

Project Title: Vocabulary Trainer

The goal of this application is to help students learn and memorize English vocabulary based on their class textbooks. By using different modes (training and testing), students can review words regularly and test their understanding effectively.

This is a console-based Python program that allows students to practice English vocabulary in an interactive way. The vocabulary is organized by units (from their school book), and students can use the program in three different modes:

1. Editing Mode
2. Training Mode
3. Testing Mode

Vocabulary is saved in a file (vocab.json).

1. Editing Mode

- Add new vocabulary words by unit.
- Delete or update existing words.

2. Training Mode

- The word (in English) appears on the screen.
- After a short delay , the meaning in Hebrew/Arabic appears.
The idea of the delay is to allow the students trying to memorize the meaning of words.
- Each word is shown **7 times** randomly across sessions to improve memory.
- Users can choose which **unit** to practice.
- Users can choose a range of words in each unit.

3. Testing Mode for each Unit!

- One-time quiz for each word.

- Score is calculated based on correct answers.
- Provides feedback at the end of the test.

Vocabulary is saved in a file (vocab.json).

1. Let's now **break down Editing Mode** and identify all required functions and coding concerns.

We'll need four core functions:

1. `add_word(unit, word, meaning)`
 - Add a new word and its meaning to a given unit.
2. `delete_word(unit, word)`
 - Remove a specific word from a unit.
3. `update_word(unit, old_word, new_word, new_meaning)`
 - Modify an existing word or its meaning.
4. `list_words(unit)`
 - Show all words and meanings in a unit (helpful before delete/update).

⚠ Things to Pay Attention To

Topic	What to Handle
✅ File Operations	Always load and save using <code>load_vocab()</code> and <code>save_vocab()</code>
❌ Duplicates	Avoid inserting the same word twice in the same unit
? Unit Existence	Check if the unit exists; optionally offer to create it
🖌 Word Formatting	Clean word input (e.g., lowercase, strip whitespace) to prevent key mismatch
📋 Input Validation	Ensure correct input (non-empty strings, word exists before deletion, etc.)

Optional:

- Search by word across all units.
- Add confirmation before deleting.
- Add CLI menu for user to select units and actions.

2. Let's now **break down Training Mode** and identify all required functions and coding concerns.

- a. Let User Choose Unit.
- b. Show list of units (e.g., "unit_1", "unit_2") :Show All English Words in That Unit
- c. Ask User: Practice Full Unit or a Range?
 - If "full unit" → proceed with all words
 - If "range" → ask for:
 - start_word (e.g., "apple")
 - end_word (e.g., "school")
 - Validate both words exist and start_word appears before end_word
- d. **For Each Word in Selected Range or Unit**
 - Repeat 7 times:
 - Show the English word
 - Wait 3 seconds (or press Enter)
 - Show the meaning
 - Update and save repeat_count for persistent tracking

Let's now **break down Testing Mode** and identify all required functions and coding concerns.

Goal:

Give students a **one-time quiz** for the words in a chosen unit and **evaluate** their recall of meanings.

Features Overview

Feature	Description
Unit Selection	User selects a unit from available ones.
One-time Attempt	Each word appears only once in the test.
User Input Required	User types the meaning of the English word.
Scoring	Score calculated based on correct answers.
Case-Insensitive Match	Compare user input to expected meaning.
Feedback	At the end, show: score, total questions, and which words were incorrect.

Testing Flow

1. **Load Vocabulary**
2. **Prompt User to Choose Unit**
3. **Loop through all words in unit**
 - Show word
 - Prompt: "What is the meaning?"
 - Compare user input to the stored meaning

- Track correct/incorrect

4. End of Test Summary

- Number of correct answers
- Score percentage
- List incorrect answers with correct meanings

Score Range	Feedback
85–100	🌟 Excellent! Great job remembering your vocabulary.
60–84	👍 Good work! A bit more practice and you'll master it.
30–59	🟡 Keep going! You're getting there, but review needed.
0–29	🔴 Needs improvement. Study the words again and try once more.

5. The `main.py` file is the entry point to your Vocabulary Trainer app. It presents a simple text menu where the user can choose between the three main modes: editing, training, and testing vocabulary.

Editing Mode
Training Mode
Testing Mode
Exit