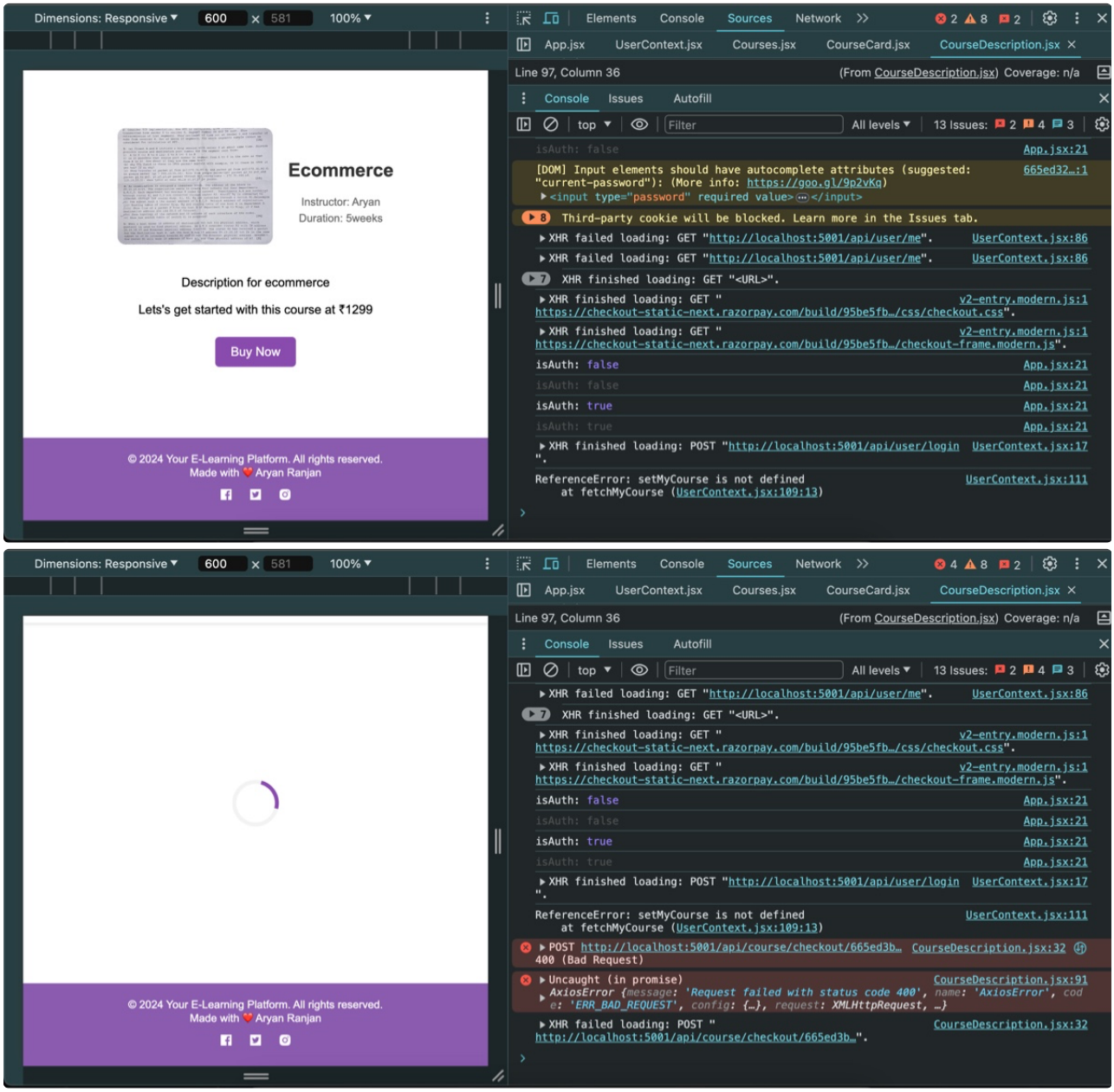
ERROR:

toh buy now par press krne ke baad razorpay ka page aana chahiye, but instead bad request aa rhi hai. Thunderclient mein check kiya toh aa rha hai already purchased ie from checkout handler in server/ course.js



/client/src/pages/coursedescription/CourseDescription.jsx

```
import React, { useEffect, useState } from "react";
import "./courseDescription.css";
import { useNavigate, useParams } from "react-router-dom";
import { CourseData } from "../../context/CourseContext";
import { server } from "../../main";
import axios from "axios";
import toast from "react-hot-toast";
import { UserData } from "../../context/UserContext";
import Loading from "../../components/loading/Loading";

const CourseDescription = ({ user }) => {
  const params = useParams();
  // console.log(params);

  const { fetchCourse, course } = CourseData();
  const navigate = useNavigate();

  const [loading, setLoading] = useState(false);

  const { fetchUser } = UserData();

  useEffect(() => {
    fetchCourse(params.id);
  }, []);

  const checkoutHandler = async () => {
    const token = localStorage.getItem("token");
    setLoading(true);

    const {
      data: { order },
    } = await axios.post(
      `${server}/api/course/checkout/${params.id}`, // server/routes/course.js se ye
endpoint mila h, aur controller mein hamko response mein order mil rha hai. toh vahi
vala order receive kar rhe hai yahan
      {},
      {
        headers: {
          token,
        },
      }
    );

    const options = {
      key: "rzp_test_R5IuwYLmHMfBGy", // .env
      amount: order.id,
      currency: "INR",
      name: "E learning",
      description: "Learn with us",
      order_id: order.id,

      //callbackURL khud se banaya yahan
      handler: async function (response) {
```

```
        const { razorpay_order_id, razorpay_payment_id, razorpay_signature } =
          response; //ye 3 aaya hai form server/controllers/course.js se

        try {
          const { data } = await axios.post(
            `${server}/api/verification/${params.id}`, //routes se aaya ye ie server/
routes/course.js
            {
              //ye req.body ki value hai
              //controller/courses.js mein paymentVerification function mein yehi 3
chizen hai in req.body se, fir body bana rhe hai orderID and paymentID se, fir issi ko
hash kiya with crypto se
              razorpay_payment_id,
              razorpay_order_id,
              razorpay_signature
            },
            { //header diya hai to show we are authenticated
              headers: {
                token,
              },
            }
          );

          await fetchUser();
          await fetchCourses();
          await fetchMyCourse();
          toast.success(data.message);
          setLoading(false);
          navigate(`/payment-success/${razorpay_payment_id}`);
        } catch (error) {
          toast.error(error.response.data.message);
          setLoading(false);
        }
      },
      theme: {
        color: "#8a4baf",
      },
    };

    const razorpay = new window.Razorpay(options);

    razorpay.open();
  };

  return (
    <>
      {loading ? (<Loading />) : (
        <>
          {course && <div className='course-description'>
            <div className='course-header'>
              <img src={`${server}/${course.image}`} alt="" className="course-image" />
              <div className='course-info'>
                <h2>{course.title}</h2>
```

```
                <p>Instructor: {course.createdBy}</p>
                <p>Duration: {course.duration}weeks</p>
              </div>


            </div>
            <p>{course.description}</p>

            <p>Lets's get started with this course at ₹{course.price}</p>
            {user && user.subscription.includes(course._id) ? (
              <button onClick={() => navigate(`/course/study/${course._id}`)}
className='common-btn'>Study</button>
            ) : (
              <button onClick={checkoutHandler} className='common-btn'>Buy Now</button>
            )}
          </div>}
        </>
      )}

    </>
  )
}

export default CourseDescription
```

/server/controllers/course.js

```
import TryCatch from "../middlewares/TryCatch.js";
import { Courses } from "../models/Courses.js";
import { Lecture } from "../models/Lecture.js";
import { User } from "../models/User.js";
import { instance } from "../index.js";
import crypto from "crypto";
import { Payment } from "../models/Payment.js";

export const getAllCourses = TryCatch(async (req, res) => {
    const courses = await Courses.find();
    res.json({ courses, });
});

export const getSingleCourse = TryCatch(async (req, res) => {
    const course = await Courses.findById(req.params.id);
    res.json({ course, });
});

export const fetchLectures = TryCatch(async (req, res) => {
    const lectures = await Lecture.find({ course: req.params.id });

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

```
    if (user.role === "admin") {
        return res.json({ lectures });
    }

    if (!user.subscription.includes(req.params.id)) {
        return res.status(400).json({
            message: "Please subscribe to access the content",
        });
    };
    res.json({ lectures });
});

export const fetchLecture = TryCatch(async (req, res) => {
    const lecture = await Lecture.findById(req.params.id);

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

    if (user.role === "admin") {
        return res.json({ lecture });
    }

    if (!user.subscription.includes(req.params.id)) {
        return res.status(400).json({
            message: "Please subscribe to access the content",
        });
    };
    res.json({ lecture });
});

export const getMyCourses = TryCatch(async (req, res) => {
    const courses = await Courses.find({ _id: req.user.subscription });
    res.json({ courses });
});

export const checkout = TryCatch(async (req, res) => {
    const user = await User.findById(req.user._id);
    const course = await Courses.findById(req.params.id);
    if (!user.subscription.includes(course._id)) {
        return res.status(400).json({ message: "You already have this course" });
    }
    const options = {
        amount: Number(course.price * 100),
        currency: "INR",
    };

    const order = await instance.orders.create(options);
    res.status(201).json({ order, course, });

});


export const paymentVerification = TryCatch(async (req, res) => {
    const { razorpay_payment_id, razorpay_order_id, razorpay_signature } = req.body;
```

```
    const body = razorpay_order_id + "|" + razorpay_payment_id;
    const expectedSignature = crypto.createHmac('sha256',
process.env.Razorpay_Secret).update(body).digest('hex');
    const isAuthentic = expectedSignature === razorpay_signature;
    if (isAuthentic) {
        await Payment.create({
            razorpay_order_id,
            razorpay_payment_id,
            razorpay_signature
        });

        const user = await User.findById(req.user._id);
        const course = await Courses.findById(req.params.id);

        user.subscription.push(course._id);
        await user.save();
        return res.status(200).json({ message: "Subscription successful" });
    } else {
        return res.status(400).json({ message: "Invalid signature sent!" });
    };
});
```

/server/routes/course.js

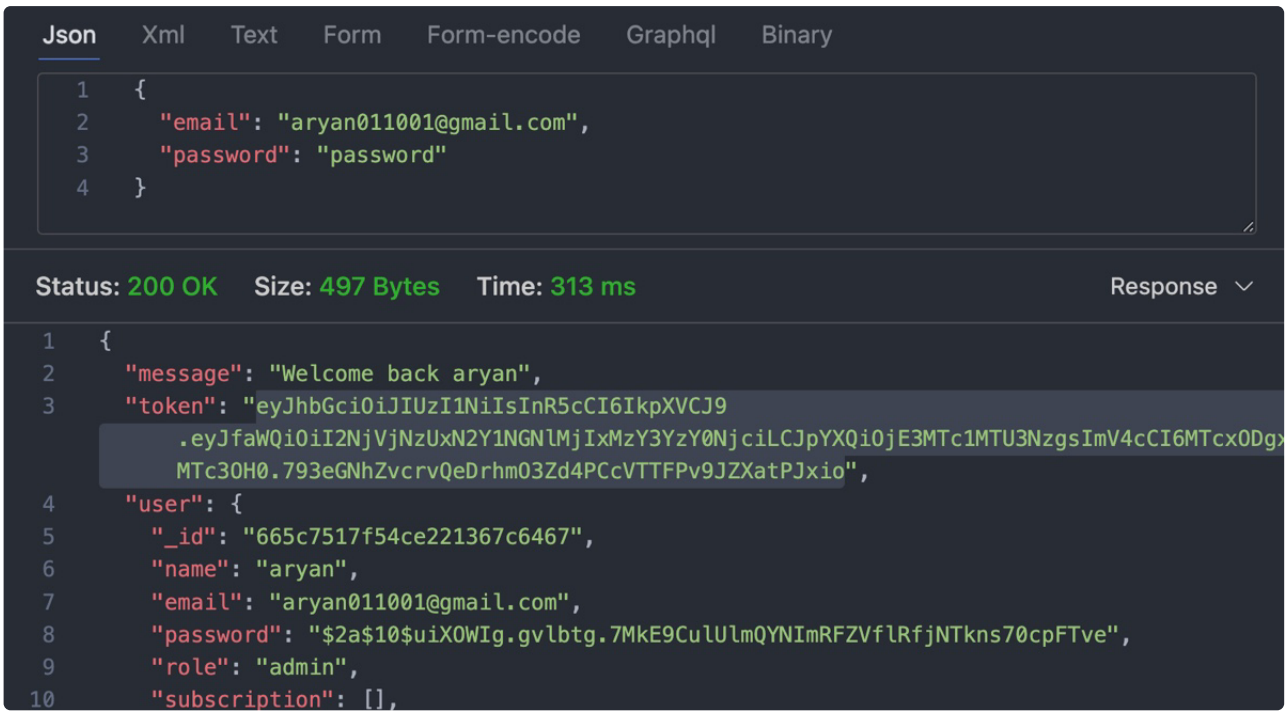
```
import express from 'express';
import { checkout, fetchLecture, fetchLectures, getAllCourses, paymentVerification }
from '../controllers/course.js';
import { getSingleCourse } from '../controllers/course.js';
import { isAuth } from '../middlewares/isAuth.js';
import { getMyCourses } from '../controllers/course.js';

const router=express.Router();

router.get('/course/all', getAllCourses);
router.get('/course/:id', getSingleCourse);
router.get('/lectures/:id', isAuth, fetchLectures);
router.get('/lecture/:id', isAuth, fetchLecture);
router.get('/mycourse', isAuth, getMyCourses);
router.post('/course/checkout/:id', isAuth, checkout);
router.post('/verification/:id', isAuth, paymentVerification);

export default router;
```

ThunderClient screenshot for login (so that token mil jaaye which will be used in checkout)



ThunderClient screenshot for checkout(isme token dala hai which we got from login)