# **TICKET RESERVATION SYSTEM**

# **(finals output)**

**(ctrl+click to skip to specific part)**

[**TICKET RESERVATION SYSTEM** 1](#_Toc62733642)

[**(finals output)** 1](#_Toc62733643)

[**1.** **Menu** 3](#_Toc62733644)

[*a.* *Code:* 3](#_Toc62733645)

[*b.* *Output:* 4](#_Toc62733646)

[**1.1** **Name with seat number** 5](#_Toc62733647)

[1.1.a Code: 5](#_Toc62733648)

[1.1.b Output: 7](#_Toc62733649)

[**1.2 Search seat if available to reserve** 8](#_Toc62733650)

[1.2.a. Code: 8](#_Toc62733651)

[1.2.b. Output: 9](#_Toc62733652)

[*1.2.b.1 Available:* 9](#_Toc62733653)

[*1.2.b.2 Reserved:* 10](#_Toc62733654)

[**1.3 Delete Reservation Record** 11](#_Toc62733655)

[1.3.a Code: 11](#_Toc62733656)

[1.3.b Output: 13](#_Toc62733657)

[*1.3.b.1 If there is no seat to delete:* 13](#_Toc62733658)

[*1.3.b.2 If seat is deleted successfully:* 14](#_Toc62733659)

[**1.4** **Reserve seat individual and by group** 15](#_Toc62733660)

[a. Output: 15](#_Toc62733661)

[**1.5 View Record** 18](#_Toc62733662)

[a. Output: 18](#_Toc62733663)

[**1.6 Exit.** 19](#_Toc62733664)

[Output: 19](#_Toc62733665)

[**2.** **Input Data: Name and Seat Number** 20](#_Toc62733666)

[*Output:* 20](#_Toc62733667)

[**3.** **Calculated data change status if reserve or not** 21](#_Toc62733668)

[*Available:* 21](#_Toc62733669)

[*Reserved:* 22](#_Toc62733670)

[**4.** **Calculated behavior is checking for continuous seats** 23](#_Toc62733671)

[*Table presents seats available:* 23](#_Toc62733672)

[*When seat is not available:* 24](#_Toc62733673)

[**5.** **Data for system should be saved on a flat file e.g. text file or txt files.** 25](#_Toc62733674)

[*Output:* 25](#_Toc62733675)

[**6.** **System can accept multiple reservation.** 26](#_Toc62733676)

[*System is running a for loop based on number of reservations user inputs:* 26](#_Toc62733677)

[**7.** **Design your own Menu.** 27](#_Toc62733678)

[*Start page:* 27](#_Toc62733679)

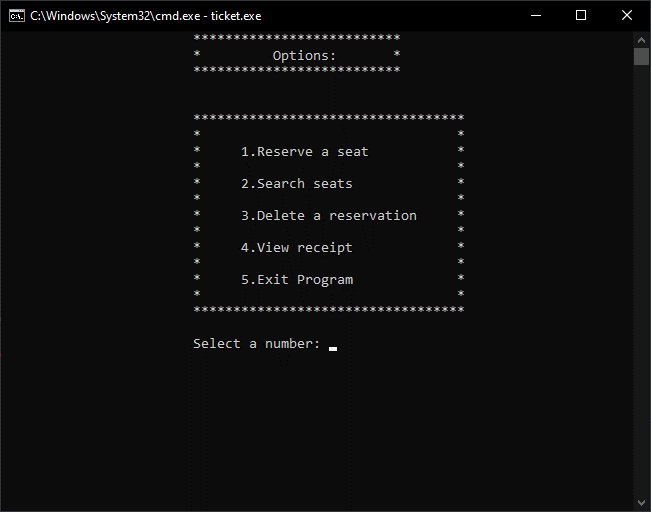
[*Menu:* 28](#_Toc62733680)

# **Menu**

## *Code:*

1. int messagemenu(void)
2. {
3. while (true)
4. {
5. do
6. {
7. system("cls");
8. cout << space() << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
9. cout << space() << "\*         Options:       \*" << endl;
10. cout << space() << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n" << endl;
11. cout << space() << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
12. cout << space() << "\*                                \*" << endl;
13. cout << space() << "\*     1.Reserve a seat           \*" << endl;
14. cout << space() << "\*                                \*" << endl;
15. cout << space() << "\*     2.Search seats             \*" << endl;
16. cout << space() << "\*                                \*" << endl;
17. cout << space() << "\*     3.Delete a reservation     \*" << endl;
18. cout << space() << "\*                                \*" << endl;
19. cout << space() << "\*     4.View receipt             \*" << endl;
20. cout << space() << "\*                                \*" << endl;
21. cout << space() << "\*     5.Exit Program             \*" << endl;
22. cout << space() << "\*                                \*" << endl;
23. cout << space() << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" << endl;
24. cout << space() << "Select a number: ";
25. quitters = getche();
26. cout << space() << endl;
27. } while (!strchr("12345", quitters));
28. return quitters - 49;
29. quitters = 0;
30. }
31. }

## *Output:*



### **Name with seat number**

#### 1.1.a Code:

void reserve(void)

{

    system("cls");

    remove("my\_receipt.txt");

    char rowNo;

    int colNo, inputNumSeats;

*//Inputs number of seats to reserve*

    cout << space() << "Enter number of seats to reserve: ";

    cin >> inputNumSeats;

*//Prints out available seats available to reserve.*

    cout << space() << "Here are the available seats you can reserve: \n";

    cinemascreen();

    cout << "\n\n";

    tickets();

    cout << "\n\n" << space() << "Enter your reservation details: \n\n";

    for (int s = 1; s <= inputNumSeats; s++)

    {

        cout << space() << "\tRow Letter: ";

        cin >> rowNo;

        rowNo = toupper(rowNo);

        cout << space() << "\tColumn Number: ";

        cin >> colNo;

        customerNames();

        cout << "\n\n";

        int transRowNo;

        switch (rowNo){

        case 'A':

            transRowNo = 1;

            break;

        case 'B':

            transRowNo = 3;

            break;

        case 'C':

            transRowNo = 5;

            break;

        case 'D':

            transRowNo = 7;

            break;

        case 'E':

            transRowNo = 9;

            break;

        }

        if (ticketTable[transRowNo + 1][colNo] != " X "){

            ticketTable[transRowNo][colNo] = customerName;

            ticketTable[transRowNo + 1][colNo] = " X ";

        }

        else if (ticketTable[transRowNo + 1][colNo] == " X "){

            char answer;

            cout << space() << "Sorry. This seat is already reserved. Reserve again? (Y/N): ";

            cin >> answer;

            answer = toupper(answer);

            if (answer == 'Y')  reserve();

            else  messagemenu();

            return;

        }

*//to get the time they reserved:*

        time\_t tt;

        struct tm \*ti;

        time(&tt);

        ti = localtime(&tt);

        fstream my\_receipt;

        my\_receipt.open("my\_receipt.txt", ios::out | ios::app);

        my\_receipt << "\tCustomer Name: " << customerName << "\n";

        my\_receipt << "\tReserved Seat Location: " << rowNo << colNo << "\n";

        my\_receipt << "\tTime you reserved: " << asctime(ti);

        my\_receipt.close();

    }

*//printout table*

    system("cls"); space(); cinemascreen();  cout << "\n\n"; tickets();

    cout << "\n\n" << space() << space() << "'X' = reserved" << endl;

    cout << space() << space() << "'A1...' = available \n\n" << endl;

    char answer;

    cout << space() << "Would you like to go back to the main menu (Y/N)?: ";

    cin >> answer;

    answer = toupper(answer);

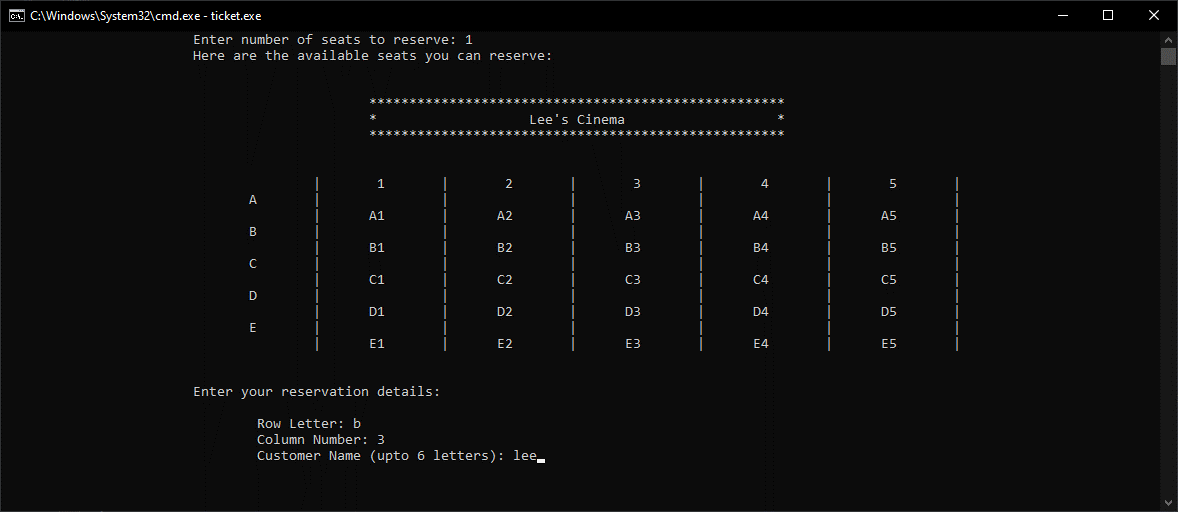
    if (answer == 'Y') messagemenu();

    else reserve();

    return;}

#### 1.1.b Output:

Asks user to input row letter, column number, and customer name.



### **1.2 Search seat if available to reserve**

#### 1.2.a. Code:

void search(void)

{

    cout << space() << "Enter row letter that you want to search:";

    cin >> searchRow;

    searchRow = toupper(searchRow);

    cout << space() << "Enter column number that you want to search: ";

    cin >> searchCol;

    cout << "\n\n";

    int transRowNo;

    switch (searchRow) {

    case 'A':

        transRowNo = 1;

        break;

    case 'B':

        transRowNo = 2;

        break;

    case 'C':

        transRowNo = 3;

        break;

    case 'D':

        transRowNo = 4;

        break;

    case 'E':

        transRowNo = 5;

        break;

    }

    if (ticketTable[transRowNo][searchCol] != "  ")

    {*//printout table*

        space();cinemascreen();cout << "\n\n";tickets();

        cout << "\n" << space() << "This seat is reserved. Would you like to search for more (Y/N)? : ";

        cin >> answer;

        answer = toupper(answer);

        if (answer == 'Y') search();

    }

    else

    {

*//printout table*

        space();cinemascreen();cout << "\n\n";tickets();cout << "\n\n";

        cout << space() << "This seat is available. Would you like to reserve a seat (Y/N)? : ";

        cin >> answer;

        answer = toupper(answer);

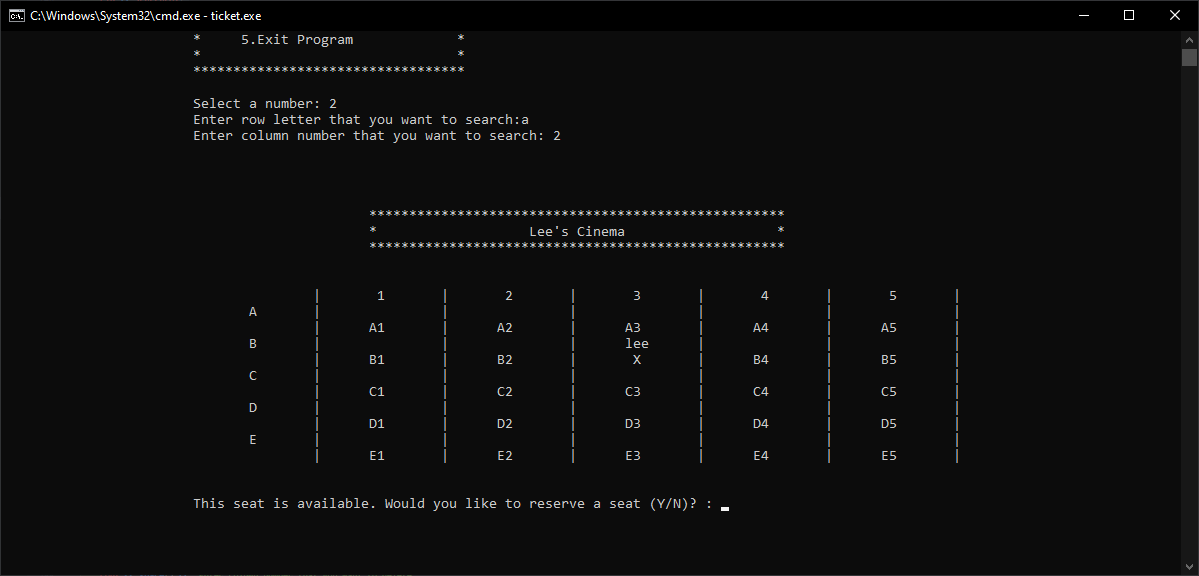
        if (answer == 'Y') reserve();

    }

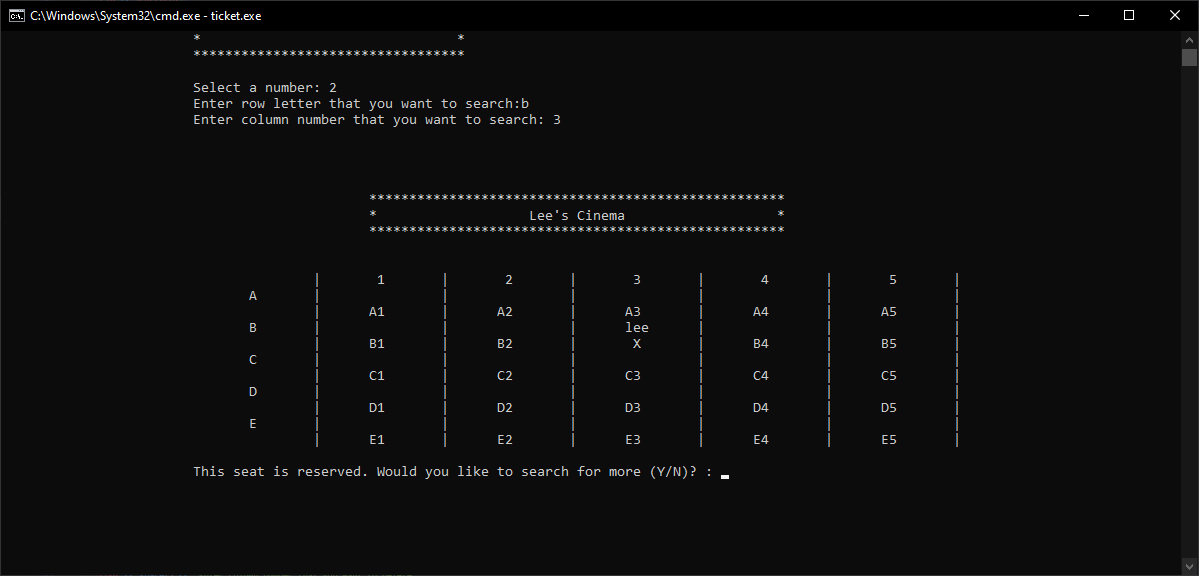
}

#### 1.2.b. Output:

Asks user to input row letter and column number they want to search. It continuously checks if seat is reserved or available

*1.2.b.1 Available:* 

##### *1.2.b.2 Reserved:*



### **1.3 Delete Reservation Record**

#### 1.3.a Code:

void erase(void)

{

    char answer;

    cout << space() << "Enter row letter that you want to delete: ";

    cin >> delRow;

    delRow = toupper(delRow);

    cout << space() << "Enter column number that you want to delete: ";

    cin >> delCol;

    int transRowNo;

    switch (delRow)

    {

    case 'A':

        transRowNo = 1;

        break;

    case 'B':

        transRowNo = 2;

        break;

    case 'C':

        transRowNo = 3;

        break;

    case 'D':

        transRowNo = 4;

        break;

    case 'E':

        transRowNo = 5;

        break;

    }

    string col = to\_string(delCol);

    string delInput = delRow + col;

    if (ticketTable[transRowNo + 1][delCol] == " X ")

    {

        ticketTable[transRowNo][delCol] = "   ";

        ticketTable[transRowNo + 1][delCol] = delInput + " ";

        remove("my\_receipt.txt");

        cout << "\n\n"  << space() << "Deleted reservation succesfully. Would you like to go back to the main menu (Y/N)? : ";

        char answer;

        cin >> answer;

        answer = toupper(answer);

        if (answer == 'Y')  messagemenu();

        else  erase();

    }

    else

    {

        cout << "No seats reserved. Would you like to reserve first? (Y/N)? : ";

        cin >> answer;

        answer = toupper(answer);

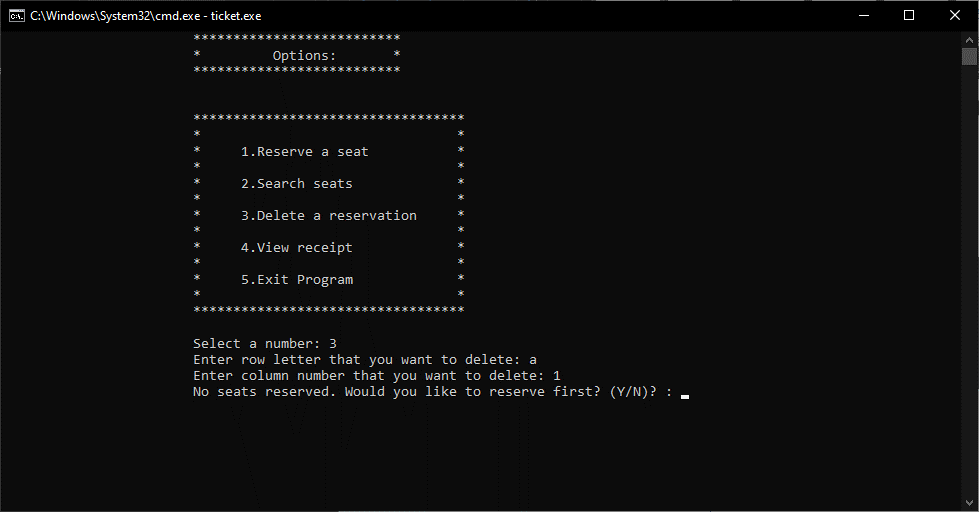
        if (answer == 'Y') reserve();

    }

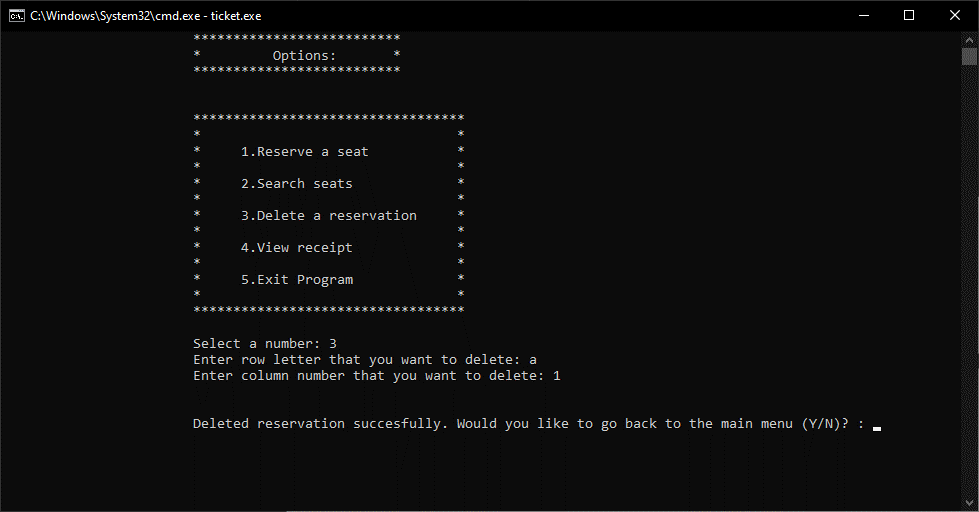
}

#### 1.3.b Output:

##### *1.3.b.1 If there is no seat to delete:*



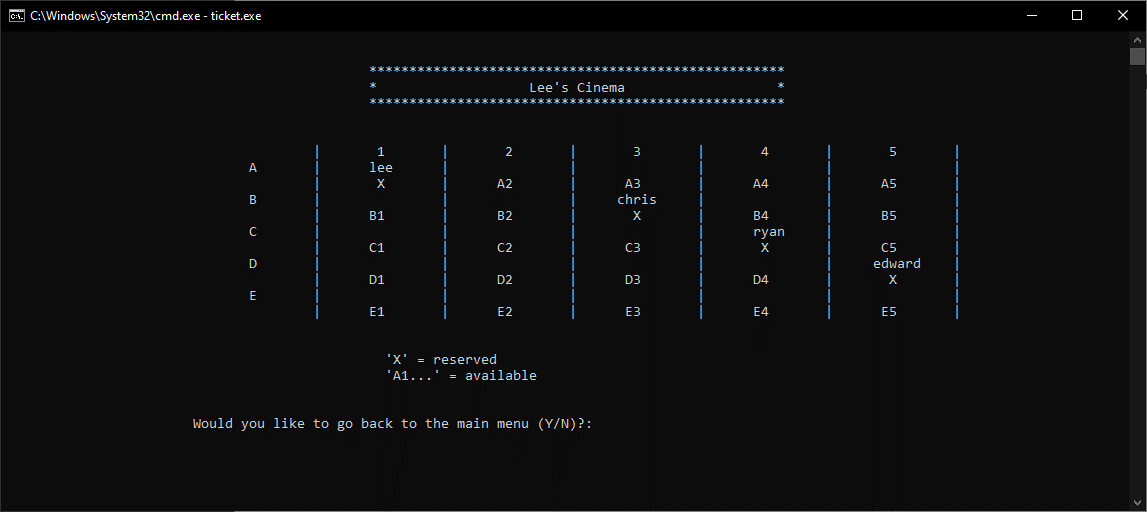
##### *1.3.b.2 If seat is deleted successfully:*



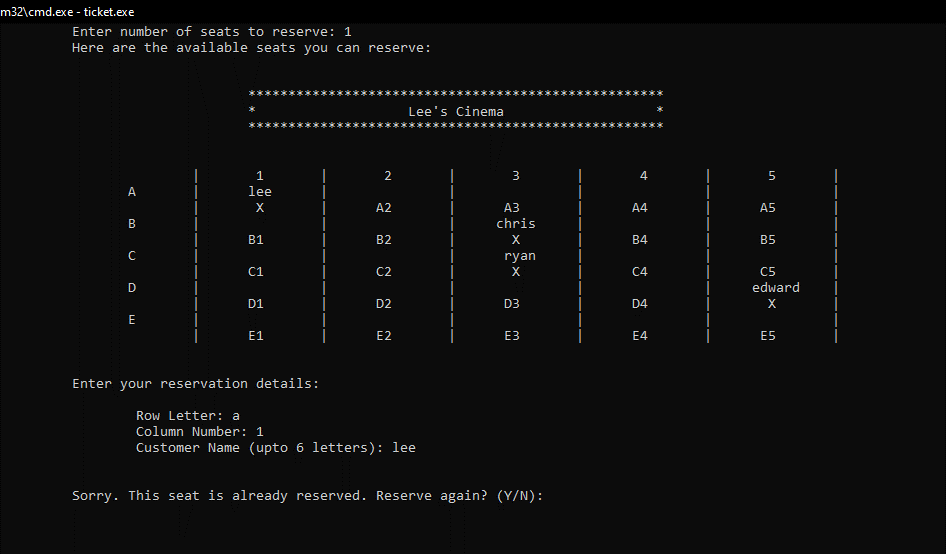
### **Reserve seat individual and by group**

#### Output:

Table:



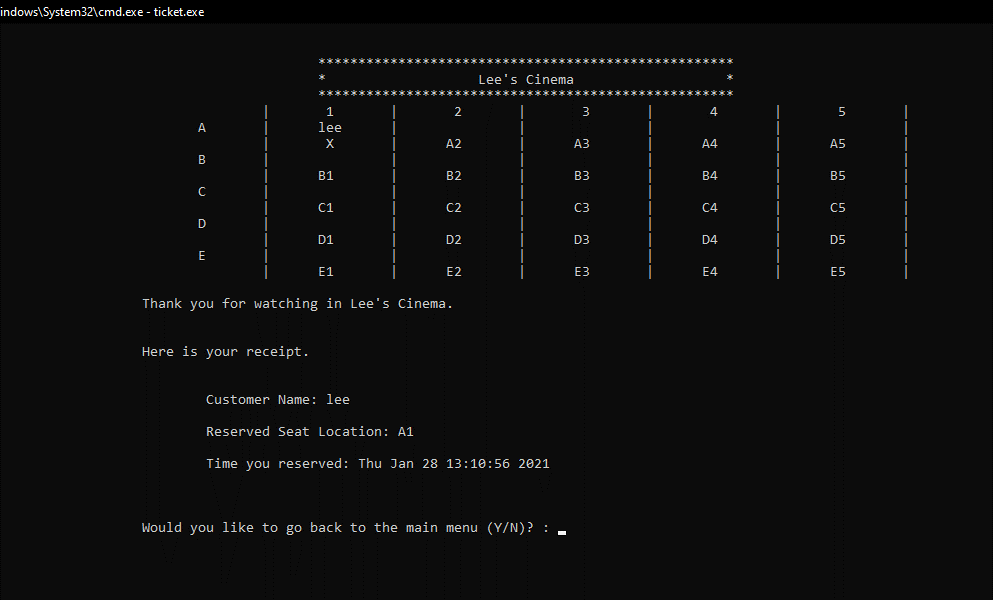
*If seat is already reserved:*



### **1.5 View Record**

#### a. Output:

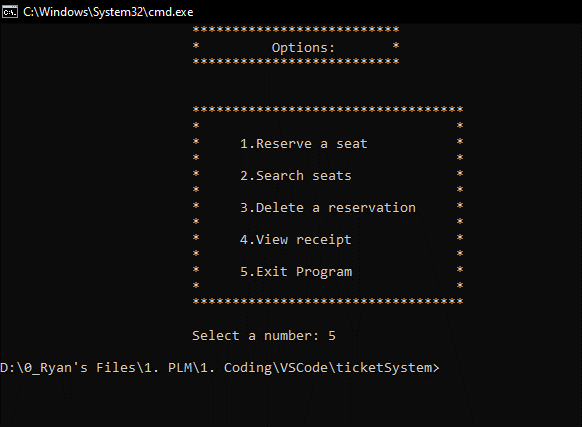
Details such as table to show your seat and a receipt with the time you reserved your seat.



### **1.6 Exit.**

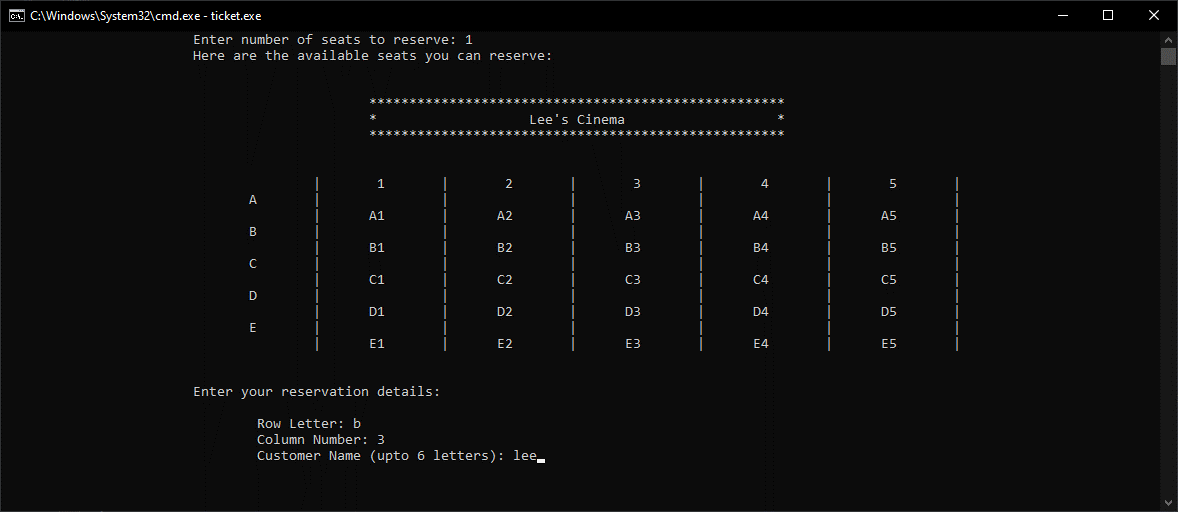
#### Output:

Typing ‘5’ will let you exit the program.



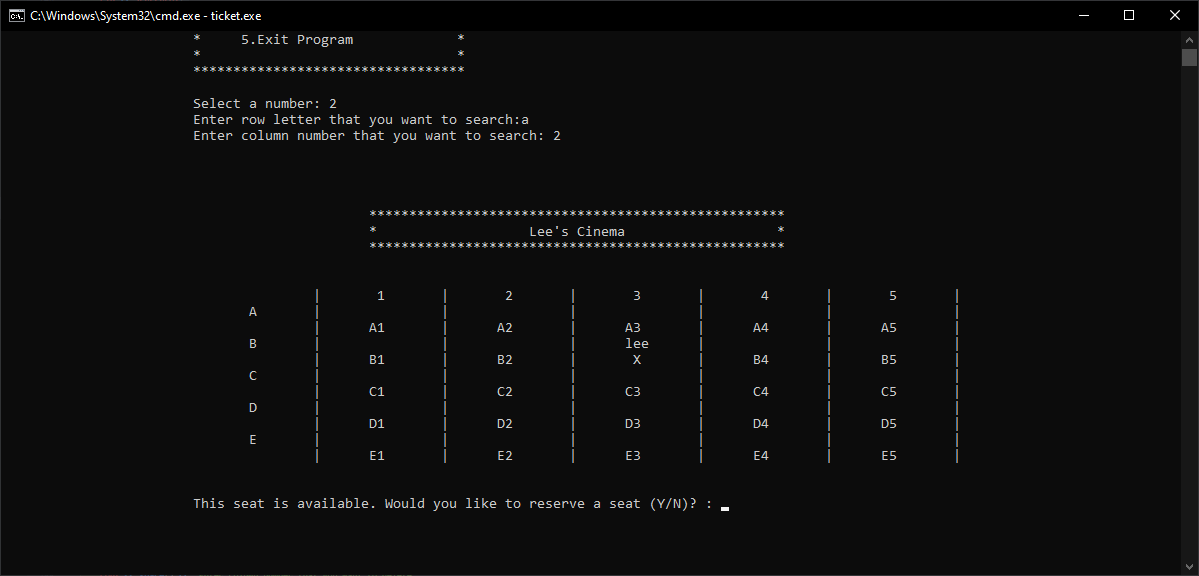
# **Input Data: Name and Seat Number**

## *Output:*

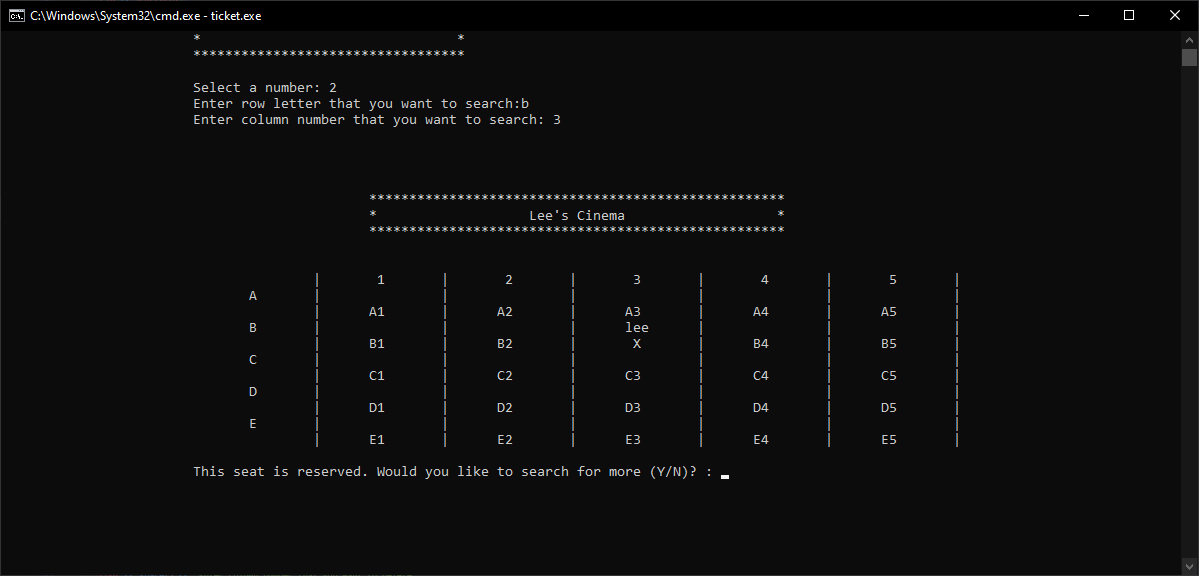


# **Calculated data change status if reserve or not**

## *Available:*

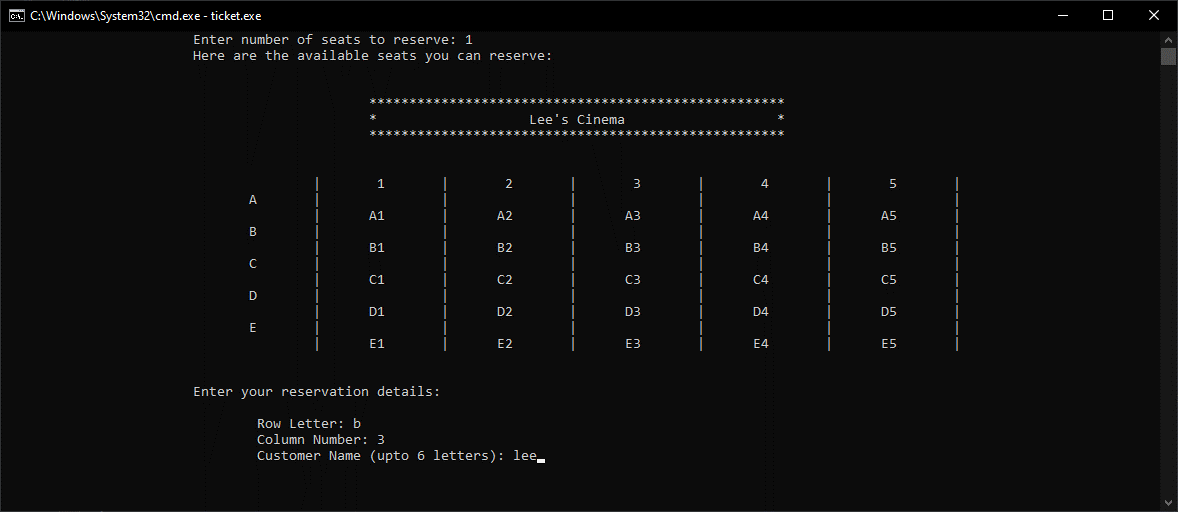


## *Reserved:*

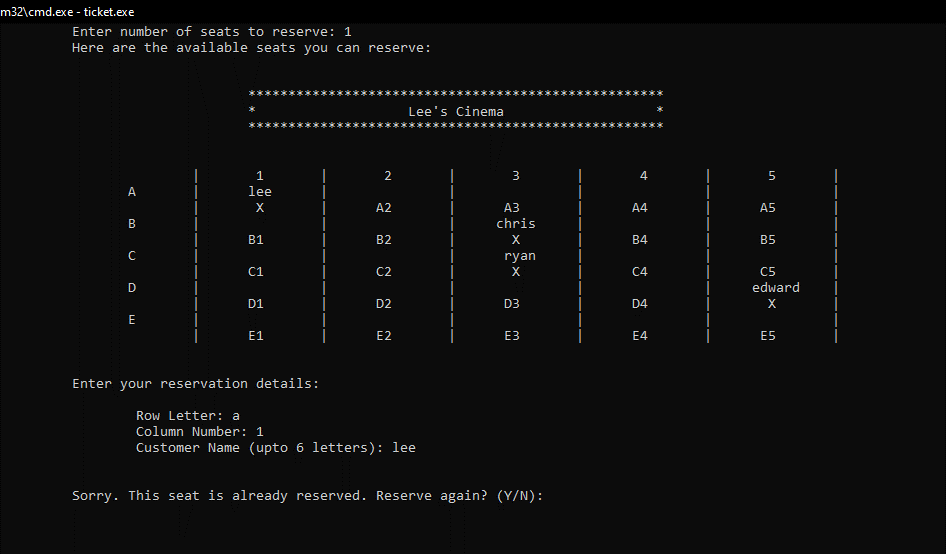


# **Calculated behavior is checking for continuous seats**

## *Table presents seats available:*

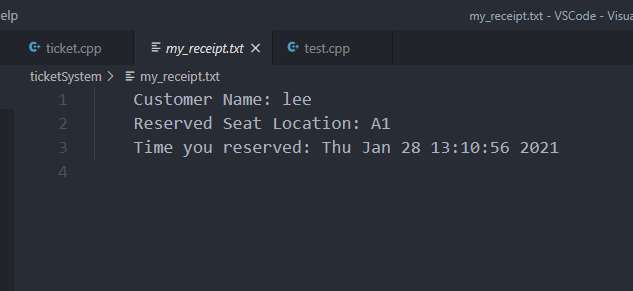


## *When seat is not available:*



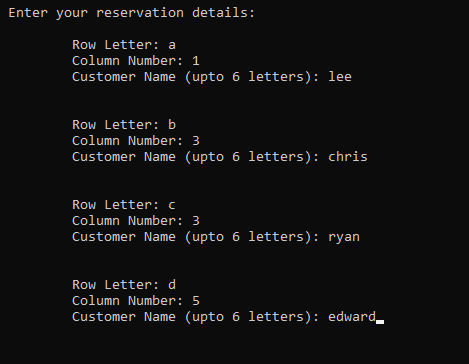
# **Data for system should be saved on a flat file e.g. text file or txt files.**

## *Output:*



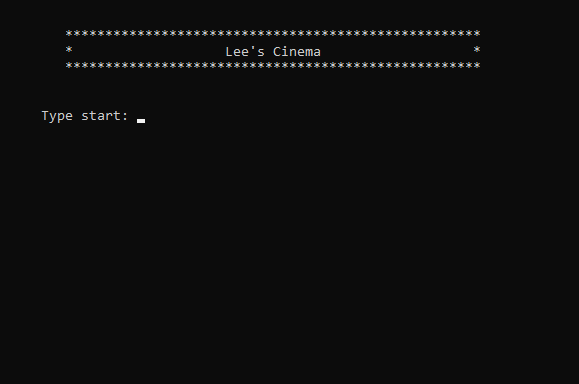
# **System can accept multiple reservation.**

## *System is running a for loop based on number of reservations user inputs:*



# **Design your own Menu.**

## *Start page:*



## *Menu:*

