A screenshot of a computer

AI-generated content may be incorrect.

please can you create a test case in @test\_ngram\_models.py to test @ngram\_analyzer.py

based on teh requirement in @ngram.md. Please do not change the DB structure in @typing\_data.db or the database setup in @database\_manager. You can use the MCP addin typing\_sqlite to determine the database structure

Please use a fixture to create a temporary database and temporary @database\_manager and use the Initialize\_tables feature to initialize the dB.  
Please use a fixture to create a test session within this db, and return this as a PracticeSession object. There is a good example of a practice session in the database – you can use the MCP typing\_sqlite to query it

Finally – please create a list of keystroke objects with 2 keystrokes as above

The first test case just asserts that we have a practice session with a valid session id and that our keystroke list has 2 entries

-------------------------  
  
please can you create a test case in @test\_ngram\_models.py to test @ngram\_analyzer.py

based on teh requirement in @ngram.md. Please do not change the DB structure in @typing\_data.db or the database setup in @database\_manager. You can use the MCP addin typing\_sqlite to determine the database structure

start with defining session ID like the attached screenshot:  
Our first test case is for 1 keystroke - just the letter T  
  
There should be zero ngrams identified of any length

Please can you add another test case in @test\_ngram\_models.py to test @ngram\_analyzer.py - very similar to the one above   
- name: 2 keystroke no errors  
- two keystrokes - T and h with no errors  
- time\_since\_previous on T is 0, time since previous on h is 500ms  
  
The ngram speed for this should be 500mg, with only one ngram via the class. in the DB should be one entry in ngram\_speed and zero entries in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures  
  
- name: 2 keystroke – error at 1

Sitation: - two keystrokes - T and h expected. However Gh was typed and keystroke number 1 is marked as is\_error = true

* time\_since\_previous on keystroke 1 is 0 with is\_error = true
* time since previous on keystroke 2 is 500ms,

**Models:**

* There should be 1 ngrams of length 2, containing “Gh” with a time of 500ms and is\_error = true

**Database:**

* One row in ngram\_errors as above
* Zero rows in ngram\_speed  
    
  please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models.py to test @ngram\_analyzer.py - very similar to the one above   
  
- name: 3 keystroke no errors - three keystrokes - T and h and e with no errors - time\_since\_previous on T is 0, time since previous on h is 500ms, time since previous on e is 1000ms   
There should be 3 ngrams - two of length 2, and one of length 3  
- Th with a time of 500ms  
- he with a time of 1000ms  
- The with a time of 750ms  
  
please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures  
  
- name: 3 keystroke – error at 1

Situation: - three keystrokes - T and h and e expected. However Ghe was typed and keystroke number 1 is marked as is\_error = true

* Keystroke 1: keystroke: G; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 1: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: true
* Keystroke 1: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false

In the object there should be 3 ngrams:

* One of length 2
  + “he”, no error, time is 1000ms
* Zero of length 3

In database:

* One row in ngram\_speed
* Zero rows in ngram\_errors
  + Zero rows

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures  
  
- name: 3 keystroke – error at 2

Situation: - three keystrokes - T and h and e expected. However Tbe was typed and keystroke number 2 is marked as is\_error = true

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 1: keystroke: b; Expected: h; Time\_since\_previous: 500 ms; isError: true
* Keystroke 1: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false

In the object there shall be multiple ngrams:

* one of length 2
  + “Th”, error, time is 500ms
* Zero of length 3
* Zero of length 4

In database:

* No rows in ngram\_speed
* One row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures  
- name: 3 keystroke – error at 3

Sitation: - three keystrokes - T and h and e expected. However Thd was typed

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 1: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 1: keystroke: d; Expected: e; Time\_since\_previous: 1000 ms; isError: true

In the object There should be multiple ngrams:

* Three of length 2
  + “Th”, no error, total time is 500ms. Avg is 500
  + “he”, error, total time is 1000ms. Avg is 1000 ms
* One of length 3
  + “The”, error, total time is 1500ms. Average is 750 ms
* 0 of length 4

In database:

* One rows in ngram\_speed
* Two row in ngram\_errors
  + One is length 2, error, text is he, time is 1000ms
  + One is length 3, error, text is The, time is 750

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p2.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures, Please use @test\_three\_keystrokes\_no\_errors and @test\_two\_keystrokes\_no\_errors as a template

- name: 4 keystroke – no errors

Sitation: - target text: “Then”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: n; Expected: n; Time\_since\_previous: 300 ms; isError: false

In the object There should be the following ngrams:

* Three of length 2
  + “Th”, no error, total time is 500ms. Avg is 500/1
  + “he”, no error, total time is 1000ms. Avg is 1000/1
  + “en”, no error, total time is 300ms. Avg is 300/1
* Two of length 3
  + “The”, no error, total time is 1500ms. Avg is 1500/2
  + “hen”, no error, total time is 1300ms. Avg is 1300/2
* 1 of length 4
  + “Then”, no error, total time is 1800 ms. Avg is 1800/3

In database:

* 6 rows in ngram\_speed
* Zero rows in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p2.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures. Please use @test\_three\_keystrokes\_error\_at\_first and @test\_two\_keystrokes\_error\_at\_first as a template

- name: 4 keystrokes – error at first

Situation: - target text: “Then”

* Keystroke 1: keystroke: G; Expected: T; Time\_since\_previous: 0; isError: true
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: n; Expected: n; Time\_since\_previous: 300 ms; isError: false

In the object There should be 3 ngrams:

* Three of length 2
  + “he”, no error, total time is 1000ms, avg is 1000/1
  + “en”, no error, total time is 300ms, avg is 300/1
* Two of length 3
  + “hen”, no error, total time is 1300ms, avg is 1300/2

In database:

* Three rows in ngram\_speed
* Zero rows in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p2.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: four keystrokes – error at second

Sitation: - target text: “Then”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: g; Expected: h; Time\_since\_previous: 500 ms; isError: true
* Keystroke 3: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: n; Expected: n; Time\_since\_previous: 300 ms; isError: false

In the object There should be multiple ngrams:

* Two of length 2
  + “Th”, error, total time is 500ms. Avg is 500/1
  + “en”, no error, total time is 300ms. Avg is 300/1
* zero of length 3
* zero of length 4

In database:

* one rows in ngram\_speed
* One row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p2.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: four keystrokes – error at third

Sitation: - target text: “Then”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: g; Expected: e; Time\_since\_previous: 1000 ms; isError: true
* Keystroke 4: keystroke: n; Expected: n; Time\_since\_previous: 300 ms; isError: false

In the object There should be multiple ngrams:

* Two of length 2
  + “Th”, no error, total time is 500ms. Avg is 500/1
  + “he”, error, total time is 1000ms. Avg is 1000/1
* One of length 3
  + “The”, error, total time is 1500ms. Avg is 1500/2
* zero of length 4

In database:

* one rows in ngram\_speed
* two rows in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p2.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: four keystrokes – error at fourth

Sitation: - target text: “Then”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: b; Expected: n; Time\_since\_previous: 300 ms; isError: true

In the object There should be multiple ngrams:

* Three of length 2
  + “Th”, no error, total time is 500ms. Avg is 500/1
  + “he”, no error, total time is 1000ms. Avg is 1000/1
  + “en”, error, total time is 300ms. Avg is 300/1
* Two of length 3
  + “The”, no error, total time is 1500ms. Avg is 1500/2
  + “hen”, error, total time is 1300ms. Avg is 1300/2
* 1 of length 4
  + “Then”, error, total time is 1800 ms. Avg is 1800/3

In database:

* Three rows in ngram\_speed
* three rows in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p3.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: 5 keystroke – backspace at 1 with mistake

Sitation: - target text: “Then”

* Keystroke 1: keystroke: G; Expected: T; Time\_since\_previous: 0; isError: true
* Keystroke 2: keystroke: backspace; Expected: h; Time\_since\_previous: 500 ms; isError: true
* Keystroke 3: keystroke: T; Expected: e; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: h; Expected: n; Time\_since\_previous: 300 ms; isError: false
* Keystroke 5: keystroke: e; Expected: e; Time\_since\_previous: 170 ms; isError: false

In the object There should be multiple ngrams:

* length 2
  + “Th”, no error, total time is 300 ms; avg time is 300ms/1
  + “he”, no error, total time is 170 ms; avg time is 170ms/1
* Length 3:
  + “The”, no error, total time is 470 ms; avg time is 470/2
* Length 4: None
* Length 5: None

In database:

* 3 row in ngram\_speed
* Zero row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p3.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: 5 keystroke – backspace at 2 with mistake

Sitation: - target text: “Then”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: g; Expected: h; Time\_since\_previous: 300 ms; isError: true
* Keystroke 3: keystroke: backspace; Expected: h; Time\_since\_previous: 500 ms; isError: true
* Keystroke 4: keystroke: h; Expected: h; Time\_since\_previous: 300 ms; isError: false
* Keystroke 5: keystroke: e; Expected: e; Time\_since\_previous: 170 ms; isError: false

In the object There should be multiple ngrams:

* length 2
  + “Th”, error, total time is 300ms; avg time is 300ms/1
  + “he”, no error, total time is 170 ms; avg time is 170ms/1
* Length 3: None
* Length 4: None
* Length 5: None

In database:

* 1 row in ngram\_speed
* 1 row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p3.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: 5 keystroke – backspace at 3 with no mistake

Sitation: - target text: “Then”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: backspace; Expected: e; Time\_since\_previous: 1000 ms; isError: true
* Keystroke 4: keystroke: h; Expected: n; Time\_since\_previous: 300 ms; isError: false
* Keystroke 5: keystroke: e; Expected: e; Time\_since\_previous: 170 ms; isError: false

In the object There should be multiple ngrams:

* length 2
  + “Th”, no error, total time is 500 ms; avg time is 500ms/1
  + “he”, no error, total time is 170 ms; avg time is 170ms/1
* Length 3: None
* Length 4: None
* Length 5: None

In database:

* 2 row in ngram\_speed
* zero row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p3.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: 5 keystroke – space at 3

Sitation: - target text: “Th en”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: space; Expected: space; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: e; Expected: e; Time\_since\_previous: 300 ms; isError: false
* Keystroke 5: keystroke: n; Expected: n; Time\_since\_previous: 170 ms; isError: false

In the object There should be multiple ngrams:

* length 2
  + “Th”, no error, total time is 500 ms; avg time is 500ms/1
  + “en”, no error, total time is 170ms; avg time is 170/1
* Length 3: None
* Length 4: None
* Length 5: None

In database:

* 2 row in ngram\_speed
* Zero row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p3.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: 5 keystroke – space at 2

Sitation: - target text: “1 cat”

* Keystroke 1: keystroke: 1; Expected: 1; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: space; Expected: space; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: c; Expected: c; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: a; Expected: a; Time\_since\_previous: 300 ms; isError: false
* Keystroke 5: keystroke: t; Expected: t; Time\_since\_previous: 170 ms; isError: false

In the object There should be multiple ngrams:

* length 2
  + “ca”, no error, total time is 300 ms; avg time is 300ms/1
  + “at”, no error, total time is 170ms; avg time is 170/1
* Length 3: None
  + “cat”, no error, total time is 470 ms; avg time is 470ms/2
* Length 4: None
* Length 5: None

In database:

* 2 row in ngram\_speed
* Zero row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

Please can you add another test case in @test\_ngram\_models\_p3.py to test @ngram\_analyzer.py - very similar to the one above. Please use the MCP addin typing\_sqlite to check and change DB schema / table structures

- name: 5 keystroke – space at 5

Sitation: - target text: “Then ”

* Keystroke 1: keystroke: T; Expected: T; Time\_since\_previous: 0; isError: false
* Keystroke 2: keystroke: h; Expected: h; Time\_since\_previous: 500 ms; isError: false
* Keystroke 3: keystroke: e; Expected: e; Time\_since\_previous: 1000 ms; isError: false
* Keystroke 4: keystroke: n; Expected: n; Time\_since\_previous: 300 ms; isError: false
* Keystroke 5: keystroke: space; Expected: space; Time\_since\_previous: 170 ms; isError: false

In the object There should be multiple ngrams:

* length 2
  + “Th”, no error, total time is 500 ms; avg time is 500ms/1
  + “he”, no error, total time is 1000 ms; avg time is 1000/1
  + “en”, no error, total time is 300ms; avg time is 300/1
* Length 3:
  + “The”, no error, total time is 1500 ms; avg time is 1500/2
  + “hen”, no error, total time is 1300 ms; avg time is 1300 ms/2
* Length 4:
  + “Then”, no error, total time is 1800 ms; avg time is 1800/3
* Length 5: None

In database:

* 6 row in ngram\_speed
* Zero row in ngram\_errors

please validate the timing of the ngrams, the count, and the length - both via the object, and via the DB?

please delete @test\_ngram\_size.py as the first step before moving forward.

Please can you create tests in a newly created @test\_ngram\_size.py (under @tests\models) to robustly test the size behaviors for @ngram.py and @ngram\_manager.py - specifically the generation of ngrams but not saving these to the database or any other features of @ngram.py or @ngram\_manager.py other than checking the size (don’t worry about duration or speed or validity or errors etc)

You will likely need a test fixture to initiate a database using @database\_manager.py along with a test category, a test snippet and a test session. Please use methods in #category\_manager.py #snippet\_manager.py and #session\_manager.py to create these required pieces

Please make sure you have at least 15 parameterized tests with keystroke lists of various lengths (from zero, one, and upwards) with 10ms between keystrokes to make sure that not only do you have the right counts of ngrams of various sizes - and also that these ngrams are correct.

this is how it should behave:

#### Size:

So for a length of text like "abcde" there are 4 ngrams of length 2 (ab, bc, cd and de), there are 3 ngrams of length 3 (abc, bcd, and cde), 2 ngrams of length 4 (abcd, and bcde) and one ngram of size 5. This continues up to ngrams of size 10 maximum. Ngrams are always sequential keystrokes, not random combinations

We do not want any ngrams shorter than 2, or longer than 10

Please can you create tests in a newly created @test\_ngram\_speed.py (under @tests\models) to robustly test the speed behaviors for @ngram.py and @ngram\_manager.py - specifically the generation of ngrams but not saving these to the database or any other features of @ngram.py or @ngram\_manager.py other than checking the speed (don’t worry about ngram validity or errors etc)

You will likely need a test fixture to initiate a database using @database\_manager.py along with a test category, a test snippet and a test session. Please use methods in #category\_manager.py #snippet\_manager.py and #session\_manager.py to create these required pieces

Please make sure you have at least 15 parameterized tests with keystroke lists of various lengths (from zero, one, and upwards) with 10ms between keystrokes to make sure that not only do you have the right counts of ngrams of various sizes - and also that these ngrams are correct.

this is how it should behave:

**#### Speed:**

The speed of an ngram is stored in ngram\_speed\_ms - this is calculated as the time between the first keystroke and the last keystroke divided by (ngram size - 1) - then converted to ms

So if you have the following keystrokes

a: 08:01:00.00

b: 08:01:01.00

c: 08:01:01.50

then:

- ngram ab has time 1000 ms (1000 / 1)

- ngram bc has time 500 ms (500 / 1)

- ngram abc has time 750ms (1500 ms / 2)

Please delete @test\_ngram\_valid.py before you do anything else

Please can you create tests in a newly created @test\_ngram\_valid.py (under @tests\models) to robustly test the valid behaviors for @ngram.py and @ngram\_manager.py - specifically the generation of ngrams but not saving these to the database or any other features of @ngram.py or @ngram\_manager.py other than checking the validity (don’t worry about ngram speed or errors etc)

Please make sure you have at least 15 (as many as you need) parameterized tests with keystroke lists with various combinations of keystrokes (from zero, one, and upwards) to make sure that not only do you have the right counts of ngrams of various sizes - and also that these ngrams are correct. It is important that you test the behaviours here below including backspaces, spaces, errors on the keystrokes, zero duration keystrokes etc.

this is how it should behave:

**#### valid status**

an ngram is valid if it doesn't hit any of the following

  - Skip n-grams with errors in any position except the last character.

    - Skip n-grams containing any backspace characters (backspaces act as sequence separators).

    - Skip n-grams with a total typing time of 0.0 ms.

    - Skip n-grams containing spaces in any position (spaces act as sequence separators).

Please can you create tests in a newly created @test\_ngram\_error.py (under @tests\models) to robustly test the error behaviors for @ngram.py and @ngram\_manager.py - specifically the generation of ngrams but not saving these to the database or any other features of @ngram.py or @ngram\_manager.py other than checking the error status on an ngram

Please make sure you have at least 15 (as many as you need) parameterized tests with keystroke lists with different combinations of errors and non-errors on keystrokes to test this error flag robustly.

this is how it should behave:

**#### Error Status**

each keystroke has a status of "iserror".      any ngrams with only an error in the last position should have an error status of true, otherwise error = false

Please can you create tests in a newly created @test\_ngram\_clean.py (under @tests\models) to robustly test the clean behaviors for @ngram.py and @ngram\_manager.py - specifically the generation of ngrams but not saving these to the database or any other features of @ngram.py or @ngram\_manager.py other than checking the error status on an ngram

Please make sure you have at least 15 (as many as you need) parameterized tests with keystroke lists with different combinations of errors and non-errors on keystrokes as well as different characters and backspaces and times to test this error flag robustly.

**#### Clean status**

A "clean" n-gram meets ALL the following criteria:

  - No errors in any position

  - No backspace characters in any position (backspaces act as sequence separators)

  - No spaces in any position (spaces act as sequence separators)

  - Total typing time is greater than 0.0 ms

Please can you create tests in a newly created file called @test\_ngram\_persist.py (under @tests\models) to robustly test the db persistence behaviours.  
  
The main behaviour you are testing is that ngrams which are “clean” and “valid” are saved to ngram\_speed table with unique IDs – and ngrams which have an “error” flag are saved to ngram\_errors table.   
  
  
You will likely need a test fixture to initiate a database using @database\_manager.py along with a test category, a test snippet and a test session. Please use methods in #category\_manager.py #snippet\_manager.py and #session\_manager.py to create these required pieces

Please make sure you have at least 15 parameterized tests with keystroke lists of various lengths (from zero, one, and upwards) with various combinations of error flag, valid & clean flags. Please make sure that the database has the right right counts of ngrams of various sizes in the table - and also that these ngrams are correct.