

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. Ігоря СІКОРСЬКОГО»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

# Кафедра системного програмування і спеціалізованих комп’ютерних систем

**Лабораторна робота №3**

з дисципліни

з дисципліни **WEB-дизайн**

на тему: “ Розробка Web-додатка засобами Javascript/VueJS”

Виконала: студентка ІII курсу

ФПМ групи КВ-01

Соліна Т.А.

Київ - 2023

**Мета**: ознайомитись із засобами фреймворка VueJS та навчитись створювати асинхронні запити до Web-сервера.

**Загальне завдання**: адаптувати програмний код ЛР№2 до вимог фреймворка VueJS та забезпечити завантаження необхідних даних з Web-сервера

**Інструменти розробки**: Javascript, VueJS, NodeJS, Express.

**Назва та опис роботи додатка: Сайт “Калькулятор”**

Головне завдання сайта: калькулятор

При розробці дизайну обрано веселкові кольори, які домінують на сторінці. Такі кольори обрано з розрахунком на тематику логотипу unicorn (единоріг), сайт буде використовуватись для роботи.

При вході на сайт користувач, якщо він не авторизований, то потратляє не сторінку реєстрації, звідти може перейти на сторінку входу, якщо він вже має свою сторінку.

Після авторизації користувач потрапляє на сторінку профіля, де він може побачити дані про свій обліковий запис.

Потім, за допомогою статичного сайдбара він може потрапити на сторінку калькулятора і обрати вид калькулятора, яким він хоче користуватися.

Sidebar вміщує в собі фото користувача, яке поки що є фото за замовчуванням, при натисненні на яке користувач потрапляє на сторінку користувача. Також присутні кнопки “About”, яка переправляє користувача на сторінку з інформацією про калькулятори. Кнопка “Log out” (кнорпка виходу) переносить користувача на сторінку входу.

Navbar вміщує в собі логотип додатка а також кнопку про додаток.

Опис основної функціональності: калькулятор

Калькулятор може працювати в двох режимах роботи. Звичайному та в програмованому. Зміна режимів може відбуватися за допомогою сайдбара.

У звичайному режимі калькулятор може виконувати обчислення шляхом додавання, віднімання, множення або ділення результуючого (значення “Result”) значення та введеного (значення “Expression”). Вибравши функцію відсотка можна визначити введений відсоток від результуючого значення. Значення логарифмів (“log” - десятковий логарифм, “ln” - натуральний логарифм), факторіала та кореня квадратного обчислюється наступним чином. Виконується зазначена раніше операція над числом в полі “Expression” та виконується над ним і результатом операція “Operation”.

У програмованому режимі можна виконувати обрахунки всі ті ж самі обрахунки, що і в звичайному тільки в різних системах числення в: двій, вісімковій, десятковій та шістнадцятковій. Можна також виконувати переведення між різними системами числення, для чого необхідно просто поміняти значення “DEC” (за замовчуванням) на іншу потрібну систему числення. Також є можливість вести обчислення між різними системами числення. Наприклад, до числа в двійковій системі числення додати число в десятковій, вісімковій чи шістнадцятковій системі числення.

**Скріни статичних веб сторінок додатка:**

Рис.1 - сторінка реєстрації

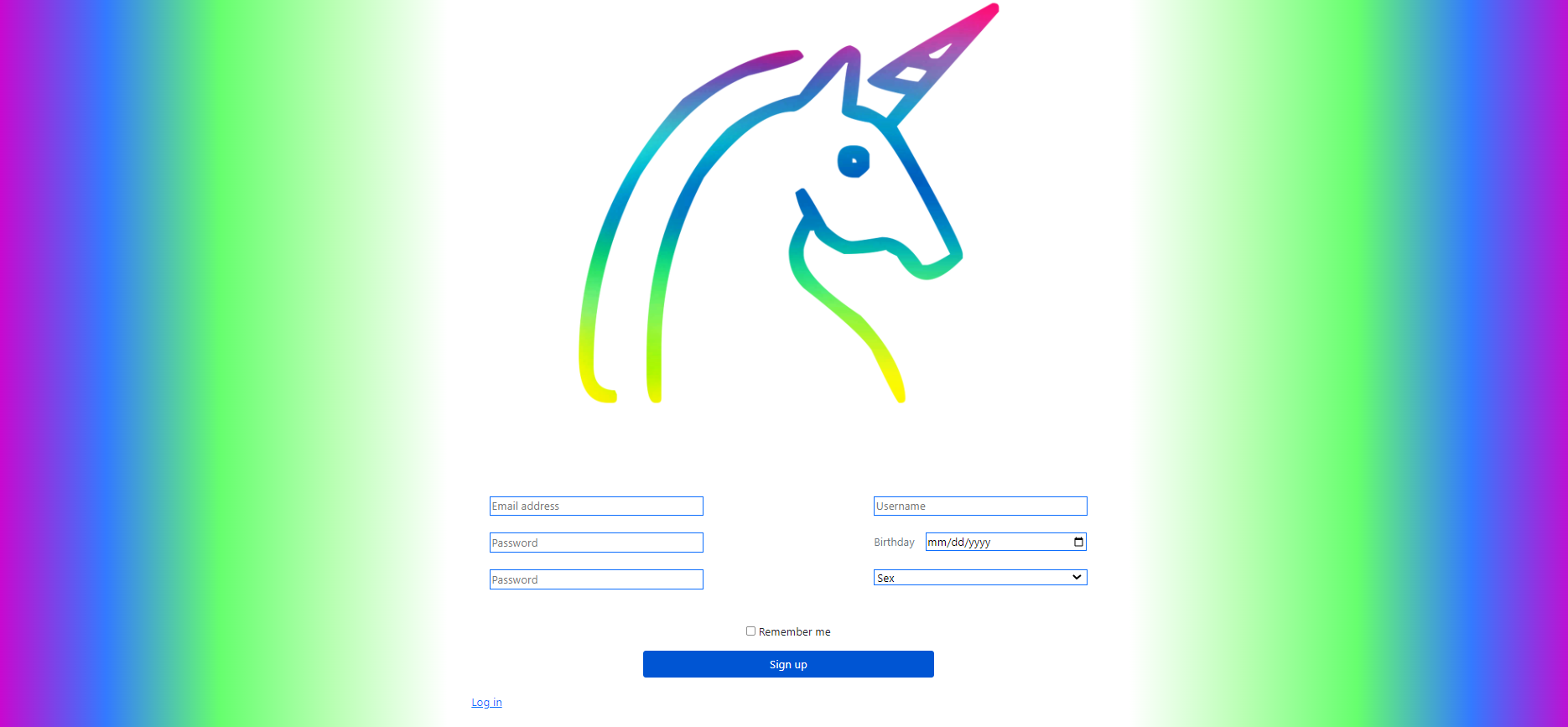


Рис.2 - сторінка входу:

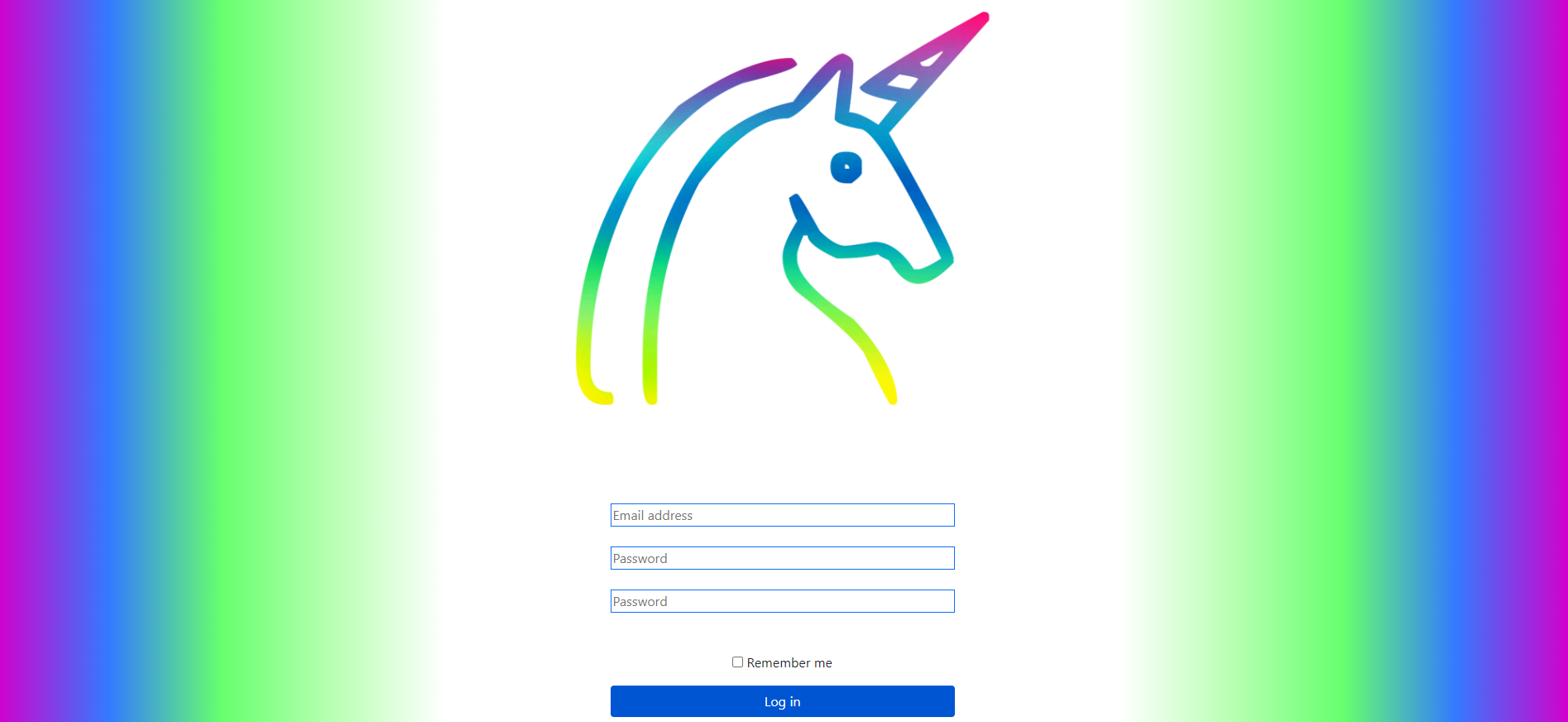


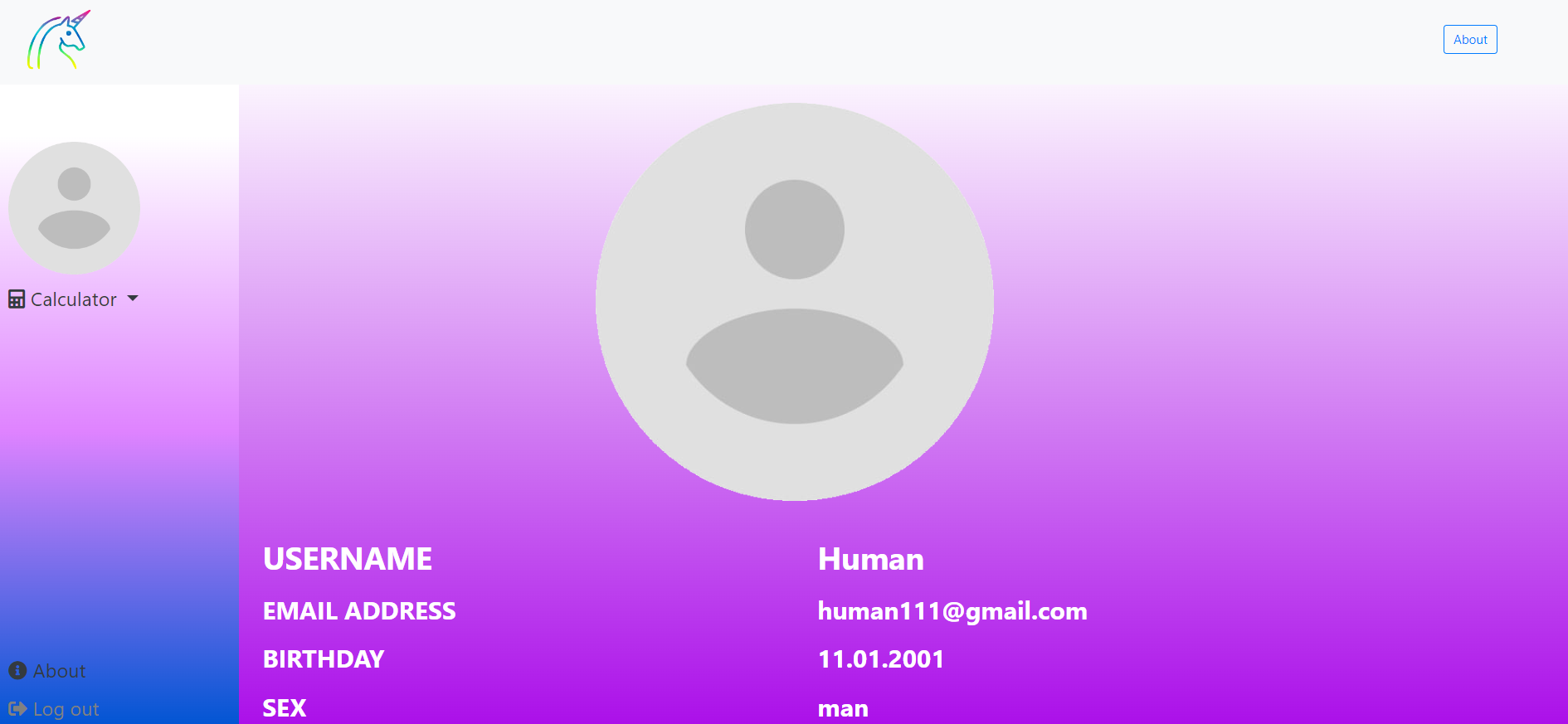
Рис.3 - сторінка профіля

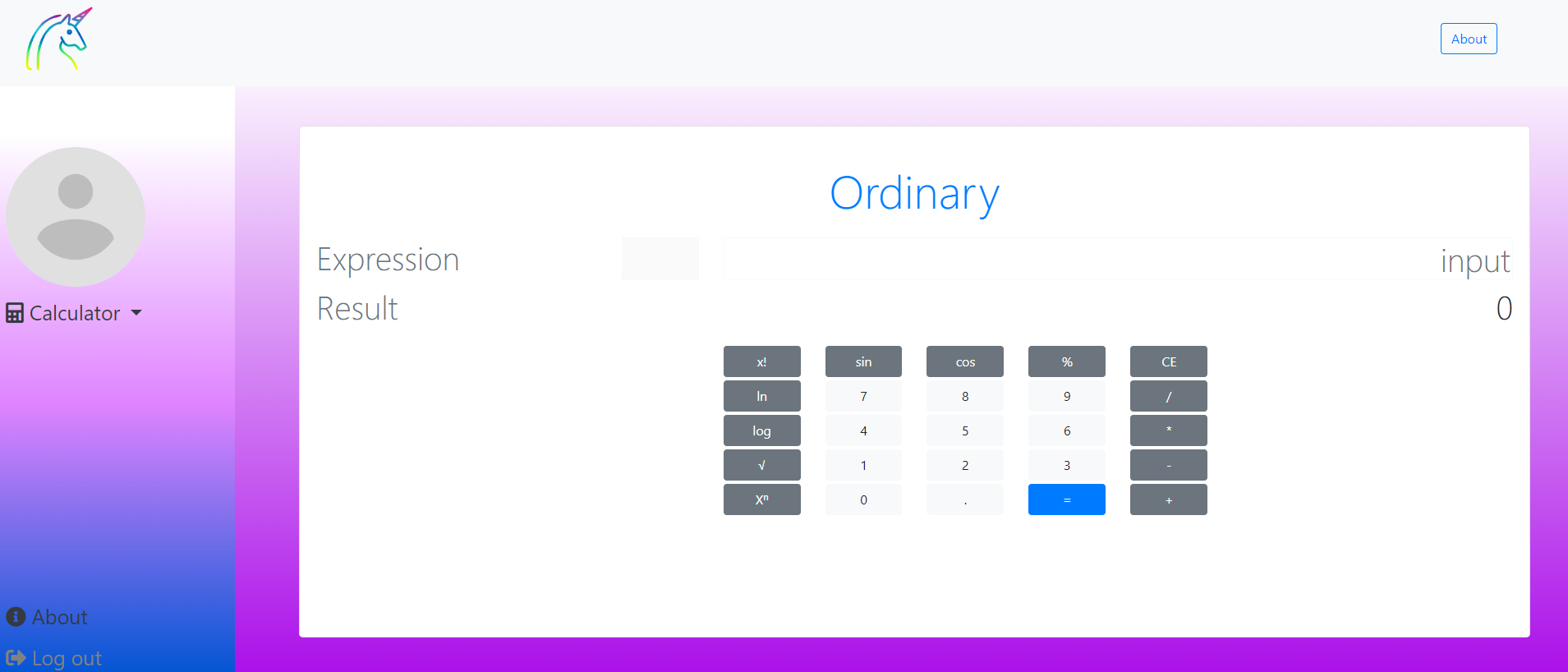
Рис.4 - сторінка звичайного калькулятора

Рис.5 - сторінка програмістського калькулятора

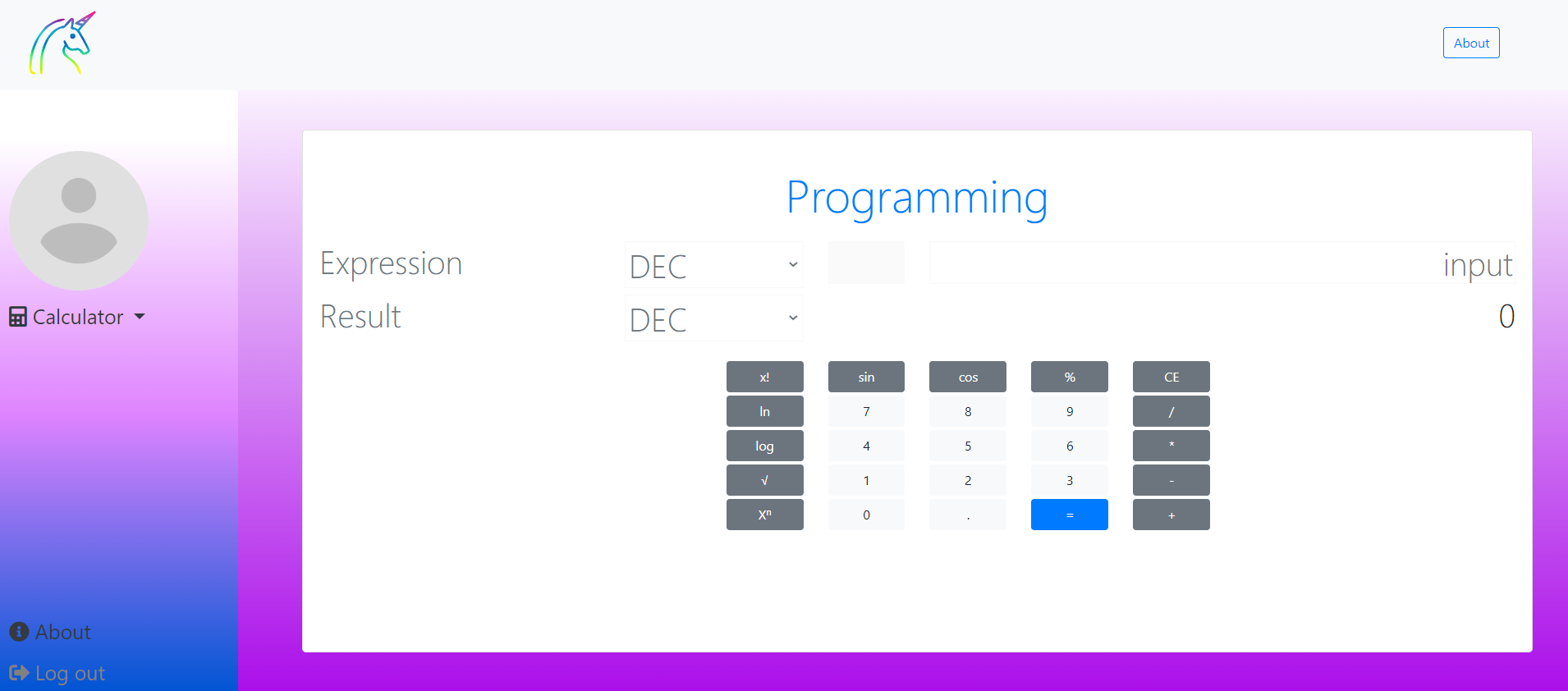
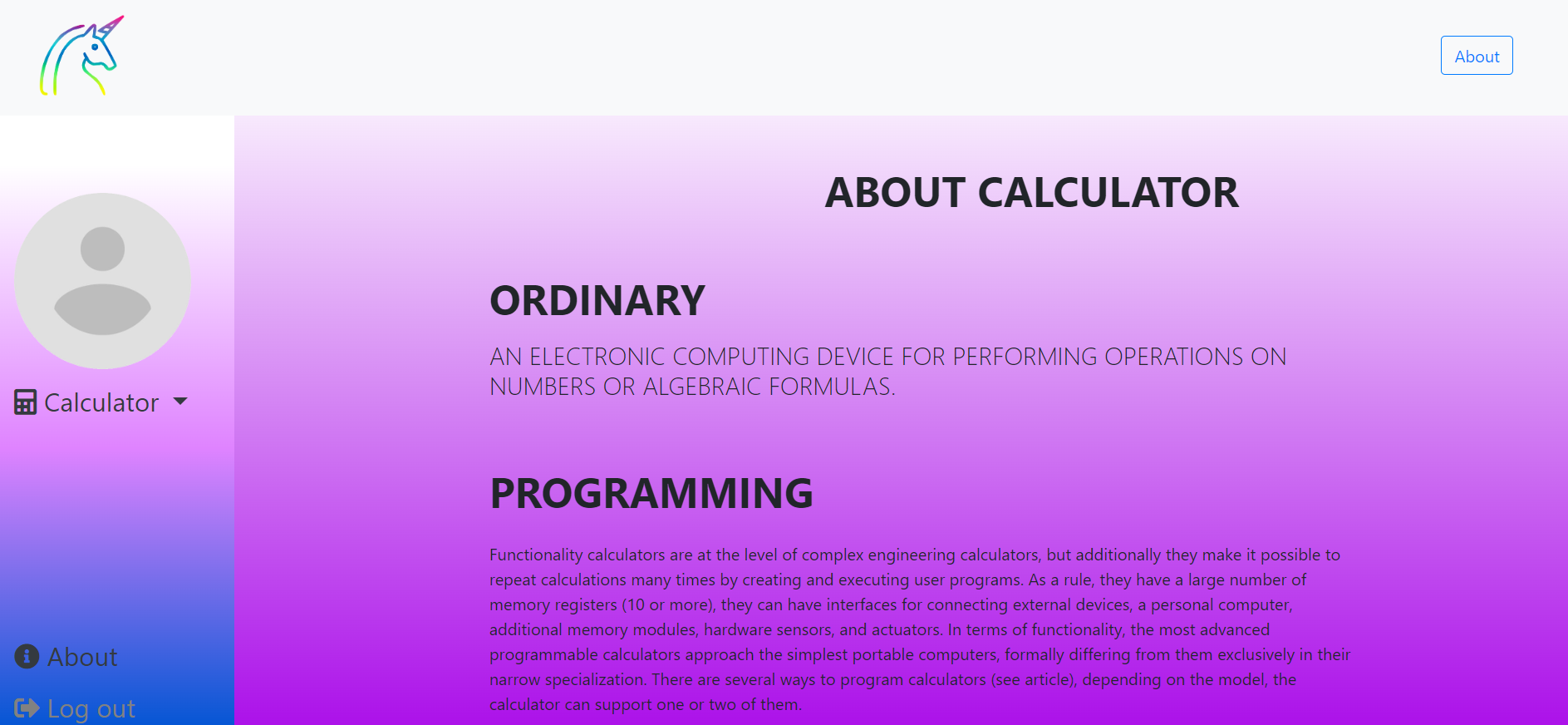


Рис.6 - сторінка про додаток (про калькулятор) 

**Текст файлів JS робочої сторінки**

<template>

    <div class="background">

        <div class="row">

            <div class="col-md-2" />

            <div class="col-md-10">

                <div class="card rounded-3 mt-5 mb-5 mx-5">

                    <div class="card-body mb-4">

                        <p class="card-text text-primary text-center display-4 mt-4">Ordinary</p>

                        <div class="conteiner">

                            <div class="row mt-4">

                                <div class="col-md-3 text-secondary display-6">Expression</div>

                                <div class="col-md-9">

                                    <input type="text" class="w-100 border border-light text-end display-6 border-invis-focus"

                                        placeholder="input" v-model="expression" v-on:input="inputExpression">

                                </div>

                            </div>

                            <div class="row mt-2 mb-4">

                                <div class="col-md-3 text-secondary display-6">Result</div>

                                <div class="col-md-9">

                                    <div class="float-end display-6">

                                        {{result}}

                                    </div>

                                </div>

                            </div>

                            <CalculatorKeyboard v-bind:result.sync="result"

                                v-bind:notationResult.sync="notationResult"

                                v-bind:expression.sync="expression"

                                v-bind:notationExpression.sync="notationExpression"

                                ref="calcKey"/>

                        </div>

                    </div>

                </div>

            </div>

        </div>

    </div>

</template>

<script>

import CalculatorKeyboard from '@/components/CalculatorKeyboard'

export default {

    beforeRouteEnter(to, from, next) {

        if (from.meta.auth === true){

            next();

        }

        else {

            next('/log-in');

        }

        to.meta.auth = from.meta.auth;

    },

    components: {

        CalculatorKeyboard

    },

    data() {

        return {

            expression : '',

            result : '0',

            notationExpression : 10,

            notationResult : 10

        }

    },

    methods: {

        inputExpression (event) {

            switch (event.data) {

                case '+':

                    this.$refs.calcKey.handlerPlus();

                    break;

                case '-':

                    this.$refs.calcKey.handlerMinus();

                    break;

                case '\*':

                    this.$refs.calcKey.handlerMultiply();

                    break;

                case '/':

                    this.$refs.calcKey.handlerDivision();

                    break;

            }

        },

    },

}

</script>

<style>

.background {

    background: linear-gradient(white 8%, #ab0fe9);

    height: 100vh;

}

</style>