|  |
| --- |
| [18-08-25](#day1) |
| [19-08-25](#day2) |
| [20-08-25](#day3) |
| [22-08-25](#day4) |
| [23-08-25](#day5) |
| [25-08-55](#day6) |
| 26-08-25 |
| [27-08-25](#day8) |
| [28-08-25](#day9) |
| [29-08-25](#day10) |
| [01-09-25](#day11) |
| [02-09-25](#day12) |
| [03-09-25](#day13) |
| [04-09-25](#day14) |
| [08-09-25](#day15) |
| [09-09-25](#day16) |

Notes = 150 page tak clear hai

Class 28 = 19:00

**18-08-25 day 1**

Frontend folder

Create next app

Backend folder

Npm init

Npm i express mongoose cors

Src/app

(admin-group) folder

(website-group) folder

(website-group)

Layout.js and page.js ko website-group me le jao

Global css ke .. kar do

Npm run dev

(admin-group)/admin

Layout.js

Page.js

Src

Component folder

Src/components/

Admincomponent folder

Websitecomponent folder

Src/component/websitecomponent/

Header.jsx

Footer.jsx

(website-group)

Page.jsx me kuch dummy card add kar lo

(website-group)

Layout.js me header and footerset kar lo {children} ke upar niche

Src/component/admincomponent

Admin sidemenu

(admin-group)

Layout.js me {children} ke upar add karo

Admincomponent

Header.jsx banao

(admin-group)

Layout.js me header add karo aur layout fix karo

(admin-group)

Admin-group ke liye dashboard banao aur page.jsx me add karo

Admin folder

Category folder

Page.jsx (is page par category ki table print honi chahiye jisme image, name, slug, action button and table ke top par add category button rahega)

Side menu flow bite

**18-08-25 day 2**

Backend

Index.js create this file

Add below code in index.js and start backend with node –watch index

const express = require('express');

const mongoose = require('mongoose')

const server = express();

server.use(express.json())

server.listen(

    5000,

    () => {

        console.log('Server Running PORT 5000')

    }

)

And connect database with backend

server.listen(

  5000,

  () => {

    console.log('Server Running PORT 5000')

    mongoose.connect("mongodb://localhost:27017/", { dbName:"ishop" }).then(

      () => {

        console.log('Database connected')

      }

    ).catch(() => {

      console.log('unable to connect database')

    })

  }

)

Module banana ke l iye MVC architecture follow karte hai

Mvc = module view controller

Backend me folder create kare

controller (isme logical code rahega)

model (isme schema rahega)

router (routing ke liye)

model

category.model.js

model/category.model.js

isme itana code likho aur given code ko chat gpt ko dedo aur validation ke liye

constmongoose = require('mongoose');

constcategorySchema = newmongoose.Schema({

  name: {

    type:String,

    required:true,

    unique:true

  },

  slug: {

    type:String,

    required:true,

    unique:true

  },

  image: {

    type:String,

    default:null

  },

  status: {

    type:Boolean,

    default:true

  }

}, {

  timestamps:true

});

Chat gpt kuch updates dega

const mongoose = require('mongoose');

const categorySchema = new mongoose.Schema({

  name: {

    type:String,

    required: [true, "Category name is required"],

    unique:true,

    trim:true, // removes extra spaces

    minlength: [3, "Category name must be at least 3 characters"],

    maxlength: [50, "Category name cannot exceed 50 characters"]

  },

  slug: {

    type:String,

    required: [true, "Slug is required"],

    unique:true,

    lowercase:true, // always stored in lowercase

    trim:true,

  },

  image: {

    type:String,

    default:null,

  },

  status: {

    type:Boolean,

    default:true

  }

}, {

  timestamps:true

});

module.exports = mongoose.model("Category", categorySchema);

router

category.router.js

router/category.router.js

isme kuch code add karo

const categoryRouter = require('express').Router()

categoryRouter.post('/create', )

Exaplaination :

Is line me hum router ko bol rahe hain ki jab **POST request** aaye /create path pe, toh **handler function** run karo.

 'post' HTTP method hai, jo usually naya data create karne ke liye use hota hai.

 '/create' route path hai — iska matlab jab koi client /create pe POST request bheje, toh yeh route activate hoga.

 handler ek function hota hai jo request ko process karta hai, jaise data lena, database me save karna, response bhejna.

aur jab data ayega to controller ke pass jayega

to category controller bana lo aur usme kuch add karos

controller/category.controller.js

const category = {

    create(req, res) {

        console.log(req);

        res.send("hello")

    }

**}**

**module.exports = category**

Explaination :

### Kya ho raha hai:

* Ye ek **controller object** hai jisme create method defined hai.
* Jab bhi /create pe POST request aayegi, ye create(req, res) function execute hoga.
* Is function me bas console.log(req) kiya gaya hai (request object ko console pe dikhane ke liye).
* Fir client ko response me "hello" bhej diya ja raha hai.

aur fir is category.controller.js ko router me add karna hai

constcategoryRouter = require('express').Router()

const {create} = require('../controller/category.controller')

categoryRouter.post('/create', create)

module.exports = categoryRouter

Explaination

**Kya ho raha hai:**

* require('express').Router() se ek **router object** banaya gaya hai, jise categoryRouter bola gaya hai.
* create function ko category.controller.js se import kiya gaya hai (jo second snippet me hai).
* categoryRouter.post('/create', create) matlab: jab koi client /create pe POST request kare, toh **controller ka create function** call hoga.
* Finally, categoryRouter ko export kiya gaya hai taaki ye app me kahin bhi import ho kar use ho sake.

index.js

const express = require('express');

const mongoose = require('mongoose')

const categoryRouter = require('./router/category.router')

const server = express();

server.use(express.json())

server.use('/category', categoryRouter)

server.listen(

    5000,

    () => {

        console.log('Server Running PORT 5000')

        mongoose.connect("mongodb://localhost:27017/", { dbName:"ishop" }).then(

            () => {

                console.log('Database connected')

            }

        ).catch(() => {

            console.log('unable to connect database')

        })

    }

)

## Step-by-step explanation:

### 1. const express = require('express');

* Express framework import kiya, jisse aap web server create kar sakte hain.

### 2. const mongoose = require('mongoose');

* Mongoose library import ki, jo MongoDB ke saath easily interact karne ke liye use hoti hai.

### 3. const categoryRouter = require('./router/cateogry.router');

* Aapka router module import kiya gaya hai jo category-related API routes handle karega.  
  Note: File name me spelling mistake lag rahi hai: cateogry.router → category.router ho sakta hai.

### 4. const server = express();

* Express app ka instance banaya, jise aap server bol rahe hain.

### 5. server.use(express.json());

* Yeh middleware hai jo **incoming requests ke JSON body ko parse** karta hai.
* Matlab agar client JSON data bhej raha hai, toh aapko req.body me data mil jayega.

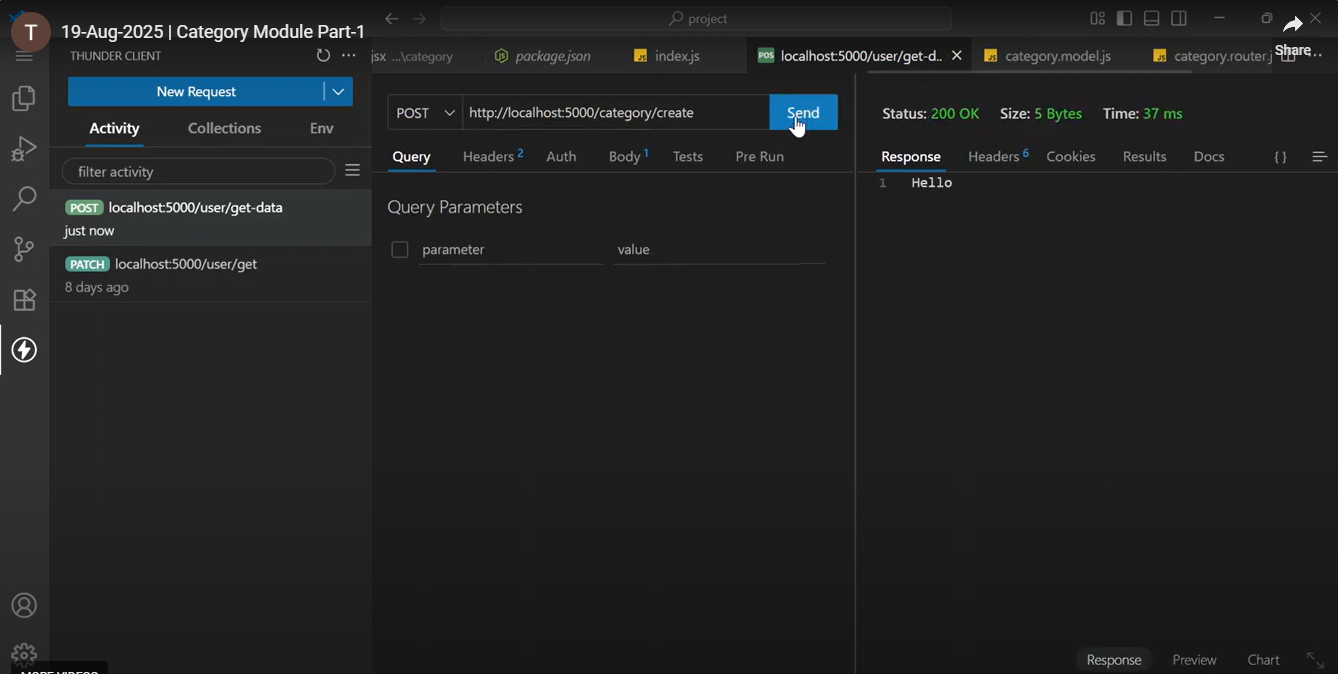
### 6. server.use('/category', categoryRouter);

* Is line ka matlab:
  + Jab bhi koi request /category path se start hoti hai, toh wo categoryRouter ko forward kar di jayegi.
  + Matlab aapka category-related saara routing logic categoryRouter me likha hoga.

### 7. server.listen(5000, () => { ... });

* Server ko **port 5000** pe start kar raha hai.
* Jab server successfully start ho jata hai, toh callback function execute hota hai.

Check on thunder



Response : hello

Category.controller.js me below wala code add karo

const categoryModel = require('../model/category.model')

constcategory = {

  asynccreate(req, res) {

    try {

      if (!req.body.name || !req.body.slug) {

        returnres.status(400).json({

          message:"name and slug is required",

          success:false

        })

      }

      constcategory = awaitcategoryModel.create({

        name:req.body.name,

        slug:req.body.slug

      })

      awaitcategory.save()

      res.status(201).json({

        message:"Category create",

        success:true

      })

    } catch (error) {

      res.status(500).json({

        message:"Internal server error",

        success:false

      })

    }

  }

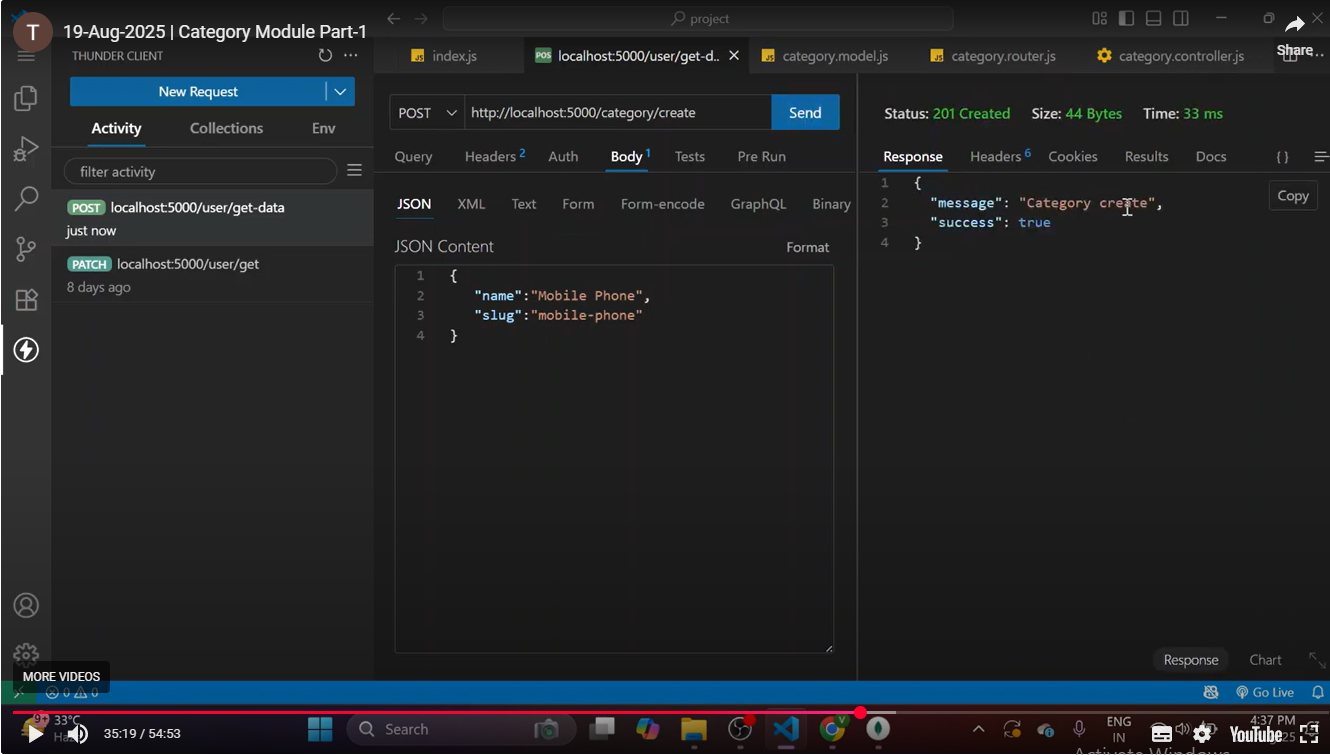
}

module.exports = category

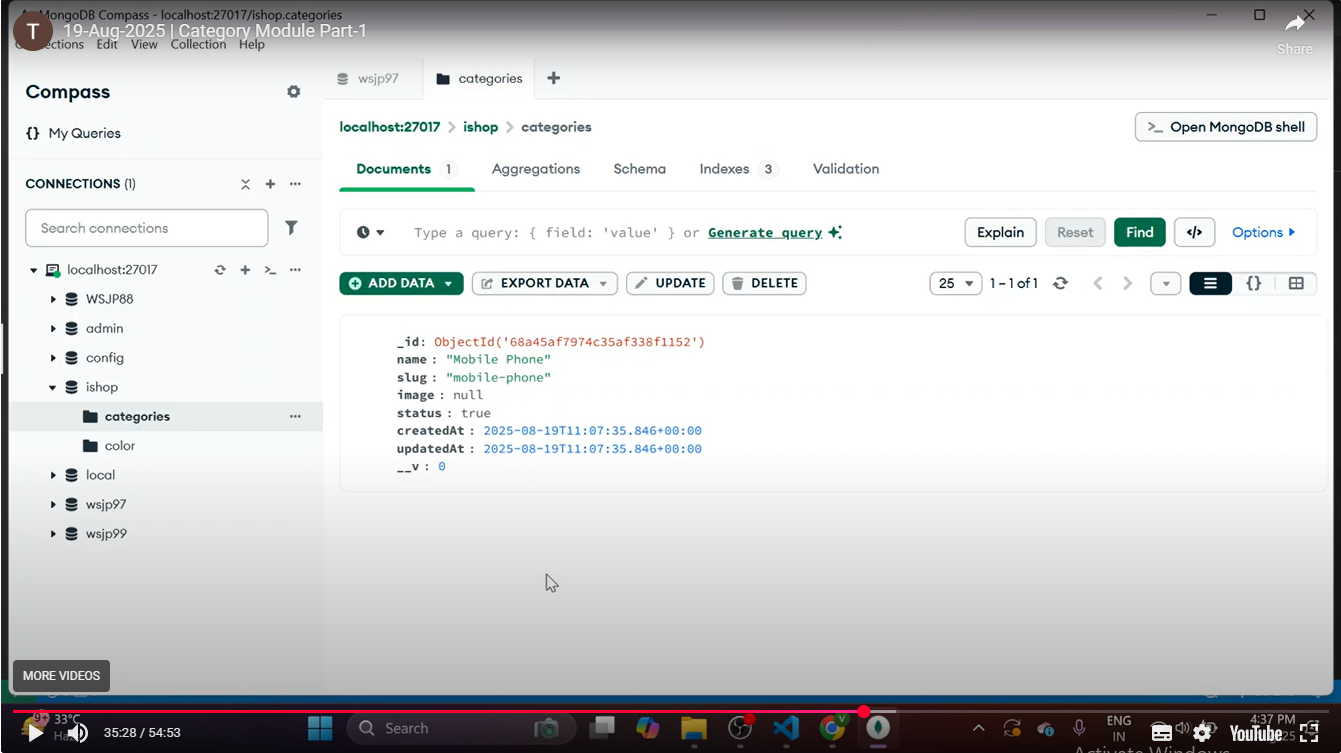
## 🧠 So, async/await ka matlab kya hai?

| **Keyword** | **Kya karta hai?** |
| --- | --- |
| async | Function ko **asynchronous** banata hai (ye promise return karega) |
| await | Kisi bhi **promise-returning operation** ko **wait** karne ke liye use hota hai |

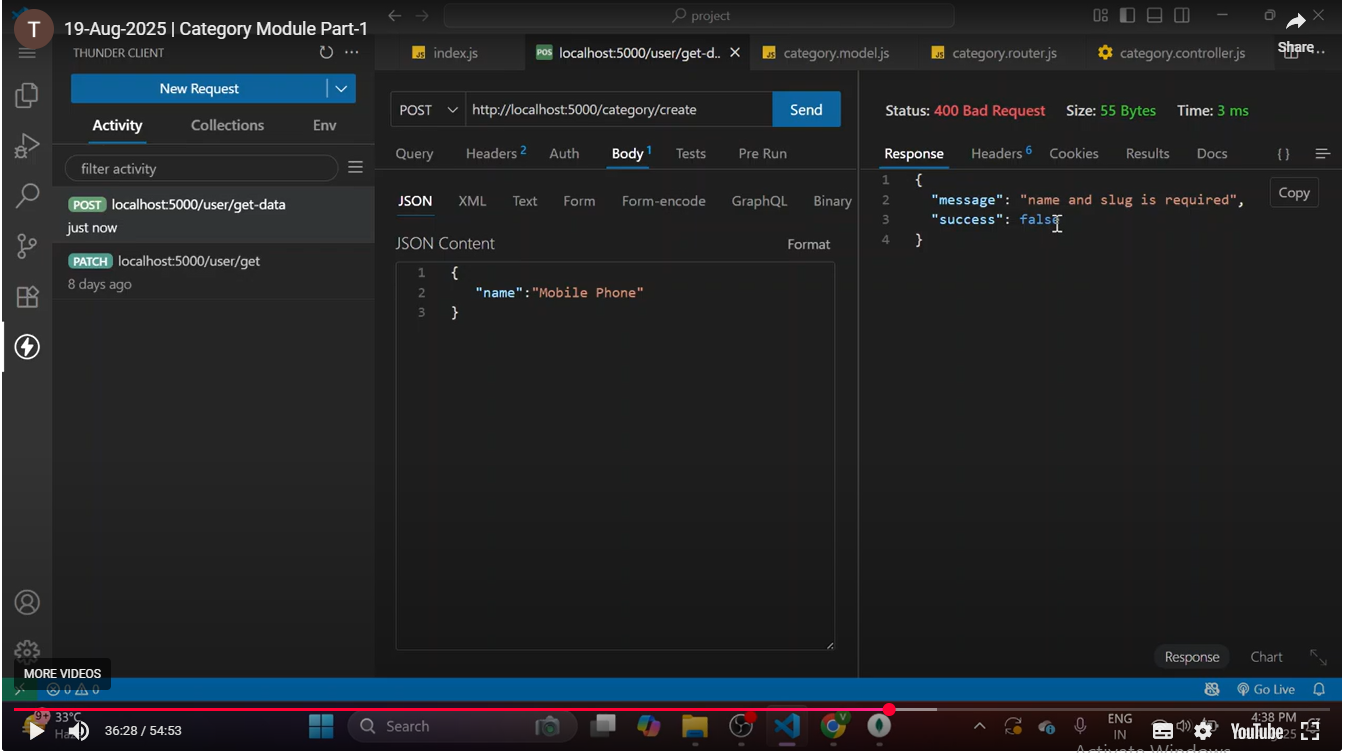
check on thunder ki category bani ya nahi



Then fir database me jakar dekho ki database me category add hui ya nahi



Body me slug pass nahi karne par

Error ayega name and slug is required

Ek folder copy karna hai backend jisme response.js file hai

utility/response.js

category.controller.js (code ko optimize karna)

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse } =require('../utility/response')

constcategory= {

    asynccreate(req, res) {

        try {

            if (!req.body.name||!req.body.slug) {

                returnnoContentResponse(res)

            }

            constcategory=awaitcategoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            awaitcategory.save()

            returncreatedResponse(res)

        } catch (error) {

            returnserverErrorResponse(res)

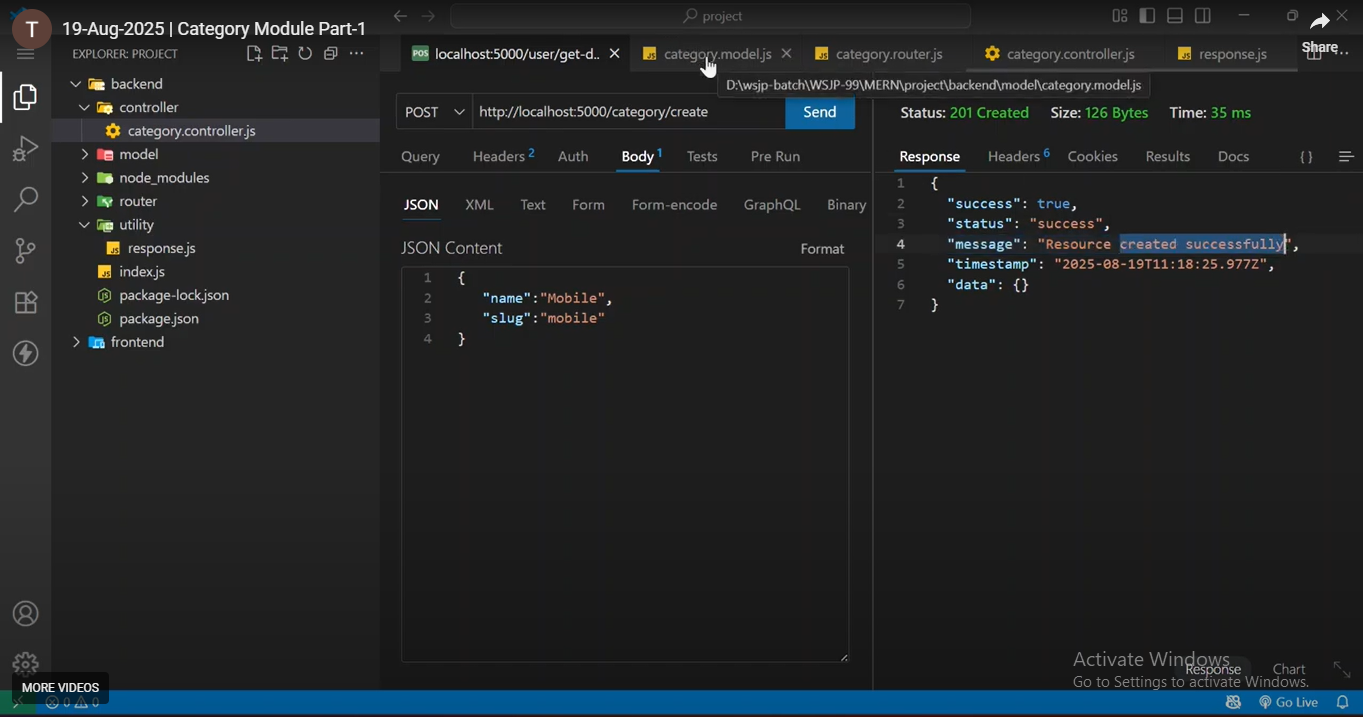
        }

    }

}

module.exports=category

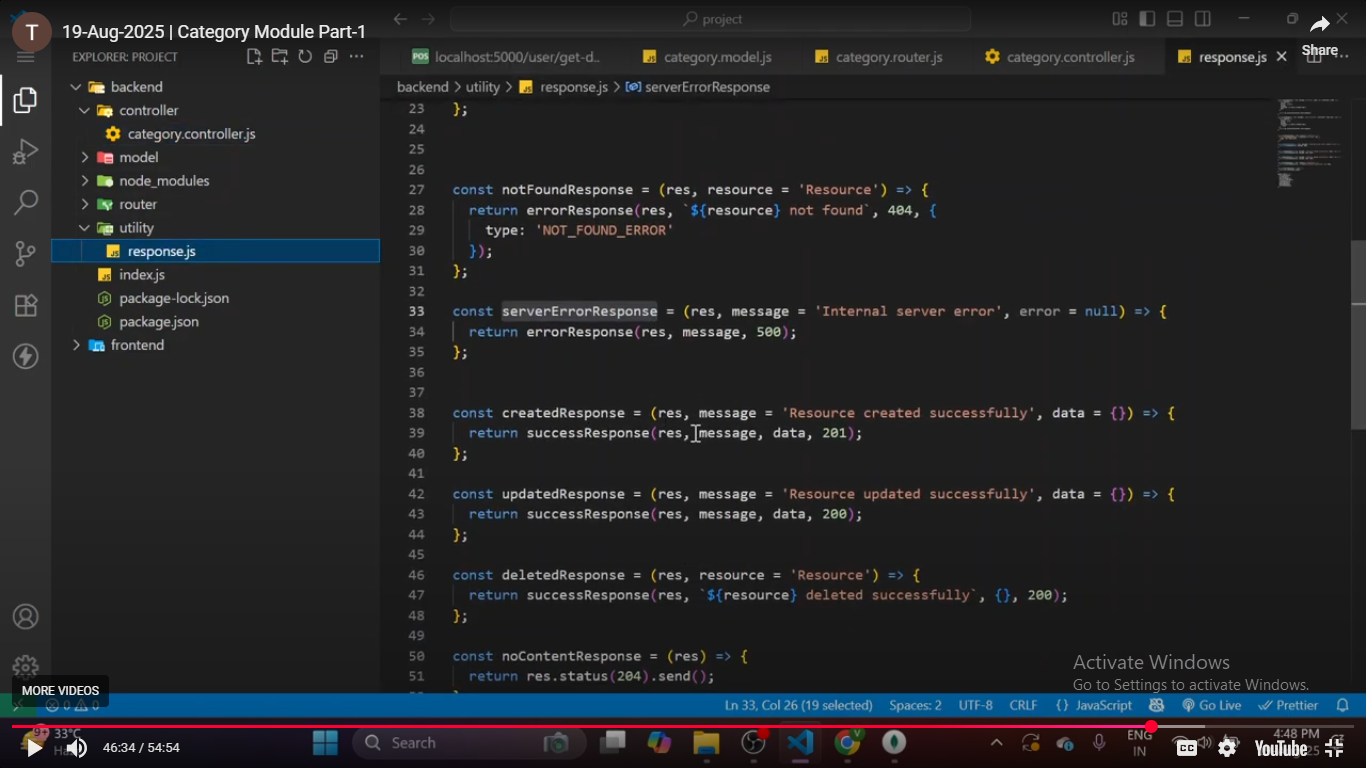
fir se category create karne par proper response aayega



Agar hum response me

“Resource created successfully” ki “Category created successfully” message dikhana chahate hai to

Pahale response.js me jao dekho kitane number hai message createdResponse()



Fir category.controll.js me jao createdResponse() me us number par message pass kar do

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse } = require('../utility/response')

const category = {

    async create(req, res) {

        try {

            if (!req.body.name || !req.body.slug) {

                return noContentResponse(res)

            }

            constcategory=awaitcategoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            awaitcategory.save()

            returncreatedResponse(res, "Category created successfully")

        } catch (error) {

            returnserverErrorResponse(res)

        }

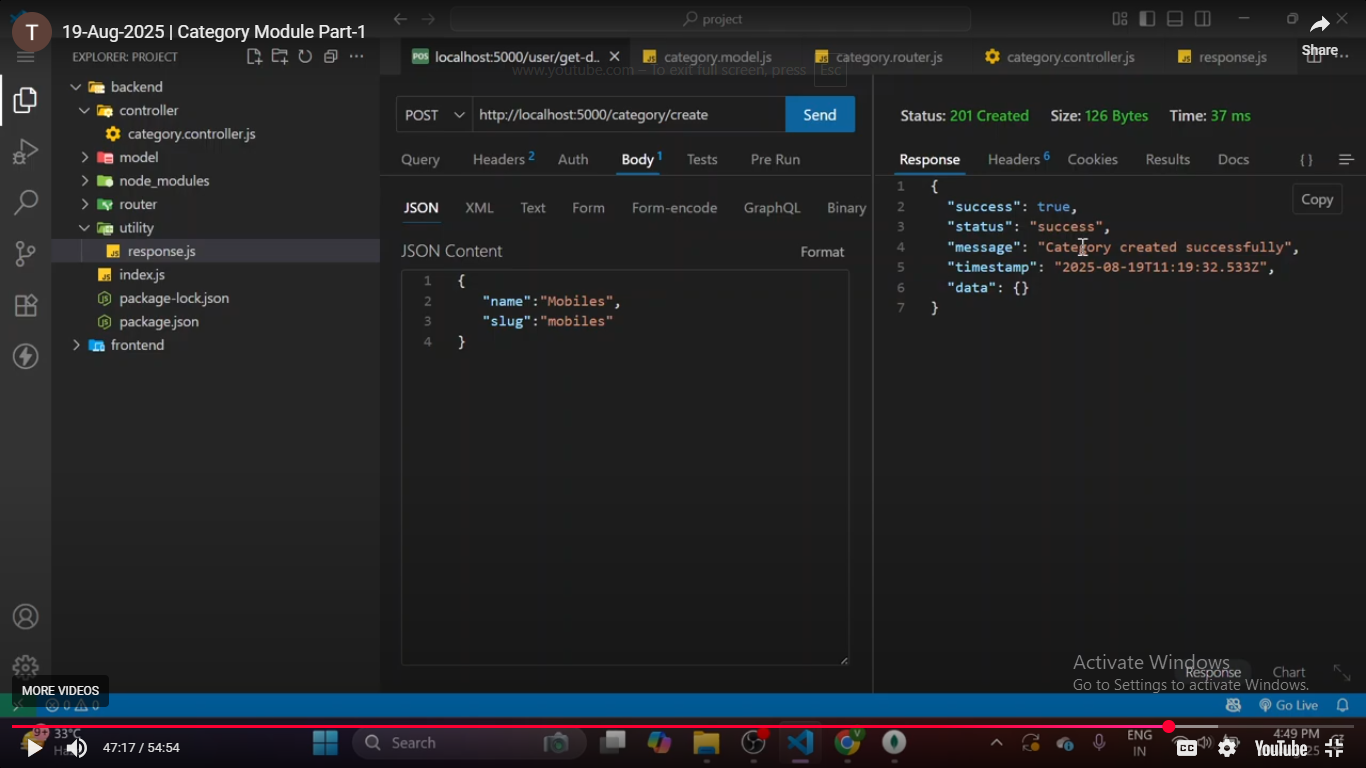
    }

}

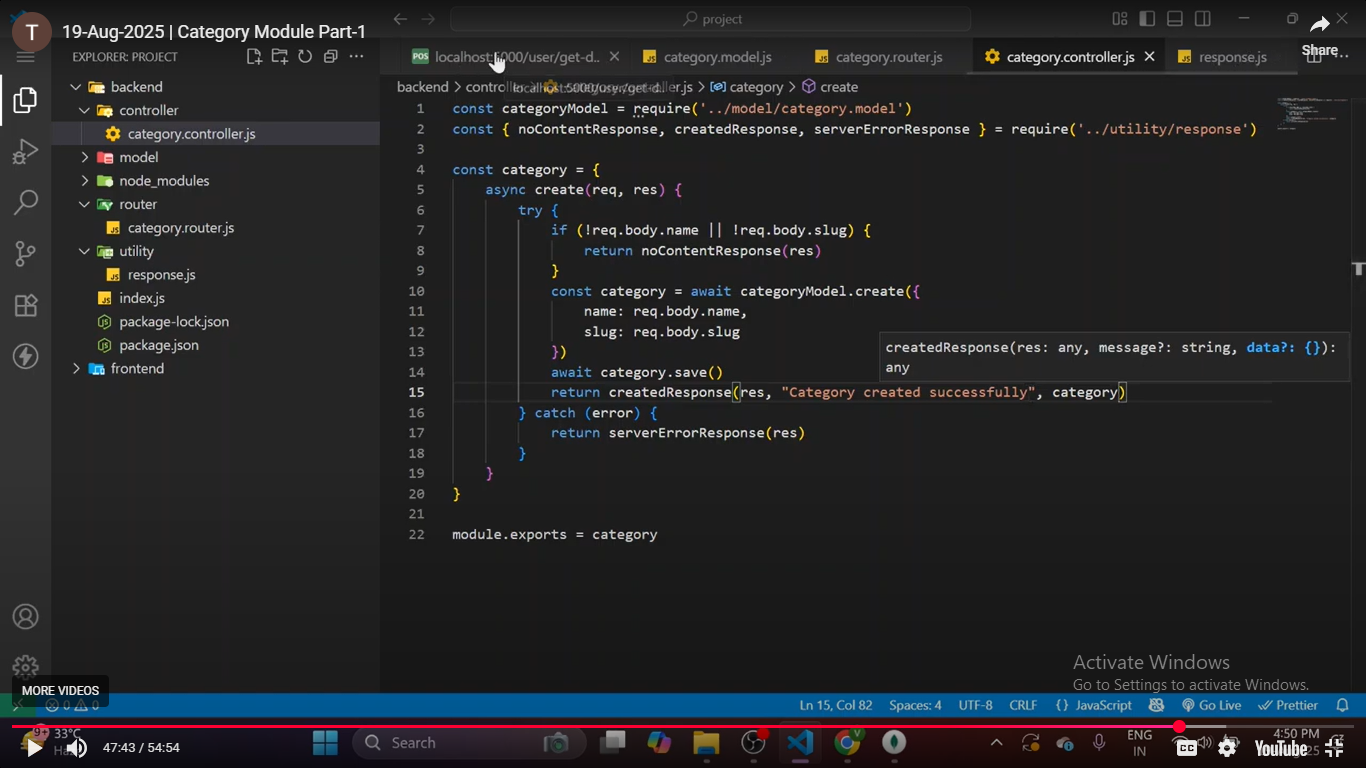
module.exports=category

now check on thunder ki category aayi hai ya nahi

lekin is bar category aur slug nayi honi chahiye



Agar jo data create huaa vo dekhana chahate hai vo bhi dekh sakte hai use bhejkar



Lekin hum yanha data ko pass nahi karenge testing kar sakte hai

Ab maanao aapne koi category pahale se bana li aur dobara usi category ko create karoge to internal server error dega

To ham ek logic lagana hai ki jo category hum create kar rahe hai vo pahale se database me hai ya nahi hai

.

constcategoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse } =require('../utility/response')

constcategory= {

    asynccreate(req, res) {

        try {

            if (!req.body.name || !req.body.slug) {

                returnnoContentResponse(res)

            }

            const existingItem=awaitcategoryModel.findOne({name: req.body.name})

            if(existingItem) {

                return  serverErrorResponse(res, "category already created")

            }

            const category=awaitcategoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            await category.save()

            return createdResponse(res, "Category created successfully")

        } catch (error) {

            return serverErrorResponse(res)

        }

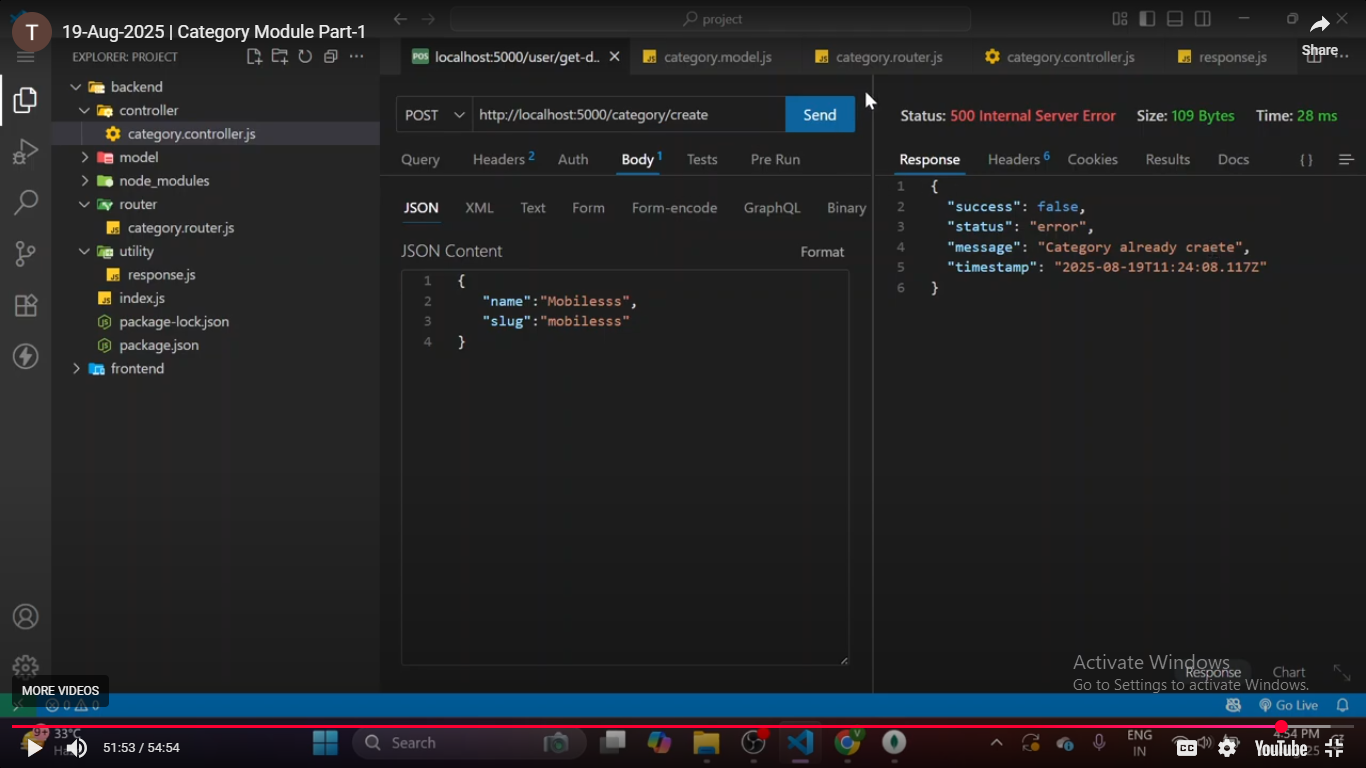
    }

}

module.exports=category

then check on thunder

agar different name ke saath try karoge to “category already created” message nahi aayega



Ab agar category create karne par “internal server error” aa raha hai to agar aap dekhana chahate hai ki error kis cheej ka hai to app dekh sakte hai catch() me serverErrorResponse(res, error.errmsg) karke dekh sakte hai

            if(existingItem) {

                return  serverErrorResponse(res, "category already created")

            }

Console me error message dekhne ke liye yah line hatani hogi

constcategoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse } =require('../utility/response')

constcategory= {

    asynccreate(req, res) {

        try {

            if (!req.body.name || !req.body.slug) {

                returnnoContentResponse(res)

            }

            const existingItem=awaitcategoryModel.findOne({name: req.body.name})

            if(existingItem) {

                return  serverErrorResponse(res, "category already created")

            }

            const category=awaitcategoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            await category.save()

            return createdResponse(res, "Category created successfully")

        } catch (error) {

            // console.log(error)

            return serverErrorResponse(res, error.errmsg)

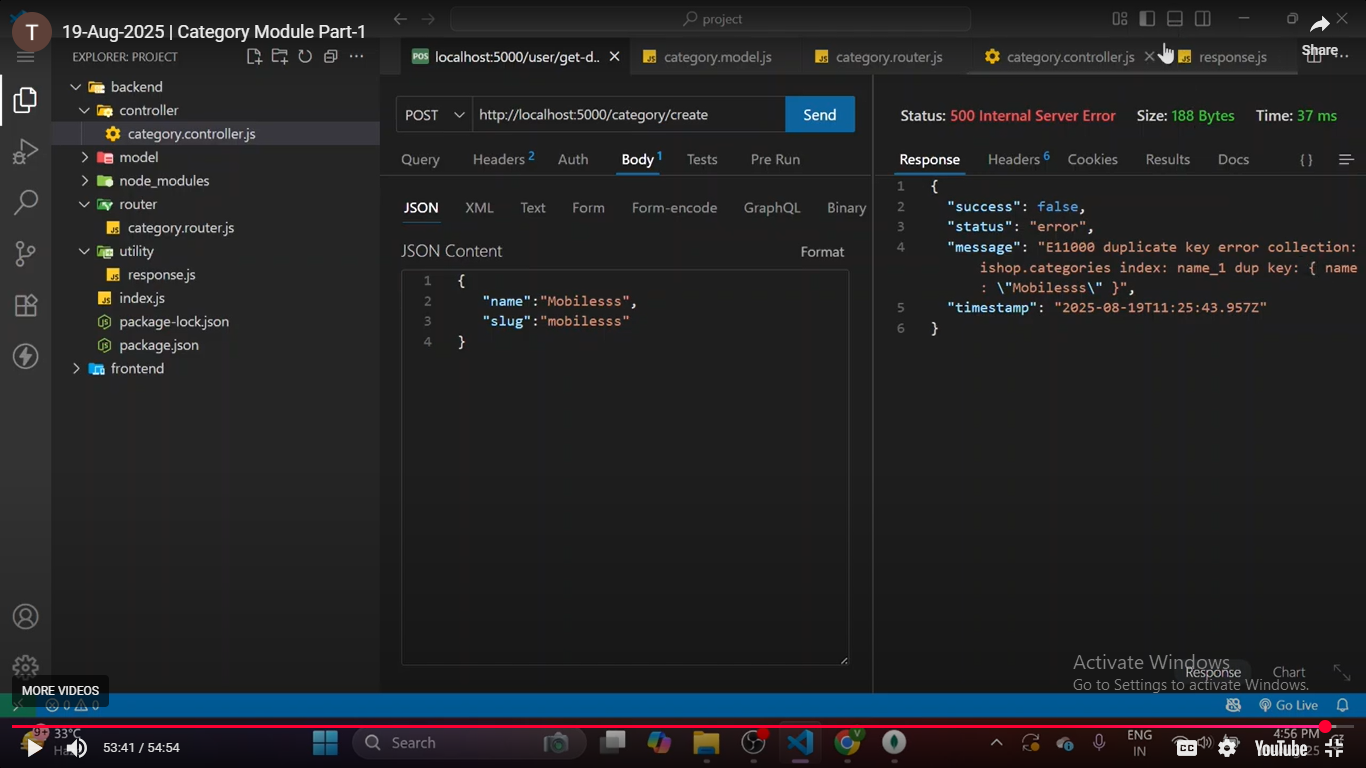
        }

    }

}

module.exports=category

now you can check on thunder



**20-08-25 day3**

Start frontend and backend

Node –watch index

Npm run dev

Read api bana hai

Category.router.js

const categoryRouter = require('express').Router()

const {create, read} =require('../controller/category.controller')

categoryRouter.post('/create', create)

categoryRouter.get('/', read)

module.exports=categoryRouter

category.controller.js

constcategoryModel=require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse } =require('../utility/response')

constcategory= {

    asynccreate(req, res) {

        try {

            if (!req.body.name||!req.body.slug) {

                returnnoContentResponse(res)

            }

            constexistingItem=awaitcategoryModel.findOne({name: req.body.name})

            if(existingItem) {

                return  serverErrorResponse(res, "category already created")

            }

            constcategory=awaitcategoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            awaitcategory.save()

            returncreatedResponse(res, "Category created successfully")

        } catch (error) {

            // console.log(error)

            returnserverErrorResponse(res, error.errmsg)

        }

    },

    asyncread(req, res) {

        try {

            constcategory=awaitcategoryModel.find()

            console.log(category)

            res.send("hello")

        } catch (error) {

            returnserverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports=category

abhi yah get api hai to ham ise kanhi bhi check kar sakte hai to hum ise url me check karenge

localhost:5000/category

browser par to hello aayega

aur backend ke console me jo category hamane abhi tak banai hai vo sabhi aa jayegi array of object

constcategoryModel=require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse } =require('../utility/response')

constcategory= {

    asynccreate(req, res) {

        try {

            if (!req.body.name||!req.body.slug) {

                returnnoContentResponse(res)

            }

            constexistingItem=awaitcategoryModel.findOne({name: req.body.name})

            if(existingItem) {

                return  serverErrorResponse(res, "category already created")

            }

            constcategory=awaitcategoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            awaitcategory.save()

            returncreatedResponse(res, "Category created successfully")

        } catch (error) {

            // console.log(error)

            returnserverErrorResponse(res, error.errmsg)

        }

    },

    asyncread(req, res) {

        try {

            constcategory=awaitcategoryModel.find()

            if(!category) errorResponse(res, "Category not found")

                returnsuccessResponse(res, "Category Found", category)

        } catch (error) {

            returnserverErrorResponse(res, error.errmsg)

        }

    }

}

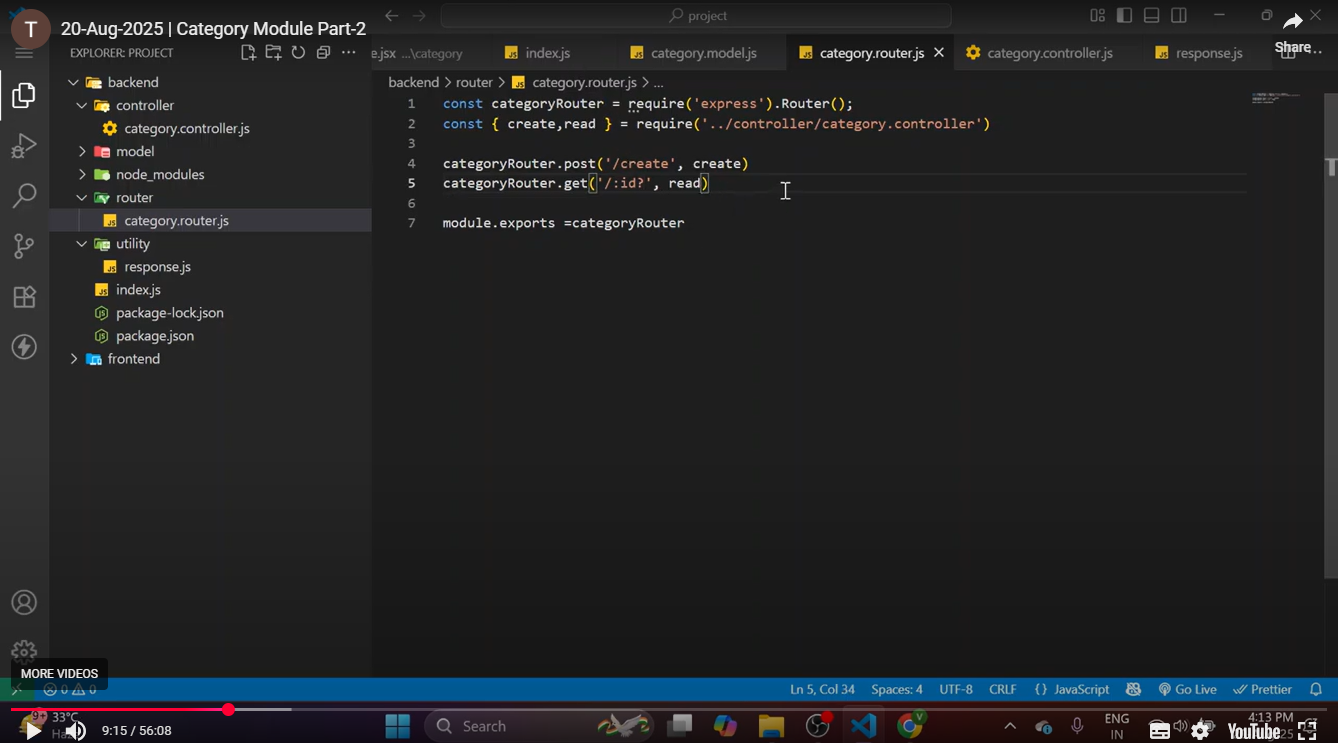
module.exports=category

ab api ko hit karenge url me

localhost:5000/category

to ham array of object mil jayega

agar category ki id jaati hai param me to keval usi id ka data return hona chahiye nahi to sabhi category ke data return hona chahiye



Update ke liye background color red lagaya hai

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const category = {

    async create(req, res) {

        try {

            if (!req.body.name || !req.body.slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({ name: req.body.name })

            if (existingItem) {

                return serverErrorResponse(res, "category already created")

            }

            const category = await categoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            await category.save()

            return createdResponse(res, "category creats successfully", category)

        } catch (error) {

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

console.log(req.params)

            const category = await categoryModel.find()

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = category

abhi hamne params ko console kiya hai

ab hum backend ke terminal me jayenge agar express ka purane version ka error aata hai to

purana express uninstall kar naya install kar le

npm uninstall express

npm i express@4

ab hamane params pass kiya hai to hum url me jayenge aur /category/category\_id denge to vahi id ka data dikhega

ab hamen id ke base par nikallana hai to hum

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const category = {

    async create(req, res) {

        try {

            if (!req.body.name || !req.body.slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({ name: req.body.name })

            if (existingItem) {

                return serverErrorResponse(res, "category already created")

            }

            const category = await categoryModel.create({

                name: req.body.name,

                slug: req.body.slug

            })

            await category.save()

            return createdResponse(res, "category creats successfully", category)

        } catch (error) {

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if(id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = category

yanha hamne id ke base par nikala hai

const id = req.params.id de denge

url me /category hi karoge all data dikhega nahi to keval id ka data dikhega

yanha hame do alag alag api nahi bani padati

destructuring se sort banan

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const category = {

    async create(req, res) {

        try {

const {name, slug} = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({ name }) //key value same name se hai

            if (existingItem) {

                return serverErrorResponse(res, "category already created")

            }

            const category = await categoryModel.create({

                name,

                slug

            })

            await category.save()

            return createdResponse(res, "category creats successfully", category)

        } catch (error) {

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if(id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = category

yanha hamari read api finish hoti hai

**status wali api aap ko hi banana hai**

api get karni hai

src/library name se folder banao ek file banao

api-calls.js

frontend me axios install karo

npm i axios

api-calls.js below code add karo

const getCategories = async () => {

  try {

    const response = await axios.get("http://localhost:5000/category")

    if (response.data) {

      return response.data

    } else {

      return null

    }

  } catch (error) {

    return null

    console.log(error)

  }

}

export { getCategories }

admin-group/admin/category/page.jsx isme hame ise get karna hai

note:- server side rendering par async laga sakte hai client side rendering par nahi laga sakte hai

// components/CategoryTable.jsx

import { getCategories } from '@/library/api-calls';

import { FiEdit, FiTrash2 } from 'react-icons/fi';

export default async function CategoryTable() {

  const categoryJSON = await getCategories()

  console.log(categoryJSON);

  // Dummy data (you can replace this with real API data)

  return (

    <div className="bg-white shadow rounded-lg p-6">

      {/\* Header \*/}

      <div className="flex items-center justify-between mb-4">

        <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

        <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

          + Add Category

        </button>

      </div>

      {/\* Table \*/}

      <div className="overflow-x-auto">

        <table className="min-w-full text-sm text-left">

          <thead className="bg-gray-50 text-gray-600 border-b">

            <tr>

              <th className="px-4 py-2">Image</th>

              <th className="px-4 py-2">Name</th>

              <th className="px-4 py-2">Slug</th>

              <th className="px-4 py-2 text-right">Actions</th>

            </tr>

          </thead>

          <tbody className="text-gray-700">

            {categories.map((cat) => (

              <tr key={cat.id} className="border-b hover:bg-gray-50">

                <td className="px-4 py-3">

                  <img

                    src={cat.image}

                    alt={cat.name}

                    className="w-12 h-12 object-cover rounded"

                  />

                </td>

                <td className="px-4 py-3">{cat.name}</td>

                <td className="px-4 py-3">{cat.slug}</td>

                <td className="px-4 py-3 text-right space-x-2">

                  <button className="text-blue-600 hover:text-blue-800 transition">

                    <FiEdit />

                  </button>

                  <button className="text-red-600 hover:text-red-800 transition">

                    <FiTrash2 />

                  </button>

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

}

Kuch get huaa hai ya nahi iske liye url me us path par jaana hai janha yah call huaa hai

localhost:3000/admin /category hit karo

console me dekho kya aa raha hai

**server null** aa rha hoga

yah isliye aa rha kyonki hamane **axios** ko install to kar liye lekin import nahi kiya **api-calls.js** me axios ko **import** karna hai

fir dobara url me api hit karoge to data aa jayega

import axios from "axios"

const getCategories = async () => {

  try {

    const response = await axios.get("http://localhost:5000/category")

    if (response.data) {

      return response.data

    } else {

      return null

    }

  } catch (error) {

    return null

    console.log(error)

  }

}

export { getCategories }

admin-group/admin/category/page.jsx is wapas jaana hai aur update karna hai

aur jo manual categories thi unko remove karna hai

ab aapki category table par vo sabhi catories print hongi jo aapne banayi thi

ab hum admincomponent/sideMenu me ek link lagaye jis par click karne par hum /admin/category table par pahunche

**Ab ek page bnao**

**Admin/category/add/page.jsx**

Isme hame ek form dikhana hai jisme category name hoga, slug hoga aur ek image rahegi, hame keval ui chahiye

Ise hum chat gpt se banwa lenge

**Admin/category/page.jsx**

Is par jo category add button hai us link laga denge route bane /**admin/category/add**

Jisase hum button par click karne par category add page par pahunch jayenge

Jab hamara form ban jayega tab hame slug banana hai jo name se hi banega

Admin/category/add

**Src/library** me Create a file helper.js usme cdoe likho aur **admin/category/add/page.jsx** me add get karo

Helper.js

function createSlug(text) {

    return text

        .toString()              // Ensure it's a string

        .trim()                  // Remove extra spaces

        .toLowerCase()           // Convert to lowercase

        .replace(/[^a-z0-9\s-]/g, '')  // Remove special chars

        .replace(/\s+/g, '-')          // Replace spaces with dashes

        .replace(/-+/g, '-');          // Replace multiple dashes with single

}

export { createSlug }

admin/category/add/page.js

'use client'

import { createSlug } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const data = {

            name: nameRef.current.value,

            slug: slugRef.current.value

        }

        axios.post("http://localhost:5000/category/create", data).then(

            (response)=> {

                console.log(response.data);

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}

>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file" />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Index.js me cors add karna hai

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const server = express();

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect("mongodb://localhost:27017/", {dbName: "rishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

**22-08-25 day 4**

Backend and frontend dono ko ek hi command se start karna

Iske liye frontend and backend ke bahar ise install karna hai

npm i [concurrently@6.1.0](mailto:concurrently@6.1.0)

or

npm i concurrently

fir package.json me kuch updates karna hai

{

  "dependencies": {

    "concurrently": "^9.2.0"

  },

  "scripts": {

    "start": "concurrently \"cd backend && node --watch index.js\" \"cd frontend && npm run dev\""

  }

}

**Ab admin/category/add/page.jsx**

response.data me internal server error bhi aa sakta hai aur

success hone par hi ek message print karwana hai aur success hone par hi ek form ko reset karwana hoga taaki dobara se dala ja sake

'use client'

import { createSlug } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const data = {

            name: nameRef.current.value,

            slug: slugRef.current.value

        }

        axios.post("http://localhost:5000/category/create", data).then(

            (response)=> {

                // console.log(response.data.success);

                if(response.data.success); {

                }

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file" />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

aur ek toastifier lagayenge

front end me toastify add karo

npm i react-toastify

knyoki hame toastify bahut jagah chahiye siliye hum toastify ko helper.js me rakhenge

toastify ki github repo par jakar hum aur extra things add kar sakte hai

position, type aur theme change kar lo

playground wale toast container ko copy kar lo

ese page par rakhana hai taki testify ko kanhi se bhi use kiya ja sake

hum admin/layout me rakh rahe hai

toasty container ke liye update

import { ToastContainer, toast } from "react-toastify";

const notify = (msg, flag) => toast(msg, {type: flag ? 'success' : 'error'})

function createSlug(text) {

    return text

        .toString()              // Ensure it's a string

        .trim()                  // Remove extra spaces

        .toLowerCase()           // Convert to lowercase

        .replace(/[^a-z0-9\s-]/g, '')  // Remove special chars

        .replace(/\s+/g, '-')          // Replace spaces with dashes

        .replace(/-+/g, '-');          // Replace multiple dashes with single

}

export { createSlug, notify }

import { Geist, Geist\_Mono } from "next/font/google";

import "./../../globals.css";

import SideMenu from "@/components/admin-components/SideMenu";

import Header from "@/components/admin-components/Header";

 import { ToastContainer} from 'react-toastify';

const geistSans = Geist({

  variable: "--font-geist-sans",

  subsets: ["latin"],

});

const geistMono = Geist\_Mono({

  variable: "--font-geist-mono",

  subsets: ["latin"],

});

export const metadata = {

  title: "admin",

  description: "Generated by create next app",

};

export default function RootLayout({ children }) {

  return (

    <html lang="en">

      <body

        className={`${geistSans.variable} ${geistMono.variable} antialiased`}

      >

        <Header />

        <div className="flex">

          <ToastContainer

            position="top-right"

            autoClose={5000}

            hideProgressBar={false}

            newestOnTop={false}

            closeOnClick={false}

            rtl={false}

            pauseOnFocusLoss

            draggable

            pauseOnHover

            theme="light"

          />

          <SideMenu />

          <div className="flex-1">

            {children}

          </div>

        </div>

      </body>

    </html>

  );

}

 transition={Bounce}

isko hata dena hai them=”light” ke niche se

admin/category/add/page.jsx

'use client'

import { createSlug, notify } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const data = {

            name: nameRef.current.value,

            slug: slugRef.current.value

        }

        axios.post("http://localhost:5000/category/create", data).then(

            (response)=> {

                // console.log(response.data.success);

                notify(response.data.message, {flag: response.data.success})

                if(response.data.success); {

                     nameRef.current.value = ""

                     slugRef.current.value = ""

                }

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file" />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Already category create hai to axios error aayegi

Ab category me image upload karwani hai

Json ke andar media nahi ja sakta jese image, audio, file, etc

'use client'

import { createSlug, notify } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // }

        console.log(e.target.category\_image.files);

        return

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        axios.post("http://localhost:5000/category/create", formData).then(

            (response)=> {

                // console.log(response.data.success);

                notify(response.data.message, {flag: response.data.success})

                if(response.data.success); {

                     nameRef.current.value = ""

                     slugRef.current.value = ""

                }

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                             />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Files ki list aa jayegi new category create karne par image ke saath

'use client'

import { createSlug, notify } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axios.post("http://localhost:5000/category/create", formData).then(

            (response)=> {

                // console.log(response.data.success);

                notify(response.data.message, {flag: response.data.success})

                if(response.data.success); {

                     nameRef.current.value = ""

                     slugRef.current.value = ""

                }

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                             />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Category.controll.js

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse  } = require('../utility/response')

const category = {

    async create(req, res) {

        try {

            console.log(req.body);

            console.log(req.files);

            const {name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if(existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const category = await categoryModel.create({

                name,

                slug

            })

            await category.save()

            return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if(id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if(!category) errorResponse(res, "Category not found")

                return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res,  error.msg)

        }

    }

}

module.exports = category

ab wapas category name, slug aur image ke saath category create karoge to backend ke console me undefined {} aayega

yah isliy aa rha hai kyonki data ko pass karne ke liye index.js me json chaiye lekin binary data jese image ko pass karne ke liye middele ware chahiye jise install karna hoga

backend me

npm i [express-fileupload@1.3.0](mailto:express-fileupload@1.3.0)

import karo

import karne ke baad ise use karna

note: midlleware do tarah se laga sakte hai global middle ware and local middle ware

to janha ise use karna hai vanha ise lagao

global middleware mean = all route

local middleware mean = particular route par

global index.js me lagega

lekin hame create karte time hi lagana hai to route me jaana hoga

category.route.js

const categoryRouter = require('express').Router()

const {create, read} = require('../controller/category.controller')

const fileupload = require('express-fileupload')

categoryRouter.post('/create',fileupload({createParentPath: true}), create)

categoryRouter.get('/:id?', read)

module.exports = categoryRouter

public/images/category is folder sabhi image aayegi

ab app fir new category create kroge image ke saath to aapko (npm start ) ke console me image ka object aa jayega

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const destination = 'public/images/category/' + categoryImg.name

            categoryImg.mv(

                destination,

             async   (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image: categoryImg.name

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    }

}

module.exports = category

abhi jab hum image le rahe hai agar new image ka name purani image se match hota hai to purani wali image replace ho raha hai

iske liye backend me helper.js bnao utility folder me

usme unique image name se function bana lo

aur category.controll.js me call karo

function createUniqueName(originalName) {

    const uniqueId = Date.now().toString(36) + Math.random().toString(36)

    return `${uniqueId + originalName}`;

}

module.exports = {createUniqueName}

ab new category create karoge to unique image name generate hoga backend ke console me

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const  {createUniqueName} = require('../utility/helper')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            console.log(image);

            return

            const destination = 'public/images/category/' + ategoryImg.name

            categoryImg.mv(

                destination,

             async   (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image: categoryImg.name

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    }

}

module.exports = category

ab kitane bhi image save karo different name se jayegi

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const  {createUniqueName} = require('../utility/helper')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

             async   (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image: name

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    }

}

module.exports = category

complete

**23-08-25 day 5**

Jab aap is path ko url me daloge to aapko backend se image get nahi hogi

<http://localhost:5000/public/images/category/jp.jpg>

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const server = express();

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use(express.static('public'));

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect("mongodb://localhost:27017/", {dbName: "rishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

<http://localhost:5000/images/category/image_name.jpg>

jo image database me save vo aa jayegi

ab admin/category/add/page.jsx

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    }

}

module.exports = category

category table image me ise jarur add kare kyoki path alag alag ho sakte hai

src={`http://localhost:5000/images/category/${cat.image}`}

ab category table par images dikhane lagegi

4:00

Frontend me src ke folder ke bahar .env file bnaani hai aur niche diya code likhana hai

.env file ko src ke bahar rakhana hai

NEXT\_PUBLIC\_API\_BASE\_URL=http://localhost:5000/

Ab admin category table me env code ko add karna hai

import React from "react";

import { getCategories } from "@/library/api-calls";

import Link from "next/link";

export default async function Page() {

    // Fetch Categories

    const categoryJSON = await getCategories();

    const categories = categoryJSON?.data || []; // API response ke andar data array aayega

    return (

        <div className="p-6">

            {/\* Top Bar with Title and Button \*/}

            <div className="flex justify-between items-center mb-6">

                <h1 className="text-2xl font-bold">Category Table</h1>

                <Link href="/admin/category/add">

                    <button className="bg-blue-600 text-white px-4 py-2 rounded-lg hover:bg-blue-700">

                        + Add Category

                    </button>

                </Link>

            </div>

            {/\* Category Table \*/}

            <div className="overflow-x-auto">

                <table className="w-full border border-gray-200 rounded-lg">

                    <thead className="bg-gray-100">

                        <tr>

                            <th className="px-4 py-2 border">#</th>

                            <th className="px-4 py-2 border">Image</th>

                            <th className="px-4 py-2 border">Name</th>

                            <th className="px-4 py-2 border">Slug</th>

                            <th className="px-4 py-2 border">Created At</th>

                        </tr>

                    </thead>

                    <tbody>

                        {categories.length > 0 ? (

                            categories.map((cat, index) => (

                                <tr key={cat.\_id} className="hover:bg-gray-50">

                                    <td className="px-4 py-2 border text-center">{index + 1}</td>

                                    {/\* Image Column \*/}

                                    <td className="px-4 py-2 border text-center">

                                        {cat.image ? (

                                            <img

                                                src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${cat.image}`}

                                                alt={cat.name}

                                                className="w-16 h-16 object-cover rounded-md mx-auto"

                                            />

                                        ) : (

                                            <span className="text-gray-400">No Image</span>

                                        )}

                                    </td>

                                    <td className="px-4 py-2 border">{cat.name}</td>

                                    <td className="px-4 py-2 border">{cat.slug}</td>

                                    <td className="px-4 py-2 border">

                                        {new Date(cat.createdAt).toLocaleDateString()}

                                    </td>

                                </tr>

                            ))

                        ) : (

                            <tr>

                                <td

                                    colSpan="5"

                                    className="px-4 py-4 border text-center text-gray-500"

                                >

                                    No categories found

                                </td>

                            </tr>

                        )}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

Ab hum env code ko axios me use karenge iske liye **helper.js** me update karenge

import { ToastContainer, toast } from "react-toastify";

const notify = (msg, flag) => toast(msg, {type: flag ? 'success' : 'error'})

import axios from "axios";

const axiosInstance = axios.create({

    baseURL: process.env.NEXT\_PUBLIC\_API\_BASE\_URL

})

function createSlug(text) {

    return text

        .toString()              // Ensure it's a string

        .trim()                  // Remove extra spaces

        .toLowerCase()           // Convert to lowercase

        .replace(/[^a-z0-9\s-]/g, '')  // Remove special chars

        .replace(/\s+/g, '-')          // Replace spaces with dashes

        .replace(/-+/g, '-');          // Replace multiple dashes with single

}

export { createSlug, notify, axiosInstance }

isase hame yah benefit hoga ki hame axios ko import karne ke jarurat nahi hogi

hum category/add/page.jsx me ise update karenge janha janha axios import kar rakha use import karne ki jarurat nahi hogi keva **axiosInstance** is import kar lenge **helper**.js se

'use client'

import { createSlug, notify, axiosInstance } from '@/library/helper';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.post("category/create", formData).then(

            (response)=> {

                // console.log(response.data.success);

                notify(response.data.message, {flag: response.data.success})

                if(response.data.success); {

                     nameRef.current.value = ""

                     slugRef.current.value = ""

                }

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                             />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Ab api calls me bhi ese hi karenge aixos ko import kar rakha hai use hata denge **axiosInstance** use karenge

import { axiosInstance } from "./helper"

const getCategories = async () => {

  try {

    const response = await axiosInstance.get("category")

    if (response.data) {

      return response.data

    } else {

      return null

    }

  } catch (error) {

    return null

    console.log(error)

  }

}

export { getCategories }

front ki .env file kaam kar rahi hai

jo .env file ka use hamane frontend me kiya hai ab .env file ka use hum backend me bhi karenge lekin backed me .env file inbuilt nahi aati hai **dotenv** packend use install karna hota hai

frontend me inbuilt aati hai isliye install nahi kiya

backend me install karna hai

npm i dotenv

index.js me add karna hai

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const server = express()

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use(express.static('public'));

server.listen(5000, ()=> {

    console.log("server Running PORT 5000");

    mongoose.connect('mongodb://localhost:27017/', {dbName: "practice-rishop4"}).then(()=> {

        console.log("Database is connected");

    }).then(()=> {

        console.log("Database is unable to connect");

    })

})

Backend me .**env** file create karo aur niche wala code add karo jisme mongodb ki connecting string hai jo index se li hai

DATABASE\_URL = mongodb://localhost:27017/

Aur index.js me string ko remove kar env code dala hai

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const server = express()

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use(express.static('public'));

server.listen(5000, ()=> {

    console.log("server Running PORT 5000");

    mongoose.connect(process.env.DATABASE\_URL, {dbName: "practice-rishop4"}).then(()=> {

        console.log("Database is connected");

    }).then(()=> {

        console.log("Database is unable to connect");

    })

})

Yanah backend ki env file complete hoti hia

**Delete ki api banayenge**

Category route ko update karenge

const categoryRouter = require('express').Router()

const {create, read, deleteById} = require('../controller/category.controller')

const fileupload = require('express-fileupload')

categoryRouter.post('/create',fileupload({createParentPath: true}), create)

categoryRouter.get('/', read)

categoryRouter.delete('/delete/:id', deleteById)

module.exports = categoryRouter

ab jab hamane category controller se **deleteById** liye to usme jayenge aur delete ki api banayenge

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

  async create(req, res) {

    try {

      // console.log(req.body);

      // console.log(req.files);

      const categoryImg = req.files.image

      const { name, slug } = req.body

      if (!name || !slug) {

        return noContentResponse(res)

      }

      const existingItem = await categoryModel.findOne({

        name

      })

      if (existingItem) {

        return serverErrorResponse(res, "Category already created", 409)

      }

      const image = createUniqueName(categoryImg.name)

      const destination = 'public/images/category/' + image

      categoryImg.mv(

        destination,

        async (error) => {

          if (error) {

            return errorResponse(res, "File not found")

          } else {

            const category = await categoryModel.create({

              name,

              slug,

              image

            })

            await category.save()

            return createdResponse(res, "category created successfully")

          }

        }

      )

      // const category = await categoryModel.create({

      //     name,

      //     slug

      // })

      // await category.save()

      // return createdResponse(res, "category created successfully")

    } catch (error) {

      console.log(error);

      return serverErrorResponse(res)

    }

  },

  async read(req, res) {

    try {

      const id = req.params.id

      let category = null

      if (id) {

        category = await categoryModel.findById(id)

      } else {

        category = await categoryModel.find()

      }

      if (!category) errorResponse(res, "Category not found")

      return successResponse(res, "Category Found", category)

    } catch (error) {

      return serverErrorResponse(res, error.msg)

    }

  },

  async deleteByID(req, res) {

    try {

      const id = req.params.id;

      const existingCat = await categoryModel.findById(id);

      if (existingCat) {

        fs.unlinkSync('./public/images/category/' + existingCat.name)

      }

      await categoryModel.findByIdAndDelete(id)

      return deletedResponse(res)

      return successResponse(res, "Category Found", category)

    } catch (error) {

      return serverErrorResponse(res, error.errmsg)

    }

  }

}

module.exports = category

upar hamari delete ki api ka code hai

hum status ki api banayenge

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

  async create(req, res) {

    try {

      // console.log(req.body);

      // console.log(req.files);

      const categoryImg = req.files.image

      const { name, slug } = req.body

      if (!name || !slug) {

        return noContentResponse(res)

      }

      const existingItem = await categoryModel.findOne({

        name

      })

      if (existingItem) {

        return serverErrorResponse(res, "Category already created", 409)

      }

      const image = createUniqueName(categoryImg.name)

      const destination = 'public/images/category/' + image

      categoryImg.mv(

        destination,

        async (error) => {

          if (error) {

            return errorResponse(res, "File not found")

          } else {

            const category = await categoryModel.create({

              name,

              slug,

              image

            })

            await category.save()

            return createdResponse(res, "category created successfully")

          }

        }

      )

      // const category = await categoryModel.create({

      //     name,

      //     slug

      // })

      // await category.save()

      // return createdResponse(res, "category created successfully")

    } catch (error) {

      console.log(error);

      return serverErrorResponse(res)

    }

  },

  async read(req, res) {

    try {

      const id = req.params.id

      let category = null

      if (id) {

        category = await categoryModel.findById(id)

      } else {

        category = await categoryModel.find()

      }

      if (!category) errorResponse(res, "Category not found")

      return successResponse(res, "Category Found", category)

    } catch (error) {

      return serverErrorResponse(res, error.msg)

    }

  },

  async deleteByID(req, res) {

    try {

      const id = req.params.id;

      const existingCat = await categoryModel.findById(id);

      if (existingCat) {

        fs.unlinkSync('./public/images/category/' + existingCat.name)

      }

      await categoryModel.findByIdAndDelete(id)

      return deletedResponse(res)

      return successResponse(res, "Category Found", category)

    } catch (error) {

      return serverErrorResponse(res, error.errmsg)

    }

  },

  async status(req, res) {

    try {

      const id = req.params.id;

      const category = await categoryModel.findById(id);

      await categoryModel.findByIdAndUpdate(

        id,

        { $set: { status: !category.status } }

      )

      return updatedResponse(res, "Status Update")

    } catch (error) {

      return serverErrorResponse(res, error.errmsg)

    }

  },

}

module.exports = category

status ki api ko router me update kar denge

const categoryRouter = require('express').Router()

const {create, read, deleteById, status} = require('../controller/category.controller')

const fileupload = require('express-fileupload')

categoryRouter.post('/create',fileupload({createParentPath: true}), create)

categoryRouter.get('/', read)

categoryRouter.delete('/delete/:id', deleteById)

categoryRouter.patch('/status/:id', status)

module.exports = categoryRouter

website-component me **deleteButton** bana lenge aur code write karenge

'use client'

import { axiosInstance, notify } from '@/library/helper'

import { useRouter } from 'next/navigation'

import React from 'react'

export default function DeleteBtn({ id }) {

  const router = useRouter()

  function deleteHandler() {

    axiosInstance.delete(`delete/${id}`).then(

      (response) => {

        notify(response.data.message, response.data.success)

        if (response.data.success) {

          router.refresh()

        }

      }

    ).catch((error) => {

      console.log(error)

    })

  }

  return (

    <button onClick={deleteHandler} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

      Delete

    </button>

  )

}

**DeleteBtn**

'use client'

import { axiosInstance, notify } from '@/library/helper'

import { useRouter } from 'next/navigation'

import React from 'react'

export default function DeleteBtn({ id }) {

  const router = useRouter()

  function deleteHandler() {

    axiosInstance.delete(`category/delete/${id}`).then(

      (response) => {

        notify(response.data.message, response.data.success)

        if (response.data.success) {

          router.refresh()

        }

      }

    ).catch((error) => {

      console.log(error)

    })

  }

  return (

    <button onClick={deleteHandler} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

      Delete

    </button>

  )

}

**StatusBtn**

'use client'

import React from 'react'

import { axiosInstance, notify } from '@/library/helper'

import { Router } from 'next/navigation'

import { useRouter } from 'next/navigation'

export default function StatusBtn({ id, status }) {

    const router = useRouter()

    function statusHandler() {

        axiosInstance.patch(`category/status/${id}`).then(

            (response) => {

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    router.refresh()

                }

            }

        ).catch((error) => {

            console.log(error)

        })

    }

    return (

        <button onClick={statusHandler} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

            {

                status ?

                    "Active"

                    :

                    "Inactive"

            }

        </button>

    )

}

**Router**

const categoryRouter = require('express').Router()

const {create, read, deleteById, status} = require('../controller/category.controller')

const fileupload = require('express-fileupload')

categoryRouter.post('/create',fileupload({createParentPath: true}), create)

categoryRouter.get('/:id?', read)

categoryRouter.delete('/delete/:id', deleteById)

categoryRouter.patch('/status/:id', status)

module.exports = categoryRouter

**category controller**

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingCat = await categoryModel.findById(id);

            if (existingCat) {

                fs.unlinkSync(`public/images/category/${existingCat.image}`)

            }

            await categoryModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const category = await categoryModel.findById(id);

            await categoryModel.findByIdAndUpdate(

                id,

                { $set: { status: !category.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

}

module.exports = category

**category table**

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getCategories } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit, FiTrash2 } from 'react-icons/fi';

export default async function CategoryTable() {

    const categories = await getCategories()

    console.log(categories);

    // Dummy data (you can replace this with real API data)

    return (

        <div className="bg-white shadow rounded-lg p-6">

            {/\* Header \*/}

            <div className="flex items-center justify-between mb-4">

                <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

                <Link href="/admin/categoryTable/add">

                    <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

                        + Add Category

                    </button>

                </Link>

            </div>

            {/\* Table \*/}

            <div className="overflow-x-auto">

                <table className="min-w-full text-sm text-left">

                    <thead className="bg-gray-50 text-gray-600 border-b">

                        <tr>

                            <th className="px-4 py-2">Image</th>

                            <th className="px-4 py-2">Name</th>

                            <th className="px-4 py-2">Slug</th>

                            <th className="px-4 py-2 text-right">Actions</th>

                        </tr>

                    </thead>

                    <tbody className="text-gray-700">

                        {categories.map((cat) => (

                            <tr key={cat.id} className="border-b hover:bg-gray-50">

                                <td className="px-4 py-3">

                                    <img

                                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${cat.image}`}

                                        alt={cat.name}

                                        className="w-12 h-12 object-cover rounded"

                                    />

                                </td>

                                <td className="px-4 py-3">{cat.name}</td>

                                <td className="px-4 py-3">{cat.slug}</td>

                                <td className="px-4 py-3 text-right space-x-2">

                                    <button className="text-blue-600 hover:text-blue-800 transition">

                                        <FiEdit />

                                    </button>

                                    <StatusBtn status={cat.status} id={cat.\_id} />

                                    <DeleteBtn id={cat.\_id} />

                                    <button className="text-red-600 hover:text-red-800 transition">

                                        <FiTrash2 />

                                    </button>

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

**25-08-25** **day6**

Ab edit ki api banayenge

categoryTable me edit folder bnana hai aur usme **[category\_id]** name folder banana hai fir usme page.jsx bnani hai

add/page.jsx ka pura code copy karna hai aur **edit/[category\_id]/page.j**sx me paste karna hai

**function name categoryEdit**

categoryTable me edit par link lagawana jisase edit par click karne par edit wala page open hona chahiye

**categoryEdit page niche hai**

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getCategories } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function CategoryTable() {

    const categories = await getCategories()

    console.log(categories);

    // Dummy data (you can replace this with real API data)

    return (

        <div className="bg-white shadow rounded-lg p-6">

            {/\* Header \*/}

            <div className="flex items-center justify-between mb-4">

                <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

                <Link href="/admin/categoryTable/add">

                    <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

                        + Add Category

                    </button>

                </Link>

            </div>

            {/\* Table \*/}

            <div className="overflow-x-auto">

                <table className="min-w-full text-sm text-left">

                    <thead className="bg-gray-50 text-gray-600 border-b">

                        <tr>

                            <th className="px-4 py-2">Image</th>

                            <th className="px-4 py-2">Name</th>

                            <th className="px-4 py-2">Slug</th>

                            <th className="px-4 py-2 text-right">Actions</th>

                        </tr>

                    </thead>

                    <tbody className="text-gray-700">

                        {categories.map((cat) => (

                            <tr key={cat.id} className="border-b hover:bg-gray-50">

                                <td className="px-4 py-3">

                                    <img

                                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${cat.image}`}

                                        alt={cat.name}

                                        className="w-12 h-12 object-cover rounded"

                                    />

                                </td>

                                <td className="px-4 py-3">{cat.name}</td>

                                <td className="px-4 py-3">{cat.slug}</td>

                                <td className="px-4 py-3 text-right space-x-2">

                                    <Link href={`/admin/categoryTable/edit/${cat.\_id}`}>

                                        <button className="text-blue-600 hover:text-blue-800 cursor-pointer transition">

                                            <FiEdit />

                                        </button>

                                    </Link>

                                    <StatusBtn status={cat.status} id={cat.\_id} />

                                    <DeleteBtn id={cat.\_id} />

                                    <button className="text-red-600 hover:text-red-800 transition">

                                    </button>

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

**Api-calls me update karna**

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) `${API}/${id}`

        console.log(API);

        // http://localhost:5000/category/

        // http://localhost:5000/category/68b00eca8029bc625c33343f

        const response = await axiosInstance.get(API)

        console.log(response.data, "resp");

        if (response) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        return null

        console.log(error)

    }

}

export { getCategories }

**edit page.jsx me update karna hai**

'use client'

import { getCategories } from '@/library/api-calls';

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useEffect, useRef, useState } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function categoryEdit({ params }) {

    const [categories, setCategories] = useState({})

    const nameRef = useRef()

    const slugRef = useRef()

    async function getCategory() {

        const categoryJSON = await getCategories(params.category\_id)

        setCategories(categoryJSON.data)

    }

    useEffect(() => {

        getCategory()

    }, [params.category\_id])

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    console.log("category", categories);

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.post("category/create", formData).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        defaultValue={category.name}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        defaultValue={category.slug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${category.image}`}

                        alt={category.name}

                        className="h-10 w-10 rounded objec t-cover shadow"

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Api-calls ko update karna hai**

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) `${API}/${id}`

        console.log(API);

        // http://localhost:5000/category/

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //inko browser par check kar sakte hai

        const response = await axiosInstance.get(API)

        console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        return null

        console.log(error)

    }

}

export { getCategories }

**ab fir se edit page.jsx ko update karna hai**

'use client'

import { getCategories } from '@/library/api-calls';

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useEffect, useRef, useState } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function categoryEdit({ params }) {

    const [categories, setCategories] = useState({})

    const nameRef = useRef()

    const slugRef = useRef()

    async function getCategory() {

        const categoryJSON = await getCategories(params.category\_id)

        console.log(categoryJSON.data, "data");

        setCategories(categoryJSON.data)

    }

    useEffect(() => {

        getCategory()

    }, [params.category\_id])

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    console.log("category", categories);

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.post("category/create", formData).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        defaultValue={category.name}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        defaultValue={category.slug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${category.image}`}

                        alt={category.name}

                        className="h-10 w-10 rounded objec t-cover shadow"

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Ab api-calls me update karna hai**

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        console.log(API);

        // http://localhost:5000/category/

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //inko browser par check kar sakte hai

        const response = await axiosInstance.get(API)

        console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        return null

        console.log(error)

    }

}

export { getCategories }

**ab category edit ko update karna hai**

'use client'

import { getCategories } from '@/library/api-calls';

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useEffect, useRef, useState } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function categoryEdit({ params }) {

    const [categories, setCategories] = useState({})

    const nameRef = useRef()

    const slugRef = useRef()

    async function getCategory() {

        const categoryJSON = await getCategories(params?.category\_id)

        console.log(categoryJSON.data, "data");

        setCategories(categoryJSON.data)

    }

    useEffect(() => {

        getCategory()

    }, [params.category\_id])

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    console.log("category", categories);

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.post("category/create", formData).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        defaultValue={category.name}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        defaultValue={category.slug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${category.image}`}

                        alt={category.name}

                        className="h-10 w-10 rounded objec t-cover shadow"

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Ab fir se category edit ko update karna hia**

'use client'

import { getCategories } from '@/library/api-calls';

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useEffect, useRef, useState } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function categoryEdit({ params }) {

    const [categories, setCategories] = useState({})

    const nameRef = useRef()

    const slugRef = useRef()

    async function getCategory() {

        const categoryJSON = await getCategories(params?.category\_id)

        console.log(categoryJSON.data, "data");

        setCategories(categoryJSON.data)

    }

    useEffect(() => {

        getCategory()

    }, [params.category\_id])

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.put("category/update", formData).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        defaultValue={category.name}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        defaultValue={category.slug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${category.image}`}

                        alt={category.name}

                        className="h-10 w-10 rounded objec t-cover shadow"

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Ab router ko update karna hai**

const categoryRouter = require('express').Router()

const {create, read, deleteById, status, update} = require('../controller/category.controller')

const fileupload = require('express-fileupload')

categoryRouter.post('/create',fileupload({createParentPath: true}), create)

categoryRouter.get('/:id?', read)

categoryRouter.delete('/delete/:id', deleteById)

categoryRouter.patch('/status/:id', status)

categoryRouter.put('/update/:id', fileupload({createParentPath: true}),  update)

module.exports = categoryRouter

**category controller me edit ki api banana hai**

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingCat = await categoryModel.findById(id);

            if (existingCat) {

                fs.unlinkSync(`public/images/category/${existingCat.image}`)

            }

            await categoryModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const category = await categoryModel.findById(id);

            await categoryModel.findByIdAndUpdate(

                id,

                { $set: { status: !category.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async update(req, res) {

        try {

            const id = req.params.id

            const categoryImg = req.files.image

            const { name, slug } = req.body

            const existingItem = await categoryModel.findById(id)

            if (existingItem) {

                return serverErrorResponse(res, "Category not found", 409)

            }

            const update = {}

            if (name) update.name = name

            if (slug) update.slug = slug

            if (categoryImg) {

                const image = createUniqueName(categoryImg.name)

                const destination = 'public/images/category/' + image

                categoryImg.mv(

                    destination,

                    async (error) => {

                        if (error) {

                            return errorResponse(res, "File not found")

                        } else {

                            fs.unlinkSync(`public/images/category/${existingItem.image}`)

                            console.log(update);

                            update.image = image

                            const category =

                             await categoryModel.findByIdAndUpdate({

                               id,

                               {$set: update}

                            })

                            return updatedResponse(res, "category updated successfully")

                        }

                    }

                )

            }

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = category

**ab category edit ko update karna hai**

'use client'

import { getCategories } from '@/library/api-calls';

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useEffect, useRef, useState } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function categoryEdit({ params }) {

    const [categories, setCategories] = useState({})

    const nameRef = useRef()

    const slugRef = useRef()

    async function getCategory() {

        const categoryJSON = await getCategories(params?.category\_id)

        console.log(categoryJSON.data, "data");

        setCategories(categoryJSON.data)

    }

    useEffect(() => {

        getCategory()

    }, [params.category\_id])

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.put(`category/update/${category.\_id}`, formData).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        defaultValue={category.name}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        defaultValue={category.slug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${category.image}`}

                        alt={category.name}

                        className="h-10 w-10 rounded objec t-cover shadow"

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Ab category controller ko update karna hai**

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingCat = await categoryModel.findById(id);

            if (existingCat) {

                fs.unlinkSync(`public/images/category/${existingCat.image}`)

            }

            await categoryModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const category = await categoryModel.findById(id);

            await categoryModel.findByIdAndUpdate(

                id,

                { $set: { status: !category.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async update(req, res) {

        try {

            const id = req.params.id

            const categoryImg = req.files.image

            console.log(categoryImg, "categoryImg")

            console.log(req.body);

            const { name, slug } = req.body

            const existingItem = await categoryModel.findById(id)

            console.log(existingItem);

            return

            if (existingItem) {

                return serverErrorResponse(res, "Category not found", 409)

            }

            const update = {}

            if (name) update.name = name

            if (slug) update.slug = slug

            if (categoryImg) {

                const image = createUniqueName(categoryImg.name)

                const destination = 'public/images/category/' + image

                categoryImg.mv(

                    destination,

                    async (error) => {

                        if (error) {

                            return errorResponse(res, "File not found")

                        } else {

                            fs.unlinkSync(`public/images/category/${existingItem.image}`)

                            console.log(update);

                            update.image = image

                             await categoryModel.findByIdAndUpdate({

                               id,

                               {$set: update}

                            })

                            return updatedResponse(res, "category updated successfully")

                        }

                    }

                )

            }

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = category

**fir se category controller ko update karo**

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingCat = await categoryModel.findById(id);

            if (existingCat) {

                fs.unlinkSync(`public/images/category/${existingCat.image}`)

            }

            await categoryModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const category = await categoryModel.findById(id);

            await categoryModel.findByIdAndUpdate(

                id,

                { $set: { status: !category.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async update(req, res) {

        try {

            const id = req.params.id

            const categoryImg = req.files.image

            console.log(categoryImg, "categoryImg")

            console.log(req.body);

            const { name, slug } = req.body

            const existingItem = await categoryModel.findById(id)

            console.log(existingItem);

            if (!existingItem) {

                return serverErrorResponse(res, "Category not found", 409)

            }

            const update = {}

            if (name) update.name = name

            if (slug) update.slug = slug

            if (categoryImg) {

                const image = createUniqueName(categoryImg.name)

                const destination = 'public/images/category/' + image

                categoryImg.mv(

                    destination,

                    async (error) => {

                        if (error) {

                            return errorResponse(res, "File not found")

                        } else {

                            fs.unlinkSync(`public/images/category/${existingItem.image}`)

                            console.log(update);

                            update.image = image

                             await categoryModel.findByIdAndUpdate(

                               id,

                               {$set: update}

                            )

                            return updatedResponse(res, "category updated successfully")

                        }

                    }

                )

            }

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = category

**yanha tak ka code uncomplete edit api update ke saath commit hai**

**controller ko update karna**

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingCat = await categoryModel.findById(id);

            if (existingCat) {

                fs.unlinkSync(`public/images/category/${existingCat.image}`)

            }

            await categoryModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const category = await categoryModel.findById(id);

            await categoryModel.findByIdAndUpdate(

                id,

                { $set: { status: !category.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async update(req, res) {

        try {

            const id = req.params.id

            const categoryImg = req.files.image

            console.log(categoryImg, "categoryImg")

            console.log(req.body);

            const { name, slug } = req.body

            const existingItem = await categoryModel.findById(id)

            console.log(existingItem);

            if (!existingItem) {

                return serverErrorResponse(res, "Category not found", 409)

            }

            const update = {}

            if (name) update.name = name

            if (slug) update.slug = slug

            if (categoryImg) {

                const image = createUniqueName(categoryImg.name)

                const destination = 'public/images/category/' + image

                categoryImg.mv(

                    destination,

                    async (error) => {

                        if (error) {

                            return errorResponse(res, "File not found")

                        } else {

                            fs.unlinkSync(`public/images/category/${existingItem.image}`)

                            console.log(update);

                            update.image = image

                             await categoryModel.findByIdAndUpdate(

                               id,

                               {$set: update}

                            )

//remove object

                            return updatedResponse(res, "category updated successfully")

                        }

                    }

                )

            }

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = category

**update api-call**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories }

**category table ko update karna**

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getCategories } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function CategoryTable() {

    const category = await getCategories()

    const categories =  category.data

    // Dummy data (you can replace this with real API data)

    return (

        <div className="bg-white shadow rounded-lg p-6">

            {/\* Header \*/}

            <div className="flex items-center justify-between mb-4">

                <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

                <Link href="/admin/categoryTable/add">

                    <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

                        + Add Category

                    </button>

                </Link>

            </div>

            {/\* Table \*/}

            <div className="overflow-x-auto">

                <table className="min-w-full text-sm text-left">

                    <thead className="bg-gray-50 text-gray-600 border-b">

                        <tr>

                            <th className="px-4 py-2">Image</th>

                            <th className="px-4 py-2">Name</th>

                            <th className="px-4 py-2">Slug</th>

                            <th className="px-4 py-2 text-right">Actions</th>

                        </tr>

                    </thead>

                    <tbody className="text-gray-700">

                        {categories.map((cat) => (

                            <tr key={cat.id} className="border-b hover:bg-gray-50">

                                <td className="px-4 py-3">

                                    <img

                                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${cat.image}`}

                                        alt={cat.name}

                                        className="w-12 h-12 object-cover rounded"

                                    />

                                </td>

                                <td className="px-4 py-3">{cat.name}</td>

                                <td className="px-4 py-3">{cat.slug}</td>

                                <td className="px-4 py-3 text-right space-x-2">

                                    <Link href={`/admin/categoryTable/edit/${cat.\_id}`}>

                                        <button className="text-blue-600 hover:text-blue-800 cursor-pointer transition">

                                            <FiEdit />

                                        </button>

                                    </Link>

                                    <StatusBtn status={cat.status} id={cat.\_id} />

                                    <DeleteBtn id={cat.\_id} />

                                    <button className="text-red-600 hover:text-red-800 transition">

                                    </button>

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

**Fir se update api-call**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories }

**category edit page ko update karana**

'use client'

import { getCategories } from '@/library/api-calls';

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useEffect, useRef, useState } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function categoryEdit({ params }) {

    const [category, setCategory] = useState({})

    const nameRef = useRef()

    const slugRef = useRef()

    async function getCategory() {

        const categoryJSON = await getCategories(params?.category\_id)

        console.log(categoryJSON.data, "data");

        setCategory(categoryJSON.data)

    }

    useEffect(() => {

        getCategory()

    }, [params.category\_id])

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.put(`category/update/${category.\_id}`, formData).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        defaultValue={category.name}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        defaultValue={category.slug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${category.image}`}

                        alt={category.name}

                        className="h-10 w-10 rounded objec t-cover shadow"

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Update api complete hoti hai**

**Ab color module start karenge**

**Color module ka Schema banao**

**model/color.model.js**

const mongoose = require('mongoose');

//const validator = require('validator'); // for advanced string validation

const colorSchema = new mongoose.Schema({

  name: {

    type: String,

    required: [true, 'Category name is required'],

    unique: true,

    trim: true,

    minlength: [3, 'Category name must be at least 3 characters'],

    maxlength: [50, 'Category name must be at most 50 characters'],

  },

  slug: {

    type: String,

    required: [true, 'Slug is required'],

    unique: true,

    lowercase: true,

  },

  hexcode: {

    type: String,

    default: null,

     unique: true,

  },

  status: {

    type: Boolean,

    default: true,

  },

}, {

  timestamps: true,

});

module.exports = mongoose.model("Color", colorSchema)

**color ka router banao**

const colorRouter = require('express').Router();

const {  create, read, deleteById, status, update } = require('../controller/color.controller');

colorRouter.post('/create', create)

colorRouter.get('/:id?', read);

colorRouter.delete('/delete/:id?', deleteById);

// categoryRouter.patch('/status/:id?', updateById);

colorRouter.patch('/status/:id?', status);

module.exports = colorRouter;

**index.js ko update karo**

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const colorRouter = require('./router/color.router')

const server = express();

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use('/color', colorRouter)

server.use(express.static('public'));

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect(process.env.DATABASE\_URL, {dbName: "rishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

**Color controller banao**

const colorModel = require('../model/color.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const color = {

    async create(req, res) {

        try {

            const { name, slug, hexcode } = req.body

            if (!name || !slug || !hexcode) {

                return noContentResponse(res)

            }

            const existingItem = await colorModel.findOne({name})

            if (existingItem) {

                return serverErrorResponse(res, "color already created", 409)

            }

            const color = await colorModel.create({name, slug, hexcode})

            await color.save()

            return createdResponse(res, "color created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let color = null

            if (id) {

                color= await colorModel.findById(id)

            } else {

                color = await colorModel.find()

            }

            if (!color) errorResponse(res, "Color not found")

            return successResponse(res, "Color Found", color)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingItem = await colorModel.findById(id);

            await colorModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const color = await colorModel.findById(id);

            await colorModel.findByIdAndUpdate(

                id,

                { $set: { status: !color.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

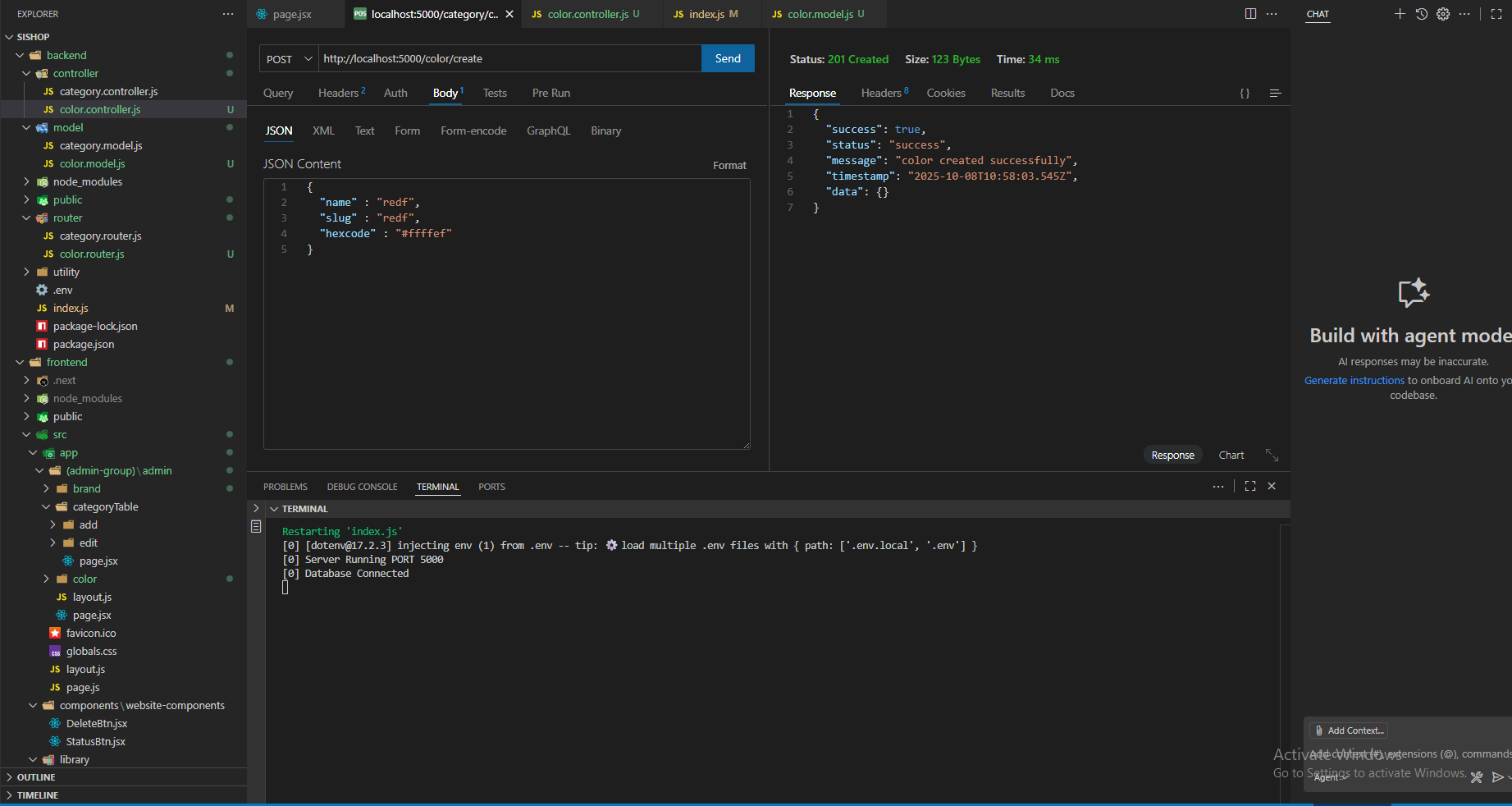
    },

}

module.exports = color

**router me se color update ka api hatao**

**ab thunder par check karo**

****

**Color ka front end banana ke liye cat eke folder ko hi copy paste kar denge**

**Aur sath hi brand ka foder bhi bna lenge**

**Ab color ka front edn banayenge**

**Color add page.jsx ko banaye karenge kyonki hamane cateogory ka hi add liya hai to usme hi update karenge**

'use client'

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const data = {

            name: nameRef.current.value,

            slug: slugRef.current.value,

            hexcode: e.target.hexcode.value

        }

        axiosInstance.post("color/create", data).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="color"

                            name="category\_image"

                        />

                        <FaRegImage className='text-5xl' />

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Fir se add page update karo**

'use client'

import { createSlug, notify, axiosInstance } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const data = {

            name: nameRef.current.value,

            slug: slugRef.current.value,

            hexcode: e.target.hexcode.value

        }

        axiosInstance.post("color/create", data).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Color Code

                    </label>

                    <input

                        type="color"

                        id="slug"

                        name="hexcode"

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

**Color ko get karana**

**15:00**

**Api call me getcolor api ko call karna hai**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if(id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors }

**ab color table me display karna hai colors ko**

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getColors } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function ColorTable() {

    const colorJSON = await getColors()

    const colors = colorJSON.data

    console.log(colors);

    // Dummy data (you can replace this with real API data)

    return (

        <div className="bg-white shadow rounded-lg p-6">

            {/\* Header \*/}

            <div className="flex items-center justify-between mb-4">

                <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

                <Link href="/admin/color/add">

                    <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

                        + Add color

                    </button>

                </Link>

            </div>

            {/\* Table \*/}

            <div className="overflow-x-auto">

                <table className="min-w-full text-sm text-left">

                    <thead className="bg-gray-50 text-gray-600 border-b">

                        <tr>

                            <th className="px-4 py-2">Image</th>

                            <th className="px-4 py-2">Name</th>

                            <th className="px-4 py-2">Slug</th>

                            <th className="px-4 py-2 text-right">Actions</th>

                        </tr>

                    </thead>

                    <tbody className="text-gray-700">

                        {colors.map((color) => (

                            <tr key={color.id} className="border-b hover:bg-gray-50">

                                <td

                                style={{background : color.hexcode}}

                                className="px-4 py-3">

                                </td>

                                <td className="px-4 py-3">{color.name}</td>

                                <td className="px-4 py-3">{color.slug}</td>

                                <td className="px-4 py-3 text-right space-x-2">

                                    <Link href={`/admin/categoryTable/edit/${color.\_id}`}>

                                        <button className="text-blue-600 hover:text-blue-800 cursor-pointer transition">

                                            <FiEdit className='text-xl' />

                                        </button>

                                    </Link>

                                    <StatusBtn url="color" status={color.status} id={color.\_id} />

                                    <DeleteBtn url="color" id={color.\_id} />

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

**Status button ko update karna**

'use client'

import React from 'react'

import { axiosInstance, notify } from '@/library/helper'

import { Router } from 'next/navigation'

import { useRouter } from 'next/navigation'

export default function StatusBtn({ id, status, url }) {

    const router = useRouter()

    function statusHandler() {

        axiosInstance.patch(`${url} /status/${id}`).then(

            (response) => {

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    router.refresh()

                }

            }

        ).catch((error) => {

            console.log(error)

        })

    }

    return (

        <button onClick={statusHandler} className="rounded-md cursor-pointer bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

            {

                status ?

                    "Active"

                    :

                    "Inactive"

            }

        </button>

    )

}

**Status button ko update karna**

'use client'

import { axiosInstance, notify } from '@/library/helper'

import { useRouter } from 'next/navigation'

import React from 'react'

import { FiTrash2 } from 'react-icons/fi';

export default function DeleteBtn({ id, url }) {

  const router = useRouter()

  function deleteHandler() {

    axiosInstance.delete(`${url} /delete/${id}`).then(

      (response) => {

        notify(response.data.message, response.data.success)

        if (response.data.success) {

          router.refresh()

        }

      }

    ).catch((error) => {

      console.log(error)

    })

  }

  return (

    <button onClick={deleteHandler} className="text-red-600 hover:text-red-800 transition rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium cursor-pointer hover:bg-gray-200">

      <FiTrash2  />

    </button>

  )

}

**Ab hum brand ki api banayenge**

**Brand ka model banana**

**Category.model.js ko copy pate karenge aur brand.model.js name denge**

**Aur kuch change karenge**

const mongoose = require("mongoose")

const brandSchema = new mongoose.Schema({

    name : {

        type : String,

        required : true,

        unique : true,

        minlength : 3,

        maxlength : 50

    },

    slug : {

        type : String,

        required : true,

        unique : true

    },

    image : {

        type : String,

        default : null

    },

    status : {

        type : Boolean,

        default : true

    }

},

    {

        timestamps : true

    }

)

module.exports = mongoose.model("Brand", brandSchema)

**brand ke liye brand.router.js banayenge**

**category.router.js me se code ko copy paste karenge aur usme update karenge**

**brand lena hai to image rahegi**

const brandRouter = require('express').Router()

const {create, read, deleteById, status, update} = require('../controller/brand.controller')

const fileupload = require('express-fileupload')

brandRouter.post('/create',fileupload({createParentPath: true}), create)

brandRouter.get('/:id?', read)

brandRouter.delete('/delete/:id', deleteById)

brandRouter.patch('/status/:id', status)

brandRouter.put('/update/:id', fileupload({createParentPath: true}),  update)

module.exports = categoryRouter

**ab brand.controller.js banana ha sam category.controller.js ko copy paste kar lenge**

aur janha janha category vanaha brand kar denge aur delete api me **existingCat** ki jagah **existingBrand** kar denge

const brandModel = require('../model/brand.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const brand = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const brandImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await brandModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Brand already created", 409)

            }

            const image = createUniqueName(brandImg.name)

            const destination = 'public/images/brand/' + image

            brandImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const brand = await brandModel.create({

                            name,

                            slug,

                            image

                        })

                        await brand.save()

                        return createdResponse(res, "brand created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let brand = null

            if (id) {

                brand = await brandModel.findById(id)

            } else {

                brand = await brandModel.find()

            }

            if (!brand) errorResponse(res, "Brand not found")

            return successResponse(res, "Brand Found", brand)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            const existingItem = await brandModel.findById(id);

            if (existingItem) {

                fs.unlinkSync(`public/images/brand/${existingItem.image}`)

            }

            await brandModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

}

module.exports = brand

**brand.router.js ko upadate kar dena hai jinke route nahi bana rakhe unko delete karna hai jese status and update**

const brandRouter = require('express').Router()

const {create, read, deleteById} = require('../controller/brand.controller')

const fileupload = require('express-fileupload')

brandRouter.post('/create',fileupload({createParentPath: true}), create)

brandRouter.get('/:id?', read)

brandRouter.delete('/delete/:id', deleteById)

module.exports = categoryRouter

ab index.js me jakar brand ka base router path update karna hai

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const colorRouter = require('./router/color.router')

const brandRouter = require('./router/brand.router')

const server = express();

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use('/color', colorRouter)

server.use('/brand', brandRouter)

server.use(express.static('public'));

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect(process.env.DATABASE\_URL, {dbName: "rishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

Ab front me ise call kar lena hai

Brand/add/page.jsx ko update karenge

Kyonki **hamane brand/add.page.jsx** ko **category/add/page.jsx** se copy kiya hai to kuch bhi change nahi hoga keval **axiosInstance** me **brand** aayega

'use client'

import { createSlug, notify, axiosInstance  } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const formData = new FormData();

        formData.append("name", nameRef.current.value);

        formData.append("slug", slugRef.current.value)

        formData.append("image", e.target.category\_image.files[0])

        axiosInstance.post("brand/create", formData).then(

            (response)=> {

                // console.log(response.data.success);

                notify(response.data.message, {flag: response.data.success})

                if(response.data.success); {

                     nameRef.current.value = ""

                     slugRef.current.value = ""

                }

            }

        ).catch((error)=> {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                        htmlFor="file\_input"

                    >

                        Upload file

                    </label>

                    <div className=" w-full flex flex-col items-center justify-center text-sm h-30 text-gray-900 border border-gray-300 rounded-lg cursor-pointer bg-gray-50 dark:text-gray-400 focus:outline-none dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400" >

                        <input

                            id="file\_input"

                            type="file"

                            name="category\_image"

                             />

                        <FaRegImage  className='text-5xl'/>

                    </div>

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Ab front ke url me add ke form par jakar brand ke name, slug and imge upload kar sakte hai+

Yanha tak front end ke add se brand successfully create ho raha hai

Ab api-call me ise get karna hai

getCategories ke get ko as it as copy paste karna aur usme kuch updae karna hai vo add kar do

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getBrands = async (id = null) => {

    try {

        let API = "brand"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if(id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors, getBrands }

ab ise brands table par call karna hai

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { **getBrands**  } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function CategoryTable() {

**const brandsJSON = await getBrands()**

**const brands = brandsJSON.data**

**console.log(brands);**

    // Dummy data (you can replace this with real API data)

    return (

        <div className="bg-white shadow rounded-lg p-6">

            {/\* Header \*/}

            <div className="flex items-center justify-between mb-4">

                <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

                <Link href="/admin/brand/add">

                    <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

                        + Add Category

                    </button>

                </Link>

            </div>

            {/\* Table \*/}

            <div className="overflow-x-auto">

                <table className="min-w-full text-sm text-left">

                    <thead className="bg-gray-50 text-gray-600 border-b">

                        <tr>

                            <th className="px-4 py-2">Image</th>

                            <th className="px-4 py-2">Name</th>

                            <th className="px-4 py-2">Slug</th>

                            <th className="px-4 py-2 text-right">Actions</th>

                        </tr>

                    </thead>

                    <tbody className="text-gray-700">

                        {brands.map((brand) => (

                            <tr key={brand.id} className="border-b hover:bg-gray-50">

                                <td className="px-4 py-3">

                                    <img

                                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/brand/${brand.image}`}

                                        alt={brand.name}

                                        className="w-12 h-12 object-cover rounded"

                                    />

                                </td>

                                <td className="px-4 py-3">{brand.name}</td>

                                <td className="px-4 py-3">{brand.slug}</td>

                                <td className="px-4 py-3 text-right space-x-2">

                                    <Link href={`/admin/brand/edit/${brand.\_id}`}>

                                        <button className="text-blue-600 hover:text-blue-800 cursor-pointer transition">

                                            <FiEdit className='text-xl' />

                                        </button>

                                    </Link>

                                    <StatusBtn url="brand" status={brand.status} id={brand.\_id} />

                                    <DeleteBtn url="brand" id={brand.\_id} />

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

Yanha tak hamane brand ko complete kar liya display create and successfully and delete bhi lekin hamane status aur update ki api nahi banai hai to vo button yanha kaam nahi karega

27-08-25 day 8

Product module banana abtak ke jitane module banaye hai vo sabhi isme add honge

Frontend me admin me **admin/product/page.jsx** banao

Admin/product/add/page.jsx banao product ko add karne ke liye

Ab pahale backend banayenge

Model/product.model.js banana hai

const mongoose = require("mongoose");

const productSchema = new mongoose.Schema(

    {

        name: {

            type: String,

            maxLength: 50,

            unique: true

        },

        slug: {

            type: String,

            maxLength: 60,

            unique: true

        },

        shortDescription: {

            type: String,

            maxLength: 200,

        },

        longDescription: {

            type: String,

        },

        originalPrice: {

            type: Number,

            default: 200

        },

        discountPercentage: {

            type: Number,

            default: 5

        },

        finalPrice: {

            type: Number,

        },

        categoryId: {

            type: mongoose.Schema.ObjectId,

            ref: "Category"

        },

        brandId: {

            type: mongoose.Schema.ObjectId,

            ref: "Brand"

        },

        colors: [

            {

                type: mongoose.Schema.ObjectId,

                ref: "Color"

            }

        ],

        thumbnail: {

            type: String,

            default: null

        },

        images: [

            {

                type: String,

            }

        ],

        stock: {

            type: Boolean,

            default: true

        },

        topSelling: {

            type: Boolean,

            default: false

        },

        status: {

            type: Boolean,

            default: true

        },

    },

    {

        timestamps: true

    }

)

const ProductModel = mongoose.model("Product", productSchema);

module.exports = ProductModel

product.router.js banana hai

const productRouter = require('express').Router()

const {create} = require('../controller/product.controller')

const fileupload = require('express-fileupload')

productRouter.post('/create',fileupload({createParentPath: true}), create)

module.exports = productRouter

product.controller ki file banana hai

**category model ke controller** file ko hi copy paste karte hai **product.controller**.js me

abhi ke liye backend ke code yanhatak hi rahane dete pahale hame frontend banana hoga ise get karne ke liye

product ke model ke according product add ya product create form bana lenge

admin/product/add/page.jsx kyonki ispar kuch logic lagenge to hum ek component banakar is par call kara lenge

components/admin/ProductAdd.jsx

'use client'

import React from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

export default function ProductAdd({category, colors, brands}) {

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", slugRef.current.value,)

        formData.append("brandId", slugRef.current.value,)

        formData.append("colors", slugRef.current.value,)

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.category\_image.files[0])

        formData.append("images", e.target.category\_image.files[0])

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select>

                    <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select>

                    {/\* Color Multi-Select \*/}

                    <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select>

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Admin/product/add/page.jsx me call karawana

import ProductAdd from '@/components/admin-components/ProductAdd'

import React from 'react'

export default function page() {

  return (

    <div>

      <ProductAdd />

    </div>

  )

}

/admin/product/add/page.jsx ko update karna

import ProductAdd from '@/components/admin-components/ProductAdd';

import { getBrands, getCategories, getColors } from '@/library/api-calls';

// import { getBrands, getCategories, getColors } from '@/library/api-call'

import React from 'react'

export default async function page() {

    const categoryJSON = await getCategories();

    const brandJSON = await getBrands();

    const colorJSON = await getColors();

    return (

        <ProductAdd category={categoryJSON.data} brands={brandJSON.data} colors={colorJSON.data} />

    )

}

Yanha tak hamane product ko create kar liya hai successfully lekin get ya display nahi kiya hai

**28-08-25 day 9**

Category aur color search ya select karne ke liye ek package ka use karenge

React slector

Npm i –save react-select

Ise frontend me add karna hai

**ProductAdd.jsx me update karna**

'use client'

import React from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", slugRef.current.value,)

        formData.append("brandId", slugRef.current.value,)

        formData.append("colors", slugRef.current.value,)

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.category\_image.files[0])

        formData.append("images", e.target.category\_image.files[0])

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Yanha tak react select package successfull kam kar raha hai

Ab ja category id select karenge to iski value select hogi to select bhi uski value pass karega

**Fir se ProductAdd.jsx ko update karenge**

Category aur brand ki value ko select karna ye single value select karenge

'use client'

import React from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", slugRef.current.value,)

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.category\_image.files[0])

        formData.append("images", e.target.category\_image.files[0])

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

ProductAdd.jsx me color ke value select karenge kyonki vo multiple hai

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.category\_image.files[0])

        formData.append("images", e.target.category\_image.files[0])

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data)=> {

                            const color = data.map(o=> o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Yanha tak hamane color ko slect kar liya hai

Ab hum image ko select karenge

Single image select karna

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        formData.append("images", e.target.category\_image.files[0])

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data)=> {

                            const color = data.map(o=> o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Ab multiple images select karna hai

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Yanha tak multiple images aa gai hai

**Ab long description me text editor add karna hai**

jodiEditor ko install karna frontend me

npm i jodit-pro-react

admin/TextEditor.jsx file banao aur

import React, {useState, useRef} from 'react';

import JoditEditor from "jodit-pro-react";

const TextEditor = ({placeholder}) => {

    const editor = useRef(null)

    const [content, setContent] = useState('')

    return (

        <JoditEditor

            ref={editor}

            value={content}

            tabIndex={1} // tabIndex of textarea

            onChange={newContent => {}}

        />

    );

}

export default TextEditor

ProductAdd ko update karna kyonki vanha par longdescription add kar rakha hai

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

import TextEditor from "@/app/(admin-group)/admin/TextEditor";

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const [longDescription, setlongDescription] = useState("")

    const name = "shivam"

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDescription.value,)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("category/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                {/\* <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                /> \*/}

                <TextEditor value={longDescription} changeHandler={(data)=>

                    setlongDescription(data)

                }/>

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Ab TextEditor.jsx ko update karenge

import React, {useState, useRef} from 'react';

import JoditEditor from "jodit-pro-react";

const TextEditor = ({placeholder, value, changeHandler}) => {

    const editor = useRef(null)

    const [content, setContent] = useState('')

    return (

        <JoditEditor

            ref={editor}

            value={value}

            tabIndex={1} // tabIndex of textarea

            onChange={(data)=> changeHandler(data)}

        />

    );

}

export default TextEditor

ab yah text editor product form me add ho jayega

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

import TextEditor from "@/app/(admin-group)/admin/TextEditor";

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const [longDescription, setlongDescription] = useState("")

    const name = "shivam"

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", longDescription)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("product/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountPercentageRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                {/\* <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                /> \*/}

                <TextEditor value={longDescription} changeHandler={(data)=>

                    setlongDescription(data)

                }/>

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Ab finaly hamane text editor set kar diya hai

Ab backend me jaana hai aur category controller update karna hai

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const product = {

    async create(req, res) {

        try {

            console.log(req.body);

            console.log(req.files);

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = product

ProductRouter ko check kar lena hai

const productRouter = require('express').Router()

const {create} = require('../controller/product.controller')

const fileupload = require('express-fileupload')

productRouter.post('/create',fileupload({createParentPath: true}), create)

module.exports = productRouter

kyonki abhi jodit editor use karne se product nahi ban raha hai isliye

productAdd ko update karna

Jese hi hum product add kare to backend ke consle me product ka object aana chahiye

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

import TextEditor from "@/app/(admin-group)/admin/TextEditor";

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

  // const [longDescription, setlongDescription] = useState(null)

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", longDescription)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("product/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountPercentageRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                {/\* <textarea

                    name="longDescription"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                /> \*/}

                <TextEditor value={longDescription} changeHandler={(data)=> setlongDescription(data)}/>

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Update productAdd

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

// import TextEditor from "@/app/(admin-group)/admin/TextEditor";

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    // const [longDescription, setlongDescription] = useState(null)

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDec.value)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        formData.append("categoryId", e.target.categoryId.value,)

        formData.append("brandId", e.target.brand.value,)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("product/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountPercentageRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDec"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* <TextEditor value={longDescription} changeHandler={(data)=> setlongDescription(data)}/> \*/}

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        } />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Index.js ko update karna

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const colorRouter = require('./router/color.router')

const brandRouter = require('./router/brand.router')

const productRouter = require('./router/product.router')

const server = express();

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use('/color', colorRouter)

server.use('/brand', brandRouter)

server.use('/product', productRouter)

server.use(express.static('public'));

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect(process.env.DATABASE\_URL, {dbName: "rishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

Yanha tak 28 wali video ke according hai

Yanha jab app product banate ho to backend ke console me pura project aa jana chahiye

**Lekin jese me index me product router add karta hun to api call ki error aa jati hai**

Galati **product.model.js** ka name sahi kar lena **prodct.model.js** se

Ab product create karne par product ka object backend ke console me aa raha hai

**Update product.controler.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const product = {

    async create(req, res) {

        try {

            console.log(req.body);

            console.log(req.files);

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = product

updae productAdd

jo niche code highlight hai vo mene chat gpt se solve karwaye hai

aur ab backend ke console me image aur data ka object aa raha hai

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const [selectedCategory, setSelectedCategory] = useState(null);

    const [selectedBrand, setSelectedBrand] = useState(null);

    // const [longDescription, setlongDescription] = useState(null)

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDec.value)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        // formData.append("categoryId", e.target.categoryId.value,)

        formData.append("categoryId", selectedCategory)

        // formData.append("brandId", e.target.brand.value,)

        formData.append("brandId", selectedBrand)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("product/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountPercentageRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDec"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* <TextEditor value={longDescription} changeHandler={(data)=> setlongDescription(data)}/> \*/}

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="category-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        }

                        value={selectedCategory}

                        onChange={(option) => setSelectedCategory(option)}

                    />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="brand-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        }

                        value={selectedBrand}

                        onChange={(option) => setSelectedBrand(option)}

                    />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="color-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Abhi product add karoge to erro aayega isliye

**productAdd me update karo**

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const [selectedCategory, setSelectedCategory] = useState(null);

    const [selectedBrand, setSelectedBrand] = useState(null);

    // const [longDescription, setlongDescription] = useState(null)

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        formData.append("longDescription", e.target.longDec.value)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        // formData.append("categoryId", e.target.categoryId.value,)

        formData.append("categoryId", selectedCategory?.value)

        // formData.append("brandId", e.target.brand.value,)

        formData.append("brandId", selectedBrand?.value)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("product/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountPercentageRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                <textarea

                    name="longDec"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                />

                {/\* <TextEditor value={longDescription} changeHandler={(data)=> setlongDescription(data)}/> \*/}

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="category-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        }

                        value={selectedCategory}

                        onChange={(option) => setSelectedCategory(option)}

                    />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="brand-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        }

                        value={selectedBrand}

                        onChange={(option) => setSelectedBrand(option)}

                    />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="color-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

Yanha tak product successfully toastify ke saath create ho gaya hai aur databse me add ho gaya hai

29-08-25 day 10

ProductAdd ko update karana

'use client'

import React, { useState } from "react";

import { useRef } from "react";

import { axiosInstance, notify, createSlug } from "@/library/helper";

import Select from 'react-select'

import TextEditor from "@/app/(admin-group)/admin/TextEditor";

export default function ProductAdd({ category, colors, brands }) {

    const [selcolor, setSelcolor] = useState([])

    const [selectedCategory, setSelectedCategory] = useState(null);

    const [selectedBrand, setSelectedBrand] = useState(null);

    const [longDescription, setlongDescription] = useState(null)

    const nameRef = useRef();

    const slugRef = useRef();

    const originalRef = useRef();

    const discountPercentageRef = useRef();

    const finalRef = useRef();

    function generateSlug() {

        const slug = createSlug(nameRef.current.value);

        slugRef.current.value = slug;

    }

    function submitHandler(e) {

        e.preventDefault();

        // const data = {

        //     name: nameRef.current.value,

        //     slug: slugRef.current.value

        // };

        // console.log(e.target.category\_image.files[0]);

        // return

        const formData = new FormData();

        formData.append("name", nameRef.current.value,)

        formData.append("slug", slugRef.current.value,)

        formData.append("shortDescription", e.target.shortDescription.value,)

        // formData.append("longDescription", e.target.longDec.value)

        formData.append("longDescription", longDescription)

        formData.append("originalPrice", originalRef.current.value,)

        formData.append("discountPercentage", discountPercentageRef.current.value,)

        formData.append("finalPrice", finalRef.current.value,)

        // formData.append("categoryId", e.target.categoryId.value,)

        formData.append("categoryId", selectedCategory?.value)

        // formData.append("brandId", e.target.brand.value,)

        formData.append("brandId", selectedBrand?.value)

        formData.append("colors", JSON.stringify(selcolor))

        formData.append("thumbnail", slugRef.current.value,)

        formData.append("thumbnail", e.target.thumbnail.files[0])

        for (let img of e.target.images.files) {

            formData.append("images", img)

        }

        // axios.post("http://localhost:5000/category/create", data) //yanha data use kiya gaya hai

        axiosInstance.post("product/create", formData) // data ki jagah formData use kiya gaya hai

            .then((response) => {

                // console.log(response.data);

                // console.log(response.data.success);

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            })

            .catch((error) => {

                console.log(error);

            });

    }

    function priceCalculate() {

        const op = originalRef.current.value;

        const dp = discountPercentageRef.current.value;

        const fp = op - (op \* dp) / 100;

        finalRef.current.value = parseInt(fp);

    }

    return (

        <div>

            <form

                onSubmit={submitHandler}

                className="max-w-5xl mx-auto grid grid-cols-1 md:grid-cols-2 gap-6 p-6 bg-white rounded-lg shadow"

            >

                {/\* Product Name \*/}

                <input

                    type="text"

                    ref={nameRef}

                    onChange={generateSlug}

                    name="name"

                    placeholder="Product Name"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Slug \*/}

                <input

                    type="text"

                    ref={slugRef}

                    readOnly

                    name="slug"

                    placeholder="Slug"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Short Description \*/}

                <textarea

                    name="shortDescription"

                    placeholder="Short Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="2"

                />

                {/\* Long Description \*/}

                {/\* <textarea

                    name="longDec"

                    placeholder="Long Description"

                    className="w-full border border-gray-300 rounded-md px-4 py-2 md:col-span-2"

                    rows="4"

                /> \*/}

                <TextEditor value={longDescription} changeHandler={(data)=> setlongDescription(data)}/>

                {/\* Pricing Row \*/}

                <div className="md:col-span-2 flex flex-col md:flex-row gap-4">

                    <input

                        type="number"

                        ref={originalRef}

                        onChange={priceCalculate}

                        name="originalPrice"

                        placeholder="Original Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        ref={discountPercentageRef}

                        onChange={priceCalculate}

                        name="discountPercentage"

                        placeholder="Discount Percentage"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                    <input

                        type="number"

                        name="finalPrice"

                        ref={finalRef}

                        readOnly

                        placeholder="Final Price"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    />

                </div>

                {/\* Category / Brand / Colors Row \*/}

                <div className="md:col-span-2 flex h-10 flex-col md:flex-row gap-4">

                    {/\* Category Select \*/}

                    {/\* <Select options={options} />

                    <select

                        name="categoryId"

                        className="w-full border border-gray-300 rounded-md px-4"

                    >

                        <option value="">Select Category</option>

                        {category?.map((cat) => (

                            <option key={cat.\_id} value={cat.\_id}>{cat.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="category-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            category.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        }

                        value={selectedCategory}

                        onChange={(option) => setSelectedCategory(option)}

                    />

                    {/\* <select

                        name="brandId"

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        <option value="">Select Brand</option>

                        {brands?.map((brand) => (

                            <option key={brand.\_id} value={brand.\_id}>{brand.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="brand-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        options={

                            brands.map((cat) => {

                                return { value: cat.\_id, label: cat.name }

                            })

                        }

                        value={selectedBrand}

                        onChange={(option) => setSelectedBrand(option)}

                    />

                    {/\* Color Multi-Select \*/}

                    {/\* <select

                        name="colors"

                        multiple

                        className="w-full border border-gray-300 rounded-md px-4 py-2"

                    >

                        {colors?.map((color) => (

                            <option key={color.\_id} value={color.\_id}>{color.name}</option>

                        ))}

                    </select> \*/}

                    <Select

                        instanceId="color-select"

                        className="w-full border border-gray-300 rounded-md px-4"

                        closeMenuOnSelect={false}

                        isMulti

                        onChange={(data) => {

                            const color = data.map(o => o.value)

                            setSelcolor(color)

                        }}

                        options={

                            colors.map((col) => {

                                return { value: col.\_id, label: col.name }

                            })

                        }

                    />

                </div>

                {/\* Thumbnail \*/}

                <input

                    type="file"

                    name="thumbnail"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Images \*/}

                <input

                    type="file"

                    name="images"

                    multiple

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                />

                {/\* Stock \*/}

                <select

                    name="stock"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">In Stock</option>

                    <option value="false">Out of Stock</option>

                </select>

                {/\* Top Selling \*/}

                <select

                    name="topSelling"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="false">No</option>

                    <option value="true">Yes</option>

                </select>

                {/\* Status \*/}

                <select

                    name="status"

                    className="w-full border border-gray-300 rounded-md px-4 py-2"

                >

                    <option value="true">Active</option>

                    <option value="false">Inactive</option>

                </select>

                {/\* Submit Button \*/}

                <div className="md:col-span-2 text-right">

                    <button

                        type="submit"

                        className="bg-blue-600 hover:bg-blue-700 text-white px-6 py-2 rounded-md"

                    >

                        Submit Product

                    </button>

                </div>

            </form>

        </div>

    );

}

TextEditor

'use client';

import React, {  useRef } from 'react';

import JoditEditor from "jodit-pro-react"; // or "jodit-react" if using free version

const TextEditor = ({  value, changeHandler }) => {

    const editor = useRef(null);

    return (

        <JoditEditor

            ref={editor}

            value={value}

            tabIndex={1}

            onChange={(data)=> changeHandler(data)}

        />

    );

};

export default TextEditor;

product add page.jsx

import ProductAdd from '@/components/admin-components/ProductAdd';

import { getBrands, getCategories, getColors } from '@/library/api-calls';

// import { getBrands, getCategories, getColors } from '@/library/api-call'

import React from 'react'

export default async function page() {

    const categoryJSON = await getCategories();

    const brandJSON = await getBrands();

    const colorJSON = await getColors();

    return (

        <ProductAdd category={categoryJSON.data} brands={brandJSON.data} colors={colorJSON.data} />

    )

}

Yanha tak product toastify ke saath successfully kam kar raha hai aur JoditEditor bhi kaam kar raha hai

**Ab product ko get api banana hai**

**Update product controller.jsx**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find()

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

get api ka route bana lo

upate product router

kyonki hamane getProducts ko getCategories se copy paste kiya hai to keval function ka name badalana hai aur export me add karna hai

const productRouter = require('express').Router()

const {create, read} = require('../controller/product.controller')

const fileupload = require('express-fileupload')

productRouter.post('/create',fileupload({createParentPath: true}), create)

productRouter.get('/:id?', read)

module.exports = productRouter

ab frontend me api-calls me get product ka function call kar lo

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getProducts = async (id = null) => {

    try {

        let API = "product"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getBrands = async (id = null) => {

    try {

        let API = "brand"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if(id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors, getBrands, getProducts }

ab product ki table me hum product ko call kar lenge

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getProducts } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function product() {

  const product = await getProducts()

  const products = product.data

  console.log(products);

  // Dummy data (you can replace this with real API data)

  return (

    <div className="bg-white shadow rounded-lg p-6">

      {/\* Header \*/}

      <div className="flex items-center justify-between mb-4">

        <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

        <Link href="/admin/product/add">

          <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

            + Add Category

          </button>

        </Link>

      </div>

      {/\* Table \*/}

      <div className="overflow-x-auto">

        <table className="min-w-full text-sm text-left">

          <thead className="bg-gray-50 text-gray-600 border-b">

            <tr>

              <th className="px-4 py-2">Image</th>

              <th className="px-4 py-2">Name</th>

              <th className="px-4 py-2">Slug</th>

              <th className="px-4 py-2 text-right">Actions</th>

            </tr>

          </thead>

          <tbody className="text-gray-700">

            {products.map((product) => (

              <tr key={product.id} className="border-b hover:bg-gray-50">

                <td className="px-4 py-3">

                  <img

                    src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                    alt={product.name}

                    className="w-12 h-12 object-cover rounded"

                  />

                </td>

                <td className="px-4 py-3">{product.name}</td>

                <td className="px-4 py-3">{product.slug}</td>

                <td className="px-4 py-3 text-right space-x-2">

                  <Link href={`/admin/product/edit/${product.\_id}`}>

                    <button className="text-blue-600 hover:text-blue-800 cursor-pointer transition">

                      <FiEdit className='text-xl' />

                    </button>

                  </Link>

                  <button className=" hover:text-blue-800 cursor-pointer transition">

                    Active

                  </button>

                  <button className=" hover:text-blue-800 cursor-pointer transition">

                    Stock

                  </button>

                  <button className=" hover:text-blue-800 cursor-pointer transition">

                    top selling

                  </button>

                  <StatusBtn url="product" status={product.status} id={product.\_id} />

                  <DeleteBtn url="product" id={product.\_id} />

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

}

Yanha tak hamane products ko frontend me display karwa diya hai ab button par kaam karana hai

Product table

Active, stock and top selling ke keval ek api banana hai

**Components/Admin/ProductBtn.jsx banana hai**

'use client'

import { useRouter } from 'next/navigation'

import { axiosInstance, notify } from '@/library/helper'

import React, { useState } from 'react'

export default function ProductBtn({ product }) {

    const [toggle, setToggle] = useState(null)

    const router = useRouter()

    function statusHandler(flag) {

        axiosInstance.patch(`product/status/${product.\_id}`, { flag }).then(

            (response) => {

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    router.refresh()

                }

            }

        ).catch((error) => {

            console.log(error)

        })

    }

    return (

        <td className="whitespace-nowrap pl-10 py-3 space-x-2">

            <button onClick={() => statusHandler(1)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                {

                    product.status ?

                        "Active"

                        :

                        "Inactive"

                }

            </button>

            <button onClick={() => statusHandler(2)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                {

                    product.stock ?

                        "In"

                        :

                        "Out"

                }

            </button>

            <button onClick={() => statusHandler(3)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                {

                    product.topSelling ?

                        "Yes"

                        :

                        "No"

                }

            </button>

            <button className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                Delete

            </button>

            <button onClick={() => setToggle(product)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                View

            </button>

            {

                toggle && <ProductView product={product} close={() => setToggle(null)} />

            }

        </td>

    )

}

function ProductView({ product, close }) {

    return (

        <div className='w-full p-10 h-[300px] fixed bottom-0 left-0 bg-teal-100'>

            <button onClick={close}>Close</button>

            <div

                className="product-long-description"

                dangerouslySetInnerHTML={{ \_\_html: product.longDescription }}

            />

        </div>

    )

}

**productBtn.jsx ko hamane product table me call kar osme ek pros pass kiya hai**

// components/CategoryTable.jsx

import ProductBtn from '@/components/admin-components/ProductBtn';

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getProducts } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function product() {

  const product = await getProducts()

  const products = product.data

  console.log(products);

  // Dummy data (you can replace this with real API data)

  return (

    <div className="bg-white shadow rounded-lg p-6">

      {/\* Header \*/}

      <div className="flex items-center justify-between mb-4">

        <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

        <Link href="/admin/product/add">

          <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

            + Add Category

          </button>

        </Link>

      </div>

      {/\* Table \*/}

      <div className="overflow-x-auto">

        <table className="min-w-full text-sm text-left">

          <thead className="bg-gray-50 text-gray-600 border-b">

            <tr>

              <th className="px-4 py-2">Image</th>

              <th className="px-4 py-2">Name</th>

              <th className="px-4 py-2">Slug</th>

              <th className="px-4 py-2 text-right">Actions</th>

            </tr>

          </thead>

          <tbody className="text-gray-700">

            {products.map((product) => (

              <tr key={product.id} className="border-b hover:bg-gray-50">

                <td className="px-4 py-3">

                  <img

                    src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                    alt={product.name}

                    className="w-12 h-12 object-cover rounded"

                  />

                </td>

                <td className="px-4 py-3">{product.name}</td>

                <td className="px-4 py-3">{product.slug}</td>

                <ProductBtn product={product} />

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

}

Ab product stats ka api router me update karenge

const productRouter = require('express').Router()

const {create, read, status} = require('../controller/product.controller')

const fileupload = require('express-fileupload')

productRouter.post('/create',fileupload({createParentPath: true}), create)

productRouter.get('/:id?', read)

productRouter.patch('/status/:id?', status)

module.exports = productRouter

ab product controller.js me status ki api bana lenge

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find()

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

yanha tak status ke button theek se kaam kar rahe hai successfull toastify ke saath

ab category ke delete button par click karte hi category delete ho jaati hai lekin ab hum chahate hai ki delete hone se pahale confirmation message aana chahiye

category delete aur status kaam nahi kar raha hai tha url ka mistake niche updated code hai

// components/CategoryTable.jsx

import DeleteBtn from '@/components/website-components/DeleteBtn';

import StatusBtn from '@/components/website-components/StatusBtn';

import { getCategories } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

export default async function CategoryTable() {

    const category = await getCategories()

    const categories = category.data

    // console.log(categories);

    // Dummy data (you can replace this with real API data)

    return (

        <div className="bg-white shadow rounded-lg p-6">

            {/\* Header \*/}

            <div className="flex items-center justify-between mb-4">

                <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

                <Link href="/admin/categoryTable/add">

                    <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

                        + Add Category

                    </button>

                </Link>

            </div>

            {/\* Table \*/}

            <div className="overflow-x-auto">

                <table className="min-w-full text-sm text-left">

                    <thead className="bg-gray-50 text-gray-600 border-b">

                        <tr>

                            <th className="px-4 py-2">Image</th>

                            <th className="px-4 py-2">Name</th>

                            <th className="px-4 py-2">Slug</th>

                            <th className="px-4 py-2 text-right">Actions</th>

                        </tr>

                    </thead>

                    <tbody className="text-gray-700">

                        {categories.map((cat) => (

                            <tr key={cat.id} className="border-b hover:bg-gray-50">

                                <td className="px-4 py-3">

                                    <img

                                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/category/${cat.image}`}

                                        alt={cat.name}

                                        className="w-12 h-12 object-cover rounded"

                                    />

                                </td>

                                <td className="px-4 py-3">{cat.name}</td>

                                <td className="px-4 py-3">{cat.slug}</td>

                                <td className="px-4 py-3 text-right space-x-2">

                                    <Link href={`/admin/categoryTable/edit/${cat.\_id}`}>

                                        <button className="text-blue-600 hover:text-blue-800 cursor-pointer transition">

                                            <FiEdit className='text-xl' />

                                        </button>

                                    </Link>

                                    <StatusBtn url="category" status={cat.status} id={cat.\_id} />

                                    <DeleteBtn url="category" id={cat.\_id} />

                                </td>

                            </tr>

                        ))}

                    </tbody>

                </table>

            </div>

        </div>

    );

}

**Ab delete confirm message banayenge**

Iske liye package ka use karenge frontend me

npm i sweetalert2

update deleteBtn

'use client'

import { axiosInstance, notify } from '@/library/helper'

import { useRouter } from 'next/navigation'

import React from 'react'

import { FiTrash2 } from 'react-icons/fi';

import Swal from 'sweetalert2'

export default function DeleteBtn({ id, url }) {

  const router = useRouter()

  function deleteHandler() {

    //confirm message code start

    Swal.fire({

      title: "Are you sure?",

      text: "You won't be able to revert this!",

      icon: "warning",

      showCancelButton: true,

      confirmButtonColor: "#3085d6",

      cancelButtonColor: "#d33",

      confirmButtonText: "Yes, delete it!"

      //confirm message code end

    }).then((result) => {

      if (result.isConfirmed) {

        Swal.fire({

          title: "Deleted!",

          text: "Your file has been deleted.",

          icon: "success"

        });

        axiosInstance.delete(`${url}/delete/${id}`).then(

          (response) => {

            notify(response.data.message, response.data.success)

            console.log(response.data.message, response.data.success);

            if (response.data.success) {

              router.refresh()

            }

          }

        ).catch((error) => {

          console.log(error)

        })

      }

    });

  }

  return (

    <button onClick={deleteHandler} className="text-red-600 hover:text-red-800 transition rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium cursor-pointer hover:bg-gray-200">

      <FiTrash2 />

    </button>

  )

}

Yanah hamane vo code kiya hai jab hum category delete karenge to confirm message show hoga successfully kaam kar raha hai

commit

Note : status and delete button admin component me rahenge

**Ab hum product table par view ka button banayenge**

**Iske liye /admin/component/ProductBtn.jsx me jao**

'use client'

import { useRouter } from 'next/navigation'

import { axiosInstance, notify } from '@/library/helper'

import React, { useState } from 'react'

export default function ProductBtn({ product }) {

    const [toggle, setToggle] = useState(null)

    const router = useRouter()

    function statusHandler(flag) {

        axiosInstance.patch(`product/status/${product.\_id}`, { flag }).then(

            (response) => {

                notify(response.data.message, response.data.success)

                if (response.data.success) {

                    router.refresh()

                }

            }

        ).catch((error) => {

            console.log(error)

        })

    }

    return (

        <td className="whitespace-nowrap pl-10 py-3 space-x-2">

            <button onClick={() => statusHandler(1)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                {

                    product.status ?

                        "Active"

                        :

                        "Inactive"

                }

            </button>

            <button onClick={() => statusHandler(2)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                {

                    product.stock ?

                        "In"

                        :

                        "Out"

                }

            </button>

            <button onClick={() => statusHandler(3)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                {

                    product.topSelling ?

                        "Yes"

                        :

                        "No"

                }

            </button>

            <button className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                Delete

            </button>

            <button onClick={() => setToggle(product)} className="rounded-md bg-gray-100 px-3 py-1.5 text-xs font-medium hover:bg-gray-200">

                View

            </button>

            {

                toggle && <ProductView product={product} close={() => setToggle(null)} />

            }

        </td>

    )

}

function ProductView({ product, close }) {

    return (

        <div className='w-full p-10 h-[300px] fixed bottom-0 left-0 bg-teal-100'>

            <button onClick={close}>Close</button>

            <div

                className="product-long-description"

                dangerouslySetInnerHTML={{ \_\_html: product.longDescription }}

            />

        </div>

    )

}

Aur fir product table me call kar lo

// components/CategoryTable.jsx

import ProductBtn from '@/components/admin-components/ProductBtn';

import DeleteBtn from '@/components/admin-components/DeleteBtn';

import StatusBtn from '@/components/admin-components/StatusBtn';

import { getProducts } from '@/library/api-calls';

import { axiosInstance } from '@/library/helper';

import Link from 'next/link';

import { FiEdit } from 'react-icons/fi';

import ProductView from '@/components/admin-components/ProductView';

export default async function product() {

  const product = await getProducts()

  const products = product.data

  // console.log(products);

  // Dummy data (you can replace this with real API data)

  return (

    <div className="bg-white shadow rounded-lg p-6">

      {/\* Header \*/}

      <div className="flex items-center justify-between mb-4">

        <h2 className="text-xl font-semibold text-gray-800">Categories</h2>

        <Link href="/admin/product/add">

          <button className="bg-blue-600 text-white px-4 py-2 text-sm rounded hover:bg-blue-700 transition">

            + Add Category

          </button>

        </Link>

      </div>

      {/\* Table \*/}

      <div className="overflow-x-auto">

        <table className="min-w-full text-sm text-left">

          <thead className="bg-gray-50 text-gray-600 border-b">

            <tr>

              <th className="px-4 py-2">Image</th>

              <th className="px-4 py-2">Name</th>

              <th className="px-4 py-2">Slug</th>

              <th className="px-4 py-2 text-right">Actions</th>

            </tr>

          </thead>

          <tbody className="text-gray-700">

            {products.map((product) => (

              <tr key={product.id} className="border-b hover:bg-gray-50">

                <td className="px-4 py-3">

                  <img

                    src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                    alt={product.name}

                    className="w-12 h-12 object-cover rounded"

                  />

                </td>

                <td className="px-4 py-3">{product.name}</td>

                <td className="px-4 py-3">{product.slug}</td>

                <td className="px-4 py-3 text-right space-x-2">

                  <ProductBtn product={product} />

                </td>

              </tr>

            ))}

          </tbody>

        </table>

      </div>

    </div>

  );

}

Yanha hamara product view button successfully kaam kar raha hai

Commit

01-09-25

Abhi hum /admin se admin ke kisi bhi page par ja sakte hai

Hum middle ware ka use karenge

**/src/middleware.js** ko banana hai

import { NextResponse } from 'next/server';

export function middleware(req) {

  const { pathname } = req.nextUrl;

  console.log(req.nextUrl);

  if (pathname.startsWith('/admin')) {

    const admin\_token = req.cookies.get('admin\_token');

    if (!admin\_token) {

      return NextResponse.redirect(new URL('/admin-login', req.url));

    }

  }

  return NextResponse.next();

}

export const config = {

  matcher: '/admin/:path\*',

};

Note : jab bhi request jaati hai network se hokar jaati hai network me pata chal jata hai ki kya pass ho raha hai.

**Admin login page banana hai**

**/src/app/admin-login/page.jsx**

**Aur layout.jsx le lena hai /admin-login folder me**

import React from 'react';

const AdminLogin = () => {

  return (

    <div className="min-h-screen flex items-center justify-center bg-gray-100">

      <div className="w-full max-w-md bg-white shadow-md rounded-lg p-8">

        <h2 className="text-2xl font-bold text-center text-gray-800 mb-6">Admin Login</h2>

        <form className="space-y-5">

          <div>

            <label className="block text-sm font-medium text-gray-700 mb-1" htmlFor="email">

              Email

            </label>

            <input

              id="email"

              type="email"

              placeholder="admin@example.com"

              className="w-full px-4 py-2 border rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500"

            />

          </div>

          <div>

            <label className="block text-sm font-medium text-gray-700 mb-1" htmlFor="password">

              Password

            </label>

            <input

              id="password"

              type="password"

              placeholder="••••••••"

              className="w-full px-4 py-2 border rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500"

            />

          </div>

          <button

            type="submit"

            className="w-full bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700 transition"

          >

            Login

          </button>

        </form>

        <p className="text-center text-sm text-gray-500 mt-6">

          © 2025 Admin Panel. All rights reserved.

        </p>

      </div>

    </div>

  );

};

export default AdminLogin;

**backend me model banana hai /admin.model.js banana hai**

const mongoose = require('mongoose')

const AdminSchema = new mongoose.Schema(

  {

    email: {

      type: String,

      required: true,

      unique: true,

      lowercase: true,

      trim: true,

    },

    password: {

      type: String,

      required: true,

      minlength: 6,

    },

    role: {

      type: String,

      enum: ["superadmin", "admin", "manager"],

      default: 'admin',

    },

    status: {

        type: Boolean,

        default: true,

    },

  },

  {

    timestamps: true,

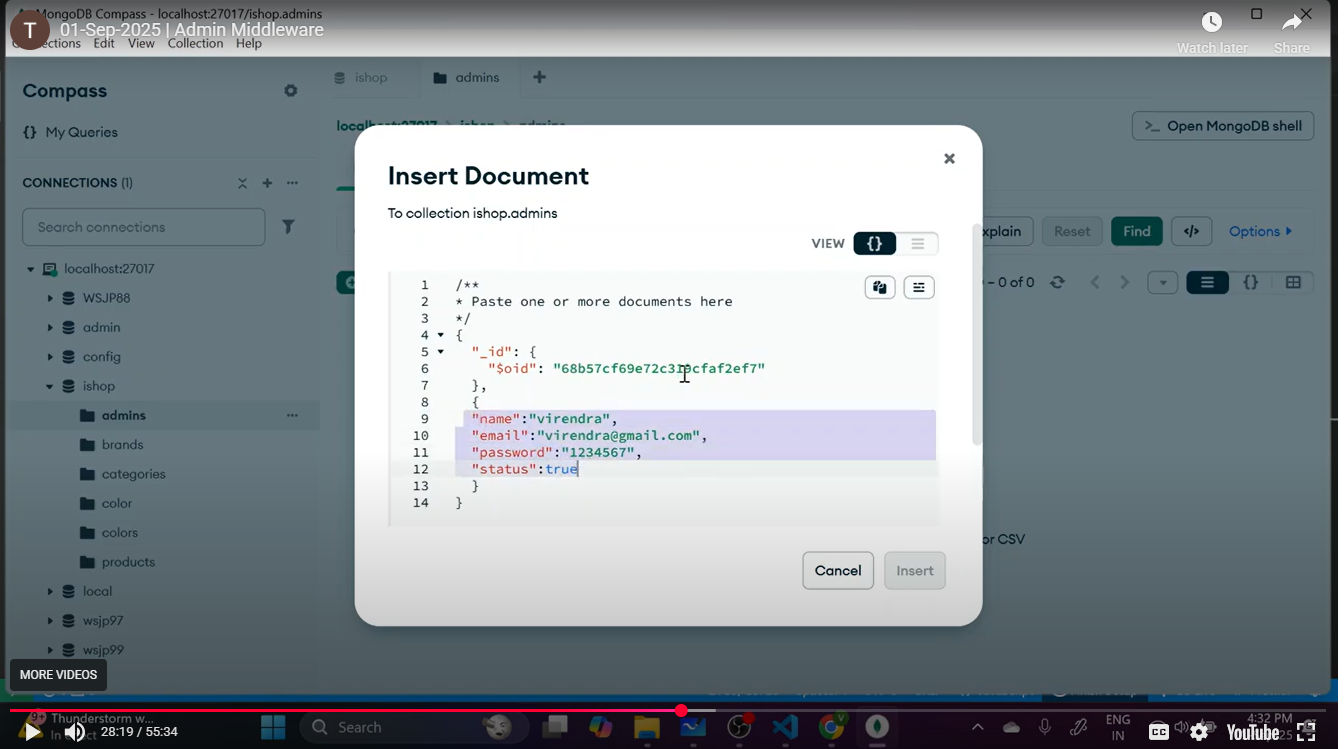
  }

);

const adminModel = mongoose.model("Admin", AdminSchema)

module.exports = adminModel

ab admin ko create kar lenge directe kyonki admin ko supar admin bana sakta hai isliye hum direct add kar reahe hai



Id mongodb khud se generate karega hame bas object insert karna hai jo box me hai upar

Ab **admin.router.js** bana lenge

const adminRouter = require('express').Router();

const { login } = require('../controller/admin.controller');

adminRouter.post('/login', login)

module.exports = adminRouter

const adminRouter = require('express').Router();

const { login } = require('../controller/admin.controller');

adminRouter.post('/login', login)

module.exports = adminRouter

ab **index .j**s ko update kar lenge

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

const categoryRouter = require('./router/category.router')

const colorRouter = require('./router/color.router')

const brandRouter = require('./router/brand.router')

const productRouter = require('./router/product.router')

const adminRouter = require('./router/admin.router')

const server = express();

server.use(cors({origin: "http://localhost:3000"}))

server.use(express.json())

server.use('/category', categoryRouter)

server.use('/color', colorRouter)

server.use('/brand', brandRouter)

server.use('/product', productRouter)

server.use('/admin', adminRouter)

server.use(express.static('public'));

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect(process.env.DATABASE\_URL, {dbName: "sishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

Ab **admin.controller.js** ko bana lenge

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

const admin = {

    async login(req, res) {

        try {

            console.log("Login route hit"); // ✅ Check if route is reached

            console.log(req.body);          // ✅ Should show the request body

            // return successResponse(res, "Login route working");

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

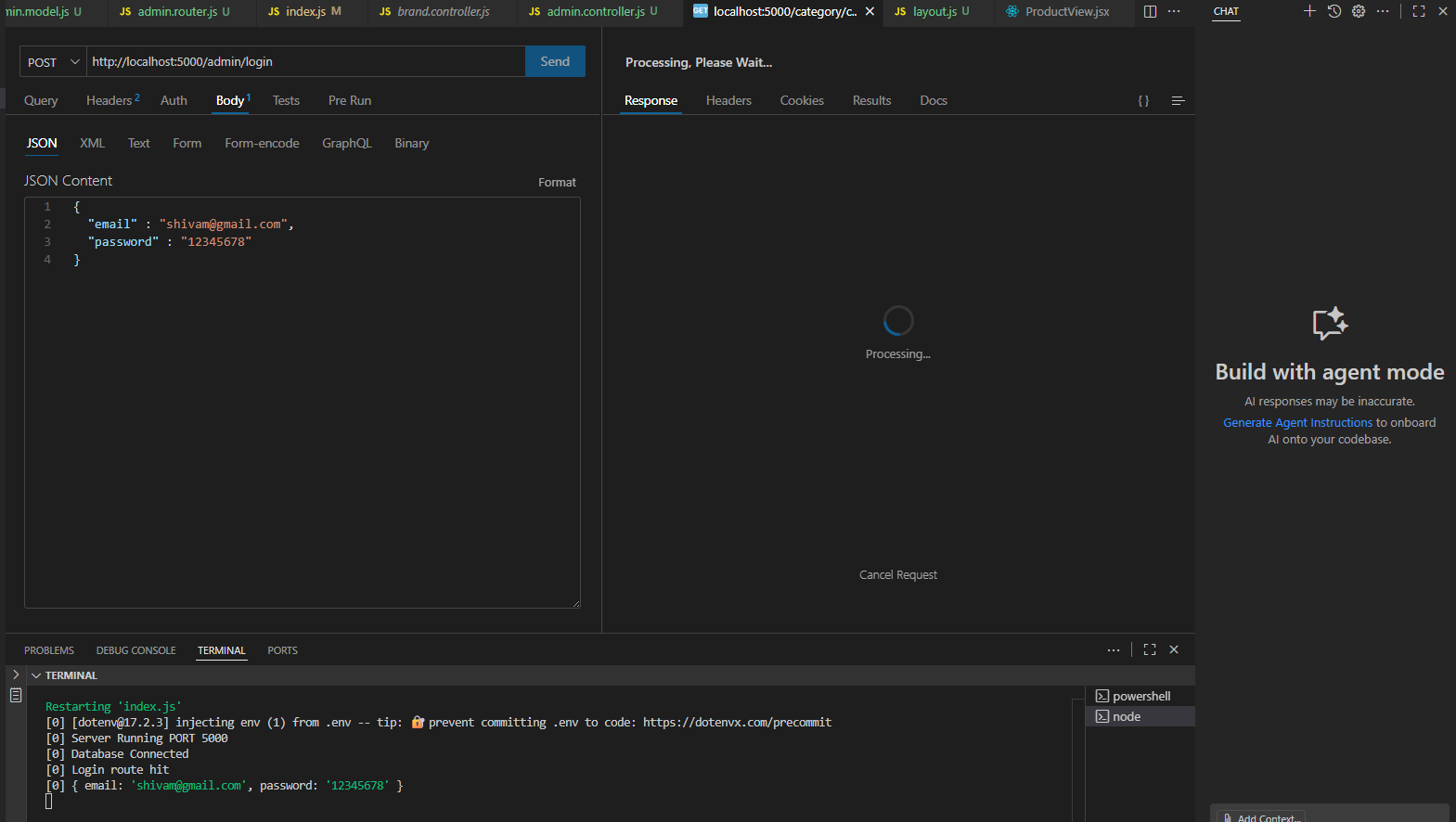
    }

}

module.exports = admin

test on thunder working or not

processing aa raha hai kyonki abhi response pass nahi kiya hai



yanha tak to sahi kaam kar raha hai req.body se email aur password object me aa raha rahe hai console me

commit

Ab response ke saath test karenge

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

const admin = {

    async login(req, res) {

        try {

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            console.log(admin);

            res.send("hello")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

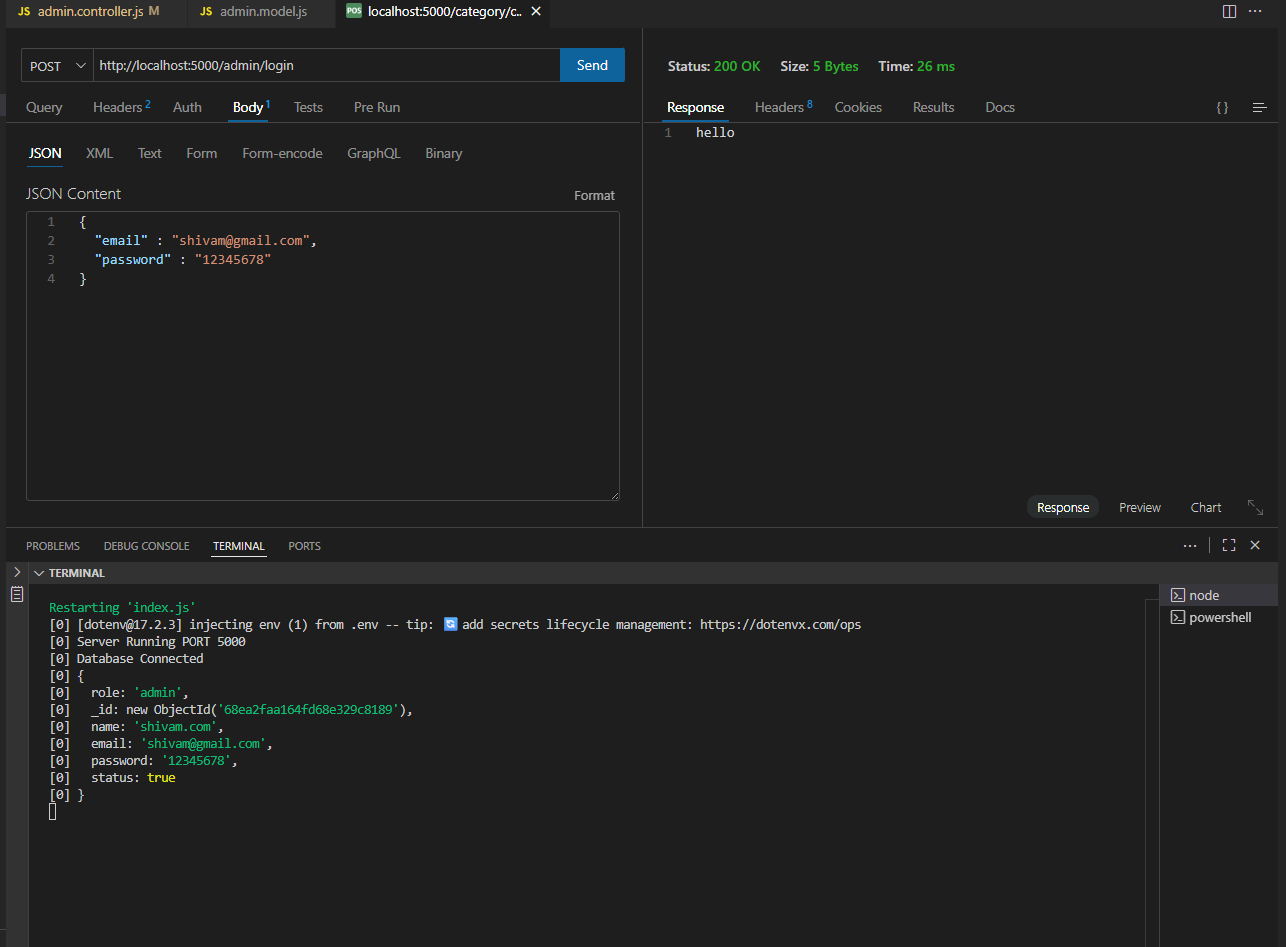
    }

}

module.exports = admin

output me object aana cahiye jab email password sahi ho

mongo db me agar singular collection banayi to vo ise plural me convert kar dega isliye isme hamesha plural me data add kare



Email galat hai to null aayega isliye email match na hone par noContentResponse pass kiya jaata hai “Admin not Found”

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

const admin = {

    async login(req, res) {

        try {

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            console.log(admin);

            res.send("hello")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

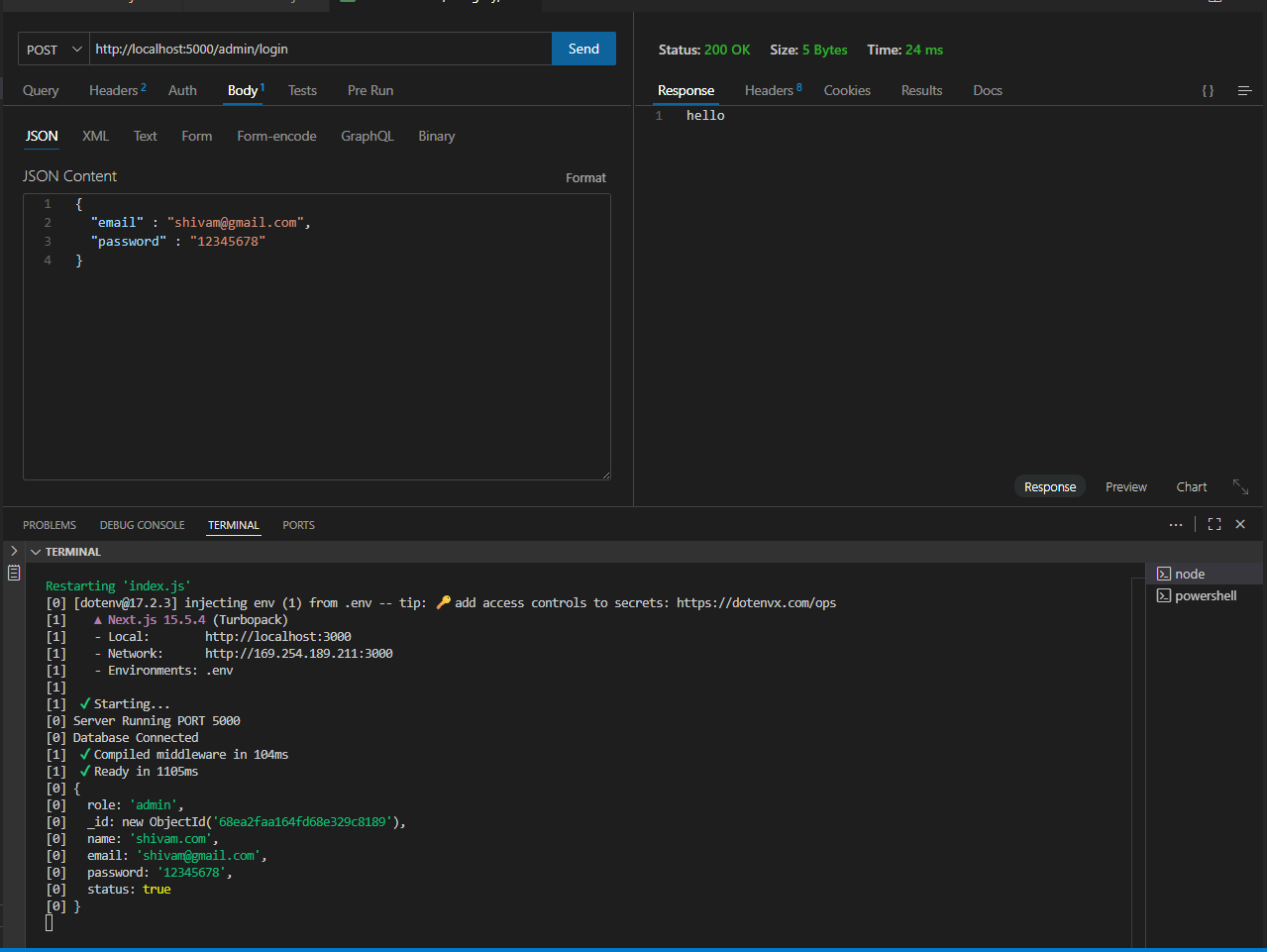
    }

}

module.exports = admin

output null ayega

agar email galat hai ya mongodb se email nahi get ho rahi hai to null aayega



Update admin controller.jsx

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

            return successResponse(res, "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

    }

}

module.exports = admin

now test on thunder



Is tarah login hoga

**Note :** Lekin jab login hoga to ek token bhi to generate hoga aur login kar raha hai uske pass token bhej do uske id ke tor par jese student ki id hoti hai

Us token us student ke bare me sari **jankari pata chal jayegi** ki vo token kab tak **valid** hai aur kya kar sakta hai

Token create karne ke liye **jwt packge** ka use karenge

|  |  |  |
| --- | --- | --- |
| JWT rakhata hai | | |
| Token | Id | validity |

Backend me install kar lena hai

npm i jsonwebtoken

**jwt ko import kar lo janha bhi import karna ho**

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

var jwt = require('jsonwebtoken');

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

                //add jwt function jo token ko create karega

            jwt.sign({

                id : admin.\_id,

                email: admin.email

            }, process.env.TOKEN\_SECRET\_KEY, { expiresIn: 60 \* 60 });

            return successResponse(res, "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

    }

}

module.exports = admin

secret key backend ke .env me difne ki gai hai

DATABASE\_URL = mongodb://localhost:27017/

TOKEN\_SECRET\_KEY=WSCUBETECH@99

Controller me update karke token ki validity define karna aur token ko ek variable me dalana

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

var jwt = require('jsonwebtoken');

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

                //add jwt function jo token ko create karega

           const token =  jwt.sign({

                id : admin.\_id,

                email: admin.email

            }, process.env.TOKEN\_SECRET\_KEY, { expiresIn: 7 \* 24 \* 60 \* 60 \* 1000 });

            return successResponse(res,  "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

    }

}

module.exports = admin

ab hamara token create ho gaya hai agar token ko dekhana hai to

update admin.controller.js

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

var jwt = require('jsonwebtoken');

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

                //add jwt function jo token ko create karega

           const token =  jwt.sign({

                id : admin.\_id,

                email: admin.email

            }, process.env.TOKEN\_SECRET\_KEY, { expiresIn: 7 \* 24 \* 60 \* 60 \* 1000 });

            return successResponse(res, token,  "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

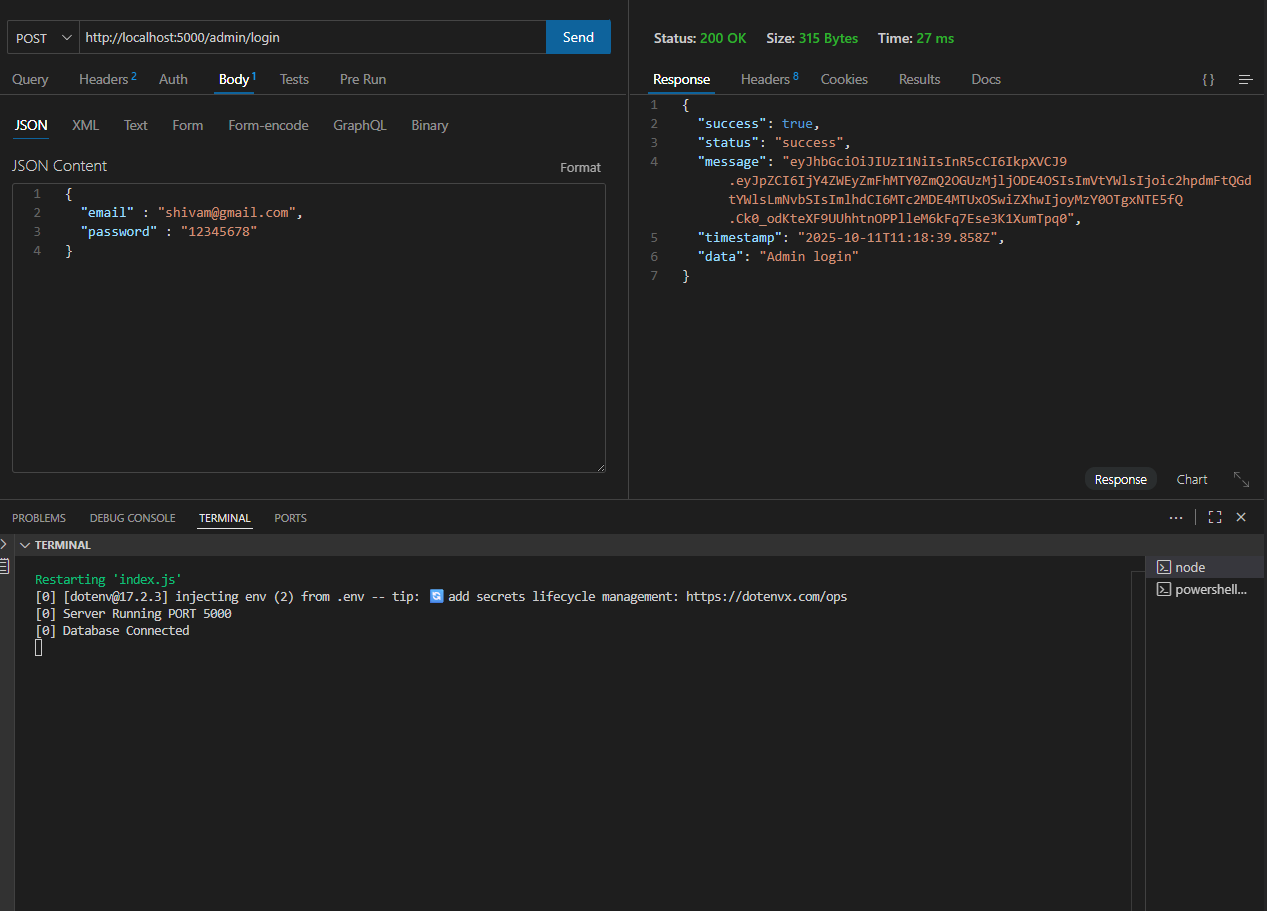
        }

    }

}

module.exports = admin

now test on thunder



upar diye code se token ban gaya aur print bhi karwa diya hai thunder par lekin ab hame ise client end me kese save karayenge vo janenges

commit

three methods hai

**localstorage**

**session storage**

**cookies**

best option hai cookies me token ko store karna kyonki local storage me history delete hote hi sab delete ho jata hai to fir se hame login karna padega

02-09-25 day 12

Cookies me token save karwane ke liye ek package install karenge backend

npm i cookie-parser

Jab req jayegi to cooki parser server ke pass token lekar jayega

Update index.js

require('dotenv').config()

const express = require('express')

const mongoose = require('mongoose')

const cors = require('cors')

var cookieParser = require('cookie-parser')

const categoryRouter = require('./router/category.router')

const colorRouter = require('./router/color.router')

const brandRouter = require('./router/brand.router')

const productRouter = require('./router/product.router')

const adminRouter = require('./router/admin.router')

const server = express();

server.use(cors({origin: "http://localhost:3000", credentials : true}))

server.use(cookieParser())

server.use(express.json())

server.use('/category', categoryRouter)

server.use('/color', colorRouter)

server.use('/brand', brandRouter)

server.use('/product', productRouter)

server.use('/admin', adminRouter)

server.use(express.static('public'));

server.listen(

    5000,

    ()=> {

        console.log('Server Running PORT 5000');

        mongoose.connect(process.env.DATABASE\_URL, {dbName: "sishop"}).then(

            ()=> {

                console.log("Database Connected");

            }

        ).catch(()=> {

            console.log('unable to connect database');

        })

    }

)

Update admin.controller.js cooki ko create karwane ke liye

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

var jwt = require('jsonwebtoken');

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

            //add jwt function jo token ko create karega

            const token = jwt.sign({

                id: admin.\_id,

                email: admin.email

            }, process.env.TOKEN\_SECRET\_KEY, { expiresIn: '7d' });

            res.cookie('admin\_token', token, {

                maxAge: 60 \* 60, // 1 hour

                httpOnly: false,    // Not accessible by JS

                secure: false,    // Only sent over HTTPS

                samesite: 'strict'    // CSRF protection

            });

            return successResponse(res, token, "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

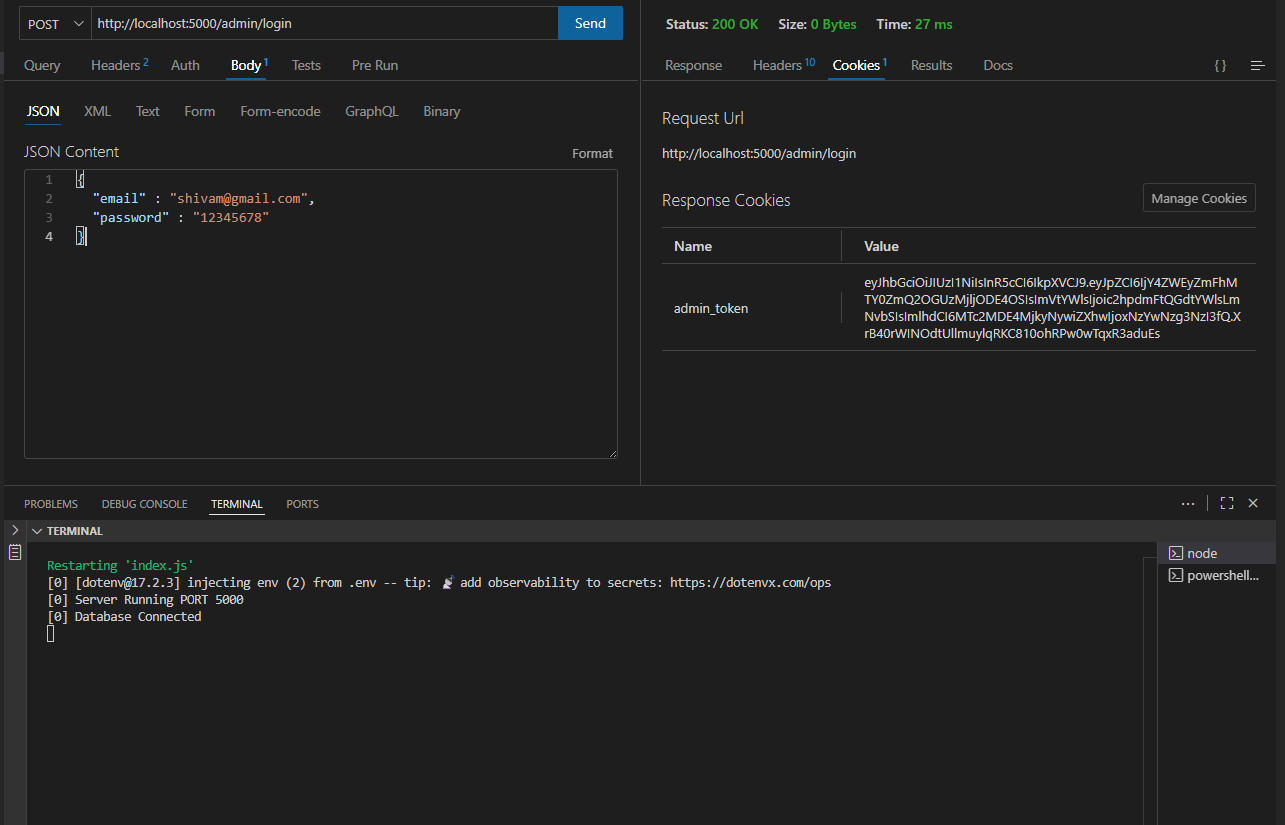
    }

}

module.exports = admin

jab ham pahale thunder par check kar rahe the bina cooki parser ke add kare to thunder cooke me cooki nahi aa raha par jab hamane ise add kar liya tab aa jayega

test on thunder



Ab frontend me cookie ko display karwana hai to ab frontend me code karenge

/admin-login/page.jsx

'use client'

import { axiosInstance, notify } from '@/library/helper';

import React from 'react';

const AdminLogin = () => {

  function loginHandler(e) {

    e.preventDefault();

    const data = {

      email: e.target.email.value,

      password: e.target.password.value

    }

    axiosInstance.post('admin/login', data).then(

      (response) => {

        notify(response.data.message, response.data.success)

        if (response.data.success) {

          console.log(response.data)

        }

      }

    ).catch(

      (error) => {

        notify("Something else", 0)

      }

    )

  }

  return (

    <div className="min-h-screen flex items-center justify-center bg-gray-100">

      <div className="w-full max-w-md bg-white shadow-md rounded-lg p-8">

        <h2 className="text-2xl font-bold text-center text-gray-800 mb-6">Admin Login</h2>

        <form onSubmit={loginHandler} className="space-y-5"  >

          <div>

            <label className="block text-sm font-medium text-gray-700 mb-1" htmlFor="email">

              Email

            </label>

            <input

              name="email"

              id="email"

              type="email"

              placeholder="admin@example.com"

              className="w-full px-4 py-2 border rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500"

            />

          </div>

          <div>

            <label className="block text-sm font-medium text-gray-700 mb-1" htmlFor="password">

              Password

            </label>

            <input

              name='password'

              id="password"

              type="password"

              placeholder="••••••••"

              className="w-full px-4 py-2 border rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500"

            />

          </div>

          <button

            type="submit"

            className="w-full bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700 transition"

          >

            Login

          </button>

        </form>

        <p className="text-center text-sm text-gray-500 mt-6">

          © 2025 Admin Panel. All rights reserved.

        </p>

      </div>

    </div>

  );

};

export default AdminLogin;

admin/layout.js me toastify ko globally rakh denge

jab hum admin-login me email aur password enter karenge to toastify success me token bhi dikh jayega

import { Geist, Geist\_Mono } from "next/font/google";

import "./../globals.css";

import { ToastContainer } from "react-toastify";

const geistSans = Geist({

  variable: "--font-geist-sans",

  subsets: ["latin"],

});

const geistMono = Geist\_Mono({

  variable: "--font-geist-mono",

  subsets: ["latin"],

});

export const metadata = {

  title: "Create Next App",

  description: "Generated by create next app",

};

export default function RootLayout({ children }) {

  return (

    <html lang="en">

      <body

        className={`${geistSans.variable} ${geistMono.variable} antialiased`}

      >

        <ToastContainer

      position="top-right"

          autoClose={5000}

          hideProgressBar={false}

          newestOnTop={false}

          closeOnClick={false}

          rtl={false}

          pauseOnFocusLoss

          draggable

          pauseOnHover

          theme="light"

/>

        {children}

      </body>

    </html>

  );

}

Update admin.controller

Second number token pass kar rakha hai isliye testify success me tokne message aa raha hai

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

var jwt = require('jsonwebtoken');

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

            //add jwt function jo token ko create karega

            const token = jwt.sign({

                id: admin.\_id,

                email: admin.email

            }, process.env.TOKEN\_SECRET\_KEY, { expiresIn: '7d' });

            res.cookie('admin\_token', token, {

                maxAge: 60 \* 60, // 1 hour

                httpOnly: false,    // Not accessible by JS

                secure: false,    // Only sent over HTTPS

                samesite: 'strict'    // CSRF protection

            });

            return successResponse(res,  "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

    }

}

module.exports = admin

Ab hame login karne ke baad amin page par redirect karna hai to hum router ka use karenge

Update admin/page.jsx

'use client'

import { axiosInstance, notify } from '@/library/helper';

import { useRouter } from 'next/navigation';

import React from 'react';

const AdminLogin = () => {

  const router = useRouter()

  function loginHandler(e) {

    e.preventDefault();

    const data = {

      email: e.target.email.value,

      password: e.target.password.value

    }

    axiosInstance.post('admin/login', data, { withCredentials : true }).then(

      (response) => {

        notify(response.data.message, response.data.success)

        if (response.data.success) {

          // console.log(response.data)

          router.push('/admin')

        }

      }

    ).catch(

      (error) => {

        notify("Something else", 0)

      }

    )

  }

  return (

    <div className="min-h-screen flex items-center justify-center bg-gray-100">

      <div className="w-full max-w-md bg-white shadow-md rounded-lg p-8">

        <h2 className="text-2xl font-bold text-center text-gray-800 mb-6">Admin Login</h2>

        <form onSubmit={loginHandler} className="space-y-5"  >

          <div>

            <label className="block text-sm font-medium text-gray-700 mb-1" htmlFor="email">

              Email

            </label>

            <input

              name="email"

              id="email"

              type="email"

              placeholder="admin@example.com"

              className="w-full px-4 py-2 border rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500"

            />

          </div>

          <div>

            <label className="block text-sm font-medium text-gray-700 mb-1" htmlFor="password">

              Password

            </label>

            <input

              name='password'

              id="password"

              type="password"

              placeholder="••••••••"

              className="w-full px-4 py-2 border rounded-md focus:outline-none focus:ring-2 focus:ring-blue-500"

            />

          </div>

          <button

            type="submit"

            className="w-full bg-blue-600 text-white py-2 rounded-md hover:bg-blue-700 transition"

          >

            Login

          </button>

        </form>

        <p className="text-center text-sm text-gray-500 mt-6">

          © 2025 Admin Panel. All rights reserved.

        </p>

      </div>

    </div>

  );

};

export default AdminLogin;

cookies me token aa gaya hai

update admin.controller

token ki age badana

const adminModel = require('../model/admin.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, deletedResponse, updatedResponse } = require('../utility/response')

var jwt = require('jsonwebtoken');

const admin = {

    async login(req, res) {

        try {

            // console.log(req.body);

            const { email, password } = req.body;

            const admin = await adminModel.findOne({ email: email });

            if (!admin) return errorResponse(res, "Admin not found")

            if (password !== admin.password) return errorResponse(res, "Password not match")

            //add jwt function jo token ko create karega

            const token = jwt.sign({

                id: admin.\_id,

                email: admin.email

            }, process.env.TOKEN\_SECRET\_KEY, { expiresIn: '7d' });

            res.cookie('admin\_token', token, {

                maxAge: 7 \* 24 \* 1000 \* 60 \* 60, // 1 hour

                httpOnly: false,    // Not accessible by JS

                secure: false,    // Only sent over HTTPS

                samesite: 'strict'    // CSRF protection

            });

            return successResponse(res,  "Admin login")

        } catch (error) {

            return serverErrorResponse(res, error.message);

        }

    }

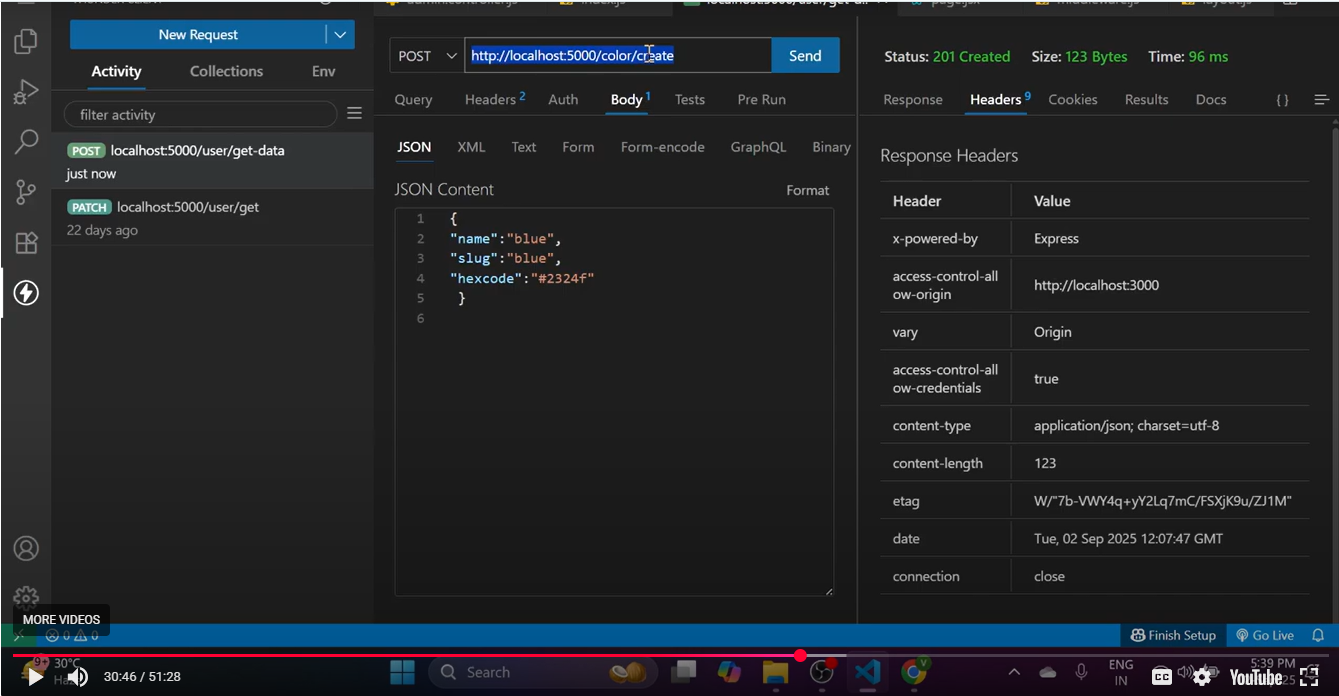
}

module.exports = admin

yanha token complete hota hai frontend me aur login karne ke baad admin par redirect ho rahe hai

commit

Ab hame lagata hai ki hamane admin par lock laga diya jab admin open karege tabhi hum api me change kar sakte hai kuch product, color and brand, etc add kar sakte hai lekin esa nahi hai hum thunder se yah sab chije kar sakte hai



To ham chahate hai ki jab bhi api me change ho token pass hona chahiye jisase kevel admin hi update kar sake

Iske liye hum backend me /middleware/authMiddleware.js banayenge

const jwt = require('jsonwebtoken');

const { errorResponse } = require('../utility/response');

const authMiddleware = (req, res, next) => {

    const token = req.headers.authorization;

    console.log(token);

    if (!token) {

        return errorResponse(res, 'No token provided, authorization denied', 401)

    }

    try {

        const decoded = jwt.verify(token, process.env.TOKEN\_SECRET\_KEY);

        req.user = decoded;

        next();

    } catch (err) {

        return errorResponse(res, 'Token is not valid', 401)

    }

};

module.exports = authMiddleware;

update helper.js

import { ToastContainer, toast } from "react-toastify";

const notify = (msg, flag) => toast(msg, {type: flag ? 'success' : 'error'})

import axios from "axios";

const axiosInstance = axios.create({

    baseURL: process.env.NEXT\_PUBLIC\_API\_BASE\_URL

})

function createSlug(text) {

    return text

        .toString()              // Ensure it's a string

        .trim()                  // Remove extra spaces

        .toLowerCase()           // Convert to lowercase

        .replace(/[^a-z0-9\s-]/g, '')  // Remove special chars

        .replace(/\s+/g, '-')          // Replace spaces with dashes

        .replace(/-+/g, '-');          // Replace multiple dashes with single

}

function getCookies(name) {

  if (typeof document === 'undefined') return null;

  const value = `; ${document.cookie}`;

  const parts = value.split(`; ${name}=`);

  if (parts.length === 2) return parts.pop().split(';').shift();

  return null;

}

export { createSlug, notify, axiosInstance, getCookies }

update /middleware/authMiddleware.js

const jwt = require('jsonwebtoken');

const { errorResponse } = require('../utility/response');

const authMiddleware = (req, res, next) => {

    const token = req.headers.authorization;

    console.log(token);

    if (!token) {

        return errorResponse(res, 'No token provided, authorization denied', 401)

    }

    try {

        const decoded = jwt.verify(token, process.env.TOKEN\_SECRET\_KEY);

        req.user = decoded;

        next();

    } catch (err) {

        return errorResponse(res, 'Token is not valid', 401)

    }

};

module.exports = authMiddleware;

ab hame is middleware ko lagana hai ham chahiye to ise globally bhi laga sakte hai lekin get api sab access kar sake isliye hum kuch route par hi lagayenge

to hum post, update, delete par hi lagayenge

pahale hum lagayenge

color.router.js me

update color.router.js

ab hum front end se color create karenge to hum nahi create kar sakte aur na hi thunder se nahi kar sakte hai

abhi hamane color ki post par middlware lagaya hai to post wali ko update nahi kar sakte hai

axios error aayegi

const colorRouter = require('express').Router();

const { create, read, deleteById, status } = require('../controller/color.controller');

const authMiddleware = require('../middleware/authMiddleware');

colorRouter.post('/create', authMiddleware, create)

colorRouter.get('/:id?', read);

colorRouter.delete('/delete/:id?', deleteById);

colorRouter.patch('/status/:id?', status);

module.exports = colorRouter;

ab hum front end me color add/page.jsx me token pass karwana chahate hai

update add/page.jsx

'use client'

import { createSlug, notify, axiosInstance, getCookies } from '@/library/helper';

import axios from 'axios';

import React, { useRef } from 'react'

import { FaRegImage } from "react-icons/fa6";

export default function page() {

    const token = getCookies('admin\_token')

    // console.log(token);

    const nameRef = useRef()

    const slugRef = useRef()

    function generateSlug() {

        // console.log(nameRef.current.value);

        const slug = createSlug(nameRef.current.value)

        // console.log(slug);

        slugRef.current.value = slug

    }

    function submitHandler(e) {

        e.preventDefault()

        const data = {

            name: nameRef.current.value,

            slug: slugRef.current.value,

            hexcode: e.target.hexcode.value

        }

        axiosInstance.post("color/create", data, {

            headers : {

                Authorization : token

            }

        }).then(

            (response) => {

                // console.log(response.data.success);

                notify(response.data.message, { flag: response.data.success })

                if (response.data.success); {

                    nameRef.current.value = ""

                    slugRef.current.value = ""

                }

            }

        ).catch((error) => {

            console.log(error);

        })

    }

    return (

        <div>

            <form className="max-w-sm mx-auto" onSubmit={submitHandler}>

                <div className="mb-5">

                    <label

                        htmlFor="text"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Category Name :

                    </label>

                    <input

                        type="text"

                        id="text"

                        ref={nameRef}

                        onChange={generateSlug}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder="Category name"

                        required=""

                    />

                </div>

                <div className="mb-5">

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Slug :

                    </label>

                    <input

                        type="text"

                        id="slug"

                        ref={slugRef}

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                    <label

                        htmlFor="slug"

                        className="block mb-2 text-sm font-medium text-gray-900 dark:text-white"

                    >

                        Color Code

                    </label>

                    <input

                        type="color"

                        id="slug"

                        name="hexcode"

                        className="bg-gray-50 border border-gray-300 text-gray-900 text-sm rounded-lg focus:ring-blue-500 focus:border-blue-500 block w-full p-2.5 dark:bg-gray-700 dark:border-gray-600 dark:placeholder-gray-400 dark:text-white dark:focus:ring-blue-500 dark:focus:border-blue-500"

                        placeholder='slug'

                        required=""

                    />

                </div>

                <button

                    type="submit"

                    className="text-white bg-blue-700 hover:bg-blue-800 focus:ring-4 focus:outline-none focus:ring-blue-300 font-medium rounded-lg text-sm w-full sm:w-auto px-5 py-2.5 text-center dark:bg-blue-600 dark:hover:bg-blue-700 dark:focus:ring-blue-800"

                >

                    Submit

                </button>

            </form>

        </div>

    )

}

Ab color create kar sakte ho aur color create ho jayega successfully with successfull toastify ke saath

Lekin thunder se nahi kar sakte hai

Update authMiddleware.js

500 ki jagah 401 ki error aani chhiye jo ki sahi hai becoz 401 mean unauthorized

const jwt = require('jsonwebtoken');

const { errorResponse } = require('../utility/response');

const authMiddleware = (req, res, next) => {

    const token = req.headers.authorization;

    console.log(token);

    if (!token) {

        return errorResponse(res, 'No token provided, authorization denied', 401)

    }

    try {

        const decoded = jwt.verify(token, process.env.TOKEN\_SECRET\_KEY);

        req.user = decoded;

        next();

    } catch (err) {

        return errorResponse(res, 'Token is not valid', 401)

    }

};

module.exports = authMiddleware;

ab hama color post api secure hai kyoki email aur password hoga tabhi login kar payega admin nahi to token generate nahi hoga aur color create nahi hoga full secure hai

yanha hamara admin complete hota hai with authMiddleware

commit

kis admin ne create kiya hai vo bhi jaan sakte ho

**03-09-25 day 13**

Front end store page

**Src/app/(website-group)**

**Layout.js**

**Page.jsx**

**Jo bahar layout.js and page.jsx hai unko copy paste karna hai aur css ko update kar lena ha**

**(website-group)/store/**

**Layout.js**

**Page.jsx**

**(website-group)/store/layout.js**

'use client'

export default function RootLayout({ children }) {

    return (

        <html lang="en">

            <body>

                <div className="space-y-10 p-5 bg-gray-50">

                    {/\* 🧭 Breadcrumb Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <ol className="flex items-center whitespace-nowrap space-x-2">

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Home

                                </a>

                                <svg

                                    className="mx-2 w-4 h-4 text-gray-400"

                                    viewBox="0 0 16 16"

                                    fill="none"

                                    xmlns="http://www.w3.org/2000/svg"

                                >

                                    <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                </svg>

                            </li>

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Shop

                                    <svg

                                        className="mx-2 w-4 h-4 text-gray-400"

                                        viewBox="0 0 16 16"

                                        fill="none"

                                        xmlns="http://www.w3.org/2000/svg"

                                    >

                                        <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                    </svg>

                                </a>

                            </li>

                            <li className="text-sm font-semibold text-gray-800 truncate">

                                Top Cell Phones & Tablets

                            </li>

                        </ol>

                    </section>

                    {/\* 📱 Top Cell Phones & Tablets Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Top Cell Phones & Tablets</h1>

                        <div className="flex flex-wrap gap-5">

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/slider3.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-3 text-white">

                                    <h1 className="text-2xl font-bold">Noise Cancelling</h1>

                                    <h3 className="text-lg">Headphone</h3>

                                    <p className="text-sm leading-tight">Boso Over-Ear Headphone <br />Wifi, Voice Assistant,<br />Low latency game mode</p>

                                    <button className="px-4 py-2 bg-white text-black rounded-lg font-semibold">BUY NOW</button>

                                </div>

                            </div>

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/banner1.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-2 text-black">

                                    <h1 className="text-2xl font-bold">Redmi Note 12 <br />Pro+ 5G</h1>

                                    <p className="text-sm text-gray-600">Rise to the challenge</p>

                                    <button className="mt-2 px-4 py-2 bg-black text-white rounded-lg">SHOP NOW</button>

                                </div>

                            </div>

                        </div>

                    </section>

                    {/\* 🗂️ Popular Categories Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Popular Categories</h1>

                        <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-5 gap-6">

                            {[

                                { title: "iPhone (iOS)", items: 74, image: "iphone" },

                                { title: "Android", items: 34, image: "android" },

                                { title: "5G Support", items: 12, image: "5gsupport" },

                                { title: "Apple Tablets", items: 22, image: "apptablet" },

                                { title: "Smartphone Chargers", items: 33, image: "smartphone\_chargers" },

                                { title: "Gaming", items: 9, image: "gaming" },

                                { title: "Xiaomi", items: 52, image: "xiaomi" },

                                { title: "Accessories", items: 29, image: "accessories" },

                                { title: "Samsung Tablets", items: 26, image: "samsung" },

                                { title: "eReader", items: 5, image: "eReader" }

                            ].map((cat, i) => (

                                <div key={i} className="flex justify-between items-center p-3 border rounded-md bg-gray-50">

                                    <div>

                                        <h1 className="font-semibold">{cat.title}</h1>

                                        <p className="text-sm text-gray-500">{cat.items} Items</p>

                                    </div>

                                    <img src={`store/popular\_categories/${cat.image}.png`} alt={cat.title} className="w-16 h-16 object-contain" />

                                </div>

                            ))}

                        </div>

                    </section>

                    {/\* 🧩 Content Area (children) \*/}

                    <section className="border rounded-md p-6 bg-white shadow">

                        <h1 className="text-xl font-bold mb-4">Main Content Area</h1>

                        <div className="flex w-full gap-6">

                            {/\* Left Sidebar: Filters \*/}

                            <div className="w-1/4 space-y-6">

                                {/\* Categories Filter \*/}

                                <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                                    <h2 className="text-lg font-bold">CATEGORIES</h2>

                                    <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                                        All Categories

                                    </button>

                                    <div className="text-sm space-y-1 pl-2">

                                        <p className="text-gray-700">Cell Phones & Tablets</p>

                                        <ul className="space-y-1 pl-3 text-gray-500">

                                            <li className="cursor-pointer hover:text-black">All</li>

                                            <li className="cursor-pointer hover:text-black">Iphone</li>

                                            <li className="cursor-pointer hover:text-black">Samsung</li>

                                            <li className="cursor-pointer hover:text-black">Xiaomi</li>

                                            <li className="cursor-pointer hover:text-black">Asus</li>

                                        </ul>

                                    </div>

                                </div>

                                {/\* Brand Filter \*/}

                                <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                                    <h2 className="text-lg font-bold">BRAND FILTER</h2>

                                    <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                                        All Brands

                                    </button>

                                    <div className="text-sm space-y-1 pl-2">

                                        <p className="text-gray-700">Cell Phones & Tablets</p>

                                        <ul className="space-y-1 pl-3 text-gray-500">

                                            <li className="cursor-pointer hover:text-black">Iphone</li>

                                            <li className="cursor-pointer hover:text-black">Samsung</li>

                                            <li className="cursor-pointer hover:text-black">Xiaomi</li>

                                            <li className="cursor-pointer hover:text-black">Asus</li>

                                        </ul>

                                    </div>

                                </div>

                                {/\* Color Filter \*/}

                                <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                                    <h2 className="text-lg font-bold">COLOR FILTER</h2>

                                    <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                                        All Colors

                                    </button>

                                    <div className="text-sm space-y-1 pl-2">

                                        <p className="text-gray-700">Available Colors</p>

                                        <ul className="space-y-1 pl-3 text-gray-500">

                                            <li className="cursor-pointer hover:text-black">Black</li>

                                            <li className="cursor-pointer hover:text-black">White</li>

                                            <li className="cursor-pointer hover:text-black">Blue</li>

                                            <li className="cursor-pointer hover:text-black">Red</li>

                                            <li className="cursor-pointer hover:text-black">Green</li>

                                        </ul>

                                    </div>

                                </div>

                            </div>

                            {/\* Right Main Content \*/}

                            <div className="w-3/4 bg-red-400 p-4 rounded-md">

                                page

                            </div>

                        </div>

                    </section>

                </div>

            </body>

        </html>

    );

}

**Ab website-component me component bana lo store product ke filter ke liye**

**Website-component/CategoryFilter.jsx**

import React from 'react'

export default function CategoryFilter() {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">CATEGORIES</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Categories

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        <li className="cursor-pointer hover:text-black">All</li>

                        <li className="cursor-pointer hover:text-black">Iphone</li>

                        <li className="cursor-pointer hover:text-black">Samsung</li>

                        <li className="cursor-pointer hover:text-black">Xiaomi</li>

                        <li className="cursor-pointer hover:text-black">Asus</li>

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Website-component/BrandFilter.jsx**

**Website-component/ColorFilter.jsx**

**Ab in CategoryFilter, BrandFilter and ColorFilter ko /storepage/layout.jsx me import karo**

**Update storepage/layout.jsx**

'use client'

import CategoryFilter from "@/components/website-components/CategoryFilter";

import BrandFilter from "@/components/website-components/BrandFilter";

import ColorFilter from "@/components/website-components/ColorFilter";

export default function RootLayout({ children }) {

    return (

        <html lang="en">

            <body>

                <div className="space-y-10 p-5 bg-gray-50">

                    {/\* 🧭 Breadcrumb Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <ol className="flex items-center whitespace-nowrap space-x-2">

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Home

                                </a>

                                <svg

                                    className="mx-2 w-4 h-4 text-gray-400"

                                    viewBox="0 0 16 16"

                                    fill="none"

                                    xmlns="http://www.w3.org/2000/svg"

                                >

                                    <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                </svg>

                            </li>

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Shop

                                    <svg

                                        className="mx-2 w-4 h-4 text-gray-400"

                                        viewBox="0 0 16 16"

                                        fill="none"

                                        xmlns="http://www.w3.org/2000/svg"

                                    >

                                        <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                    </svg>

                                </a>

                            </li>

                            <li className="text-sm font-semibold text-gray-800 truncate">

                                Top Cell Phones & Tablets

                            </li>

                        </ol>

                    </section>

                    {/\* 📱 Top Cell Phones & Tablets Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Top Cell Phones & Tablets</h1>

                        <div className="flex flex-wrap gap-5">

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/slider3.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-3 text-white">

                                    <h1 className="text-2xl font-bold">Noise Cancelling</h1>

                                    <h3 className="text-lg">Headphone</h3>

                                    <p className="text-sm leading-tight">Boso Over-Ear Headphone <br />Wifi, Voice Assistant,<br />Low latency game mode</p>

                                    <button className="px-4 py-2 bg-white text-black rounded-lg font-semibold">BUY NOW</button>

                                </div>

                            </div>

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/banner1.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-2 text-black">

                                    <h1 className="text-2xl font-bold">Redmi Note 12 <br />Pro+ 5G</h1>

                                    <p className="text-sm text-gray-600">Rise to the challenge</p>

                                    <button className="mt-2 px-4 py-2 bg-black text-white rounded-lg">SHOP NOW</button>

                                </div>

                            </div>

                        </div>

                    </section>

                    {/\* 🗂️ Popular Categories Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Popular Categories</h1>

                        <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-5 gap-6">

                            {[

                                { title: "iPhone (iOS)", items: 74, image: "iphone" },

                                { title: "Android", items: 34, image: "android" },

                                { title: "5G Support", items: 12, image: "5gsupport" },

                                { title: "Apple Tablets", items: 22, image: "apptablet" },

                                { title: "Smartphone Chargers", items: 33, image: "smartphone\_chargers" },

                                { title: "Gaming", items: 9, image: "gaming" },

                                { title: "Xiaomi", items: 52, image: "xiaomi" },

                                { title: "Accessories", items: 29, image: "accessories" },

                                { title: "Samsung Tablets", items: 26, image: "samsung" },

                                { title: "eReader", items: 5, image: "eReader" }

                            ].map((cat, i) => (

                                <div key={i} className="flex justify-between items-center p-3 border rounded-md bg-gray-50">

                                    <div>

                                        <h1 className="font-semibold">{cat.title}</h1>

                                        <p className="text-sm text-gray-500">{cat.items} Items</p>

                                    </div>

                                    <img src={`store/popular\_categories/${cat.image}.png`} alt={cat.title} className="w-16 h-16 object-contain" />

                                </div>

                            ))}

                        </div>

                    </section>

                    {/\* 🧩 Content Area (children) \*/}

                    <section className="border rounded-md p-6 bg-white shadow">

                        <h1 className="text-xl font-bold mb-4">Main Content Area</h1>

                        <div className="flex w-full gap-6">

                            {/\* Left Sidebar: Filters \*/}

                            <div className="w-1/4 space-y-6">

                                {/\* Categories Filter \*/}

                                <CategoryFilter />

                                {/\* Brand Filter \*/}

                                <BrandFilter />

                                {/\* Color Filter \*/}

                                <ColorFilter />

                            </div>

                            {/\* Right Main Content \*/}

                            <div className="w-3/4 bg-red-400 p-4 rounded-md">

                                page

                            </div>

                        </div>

                    </section>

                </div>

            </body>

        </html>

    );

}

**Ab website-component me product cartd ka component banana hai**

import React from 'react'

export default function ProductCard() {

    return (

        <div>

            <div className="w-full max-w-xs bg-white border rounded-xl shadow hover:shadow-lg transition p-4 space-y-4">

                {/\* Product Image with 'Top' Badge \*/}

                <div className="relative">

                    <img

                        src="/store/products/product1.png"

                        alt="Product Thumbnail"

                        className="w-full h-48 object-cover rounded-md"

                    />

                    <span className="absolute top-2 left-2 bg-red-500 text-white text-xs font-semibold px-2 py-1 rounded">

                        Top

                    </span>

                </div>

                {/\* Product Info \*/}

                <div className="space-y-2">

                    {/\* Description \*/}

                    <h2 className="text-sm text-gray-700 font-medium line-clamp-2">

                        Wireless Bluetooth Over-Ear Headphones with Noise Cancellation

                    </h2>

                    {/\* Price \*/}

                    <div className="flex items-center gap-2">

                        <span className="text-gray-500 line-through text-sm">₹3,999</span>

                        <span className="text-red-600 font-semibold text-base">₹2,499</span>

                    </div>

                    {/\* Color Options \*/}

                    <div className="flex items-center gap-2">

                        {["bg-black", "bg-red-500", "bg-blue-500", "bg-green-600"].map((color, index) => (

                            <span

                                key={index}

                                className={`w-4 h-4 rounded-full border border-gray-300 ${color} cursor-pointer`}

                            />

                        ))}

                    </div>

                </div>

                {/\* Add to Cart Button \*/}

                <button className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                    Add to Cart

                </button>

            </div>

        </div>

    )

}

**Storepage/layout.js ke right main content me hame store ka home page.jsx chahiye to hum given code ko {children} se replace kar denge taki page.jsx ka content dikhe**

   {/\* Right Main Content \*/}

                            <div className="w-3/4 bg-red-400 p-4 rounded-md">

                                home page

                            </div>

**Update storepage/layout.js**

import CategoryFilter from "@/components/website-components/CategoryFilter";

import BrandFilter from "@/components/website-components/BrandFilter";

import ColorFilter from "@/components/website-components/ColorFilter";

export default function RootLayout({ children }) {

    return (

        <html lang="en">

            <body>

                <div className="space-y-10 p-5 bg-gray-50">

                    {/\* 🧭 Breadcrumb Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <ol className="flex items-center whitespace-nowrap space-x-2">

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Home

                                </a>

                                <svg

                                    className="mx-2 w-4 h-4 text-gray-400"

                                    viewBox="0 0 16 16"

                                    fill="none"

                                    xmlns="http://www.w3.org/2000/svg"

                                >

                                    <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                </svg>

                            </li>

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Shop

                                    <svg

                                        className="mx-2 w-4 h-4 text-gray-400"

                                        viewBox="0 0 16 16"

                                        fill="none"

                                        xmlns="http://www.w3.org/2000/svg"

                                    >

                                        <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                    </svg>

                                </a>

                            </li>

                            <li className="text-sm font-semibold text-gray-800 truncate">

                                Top Cell Phones & Tablets

                            </li>

                        </ol>

                    </section>

                    {/\* 📱 Top Cell Phones & Tablets Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Top Cell Phones & Tablets</h1>

                        <div className="flex flex-wrap gap-5">

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/slider3.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-3 text-white">

                                    <h1 className="text-2xl font-bold">Noise Cancelling</h1>

                                    <h3 className="text-lg">Headphone</h3>

                                    <p className="text-sm leading-tight">Boso Over-Ear Headphone <br />Wifi, Voice Assistant,<br />Low latency game mode</p>

                                    <button className="px-4 py-2 bg-white text-black rounded-lg font-semibold">BUY NOW</button>

                                </div>

                            </div>

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/banner1.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-2 text-black">

                                    <h1 className="text-2xl font-bold">Redmi Note 12 <br />Pro+ 5G</h1>

                                    <p className="text-sm text-gray-600">Rise to the challenge</p>

                                    <button className="mt-2 px-4 py-2 bg-black text-white rounded-lg">SHOP NOW</button>

                                </div>

                            </div>

                        </div>

                    </section>

                    {/\* 🗂️ Popular Categories Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Popular Categories</h1>

                        <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-5 gap-6">

                            {[

                                { title: "iPhone (iOS)", items: 74, image: "iphone" },

                                { title: "Android", items: 34, image: "android" },

                                { title: "5G Support", items: 12, image: "5gsupport" },

                                { title: "Apple Tablets", items: 22, image: "apptablet" },

                                { title: "Smartphone Chargers", items: 33, image: "smartphone\_chargers" },

                                { title: "Gaming", items: 9, image: "gaming" },

                                { title: "Xiaomi", items: 52, image: "xiaomi" },

                                { title: "Accessories", items: 29, image: "accessories" },

                                { title: "Samsung Tablets", items: 26, image: "samsung" },

                                { title: "eReader", items: 5, image: "eReader" }

                            ].map((cat, i) => (

                                <div key={i} className="flex justify-between items-center p-3 border rounded-md bg-gray-50">

                                    <div>

                                        <h1 className="font-semibold">{cat.title}</h1>

                                        <p className="text-sm text-gray-500">{cat.items} Items</p>

                                    </div>

                                    <img src={`store/popular\_categories/${cat.image}.png`} alt={cat.title} className="w-16 h-16 object-contain" />

                                </div>

                            ))}

                        </div>

                    </section>

                    {/\* 🧩 Content Area (children) \*/}

                    <section className="border rounded-md p-6 bg-white shadow">

                        <h1 className="text-xl font-bold mb-4">Main Content Area</h1>

                        <div className="flex w-full gap-6">

                            {/\* Left Sidebar: Filters \*/}

                            <div className="w-1/4 space-y-6">

                                {/\* Categories Filter \*/}

                                <CategoryFilter />

                                {/\* Brand Filter \*/}

                                <BrandFilter />

                                {/\* Color Filter \*/}

                                <ColorFilter />

                            </div>

                            {/\* Right Main Content \*/}

                            <div className="w-3/4 bg-red-400 p-4 rounded-md">

                                {children}

                            </div>

                        </div>

                    </section>

                </div>

            </body>

        </html>

    );

}

**Ab storepage/page.js me website-component/ProductCard.jsx ko import karenge**

import ProductCard from '@/components/website-components/ProductCard'

import React from 'react'

export default function page() {

    return (

        <div>

            <ProductCard />

        </div>

    )

}

**Ab storepage.jsx me product ko get karenge api se**

**Update storepage/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page() {

    const productJSON = await getProducts(null)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map(product => (

                    <ProductCard key={product.id} product={product} />

                ))

            }

        </div>

    )

}

**Ab website-component/ProductCard.jsx me props pass kiay hai use update karenge**

**Update website-component/ProductCard.jsx**

import React from 'react'

export default function ProductCard({ product }) {

    const { name, finalPrice, discountPercentage, originalPrice, thumbnail } = product;

    return (

        <div>

            <div className="w-full max-w-xs bg-white border rounded-xl shadow hover:shadow-lg transition p-4 space-y-4">

                {/\* Product Image with 'Top' Badge \*/}

                <div className="relative">

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                        alt="Product Thumbnail"

                        className="w-full h-48 object-cover rounded-md"

                    />

                    <span className="absolute top-2 left-2 bg-red-500 text-white text-xs font-semibold px-2 py-1 rounded">

                        Top

                    </span>

                </div>

                {/\* Product Info \*/}

                <div className="space-y-2">

                    {/\* Description \*/}

                    <h2 className="text-sm text-gray-700 font-medium line-clamp-2">

                       {name}

                    </h2>

                    {/\* Price \*/}

                    <div className="flex items-center gap-2">

                        <span className="text-gray-500 line-through text-sm">₹{originalPrice}</span>

                        <span className="text-red-600 font-semibold text-base">₹{finalPrice}</span>

                    </div>

                    {/\* Color Options \*/}

                    <div className="flex items-center gap-2">

                        {["bg-black", "bg-red-500", "bg-blue-500", "bg-green-600"].map((color, index) => (

                            <span

                                key={index}

                                className={`w-4 h-4 rounded-full border border-gray-300 ${color} cursor-pointer`}

                            />

                        ))}

                    </div>

                </div>

                {/\* Add to Cart Button \*/}

                <button className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                    Add to Cart

                </button>

            </div>

        </div>

    )

}

**Yanah hamane product ko successfully storepage me get karwakar display karwa diya hai**

**Commit**

**Ab CategoryFilter me category, brands and colors ko get kar storepage pa dikhani hai**

**Update storepage/layout.js**

import CategoryFilter from "@/components/website-components/CategoryFilter";

import BrandFilter from "@/components/website-components/BrandFilter";

import ColorFilter from "@/components/website-components/ColorFilter";

import { getBrands, getCategories, getColors } from "@/library/api-calls";

export default async function RootLayout({ children }) {

    {/\* 🧭 get categories \*/ }

    const categoryJSON = await getCategories(null)

    const categories = categoryJSON.data

    {/\* 🧭 get brands \*/ }

    const brandsJSON = await getBrands(null)

    const brands = brandsJSON.data

    {/\* 🧭 get colors \*/ }

    const colorJSON = await getColors(null)

    const colors = colorJSON.data

    return (

        <html lang="en">

            <body>

                <div className="space-y-10 p-5 bg-gray-50">

                    {/\* 🧭 Breadcrumb Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <ol className="flex items-center whitespace-nowrap space-x-2">

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Home

                                </a>

                                <svg

                                    className="mx-2 w-4 h-4 text-gray-400"

                                    viewBox="0 0 16 16"

                                    fill="none"

                                    xmlns="http://www.w3.org/2000/svg"

                                >

                                    <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                </svg>

                            </li>

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Shop

                                    <svg

                                        className="mx-2 w-4 h-4 text-gray-400"

                                        viewBox="0 0 16 16"

                                        fill="none"

                                        xmlns="http://www.w3.org/2000/svg"

                                    >

                                        <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                    </svg>

                                </a>

                            </li>

                            <li className="text-sm font-semibold text-gray-800 truncate">

                                Top Cell Phones & Tablets

                            </li>

                        </ol>

                    </section>

                    {/\* 📱 Top Cell Phones & Tablets Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Top Cell Phones & Tablets</h1>

                        <div className="flex flex-wrap gap-5">

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/slider3.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-3 text-white">

                                    <h1 className="text-2xl font-bold">Noise Cancelling</h1>

                                    <h3 className="text-lg">Headphone</h3>

                                    <p className="text-sm leading-tight">Boso Over-Ear Headphone <br />Wifi, Voice Assistant,<br />Low latency game mode</p>

                                    <button className="px-4 py-2 bg-white text-black rounded-lg font-semibold">BUY NOW</button>

                                </div>

                            </div>

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/banner1.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-2 text-black">

                                    <h1 className="text-2xl font-bold">Redmi Note 12 <br />Pro+ 5G</h1>

                                    <p className="text-sm text-gray-600">Rise to the challenge</p>

                                    <button className="mt-2 px-4 py-2 bg-black text-white rounded-lg">SHOP NOW</button>

                                </div>

                            </div>

                        </div>

                    </section>

                    {/\* 🗂️ Popular Categories Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Popular Categories</h1>

                        <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-5 gap-6">

                            {[

                                { title: "iPhone (iOS)", items: 74, image: "iphone" },

                                { title: "Android", items: 34, image: "android" },

                                { title: "5G Support", items: 12, image: "5gsupport" },

                                { title: "Apple Tablets", items: 22, image: "apptablet" },

                                { title: "Smartphone Chargers", items: 33, image: "smartphone\_chargers" },

                                { title: "Gaming", items: 9, image: "gaming" },

                                { title: "Xiaomi", items: 52, image: "xiaomi" },

                                { title: "Accessories", items: 29, image: "accessories" },

                                { title: "Samsung Tablets", items: 26, image: "samsung" },

                                { title: "eReader", items: 5, image: "eReader" }

                            ].map((cat, i) => (

                                <div key={i} className="flex justify-between items-center p-3 border rounded-md bg-gray-50">

                                    <div>

                                        <h1 className="font-semibold">{cat.title}</h1>

                                        <p className="text-sm text-gray-500">{cat.items} Items</p>

                                    </div>

                                    <img src={`store/popular\_categories/${cat.image}.png`} alt={cat.title} className="w-16 h-16 object-contain" />

                                </div>

                            ))}

                        </div>

                    </section>

                    {/\* 🧩 Content Area (children) \*/}

                    <section className="border rounded-md p-6 bg-white shadow">

                        <h1 className="text-xl font-bold mb-4">Main Content Area</h1>

                        <div className="flex w-full gap-6">

                            {/\* Left Sidebar: Filters \*/}

                            <div className="w-1/4 space-y-6">

                                {/\* Categories Filter \*/}

                                <CategoryFilter categories={categories} />

                                {/\* Brand Filter \*/}

                                <BrandFilter brands={brands} />

                                {/\* Color Filter \*/}

                                <ColorFilter  colors={colors}/>

                            </div>

                            {/\* Right Main Content \*/}

                            <div className="w-3/4 bg-red-400 p-4 rounded-md">

                                {children}

                            </div>

                        </div>

                    </section>

                </div>

            </body>

        </html>

    );

}

**Update CategoryFilter.jsx**

**Categories ko props lo storepage/page.jsx and display cateogry**

import React from 'react'

export default function CategoryFilter({ categories }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">CATEGORIES</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Categories

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            categories && categories.map(category => (

                                <li key={category.id}

                                 className="cursor-pointer hover:text-black">{category.name}</li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Update BrandsFilter.jsx**

**Brands ko props lo storepage/page.jsx and display brands**

import React from 'react'

export default function BrandFilter({ brands }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">BRAND FILTER</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Brands

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            brands && brands.map(brand => (

                                <li  key={brand.\_id} className="cursor-pointer hover:text-black">{brand.name}</li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Update ColorsFilter.jsx**

**Colors ko props lo storepage/page.jsx and display colors**

import React from 'react'

export default function ColorFilter({ colors }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">COLOR FILTER</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Colors

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Available Colors</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            colors && colors.map((color) => (

                                <li key={color.\_id}

                                    style={{ background: color.hexcode }}

                                    className="cursor-pointer hover:text-black w-5 h-5 rounded-4xl" ></li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Yanaha tak hamane category, brand and color filter ko complete kar liya hai next kis category ke kitane item hai show karwna hai**

**Commit**

**Category me kitane item hai**

**Update category.controller.js**

const categoryModel = require('../model/category.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const productModel = require('../model/product.model')

const fs = require('fs')

const category = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            const categoryImg = req.files.image

            const { name, slug } = req.body

            if (!name || !slug) {

                return noContentResponse(res)

            }

            const existingItem = await categoryModel.findOne({

                name

            })

            if (existingItem) {

                return serverErrorResponse(res, "Category already created", 409)

            }

            const image = createUniqueName(categoryImg.name)

            const destination = 'public/images/category/' + image

            categoryImg.mv(

                destination,

                async (error) => {

                    if (error) {

                        return errorResponse(res, "File not found")

                    } else {

                        const category = await categoryModel.create({

                            name,

                            slug,

                            image

                        })

                        await category.save()

                        return createdResponse(res, "category created successfully")

                    }

                }

            )

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            const id = req.params.id

            let category = null

            if (id) {

                category = await categoryModel.findById(id)

            } else {

                category = await categoryModel.find()

                const data = await Promise.all(

                    category.map(

                        async (cat) => {

                            const productCount = await productModel.countDocuments({

                                categoryId:

                                    cat.\_id

                            });

                            return {

                                ...cat.toObject(),

                                productCount

                            }

                        }

                    )

                )

                return successResponse(res, "Category Found", data)

            }

            if (!category) errorResponse(res, "Category not found")

            return successResponse(res, "Category Found", category)

        } catch (error) {

            return serverErrorResponse(res, error.msg)

        }

    },

    async deleteById(req, res) {

        try {

            const id = req.params.id;

            console.log(id);

            const existingCat = await categoryModel.findById(id);

            if (existingCat) {

                fs.unlinkSync(`public/images/category/${existingCat.image}`)

            }

            await categoryModel.findByIdAndDelete(id)

            return deletedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const id = req.params.id;

            const category = await categoryModel.findById(id);

            await categoryModel.findByIdAndUpdate(

                id,

                { $set: { status: !category.status } }

            )

            return updatedResponse(res, "Status Update")

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async update(req, res) {

        try {

            const id = req.params.id

            const categoryImg = req.files.image

            console.log(categoryImg, "categoryImg")

            console.log(req.body);

            const { name, slug } = req.body

            const existingItem = await categoryModel.findById(id)

            console.log(existingItem);

            if (!existingItem) {

                return serverErrorResponse(res, "Category not found", 409)

            }

            const update = {}

            if (name) update.name = name

            if (slug) update.slug = slug

            if (categoryImg) {

                const image = createUniqueName(categoryImg.name)

                const destination = 'public/images/category/' + image

                categoryImg.mv(

                    destination,

                    async (error) => {

                        if (error) {

                            return errorResponse(res, "File not found")

                        } else {

                            fs.unlinkSync(`public/images/category/${existingItem.image}`)

                            console.log(update);

                            update.image = image

                            await categoryModel.findByIdAndUpdate(

                                id,

                                { $set: update }

                            )

                            return updatedResponse(res, "category updated successfully")

                        }

                    }

                )

            }

            // const category = await categoryModel.create({

            //     name,

            //     slug

            // })

            // await category.save()

            // return createdResponse(res, "category created successfully")

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

}

module.exports = category

import React from 'react'

export default function CategoryFilter({ categories }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">CATEGORIES</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Categories

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            categories && categories.map(category => (

                                <li key={category.id}

                                 className="cursor-pointer hover:text-black">{category.name}

                                 <span>({category.productCount})</span>

                                 </li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Yanha tak category ka ka count display karwa diya hai**

**Commit**

**Ise hi hum brand and ka count display karwa sakte hais**

**04-09-25 day** **14**

**Category ke base par product filter karna**

**Update CategoryFilter.jsx**

import Link from 'next/link'

import React from 'react'

export default function CategoryFilter({ categories }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">CATEGORIES</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Categories

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            categories && categories.map(category => (

                                <Link href={`/store/${category.slug}`} key={category.\_id} >

                                    <li

                                        className="cursor-pointer hover:text-black">{category.name}

                                        <span>({category.productCount})</span>

                                    </li>

                                </Link>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Creae a foler in store**

**Storepage/[category\_slug]/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({params}) {

    const productJSON = await getProducts(null, params.category\_slug)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Update api-calls.js**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getProducts = async (id = null, categorySlug = null) => {

    try {

        let API = "product"

        if(id != null) API += `/${id}`

        const query = new URLSearchParams()

        if(categorySlug) query.append("categorySlug", categorySlug)

        const response = await axiosInstance.get(`${API}?${query.toString()}`)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getBrands = async (id = null) => {

    try {

        let API = "brand"

        if(id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if(id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors, getBrands, getProducts }

**update product.controller.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const categoryModel = require('../model/category.model')

const { log } = require('console')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            // console.log(req.query);

            const {categorySlug} = req.query

            const id = req.params.id

            const filterQuery = {}

            if(categorySlug) {

                const category = await categoryModel.findOne({slug : categorySlug})

                if(category) {

                    filterQuery.categoryId = category.\_id

                }

                // console.log(category);

            }

   // console.log(filterQuery);

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find(filterQuery)

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

**yanah tak hamane category filter se product ko filter karwa liya hai ab product category ke base par filter ho rahe hai**

**commit**

**All product lane ke liye**

**Update CategoryFilter**

import Link from 'next/link'

import React from 'react'

export default function CategoryFilter({ categories }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">CATEGORIES</h2>

                <Link href="/storepage">

                    <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                        All Categories

                    </button>

                </Link>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3   text-gray-500">

                        {

                            categories && categories.map(category => (

                                <Link href={`/storepage/${category.slug}`} key={category.\_id} >

                                    <li

                                        className="cursor-pointer hover:text-black">{category.name}

                                        <span>({category.productCount})</span>

                                    </li>

                                </Link>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Yanha tak All product par click karne par sabhi product aa rahe hai**

**Commit**

**Ab brand par filter karwayenge**

**Update Brand Filter.jsx**

'use client'

import React, { useEffect, useState } from 'react'

import { useRouter } from 'next/navigation'

export default function BrandFilter({ brands }) {

    const router = useRouter()

    const [selbrand, setSalbrand] = useState(null)

    useEffect(

        () => {

            const brand = new URLSearchParams({ brand: selbrand })

            if (selbrand) {

                router.push(`/storepage?${brand.toString()}`)

            }

        },

        [selbrand]

    )

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">BRAND FILTER</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Brands

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            brands && brands.map(brand => (

                                <li

                                    onClick={()=> setSalbrand(brand.slug)}

                                key={brand.\_id} className="cursor-pointer hover:text-black">{brand.name}</li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Now update storepage/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({searchParams}) {

    const brand = searchParams.brand ?? null

    const productJSON = await getProducts(null, brand)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Update api-calls**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if (id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getProducts = async (id = null, categorySlug = null, brandSlug = null) => {

    try {

        let API = "product"

        if (id != null) API += `/${id}`

        const query = new URLSearchParams()

        if (categorySlug) query.append("categorySlug", categorySlug)

        if (brandSlug) query.append("brandSlug", brandSlug)

        const response = await axiosInstance.get(`${API}?${query.toString()}`)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getBrands = async (id = null) => {

    try {

        let API = "brand"

        if (id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if (id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors, getBrands, getProducts }

**now update product.controler.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const categoryModel = require('../model/category.model')

const { log } = require('console')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            // console.log(req.query);

            const {categorySlug, brandSlug} = req.query

            console.log(categorySlug, brandSlug);

            const id = req.params.id

            const filterQuery = {}

            if(categorySlug) {

                const category = await categoryModel.findOne({slug : categorySlug})

                if(category) {

                    filterQuery.categoryId = category.\_id

                }

                // console.log(category);

            }

            // console.log(filterQuery);

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find(filterQuery)

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

**now update storepage/[category\_slug]/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({params, searchParams}) {

    const productJSON = await getProducts(null, params.category\_slug, searchParams.brand ?? null)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Now update product.controller.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const categoryModel = require('../model/category.model')

const brandModel = require('../model/brand.model')

const { log } = require('console')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            // console.log(req.query);

            const {categorySlug, brandSlug} = req.query

            console.log(categorySlug, brandSlug);

            const id = req.params.id

            const filterQuery = {}

            if(categorySlug) {

                const category = await categoryModel.findOne({slug : categorySlug})

                if(category) {

                    filterQuery.categoryId = category.\_id

                }

                // console.log(category);

            }

            // console.log(filterQuery);

              if(brandSlug) {

                const brand = await brandModel.findOne({slug : brandSlug})

                if(brand) {

                    filterQuery.brandId = brand.\_id

                }

                // console.log(category);

            }

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find(filterQuery)

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

**now update storepage/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({searchParams}) {

    const brand = searchParams.brand ?? null

    const productJSON = await getProducts(null, null,  brand)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Yanha tak brand ka filter kaam kar raha hai lekin all brand wala uncomplete hai aur humko yah functionality bhi add karni hai future me ki category par bhi filter lage aur brand par bhi dono me ek saath lagana chahiey**

**Commit**

**Ab color filter ke base par filter karenge**

**Update color Filter.jsx**

'use client'

import { useRouter } from 'next/navigation'

import React, { useEffect, useState } from 'react'

export default function ColorFilter({ colors }) {

    const router = useRouter()

    const [selColor, setSelColor] = useState(null)

    useEffect(

        () => {

            const color = new URLSearchParams({ color: selColor })

            if (selColor) {

                router.push(`/storepage?${color.toString()}`)

            }

        },

        [selColor]

    )

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">COLOR FILTER</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Colors

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Available Colors</p>

                    <ul className="space-y-1  flex gap-3 pl-3 text-gray-500">

                        {

                            colors && colors.map((color) => (

                                <li

                                    onClick={() => setSelColor(color.slug)}

                                    key={color.\_id}

                                    style={{ background: color.hexcode }}

                                    className="cursor-pointer hover:text-black w-5 h-5 rounded-4xl" ></li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Now update storepage/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({searchParams}) {

    const brand = searchParams.brand ?? null

    const color = searchParams.color ?? null

    const productJSON = await getProducts(null, null,  brand, color)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Now update api-calls.js**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if (id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getProducts = async (id = null, categorySlug = null, brandSlug = null, colorSlug = null) => {

    try {

        let API = "product"

        if (id != null) API += `/${id}`

        const query = new URLSearchParams()

        if (categorySlug) query.append("categorySlug", categorySlug)

        if (brandSlug) query.append("brandSlug", brandSlug)

        if (colorSlug) query.append("colorSlug", colorSlug)

        const response = await axiosInstance.get(`${API}?${query.toString()}`)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getBrands = async (id = null) => {

    try {

        let API = "brand"

        if (id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if (id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors, getBrands, getProducts }

**now update product.controller.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const categoryModel = require('../model/category.model')

const brandModel = require('../model/brand.model')

const colorModel = require('../model/color.model')

const { log } = require('console')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            // console.log(req.query);

            const {categorySlug, brandSlug, colorSlug} = req.query

            console.log(categorySlug, brandSlug, colorSlug);

            const id = req.params.id

            const filterQuery = {}

            if(categorySlug) {

                const category = await categoryModel.findOne({slug : categorySlug})

                if(category) {

                    filterQuery.categoryId = category.\_id

                }

                // console.log(category);

            }

            // console.log(filterQuery);

              if(brandSlug) {

                const brand = await brandModel.findOne({slug : brandSlug})

                if(brand) {

                    filterQuery.brandId = brand.\_id

                }

                // console.log(category);

            }

               if(colorSlug) {

                const color = await colorModel.findOne({slug : colorSlug})

                if(color) {

                    filterQuery.colors = color.\_id

                    //colors == colorId kyoki database me colors ke name se save hai id

                }

                // console.log(category);

            }

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find(filterQuery)

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

**yanha hama hamara color filter bhi filter ho raha hai**

**commit**

**08-09-25 day** **15**

**Update productCard.jsx**

**kyonki color url me to filter ho rahe hai lekin card par jo color hai vo menual hai isliye product me se color nikalkar color wala array print karwana hai product card me**

import React from 'react'

export default function ProductCard({ product }) {

    const { name, finalPrice, discountPercentage, originalPrice, thumbnail, colors } = product;

    console.log(colors);

    return (

        <div>

            <div className="w-full max-w-xs bg-white border rounded-xl shadow hover:shadow-lg transition p-4 space-y-4">

                {/\* Product Image with 'Top' Badge \*/}

                <div className="relative">

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                        alt="Product Thumbnail"

                        className="w-full h-48 object-cover rounded-md"

                    />

                    <span className="absolute top-2 left-2 bg-red-500 text-white text-xs font-semibold px-2 py-1 rounded">

                        Top

                    </span>

                </div>

                {/\* Product Info \*/}

                <div className="space-y-2">

                    {/\* Description \*/}

                    <h2 className="text-sm text-gray-700 font-medium line-clamp-2">

                        {name}

                    </h2>

                    {/\* Price \*/}

                    <div className="flex items-center gap-2">

                        <span className="text-gray-500 line-through text-sm">₹{originalPrice}</span>

                        <span className="text-red-600 font-semibold text-base">₹{finalPrice}</span>

                    </div>

                    {/\* Color Options \*/}

                    <div className="flex items-center gap-2">

                        {

                            colors.map((color, index) => (

                                <span

                                    key={index}

                                    style={{ background: color.hexcode }}

                                    className={`w-4 h-4 rounded-full border border-gray-300 ${color} cursor-pointer`}

                                />

                            ))

                        }

                    </div>

                </div>

                {/\* Add to Cart Button \*/}

                <button className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                    Add to Cart

                </button>

            </div>

        </div>

    )

}

**update brandFilter and ColorFilter category ke base par filter ke liye**

**Update brandFilter.jsx**

'use client'

import React, { useEffect, useState } from 'react'

import { useRouter } from 'next/navigation'

export default function BrandFilter({ brands }) {

    const router = useRouter()

    const [selbrand, setSalbrand] = useState(null)

    useEffect(

        () => {

            const brand = new URLSearchParams({ brand: selbrand })

            if (selbrand) {

                router.push(`?${brand.toString()}`)

            }

        },

        [selbrand]

    )

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">BRAND FILTER</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Brands

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Cell Phones & Tablets</p>

                    <ul className="space-y-1 pl-3 text-gray-500">

                        {

                            brands && brands.map(brand => (

                                <li

                                    onClick={()=> setSalbrand(brand.slug)}

                                key={brand.\_id} className="cursor-pointer hover:text-black">{brand.name}</li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Update colorFilter.jsx**

'use client'

import { useRouter } from 'next/navigation'

import React, { useEffect, useState } from 'react'

export default function ColorFilter({ colors }) {

    const router = useRouter()

    const [selColor, setSelColor] = useState(null)

    useEffect(

        () => {

            const color = new URLSearchParams({ color: selColor })

            if (selColor) {

                router.push(`?${color.toString()}`)

            }

        },

        [selColor]

    )

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">COLOR FILTER</h2>

                <button className="w-full border px-3 py-2 rounded-md font-medium text-sm hover:bg-gray-100">

                    All Colors

                </button>

                <div className="text-sm space-y-1 pl-2">

                    <p className="text-gray-700">Available Colors</p>

                    <ul className="space-y-1  flex gap-3 pl-3 text-gray-500">

                        {

                            colors && colors.map((color) => (

                                <li

                                    onClick={() => setSelColor(color.slug)}

                                    key={color.\_id}

                                    style={{ background: color.hexcode }}

                                    className="cursor-pointer hover:text-black w-5 h-5 rounded-4xl" ></li>

                            ))

                        }

                    </ul>

                </div>

            </div>

        </div>

    )

}

**Update product.controller.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const categoryModel = require('../model/category.model')

const brandModel = require('../model/brand.model')

const colorModel = require('../model/color.model')

const { log } = require('console')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            // console.log(req.query);

            const {categorySlug, brandSlug, colorSlug} = req.query

            console.log(categorySlug, brandSlug, colorSlug, "all three");

            const id = req.params.id

            const filterQuery = {}

            if(categorySlug) {

                const category = await categoryModel.findOne({slug : categorySlug})

                if(category) {

                    filterQuery.categoryId = category.\_id

                }

                // console.log(category);

            }

            // console.log(filterQuery);

              if(brandSlug) {

                const brand = await brandModel.findOne({slug : brandSlug})

                if(brand) {

                    filterQuery.brandId = brand.\_id

                }

                // console.log(category);

            }

               if(colorSlug) {

                const color = await colorModel.findOne({slug : colorSlug})

                if(color) {

                    filterQuery.colors = color.\_id

                    //colors == colorId kyoki database me colors ke name se save hai id

                }

                // console.log(category);

            }

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find(filterQuery).populate("colors");

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

**yanah tak abhi bhi kisi ek category ke multiple brand hone par bhi brand filter ho raha hai aur individual bhi filter ho raha hai**

**lekin kisi ek particular category ke product par kisi particular color par product filter nahi ho rahe hai**

**commit**

**is problem ko fix karne ke liye**

**update storepage/[category\_slug]/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({params, searchParams}) {

    const productJSON = await getProducts(null, params.category\_slug, searchParams.brand ?? null, searchParams.color ?? null)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Yanhapar kisi particular category par click ko filter karne ke baad kisi particular color ko filter karne par vah color us category me filter ho raha hai**

**Commit**

**Ab ek task hame assign kiya gaya hai pagination ya limit product filter.**

**Ab hame price filter banana hai**

**Website-component/PriceFilter.jsx**

'use client'

import { useRouter } from 'next/navigation'

import React, { useEffect, useState } from 'react'

export default function PriceFilter({ colors }) {

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">PRICE FILTER</h2>

                <input type="range" name="" id="" />

            </div>

        </div>

    )

}

**Import in /storepage/layout.js**

import CategoryFilter from "@/components/website-components/CategoryFilter";

import BrandFilter from "@/components/website-components/BrandFilter";

import ColorFilter from "@/components/website-components/ColorFilter";

import PriceFilter from "@/components/website-components/PriceFilter";

import { getBrands, getCategories, getColors } from "@/library/api-calls";

export default async function RootLayout({ children }) {

    {/\* 🧭 get categories \*/ }

    const categoryJSON = await getCategories(null)

    const categories = categoryJSON.data

    {/\* 🧭 get brands \*/ }

    const brandsJSON = await getBrands(null)

    const brands = brandsJSON.data

    {/\* 🧭 get colors \*/ }

    const colorJSON = await getColors(null)

    const colors = colorJSON.data

    return (

        <html lang="en">

            <body>

                <div className="space-y-10 p-5 bg-gray-50">

                    {/\* 🧭 Breadcrumb Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <ol className="flex items-center whitespace-nowrap space-x-2">

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Home

                                </a>

                                <svg

                                    className="mx-2 w-4 h-4 text-gray-400"

                                    viewBox="0 0 16 16"

                                    fill="none"

                                    xmlns="http://www.w3.org/2000/svg"

                                >

                                    <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                </svg>

                            </li>

                            <li className="inline-flex items-center">

                                <a className="text-sm text-gray-500 hover:text-blue-600" href="#">

                                    Shop

                                    <svg

                                        className="mx-2 w-4 h-4 text-gray-400"

                                        viewBox="0 0 16 16"

                                        fill="none"

                                        xmlns="http://www.w3.org/2000/svg"

                                    >

                                        <path d="M6 13L10 3" stroke="currentColor" strokeLinecap="round" />

                                    </svg>

                                </a>

                            </li>

                            <li className="text-sm font-semibold text-gray-800 truncate">

                                Top Cell Phones & Tablets

                            </li>

                        </ol>

                    </section>

                    {/\* 📱 Top Cell Phones & Tablets Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Top Cell Phones & Tablets</h1>

                        <div className="flex flex-wrap gap-5">

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/slider3.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-3 text-white">

                                    <h1 className="text-2xl font-bold">Noise Cancelling</h1>

                                    <h3 className="text-lg">Headphone</h3>

                                    <p className="text-sm leading-tight">Boso Over-Ear Headphone <br />Wifi, Voice Assistant,<br />Low latency game mode</p>

                                    <button className="px-4 py-2 bg-white text-black rounded-lg font-semibold">BUY NOW</button>

                                </div>

                            </div>

                            <div className="relative w-full sm:w-[48%] h-80">

                                <img src="store/top cell phones & tablets/banner1.png" alt="" className="w-full h-full object-cover rounded-md" />

                                <div className="absolute top-10 left-10 space-y-2 text-black">

                                    <h1 className="text-2xl font-bold">Redmi Note 12 <br />Pro+ 5G</h1>

                                    <p className="text-sm text-gray-600">Rise to the challenge</p>

                                    <button className="mt-2 px-4 py-2 bg-black text-white rounded-lg">SHOP NOW</button>

                                </div>

                            </div>

                        </div>

                    </section>

                    {/\* 🗂️ Popular Categories Section \*/}

                    <section className="border rounded-md p-4 bg-white shadow">

                        <h1 className="text-xl font-bold uppercase mb-4">Popular Categories</h1>

                        <div className="grid grid-cols-2 sm:grid-cols-3 lg:grid-cols-5 gap-6">

                            {[

                                { title: "iPhone (iOS)", items: 74, image: "iphone" },

                                { title: "Android", items: 34, image: "android" },

                                { title: "5G Support", items: 12, image: "5gsupport" },

                                { title: "Apple Tablets", items: 22, image: "apptablet" },

                                { title: "Smartphone Chargers", items: 33, image: "smartphone\_chargers" },

                                { title: "Gaming", items: 9, image: "gaming" },

                                { title: "Xiaomi", items: 52, image: "xiaomi" },

                                { title: "Accessories", items: 29, image: "accessories" },

                                { title: "Samsung Tablets", items: 26, image: "samsung" },

                                { title: "eReader", items: 5, image: "eReader" }

                            ].map((cat, i) => (

                                <div key={i} className="flex justify-between items-center p-3 border rounded-md bg-gray-50">

                                    <div>

                                        <h1 className="font-semibold">{cat.title}</h1>

                                        <p className="text-sm text-gray-500">{cat.items} Items</p>

                                    </div>

                                    <img src={`store/popular\_categories/${cat.image}.png`} alt={cat.title} className="w-16 h-16 object-contain" />

                                </div>

                            ))}

                        </div>

                    </section>

                    {/\* 🧩 Content Area (children) \*/}

                    <section className="border rounded-md p-6 bg-white shadow">

                        <h1 className="text-xl font-bold mb-4">Main Content Area</h1>

                        <div className="flex w-full gap-6">

                            {/\* Left Sidebar: Filters \*/}

                            <div className="w-1/4 space-y-6">

                                {/\* Categories Filter \*/}

                                <CategoryFilter categories={categories} />

                                {/\* Brand Filter \*/}

                                <BrandFilter brands={brands} />

                                {/\* Color Filter \*/}

                                <ColorFilter  colors={colors}/>

                                {/\* Price Filter \*/}

                                <PriceFilter />

                            </div>

                            {/\* Right Main Content \*/}

                            <div className="w-3/4 bg-red-400 p-4 rounded-md">

                                {children}

                            </div>

                        </div>

                    </section>

                </div>

            </body>

        </html>

    );

}

**Both side se filter ek package ka use karte hai frontend me**

npm i react-range-slider-input

**update PriceFilter.jsx**

'use client'

import { useRouter, useSearchParams } from 'next/navigation'

import React, { useEffect, useState } from 'react'

import RangeSlider from 'react-range-slider-input';

import 'react-range-slider-input/dist/style.css';

export default function PriceFilter({ colors }) {

    const router = useRouter()

    const searchParams = useSearchParams()

    const [price, setPrice] = useState([])

    useEffect(

        ()=> {

            const min = searchParams.get('min')

            const max = searchParams.get('max')

            // console.log(min, max);

            setPrice([Number(min), Number(max)])

        },[searchParams]

    )

    function changeHandler(data) {

        // console.log(data);

        setPrice(data)

        const price = new URLSearchParams()

        price.set("min", data[0])

        price.set("max", data[1])

        router.push(`?${price.toString()}`)

    }

    return (

        <div>

            <div className="bg-white p-4 rounded-xl shadow border space-y-4">

                <h2 className="text-lg font-bold">PRICE FILTER</h2>

                <div className='my-6'>

                    <label htmlFor="">{price[0]}</label>

                    <span>-</span>

                    <label htmlFor="">{price[1]}</label>

                </div>

                <RangeSlider min="200" max="20000" value={price} onInput={changeHandler} />

            </div>

        </div>

    )

}

**Default value ke liye usestate me initial value dal do**

const [price, setPrice] = useState([])

**now update /storepage/page.jsx**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({searchParams}) {

    const brand = searchParams.brand ?? null

    const color = searchParams.color ?? null

    const min = searchParams.min ?? null

    const max = searchParams.max ?? null

    const productJSON = await getProducts(null, null,  brand, color, min, max)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Now update api-calls.js**

import { axiosInstance } from "./helper";

const getCategories = async (id = null) => {

    try {

        let API = "category"

        if (id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getProducts = async (id = null, categorySlug = null, brandSlug = null, colorSlug = null, min = null, max = null) => {

    try {

        let API = "product"

        if (id != null) API += `/${id}`

        const query = new URLSearchParams()

        if (categorySlug) query.append("categorySlug", categorySlug)

        if (brandSlug) query.append("brandSlug", brandSlug)

        if (colorSlug) query.append("colorSlug", colorSlug)

        if (min) query.append("min", min)

        if (max) query.append("max", max)

        const response = await axiosInstance.get(`${API}?${query.toString()}`)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getBrands = async (id = null) => {

    try {

        let API = "brand"

        if (id != null) API += `/${id}`

        // console.log(API);

        // http://localhost:5000/category/ //yah to chal rahi broser me

        // http://localhost:5000/category/68b00eca8029bc625c33343f   //yah nahi chal rahi hai

        const response = await axiosInstance.get(API)

        // console.log(response.data, "resp");

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

const getColors = async (id = null) => {

    try {

        let API = "color"

        if (id != null) API += `/${id}`

        const response = await axiosInstance.get(API)

        if (response.data.success) {

            return response.data

        } else {

            return null

        }

    }

    catch (error) {

        console.log(error)

        return null

    }

}

export { getCategories, getColors, getBrands, getProducts }

**now update product.controller.js**

const productModel = require('../model/product.model')

const { noContentResponse, createdResponse, serverErrorResponse, errorResponse, successResponse, updatedResponse, deletedResponse } = require('../utility/response')

const { createUniqueName } = require('../utility/helper')

const fs = require('fs')

const categoryModel = require('../model/category.model')

const brandModel = require('../model/brand.model')

const colorModel = require('../model/color.model')

const { log } = require('console')

async function saveFile(imageObj) {

    const imageName = createUniqueName(imageObj.name)

    const destination = 'public/images/product/' + imageName

    await imageObj.mv(destination)

    return imageName

}

const product = {

    async create(req, res) {

        try {

            // console.log(req.body);

            // console.log(req.files);

            // Save thumbnail

            const thumbnail = req.files?.thumbnail

                ? await saveFile(req.files.thumbnail)

                : null

            // Save multiple images

            const images = req.files?.images

                ? await Promise.all(

                    (Array.isArray(req.files.images)

                        ? req.files.images

                        : [req.files.images]

                    ).map((img) => saveFile(img))

                )

                : []

            // Create product

            await productModel.create({

                ...req.body,

                colors: req.body.colors ? JSON.parse(req.body.colors) : [],

                thumbnail,

                images

            })

            return res.status(201).json({ success: true, message: "Product created successfully" })

        } catch (error) {

            console.log(error);

            return serverErrorResponse(res)

        }

    },

    async read(req, res) {

        try {

            // console.log(req.query);

            const {categorySlug, brandSlug, colorSlug, min, max} = req.query

            console.log(categorySlug, brandSlug, colorSlug, "all three");

            const id = req.params.id

            const filterQuery = {}

            if(categorySlug) {

                const category = await categoryModel.findOne({slug : categorySlug})

                if(category) {

                    filterQuery.categoryId = category.\_id

                }

                // console.log(category);

            }

            // console.log(filterQuery);

              if(brandSlug) {

                const brand = await brandModel.findOne({slug : brandSlug})

                if(brand) {

                    filterQuery.brandId = brand.\_id

                }

                // console.log(category);

            }

               if(colorSlug) {

                const color = await colorModel.findOne({slug : colorSlug})

                if(color) {

                    filterQuery.colors = color.\_id

                    //colors == colorId kyoki database me colors ke name se save hai id

                }

                // console.log(category);

            }

             if(min && max) {

                filterQuery.finalPrice={

                    $gte: min,

                    $lte: max

                }

            }

            let product = null;

            if (id) {

                product = await productModel.findById(id)

            } else {

                product = await productModel.find(filterQuery).populate("colors");

            }

            if (!product) errorResponse(res, "product not found")

            return successResponse(res, "Product Found", product)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    },

    async status(req, res) {

        try {

            const { flag } = req.body

            const id = req.params.id;

            const product = await productModel.findById(id);

            if (!product) return noContentResponse(res);

            const updateKey = {};

            if (flag == 1) {

                updateKey.status = !product.status

            } else if (flag == 2) {

                updateKey.stock = !product.stock

            } else if (flag == 3) {

                updateKey.topSelling = !product.topSelling

            }

            await productModel.findByIdAndUpdate(

                id,

                {

                    $set: updateKey

                }

            )

            return updatedResponse(res)

        } catch (error) {

            return serverErrorResponse(res, error.errmsg)

        }

    }

}

module.exports = product

**agar jyada range ke product hai to FilterPrice.jsx me jakar RangeSlider me min ya max ki value manage kar sakte hai**

                <RangeSlider min="200" max="20000" value={price} onInput={changeHandler} />

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({params, searchParams}) {

    const productJSON = await getProducts(null, params.category\_slug, searchParams.brand ?? null, searchParams.color ?? null, searchParams,min || null, searchParams.max || null)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**Yanha hamara price range task complete hota hai**

**Commit**

**Ab hum react silk slider lagayenge home page par**

npm install react-slick –save

**isem frontend me add karo**

**create a component in**

**website-component/ImageSlider.jsx**

'use client'

import React from "react";

import Slider from "react-slick";

function ImageSlider() {

  var settings = {

    dots: true,

    infinite: false,

    speed: 500,

    slidesToShow: 4,

    slidesToScroll: 4,

    initialSlide: 0,

    responsive: [

      {

        breakpoint: 1024,

        settings: {

          slidesToShow: 3,

          slidesToScroll: 3,

          infinite: true,

          dots: true

        }

      },

      {

        breakpoint: 600,

        settings: {

          slidesToShow: 2,

          slidesToScroll: 2,

          initialSlide: 2

        }

      },

      {

        breakpoint: 480,

        settings: {

          slidesToShow: 1,

          slidesToScroll: 1

        }

      }

    ]

  };

  return (

    <div className="slider-container">

      <Slider {...settings}>

        <div>

          <h3>1</h3>

        </div>

        <div>

          <h3>2</h3>

        </div>

        <div>

          <h3>3</h3>

        </div>

        <div>

          <h3>4</h3>

        </div>

        <div>

          <h3>5</h3>

        </div>

        <div>

          <h3>6</h3>

        </div>

        <div>

          <h3>7</h3>

        </div>

        <div>

          <h3>8</h3>

        </div>

      </Slider>

    </div>

  );

}

export default ImageSlider;

**call in**

**(website-group)/page.jsx**

import ImageSlider from '@/components/website-components/ImageSlider'

import React from 'react'

export default function page() {

  return (

    <div>

      <ImageSlider />

    </div>

  )

}

**Slider me css add karan**

**Install package and import links in front end**

npm install slick-carousel –save

**now update (website-componet)/page.jsx**

import ImageSlider from '@/components/website-components/ImageSlider'

import React from 'react'

import "slick-carousel/slick/slick.css";

import "slick-carousel/slick/slick-theme.css";

export default function page() {

  return (

    <div>

      <ImageSlider />

    </div>

  )

}

**Yanah tak hamane ek basic slider lagakar dekha lekin us par completely apply nahi kiya hai kyonki abhi hamane kisi product yanaha call nahi karwaya ya image set nahi ki hai**

**08-09-25 ki video dekh sakte thoda aur explore karne ke liye from 44:00 time se 49:00**

**Commit**

**/storepage/[category\_slug]/page.jsx ko update kar lena kyonki searchParams,min tha aur uske pahale hi commit ho gaya isliye ise sahi kar lena**

import ProductCard from '@/components/website-components/ProductCard'

import { getProducts } from '@/library/api-calls'

import React from 'react'

export default async function page({params, searchParams}) {

    const productJSON = await getProducts(null, params.category\_slug, searchParams.brand ?? null, searchParams.color ?? null, searchParams.min || null, searchParams.max || null)

    const products = productJSON.data

    return (

        <div className='grid grid-cols-4 gap-4'>

            {

                products.map((product) => (

                    <ProductCard key={product.\_id}  product={product} />

                ))

            }

        </div>

    )

}

**09-09-25 day**

**Redux Introduction**

**10-09-25**

**Add to Cart**

**Frontend me install karo**

npm install @reduxjs/toolkit react-redux

create a file

src/redux/store.js

import { configureStore } from '@reduxjs/toolkit'

const store =  configureStore({

  reducer: {},

})

export default store

now create a file in website-component/StoreProvider.jsx

import React from 'react'

export default function StoreProvide({children}) {

  return (

    <div>

    </div>

  )

}

**Ab hum isko (website-group)/layout.jsx me hi apply karenge aur yah usi par apply karega admin par apply nahi karega**

import { Geist, Geist\_Mono } from "next/font/google";

import "./../globals.css";

import StoreProvide from "@/components/StoreProvider";

const geistSans = Geist({

  variable: "--font-geist-sans",

  subsets: ["latin"],

});

const geistMono = Geist\_Mono({

  variable: "--font-geist-mono",

  subsets: ["latin"],

});

export const metadata = {

  title: "Create Next App",

  description: "Generated by create next app",

};

export default function RootLayout({ children }) {

  return (

    <html lang="en">

      <body

        className={`${geistSans.variable} ${geistMono.variable} antialiased`}

      >

        <StoreProvide>

          {/\* agar header and footer honge to vo bhi StoreProvider ke andar hi honge \*/}

          {children}

        </StoreProvide>

      </body>

    </html>

  );

}

**Now update StoreProvider.jsx**

import React from 'react'

export default function StoreProvider({children}) {

  return (

    <div>

      {children}

    </div>

  )

}

**Now again update StoreProvider.jsx**

import React from 'react'

import store from '@/redux/store'

import { Provider } from 'react-redux'

export default function StoreProvider({ children }) {

  return (

    <div>

      <Provider store={store}>

        {children}

      </Provider>

    </div>

  )

}

**Create a file**

**Src/redux/features/cartSlice.js**

import { createSlice } from '@reduxjs/toolkit'

export const cartSlice = createSlice({

    name: 'cart',

    initialState: {

        items: [],

        final\_total: 0,

        original\_total: 0

    },

    reducers: {

        addtoCart: (state) => {

            console.log(state);

        },

    },

})

// Action creators are generated for each case reducer function

export const { addtoCart } = cartSlice.actions

export default cartSlice.reducer

**now update redux/store.js**

import { configureStore } from '@reduxjs/toolkit'

import { cartSlice } from './features/cartSlice'

const store =  configureStore({

  reducer: {

    cart: cartSlice

  },

})

export default store

**update StoreProvide.jsx**

**make the StoreProvider.jsx ‘use client’**

'use client'

import React from 'react'

import store from '@/redux/store'

import { Provider } from 'react-redux'

export default function StoreProvider({ children }) {

  return (

    <div>

      <Provider store={store}>

        {children}

      </Provider>

    </div>

  )

}

**Creaet a component in**

**Website-component/CartBtn.jsx**

import React from 'react'

export default function CartBtn() {

    return (

        <div>

            <button className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                Add to Cart

            </button>

        </div>

    )

}

**Call CartBtn.jsx in ProductCart.jsx**

import React from 'react'

import CartBtn from './CartBtn';

export default function ProductCard({ product }) {

    const { name, finalPrice, discountPercentage, originalPrice, thumbnail, colors } = product;

    // console.log(colors, "products color");

    return (

        <div>

            <div className="w-full max-w-xs bg-white border rounded-xl shadow hover:shadow-lg transition p-4 space-y-4">

                {/\* Product Image with 'Top' Badge \*/}

                <div className="relative">

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                        alt="Product Thumbnail"

                        className="w-full h-48 object-cover rounded-md"

                    />

                    <span className="absolute top-2 left-2 bg-red-500 text-white text-xs font-semibold px-2 py-1 rounded">

                        Top

                    </span>

                </div>

                {/\* Product Info \*/}

                <div className="space-y-2">

                    {/\* Description \*/}

                    <h2 className="text-sm text-gray-700 font-medium line-clamp-2">

                        {name}

                    </h2>

                    {/\* Price \*/}

                    <div className="flex items-center gap-2">

                        <span className="text-gray-500 line-through text-sm">₹{originalPrice}</span>

                        <span className="text-red-600 font-semibold text-base">₹{finalPrice}</span>

                    </div>

                    {/\* Color Options \*/}

                    <div className="flex items-center gap-2">

                        {

                            colors?.map((color) => {

                                return <span

                                key={color.\_id}

                                 style={{ background: color.hexcode }} className="w-5 h-5 rounded-full cursor-pointer border"></span>

                            })

                        }

                    </div>

                </div>

                {/\* Add to Cart Button \*/}

                {/\* <button className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                    Add to Cart

                </button> \*/}

                <CartBtn />

            </div>

        </div>

    )

}

**Now update ProductCart.jsx**

import React from 'react'

import CartBtn from './CartBtn';

export default function ProductCard({ product }) {

    const { name, finalPrice, discountPercentage, originalPrice, thumbnail, colors, \_id } = product;

    // console.log(colors, "products color");

    return (

        <div>

            <div className="w-full max-w-xs bg-white border rounded-xl shadow hover:shadow-lg transition p-4 space-y-4">

                {/\* Product Image with 'Top' Badge \*/}

                <div className="relative">

                    <img

                        src={`${process.env.NEXT\_PUBLIC\_API\_BASE\_URL}images/product/${product.thumbnail}`}

                        alt="Product Thumbnail"

                        className="w-full h-48 object-cover rounded-md"

                    />

                    <span className="absolute top-2 left-2 bg-red-500 text-white text-xs font-semibold px-2 py-1 rounded">

                        Top

                    </span>

                </div>

                {/\* Product Info \*/}

                <div className="space-y-2">

                    {/\* Description \*/}

                    <h2 className="text-sm text-gray-700 font-medium line-clamp-2">

                        {name}

                    </h2>

                    {/\* Price \*/}

                    <div className="flex items-center gap-2">

                        <span className="text-gray-500 line-through text-sm">₹{originalPrice}</span>

                        <span className="text-red-600 font-semibold text-base">₹{finalPrice}</span>

                    </div>

                    {/\* Color Options \*/}

                    <div className="flex items-center gap-2">

                        {

                            colors?.map((color) => {

                                return <span

                                key={color.\_id}

                                 style={{ background: color.hexcode }} className="w-5 h-5 rounded-full cursor-pointer border"></span>

                            })

                        }

                    </div>

                </div>

                {/\* Add to Cart Button \*/}

                {/\* <button className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                    Add to Cart

                </button> \*/}

                <CartBtn finalPrice={finalPrice} productId={\_id} originalPrice={originalPrice}/>

            </div>

        </div>

    )

}

**Now update CartBtn.jsx**

import { addtoCart } from '@/redux/features/cartSlice'

import React from 'react'

import { useDispatch } from 'react-redux'

export default function CartBtn(originalPrice, finalPrice, productId) {

    //Payload

    const dispacher=useDispatch()

    return (

        <div>

            <button

            onClick={()=> {

                dispacher(

                    addtoCart({

                        productId,

                        originalPrice,

                        finalPrice

                    })

                )

            }}

            className="w-full py-2 bg-blue-600 text-white text-sm font-semibold rounded-md hover:bg-blue-700 transition">

                Add to Cart

            </button>

        </div>

    )

}

**Yanha me is database localmongoDB se atles MongoDB par le ja raha hun yah coaching ka part nahi hai yah me khud se kar raha hun**

**Backend ke .env file me update karna**

**PORT=5000**

**DATABASE\_URL=mongodb+srv://shivamAtlas:xYFTJ5QXvXMFWgiO@cluster0.sz7pqxo.mongodb.net/sishop?retryWrites=true&w=majority&appName=Cluster0**

**TOKEN\_SECRET\_KEY=WSCUBETECH@99**

**Backend ke index.js me update karna**

**require('dotenv').config()**

**const express = require('express')**

**const mongoose = require('mongoose')**

**const cors = require('cors')**

**var cookieParser = require('cookie-parser')**

**const categoryRouter = require('./router/category.router')**

**const colorRouter = require('./router/color.router')**

**const brandRouter = require('./router/brand.router')**

**const productRouter = require('./router/product.router')**

**const adminRouter = require('./router/admin.router')**

**const server = express();**

**server.use(cors({origin: "http://localhost:3000", credentials : true}))**

**server.use(cookieParser())**

**server.use(express.json())**

**server.use('/category', categoryRouter)**

**server.use('/color', colorRouter)**

**server.use('/brand', brandRouter)**

**server.use('/product', productRouter)**

**server.use('/admin', adminRouter)**

**server.use(express.static('public'));**

**server.listen(**

**process.env.PORT,**

**()=> {**

**console.log('Server Running PORT 5000');**

**mongoose.connect(process.env.DATABASE\_URL, {dbName: "sishop"}).then(**

**()=> {**

**console.log("Database Connected");**

**}**

**).catch(()=> {**

**console.log('unable to connect database');**

**})**

**}**

**)**

**Frontend ki .env code preserve hona chahiye**

**NEXT\_PUBLIC\_API\_BASE\_URL=http://localhost:5000/**

**Frontend ke helper.js me kuch update**

**import { ToastContainer, toast } from "react-toastify";**

**const notify = (msg, flag) => toast(msg, {type: flag ? 'success' : 'error'})**

**import axios from "axios";**

**const axiosInstance = axios.create({**

**baseURL: process.env.NEXT\_PUBLIC\_API\_BASE\_URL,  withCredentials: true**

**})**

**s**

**function createSlug(text) {**

**return text**

**.toString()              // Ensure it's a string**

**.trim()                  // Remove extra spaces**

**.toLowerCase()           // Convert to lowercase**

**.replace(/[^a-z0-9\s-]/g, '')  // Remove special chars**

**.replace(/\s+/g, '-')          // Replace spaces with dashes**

**.replace(/-+/g, '-');          // Replace multiple dashes with single**

**}**

**function getCookies(name) {**

**if (typeof document === 'undefined') return null;**

**const value = `; ${document.cookie}`;**

**const parts = value.split(`; ${name}=`);**

**if (parts.length === 2) return parts.pop().split(';').shift();**

**return null;**

**}**

**export { createSlug, notify, axiosInstance, getCookies }**

**commit**

**yanha tak atles me color add karne par color add ho raha hai yani backend kaam kar raha hai**