

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ІНФОРМАТИКИ ТА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**ЗВІТ**

лабораторної роботи №4

з курсу «Безпека програмного забезпечення»

**Тема:** «Засвоювання базових навичок OAuth2 авторизаційного протокола»

|  |  |
| --- | --- |
| **Виконав:**  студент групи - ІТ-94  Тріпак Крістіна  Дата здачі  Захищено з балом | **Перевірив:**  викладач кафедри ІПІ  Волокита Артем Миколайович |

Київ 2023

# Завдання

* + Використовуючи наведені налаштування з лабораторної роботи 2 - 3 та приведених запитів модифікувати аплікейшен <https://github.com/Kreolwolf1/auth_examples/tree/main/token_auth>
  + Використовуючи перевірку юзера та отримання токена з auth0 (**password grant type).** Надати код модифікованного аплікейшена.
* **Для отримання додаткового балу:** додатково розшири аплікайшен створенням юзера та перевіркою життя токена (у разі близького завершення – оновити токен використовуючи **refresh-token** grant type)

Посилання на проєкт:

# Хід роботи

* + Використовуючи наведені налаштування з лабораторної роботи 2 - 3 та приведених запитів модифікувати аплікейшен <https://github.com/Kreolwolf1/auth_examples/tree/main/token_auth>. Використовуючи перевірку юзера та отримання токена з auth0 (**password grant type).** Надати код модифікованного аплікейшена.

app.post('/api/login', (req, res) => {

  const { login, password } = req.body;

  var login\_options = {

    method: 'POST',

    url: `https://${process.env.DOMAIN}/oauth/token`,

    headers: { 'content-type': 'application/x-www-form-urlencoded' },

    form:

    {

        audience: process.env.AUDIENCE,

        client\_id: process.env.CLIENT\_ID,

        client\_secret: process.env.CLIENT\_SECRET,

        username: login,

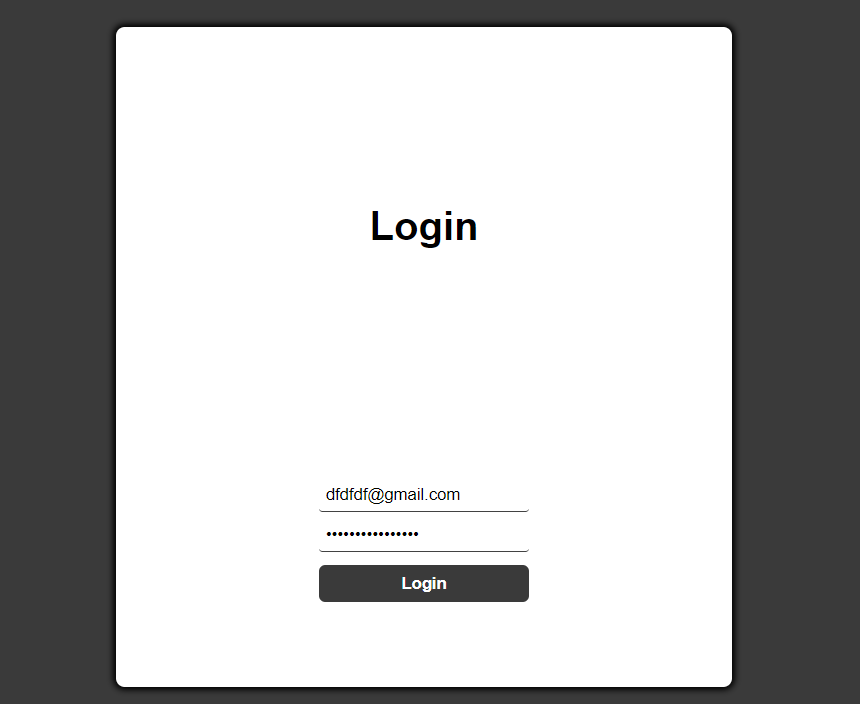
        password: password,

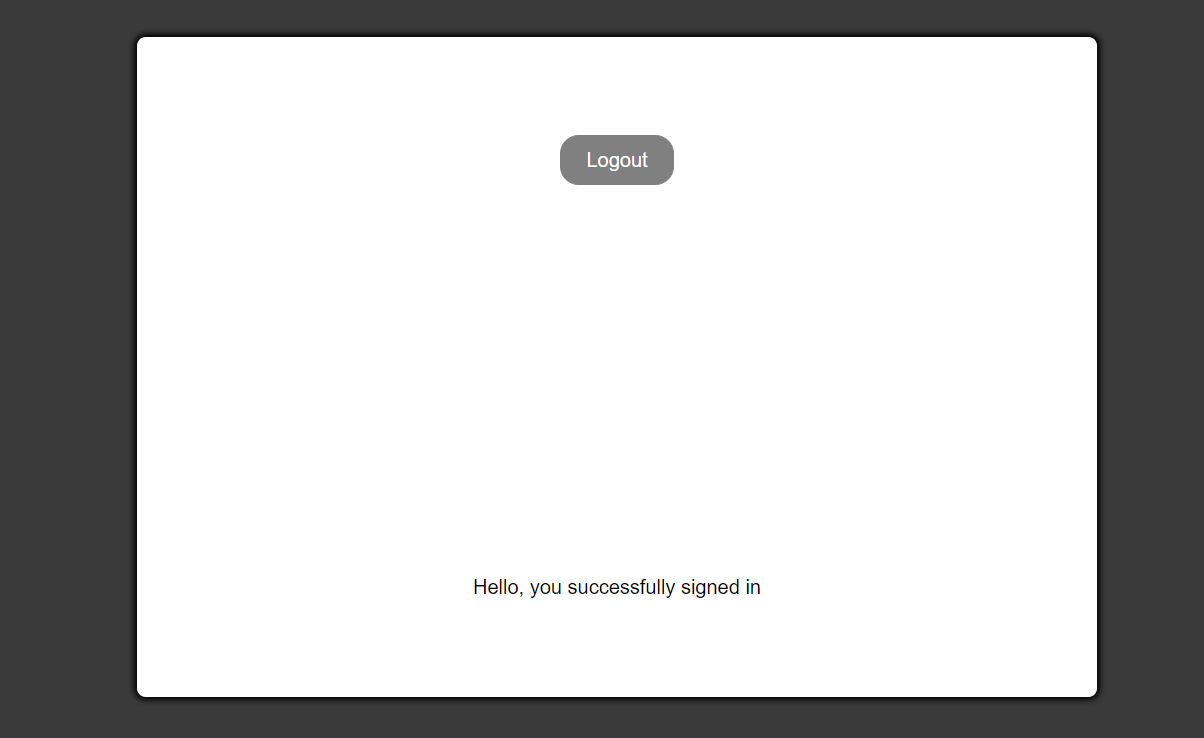
        scope: 'offline\_access',

        grant\_type: 'password'

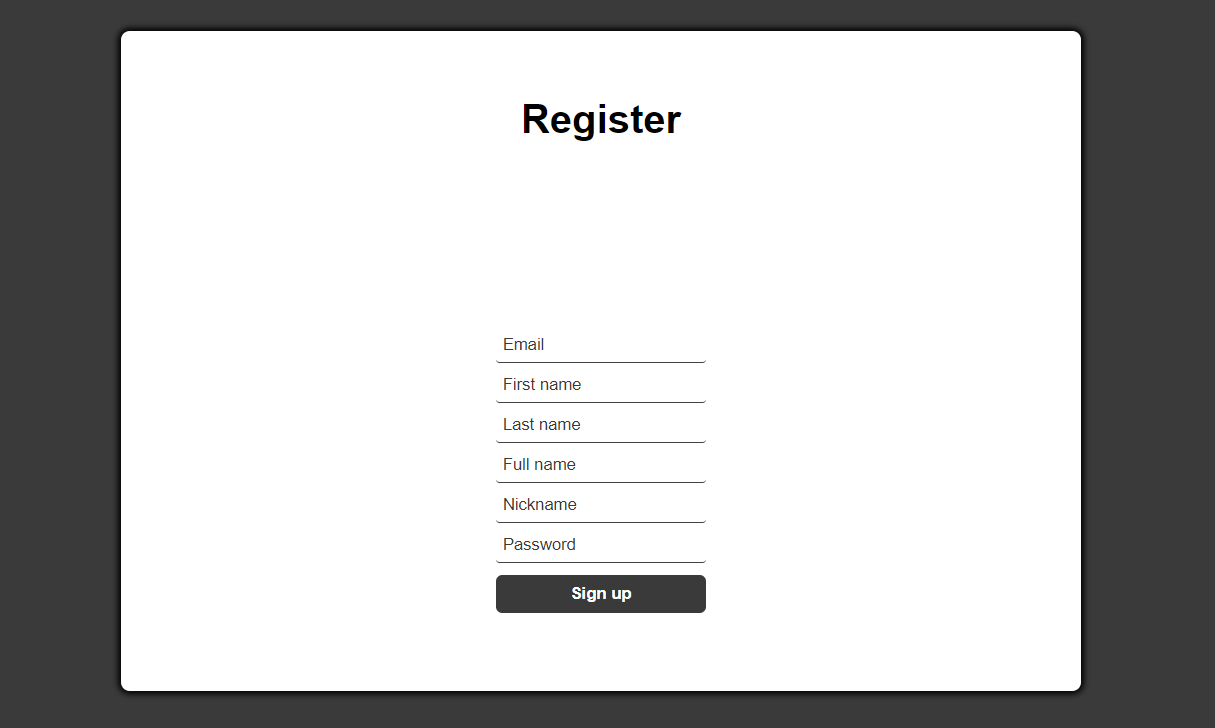
    }

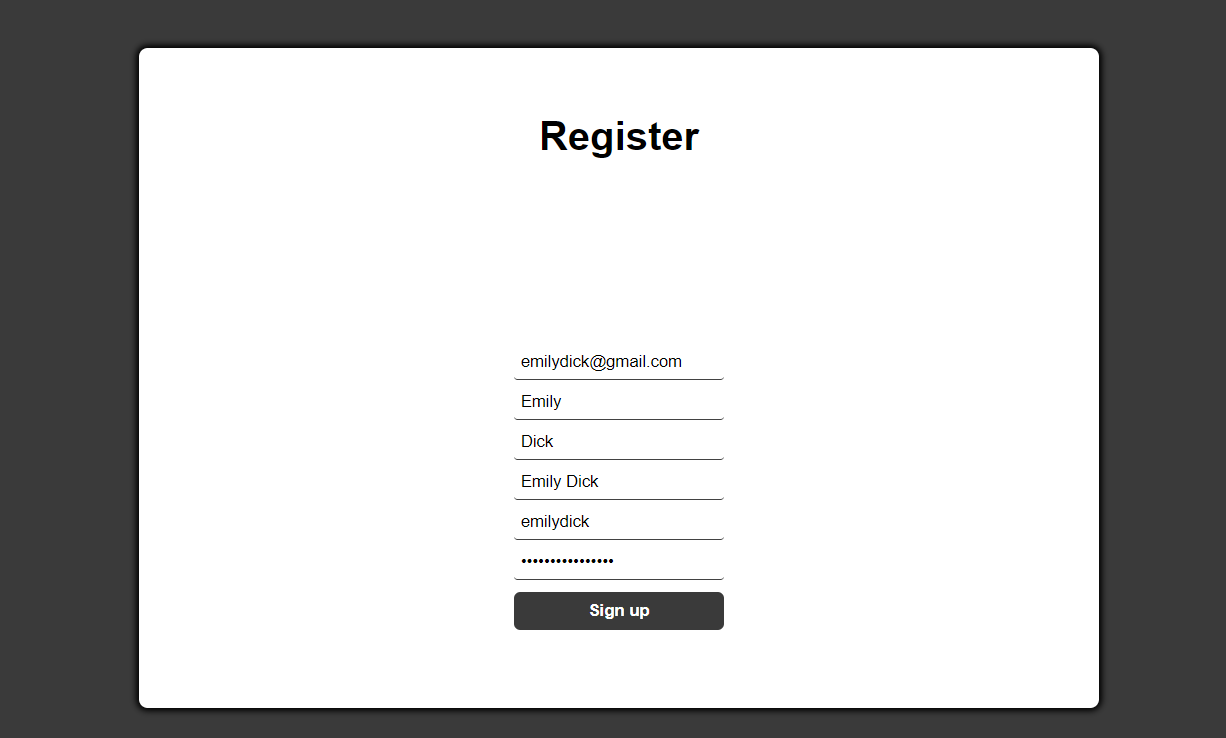
};



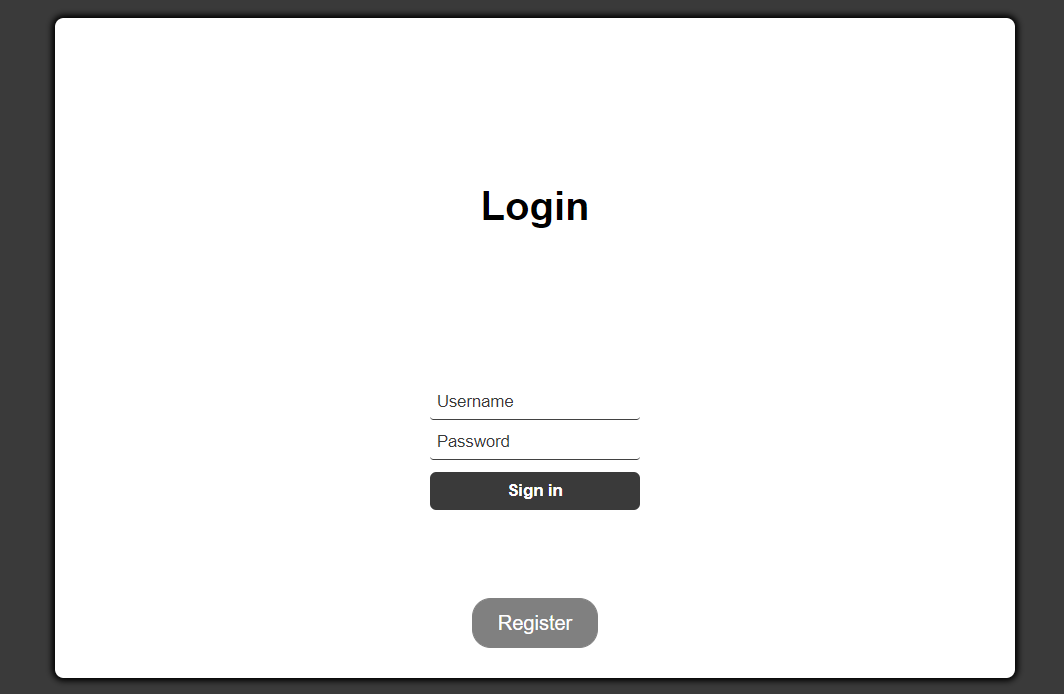


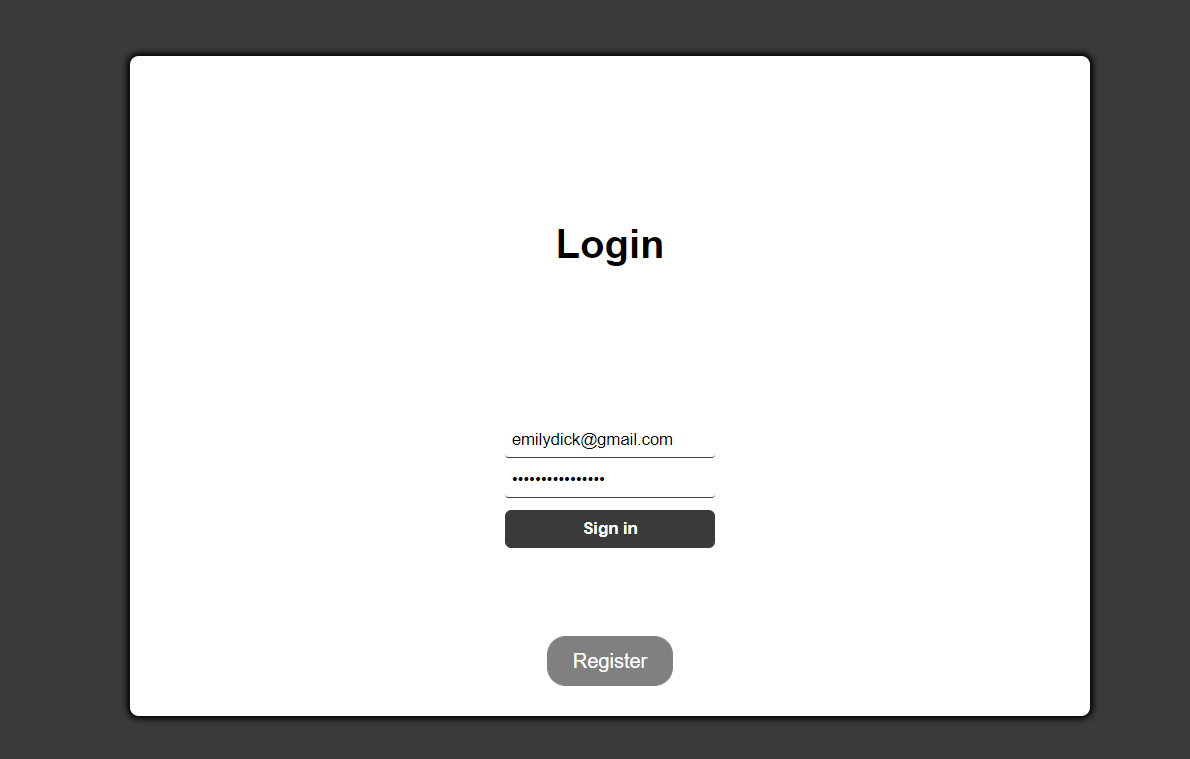
* **Для отримання додаткового балу:** додатково розшири аплікайшен створенням юзера та перевіркою життя токена (у разі близького завершення – оновити токен використовуючи **refresh-token** grant type)

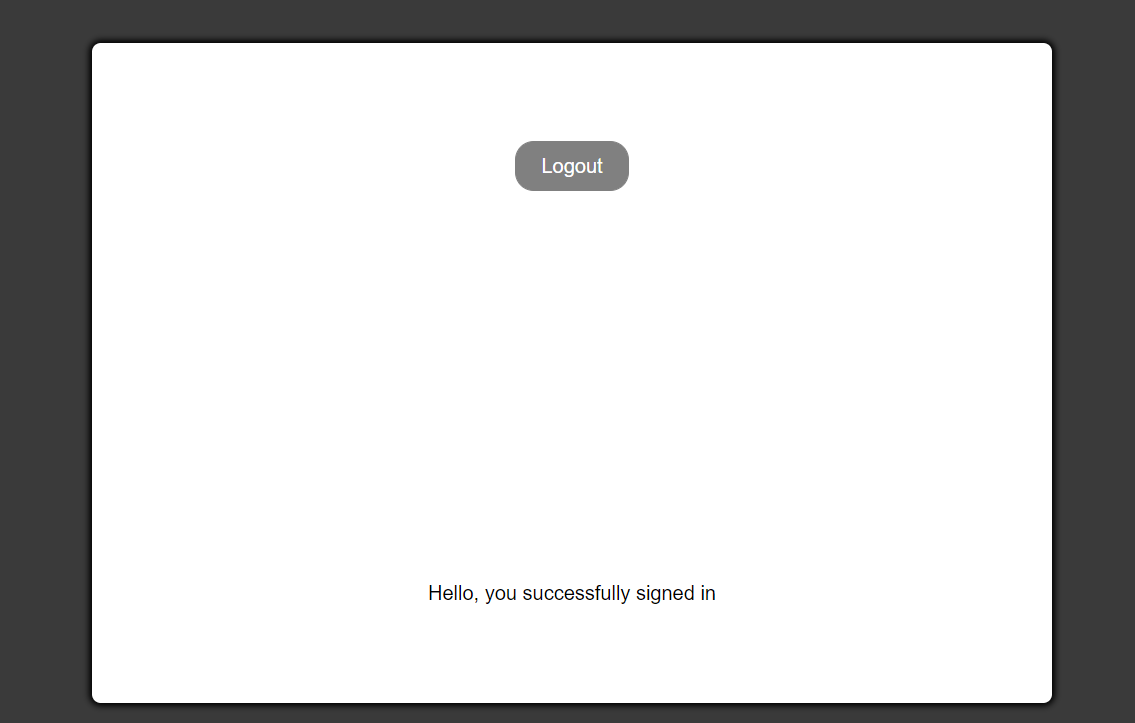
****

****

****

****

****

****

****

function post\_user(res, email, first\_name, last\_name, full\_name, nickname, password) {

  var post\_user\_options = {

      method: 'POST',

      url: `https://${process.env.DOMAIN}/api/v2/users`,

      headers: {

          'content-type': 'x-www-form-urlencoded',

          'Authorization': `Bearer ${process.env.ACCESS\_TOKEN}`

      },

      form: {

          "email": email,

          "given\_name": first\_name,

          "family\_name": last\_name,

          "name": full\_name,

          "nickname": nickname,

          "connection": "Username-Password-Authentication",

          "password": password

      }

  };

  request(post\_user\_options, function (error, response, body) {

    console.log('Статус створення юзера: '+ response.statusMessage)

    res.status(response.statusCode).send();

  });

}

app.use((req, res, next) => {

  let getToken = req.get(SESSION\_KEY);

  if (getToken) {

    let token = JSON.parse(getToken)

    const date = new Date();

    date.setSeconds(date.getSeconds() + 5)

    if (token.expires < date) {

      var refresh\_token\_options = {

        method: 'POST',

        url: `https://${process.env.DOMAIN}/oauth/token`,

        headers: {

            'content-type': 'x-www-form-urlencoded',

            'Authorization': `Bearer ${process.env.USER\_ACCESS\_TOKEN}`

        },

        form: {

            audience: process.env.AUDIENCE,

            client\_id: process.env.CLIENT\_ID,

            client\_secret: process.env.CLIENT\_SECRET,

            refresh\_token: process.env.USER\_REFRESH\_TOKEN,

            grant\_type: 'refresh\_token'

        }

      };

      request(refresh\_token\_options, function (error, response, body) {

          if (error) throw new Error(error);

          const information = JSON.parse(body);

          process.env.USER\_ACCESS\_TOKEN = information.access\_token

          console.log(`Refreshed Token: ${information.access\_token}`);

          token = JSON.stringify({ token: information.access\_token, expires: information.expires\_in})

      });

    }

    req.token = token;

  }

  next();

});

**Висновок**

На цій лабораторній роботі ми засвоїли базові навички OAuth2 авторизаційного протокола. Навчилися використовувати токен refresh-token і password grant type. Також зробили перевірку життя точену, щоб у разі близького завершення – оновлювати токен.