

# Linked List Data Structure

Presentation



---

# Outline

The Problem

Solution Proposal

Components

Next Steps



# The Problem



---

# Problem statement

Our project, 'Linked List Practice Problems,' addresses the challenge students face in mastering the Linked List data structure efficiently.




---

# What users do today

Understanding Linked Lists is fundamental to computer science education, and students often struggle due to the lack of effective practice tools.

Our team believes that providing a platform for targeted practice will enhance learning outcomes and streamline the educational experience for students tackling Linked Lists.

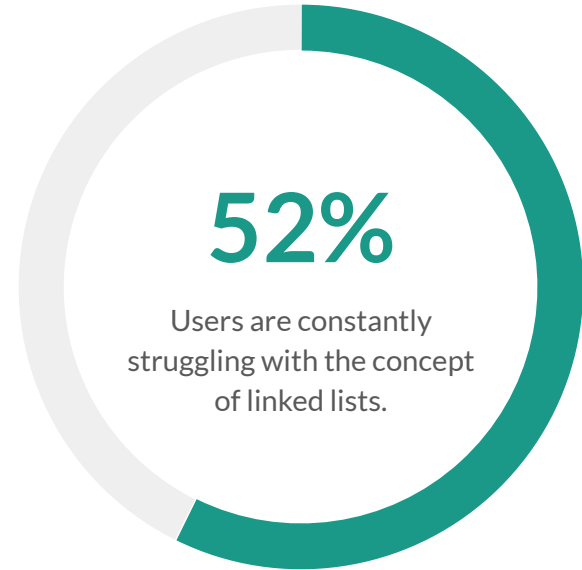


# Supporting information

## O1

-Our primary focus on students stems from recognizing the challenges they face in grasping Linked List concepts. By tailoring the system to their learning needs, we aim to optimize their learning journeys and boost comprehension.

-Our attention to lecturers is driven by the belief that by providing them with tools to customize quizzes and monitor student progress, we can enhance the teaching experience and tailor the material to their specific curriculum



# Supporting information

02

-Our collaboration began when William expressed concerns about students' challenges in mastering LinkedIn. They approached us seeking a solution that could enhance learning experiences.



**William Hill**

Client

The client articulated the need for a platform that not only generates LinkedIn problems but also guides students through the problem-solving process. The goal was to address common misconceptions and provide tailored feedback.



# Use cases / user stories

- Use cases, user stories, notes to set up the wireframes. Such as...
- “As a student, *I want a clear and intuitive user interface*. so that I can easily navigate the system without feeling overwhelmed, even if I am not technologically proficient.”
- “As a student *I want to practice Linked List problems*, so that I can reinforce my understanding of Linked List implementation and improve my problem-solving skills.”
- “As a Lecturer, *I want to customize problem sets for my students*, so that I can tailor the learning experience to align with my curriculum and address specific challenges my students may be facing.”



# Assumptions

- Our project assumes that providing a user-friendly interface with instant feedback will significantly improve students' understanding of Linked Lists
  - The effectiveness of our solution relies on consistent user engagement. We assume that students will find value in the practice problems and feedback, fostering regular usage.
  - We assume that immediate feedback highlighting common mistakes will enhance the learning experience. This assumes that students will use this feedback constructively to improve their understanding.
  - The platform aims to simplify the learning process, but it assumes that users have a basic understanding of using online platforms. If users struggle with basic technology, the learning curve may still pose a challenge.
  - For widespread adoption, we assume that educational institutions may consider integrating or endorsing our platform. This would enhance accessibility and encourage student participation.
-

---

# Solution Proposal



# Solution description

Our solution, "Linked List Practise Problems," provides students with a user-friendly platform to practice and enhance their understanding of Linked List data structures through generated problems, immediate feedback, and a customizable interface for educators..



# Why it's better than existing solutions

Compared to traditional methods of teaching and practicing Linked List concepts, our solution stands out with its interactive and user-centric approach. Unlike static textbooks or one-size-fits-all online resources, our platform tailors problem generation, feedback, and user interface to individual learning needs, ensuring a more personalized and effective learning experience for students.



# Components

Our solution comprises three key components:

1. Problem Generator:

- Function: Lecture creates dynamic Linked List problems based on his project scope.
- Importance: Offers a diverse range of challenges, promoting comprehensive understanding and skill development.

2. Interactive User Interface:

- Function: Enables students to perform Linked List operations (e.g., add, delete, swap nodes) with immediate feedback.
- Importance: Enhances hands-on learning, reinforcing theoretical knowledge through practical application.

3. Feedback Mechanism:

- Function: Provides personalized feedback to students, pinpointing common mistakes and suggesting improvements.
- Importance: Facilitates targeted learning, helping students address specific weaknesses and reinforcing correct practices.

## Welcome to the Linked Lists Visualization

Student

For students

Lecturer

For lectures

## Lecturer Dashboard

Add New Quiz

generate Feedback

Quiz1

Quiz2

test1

Quiz3

Quiz4

Quiz5

Add a new quiz/ test

feedback from  
students

## Lecturer Dashboard

Save Quiz Name

Question Type:

Multiple Choice

Question Mark:

Enter mark e.g 5

Question Text:

Type your question

Options:

Option 1

Add Option

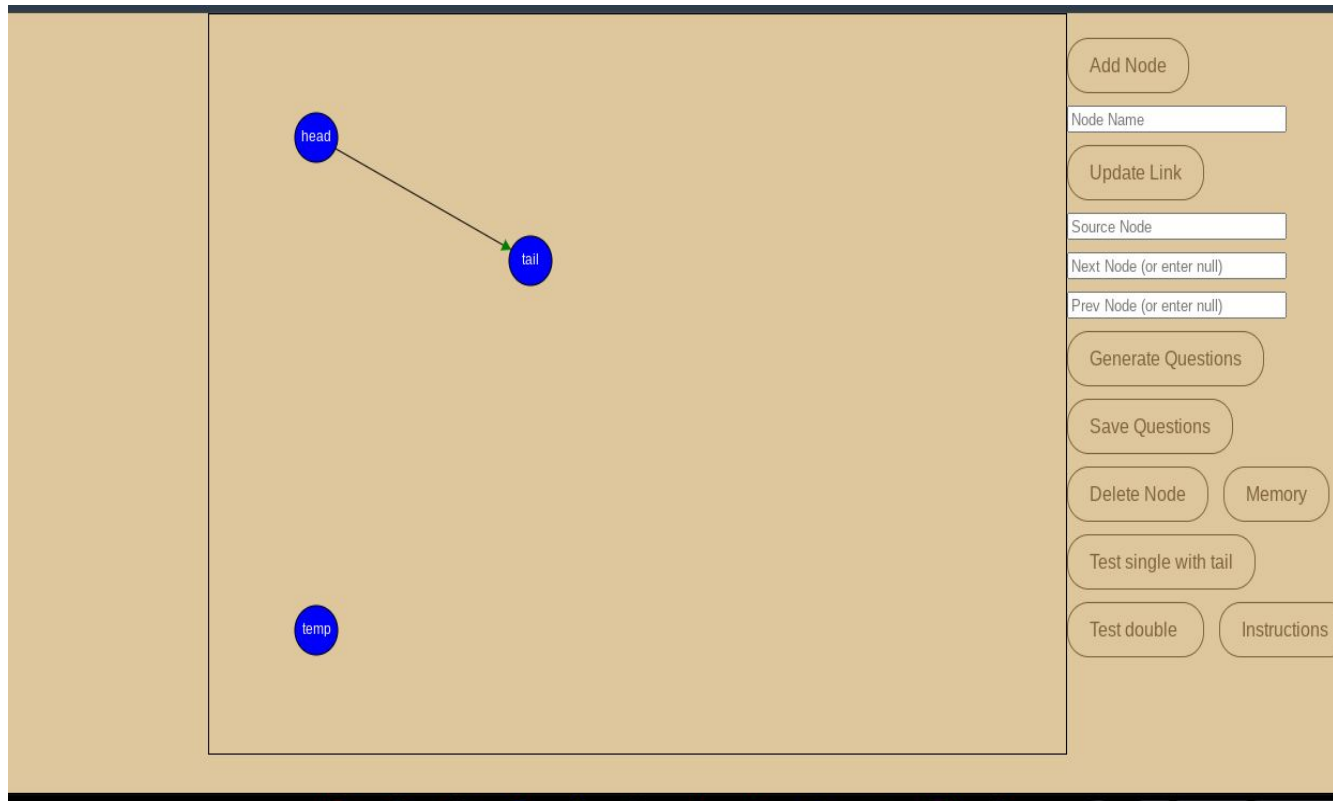
Upload Image (optional):

Choose File

No file chosen

Add Question





Node graph to practise basic operations on linked lists.

### Instructions

Instruction 1: Use "Add Note" to add a new Node.

Instruction 2: Use "Update Link" to create different links.

Instruction 3: Use "Generate Questions" to add new Questions

Instruction 4: Use "Save Questions" To save and update questions

Instruction 5: Use "test Single LinkedList" to test your linkedlist if it matches the tests generated

Instruction 6: Use "test Double" to test your linkedlist if it matches the tests generated for Double Linked list

Instruction 7: Use "Memory" to see which nodes were lost as Memory loss

Instruction 8: Use "Delete Node" to remove or delete nodes from the canvas/ Drag the node to the button

Instruction 9: Use "Memory" to see which nodes were lost as Memory loss

Instruction 10: The Green Arrow shows that is a next pointer and it can be used in both single and double linked list

Instruction 11: The red arrow is a previous link and can be used in Double Linked list to link nodes for a reverse

Instruction 12: The Score for Adding and Removing depends on number of Elements you want to add/remove : if you added 2/3 successfully you would score 66.66%

Instruction 13: The score for all the push backs is either 100% or 0% in a single linked list

Instruction 14: The score for all the push and pops will be either 100%, 50% or 0% : you will score 100 if you added or pushed successfully and all links connected else you will get 50% if some links are missing or get 0% if the node is not added/removed at all

Instructions on how to use the node graph



## Removing



Pop Front

Pop Back

Enter Index:

Pop Node at Index



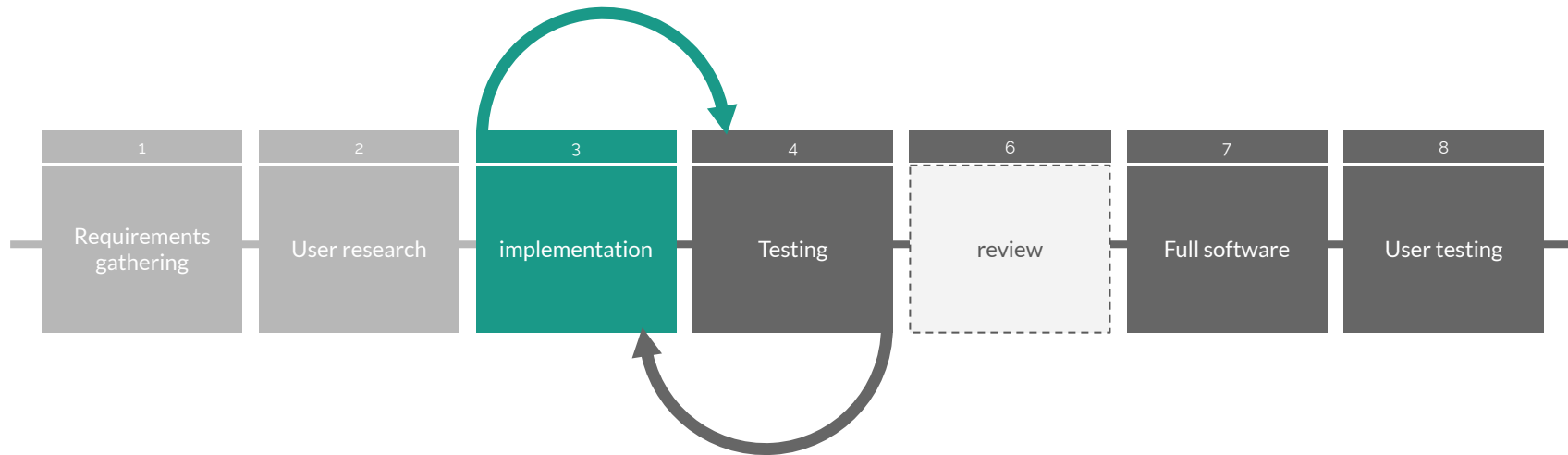
## Next Steps



# What next?

- Present the timeline.
- Extra documents presentation
- Questions

# Timeline



Questions?

---



# References

[Tips for Presenting Your Wireframes](#)

[3 Steps to Better UI Wireframes](#)

[Wireframing for Beginners](#)