Week 3

Date: 24.07.2025

Circular Linked List, Stack & its Application

1. Circular Linked List

You are tasked with building a **music player playlist manager**. The system should store songs in a way that, after the last song is played, it automatically moves to the first song without restarting the playlist manually. You decide to implement this using a **circular linked list**.

Requirements:

- Add a new song to the playlist
- Delete a song by title
- Display the current playlist in a loop
- Navigate to the next song continuously using circular logic

2. Stack (Array Implementation)

A **book-keeping system** in a small library records all return transactions for a day. The librarian wants a “Last-In-First-Out” way to undo the latest transactions in case of errors. You are to implement this using a **stack with array**.

Requirements:

- Push: Add a transaction (book ID)
- Pop: Undo last transaction
- Peek: See the most recent transaction
- Display all return transactions in LIFO order

3. Stack (Linked List Implementation)

A **browser application** keeps track of recently visited web pages. The user can go back to the previous page using a "Back" button. You must implement the "Back" feature using a **stack implemented via a linked list**.

Requirements:

- Push: Visit a new page (store URL)
- Pop: Go back to the previous page
- Display: Show current browsing history from top to bottom

4. Infix to Postfix Conversion & Evaluation (Linked List)

A **smart billing system** at a retail store calculates complex offers, discounts, and taxes on the total amount. The logic is often represented in infix mathematical expressions. You are asked to build a parser and calculator that can convert these expressions to **postfix** and evaluate them dynamically using a **stack implemented with a linked list**.

Requirements:

- Accept an infix expression (e.g., (500 + 20% of 500) - 50)
- Convert the expression to postfix using a linked list-based stack
- Evaluate the postfix expression
- Print result (Postfix expression & the final billing amount)

****