Faculty of Engineering and Technology

Department of Computer Science and Engineering

Jain Global Campus, Kanakapura Taluk - 562112

Ramanagara District, Karnataka, India

A Project Report on "Book Collection System"

For the partial fulfillment of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

MANDARI SAI JAGAPATHI
17BTRCS097
RISWITHA
17BTRCS088

Faculty of Engineering & Technology

Department of Computer Science and Engineering

Jain Global campus
Kanakapura Taluk - 562112
Ramanagara District
Karnataka, India

CERTIFICATE

This is to certify that the Project work titled **"Book Management System"** for the course **Scripting Language Lab** during **5th semester,** is carried out by **Risiwtha (17btrcs088), M.Sai.Jagapathi (17btrcs097)** are bonafide students at the Faculty of Engineering & Technology, JAIN (Deemed-to-be-University), Bangalore in partial fulfilment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2019 - 2020**.

Prof. Deepa T.P	**Prof. Prasanth**	**DR.Narayanaswamy.R**
Assistant Professor Dept. of Computer Science and Engineering , Faculty of Engineering & Technology, JAIN (Deemed-to-be-University	Assistant Professor Dept. of Computer Science and Engineering , Faculty of Engineering & Technology, JAIN (Deemed-to-be-University)	Head Of Department Dept. of Computer Science and Engineering , Faculty of Engineering & Technology, JAIN (Deemed-to-be-University)

DECLARATION

We, **Riswitha (17btrcs088), M. Sai Jagapathi (17btrcs097)** are the students of fifth semester B.Tech in **Computer Science & Engineering**, at the Faculty of Engineering & Technology, **Jain University**. We hereby declare that the mini-project titled **"Book Collection System"** has been carried out by us and submitted in partial fulfilment for the course **17CS57S-SCRIPTING LANGUAGE LAB** as part of our degree in **Bachelor of Technology in Computer Science & Engineering** during the academic year **2019-2020**.

Name: Riswitha USN :17btrcs088
Name: M.Sai.Jagapathi USN :17btrcs097
Place : Bangalore Date :05-12-2019

ACKNOWLEDGEMENT

It is a great pleasure for me to acknowledge the assistance and support of a large number of individuals who have been responsible for the successful completion of this mini-project work.
We would like to thank ***Dr. Hariprasad S A,*** Director, Faculty of Engineering & Technology, Jain University and ***Dr. Kuldeep Sharma,*** Dean, School of Computer Science & Engineering, Jain University for their constant encouragement and expert advice.
It is a matter of immense pleasure to express our sincere thanks to ***Dr. Narayanaswamy R*** , Head of the department, Computer Science & Engineering, Jain University, for providing right academic guidance tha Faculty at made our task possible.
We would like to thank our guide **Prof.Prasanth** Assistant Professor, Dept. of Computer Science & Engineering, Jain University, for sparing his valuable time to extend help in the project work.
We would like to thank one and all who directly or indirectly helped us in completing the project work successfully.

Signature of Students

Abstract

Bookstore management system report in python. Today it is becoming very difficult to maintain records manually. Software system easily does the job of maintaining daily records as well as the transaction according to the user requirements. Only basic knowledge of computers is required for operations. The software system consists of all information of books and sold to the customer. The proposed system provides lots of facilities to the user to store information of the books and it provide information in quick time in a systematic manner. The processing time on the data is very fast .An online bookstore software projects that acts as a central database containing various books in stock along with their title, author and cost. This project is a website that acts as a central book store. This web project is developed using asp.net as the front end and sql as a back-end. The sql database stores various book related details. A user visiting the website can see a wide range of books arranged in respective categories. The user may select desired book and view its price. The user may even search for specific books on the website. Once the user selects a book , he then has to fill in a form and the book is booked for the user. It provides required data quickly to the user and also in specified manner to the user. There is a lot of duplicate woks, and chance of mistake when the records are changed they need to update each and every excel file. There is no option to find and print previous saved records there is no security, anybody can access any report and sensitive data. This bookshop management system is used to overcome the entire problem which they are facing currently, and making complete automation of manual systems to computerised system.

Chapter 1

Introduction

The proposed system provides lot of facility to the user to store information of the Books and it provides information in quick time in a systematic manner. The processing time on the data is very fast. It provides required data quickly to the user and also in specified manner to the user. All the information of Books changes is given to the user and also the reports are also generated according to the requirement of the user.

Today it is becoming very difficult to maintain record manually. This software system easily does the job of maintaining daily records as well as the transaction according to the user requirements.

Some advantages are:

User friendly software

Easy to record all data about the Books and its details.

It saves time of the user as well as prevent paper work

It provides the security to the database that is large amount of important data of the book.

It provides more reliability for keeping information.

1.1. Problem Definition

In the existing system all the transaction of books are done manually, So taking more time for a transaction like borrowing a book or returning a book and also for searching of members and books. Another major problem is on preparing the list of books borrowed and the available books in the library will take more time, currently it is doing as a one day process for verifying all records. So after conducting the feasibility study a decision to make the manual Library management system to be computerized one was taken to consideration. Some of the problems being faced that brought about the new system design are as follows

. Fast report generation is not possible.

Tracing a book is difficult.

➔ Information about issue/return of the books is not properly maintained.
➔ No central database can be created as information is
➔ Records are not available in database.

Objectives

- To enable administrator to access the system from anywhere.
- To reduce data redundancy and pruning ambiguous data.
- To process data in possibly the least amount of time.
- To secure the information and obstruct the access of unauthorized persons.
- To avail the information immediately in demand.
- To analyze the data for providing analytical decisions

1.2. Methodology

The problems which were faced in the bookstore/library can be solved with the help of this project by making a user friendly software. We planned to solve this project with python and mysql where python will give a frontend and backend will be given by mysql. The purpose of Software Quality Assurance Plan is to define all the techniques, procedures, and methodologies that will be used in the project to assure timely delivery of software that meets specified requirements within project resources. Software Quality Assurance involves reviewing and auditing the software products and activities to verify that they comply with the applicable procedures and standards and providing the software project and other appropriate managers with the results of these reviews and audits.

1.3. Software Requirements

- The software used for this project is python IDE.
- We have tkinter pre installed in pythons latest version.
- SQLite3 is inbuilt pip file which can be downloaded using cmd or python cmd in pycharm. As it is inbuilt the connection between the sql and python is easier to do.

1.4. Tool Description

Python: To run this we need Python. Python has inbuilt sql and tkinter in the lastest version.
Tkinter: To connect with this project we need Tkinter package.
SQLite: To run this database we need a mysql database.

Chapter 2

Implementation

2.1. Design & Implementation

This project can be used as a database input for library management, book store management by importing the front end and back end using. This will help in managing and storing the data of the books in a library and a few other sectors. Furthermore, we are able to keep the systems logic abstracted from both the end and the DBMS by sqlite3 to create a third tier architecture. Tkinter is used as GUI for letting the user interact with the database indirectly. We can import the front end and back end in other projects to solve various problems as we have a store already which gives a head start to a project. Data manipulation language DML is SQL is the statement which used to insert, update, delete and retrieve the data from the table or multiple tables as per requirement. The INSERT statement is used to insert a record in a table. UPDATE statement is used to modify the value of the attribute in a record and DELETE statement is used to delete a record from the table. In software engineering, a system development methodology refers to the framework that is used to structure, plan, and control the process of developing an information system. Software development methodologies define the processes we use to build software. These methodologies are also referred to as Software Development Process Models. Each methodology follows a series of steps unique to its type, to ensure success in the process of software development. A software process is a set of related activities that leads to the production of a software product. There are a lot of software processes but they all include four activities:

- ➔ Specification
- ➔ Development
- ➔ Validation
- ➔ Evolution

2.1.2. Source Code

FRONTEND

- This imports tkinter and connects python and database.

```
from tkinter import *
from backend import  Database
database = Database("books.db")
```

- Here we are creating text boxes and labels to take input from the user.

```
class Window(object):
   def __init__(self,window):
      self.window = window
      self.window.wm_title("The Book Store")
      l1 = Label(window, text="Title")
      l1.grid(row=0, column=0)
      l2 = Label(window, text="Author")
      l2.grid(row=0, column=2)
      l3 = Label(window, text="Year")
      l3.grid(row=1, column=0)
      l4 = Label(window, text="ISBN")
      l4.grid(row=1, column=2)
      self.title_text = StringVar()
      self.e1 = Entry(window, textvariable=self.title_text)
      self.e1.grid(row=0, column=1)
      self.author_text = StringVar()
      self.e2 = Entry(window, textvariable=self.author_text)
      self.e2.grid(row=0, column=3)

      self.year_text = IntVar()
      self.e3 = Entry(window, textvariable=self.year_text)
      self.e3.grid(row=1, column=1)
      self.ISBN_text = StringVar()
      self.e4= Entry(window, textvariable=self.ISBN_text)
      self.e4.grid(row=1, column=3)
      self.list1 = Listbox(window, height=6, width=35)
      self.list1.grid(row=2, column=0, rowspan=6, columnspan=2)
      self.list1.bind('<<ListboxSelect>>', self.get_selected_row)
```

- Now we need to attach a scrollbar to the listbox, and the other direction,too

```
      sb1 = Scrollbar(window)
```

```
        sb1.grid(row=2, column=2, rowspan=6)
        self.list1.config(yscrollcommand=sb1.set)
        sb1.config(command=self.list1.yview)
        b1 = Button(window, text="View all", width=12, command=self.view_command)
        b1.grid(row=2, column=3)
        b2 = Button(window, text="Search entry", width=12, command=self.search_command)
        b2.grid(row=3, column=3)
        b3 = Button(window, text="Add entry", width=12, command=self.add_command)
        b3.grid(row=4, column=3)
        b4 = Button(window, text="Update selected", width=12, command=self.update_command)
        b4.grid(row=5, column=3)
        b5 = Button(window, text="Delete selected", width=12, command=self.delete_command)
        b5.grid(row=6, column=3)
        b6 = Button(window, text="Close", width=12, command=window.destroy)
        b6.grid(row=7, column=3)
    def get_selected_row(self,event):   #the "event" parameter is needed b/c we've binded this
function to the listbox
        try:
            index = self.list1.curselection()[0]
            self.selected_tuple = self.list1.get(index)
            self.e1.delete(0,END)
            self.e1.insert(END,self.selected_tuple[1])
            self.e2.delete(0, END)
            self.e2.insert(END,self.selected_tuple[2])
            self.e3.delete(0, END)
            self.e3.insert(END,self.selected_tuple[3])
            self.e4.delete(0, END)
            self.e4.insert(END,self.selected_tuple[4])
        except IndexError:
            pass            #in the case where the listbox is empty, the code will not execute

        def view_command(self):
        self.list1.delete(0, END)
# make sure we've cleared all entries in the listbox every time we press the View all button
        for row in database.view():
            self.list1.insert(END, row)
    def search_command(self):
        self.list1.delete(0, END)
        for row in database.search(self.title_text.get(), self.author_text.get(), self.year_text.get(),
self.ISBN_text.get()):
            self.list1.insert(END, row)
    def add_command(self):
        if(Database.chenul(self.title_text.get()) and Database.chenul(self.author_text.get()) and
Database.chenul(self.ISBN_text.get())):
```

```
        database.insert(self.title_text.get(), self.author_text.get(), self.year_text.get(),
self.ISBN_text.get())
        self.list1.delete(0, END)
        self.list1.insert(END, (self.title_text.get(), self.author_text.get(), self.year_text.get(),
self.ISBN_text.get()))
    def delete_command(self):
        database.delete(self.selected_tuple[0])
        self.view_command()
    def update_command(self):
        #be careful for the next line ---> we are updating using the texts in the entries, not the selected
tuple
        database.update(self.selected_tuple[0],self.title_text.get(), self.author_text.get(),
self.year_text.get(), self.ISBN_text.get())
        self.view_command()
#code for the GUI (front end)
window = Tk()
Window(window)

window.mainloop()
```

BACKEND

```
import sqlite3
class Database:
    def __init__(self,db):
        self.conn = sqlite3.connect(db)
        self.cur = self.conn.cursor()
        self.cur.execute("CREATE TABLE IF NOT EXISTS book (id INTEGER , title TEXT,author TEXT, year
DATE, isbn INTEGER  PRIMARY KEY)")
        self.conn.commit()
    def insert(self,title, author, year, isbn):
        #the NULL parameter is for the auto-incremented id
        self.cur.execute("INSERT INTO book VALUES(NULL,?,?,?,?)", (title,author,year,isbn))
        self.conn.commit()
    def view(self):
        self.cur.execute("SELECT * FROM book")
        rows = self.cur.fetchall()
        return rows
    def search(self,title="", author="", year="YYYY", isbn=""):
        self.cur.execute("SELECT * FROM book WHERE title = ? OR author = ? OR year = ? OR isbn = ?",
(title, author, year, isbn))
        rows = self.cur.fetchall()
        return rows
    def delete(self,id):
        self.cur.execute("DELETE FROM book WHERE id = ?", (id,))
        self.conn.commit()
        #conn.close()
    def chenul(val):
        if val!="":
            return True
        return False
    def update(self,id, title, author, year, isbn):
        self.cur.execute("UPDATE book SET title = ?, author = ?, year = ?, isbn = ? WHERE id = ?", (title,
author, year, isbn, id))
        self.conn.commit()
    #destructor-->now we close the connection to our database here
    def __del__(self):
        self.conn.close()
```

Chapter 3

Results

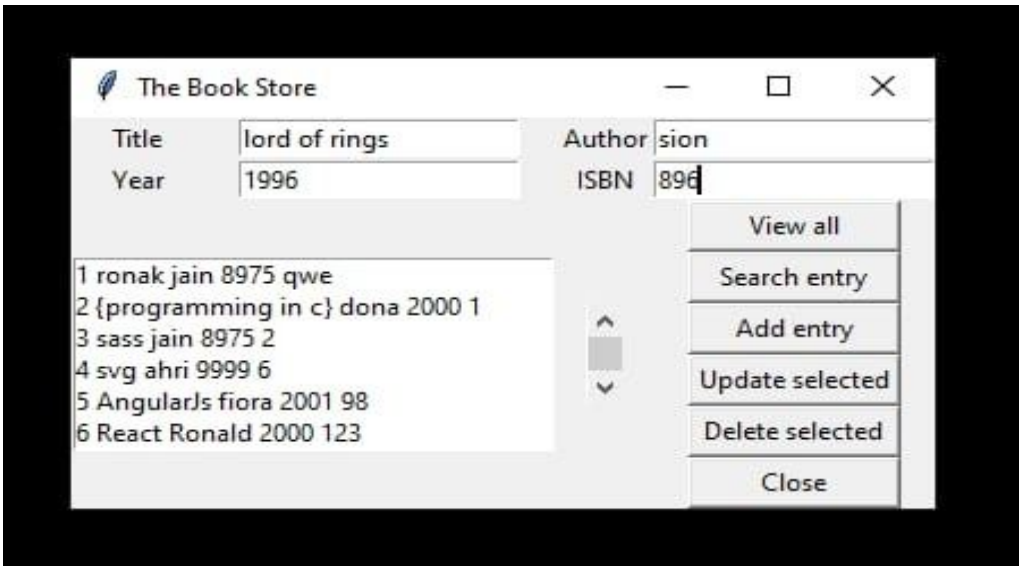

Fig. 3.1

Chapter 4

Conclusion

Book collection system report in python is an attempt to overcome the present in efficient and time consuming process of locating reserving and purchasing quality reading materials available in the shop. Through automated book shop solution, provide an easy way of searching for reserving and purchasing of books. It's worth analysing and identifying the benefits as it would directly influence the productivity of the shop.This software can be easily implemented under various situations. Any education institute can make use of it for providing information about the author, the content of the available books in their library. Modifications can be easily done according to requirements and when necessary. It can be used in any type of Book Shop for managing all the sales and purchasing activities and managing the data records related to Book house.

References

1. Google
2. Python.docx
3. Sql
4. Youtube
5. Github
6. Quora