Industrial Training Report
(Summer Internship)
Presentation

On

User Authentication

Submitted To

Mr. Basudeo Singh Roohani

Ms. Shanu Priya Chauhan

(Internship Coordinators)

Submitted By

Prince

2201430109011

7th Sem, CSE-1

Supervised By

Aditya Batra

Assistant Professor

CSE Department

Department of Computer Science & Engineering

IMS Engineering College, Ghaziabad, India

(2024-25)

INTRODUCTION

[COMPANY PROFILE ↓]

CETPA INFOTECH COMPANY PROFILE

- **ABOUT**
  - The aim of the company is to furnish the decisive and applicable products in a very economical time.
  - CETPA offers College Campus Training Programs to students pursuing B.Tech, B.E., M.Tech, BCA, MCA etc., within the college campus itself.
- **SPECIALITIES**
  - CETPA has specialization in 3 important domains namely:
    - TRAINING
    - DEVELOPMENT
    - CONSULTANCY.
- **SERVICES**
  - The company provides specialized training in 50+ leading technologies like . NET, Java Ethical Hacking, and many more

REQUIREMENT ANALYSIS

- **User Registration**
  - Name
  - Email Id
  - Password
  - Confirm Password
  - Terms and Condition (in checkbox from)
- **User Login**
  - Email
  - Password
- **Profile Page**
  - information

PROBLEM STATEMENT

- Develop a secure and scalable user authentication system using the MERN stack to enable seamless registration, login, Authorization and profile for users of an online platform.

OBJECTIVES

- To implement a secure and efficient user authentication system in a MERN stack application by leveraging MongoDB, Express.js, React, and Node.js.
- This system will ensure protection of user data through
  - JWT (JSON Web Tokens) for secure session management.
  - bcrypt for password hashing.
  - HTTPS for encrypted communication.
- The goal is to create a seamless user experience while maintaining high standards of security and data integrity, facilitating safe and reliable user registration, login, and session management.

SYSTEM REQUIREMENT SPECIFICATION

1. Hardware Requirements For This Project :

a) Deployment Machine

- **Processor :** Intel i5/Ryzen 5 or better (modern multi-core processors are preferred for smooth development).
- **RAM :** 8 GB or more (16 GB recommended for handling multiple applications and services simultaneously).
- **Storage :** SSD with at least 256 GB (faster read/write speeds help with development efficiency).
- **Network :** Stable internet connection.

2. Software Requirements For This Project :

a) Operation System

b) Code Editor

c) MongoDB Compass

d) Framework And There Packages

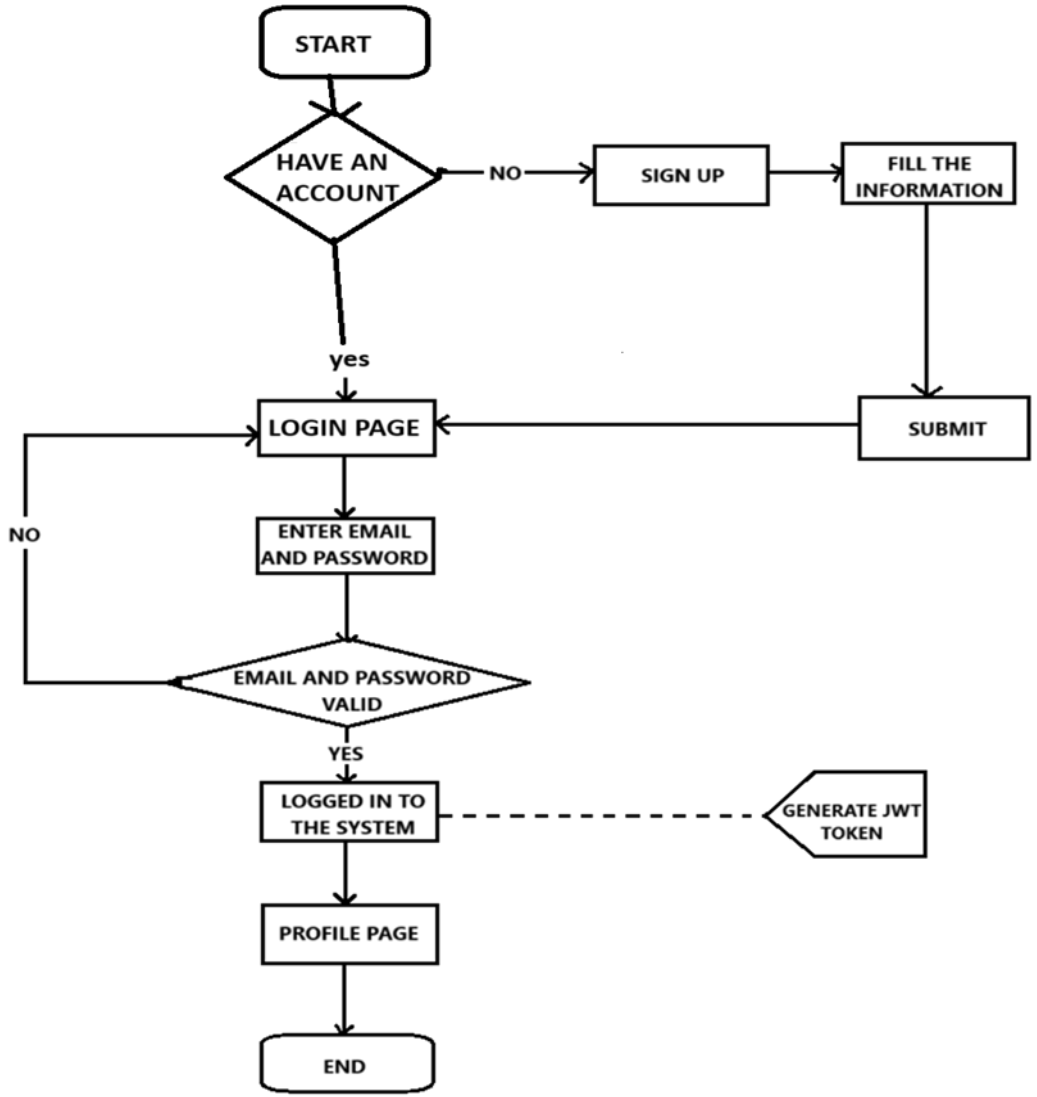
- For Frontend
  - Axios
  - React-Router-Dom
  - Tailwind CSS
- For Backend
  - JWT(JsonWebToken)
  - Bcrypt
  - Nodemon

TECHNOLOGY USED

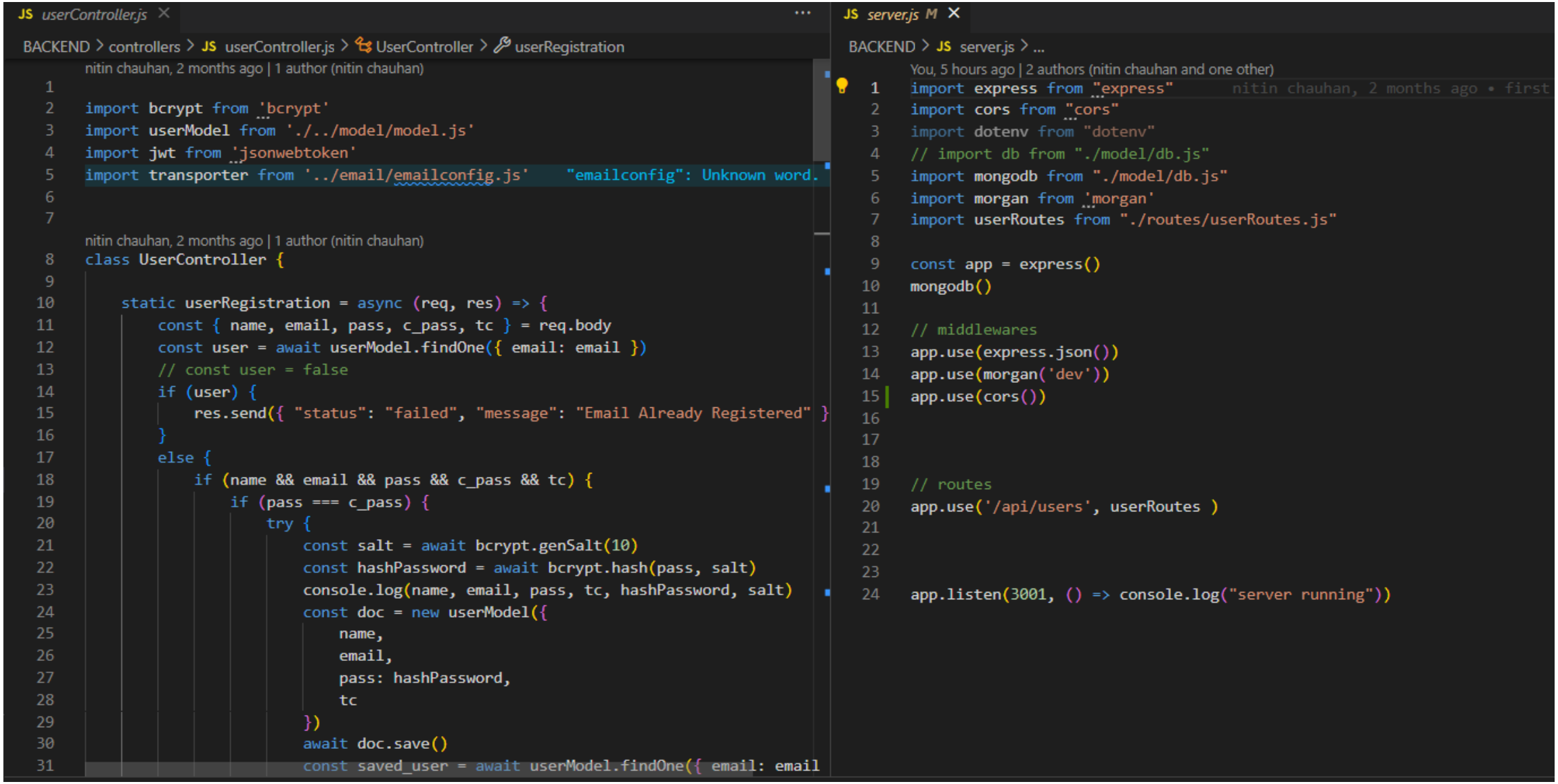
- **VS CODE EDITOR**
- **We Used In Frontend**
  - ReactJS : Library to create web interface.
  - Redux : For State-management.
  - React-Router-Dom : It is help to create routes for web pages.
  - Tailwind CSS : It is Open source CSS Framework.
  - Axios : Through this we can create HTTP request.
- **We Used In Backend**
  - NodeJS **:** It is use for the Backend Framework.
  - ExpressJS : To Create Server.
  - MongoDB Compass : For storing data in Database.

METHODOLOGY

- **Flow Chart**



IMPLEMENTATION



userController.js

BACKEND > controllers > userController.js > UserController > userRegistration

nitin chauhan, 2 months ago | 1 author (nitin chauhan)

```
import bcrypt from 'bcrypt'
import userModel from './../model/model.js'
import jwt from 'jsonwebtoken'
import transporter from '../email/emailconfig.js'    "emailconfig": Unknown word.


nitin chauhan, 2 months ago | 1 author (nitin chauhan)
class UserController {

    static userRegistration = async (req, res) => {
        const { name, email, pass, c_pass, tc } = req.body
        const user = await userModel.findOne({ email: email })
        // const user = false
        if (user) {
            res.send({ "status": "failed", "message": "Email Already Registered" }
        }
        else {
            if (name && email && pass && c_pass && tc) {
                if (pass === c_pass) {
                    try {
                        const salt = await bcrypt.genSalt(10)
                        const hashPassword = await bcrypt.hash(pass, salt)
                        console.log(name, email, pass, tc, hashPassword, salt)
                        const doc = new userModel({
                            name,
                            email,
                            pass: hashPassword,
                            tc
                        })
                        await doc.save()
                        const saved_user = await userModel.findOne({ email: email
```

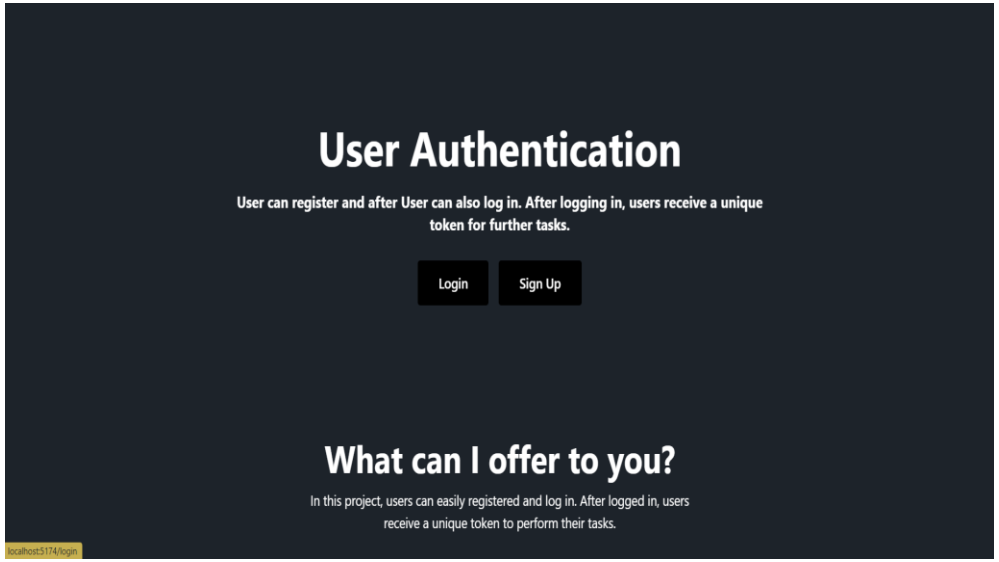
server.js M

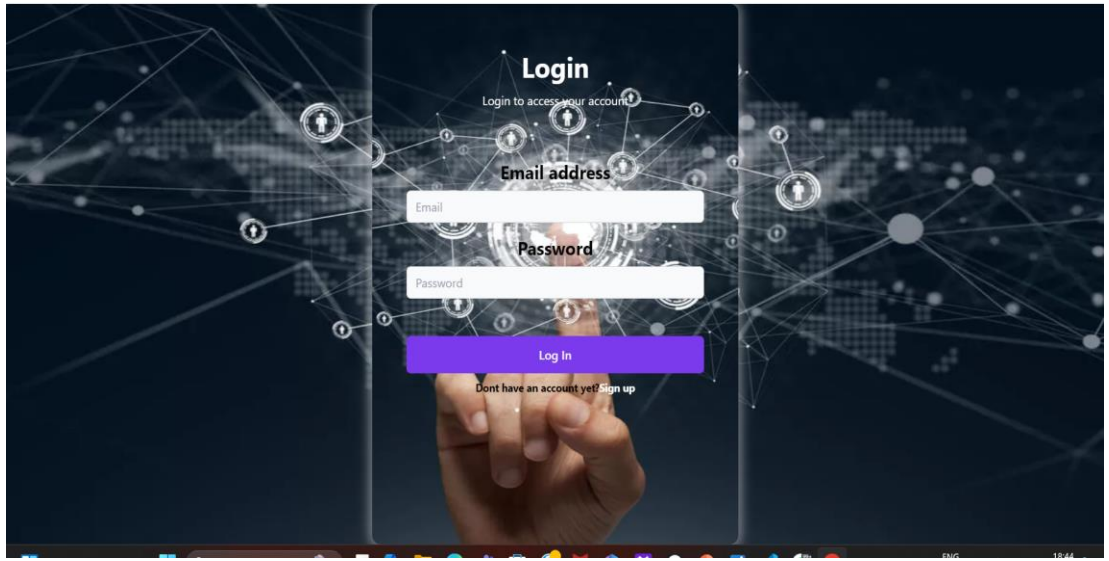
BACKEND > server.js > ...

You, 5 hours ago | 2 authors (nitin chauhan and one other)

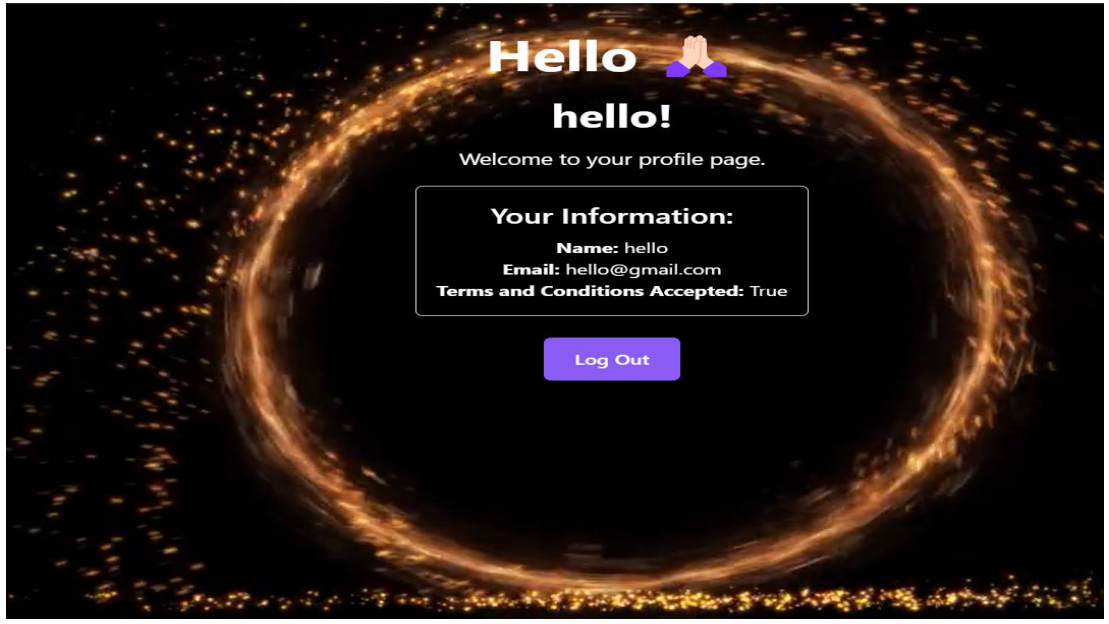
```
import express from "express"      nitin chauhan, 2 months ago • first
import cors from "cors"
import dotenv from "dotenv"
// import db from "./model/db.js"
import mongodb from "./model/db.js"
import morgan from 'morgan'
import userRoutes from "./routes/userRoutes.js"

const app = express()
mongodb()

// middlewares
app.use(express.json())
app.use(morgan('dev'))
app.use(cors())



// routes
app.use('/api/users', userRoutes )



app.listen(3001, () => console.log("server running"))
```

RESULTS


User Authentication
User can register and after User can also log in. After logging in, users receive a unique token for further tasks.
Login
Sign Up
What can I offer to you?
In this project, users can easily registered and log in. After logged in, users receive a unique token to perform their tasks.
localhost:5174/login


Login
Login to access your account
Email address
Email
Password
Password
Log In
Dont have an account yet? Sign up


Sign Up
Create your account
Name
Name
Email address
Email
Password
Password
Confirm Password
Confirm Password
Terms & Conditions
I agree to the terms and conditions
Sign Up
Already have an account? Log In


Hello
hello!
Welcome to your profile page.
Your Information:
Name: hello
Email: hello@gmail.com
Terms and Conditions Accepted: True
Log Out

CONCLUSION

- In my Project I used the MERN stack with jsonwebtoken which help us build a safe authentication system. It keeps user data secure and makes it easy to manage user sessions while providing a smooth experience for users logging in and signing up.

REFERENCES

- [1]S.W. Shah and S. S. Kanhere, “Recent Trends in User Authentication – A Survey”, in IEEE Access, vol. 7, pp. 112505-112519, 2023, doi: 10.1109/ACCESS.2023.2932400.
- [2] Koushik Sen build “Highly scalable, developer-friendly API for the modern web with JavaScript and Node.js”, published on 2023 page No.32.
- [3] Desai, K. and Fiaidhi, J. “Developing a Social Platform using MERN Stack”, Authorea Preprints.Vancouver. 2022.
- [4] Mahindrakar, P., & Pujeri, U. “Security Implications for Json web Token Used in MERN Stack for Developing E-Commerce Web Application”, International Journal of Engineering and Advanced Technology (IJEAT), 10(1). (2020).

Thank You