**Challenge 1**

Bonus: Suggest how this app can be tested (No need to code this part)

Answer:   
This app can be tested using Postman application to send HTTP query to localhost:port in local PC.

**Part 2**  
Explain how to improve on token-based authentication.

Answer:   
There are some ways to improve such as:

2.1 Keep the signing key secretly and safe.

2.2 Define an expiration for token, so that user has to re-authenticate

2.3 Send token over HTTPS, not http.  
  
**Challenge 2**

Note: You do not need to code anything. You can describe and explain in words.

* How would you implement a pagination api that needs to retrieve records from a high volume database?

Answer:   
The api needs to set limit and offset to certain number in order to pretend overload/slow data retrieving from dense database. It returns a subset of data for user, and will continue next offset if user chooses to search further.

* Discuss at least two pagination schemes and their pros/cons

Answer:   
There are some common pagination schemes, such as “Offset-based”, “Cursor-based”

1. “Offset”:

User specifics limit and offset of expected data, then server will return results from database accordingly.

Pros:   
 - Simple to implement for client and server.  
 - Possible to jump to arbitrary pages.  
 - Suitable for small datasets, clients need to jump around pages.

Cons:  
 - Unreliable results if database is changed frequently due to insert/delete records.  
 - Inefficient for large datasets, when offset is big, due to counting.

1. “Cursor”:

To solve the disadvantage of “Offset-based” scheme.  
Client requests with a limit  
Then server will respond with results and a next-cursor.  
After that, client includes this next-cursor in subsequent requests. Server can use this cursor to continue move along the records without counting from index 0.

Pros:  
 - Improve the performance.  
 - Consistent results.  
 - Suitable for large datasets, and traverse pages is acceptable.

Cons:  
 - Clients need to traverse through page by page, one by one.  
 - Records need to be added sequentially to the DB.  
 - Clients need to manage the next cursor.

* Give examples of client request messages and server response messages in each scheme

Answer:

1. “Offset-based”:

Client: GET /products?limit=50&offset=100

Server: return items from 101 to 150 (offset 100 from top, row counts 50 per time)

1. “Cursor”:  
   Client: limit = 100, cursor = 0 (first query)

Server: returns first 100 result (1-100), with cursor point to 100.  
  
Client: limit = 100, cursor = 100 (use cursor value assigned by server)

Server: return next 100 result (101-200), starting from cursor point 100.

Mention any third-party library/tools you would use.  
 Answer:

express-paginate