users.service.js

```
import UsersModel from '../models/users.model.js'
import bcrypt from 'bcrypt'

const create = async (data) => {
  try {
    let result = await UsersModel.create(data)
    console.log(result)
    return result
  } catch (error) {
    throw error.message
  }
}

const update = async (data, id) => {
  try {
    let result = await UsersModel.updateOne({ _id: id }, data)
    return result
  } catch (error) {
    throw error.message
  }
}
const getAll = async () => {
  try {
    let result = await UsersModel.find()
    return result
  } catch (error) {
    throw error.message
  }
}
const getOne = async (id) => {
  try {
    let result = await UsersModel.findById({ _id: id })
    return result
  } catch (error) {
    throw error.message
  }
}

const deleteOne = async (id) => {
  try {
    let result = await UsersModel.deleteOne({ _id: id })
    return result
  } catch (error) {
    throw error.message
  }
}

const register = async (data) => {
  try {
    let result = await UsersModel.create(data)
    console.log(result)
    return result
  } catch (error) {
    throw error.message
  }
}

const login = async (data) => {
  try {
    let email = data.email
    let password = data.password
    let user = await UsersModel.findOne({ email }).lean()
    if (!user) throw new Error('User Not Fond!')
    let result = await bcrypt.compare(password, user.password)
    if (!result) throw new Error('Passowrd Not matched!')
    delete user.password
    return user
  } catch (error) {
    throw error.message
  }
}

const UsersService = {
  create,
  update,
  getAll,
  getOne,
  deleteOne,
  register,
  login,
}

export default UsersService
```