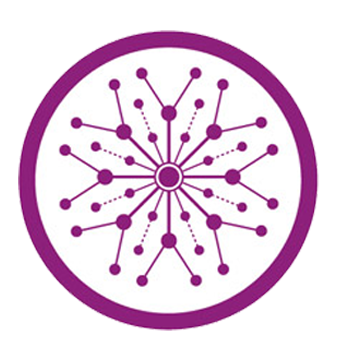
**Data Structure and Algorithm (DSA)**

**Final Project**

**Hospital Management System**

****

**Name: Zunaira Javaid**

**Roll-No: BSEM-F19-090**

**Section: 3A**

**Submitted to Ma’am Asma**

**HOSPITAL MANAGEMENT SYSTEM**

**Description:**

Hospital Management System provides the beneﬁts of streamlined operations, enhanced administration, control, superior patient care, strict cost control and improved proﬁtability.

Hospital Management System is powerful, ﬂexible, and easy to use and is designed and developed to deliver real conceivable beneﬁts to hospitals. More importantly it is backed by reliable and dependable support.

The project ‘Hospital Management System’ is based on the file handling, DSA C++ and object oriented programming. We keep the records in excel file for which we are using .csv extension. Hospital Management System is custom built to meet the speciﬁc requirement of the mid and large size hospitals.

It covers all the required modules right from Patient Registration, patient record details, patient appointment details, Doctors details, record modiﬁcation, discharge details etc.

**Project objectives:**

Keeping record of all the activities in hospital on paper is very inconvenient and error prone. It is also very inefficient and a time-consuming process. Recording and maintaining all records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain records on paper. Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “Hospital management system”.

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides reliable storage and backup facilities.

**Why DSA?**

DSA C++, Really fast as compared to other language. We use singly linked list in our project. The Time complexity of the Singly Linked List is O (n) and same for Space complexity.

**Operations:**

1. Create Record
2. Check Record Existing or not
3. Append Record
4. Prepend Record
5. Insert Record
6. Delete Record
7. Search Record
8. Create Node
9. Display Record
10. Update Record

**Concepts:**

* Switch
* Conditions
* Classes.
* Constructors.
* Parameterized constructor
* Unconditional transfer of control
* Singly linked list
* File handling.