

Lab 6

- Due Mar 22 by 11:59pm
- Points 100
- Submitting a file upload
- File Types zip
- Available until Mar 29 at 11:59pm

This assignment was locked Mar 29 at 11:59pm.

CS-546 Lab 6

A Product API

For this lab, you will create a simple server that provides an API for someone to Create, Read, Update, and Delete products and also product reviews.

We will be practicing:

- Separating concerns into different modules:
- Database connection in one module
- Collections defined in another
- Data manipulation in another
- Practicing the usage of **async / await** for asynchronous code
- Continuing our exercises of linking these modules together as needed
- Developing a simple (10 route) API server

Packages you will use:

You will use the [mongodb](https://mongodb.github.io/node-mongodb-native/) package to hook into MongoDB

You may use the [lecture 4 code](https://github.com/stevens-cs546-cs554/CS-546/tree/master/lecture_04) and the [lecture 5 code](https://github.com/stevens-cs546-cs554/CS-546/tree/master/lecture_05) and [lecture 6 code](https://github.com/stevens-cs546-cs554/CS-546/tree/master/lecture_06) as a guide.

You can read up on [express](http://expressjs.com/) on its home page. Specifically, you may find the [API Guide section on requests](http://expressjs.com/en/4x/api.html#req) useful.

You must save all dependencies you use to your package.json file

Folder Structure

YOU MUST use the directory and file structure in the code stub or points will be deducted. You can download the starter template here: [Lab6_stub.zip](#)

(<https://sit.instructure.com/courses/71954/files/13156840?wrap=1>)_ ↓

(https://sit.instructure.com/courses/71954/files/13156840/download?download_frd=1)

PLEASE NOTE: THE STUB DOES NOT INCLUDE THE PACKAGE.JSON FILE. YOU WILL NEED TO CREATE IT! DO NOT FORGET TO ADD THE START COMMAND AND "type": "module" property.

YOU MUST TRIM ALL INPUTS THAT ARE STRINGS! No cases of empty strings or strings with just spaces are acceptable

Database Structure

You will use a database with the following structure:

- The database will be called **FirstName_LastName_lab6**
- The collection you use to store products will be called `products` you will store a sub-document of `reviews`
- **You must only use ONE collection. The reviews are stored in the products collection as an array of objects. If you use two collections, you will lose 50% of your grade!**

products

The schema for a product is as followed:

```
{
  _id: ObjectId,
  productName: string,
  productDescription: string,
  modelNumber: string,
  price: number/float (9.99, 199.99, 59.00, 100, 5 etc),
  manufacturer: string,
  manufacturerWebsite: string (must contain full url http://www.phillips.com),
  keywords: [array, of, strings],
  categories: [array, of, strings],
  dateReleased: string date in mm/dd/yyyy format,
  discontinued: boolean,
  reviews:[], (an array of review objects, you will initialize this field to be an empty array when a product is created),
  averageRating: number (from 0 to 5 this will be a computed average from all the product reviews posted for a product, the initial value of this field will be 0 when a product is created)
}
```

The `_id` field will be automatically generated by MongoDB when a product is inserted, so you do not need to provide it when a product is created.

An example of how the 83 inch LG C3 OLED TV would be stored in the DB:

```
{
  _id: ObjectId("507f1f77bcf86cd799439011"),
  productName: "83 inch LG C3 OLED TV",
  productDescription: "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished wa
```

tching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories."

```
{
  modelNumber: "OLED83C3PUA",
  price: 4757.29,
  manufacturer: "LG",
  manufacturerWebsite: "http://www.lgelectronics.com",
  keywords: ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  categories: ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  dateReleased: "02/27/2023",
  discontinued: false,
  reviews: [],
  averageRating: 0
}
```

The Product Review Sub-document (stored within the Product document)

```
{
  _id: ObjectId,
  title: string,
  reviewDate: string (string value of a date in MM/DD/YYYY format),
  reviewerName: string,
  review: string,
  rating: number ranging from 1-5 (floats will be accepted as long as they are in the range of 1-5. 1.5 or 4.8 for example would be valid. .5, 5.5 or 3.65, would not be. We will only use one decimal place)
}
```

An example of the review sub-document:

```
{
  _id: ObjectId("603d992b919a503b9afb856e"),
  title: "Wow!!",
  reviewDate: "10/13/2022",
  reviewerName: "Patrick Hill",
  review: "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!",
  rating: 5
}
```

For the `_id` field, you will need to generate a new `objectId` and pass it in when you create the review, (there was some sample code in lab 4's specs that showed generating a new `ObjectId`)

data/products.js

In `products`, you will create and export 5 methods. Create, Read (one for getting all and also one getting by id), Update, and Delete. **You must do FULL error handling and input checking for ALL functions as you have in previous labs, checking if the input is supplied, correct type, range etc. and throwing errors when you encounter bad input. You must also trim all inputs! You can use the functions you used for lab 4 however, you will need to create an update function and also modify your create and getAll functions. rename from lab 4 will not be used.**

As a reminder, you will keep the same function names you used in lab 4:

```
create(productName, productDescription, modelNumber, price, manufacturer, manufacturerWebsite, keywords,
categories, dateReleased, discontinued)
getAll()
get(id)
remove(id)
update(id, productName, productDescription, modelNumber, price, manufacturer, manufacturerWebsite, keywords,
categories, dateReleased, discontinued) Note: this is the new function you will create.
```

For the functions you did in lab 4, all the same input requirements apply in this lab, for new update function, those requirements are stated below.

NOTE: `averageRating` and reviews are not passed into either create or update functions.

For create: When a product is created, in your DB function, you will initialize the reviews array to be an empty array. You will also initialize `averageRating` to be 0 when a product is created.

For getAll: You will have to modify this function so it just returns the `_id` and name of the products as shown below in the GET /products route example. (Hint: use a projection!)

For update: You will not modify the `averageRating` or reviews in the update function, you must keep them intact as they were before the update.

`async update(id, productName, productDescription, modelNumber, price, manufacturer, manufacturerWebsite, keywords, categories, dateReleased, discontinued)`

This function will update **all** the data of the product currently in the database.

If `id, productName, productDescription, modelNumber, price, manufacturer, manufacturerWebsite, keywords, categories, dateReleased, discontinued` are not provided at all, the method should throw. (**All fields need to have valid values**);

If `id, productName, productDescription, modelNumber, manufacturer, manufacturerWebsite, dateReleased` are not `strings` or are empty strings, the method should throw.

If `id` is not a valid `ObjectId`, the method should throw.

If `price` is not a number greater than 0, whole numbers and decimals allowed (**only allow 2 decimal points for the cents, no more than two!**), the method should throw.

If `manufacturerWebsite` does not start with `http://www.` and end in a `.com`, and have at least 5 characters in-between the `http://www.` and `.com` this method will throw.

If `keywords, categories` are not arrays and if they do not have **at least** one element in each of them that is a valid `string`, or are empty strings the method should throw. (each element should be a valid string but the arrays should contain at LEAST one element that's a valid string).

If `dateReleased` is not a valid date in mm/dd/yyyy format then the method should throw. You will not have to take into account leap years but it must be a valid date. For example:09/31/2022 would not

be valid as there are not 31 days in September. 02/30/1995 would not be valid as there are never 30 days in Feb.

If `discontinued` is not a boolean, then the method should throw.

Note: FOR ALL INPUTS: Strings with empty spaces are NOT valid strings. So no cases of " " are valid. You should also trim all string inputs!

If the update succeeds, return the entire product object as it is after it is updated.

data/reviews.js

In reviews, you will create and export 5 methods. Create, Read (one for getting all and also one getting by id), update and Delete. **You must do FULL error handling and input checking for ALL functions as you have in previous labs, checking if input is supplied, correct type, range etc. and throwing errors when you encounter bad input. You must also trim all inputs!**

Function Names:

```
createReview(productId, title, reviewerName, review, rating) Note: Notice we are not passing reviewDate,
this will be set to the current date when the review is created
getAllReviews(productId)
getReview(reviewId)
removeReview(reviewId)
updateReview(reviewId, reviewObj)
```

`async createReview(productId, title, reviewerName, review, rating);`

This async function will return to the newly created review object, with **all** of the properties listed in the above schema.

If `productId, title, reviewerName, review, rating` are not provided at all, the method should throw. **(All fields need to have valid values);**

If `productId, title, reviewerName, review` are not `strings` or are empty strings, the method should throw.

If the `productId` provided is not a valid `ObjectId`, the method should throw

If the product does not exist with that `productId`, the method should throw.

If `rating` is not a number in the range of 1 to 5, the method should throw (floats will be accepted as long as they are in the valid range of 1-5. 1.5 or 4.8 for example, would be valid. .5, 5.5 or 3.65 would not be valid We will only use one decimal place).

If `rating` has more than one decimal place, the method should throw.

REMINDER: reviewDate is not passed into this function, you will set that to be the current date when the review is added. It MUST be in mm/dd/yyyy format.

NOTE: When a review is added, you will need to recalculate the average for all the reviews for that product and then update the averageRating field in the main document to reflect the new average.

For the `_id` field, you will need to generate a new objectId and pass it in when you create the review, (there was some sample code in lab 4's specs that showed generating a new ObjectId)

async getAllReviews(productId);

This function will return an array of objects of the reviews given the `productId` return ONLY the reviews for the product, not any of the other product data. **If there are no reviews for the product, this function will return an empty array**

If the `productId` is not provided, the method should throw.

If the `productId` provided is not a string, or is an empty string, the method should throw.

If the `productId` provided is not a valid `ObjectId`, the method should throw

If the product doesn't exist with that `productId`, the method should throw.

async getReview(reviewId);

When given a `reviewId`, this function will return a review from a product. Return ONLY the review object and not all of the product data.

If the `reviewId` is not provided, the method should throw.

If the `reviewId` provided is not a string, or is an empty string, the method should throw.

If the `reviewId` provided is not a valid `ObjectId`, the method should throw

If the review doesn't exists with that `reviewId`, the method should throw.

```
{
  _id: ObjectId("603d992b919a503b9afb856e"),
  title: "Wow!!",
  reviewDate: "10/13/2022",
  reviewerName: "Patrick Hill",
  review: "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!",
  rating: 5
}
```

Notice, you are only returning ONE single review object from the reviews array, do NOT return any of the product information, do not return an array with one single review object, your function will only return a single object as shown above. Use a projection and there is an exact example of how to do this in the lecture code.

async updateReview(reviewId, updateObject):

This function will update a review in the database and then return the product object that the review belongs to show that the review sub-document was updated. You will pass all the data from the request body into this route from the `PATCH /reviews/review/reviewId` route. Since this update is for a PATCH request, one or MORE of the fields from the review object can be passed in. It's easier to make the 2nd parameter an object since it can contain one or more fields to be updated.

If the `reviewId` is not provided, the method should throw.

If the `reviewId` provided is not a string, or is an empty string, the method should throw.

If the `reviewId` provided is not a valid `ObjectId`, the method should throw

If the review doesn't exist with that `reviewId`, the method should throw.

If `reviewId`, `title`, `reviewerName`, `review` are not `strings` or are empty strings, the method should throw. (if any of the fields are supplied in the `updateObject` parameter)

If `rating` is supplied in the `updateObject` parameter and is not a number in the range of 1 to 5, the method should throw (floats will be accepted as long as they are in the valid range of 1-5. 1.5 or 4.8 for example, would be valid. .5, 5.5 or 3.65 would not be valid. We will only use one decimal place).

If `rating` is supplied in the `updateObject` parameter and has more than one decimal place, the method should throw.

REMINDER: reviewDate is not passed into this function, you will set that to be the current date when the review is updated. It MUST be in mm/dd/yyyy format.

NOTE: When a review is updated, if the rating was supplied and updated, you will need to recalculate the average for all the reviews for that product and then update the averageRating field in the main document to reflect the new average.

Returns the full product object showing the review has been updated:

```
{
  _id: ObjectId("507f1f77bcf86cd799439011"),
  productName: "83 inch LG C3 OLED TV",
  productDescription: "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  modelNumber: "OLED83C3PUA",
  price: 4757.29,
  manufacturer: "LG",
  manufacturerWebsite: "http://www.lgelectronics.com",
  keywords: ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  categories: ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  dateReleased: "02/27/2023",
  discontinued: false,
```



```

    reviews:[{"_id":ObjectId("607f1f77bcf86cd799439022"),"title": "Amazing!!! Wow!!","reviewDate": "03/05/2024", "reviewerName": "Patrick Hill","review": "This TV is amazing! I don't know how I'll ever go back after experiencing OLED!", "rating": 5}],
    ],
    averageRating: 5
}

```

async removeReview(reviewId):

This function will remove the review from the product in the database and then return the product object that the review belonged to show that the review sub-document was removed from the product document.

If the `reviewId` is not provided, the method should throw.

If the `reviewId` provided is not a string, or is an empty string, the method should throw.

If the `reviewId` provided is not a valid `ObjectId`, the method should throw

If the review doesn't exist with that `reviewId`, the method should throw.

Returns the full product object showing the review has been removed:

```

{
  _id: ObjectId("507f1f77bcf86cd799439011"),
  productName: "83 inch LG C3 OLED TV",
  productDescription: "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  modelNumber: "OLED83C3PUA",
  price: 4757.29,
  manufacturer: "LG",
  manufacturerWebsite: "http://www.lgelectronics.com",
  keywords: ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  categories: ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  dateReleased: "02/27/2023",
  discontinued: false,
  reviews:[],
  averageRating: 0
}

```

NOTE: When a review is deleted, you will need to recalculate the average for all the reviews for that product and then update the averageRating field in the main document to reflect the new average.

routes/products.js

You must do FULL error handling and input checking for ALL routes! checking if input is supplied, correct type, range etc. and responding with proper status codes when you encounter bad input. All the input types, values and ranges that apply to the DB functions apply to the routes as well so you will do all the same checks in the routes that you do in the DB functions before sending the data to the DB function

GET /products

Responds with an array of all products in the format of `{"_id": "product_id", "productName": "product_name"}` Note: Notice you are **ONLY** returning the product ID, and product name. HINT: USE A PROJECTION!

```
[{ "_id": "603d965568567f396ca44a72", "productName": "83 inch LG C3 OLED TV"}, { "_id": "704f456673467g306fc44c34", "productName": "iPhone 14 Pro - 1TB Space Gray"}, ...]
```

POST /products

Creates a product with the supplied data in the request body, and returns the new product (**ALL FIELDS MUST BE PRESENT AND CORRECT TYPE**).

You should expect the following JSON to be submitted in the request.body:

```
{
  "productName": "83 inch LG C3 OLED TV",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  "modelName": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronics.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false
}
```

If `productName`, `productDescription`, `modelName`, `price`, `manufacturer`, `manufacturerWebsite`, `keywords`, `categories`, `dateReleased`, `discontinued` are not provided at all, the route should issue a 400 status code and end the request. (**All fields need to have valid values**);

If `productName`, `productDescription`, `modelName`, `manufacturer`, `manufacturerWebsite`, `dateReleased` are not `strings` or are empty strings, the route should issue a 400 status code and end the request.

If `price` is not a number greater than 0, whole numbers and decimals allowed (**only allow 2 decimal points for the cents, no more than two!**), the route should issue a 400 status code and end the request.

If `manufacturerWebsite` does not start with `http://www.` and end in a `.com`, and have at least 5 characters in-between the `http://www.` and `.com` this route should issue a 400 status code and end the request.

If `keywords`, `categories` are not arrays and if they do not have **at least** one element in each of them that is a valid `string`, or are empty strings this route should issue a 400 status code and end the request. (each element should be a valid string but the arrays should contain at LEAST one element that's a valid string).

If `dateReleased` is not a valid date in mm/dd/yyyy format then this route should issue a 400 status code and end the request. You will not have to take into account leap years but it must be a valid date. For example: 09/31/2022 would not be valid as there are not 31 days in September. 02/30/1995 would not be valid as there are never 30 days in Feb.

If `discontinued` is not a boolean, then this route should issue a 400 status code and end the request.

If the JSON provided does not match the above schema or fails the conditions listed above, you will issue a 400 status code and end the request.

If the JSON is valid and the product can be created successfully, you will return the newly created product (as shown below) with a 200 status code.

```
{
  "_id": "507f1f77bcf86cd799439011"),
  "productName": "83 inch LG C3 OLED TV",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  "modelName": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronics.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false,
  "reviews": [],
  "averageRating": 0
}
```

GET /products/:productId

Example: `GET /products/507f1f77bcf86cd799439011`

Responds with the full content of the specified product. So you will return all details of the product.

if `_id` is not a valid `ObjectId`, you will issue a 400 status code and end the request

If no product with that `_id` is found, you will issue a 404 status code and end the request.

You will return the product (as shown below) with a 200 status code along with the product data if found.

```
{
  "_id": "507f1f77bcf86cd799439011",
  "productName": "83 inch LG C3 OLED TV",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  "modelNumber": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronics.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false,
  "reviews": [{"_id": "607f1f77bcf86cd799439022", "title": "Wow!", "reviewDate": "10/13/2022", "reviewerName": "Patrick Hill", "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!", "rating": 5}],
  "averageRating": 5
}
```

`PUT /products/:productId`

Example: `PUT /products/507f1f77bcf86cd799439011`

This request will update a product with information provided from the PUT body. Updates the specified product **by replacing** the product data with the new product data, and returns the updated product. **(All fields need to be supplied in the request.body, even if you are not updating all fields)**

You should expect the following JSON to be submitted:

```
{
  "productName": "83 inch LG C3 OLED TV UPDATED!!!!",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more.
```

re. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories."

```
{
  "modelName": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronicupdated.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false
}
```

if the `id` url parameter is not a valid `ObjectId`, you will issue a 400 status code and end the request

If no product exists with the `:id` provided, return a 404 and end the request.

If `productName`, `productDescription`, `modelName`, `price`, `manufacturer`, `manufacturerWebsite`, `keywords`, `categories`, `dateReleased`, `discontinued` are not provided at all, the route should issue a 400 status code and end the request. **(All fields need to have valid values);**

If `productName`, `productDescription`, `modelName`, `manufacturer`, `manufacturerWebsite`, `dateReleased` are not `strings` or are empty strings, the route should issue a 400 status code and end the request.

If `price` is not a number greater than 0, whole numbers and decimals allowed (**only allow 2 decimal points for the cents, no more than two!**), the route should issue a 400 status code and end the request.

If `manufacturerWebsite` does not start with `http://www.` and end in a `.com`, and have at least 5 characters in-between the `http://www.` and `.com` this route should issue a 400 status code and end the request.

If `keywords`, `categories` are not arrays and if they do not have **at least** one element in each of them that is a valid `string`, or are empty strings this route should issue a 400 status code and end the request. (each element should be a valid string but the arrays should contain at LEAST one element that's a valid string).

If `dateReleased` is not a valid date in mm/dd/yyyy format then this route should issue a 400 status code and end the request. You will not have to take into account leap years but it must be a valid date. For example: 09/31/2022 would not be valid as there are not 31 days in September. 02/30/1995 would not be valid as there are never 30 days in Feb.

If `discontinued` is not a boolean, then this route should issue a 400 status code and end the request.

If the JSON provided does not match the above schema or fails the conditions listed above, you will issue a 400 status code and end the request.

For PUT: you will not modify the `averageRating` or `reviews` properties in the PUT route, you must keep them intact as they were before the update. (I know this is not technically how a put

should work, but we want to retain the subdocuments and averageRating for the purpose of the assignment)

If the JSON provided in the PUT body is not as stated above or fails any of the above conditions, fail the request with a 400 error and end the request.

If the update was successful, then respond with that updated product (as shown below) with a 200 status code

```
{
  "_id": "507f1f77bcf86cd799439011",
  "productName": "83 inch LG C3 OLED TV UPDATED!!!",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  "modelName": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronicupdated.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false,
  "reviews": [{"_id": "607f1f77bcf86cd799439022", "title": "Wow!!", "reviewDate": "10/13/2022", "reviewerName": "Patrick Hill", "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!", "rating": 5}],
  "averageRating": 5
}
```

DELETE /products/:productId

Example: **DELETE /products/507f1f77bcf86cd799439011**

if the **id** url parameter is not a valid **ObjectId**, you will issue a 400 status code and end the request

If no product exists with the provided id url parameter, return a 404 and end the request.

Deletes the product, sends a status code 200 and returns:

```
{"_id": "507f1f77bcf86cd799439011", "deleted": true}
```

routes/reviews.js

You must do FULL error handling and input checking for ALL routes! checking if input is supplied, correct type, range etc. and responding with proper status codes when you encounter bad input. All the input types, values and ranges that apply to the DB functions

apply to the routes as well so you will do all the same checks in the routes that you do in the DB functions before sending the data to the DB function

GET /reviews/:productId

Example: GET /reviews/306f1f77bcf86cd799431423

Getting this route will return an array of all reviews in the system for the specified product id. You are only returning the array of reviews from the product, not any of the other product data!

if the `productId` is not a valid `ObjectId`, you will issue a 400 status code and end the request

If no reviews for the `productId` are found, you will issue a 404 status code and end the request.

if the `productId` is not found in the system, you will issue a 404 status code and end the request

You will return the array of reviews (as shown below) with a 200 status code along with the review data if found. Note: the `{...}` is just to show there are more review objects in the array Obviously you should not be returning "`{...}`" literally

```
[{ "_id": "603d992b919a503b9afb856e", "title": "Wow!!",
  "reviewDate": "10/13/2022", "reviewerName": "Patrick Hill",
  "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!",
  "rating": 5 }, {...}, {...}
]
```

POST /reviews/:productId

Example: POST /reviews/306f1f77bcf86cd799431423

Creates a review sub-document with the supplied data in the request body, and returns all the product data showing the reviews (**ALL FIELDS MUST BE PRESENT AND CORRECT TYPE**).

You should expect the following JSON to be submitted in the request.body:

```
{
  "title": "Wow!!",
  "reviewerName": "Patrick Hill",
  "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!",
  "rating": 5
}
```

If `title`, `reviewerName`, `review`, `rating` are not provided at all, the route will respond with a status code of 400 and end the request. (**All fields need to have valid values**);

If `title`, `reviewerName`, `review` are not `strings` or are empty strings, the route will respond with a status code of 400 and end the request.

If the `productId` URL parameter provided is not a valid `ObjectId`, the route will respond with a status code of 400 and end the request.

If the product doesn't exist with that `productId`, the route will respond with a status code of 404 and end the request.

If `rating` is not a number in the range of 1 to 5, the route will respond with a status code of 400 and end the request. (floats will be accepted as long as they are in the valid range of 1-5. 1.5 or 4.8 for example, would be valid. .5, 5.5 or 3.65 would not be valid We will only use one decimal place).

If `rating` has more than one decimal place, the route will respond with a status code of 400 and end the request.

REMINDER: reviewDate is not passed into this route from the request body, you will set that to be the current date when the review is added.

If the JSON provided does not match that schema above or it fails any of the above conditions, you will issue a 400 status code and end the request.

If the JSON is valid and the review can be created successful, you will return all the product data showing the reviews (as shown below) with a 200 status code.

```
{
  "_id": "507f1f77bcf86cd799439011",
  "productName": "83 inch LG C3 OLED TV UPDATED!!!",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  "modelName": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronicsonline.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false,
  "reviews": [{"_id": "607f1f77bcf86cd799439022", "title": "Wow!!", "reviewDate": "10/13/2022", "reviewerName": "Patrick Hill", "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!", "rating": 5}],
  "averageRating": 5
}
```

GET /reviews/review/:reviewId

Example: **GET /reviews/review/603d992b919a503b9afb856e**

If the `reviewId` is not a valid `ObjectId`, you will issue a 400 status code and end the request

If no review for the `reviewId` are found, you will issue a 404 status code and end the request.

You will return the review (as shown below) with a 200 status code.

```
{
  "_id": "603d992b919a503b9afb856e",
  "title": "Wow!!",
  "reviewDate": "10/13/2022",
  "reviewerName": "Patrick Hill",
  "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!",
  "rating": 5
}
```

Notice, you are only returning ONE single review object from the reviews array, do NOT return any of the product information, do not return an array with one single review object, your function will only return a single object as shown above. Use a projection and there is an exact example of how to do this in the lecture code.

PATCH /reviews/review/:reviewId

Example: **PATCH /reviews/review/603d992b919a503b9afb856e**

This request will update a review with information provided from the PATCH body. Updates the specified review by only updating the fields that are supplied in the req.body. Any of the fields that were supplied when a review was originally created can be supplied.

These were the required fields when we created a review:

```
{
  "title": "Wow!!",
  "reviewerName": "Patrick Hill",
  "review": "This TV was amazing! I don't know how I'll ever go back after experiencing OLED!",
  "rating": 5
}
```

So for the PATCH request, one or more of these fields could be supplied in the req.body. There should be at LEAST one field in the req.body that you are updating. If there is not at least one field with a valid value, you will fail the request with a 400 status code. For the fields supplied in the req.body, the same input validation needs to be done for them that you did when you created the review.

If **title**, **reviewerName**, **review** are not **strings** or are empty strings, the route will respond with a status code of 400 and end the request (if any of these fields are supplied in the req.body obviously, you only have to check the ones that are supplied in the req.body).

If the **reviewId** URL parameter provided is not a valid **ObjectId**, the route will respond with a status code of 400 and end the request.

If the review doesn't exist with that **reviewId**, the route will respond with a status code of 404 and end the request.

If **rating** is supplied in the req.body and is not a number in the range of 1 to 5, the route will respond with a status code of 400 and end the request. (floats will be accepted as long as they are in the valid

range of 1-5. 1.5 or 4.8 for example, would be valid. .5, 5.5 or 3.65 would not be valid We will only use one decimal place).

If `rating` is supplied in the req.body and has more than one decimal place, the route will respond with a status code of 400 and end the request.

REMINDER: reviewDate is not passed into this route from the request body, you will update that field to be the current date when the review is updated. If rating was supplied, then you will need to also recalculate the averageRating field in the main document.

If the review was updated, this route will respond with the entire product document, showing the update:

```
{
  "_id": "507f1f77bcf86cd799439011",
  "productName": "83 inch LG C3 OLED TV",
  "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
  "modelNumber": "OLED83C3PUA",
  "price": 4757.29,
  "manufacturer": "LG",
  "manufacturerWebsite": "http://www.lgelectronics.com",
  "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
  "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
  "dateReleased": "02/27/2023",
  "discontinued": false,
  "reviews": [{"_id": "607f1f77bcf86cd799439022", "title": "Amazing!!! Wow!!", "reviewDate": "03/05/2024", "reviewerName": "Patrick Hill", "review": "This TV is amazing! I don't know how I'll ever go back after experiencing OLED!", "rating": 5}],
  "averageRating": 5
}
```

Delete /reviews/review/:reviewId

Example: `DELETE /reviews/review/603d992b919a503b9afb856e`

Deletes the specified review for the specified `reviewId`, and then sends a 200 status code and returns the full product data that the review belonged to to show the review has been removed (The below example only had one review so the review has been removed and averageRating set back to be 0):

```
{
  "_id": "507f1f77bcf86cd799439011"),
  "productName": "83 inch LG C3 OLED TV UPDATED!!!",
```

```

    "productDescription": "The advanced LG OLED evo C-Series is better than ever. The LG OLED evo C3 is powered by the next-gen a9 AI Processor Gen6—exclusively made for LG OLED—for ultra-realistic picture and sound. And the Brightness Booster improves brightness so you get luminous picture and high contrast, even in well-lit rooms.* AI-assisted deep learning analyzes what you're watching to choose the best picture and sound setting for your content. The LG OLED evo C3 not only performs great, but looks great as well. With an almost invisible bezel, it will blend into the background for a seamless look. When you're finished watching, display paintings, photos and other content to blend the LG OLED evo C3 into your space even more. But that's not all. Experience less searching and more streaming, thanks to the next generation of AI technology from LG webOS 23. Every LG OLED comes loaded with Dolby Vision for extraordinary color, contrast and brightness, plus Dolby Atmos** for wrap-around sound. And LG's FILMMAKER MODE allows you to see films just as the director intended. Packed with gaming features, the LG OLED evo C-Series comes with everything you need to win like a 0.1ms response time, native 120Hz refresh rate and four HDMI 2.1 inputs. *Based on LG internal testing: 55/65/77/83 LG OLED evo C3 models are brighter than non-OLED evo B3 models and excludes the 42 and 48 LG OLED evo C3. **Dolby, Dolby Atmos and the double-D symbol are registered trademarks of Dolby Laboratories.",
    "modelName": "OLED83C3PUA",
    "price": 4757.29,
    "manufacturer": "LG",
    "manufacturerWebsite": "http://www.lgelectronicupdated.com",
    "keywords": ["TV", "Smart TV", "OLED", "LG", "Big Screen", "83 Inch"],
    "categories": ["Electronics", "Television & Video", "Televisions", "OLED TVs"],
    "dateReleased": "02/27/2023",
    "discontinued": false,
    "reviews": [],
    "averageRating": 0
  }

```

if the `reviewId` url parameter is not a valid `ObjectId`, you will issue a 400 status code and end the request.

If no review with that `reviewId` is found, you will issue a 404 status code and end the request.

NOTE: If a review is deleted, you need to recalculate and update the average for the `averageRating` field in the main product document

app.js

Your app.js file will start the express server on **port 3000**, and will print a message to the terminal once the server is started.

Tip for testing:

You should create a seed file that populates your DB with initial data for both products and reviews. This will GREATLY improve your debugging as you should have enough sample data to do proper testing and it would be rather time consuming to enter a product and reviews for that product one by one through the API. A seed file is not required and is optional but is highly recommended. You should have a DB with at least 10 products and multiple reviews for each product for proper testing (again, this is not required, but it is to ensure you can test thoroughly.)

General Requirements

1. You **must not submit** your node_modules folder
2. You **must remember** to save your dependencies to your package.json folder
3. You must do basic error checking in each function
4. Check for arguments existing and of proper type.

5. Throw if anything is out of bounds (ie, trying to perform an incalculable math operation or accessing data that does not exist)
6. If a function should return a promise, you should mark the method as an `async` function and return the value. Any promises you use inside of that, you should *await* to get their result values. If the promise should throw, then you should throw inside of that promise in order to return a rejected promise automatically. Thrown exceptions will bubble up from any awaited call that throws as well, unless they are caught in the async method.
7. You **must remember** to update your package.json file to set `app.js` as your starting script!
8. You **must** submit a zip file named in the following format: `LastName_FirstName_CS546_SECTION.zip`

Lab 6 Rubric Spring 2024

Criteria	Ratings		Pts
<p>GET /products</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the route.</p>	<p>5 to >0.0 pts 5 Points/Valid Input Test Case 1 Valid Input Test Case. The route will send the correct response if it is implemented correctly, and few test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.</p>	5 pts
<p>GET /products/:productId</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected.</p>	<p>5 to >0.0 pts 5 Points/Error Handling Test Case & 4 Points/Valid Input Test Case 2 Error Handling Cases & 1 Valid Input Test Case. The route is implemented correctly, sends the correct response, and few test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.</p>	7 pts
<p>GET /reviews/:productId</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected.</p>	<p>5 to >0.0 pts 5 Points/Error Handling Test Case & 4 Points/Valid Input Test Case 2 Error Handling Cases & 1 Valid Input Test Case. The route is implemented correctly, sends the correct response, and few test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.</p>	7 pts
<p>GET /reviews/review/:reviewId</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected.</p>	<p>5 to >0.0 pts 5 Points/Error Handling Test Case & 4 Points/Valid Input Test Case 2 Error Handling Cases & 1 Valid Input Test Case. The route is implemented correctly, sends the correct response, and few test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.</p>	7 pts
<p>DELETE /products/:productId</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected. The database must not contain deleted items.</p>	<p>8 to >0.0 pts 2 Points/Error Handling Test Case & 4 Points/Valid Input Test Case 2 Error Handling Cases & 1 Valid Input Test Case. The route is implemented correctly, sends the correct response, and few test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.</p>	8 pts
<p>DELETE /reviews/review/:reviewId</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected. The database must not contain deleted items.</p>	<p>8 to >0.0 pts 2 Points/Error Handling Test Case & 4 Points/Valid Input Test Case 2 Error Handling Cases & 1 Valid Input Test Case. The route is implemented correctly, sends the correct response, and few test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.</p>	8 pts
<p>POST /products</p> <p>You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will</p>	<p>20 to >0.0 pts 1 Point/Error Handling Test Case & 19 Points/Valid Input Test Case 1 Error Handling Case & 19 Valid Input Test Cases. The route is implemented correctly, sends the correct response, and all test cases pass.</p>	<p>0 pts All Test Cases Failed Incorrect implementation, none of</p>	20 pts

Criteria	Ratings		Pts
result in FULL points off for the test case(s) affected. The database must contain the new data EXACTLY as expected.	Points/Valid Input Test Case 10 Error Handling Cases & 1 Valid Input Test Case.	the test cases pass or the route crashes, hangs, or does not send any response.	
POST /reviews/:productId You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected. The database must contain the new data EXACTLY as expected.	10 to 20 pts Points/Error Handling Test Case & 8 Points/Valid Input Test Case 5 Error Handling Cases & 1 Valid Input Test Case.	0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.	18 pts
PUT /products/:productId You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected. The database must contain the altered data.	10 to 20 pts Points/Error Handling Test Case & 5 Points/Valid Input Test Case 5 Error Handling Cases & 1 Valid Input Test Case.	0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.	10 pts
PATCH /reviews/review/:reviewId You MUST return and send the response exactly as shown with the proper status code. Having an incorrect key, value, or output will result in FULL points off for the test case(s) affected. The database must contain the altered data.	10 to 20 pts Points/Error Handling Test Case & 5 Points/Valid Input Test Case 5 Error Handling Cases & 1 Valid Input Test Case.	0 pts All Test Cases Failed Incorrect implementation, none of the test cases pass or the route crashes, hangs, or does not send any response.	10 pts
<div>Total Points: 100</div>			