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.

2) Done setting up the project with winston and morgan