

Міністерство освіти і науки України

Національний технічний університет України „КПІ

імені Ігоря Сікорського ”

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

**ЗВІТ**

лабораторної работи №1

курсу «Безпека програмного забезпечення»

Тема: «Основні методи авторизації»»

Перевірив: Виконав:

Курченко О. А. Група ІП-02

Трофимов Данило

Київ 2023

Завдання:

Викачати репозиторій з лекціями https://github.com/Kreolwolf1/auth\_examples

Запустити кожен з 3 аплікейшенів та зробити скріншити запитів до серверу.

Для отримання додаткового балу: модифікувати token\_auth аплікейшен змінивши

токен на JWT.

Хід роботи:

1. Для репозиторію basic\_auth було запущено index.js. За адресою localhost:3000 було показано вікно для введення логіну та паролю, яке зображено на рисунку 1.

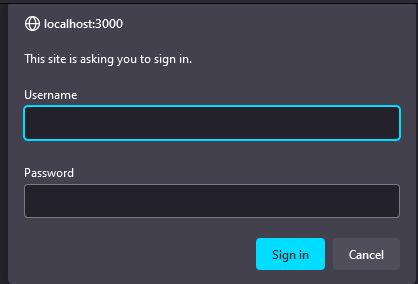


Рисунок 1 – Вікно для введення даних авторизації

У ході аналізу коду, було помічено, що дані авторизації передаються в base64 закодованому вигляді, після чого розкодовуються і порівнюються з логіном та паролем, що наявні у коді без кодування, що не є безпечним. Також при кожному заході на сторінку потрібно авторизовуватись спочтку.

1. Для репозиторію forms\_auth також було запущено index.js. В цьому методі, на відміну від попереднього, пристунє зберігання сесії у файлі на сервері та використання cookie. Таким чином не доведеться вводити логін та пароль при кожному оновлені сторінки. Також у цьому методі реалізована функція logout.

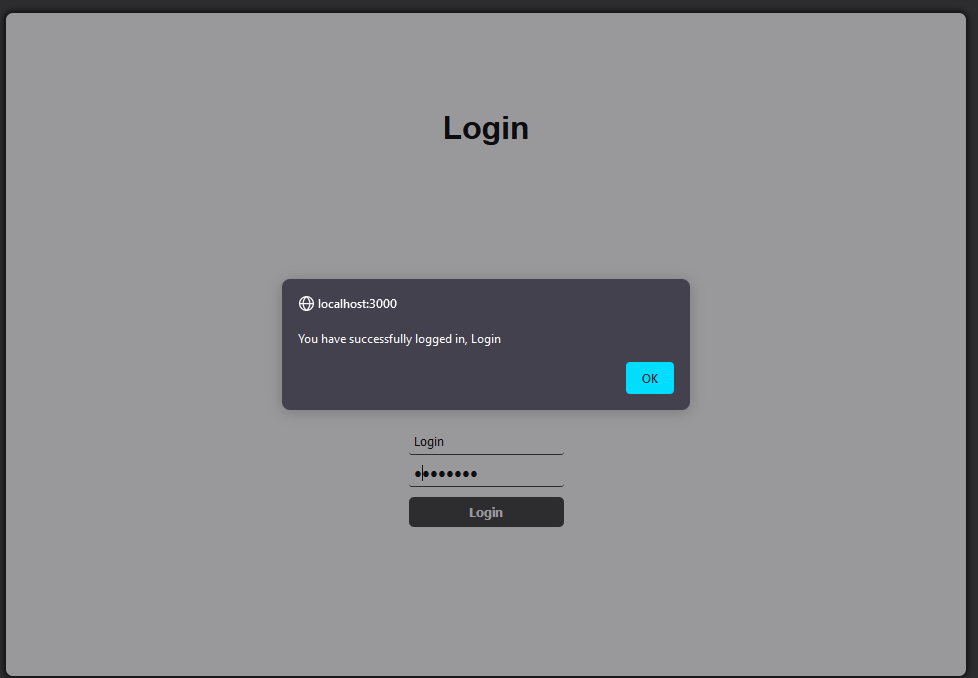


Рисунок 2 – Вигляд сторінки при успішній авторизації.

1. Для репозиторію token\_auth також було запущено index.js. Відкрилася схожа з попередньою сторінка. При аналізі коду було помічено, що у якості токена передається id сесії з файлу з попередньої роботи, а сам метод авторизації майже повністю повторює попередній. Тому було розроблено функціонал для того, аби уникнути роботу з сесіями та використовувати JsonWebToken.

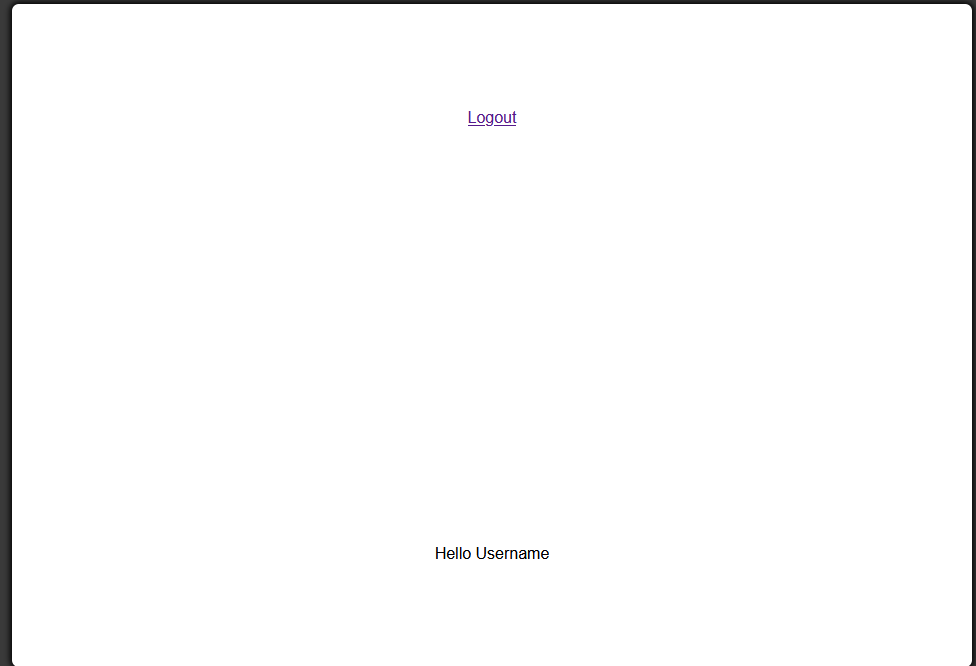


Рисунок 3 – Вигляд сторінки при успішній авторизації.

Лістинг коду файла token\_auth/index.js.

const uuid = require('uuid');

const express = require('express');

const onFinished = require('on-finished');

const bodyParser = require('body-parser');

const path = require('path');

const port = 3000;

const fs = require('fs');

const jwt = require('jsonwebtoken');

const app = express();

app.use(bodyParser.json());

app.use(bodyParser.urlencoded({ extended: true }));

const JWT\_SECRET = 'secret';

const users = [

    {

        login: 'Login',

        password: 'Password',

        username: 'Username',

    },

    {

        login: 'Login1',

        password: 'Password1',

        username: 'Username1',

    }

]

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

    res.json({

        username: req.user.username,

        logout: 'http://localhost:3000/logout'

    });

})

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

    res.redirect('/');

});

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

    const { login, password } = req.body;

    const user = users.find((user) => {

        return user.login === login && user.password === password;

    });

    if (user) {

        const token = jwt.sign({ username: user.username, login: user.login }, JWT\_SECRET);

        res.json({ token });

    } else {

        res.status(401).send('Unauthorized');

    }

});

app.listen(port, () => {

    console.log(`Example app listening on port ${port}`)

})

function verifyToken(req, res, next) {

    const token = req.header('Authorization');

    if (!token) {

        return res.sendFile(path.join(\_\_dirname + '/index.html'));

    }

    jwt.verify(token, JWT\_SECRET, (err, decoded) => {

        if (err) {

            return res.sendFile(path.join(\_\_dirname + '/index.html'));

        }

        req.user = decoded;

        next();

    });

}

Висновки: У ході виконання лабораторної роботи було ознайомдено з основними методамт авторизації на веб-сервісах. Було порівняно базову авторизацію, авторизацію, що будується на сесіях та на токенах. Також було розроблено власне програмне забезпечення для авторизації з використанням JWT токену.