 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №8**

з дисципліни «Веб-технології та веб-дизайн»

**Виконав:**

студент гр. БС-81

Сєров О. В.

**Перевірив:**

ас. Матвійчук О. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ – 2020

Завдання:

1. Створити динамічну галерею зображень (index.html).
2. Вибрати фреймворк для реалізації галереї (**Vue**, React, Angular)
3. Реалізувати галерею зображень як у лабораторній роботі #6 (включаючи режим preview)
4. В Preview режимі відображати картинки сусіди (по 1 з кожної сторони, на кільцевому списку. В галереї в кожен момент часу має бути не менше 3х зображень).
5. Додати функціонал з видалення зображень
6. Додати функціонал з розширення галереї (форму для додавання нових зображень)
7. Галерею реалізувати за паттерном Model-View-ViewModel

index.html:

<!DOCTYPE html>

<html lang="en">

<head>

  <title>ALXSRVV</title>

  <meta charset="utf-8">

  <link rel="shortcut icon" href="../src/favicon.ico" type="image/x-icon">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width,initial-scale=1.0">

</head>

<body>

  <div id="app"></div>

</body>

</html>

file-upload.fake.service.js:

function upload(formData) {

    const photos = formData.getAll('photos');

    const promises = photos.map((x) => getImage(x)

        .then(img => ({

            id: img,

            originalName: x.name,

            fileName: x.name,

            url: img

        })));

    return Promise.all(promises);

}

function getImage(file) {

    return new Promise((resolve, reject) => {

        const fReader = new FileReader();

        const img = document.createElement('img');

        fReader.onload = () => {

            img.src = fReader.result;

            resolve(getBase64Image(img));

        }

        fReader.readAsDataURL(file);

    })

}

function getBase64Image(img) {

    const canvas = document.createElement('canvas');

    canvas.width = img.width;

    canvas.height = img.height;

    const ctx = canvas.getContext('2d');

    ctx.drawImage(img, 0, 0);

    const dataURL = img.src;

    return dataURL;

}

export { upload }

file-upload.service.js:

import \* as axios from 'axios';

const BASE\_URL = 'http://localhost:3001';

function upload(formData) {

    const url = `${BASE\_URL}/photos/upload`;

    return axios.post(url, formData)

        // get data

        .then(x => x.data)

        // add url field

        .then(x => x.map(img => Object.assign({},

            img, { url: `${BASE\_URL}/images/${img.id}` })));

}

export { upload }

main.js:

import Vue from 'vue'

import App from './App.vue'

import router from './router'

Vue.config.productionTip = false

new Vue({

  router,

  render: h => h(App)

}).$mount('#app')

router.js:

import Vue from 'vue';

import Router from 'vue-router';

import Home from './views/Home.vue';

import Photo from './views/Photo.vue';

Vue.use(Router);

export default new Router({

  base: process.env.BASE\_URL,

  routes: [

    {

      path: '/',

      name: 'home',

      component: Home,

    },

    {

      path: '/photo/:id',

      name: 'photo',

      component: Photo,

    },

  ],

});

utils.js:

// utils to delay promise

function wait(ms) {

    return (x) => {

        return new Promise(resolve => setTimeout(() => resolve(x), ms));

    };

}

export { wait }