**Muhammad Abdullah**

**SE(6A) | 19F-0916**

Web Engineering Lab

Lab 12: Pagination

**Task 1**

**NodeJS Code**

const express = require ('express');

const app = express();

users = [

    {id: 1, Name: 'User 1'},

    {id: 2, Name: 'User 2'},

    {id: 3, Name: 'User 3'},

    {id: 4, Name: 'User 4'},

    {id: 5, Name: 'User 5'},

    {id: 6, Name: 'User 6'},

    {id: 7, Name: 'User 7'},

    {id: 8, Name: 'User 8'},

    {id: 9, Name: 'User 9'},

    {id: 10, Name: 'User 10'},

]

app.get('/Task1', (req,res) => {

    res.json(users)

})

app.listen(8000);

**Installing REST Client**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Get called for Local Browser**

**Graphical user interface

Description automatically generated with medium confidence**

**Get called for Rest Client**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Task 2**

**NodeJS Code**

const express = require ('express');

const app = express();

users = [

    {id: 1, Name: 'User 1'},

    {id: 2, Name: 'User 2'},

    {id: 3, Name: 'User 3'},

    {id: 4, Name: 'User 4'},

    {id: 5, Name: 'User 5'},

    {id: 6, Name: 'User 6'},

    {id: 7, Name: 'User 7'},

    {id: 8, Name: 'User 8'},

    {id: 9, Name: 'User 9'},

    {id: 10, Name: 'User 10'},

    {id: 11, Name: 'User 11'},

    {id: 12, Name: 'User 12'},

    {id: 13, Name: 'User 13'},

    {id: 14, Name: 'User 14'},

    {id: 15, Name: 'User 15'},

    {id: 16, Name: 'User 16'},

    {id: 17, Name: 'User 17'},

    {id: 18, Name: 'User 18'},

    {id: 19, Name: 'User 19'},

    {id: 20, Name: 'User 20'},

    {id: 21, Name: 'User 21'},

    {id: 22, Name: 'User 22'},

    {id: 23, Name: 'User 23'},

    {id: 24, Name: 'User 24'},

    {id: 25, Name: 'User 25'},

    {id: 26, Name: 'User 26'},

]

app.get('/Task2', (req,res) => {

    const page = parseInt(req.query.page)

    const limit = parseInt(req.query.limit)

const startIndex = (page - 1) \* limit

const endIndex = page \* limit

const results = {}

if (endIndex < users.length){

    results.next = {

    page: page + 1,

    limit: limit

    }

}

if (startIndex > 0) {

    results.previous = {

    page: page - 1,

    limit: limit

    }

}

results.results = users.slice(startIndex,endIndex)

    res.json(results)

})

app.listen(8000);

**For Page 1 And Limit 6**

**A screenshot of a computer

Description automatically generated with medium confidence**

**For Page 2 And Limit 6**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Task 3**

**NodeJS Code**

const express = require ('express');

const app = express();

const users = [

    {id: 1, Name: 'User 1'},

    {id: 2, Name: 'User 2'},

    {id: 3, Name: 'User 3'},

    {id: 4, Name: 'User 4'},

    {id: 5, Name: 'User 5'},

    {id: 6, Name: 'User 6'},

    {id: 7, Name: 'User 7'},

    {id: 8, Name: 'User 8'},

    {id: 9, Name: 'User 9'},

    {id: 10, Name: 'User 10'},

]

const admin = [

    {id: 1, Name: 'admin 1'},

    {id: 2, Name: 'admin 2'},

    {id: 3, Name: 'admin 3'},

    {id: 4, Name: 'admin 4'},

    {id: 5, Name: 'admin 5'},

    {id: 6, Name: 'admin 6'},

    {id: 7, Name: 'admin 7'},

    {id: 8, Name: 'admin 8'},

    {id: 9, Name: 'admin 9'},

    {id: 10, Name: 'admin 10'},

]

const customer = [

    {id: 1, Name: 'customer 1'},

    {id: 2, Name: 'customer 2'},

    {id: 3, Name: 'customer 3'},

    {id: 4, Name: 'customer 4'},

    {id: 5, Name: 'customer 5'},

    {id: 6, Name: 'customer 6'},

    {id: 7, Name: 'customer 7'},

    {id: 8, Name: 'customer 8'},

    {id: 9, Name: 'customer 9'},

    {id: 10, Name: 'customer 10'},

]

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

    res.json(res.PaginatedResults)

})

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

    res.json(res.PaginatedResults)

})

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

    res.json(res.PaginatedResults)

})

function PaginatedResults(model){

    return (req,res,next) => {

        const page = parseInt(req.query.page)

        const limit = parseInt(req.query.limit)

        const startIndex = (page - 1) \* limit

        const endIndex = page \* limit

        const results = {}

    if (endIndex < model.length){

        results.next = {

        page: page + 1,

        limit: limit

        }

    }

    if (startIndex > 0) {

        results.previous = {

        page: page - 1,

        limit: limit

        }

    }

    results.results = model.slice(startIndex,endIndex)

    res.PaginatedResults = results

    next()

    }

}

app.listen(8000);

**For Admin**

**A screenshot of a computer

Description automatically generated with medium confidence**

**For Customer**

**A screenshot of a computer

Description automatically generated with medium confidence**

**For User**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Task 4**

**NodeJS Code**

const express = require ('express');

const app = express()

const T4schema = require ('E:/FAST-NU/6th Semester/Web Engineering/Lab/Assignments/9/task4model.js')

const mongoose = require ('mongoose');

const DBURI = 'mongodb+srv://abdullah1912:abcd1234@webproject.7w438.mongodb.net/WebLab12?retryWrites=true&w=majority'

mongoose.connect(DBURI)

.then (() => console.log('MongoDB Connected'))

.catch(err => console.log(err));

app.get('/user' ,function (req,res) {

    const T4Data = new T4schema({

        name: 'user 10'

    });

    T4Data.save()

    .then ((result)=>{

        res.send(result);

    })

    .catch((err)=> {

        console.log(err);

    })

})

app.get('/usersData', PaginatedResults(T4schema), (req,res) => {

    res.json(res.PaginatedResults)

})

function PaginatedResults(model){

    return async (req,res,next) => {

        const page = parseInt(req.query.page)

        const limit = parseInt(req.query.limit)

        const startIndex = (page - 1) \* limit

        const endIndex = page \* limit

        const results = {}

    if (endIndex < model.length){

        results.next = {

        page: page + 1,

        limit: limit

        }

    }

    if (startIndex > 0) {

        results.previous = {

        page: page - 1,

        limit: limit

        }

    }

    try{

        results.results = await model.find().limit(limit).skip(startIndex).exec()

        res.PaginatedResults = results

        next()

    } catch (e)  {

        res.status(500).json({mesaage: e.mesaage})

    }

    }

}

app.listen(8000);

**Model Code:**

const mongoose = require ('mongoose');

const Schema = mongoose.Schema;

const task4Schema = new Schema ({

    name: {

        type: String,

        required: true

    }

})

const task4data = mongoose.model('Task4',task4Schema)

module.exports = task4data;

**Data in DB**

**Graphical user interface, text, application

Description automatically generated**

**Data Shown in Rest Client**

**A screenshot of a computer

Description automatically generated with medium confidence**