ByteQuest Screening Task APIs

Github link for the API code.

Video Link for the API demonstration.

1. GET '/products/'

This API returns the first five products sorted alphabetically by their title.

2. GET '/products/page?pageNumber='

This API takes in an optional page number (defaults to 1 if not provided) and shows 5 products per page.

3. GET '/products/:id'

This API takes a product ID and returns its information if it exists. If not, it responds with an error message.

4. POST '/products'

This API takes in the following request object.

{

Title: Required String with maximum 100 characters.

Description: Required String with maximum 800 characters.

Category: Required String with maximum 30 characters.

Price: Required Number.

Stock: Required Number.

Images: An array of image URLs.

}

The API validates this object, responds with error message if any, and then saves the object to the database and responds with a success message.

5. PUT '/:id'

This API takes in as few or as many attributes of the product object shown above and updates it atomically in the database, so that no concurrency issues arise.

6. DELETE '/:id'

This API takes in the ID of an existing product and deletes it from the database if it exists.

Some extra notes from me:

1. This project was put together in 3 hours. Yes, I see plenty of ways to improve this. A few examples would be to add a user authentication system with different classes, like buyer, seller etc. Then we could introduce things like creating orders and reviews and ratings etc. But that is beyond the scope of this project.
2. The APIs for this were not made with a specific front-end in mind. So, they're very general and hence might not be as clean as they can be. I was only told to make the APIs so I did the best I could.
3. All of the APIs are working on a MongoDB Atlas collection. I have not exposed the URL on Github so you won't be able to run the APIs on your machine. Which is why I will attach a video link of me executing them and showing my screen.
4. Lastly, thank you for reading all of this. I hope you liked my project and it was up to the standard you were looking for. I have done more complex work than this and am capable of doing much more, given the chance.