Security Features Implemented

1. **Cryptographically Secure Random Generation**
    - **Uses Web Crypto API's getRandomValues() instead of Math.random()**
    - **Prevents predictability in generated passwords**
2. **Shannon Entropy Calculation**
    - **Scientific measurement of password strength**
    - **Bits of entropy accurately reflect resistance to brute force attacks**
3. **Modulo Bias Mitigation**
    - **Implements rejection sampling to ensure uniform distribution**
    - **Prevents statistical bias in character selection**
4. **Comprehensive Character Sets**
    - **Flexible options for various character types**
    - **Option to exclude ambiguous characters for better usability**
5. **Password Crack Time Estimation**
    - **Realistic assessment based on modern computing capabilities**
    - **Helps users understand the practical strength of their password**
6. **Zero-Server Interaction**
    - **All generation happens client-side in the browser**
    - **No passwords are ever transmitted over the network**
7. **Temporary Password History**
    - **Keeps recent passwords in memory only (not localStorage)**
    - **Automatically cleared when the page is refreshed**
8. **Visual Strength Indicators**
    - **Color-coded indicators help users assess password quality**
    - **Animation provides visual feedback when generating new passwords**

This implementation adheres to NIST Special Publication 800-63B guidelines for secure password handling and generation, making it suitable for educational and institutional environments where security is a priority.