Лабораторна робота №4

Mongoose.

# Виконання роботи

Скопіювати проект попередньої лабораторної роботи та вдосконалити використовуючи ODM-бібліотеку.

## Встановити Mongoose та створити схему користувача.

### Результат встановлення mongoose



### Лістинг створення моделі

const { Schema, model } = require("mongoose");

const userScheme = new Schema({

  name: { type: String, required: true },

  age: { type: Number, required: true },

});

module.exports = {

  UserModel: model("User", userScheme),

};

## Замінити методи роботи з бд на методи mongoose

### Лістинг методів взаємодії з бд

const { UserModel } = require("./schemes");

const mongoose = require("mongoose");

const uri = "mongodb://localhost:27017";

async function connectToMongo() {

  try {

    await mongoose.connect(uri);

    console.log("Connected to MongoDB");

  } catch (error) {

    console.error("Error connecting to MongoDB:", error);

    throw new Error(error?.message);

  }

}

async function closeMongo() {

  try {

    await mongoose.close();

    console.log("MongoDB connection closed.");

  } catch (error) {

    console.error("Error closing MongoDB connection:", error);

    throw new Error(error?.message);

  }

}

async function getUser(query) {

  const user = await UserModel.findOne(query);

  if (!user) throw new Error("User not found");

  return user;

}

async function insertUser(user) {

  const newUser = new UserModel(user);

  const result = await newUser.save();

  return result;

}

async function updateUser(query, update) {

  const result = await UserModel.updateOne(query, update);

  if (result.nModified === 0) throw new Error("User not found");

  return result;

}

async function deleteUser(query) {

  const result = await UserModel.deleteOne(query);

  if (result.deletedCount === 0) throw new Error("User not found");

  return result;

}

async function getUsers(query) {

  return await UserModel.find(query);

}

async function deleteUsers(ids) {

  if (!ids?.length) throw new Error("No IDs provided");

  const result = await UserModel.deleteMany({ \_id: { $in: ids } });

  if (result.deletedCount === 0) throw new Error("Users not found");

  return result;

}

module.exports = {

  connectToMongo,

  closeMongo,

  getUser,

  insertUser,

  updateUser,

  deleteUser,

  getUsers,

  deleteUsers,

};

# Висновок

В ході лабораторної роботи я модифікував попередній проект використавши ODM-бібліотеку **mongoose**. Виконавши лабораторну роботу я ознайомився з ODM-бібліотекою **mongoose**.