A white building with a black background

Description automatically generated

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:

Курченко О.А.

Виконав:

Група ІП-02, Сергєнко Яків

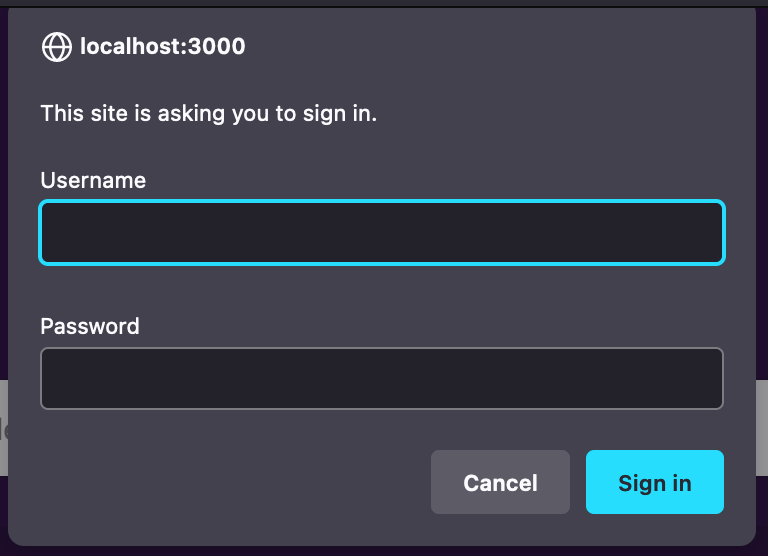
Київ 2023

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:

Завдання 1.

В репозиторії basic\_auth було запущено файл index.js. За адресою localhost:3000 було показано вікно з формою для введення логіну та поаролю, яке зображено на рисунку нижче.



При аналізу коду застосунку, було помічено, що при введенні логіну «DateArt» та паролю «2408» можна залогінитись, тоді з’явиться нова сторінка, яку можна побачити на рисунку нижче.

A black and white logo

Description automatically generated with medium confidence

При авторизації дані передаються в base64 кодуванні, після чого розкодовуються і порівнюються з логіном та паролем, що наявні в коді без кодування, що не є безпечним підходом.

Завдання 2.

Для репозиторію forms\_auth також було запущено файл index.js На відміну від попереднього завдання, тут присутнє зберігання сесії у файлі на сервері та використовуються cookies, таким чином не потрібно кожного разу вводити логін та пароль при оновленні сторінки, також тут пристуній роут logout, що дозволяє вийти. На рисунку нижче можна побачити сторінку при успішній авторизації.

A screenshot of a login box

Description automatically generated

Завдання 3.

Для репозиторію token\_auth також було запущено файл index.js. При аналізу коду було помічено, що у вигляді токену передається id сесії з файлу, з попередньої роботи, також сам метод майже повністю повторює попередній. Тому було розроблено функціонал для того, щоб уникнути роботи з сесіями, а замість них використовувати JsonWebToken. На рисунку нижче можна побачити сторінку при успішному логіні.

A white background with purple text

Description automatically generated

Лістинг коду файлу token\_auth/index.js з використанням JsonWebToken:

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 = 'JWT\_SECRET';

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

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

if (token) {

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

if (err) {

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

} else {

return res.json({

username: decoded.username,

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

})

}

});

} else {

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

}

})

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

res.redirect('/');

});

const users = [

{

login: 'Login',

password: 'Password',

username: 'Username',

},

{

login: 'Login1',

password: 'Password1',

username: 'Username1',

}

]

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) {

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

}

res.status(401).send();

});

app.listen(port, () => {

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

})

Висновки.

У ході виконання даної лабораторної роботи я ознайомився з основними методами атворизації на веб-сервісах, було порівняно базову, авторизацію з тими, що будується на сесіях та токенах. Також, в рамках додаткового завдання було модифіковано код третього завдання в якому для авторизації використовується JWT токен.