**Array for patients; linked list for appointments.**

**Project report submitted in partial fulfillment of the Requirements for the Award of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

|  |  |
| --- | --- |
| **P.CHARU HASINI** | **24KB1A05FE** |
| **SHAIK FARSHIYA** | **24KB1A05HG** |
| **SHAIK SAMERA** | **24KB1A05JJ** |
| **VENKATA SRAVYA** | **24KB1A05JX** |

**Under the Guidance of**

**SIVANRAJ SIR**

**SUNEETHA MAM**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**NBKRIST, VIDYANAGAR, TIRUPATI.**

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



###### CERTIFICATE

This is to certify that the project report entitled YOUR PROJECT TITLE being submitted by

|  |  |
| --- | --- |
| **P.CHARU HASINI** | **24KB1A05FE** |
| **SHAIK FARSHIYA** | **24KB1A05HG** |
| **SHAIK SAMERA** | **24KB1A05JJ** |
| **VENKATA SRAVYA** | **24KB1A05JX** |

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the **NBKRIST SCIENCE AND TECHNOLOGY, VIDYANAGAR,TIRUPATI DIST.** is a record of bondified work carried out under my guidance and supervision.

|  |  |
| --- | --- |
| **SIVANRAJ SIR,**  **SUNEETHA MAM.**  **Designation** | **Dr. HOD RAVEENDRA REDDY**  **M. Tech , Ph. D**  **Head of the Department** |

**DECLARATION**

I hereby declare that the dissertation entitled **Array for patients; linked list for appointments** submitted for the B.Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associate ship, fellowship or any other similar titles.

Place:

|  |  |
| --- | --- |
| **NELLORE** | **P.CHARU HASINI** |
| **NELLORE**  **NELLORE**  **NELLORE** | **SHAIK FARSHIYA** |
|  | **SHAIK SAMERA** |
|  | **VENKATA SRAVYA** |

Date: 07-05-2025 24KB1A05FE

07-05-2025 24KB1A05HG

07-05-2025 24KB1A05JJ

07-05-2025 24KB1A05JX

**Acknowledgement**

We acknowledge the implementation of an efficient data management system that leverages an array structure for storing patient information and a linked list for handling appointment scheduling. The use of an array allows for quick access to patient records due to its indexed nature, while the linked list provides flexibility in managing dynamic appointment entries, including insertions and deletions. This combination enhances the overall performance and scalability of the system, ensuring smooth operation and ease of maintenance.

**INTRODUCTION**

In healthcare management systems, efficient data organization is crucial. Arrays are well-suited for storing patient records due to their fixed size and fast access time, allowing for quick retrieval and indexing of patient information. On the other hand, appointments are better managed using linked lists, as they provide flexibility in dynamic insertion and deletion—perfect for handling frequent scheduling changes without the need to shift data.

### ****Abstract****

This project presents a simplified patient management system designed to efficiently handle basic hospital operations such as storing patient data and managing appointment schedules. To achieve optimal performance and organization, two distinct data structures have been utilized: an **array** for managing patient records and a **linked list** for handling appointment bookings. The array enables fast and direct access to patient details through index-based retrieval, ideal for fixed-size datasets. On the other hand, the linked list provides dynamic memory allocation, allowing appointments to be added or removed easily without the need for data shifting. This structure is particularly suited to the variable nature of scheduling systems. Together, these data structures provide a functional and scalable foundation for a lightweight, responsive management system.

| **Component** | **Data**  **Structure**  **Used** | **Purpose** | **Advantages** |
| --- | --- | --- | --- |
| Patients | Array | Store and manage  records | - Fast access using index- Easy to implement- Suitable for fixed-size data |
| Appointments | Linked List | Manage dynamic  appointment entries  (add, remove, modify) | - Efficient insertions and deletions- Dynamic size- Memory-efficient |

**Comparison Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | **Aspect** | **Patients (Array)** | **Appointments (Linked List)** | | --- | --- | --- | | Data Structure Type | Static (Fixed size) | Dynamic (Expandable at runtime) | | Access Speed | Fast (Direct indexing) | Slower (Sequential access) | | Insertion/Deletion | Costly (May require shifting elements) | Efficient (Just update pointers) | | Memory Allocation | Predefined (contiguous block) | Allocated as needed (non-contiguous memory) | | Use Case in System | Store patient profiles | Schedule and manage daily appointments | |  |  |  |
| **ARRAY OF PATIENTS:** |  |  |  |
|  |  |  |  |
| * Each slot holds a patient's data (e.g., ID, name, age, etc.) * Accessed directly using index: patients[2] gives Patient 2 |  |  |  |
|  |  |  |  |

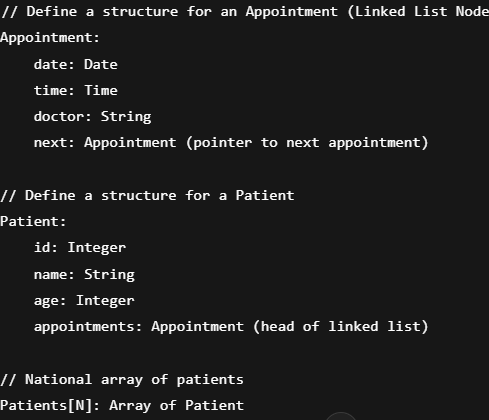
**Linked List for Appointments:**

****

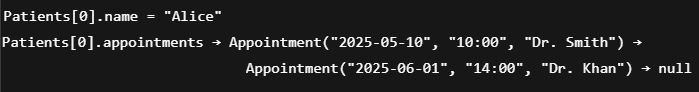
 Each node contains appointment details and a pointer to the next

 Easily add or remove appointments without shifting elements

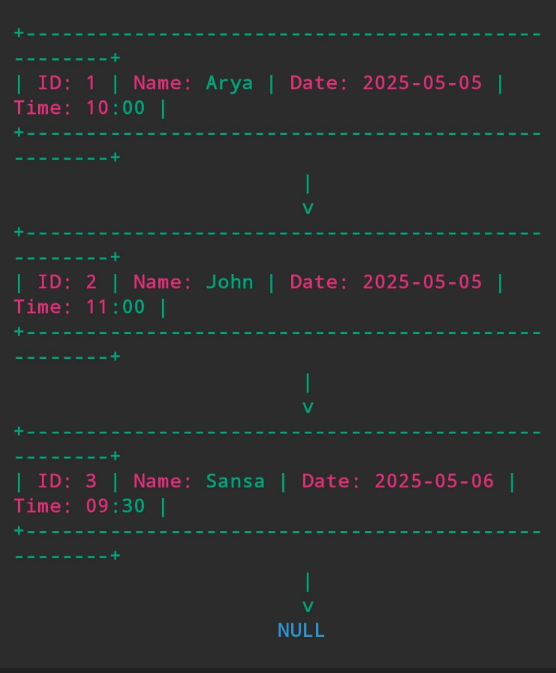
**Notation**

****

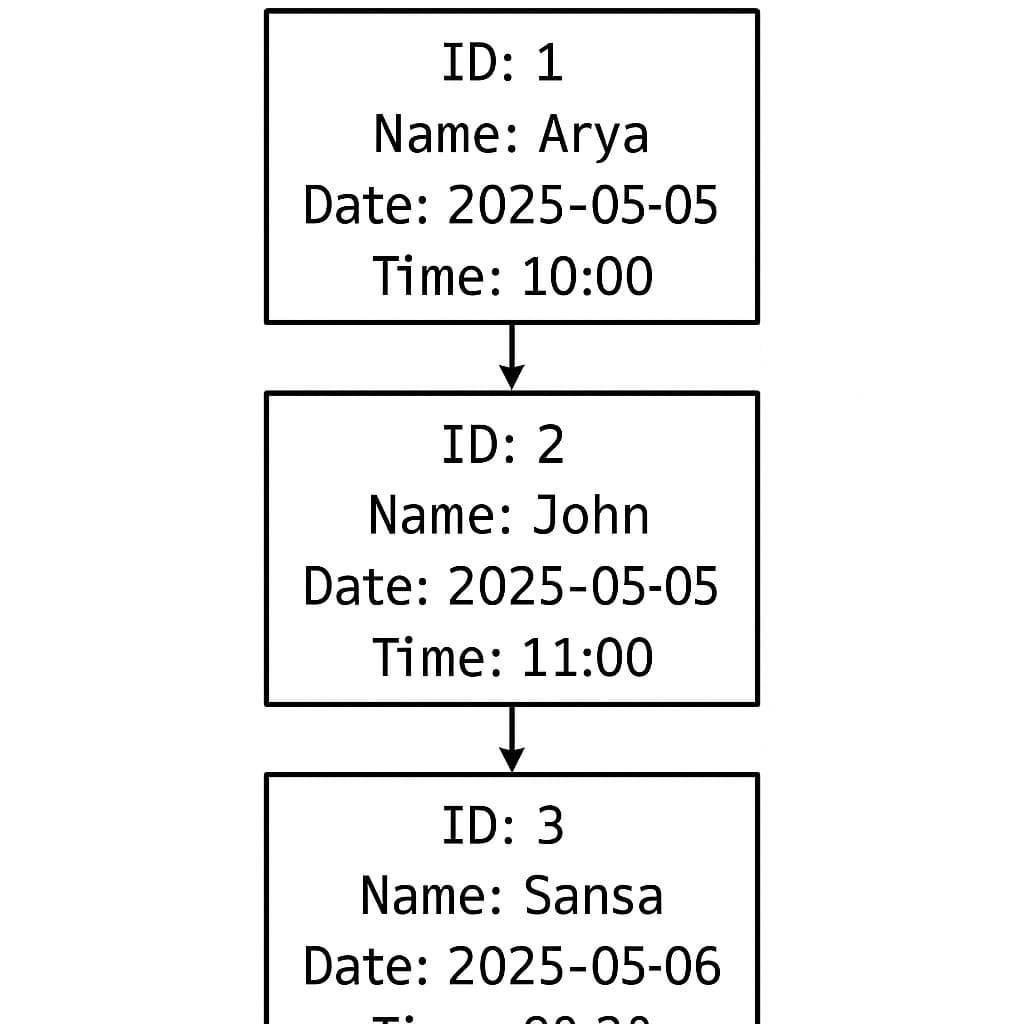
**Example Usage**

****

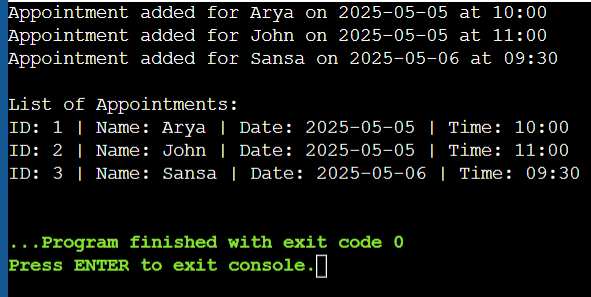
**Figures**

****

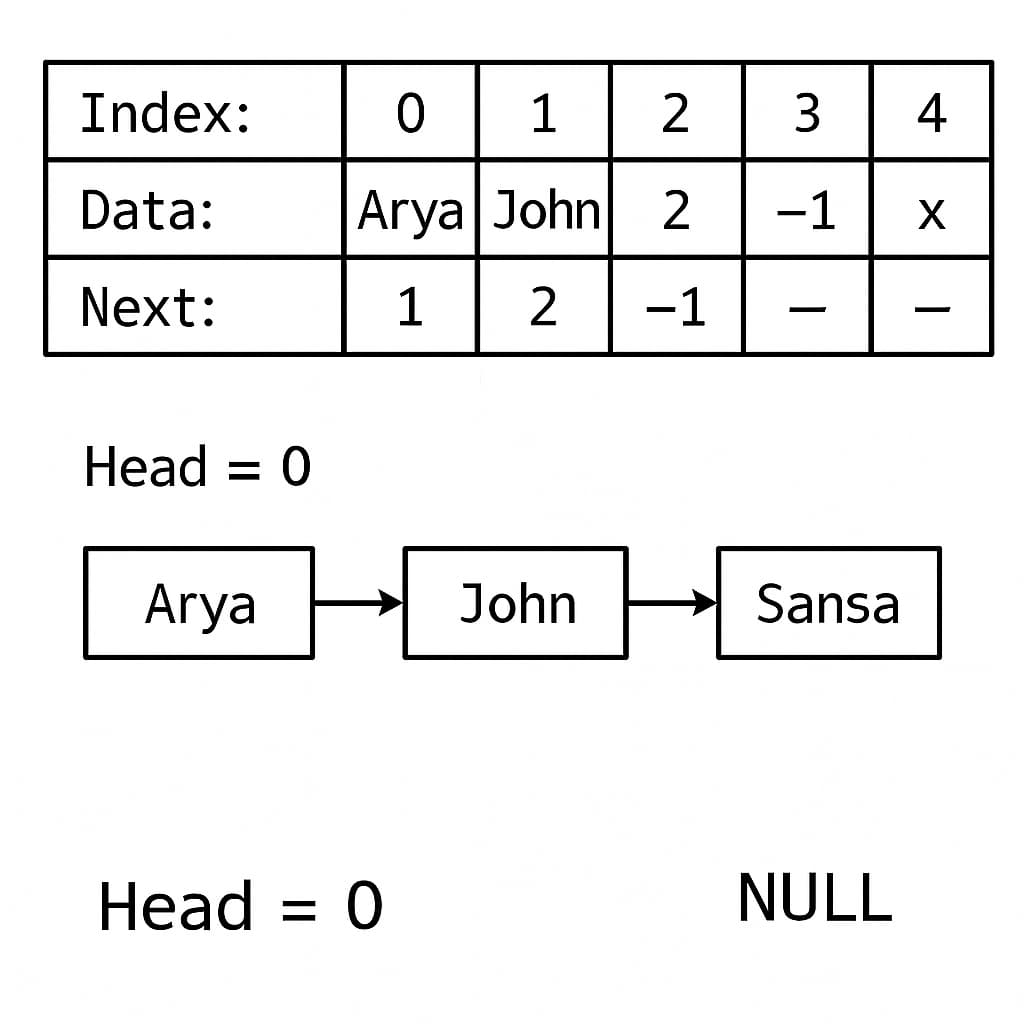
**Each box represents a node in the linked list. The arrow (v) shows the pointer to the next node. The list starts with Arya’s appointment and ends at NULL, indicating the end of the list.**

****

**OUTPUT**

****

**GRAPHS:**



### References

Books

1. "Let Us C" by Yashavant Kanetkar

Great beginner-level book to understand C and linked list concepts.

2. "Data Structures Using C" by Reema Thareja

Focused on implementing data structures like linked lists, stacks, queues in C.

3. "Data Structures and Program Design in C" by Robert L. Kruse

A more in-depth look at linked lists and pointers.

**Online Tutorials**

1. Geeks for Geeks – Linked List in C

https://www.geeksforgeeks.org/data-structures/linked-list/

Covers single, double, circular linked lists with examples.

2. Tutorials Point – C Linked List

https://www.tutorialspoint.com/data\_structures\_algorithms/linked\_list\_program\_in\_c.htm

Basic syntax and patient scheduling adaptations are easy.

3. Programiz – Linked List in C

https://www.programiz.com/dsa/linked-list