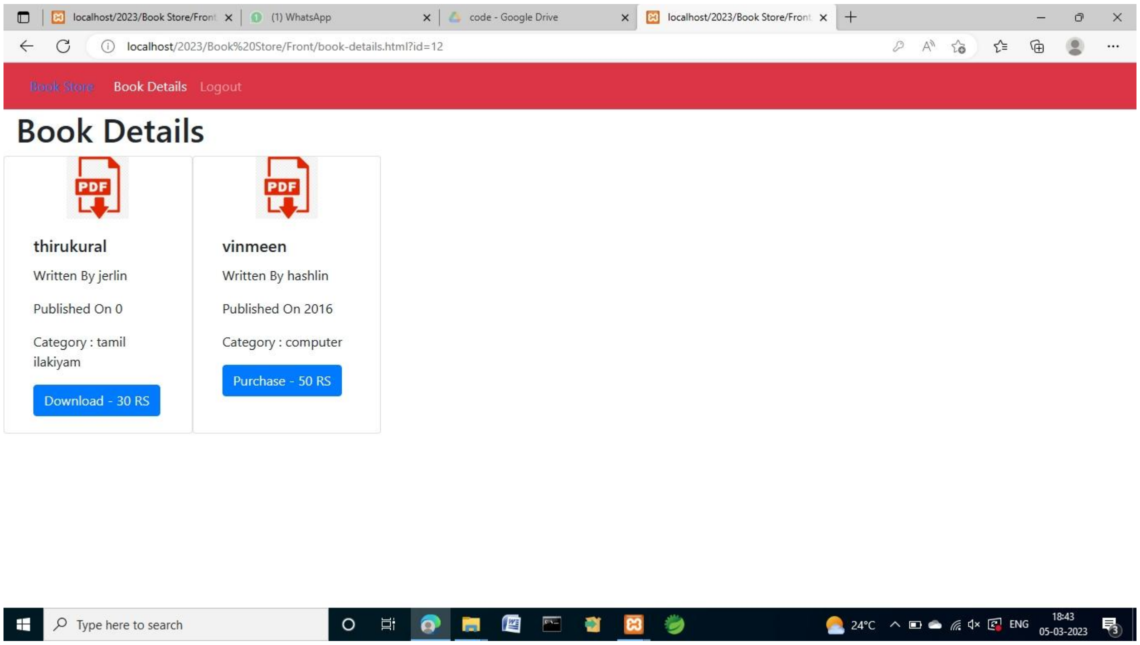
Book Name:

thirukural

Price:

30

AFTER BOOK PURCHASE:


localhost/2023/Book Store/Front
(1) WhatsApp
code - Google Drive
localhost/2023/Book Store/Front

localhost/2023/Book%20Store/Front/book-details.html?id=12

Book Details

thirukural

Written By jerlin

Published On 0

Category : tamil ilakiyam

vinmeen

Written By hashlin

Published On 2016

Category : computer