CODE – For managing the Book Records

import streamlit as st

from create import create

from database import create\_table

from delete import delete

from read import read

from update import update

def main():

    st.title("Library Management System")

    menu = ["Add Books", "View Books", "Edit Books", "Remove Books"]

    choice = st.sidebar.radio("Menu", menu)

    create\_table()

    if choice == "Add Books":

        st.subheader("Enter Book Details:")

        create()

    elif choice == "View Books":

        st.subheader("View created records")

        read()

    elif choice == "Edit Books":

        st.subheader("Update created records")

        update()

    elif choice == "Remove Books":

        st.subheader("Delete created records")

        delete()

    else:

        st.subheader("About tasks")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

CODE – For managing the Student Membership

import streamlit as st

from createst import create

from databasest import create\_table

from deletest import delete

from readst import read

from updatest import update

def main():

    st.title("Student Membership Details")

    menu = ["Add Member", "View all members", "Edit members", "Remove members"]

    choice = st.sidebar.radio("Here is the list of operations you can perform", menu)

    create\_table()

    if choice == "Add Member":

        st.subheader("Enter student Details:")

        create()

    elif choice == "View all members":

        st.header("View added records")

        read()

    elif choice == "Edit members":

        st.subheader("Update added records")

        update()

    elif choice == "Remove members":

        st.subheader("Delete added records")

        delete()

    else:

        st.subheader("About tasks")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

CODE – For generating Report Records

import streamlit as st

from createre import create

from databasere import create\_table

from deletere import delete

from readre import read

from updatere import update

def main():

    st.title("Book Borrow Report")

    menu = ["Generate Report", "View all reports", "Extend date", "Delete report"]

    choice = st.sidebar.radio("Here is the list of operations you can perform", menu)

    create\_table()

    if choice == "Generate Report":

        st.subheader("Enter report details:")

        create()

    elif choice == "View all reports":

        st.subheader("View present reports")

        read()

    elif choice == "Extend date":

        st.subheader("Extend date")

        update()

    elif choice == "Delete report":

        st.subheader("Book has been returned")

        delete()

    else:

        st.subheader("About tasks")

if \_\_name\_\_ == '\_\_main\_\_':

    main()

Create

import streamlit as st

from database import add\_data

def create():

    col1, col2, col3, col4 = st.columns(4)

    with col1:

        num = st.text\_input("Book No")

        author = st.text\_input("Author")

    with col2:

        title = st.text\_input("Book Title")

        publisher = st.text\_input("Publisher")

    with col3:

        genre = st.text\_input("Genre")

        availability = st.selectbox("Availability",["Yes","No","Not supplying anymore"])

    with col4:

        des = st.text\_input("mini description")

        copies = st.number\_input("Number of copies")

    if st.button("Add this book to the library"):

        add\_data(num,title,genre,des,author,publisher,availability,copies)

        st.success("Successfully added the book: {}".format(title))

Database

import mysql.connector

mydb = mysql.connector.connect(

    host="localhost",

    user="root",

    password="Safiyasql@13",

    database="lib"

)

c = mydb.cursor()

def create\_table():

    c.execute('CREATE TABLE IF NOT EXISTS reports(repnum INT, sname TEXT, bnum INT, sid INT, issued DATE, returnd DATE)')

def add\_data(repnum,sname,bnum,sid,issued,returnd):

    c.execute('INSERT INTO reports(repnum,sname,bnum,sid,issued,returnd) VALUES (%s,%s,%s,%s,%s,%s)',(repnum,sname,bnum,sid,issued,returnd))

    mydb.commit()

def view\_all\_data():

    c.execute('SELECT \* FROM reports')

    data = c.fetchall()

    return data

def view\_only\_repnum():

    c.execute('SELECT repnum FROM reports')

    data = c.fetchall()

    return data

def get\_repnum(repnum):

    c.execute('SELECT \* FROM reports WHERE repnum="{}"'.format(repnum))

    data = c.fetchall()

    return data

def edit\_data(nrepnum,nsname,nbnum,nsid,nissued,nreturnd,repnum,sname,bnum,sid,issued,returnd):

    c.execute("UPDATE reports SET repnum=%s, sname=%s, bnum=%s, sid=%s, issued=%s, returnd=%s  WHERE repnum=%s and sname=%s and bnum=%s and sid=%s and issued=%s and returnd=%s ", (nrepnum,nsname,nbnum,nsid,nissued,nreturnd,repnum,sname,bnum,sid,issued,returnd))

    mydb.commit()

    data = c.fetchall()

    return data

def delete\_data(repnum):

    c.execute('SET FOREIGN\_KEY\_CHECKS=0')

    c.execute('DELETE FROM reports WHERE repnum="{}"'.format(repnum))

    mydb.commit()

import mysql.connector

mydb = mysql.connector.connect(

    host="localhost",

    user="root",

    password="Safiyasql@13",

    database="lib"

)

c = mydb.cursor()

def create\_table():

    c.execute('CREATE TABLE IF NOT EXISTS studs(sid INT, name TEXT, address TEXT,phone INT, expiry DATE, mem TEXT)')

def add\_data(sid,name,address,phone,expiry,mem):

    c.execute('INSERT INTO studs(sid,name,address,phone,expiry,mem) VALUES (%s,%s,%s,%s,%s,%s)',(sid,name,address,phone,expiry,mem))

    mydb.commit()

def view\_all\_data():

    c.execute('SELECT \* FROM studs')

    data = c.fetchall()

    return data

def view\_only\_sid():

    c.execute('SELECT sid FROM studs')

    data = c.fetchall()

    return data

def get\_sid(sid):

    c.execute('SELECT \* FROM studs WHERE sid="{}"'.format(sid))

    data = c.fetchall()

    return data

def edit\_stud\_data(nsid,nname,naddress,nphone,nexpiry,nmem,sid,name,address,phone,expiry,mem):

    c.execute("UPDATE studs SET sid=%s, name=%s, address=%s, phone=%s, expiry=%s, mem=%s WHERE sid=%s and name=%s and address=%s and phone=%s and expiry=%s and mem=%s ", (nsid,nname,naddress,nphone,nexpiry,nmem,sid,name,address,phone,expiry,mem))

    mydb.commit()

    data = c.fetchall()

    return data

def delete\_data(sid):

    c.execute('SET FOREIGN\_KEY\_CHECKS=0')

    c.execute('DELETE FROM studs WHERE sid="{}"'.format(sid))

    mydb.commit()

Update

import datetime

import pandas as pd

import streamlit as st

from databasest import view\_all\_data, view\_only\_sid, get\_sid, edit\_stud\_data

def update():

    result = view\_all\_data()

    df = pd.DataFrame(result, columns=['Student ID Number','Student Name','Address', 'Phone Number', 'Expiry date','Membership for a year'])

    with st.caption("Current Students"):

        st.dataframe(df)

    list\_of\_studs = [i[0] for i in view\_only\_sid()]

    selected\_stud = st.selectbox("Student details to Edit", list\_of\_studs)

    selected\_result = get\_sid(selected\_stud)

    if selected\_result:

        sid = selected\_result[0][0]

        name = selected\_result[0][1]

        address = selected\_result[0][2]

        phone = selected\_result[0][3]

        expiry = selected\_result[0][4]

        mem = selected\_result[0][5]

    col1, col2, col3 = st.columns(3)

    with col1:

        sid = st.text\_input("Student ID Number",sid)

        phone = st.number\_input("Phone Number",phone)

    with col2:

        name = st.text\_input("Student Name",name)

        expiry = st.date\_input("Membership Expiry Date",expiry)

    with col3:

        address = st.text\_input("Address",address)

        mem = st.selectbox("1 year membership",["Yes","No"])

        if st.button("Update Student Details"):

            edit\_stud\_data(nsid,nname,naddress,nphone,nexpiry,nmem,sid,name,address,phone,expiry,mem)

            st.success("Successfully updated details for student:: {} to ::{}".format(name,nname))

    result2 = view\_all\_data()

    df2 = pd.DataFrame(result2, columns=['Student ID Number','Student Name','Address', 'Phone Number', 'Expiry date','Membership for a year'])

    with st.expander("Updated data"):

        st.dataframe(df2)

Read

import pandas as pd

import streamlit as st

from database import view\_all\_data

def read():

    result = view\_all\_data()

    df = pd.DataFrame(result, columns=['Book no','Book Title','Genre', 'Description', 'Author','Publisher', 'Availability', 'copies'])

    with st.caption("View all Book Records"):

        st.dataframe(df)