Longest common prefix

To find the longest common prefix among a list of strings, here's a structured approach with detailed guidance:

Strategy: Vertical Scanning (character by character)

This is one of the most intuitive and efficient methods.

Idea:
1. Pick the first string as a reference.
2. For each character position `i` in the reference string:
   - Compare it with the character at position `i` in all other strings.
   - If any string is too short, or a character doesn't match, stop.
3. Return the prefix built so far.

Example: For `["flower", "flow", "flight"]`
- compare 'f' at index 0 in all → ✓
- compare 'l' at index 1 → ✓
- compare 'o' at index 2 → "flight" has 'i' → ✗ stop
  Return "fl"

Edge Cases:
- empty input → return `""`
- only one string → return it
- one string is empty → return `""`
- no common prefix at all → return `""`

Pseudocode Outline:

```
if strs is empty:
    return ""
prefix = strs[0]
for each string in strs[1:]:
    while current string doesn't start with prefix:
        remove last char from prefix
```

if prefix becomes empty:
return ""

return prefix

This method avoids full character-by-character comparison and uses the "startWith" logic efficiently.