Одеський коледж комп’ютерних технології «Сервер»

Звіт з лабораторної роботи №9

«Основи роботи із Express 2»

Виконував :

Студент групи К21.1

Машков Володимир

Перевірив:  
Рудниченко Н.Д.

Одеса 2024

# Зміст

[Зміст 2](#_Toc160503319)

[Введения 3](#_Toc160503320)

[Теоретична частина 4](#_Toc160503321)

[Практична частина 6](#_Toc160503322)

[Завдання 1 6](#_Toc160503323)

[Завдання 2 8](#_Toc160503324)

[Завдання 3 9](#_Toc160503325)

[Завдання 4 10](#_Toc160503326)

[Завдання 5 12](#_Toc160503327)

[Завдання 6 14](#_Toc160503328)

[Завдання 7 16](#_Toc160503329)

[Підсумок 18](#_Toc160503330)

[Додаткові матеріали 19](#_Toc160503331)

# Введения

Вивчити та застосувати на практиці знання та навички про роботу із стрічками в JS. Вивчити та застосувати базових функції для роботи із стрічками. Регулярні вирази. Описання вмісту текстової стрічки, за допомогою спец. Правил із метою перевірки на правильність заповнення форми.

# Практична частина

# Завдання :

Створити БД Монга.

Результат виконанної роботи :

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Код програми :

Server.js :

const express = require('express');

const { MongoClient, ObjectID } = require('mongodb');

const app = express();

const port = 3000;

const uri = "mongodb+srv://mashkov:mashkov@cluster0.6znnou0.mongodb.net/";

async function connectToMongoDB() {

    const client = new MongoClient(uri);

    await client.connect();

    return client;

}

async function listDatabases(client) {

    try {

        const databasesList = await client.db().admin().listDatabases();

        console.log("Databases:");

        databasesList.databases.forEach(db => console.log(` - ${db.name}`));

        return databasesList;

    } catch (error) {

        console.error(error);

    }

}

app.get('/', async (req, res) => {

    try {

        const client = await connectToMongoDB();

        const databasesList = await listDatabases(client);

        res.send(databasesList);

    } catch (error) {

        console.error(error);

        res.sendStatus(500);

    }

});

app.get('/artists', async (req, res) => {

    try {

        const client = await connectToMongoDB();

        const db = client.db();

        const artists = await db.collection('artists').find().toArray();

        res.send(artists);

    } catch (error) {

        console.error(error);

        res.sendStatus(500);

    }

});

app.get('/artists/:id', async (req, res) => {

    try {

        const client = await connectToMongoDB();

        const db = client.db();

        const artist = await db.collection('artists').findOne({ \_id: ObjectID(req.params.id) });

        if (!artist) {

            return res.sendStatus(404);

        }

        res.send(artist);

    } catch (error) {

        console.error(error);

        res.sendStatus(500);

    }

});

app.post('/artists', function (req, res) {

    var artist = {

      name: req.body.name

    };

    db.collection('artists').insert(artist, (err, result) => {

      if (err) {

        console.log(err);

        return res.sendStatus(500);

      }

      res.send(artist);

    })

  })

  app.put('/artists/:id', function (req, res) {

    db.collection('artists').updateOne(

      { \_id: ObjectID(req.params.id) },

      { name: req.body.name },

      function (err, result) {

        if (err) {

          console.log(err);

          return res.sendStatus(500);

        }

        res.sendStatus(200);

      })

  })

  app.delete('/artists/:id', function (req, res) {

    db.collection('artists').deleteOne(

      { \_id: ObjectID(req.params.id) },

      function (err, result) {

        if (err) {

          console.log(err);

          return res.sendStatus(500);

        }

      res.sendStatus(200);

    })

  })

app.listen(port, () => {

    console.log(`Server is listening at http://localhost:${port}`);

});

# Підсумок

У результат роботи та виконання цієї роботи, ми навчилися працювати із Express.JS

# Додаткові матеріали

Мій GitHub :

<https://github.com/ActaniumTime/pr9>