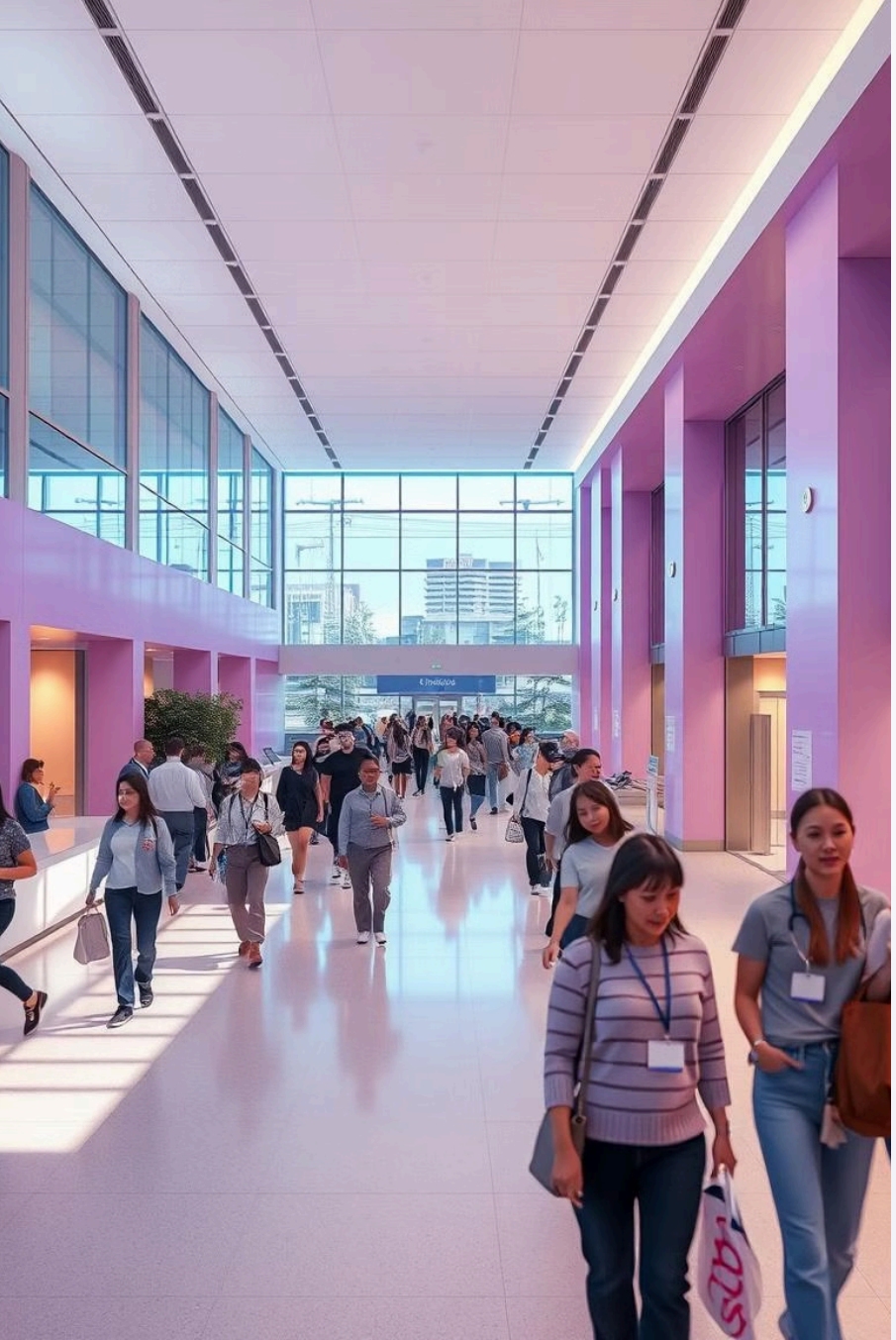
Welcome

Department of Computer Science



Hospital Management System in C++

This presentation will discuss the design and implementation of a hospital management system in C++. We'll explore key features, functionality, and future enhancements, ultimately demonstrating the system's ability to streamline operations and improve patient care.

Team Members

Aeliya Haider

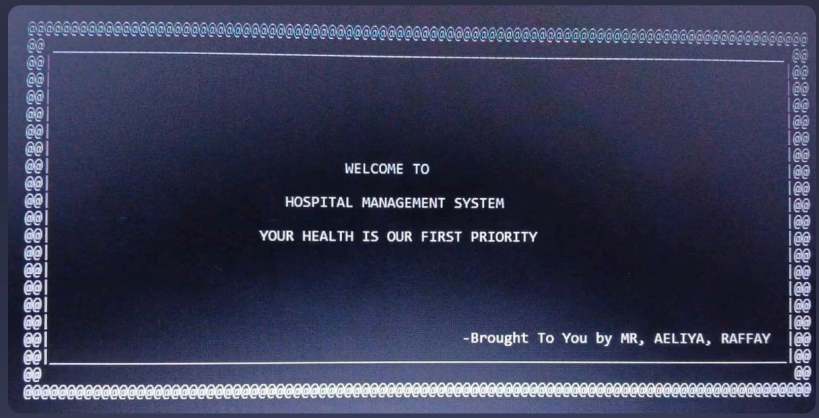
Mohsin Ahmed

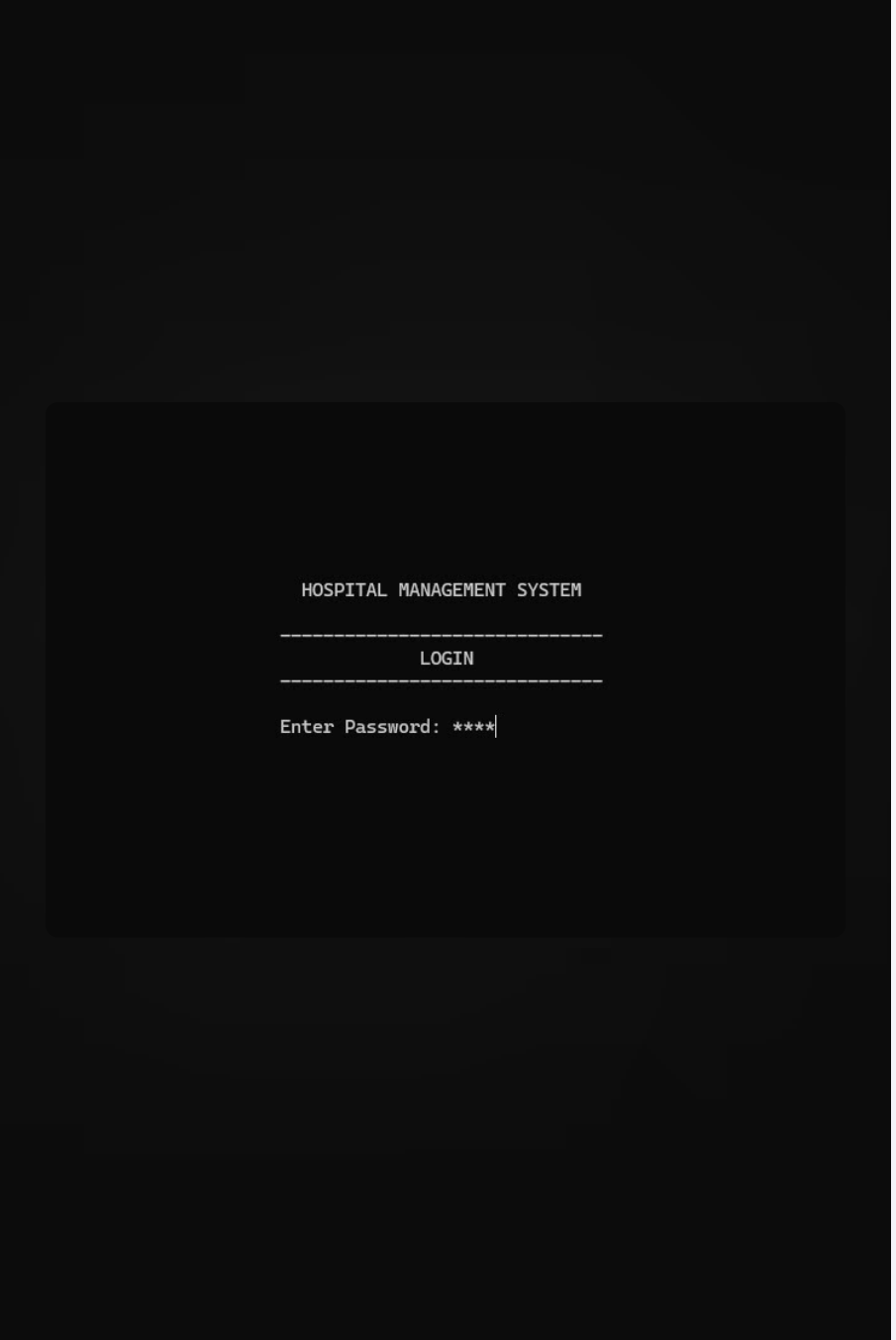
Abdul Rafay

Department of Computer Science

About the System

The Hospital Management System is designed to manage patient records and diagnosis information efficiently. It features a secure login system, allowing authorized users to add new patient records, update diagnoses, and view comprehensive patient histories. This system simplifies data management and enhances overall efficiency.

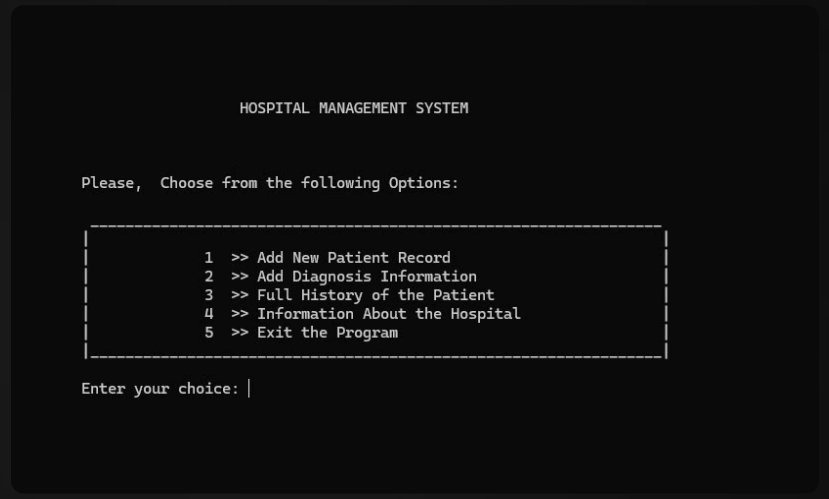




Key Features and Functionality

Secure Login

Ensure data security and confidentiality through a robust user authentication system, limiting access to authorized personnel.

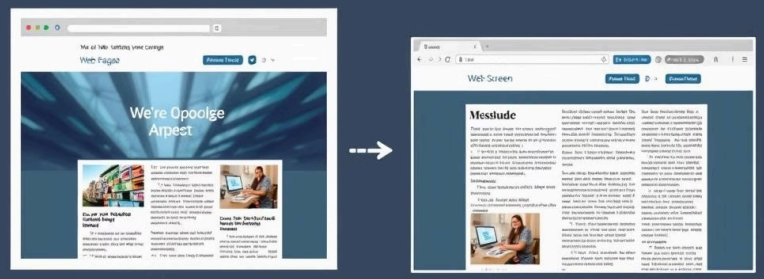


Diagnosis Information

- Users can update a patient’s diagnosis by selecting their file and providing:
- Symptoms, diagnosis, and prescribed medicines.
- Admission status and ward type (if applicable).

Patient Records

- Includes full name, age, gender, contact information, address, and a unique patient ID for easy identification.
- **Medical History**: Maintains a detailed record of past diagnoses, treatments, surgeries, and chronic conditions for better continuity of care.
- **Medications**: Tracks current and past prescriptions, including dosage and frequency, ensuring proper treatment and avoiding drug interactions.



Important Notes

Fullscreen Mode

The system is optimized for full-screen view. The content may display incorrectly or be difficult to read on smaller screens.

C++ Code

Explaination

- The welcome screen is designed using repeated @ symbols for the border. It ensures that the output appears visually neat and aligned.
- The code displays an attractive and user-friendly welcome screen, setting a professional tone for the **Hospital Management System** program. The emphasis on formatting (using @ and spacing) ensures the output looks visually aligned, especially when run in fullscreen mode

```
#include<iostream>-proj
#include<fstream>
#include<cstdlib>
#include<conio.h>
#include<time.h>
#include<iomanip>

using namespace std;
int main()

//NOTE: RUN THE PROGRAM IN FULL SCREEN ONLY

{
char fname[20];
time_t rawtime;
struct tm * timeinfo;

time ( &rawtime );
timeinfo = localtime ( &rawtime );

//printing the welcome note
re:
cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";
cout<<"\t\t\t\t\t@@ ______________________________________________________________________ @@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                            WELCOME TO                                  |@@\n";
```

```
struct tm * timeinfo;

time ( &rawtime );
timeinfo = localtime ( &rawtime );

//printing the welcome note
re:
cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n";
cout<<"\t\t\t\t\t@@ ______________________________________________________________________ @@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                            WELCOME TO                                  |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|                      HOSPITAL MANAGEMENT SYSTEM                        |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|                                                                        |@@\n";
cout<<"\t\t\t\t\t@@|________________________________________________________________________|@@\n";
cout<<"\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n\n\n\n\t\t\t\t\t";
system("pause");
system("cls");
```

Login :

- The program displays a login screen with a header ("HOSPITAL MANAGEMENT SYSTEM") and a prompt for entering a password (`Enter Password:`).
- This is done using `cout`, which outputs text to the console, formatted with `\n` (newlines) and `\t` (tabs) for spacin

```
int login(){
   string pass ="";
   char ch;
   cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t  HOSPITAL MANAGEMENT SYSTEM \n\n";
   cout<<"\t\t\t\t\t\t\t\t\t----------------------------";
   cout<<"\n\t\t\t\t\t\t\t\t\t     LOGIN \n";
   cout<<"\t\t\t\t\t\t\t\t\t----------------------------\n\n";
   cout << "\t\t\t\t\t\t\t\tEnter Password: ";
   ch = _getch();
   while(ch != 13){  //character 13 is enter
      pass.push_back(ch);
      cout << '*';
      ch = _getch();
   }
   if(pass == "pass"){
      cout << "\n\n\t\t\t\t\t\t\t\tAccess Granted! \n";
      system("PAUSE");
      system ("CLS");
   }else{
      cout << "\n\n\t\t\t\t\t\t\t\tAccess Aborted...\n\t\t\t\t\t\t\t\tPlease Try Again\n\n";
      system("PAUSE");
      system("CLS");
      login();
   }
}
```

Menu :

The program begins by displaying the **title** of the system, properly formatted with newlines and tab spaces for alignment.

- The program provides **five choices**:

1. Add a new patient record.
2. Add diagnosis information.
3. View the full history of a patient.
4. Get information about the hospital.
5. Exit the program.

```
system("cls");



int i;
int login();
login();
//giving option to the user for their choice
b:
cout<<"\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t\t\t\t\t  HOSPITAL MANAGEMENT SYSTEM \n\n";
cout<<"\n\n\t\t\t\t\tPlease,  Choose from the following Options: \n\n";
cout<<"\t\t\t\t\t\t _____________________________________________________ \n";
cout<<"\t\t\t\t\t\t|                                                     |\n";
cout<<"\t\t\t\t\t\t|             1  >> Add New Patient Record            |\n";
cout<<"\t\t\t\t\t\t|             2  >> Add Diagnosis Information         |\n";
cout<<"\t\t\t\t\t\t|             3  >> Full History of the Patient       |\n";
cout<<"\t\t\t\t\t\t|             4  >> Information About the Hospital    |\n";
cout<<"\t\t\t\t\t\t|             5  >> Exit the Program                  |\n";
cout<<"\t\t\t\t\t\t|_____________________________________________________|\n\n";
a:cout<<"\t\t\t\t\t\tEnter your choice: ";cin>>i;
if(i>5||i<1){cout<<"\n\n\t\t\t\t\t\tInvalid Choice\n";cout<<"\t\t\t\t\t\tTry again.....\n\n";goto a;} //if inputed choice is other than given choice
```

Patient information:

- A structure `patient_info` is defined to store patient details.
- **Fields**:
- `name`: Stores the patient's name (max 20 characters).
- `address`: Stores the patient's address (max 100 characters).
- `contact`: Stores the patient's contact number (max 10 digits).
- `age`: Stores the patient's age (max 5 characters).
- `sex`: Stores the patient's gender (max 8 characters, e.g., Male/Female).
- `blood_gp`: Stores the blood group (max 5 characters, e.g., O+, A-).
- `disease_past`: Stores information on any past major diseases (max 50 characters).
- `id`: Stores the patient ID (max 15 characters).

```
            char name[20];
            char address[100];
            char contact[10];
            char age[5];
            char sex[8];
            char blood_gp[5];
            char disease_past[50];
            char id[15];
        };

patient_info ak;
cout<<"\n*************************************************************************\n";pat_file<<"\n*************************************************************************
cout<<"\nName : ";pat_file<<"Name : ";gets(ak.name);pat_file<<ak.name<<"\n";
cout<<"\nAddress : ";pat_file<<"Address : ";gets(ak.address);pat_file<<ak.address<<"\n";
cout<<"\nContact Number : ";pat_file<<"Contact Number : ";gets(ak.contact);pat_file<<ak.contact<<"\n";
cout<<"\nAge : ";pat_file<<"Age : ";gets(ak.age);pat_file<<ak.age<<"\n";
cout<<"\nSex : ";pat_file<<"Sex : ";gets(ak.sex);pat_file<<ak.sex<<"\n";
cout<<"\nBlood Group : ";pat_file<<"Blood Group : ";gets(ak.blood_gp);pat_file<<ak.blood_gp<<"\n";
cout<<"\nAny Major disease suffered earlier : ";pat_file<<"Any Major disease suffered earlier : ";gets(ak.disease_past);pat_file<<ak.disease_past<<"\n";
cout<<"\nPatient ID : ";pat_file<<"Patient ID : ";gets(ak.id);pat_file<<ak.id<<"\n";
cout<<"\n*************************************************************************\n";pat_file<<"\n*************************************************************************
cout<<"\nInformation Saved Successfully\n";
}
use");
s");
```

Conclusion and Future Enhancements

The **Hospital Management System** in C++ is a simple project for managing patient records and diagnosis details. It features secure login, patient registration, diagnosis updates, and history viewing using file handling. While deletion requires manual file removal, the project is a great educational tool to learn C++ concepts like file operations and user interaction. Despite its limitations, it provides an efficient and straightforward approach to hospital record management.

Thank You

the end

Department of Computer Science