**Mern Stack project 1**

**MongoDB database**

1. **I created an mongodb account on** [**https://cloud.mongodb.com/v2/61b712d921d1275467fafcb5#setup/access?includeToast=true**](https://cloud.mongodb.com/v2/61b712d921d1275467fafcb5#setup/access?includeToast=true)
2. **The created a project name(mern\_stack\_project1) and database(Mernstackcluster)**
3. **Create user and password**
4. **Use connection method(mongodb Compass)**
5. **Download and install Mongodb compass.**
6. **Then I used the connection url from mongodb account to create database and also used it on index.js file as shown below:**

Graphical user interface, text, application, email

Description automatically generated

**NB: change the password to your chosen password(Mernstack\_user1)**

**GUI Mongodb database on my laptop**

Graphical user interface, application, table

Description automatically generated

Text

Description automatically generated

**To connect the database to the application follow the below steps:**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

**Replace the former url with the application url as shown below and then create the real database and change the FirstProject Database in the url to the one you created.**

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

1. **Then in the VS code, I created a folder for server and client**
2. **In the server folder, and issued “npm init -y” to build package.json file automatically**

Text

Description automatically generated

1. **Express- to created backend and mongoose- a module to make backend communicate with database, cors- to create a react app in the backend, nodemon**

**Npm install express mongoose cors nodemon**

Text

Description automatically generated

**Create Index.js file( to write code for API and database setup**

Text

Description automatically generated

**Nodemon to track changes. Go to the package.json file to adjust the code to use nodemon as shown below:**

**Then ‘npm start’ to run**

Text

Description automatically generated

**Setting the fields that are present in mongodb in the server, create another folder(named models)**

**And create users.js file inside it as shown below:**

**User.js**

//import mongoose

const mongoose = require('mongoose')

//creating a schema for users

const usersSchema = new mongoose.Schema({

    name: {

        type: String,

        required: true,

    },

    age: {

        type: Number,

        required: true,

    },

    username: {

        type: String,

        required: true

    }

});

const userModel = mongoose.model("users", usersSchema)

//to have access to userModel outside of this file:

module.exports = userModel;

**index.js**

//variable for express and app

const express = require('express');

const app = express()

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

//to connect API to the frontend(client)

const cors = require("cors");

app.use(cors());

//for postman not to give an error use:

app.use(express.json());

//setting mongodb connection

const mongoose = require('mongoose')

mongoose.connect("mongodb+srv://Mernstack\_user1:Mernstack\_user1@mernstackcluster1.jkoep.mongodb.net/mern\_social\_database?retryWrites=true&w=majority")

//to make request to the database

app.get("/getUsers", (req, res)=>{

    userModel.find({}, (err,result)=> {

        if (err){res.json(error)

        } else {

            res.json(result);

        }

    });

});

//to send request to the database

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

    const user = req.body

    const newUser = new userModel(user);

    await newUser.save();

    res.json(user)

});

//for Api to start

app.listen(3001, ()=> {

    console.log("Server runs perfectly...");

});

**Testing with postman**

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

**For Client side**

1. Using ‘npx install create-react-app .’ you can install the frontend development modules
2. Npm Install axios( a library that enable easy communication with the API request)

**App.js**

import './App.css';

import { useState, useEffect } from "react";

import Axios from 'axios';

function App() {

  const [listOfUsers, setListOfUsers] = useState([]);

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

  const [age, setAge] = useState(0);

  const [username, setUsername] = useState("");

  useEffect(() => {

    Axios.get("http://localhost:3001/getUsers").then((response)=> {

      setListOfUsers(response.data)

    })

  }, [])

  //a function that will submit request to backend

  const createUser = () => {

    Axios.post("http://localhost:3001/createUsers", {

    name: name,

    age:age,

    username:username,

  }).then((response)=>{

      setListOfUsers([...listOfUsers, {name: name,

    age:age,

    username:username}]);

    });

  };

  return (

    <div className="App">

    <div className="usersDisplay">

    {listOfUsers.map((user) => {

      return(<div>

        <h1>Name: {user.name}</h1>

        <h1>Age: {user.age}</h1>

        <h1>Username: {user.username}</h1>

        </div>);

    })}

    </div>

    <div>

    <input type="text" placeholder='Name....' onChange={(event)=> {setName(event.target.value);}}/>

      <input type="number" placeholder='Age....' onChange={(event)=> {setAge(event.target.value);}} />

        <input type="text" placeholder='UsernameName....' onChange={(event)=> {setUsername(event.target.value);}}/>

        <button onClick={createUser}> Create User</button>

    </div>

    </div>

  );

}

export default App;

Graphical user interface, text, application, email

Description automatically generated