**Project Proposal: Car Parking Systems**

Submitted by   
 Rehan Ali (su92-bscsm-f23-692)

**1. Objective**

Develop an efficient car parking system using **Doubly Linked List**, **Binary Search Tree**, and **Queue** to manage car records, enable fast searches, and handle a waiting queue.

**2. Features**

1. **Insert Record**: Add a car to the parking lot.
2. **Search Record**: Find car details using the car number.
3. **Update Record**: Modify existing car details.
4. **Delete All Records**: Clear all parking records.
5. **Show All Records**: Display all car records.
6. **Add to Waiting Queue**: Manage overflow cars.
7. **Show Waiting Queue**: View cars in the queue.
8. **Exit**: Close the application.

**3. Data Structures**

1. **Doubly Linked List**: Store and manage car records.
2. **Binary Search Tree (BST)**: Enable fast searches.
3. **Queue**: Handle waiting cars efficiently.

**4. Algorithm Highlights**

* **Insert**: Add car details to the list and BST.
* **Search**: Use BST for quick retrieval.
* **Update**: Traverse the list to modify records.
* **Delete**: Clear the list, BST, and queue.

**5. Tools**

* **Language**: C++
* **IDE**: Visual Studio Code, Dev C++, or Code::Blocks

**6. Timeline**

| **Task** | **Time** |
| --- | --- |
| Analysis & Design | 2 Days |
| Implementation | 7 Days |
| Testing | 3 Days |
| Documentation | 2 Days |

**7. Expected Outcomes**

Efficient car management with fast searches, organized storage, and seamless queue handling.