**PRISON MANAGEMENT SYSTEM**

**OBJECTIVE :**

To make a system that can manage the records of the prisoner.

**INTRODUCTION :**

* This system can be used to manage the record of the prisoners by the authorities.
* It has various functions like to add a record, view the information of the prisoners altogether, edit the record, delete a record and search a record.
* Each submenu has a particular function which are as follows :

1. **addRecord**

It is used to add information of prisoners in the system like name, gender, age, weight, height, crime, lawyer, conviction, the date when punishment started and the date when punishment ended.

1. **searchRecord**

It is used to search information of prisoners in the system and is then displayed on the screen.

1. **editRecord**

It is used to edit information of the prisoner in the system which can be done in any category and also the whole record can be edited at once which is then stored in the file.

1. **viewRecord**

It is used to view the information of all prisoners at once in a tabular form.

1. **deleteRecord**

It is used to delete a particular record of a prisoner from the system.

**METHODOLOGY :**

**Pseudocode :**

**Login Form**

The code is for a program that will display the login menu.

The has to enter the username and password

If the systemUsername and inputUsername is same and the systemPassword and inputPasswod are same then

You will enter the main menu(and 1 is returned to the login function)

If they are not same then

Then you will have to again write the credentials as they were wrong previously

The user will be given three chances for entering the correct credentials as it is the maximum limit

**Main**

When 1 is returned by login then the main menu is opened.

Prison management system is printed along with the different submenus like addRecord, searchRecord, editRecord, viewRecord, deleteRecord and exit and each sunmenu has a number assigned to it.

Then the user is asked to enter their choice and then the particular menu is opened according to the choice.

If addRecord is chosen then the add menu is opened and so on.

If the exit menu is chosen the thank you is displayed on the screen.

If any other choice is entered then it shows that the choice is invalid.

**AddRecord**

The code is for a program that will display an add menu.

Next a pointer named fp is assigned to the file

If it is equal to null than

it doesn't exist

If is not equal to Null than

The file named “PrisonRecord” can be opened

The the system asks to enter the prisoner’s ID

If the system ID and the input ID are equal then

The record already exists

If both are not equal

Then the system asks to enter the information of the prisoner.

After filling information once, the system asks whether you want to add another record or not

If yes then

The above process will be repeated once again

If no then

The file will be closed using fclose.

**SearchRecord**

The code is for a program that will display a search menu.

Next a pointer named fp is assigned to the file.

The file named as “PrisonRecord” will be opened using fopen

Then the user has to enter the prisoner ID which has to be searched.

The “PrisonRecord” file is read by using the function fread and if the input ID is matched with anyID of the system then

The record is displayed

Else

The record does not exist in the system

After searching the information once, the system asks whether the user wants to search another record or not

If yes then

The above process will be repeated once again.

If no then

The file will be closed using fclose.

**EditRecord**

The code is for a program that will display an edit menu.

The the system asks to enter the prisoner’s ID

Next a pointer named fp is assigned to the file named “PrisonRecord”

If it is equal to null than

File doesn't exist

If is not equal to Null than

The file named “PrisonRecord” will be opened

A pointer named ft is assigned to a file named “temp” which is empty

If it is equal to null than

File doesn't exist

If is not equal to Null than

The file named “temp” will be opened

The “PrisonRecord” file is read by using the function fread

Then if the input ID is matched with anyID of the system then

The old record of the prisoner and the categories in which changes can be made are displayed.(Categories are assigned from numbers 1 to 12)

If they are not matched then

The record does not exist.

Continuing the case when both ID’s are matched

The user is asked to enter the choice of category which they want to edit or the whole record can also be edited.

Then the user can enter the new data of the particular category or the whole record.

If the user enters the number which is not for the categories i.e, not from 1 to 12 then the system shows that something wrong is typed so please try again.

The new edited record is then printed on the screen

The new edited record and the untouched records are written in the “temp” file using fwrite.

Then both the files are closed using fclose.

“PrisonRecord” file is deleted and the “temp” file is renamed as “PrisonRecord”.

After editing the information once, the system asks whether the user wants to edit another record or not

If yes then

The above process will be repeated once again.

If no then

Then the process will not be repeated again.

The user can then press any key to exit this menu.

**ViewRecord**

The code is for a program that will display the list of prisoners with their details.

Next a pointer named fp is assigned to the file named “PrisonRecord” and it is opened using fopen.

Then the name of categories of information is printed.

The file ”PrisonRecord” is read using the function fread and the information of all prisoners in different categories is printed.

The file is closed using the fclose function.

The user can then press any key to exit this menu.

**DeleteRecord**

The code is for a program that will display a delete menu.

The user has to enter the password and

If systemPassword and inputPassword are same then

Access is granted

If they are not same then

Access in not granted

After getting the access a pointer named fp is assigned to the file named “PrisonRecord”

If it is equal to null than

File doesn't exist

If is not equal to Null than

The file named “PrisonRecord” will be opened

A pointer named ft is assigned to a file named “temp” which is empty

If it is equal to null than

File doesn't exist

If is not equal to Null than

The file named “temp” will be opened

The ID of the record which has to be deleted is entered by the user.

The file “PrisonRecord” is read using the fread function.

The input ID is compared with each and every ID which is stored in the system then

The ID which are not matched with input ID

Are written in the temp file using fwrite

The ID which is matched

Is given a flag named “recordEntry” and its value is increased to 1.

Both the files are closed.

“PrisonRecord” file is deleted and the “temp” file is renamed as “PrisonRecord”.

Then if the ID has the flag named “recordEntry” with value 1 the system shows that

The record is deleted

Else the system show that

The record does not exist in the system

After deleting the record once, the system asks whether the user wants to delete another record or not

If yes then

The above process will be repeated once again.

If no then

Then the process will not be repeated again.

The user can then press any key to exit this menu.

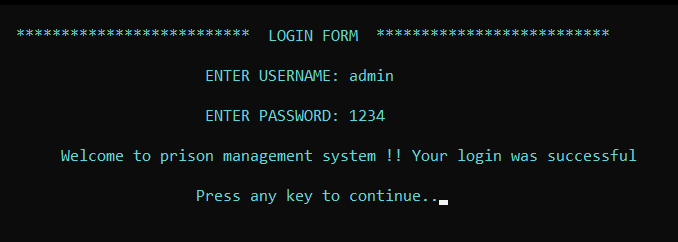
**User Credentials :**

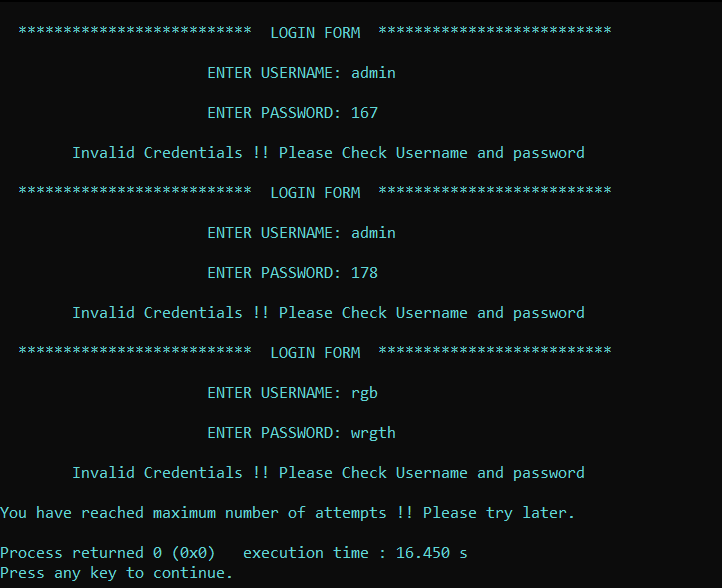
**Username :** admin

**Password :** 1234

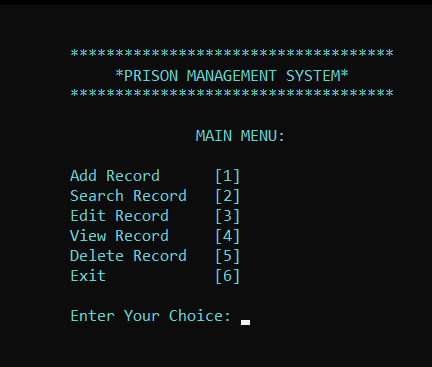
**Login Form**

**Case:** Login Successful



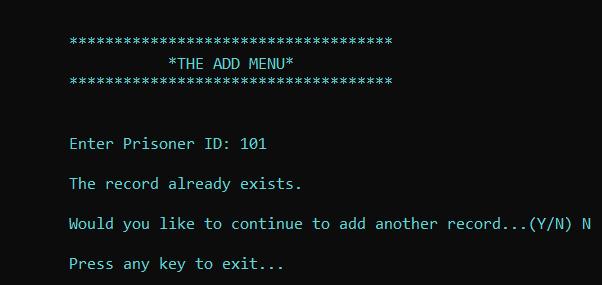
**Case:** Login Failed

**Main Menu**

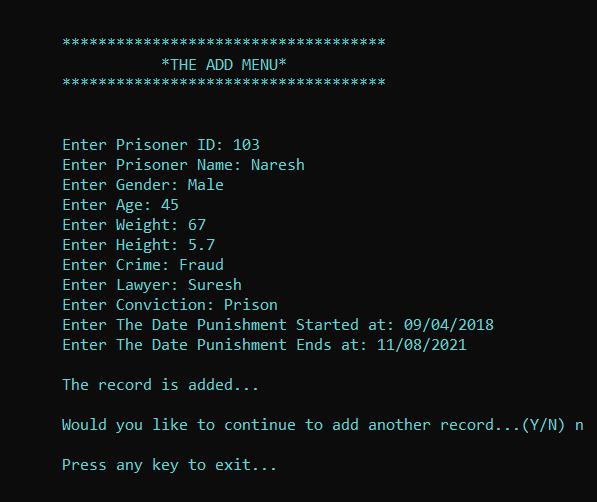


**Add Menu**

**Case:** Record Already Exists

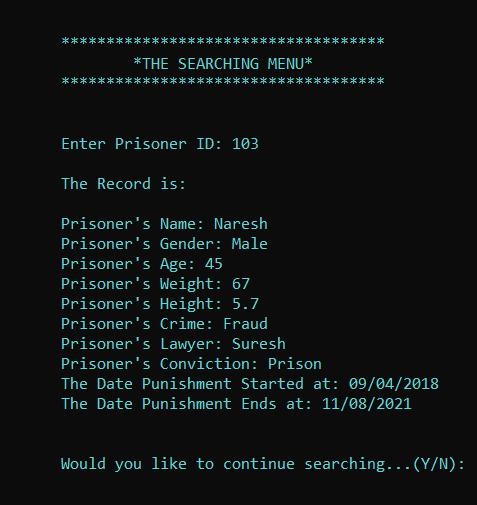


**Case:** New Record Added

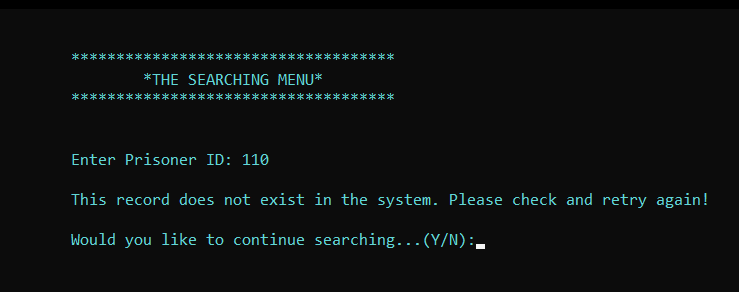


**Search Menu**

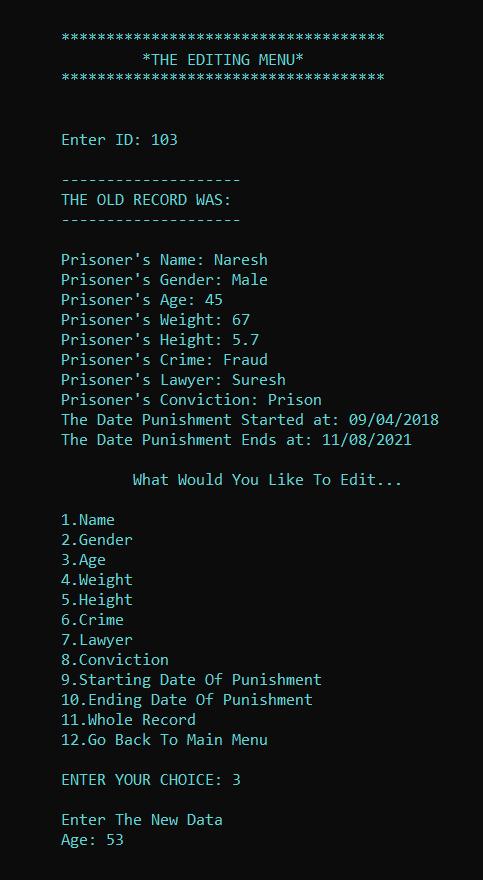
**Case:** Record Exists

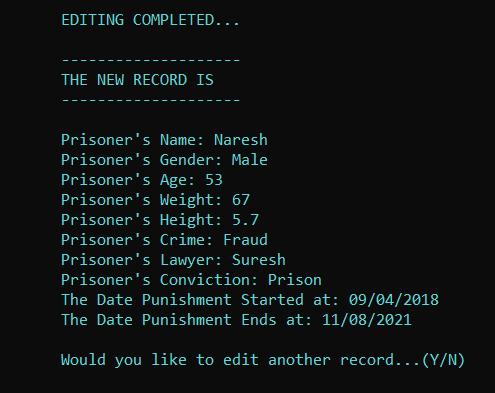


**Case:** Record Does Not Exist

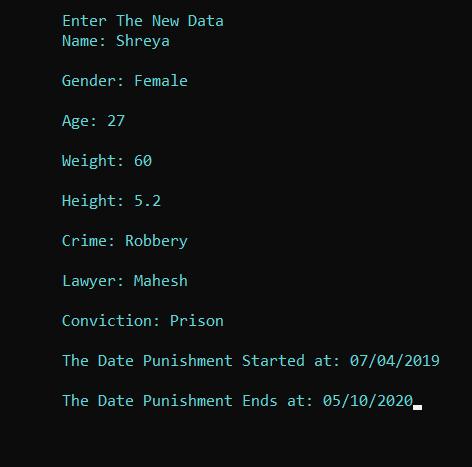


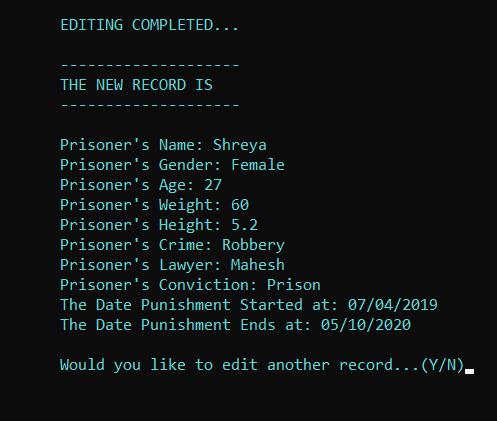
**Edit Menu**









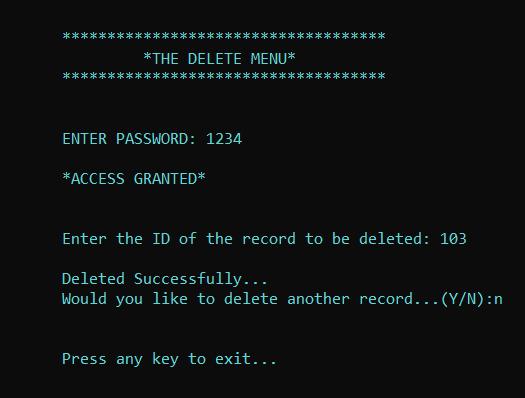


**View Menu**

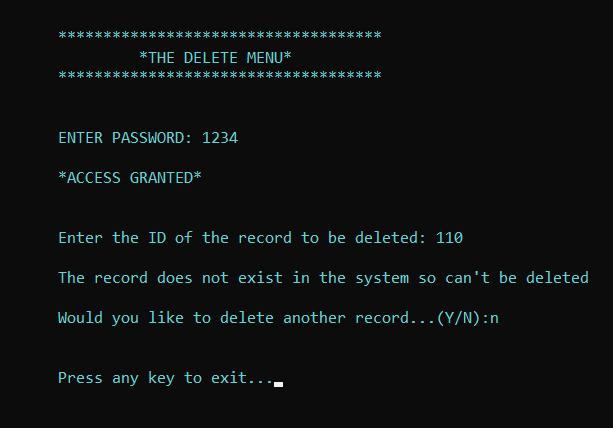


**Delete Menu**

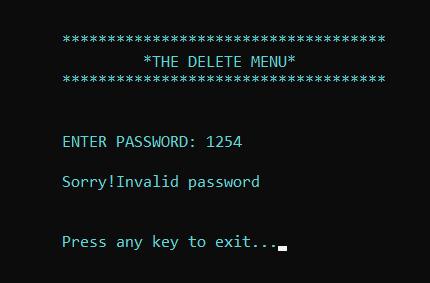
**Case:** Record Exists



**Case:** Record Does Not Exist



**Case:** Invalid Access



**Exit Menu**

