# Development of Teaching computer application

# 1. Existing features

### 1.1 Processor Architecture Visualization

- Registers (A, B, PC, SP, IR, AR, DR, OF, BR, etc.)
- ALU operations
- Memory (byte-addressable)
- Control unit
- Interrupt preview
- Data bus & address bus visualization

### 1.2 Program Execution

- Load program (assembly-like instructions)
- Step-by-step execution
- Run continuously
- Pause / Stop
- View changing register values live

### 1.3 Assembly Editor

- Basic built-in editor for writing LC-1 code
- Automatic memory mapping
- Validate instructions (LDA, STA, JMP, CAL, RET, ADD, SUB, NOT, AND, etc.)

### 1.4 Instruction Set Support

- Data movement: LDA, LDB, MOV
- Arithmetic: ADD, SUB
- Logic: AND, NOT
- Control flow: JMP, JPS, JPO
- Subroutine: CAL, RET
- Rotate: RRA, RLA
- HLT (halt)

# 2. Additional Features

### 2.1 Improved UI / UX

- Modern, responsive interface

### 2.2 Real-Time Graphical Animations

- Data bus lines lighting up during execution
- Register updates with animation
- ALU operations visually highlighted
- Animated instruction cycle (fetch → decode → execute → writeback)

### 2.3 Instruction-Level Tooltips

When hovering over `LDA`, user sees:

- Meaning
- Binary encoding
- Execution cycle steps

### 2.4 Memory Visualization Enhancements

- Hex viewer
- Decimal viewer
- Binary viewer
- Change memory values manually
- Highlight updated cells in yellow for 1 second

### 2.5 Assembly → Binary Compiler

- Let the user see machine code
- Side-by-side view:  
  **Assembly | Machine Code | Micro-operations**

### 2.6 Error Highlighting

- Memory overflow
- Invalid instruction
- Stack underflow
- Illegal jump address

### 2.7 "Learning Mode" for students

- Guided tutorials built into the app

- “Show micro-operations” toggle

# 4. Technical Requirements

## 4.1 Tech Stack

- **JavaScript + HTML5 + CSS3**

## 4.2 Architecture

- **CPU Core Simulation**
    - Clock cycle
    - Fetch/Decode/Execute
    - Register operations
    - Memory operations
- **Visualization Layer**
    - Renders architecture
    - Animates data flow
    - Displays register state