**Registration || Login form**

# **Register a User**

* Save data in Data-Base using app.post

app.post("/register", async (req, res) => {

    try {

        const pass = await req.body.password;

        const cpass = await req.body.cpassword;

        if (pass === cpass) {

            const registerEmploye = new Register({

                //schema = user fill data in input field

                firstname:req.body.firstName,

                lastname:req.body.lastName,

                email:req.body.emailAdress,

                phone:req.body.phoneNumber,

                age:req.body.myAge,

                password:req.body.password,

                confirmpassword:req.body.cpassword

            });

            const registered = await registerEmploye.save()

            res.send("index.hbs").status(201)

        } else {

            res.send("your password is not matching either")

        }

    } catch (error) {

        res.send(error).status(401)

    }

})

# **Login User**

app.post("/login", async (req, res) => {

    try {

        const lEmail = await req.body.lEmail;

        const lPassword = await req.body.lPassword;

        // console.log(`${lEmail} and ${lPassword}`);

        const userEmail = await Register.findOne({ email: lEmail }) //findOne give you promise and a whole collection present in database

        if (userEmail.password===lPassword) {

            res.send("loged in")

        } else {

            res.send("invalid Login details")

        }

    } catch (error) {

        res.send("invalid details").status(401)

    }

})

# **Bcrypt Algorithm to Save Password**

* Past below the defined Schema

1. const bcrypt = require("bcryptjs");
2. employeSchema.pre("save" , async function(next){
3. //this.password gives you input user password
4. //input field password = us ko bcrypt kr do bus
5. if(this.isModified("password")){
7. this.password = await bcrypt.hash(this.password,10)
8. this.confirmpassword = await bcrypt.hash(this.confirmpassword,10)
9. }
11. next();
13. })

# **Login User with Bcrypt Compare**

app.post("/login", async (req, res) => {

    try {

        const lEmail = await req.body.lEmail;

        const lPassword = await req.body.lPassword;

        const userEmail = await Register.findOne({ email: lEmail })

//findOne give you promise and a whole collection present in database

        const isMatch = await bcrypt.compare(lPassword,userEmail.password)

        if (isMatch) {

            res.send("loged in")

        } else {

            res.send("invalid Login details")

        }

    } catch (error) {

        res.send("invalid details").status(401)

    }

})

# **Cookies**