Department of IT and Computer Science

Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur, Pakistan

Project Proposal

Submitted to

Dr. Ihtisham Ali
Operating Systems

Mr. Syed Ahsan
Operating Systems

Submitted By

Ahmed Raza	B20F0436CS031
Saad Bin Khalid	B20F0247CS010
Muhammad Waseem Faiz	B20F0286CS016

Contents

Project Proposal 1
Submitted to 1
Submitted By 1
Project Description 3
Display Menu: 3
Add Book: 3
Display Books: 4
Buy Book: 6
Modify Book: 6
Add Customer: 7
Display Customers: 8
Delete Book: 9
Delete All Data: 10
Generate Receipt: 10
Exit: 10
Conclusion 11

ABSTRACT

Project Description

Our project is Book Store Management System (BSMS) in shell script. The script provides several functions for managing a book inventory, including adding books, displaying books, buying books, modifying books, adding customer details, displaying customer details, deleting books, deleting all data, generating receipts, and exiting the program. The script uses MongoDB as the database for storing the book inventory and customer details. The script interacts with the database using the mongosh command-line tool. The functions in the script perform various operations on the database, such as adding books, finding books by ISBN number, updating book details, and counting the number of books in the inventory.

Display Menu: This function displays a menu of options for the user to choose from.
The menu options include:

- Add a book
- Display all books
- Buy a book
- Modify a book
- Add customer details
- Display all customers
- Delete a book
- Delete all data
- Generate receipt
- Exit The function uses the **echo** command to display the menu and the **read** command to prompt the user for a **choice**. The user's choice is stored in a variable named choice.

Add Book: This function allows the user to add a new book to the inventory.
The function prompts the user to enter the following details for the new book:

- Title
- Author
- Price
- Stock
- ISBN number

The function uses the MongoDB shell (**mongosh**) to add the book to the "Inventory" collection in the "BSMS" database. The details of the book are stored as a JSON document, with each field (e.g., title, author, price, etc.) as a key-value pair. The **insertOne** method is used to add the document to the collection.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 1
Enter the book's title: OS
Enter the book's author: Ihtesham
Enter the book's price: 500
Enter the book's stock: 20
Enter the book's ISBN number: 811
{
  acknowledged: true,
  insertedId: ObjectId("63d78cf033183574d7793f0b")
}
Book added successfully!
```

Display Books: This function displays all of the books in the inventory. The function uses the MongoDB shell to retrieve all documents from the "Inventory" collection in the "BSMS" database. The find method is used to retrieve the documents, and the pretty method is used to display the output in a nicely formatted way.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 2
Current Mongosh Log ID: 63d78c493fa3bb37bc88a927
Connecting to:          mongodb://127.0.0.1:27017/BSMS?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.1
Using MongoDB:          6.0.3
Using Mongosh:          1.6.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2022-12-26T23:17:05.484+05:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
   2022-12-26T23:17:06.197+05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2022-12-26T23:17:06.198+05:00: vm.max_map_count is too low
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------
```

```
[
  {
    _id: ObjectId("63d6f2ac0d50e115ec13571c"),
    title: 'Harry Porter',
    author: 'JK Rowling',
    Price: '999',
    Stock: '21',
    isbn: '909',
    'Book ID': ''
  },
  {
    _id: ObjectId("63d6f2cec4eb838df5753886"),
    title: 'Legends of Tomorrow',
    author: 'John Nolan',
    Price: '1000',
    Stock: '10',
    isbn: '213',
    'Book ID': ''
  },
  {
    _id: ObjectId("63d6fc7bdcf6335150ea0f91"),
    title: 'Percy Jackson',
    author: 'Bale',
    Price: '799',
    Stock: '18',
    isbn: '880',
    'Book ID': ''
  },
  {
    _id: ObjectId("63d6fe55182cbb037cff2a1a"),
    title: 'Beind the Door',
    author: 'John',
    Price: '799',
    Stock: '21',
    isbn: '989',
    'Book ID': ''
  }
]
```

Buy Book: This function allows the user to buy a book by ISBN number. The function prompts the user to enter the ISBN number of the book they want to buy. The function then uses the MongoDB shell to check if a book with the specified ISBN number exists in the "Inventory" collection. If the book does not exist, an error message is displayed. If the book exists, the function retrieves the current stock of the book. If the stock is 0, an error message is displayed. If the stock is greater than 0, the function prompts the user to enter the number of copies they want to buy. If the number of copies requested is greater than the stock, an error message is displayed. If the number of copies requested is less than or equal to the stock, the stock is updated, and a success message is displayed.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 3
Enter the ISBN number of the book you want to buy: 989
Enter the number of copies you want to buy: 12
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Success: You have successfully bought 12 copy/copies of the book with ISBN number 989.
```

Modify Book: This function allows the user to modify the details of a book in the inventory. The function prompts the user to enter the ISBN number of the book they want to modify. The function then uses the MongoDB shell to check if a book with the specified ISBN number exists in the "Inventory" collection. If the book does not exist, an error message is displayed. If the book exists, the function retrieves the current details of the book (title, author, price, and stock). The function then prompts the user to enter the updated details for each field. If the user does not want to update a field, they can simply press Enter. The function uses the MongoDB shell to update the document in the "Inventory" collection with the updated details.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 4
Enter the ISBN number of the book you want to modify: 989
Enter the updated title [Beind the Door]: Next Door
Enter the updated author [John]: John
Enter the updated price [799]: 600
Enter the updated stock [9]: 19
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Success: The details of the book with ISBN number 989 have been updated.
```

Add Customer: This function allows the user to add a new customer to the list of customers. The function prompts the user to enter the following details for the new customer:

- Name
- Address
- Phone number
- Email

The function uses the MongoDB shell to add the customer to the "Customers" collection in the "BSMS" database. The details of the customer are stored as a JSON document, with each field (e.g., name, address, phone number, etc.) as a key-value pair. The insertOne method is used to add the document to the collection.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 5
Enter the customer's name: Waseem Faiz
Enter the customer's CNIC: 6376187366
Enter the book the customer wants to buy: OS
Enter the customer's address: Pak Austria
{
  acknowledged: true,
  insertedId: ObjectId("63d78effdd595b41457ac05f")
}
Customer added successfully!
```

Display Customers: This function displays all of the customers in the list. The function uses the MongoDB shell to retrieve all documents from the "Customers" collection in the "BSMS" database. The find method is used to retrieve the documents, and the pretty method is used to display the output in a nicely formatted way.

```
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 6
Current Mongosh Log ID: 63d78f5890ec8eed6ca1e44f
Connecting to:          mongodb://127.0.0.1:27017/BSMS?directConnection=true&serverSelectionTimeoutMS=2000&appName=mong
Using MongoDB:          6.0.3
Using Mongosh:          1.6.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2022-12-26T23:17:05.484+05:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine.
   2022-12-26T23:17:06.197+05:00: Access control is not enabled for the database. Read and write access to data and con
   2022-12-26T23:17:06.198+05:00: vm.max_map_count is too low
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

[
  {
    _id: ObjectId("63d78effdd595b41457ac05f"),
    name: 'Waseem Faiz',
    CNIC: '6376187366',
    book: '',
    address: 'Pak Austria'
  }
]
```

Delete Book: This function allows the user to delete a book from the inventory. The function prompts the user to enter the ISBN number of the book they want to delete. The function then uses the MongoDB shell to check if a book with the specified ISBN number exists in the "Inventory" collection. If the book does not exist, an error message is displayed. If the book exists, the function uses the deleteOne method to delete the document from the "Inventory" collection.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 7
Enter the ISBN number of the book you want to delete: 811
{ acknowledged: true, deletedCount: 1 }
Success: Book with ISBN number 811 has been deleted from the inventory.
```

Delete All Data: This function allows the user to delete all books and customer details from the "BSMS" database. The function uses the MongoDB shell to delete all documents from both the "Inventory" and "Customers" collections in the "BSMS" database.

Generate Receipt: This function generates a receipt for a book purchase. The function prompts the user to enter the ISBN number of the book they want to buy. The function then uses the MongoDB shell to check if a book with the specified ISBN number exists in the "Inventory" collection. If the book does not exist, an error message is displayed. If the book exists, the function retrieves the details of the book (title, author, price, and stock) and prompts the user to enter their name and address. The function then generates a receipt in the form of a JSON document, which includes the details of the book and the customer, as well as the date and time of the purchase. The receipt is stored in the "Receipts" collection in the "BSMS" database.

```
Book Store Management System
1. Add a book
2. Display all books
3. Buy a book
4. Modify a book
5. Add customer details
6. Display all customers
7. Delete a book
8. Delete all data
9. Generate receipt
10. Exit
Enter your choice: 9
Enter the ISBN number of the book: 989
Enter the number of copies you want to buy: 3
----------------------------
Book Store Management System
----------------------------

Book Title: Next Door
Author: John
ISBN: 989
Number of Copies: 3
Price per Copy: 600
Total Price: 1800
```

Exit: This function simply exits the program. The function uses the exit command to terminate the shell script.

Conclusion

In conclusion, the project is a simple implementation of a bookstore management system that uses a MongoDB database to store and manage the information of books and customers. The system provides a menu-driven interface that allows the user to perform various tasks, such as adding a new book, displaying all books, buying a book, modifying a book, adding customer details, displaying all customers, deleting a book, deleting all data, generating a receipt, and exiting the program.

From this project, we have learned how to interact with a MongoDB database using the MongoDB shell and how to perform basic database operations such as inserting, updating, and retrieving documents. We have also learned how to create a menu-driven interface using shell scripts and how to perform basic input and output operations in shell scripts.

In addition, the project has provided us with a simple example of how to design and implement a basic database-driven application, which is a common requirement in many real-world scenarios. This project has also shown the importance of proper database design and the use of a database management system in maintaining the consistency and integrity of data in a system.