CPE 593 Progress Report v2

GitHub: https://github.com/joyson0914/CPE593_finalproject.git

Project Topic: Contact Management System- Arrays, Linked list, Hash Tables

From Project Plan:

1. Add Contact
    - User Input: Determine what contact data is needed (e.G., name, telephone quantity, address, electronic mail).
    - Validation: Implement fundamental validation to ensure the statistics provided are correct (e.G., email layout, cellphone variety format).
    - Data Structure: Decide where and the way this touch statistics might be saved (e.G., in a related list or array).
2. Delete Contact
    - Identify Contact: Define how users will pick the touch to delete (e.G., with the aid of a call or ID).
    - Confirm Deletion: Include an affirmation step to prevent accidental deletions.
    - Update Data Structure: Update your information structure to mirror the elimination of a touch.
3. Update Contact
    - Identify Contact: Determine how customers will discover the touch to replace
    - Update Fields: Allow customers to alter precise fields without overwriting others.
    - Data Synchronization: Ensure the hash desk and connected listing are updated consistently.
4. Search for Contact
    - Search by way of Attribute: Define which attributes can be used for looking (e.G., name, email, cellphone).
    - Search Efficiency: Implement the quest the use of a hash table for brief lookups.
    - User Experience: Decide on person enter format for seek and the output show.
5. List Contacts
    - Display Contacts: Create a way to list all contacts. This could be as easy as printing to the console or more complex like a graphical user interface (GUI).
    - Sort Contacts: Offer sorting options (e.G., by using call, telephone quantity).
    - Pagination: If the list turns into long, recollect including pagination for less complicated navigation.
6. Add UI
    - User Interface Options: Decide whether you will construct a graphical user interface or a command-line interface.

- User Interaction Design: Plan how customers will have interaction with the device to perform exclusive operations.
- Accessibility and Usability: Ensure the UI is user-friendly and accessible.

7. Search Function
   - Search Optimization: Consider approaches to optimize the hunt (e.G., index-primarily based looking, caching).
   - Error Handling: Implement robust blunders dealing with instances wherein the hunt does not go back results.

Rough Project plan:

April 23rd- May 2nd

- Testing major functions
- Optimization
- Final research paper
    - Add details
    - Examples
    - Result
    - Conclusion

May 8th:

- Presentation slides done
- Presentation