CPP104: LinkedList Operations Activity: Patient Management System

Instructions:

Objective:

The objective of this activity is to teach students how to implement and understand linked list operations in Python by applying them to a real-life scenario: managing a healthcare clinic's patient records.

Overview:

In this activity, students will implement a linked list data structure to manage patient records. They will perform operations such as adding new patients, serving patients, and viewing the current patient list. The process for each operation will be documented step-by-step to provide a clear understanding of how linked lists can be applied in real-life situations.

Create documentation for your program including a user manual explaining how to use the system and any technical documentation necessary for maintenance and future development.

Documentation Outline

System Overview:

- Describe the overall purpose and functionality of the patient management system.
- Explain the significance of linked list operations in managing patient records.
- Discuss how users can interact with your system to add, serve, and view patient records.
- Highlight any unique features or capabilities of your system.

Order Management Operations:

- **Add Patient:** Adds a new patient to the linked list.
- **Serve Patient:** Removes the first patient from the linked list and marks them as served.
- **View Current Patient List:** Displays all patients in the linked list.
- **Check if Patients are Pending:** Checks if there are any patients left to serve.
- **Check if Patient List is Full:** Checks if the list has reached its maximum capacity (if a maximum size is defined).

User Manual:

- Provide a step-by-step guide on how to implement and use the patient management system.
- Explain the initial setup and any required Python libraries.
- Detail the process for each patient management operation, including pseudocode and flowcharts if necessary.
- Include code snippets and explanations for key parts of the code.
- Show examples of patient management with initial and final states.

Technical Documentation:

- Describe the data structures used in the implementation of the linked list for patient management.
- Include pseudocode outlining the logic of your patient management operations.
- Provide flowcharts to visualize the steps needed to implement each patient management operation.

Known Issues and Limitations:

- Document any known issues, bugs, or limitations of the patient management implementation.
- Provide recommendations for addressing or mitigating these issues.

Future Enhancements:

- Suggest potential improvements and additional features that could be added to the system in the future.

Appendices:

- Include any additional information or resources relevant to the patient management activity, such as sample input/output data or code snippets.
- If you use AI tools (e.g. ChatGPT) during the development process, include details on how they were utilized.
- List any online resources used, such as YouTube, StackOverflow, or other relevant websites.

Members:

- Include the names of all team members in your project documentation.
- Include photos and descriptions of each team member.
- Describe each member's contributions to the project.

Rubrics:

Criterion	Description	Points
User-friendly Design	Evaluate the intuitiveness, ease of use, and visual appeal of the GUI.	**10**
Functionality	Assess the implementation of linked list methods (Add Patient, Serve Patient, View Patient List).	**25**
Error Handling	Check the robustness of the system in handling invalid inputs and unexpected errors.	**10**
Data Handling	Evaluate the efficiency and accuracy in processing and storing data inputs and outputs.	**10**
Documentation	Assess the clarity, completeness, and helpfulness of user manuals and technical documentation.	**40**
Neat Source Code	Check for readability, organization, and appropriate comments in the source code.	**5**
Total	Sum of all points across the criteria.	**100**