***SMESTER PROJECT***

***RAO ALI AHMAD (223710)***

***M SAJJAD (223631)***

***PROJECT NAME***



***SUBMITTED TO:***

***PROF HAFIZ MUHAMMAD ANAS ALI***

### ****GreenShare Project Report****

### ****1. Project Overview****

* **Objective**: The project implements a mobile energy-sharing application called GreenShare, utilizing a console-based approach to manage posts and payments.
* **Features**:
  1. Allows owners to post details about shared solar energy.
  2. Clients can search, view, and make payments for posts.
  3. Admin (owner) and client login system.
  4. Payment processing simulation.

### ****2. Code Structure****

**Main Components**:

* + **Node Class**: Represents a single post with attributes like name, address, description, quantity, price, and payment status.
  + **Doubly Class**: Implements a doubly linked list to store and manage posts.
  + **Menu Functionality**: Handles owner and client interactions.

**Key Functions:**

* 1. add: Adds a new post to the doubly linked list.
  2. print: Displays all posts.
  3. searchaddress: Searches posts by address.
  4. searchandselete: Deletes a post by name.
  5. showdescription: Displays details of a post by name.
  6. makePayment: Simulates payment processing for a post.

### ****3. Code Flow****

**Login System:**

* + Owners and clients log in with predefined credentials.
  + Email and password are verified for access.

**Owner Features:**

* + **Add Post**: Add details of solar energy availability (quantity, price, etc.).
  + **Delete Post**: Search and delete posts by name.
  + **View Posts**: View all available posts with total payment details.

**Client Features**:

* + **Search Post**: Search posts by address.
  + **View Posts**: Display all posts.
  + **Make Payment**: Pay for a selected post.

### ****4. Highlights of the Implementation****

* **Data Structure**: Efficiently manages posts using a doubly linked list.
* **Error Handling**: Input validation for integers and non-empty fields.
* **Security**: Masked password entry during login.
* **User Interaction**: Provides a clear and simple menu-driven interface.

### ****5. Challenges and Considerations****

* **Scalability**: The current implementation is suitable for limited data but may require optimization for large datasets.
* **Data Persistence**: Post data and payments are not saved after the program terminates. Future improvements could include file-based or database storage.
* **Dynamic Memory Management**: Manual handling of node deletion is implemented but requires careful debugging for memory leaks.

### ****6. Future Enhancements****

* Add file-based persistence to store posts and payment history.
* Improve UI/UX with graphical interfaces.
* Incorporate advanced user authentication mechanisms.
* Integrate APIs for real-time energy sharing and payment gateways.