ONLINE BOOK REUSE SYSTEM
Project Report

2020-2022

Submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications of APJ Abdul Kalam Technological University

Submitted by:

NAVYA XAVIER
SJC20MCA-2039

MASTER OF COMPUTER APPLICATIONS

ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI
CHOONDACHERRY P.O, KOTTAYAM
KERALA

ST. JOSEPH' S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

(*An ISO 9001: 2008 Certified College*)

CHOONDACHERRY P.O, KOTTAYAM KERALA

CERTIFICATE

This is to certify that the project work entitled **Online book reuse system** *submitted by* ***Navya Xavier***
student of **Third** *semester* ***MCA*** *at* ***ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY***, ***PALAI*** *in partial fulfillment for the award of Master of Computer Applications.*

Mrs. Liz George	**Dr. Rahul Shajan**	**Mr. Anish Augustine**
(Project Guide)	**(Project Coordinator)**	**(HoD-MCA)**

Submitted for the Viva-Voce Examination held on

Examiner 1:

Examiner 2:

DECLARATION

I, ***Navya Xavier*** do hereby declare that the project titled **Online Book Reuse System** is a record of work carried out under the guidance of Mrs. Liz George, Asst. Professor, Department of Computer Science and Applications, SJCET, Palai as per the requirement of the curriculum of Master of Computer Applications Programme of APJ Abdul Kalam Technological University, Thiruvanthapuram. Further, we also declare that this report has not been submitted, full or part thereof, in any University / Institution for the award of any Degree / Diploma.

Place: Choondacherry
Date:

Navya Xavier

SJC20MCA-2039

ACKNOWLEDGEMENT

The success of any project depends largely on the encouragement and guidelines of many others. We would like to take this opportunity to express our gratitude to those people who have been instrumental in the successful completion of this project.

First and foremost, we give all glory, honor and praise to **God Almighty** who gave us wisdom and enabled me to complete the project successfully.

We also express sincere thanks, from the bottom of our heart, to our parents for their encouragement and support in all our endeavors and especially in this project.

Words are inadequate to express our deep sense of gratitude to **Dr. J. David, Principal, SJCET, Palai** for allowing us to utilize all the facilities of our college and also for his encouragement.

We extend our sincere gratitude to **Mr. Anish Augustine, Head of the Department of MCA**, **SJCET, Palai** who has been a constant source of inspiration and without his tremendous help and support this project would not have been materialized.

We extend our sincere gratitude to our project coordinator **Dr. Rahul Shajan, Asst. Professor, Department of MCA**, **SJCET, Palai** who has been a constant source of inspiration and without his tremendous help and support this project would not have been materialized.

We owe a particular debt of gratitude to our internal project guide, **Mrs. Liz George, Asst. Professor, Department of MCA, SJCET, Palai** for all the necessary help and support that she has extend to me. her valuable suggestions, corrections and the sincere efforts to accomplish our project even under a tight time schedule were crucial in the successful completion this project. We extend our sincere thanks to all of our teachers and non-teaching staff of SJCET, Palai for the knowledge they have imparted to us over the last three years.

We would also like to express our appreciation to all our friends for their comments, help and support.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 GENERAL INTRODUCTION

One of the major contribution of the 20th century is computer is being used in almost every field of life one cannot think about a world without computer. There has been a rapid and widespread growth in every sphere of the life due to the arrival of computers. They are very much reliable and that is why they are favourites of men in almost every department or section of work. They are indispensable to engineers, scientists, managers, business executives, administrators, accountants, teachers, students they have strengthened man's power in numerical computations and information processing and here by increasing the effectiveness of the organization.

The online Book Reuse System is a flexible web-based platform and will be very useful for buy, sell and borrow of books. This application is developed in programming platform of Python and MySQL for managing databases.

This project focuses on buying books as well as borrowing books and also user can sell the books available in their hands that might be useless for them. Mainly second hand books will be selling through this platform. Users can be able to manage their profile in this site. There will be an admin who will add new categories of books to this site and corresponding to this category new books can be added. Apart from this admin can view, edit and delete all the categories and books available in the site. Admin will approve the books that are submitted by the user for selling purpose. Only the books that are approved by the admin will be sell through this site. Admin can view all the orders happened in the site. Users can view all the available books in the site.

The main aim of this mini project is to design and develop a flexible platform for users to share the books that are in their hand and can also buy, borrow the books they needed. Thus the system will be time efficient and accurate.

1.2 **SOFTWARE INTRODUCTION**

1.2.1 **FRONT END**

PYTHON 3.7

Python is an interpreter, object-oriented, high level programming language with dynamic semantics. Its high level built-in Data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development as well as for use as a scripting or glue language to connect existing components together. Python's easy to learn syntax emphasizes readability and therefore reduce cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-text-debug cycle is incredibly fast. Debugging Python program is easy: a bug or a bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, interpreter prints a stack trace. A source level debugger allows inspection of the local and global variables, evaluation of arbitrary expressions, setting break points, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying the Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effectively.

1.2.2 **BACK END**

MySQL 5.8

Backend is the most important part in the working of the system. It is the back end that manages all the data. So it should be capable of managing, manipulating, protected data and provides sufficient security for an authorized access of database. Considering the above said requirements we have wide range of products available in the market such as Oracle, Oracle8i, Microsoft access, Microsoft SQL server, MS Visual FoxPro, paradox, MySQL etc. and many server scripting languages like Perl, Python, PHP.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create Maria DB. High Availability: Ensure business continuity with the highest levels of system availability through technologies that protect data against costly human errors and minimize disaster recovery downtime.

2. ABOUT THE ORGANIZATION

2.1 OVERVIEW

The establishment of St. Joseph’s College of Engineering, was the fulfillment of a long cherished dream of providing facilities for higher education to the people of the diocese and surrounding regions. The main objective is to develop a college with a distinct identity and character, where education and training are imparted in a truly Christian environment conducive to fostering Christian values such as faith in God, love for their fellow men and devotion to the motherland. Every facility is provided in the campus to create an environment fully conducive to realizing this objective.

Discipline, hard work, positive thinking, commitment to excellence and abiding faith in the Almighty are the guiding principles that propel the college to its vision of emerging as a Centre of Excellence in technical education in the country. Value systems such as eco-friendliness, quality consciousness and work ethics are also being instilled through the special work culture and campus life existing in the college.

The college aims to provide an education that WORKS! – an education that helps the students in ensuring a challenging and satisfying career after the course.

2.2 VISION

Developing into a world-class, pace-setting Institute of Engineering and Technology with distinct identity and character, meeting the goals and aspirations of the society.

2.3 MISSION

• To maintain a conducive infrastructure and learning environment for world class education.
• To nurture a team of dedicated, competent and research oriented faculty.

• To develop students with moral & ethical values, for their successful career by offering variety of programmes and services.

2.4 OBJECTIVES

St. Joseph’s College of Engineering and Technology, Palai was instituted with the objective of developing a center of professional learning with a distinct identity and character, for imparting education and training in a truly Christian environment, fostering Christian values of faith and love to God and fellowmen. The college aims to provide the kind of education that helps to achieve academic excellence and thereby ensures a challenging and satisfying career for the students on the successful completion of the programme. With this perspective, training is organized on a regular basis for the development of personality, learning and communication skills as well as employability skills.

3. SYSTEM ANALYSIS

System analysis is a structured method for identifying and solving problems. Analysis implies breaking something into its parts so that the whole may be understood. The definition of systemanalysis not only process analysis but also that of synthesis, which implies the process of puttingparts together to form a new whole. All the activities relating to the life cycle phase must be performed managed and document. To design a system, we need requirements of the system and the specification document are prepared in this phase. The purpose of this document is to specify the functional requirement of the software that is to build. The specifications are intended to guide the activities, relationships and all other objectives.

The main thing is to find what is to be done to solve the problems with the current system. In the phase the problems or drawbacks of the current system is identified and the necessary actions tosolve these problems are recommended.

3.1 EXISTING SYSTEM

Traditionally the system is totally manual system. Actually there is no existing system currently. The existing system can be defined as two systems mainly: The Library Management System and the Online purchasing of books.

The library management system is a system which is same as the traditional library system in which users can borrow the books from library for a specific period (for example 2 or 3 weeks)and after this specific period the user want to return the books taken. There is no scope of buying the books from the library by paying the corresponding money of the book.

In online purchasing of books is a system in which user can buy the books they needed in online by paying the full amount. We do not have the provision to borrow books from an online book purchasing site.

The limitations of the existing systems are:

- In library management we can’t buy books from library.
- In online purchasing we can’t borrow books from these sites.
- We don’t have the provision to sell the books that are in our hand(mainly second hand books).

Based on the drawbacks and inadequacies of the existing system, the new system is designed which could rectify all the existing system. For that discussions were carried out to choose the best package for developing new systems.

3.2 PROPOSED SYSTEM

Proposed website is an interactive way of overcoming all the drawbacks. The online book reuse system is a platform to buy, sell and borrow books so that users can buy or borrow books according to their needs. Users can also the secondhand books available in their hands to this site. The online book reuse system actually combines the good sides of both the library management system and online purchasing of books. Admin will be able to approve the books that are submitted by the user for selling purpose. Admin can view all the orders happening in the system. User can view all the available books in the site.

The advantages of this system are:

- This system saves the precious time of user and very efficient to use.
- Users can buy book easily by making online payment.
- Users can have the provision to sell the books that are available in their hand which will be a help for others.
- Save the resources which are used to make books.
- Can take a step to prevent deforestation.

3.3 SOFTWARE REQUIREMENT SPECIFICATION (SRS)

3.4 FEASIBILITY ANALYSIS

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product. The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on thesedata, the output data required to be produced by the system as well as various constraints on the behavior of the system. This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.
- Facility to communicate data to distant locations.

In examining technical feasibility, configuration of the system is given more importance than the actual makes of hardware. The configuration should give the complete picture about the system'srequirements:

1. How many workstations are required, how these units are interconnected so that they could operate and communicate smoothly.
2. What speeds of input and output should be achieved at particular quality of printing?

3.4.1 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost / Benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. This is an outgoing effort that improves in accuracy at each phase of the system life cycle.

3.4.2 OPERATIONAL FEASIBILITY

This is mainly related to human organizational and political aspects. The points to be considered are:

• What changes will be brought with the system?
• What organizational structure are disturbed?
• What new skills will be required? Do the existing staff members have these skills? If not, can they be trained in due course of time?

This feasibility study is carried out by a small group of people who are familiar with information system technique and are skilled in system analysis and design process.
Proposed projects are beneficial only if they can be turned into information system that will meet the operating requirements of the organization. This test of feasibility asks if the system will work when it is developed and installed.

3.4.3 TECHNICAL FEASIBILITY

Technical Feasibility centers on the existing computer system (hardware, software etc and to what extent it can support the proposed software. The hardware and software requirements of the system are industry standards. Here no extra expenditure is expected to incur. This system is technically feasible. The considerations that are normally associated with technical feasibility include:

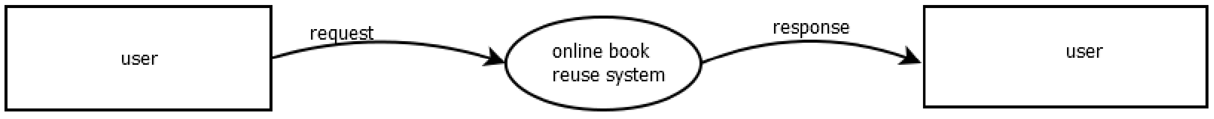
 Development risk can the system element be designed so that to the necessary function and performance is achieved within the constraints uncovered during the analysis.
 Resource availability is competent staff available to develop to the system element in question. Are other necessary resources (hardware and software) available to build the system?
 Technology has the relevant technology progressed to a state that will support the system.

3.5 DATA FLOW DIAGRAM (DFD)

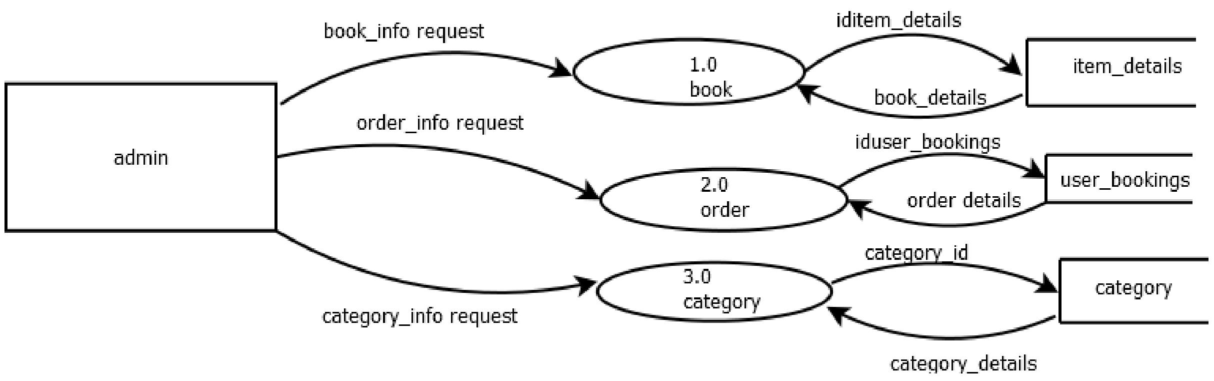
A Data Flow Diagram (DFD) is a graphical representation of the flow of data through an information system, modeling its process aspects. Larry Constantine first developed the DFD as a way of expressing system requirements in a graphical form lead to module design. Often they are preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). So it is the starting point of design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consists of a series of bubbles joined by lines and its also known as a “bubble chart”.
DFD Symbols:

- A system defined source or destination of data.
- An array identifies data flow, data in motion.
- A circle represents the process that transforms incoming data flow to outgoing data flow.
- An open rectangular is data store-data at rest or temporary repository of data.

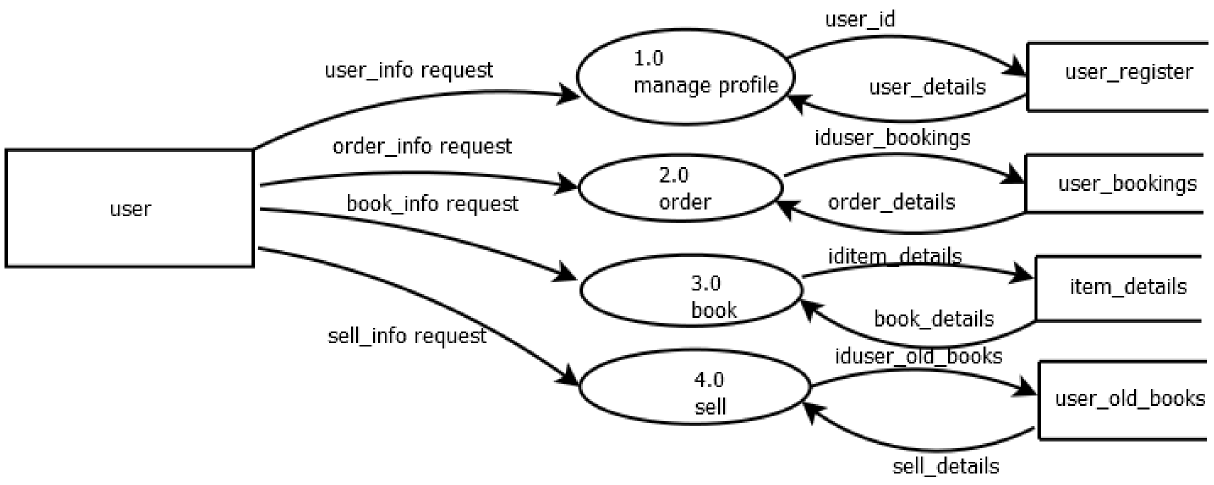
CONTEXT LEVEL- LEVEL 0 DFD OF ONLINE BOOK REUSE SYSTEM



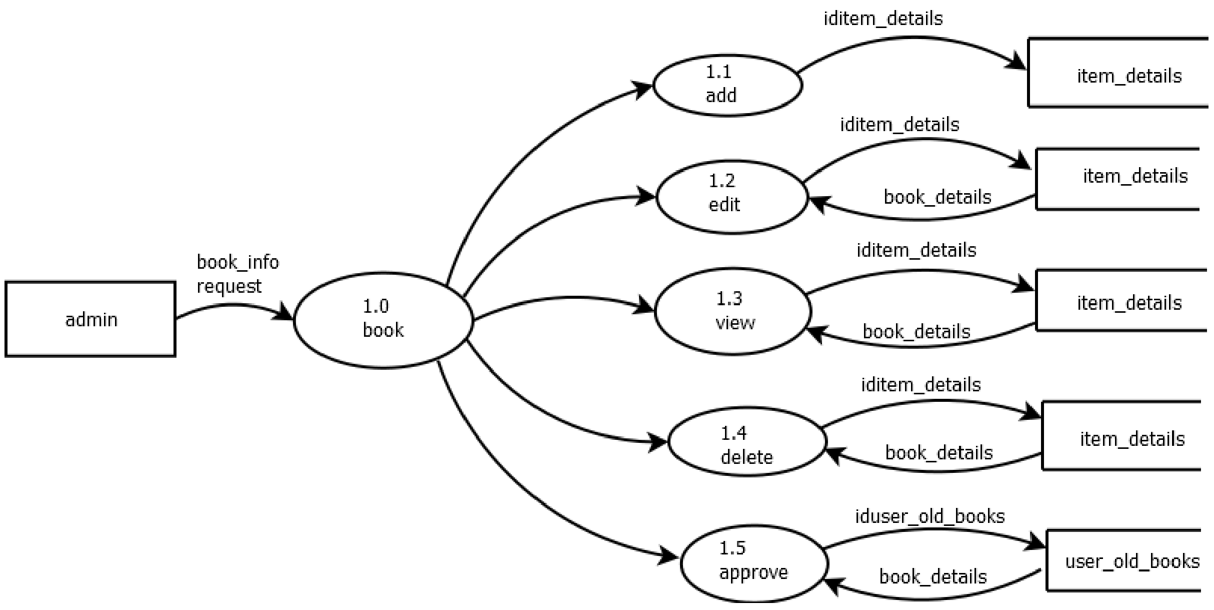
LEVEL 1 DFD FOR ADMIN



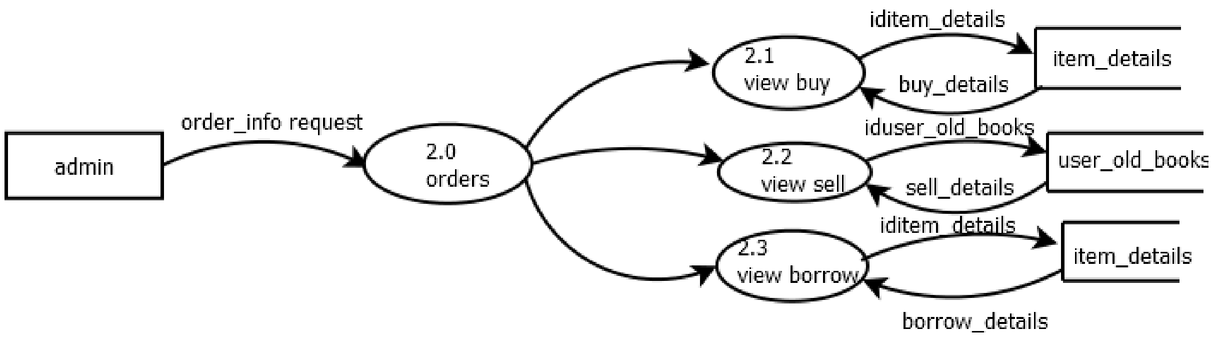
LEVEL 1 DFD FOR USER



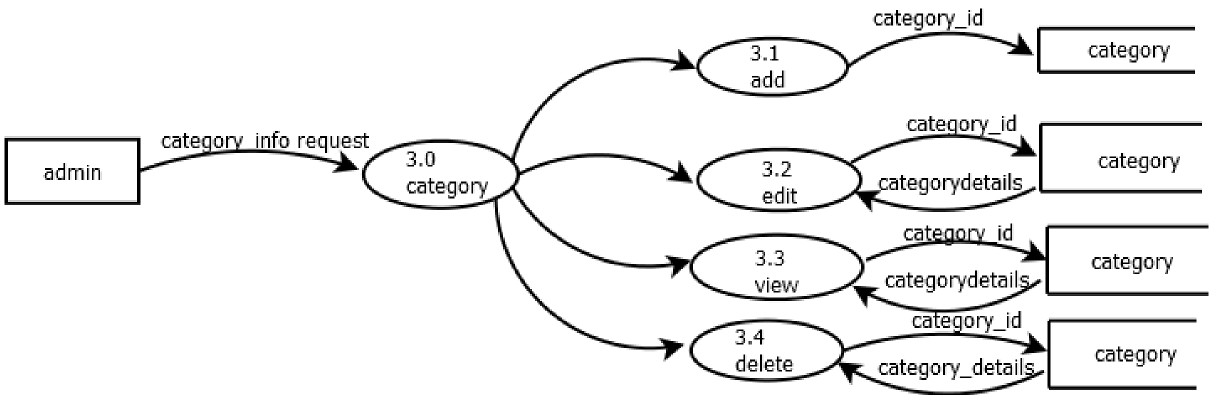
LEVEL 2 DFD FOR ADMIN (BOOK)



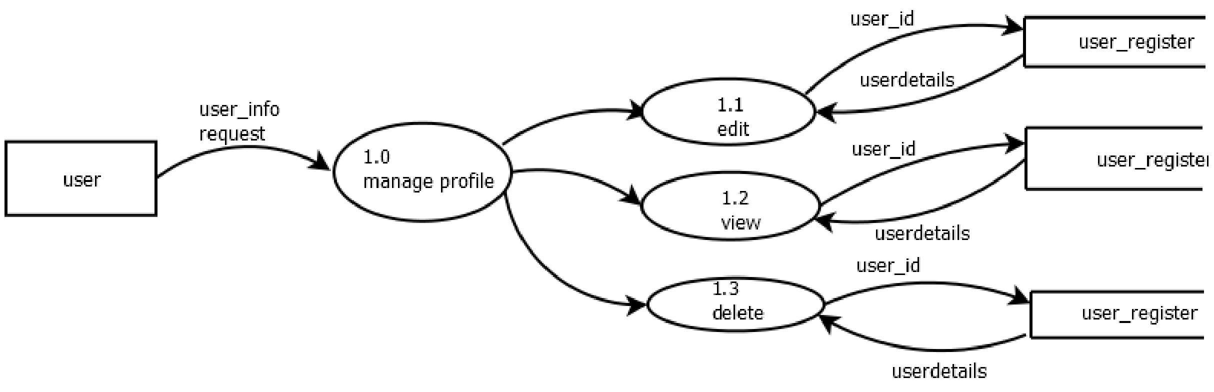
LEVEL 2 DFD FOR ADMIN (ORDER)



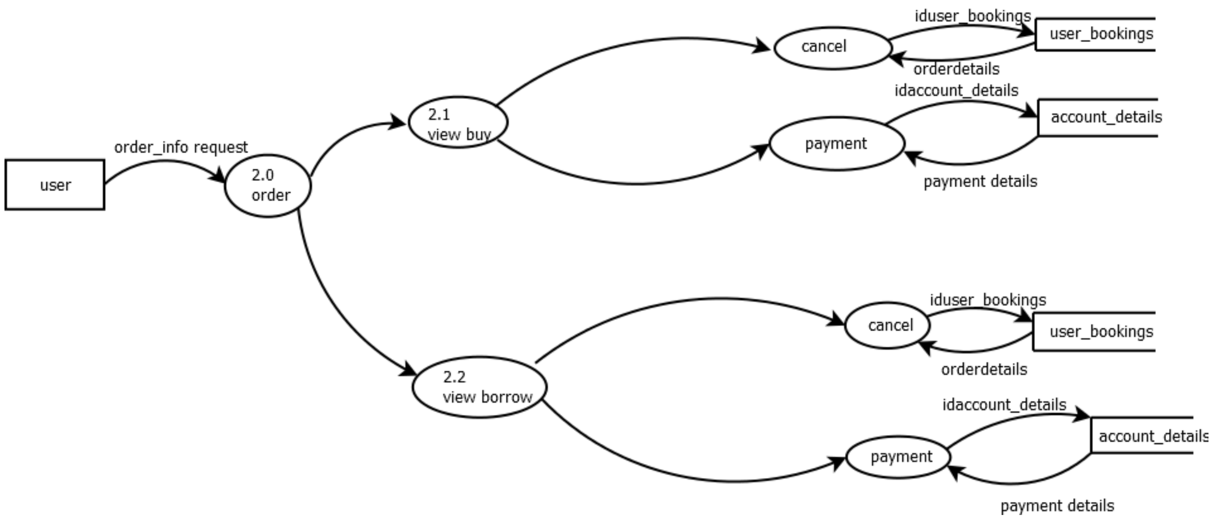
LEVEL 2 DFD FOR ADMIN (CATEGORY)



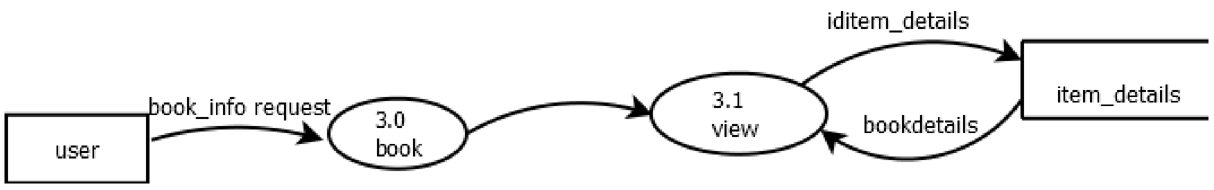
LEVEL 2 DFD FOR USER (MANAGE PROFILE)



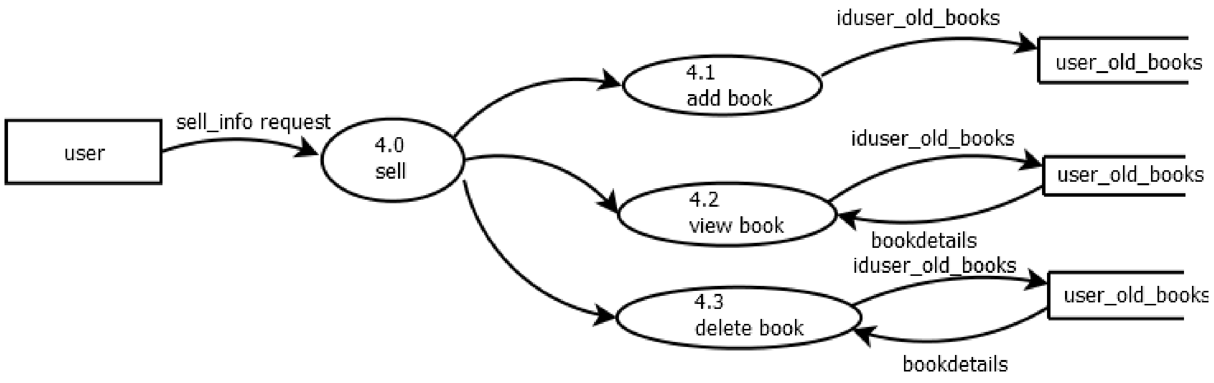
LEVEL 2 DFD FOR USER (ORDER)



LEVEL 2 DFD FOR USER (BOOK)



LEVEL 2 DFD FOR USER (SELL)



4. SYSTEM DESIGN

System design is the second phase of the software life cycle. The system goes through logical and physical state of development. The user-oriented performance specification is extended intoa design specification, while designing the needed system. The design phase begins when the requirement specification document for the software to be developed is available. When the Requirement Specification activity is entirely in the problem domain, design is the first step to move from the problem domain to the solution domain. Design is essentially the bridge betweenthe requirements specification and the final solution for satisfying these requirements.

4.1 INPUT DESIGN

Input design is the process of converting a user-oriented description of the inputs to a computer- based business system into a programmer-oriented specification. The design decision for handling input specify how data are accepted for computer processing. Input design is a part of overall design that needs careful attention. The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a way so as to get the relevant information, extreme care is taken to obtain the pertinent information. If the data going into the system is incorrect then the processing and outputs will magnify these errors. The goal of designing input data is to make data entry as easy, logical and free from errors as possible. The following are the objectives of input design:

• To produce a cost-effective method of input.

• To ensure validation.

Effort has been made to ensure that input data remains accurate from the stage at which it is recorded and documented to the stage at which it is accepted by the computer. Validation procedures are also present to detect errors in data input, which is beyond control procedures. Validation procedures are designed to check each record, data item or field against certain criteria.

In my proposed system of book reuse system, data has to be accurate and complete. The data validation, a procedure of the proposed system, provides program checks for the completeness, consistency, reasonableness and sequence of the system.

Maximum care has been taken to ensure that user types in only minimum data into the system, as all he/she will have to do is to move and click the mouse or strike a key to select the desired data at the desired position.

The screens are designed in such a way that the user can find the needed like options, actions etc. with ease of use. The needed columns, where interaction is needed, like labels, buttons are also simple. The related data columns are clubbed together as groups, so that the user can understands the related data easily.

The input design is the link between the information system and the user. It comprises developing specifications and procedures for data preparation and those steps that are necessary to put input data into a usable form for processing data entry. The design of inputs focuses on controlling the number of inputs required, controlling errors, avoiding delay, avoiding extra steps and keeping the process simple.

4.2 OUTPUT DESIGN

The output design phase of the system design is concerned with the conveyance of information to the end users in user-friendly manner. The output design should be efficient, intelligible so that the system relationship with the end user is improved and thereby enhancing the process of decision making. The output design is an ongoing activity almost from the beginning of the project, efficient and well-defined output design improves the relation of the system and the user. The primary considerations in the design of the output are the requirement of the information and the objective of the end user. There are various types of outputs required by most of the systems, but outputs of online book reuse Website are purely interactive outputs-which involve the user in communicating with the computer.
The system output may be of any of the following:

- A report
- A document
- A notification

The output design specification is made in such a way that it is unambiguous and comprehensive. The approach to output design is very dependent on the type of output and nature of data. Special attention has to be made to data editing. The choice of appropriate output medium is also an important task. The output designed must be specified and documented, data items have to be accurately defined and arranged for clarity. The layout of the output will be normally specified on a layout chart. The final design layout must be approved by the user, communicated in detailed to the programmer. The user's requirements are quite different from that of the programmer.

Before preparing a specification for the programmer, it is prudent to ensure that the design is acceptable to the user.

4.3 TABLE DESIGN

1. LOGIN

Field name	**Data type**	**constraints**	**description**
Admin_id	varchar	Primary key	username of admin
password	varchar	Not null	password

2. user_register

Field name	**Data type**	**constraints**	**description**
user_id	varchar	Primary key	User name
first_name	varchar	Not null	First name
last_name	varchar	Not null	Last name
email	varchar	Not null	email
phone	bigint	Not null	phonenumber
housename	varchar	Not null	housename

city	varchar	Not null	city
pincode	bigint	Not null	pincode
country	varchar	Not null	country
state	varchar	Not null	state
password	varchar	Not null	password

3. user_old_books

Field name	**Data type**	**constraints**	**description**
Iduser_old_books	int	Primary key	Id of book
User_id	varchar	Foreign key	name of user
Category_id	varchar	Not null	Category of book
Book_name	varchar	Not null	Book name
author	varchar	Not null	author
year	int	Not null	year
Book_description	varchar	Not null	Book description
status	varchar	Not null	status
reply	varchar	Not null	reply
price	varchar	Not null	price
filepath	varchar	Not null	Image path of book

4. Category

Field name	**Data type**	**constraints**	**description**
Category_id	int	Primary key	Category id
name	varchar	Not null	Category name

5. item_details

Field name	**Data type**	**constraints**	**description**
Iditem_details	int	Primary key	Id of item
Category_id	int	Foreign key	Category id
Posted_date	date	Not null	Date of posting
Book_name	varchar	Not null	Book name
Purchase_price	int	Not null	Purchase price
status	varchar	Not null	status
Rental_price	int	Not null	Rental price
Book_description	varchar	Not null	Book description

6. user_bookings

Field name	**Data type**	**Constraints**	**description**
Iduser_bookings	int	Primary key	id
User_id	varchar	Foreign key	username
Iditem_details	int	Foreign key	Itemdetails id
Booking_date	date	Not null	Booking date
quantity	int	Not null	quantity
Total_price	int	Not null	price

status	varchar	Not null	status
Ownership_type	varchar	Not null	Ownership type

7. account_details

Field name	**Data type**	**constraints**	**description**
Idaccount_details	int	Primary key	Id of account
Card_no	int	Not null	Card number
Card_cvv	int	Not null	cvv
Card_holder_name	varchar	Not null	Name of cardholder
Card_expiry_date	date	Not null	Expiration date

4.4 PROCESS DESIGN

A successful process design has to take into account the appropriateness of the process to overall organization objective. Process design requires a broad view of the whole organization and should not have a myopic outlook. And the process should deliver customer value with constant involvement of the management at various stages.

In order to achieve a good process design, effective process strategy is required, which deals with a singular line items required to manufacture the end product. Effective process strategy deals with raw material procurement, customer participation, technology investment, etc.

Over a period of time process design has undergone change and new concepts like flexible manufacturing systems have been developed, which delivers efficient and effective production design and analysis.

4.4.1 MODULE DESCRIPTION

USER

- User registration
- Login to the system
- Can submit request for selling books
- View available books
- Buy books
- Borrow books

ADMIN

- Login to the system
- View and approve user selling request
- View all the orders
- Add categories of books
- Edit, view and delete categories of books
- Add books
- Edit , view and delete books

5. SYSTEM TESTING & IMPLEMENTATION

5.1 SYSTEM TESTING

It is a systematic technique for the program structure while at the same time conducting test to uncover errors associated with the interfaces. The objective is to take unit tested module and build the program structure that has been dedicated by design. All modules are combined in thistesting step. Then the entire program is tested as whole. If a set of errors is encountered correctionis difficult because the isolation of cause is complicated by vastness of the entire program.
Using integrated test plans prepared in the design phase of the system developed as a guide, the integration was carried out. All the errors found in the system were corrected for the next step.

5.1.1 VALIDATION TESTING

At the end of the integration testing, software is assembled as a package, interfacing errors have been uncovered and corrected and final series of software validation testing begins. Validation testing can be defined in any ways, but a simple definition is that validation succeed when the software function in a manner that can be reasonably accepted by the user. Software validation is achieved through a series of black box that demonstrate conformity with the requirements.
After validation test has been completed, one of the following two possible conditions exists:

- The function or performed characteristics confirm to specification and are accepted.
- A deviation from specification is uncovered at this step in this project is corrected prior to the completion of the project is the help of user by negotiating to establish a method resolving deficiencies. Thus, the proposed system under consideration has

been tested by using validation testing and found to be working satisfactory.

5.1.2 OUTPUT TESTING

After performing validation testing the next step is to perform the output testing of the proposed system. Since no system could be useful if it does not produce required output in the specified format. The output generated are displayed by the user under consideration are tested by the company with the format required by the user. Here the output format is considered in two ways. One is onscreen and the other is printed format. The output format on the screen is found to be correct as the system design phase accounting to the user hence: the output testing does not result in any correction in the system.

5.1.3 USER ACCEPTANCE TESTING

User acceptance of a system is a key factor to the success of any system. The system under consideration was tested for user acceptance by constantly keeping in touch with the proposed system user at the time of developing and making changes whenever required. This is done with regard to the following points.

- Input screen design.
- Output screen design.
- Format of the reports and other output.

5.1.4 BLACK BOX TESTING

Knowing the specified function that the product has been designed to perform, test can be conducted that each function is fully operational. Black box test is carried out to test that input toa function is properly accepted and output is correctly produced. A black box test examines someaspects of a system little regard for the internal structure of the software. Errors in the followingcategories were found through Black box testing:

- Incorrect or missing function.
- Interface errors.
- Errors in database structure or external database access.
- Performance errors
- Initialization and termination errors

5.1.5 WHITE BOX TESTING

White box testing of software is predicated on a close examination of procedural detail. The status of the project may be tested at various points to determine whether the exposed or asserted status is corresponding to the actual status.

Using these following test cases can be derived:

- Exercise all logical functions on their true and false side.
- Execute all loops within their boundaries and their operation bounds.
- Exercise internal data structure to ensure their validity.

5.2 IMPLEMENTATION

Implementation includes placing the system into operation and providing the users and operation personnel with the necessary documentation to use and maintain the new system. Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not guarantee improvement in the organization using the new system, as well as, improper installation will prevent. There are four methods for handling a system conversion.
Parallel approach: The old system is operated with the new system. Direct cut over method: The old system is replaced with the new system. Pilot approach: Working version of the system is implemented in one part of the organization based on the feedback, changes are made and the system is installed in the rest of the organization by one of the other methods. Phase-in-method: Gradually implements the system across all users. We have used the direct

cut over method in our implementation.

The Implementation Plan describes how the information system will be deployed, installed and transitioned into an operational system. The plan contains an overview of the system, a brief description of the major tasks involved in the implementation, the overall resources needed to support the implementation effort, and any site-specific implementation requirements. The plan is developed during the Design Phase and is updated during the Development Phase the final version is provided in the Integration and Test Phase and is used for guidance during the Implementation Phase.

There are three types of implementation:

- Conversion
- User training
- Documenting the system

6. MAINTENANCE

Once the system has been implemented, it cannot be considered as the end of the system life cycle. After the implementation it is necessary that the system be constantly monitored so that it may be decided as how the system is working. If any problem is encountered it is necessary that the in-charge person rectifies the problem so that the clients may not be affected by the problem. This phase of the system development life cycle is known as the maintenance period.

There are three types of maintenances:

- Correctives (fixing bugs/errors)
- Adaptive (updates due to environment changes)
- Perfective (enhancements, requirements change)

7. CONCLUSION

“The Online Book Reuse System” is a flexible platform for the very useful for users to share the books in their hand mostly second hand books and also can buy, borrow books according to their needs. Without wasting time in going to library for borrowing of books and without spending more amount in shops users can get their books at their doorstep by making online payment. Thus, this system will be time efficient and accurate.

7.1 SCOPE FOR FUTURE ENHANCEMENTS

The project will be very helpful in future. It saves time, effort and makes a better user friendly environment to all the users who use this project. Making enhancement is all about perfective maintenance. It means adding, modifying or redeveloping the code to support changes in the specification. It is necessary to keep up with changing user needs and the operational environment.
The following are the future scope for the project.

- Outside Kerala and India delivery will be done.
- Users will get reviews from buyers who buy their secondhand books.
- Notifications regarding new arrival of books will be done in the system.

8. BIBLIOGRAPHY

8.1 REFERENCES

1. Wesley J. Chun, **“Core Python Applications Programming”**, 3rd Edition , Pearson Education, 2016
2. Charles Dierbach, **“Introduction to Computer Science using Python”**, Wiley, 2015
3. Jeeva Jose, **“Taming Python by Programming”**, Khanna Publishers, New Delhi, 2018
4. Downey, A. et al., **"How to think like a Computer Scientist: Learning with Python"**, John Wiley, 2015

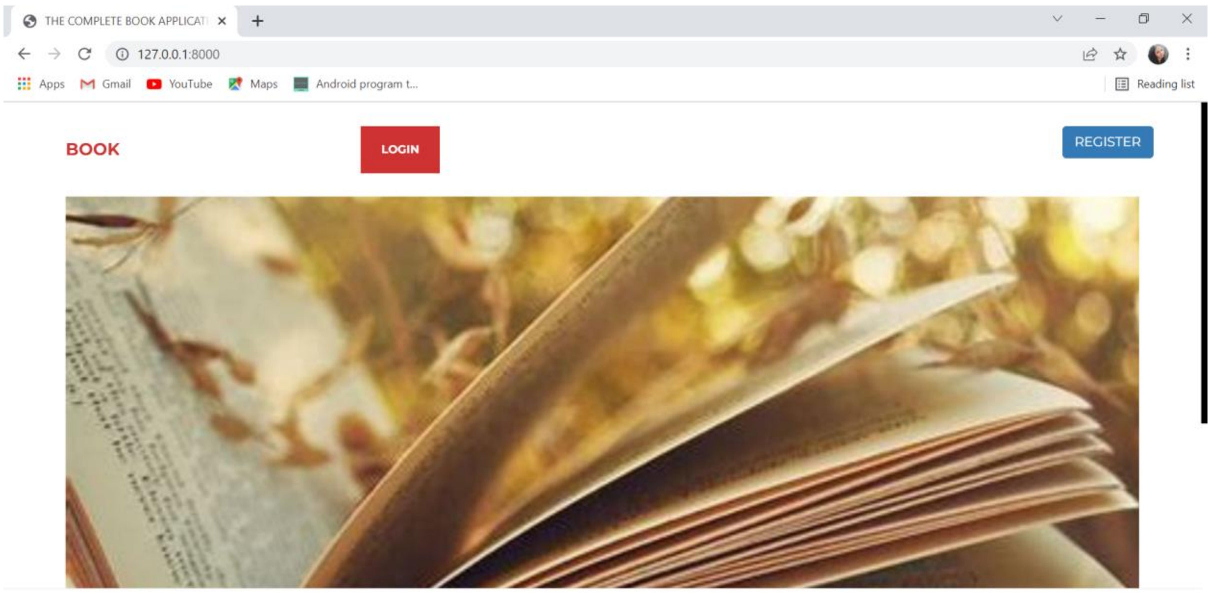
8.2 WEBSITES

- **https://www.learnpython.org**
- **https://www.python.org/about/gettingstarted**
- **https://www.programiz.com/python-programming**
- https://www.econema.org/reuse-recycling-textbooks/

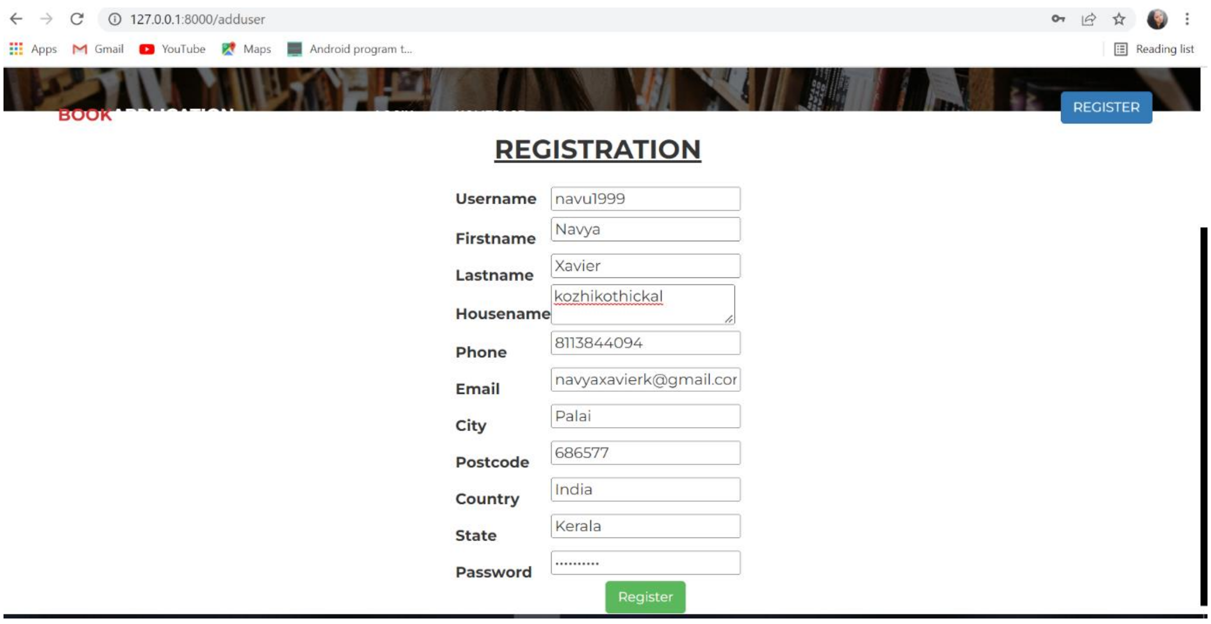
9. APPENDIX

9.1 SCREEN SHOTS

HOMEPAGE- ONLINE BOOK REUSE SYSTEM



REGISTRATION OF USER

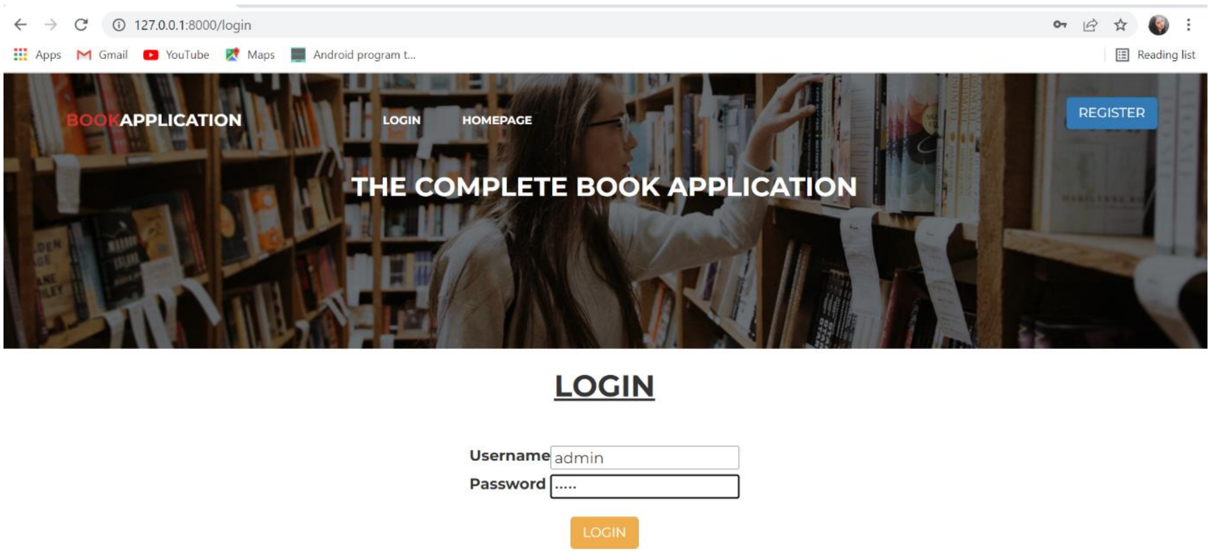


REGISTRATION

Username	navu1999
Firstname	Navya
Lastname	Xavier
Housename	kozhikothickal
Phone	8113844094
Email	navyaxavierk@gmail.cor
City	Palai
Postcode	686577
Country	India
State	Kerala
Password	....

Register

ADMIN LOGIN

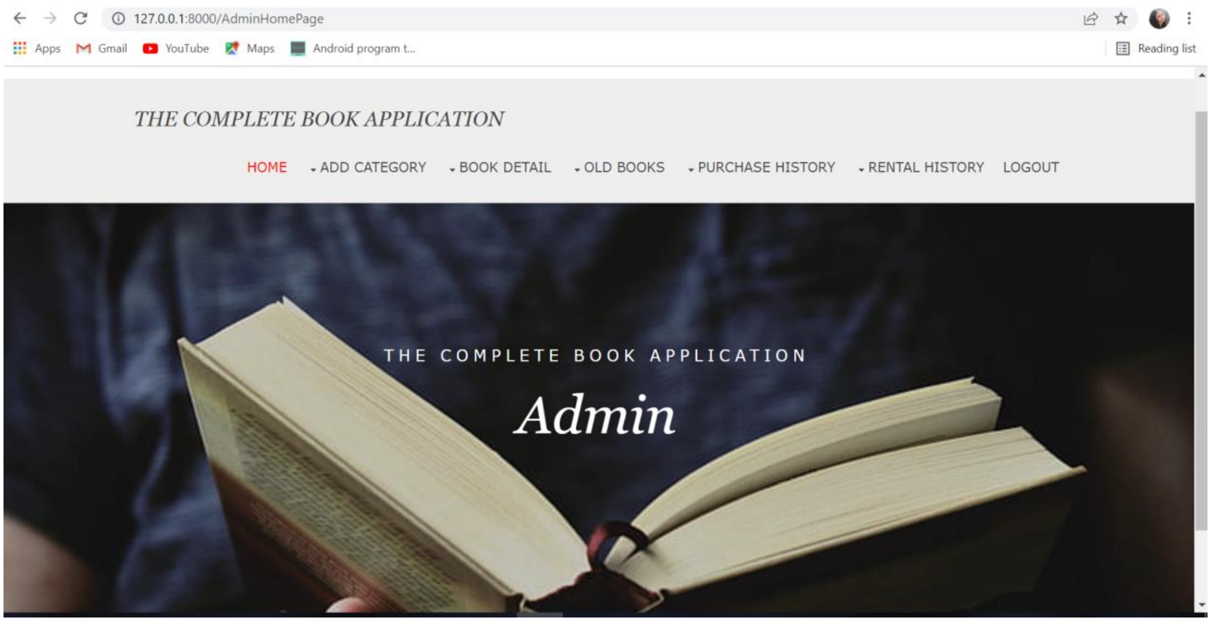


LOGIN

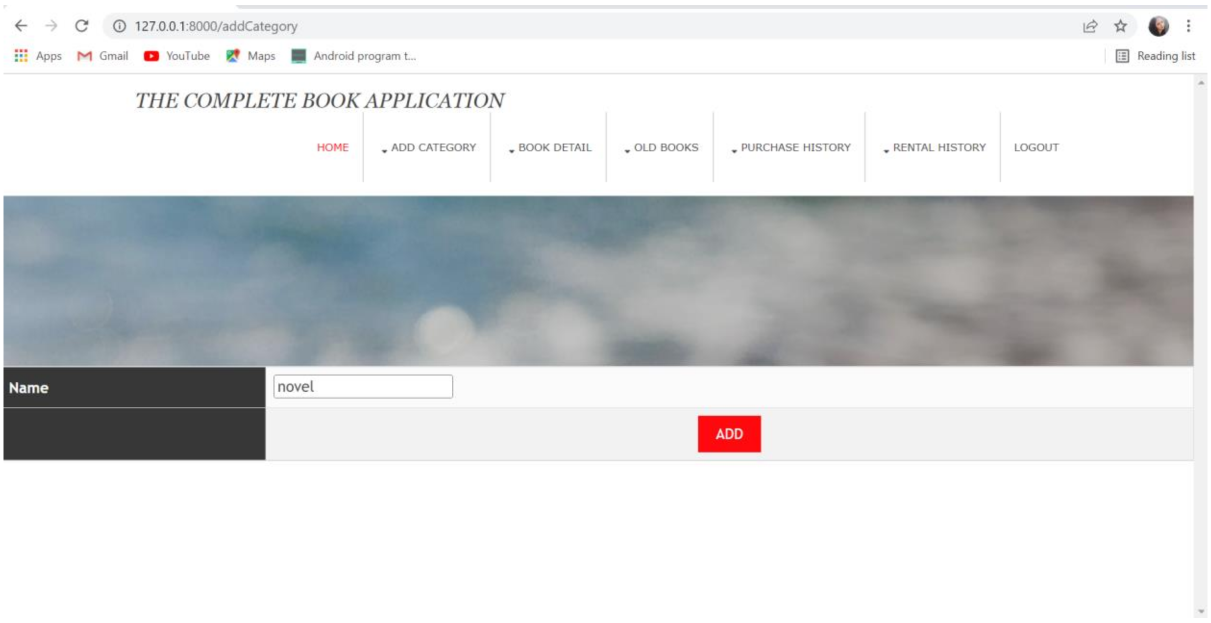
Username	admin
Password	.....

LOGIN

ADMIN HOMEPAGE

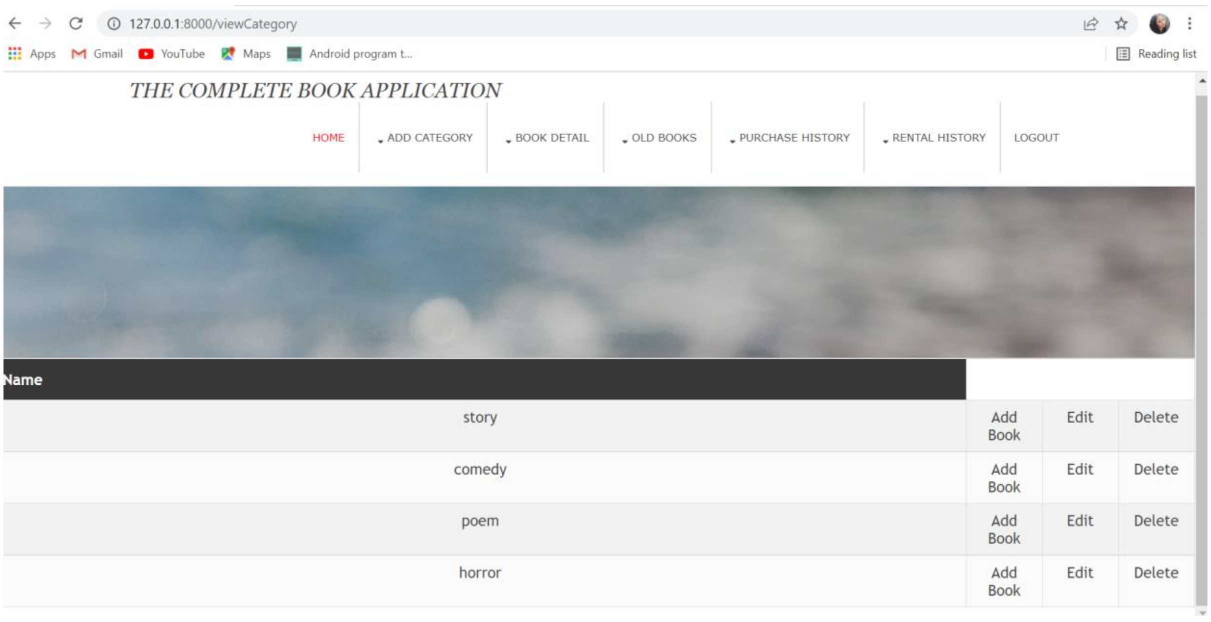


ADMIN ADD CATEGORY



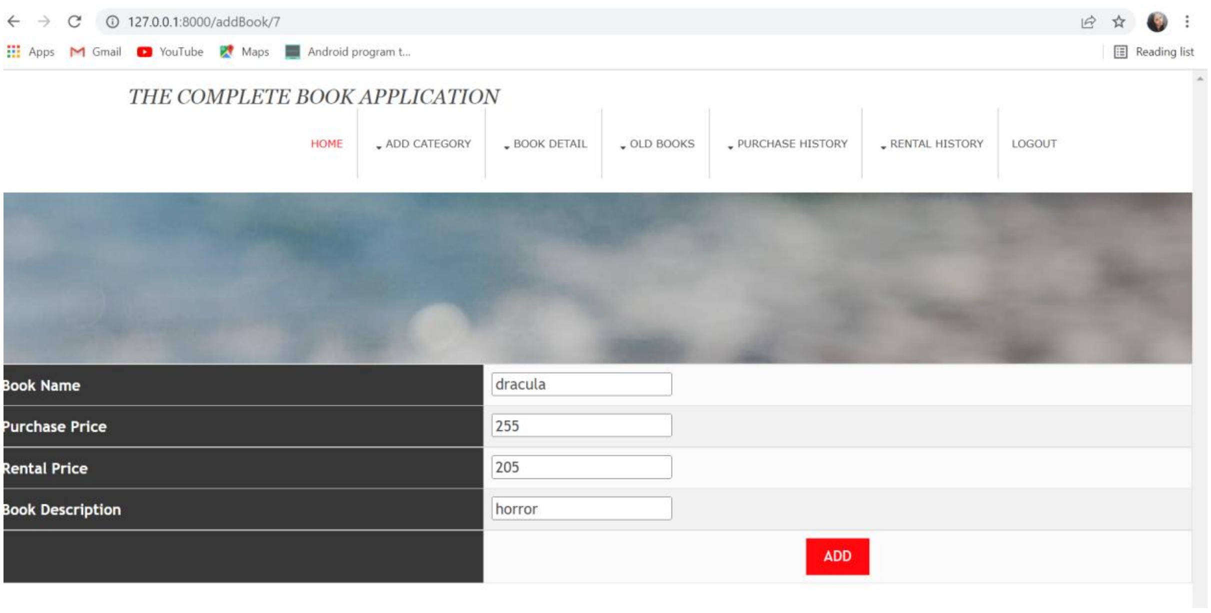
Name: novel

ADMIN VIEW CATEGORY



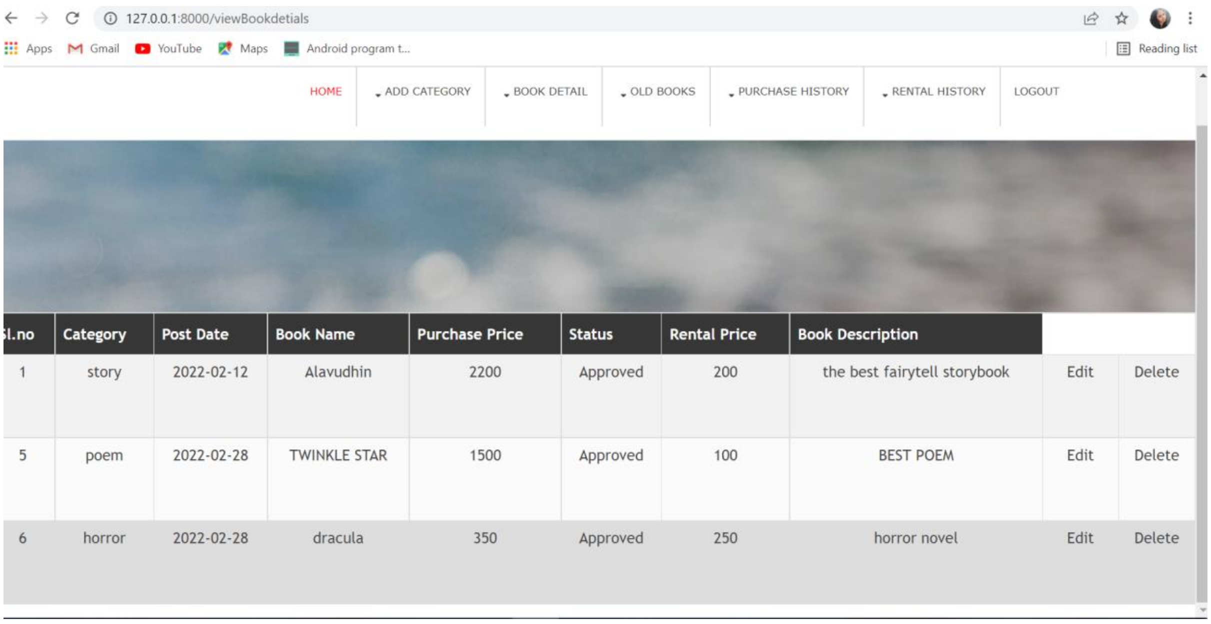
Name			
story	Add Book	Edit	Delete
comedy	Add Book	Edit	Delete
poem	Add Book	Edit	Delete
horror	Add Book	Edit	Delete

ADMIN ADD BOOKS



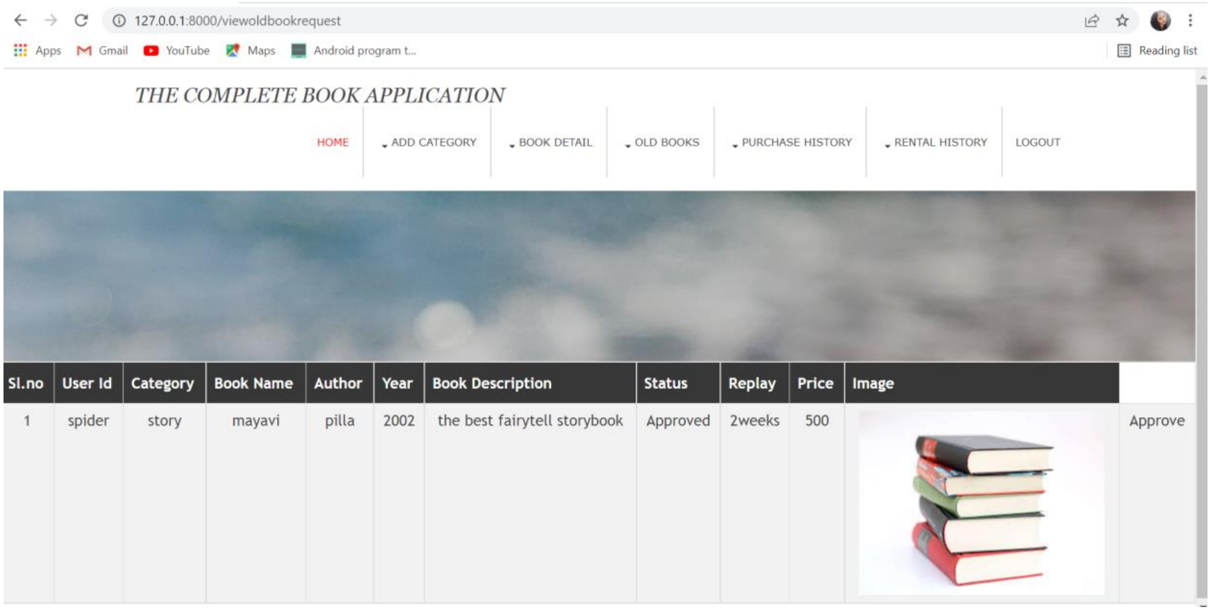
Book Name	dracula
Purchase Price	255
Rental Price	205
Book Description	horror

ADMIN VIEW BOOK DETAILS



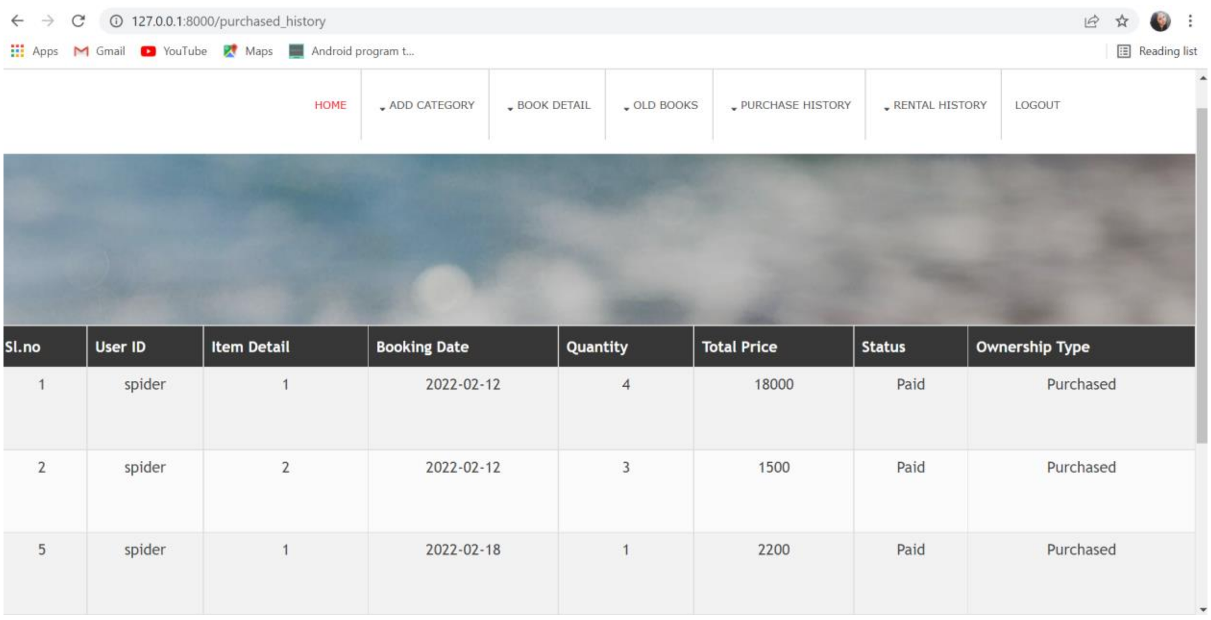
l.no	Category	Post Date	Book Name	Purchase Price	Status	Rental Price	Book Description		
1	story	2022-02-12	Alavudhin	2200	Approved	200	the best fairytell storybook	Edit	Delete
5	poem	2022-02-28	TWINKLE STAR	1500	Approved	100	BEST POEM	Edit	Delete
6	horror	2022-02-28	dracula	350	Approved	250	horror novel	Edit	Delete

ADMIN APPROVE BOOKS



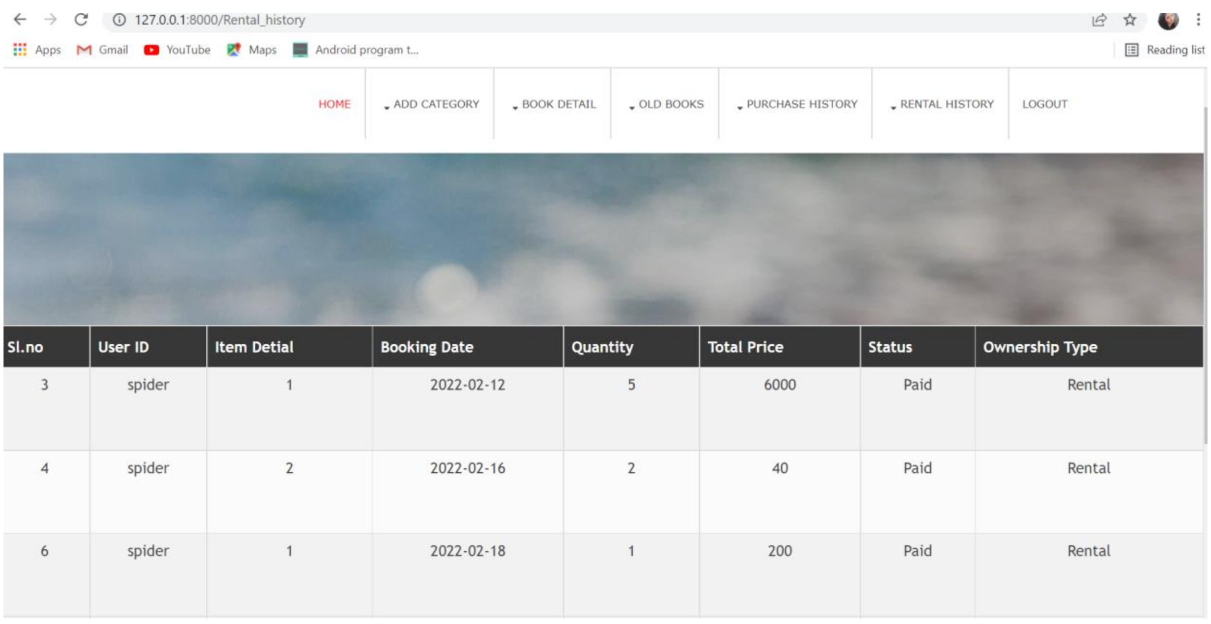
Sl.no	User Id	Category	Book Name	Author	Year	Book Description	Status	Replay	Price	Image	
1	spider	story	mayavi	pilla	2002	the best fairytell storybook	Approved	2weeks	500		Approve

ADMIN VIEW BUY ORDERS



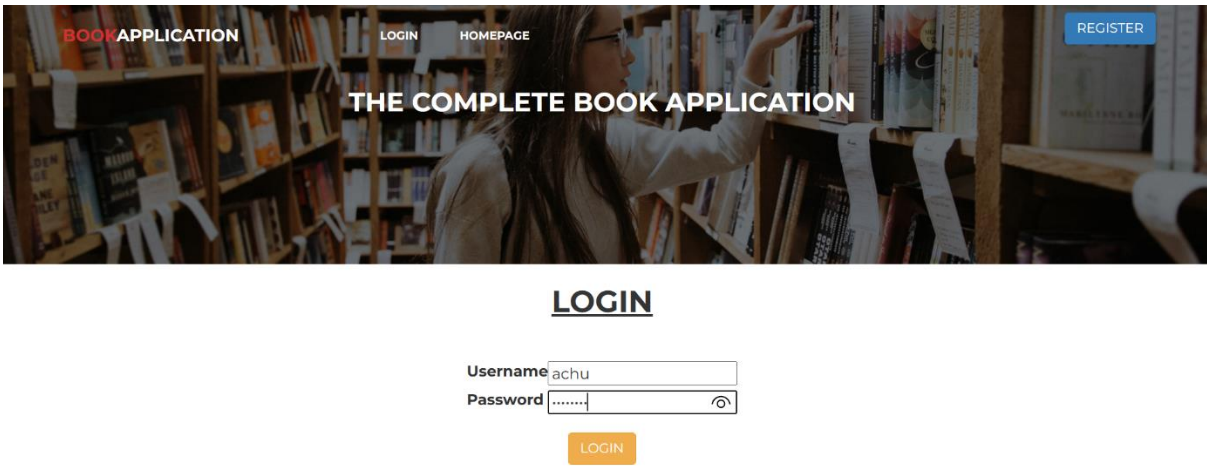
Sl.no	User ID	Item Detail	Booking Date	Quantity	Total Price	Status	Ownership Type
1	spider	1	2022-02-12	4	18000	Paid	Purchased
2	spider	2	2022-02-12	3	1500	Paid	Purchased
5	spider	1	2022-02-18	1	2200	Paid	Purchased

ADMIN VIEW RENTAL ORDERS

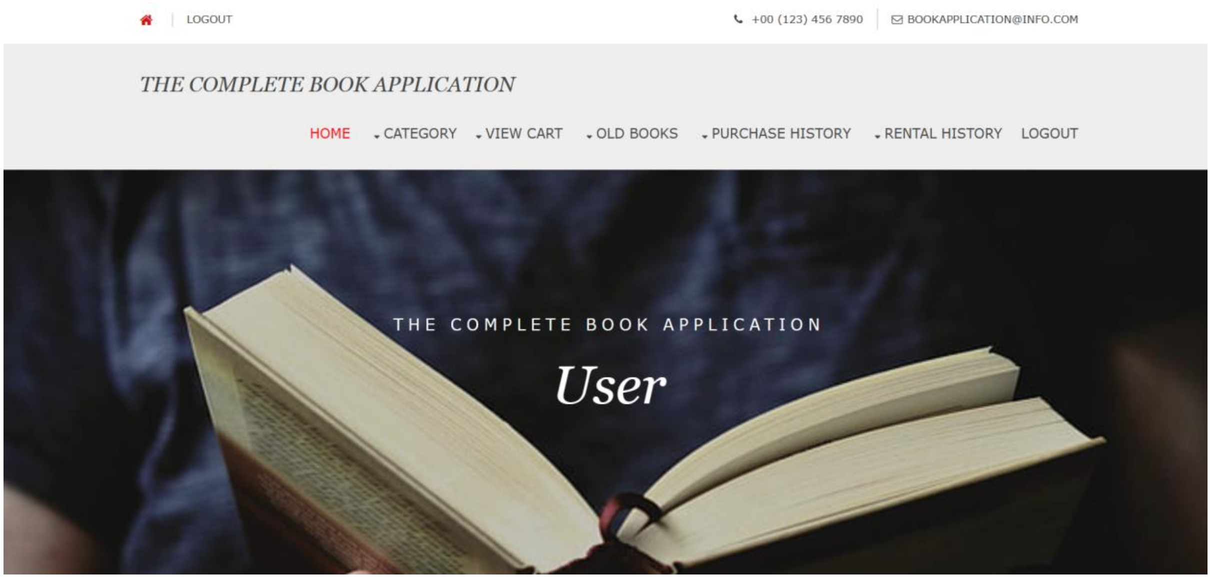


Sl.no	User ID	Item Detial	Booking Date	Quantity	Total Price	Status	Ownership Type
3	spider	1	2022-02-12	5	6000	Paid	Rental
4	spider	2	2022-02-16	2	40	Paid	Rental
6	spider	1	2022-02-18	1	200	Paid	Rental

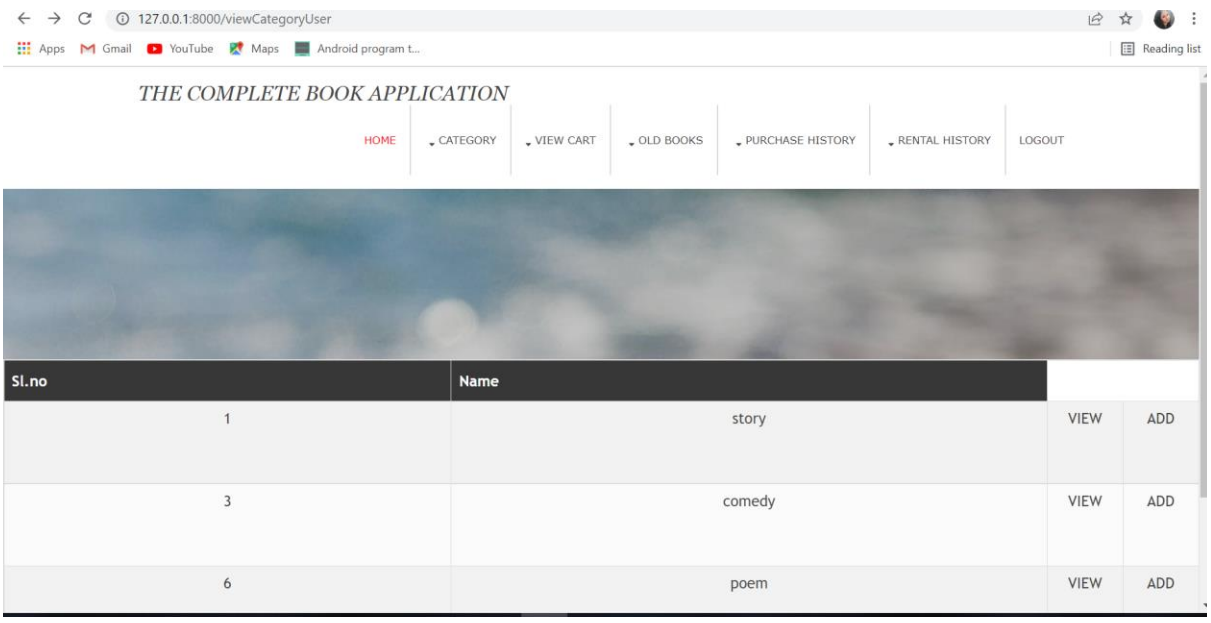
USER LOGIN



USER HOMEPAGE

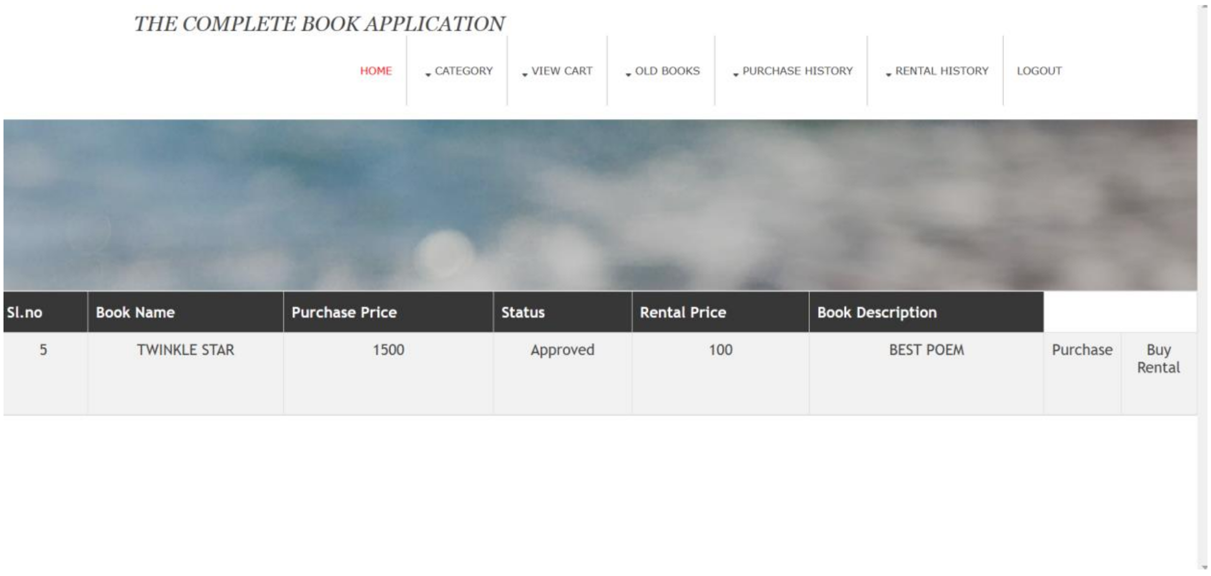


USER VIEW CATEGORY



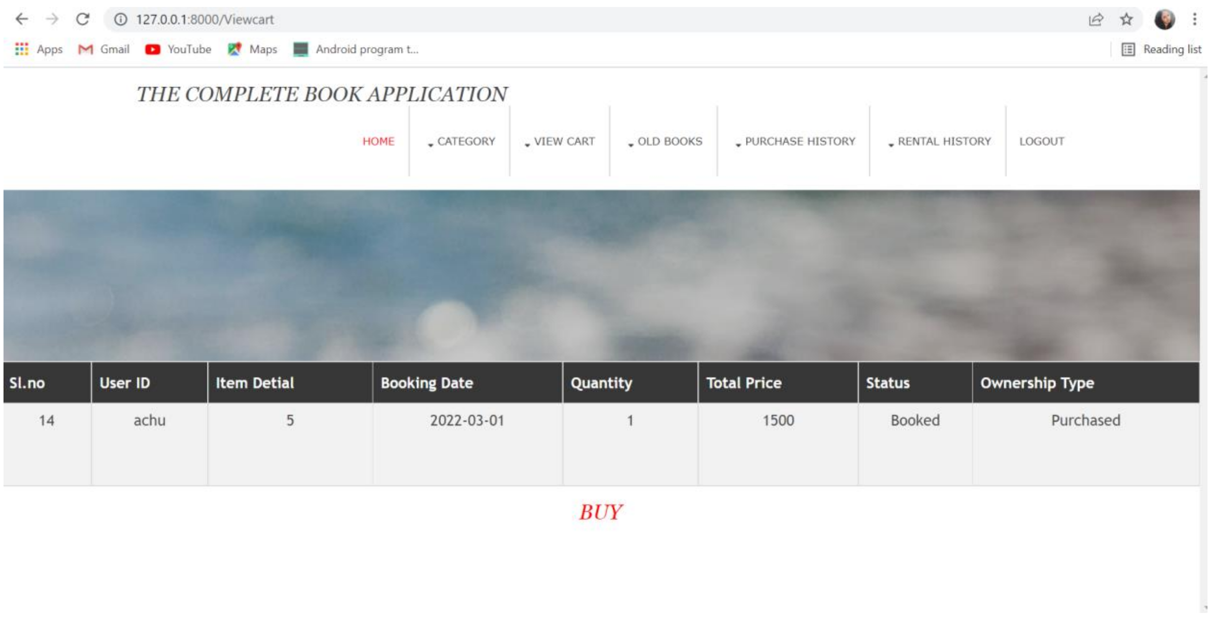
Sl.no	Name		
1	story	VIEW	ADD
3	comedy	VIEW	ADD
6	poem	VIEW	ADD

USER VIEW BOOKS



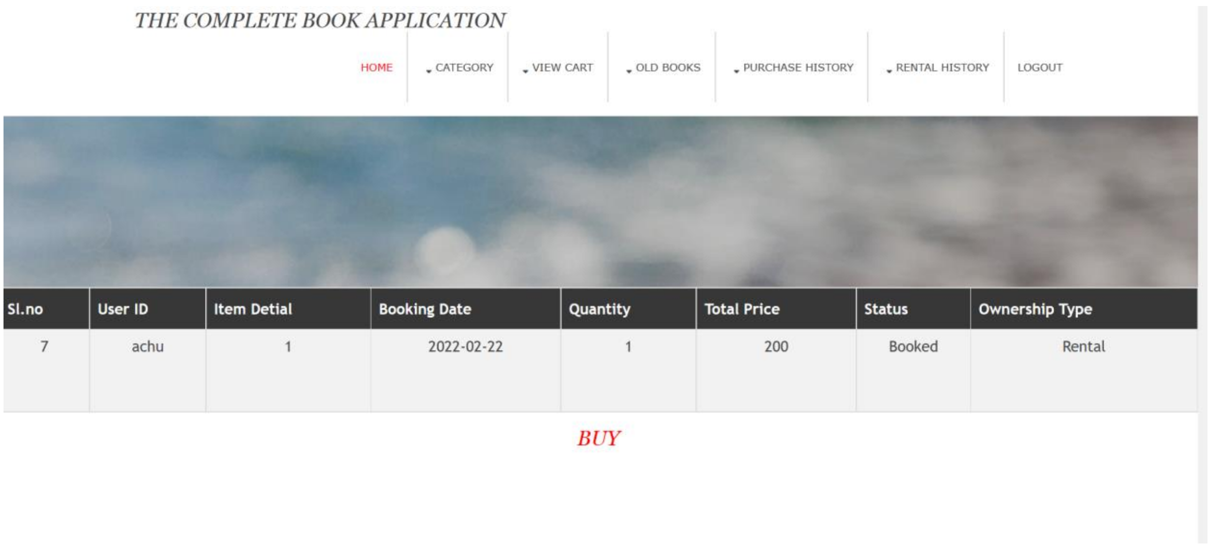
Sl.no	Book Name	Purchase Price	Status	Rental Price	Book Description		
5	TWINKLE STAR	1500	Approved	100	BEST POEM	Purchase	Buy Rental

USER BUY BOOKS



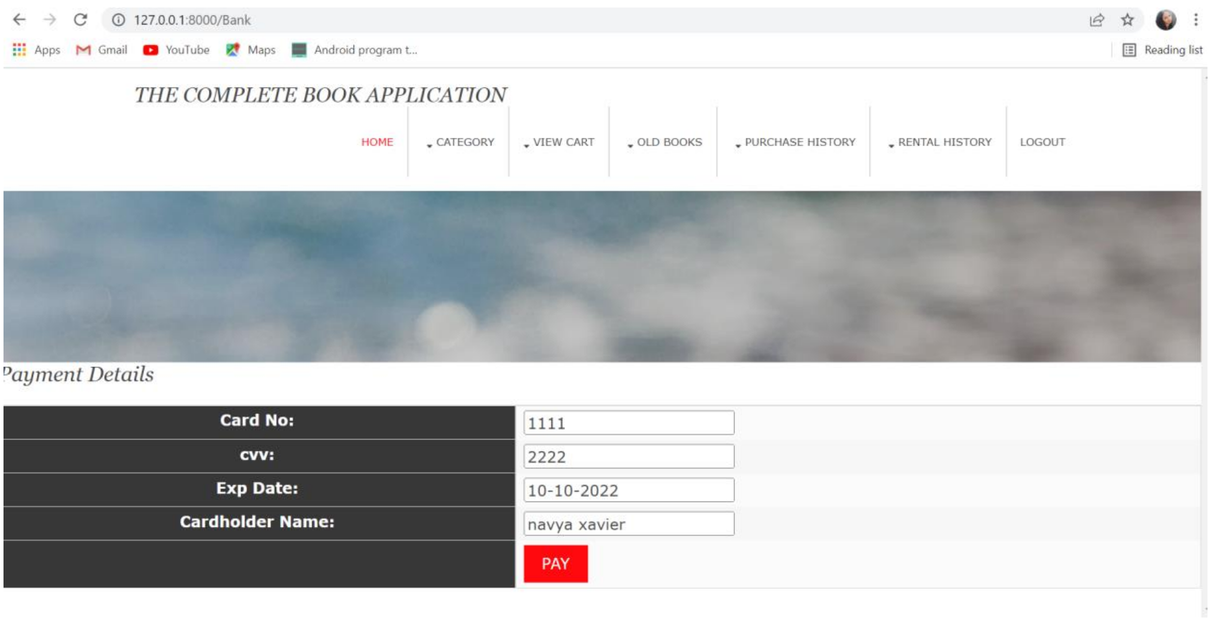
Sl.no	User ID	Item Detial	Booking Date	Quantity	Total Price	Status	Ownership Type
14	achu	5	2022-03-01	1	1500	Booked	Purchased

USER BORROW BOOKS



Sl.no	User ID	Item Detial	Booking Date	Quantity	Total Price	Status	Ownership Type
7	achu	1	2022-02-22	1	200	Booked	Rental

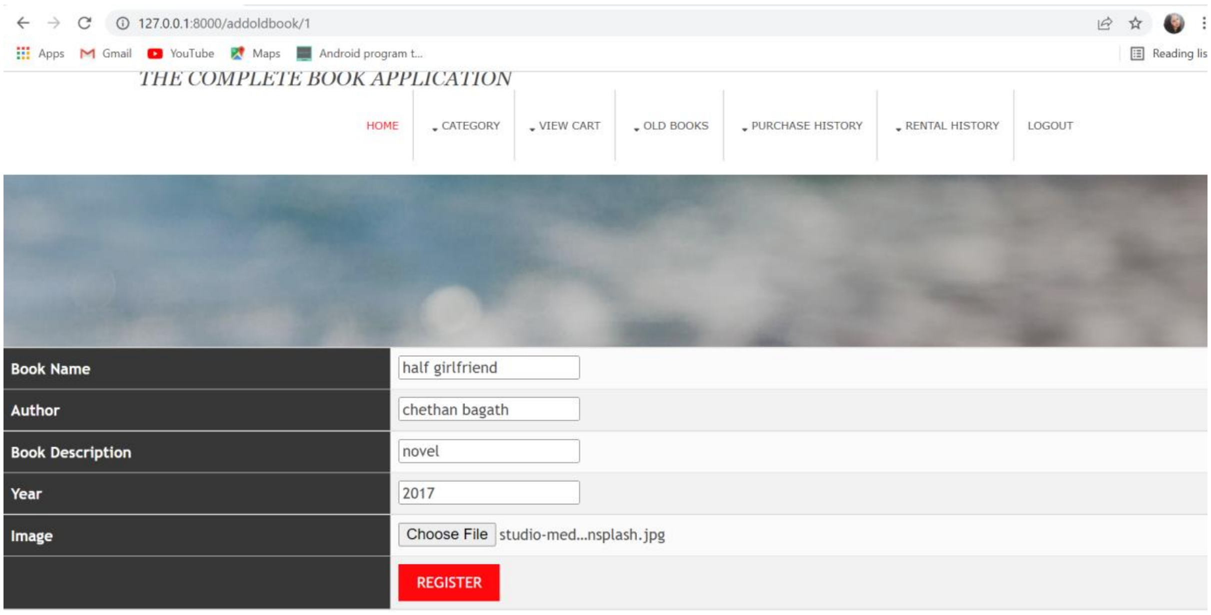
USER PAYMENT FOR BUY AND BORROW



Payment Details

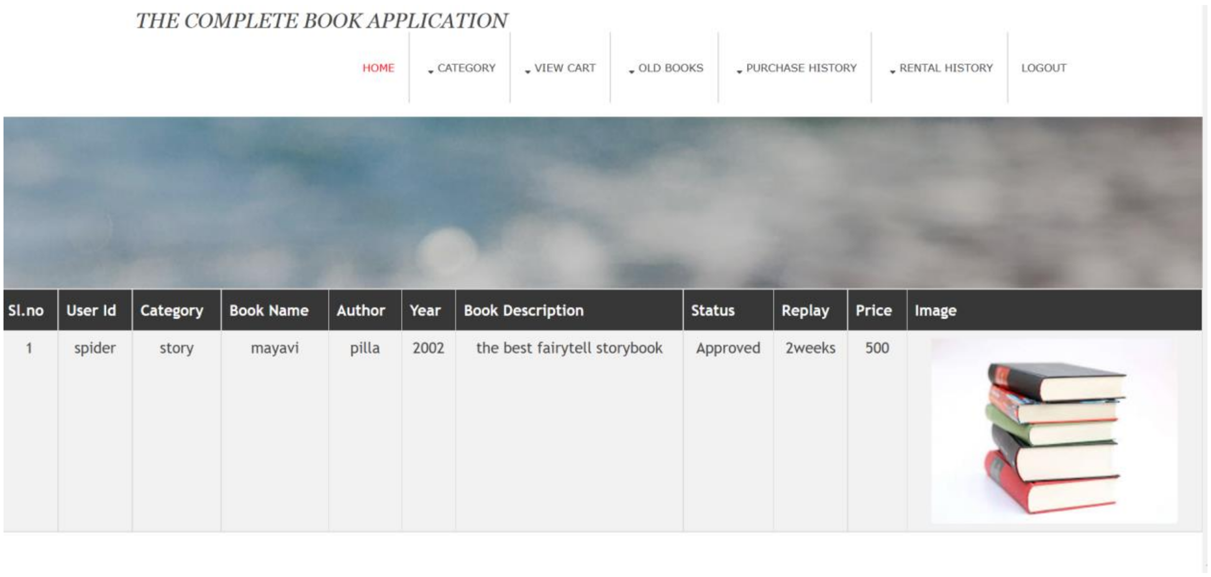
Card No:	1111
cvv:	2222
Exp Date:	10-10-2022
Cardholder Name:	navya xavier

USER ADD BOOKS FOR SELLING



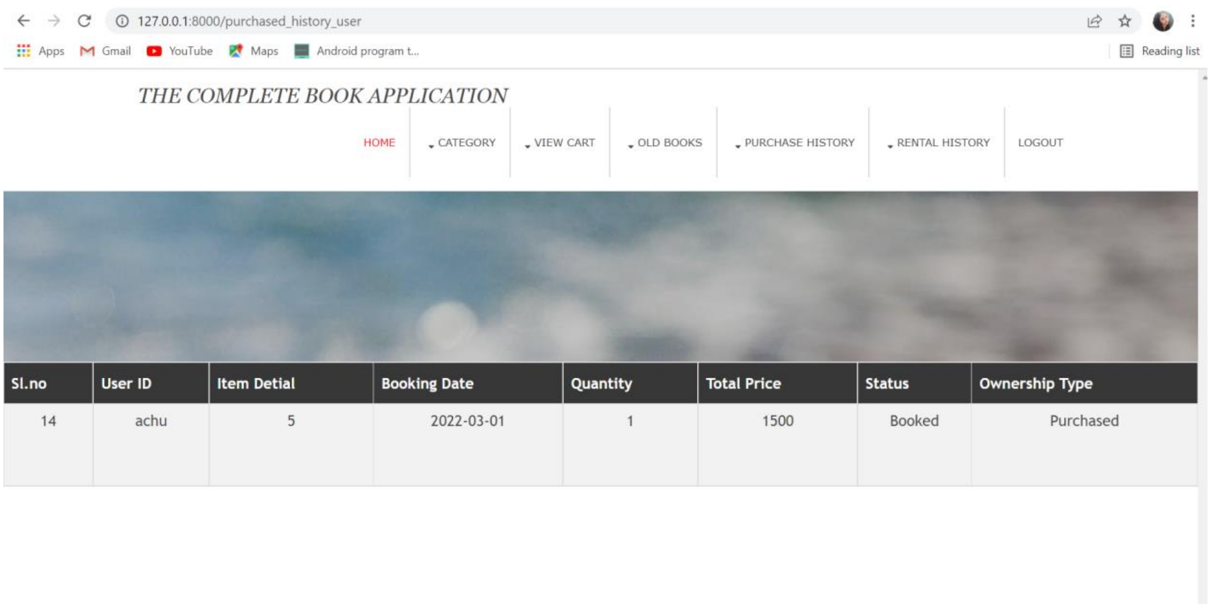
Book Name	half girlfriend
Author	chethan bagath
Book Description	novel
Year	2017
Image	Choose File studio-med...nsplash.jpg

USER CAN VIEW THE APPROVED REQUEST



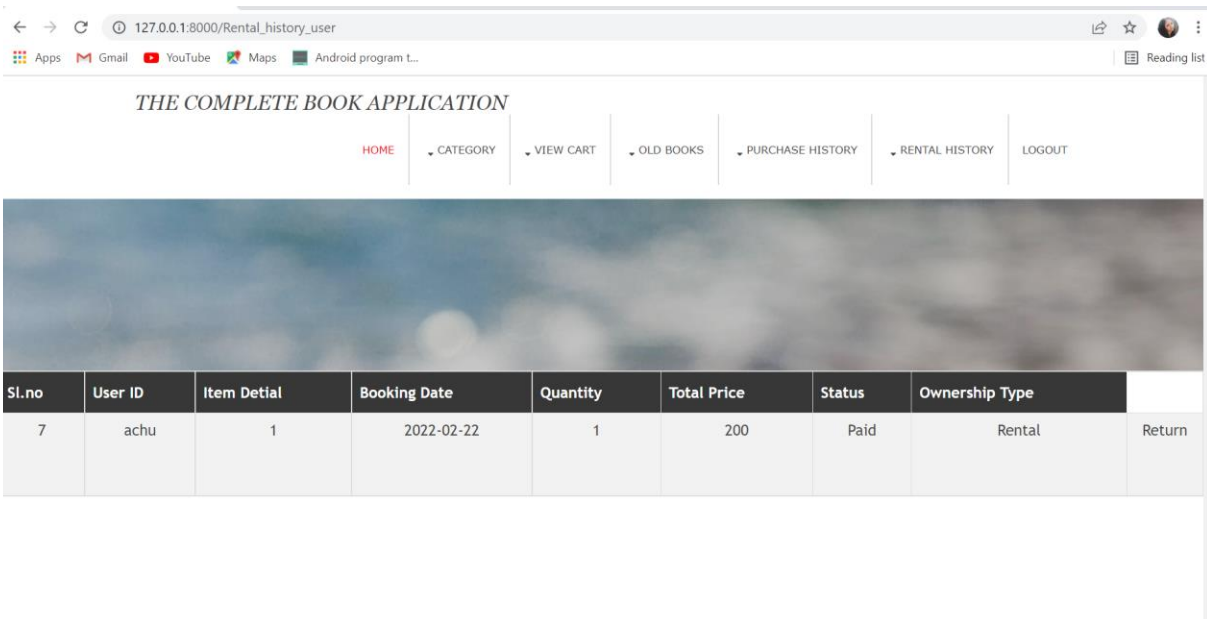
Sl.no	User Id	Category	Book Name	Author	Year	Book Description	Status	Replay	Price	Image
1	spider	story	mayavi	pilla	2002	the best fairytell storybook	Approved	2weeks	500	

USER CAN VIEW BUY ORDERS



Sl.no	User ID	Item Detial	Booking Date	Quantity	Total Price	Status	Ownership Type
14	achu	5	2022-03-01	1	1500	Booked	Purchased

USER CAN VIEW RENTAL ORDERS



Sl.no	User ID	Item Detial	Booking Date	Quantity	Total Price	Status	Ownership Type	
7	achu	1	2022-02-22	1	200	Paid	Rental	Return

9.2 CODE

Views.py

```
import mimetypes
import os
from datetime import date
from urllib import request

from django.db import connection
from django.http import HttpResponse, FileResponse
from django.shortcuts import render, redirect
from django.template.smartif import key


from bookapp.models import item_details, category, user_old_books


def index(request):
    return render(request,'HomePage.html')
```

```
def AdminHomePage(request):
    return render(request, 'Admin/AdminHomePage.html')


def UserHome(request):
    return render(request, 'User/UserHome.html')


def login(request):
    return render(request,"login.html")


def login1(request):
    if request.method == "POST":
        name = request.POST['un']
        password = request.POST['pass']
        request.session['lid']=name
        cursor = connection.cursor()
        cursor.execute ("select * from login where admin_id='" + name + "' and 
password='" + password + "'")
        print("select * from login where admin_id='" + name + "' and password='" + 
password + "'")
        pins = cursor.fetchone()
        flag='error'
        if pins == None:
            print("not admin")
            cursorF = connection.cursor()
            cursorF.execute("select user_id from user_register where user_id='" + 
name + "' and password='" + password + "'")
            print("select user_id from user_register where user_id='" + name + "' 
and password='" + password + "'")
            res = cursorF.fetchall()
            if res is not None:
                n = res[0]
                x = n[0]
                print(x)
                request.session['sid'] = x
                return redirect("/UserHome")
            return 
HttpResponse("<script>alert('invalid');window.location='/';</script>")
        else:
            flag="admin"
            print("this is admin")
    print("flag is:"+flag)
    if flag=="admin":
        return redirect("/AdminHomePage")
    if flag=="res":
        return redirect("/UserHome")
    if flag == "error":
        return 
HttpResponse("<script>alert('invalid');window.location='login';</script>")

    return 
HttpResponse("<script>alert('invalid');window.location='login';</script>")

def logout(request):
    return render(request,'logout.html')
```

```
def addCategory(request):
    if request.method == "POST":
        name = request.POST['txtname']
        data = category(name=name)
        data.save()
        return HttpResponse("<script>alert('Category
Added');window.location='/AdminHomePage';</script>")
    return render(request,"Admin/addCategory.html")


def viewCategory(request):
    cursor = connection.cursor()
    cursor.execute("select * from category")
    pin = cursor.fetchall()
    print(pin)
    return render(request,"Admin/viewCategory.html",{'data':pin})



def editcategory(request,id):
    pro_data=category.objects.get(id=id)
    print(pro_data)
    return render(request,"Admin/editcategory.html",{'data':pro_data})

def categoryedit(request,id):
    if request.method == "POST":
        name = request.POST['name']
        cursor = connection.cursor()
        cursor.execute("update category set name='" + name + "' where   id  ='" +
str(id) + "'")
        return
HttpResponse("<script>alert('Updated');window.location='/viewCategory';</script>")
    return render(request,"Admin/editcategory.html")


def deleteCategory(request,id):
    cursor = connection.cursor()
    cursor.execute ("delete from category where id='"+str(id)+"'")
    return HttpResponse( "<script>alert('Deleted
Succesfully');window.location='/viewCategory';</script>")



def addBook(request,id):
    if request.method == "POST":
        category_id = id
        book_name=request.POST['book_name']
        purchase_price=request.POST['purchase_price']
        rental_price=request.POST['rental_price']
        book_description=request.POST['book_description']
        cursor = connection.cursor()
        cursor.execute ("insert into item_details values(null,'" +
str(category_id) + "',curdate(),'" + book_name + "','" + purchase_price +
"','Approved','" + rental_price + "','" + book_description + "')")
        return HttpResponse("<script>alert('Item
Added');window.location='/AdminHomePage';</script>")
```

```
    return render(request,"Admin/addBook.html")


def viewBookdetials(request):
    cursor = connection.cursor()
    cursor.execute("select c.name,item_details.* from item_details  join category
as c on c.id=item_details.category_id")
    pin = cursor.fetchall()
    return render(request,"Admin/viewBookdetials.html",{'data':pin})


def editBookdetial(request,id):
    pro_data=item_details.objects.get(id=id)
    print(pro_data)
    return render(request,"Admin/editBookdetial.html",{'data':pro_data})


def bookedit(request,id):
    if request.method == "POST":
        book_name = request.POST['book_name']
        purchase_price = request.POST['purchase_price']
        rental_price = request.POST['rental_price']
        book_description = request.POST['book_description']
        cursor = connection.cursor()
        cursor.execute("update item_details set book_name='" + book_name +
"',purchase_price='" + purchase_price + "',rental_price='" + rental_price +
"',book_description='" + book_description + "' where  id  ='" + str(id) + "'")
        return
HttpResponse("<script>alert('Updated');window.location='/viewBookdetials';</script
>")
    return render(request,"Admin/editBookdetial.html")


def deletebook(request,id):
    cursor = connection.cursor()
    cursor.execute ("delete from item_details where id='"+str(id)+"'")
    return HttpResponse( "<script>alert('Deleted
Succesfully');window.location='/viewBookdetials';</script>")


def viewoldbookrequest(request):
    cursor = connection.cursor()
    cursor.execute("select c.name,user_old_books.* from user_old_books  join
category as c on c.id=user_old_books.category_id ")
    pin = cursor.fetchall()
    return render(request,"Admin/viewoldbookrequest.html",{'data':pin})


def approverequest(request,id):
    cursor = connection.cursor()
    cursor.execute("update user_old_books set status='Approved' where
iduser_old_books='" + str(id) + "'")
    return redirect("/viewoldbookrequest")



def purchased_history(request):
    cursor = connection.cursor()
    cursor.execute("select * from user_bookings where ownership_type='Purchased'
")
    pin = cursor.fetchall()
    return render(request,"Admin/purchased_history.html",{'data':pin})



def Rental_history(request):
    cursor = connection.cursor()
    cursor.execute("select * from user_bookings where ownership_type='Rental' ")
    pin = cursor.fetchall()
    return render(request,"Admin/Rental_history.html",{'data':pin})


#-----------------------------------------------------------User------------------
----------------------------------------------------------#


def adduser(request):
    if request.method == "POST":
        user_id = request.POST['seller_id']
        first_name = request.POST['first_name']
        last_name = request.POST['last_name']
        address = request.POST['TxtAddress']
        phone = request.POST['TxtPhone']
        email = request.POST['TxtEmail']
        city = request.POST['city']
        postcode = request.POST['postcode']
        country = request.POST['country']
        state = request.POST['state']
        password = request.POST['password']
        cursor = connection.cursor()
        cursor.execute ("insert into user_register values('" + user_id + "','" +
first_name + "','" + last_name + "','" + address + "','" + phone + "','" + email +
"','" + city + "','" + postcode + "','" + country + "','" + state + "','" +
password + "')")
        return
HttpResponse("<script>alert('Registered');window.location='/login';</script>")
    return render(request,"User/adduser.html")


def viewCategoryUser(request):
    cursor = connection.cursor()
    cursor.execute("select * from category")
    pin = cursor.fetchall()
    print(pin)
    return render(request,"User/viewCategoryUser.html",{'data':pin})



def viewBookUser(request,id):
    cursor = connection.cursor()
    cursor.execute("select * from item_details where category_id='"+str(id)+"'")
    pin = cursor.fetchall()
    return render(request,"User/viewBookUser.html",{'data':pin})



def Purchase(request,id,pid):
    uid=request.session['sid']
    n=str(pid)
    if request.method == "POST":
        user_id=uid
        iditem_details=id
        quantity=request.POST['quantity']
        total_price=int(quantity)*int(n)
        cursor = connection.cursor()
        cursor.execute ("insert into user_bookings values(null,'" + str(user_id) +
"','" + str(iditem_details) + "',curdate(),'" + quantity + "','" +
str(total_price) + "','Booked','Purchased')")
        return HttpResponse("<script>alert('ADDED TO
CART');window.location='/UserHome';</script>")
    return render(request,"User/Purchase.html")



def Rental(request,id,rid):
    uid=request.session['sid']
    n=str(rid)
    if request.method == "POST":
        user_id=uid
        iditem_details=id
        quantity=request.POST['quantity']
        total_price=int(quantity)*int(n)
        cursor = connection.cursor()
        cursor.execute ("insert into user_bookings values(null,'" + str(user_id) +
"','" + str(iditem_details) + "',curdate(),'" + quantity + "','" +
str(total_price) + "','Booked','Rental')")
        return HttpResponse("<script>alert('Item
Added');window.location='/UserHome';</script>")
    return render(request,"User/Rental.html")


def Viewcart(request):
    uid = request.session['sid']
    cursor = connection.cursor()
    cursor.execute("select * from user_bookings where user_id='"+str(uid)+"' and
status='Booked' ")
    pin = cursor.fetchall()
    return render(request,"User/Viewcart.html",{'data':pin})



def Bank(request):
    uid = request.session['sid']
    if request.method == "POST":
        card_no =request.POST['card_no']
        card_cvv = request.POST['cvv']
        card_holder_name =request.POST['card_holder_name']
        card_expiry_date =request.POST['card_expiry_date']
        cursor = connection.cursor()
        cursor.execute ("select * from account_details where card_no='" + card_no
+ "' and card_cvv='" + card_cvv + "' and card_holder_name='" + card_holder_name +
"' and card_expiry_date='" + card_expiry_date + "' ")
        cursor.execute("update user_bookings set status='Paid' where user_id='" +
str(uid) + "'")
        return
HttpResponse("<script>alert('PAIDED');window.location='/UserHome';</script>")
    return render(request,"User/Bank.html")
```

```
def addoldbook(request,id):
    uid=request.session['sid']
    if request.method == "POST":
        user_id=uid
        category_id=str(id)
        book_name = request.POST['book_name']
        author = request.POST['author']
        year = request.POST['year']
        book_description = request.POST['book_description']
        file_path =request.FILES['file_path']
        data =
user_old_books(user_id=user_id,category_id=category_id,book_name=book_name,author=
author,year=year,book_description=book_description,file_path=file_path)
        data.status='request'
        data.reply='pending'
        data.price='pending'
        data.save()
        return
HttpResponse("<script>alert('Added');window.location='/UserHome';</script>")
    return render(request,"User/addoldbook.html")


def viewoldbook(request):
    uid = request.session['sid']
    cursor = connection.cursor()

    cursor.execute("SELECT c.name,user_old_books.* FROM  user_old_books join
category as c on c.id=user_old_books.category_id where
user_old_books.user_id='"+str(uid)+"' ")
    pin = cursor.fetchall()
    return render(request,"User/viewoldbook.html",{'data':pin})




def purchased_history_user(request):
    uid = request.session['sid']
    cursor = connection.cursor()
    cursor.execute("select * from user_bookings where ownership_type='Purchased'
and user_id='"+str(uid)+"' ")
    pin = cursor.fetchall()
    return render(request,"User/purchased_history_user.html",{'data':pin})


def Rental_history_user(request):
    uid = request.session['sid']
    cursor = connection.cursor()
    cursor.execute("select * from user_bookings where ownership_type='Rental' and
user_id='"+str(uid)+"' ")
    pin = cursor.fetchall()
    return render(request,"User/Rental_history_user.html",{'data':pin})


def returns(request,id):
    cursor = connection.cursor()
    cursor.execute("update user_bookings set status='return' where
iduser_bookings='" + str(id) + "'")
    return redirect("/Rental_history_user")
```

urls.py

```
from django.contrib import admin
from django.urls import path, include

from bookapp import views

urlpatterns = [
    path('', views.index, name='index'),
    path('login', views.login, name='login'),
    path('login1', views.login1, name='login1'),

    path('logout', views.logout, name='logout'),
    path('AdminHomePage', views.AdminHomePage, name='AdminHomePage'),
    path('UserHome', views.UserHome, name='UserHome'),
    path('addCategory', views.addCategory, name='addCategory'),
    path('viewCategory', views.viewCategory, name='viewCategory'),
    path('editcategory/<int:id>', views.editcategory, name='editcategory'),
    path('categoryedit/<int:id>', views.categoryedit, name='categoryedit'),
    path('deleteCategory/<int:id>', views.deleteCategory, name='deleteCategory'),
    path('addBook/<int:id>', views.addBook, name='addBook'),
    path('viewBookdetials', views.viewBookdetials, name='viewBookdetials'),
    path('editBookdetial/<int:id>', views.editBookdetial, name='editBookdetial'),
    path('bookedit/<int:id>', views.bookedit, name='bookedit'),
    path('deletebook/<int:id>', views.deletebook, name='deletebook'),
    path('viewoldbookrequest', views.viewoldbookrequest,
name='viewoldbookrequest'),
    path('approverequest/<int:id>', views.approverequest, name='approverequest'),
    path('purchased_history', views.purchased_history, name='purchased_history'),
    path('Rental_history', views.Rental_history, name='Rental_history'),
    path('adduser', views.adduser, name='adduser'),
    path('viewCategoryUser', views.viewCategoryUser, name='viewCategoryUser'),
    path('viewBookUser/<int:id>', views.viewBookUser, name='viewBookUser'),
    path('Purchase/<int:id>/<str:pid>', views.Purchase, name='Purchase'),
    path('Rental/<int:id>/<str:rid>', views.Rental, name='Rental'),
    path('Viewcart', views.Viewcart, name='Viewcart'),
    path('Bank', views.Bank, name='Bank'),
    path('addoldbook/<int:id>', views.addoldbook, name='addoldbook'),
    path('viewoldbook', views.viewoldbook, name='viewoldbook'),
    path('purchased_history_user', views.purchased_history_user,
name='purchased_history_user'),
    path('Rental_history_user', views.Rental_history_user,
name='Rental_history_user'),
    path('returns/<int:id>', views.returns, name='returns'),


]
```