  
  
**Assignment Cover Sheet**

|  |  |
| --- | --- |
| **Subject Code:** | CSCI212 |
| **Subject Name:** | Interacting systems |
| **Submission Type:** | Program |
| **Assignment Title:** | Shell Program |
| **Student Name:** | Mohammad Fawaz Siddiqi |
| **Student Number:** | 5465163 |
| **Student Phone/Mobile No.** | +971585965669 |
| **Student E-mail:** | ms995@uowmail.edu.au |
| **Lecturer Name:** | Dr Soly |
| **Due Date:** | 21/11/18 |
| **Date Submitted:** | 21/11/18 |

|  |  |
| --- | --- |
| **PLAGIARISM:** The penalty for deliberate plagiarism is FAILURE in the subject. Plagiarism is cheating by using the written ideas or submitted work of someone else. UOWD has a strong policy against plagiarism.  The University of Wollongong in Dubai also endorses a policy of non-discriminatory language practice and presentation.  **PLEASE NOTE:**STUDENTS MUST RETAIN A COPY OF ANY WORK SUBMITTED | **DECLARATION:** I/We certify that this is entirely my/our own work, except where I/we have given fully-documented references to the work of others, and that the material contained in this document has not previously been submitted for assessment in any formal course of study. I/we understand the definition and consequences of plagiarism.  **Signature of Student:** |

|  |  |  |
| --- | --- | --- |
| |  | | --- | | **Optional Marks:** | | **Comments:** | |

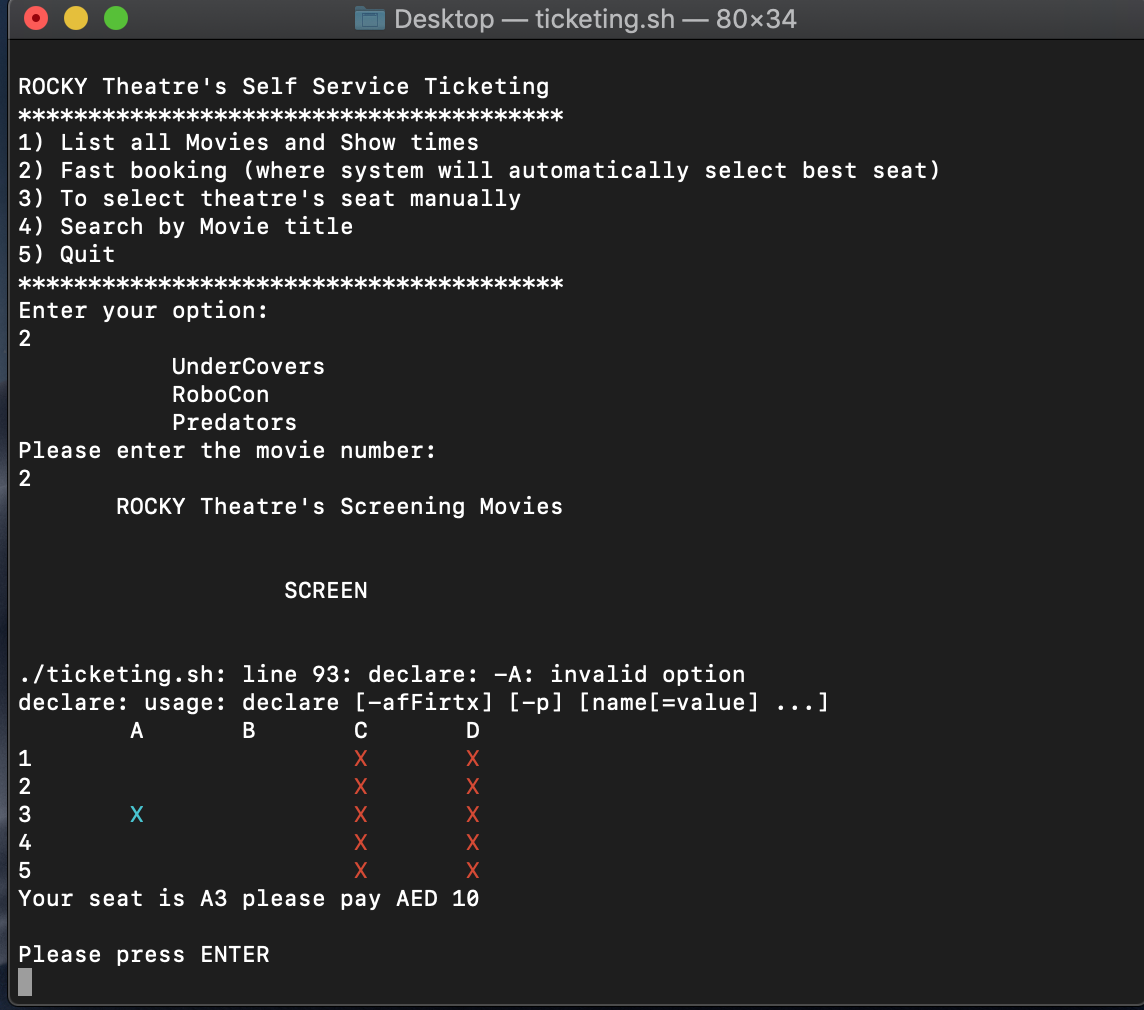
https://my.uowdubai.ac.ae/images/scissors.gif

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Lecturer Assignment Receipt**(To be filled in by student and retained by Lecturer upon return of assignment) | | | **Subject:** | **Assignment Title:** | | **Student Name:** | **Student Number:** | | **Due Date:** | **Date Submitted:** | | **Signature of Student:** | | |

https://my.uowdubai.ac.ae/images/scissors.gif

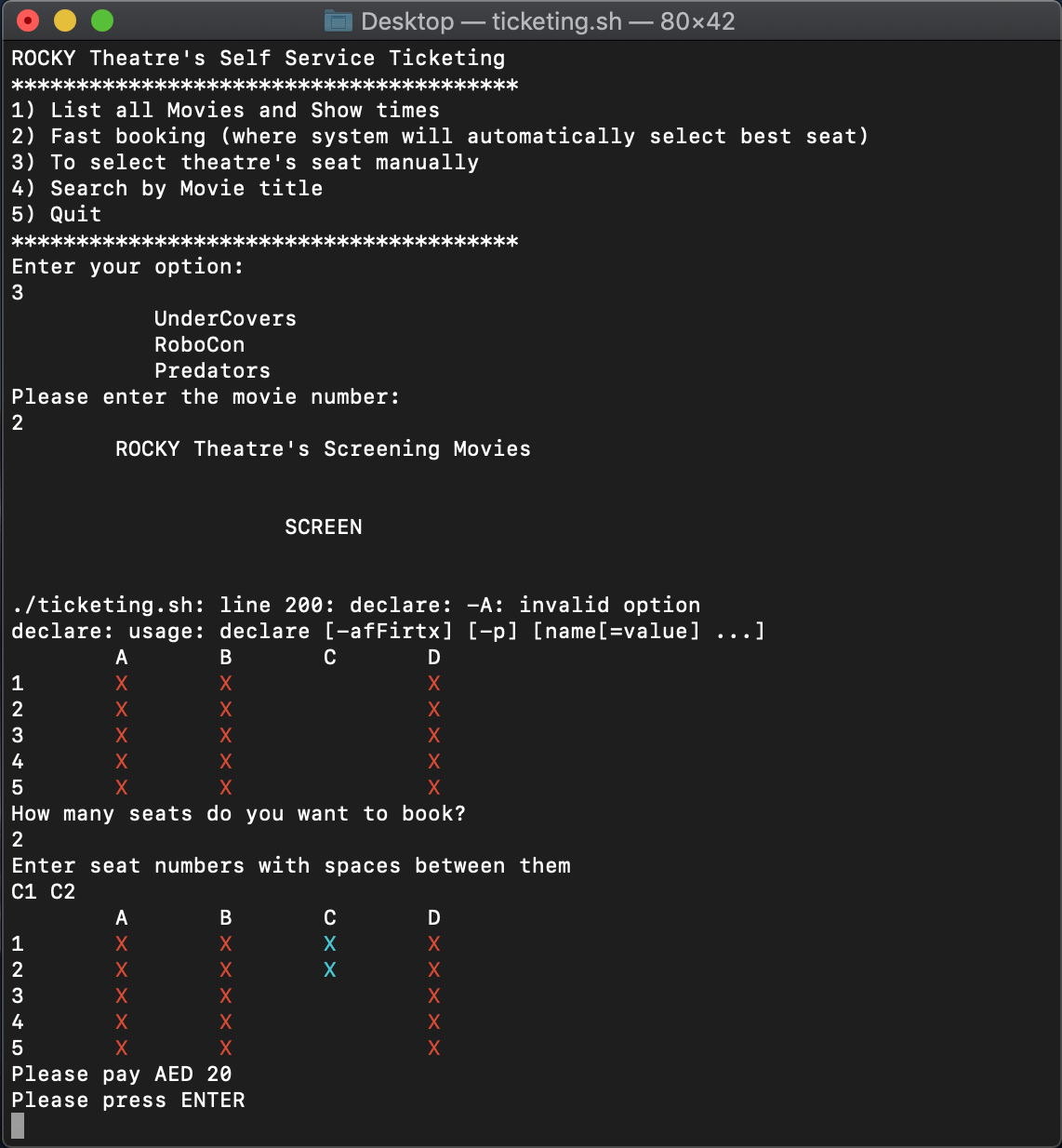
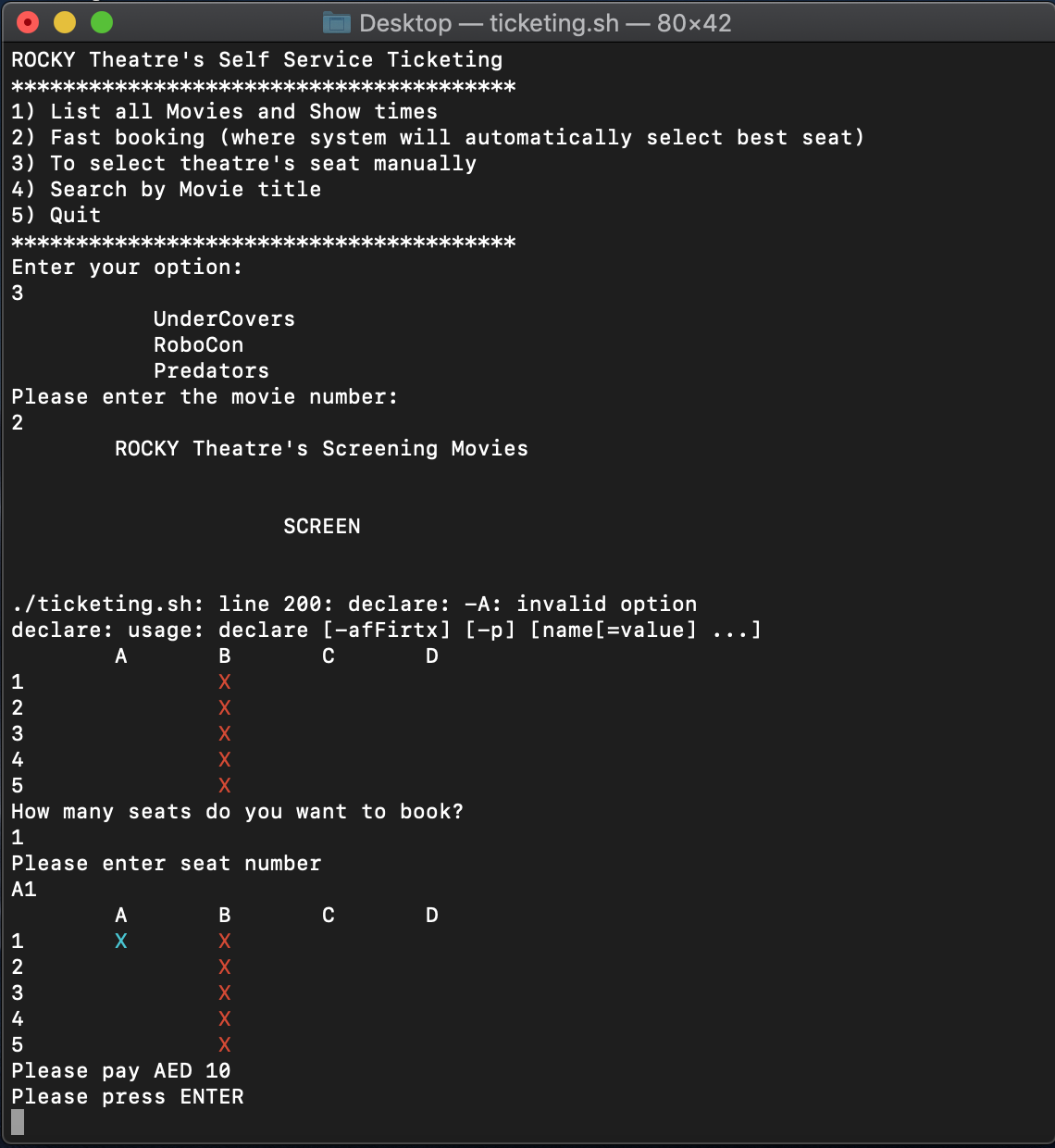
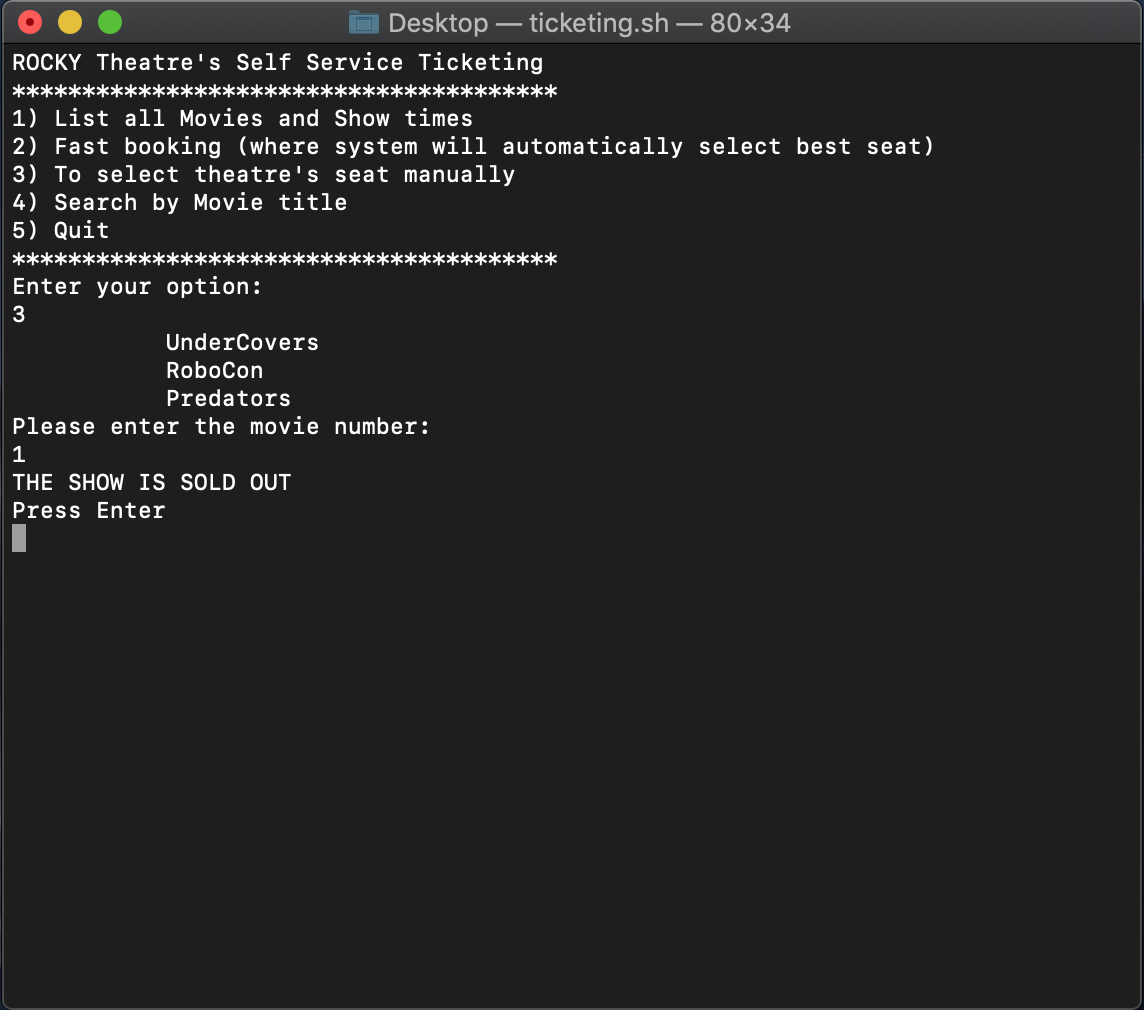
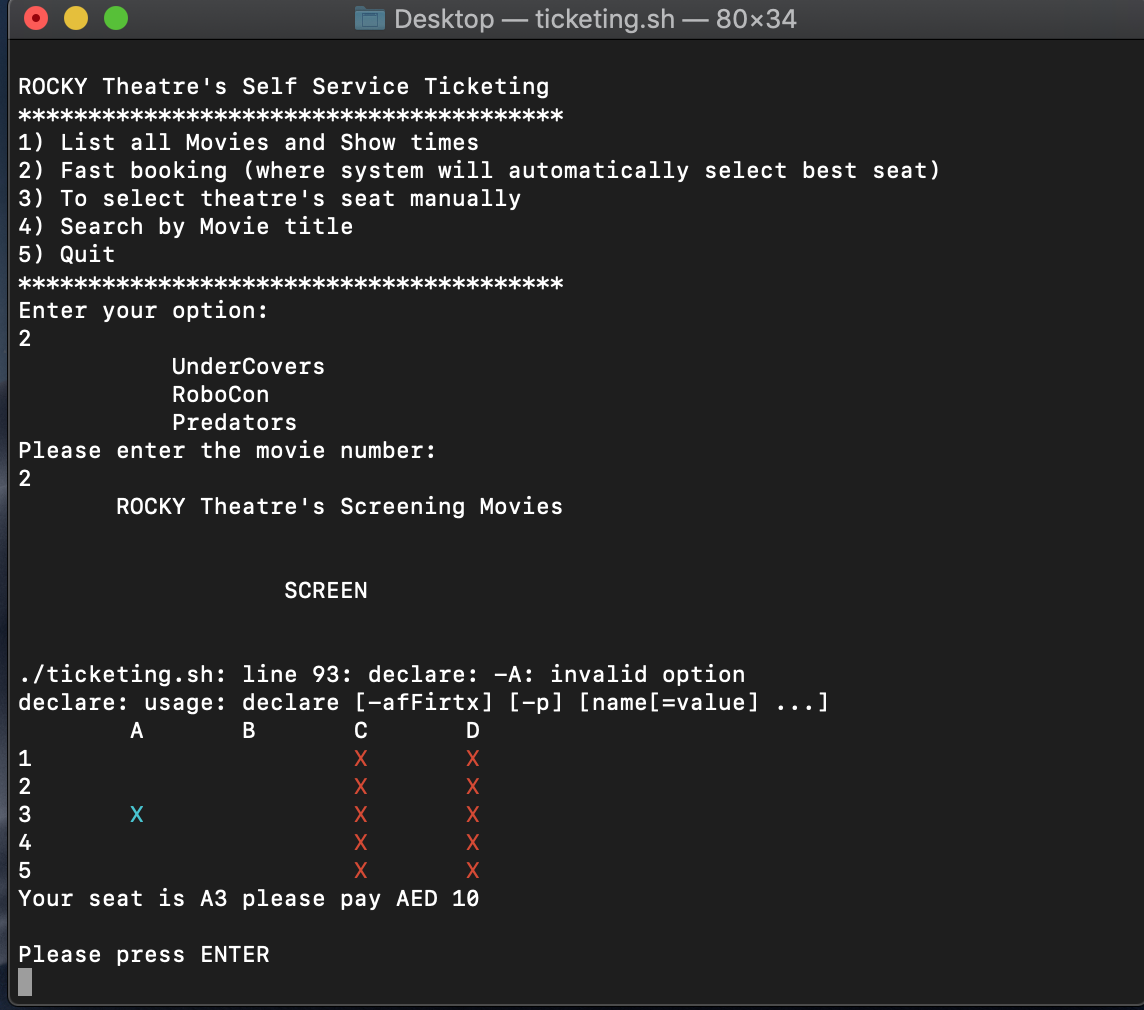
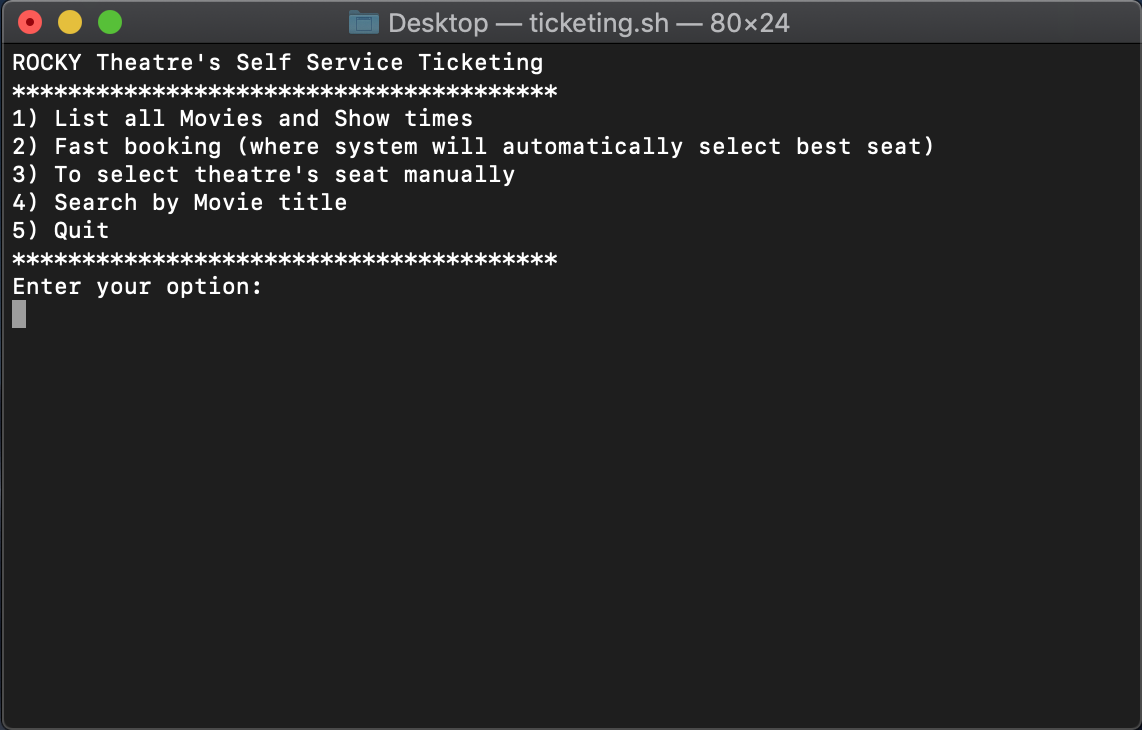
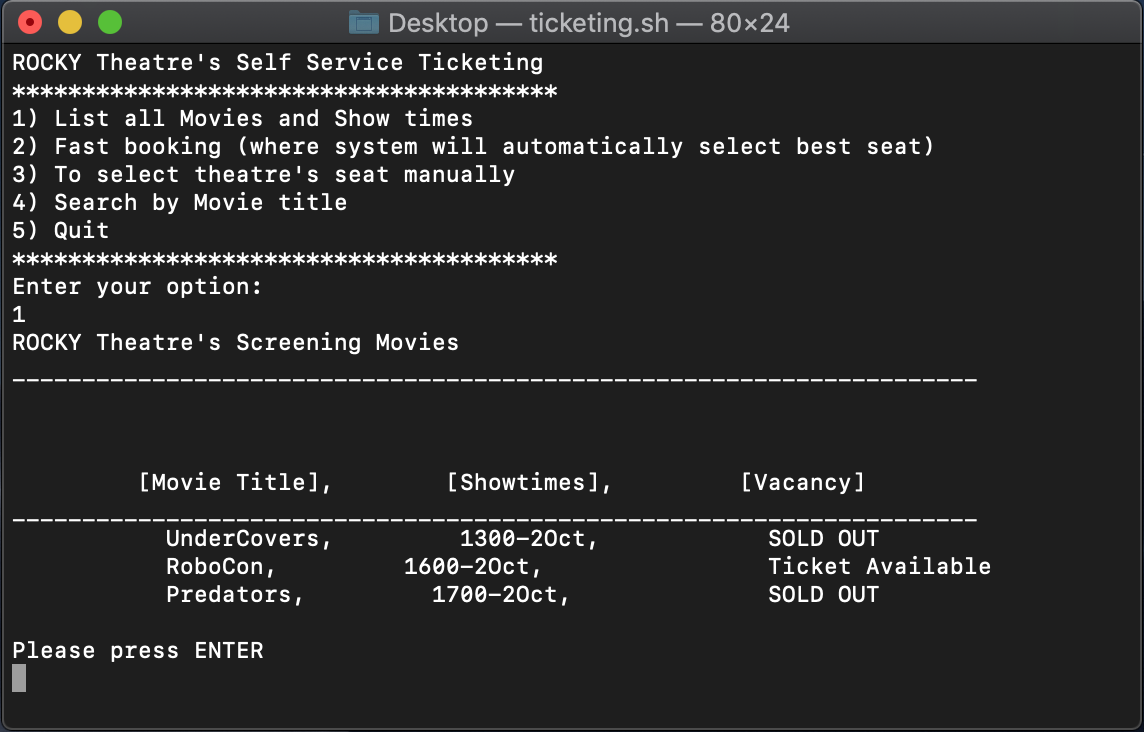
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Student Assignment Receipt** (To be filled in and retained by Student upon submission of assignment) | | | **Subject:** | **Assignment Title:** | | **Student Name:** | **Student Number:** | | **Due Date:** | **Date Submitted:** | | **Signature of Lecturer** | | |

Please follow the instructions on the utility since it is menu based and uses numbers and letters only when booking the seat number and searching for the movie.

There are some errors in the utility for example:

Whereas the program does its functionality.

Screenshots are provided below:



Program code:

#!/bin/bash

#tputcolors

function main\_menu #displays main menu

{

while :

do

clear

echo -e "ROCKY Theatre's Self Service Ticketing"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo -e "1) List all Movies and Show times"

echo -e "2) Fast booking (where system will automatically select best seat)"

echo -e "3) To select theatre's seat manually"

echo -e "4) Search by Movie title"

echo -e "5) Quit"

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"

echo "Enter your option:"

read mainc;

case $mainc in #switch case followed by the input in mainc

1) choice1;

echo "Please press ENTER";

read;;

2) choice2;

echo "Please press ENTER";

read;;

3) choice3;

echo "Please press ENTER";

read;;

4) choice4;

echo "Please press ENTER";

read;;

5) exit 0;;

\*) echo "Invalid selection!! Please Select a valid option";

echo "Please press ENTER";

read;;

esac

done

}

function choice1 #to list movies

{

echo -e "ROCKY Theatre's Screening Movies"

echo "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"

filename="./MoviesList.txt"

while IFS= read -r line

do

sed 's/^\([^,]\*,\)\([^,]\*,\)\([^,]\*\)/\2\1\3/' #stream edidor used tto manipulate file

sed 's/Yes/SOLD OUT/g'

sed 's/No/Ticket Available/g'

printf '%s\n' "$line"

done <"$filename"

}

function choice2 #selects seats in random

{

awk -F ',' 'NR > 6 {print $2}' MoviesList.txt #awk used to print

echo "Please enter the movie number:"

read movienum;

final\_movienum=`expr $movienum + 6`

ifavail=$(sed -n "$final\_movienum"'p' < MoviesList.txt | awk -F ',' '{print $3}') #checks the availability from the file

if [[ $ufavail == \*SOLD\* ]];

then

echo "SHOW IS SOLD OUT"

echo "Press Enter"

read

continue

elif [[ -z $ifavail ]];

then

echo "INVALID SELECTION!! Press Enter to Continue"

read

continue

else

echo -e " ROCKY Theatre's Screening Movies "

echo " "

echo -e " SCREEN "

echo " "

fi

declare -A randarr #3d array declared

rows=5

columns=4

for ((i=0;i<rows;i++))

do

for ((j=0;j<columns;j++))

do

randarr[$i,$j]=$((RANDOM % 2))

done

done

best=("A3" "A5" "D2" "D3" "B5" "C3" "B1" "B4" "A1" "C5" "C4" "C2" "D4" "A4" "B3" "A2" "D1" "B2" "D5" "C1") #random seats declared

char=( {A..Z} )

for i in ${!best[@]}

do

first[$i]=${best[i]:0:1}

num[$i]=${best[i]:1:2}

done

for k in ${!best[@]}

do

for ((i=0;i<rows;i++))

do

for ((j=0;j<columns;j++))

do

if [ ${first[$k]} = ${char[$(($j))]} ] && [ ${num[$k]} -eq $(($i+1)) ] && [ ${randarr[$i,$j]} -eq 0 ] #checks if seats are free then places the seat for the customer and then goes on to the next loop to get number

then

best[$k]="00"

fi

done

done

done

for i in ${!best[@]}

do

if test "${best[$i]}" != "00"

then

bestfinal=${best[$i]}

break

fi

done

for ((i=0;i<=rows;i++))

do

for ((j=0;j<=columns;j++))

do

if test $i -eq 0 -a $j -eq 0

then

printf " "

elif test $i -eq 0 -a $j -ne 0

then

printf "${char[$(($j-1))]} "

elif test $j -eq 0 -a $i -ne 0

then

printf "$i "

else

if test ${randarr[$(($i-1)),$(($j-1))]} -eq 0

then

printf "\e[31mX \e[0m"

elif test ${bestfinal:0:1} = ${char[$(($j-1))]} -a ${bestfinal:1:2} -eq $i

then

printf "\e[36mX \e[0m"

else

printf " "

fi

fi

done

echo

done

echo "Your seat is $bestfinal please pay AED 10"

echo

}

function choice3

{

awk -F ',' 'NR > 6 {print $2}' MoviesList.txt

echo "Please enter the movie number: "

read movienum; #checks customer input and based on that tells if the show is sold out or Tickets are available

final\_movienum=`expr $movienum + 6`

ifavail=$(sed -n "$final\_movienum"'p' < MoviesList.txt | awk -F ',' '{print $3}')

if [[ $ifavail == \*SOLD\* ]];

then

echo "THE SHOW IS SOLD OUT"

echo "Press Enter"

read

continue

elif [[ -z $ifavail ]];

then

echo "INVALID SELECTION!!! Press Enter"

read

continue

else

echo -e " ROCKY Theatre's Screening Movies "

echo " "

echo -e " SCREEN "

echo " "

fi

declare -A randarr2

rows=5

columns=4

for ((i=0;i<rows;i++))

do

for ((j=0;j<columns;j++))

do

randarr2[$i,$j]=$((RANDOM % 2))

done

done

char=( {A..Z} )

for ((i=0;i<=rows;i++))

do

for ((j=0;j<=columns;j++))

do

if test $i -eq 0 -a $j -eq 0

then

printf " "

elif test $i -eq 0 -a $j -ne 0

then

printf "${char[$(($j-1))]} "

elif test $j -eq 0 -a $i -ne 0

then

printf "$i "

else

if test ${randarr2[$(($i-1)),$(($j-1))]} -eq 0

then

printf "\e[31mX \e[0m"

else

printf " "

fi

fi

done

echo

done

echo "How many seats do you want to book?"

read seatamount

if ((seatamount>1))

then

echo "Enter seat numbers with spaces between them"

read -a newseat

for i in ${!newseat[@]}

do

first[$i]=${newseat[i]:0:1}

num[$i]=${newseat[i]:1:2}

done

for ((i=0;i<=rows;i++))

do

for ((j=0;j<=columns;j++))

do

completed=0

if test $i -eq 0 -a $j -eq 0

then

printf " "

elif test $i -eq 0 -a $j -ne 0

then

printf "${char[$(($j-1))]} "

elif test $j -eq 0 -a $i -ne 0

then

printf "$i "

else

for k in ${!newseat[@]}

do

if [ ${first[$k]} = ${char[$(($j-1))]} ] && [ ${num[$k]} -eq $i ] && [ $completed -eq 0 ]

then

printf "\e[36mX \e[0m"

completed=1

fi

done

if (($completed==0))

then

if test ${randarr2[$(($i-1)),$(($j-1))]} -eq 0

then

printf "\e[31mX \e[0m"

else

printf " "

fi

fi

fi

done

echo

done

price=`expr 10 \\* ${#newseat[@]}` #calculates the total price if more than 1 seats books

echo "Please pay AED $price"

else

echo "Please enter seat number"

read newseat

for ((i=0;i<=rows;i++))

do

for ((j=0;j<=columns;j++))

do

completed=0

if test $i -eq 0 -a $j -eq 0

then

printf " "

elif test $i -eq 0 -a $j -ne 0

then

printf "${char[$(($j-1))]} "

elif test $j -eq 0 -a $i -ne 0

then

printf "$i "

else

if test ${newseat:0:1} = ${char[$(($j-1))]} -a ${newseat:1:2} -eq $i

then

printf "\e[36mX \e[0m"

elif test ${randarr2[$(($i-1)),$(($j-1))]} -eq 0

then

printf "\e[31mX \e[0m"

else

printf " "

fi

fi

done

echo

done

echo "Please pay AED 10"

fi

}

function choice4

{

function movie #uses grep based on name to find the name of the movie

{

echo "Enter the name of the movie"

read name

grep -i "$name" MoviesList.txt

if [ -z "$(grep -i "$name" MoviesList.txt)" ]

then

echo "Movie not available!"

fi

}

function timing

{

echo "Enter timing in the format 1300 for 1 PM"

read time

grep -i "$time" MoviesList.txt #uses grep based on time to find the name of the movie

if [ -z "$(grep -i "$time" MoviesList.txt)" ]

then

echo "Timing not available!"

fi

}

echo "1) According to movie name"

echo "2) According to movie time"

read moviemenu;

case $moviemenu in

1) movie;;

2) timing;;

\*) echo "Invalid selection!! Please Select a valid option";

echo "Press ENTER to Continue";

read;;

esac

}

main\_menu #program jumps to main menu

-----------------------------END OF PROGRAM CODE---------------------------------

Text file:

[Showtimes], [Movie Title], [Vacancy]

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1300-2Oct, UnderCovers, SOLD OUT

1600-2Oct, RoboCon, Ticket Available

1700-2Oct, Predators, SOLD OUT

Developed on macOS terminal thus not much color coding is supported.