**DEPARTMENT OF INFORMATION TECHNOLOGY**

|  |
| --- |
| NAME: Anish Sharma  ROLL NO: I011  SAP ID: 60003220045  BRANCH: Information Technology  BATCH: 1 |

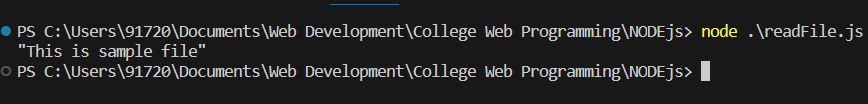
# EXPERIMENT NO. 9

**CO/LO:** Develop web applications.

**AIM / OBJECTIVE:** Program based on inbuilt functions in Node.js.

**Task 1: Working with File System**

1. **Reading a File:**
2. const fs = require('fs');
3. fs.readFile('sample.txt', 'utf8', (err, data) => {
4. if (err) {
5. console.error(err); return;
6. } console.log(data);
7. });



**2. Writing to a File:**

const fs = require('fs'); fs.writeFile('output.txt', 'Hello, Node.js!', (err) => {

    if (err) {

        console.error(err);

        return;

    } console.log('File written successfully!');

});

****

**3. Checking if a File Exists:**

const fs = require('fs');

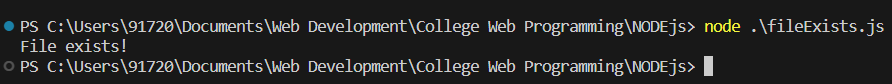
fs.access('check.txt', fs.constants.F\_OK, (err) => {

    if (err) {

        console.error(err); return;

    } console.log('File exists!');

});

****

**Task2: Handling HTTP Requests**

1. **Creating a Basic Web Server:**

const http = require('http');

const server = http.createServer((req, res) => {

    res.writeHead(200, { 'Content-Type': 'text/html' });

    res.end('<h1>Hello, World!</h1>');

});

server.listen(3000, '127.0.0.1', () => {

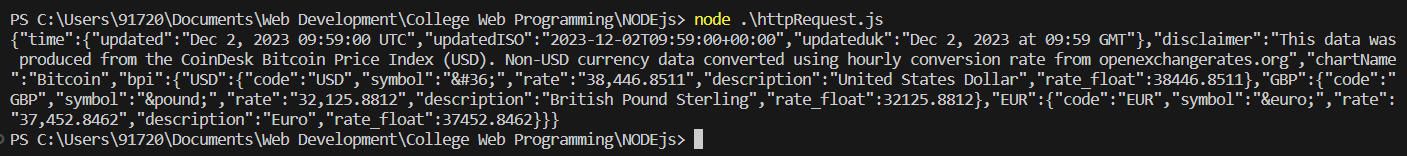
    console.log('Server is running on http://127.0.0.1:3000/');

});

****

****

1. **Sending an HTTP Request:**
2. const http = require('http');
3. const options = {
4. hostname: 'api.coindesk.com',
5. path: '/v1/bpi/currentprice.json',
6. method: 'GET',
7. port: 80,
8. };
9. const req = http.request(options, (res) => {
10. let data = '';
11. res.on('data', (chunk) => {
12. data += chunk;
13. });
14. res.on('end', () => {
15. console.log(data);
16. });
17. });
18. req.on('error', (e) => {
19. console.error(`Problem with request: ${e.message}`);
20. });
21. req.end();

****

**BOOKS AND WEB RESOURCES:**

1. Tailwind CSS From Scratch | Learn By Building Projects by Brad Traversy
2. Tailwind CSS: A Modern Way To Build Websites Using CSS

**CONCLUSION:** I learn**ed** about file System module and implemented http requests.

**DEPARTMENT OF INFORMATION TECHNOLOGY**

|  |
| --- |
| NAME: Anish Sharma  ROLL NO: I011  SAP ID: 60003220045  BRANCH: Information Technology  BATCH: 1 |

# EXPERIMENT NO. 10

**CO/LO:** Develop web applications.

**AIM / OBJECTIVE:** Program based on inbuilt functions in Node.js.

**BACKEND**

Index.js

const express = require('express');

const Users = require('./models/users');

const cors = require('cors');

require('./models/config');

const app = express();

app.use(express.json());

app.use(cors());

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

    let data = new Users(req.body);

    let result = await data.save();

    result = result.toObject();

    delete result.password;

    if (result) {

        if (err) {

            res.send({ result: "Something went wrong, Please tryagain later" })

        }

        res.send({ result, auth: token });

    }

});

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

    if (req.body.password && req.body.email) {

        let user = await Users.findOne(req.body).select("-password");

        if (user) {

            if (err) {

                res.send({ result: "Something went wrong, Please tryagain later" })

            }

            res.send({ user, auth: token });

        }

        else {

            res.send({ result: "No User found" })

        }

    }

    else {

        res.send({ result: "No User found" })

    }

});

app.listen(3000);

models/config.js

const mongoose = require('mongoose');

mongoose.connect("mongodb+srv://gofood:mern123@cluster0.otuonrs.mongodb.net/gofoodmern?retryWrites=true&w=majority");

models/Users.js

const mongoose = require('mongoose');

const userSchema = new mongoose.Schema({

    name:String,

    email:String,

    password:String

});

module.exports= mongoose.model('users',userSchema);

**FRONTEND**

\src\App.js

import './App.css';

import Nav from './Components/Nav';

import SignUp from './Components/SignUp';

import PrivateComponent from './Components/PrivateComponent';

import { BrowserRouter, Routes, Route } from 'react-router-dom';

import Login from './Components/Login';

function App() {

  return (

    <div className="App">

      <BrowserRouter>

      <Nav/>

      <Routes>

        <Route element={<PrivateComponent />} >

<Route path="/" element={<Home/>} />

        </Route>

        <Route path="/signup" element= {<SignUp/>} />

        <Route path="/login" element={<Login/>} />

      </Routes>

      </BrowserRouter>

    </div>

  );

}

export default App;

src\components\SignUp.js

import React,{useState, useEffect} from "react";

import { useNavigate } from "react-router-dom"; //used to redirect after signup

const SignUp = () =>{

    const navigate=useNavigate(); //imp to use with navigate

    useEffect(() => {             //very imp to note

        const auth = JSON.parse(localStorage.getItem("user"));

        if (auth) {

          navigate("/");

        }

      }, []);

    const [name, setName]=useState("");

    const [email, setEmail]=useState("");

    const [password, setPassword]=useState("");

    const collectData =async () =>{

        console.warn(name,email,password);

        let result = await fetch("http://localhost:3000/signup",{

            method:'post',

            body:JSON.stringify({name,email,password}),

            headers:{

                'Content-Type':'application/json'

            }

        });

        result= await result.json();

        console.warn(result);

        localStorage.setItem("user",JSON.stringify(result)); //storing data locally in browser

        if(result)

        {

            navigate('/');

        }

    }

    return(

<div className="registerContainer">

          <div className="register">

            <h1>REGISTER</h1>

            <input

              className="inputBox"

              value={name}

              onChange={(e) => setName(e.target.value)}

              type="text"

              placeholder="Enter name"

            />

            <input

              className="inputBox"

              value={email}

              onChange={(e) => setEmail(e.target.value)}

              type="text"

              placeholder="Enter email"

            />

            <input

              className="inputBox"

              value={password}

              onChange={(e) => setPassword(e.target.value)}

              type="password"

              placeholder="Enter password"

            />

            <button

              className="signupbutton"

              type="button"

              onClick={collectData}

            >

              SignUp

            </button>

          </div>

        </div>

    )

}

export default SignUp;

src\components\Login.js

import React, { useState, useEffect } from "react";

import { useNavigate } from "react-router-dom";

const Login = () => {

  const navigate = useNavigate();

  useEffect(() => {

    const auth = JSON.parse(localStorage.getItem("user"));

    if (auth) {

      navigate("/");

    }

  }, [navigate]);

  const [email, setEmail] = useState("");

  const [password, setPassword] = useState("");

  const handleLogin = async () => {

    console.warn(email, password);

    try {

      const response = await fetch("http://localhost:3000/login", {

        method: "post",

        body: JSON.stringify({ email, password }),

        headers: {

          "Content-Type": "application/json",

        },

      });

      if (!response.ok) {

        throw new Error("Login failed"); // Handle non-200 status codes

      }

      const result = await response.json();

      console.warn(result);

      if (result.name) {

        localStorage.setItem("user", JSON.stringify(result));

        navigate("/");

      } else {

        alert("Enter correct password/username");

      }

    } catch (error) {

      console.error("Error during login:", error.message);

      alert("Login failed. Please try again later.");

    }

  };

  return (

    <>

      <h1>LOGIN</h1>

      <div className="login">

        <input

          className="inputBox"

          type="text"

          value={email}

          onChange={(e) => setEmail(e.target.value)}

          placeholder="Enter Email"

        />

        <input

          className="inputBox"

          type="password" // Change the input type to 'password'

          value={password}

          onChange={(e) => setPassword(e.target.value)}

          placeholder="Enter Password"

        />

      </div>

      <button className="loginbutton" type="button" onClick={handleLogin}>

        LOGIN

      </button>

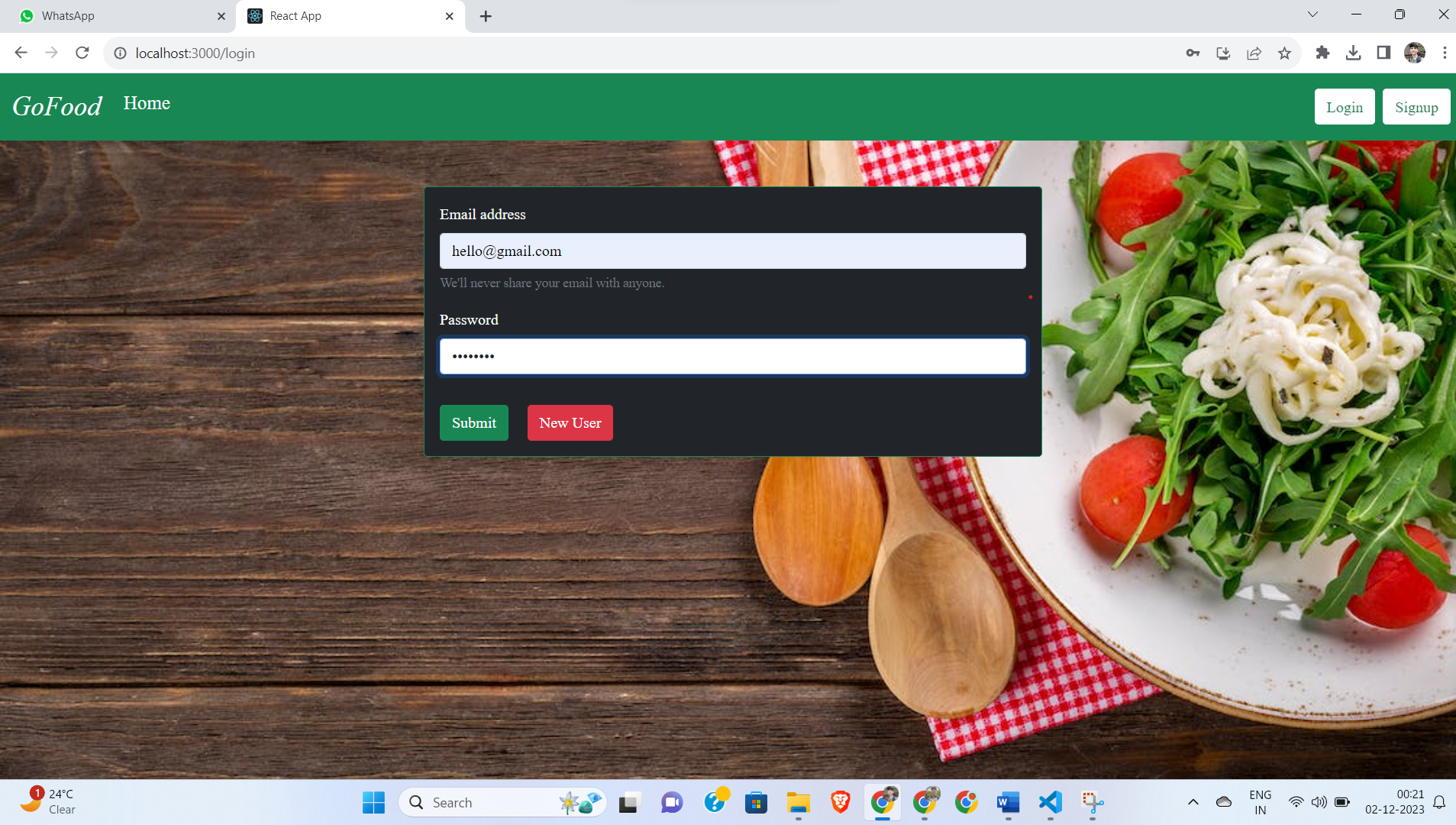
    </>

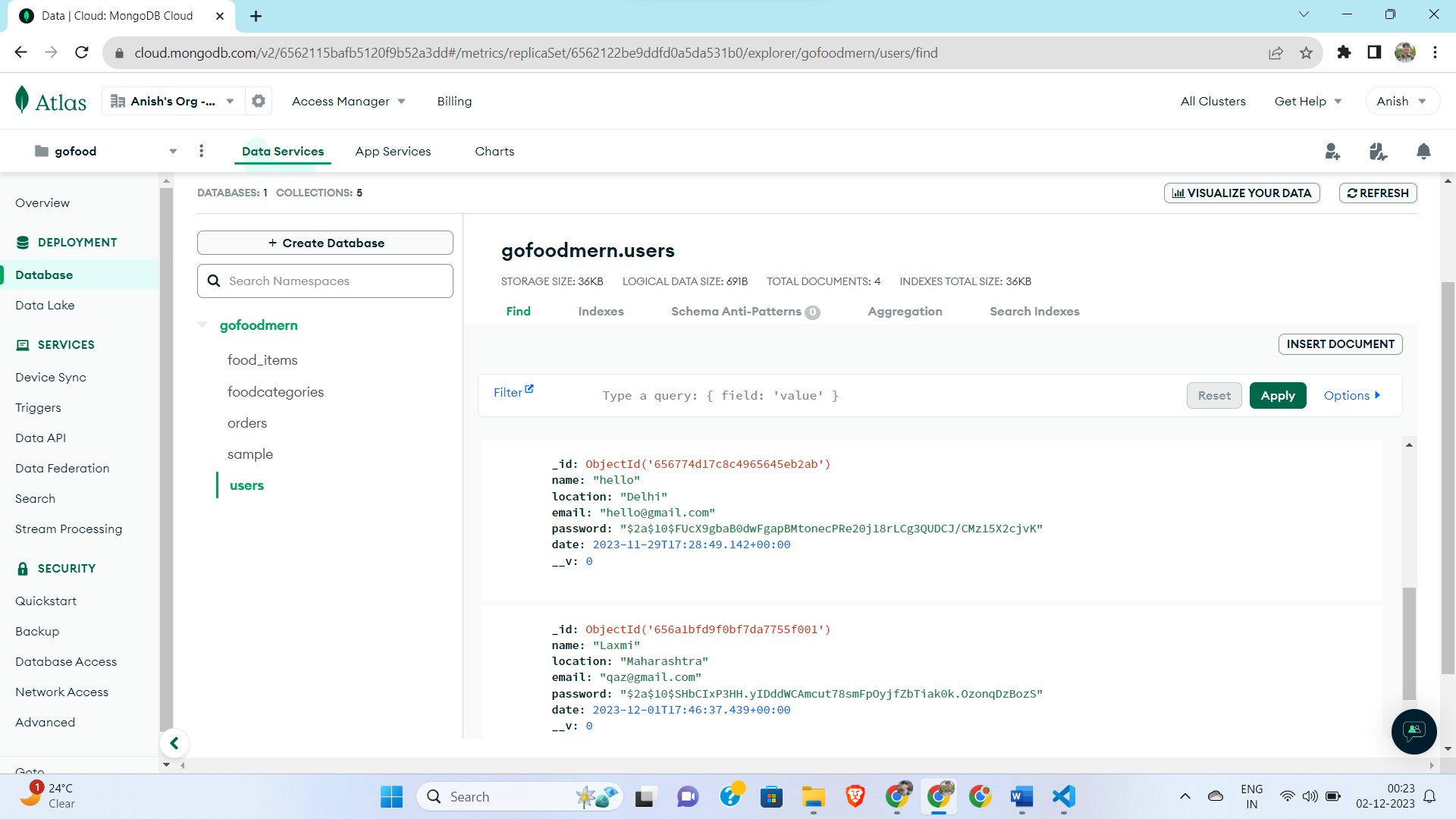
  );

};

export default Login;

****

****

****

**BOOKS AND WEB RESOURCES:**

1. Tailwind CSS From Scratch | Learn By Building Projects by Brad Traversy
2. Tailwind CSS: A Modern Way To Build Websites Using CSS

**CONCLUSION:** I learnt to integrate mongoDB with frontend and backend and successfully created and integrated signup and login api.