CRIME MANAGEMENT SYSTEM

MINI PROJECT REPORT

By

DHANUSH A [RA2211047010031]

Under the guidance of

Dr. RESHMY A K - Assistant Professor

In partial fulfilment for the Course

of

21CSC201J – DATA STRUCTURES AND ALGORITHMS

B.TECH in Artificial Intelligence

COLLEGE OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTATIONAL INTELLIGENCE

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR

NOVEMBER 2023

CONTENT

CONTRIBUTION TABLE

NAME	REG NO	CONTRIBUTION
PREJAN RAJA S	RA2211047010019	IMPLEMENTATION
SRI KRISHNA C	RA2211047010028	ANALYSIS
PRAGATISH A M	RA2211047010055	ALGORITHMS
DEEPAK KUMAR N	RA2211047010059	DOCUMENTATION

Abstract:

This Crime Management System in C programming is designed to manage prisoner records effectively. It allows administrators to perform CRUD (Create, Read, Update, Delete) operations on prisoner details. The system begins with a login process and provides functionalities such as adding new records, searching for specific prisoner information, editing existing records, viewing all records, and deleting records with proper authentication.

The system maintains various prisoner attributes including name, gender, age, weight, height, physical descriptions (hair and eye color), crime details, conviction information (court, lawyer, punishment), emergency contact details, and dates related to punishment. It provides an interface with a series of menus and options for administrators to interact and manage prisoner data efficiently.

Utilizing file handling in C, the system operates on a file named "filename" to store and manipulate prisoner records. It maintains data integrity by preventing duplicate records and enables the user to search for specific prisoner information by name and ID. Additionally, it incorporates a secure deletion mechanism by requiring a password for access.

The program ensures a secure login process to grant access to the system's functionalities. Although functional, it's essential to note that the system might benefit from improvements in security measures, error handling, and optimized file handling methods to enhance robustness and efficiency.

Overall, this Crime Management System provides a basic framework for managing prisoner records in a structured manner, catering to various administrative needs.

Please note, while the code offers a functional framework, it's essential to implement additional security measures, error handling, and efficiency improvements for real-world applications. Additionally, user Input validation and data integrity checks are crucial to prevent unexpected behavior or data corruption.

INTRODUCTION

The Crime Management System in C programming presents a comprehensive solution designed to efficiently manage and maintain records of individuals incarcerated within a correctional facility or involved in criminal activities. This system caters to the complex administrative tasks associated with tracking, storing, updating, and managing information pertaining to prisoners' personal details, crimes committed, legal proceedings, and associated contacts.

In modern society, the management of criminal records demands a structured approach to organize vast amounts of data concerning individuals detained or convicted for various offenses. Effective management of this information ensures streamlined administration within correctional facilities, aiding law enforcement agencies, courts, and legal practitioners in their daily operations.

This system's primary objective is to provide a user-friendly interface for authorized personnel, typically administrative staff or law enforcement officers, to perform essential tasks related to prisoner management. This includes adding new records for individuals entering the correctional system, searching for specific inmate details, modifying existing records, viewing a comprehensive list of incarcerated individuals, and securely deleting records when necessary.

The system incorporates fundamental functionalities such as user authentication through a login process to ensure secure access to its features. It also leverages file handling mechanisms in the C programming language to maintain a structured database of prisoner information, safeguarding data integrity and confidentiality.

Moreover, this system aims to enhance operational efficiency by facilitating quick and accurate retrieval of prisoner details, assisting authorities in making informed decisions regarding legal proceedings, punishment, and emergency contacts for inmates.

While this Crime Management System provides a foundational framework for managing prisoner records, it is imperative to continuously refine and fortify the system's capabilities. Strengthening security measures, implementing error handling mechanisms, and optimizing data management techniques are essential steps towards improving the system's reliability, security, and performance.

Overall, the Crime Management System in C programming offers a robust starting point for administrators and law enforcement personnel to oversee prisoner records systematically, ensuring effective management within correctional environments and supporting legal proceedings related to criminal activities.

Algorithm:

1. Initialize Variables and Data Structures:
   - Set up necessary variables for file handling, user authentication, and record management.
   - Define a structure to store prisoner details including ID, name, age, gender, physical attributes, crime information, legal proceedings, and emergency contacts.

2. Login Function:
   - Prompt the user for a username and password.
   - Allow a limited number of attempts for login.
   - Grant access upon successful authentication; otherwise, deny access.

3. Add Record Function:
   - Open the file for appending or create a new file if it doesn't exist.
   - Prompt the user for prisoner details like ID, name, gender, age, physical features, crime details, court information, lawyer, conviction, punishment dates, and emergency contacts.
   - Validate and add the new record to the file.

4. Search Record Function:
   - Prompt the user to input the prisoner's name or ID for search.
   - Open the file and search for matching records.
   - Display prisoner details if found; else, show a "Record Not Found" message.

5. Edit Record Function:
   - Prompt the user to input the prisoner's name or ID for editing.
   - Open the file, find the corresponding record, and display the existing details.
   - Allow the user to update specific fields or the entire record.
   - Save the updated information back to the file.

6. View Record Function:
   - Open the file and display all prisoner records one by one.
   - Implement navigation options for the user to scroll through records.

7. Delete Record Function:
   - Request a password to confirm the deletion process.
   - Prompt the user to input the name or ID of the prisoner to be deleted.
   - Open the file, remove the specified record, and update the file accordingly.

8. Error Handling and Security Measures:
   - Implement robust error handling mechanisms to manage unexpected inputs or file-related issues.
   - Ensure secure file handling techniques and user authentication to prevent unauthorized access.

9. Optimization and Refinement:
   - Evaluate and optimize file handling methods to improve data retrieval and storage efficiency.
   - Refine user interface elements for better usability and navigation within the system.

10. Future Enhancements:
    - Outline plans for future improvements such as incorporating advanced search capabilities, data analytics for crime trends, and integration with law enforcement databases.

This structured algorithm provides a comprehensive overview of the step-by-step process involved in building and utilizing the Crime Management System. Each step needs to be implemented in the programming code to create a functional system meeting the specified requirements.

IMPLEMENTATION/PROGRAM

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<string.h>
void addrecord();
void viewrecord();
void editrecord();
void searchrecord();
void deleterecord();
void login();
struct record
{
   char id[10];
   char name[30];
   char age[6];
   char gender[10];
   char weight[20];

   char height[20];

   char haircolor[20];

   char eyecolor[20];

   char crime[40];

   char convictedf[20];

   char court[20];
   char lawyer[20];
```

```
    char punishment[50];

    char punishments[20];

    char punishmente[20];

    char emergencyc[20];

    char emergencyr[20];

    char emergencyn[20];

} a;

int main()

{
        login();

    int ch;
    printf("\n\n\t*************************************\n");
    printf("\t\t*CRIME MANAGEMENT SYSTEM*\n");
    printf("\t*************************************");

    while(1)
    {

        printf("\n\n\t\tMAIN MENU:");
        printf("\n\n\tADD RECORD\t[1]");
        printf("\n\tSEARCH RECORD\t[2]");
        printf("\n\tEDIT RECORD\t[3]");
        printf("\n\tVIEW RECORD\t[4]");
```

```
            printf("\n\tDELETE RECORD\t[5]");
printf("\n\tEXIT\t\t[6]");
printf("\n\n\tENTER YOUR CHOICE:");
scanf("%d",&ch);

switch(ch)
{

case 1:
   addrecord();
   break;

case 2:
   searchrecord();
   break;

case 3:
   editrecord();
   break;

case 4:
   viewrecord();
   break;

case 5:
   deleterecord();
   break;

case 6:
   system("cls");
   printf("\n\n\t\tTHANK YOU FOR USING THIS SOFTWARE \n\t\t  ");
   getch();
```

```
            exit(0);

        default:
            printf("\nYOU ENTERED WRONG CHOICE.");
            printf("\nPRESS ANY KEY TO TRY AGAIN");
            getch();
            break;

        }

        system("cls");

    }
    return 0;

}

void addrecord( )
{
    system("cls");
    FILE *fp ;
    char another = 'Y' ,id[10];
    char filename[30];
    int choice;

    printf("\n\n\t\t*************************\n");
    printf("\t\t* WELCOME TO THE ADD MENU *");
    printf("\n\t\t*************************\n\n");
    printf("\n\n\tENTER FIRST NAME OF PRISONER:\t");
    fflush(stdin);
    gets(filename);
```

```
fp = fopen ("filename", "ab+" ) ;

if ( fp == NULL )
{

   fp=fopen("filename","wb+");
   if(fp==NULL)
   {

      printf("\nSYSTEM ERROR...");
      printf("\nPRESS ANY KEY TO EXIT");
      getch();
      return ;

   }
}
while ( another == 'Y'|| another=='y' )

{
   choice=0;
   fflush(stdin);

            printf ( "\n\tENTER PRISONER ID:\t");
   scanf("%s",id);

            rewind(fp);

   while(fread(&a,sizeof(a),1,fp)==1)
   {
      if(strcmp(a.id,id)==0)
      {
```

```
        printf("\n\tTHE RECORD ALREADY EXISTS.\n");
        choice=1;

    }

}

if(choice==0)
{
    strcpy(a.id,id);

    printf("\tENTER PRISONER NAME: ");
    fflush(stdin);
    gets(a.name);

                    printf("\tENTER GENDER: ");
    gets(a.gender);
    fflush(stdin);

    printf("\tENTER AGE: ");
    gets(a.age);
    fflush(stdin);

    printf("\tENTER WEIGHT: ");
    gets(a.weight);
    fflush(stdin);

    printf("\tENTER HEIGHT: ");
    gets(a.height);
                    fflush(stdin);

    printf("\tENTER HAIRCOLOR: ");
```

```
gets(a.haircolor);
fflush(stdin);

printf("\tENTER EYECOLOR: ");
gets(a.eyecolor);
fflush(stdin);

printf("\tENTER CRIME: ");
gets(a.crime);
fflush(stdin);

printf("\tENTER THE PLACE WHERE THE PRISONER WAS CONVICTED: ");
gets(a.convictedf);
fflush(stdin);
printf("\tENTER COURT: ");
gets(a.court);
fflush(stdin);

printf("\tENTER LAWYER: ");
gets(a.lawyer);
fflush(stdin);
printf("\tENTER CONVICTION: ");
gets(a.punishment);

fflush(stdin);

printf("\tENTER THE DATE PUNISHMENT STARTED AT: ");
gets(a.punishments);
fflush(stdin);

printf("\tENTER THE DATE PUNISHMENT ENDS AT: ");
gets(a.punishmente);
```

```
            fflush(stdin);

            printf("\tENTER NAME OF EMERGENCY CONTACT: ");
            gets(a.emergencyc);
            fflush(stdin);

            printf("\tENTER RELATION OF EMERGENCY CONTACT WITH PRISONER: ");
            gets(a.emergencyr);
            fflush(stdin);

            printf("\tENTER PHONE NUMBER OF EMERGENCY CONTACT: ");
            gets(a.emergencyn);

            fwrite ( &a, sizeof ( a ), 1, fp ) ;
            printf("\nYOUR RECORD IS ADDED...\n");

        }

        printf ( "\n\tADD ANOTHER RECORD...(Y/N) \t" ) ;
        fflush ( stdin ) ;
        another = getch( ) ;

    }

    fclose ( fp ) ;
    printf("\n\n\tPRESS ANY KEY TO EXIT...");
    getch();

}

void searchrecord( )
```

```
{
        system("cls");
   FILE *fp ;
        char id[16],choice,filename[14];
   int ch;

   printf("\n\n\t\t*********************************\n");
   printf("\t\t* HERE IS THE SEARCHING MENU *");
   printf("\n\t\t*********************************\n\n");

   do
        {

                printf("\n\tENTER THE PRISONER NAME TO BE SEARCHED:");
     fflush(stdin);
     gets(filename);

     fp = fopen ( "filename", "rb" ) ;



                printf("\nENTER PRISONER ID:");
        gets(id);
        system("cls");
        printf("\nTHE WHOLE RECORD IS:");



        while ( fread ( &a, sizeof ( a ), 1, fp ) == 1 )

       if(strcmpi(a.id,id)==0)
           { printf("\n");
            printf("\nPRISONER'S NAME IS: %s",a.name);
            printf("\nPRISONER'S GENDER IS: %s",a.gender);
```

```
            printf("\nPRISONER'S AGE IS: %s",a.age);
            printf("\nPRISONER'S WEIGHT IS: %s",a.weight);
            printf("\nPRISONER'S HEIGHT IS: %s",a.height);
            printf("\nPRISONER'S HAIRCOLOR IS: %s",a.haircolor);
                                    printf("\nPRISONER'S EYECOLOR IS: %s",a.eyecolor);
            printf("\nPRISONER'S CRIME IS: %s",a.crime);
            printf("\nTHE PLACE WHERE THE PRISONER WAS CONVICTED IS: %s",a.convictedf);
            printf("\nCOURT IS: %s",a.court);
            printf("\nPRISONER'S LAWYER IS: %s",a.lawyer);
            printf("\nPRISONER'S CONVICTION IS: %s",a.punishment);
            printf("\nTHE DATE PUNISHMENT STARTED AT IS: %s",a.punishments);
            printf("\nTHE DATE PUNISHMENT ENDS AT IS: %s",a.punishmente);
            printf("\nPRISONER'S EMERGENCY CONTACT IS: %s",a.emergencyc);
            printf("\nRELATION OF EMERGENCY CONTACT WITH PRISONER IS:
%s",a.emergencyr);
            printf("\nEMERGENCY CONTACT'S PHONE NUMBER IS: %s",a.emergencyn);
            printf("\n");
          }

         // }


      printf("\n\nWOULD YOU LIKE TO CONTINUE VIEWING...(Y/N):");
      fflush(stdin);

      scanf("%c",&choice);

   }
   while(choice=='Y'||choice=='y');

   fclose ( fp) ;
         getch();
```

```
    return ;
getch();
}


void editrecord()

{

    system("cls");
    FILE *fp ;
    char id[10],choice,filename[14];
    int num,count=0;

    printf("\n\n\t\t*********************************\n");
    printf("\t\t* WELCOME TO THE EDITING MENU *");
    printf("\n\t\t*********************************\n\n");

    do
    {

        printf("\n\tENTER THE PRISONER NAME TO BE EDITED:");
        fflush(stdin);
        gets(filename);

        printf("\n\tENTER ID:");
            gets(id);

        fp = fopen ( "filename", "rb+" ) ;


        while ( fread ( &a, sizeof ( a ), 1, fp ) == 1 )
```

```
{

   if(strcmp(a.id,id)==0)
   {

      printf("\nYOUR OLD RECORD WAS AS:");
      printf("\nPRISONER'S NAME: %s",a.name);
      printf("\nPRISONER'S GENDER: %s",a.gender);
      printf("\nPRISONER'S AGE: %s",a.age);
      printf("\nPRISONER'S WEIGHT: %s",a.weight);
      printf("\nPRISONER'S HEIGHT: %s",a.height);
      printf("\nPRISONER'S HAIRCOLOR: %s",a.haircolor);
                              printf("\nPRISONER'S EYECOLOR: %s",a.eyecolor);
      printf("\nPRISONER'S CRIME: %s",a.crime);
      printf("\nTHE PLACE WHERE THE PRISONER WAS CONVICTED: %s",a.convictedf);
      printf("\nCOURT: %s",a.court);
      printf("\nPRISONER'S LAWYER: %s",a.lawyer);
      printf("\nPRISONER'S CONVICTION: %s",a.punishment);
      printf("\nTHE DATE PUNISHMENT STARTED AT: %s",a.punishments);
      printf("\nTHE DATE PUNISHMENT ENDS AT: %s",a.punishmente);
      printf("\nPRISONER'S EMERGENCY CONTACT: %s",a.emergencyc);
      printf("\nRELATION OF EMERGENCY CONTACT WITH PRISONER: %s",a.emergencyr);
      printf("\nEMERGENCY CONTACT'S PHONE NUMBER: %s",a.emergencyn);



      printf("\n\n\t\tWHAT WOULD YOU LIKE TO EDIT..");



      printf("\n1.NAME.");
      printf("\n2.GENDER.");
```

```
printf("\n3.AGE.");
printf("\n4.WEIGHT.");
printf("\n5.HEIGHT.");
printf("\n6.HAIRCOLOR.");
printf("\n7.EYECOLOR.");
printf("\n8.CRIME.");
printf("\n9.PLACE WHERE THE PRISONER WAS CONVICTED.");
printf("\n10.COURT.");
printf("\n11.LAWYER.");
printf("\n12.CONVICTION.");
printf("\n13.STARTING DATE OF PUNISHMENT.");
printf("\n14.ENDING DATE OF PUNISHMENT.");
printf("\n15.EMERGENCY CONTACT.");
printf("\n16.RELATION OF EMERGENCY CONTACT.");
printf("\n17.EMERGENCY CONTACT'S PHONE NUMBER.");
printf("\n18.WHOLE RECORD.");
printf("\n19.GO BACK TO MAIN MENU.");

do
{

   printf("\n\tENTER YOUR CHOICE:");
   fflush(stdin);
   scanf("%d",&num);
   fflush(stdin);

   switch(num)
   {



   case 1:
      printf("\nENTER THE NEW DATA:");
```

```
    printf("\nNAME:");
    gets(a.name);
    break;

case 2:
    printf("\nENTER THE NEW DATA:");
    printf("\nGENDER:");
    gets(a.gender);
    break;

case 3:
    printf("\nENTER THE NEW DATA:");
    printf("\nAGE:");
    gets(a.age);
    break;

case 4:
    printf("\nENTER THE NEW DATA:");
    printf("\nWEIGHT:");
    gets(a.weight);
    break;

case 5:
    printf("ENTER THE NEW DATA:");
    printf("\nHEIGHT:");
    gets(a.height);
    break;

case 6:
    printf("ENTER THE NEW DATA:");

    printf("\nHAIRCOLOR:");
```

```
    gets(a.haircolor);
    break;

case 7:
    printf("ENTER THE NEW DATA:");
    printf("\nEYECOLOR:");
    gets(a.eyecolor);
    break;

case 8:
    printf("ENTER THE NEW DATA:");
    printf("\nCRIME:");
    gets(a.crime);
    break;

case 9:
    printf("ENTER THE NEW DATA:");
    printf("\nPLACE:");
    gets(a.convictedf);
    break;

case 10:
    printf("ENTER THE NEW DATA:");
    printf("\nCOURT:");
    gets(a.court);
    break;

case 11:
    printf("ENTER THE NEW DATA:");
    printf("\nLAWYER:");
    gets(a.lawyer);
    break;
```

```
case 12:
   printf("ENTER THE NEW DATA:");
   printf("\nCONVICTION:");
   gets(a.punishment);
   break;

case 13:
   printf("ENTER THE NEW DATA:");
   printf("\nSTARTING DATE OF PUNISHMENT:");
   gets(a.punishments);
   break;

case 14:
   printf("ENTER THE NEW DATA:");
   printf("\nENDING DATE OF PUNISHMENT:");
   gets(a.punishmente);
   break;

case 15:
   printf("ENTER THE NEW DATA:");
   printf("\nEMERGENCY CONTACT:");
   gets(a.emergencyc);
   break;

case 16:
   printf("ENTER THE NEW DATA:");
   printf("\nRELATION OF EMERGENCY CONTACT:");
   gets(a.emergencyr);
   break;
```

```
case 17:
   printf("ENTER THE NEW DATA:");
   printf("\nPHONE NUMBER OF EMERGENCY CONTACT:");
   gets(a.emergencyc);
   break;
case 18:
   printf("ENTER THE NEW DATA:");
   printf("\tPRISONER NAME:");
                  gets(a.name);
                                 printf("\tGENDER:");
                  gets(a.gender);
                  printf("\tAGE:");
                  gets(a.age);
                  printf("\tWEIGHT:");
                  gets(a.weight);
                  printf("\tHEIGHT:");
                  gets(a.height);
                  printf("\tHAIRCOLOR:");
                  gets(a.haircolor);
                  printf("\tEYECOLOR:");
                  gets(a.eyecolor);
                  printf("\tCRIME:");
                  gets(a.age);
                  printf("\tTHE PLACE WHERE THE PRISONER WAS CONVICTED:");
                  gets(a.convictedf);
                  printf("\tCOURT:");
                  gets(a.court);
                  printf("\tLAWYER:");
                  gets(a.lawyer);
                  printf("\tCONVICTION:");
                  gets(a.punishment);
```

```
                    printf("\tTHE DATE PUNISHMENT STARTED AT:");

                    gets(a.punishments);
                    printf("\tTHE DATE PUNISHMENT ENDS AT:");
                    gets(a.punishmente);
                    printf("\tNAME OF EMERGENCY CONTACT:");
                    gets(a.emergencyc);
                    printf("\tRELATION OF EMERGENCY CONTACT WITH PRISONER:");
                    gets(a.emergencyr);
                    printf("\tPHONE NUMBER OF EMERGENCY CONTACT:");
                    gets(a.emergencyn);
      break;

   case 19:
      printf("\nPRESS ANY KEY TO GO BACK...\n");
      getch();
      return ;
      break;

   default:
      printf("\nYOU TYPED SOMETHING ELSE...TRY AGAIN\n");
      break;

   }

}
while(num<1||num>20);

fseek(fp,-sizeof(a),SEEK_CUR);

fwrite(&a,sizeof(a),1,fp);
```

```
        fseek(fp,-sizeof(a),SEEK_CUR);


        fread(&a,sizeof(a),1,fp);

        choice=5;

        break;

    }
}

if(choice==5)

{

    system("cls");

    printf("\n\t\tEDITING COMPLETED...\n");
    printf("--------------------\n");
    printf("THE NEW RECORD IS:\n");
    printf("--------------------\n");
    printf("\nPRISONER'S NAME IS: %s",a.name);
    printf("\nPRISONER'S GENDER IS: %s",a.gender);
    printf("\nPRISONER'S AGE IS: %s",a.age);
    printf("\nPRISONER'S WEIGHT IS: %s",a.weight);
    printf("\nPRISONER'S HEIGHT IS: %s",a.height);
    printf("\nPRISONER'S HAIRCOLOR IS: %s",a.haircolor);
                    printf("\nPRISONER'S EYECOLOR IS: %s",a.eyecolor);
    printf("\nPRISONER'S CRIME IS: %s",a.crime);
    printf("\nTHE PLACE WHERE THE PRISONER WAS CONVICTED IS: %s",a.convictedf);
    printf("\nCOURT IS: %s",a.court);
```

```
        printf("\nPRISONER'S LAWYER IS: %s",a.lawyer);
                printf("\nPRISONER'S CONVICTION IS: %s",a.punishment);
        printf("\nTHE DATE PUNISHMENT STARTED AT IS: %s",a.punishments);

        printf("\nTHE DATE PUNISHMENT ENDS AT IS: %s",a.punishmente);
        printf("\nPRISONER'S EMERGENCY CONTACT IS: %s",a.emergencyc);
        printf("\nRELATION OF EMERGENCY CONTACT WITH PRISONER IS: %s",a.emergencyr);
        printf("\nEMERGENCY CONTACT'S PHONE NUMBER IS: %s",a.emergencyn);

        fclose(fp);

        printf("\n\n\tWOULD YOU LIKE TO EDIT ANOTHER RECORD.(Y/N)");
        scanf("%c",&choice);
        count++;

      }
      else
      {

        printf("\nTHE RECORD DOES NOT EXIST::\n");
        printf("\nWOULD YOU LIKE TO TRY AGAIN...(Y/N)");
        scanf("%c",&choice);

      }
   }
   while(choice=='Y'||choice=='y');

   fclose ( fp ) ;
   printf("\tPRESS ENTER TO EXIT EDITING MENU.");
   getch();

}
```

```
void viewrecord()



{
   system("cls");
   FILE *fp;
   char filename[30];

   printf("\n\n\t\t**************************\n");
   printf("\t\t   * LIST OF PRISONERS *");
   printf("\n\t\t**************************\n\n");

   fp=fopen("filename","rb");
   rewind(fp);
   while((fread(&a,sizeof(a),1,fp))==1)
   {
        printf("||NOTE:-PRESS ENTER KEY TO LOAD OTHER RECORDS||\n");
      printf("\nPRISONER'S NAME IS: %s",a.name);
      printf("\nPRISONER'S GENDER IS: %s",a.gender);
      printf("\nPRISONER'S AGE IS: %s",a.age);
      printf("\nPRISONER'S WEIGHT IS: %s",a.weight);
      printf("\nPRISONER'S HEIGHT IS: %s",a.height);
      printf("\nPRISONER'S HAIRCOLOR IS: %s",a.haircolor);
                printf("\nPRISONER'S EYECOLOR IS: %s",a.eyecolor);
      printf("\nPRISONER'S CRIME IS: %s",a.crime);
      printf("\nTHE PLACE WHERE THE PRISONER WAS CONVICTED IS: %s",a.convictedf);
      printf("\nCOURT IS: %s",a.court);
      printf("\nPRISONER'S LAWYER IS: %s",a.lawyer);
      printf("\nPRISONER'S CONVICTION IS: %s",a.punishment);
      printf("\nTHE DATE PUNISHMENT STARTED AT IS: %s",a.punishments);
```

```
        printf("\nTHE DATE PUNISHMENT ENDS AT IS: %s",a.punishmente);
        printf("\nPRISONER'S EMERGENCY CONTACT IS: %s",a.emergencyc);
        printf("\nRELATION OF EMERGENCY CONTACT WITH PRISONER IS: %s",a.emergencyr);
        printf("\nEMERGENCY CONTACT'S PHONE NUMBER IS: %s",a.emergencyn);
        getch();

     printf("\n\n");

    }
    fclose(fp);
          getch();

}


void deleterecord( )

{

    system("cls");
    FILE *fp,*ft ;
    struct record file ;
    char filename[15],another = 'Y' ,id[16];;
    int choice,check;
    int j=0;
    char pass[8];

    printf("\n\n\t\t**************************\n");
    printf("\t\t* WELCOME TO DELETE MENU*");
    printf("\n\t\t**************************\n\n");

    printf("\nENTER PASSWORD\n");
```

```
int i;
for(  i=0;i<4;i++)
{
        pass[i]=getch();
        printf("*");
        }

        if (strcmpi(pass,"pass")==0)
        {

  printf("\n\t\t*ACCESS GRANTED*\n\n");
while ( another == 'Y' || another == 'y' )

{

        printf("\n\tENTER THE NAME OF PRISONER TO BE DELETED:");
                fflush(stdin);
    gets(filename);
    fp = fopen ("filename", "rb" ) ;
    if ( fp == NULL )
          {
             printf("\nTHE FILE DOES NOT EXIST");
             printf("\nPRESS ANY KEY TO GO BACK.");
             getch();
             return ;
                                }
                                ft=fopen("temp","wb");

                                if(ft==NULL)
          {
                printf("\nSYSTEM ERROR");
             printf("\nPRESS ANY KEY TO GO BACK");
```

```
            getch();
            return ;
        }
        printf("\n\tENTER THE ID OF RECORD TO BE DELETED:");
        fflush(stdin);
        gets(id);


            while(fread(&file,sizeof(file),1,fp)==1)

        {

            if(strcmp(file.id,id)!=0)

                fwrite(&file,sizeof(file),1,ft);

        }
    fclose(ft);
    fclose(fp);
    remove("filename");
    rename("temp","filename");
    printf("\nDELETED SUCCESFULLY...");
    getch();

    printf("\n\tDO YOU LIKE TO DELETE ANOTHER RECORD.(Y/N):");
    fflush(stdin);
    scanf("%c",&another);

}


printf("\n\n\tPRESS ANY KEY TO EXIT...");
```

```
   getch();
}
        else
        {
                printf("\nSorry!Invalid password\n");
                exit(0);
        }

}

void login()
{
        int a=0,i=0;
   char uname[10],c=' ';
   char pword[10],code[10];
   char user[10]="user";
   char pass[10]="pass";
   do
{

   printf("\n  *************************  LOGIN FORM  *************************  ");
   printf(" \n                    ENTER USERNAME:-");
        scanf("%s", &uname);
        printf(" \n                    ENTER PASSWORD:-");
        while(i<10)
        {
           pword[i]=getch();
           c=pword[i];
           if(c==13) break;
           else printf("*");
           i++;
```

```
        }
        pword[i]='\0';
        //char code=pword;
        i=0;
        //scanf("%s",&pword);
                if(strcmp(uname,"user")==0 && strcmp(pword,"pass")==0)
        {
        printf("   \n\n\n        WELCOME TO PRISON MANAGEMENT SYSTEM !! YOUR LOGIN IS SUCCESSFUL");

        printf("\n\n\n\t\t\t\tPress any key to continue...");
        getch();
        break;
        }
        else
        {
                printf("\n       SORRY !!!!  LOGIN IS UNSUCESSFUL");
                a++
                getch();//holds the screen
        }
}
        while(a<=2);
        if (a>2)
        {
                printf("\nSorry you have entered the wrong username and password for four times!!!");
                getch();
                }
                system("cls");
}
```

The provided Crime Management System in C implements functions to manage prisoner records. It includes functionalities like user authentication, adding, searching, editing, deleting, and viewing records

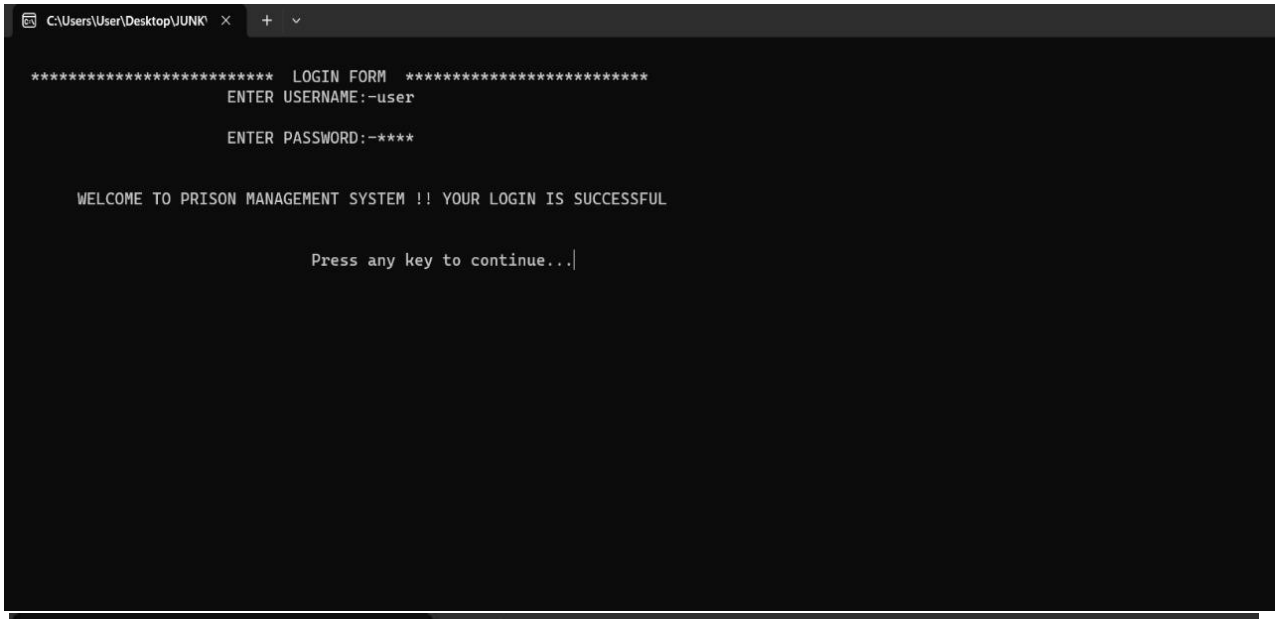
stored in a file. Here's a concise breakdown:

- Libraries: Imports necessary C libraries for input/output, memory allocation, and string manipulation.
- Structure: Defines a 'record' structure to hold various prisoner details.
- Functions:
- login(): Handles user authentication with predefined credentials.
- addRecord(): Adds new prisoner details to a file.
- searchRecord(): Searches and displays specific prisoner records.
- editRecord(): Allows modification of existing records.
- deleteRecord(): Deletes records with password confirmation.

- viewRecord(): Displays all stored prisoner records.
- Main Function: Controls the program's flow by presenting a menu for user interactions.
- File Operations: Manages prisoner data by interacting with a file named "filename".
- User Interaction: Prompts users for input, displays retrieved data, and provides confirmation messages.
- Password Protection: Secures critical functions, like record deletion, with a specific password.

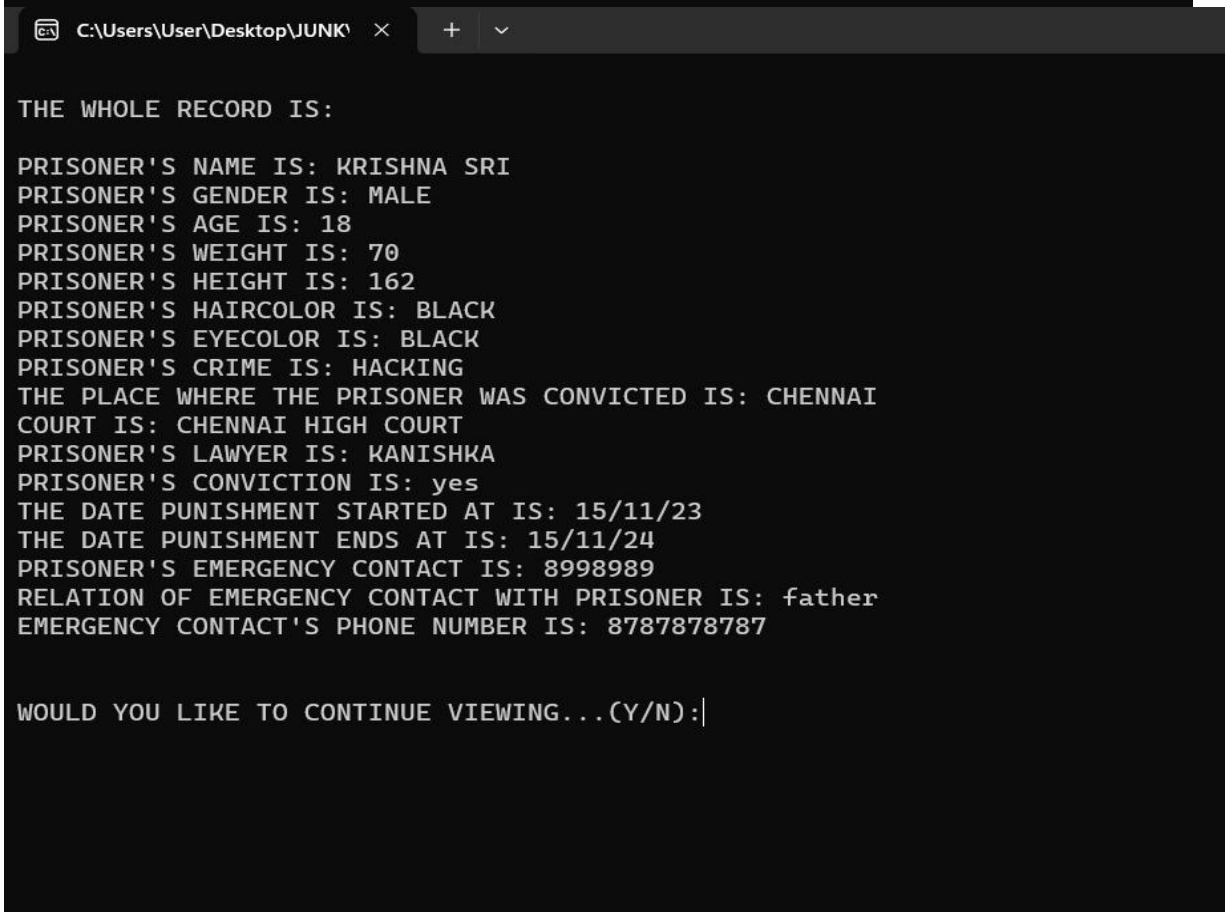
This C program employs file handling, user authentication, conditional statements, and input/output operations to effectively manage prisoner records through a text-based interface.

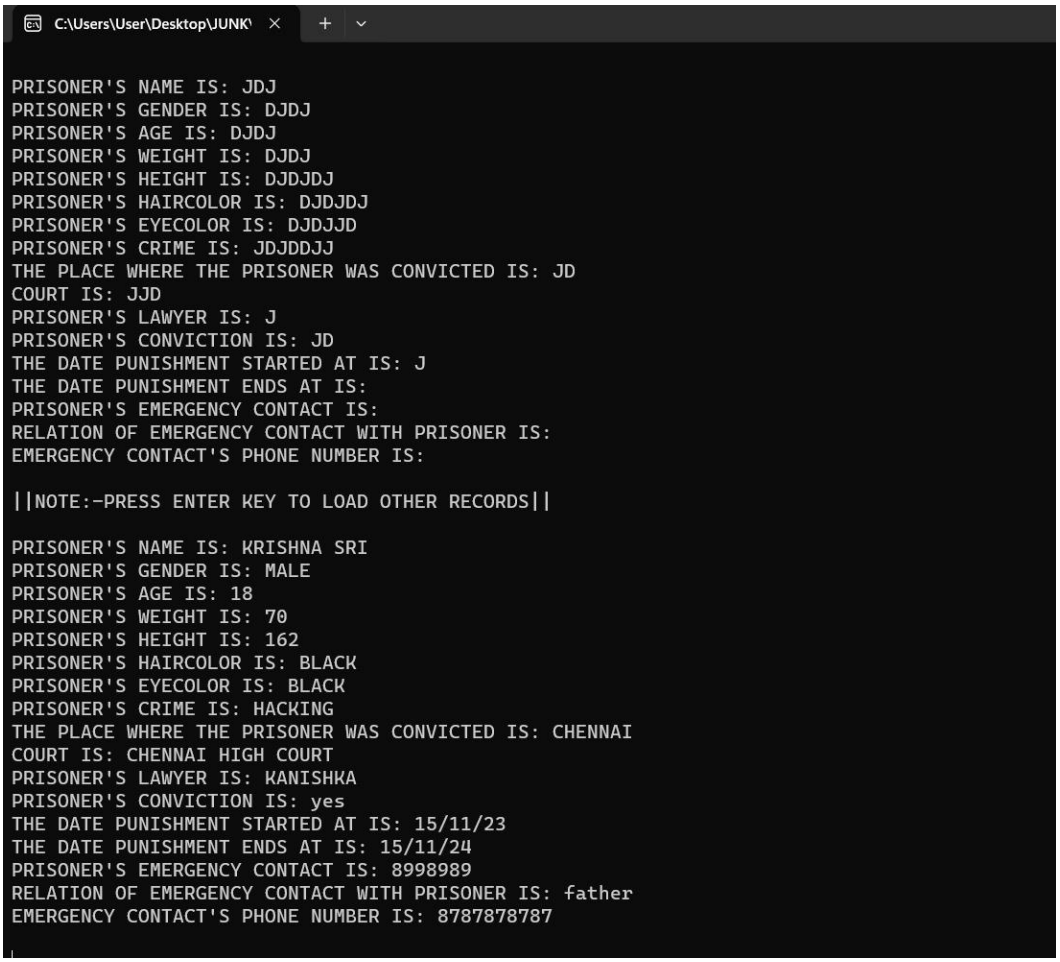
OUTPUT

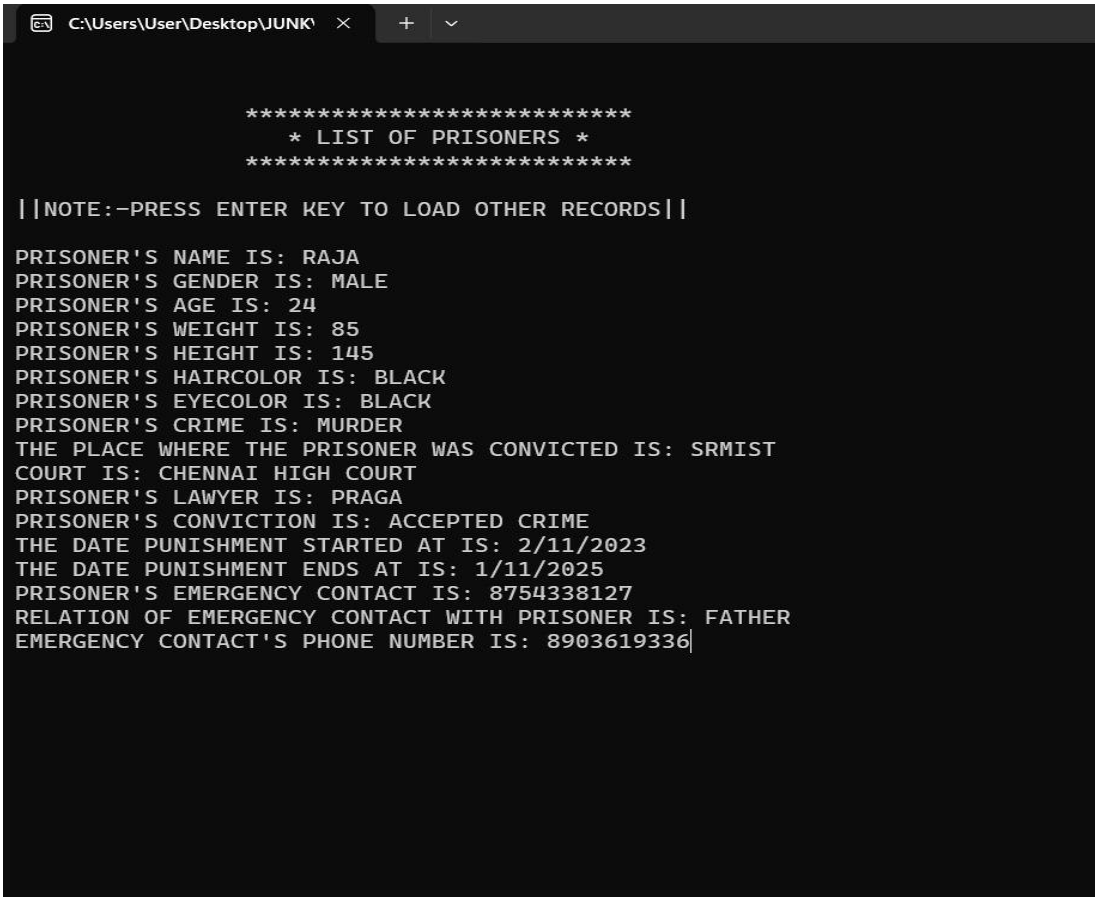

C:\Users\User\Desktop\JUNK
********** LOGIN FORM **********
ENTER USERNAME:-user
ENTER PASSWORD:-****
WELCOME TO PRISON MANAGEMENT SYSTEM !! YOUR LOGIN IS SUCCESSFUL
Press any key to continue...


C:\Users\User\Desktop\JUNK
****
PRISONER RECORD
****
MAIN MENU:
ADD RECORD [1]
SEARCH RECORD [2]
EDIT RECORD [3]
VIEW RECORD [4]
DELETE RECORD [5]
EXIT [6]
ENTER YOUR CHOICE:

```
               *************************
               * WELCOME TO THE ADD MENU *
               *************************



       ENTER FIRST NAME OF PRISONER:   KRISHNA

       ENTER PRISONER ID:      2
       ENTER PRISONER NAME: KRISHNA SRI
       ENTER GENDER: MALE
       ENTER AGE: 18
       ENTER WEIGHT: 70
       ENTER HEIGHT: 162
       ENTER HAIRCOLOR: BLACK
       ENTER EYECOLOR: BLACK
       ENTER CRIME: HACKING
       ENTER THE PLACE WHERE THE PRISONER WAS CONVICTED: CHENNAI
       ENTER COURT: CHENNAI HIGH COURT
       ENTER LAWYER: KANISHKA
       ENTER CONVICTION: yes
       ENTER THE DATE PUNISHMENT STARTED AT: 15/11/23
       ENTER THE DATE PUNISHMENT ENDS AT: 15/11/24
       ENTER NAME OF EMERGENCY CONTACT: 8998989
       ENTER RELATION OF EMERGENCY CONTACT WITH PRISONER: father
       ENTER PHONE NUMBER OF EMERGENCY CONTACT: 8787878787

YOUR RECORD IS ADDED...

       ADD ANOTHER RECORD...(Y/N)
```




C:\Users\User\Desktop\JUNK\
PRISONER'S NAME IS: JDJ
PRISONER'S GENDER IS: DJDJ
PRISONER'S AGE IS: DJDJ
PRISONER'S WEIGHT IS: DJDJ
PRISONER'S HEIGHT IS: DJDJDJ
PRISONER'S HAIRCOLOR IS: DJDJDJ
PRISONER'S EYECOLOR IS: DJDJJD
PRISONER'S CRIME IS: JDJDDJJ
THE PLACE WHERE THE PRISONER WAS CONVICTED IS: JD
COURT IS: JJD
PRISONER'S LAWYER IS: J
PRISONER'S CONVICTION IS: JD
THE DATE PUNISHMENT STARTED AT IS: J
THE DATE PUNISHMENT ENDS AT IS:
PRISONER'S EMERGENCY CONTACT IS:
RELATION OF EMERGENCY CONTACT WITH PRISONER IS:
EMERGENCY CONTACT'S PHONE NUMBER IS:
||NOTE:-PRESS ENTER KEY TO LOAD OTHER RECORDS||
PRISONER'S NAME IS: KRISHNA SRI
PRISONER'S GENDER IS: MALE
PRISONER'S AGE IS: 18
PRISONER'S WEIGHT IS: 70
PRISONER'S HEIGHT IS: 162
PRISONER'S HAIRCOLOR IS: BLACK
PRISONER'S EYECOLOR IS: BLACK
PRISONER'S CRIME IS: HACKING
THE PLACE WHERE THE PRISONER WAS CONVICTED IS: CHENNAI
COURT IS: CHENNAI HIGH COURT
PRISONER'S LAWYER IS: KANISHKA
PRISONER'S CONVICTION IS: yes
THE DATE PUNISHMENT STARTED AT IS: 15/11/23
THE DATE PUNISHMENT ENDS AT IS: 15/11/24
PRISONER'S EMERGENCY CONTACT IS: 8998989
RELATION OF EMERGENCY CONTACT WITH PRISONER IS: father
EMERGENCY CONTACT'S PHONE NUMBER IS: 8787878787


C:\Users\User\Desktop\JUNK\
****
* LIST OF PRISONERS *
****
||NOTE:-PRESS ENTER KEY TO LOAD OTHER RECORDS||
PRISONER'S NAME IS: RAJA
PRISONER'S GENDER IS: MALE
PRISONER'S AGE IS: 24
PRISONER'S WEIGHT IS: 85
PRISONER'S HEIGHT IS: 145
PRISONER'S HAIRCOLOR IS: BLACK
PRISONER'S EYECOLOR IS: BLACK
PRISONER'S CRIME IS: MURDER
THE PLACE WHERE THE PRISONER WAS CONVICTED IS: SRMIST
COURT IS: CHENNAI HIGH COURT
PRISONER'S LAWYER IS: PRAGA
PRISONER'S CONVICTION IS: ACCEPTED CRIME
THE DATE PUNISHMENT STARTED AT IS: 2/11/2023
THE DATE PUNISHMENT ENDS AT IS: 1/11/2025
PRISONER'S EMERGENCY CONTACT IS: 8754338127
RELATION OF EMERGENCY CONTACT WITH PRISONER IS: FATHER
EMERGENCY CONTACT'S PHONE NUMBER IS: 8903619336

```
                ************************
                * WELCOME TO DELETE MENU*
                ************************


ENTER PASSWORD
****
                *ACCESS GRANTED*


        ENTER THE NAME OF PRISONER TO BE DELETED:KRISHNA

        ENTER THE ID OF RECORD TO BE DELETED:2

DELETED SUCCESFULLY...
        DO YOU LIKE TO DELETE ANOTHER RECORD.(Y/N):N


        PRESS ANY KEY TO EXIT...
```

```
                THANK YOU FOR USING THIS SOFTWARE

--------------------------------
Process exited after 138.1 seconds with return value 0
Press any key to continue . . .
```

Applications:

1. Law Enforcement Agencies: Helps law enforcement agencies maintain a comprehensive database of criminals, aiding in tracking, monitoring, and managing criminal records efficiently.
2. Prison Administration: Assists prison authorities in managing inmate information, including personal details, convictions, legal proceedings, and emergency contacts.
3. Legal Firms and Courts: Provides a repository for legal professionals to access information about criminals, their cases, court proceedings, and associated legal details.
4. Criminal Investigation: Facilitates investigations by storing and retrieving crime-related information, assisting detectives and investigators in tracking criminal histories and patterns.
5. Crime Analysis and Research: Offers data for statistical analysis, helping researchers and analysts in understanding crime trends, demographics, and patterns for policymaking and preventive measures.
6. Emergency Response and Contact Management: Stores emergency contact details, aiding emergency response units in contacting next of kin or designated contacts if required.
7. Historical Records and Reporting: Maintains historical data of criminal activities, providing a foundation for generating reports and statistical analyses for various purposes.
8. Efficient Information Retrieval: Enables swift retrieval of specific prisoner information, aiding officials in making informed decisions swiftly.
9. Security and Access Control: Ensures data security by implementing authentication mechanisms for accessing sensitive information, preventing unauthorized access or alterations.
10. Compliance and Governance: Helps in complying with legal regulations by maintaining accurate and up-to-date records, contributing to governance and legal compliance standards.

The Crime Management System serves as a robust tool for various stakeholders involved in law enforcement, legal proceedings, and crime analysis, enhancing operational efficiency and information management in dealing with criminal cases and prison administration.

Conclusion:

The Crime Management System developed in C programming offers a crucial tool for law enforcement agencies, prison administrations, legal professionals, and investigative bodies in managing and accessing comprehensive prisoner records. This project embodies the essence of effective data organization, user interaction, and file handling capabilities to streamline the management of criminal information.

One of its significant strengths lies in providing an efficient and user-friendly interface for adding, searching, editing, deleting, and viewing prisoner records. Through functionalities such as user authentication and password protection, the system ensures data security, preventing unauthorized access to sensitive information and enabling controlled management of records.

Moreover, the system's practical applications span across various domains. It aids law enforcement by facilitating criminal tracking, assists legal professionals in accessing pertinent information for court proceedings, and supports researchers in analyzing crime patterns for policy formulation. Its utility in emergency response scenarios by maintaining emergency contact details further enhances its relevance and usefulness.

Additionally, the project underscores the significance of data management in crime analysis and law enforcement. By maintaining a structured database and offering features for quick data retrieval, it contributes to informed decision-making and efficient handling of criminal cases. However, the system might benefit from enhancements such as improved error handling, better data validation, and possibly a more robust data encryption mechanism for heightened security.

In conclusion, the Crime Management System stands as an essential tool in the realm of law enforcement and criminal justice, demonstrating the importance of organized data systems in effectively managing, analyzing, and utilizing information pertaining to criminal records and proceedings. Its utility, though robust, can further evolve with advancements in technology and continuous refinement to meet the ever-growing demands of the criminal justice landscape.

References:

- "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie
- "C Programming Absolute Beginner's Guide" by Greg Perry and Dean Miller
- "Programming in C" by Stephen G. Kochan
- Online Resources and Tutorials
- GitHub Repositories
- https://google.com/
- https://chatgpt.com//
- https://youtube.com/