1. Presentation Slides (10 Slides)

Prepare 10 slides explaining your project.

The slides should include:

1. Title Slide – Pocket Tool for Electrical Engineers, M. Abdullah 2023324, Mian AbdulMoiez 2023314, Mehadi Abbas 2023310, M. Aimal Khan 2023369. Data Structure and Algorithm (EE-222)

2. Problem Statement – The problem is to design and implement a Digital Systems Assistance (DSA) Tool that will assist electrical engineers in performing a variety of essential calculations and operations. The tool must provide functionalities for converting between decimal and binary number systems, performing Delta-to-Wye and Wye-to-Delta resistor configuration conversions, verifying resistor values using color bands, and calculating Norton and Thevenin equivalents for circuits. Furthermore, it should support the generation and verification of Hamming codes for error detection. The tool should be user-friendly, offering a menu-driven interface to easily interact with these functions. It must also log the results of each operation for future reference and provide real-time updates on the status of ongoing calculations. The goal is to create a versatile tool that improves the efficiency and accuracy of engineers in their work with circuit analysis, coding, and troubleshooting tasks.

3. Objectives – **Tool** for electrical engineers to perform various calculations and operations.

- Convert between decimal and binary numbers, including both decimal-to-binary and binary-to-decimal conversions.
- Perform Delta-to-Wye and Wye-to-Delta resistor configuration conversions for circuit analysis.
- Verify resistor values based on color bands, including tolerance and multiplier calculations.
- Calculate Norton and Thevenin equivalents for electrical circuits.
- Generate and check Hamming codes for error detection and correction.
- Provide a menu-driven interface for easy interaction and navigation through different functions.
- Support engineers in circuit analysis, enhancing accuracy and efficiency in daily tasks.

4. Proposed Solution – The proposed solution is to develop a **Pocket Tool** that integrates essential functionalities for electrical engineers into a single, intuitive platform. The tool will offer a menu-driven interface to easily perform various operations such as converting between decimal and binary numbers, using recursive algorithms to streamline number system conversions. It will also provide the capability to convert Delta-to-Wye and Wye-to-Delta resistor configurations, enabling engineers to quickly calculate equivalent resistances in circuits. The tool will allow for resistor value verification based on color bands, calculating both resistance values and tolerance. Additionally, the tool will calculate Norton and Thevenin equivalents for simplifying circuits and support Hamming code generation and error detection. A key feature of the solution is its logging and history management functionality, which will record each operation's results for future

reference. The tool will use efficient data structures such as stacks, vectors, and arrays to optimize performance. By consolidating these capabilities into one platform, the proposed solution aims to reduce the time engineers spend on routine calculations, improve the accuracy of their work, and enhance productivity through real-time updates and easy access to past operations.

5. Architecture/Design – The architecture of the Traffic Management System consists of the following key components:

1. User Interface: A menu-driven interface for easy navigation, allowing users to select operations like decimal/binary conversion, resistor calculations, and Hamming code generation.
2. Core Functionality: Includes operations like Delta-to-Wye resistor conversion, resistor value verification, Norton and Thevenin equivalent calculations, and Hamming code generation/error detection.
3. Data Structures: Utilizes stacks, vectors, and arrays for efficient data storage and operations, ensuring quick calculations and smooth user interaction.
4. Logging and History Management: Stores results of operations for future reference, allowing users to review past calculations and track their work.
5. Error Handling: Validates inputs to ensure correct data entry, providing error messages for invalid inputs and ensuring the accuracy of calculations.

6. Tools and Technologies – **Tools, Languages, and Frameworks Used:**

- **Programming Language**: C++
- **Libraries**:
    - iostream, vector, string, fstream, stack, cmath queue, list, algorithm
- **IDE/Compiler**: Dev, Visual Studio, Code::Blocks, or GCC

7. Features/Modules – Modules added in our project are:

- Decimal and Binary Conversion: Allows conversion between decimal and binary formats using recursive algorithms for efficient calculations.
- Resistor Value Verification: Calculates resistor values based on color bands and provides tolerance values, ensuring accurate resistor specifications.
- Delta-to-Wye and Wye-to-Delta Conversion: Supports quick calculations of equivalent resistances for Delta and Wye resistor configurations, aiding in circuit analysis.
- Hamming Code Generation and Error Detection: Generates Hamming codes and checks received codes for errors, ensuring reliable data transmission.
- Norton and Thevenin Equivalent Calculations: Computes Norton and Thevenin equivalents for simplifying complex electrical circuits, enhancing circuit analysis efficiency.
- History Logging: Records the results of each operation, allowing users to track past calculations and retrieve them as needed for reference.

- User-Friendly Interface: A simple, menu-driven interface with real-time updates and clear prompts for smooth user interaction and error-free operations.

9. Challenges and Solutions – Challenges Faced and their solutions are:

1. Challenge 1: Input Validation and Error Handling
    - *Solution*: Implement strict input validation to ensure users provide correct data types and values (e.g., valid binary digits, resistor color bands). Error messages guide users to input correct data and avoid crashes.
2. Challenge 2: Handling Complex Calculations Efficiently
    - *Solution*: Use efficient algorithms and data structures such as recursion for number conversions, stacks for binary conversion, and arrays/vectors for storing results, ensuring fast and accurate calculations.
3. Challenge 3: Real-Time Updates and Result Logging
    - *Solution*: Create a logging system to save operation results while providing real-time updates on the screen. This enables users to track previous calculations and reduces the risk of data loss.

10. Future Scope – Integration with Additional Engineering Tools

- Improved User Interface (UI) Enhancements
- Cloud-Based Synchronization
- Mobile Application Development
- Machine Learning Integration
- Support for Simulation and Visualization