1. QWERTY-Braille Mapping
Braille characters are formed using combinations of six specific QWERTY keys:

*D, W, Q, K, O, P → represent dots 1 to 6 in a 2x3 Braille grid.
A custom Braille-to-character mapping is created using these combinations, where each unique set maps to an English alphabet letter.

2. Input Decoding
User input is a sequence of key sets (e.g., {'D', 'K'}), each representing one Braille character.

*These sets are converted into their corresponding letters using the predefined mapping.

*The full sequence is then joined to form the decoded word.

3. Error Handling and Suggestions
To correct errors in the decoded word, we compare it against a known dictionary using:

*Levenshtein Distance – Measures the number of edits (insertions, deletions, substitutions) required to convert one word to another.

*The closest matches (with the lowest distances) are returned as suggested corrections.

4. Efficiency Considerations
For small dictionaries, linear search with Levenshtein distance is sufficient.
For large-scale use, future improvements could include:

*BK-Trees for approximate search.

*Tries for prefix-based pruning and fast lookup.

*Caching frequently corrected words (learning mechanism).