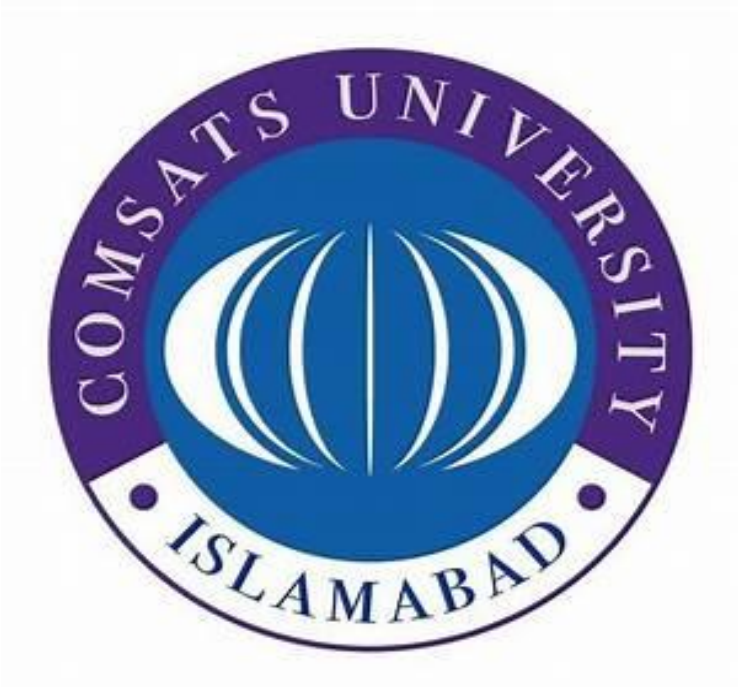
DSA # SEMISTER PROJECT

Hospital Management System

SUBMITTED BY: *Aalliyan Waheed Alvi.* **(No Group)**

REG NO # *SP20-BCS-002.*

SUBMITTED TO: *Dr. Inayat Ur Rehman .*



COMSATS UNIVERSITY ISLAMABAD

***Topic**: Hospital Management System*

I have choosed the topic hospital management system as my semester project of Data Structures and Algorithm, This project is made by one person only i.e me (~Alliyan waheed Alvi). I tried to cover the all CLO's topics in my project such as Linked list, searching, sorting, queue, Binary Search Tree and its all functions, and graph i.e (Dijkstra's Algorithm). Most of the project is made from CLO-4 topics. Whole project is mainly based on Binary Search Tree and the Dijkstra's Algorithm.

Let's discuss all the methods in the code and their working.

Four Basic Screens:

There are four basic screen in my project i.e:

1) Register and login screen:

At this stage user will register his account and then login to use this project.

```
cout<<"\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@@|                           *** WELCOME TO ***                         |@@\n";
cout<<"\t\t\t\t\t@@|                   *** HOSPITAL MANAGEMENT SYSTEM ***                 |@@\n";
cout<<"\t\t\t\t\t@@|                                                             |@@\n";
cout<<"\t\t\t\t\t@@|                       ### Register Your Account ###                  |@@\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@@|   Please Register Your Account, then Login Below...                  |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|     Press Any Key to continue...                                     |@@\n";
cout<<"\t\t\t\t\t@@|                                                                      |@@\n";
cout<<"\t\t\t\t\t@@|                                    ~Code By Alliyan Waheed Alvi     |@@\n";
cout<<"\t\t\t\t\t@@|______________________________________________________________________|@@\n";
cout<<"\t\t\t\t\t@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@\n\n\n\n";
```

2) Dashboard:

```
    cout<<"\n\n\t\t\t\t\t\t\t\t     *###* HOSPITAL MANAGEMENT SYSTEM *###*";
    cout<<"\n\t\t\t\t\t\t\t\t               *** DASHBOARD ***\n";

cout<<"\n\t\t\t\t\t\tPlease,  Choose from the following Options: \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  >> Enter into Patients Database               |\n";
cout<<"\t\t\t\t\t\t|            2  >> Enter into Doctors Database                |\n";
cout<<"\t\t\t\t\t\t|            3  >> Information About Hospital                 |\n";
cout<<"\t\t\t\t\t\t|            0  >> Exit the Program                          |\n";
cout<<"\t\t\t\t\t\t|_____________________________________________________________|\n\n";
```

3) Patient Database Screen:

```
cout<<"\n\n\t\t\t\t\t\t\t\t     *###* HOSPITAL MANAGEMENT SYSTEM *###*\n";
cout<<      "\t\t\t\t\t\t\t\t   *** (Patients Database) ***\n";

cout<<"\n\t\t\t\t\t\tPlease,  Choose from the following Options: \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  >> List of patients in the hospital         |\n";
cout<<"\t\t\t\t\t\t|              3  >> Search a patient with his ID             |\n";
cout<<"\t\t\t\t\t\t|              4  >> Discharge a patient                      |\n";
cout<<"\t\t\t\t\t\t|              5  >> Total number of Patients in the Hospital |\n";
cout<<"\t\t\t\t\t\t|              6  >> Priority wise appointments of Patients   |\n";
cout<<"\t\t\t\t\t\t|              7  >> Generate Patient Bill                    |\n";
cout<<"\t\t\t\t\t\t|              8  >> Go back to Dashboard                     |\n";
cout<<"\t\t\t\t\t\t|_____________________________________________________________|\n\n";
```

- ### void add_new_patient():

  This function adds new patient in the hospital

- ### void traverse_patients_list(patients* root):

  This Function will traverse the list of all the patients in the list and show the result

- ### int total_num_of_patients(patients* root):

  This function calculates the total number of patients in the hospital

- `void search_specific_patient(int key)`

  This function will search for specific patient by his id number

- `void remove_specific_patient(int value`

  This function will discharge specific patient from the hospital.

- `void enqueue(patients* current)`

  Here I have used priority queue to show the priority of patients at which they should be treated according to their diseases.

- `void dlisplay_patients_priority()`

  This function displays the list of priorities of patients.

- `void patients_bill()`

  This function generates the bill of patients according to their treatement , pharmacy charges and doctor fee.

Dijkstra's Algorithm:

We have no Covid-19 ward in our hospital for this reason we reffer Covid-19 patients to our other nearest Hospitals, we have 3 hospitals in our city that has working corona wards, So with the help of Dijkstra's Algorithm we will find the nearest corona hospital from our hospital and will refer our corona patient to that hospital.

```
1156
1157   // ************ Applying Dijkstr's Algorithm ************
1158
1159
1160   /*                  (Graph)
1161           (Map)
1162                      C--------E (Corona Hospital)
1163                     / \      /  \--1
1164                20 /   \ 20/      |
1165                  /     \ /       F (Corona Hospital)
1166   (Our Hospital) A       \       /
1167                   \  10 / \     /
1168                10 \   /   \   /2
1169                     \ /     \ /
1170                      B-------D (Corona Hospital)
1171                        50
1172
1173   From figure:
1174   >> Hospital A is our Hospital i.e Source point
1175   >> Hospital E F and D are Corona Hospitals
1176   >> We have to find nearest Corona hospital
1177      for Covid-19 patient
1178
1179   */
```

Functions used in Dijkstras Algorithm:

- `int minimumDist(int dist[], bool Tset[]`
- `void Dijkstra(int graph[6][6],int src)`
- `void corona_hospital()`

Map:

```
cout<<"\n                      (Graph)                          ";
cout<<"\n         (Map)                                         ";
cout<<"\n                    C--------E (Corona Hospital)       ";
cout<<"\n                   / \\     / \\ 1                      ";
cout<<"\n               20 /   \\20 /   \\                      ";
cout<<"\n                 /     \\ /     F (Corona Hospital)   ";
cout<<"\n (Our Hospital) A       \\      /                      ";
cout<<"\n                 \\  10 / \\    /                      ";
cout<<"\n               10 \\   /   \\  /2                     ";
cout<<"\n                   \\ /     \\/                       ";
cout<<"\n                    B-------D (Corona Hospital)       ";
cout<<"\n                        50                           ";
```

4) Doctors Database Screen:

```
    cout<<"\n\n\t\t\t\t\t\t\t\t    *###* HOSPITAL MANAGEMENT SYSTEM *###*";
    cout<<"\n\t\t\t\t\t\t\t\t          *** Doctors Database ***\n";

cout<<"\n\t\t\t\t\t\t Please,  Choose from the following Options: \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 Doctor's Record              |\n";
cout<<"\t\t\t\t\t\t|          2  >> List of Doctors in the hospital      |\n";
cout<<"\t\t\t\t\t\t|          3  >> Search a Doctor with his ID          |\n";
cout<<"\t\t\t\t\t\t|          4  >> Remove a doctor's record from the list|\n";
cout<<"\t\t\t\t\t\t|          5  >> Total number of Docters in the Hospital|\n";
cout<<"\t\t\t\t\t\t|          6  >> Go back to Dashboard                 |\n";
cout<<"\t\t\t\t\t\t|_____________________________________________________|\n\n";
```

For doctors, I have used the same BST structure , I am just showing the names of functions used in Doctors Database to concise the report.

- `void add_new_doctor()`

- `void traverse_doctors_list(doctors* d_root)`

- `int total_num_of_doctors(doctors* d_root)`

```
➢ void search_specific_doctor(int key)
```

```
➢ void remove_specific_doctor(int value)
```

****