**ECOMMERCE - BACKEND - INTERNSHIP**

**Functionality :**

**Seller Product Functionality :**

1) Seller store creation ( one seller can have one store )

2) Seller add product to his store , catid is selected for a particular product from available categories / images of product uploaded to cloudinary

3) Products stock and price can be updated based on product id

4) Route for checking total no of products in a store and no of PRODUCTS BELONGS TO EACH CATEGORY IN A STORE

5) Seller can delete a product based on category id and store id

6) Get routes for products , product by categories

7) Get all categories on platform

**Seller Order Functionality :**

1) Total Order Revenue of a store

2) Total Pending Orders

3) Total Product Count in Orders in a specific category in a store

All 10 working

First deal with image uploads then apply delete and other functionality

**Admin Functionality :**

**SIDE INFO :**

1) We are passing the data as Multipart from front end so body checks wont work on it because they work on req.body not on req.multipart

So yes — you're absolutely right:

🟡 "body checks won't work because they work on req.body, but we are sending multipart/form-data, and req.body is not parsed yet."

**✅ What actually handles multipart/form-data?**

**Multer** does — and only when uploads.single(...) or uploads.any() runs.

Once Multer runs:

* It **populates** req.body (for form text fields)
* And adds the uploaded file to req.file

**NEXT TASK**

• uploads.single('imageUrl') processes a single file upload with the field name imageUrl.

• Multer populates req.file with file details (if a file is uploaded) and req.body with text fields.

• **Potential Issue**: The validation for imageUrl in the Express Validator chain expects imageUrl to be a string in req.body. However, if you’re uploading a file, imageUrl will likely be in req.file (e.g., req.file.path or a generated URL after saving the file), not req.body.imageUrl. This could cause validation to fail or behave unexpectedly if imageUrl is not sent as a text field.

The imageUrl validation in the array (body('imageUrl').optional().isString()) is problematic because, with multipart/form-data, the imageUrl field is typically the file itself, not a string in req.body. If a file is uploaded, req.body.imageUrl will be undefined, and req.file will contain the file details.

the multer populates the req.file with following

{ "fieldname": "image", "originalname": "pic.jpg", "mimetype": "image/jpeg", "buffer": <binary data>, "size": 12345 } Multer parses that request and attaches info about the uploaded file to req.file (for single) or req.files (for multiple).

2) Done setting up the project with winston and morgan

3) The global error handling works as follows :

All the errors caught in async function will result in promise failure they will propage to top of application and will be caught in index.js by error middleware , if we specify status code and message then they will be displayed otherwise 500 and server erro

So it is best to catch the error with display message and status code in each controller side logic using AppError

