

بسم الله الرحمن الرحيم

گزارش کار پروژه سیستم مدیریت کتابخانه

محمد امین کمالی

9712358038

سید علیرضا امامی

9712358001

استاد: جناب دکتر مرتضی یوسف صنعتی

کمک استاد: سرکار خانم مهندس باب الحوائجی

فهرست

3.....	مقدمه
3.....	ارکان اصلی
3.....	صندوقدار
4.....	بخشدار
4.....	اعضا
5.....	کتاب ها
5.....	قفسه ها
6.....	فروش کتابها
6.....	امانت گذاری کتابها
7.....	رویت اطلاعات
13.....	اضافه کردن اطلاعات
19.....	حذف کردن اطلاعات
22.....	Homepage
23.....	Showingpage
24.....	Addpage
25.....	Deletepage
26.....	Borrowingpage
27.....	Buyingpage

مقدمه:

همانطور که در پروپزال به آن اشاره کردیم کار اصلی ما در کتابخانه فروش و قرض دادن کتاب به اعضای ثبت نام شده در کتابخانه هست .

در کتابخانه رکن های اصلی وجود دارد که این ارکان اصلی در کنار یکدیگر به طور درست یک کتابخانه کامل را تشکیل میدهند.

زبان های مورد استفاده در پروژه:-(Django(backend)-Microsoft SQL Server(database)
html(frontend)

از ارکان اصلی کتابخانه میتوان به :صندوقدار- بخشدار- اعضا – کتاب - قفسه ها اشاره کرد که هر کدام را به طور کامل توضیح داده میشود:

1. صندوق دار:

همانطور که از اسم آن پیدا هست بخش اصلی وظیفه صندوقدار در جابه جایی کتاب ها به صورت فروش یا امانت و ثبت نام اعضای جدید است.

از اطلاعات یک صندوقدار میتوان به پارامتر های زیر اشاره کرد:

نام و نام خانوادگی-کد صندوقداری(مهمترین رکن اطلاعات یک صندوقدار)-تاریخ تولد- میزان حقوق-شماره تلفن-محل سکونت که به سه قسمت شهر خیابان و پلاک تقسیم میشود.

cssn (PK, int, not null)	
Fname (varchar(15), not null)	cassn (PK, FK, int, not null)
Lname (varchar(15), not null)	hnum (PK, varchar(15), not null)
Birthday (date, null)	Cpssn (PK, FK, int, not null)
Salary (int, not null)	street (PK, varchar(15), not null)
	Phonenum (PK, int, not null)
	city (PK, varchar(15), not null)

تمام اطلاعات بالا اطلاعاتی است که در قبل آنها را ذکر کردیم به صورت SQL

علت سه دستگی آنها برای این است که میدانییم شماره تلفن یک پارامتر چند مقداره است یعنی میتوان از آن به تعداد بیشتر از یکی برای یک صندوقدار داشت.

دسته بعدی که جدا شده اطلاعات مکانی یک صندوقدار است که میدانییم یک پارامتر سه متغیره است به همین دلیل آنها نیز در یک جدول جدا هستند ولی این جدا بودن جدول ها به معنای جدایی کامل آنها نیست بلکه این دو جدول با کلید های خارجی به جدول اصلی صندوقدار وصل هستند و اطلاعات شماره تلفن و آدرس نیز قابلیت ذخیره سازی یا حذف را دارد.

2. بخشدار:

بخشدار مانند صندوقدار رکن مهم و اساسی به حساب می آید به این صورت که وظیفه او راهنمایی اعضا به قفسه های مرتبط به کتاب مورد نیاز آنها است.

از اطلاعات یک بخشدار میتوان به پارامتر های زیر اشاره کرد:

نام و نام خانوادگی-کد بخشداری(مهمترین رکن اطلاعات یک بخشدار)-تاریخ تولد-میزان حقوق-شماره تلفن-محل سکونت که به سه قسمت شهر خیابان و پلاک تقسیم میشود. قابل ذکر است که هر قفسه به یک بخشدار مرتبط است این برای سهولت کار در کتابخانه و تسریع کارها هست.

Shssn (PK, int, not null)	
Fname (varchar(15), not null)	shssn (PK, FK, int, not null)
Lname (varchar(15), not null)	hnum (PK, varchar(15), not null)
Birthday (date, null)	shssn (PK, FK, int, not null)
Salary (int, not null)	pnum (PK, int, not null)
	street (PK, varchar(15), not null)
	city (PK, varchar(15), not null)

بخشدار نیز مثل صندوقدار از سه قسمت برای ثبت اطلاعات استفاده میکند که این سه جدول با کلید های خارجی به یکدیگر متصل هستند.

3. اعضا:

اعضا همان مراجعہ کنندگان به کتابخانه هستند و میدانیم که هر مراجعہ کننده برای خرید یا قرض یک کتاب نیاز به یک حساب کاربری یا به اصطلاح یک کارت عضویت دارد تا با آن بتوان هویت فرد را فهمید. همیشه عضویت ها تا یک تاریخ مشخص شده وجود دارد.

از اطلاعات هر یک از اعضا میتوان به پارامتر های زیر اشاره کرد:

نام و نام خانوادگی- کد عضویت(مهم ترین رکن اطلاعات یک عضو کتابخانه)-تاریخ تولد- تاریخ عضویت-تاریخ انقضای عضویت-بخشداری که او را عضو کتابخانه کرده-شماره تلفن.

Ssn (PK, int, not null)	
Fname (varchar(15), not null)	
Lname (varchar(15), not null)	
Birthday (date, null)	
Regdate (date, not null)	
Expdate (date, not null)	Mssn (PK, FK, int, not null)
Cpsn (FK, int, not null)	Phonenum (PK, int, not null)

4. کتابها:

از مهمترین قسمت های یک کتاب خانه کتاب های آن است که اصلا به همان اسم هم شناخته میشوند.

در هر کتاب خانه کتاب های متفاوتی از هر ژانر هنری وجود دارد که هر کس بنا به علاقه و نیاز خود این کتاب ها را انتخاب میکند یا برای خرید و یا به امانت گرفتن.

ما کتاب هارا از روی ژانر آنها میشناسیم ولی باید به این دید برسیم که هر کتاب یک ژانر اصلی و یک ژانر خاص برای خود دارد که در ادامه این قسمت توضیح داده میشود

از اطلاعات هر یک از کتابها میتوان به پارامتر های زیر اشاره کرد:

اسم کتاب-کد کتاب-تاریخ چاپ یک کتاب-ژانر یک کتاب-قفسه ای که در آن کتاب موجود است

```
PK Bcode (PK, int, not null)
Bname (varchar(15), null)
Mdate (date, not null)
Genre (varchar(15), not null)
Price (int, not null)
BSid (FK, int, not null)
```

منظور از قفسه کتاب این است که هر کتاب در درون خود یک ژانر دارد مثل شاهنامه که به آن کتاب ادبیاتی گفته میشود ولی در ژانر کتاب های افسانه ای و داستانی نیز حضور دارد. برای همین هست که میگوییم یک کتاب دو ژانر متفاوت دارید یکی ژانر مخصوص کتاب و دیگر ژانر کلی آن است.

5. قفسه ها:

از اطلاعات هر یک از قفسه ها میتوان به پارامتر های زیر اشاره کرد:

ژانر قفسه-کد قفسه-بخشدار ی که وظیفه مراقبت و راهنمایی قفسه را دارد.

```
PK Id (PK, int, not null)
Genre (varchar(15), null)
Sgssn (FK, int, not null)
```

دلیل اتصال قفسه به بخشدار برای این است که هر بخشدار یک قفسه خاص را میتواند کنترل کند نسبت به اطلاعات خود و علاقه قبلی.

ما در کتاب خانه ها علاوه بر رکن های اصلی بالا دو رکن دیگر به اسم های فروش و یا امانت گذاری کتاب هارا نیز داریم که به صورت زیر هستند:

1. فروش کتاب:

مراحل خرید یک کتاب در کتاب خانه به این صورت است که یک عضو با کد عضویت به یک صندوقدار با شماره صندوقداری مراجعه میکند و کتاب را از طریق صندوقدار خریداری میکند.

از پارامتر های وجودی در فروش کتاب میتوان به پارامتر های پایین اشاره کرد:
کد اعضا- کد صندوق داری که کتاب را به فروش میرساند-کد کتاب فروخته شده

- ☛ sspssn (PK, FK, int, not null)
- ☛ sbcode (PK, FK, int, not null)
- ☛ smssn (PK, FK, int, not null)

2. امانت گذاری کتاب:

هر کتابی که به امانت گذاشته میشود یک مدت زمان محدودی برای استفاده و برگردتن به کتاب به کتابخانه را داراست.

از پارامتر های وجودی در امانت گذاری کتاب میتوان به پارامتر های پایین اشاره کرد:
کد اعضا- کد صندوق داری که کتاب را به امانت میرساند-کد کتاب فروخته شده-مدت زمان امانت گذاری شده.

- ☛ Ccssn (PK, FK, int, not null)
- ☛ Sbcode (PK, FK, int, not null)
- ☛ Mssn (PK, FK, int, not null)
- ☐ Bexdate (date, null)

پس از تعریف دیتا بیس اصلی کتاب خانه حال باید به این بپردازیم که چگونه میتوان اطلاعات را رویت کرد یا ویراش کرد(حذف و اضافه کردن اطلاعات) و اصلا چگونه میتوان یک کتاب را خرید یه به امانت گرفت حال وقت آن است که دست به کار شویم و راه هایی برای ثبت یا حذف اطلاعات بگذاریم:

1.رویت اطلاعات:

رویت اطلاعات برای هر 5 بخش مثل یکدیگر هست تنها فرق آنها در کویری آنها و فراخوانی اطلاعات مورد نیاز است. به دلیل شباهت بالا کلیات همه 5 بخش را توضیح میدهیم و سپس کویری های متفاوت آنها را نیز به نمایش میگذاریم: به طور مثال برای نشان دادن اطلاعات یک عضو در ابتدا باید قسمتی طراحی کنیم به اسم member که میشود اپلیکشین ما و تمام تغییرات به مرور به آن اضافه میشود بدین صورت:

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'cashier',  
    'member',  
    'shelfguy',  
    'book',  
    'bookshelf',  
    'borrow',  
    'buy',  
]
```

در قسمت setting.py و در قسمت installed apps مثل تعریف بالا تمام اپلیکشن ها را تعریف میکنیم. تعریفی های این پروژه 7 اپلیکشن انتهایی هستند.

در ادامه باید به طراحی قسمتی برسیم که به پایگاه داده خود اتصال پیدا کنیم به این صورت:

```
conn=pyodbc.connect('Driver={ODBC Driver 17 for SQL Server};'  
                    'Server=DESKTOP-IVD447A;'  
                    'Database=librarymg;'  
                    'Trusted Connection=yes;')
```

این دستور به این معنا است که شما الان میتوانید از دیتابیس خود استفاده کامل را داشته باشید.

حال بعد از اتصال به پایگاه داده ما باید اطلاعات را به صورت کوئری به دیتابیس بدهیم تا آنها را دسته بندی کند به صورت زیر:

```
cursor = conn.cursor()  
cursor.execute("select *,DATEDIFF(year, [Birthday], GETDATE()) AS age from  
[librarymg.Member] m left join [librarymg.Memphone] p on p.Mssn=m.Ssn")  
result = cursor.fetchall()
```

اگر به عکس بالا دقت کنید متوجه آن میشوید که ما اطلاعات را به صورت یک کوئری می‌دهیم به دیتابیس تا آنها را برای ما مرتب کند

حال باید به این مرحله برسیم که چگونه میتوان اصلا اطلاعات را نمایش داد.

برای این کار به یک زبان فرانت نیاز داریم مثل html:

در این مرحله ما یک فایل به زبان html تولید میکنیم

کلیت تمام فایل های فرانتی یکی است ولی قسمت متفاوت آنها قسمت پایین است:

```
<table border="1">  
  <tr>  
    <th>Ssn</th>  
    <th>Fname</th>  
    <th>Lname</th>  
    <th>Phone Number</th>  
    <th>Birthday</th>  
    <th>Age</th>  
    <th>Register Date</th>  
    <th>Expire Date</th>  
    <th>Refrenced Cashier Number</th>  
  </tr>  
  {%for datadisplay in sqlservconn%}  
    <tr>  
      <td>{{datadisplay.Ssn}}</td>  
      <td>{{datadisplay.Fname}}</td>  
      <td>{{datadisplay.Lname}}</td>  
      <td>{{datadisplay.Phonenum}}</td>  
      <td>{{datadisplay.Birthday}}</td>  
      <td>{{datadisplay.age}}</td>  
      <td>{{datadisplay.Regdate}}</td>  
      <td>{{datadisplay.Expdate}}</td>  
      <td>{{datadisplay.Cpssn}}</td>  
    </tr>  
  {%endfor%}  
</table>
```


در این مرحله در فایل فرانتی ما هیچ چیز قابل نمایش نیست چون اصلا به قسمت قبلی که توضیح دادیم اتصالی ندارد پس ما باید برای آنها اتصال ایجاد کنیم:

```
return render(request, 'member\member.html',{'sqlservconn':result})
```

حال با خط بالا میتوانیم اطلاعات را به نمایش در بیاوریم

این کلیت تمام 7 جدول موجود در کتابخانه هست تنها قسمت متفاوت آنها در کوئری مورد نظر و نمایش html آن است که در پایین هر کدام را برای راحتی کار خواننده به نمایش در میاوریم:

برای صندوقدار :

```
cursor = conn.cursor()
cursor.execute("select *, DATEDIFF(year, [Birthday], GETDATE()) AS age from
[librarymg.Cashier] left join [librarymg.Cashphone] on cssn=Cpssn left join
[librarymg.Cashaddress] on cssn=cassn")
result = cursor.fetchall()
```

فایل فرانتی آن:

```
<table border="1">
  <tr>
    <th>cssn</th>
    <th>Fname</th>
    <th>Lname</th>
    <th>birthday</th>
    <th>salary</th>
    <th>phone number</th>
    <th>city</th>
    <th>street</th>
    <th>house number</th>
    <th>age</th>
  </tr>
  {%for datadisplay in sqlservconn%}
  <tr>
    <td>{{datadisplay.cssn}}</td>
    <td>{{datadisplay.Fname}}</td>
    <td>{{datadisplay.Lname}}</td>
    <td>{{datadisplay.Birthday}}</td>
    <td>{{datadisplay.Salary}}</td>
    <td>{{datadisplay.Phonenum}}</td>
    <td>{{datadisplay.city}}</td>
    <td>{{datadisplay.street}}</td>
    <td>{{datadisplay.hnum}}</td>
    <td>{{datadisplay.age}}</td>
  </tr>
  {%endfor%}
</table>
```

برای بخشداری:

```
cursor = conn.cursor()
cursor.execute("select *,DATEDIFF(year, [Birthday], GETDATE()) AS age from
[librarymg.Shelfguy] s left join [librarymg.Shelfphone] p on p.shssn=s.Shssn
left join [librarymg.shelfaddress] a on s.Shssn=a.shssn")
result = cursor.fetchall()
```

قسمت فرانتی آن:

```
<table border="1">
  <tr>
    <th>Shelfguy_ssn</th>
    <th>Fname</th>
    <th>Lname</th>
    <th>birthday</th>
    <th>salary</th>
    <th>phone number</th>
    <th>city</th>
    <th>street</th>
    <th>house number</th>
    <th>age</th>

  </tr>
  {%for datadisplay in sqlservconn%}
  <tr>
    <td>{{datadisplay.Shssn}}</td>
    <td>{{datadisplay.Fname}}</td>
    <td>{{datadisplay.Lname}}</td>
    <td>{{datadisplay.Birthday}}</td>
    <td>{{datadisplay.Salary}}</td>
    <td>{{datadisplay.pnum}}</td>
    <td>{{datadisplay.city}}</td>
    <td>{{datadisplay.street}}</td>
    <td>{{datadisplay.hnum}}</td>
    <td>{{datadisplay.age}}</td>
  </tr>
  {%endfor%}
```

برای کتاب ها:

```
cursor = conn.cursor()
cursor.execute("select * from [librarymg.Book]")
result = cursor.fetchall()
```

قسمت فرانتی آن:

```
<table border="1">
  <tr>
    <th>Book Code</th>
    <th>Book Name</th>
    <th>Manufacture Date</th>
    <th>Genre</th>
    <th>Price</th>
    <th>Book_Shelf Number</th>
  </tr>
  {%for datadisplay in sqlservconn%}
  <tr>
    <td>{{datadisplay.Bcode}}</td>
    <td>{{datadisplay.Bname}}</td>
    <td>{{datadisplay.Mdate}}</td>
    <td>{{datadisplay.Genre}}</td>
    <td>{{datadisplay.Price}}</td>
    <td>{{datadisplay.Bsid}}</td>
  </tr>
  {%endfor%}
</table>
```

برای قفسه ها:

```
cursor = conn.cursor()
cursor.execute("select * from [librarymg.Bookshelf] a, [librarymg.Book] b
where b.Bsid=a.Id and b.Bcode not in( select sbcode from [librarymg.Buy] )
and b.Bcode not in( select Sbcode from [librarymg.Borrow] )")
result = cursor.fetchall()
```

قسمت فرانتی آن:

```
<table border="1">
  <tr>
    <th>Shelf Number</th>
    <th>Shelf Genre</th>
    <th>Shelfguy Number</th>
    <th>books</th>
  </tr>
  {%for datadisplay in sqlservconn%}
  <tr>
    <td>{{datadisplay.Id}}</td>
    <td>{{datadisplay.Genre}}</td>
    <td>{{datadisplay.Sgssn}}</td>
    <td>{{datadisplay.Bname}}</td>
  </tr>
  {%endfor%}
</table>
```

برای کتاب های فروخته شده:

```
cursor = conn.cursor()
cursor.execute("select * from [librarymg.Buy] b join [librarymg.Member] m on b.smssn=m.Ssn join
[librarymg.Book] a on b.sbcode=a.Bcode")
result = cursor.fetchall()
```

قسمت فرانتی آن:

```
<table border="1">
  <tr>
    <th>member Number</th>
    <th>member lname</th>
    <th>cashier Number</th>
    <th>book code</th>
    <th> book name</th>
    <th>book price</th>
  </tr>
  {%for datadisplay in sqlservconn%}
  <tr>
    <td>{{datadisplay.smssn}}</td>
    <td>{{datadisplay.Lname}}</td>
    <td>{{datadisplay.sspssn}}</td>
    <td>{{datadisplay.sbcode}}</td>
    <td>{{datadisplay.Bname}}</td>
    <td>{{datadisplay.Price}} T</td>
  </tr>
  {%endfor%}
</table>
```

و برای کتاب های امانت گذاشته شده:

```
cursor = conn.cursor()
cursor.execute("select *,DATEDIFF(day, GETDATE(), [Bexdate] ) AS reamaining_time from
[librarymg.Borrow] b join [librarymg.Member] m on b.Mssn=m.Ssn join [librarymg.Book] a on
b.Sbcode=a.Bcode")
result = cursor.fetchall()
```

قسمت فرانتی آن:

```
<table border="1">
  <tr>
    <th>member Number</th>
    <th>member lname</th>
    <th>cashier Number</th>
    <th>book code</th>
    <th> book name</th>
    <th>expire date</th>
    <th>remaining days</th>
  </tr>
  {%for datadisplay in sqlservconn%}
  <tr>
    <td>{{datadisplay.Mssn}}</td>
    <td>{{datadisplay.Lname}}</td>
    <td>{{datadisplay.Cssn}}</td>
    <td>{{datadisplay.Sbcode}}</td>
    <td>{{datadisplay.Bname}}</td>
    <td>{{datadisplay.Bexdate}}</td>
    <td>{{datadisplay.reamaining_time}} Days</td>
  </tr>
  {%endfor%}
</table>
```

2. اضافه کردن اطلاعات:

مانند رویت کردن اطلاعات به تعریف اضافه کردن اطلاعات میپردازیم:
دوباره در ابتدا به دیتا بیس مورد نظر خود وصل میشویم

```
def insert (request):  
    conn=pyodbc.connect('Driver={ODBC Driver 17 for SQL Server};'  
                        'Server=DESKTOP-IVD447A;'  
                        'Database=librarymg;'  
                        'Trusted_Connection=yes;')
```

سپس به قسمت فرانتی مراجعه میکنیم :

```
<form method="POST">  
    {% csrf_token %}  
    <table border="1" style=" background-color:rgb(0, 247, 255)">  
        <tr>  
            <td>SSN</td>  
            <td><input type="text" placeholder="member Ssn.." name="member_ssn"></td>  
        </tr>  
        <tr>  
            <td>Fname</td>  
            <td><input type="text" placeholder="member Fname.." name="member_fname"></td>  
        </tr>  
        <tr>  
            <td>Lname</td>  
            <td><input type="text" placeholder="member Lname.." name="member_lname"></td>  
        </tr>  
        <tr>  
            <td>Birthday</td>  
            <td><input type="text" placeholder="member Birthday.." name="member_birthday"></td>  
        </tr>  
        <tr>  
            <td>Rgister date</td>  
            <td><input type="text" placeholder="member Date.." name="member_regdate"></td>  
        </tr>  
        <tr>  
            <td>Expire date</td>  
            <td><input type="text" placeholder="member Expdate.." name="member_expdate"></td>  
        </tr>  
        <tr>  
            <td>Cashier SSN</td>  
            <td><input type="text" placeholder="member Cpssn.." name="member_cpssn"></td>  
        </tr>  
        <tr>  
            <td>Member Phone</td>  
            <td><input type="text" placeholder="memphone Phonenum.." name="memphone_phonenum"></td>  
        </tr>  
    </table>
```

سپس به این میپردازیم که اطلاعات فرانتی را به صورت کویری در بیاوریم:

```
if request.method=="POST":
    if request.POST.get('member_ssn') and request.POST.get('member_fname')
and request.POST.get('member_lname') and request.POST.get('member_birthday')
and request.POST.get('member_regdate') and request.POST.get('member_expdate')
and request.POST.get('member_cpssn'):
        cursor = conn.cursor()
        cursor2= conn.cursor()
        cursor.execute("insert into [librarymg.Member] values(' " +
request.POST.get('member_ssn')+ " ', ' " + request.POST.get('member_fname')
+ " ', ' " + request.POST.get('member_lname') + " ', ' " +
request.POST.get('member_birthday') + " ', ' " +
request.POST.get('member_regdate') + " ', ' " +
request.POST.get('member_expdate') + " ', ' " +
request.POST.get('member_cpssn') + " ')" )
        cursor2.execute("insert into [librarymg.Memphone] values(' " +
request.POST.get('member_ssn') + " ', ' " +
request.POST.get('memphone_phonenum') + " ')" )
        cursor.commit()
        cursor2.commit()
```

سپس هم فرانت هم کویری را به هم وصل میکنیم:

```
return render( request, 'member\insert.html')
```

اینها کلیت قسمت اضافه کردن هستند

حال به قسمت هایی میپردازیم که قسمت های خاص هر کدام میپردازیم:
برای صندوق دار:

```
cursor = conn.cursor()
cursor2 = conn.cursor()
cursor3 = conn.cursor()
cursor.execute("insert into [librarymg.Cashier] values(' " +
request.POST.get('cashier_cssn')+ " ', ' " +
request.POST.get('cashier_fname') + " ', ' " +
request.POST.get('cashier_lname') + " ', ' " +
request.POST.get('cashier_birthday') + " ', ' " +
request.POST.get('cashier_salary') + " ')" )
cursor2.execute("insert into [librarymg.Cashphone] values(' " +
request.POST.get('cashier_cssn') + " ', ' " +
request.POST.get('cashier_phonenum') + " ')" )
cursor3.execute("insert into [librarymg.Cashaddress] values(' " +
request.POST.get('cashier_cssn') + " ', ' " +
request.POST.get('cashier_hnum') + " ', ' " +
request.POST.get('cashier_street') + " ', ' " +
request.POST.get('cashier_city') + " ')" )
cursor.commit()
cursor2.commit()
cursor3.commit()
```

```
<form method="POST">
  {% csrf_token %}
  <table border="1" style=" background-color:rgb(0, 247, 255)">
    <tr>
      <td>SSN</td>
      <td><input type="text" placeholder="cashier ssn.."
name="cashier_cssn"></td>
    </tr>
    <tr>
      <td>Fname</td>
      <td><input type="text" placeholder="cashier Fname.."
name="cashier_fname"></td>
    </tr>
    <tr>
      <td>Lname</td>
      <td><input type="text" placeholder="cashier Lname.."
name="cashier_lname"></td>
    </tr>
    <tr>
      <td>Birthday</td>
      <td><input type="text" placeholder="cashier Birthday.."
name="cashier_birthday"></td>
    </tr>
    <tr>
      <td>Salary</td>
      <td><input type="text" placeholder="cashier Salary.."
name="cashier_salary"></td>
    </tr>
    <tr>
      <td>Phone Number</td>
      <td><input type="text" placeholder="cashier phone.."
name="cashier_phonenum"></td>
    </tr>
    <tr>
      <td>City</td>
      <td><input type="text" placeholder="cashier address.."
name="cashier_city"></td>
    </tr>
    <tr>
      <td>Street</td>
      <td><input type="text" placeholder="cashier address.."
name="cashier_street"></td>
    </tr>
    <tr>
      <td>House Number</td>
      <td><input type="text" placeholder="cashier address"
name="cashier_hnum"></td>
    </tr>
  </table>
```

برای بخشداری:

```
cursor = conn.cursor()
cursor2 = conn.cursor()
cursor3 = conn.cursor()
cursor.execute("insert into [librarymg.Shelfguy] values(' " +
request.POST.get('shguy_cssn')+ " ' , ' " + request.POST.get('shguy_fname') + " ' , ' " +
request.POST.get('shguy_lname') + " ' , ' " + request.POST.get('shguy_birthday') +
" ' , ' " + request.POST.get('shguy_salary') + " ' )" )
cursor2.execute("insert into [librarymg.Shelfphone] values(' " +
request.POST.get('shguy_cssn') + " ' , ' " + request.POST.get('shguy_phonenum') + " ' )" )
cursor3.execute("insert into [librarymg.Shelfaddress] values(' " +
request.POST.get('shguy_cssn') + " ' , ' " + request.POST.get('shguy_hnum') + " ' , ' " +
request.POST.get('shguy_street') + " ' , ' " + request.POST.get('shguy_city') + " ' )" )
cursor.commit()
cursor2.commit()
cursor3.commit()
```

قسمت فرانتی آن:

```
<form method="POST">
  {% csrf_token %}
  <table border="1" style=" background-color:rgb(0, 247, 255)">
    <tr>
      <td>SSN</td>
      <td><input type="text" placeholder="shguy ssn.." name="shguy_cssn"></td>
    </tr>
    <tr>
      <td>Fname</td>
      <td><input type="text" placeholder="shguy Fname.." name="shguy_fname"></td>
    </tr>
    <tr>
      <td>Lname</td>
      <td><input type="text" placeholder="shguy Lname.." name="shguy_lname"></td>
    </tr>
    <tr>
      <td>Birthday</td>
      <td><input type="text" placeholder="shguy Birthday.." name="shguy_birthday"></td>
    </tr>
    <tr>
      <td>Salary</td>
      <td><input type="text" placeholder="shguy Salary.." name="shguy_salary"></td>
    </tr>
    <tr>
      <td>Phone Number</td>
      <td><input type="text" placeholder="shguy phone.." name="shguy_phonenum"></td>
    </tr>
    <tr>
      <td>City</td>
      <td><input type="text" placeholder="shguy address.." name="shguy_city"></td>
    </tr>
    <tr>
      <td>Street</td>
      <td><input type="text" placeholder="shguy address.." name="shguy_street"></td>
    </tr>
    <tr>
      <td>House Number</td>
      <td><input type="text" placeholder="shguy address" name="shguy_hnum"></td>
    </tr>
  </table>
```


برای کتاب ها:

```
cursor = conn.cursor()
cursor.execute("insert into [librarymg.Book] values(' " +
request.POST.get('book_bcode')+ " ', ' " + request.POST.get('book_bname') +
" ', ' " + request.POST.get('book_mdate') + " ', ' " +
request.POST.get('book_genre') + " ', ' " + request.POST.get('book_price')
+ " ', ' " + request.POST.get('book_bsid') + " ')" )
cursor.commit()
```

قسمت فرانتی آن:

```
<form method="POST">
    {% csrf_token %}
    <table border="1" style=" background-color:rgb(0, 247, 255)">
        <tr>
            <td>Book Code</td>
            <td><input type="text" placeholder="book Bookcode.."
name="book_bcode"></td>
        </tr>
        <tr>
            <td>Book Name</td>
            <td><input type="text" placeholder="book name.."
name="book_bname"></td>
        </tr>
        <tr>
            <td>Man_Date</td>
            <td><input type="text" placeholder="book manufature_date.."
name="book_mdate"></td>
        </tr>
        <tr>
            <td>Genre</td>
            <td><input type="text" placeholder="book Genre.."
name="book_genre"></td>
        </tr>
        <tr>
            <td>Price</td>
            <td><input type="text" placeholder="book Price.."
name="book_price"></td>
        </tr>
        <tr>
            <td>Shelf_book Code</td>
            <td><input type="text" placeholder="book shelf_id.."
name="book_bsid"></td>
        </tr>
    </table>
```

برای خرید کتابها:

```
cursor = conn.cursor()
cursor.execute("insert into [librarymg.Buy] values('" + request.POST.get('buy_cssn') +
" , ' " + request.POST.get('buy_bcode') + " , ' " + request.POST.get('buy_mssn') + "
')")
cursor.commit()
```

قسمت فرانتی آن ها:

```
<form method="POST">
    {% csrf_token %}
    <table border="1" style=" background-color:rgb(0, 247, 255)">
        <tr>
            <td>Cashier SSN</td>
            <td><input type="text" placeholder="buy cssn.." name="buy_cssn"></td>
        </tr>
        <tr>
            <td>Book Code</td>
            <td><input type="text" placeholder="buy bcode.." name="buy_bcode"></td>
        </tr>
        <tr>
            <td>Borrower SSN</td>
            <td><input type="text" placeholder="buy mssn.." name="buy_mssn"></td>
        </tr>
    </table>
```

برای امانت گذاری کتابها:

```
cursor = conn.cursor()
cursor.execute("insert into [librarymg.Borrow] values('" +
request.POST.get('borrow_cssn') + " , ' " + request.POST.get('borrow_bcode') + " , '
" + request.POST.get('borrow_mssn') + " , ' " + request.POST.get('borrow_bexdate') + "
' ")")
cursor.commit()
```

قسمت فرانتی آن ها:

```
<form method="POST">
    {% csrf_token %}
    <table border="1" style=" background-color:rgb(0, 247, 255)">
        <tr>
            <td>Cashier SSN</td>
            <td><input type="text" placeholder="borrow cssn.." name="borrow_cssn"></td>
        </tr>
        <tr>
            <td>Book Code</td>
            <td><input type="text" placeholder="borrow bcode.." name="borrow_bcode"></td>
        </tr>
        <tr>
            <td>Borrower SSN</td>
            <td><input type="text" placeholder="borrow mssn.." name="borrow_mssn"></td>
        </tr>
        <tr>
            <td>Expire Date</td>
            <td><input type="text" placeholder="borrow expdate.." name="borrow_bexdate"></td>
        </tr>
    </table>
```

3. حذف کردن اطلاعات:

پس از رویت و اضافه کردن اطلاعات حال به قسمت دیگری میپردازیم که به آن حذف اطلاعات گفته میشود.

برای حذف کردن کافیه که ما از کلید اصلی هر جدول استفاده کنیم اینگونه به راحتی یک ردیف از اطلاعات را در هر جدولی که بخوایم از بین ببریم. در ابتدا با یک جدول کامل توضیحات را میگوییم و سپس قسمت های متفاوت را به نمایش میگذاریم

در ابتدا سعی میکنیم به دیتا بیس متصل شویم پس مثل همیشه:

```
conn=pyodbc.connect('Driver={ODBC Driver 17 for SQL Server};'  
                    'Server=DESKTOP-IVD447A; '  
                    'Database=librarymg; '  
                    'Trusted_Connection=yes;')
```

سپس به قسمت فرانتی کد میرویم و در آنجا اطلاعات مربوط به حذف را دریافت میکنیم:

```
<form method="POST">  
    {% csrf_token %}  
    <table border="1" style=" background-color:rgb(0, 247, 255)">  
        <tr>  
            <td>SSN</td>  
            <td><input type="text" placeholder="member ssn.." name="member_ssn"></td>  
        </tr>  
    </table>  
    <hr>  
    <input type="submit" value="Insert" onclick="return alert('Record is saved successfully')"/>  
</form>
```

سپس دوباره به قسمتی که دیتابیس را وصل کرده برمیگردیم و کوئری مورد نظر را وارد میکنیم:

```
if request.method=="POST":  
    if request.POST.get('member_ssn'):  
        cursor = conn.cursor()  
        cursor.execute("delete from [librarymg.Member] where Ssn=" +  
request.POST.get('member_ssn'))  
        cursor.commit()
```

سپس این دو قسمت را با کد پایین به هم وصل میکنیم:

```
return render( request, 'member\delete.html')
```

حال به قسمت اختصاصی هر یک از جداول میرسیم:

برای صندوقدار:

```
if request.method=="POST":
    if request.POST.get('cashier_cssn'):
        cursor = conn.cursor()
        cursor.execute("delete from [librarymg.Cashier] where cssn=" +
request.POST.get('cashier_cssn'))
        cursor.commit()
```

قسمت فرانتی آن:

```
<form method="POST">
    {% csrf_token %}
    <table border="1" style=" background-color:rgb(0, 247, 255)">
        <tr>
            <td>SSN</td>
            <td><input type="text" placeholder="cashier ssn.."
name="cashier_cssn"></td>
        </tr>
    </table>
    <hr>
    <input type="submit" value="Insert" onclick="return alert('Record is
saved successfully')"/>
</form>
```

برای بخشدار:

```
if request.method=="POST":
    if request.POST.get('shguy_cssn'):
        cursor = conn.cursor()
        cursor.execute("delete from [librarymg.shelfguy] where Shssn=" +
request.POST.get('shguy_cssn'))
        cursor.commit()
```

قسمت فرانتی آن:

```
<form method="POST">
    {% csrf_token %}
    <table border="1" style=" background-color:rgb(0, 247, 255)">
        <tr>
            <td>SSN</td>
            <td><input type="text" placeholder="shelfguy ssn.."
name="shguy_cssn"></td>
        </tr>
    </table>
    <hr>
    <input type="submit" value="Insert" onclick="return alert('Record is
saved successfully')"/>
</form>
```

برای کتاب ها:

```
if request.method=="POST":
    if request.POST.get('bcode'):
        cursor = conn.cursor()
        cursor.execute("delete from [librarymg.Book] where Bcode=" +
request.POST.get('bcode'))
        cursor.commit()
```

قسمت فرانتی آن:

```
<form method="POST">
    {% csrf_token %}
    <table border="1" style=" background-color:rgb(0, 247, 255)">
        <tr>
            <td>SSN</td>
            <td><input type="text" placeholder="book code.."
name="bcode"></td>
        </tr>
    </table>
    <hr>
    <input type="submit" value="Insert" onclick="return alert('Record is
saved successfully')"/>
</form>
```

حال بعد از همه توضیحاته کدی بهتر است به شکل ظاهری برسیم و نمایی از آن را داشته باشیم:

1.Home page:

در ابتدای بارگزاری صفحه نخست را میبینم که در آن میتوانم به هر قسمتی مراجعه کنیم

WELCOME TO
Library Management System

show all books history	show members	show cashiers	show shelfguys	show bookshelves	show buyed books	show borrowed books
------------------------	--------------	---------------	----------------	------------------	------------------	---------------------

add new book	add new member	add new cashier	add new shelfguy
--------------	----------------	-----------------	------------------

delete a book	delete a member	delete a cashier	delete a shelfguy
---------------	-----------------	------------------	-------------------

buy a book	borrow a book
------------	---------------

2.Showing page:

در کل تمام صفحه هایی که اطلاعات رو به نمایش میگذارند یکی هستند

صفحات نمایش دهنده اطلاعات:

Show all books history-show members-show chashiers-show shelfguys-
show book shelves-show buyed books-show borrowed books

و به طور مثال صفحه پایین یک صفحه نمایش اطلاعات هست:

Ssn	Fname	Lname	Phone Number	Birthday	Age	Register Date	Expire Date	Refrenced Cashier Number
10	Alireza	Emami	21859	March 5, 1999	22	Aug. 15, 2018	Nov. 25, 2023	105
10	Alireza	Emami	858585	March 5, 1999	22	Aug. 15, 2018	Nov. 25, 2023	105
11	Amin	Kamali	25471	May 15, 2000	21	Nov. 25, 2019	Nov. 5, 2024	105
12	Parsa	Darvishi	None	Nov. 15, 2001	20	Aug. 14, 2020	Aug. 17, 2025	104
13	Sajad	Jalali	None	May 19, 2001	20	Aug. 15, 2018	Nov. 25, 2025	104
14	Sina	Aby	None	April 25, 2001	20	Jan. 12, 2020	Jan. 14, 2024	106
15	Ahmad	Jafari	None	July 14, 1999	22	March 26, 2020	March 26, 2024	106

[Go Back Home](#)

3.Add page:

صفحاتی که به اضافه کردن اطلاعات میپردازند در حالت کلی یکی هستند و تنها فرق آنها در پارامتر های اضافه کننده است.

صفحات اضافه کننده اطلاعات عبارتند از:

Add new member-add new book-add new cashier-add new shelfguy

به طور مثال صفحه پایین یک صفحه برای اضافه کردن اطلاعات است:

Enter New Member Data

SSN	member Ssn..
Fname	member Fname..
Lname	member Lname..
Birthday	member Birthday..
Register date	member Date..
Expire date	member Expdate..
Cashier SSN	member Cpsn..
Member Phone	memphone Phonenum..

Insert

Go Back Home

4.delete page:

صفحات پاک کننده اطلاعات با گرفتن کلید اصلی هر ردیف موجب به حذف کامل آن ردیف میشوند.

صفحات پاک کننده اطلاعات عبارتند از:

Delete a book-delete a member-delete a cashier-delet a shelfguy

به طور مثال صفحه پایین یک صفحه برای حذف کردن اطلاعات است:

Remove a Member

SSN

5.borrowing page:

در این صفحه اطلاعات مورد نیاز برای امانت گرفتن کتاب دریافت میشود و تا زمان معین شده کتاب به امانت فرد مورد نظر گذاشته میشود.

حتی میتوان در این قسمت مشاهده کرد که چه مدت از زمان امانت گذاری باقی مانده یا چند رو اضافه تر از امانت گذاری کتاب گذاشته است.

صفحه پایین صفحه امانت گذاری کتاب است:

Enter Borrow Data

Cashier SSN	<input type="text" value="borrow cssn."/>
Book Code	<input type="text" value="borrow bcode."/>
Borrower SSN	<input type="text" value="borrow mssn."/>
Expire Date	<input type="text" value="borrow expdate."/>

6.buying page:

در این صفحه اطلاعات مورد نیاز برای خرید کتاب دریافت میشود

صفحه پایین صفحه امانت گذاری کتاب است:

Enter Buy Data

Cashier SSN	buy cssn..
Book Code	buy bcode..
Borrower SSN	buy mssn..

Insert

Go Back Home