



Міністерство освіти і науки України

Національний технічний університет України «КПІ

імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

ЗВІТ

лабораторної роботи №1

з курсу «Безпека програмного забезпечення»

Перевірив:

Курченко О. А.

Виконав:

Студент Гр. ІІІ-01

Пашковський Є. С.

Київ 2023

Завдання

Викачати

репозиторій

з

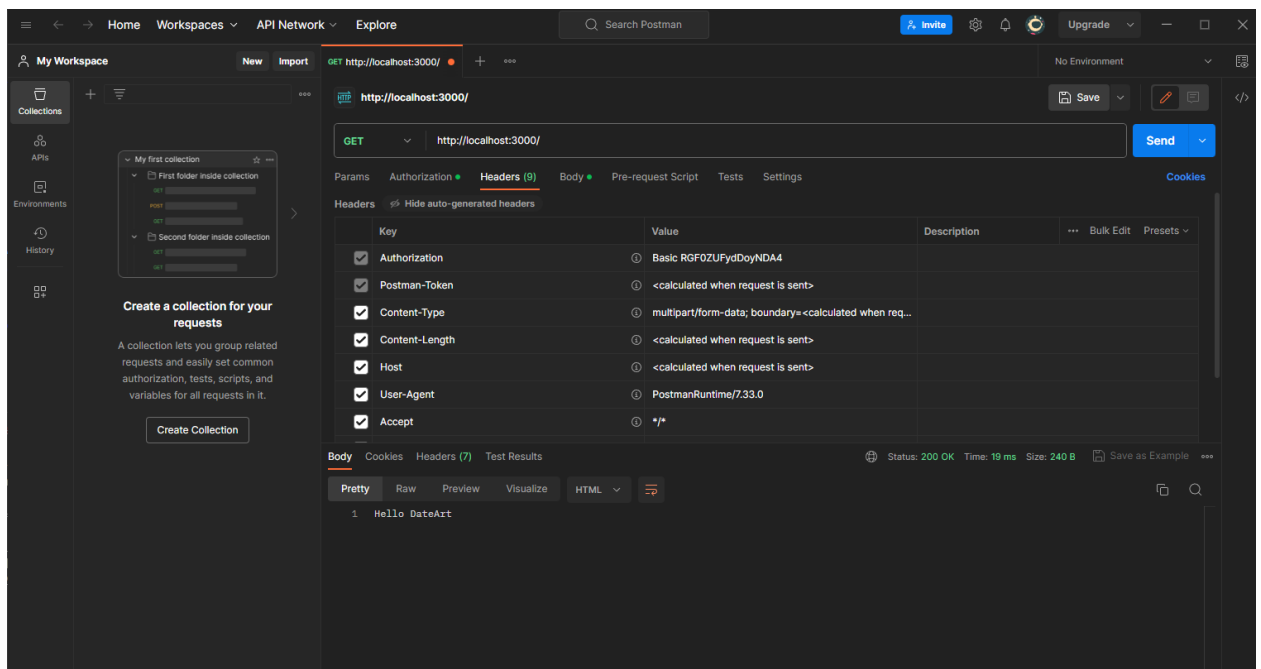
лекціями

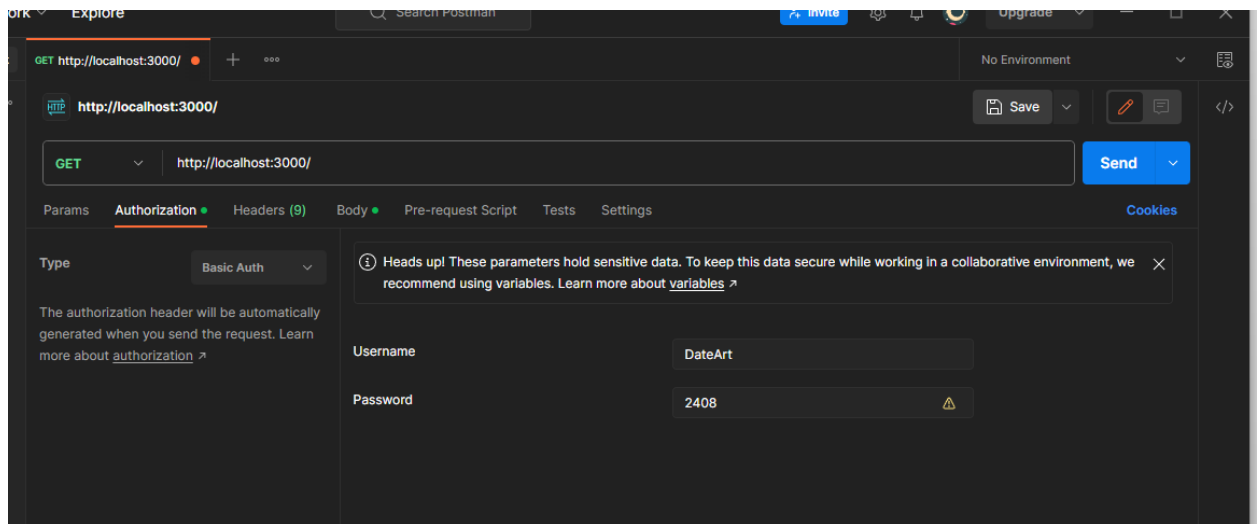
https://github.com/Kreolwolf1/auth_examples. Запустити кожен з 3 апікейшенів та зробити скріншоти запитів до серверу. Для отримання додаткового балу: модифікувати token_auth апікейшен змінивши токен на JWT.

Хід роботи

1. Запуск та тестування першого додатку

```
PS E:\Projects\security-labs> node .\lab1\basic_auth\
Example app listening on port 3000
```



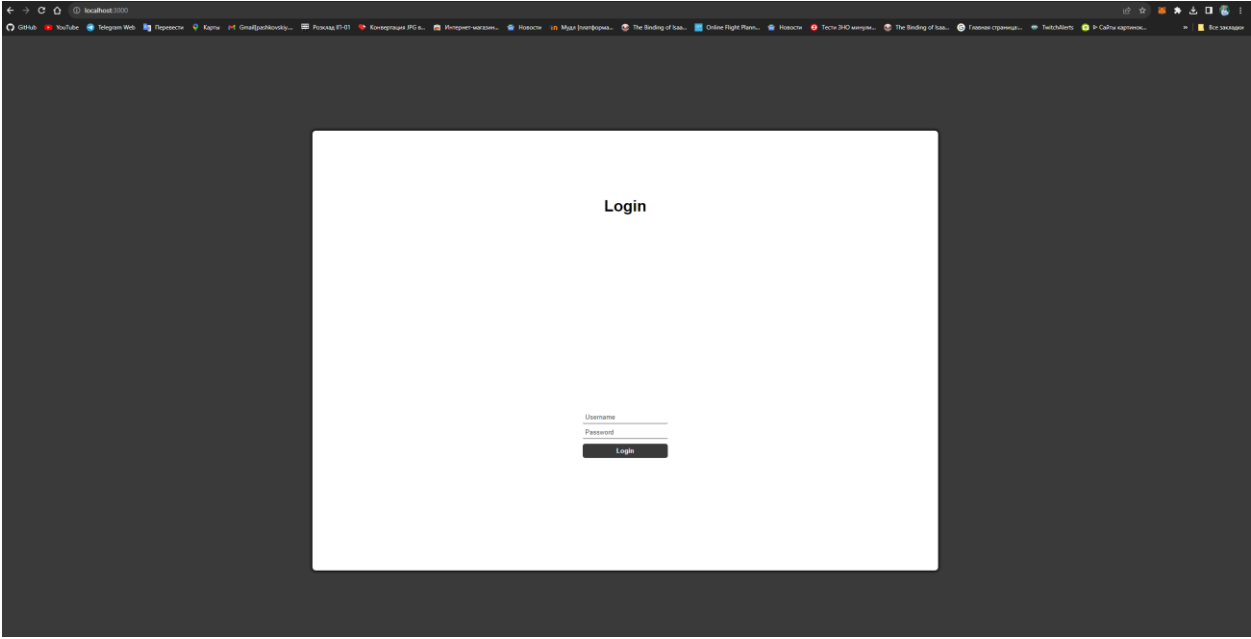


```
PS E:\Projects\security-labs> node .\lab1\basic_auth\  
Example app listening on port 3000  
  
=====
```

```
authorizationHeader Basic RGF0ZUFydDoyNDA4  
decodedAuthorizationHeader DateArt:2408  
Login/Password DateArt 2408
```

2. Запуск та тестування другого додатку

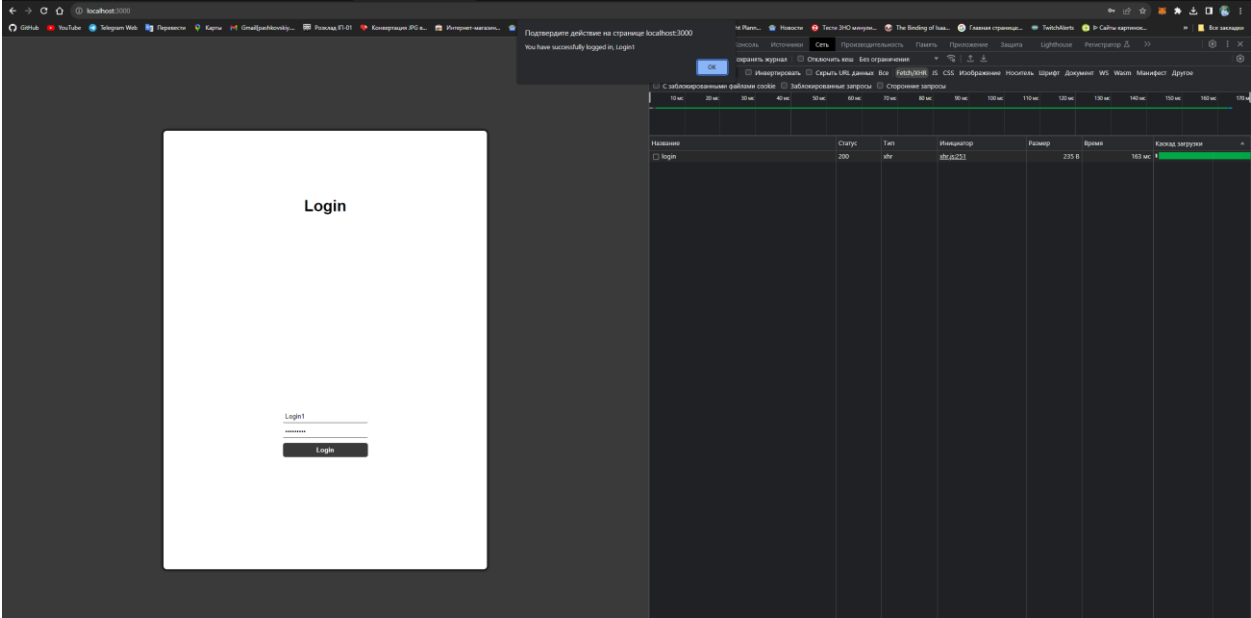
```
PS E:\Projects\security-labs> node .\lab1\forms_auth\  
Example app listening on port 3000
```



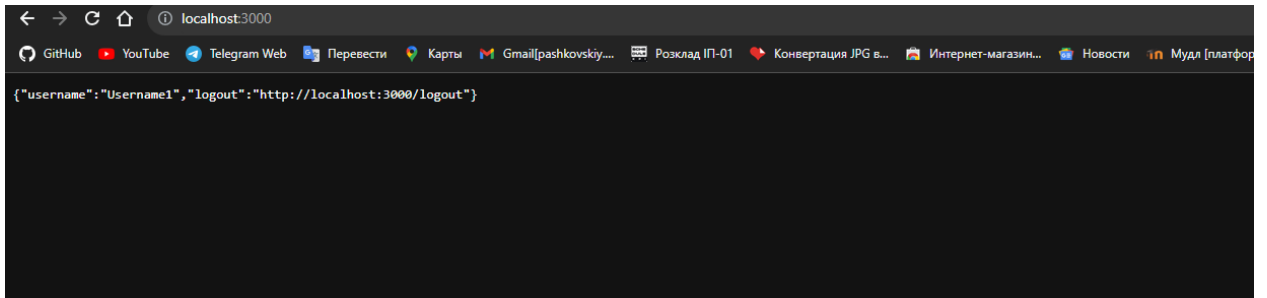
Login1

.....

Login



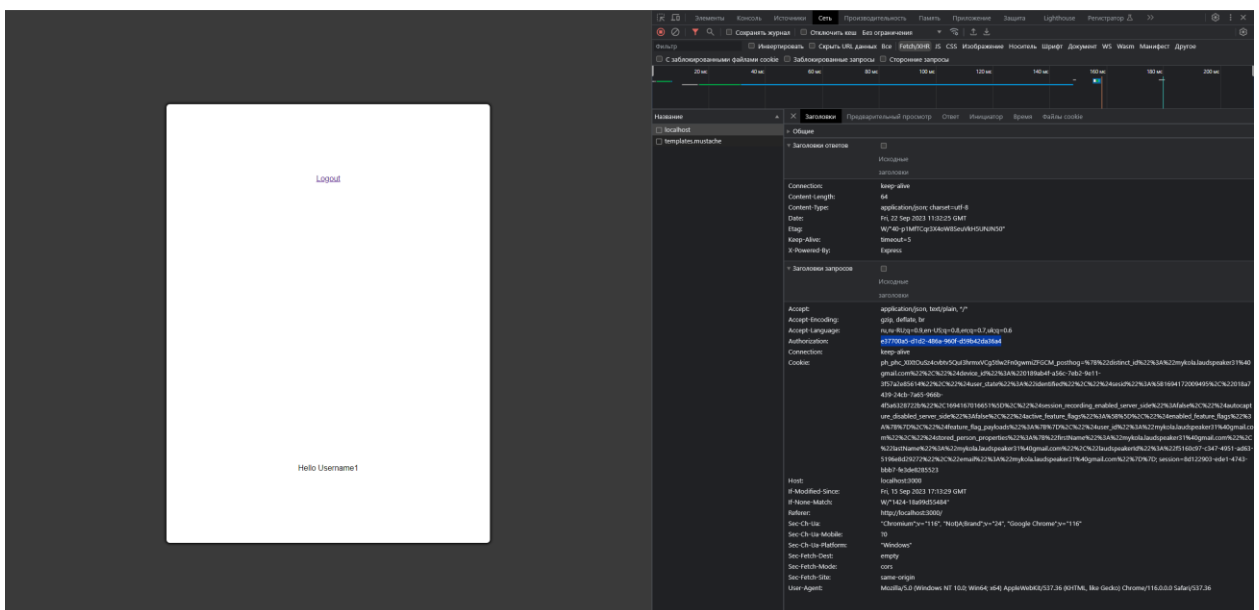
Название	Заголовки	Полезная нагрузка	Предварительный просмотр	Ответ	Инициатор	Время	Файлы cookie
login	Запрос сведений о полезной нагрузке посмотреть ресурс {login: "Login1", password: "Password1"} login: "Login1" password: "Password1"						



```
PS E:\Projects\security-labs> node .\lab1\forms_auth\
Example app listening on port 3000
{}
{ username: 'Username1', login: 'Login1' }
```

3. Запуск та тестування третього додатку

```
PS E:\Projects\security-labs> node .\lab1\token_auth\
Example app listening on port 3000
```



4. Авторизация по токenu за допомогою json web token:

```
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 }));

// get from env
const JWT_SECRET = "jwt_secret";

const users = [
  {
    login: "Login",
    password: "Password",
    username: "Username",
  },
  {
    login: "Login1",
    password: "Password1",
    username: "Username1",
  },
];

app.get("/", (req, res) => {
  try {
    const token = req.get("Authorization")?.replace("Bearer ", "");

    const tokenPayload = jwt.verify(token, JWT_SECRET);
```

```
const user = users.find(({ login }) => login === tokenPayload.login);

if (user) {
  return res.json({
    username: user.username,
    logout: "http://localhost:3000/logout",
  });
}
} catch (e) {
  console.error(e);
  res.status(401);
}

res.sendFile(path.join(__dirname + "/index.html"));
});

app.get("/logout", (req, res) => {
  res.redirect("/");
});

app.post("/api/login", (req, res) => {
  const { login, password } = req.body;

  const user = users.find((user) => {
    if (user.login === login && user.password === password) {
      return true;
    }
    return false;
  });

  if (user) {
    const token = jwt.sign({ login }, JWT_SECRET, {
```

