**Project Overview**

**Objective:** Implement a **Doubly Linked List in C** to manage and display a sequence of roll numbers, supporting insertion and deletion at specific positions.  
 **Team Collaboration:** Utilizes **GitHub** for version control, including branches, commits, and pull requests.  
 **Visualization:** Incorporates **Figma diagrams** to illustrate doubly linked list operations and structure.  
 **Documentation:** Includes a **README file** detailing the project's purpose, setup instructions, and sample outputs.

**Key Features**

**Doubly Linked List Implementation**

The program supports the following operations:

* **Insertion at Position**: Insert a new roll number at a given position in the list.
* **Deletion at Position**: Delete a roll number from a given position in the list.
* **Display**: Traverse and display the roll numbers in both forward direction (head → tail).

**Sample Output**

Example program execution:

Insertion at 1st Position: 10 <-> NULL

Insertion at 2nd Position: 10 <-> 20 <-> NULL

Insertion at 3rd Position: 10 <-> 20 <-> 30 <-> NULL

Deletion at 2nd Position: 10 <-> 30 <-> NULL

Deletion at 1st Position: 30 <-> NULL

**GitHub Collaboration**

* **Branches**: Each teammate created a branch named after their roll number to contribute features.
* **Commits**: Regular commits document development progress and code changes.
* **Pull Requests**: Used for reviewing and merging contributions into the main branch, ensuring code quality and consistency.
* **Merge Conflict Resolution**: Handled collaboratively when multiple contributors modified related parts of the code.

**Visualization**

* **Figma Diagrams**: Used to visually represent:  
    
  + Insertion at a given position
  + Deletion at a given position
  + Traversal of the doubly linked list
* These diagrams help explain pointer updates (prev and next) and overall structure.