```
1: #include <iostream>
 2: #include <fstream>
 3: #include <conio.h>
4: #include <string>
5: #include <windows.h>
6: #include <iomanip>
7: using namespace std;
8: struct Detail {
        string name;
10:
        string authur;
11:
        double price;
12:
        int quantity;
13: };
14: int length = 1, Index = 0;
15: Detail Books[1000];
16: Detail SoldBooks[1000];
17: int SaleIndex = 0, SaleLength = 1;
18: int BookFound1 = 0;
19: int BookFound = 0;
20: int choice = 0;
21: int Design = 0;
22: ofstream File;
23: ifstream Infile;
24: void SaleBook();
25: void ViewSoldBook();
26: void SaveSoldData();
27: void ReadSoldData();
28: void MainFunction();
29: int main();
30: void SaveData() {
        File.open("TestBook.J", ios::app);
32:
        for (int i = 0; i < Books[Index].name.length();</pre>
33:
34:
             if (Books[Index].name[i] == 32) {
35:
                 Books[Index].name[i] = 95;
36:
37:
38:
        for (int i = 0; i < Books[Index].authur.length();</pre>
39:
             i++) {
40:
             if (Books[Index].authur[i] == 32) {
41:
                 Books[Index].authur[i] = 95;
42:
43:
44:
        for (int i = length - 1; i < length; i++) {</pre>
            File << " " << Books[Index].name << endl</pre>
45:
46:
                 << Books[Index].authur << endl</pre>
47:
                 << Books[Index].price << endl</pre>
48:
                 << Books[Index].quantity << endl</pre>
49:
                 << endl;
50:
51:
        File.close();
52: }
53: void ReadData() {
54:
        Infile.open("TestBook.J");
55:
        while (!Infile.eof()) {
56:
57:
            Infile >> Books[Index].name;
58:
             Infile >> Books[Index].authur;
59:
            Infile >> Books[Index].price;
            Infile >> Books[Index].quantity;
60:
            if (!Infile.eof()) {
61:
```

```
62:
63:
      Index = Index + 1;
64:
      length++;
65:
     }
66:
67:
   Infile.close();
68: }
69: bool IsValid(int Num) {
70:
   return ((Num >= 1) && (Num <= 9));
71: }
72: int SearchBook1() {
73: retry:
74:
   system("cls");
75:
   cout << endl << endl;</pre>
76:
   77:
     78:
     "\n";
79:
   80:
81:
     82:
     "\n";
83:
   84:
85:
     "Book Search
86:
     "\n";
87:
   88:
89:
     90:
     "\n";
91:
   92:
93:
     94:
     "\n";
95:
96:
   cout << endl;</pre>
97:
   cout << "\t\tEnter Book name: ";</pre>
98:
   string BookSearch;
99:
   cin.ignore();
   getline(cin, BookSearch);
100:
101:
   for (int i = 0; i < BookSearch.length(); i++) {</pre>
102:
     if (BookSearch[i] == 32) {
103:
      BookSearch[i] = 95;
104:
     }
105:
   }
106:
107:
   for (int i = 0; i < length; i++) {</pre>
     if (BookSearch == Books[i].name) {
108:
109:
      BookFound = 1;
      BookFound1 = 1;
110:
      int Index = i;
111:
112:
      return Index;
113:
     }
114:
115:
   }
116: }
117: int SearchBook() {
118: retrv:
119:
   system("cls");
120:
   cout << endl << endl;</pre>
   121:
122:
```

```
123:
       "\n";
124:
125:
     126:
       127:
       "\n";
128:
     129:
130:
       "Book Search
131:
       "\n";
132:
     133:
134:
       135:
       "\n";
136:
     137:
138:
       139:
       "\n";
140:
141:
     cout << endl;
142:
     cout << "\t\tEnter Book name: " << endl;</pre>
143:
     string BookSearch;
144:
145:
     getline(cin, BookSearch);
    " || "(int i = 0; i < BookSearch.length(); i++
146:
147:
       if (BookSearch[i] == 32) {
148:
          BookSearch[i] = 95;
149:
150:
151:
     for (int i = 0; i < SaleLength; i++) {</pre>
152:
       if (BookSearch == SoldBooks[i].name) {
153:
154:
          cout << "\t\tBook Found: ";</pre>
          cout << "How Much Custumer Want To Buy: ";
155:
156:
          int tempQ;
157:
          cin >> tempQ;
158:
          if (cin.fail()) {
159:
            cin.ignore();
160:
            cin.clear();
161:
            cout << "Enter A Valid Quantity: ";</pre>
162:
            Sleep(500);
163:
            goto retryS;
164:
165:
          SoldBooks[i].quantity = SoldBooks[i].quantity + tempQ;
166:
          File.open("Salerecord.J");
167:
          for (int i = 0; i < SaleIndex; i++) {</pre>
            File << " " << SoldBooks[i].name << endl
168:
169:
              << SoldBooks[i].authur << endl</pre>
170:
              << SoldBooks[i].price << endl</pre>
171:
              << SoldBooks[i].quantity << endl</pre>
172:
              << endl:
173:
174:
          File.close();
175:
          int Pricee;
176:
          for (int k = 0; k < length; k++) {
177:
            if (SoldBooks[i].name == Books[k].name) {
178:
              Books[k].quantity = Books[k].quantity - tempQ;
              cout << "The Total Bill Is: " << Books[k].price * tempQ << endl;</pre>
179:
180:
              break;
181:
            }
182:
          }
183:
```

```
184:
        File.open("TestBook.J");
185:
        for (int i = 0; i < Index; i++) {
186:
          File << " " << Books[i].name << endl</pre>
187:
            << Books[i].authur << endl</pre>
188:
            << Books[i].price << endl</pre>
189:
            << Books[i].quantity << endl</pre>
190:
            << endl:
191:
192:
        File.close();
193:
        system("pause");
194:
        system("cls");
195:
        MainFunction();
196:
197:
198:
    for (int i = 0; i < length; i++) {</pre>
199:
200:
      if (BookSearch == Books[i].name) {
201:
        BookFound = 1;
202:
        BookFound1 = 1;
203:
        int Index = i;
204:
        return Index;
205:
      }
206:
207: }
208: int AuthurBookCount(string Authur) {
209:
    int AuthurCount = 0;
210:
    for (int i = 0; i < length; i++) {</pre>
211:
      if (Authur == Books[i].authur) {
212:
        AuthurCount++;
213:
214:
215:
    return AuthurCount;
216: }
217: void AuthurShowBooks(string Authur) {
    system("cls");
218:
219:
    cout << endl << endl;</pre>
220:
    221:
      222:
      "\n";
223:
    224:
225:
      226:
      "\n";
227:
    228:
229:
      "Book Search
230:
      "\n";
231:
    232:
233:
      234:
      "\n";
235:
236:
    237:
      238:
      "\n";
239:
240:
    cout << endl:</pre>
241:
    cout << "\t\tBooks By The Same Authur" << endl;</pre>
242:
    for (int i = 0; i < length; i++) {</pre>
243:
      if (Authur == Books[i].authur) {
        cout << "************
244:
```

```
245:
          << endl;
246:
        cout << "Book Name: " << Books[i].name << " || "</pre>
247:
          << " Authur Name: " << Books[i].authur << " | | "</pre>
          << " Book Price: " << Books[i].price << " || "</pre>
248:
          << " Quantity: " << Books[i].quantity << endl;</pre>
249:
        250:
251:
          << endl;
252:
      }
253:
254: }
255: void SearchAuthur() {
256:
    system("cls");
257:
    cout << endl << endl;</pre>
258:
    259:
      260:
261:
    262:
263:
      264:
265:
    266:
      "Book Search
267:
      268:
269:
    270:
271:
      272:
      273:
    274:
275:
      276:
277:
278:
    cout << endl;</pre>
279:
    cout << "\t\tEnter Auther's name: ";</pre>
280:
    int Authurcheck = 0;
281:
    string AuthurSearch;
282:
283:
    cin.ignore();
284:
    getline(cin, AuthurSearch);
285:
    for (int i = 0; i < AuthurSearch.length(); i++) {</pre>
286:
      if (AuthurSearch[i] == 32) {
287:
        AuthurSearch[i] = 95;
288:
289:
290:
    for (int i = 0; i < length; i++) {</pre>
291:
      if (AuthurSearch == Books[i].authur) {
        292:
293:
        cout << "Book Name: " << Books[i].name << " || "</pre>
294:
          << " Authur Name: " << Books[i].authur << " | | "</pre>
295:
          << " Book Price: " << Books[i].price << " || "</pre>
296:
297:
   " || "
          << " Quantity: " <<</pre>
        298:
299:
          << endl;
300:
        Authurcheck++;
301:
302:
303:
    if (Authurcheck != 0) {
      cout << Authurcheck <<
304:
        " Books Found By the Authur's Name: " << endl;
305:
```

```
306:
    }
307:
308:
    if (Authurcheck == 0) {
309:
      cout << "No Book Found Under The Given Authur name: "
310:
311:
312:
    system("pause");
313: }
314: void MainFunction(){
315:
    while (choice != 8) {
316:
    retry:
317:
      system("cls");
318:
      cout << endl << endl;</pre>
319:
      320:
        321:
        "\xdb\xdb\xdb\n";
322:
323:
      324:
        325:
        "\xdb\xdb\xdb\n";
326:
327:
      328:
        "Book Sales management "
        329:
330:
331:
      332:
        333:
        "\xdb\xdb\xdb\n";
334:
335:
      336:
        337:
        "\xdb\xdb\xdb\n";
338:
339:
      cout << endl;
340:
      cout << "\t\t1. Add Book" << endl;</pre>
341:
      cout << "\t\t2. Delete Book" << endl;</pre>
      cout << "\t\t3. Update Book" << endl;</pre>
342:
      cout << "\t\t4. View All books" << endl;</pre>
343:
      cout << "\t\t5. Search Book" << endl;</pre>
344:
345:
      cout << "\t\t6. Sell a Book" << endl;</pre>
      cout << "\t\t7. View Sold Book" << endl;</pre>
346:
347:
348:
        cout << "\t\t8. Exit Program" << endl;</pre>
349:
      cin >> choice;
350:
      if (cin.fail()) {
351:
        cin.clear();
352:
        cin.ignore();
353:
        goto retry;
354:
355:
      if (!IsValid(choice)) {
356:
        goto retry;
357:
358:
      cin.ignore();
359:
      if (choice > 0 && choice <= 8) {
360:
361:
        if (choice == 1) {
362:
          int AuthurTaker = 1, lengthIn = 0;
363:
364:
          system("cls");
365:
          cout << endl << endl;</pre>
          366:
```

```
367:
          368:
369:
          "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
370:
        371:
          372:
          \xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
373:
374:
        375:
          "\xb2
              Book Addition
          376:
          "\xdb\xdb\xdb\n";
377:
378:
        379:
          380:
          \xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
381:
382:
        383:
          384:
          "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
385:
386:
        cout << endl;</pre>
387:
        for (int i = length - 1; i < length; i++) {</pre>
388:
389:
          cout << "\t\tbook Name:</pre>
390:
          getline(cin, Books[Index].name);
391:
          for (int i = 0; i < Books[Index].name.length();</pre>
392:
           i++) {
393:
           if (Books[Index].name[i] == 32) {
394:
             Books[Index].name[i] = 95;
395:
396:
397:
          for (int i = Index - 1; i >= 0; i--) {
398:
           if (Books[Index].name == Books[i].name) {
399:
             AuthurTaker = 0;
400:
             lengthIn = 0;
401:
             cout << "Book Already Exist: " << endl;</pre>
402:
             cout << "Do You Want to Add Book Quantity: y/n"
403:
               << endl;
404:
             char Temp Choice;
405:
             cin >> Temp Choice;
406:
             if (Temp Choice == 'y') {
407:
               system("cls");
408:
               Design = 1;
409:
               if (Design) {
410:
                cout << endl << endl;</pre>
411:
                412:
                  413:
                  414:
                  415:
416:
                417:
                  418:
                  419:
                  \xb2\xb2\xb2\xb2\xb2\xbb\xdb\xdb\n";
420:
421:
                422:
                  "\xb2\xb2\xb2\xb2\xb2\xb2
                              Book Addition
                  423:
424:
                  \xb2\xb2\xb2\xb2\xdb\xdb\n";
425:
                426:
                  427:
```

```
428:
                   \xb2\xb2\xb2\xb2\xb2\xb2\xbb\xdb\xdb\n";
429:
430:
                 431:
                   432:
                   433:
                   434:
435:
                 cout << endl;
436:
                 Design = 0;
437:
438:
              RetryTemp Quantity:
439:
                if (Design) {
440:
                 cout << endl << endl;</pre>
441:
                 442:
                   443:
                   444:
                   445:
446:
                 447:
                   448:
                   449:
                   \xb2\xb2\xb2\xb2\xb2\xb2\xbb\xdb\xdb\n";
450:
451:
                 452:
                   "\xb2\xb2\xb2\xb2\xb2\xb2
                                Book Addition
                   453:
                   \xb2\xb2\xb2\xb2\xdb\xdb\n";
454:
455:
                 456:
                   457:
458:
                   \xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
459:
460:
                 461:
                   462:
                   463:
                   464:
465:
                 cout << endl;
466:
                 Design = 0;
467:
                int Temp Quantity;
468:
469:
                cout << "\t\tCurrent Quantity of Books is: " <<</pre>
470:
                 Books[i].quantity
471:
                 << endl:
472:
                cout << "\t\tHow Much You Want to Add: ";</pre>
473:
                cin >> Temp Quantity;
474:
                if (cin.fail()) {
475:
                 cin.clear();
476:
                 cin.ignore();
477:
                 cout << "Please Enter A Valid Input this time: "</pre>
478:
                   << endl;
479:
                 Sleep(1000);
480:
                 system("cls");
481:
                 Design = 1;
482:
                 goto RetryTemp Quantity;
483:
484:
                cin.ignore();
485:
                Books[i].quantity = Books[i].quantity +
486:
                 Temp_Quantity;
487:
                cout << "The New Quantity is: " <<
                 Books[i].quantity << endl;</pre>
488:
```

```
" 11 "
489:
490:
491:
                 File.open("TestBook.J");
492:
                 for (int i = 0; i < Index; i++) {</pre>
                   File << " " << Books[i].name << endl</pre>
493:
494:
                     << Books[i].authur << endl</pre>
495:
                     << Books[i].price << endl</pre>
496:
                     << Books[i].quantity << endl</pre>
497:
                     << endl;
498:
499:
                 File.close();
500:
               }
501:
502:
             if (AuthurTaker) {
503:
               lengthIn = 1;
504:
               cout << "\t\tAuthur Name:</pre>
505:
               getline(cin, Books[Index].authur);
506:
               for (int i = 0; i < Books[Index].authur.length();</pre>
507:
508:
                 if (Books[Index].authur[i] == 32) {
509:
                   Books[Index].authur[i] = 95;
510:
511:
               }
512:
             RetryPrice:
513:
               if (Design) {
514:
                 cout << endl << endl;</pre>
515:
                 516:
                   517:
                   518:
                   "\xdb\xdb\xdb\n";
519:
520:
                 521:
                   522:
                   523:
                   "\xdb\xdb\xdb\n";
524:
525:
                 526:
                   "\xb2\xb2\xb2\xb2
                              Book Addition
                   527:
                   \xb2\xb2\xdb\xdb\n";
528:
529:
                 530:
                   531:
                   532:
                   "\xdb\xdb\xdb\n";
533:
534:
                 535:
                   536:
                   537:
                   "\xdb\xdb\xdb\n";
538:
539:
                 cout << endl;</pre>
540:
                 Design = 0;
541:
542:
               cout << "\t\tPrice of the Book:</pre>
                                 ";
543:
               cin >> Books[Index].price;
544:
               if (cin.fail()) {
545:
                 cin.clear();
546:
                 cin.ignore();
547:
                 cout << "Please Enter A Valid Input this time: "
548:
                   << endl;
549:
                 Sleep(1000);
```

```
550:
               Design = 1;
551:
               system("cls");
552:
               goto RetryPrice;
553:
554:
             cin.ignore();
555:
           RetryQuantity:
556:
             if (Design) {
557:
               cout << endl << endl;</pre>
558:
               559:
                 560:
                 561:
                 "\xdb\xdb\xdb\n";
562:
563:
               564:
                 565:
                 566:
                 "\xdb\xdb\xdb\n";
567:
568:
               569:
                 "\xb2\xb2\xb2\xb2
                            Book Addition
                 570:
                 \xb2\xb2\xdb\xdb\n";
571:
572:
               573:
                 574:
                 575:
                 "\xdb\xdb\xdb\n";
576:
577:
               578:
                 579:
                 580:
                 "\xdb\xdb\xdb\n";
581:
582:
               cout << endl;</pre>
583:
               Design = 0;
584:
585:
             cout << "\t\tQuantity of the Book: ";</pre>
586:
             cin >> Books[Index].quantity;
587:
             if (cin.fail()) {
588:
               cin.clear();
589:
               cin.ignore();
590:
               cout << "Please Enter A Valid Input this time: "
591:
                 << endl;
592:
               Sleep(1000);
593:
               system("cls");
594:
               Design = 1;
595:
               goto RetryQuantity;
596:
597:
             cin.ignore();
598:
             SaveData();
599:
             cout << "\t\tThe book Is Added: " << endl;</pre>
600:
             system("pause");
601:
602:
603:
          if (lengthIn) {
604:
           length++;
605:
           Index++;
606:
607:
608:
        else if (choice == 2) {
609:
         string Delete;
610:
          system("cls");
```

```
611:
      cout << endl << endl;</pre>
612:
      613:
       614:
       "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
615:
616:
      617:
       618:
       \xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
619:
620:
      621:
       "\xb2
           Delete Book
       622:
       "\xdb\xdb\xdb\n";
623:
624:
      625:
       626:
       \xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
627:
628:
      629:
       630:
       "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
631:
632:
      cout << endl;</pre>
633:
      cout << "\t\tEnter The Book Name: ";</pre>
634:
      getline(cin, Delete);
635:
      for (int i = 0; i < Delete.length(); i++) {</pre>
636:
       if (Delete[i] == 32) {
637:
         Delete[i] = 95;
638:
639:
640:
      int BookFoundChecker = 0;
641:
      for (int i = 0; i < length; i++) {
642:
       if (Delete == Books[i].name) {
         cout << "Book Found: " << endl;</pre>
643:
644:
       RetryDeleteDecision:
645:
         if (Design) {
646:
          cout << endl << endl;</pre>
647:
          648:
           649:
           650:
           "\xdb\xdb\xdb\n";
651:
652:
          653:
           654:
           655:
           "\xdb\xdb\xdb\n";
656:
657:
          658:
           "\xb2\xb2\xb2\xb2
                  Delete Book
           659:
           \xb2\xb2\xdb\xdb\n";
660:
661:
          662:
           663:
           664:
           "\xdb\xdb\xdb\n";
665:
666:
          667:
           668:
           669:
           "\xdb\xdb\xdb\n";
670:
          cout << endl;
671:
```

```
672:
                  Design = 0;
673:
674:
                cout << "\t\t1. Delete Entire Book From System: "</pre>
675:
676:
                cout << "\t\t2. Reduce The Quanity Of Book: " <<
677:
                  end1:
678:
                int DeleteChoice;
679:
                cin >> DeleteChoice;
680:
                if (cin.fail()) {
681:
                  cin.clear();
682:
                  cin.ignore();
683:
                  cout << "Please Enter A Valid Input: ";</pre>
684:
                  Sleep(250);
685:
                  system("cls");
686:
                  Design = 1;
                  goto RetryDeleteDecision;
687:
688:
689:
                if (DeleteChoice == 1) {
690:
                  system("cls");
691:
                  Design = 1;
692:
                  if (Design) {
693:
                    cout << endl << endl;</pre>
694:
                    695:
                      696:
                      697:
                      "\xdb\xdb\xdb\xdb\xdb\xdb\n";
698:
699:
                    700:
                      701:
                      702:
                      \xb2\xb2\xb2\xdb\xdb\xdb\n";
703:
704:
                    705:
                      "\xb2\xb2\xb2\xb2\xb2
                                     Delete Book
                      706:
707:
                      \xb2\xb2\xb2\xdb\xdb\n";
708:
                    709:
                      710:
                      711:
                      "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
712:
713:
                    714:
                      715:
                      716:
                      "\xdb\xdb\xdb\xdb\xdb\xdb\n";
717:
718:
                    cout << endl;</pre>
719:
                    Design = 0;
720:
721:
                  for (int k = i; k < length; k++) {</pre>
722:
                    if (k + 1 <= length - 1) {</pre>
723:
                      string temp = Books[k].name;
724:
                      Books[k].name = Books[k + 1].name;
725:
                      Books[k + 1].name = temp;
726:
                    if (k + 1 <= length - 1) {</pre>
727:
728:
                      string temp = Books[k].authur;
729:
                      Books[k].authur = Books[k + 1].authur;
730:
                      Books[k + 1].authur = temp;
731:
732:
                    if (k + 1 <= length - 1) {</pre>
```

```
733:
                   float temp = Books[k].price;
734:
                   Books[k].price = Books[k + 1].price;
735:
                   Books[k + 1].price = temp;
736:
                 if (k + 1 <= length - 1) {</pre>
737:
738:
                   int temp = Books[k].quantity;
739:
                   Books[k].quantity = Books[k + 1].quantity;
740:
                   Books[k + 1].quantity = temp;
741:
742:
743:
               length = length - 1;
744:
               Index = Index - 1;
745:
               BookFoundChecker++;
746:
               cout << "Book Deleted Successfully: " << endl;</pre>
747:
748:
               Sleep(500);
749:
750:
             else if (DeleteChoice == 2) {
751:
               system("cls");
752:
               Design = 1;
753:
               if (Design) {
754:
                 cout << endl << endl;</pre>
755:
                 756:
                   757:
                   758:
                   "\xdb\xdb\xdb\xdb\xdb\xdb\n";
759:
760:
                 761:
                   762:
                   763:
                   \xb2\xb2\xb2\xdb\xdb\xdb\n";
764:
765:
                 766:
                   "\xb2\xb2\xb2\xb2\xb2
                                Delete Book
                   767:
                   "\xb2\xb2\xb2\xb2\xdb\xdb\n";
768:
769:
                 770:
                   771:
                   772:
                   "\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
773:
774:
                 775:
                   776:
                   777:
                   "\xdb\xdb\xdb\xdb\xdb\xdb\n";
778:
779:
                 cout << endl;</pre>
780:
                 Design = 0;
781:
782:
               cout << "\t\tThe Current Quantity Of the Book Named: "</pre>
783:
                 << Books[i].name</pre>
784:
                 << " is: " << Books[i].quantity << endl;</pre>
785:
             RetryTempDel:
786:
               if (Design) {
787:
                 cout << endl << endl;</pre>
788:
                 789:
                   790:
                   791:
                   "\xdb\xdb\xdb\xdb\xdb\xdb\n";
792:
793:
```

```
794:
                           795:
796:
                           797:
                           \xb2\xb2\xb2\xdb\xdb\n";
798:
                        799:
                           "\xb2\xb2\xb2\xb2\xb2
                                            Delete Book
                           800:
                           \xb2\xb2\xb2\xdb\xdb\xdb\n";
801:
802:
                        803:
                           804:
                           805:
                           \xb2\xb2\xb2\xdb\xdb\xdb\n";
806:
807:
                        808:
                           809:
                           810:
                           "\xdb\xdb\xdb\xdb\xdb\xdb\n";
811:
812:
                        cout << endl;
813:
                        Design = 0;
814:
815:
                     cout << "\t\tHow Much You Want To Reduce: ";</pre>
816:
                     int TempDelQuantity;
817:
                     cin >> TempDelQuantity;
818:
                     if (cin.fail()) {
819:
                        cin.clear();
820:
                        cin.ignore();
821:
                        cout << "Enter A valid Input: ";</pre>
822:
                        Sleep(250);
823:
                        system("cls");
824:
                        Design = 1;
825:
                        goto RetryTempDel;
826:
827:
                     Books[i].quantity = Books[i].quantity -
828:
                        TempDelQuantity;
829:
                     cout << "Quantity Updated: " << endl;</pre>
                     cout << "The New Quantity is: " <<</pre>
830:
831:
                        Books[i].quantity << endl;</pre>
832:
                     Sleep(300);
833:
                   else {
834:
835:
                     cout << "Please Enter From Given Number: " <<</pre>
836:
837:
                     Sleep(250);
838:
                     system("cls");
839:
                     Design = 1;
840:
                     goto RetryDeleteDecision;
841:
                   }
842:
                }
843:
844:
             if (BookFoundChecker == 0) {
845:
                cout << "Book Not Found: ";</pre>
846:
             File.open("TestBook.J");
847:
848:
             for (int i = 0; i < Index; i++) {</pre>
                File << " " << Books[i].name << endl
849:
850:
                   << Books[i].authur << endl</pre>
851:
                   << Books[i].price << endl</pre>
852:
                   << Books[i].quantity << endl</pre>
853:
                   << endl;
854:
             }
```

```
855:
        File.close();
856:
       }
857:
858:
       else if (choice == 3) {
859:
        system("cls");
860:
        int BookIndex = SearchBook();
861:
        if (BookFound1) {
          cout << "Book Found: " << endl;</pre>
862:
863:
          char LoopChoice;
864:
          do {
865:
          RetryDecision:
866:
            system("cls");
867:
            cout << endl << endl;</pre>
868:
            869:
              870:
              871:
872:
            873:
              874:
              875:
876:
            877:
              "\xb2\xb2\xb2 Book Sales management "
              878:
              "\xb2\xb2\xdb\xdb\n";
879:
880:
            881:
              882:
              \xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xbb\xdb\xdb\n";
883:
884:
            885:
              886:
              887:
888:
            cout << endl;</pre>
889:
            cout << "\t\t1. Update Price: " << endl;</pre>
890:
            cout << "\t\t2. Update Quantity: " << endl;</pre>
891:
            int UpdateDecision;
892:
            cin >> UpdateDecision;
893:
            if (cin.fail()) {
894:
              cin.clear();
895:
              cin.ignore();
896:
              cout << "Please Enter A Valid Input this time: "
897:
               << endl;
898:
              Sleep(1000);
899:
              goto RetryDecision;
900:
901:
            if (UpdateDecision == 1 | UpdateDecision == 2) {
902:
              cin.ignore();
903:
              BookFound = 0;
904:
              switch (UpdateDecision) {
905:
              case 1: {
906:
              RetryTempPrice:
907:
               system("cls");
908:
               cout << endl << endl;</pre>
909:
               910:
                 911:
                 912:
                 913:
914:
               915:
```

```
916:
                  917:
918:
                  \xb2\xb2\xb2\xb2\xb2\xbb\xdb\xdb\n";
919:
                920:
                  "\xb2\xb2\xb2\xb2\xb2\xb2
                              Price Updating
                  921:
                  \xb2\xb2\xb2\xb2\xdb\xdb\n";
922:
923:
                924:
                  925:
                  926:
                 \label{linear_condition} $$ \xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
927:
928:
                929:
                  930:
                  931:
                  932:
933:
                cout << endl;
934:
                cout << "\t\tThe Old Price Is :" <<</pre>
935:
                 Books[BookIndex].price
936:
                  << endl;
937:
                cout << "\t\tEnter New Price: ";</pre>
938:
                double TempPrice;
939:
                cin >> TempPrice;
940:
                if (cin.fail()) {
941:
                 cin.clear();
942:
                 cin.ignore();
943:
                 cout << "Please Enter A Valid Input this time: "</pre>
944:
                   << endl;
945:
                 Sleep(1000);
946:
                 goto RetryTempPrice;
947:
948:
                Books[BookIndex].price = TempPrice;
                cout << "Price Updated:" << endl;</pre>
949:
950:
                cout << "Do You Want To update this Book Further: y/n";
951:
                cin >> LoopChoice;
952:
                break;
953:
954:
              case 2: {
955:
              RetryTempQuantity:
956:
                system("cls");
957:
                cout << endl << endl;</pre>
958:
                959:
                  960:
                  961:
962:
                  963:
                964:
                  965:
                  966:
967:
                 \xb2\xb2\xb2\xb2\xb2\xb2\xbb\xdb\xdb\n";
968:
                969:
                  "\xb2\xb2\xb2\xb2\xb2\xb2
                              Quantity Updating
                  970:
                  \xb2\xb2\xb2\xb2\xdb\xdb\n";
971:
972:
                973:
                  974:
                  975:
                  "\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
976:
```

```
977:
                         978:
                           979:
                           980:
                           981:
982:
                         cout << endl:
983:
                         cout << "\t\t0ld Quantity Of the Book Is :"</pre>
984:
                           << Books[BookIndex].quantity << endl;</pre>
985:
                         cout << "\t\tEnter New Quantity: ";</pre>
986:
                         int TempQuantity;
987:
                         cin >> TempQuantity;
988:
                         if (cin.fail()) {
989:
                           cin.clear();
990:
                           cin.ignore();
991:
                           cout << "Please Enter A Valid Input this time: "</pre>
992:
                              << endl;
993:
                           Sleep(1000);
994:
                           goto RetryTempQuantity;
995:
996:
                         Books[BookIndex].quantity = TempQuantity;
                         cout << "Quantity Updated:" << endl;</pre>
997:
                         cout << "Do You Want To update this Book Further: y/n";
998:
999:
                         cin >> LoopChoice;
1000:
                         break;
1001:
1002:
                      }
1003:
1004:
1005:
                      cout << "Please Input From given values: " <<</pre>
1006:
                         end1;
1007:
                      Sleep(500);
1008:
                      goto RetryDecision;
1009:
1010:
                 } while (LoopChoice == 'y');
1011:
              }
1012:
              else
1013:
                 cout << "Book Not Found";
1014:
              File.open("TestBook.J");
1015:
              for (int i = 0; i < Index; i++) {
                 File << " " << Books[i].name << endl
1016:
                   << Books[i].authur << endl</pre>
1017:
1018:
                   << Books[i].price << endl</pre>
1019:
                   << Books[i].quantity << endl</pre>
                   << endl;
1020:
1021:
1022:
              File.close();
1023:
              system("pause");
1024:
1025:
           else if (choice == 5) {
1026:
           RetrySearch:
1027:
              system("cls");
1028:
              cout << endl << endl;</pre>
1029:
              1030:
                 1031:
                 "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
1032:
1033:
              1034:
                 1035:
1036:
                 \xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
1037:
```

```
"\xb2
1038:
                        Book Search
                 1039:
1040:
                 "\xdb\xdb\xdb\n";
1041:
              1042:
                 1043:
                 \xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
1044:
1045:
              1046:
                 1047:
                 "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
1048:
1049:
              cout << endl;</pre>
1050:
              cout << "\t\t1. Search Book By Name: " << endl;</pre>
1051:
              cout << "\t\t2. Search Book By Authur: " << endl;</pre>
1052:
              int SearchChoice;
              cin >> SearchChoice;
1053:
1054:
              if (cin.fail()) {
1055:
                 cin.clear();
1056:
                 cin.ignore();
1057:
                 cout << "Please Enter A Valid Input this time: "
1058:
                    << endl;
1059:
                 Sleep(1000);
1060:
                 goto RetrySearch;
1061:
1062:
              if (SearchChoice == 1 | SearchChoice == 2) {
1063:
                 switch (SearchChoice) {
1064:
                 case 1: {
1065:
                    int BookIndex = SearchBook1();
1066:
                    if (BookFound != 0) {
1067:
                      cout << "Book Found: " << endl;</pre>
                      1068:
                         "******
1069:
1070:
                      cout << "Book Name: " << Books[BookIndex].name <<</pre>
1071:
1072:
                         "||"
                         << " Authur Name: " << Books[BookIndex].authur <<</pre>
1073:
                         "[["
1074:
                         " Book Price: " << Books[BookIndex].price <<</pre>
1075:
                         "11"
1076:
                         << " Quantity: " << Books[BookIndex].quantity <<</pre>
1077:
1078:
                      1079:
                         "**********
1080:
1081:
1082:
                      if (AuthurBookCount(Books[BookIndex].authur) ==
1083:
1084:
                         cout << "No Other Books Found By This Authur" <<
1085:
                         1086:
                            "********
1087:
1088:
                            << endl;
1089:
                      }
1090:
                      else {
1091:
                         cout << AuthurBookCount(Books[BookIndex].authur)</pre>
1092:
                            << " Books Found By this Authur: Press y to Show All His "</pre>
1093:
                            "Books: "
1094:
                         1095:
                            "*******
1096:
1097:
                            << endl;
1098:
                         char AuthurShow;
```

```
1099:
                   cin >> AuthurShow;
1100:
                   if (AuthurShow == 'y' | AuthurShow == 'Y') {
1101:
                     AuthurShowBooks(Books[BookIndex].authur);
1102:
                   }
1103:
                 }
               }
1104:
1105:
               else
1106:
                 cout << "Book Not Found: ";
1107:
               system("pause");
1108:
               break;
1109:
1110:
             case 2: {
               SearchAuthur();
1111:
1112:
               break;
1113:
1114:
             }
1115:
1116:
           else {
             cout << "Please Enter From Given Numbers: " <<
1117:
               end1;
1118:
1119:
             Sleep(500);
1120:
             goto RetrySearch;
1121:
           }
1122:
         }
1123:
1124:
1125:
         else if (choice == 6) {
1126:
           SaleBook();
1127:
1128:
         else if (choice == 7) {
1129:
           ViewSoldBook();
1130:
1131:
         else if (choice == 8) {
1132:
           return 0;
1133:
         else if (choice == 4) {
1134:
1135:
           system("cls");
           cout << endl << endl;</pre>
1136:
1137:
           1138:
             1139:
             1140:
             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
1141:
           1142:
             1143:
             \xb2\xb2\xb2\xb2\xb2\xdb\xdb\xdb\n";
1144:
1145:
           1146:
             "\xb2
                     Display
             1147:
             "\xdb\xdb\xdb\n";
1148:
1149:
           1150:
             1151:
             \xb2\xb2\xb2\xb2\xb2\xdb\xdb\n";
1152:
           1153:
             1154:
             1155:
             "\xdb\xdb\xdb\xdb\xdb\xdb\xdb\xdb\n";
1156:
1157:
           cout << endl;
           int BooksCount = 0;
1158:
           for (int i = 0; i < length - 1; i++) {</pre>
1159:
```

```
1160:
                    "*****
1161:
1162:
                    << endl;
1163:
                 cout << " Book Name: " << Books[i].name << endl</pre>
                    << " Authur Name: " << Books[i].authur << endl</pre>
1164:
                    << " Book Price: " << Books[i].price << endl</pre>
1165:
                    << " Quantity:</pre>
                                " << Books[i].quantity << endl;
1166:
1167:
                 BooksCount++;
1168:
              1169:
                 "***"
1170:
1171:
              cout << BooksCount << " Books Found" << endl;</pre>
1172:
              1173:
                 "***"
1174:
1175:
                 << endl;
1176:
              system("pause");
1177:
            }
1178:
1179:
         else {
1180:
            cout << "Please Enter From Given Number: " <<</pre>
              end1;
1181:
            Sleep(1000);
1182:
1183:
         }
1184:
      }
1185:
1186:
      system("pause");
1187:
      return 0;
1188: }
1189: int main() {
1190:
      ReadData();
1191:
      ReadSoldData();
1192:
      system("color b2");
1193:
      system("title BOOK SALE MANAGMENT ");
1194:
      cout << endl << endl;</pre>
1195:
      cout << "\n\n\t\t "</pre>
1196:
         1197:
1198:
      cout << "\n\n\t\t "
1199:
         1200:
1201:
         1202:
      cout << endl;</pre>
1203:
      cout << "
                 \t ";
      for (int i = 0; i < 11; i++) {
1204:
1205:
         Sleep(200);
1206:
         cout << "_";
         cout << "_";
1207:
         cout << "_";
1208:
1209:
         Sleep(200);
1210:
1211:
      MainFunction();
1212: }
1213: void SaleBook() {
      int TempIndex = SearchBook();
1214:
1215: retryQuant:
      system("cls");
1216:
      1217:
1218:
1219:
      cout << "Book Name: " << Books[TempIndex].name <<</pre>
1220:
         " 11 "
```

```
<< " Authur Name: " << Books[TempIndex].authur <<</pre>
1221:
              " || "
1222:
1223:
              << " Book Price: " << Books[TempIndex].price <<</pre>
1224:
              " 11 "
1225:
              << " Quantity: " << Books[TempIndex].quantity <<</pre>
1226:
              end1;
          1227:
1228:
1229:
          cout << "\t\tHow Many Books The custmer want to Buy: ";</pre>
1230:
          int TempQuant;
1231:
          cin >> TempQuant;
1232:
          if (cin.fail()) {
1233:
              cin.ignore();
1234:
              cin.clear();
1235:
              cout << "Enter a Valid Quantity";</pre>
1236:
              Sleep(500);
1237:
              goto retryQuant;
1238:
        " | Books[TempIndex].quantity
1239:
1240:
              Books TempIndex quantity =
1241:
                  Books[TempIndex].quantity - TempQuant;
1242:
              SoldBooks[SaleIndex].name = Books[TempIndex].name;
1243:
              SoldBooks[SaleIndex].authur =
1244:
                  Books[TempIndex].authur;
1245:
              SoldBooks[SaleIndex].price =
1246:
                  Books[TempIndex].price;
1247:
              SoldBooks[SaleIndex].quantity = TempQuant;
1248:
              SaveSoldData();
1249:
              SaleIndex++;
1250:
              SaleLength++;
1251:
              cout << "\n\t\tThe Total Bill is: " <<</pre>
1252:
                  (Books[TempIndex].price) * TempQuant;
1253:
              system("Pause");
1254:
              File.open("TestBook.J");
1255:
              for (int i = 0; i < Index; i++) {
                  File << " " << Books[i].name << endl</pre>
1256:
1257:
                      << Books[i].authur << endl</pre>
1258:
                      << Books[i].price << endl</pre>
1259:
                      << Books[i].quantity << endl</pre>
1260:
                      << endl;
1261:
1262:
              File.close();
1263:
1264:
1265:
1266:
          else {
1267:
              cout << "Not enough Books Found: ";
1268:
          }
1269: }
1270: void SaveSoldData() {
1271:
          File.open("Salerecord.J", ios::app);
1272:
          for (int i = 0;
1273:
              i < SoldBooks[SaleIndex].name.length(); i++) {</pre>
1274:
              if (SoldBooks[SaleIndex].name[i] == 32) {
1275:
                  SoldBooks[SaleIndex].name[i] = 95;
1276:
              }
1277:
1278:
          for (int i = 0;
1279:
              i < SoldBooks[SaleIndex].authur.length(); i++) {</pre>
              if (SoldBooks[SaleIndex].authur[i] == 32) {
1280:
1281:
                  SoldBooks[SaleIndex].authur[i] = 95;
```

```
1282:
            }
1283:
1284:
         for (int i = length - 1; i < length; i++) {</pre>
1285:
            File << " " << SoldBooks[SaleIndex].name << endl</pre>
1286:
                << SoldBooks[SaleIndex].authur << endl</pre>
1287:
                << SoldBooks[SaleIndex].price << endl</pre>
1288:
                << SoldBooks[SaleIndex].quantity << endl</pre>
1289:
                << endl;
1290:
1291:
         File.close();
1292: }
1293: void ReadSoldData() {
1294:
         Infile.open("Salerecord.J");
1295:
         for (int i = 0; i < SaleLength; i++) {</pre>
            while (!Infile.eof()) {
1296:
1297:
1298:
                Infile >> SoldBooks[SaleIndex].name;
                Infile >> SoldBooks[SaleIndex].authur;
1299:
                Infile >> SoldBooks[SaleIndex].price;
1300:
1301:
                Infile >> SoldBooks[SaleIndex].quantity;
1302:
                if (!Infile.eof()) {
1303:
                    SaleIndex = SaleIndex + 1;
1304:
1305:
                    SaleLength++;
1306:
                }
1307:
1308:
            Infile.close();
1309:
1310: }
1311: void ViewSoldBook() {
         int max = 0, Indexmax;
1312:
1313:
         for (int i = 0; i < SaleIndex; i++) {</pre>
            cout << "*****************
                                             **************
1314:
                "*****
1315:
1316:
                << endl;
            cout << " Book Name: " << SoldBooks[i].name <<</pre>
1317:
1318:
                endl
                << " Authur Name: " << SoldBooks[i].authur << endl</pre>
1319:
                << " Book Price: " << SoldBooks[i].price << endl</pre>
1320:
                                " << SoldBooks[i].quantity <<
1321:
                << " Quantity:</pre>
1322:
                endl;
1323:
1324:
            if (SoldBooks[i].quantity > max) {
1325:
                max = SoldBooks[i].quantity;
1326:
                Indexmax = i;
1327:
            }
1328:
         1329:
1330:
1331:
1332:
         cout << "\t\tThe Book With Maximum Sale" << endl;</pre>
         1333:
            "*****
1334:
1335:
            << endl;
         1336:
            "*****
1337:
1338:
1339:
         cout << " Book Name: " << SoldBooks[Indexmax].name</pre>
            << "||Authur Name: " << SoldBooks[Indexmax].authur</pre>
1340:
             << "||Book Price: " << SoldBooks[Indexmax].price <<</pre>
1341:
             " ||Quantity: " << SoldBooks[Indexmax].quantity <<
1342:
```