

## DBMS LAB 7

NAME : NAGAVENI L G

SRN : PES2UG21CS315

SEC : 5F

### ONLINE BOOK STORE MANAGEMENT

#### xml\_file1.xml

```
<root>
  <books>
    <row>
      <BookID>1</BookID>
      <Title>Book 1</Title>
      <Author>Author 1</Author>
      <Price>19.99</Price>
      <Quantity>10</Quantity>
    </row>
    <row>
      <BookID>2</BookID>
      <Title>Book 2</Title>
      <Author>Author 2</Author>
      <Price>29.99</Price>
      <Quantity>15</Quantity>
    </row>
    <row>
      <BookID>3</BookID>
      <Title>Book 3</Title>
      <Author>Author 3</Author>
      <Price>24.99</Price>
      <Quantity>8</Quantity>
    </row>
    <row>
      <BookID>4</BookID>
      <Title>Book 4</Title>
      <Author>Author 4</Author>
      <Price>39.99</Price>
      <Quantity>12</Quantity>
    </row>
    <row>
```

```

<BookID>5</BookID>
<Title>Book 5</Title>
<Author>Author 5</Author>
<Price>14.99</Price>
<Quantity>20</Quantity>
</row>
</books>
<users>
<row>
<UserID>1</UserID>
<Username>User1</Username>
<Password>Password1</Password>
</row>
<row>
<UserID>2</UserID>
<Username>User2</Username>
<Password>Password2</Password>
</row>
</users>
<carts>
<row>
<CartID>1</CartID>
<UserID>1</UserID>
<BookID>1</BookID>
<Quantity>2</Quantity>
</row>
<row>
<CartID>2</CartID>
<UserID>1</UserID>
<BookID>3</BookID>
<Quantity>1</Quantity>
</row>
<row>
<CartID>3</CartID>
<UserID>2</UserID>
<BookID>5</BookID>
<Quantity>3</Quantity>
</row>
</carts>
</root>

```

## Xml\_file2.xml

```

<root>
  <books>
    <row>

```

```

<BookID>1</BookID>
<Title>Updated Book 1</Title>
<Author>New Author</Author>
<Price>24.99</Price>
<Quantity>15</Quantity>
</row>
</books>
</root>

```

## Frontend.py

```

# frontend.py

import streamlit as st
import pandas as pd
from backend import insert_data, delete_records, update_data, get_table_data
from backend import execute_query
import time

def display_table(table_name):
    columns, data = get_table_data(table_name)
    df = pd.DataFrame(data, columns=columns)
    st.write(f"Displaying data from the {table_name} table:")
    st.write(df)

def update_users():
    st.subheader('Update Information in Users Table:')
    update_columns = {}
    for column in ['Username', 'Password']:
        update_columns[column] = st.text_input(column, '')
    condition = st.text_input('Enter the condition for updating (e.g., UserID=1)', '')
    if st.button('Update Information'):
        # Ensure that the condition is properly formatted
        condition = f"UserID={condition.strip()}"
        update_data('users', update_columns, condition)
        st.success('Information updated successfully in the Users table!')

# frontend.py

def update_carts():

```

```

st.subheader('Update Information in Carts Table:')
update_columns = {}
for column in ['BookID', 'Quantity']:
    update_columns[column] = st.text_input(column, '')

condition = st.text_input('Enter the condition for updating (e.g., CartID=1)', '')
if st.button('Update Information'):
    # Ensure that the condition is properly formatted
    condition = f"CartID={condition.strip()}""

update_data('carts', update_columns, condition)
st.success('Information updated successfully in the Carts table!')


def update_books():
    st.subheader('Update Information in Books Table:')
    update_columns = {}
    for column in ['Title', 'Author', 'Price', 'Quantity']:
        update_columns[column] = st.text_input(column, '')
    condition = st.text_input('Enter the condition for updating (e.g., BookID=1)', '')
    if st.button('Update Information'):
        # Ensure that the condition is properly formatted
        condition = f"BookID={condition.strip()}""
        update_data('books', update_columns, condition)
        st.success('Information updated successfully in the Books table!')
# frontend.py

# ...

def delete_records_page():
    st.header('Delete Records')

    # Assuming 'users', 'books', and 'carts' are valid table names
    table_name = st.selectbox('Select a table to delete records from', ['users', 'books', 'carts'])

    st.write(f"Delete records from the {table_name} table:")

    condition = st.text_input(f'Enter the condition for deletion (e.g., {table_name}ID=1)', '')

    if st.button('Delete Records'):
        success, message = delete_records(table_name, condition)
        if success:
            st.success(message)
        else:

```

```
st.error(message)

# ...

def main():
    st.title('Online Bookstore Management')

    st.sidebar.header('Navigation')
    page = st.sidebar.radio("Select a page", ["View Tables", "Insert Data",
    "Delete Records", "Update Information"])

    if page == "View Tables":
        st.header('View Tables')
        table_name = st.selectbox('Select a table to view', ['books', 'users',
        'carts'])
        display_table(table_name)

    elif page == "Insert Data":
        st.header('Insert Data')

        if st.checkbox('Insert data into "users" table'):
            st.subheader('Insert Data into Users Table:')
            user_values = {}
            for column in ['Username', 'Password']:
                user_values[column] = st.text_input(column, '')
            if st.button('Insert User Data'):
                insert_data('users', user_values)
                st.success('User data inserted successfully!')

        if st.checkbox('Insert data into "books" table'):
            st.subheader('Insert Data into Books Table:')
            book_values = {}
            for column in ['Title', 'Author', 'Price', 'Quantity']:
                book_values[column] = st.text_input(column, '')
            if st.button('Insert Book Data'):
                insert_data('books', book_values)
                st.success('Book data inserted successfully!')

    # frontend.py

# ...

if st.checkbox('Insert data into "carts" table'):
    st.subheader('Insert Data into Carts Table:')
```

```

cart_values = {}
for column in ['UserID', 'BookID', 'Quantity']:
    cart_values[column] = st.text_input(column, '')

if st.button('Insert carts Data'):
    insert_data('carts', cart_values)
    st.success('cart data inserted successfully!')

# ...

elif page == "Delete Records":
    delete_records_page()

# ...

elif page == "Update Information":
    st.header('Update Information')
    table_name = st.selectbox('Select a table to update information in',
    ['books', 'users', 'carts'])
    st.write(f"Update information in the {table_name} table:")

    if table_name == 'users':
        update_users()
    elif table_name == 'books':
        update_books()
    elif table_name == 'carts':
        update_carts()

if __name__ == '__main__':
    main()

```

## backend.py

```

# backend.py

import mysql.connector
import streamlit as st
import xml.etree.ElementTree as ET
from mysql.connector import IntegrityError
# backend.py

def connect_to_database():

```

```
conn = mysql.connector.connect(
    host="localhost",
    user="root",
    password="12345678",
    database="onlinebookstore"
)
return conn

def execute_query(query):
    conn = connect_to_database()
    cursor = conn.cursor()
    cursor.execute(query)
    if "SELECT" in query:
        results = cursor.fetchall()
        columns = [desc[0] for desc in cursor.description]
        conn.close()
        return columns, results
    else:
        conn.commit()
        conn.close()

def insert_data(table_name, values):
    print(f"Inserting data into {table_name} table.")
    columns = ', '.join(values.keys())
    values_str = ', '.join(f"'{value}'" for value in values.values())
    query = f"INSERT INTO {table_name} ({columns}) VALUES ({values_str})"
    execute_query(query)
    st.success(f"Data inserted successfully into '{table_name}' table.")

def delete_records(table_name, condition):
    try:
        # Check if the record exists
        check_record_query = f"SELECT COUNT(*) FROM {table_name} WHERE {condition}"
        count_result = execute_query(check_record_query)
```

```

        print("count_result:", count_result) # Add this line to print
count_result

        if count_result and count_result[1][0][0] > 0:
            # Record exists, delete it
            delete_query = f"DELETE FROM {table_name} WHERE {condition}"
            execute_query(delete_query) # Remove fetch=False
            return True, f'Record deleted successfully from the {table_name}
table!'
        else:
            return False, f'Record not found in the {table_name} table.'
    except IntegrityError as e:
        return False, f'Error deleting record: {e}'


def update_data(table_name, update_columns, condition):
    set_values = ', '.join(f'{column} = {value}' for column, value in
update_columns.items())
    query = f"UPDATE {table_name} SET {set_values} WHERE {condition}"
    execute_query(query)

def get_table_data(table_name):
    query = f"SELECT * FROM {table_name}"
    return execute_query(query)

def insert_data_from_xml(xml_file):
    tree = ET.parse(xml_file)
    root = tree.getroot()

    for table in root:
        table_name = table.tag
        for row in table:
            columns = ', '.join(row.tag for row in row)
            values = ', '.join(f'{row.text}' for row in row)
            query = f"INSERT INTO {table_name} ({columns}) VALUES ({values})"
            execute_query(query)

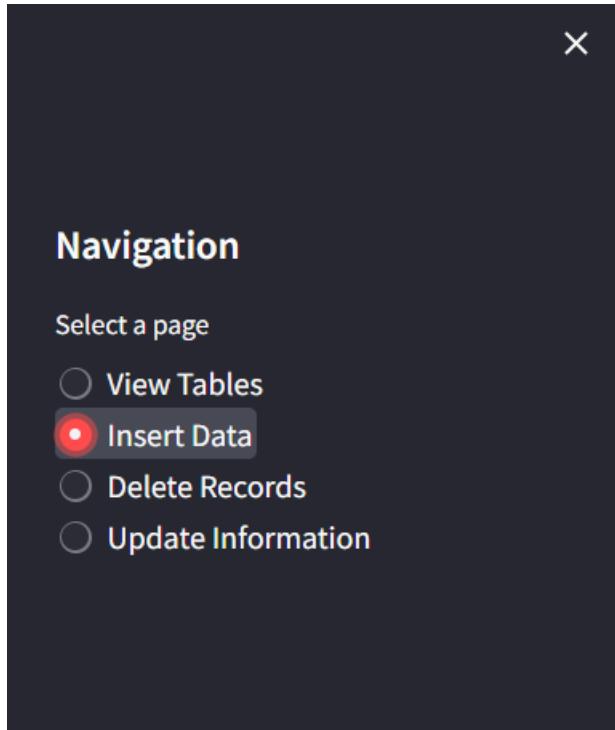
def update_book_from_xml(xml_file):
    tree = ET.parse(xml_file)
    root = tree.getroot()

    for table in root:
        table_name = table.tag
        for row in table:
            book_id = row.find('BookID').text

```

```
        update_columns = ', '.join(f'{row.tag} = {row.text}' for row in
row if row.tag != 'BookID')
        query = f"UPDATE {table_name} SET {update_columns} WHERE BookID =
{book_id}"
        execute_query(query)

if __name__ == '__main__':
    # Run necessary backend tasks, such as populating the database from XML
files
    insert_data_from_xml('populate_db.xml')
    update_book_from_xml('update_1book.xml')
```



# Online Bookstore Management

## Insert Data

- Insert data into "users" table
- Insert data into "books" table
- Insert data into "carts" table

# Online Bookstore Management

## View Tables

Select a table to view

books

Displaying data from the books table:

	BookID	Title	Author	Price	Quantity
0	1	The Great Gatsby	F. Scott Fitzgerald	90	7
1	2	To Kill a Mockingbird	by Harper Lee	100	4
2	3	The Da Vinci Code	Dan Brown	45	12
3	4	Dune	Frank Herbert	85	10

# Insert Data

Insert data into "users" table

## Insert Data into Users Table:

Username

Nagaveni Gowda

Password

nagu1234

Insert User Data

Data inserted successfully into 'users' table.

User data inserted successfully!

# View Tables

Select a table to view

users

Displaying data from the users table:

	UserID	Username	Password
0	1	Nagaveni Gowda	nagu1234
1	2	Tejas Gowda	Teju9090
2	3	Simran	simk756
3	4	smith	sm1234

# Online Bookstore Management

## View Tables

Select a table to view

carts

Displaying data from the carts table:

	CartID	UserID	BookID	Quantity
0	1	1	2	2
1	2	2	3	1

# Online Bookstore Management

## Delete Records

Select a table to delete records from

books

Delete records from the books table:

Enter the condition for deletion (e.g., booksID=1)

bookID=1

Delete Records

Record deleted successfully from the books table!

AFTER DELETING

# View Tables

Select a table to view

books

Displaying data from the books table:

	BookID	Title	Author	Price	Quantity
0	2	To Kill a Mockingbird	by Harper Lee	100	4
1	3	The Da Vinci Code	Dan Brown	45	12
2	4	Dune	Frank Herbert	85	10

After deleting CartID=1

# View Tables

Select a table to view

carts|

Displaying data from the carts table:

	CartID	UserID	BookID	Quantity
0	2	2	3	1

After deleting userID=1

# View Tables

Select a table to view

users|

Displaying data from the users table:

	UserID	Username	Password
0	2	Tejus Gowda	Teju9090
1	3	Simran	simk756
2	4	smith	sm1234

After updating quantity of the book to 4 for cartID=2

## Update Information

Select a table to update information in

carts

Update information in the carts table:

 **Update Information in Carts Table:**

BookID  
2

Quantity  
4

Enter the condition for updating (e.g., CartID=1)  
2

**Update Information**

Information updated successfully in the Carts table!

## View Tables

Select a table to view

carts

Displaying data from the carts table:

	CartID	UserID	BookID	Quantity
0	2	2	2	4

After updating users table

# Update Information

Select a table to update information in

users

Update information in the users table:

## Update Information in Users Table:

Username

smriti

Password

sm1234

Enter the condition for updating (e.g., UserID=1)

4

**Update Information**

Information updated successfully in the Users table!

# View Tables

Select a table to view

users

Displaying data from the users table:

	UserID	Username	Password
0	2	Tejas Gowda	Teju9090
1	3	Simran	simk756
2	4	smriti	sm1234
3	5	Nagaveni Gowda	nagu8776

After Updating Books Table

Update information in the books table:

## Update Information in Books Table:

Title

The Hunger Games

Author

Suzanne Collins

Price

90

Quantity

9

Enter the condition for updating (e.g., BookID=1)

4

**Update Information**

Information updated successfully in the Books table!

## View Tables

Select a table to view

books

Displaying data from the books table:

	BookID	Title	Author	Price	Quantity
0	2	To Kill a Mockingbird	by Harper Lee	100	4
1	3	The Da Vinci Code	Dan Brown	45	12
2	4	The Hunger Games	Suzanne Collins	90	9

Thank you 😊