BOOK INVENTORY MANAGEMENT SYSTEM

PROJECT REPORT

ICT502

DATABASE ENGINEERING

SEMESTER OCTOBER 2022 – FEBRUARY 2023

GROUP: M3CS2303D

AMIRUL ADLI FAHMI BIN AZAM (2022487626)
AYU NATASYA FARISAH BINTI FAIZUL (2022494566)
NURUL SYAFIQAH NADIA BINTI HUSAIN (2022463912)
MOHAMAD AZFAR SYAZANI BIN MD YUSOF (2022645724)
MUHAMMAD AZRI BIN NAZIR (2022842452)

SIR MUHAMMAD HAMIZ MOHD RADZI
FSKM, UiTM JASIN

Table of Contents

1.0 Company Background

BookXcess, which first operated in 2007, has reinvigorated and redefined bookselling in Malaysia and beyond, offering an unrivaled selection of reasonably priced books ranging from classic novels to children's pop-ups to bestselling self-help titles.

Their mission is to create, inspire, and empower readers, as well as to instill the habit of reading by making books accessible and affordable to all.

They deliver millions of books to readers worldwide through their seamless digital and retail experience, and they are dynamic, creative, and innovative, with a rapidly growing network of ground-breaking and inspirational stores.

While rapidly growing in business, they realized they need a better approach in managing all the books that they have in stock. Thus, a book inventory management system is the answer for them to manage their stocks and suppliers in a decent and practical way.

2.0 Case Study

2.1 Problem Statement

The current system used by the organization is a file-based system, which is inefficient and lacks performance, leading to several problems.

I. Lack of security

The systems used by the organization lack security. Data should be accessible to the user by his requirements only. For example, suppliers can’t see the details or data of staff like their salary. This is supposed to be avoided as it is confidential information. The system also didn’t have tight security, which will lead to stolen data. This can be a threat to the organization.

II. Data redundancy

Besides that, the organization also has a problem with data redundancy. Since the current system used by the organization relies on text instead of structural data, any data that wants to be updated will need to be done manually. It is possible that the same information may be duplicated in different files. This leads to data redundancy resulting in memory wastage. Because of data redundancy, it is possible that data may not be in a consistent state. For example, if one file contains an address record of Staff A, another file that uses address information on Staff A must recreate that data. This means that the address data on Staff A exists in two files at once.

III. Limited user access

Next, the current problem faced by the organization is having limited user access. This means that multiple users at different workstations cannot access the same data simultaneously, access to important data will be limited if multiple users search for the same data at the same time. For example, staff A which is in the workplace want to see a record of resident A, but staff B which is currently in another workstation also want to see the same record, because the record or the data has only one copy for each of it, data need to be shared by scanning the data or snapshot and send it personally to staff B, which is inconvenient.

IV. Data loss

Furthermore, data loss also might occur. File systems usually are not backup so it will be hard to recover. For example, natural disasters such as floods might happen, and this will destroy the file of data as data is recorded manually.

2.2 Objective

By developing the system, we can solve the problems which are affecting the organization.

I. Improve data security

This system is designed to increase data security. For the user to access the system, they will need an email and password. For example, the staff will need to sign in using their email and password to make an update, store and view the data. This helps to control the limit of what every stage of the user can see. The data stored can also be encrypted to avoid unauthorized access. Encryption is the process of converting readable data into unreadable characters to make sure the data is safe and secured. By combining both methods, the possibility of data leaking will be much lower than using a manual system.

II. Data consistency

Furthermore, this system is developed to help the staff key in all the details and keep track of data needed for the organization. This can help in preventing data redundancy to happen. Using the manual system might create duplicate data as each data has its own file. However, by using the system, the whole data is stored only once in a single place so that there is no chance of data redundancy.

III. Easy data sharing

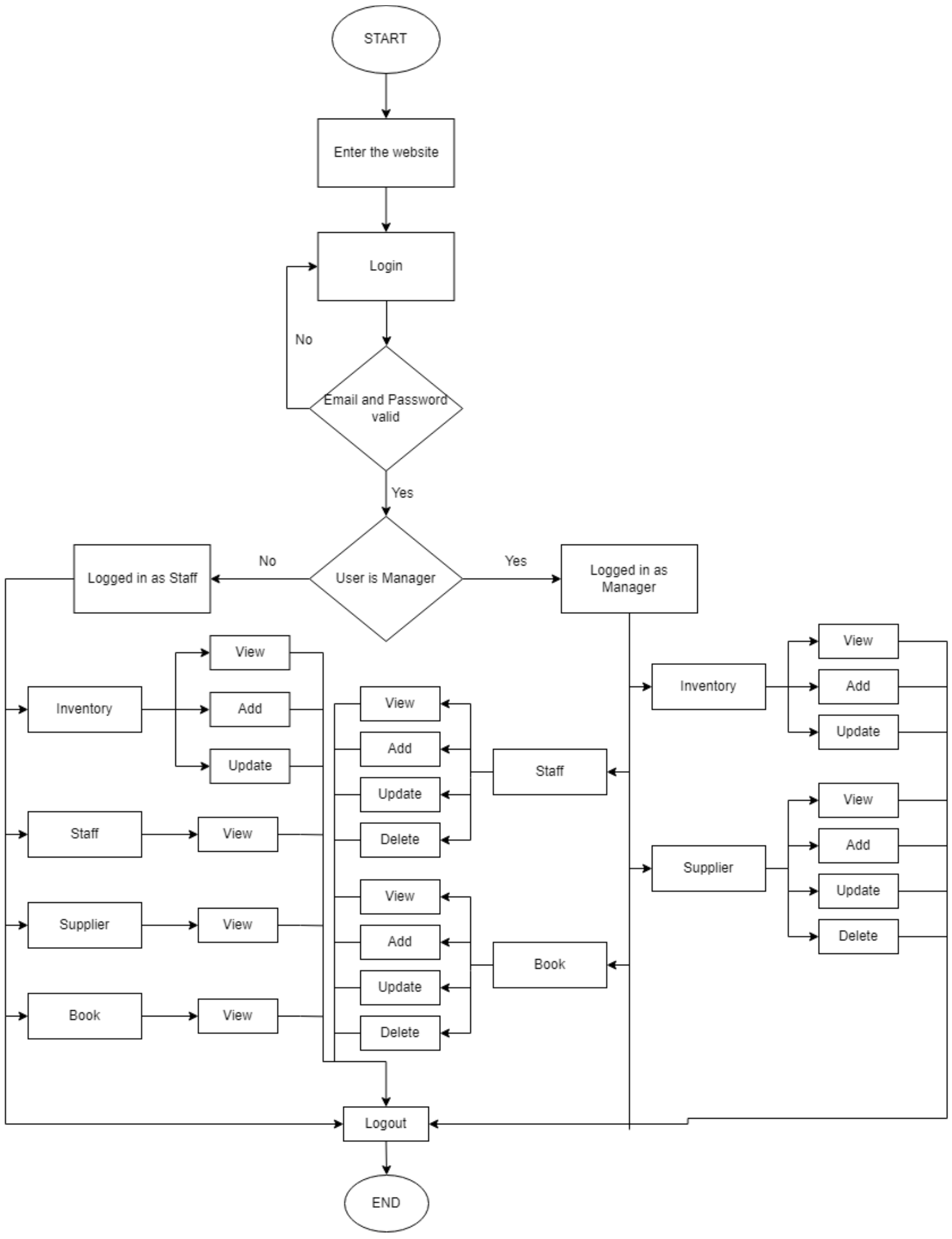
Moreover, web-based systems significantly simplify the exchange of data and project collaboration. Instead of having to redefine all the data needs, new applications can build on the existing data in the database and add the data that is not currently stored. This will result in significant time savings.

IV. Improved backup and recovery

The web-based system handles both backup and recovery automatically. Users are not required to do periodical backups as the system will handle it for them. To prevent a system failure or crash, it also restores a database to its prior state. The organization wouldn’t have to worry about losing data.

3.0 System Design

3.1 Flow Chart of System



3.2 10 SQL Queries

1) Displaying all data in table STAFF.

```
SELECT *
FROM STAFF;
```

2) Inserting data into table STAFF

```
INSERT INTO STAFF (staffid, first_name, last_name, phone_number, salary, hire_date,
password, position, supervisor_id, email, address)
VALUES (3, 'Azri', 'Doe', '1234567891', 60000, '01-OCT-2001', 'password', 'Manager',
NULL, 'azridoe@email.com', '123 Main St');
```

3) Updating data from table STAFF.

```
UPDATE STAFF
SET POSITION = 'Staff'
WHERE STAFFID = 3;
```

4) Joining two tabel to display Manager

```
SELECT *
FROM STAFF
JOIN MANAGER
ON STAFF.STAFFID = MANAGER.STAFFID;
```

5) Delete one supplier in table SUPPLIER

```
DELETE FROM SUPPLIER
WHERE SUPPLIER_ID = 4;
```

6) Inserting using sequence

```
INSERT INTO STAFF (staffid, first_name, last_name, phone_number, salary, hire_date,
password, position, supervisor_id, email, address)
VALUES (STAFF_ID_SEQ.NEXTVAL, 'Joe', 'Doe', '1234567891', 60000, '01-SEP-2001',
'password', 'Staff', 1, 'joedoe@email.com', '123 Main St');
```

7) Find book_price that is less than maximum book_price

```
SELECT BOOK_NAME, BOOK_AUTHOR, BOOK_PRICE
FROM BOOK
WHERE BOOK_PRICE <
                    (SELECT MAX(BOOK_PRICE)
                    FROM BOOK);
```

8) Find staff that hired after ‘01-SEP-2001’;

```
SELECT FIRST_NAME, HIRE_DATE
FROM STAFF
WHERE HIRE_DATE > TO_DATE('01-SEP-2001', 'DD-MON-YYYY');
```

9) Display all Staff with their Manager

```
SELECT S.FIRST_NAME "STAFF", M.FIRST_NAME "MANAGER"
FROM STAFF S
```

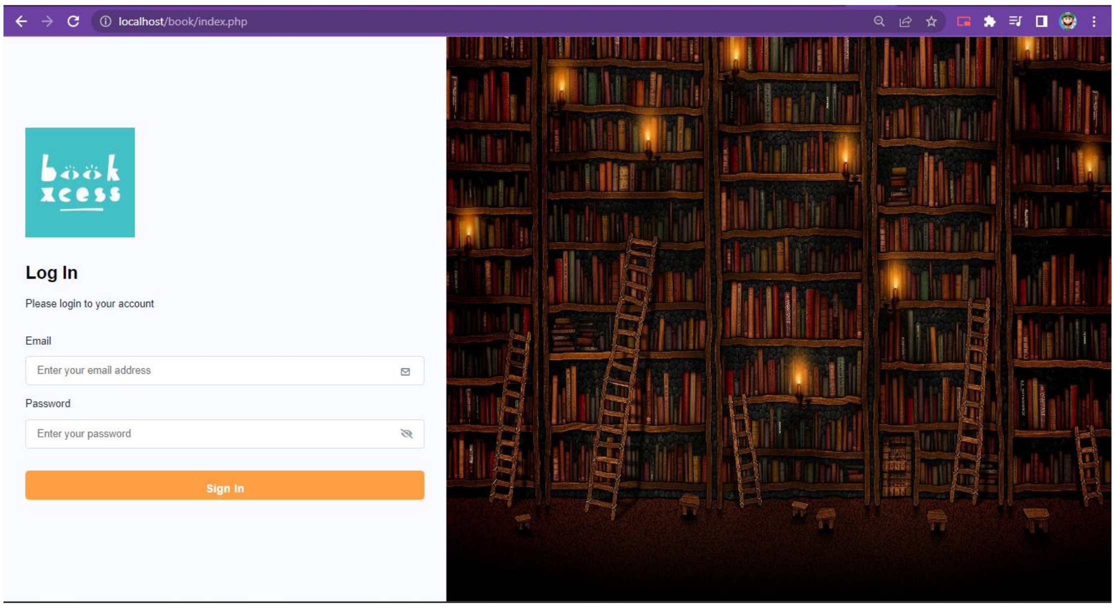
```
JOIN STAFF M
ON S.SUPERVISOR_ID = M.STAFFID;
```

10) MIN salary in table STAFF

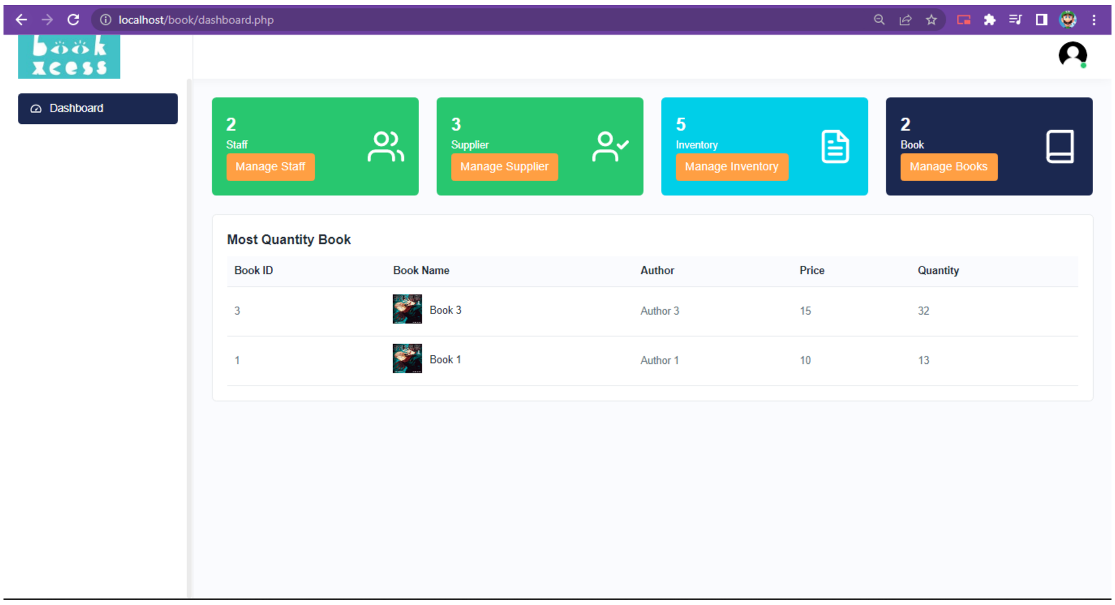
```
SELECT MIN(SALARY)
FROM STAFF;
```

3.3 System Development Sample Screen

LOGIN PAGE



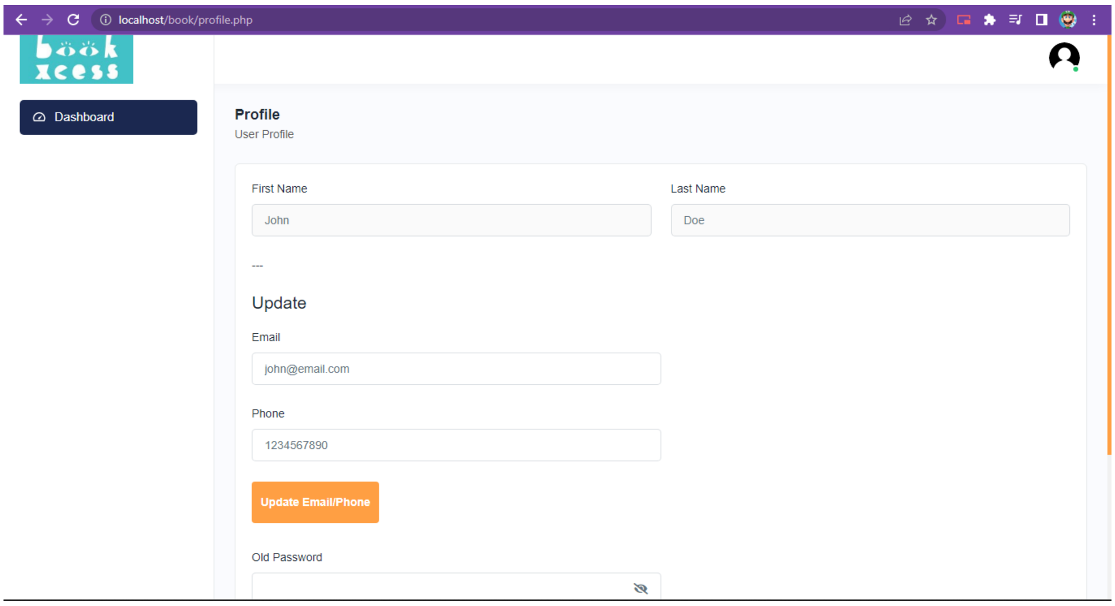
HOME PAGE



Most Quantity Book

Book ID	Book Name	Author	Price	Quantity
3	Book 3	Author 3	15	32
1	Book 1	Author 1	10	13

PROFILE PAGE


localhost/book/profile.php

Dashboard

Profile
User Profile

First Name: John
Last Name: Doe

Update

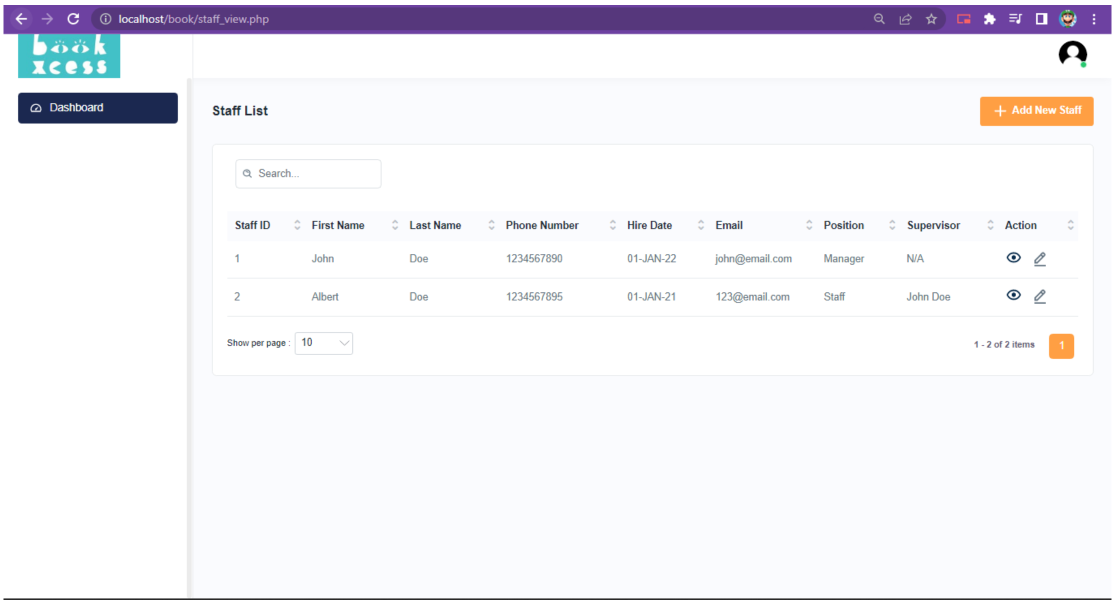
Email: john@email.com

Phone: 1234567890

Update Email/Phone

Old Password

VIEW STAFF


localhost/book/staff_view.php

Dashboard

Staff List

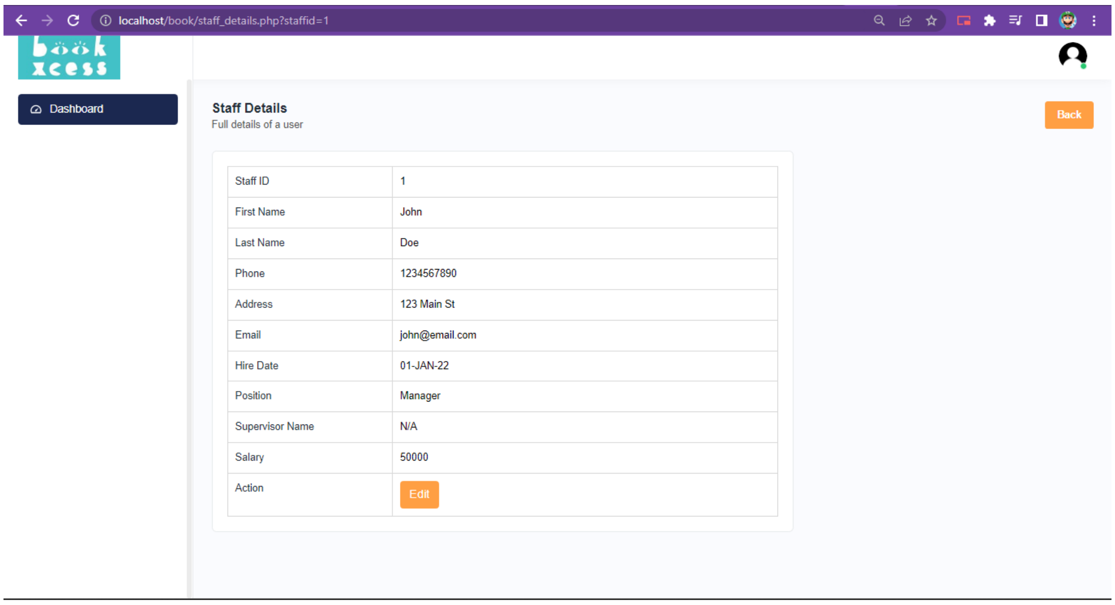
+ Add New Staff

Search...

Staff ID	First Name	Last Name	Phone Number	Hire Date	Email	Position	Supervisor	Action
1	John	Doe	1234567890	01-JAN-22	john@email.com	Manager	N/A	
2	Albert	Doe	1234567895	01-JAN-21	123@email.com	Staff	John Doe	

Show per page : 10

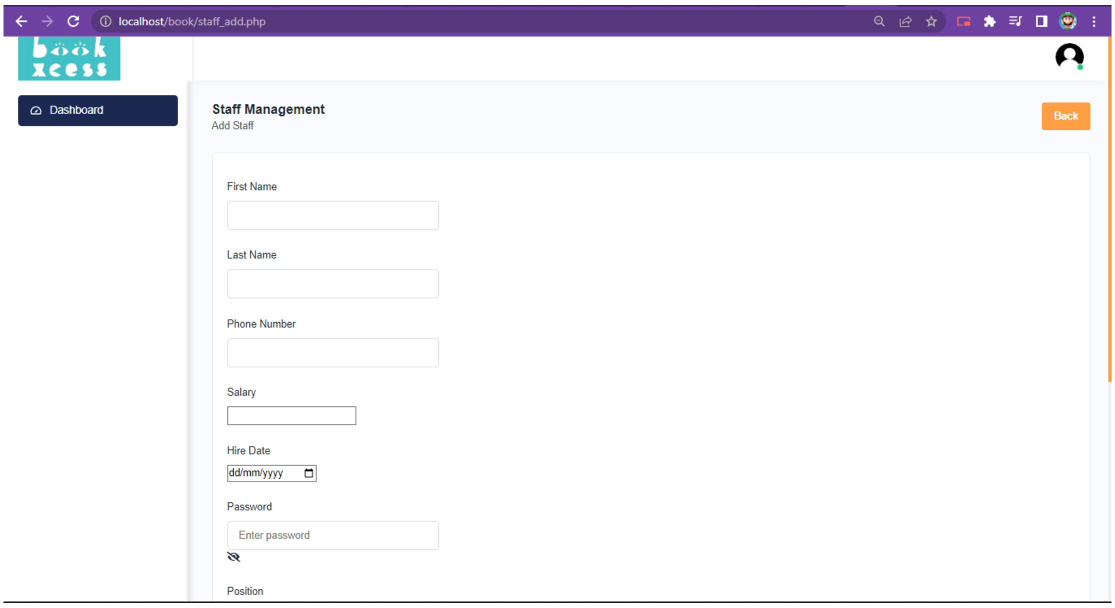
1 - 2 of 2 items 1


localhost/book/staff_details.php?staffid=1

Staff Details
Full details of a user

Staff ID	1
First Name	John
Last Name	Doe
Phone	1234567890
Address	123 Main St
Email	john@email.com
Hire Date	01-JAN-22
Position	Manager
Supervisor Name	N/A
Salary	50000
Action	Edit

ADD STAFF


localhost/book/staff_add.php

Staff Management
Add Staff

First Name

Last Name

Phone Number

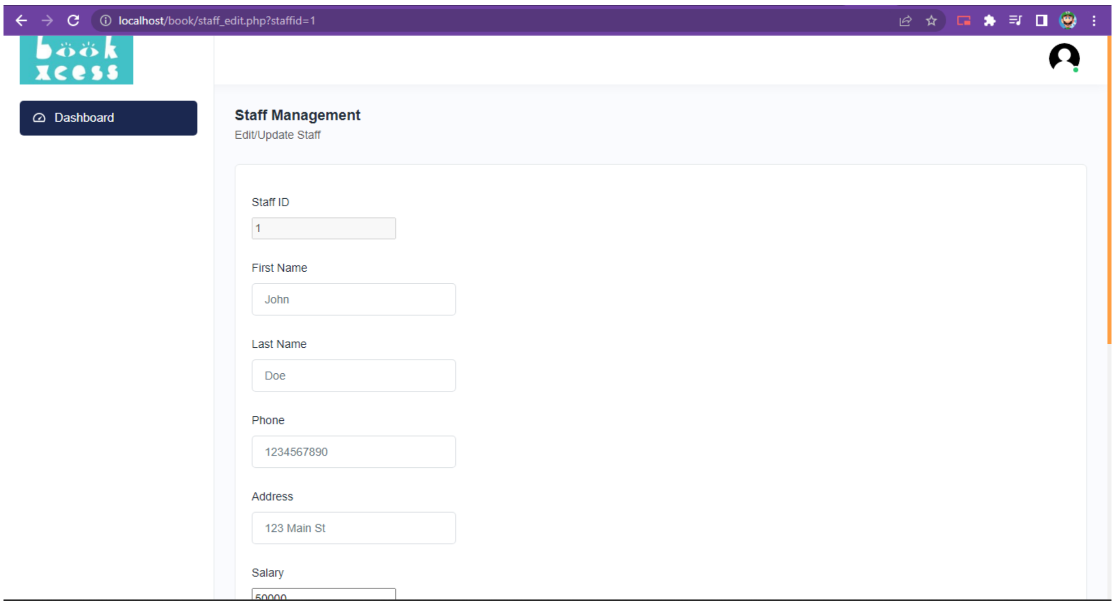
Salary

Hire Date
dd/mm/yyyy

Password
Enter password

Position

EDIT STAFF



localhost/book/staff_edit.php?staffid=1

Dashboard

Staff Management

Edit/Update Staff

Staff ID: 1

First Name: John

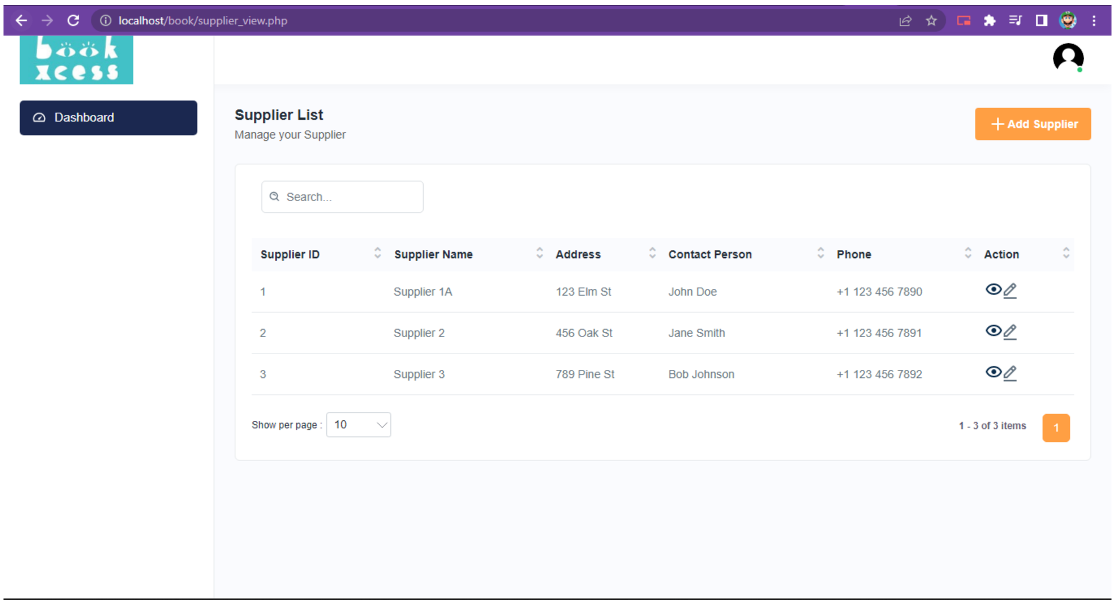
Last Name: Doe

Phone: 1234567890

Address: 123 Main St

Salary: 50000

VIEW SUPPLIER



localhost/book/supplier_view.php

Dashboard

Supplier List

Manage your Supplier

+ Add Supplier

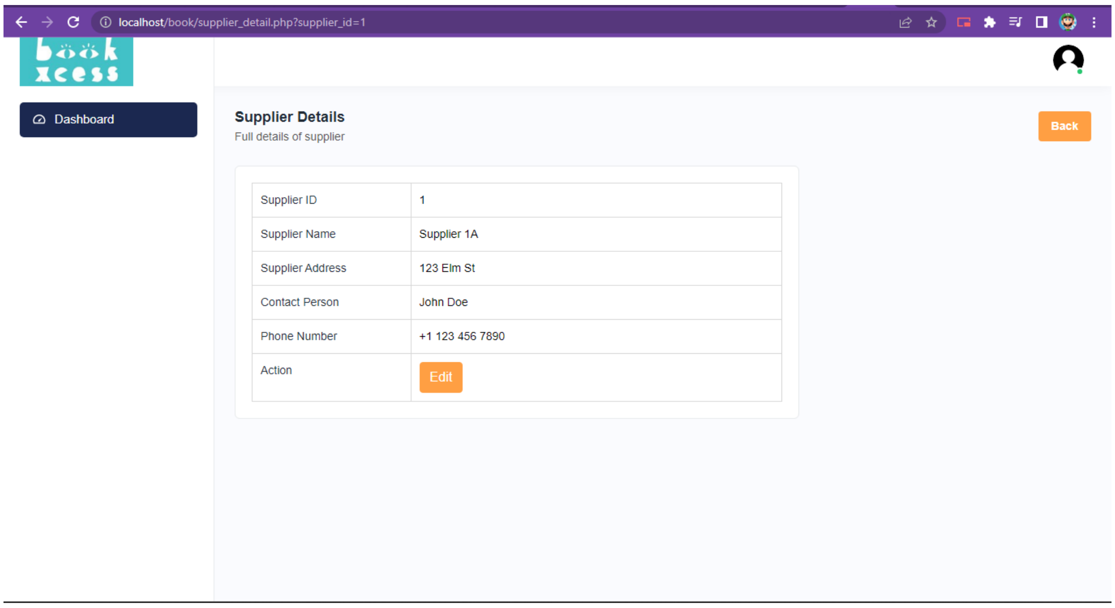
Search...

Supplier ID	Supplier Name	Address	Contact Person	Phone	Action
1	Supplier 1A	123 Elm St	John Doe	+1 123 456 7890	
2	Supplier 2	456 Oak St	Jane Smith	+1 123 456 7891	
3	Supplier 3	789 Pine St	Bob Johnson	+1 123 456 7892	

Show per page : 10

1 - 3 of 3 items

1


localhost/book/supplier_detail.php?supplier_id=1

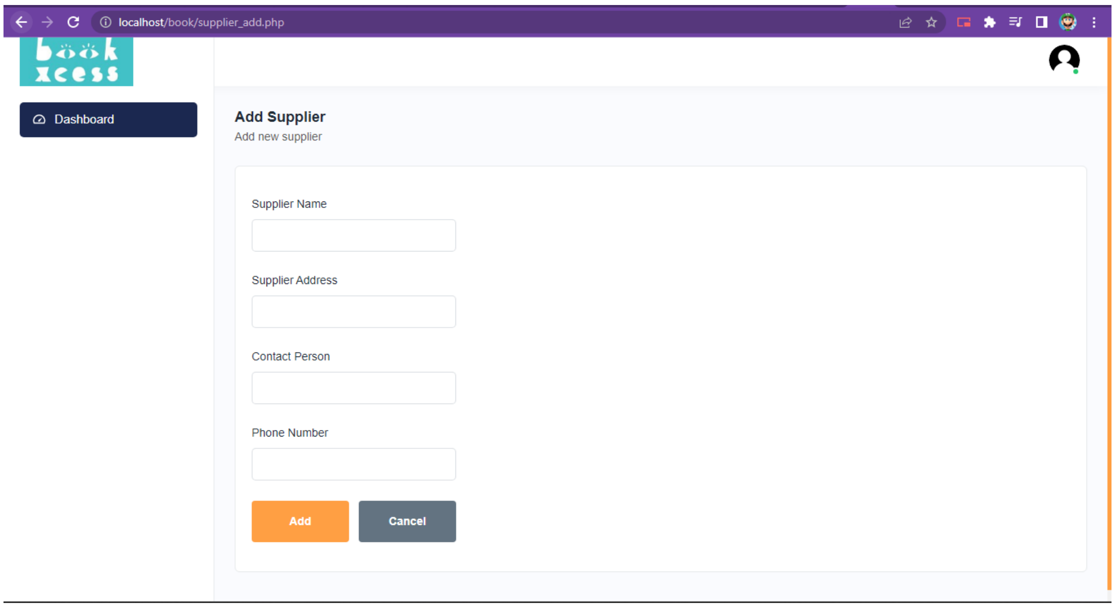
Dashboard

Supplier Details
Full details of supplier

Back

Supplier ID	1
Supplier Name	Supplier 1A
Supplier Address	123 Elm St
Contact Person	John Doe
Phone Number	+1 123 456 7890
Action	Edit

ADD SUPPLIER


localhost/book/supplier_add.php

Dashboard

Add Supplier
Add new supplier

Supplier Name

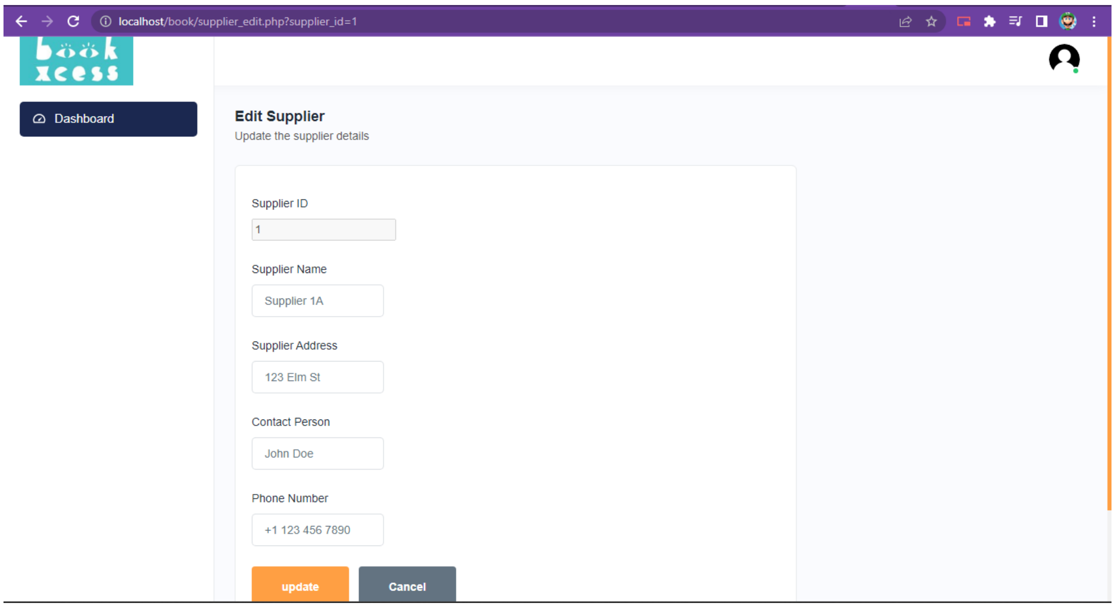
Supplier Address

Contact Person

Phone Number

Add Cancel

EDIT SUPPLIER


localhost/book/supplier_edit.php?supplier_id=1

Dashboard

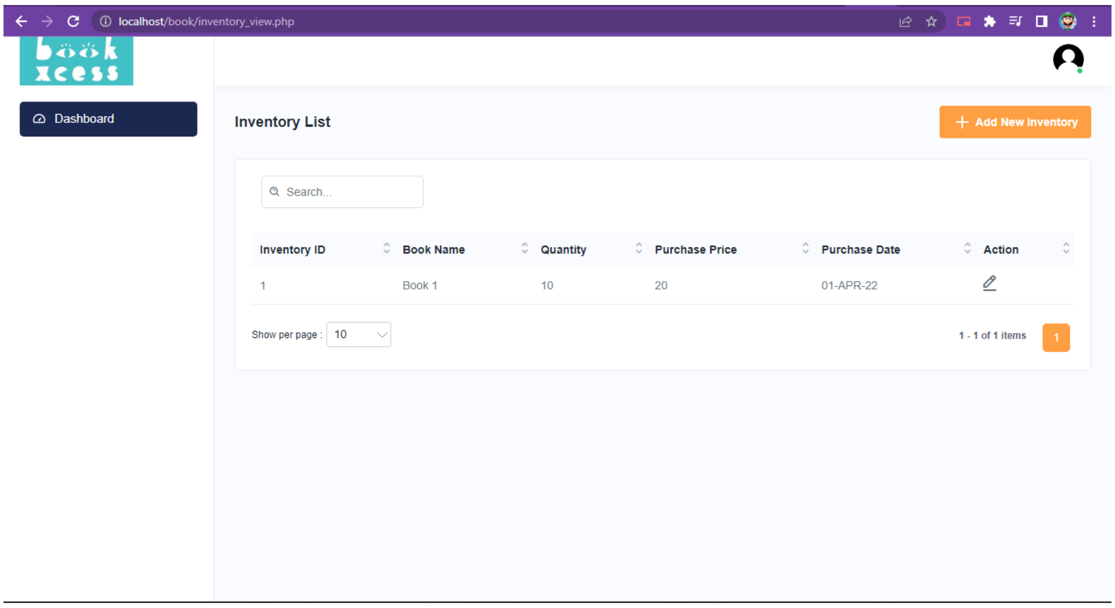
Edit Supplier

Update the supplier details

Supplier ID	1
Supplier Name	Supplier 1A
Supplier Address	123 Elm St
Contact Person	John Doe
Phone Number	+1 123 456 7890

update Cancel

VIEW INVENTORY


localhost/book/inventory_view.php

Dashboard

Inventory List

Add New Inventory

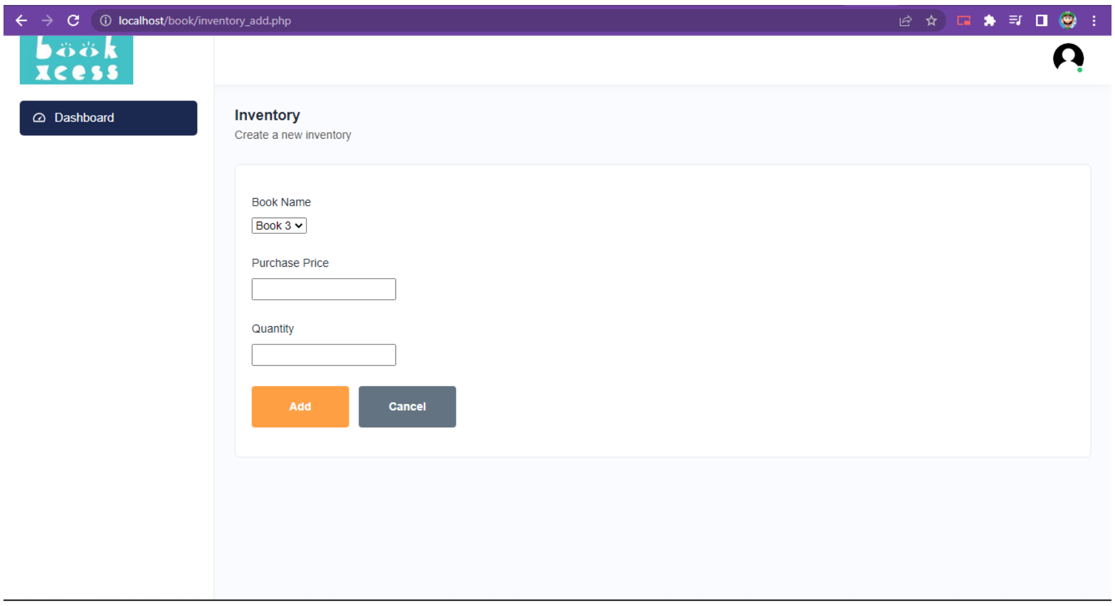
Search...

Inventory ID	Book Name	Quantity	Purchase Price	Purchase Date	Action
1	Book 1	10	20	01-APR-22	

Show per page : 10

1 - 1 of 1 items 1

ADD INVENTORY



localhost/book/inventory_add.php

Dashboard

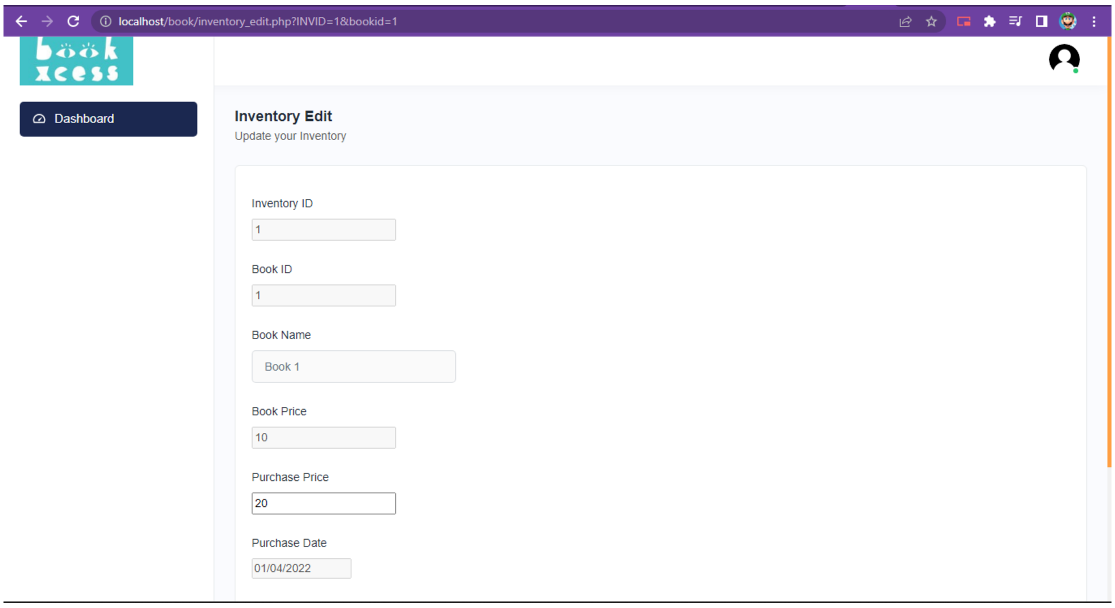
Inventory
Create a new inventory

Book Name
Book 3

Purchase Price

Quantity

EDIT INVENTORY



localhost/book/inventory_edit.php?INVID=1&bookid=1

Dashboard

Inventory Edit
Update your Inventory

Inventory ID
1

Book ID
1

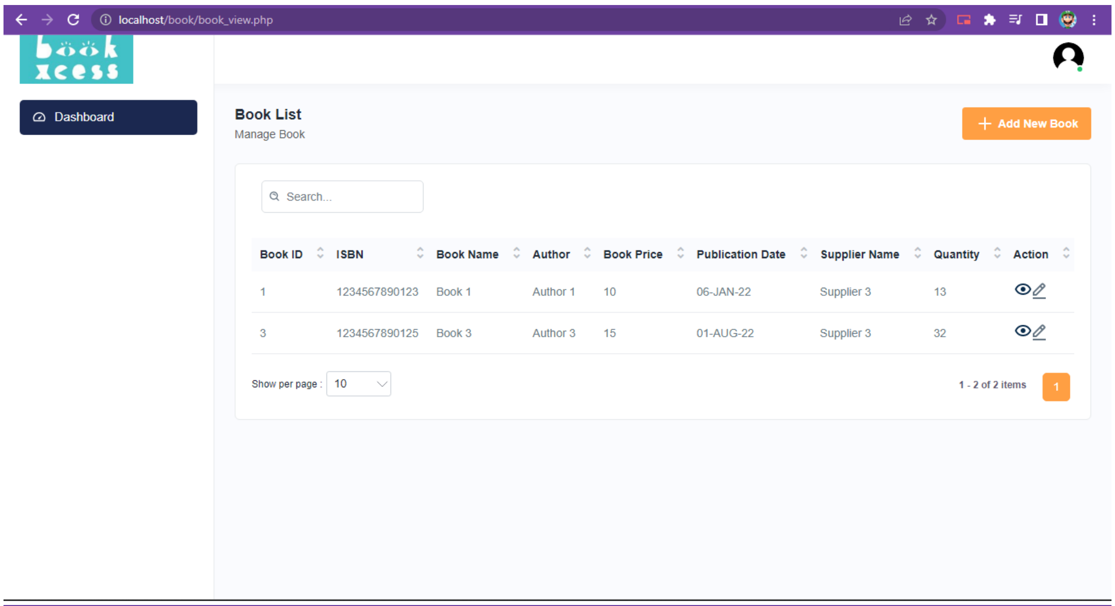
Book Name
Book 1

Book Price
10

Purchase Price
20

Purchase Date
01/04/2022

VIEW BOOK


localhost/book/book_view.php

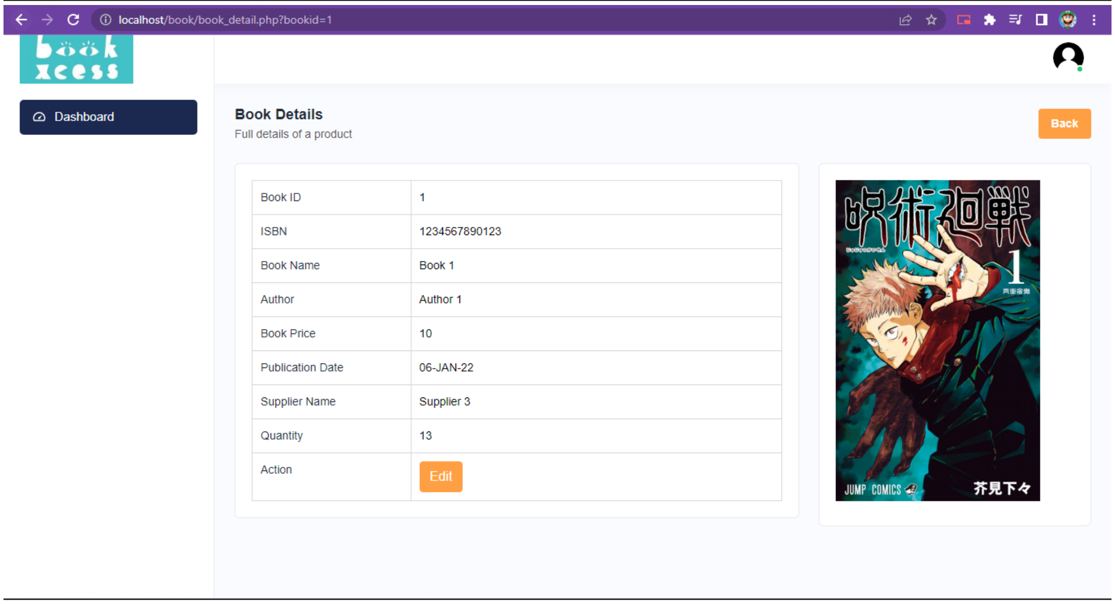
Book List

Manage Book

Book ID	ISBN	Book Name	Author	Book Price	Publication Date	Supplier Name	Quantity	Action
1	1234567890123	Book 1	Author 1	10	06-JAN-22	Supplier 3	13	
3	1234567890125	Book 3	Author 3	15	01-AUG-22	Supplier 3	32	

Show per page : 10

1 - 2 of 2 items

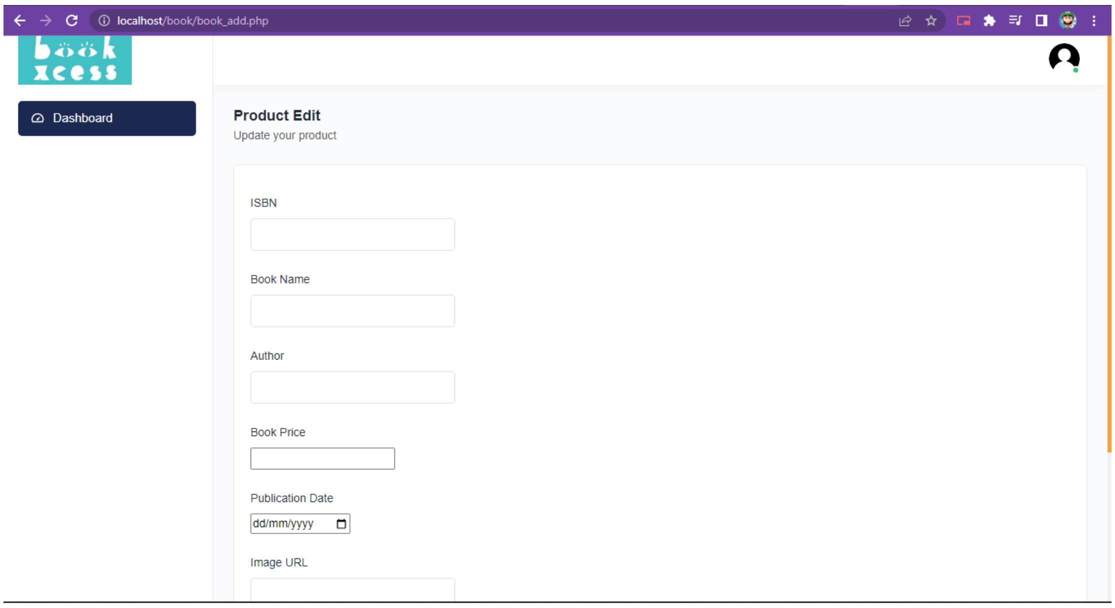

localhost/book/book_detail.php?bookid=1

Book Details

Full details of a product

Book ID	1
ISBN	1234567890123
Book Name	Book 1
Author	Author 1
Book Price	10
Publication Date	06-JAN-22
Supplier Name	Supplier 3
Quantity	13
Action	Edit

ADD BOOK


localhost/book/book_add.php

Product Edit
Update your product

ISBN

Book Name

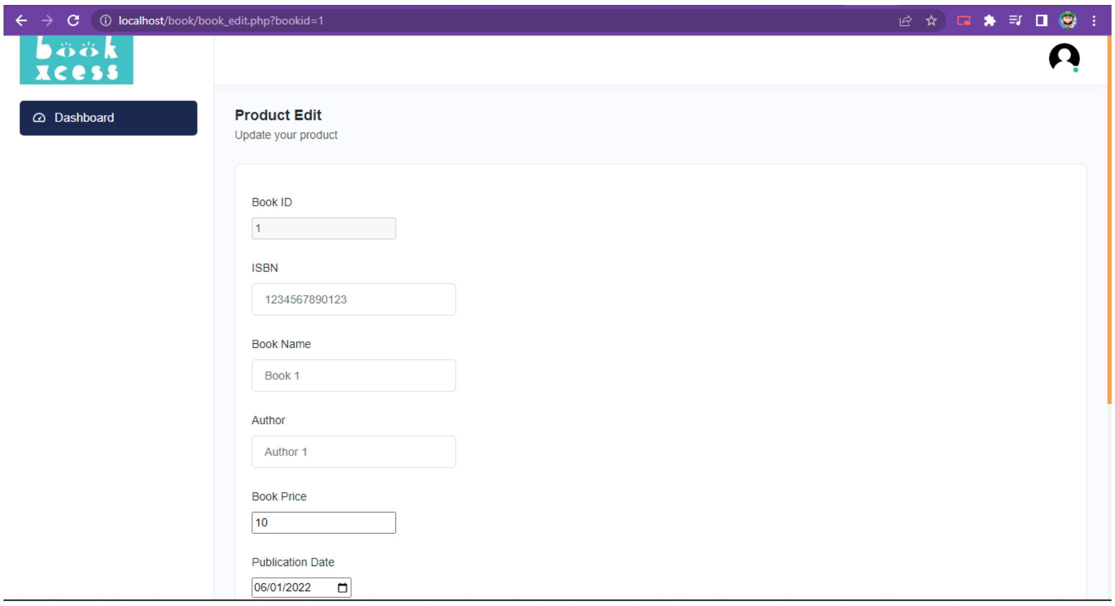
Author

Book Price

Publication Date
dd/mm/yyyy

Image URL

EDIT BOOK


localhost/book/book_edit.php?bookid=1

Product Edit
Update your product

Book ID
1

ISBN
1234567890123

Book Name
Book 1

Author
Author 1

Book Price
10

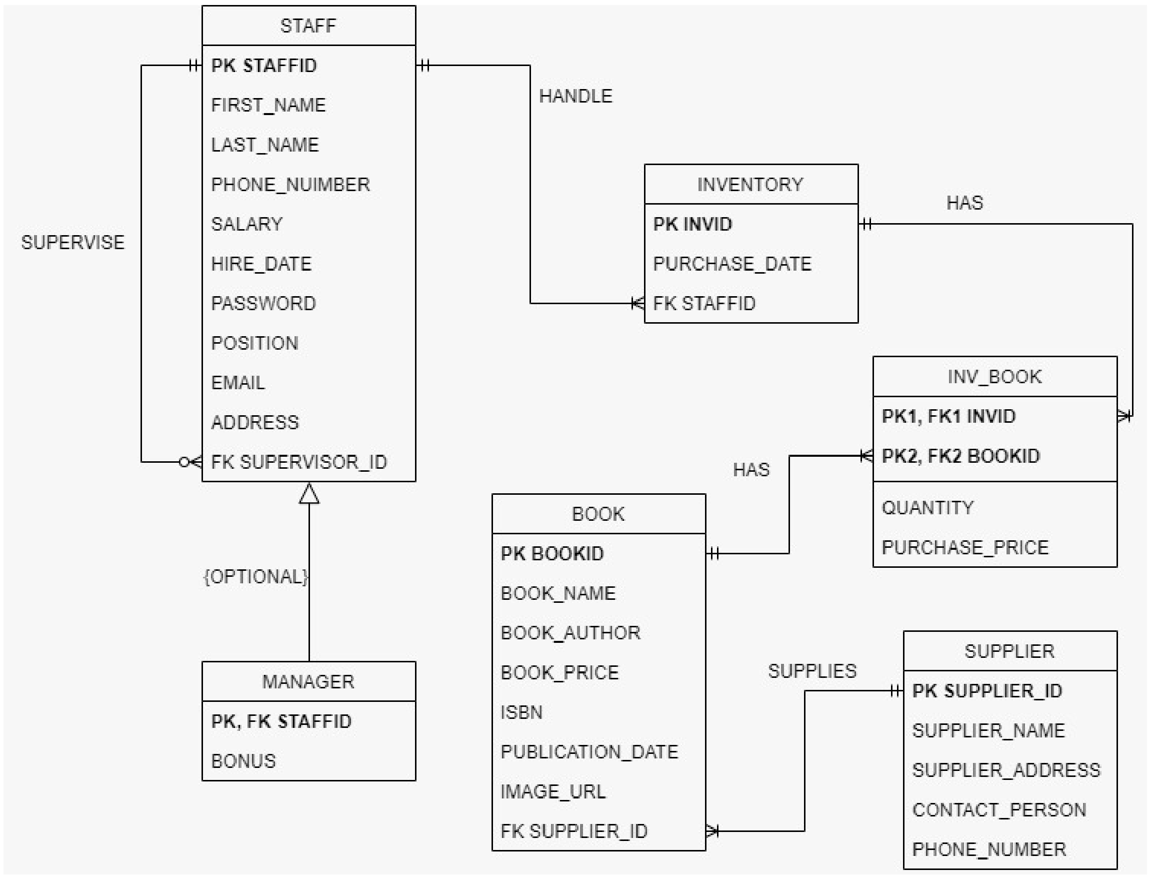
Publication Date
06/01/2022

4.0 Conclusion

In conclusion, a bookstore database using Oracle can be a powerful tool for managing the inventory and sales of a bookstore. It can help track books, customers, and orders, as well as generate reports and analyze sales data. However, it is important to plan and design the database carefully, taking into account the specific requirements of the bookstore and the capabilities of the Oracle database management system. Additionally, it is also important to ensure that the database is properly implemented and maintained to ensure reliable and accurate data.

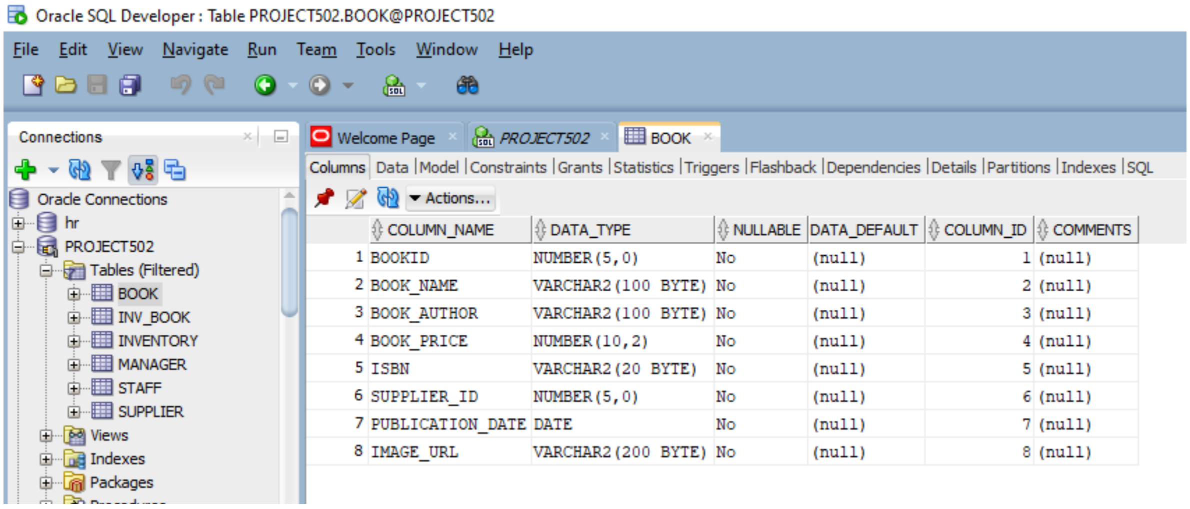
5.0 Appendix

A: Entity Relationship Diagram (ERD)



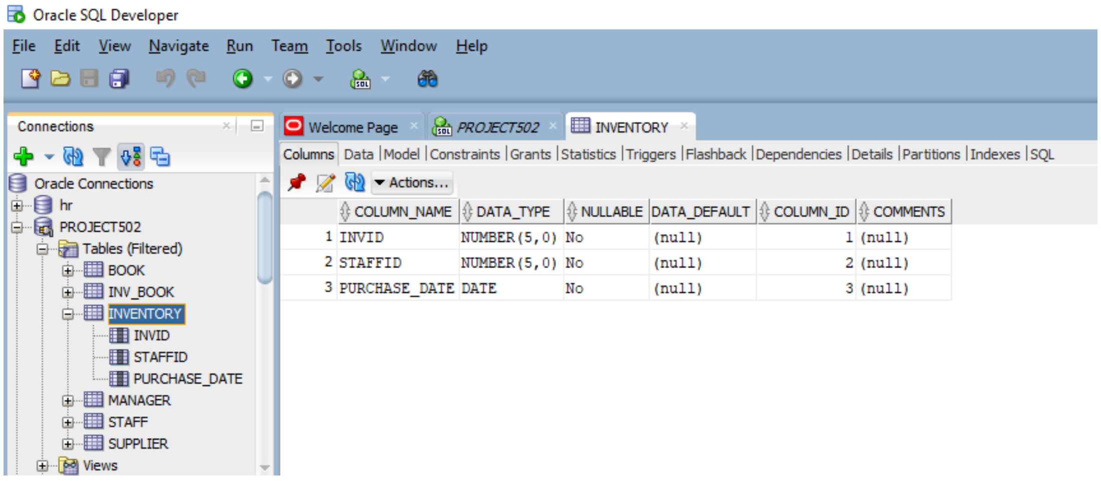
B: Data Dictionary

Book Table


Oracle SQL Developer : Table PROJECT502.BOOK@PROJECT502
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
hr
PROJECT502
Tables (Filtered)
BOOK
INV_BOOK
INVENTORY
MANAGER
STAFF
SUPPLIER
Views
Indexes
Packages
Welcome Page
PROJECT502
BOOK
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Actions...

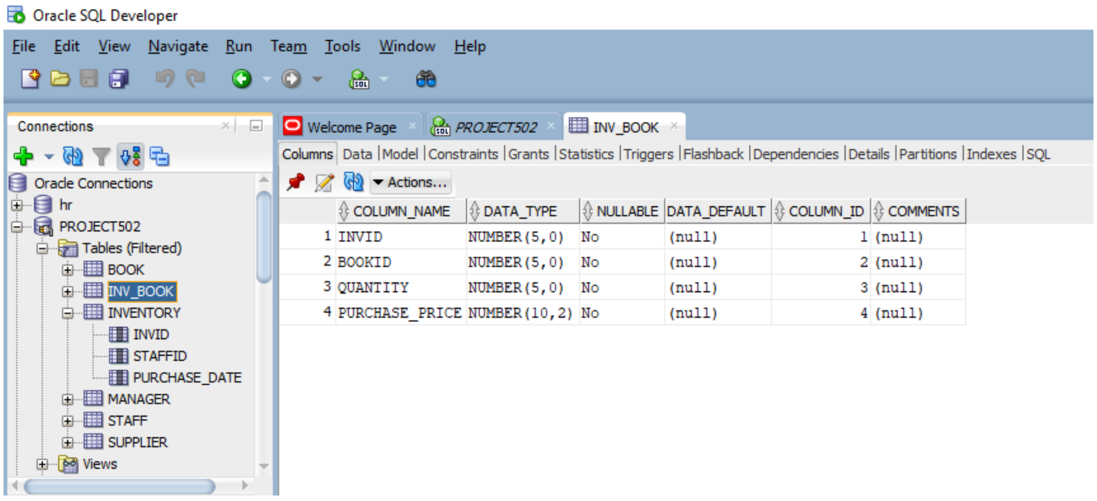
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	BOOKID	NUMBER(5,0)	No	(null)	1	(null)
2	BOOK_NAME	VARCHAR2(100 BYTE)	No	(null)	2	(null)
3	BOOK_AUTHOR	VARCHAR2(100 BYTE)	No	(null)	3	(null)
4	BOOK_PRICE	NUMBER(10,2)	No	(null)	4	(null)
5	ISBN	VARCHAR2(20 BYTE)	No	(null)	5	(null)
6	SUPPLIER_ID	NUMBER(5,0)	No	(null)	6	(null)
7	PUBLICATION_DATE	DATE	No	(null)	7	(null)
8	IMAGE_URL	VARCHAR2(200 BYTE)	No	(null)	8	(null)

Table Inventory



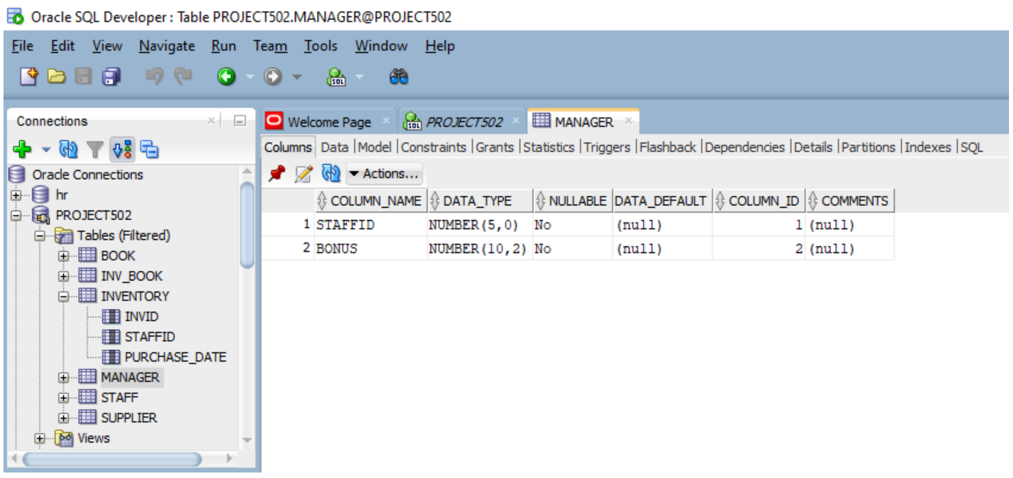
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INVID	NUMBER(5,0)	No	(null)	1	(null)
2	STAFFID	NUMBER(5,0)	No	(null)	2	(null)
3	PURCHASE_DATE	DATE	No	(null)	3	(null)

Table Inv_Book



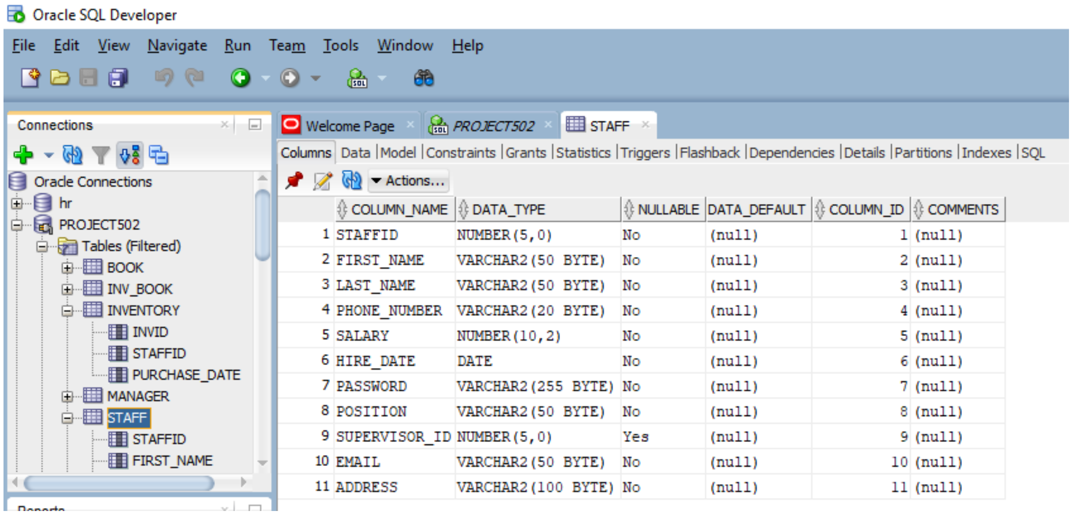
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	INVID	NUMBER(5,0)	No	(null)	1	(null)
2	BOOKID	NUMBER(5,0)	No	(null)	2	(null)
3	QUANTITY	NUMBER(5,0)	No	(null)	3	(null)
4	PURCHASE_PRICE	NUMBER(10,2)	No	(null)	4	(null)

Table Manager


Oracle SQL Developer : Table PROJECT502.MANAGER@PROJECT502
File Edit View Navigate Run Team Tools Window Help
Connections
Oracle Connections
hr
PROJECT502
Tables (Filtered)
BOOK
INV_BOOK
INVENTORY
INVID
STAFFID
PURCHASE_DATE
MANAGER
STAFF
SUPPLIER
Views
Welcome Page
PROJECT502
MANAGER
Columns Data Model Constraints Grants Statistics Triggers Flashback Dependencies Details Partitions Indexes SQL
Actions...

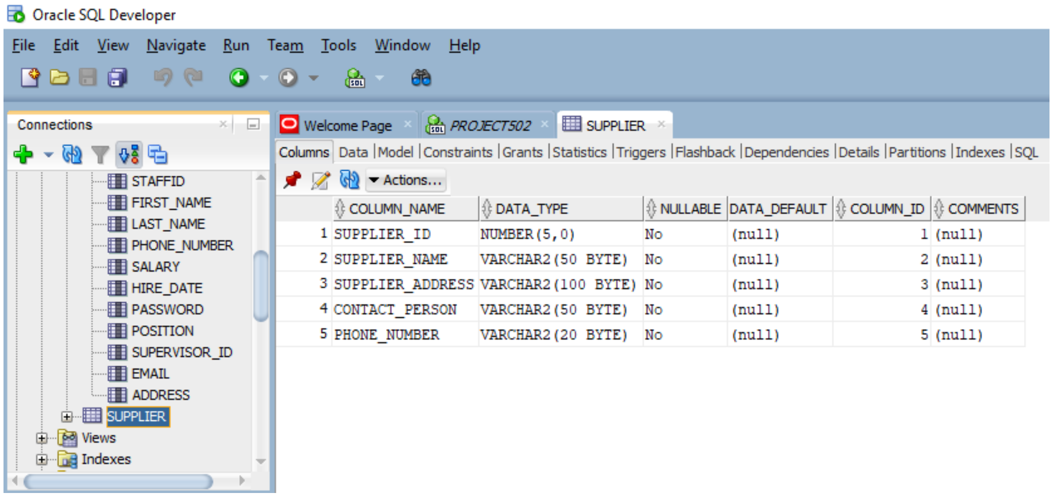
	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STAFFID	NUMBER(5,0)	No	(null)	1	(null)
2	BONUS	NUMBER(10,2)	No	(null)	2	(null)

Table Staff



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	STAFFID	NUMBER(5,0)	No	(null)	1	(null)
2	FIRST_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	LAST_NAME	VARCHAR2(50 BYTE)	No	(null)	3	(null)
4	PHONE_NUMBER	VARCHAR2(20 BYTE)	No	(null)	4	(null)
5	SALARY	NUMBER(10,2)	No	(null)	5	(null)
6	HIRE_DATE	DATE	No	(null)	6	(null)
7	PASSWORD	VARCHAR2(255 BYTE)	No	(null)	7	(null)
8	POSITION	VARCHAR2(50 BYTE)	No	(null)	8	(null)
9	SUPERVISOR_ID	NUMBER(5,0)	Yes	(null)	9	(null)
10	EMAIL	VARCHAR2(50 BYTE)	No	(null)	10	(null)
11	ADDRESS	VARCHAR2(100 BYTE)	No	(null)	11	(null)

Table Supplier



	COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1	SUPPLIER_ID	NUMBER(5,0)	No	(null)	1	(null)
2	SUPPLIER_NAME	VARCHAR2(50 BYTE)	No	(null)	2	(null)
3	SUPPLIER_ADDRESS	VARCHAR2(100 BYTE)	No	(null)	3	(null)
4	CONTACT_PERSON	VARCHAR2(50 BYTE)	No	(null)	4	(null)
5	PHONE_NUMBER	VARCHAR2(20 BYTE)	No	(null)	5	(null)

C: Data Definition Language (DDL)

```
CREATE TABLE Staff
(
staffid NUMBER(5) CONSTRAINT staffid_pk PRIMARY KEY,
first_name VARCHAR2(50) CONSTRAINT first_name_nn NOT NULL,
last_name VARCHAR2(50) CONSTRAINT last_name_nn NOT NULL,
phone_number VARCHAR2(20) CONSTRAINT phone_number_nn NOT NULL,
salary NUMBER(10,2) CONSTRAINT salary_nn NOT NULL,
hire_date DATE CONSTRAINT hire_date_nn NOT NULL,
password VARCHAR2(255) CONSTRAINT password_nn NOT NULL,
position VARCHAR2(50) CONSTRAINT position_nn NOT NULL,
supervisor_id NUMBER(5) CONSTRAINT supervisor_id_fk REFERENCES
Staff(staffid) ON UPDATE CASCADE,
email VARCHAR2(50) CONSTRAINT email_nn NOT NULL,
address VARCHAR2(100) CONSTRAINT address_nn NOT NULL
);

CREATE TABLE Manager
(
staffid NUMBER(5) PRIMARY KEY,
bonus NUMBER(10,2) CONSTRAINT bonus_nn NOT NULL,
FOREIGN KEY (staffid) REFERENCES Staff(staffid) ON UPDATE CASCADE
);

CREATE TABLE Inventory
(
invid NUMBER(5) CONSTRAINT invid_pk PRIMARY KEY,
staffid NUMBER(5) CONSTRAINT staffid_fk NOT NULL,
purchase_date DATE CONSTRAINT purchase_date_nn NOT NULL,
FOREIGN KEY (staffid) REFERENCES Staff(staffid) ON UPDATE CASCADE
);

CREATE TABLE Supplier
(
supplier_id NUMBER(5) CONSTRAINT supplier_id_pk PRIMARY KEY,
supplier_name VARCHAR2(50) CONSTRAINT supplier_name_nn NOT NULL,
supplier_address VARCHAR2(100) CONSTRAINT supplier_address_nn NOT
NULL,
contact_person VARCHAR2(50) CONSTRAINT contact_person_nn NOT NULL,
phone_number VARCHAR2(20) CONSTRAINT sp_phone_number_nn NOT NULL
);
```

```
CREATE TABLE Book
(
bookid NUMBER(5) CONSTRAINT bookid_pk PRIMARY KEY,
book_name VARCHAR2(100) CONSTRAINT book_name_nn NOT NULL,
book_author VARCHAR2(100) CONSTRAINT book_author_nn NOT NULL,
book_price NUMBER(10,2) CONSTRAINT book_price_nn NOT NULL,
isbn VARCHAR2(20) CONSTRAINT isbn_nn NOT NULL,
supplier_id NUMBER(5) CONSTRAINT supplier_id_fk NOT NULL,
publication_date DATE CONSTRAINT publication_date_nn NOT NULL,
image_url VARCHAR2(200) CONSTRAINT image_url_nn NOT NULL,
FOREIGN KEY (supplier_id) REFERENCES Supplier(supplier_id) ON UPDATE
CASCADE
);

CREATE TABLE inv_book
(
invid NUMBER(5) CONSTRAINT invid_fk NOT NULL,
bookid NUMBER(5) CONSTRAINT bookid_fk NOT NULL,
quantity NUMBER(5) CONSTRAINT quantity_nn NOT NULL,
purchase_price NUMBER(10,2) CONSTRAINT purchase_price_nn NOT NULL,
PRIMARY KEY (invid, bookid),
FOREIGN KEY (invid) REFERENCES Inventory(invid) ON UPDATE CASCADE,
FOREIGN KEY (bookid) REFERENCES Book(bookid) ON UPDATE CASCADE
);

ALTER TABLE Staff ADD CONSTRAINT email_uk UNIQUE (email);

-- sequence
CREATE SEQUENCE book_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE inv_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE manager_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE staff_id_seq START WITH 1 INCREMENT BY 1;
CREATE SEQUENCE supplier_id_seq START WITH 1 INCREMENT BY 1;
```

D: Data Manipulation Language (DML)

```
SELECT EMAIL, PASSWORD FROM STAFF WHERE EMAIL = :email;

SELECT STAFFID FROM STAFF WHERE EMAIL = :email;

SELECT STAFFID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, SALARY,
HIRE_DATE, POSITION, EMAIL, ADDRESS, SUPERVISOR_ID FROM STAFF WHERE
STAFFID = :staffid;

SELECT POSITION FROM STAFF WHERE STAFFID = :staffid;

SELECT FIRST_NAME || ' ' || LAST_NAME AS FULLNAME FROM STAFF WHERE
STAFFID = :staffid;

SELECT COUNT(*) AS TOTAL FROM STAFF;

SELECT STAFFID, FIRST_NAME, LAST_NAME, PHONE_NUMBER, HIRE_DATE,
EMAIL, POSITION, SUPERVISOR_ID FROM STAFF;

SELECT FIRST_NAME || ' ' || LAST_NAME AS FULLNAME FROM STAFF WHERE
STAFFID = :supervisorid;

SELECT STAFFID, FIRST_NAME || ' ' || LAST_NAME AS FULLNAME FROM STAFF
WHERE POSITION = 'Manager';

UPDATE STAFF SET FIRST_NAME = :firstname, LAST_NAME = :lastname,
PHONE_NUMBER = :phonenumber, EMAIL = :email, ADDRESS = :address,
POSITION = :position, SALARY = :salary, SUPERVISOR_ID = :supervisorid
WHERE STAFFID = :staffid;

DELETE FROM STAFF WHERE STAFFID = :staffid;

INSERT INTO STAFF (STAFFID, FIRST_NAME, LAST_NAME, PHONE_NUMBER,
HIRE_DATE, EMAIL, ADDRESS, POSITION, SALARY, SUPERVISOR_ID, PASSWORD)
VALUES (STAFF_ID_SEQ.nextval, :firstname, :lastname, :phonenumber,
TO_DATE(:hiredate), :email, :address, :position, TO_NUMBER(:salary,
9999999999.99), TO_NUMBER(:supervisorid, 99999), :password);

SELECT STAFF_ID_SEQ.CURRVAL FROM DUAL;
SELECT STAFF_ID_SEQ.NEXTVAL FROM DUAL;

UPDATE STAFF SET EMAIL = :email, PHONE_NUMBER = :phone WHERE STAFFID
= :staffid;

UPDATE STAFF SET PASSWORD = :password WHERE STAFFID = :staffid;

SELECT PASSWORD FROM STAFF WHERE STAFFID = :staffid;

SELECT b.bookid, b.book_name, b.book_author, b.book_price,
b.image_url, SUM(ib.quantity) AS purchase_count
        FROM inv_book ib
        JOIN Book b ON ib.bookid = b.bookid
        WHERE ROWNUM <= 10
        GROUP BY b.book_name, b.book_author, b.book_price, b.bookid,
b.image_url
        ORDER BY purchase_count DESC;

SELECT * FROM Inventory;

SELECT * FROM Inventory WHERE INVID = :invid;

SELECT * FROM inv_book WHERE INVID = :invid;

SELECT * FROM inv_book ib
                JOIN Book b ON ib.bookid = b.bookid
                WHERE ib.invid = :invid;

SELECT I.INVID, b.bookid, b.book_name, ib.quantity,ib.purchase_price,
b.book_price, i.purchase_date
        FROM Inventory i
        JOIN inv_book ib ON i.invid = ib.invid
        JOIN Book b ON ib.bookid = b.bookid
        WHERE i.staffid = :staffid;

SELECT I.INVID, b.bookid, b.book_name, ib.quantity,ib.purchase_price,
b.book_price, i.purchase_date
        FROM Inventory i
        JOIN inv_book ib ON i.invid = ib.invid
        JOIN Book b ON ib.bookid = b.bookid
        WHERE i.INVID = :INVID AND b.bookid = :bookid;
```

```
UPDATE inv_book SET quantity = :quantity, purchase_price =
:purchase_price WHERE invid = :invid AND bookid = :bookid;

INSERT INTO Inventory (INVID, STAFFID, PURCHASE_DATE) VALUES
(INV_ID_SEQ.NEXTVAL, :staffid, SYSDATE);

INSERT INTO inv_book (INVID, BOOKID, QUANTITY, PURCHASE_PRICE) VALUES
(:invid, :bookid, :quantity, :purchase_price);

SELECT INV_ID_SEQ.CURRVAL AS INVID FROM DUAL;

SELECT COUNT(*) AS total FROM Supplier;

SELECT * FROM Supplier;

SELECT supplier_id, supplier_name FROM Supplier;

SELECT * FROM Supplier WHERE supplier_id = :supplier_id;

UPDATE Supplier SET supplier_name = :supplier_name, supplier_address
= :supplier_address, contact_person = :contact_person, phone_number =
:phone_number WHERE supplier_id = :supplier_id;

INSERT INTO Supplier (supplier_id, supplier_name, supplier_address,
contact_person, phone_number) VALUES (supplier_id_seq.nextval,
:supplier_name, :supplier_address, :contact_person, :phone_number);

SELECT supplier_id_seq.currval AS supplier_id FROM dual;

DELETE FROM Supplier WHERE supplier_id = :supplier_id;

SELECT SUM(quantity) AS total FROM inv_book;

SELECT COUNT(*) AS total FROM inv_book;

SELECT COUNT(*) AS total FROM Book;

SELECT b.bookid, b.isbn, b.book_name, b.book_author, b.book_price,
b.publication_date, s.supplier_name, SUM(ib.quantity) as quantity,
b.image_url
        FROM inv_book ib
        RIGHT OUTER JOIN Book b ON ib.bookid = b.bookid
        JOIN Supplier s ON b.supplier_id = s.supplier_id
        GROUP BY b.bookid, b.isbn, b.book_name, b.book_author,
b.book_price, b.publication_date, s.supplier_name, b.image_url;

SELECT b.bookid, b.isbn, b.book_name, b.book_author, b.book_price,
b.publication_date, s.supplier_name, SUM(ib.quantity) as quantity,
b.image_url
        FROM inv_book ib
        RIGHT OUTER JOIN Book b ON ib.bookid = b.bookid
        JOIN Supplier s ON b.supplier_id = s.supplier_id
        WHERE b.bookid = :bookid
        GROUP BY b.bookid, b.isbn, b.book_name, b.book_author,
b.book_price, b.publication_date, s.supplier_name, b.image_url;

SELECT bookid, isbn, book_name, book_author, book_price,
publication_date, image_url, supplier_id FROM Book WHERE bookid =
:bookid;

UPDATE BOOK SET ISBN = :isbn, BOOK_NAME = :book_name, BOOK_AUTHOR =
:book_author, BOOK_PRICE = :book_price, PUBLICATION_DATE =
to_date(:publication_date, 'dd-mon-yyyy'), IMAGE_URL = :image_url,
SUPPLIER_ID = :supplier_id WHERE BOOKID = :bookid;

DELETE FROM BOOK WHERE BOOKID = :bookid;

INSERT INTO BOOK (BOOKID, ISBN, BOOK_NAME, BOOK_AUTHOR, BOOK_PRICE,
PUBLICATION_DATE, IMAGE_URL, SUPPLIER_ID) VALUES
(book_id_seq.NEXTVAL, :isbn, :book_name, :book_author, :book_price,
to_date(:publication_date, 'dd-mon-yyyy'), :image_url, :supplier_id);

SELECT book_id_seq.currval FROM DUAL;

SELECT DISTINCT BOOK_NAME, BOOKID FROM BOOK;
```

RUBRICS

ITEM	**MARKS**	**GROUP**
Table of Content (1 Mark)		CLASS: 3D MEMBERS:
INTRODUCTION		
Company Background (2 Marks) **2 Marks** If the company background is presented		
CASE STUDY		
Problem Statement (5 Marks) **1-3 Marks** If they did not state that the current system is Manual or File-based Approach. **4-5 Marks** If they state that the current system is Manual or File-based Approach with some relevant sub problems because of the manual system.		
Objective (5 Marks) **1-3 Marks** If they state the system objective **4-5 Marks** If they state that they want to design, develop and test as the objective.		
SYSTEM DESIGN		
Flow Chart of System **(10 Marks)** **1-5 Marks** if there is flow chart but it is not reflecting the whole system **6-10 Marks** if the flowchart reflect the whole system		
10 SQL Queries **(20 Marks)** 2 Marks for each query if they use different kind of SQL. 1 Mark is for the repeated SQL For example: UPDATE EMP SET empID = 100 WHERE name = ‘Hamiz’; □ **2 Marks** Update DEPT SET deptName = ‘Finance’ WHERE deptID = ‘10’; □ **1 Mark as the operation is almost the same as previous SQL.**		

System Development Sample Screen (20 Marks) Read Insert Update Delete Bridge Recursive Inheritance Extra 2 marks 2 marks 2 marks 2 marks 3 marks 3 marks 3 marks 3 marks		
Extra can be anything related to database function. For example, use sequence for primary key (get 1 mark).	Total:	
Conclusion (5 Marks) **5 Marks** will be given if they have stated what is the conclusion from the project that have been developed.		
APPENDIX A: ERD **20 Marks** **Rubric for** **ERD:** **Entity 4** **Attributes 3** **Relationship 3** **Relationship Name 2** **Cardinality/Modality 2** **Inheritance 2** **Recursive 2** **Bridge 2** **TOTAL 20 Marks**		
APPENDIX B: Data Dictionary **3 Marks** will be given as long as it is being inserted in report		
APPENDIX C: DDL **3 Marks** will be given if all the DDL for ALL tables are presented.		
APPENDIX D: DML **3 Marks** will be given if all the DML for ALL tables are presented. At least 2 DML for each table.		
APPENDIX F: CD/FLASH DRIVE **3 Marks** will be given if they attached the CD/FLASH DRIVE at report during presentation.		**TOTAL MARKS** **100**