Cloudinary

<https://www.npmjs.com/package/express-fileupload>

* Install express-fileupload (When you upload a file, the file will be accessible from req.files) and open the app.js file in backend

const fileUpload = require("express-fileupload")

app.use(fileUpload())

<https://www.npmjs.com/package/cloudinary>

* Install Cloudinary Node SDK (The Cloudinary Node SDK allows you to quickly and easily integrate your application with Cloudinary. Effortlessly optimize, transform, upload and manage your cloud's assets.)
* After connecting Database in backend use **cloudinary and you will find** CLOUDINARY\_NAME, CLOUDINARY\_API\_KEY, CLOUDINARY\_API\_SECRET after login to cloudinary website

const cloudinary = require("cloudinary")

connectTodataBase();

//cloudinary attachment

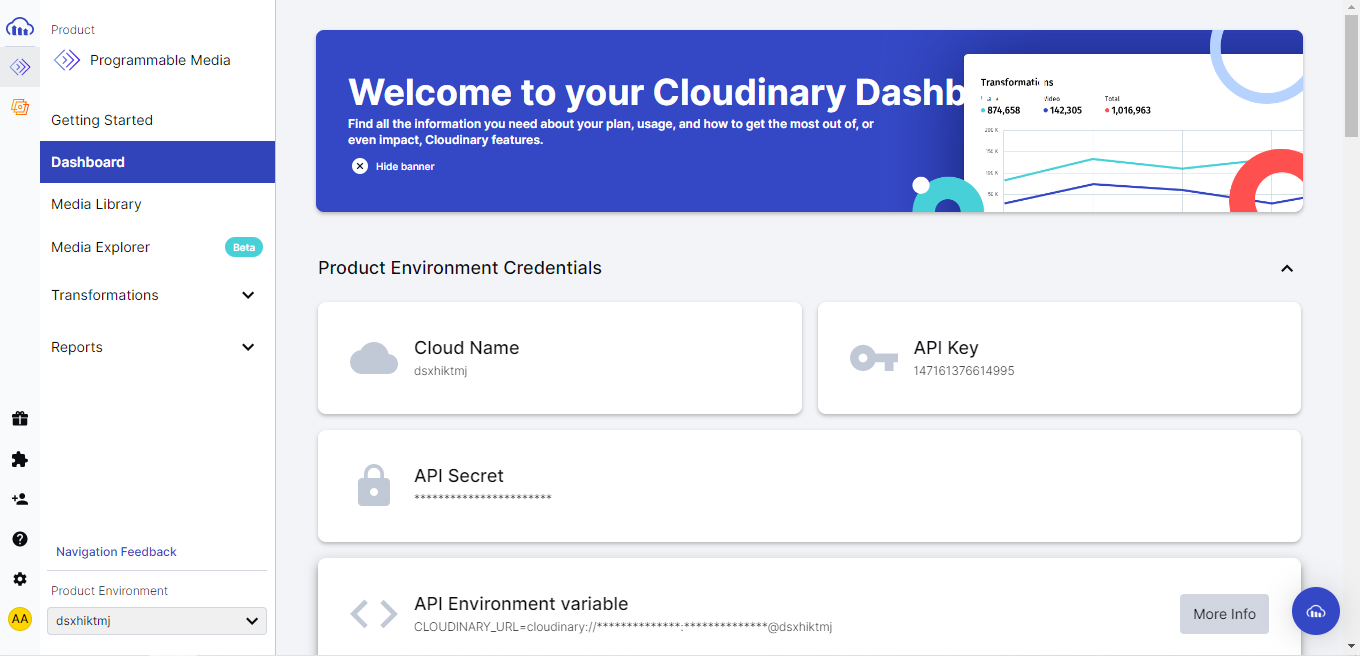
cloudinary.config({

  cloud\_name:process.env.CLOUDINARY\_NAME,

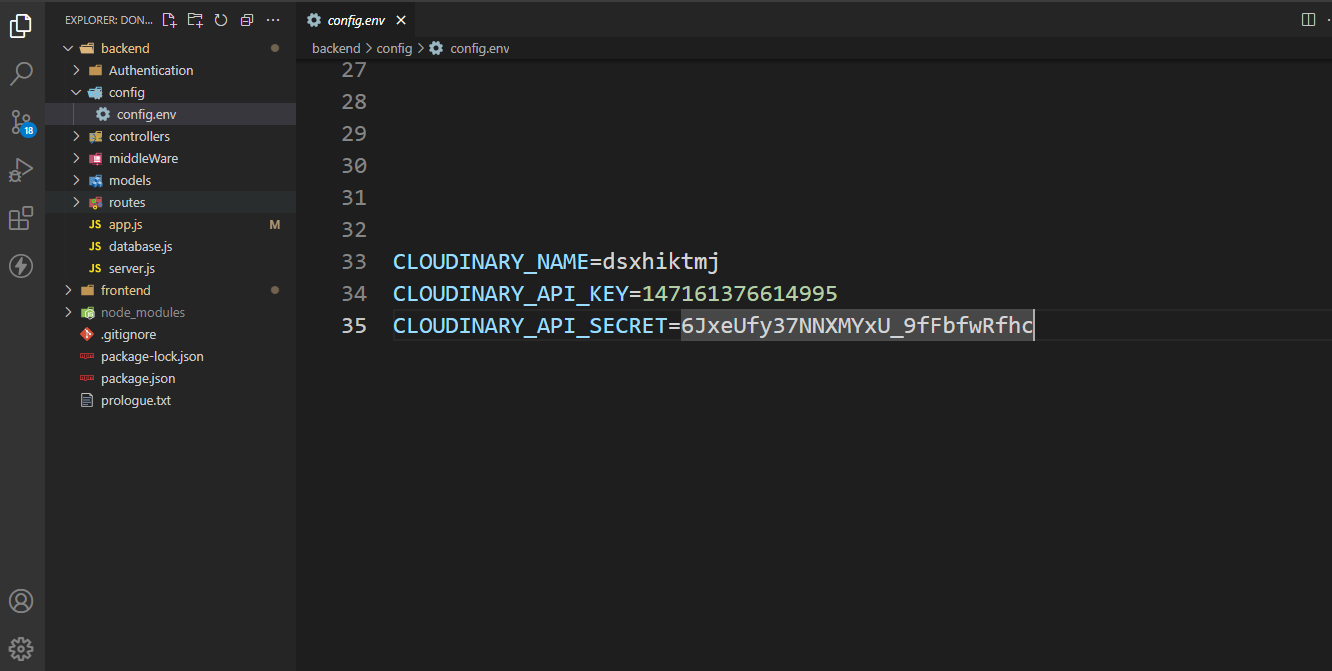
  api\_key:process.env.CLOUDINARY\_API\_KEY,

  api\_secret:process.env.CLOUDINARY\_API\_SECRET,

})



* Add these credentials in config file



* The following is the best example of cloudinary uploading

const register = async (req, res) => {

  try {

    const myCloud = await cloudinary.v2.uploader.upload(req.body.avatar, {

      folder: "avatars",

      effect: "trim:20",

      width: 250,

      crop: "scale",

      quality: 100,

    });

    const { name, email, password } = req.body;

    const user = await User.create({

      name,

      email,

      password,

      avatar: {

        public\_id: myCloud.public\_id,

        url: myCloud.secure\_url,

      },

    });

    const message = "Register Successfully";

    await sendToken(res, user, 201, message);

  } catch (error) {

    return res.status(500).json({

      success: false,

      error,

    });

  }

};

* The following example is for destroying

const updateProfile = async (req, res) => {

  try {

    const userData = {

      name: req.body.name,

      email: req.body.email,

    };

    if (req.body.avatar !== "") {

      //find and destroy image

      const user = await User.findById(req.user.\_id);

      const imageId = user.avatar.public\_id;

      await cloudinary.v2.uploader.destroy(imageId);

      //update image

      const myCloud = await cloudinary.v2.uploader.upload(req.body.avatar, {

        folder: "avatars",

        effect: "trim:20",

        width: 250,

        crop: "scale",

        quality: 100,

      });

      userData.avatar ={

        public\_id:myCloud.public\_id,

        url:myCloud.secure\_url,

      }

    }

    await User.findByIdAndUpdate(req.user.\_id, userData);

    res.status(200).json({

      success: true,

      message: "Update Successfully",

    });

  } catch (error) {

    return res.status(500).json({

      success: false,

      message: error.message,

    });

  }

};