**+ Hospital Management System +**

**Table of Contents**

**Acknowledgement**

1. **Introduction**
2. **Abstract**
3. **C Programming Concepts**

**(i)** Preprocessor Directives

**(ii)**  Pointer’s Concepts

**(iii)** Structures (user-defined)

**(iv)** Error Handling

**(v)** File Handling

**4. Coding**

**5. References**

**Acknowledgement**

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got this all along the completion of our project work. Whatever we have done is only due to such guidance and assistance and we would not forget to thank them.

We respect and thank our respected H.O.D Dr. Ishan Bhardwaj , for giving me this opportunity to do the mini project work in our domain **HOSPITAL MANAGEMENT SYSTEM** for providing us all the support and guidance which made me complete the project on time. We are extremely grateful to her for providing such a support and guidance through her busy schedule managing the Department affairs.

The successfully completion of our project would not have possible without their dedicated Support.

**INTRODUCTION**

Hospital Management System covering all the data storation of patients in hospital. This Project records patient vitals and progress notes through patient management system module all the way to an efficient patient billing system.

This healthcare software solution covers healthcare and works as complete healthcare management system for Healthcare units and all stake holders including Patient with , Doctors , Nurses effectively creating best solution.

**Abstract**

We intend to use VS CODE editor as the tools to create this Project in C Programming Language. Our webpage is intended to have drop down menus containing various sections. Menu section to allow our clients to connect with operation. We also intend to provide the user with an option to tell us which information they want to make private such that we can't make it publicly available. Also our webpage gives the user the facility to see detailed information about the Data stored via user in File Handling.

**C Programming Concepts**

**Preprocessor Directives -** The **C Preprocessor** is not a part of the compiler, but is a separate step in the compilation process. In simple terms, a C Preprocessor is just a text substitution tool and it instructs the compiler to do required pre-processing before the actual compilation. We'll refer to the C Preprocessor as CPP.

All preprocessor commands begin with a hash symbol (#). It must be the first nonblank character, and for readability, a preprocessor directive should begin in the first column. The following section lists down all the important preprocessor directives –

**#define,**

**#include**

**Pointer’s Concepts -** The pointer in C language is **a variable which stores the address of another variable**. This variable can be of type int, char, array, function, or any other pointer. The size of the pointer depends on the architecture. However, in 32-bit architecture the size of a pointer is 2 byte.

1. **int** n = 10;
2. **int**\* p = &n;
3. **Structures (user-defined) - A structure is a key word that create user defined data type in C/C++**. A structure creates a data type that can be used to group items of possibly different types into a single type. How to create a structure? 'struct' keyword is used to create a structure.
4. **struct** structure\_name
5. {
6. data\_type member1;
7. data\_type member2;
8. .
9. .
10. data\_type member N ;
11. };

**File Handling -** In programming, we may require some specific input data to be generated several numbers of times. Sometimes, it is not enough to only display the data on the console. The data to be displayed may be very large, and only a limited amount of data can be displayed on the console, and since the memory is volatile, it is impossible to recover the programmatically generated data again and again. However, if we need to do so, we may store it onto the local file system which is volatile and can be accessed every time. Here, comes the need of file handling in C.

**fopen() , fprintf()**

**fscanf() ,**

**etc.**