|  |  |
| --- | --- |
| 스터디 날짜 | 22.05.02 |
| 제출일 | 22.05.10 |
| 학과 | 컴퓨터정보공학부 |
| 학번 | 21 |
| 이름 | 노지혜 |

2022 CAT-CERT Junior Web study 4

**목차**

1. **실습**

**1,2,3**

1. **Web Storage조사 및 실습**
2. **정규표현식 조사 및 실습**
3. **Node.js와 npm, express 조사**
4. **게임 save/load 기능 추가**
5. **php 베스킨라빈스31**
6. **node.js 스마트 출석 만들기**

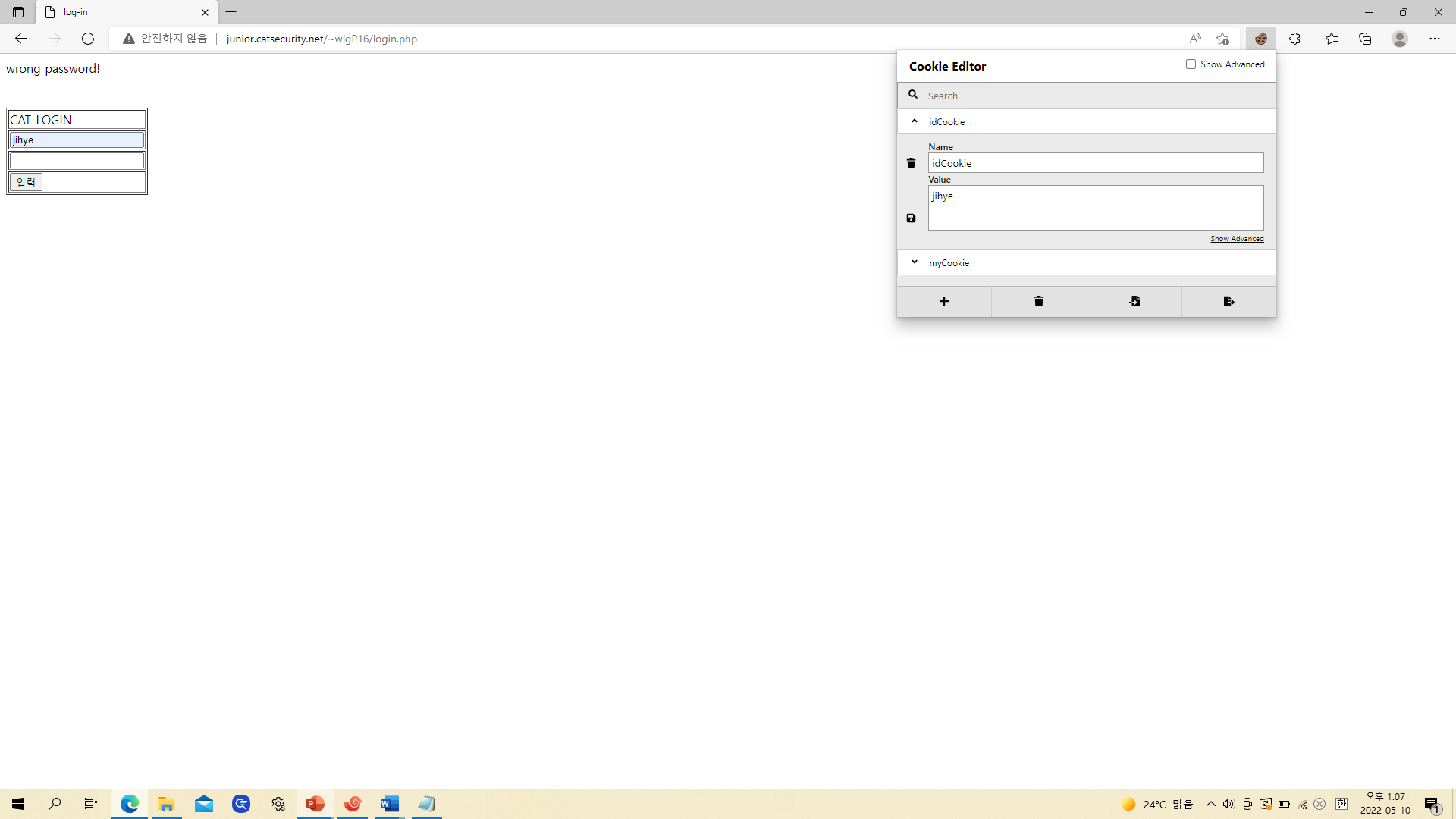
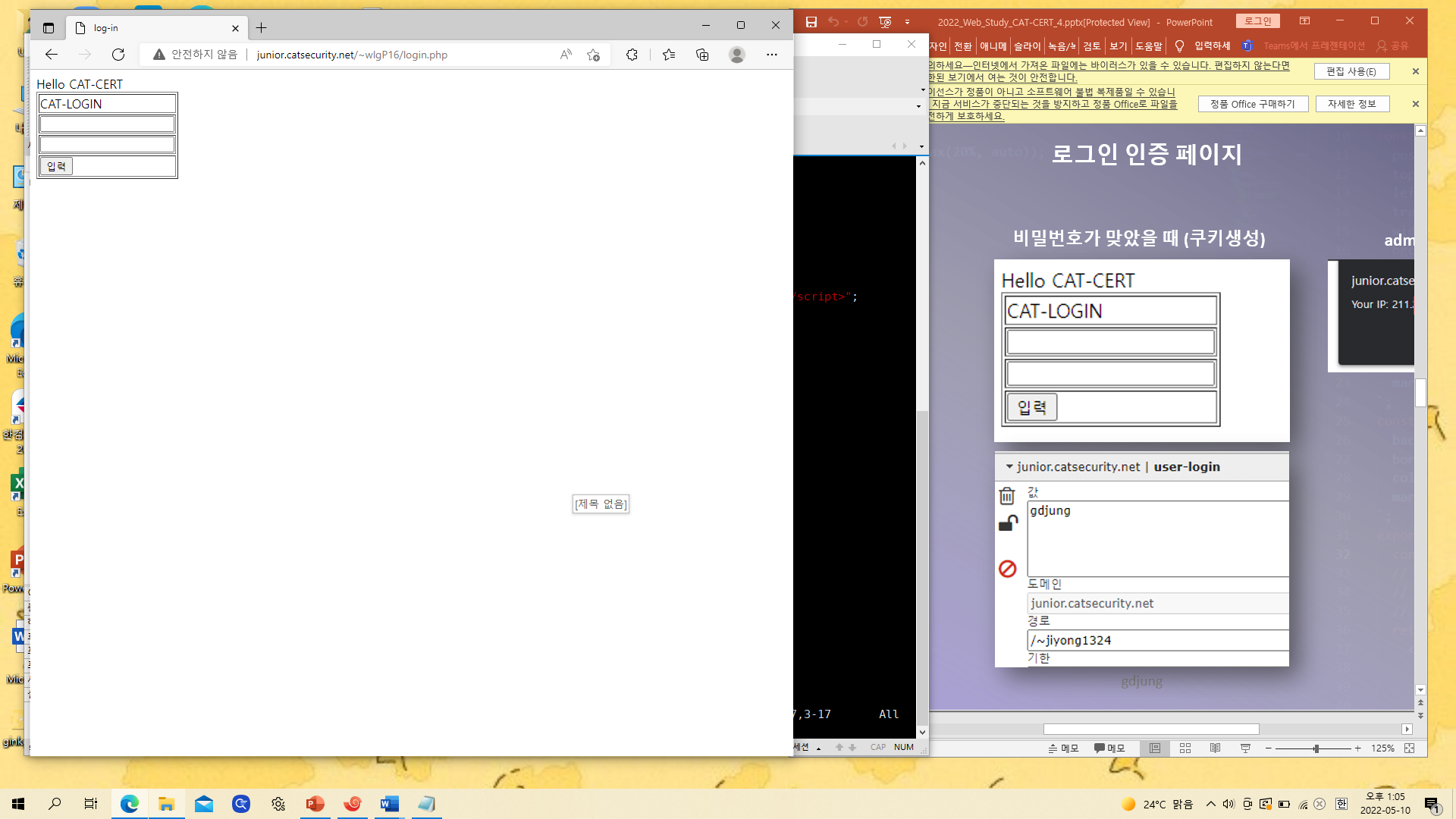
**1. 실습**

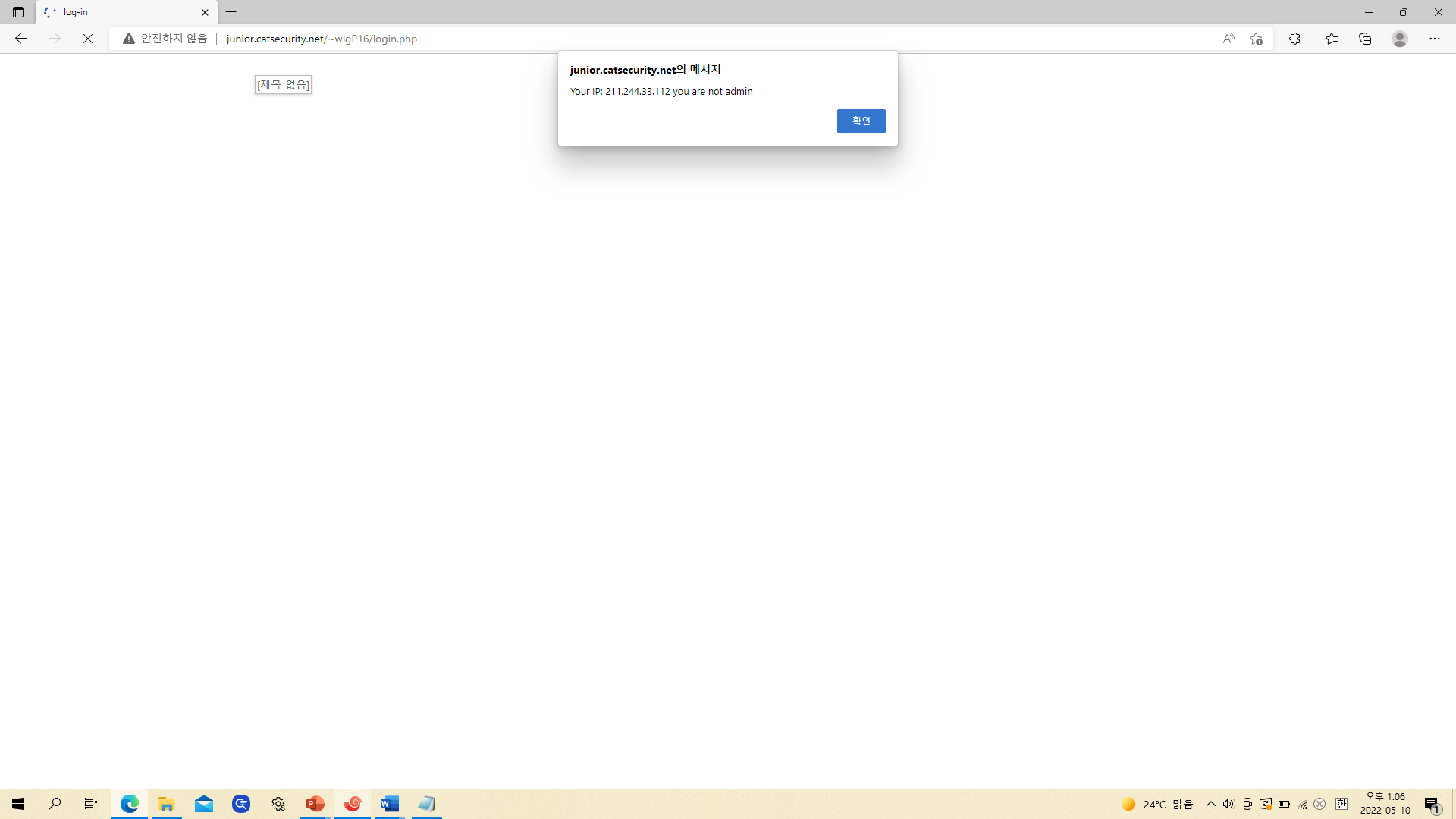
-1 http://junior.catsecurity.net/~wlgP16/login.php

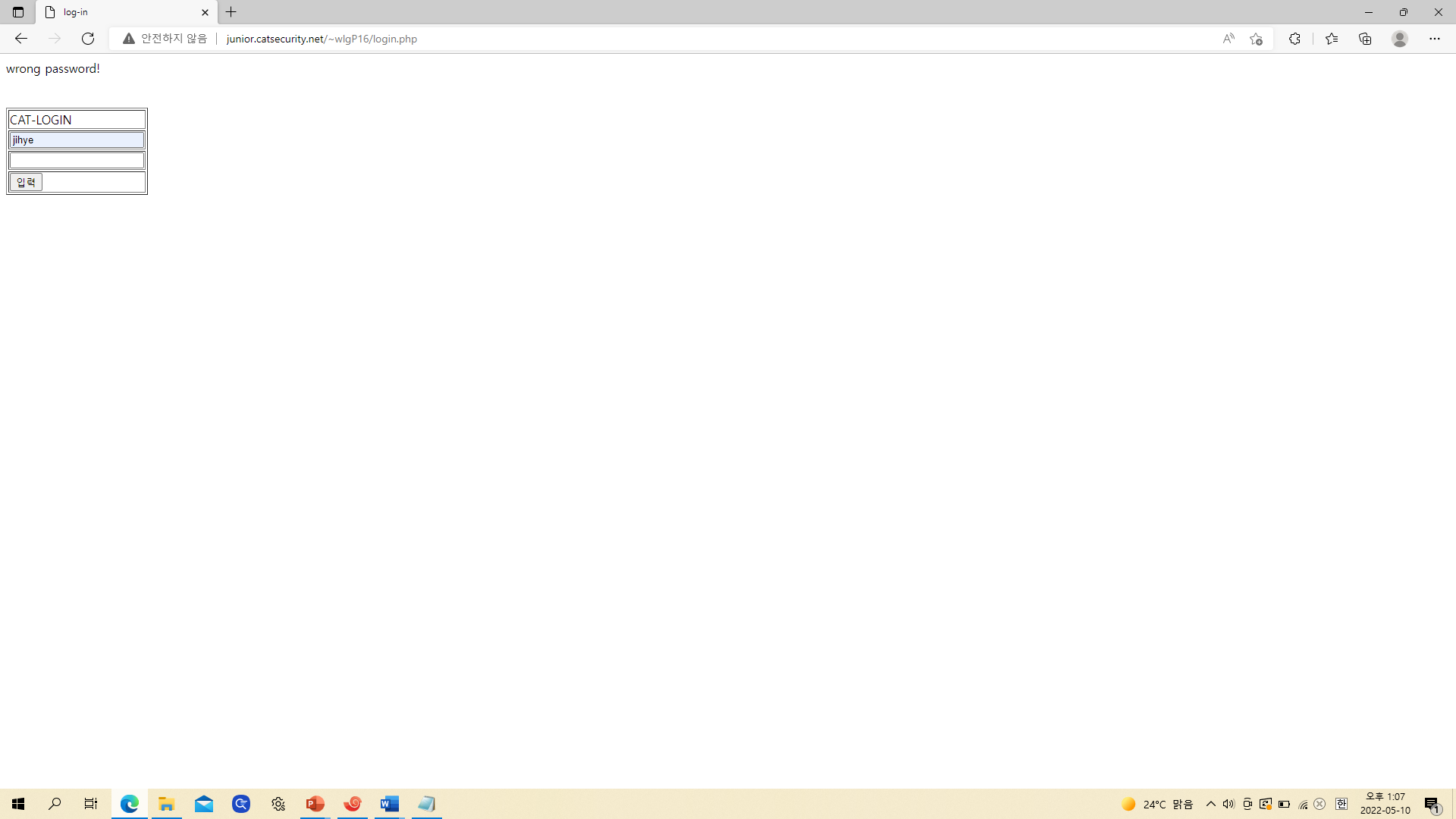
소스코드

|  |
| --- |
| <html>  <head> <title> log-in </title> </head>  <body>  <?php  $pwd="catholic";  $ip=$\_SERVER['REMOTE\_ADDR'];  if($\_POST['submit']==NULL){  if($\_POST['logid']=='admin'){  echo "<script>alert('Your IP: {$ip} you are not admin');</script>";  }  else if( $\_POST['logpwd']==$pwd){  setcookie("idCookie",$\_POST['logid'],time()+(60\*60\*24));  echo("Hello CAT-CERT");}  else echo('wrong password!'.'<br>'.'<br>'.'<br>');  }  ?>  <form method="POST" action="">  <table border="1">  <tr><td> CAT-LOGIN</td> </tr>  <tr><td> <input type="text" name="logid"/></td> </tr>  <tr><td><input type="text" name="logpwd"/></td></tr>  <tr><td><input type="submit" value="입력"/></td></tr>  </table>  </form>  </body>  </html> |

실행 결과







-2

소스코드

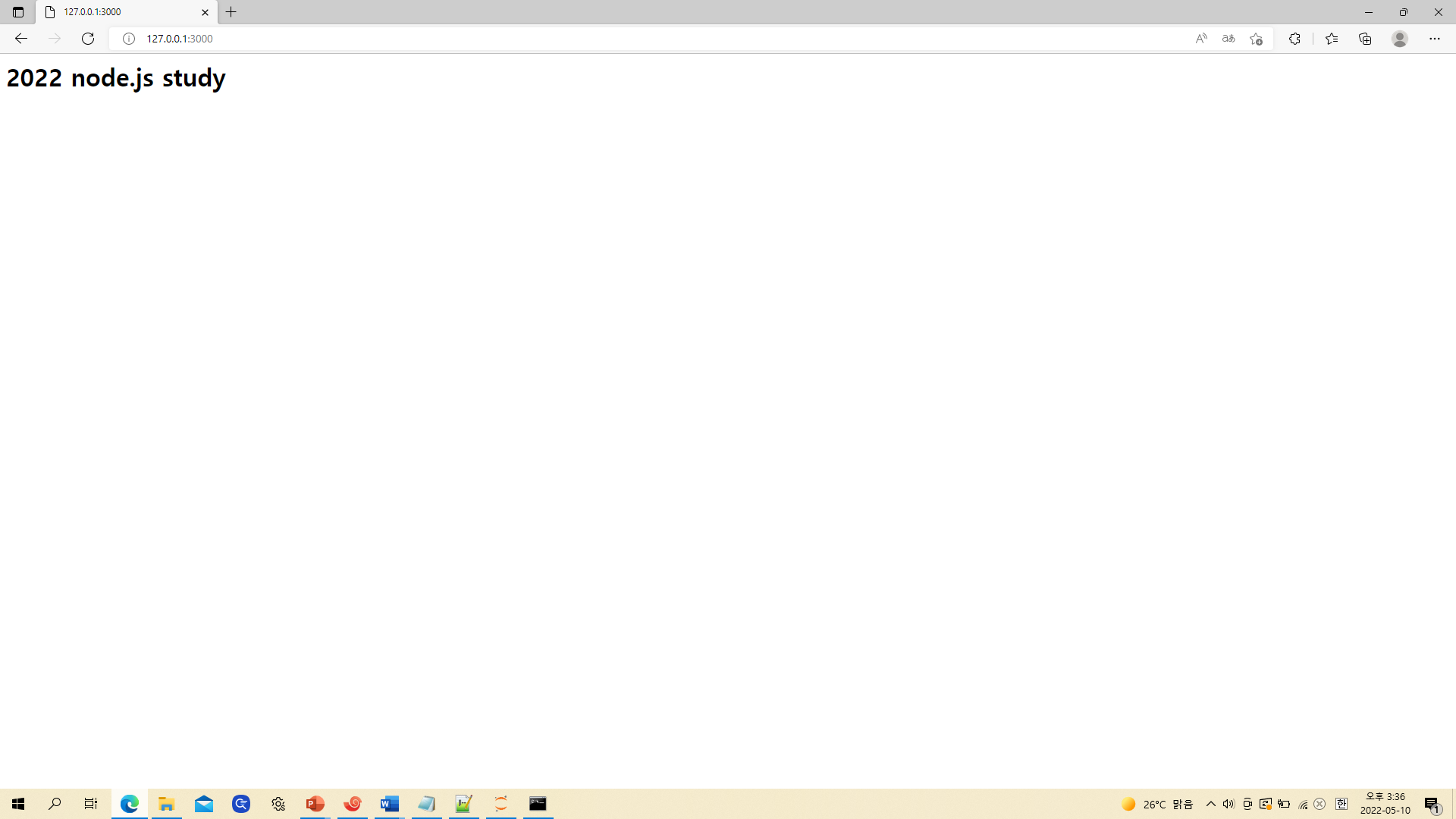
app.js

|  |
| --- |
| var express = require("express");  var app=express();  app.listen(3000,function(){  console.log("listening in port 3000");  });  app.get("/",function(req,res){  res.sendFile(\_\_dirname+"/index.html");  }); |

index.html

|  |
| --- |
| <html>  <head></head>  <body>  <h1> 2022 node.js study </h1>  </body>  </html> |

실행 결과



-3

소스코드

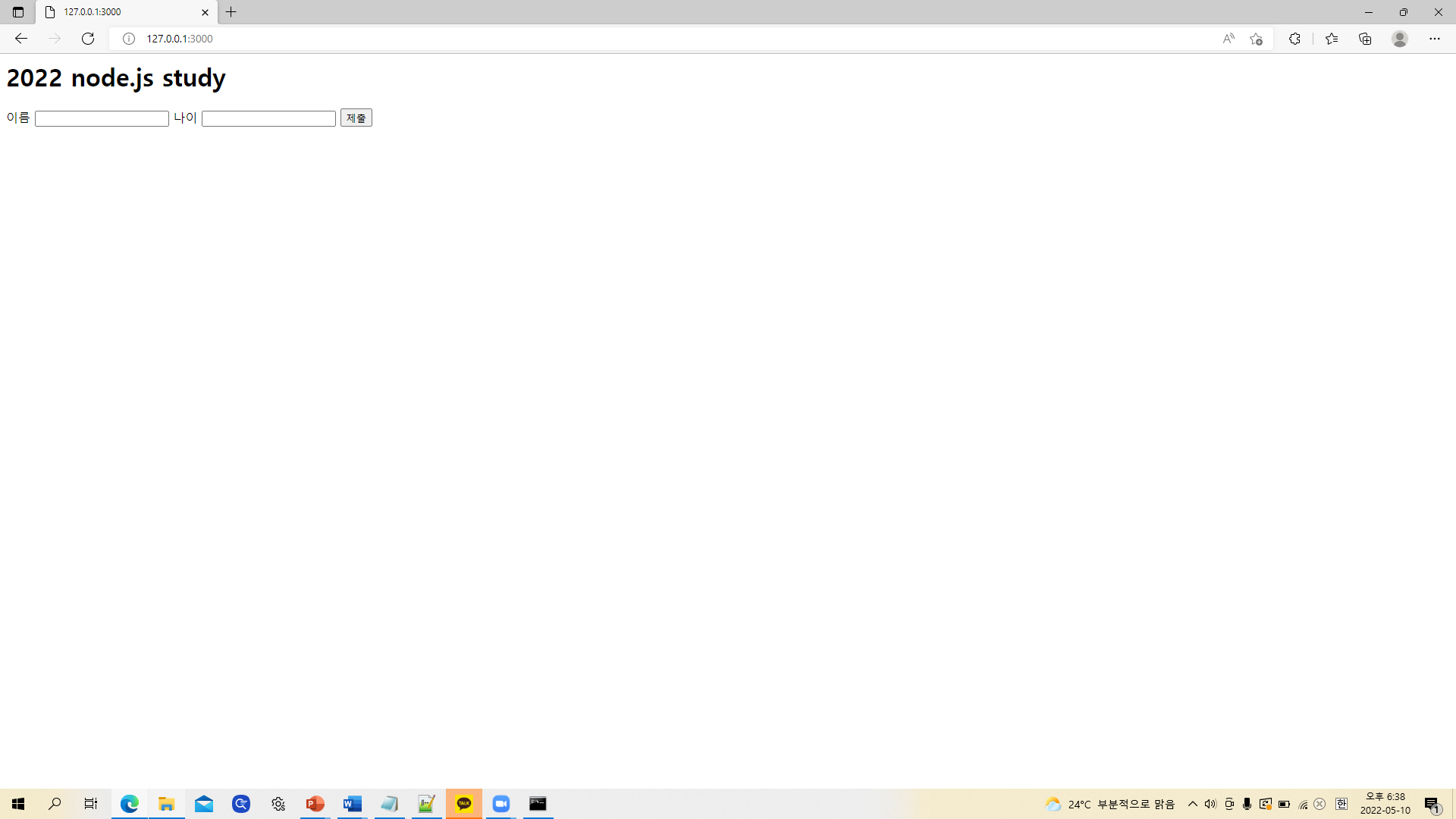
name.js

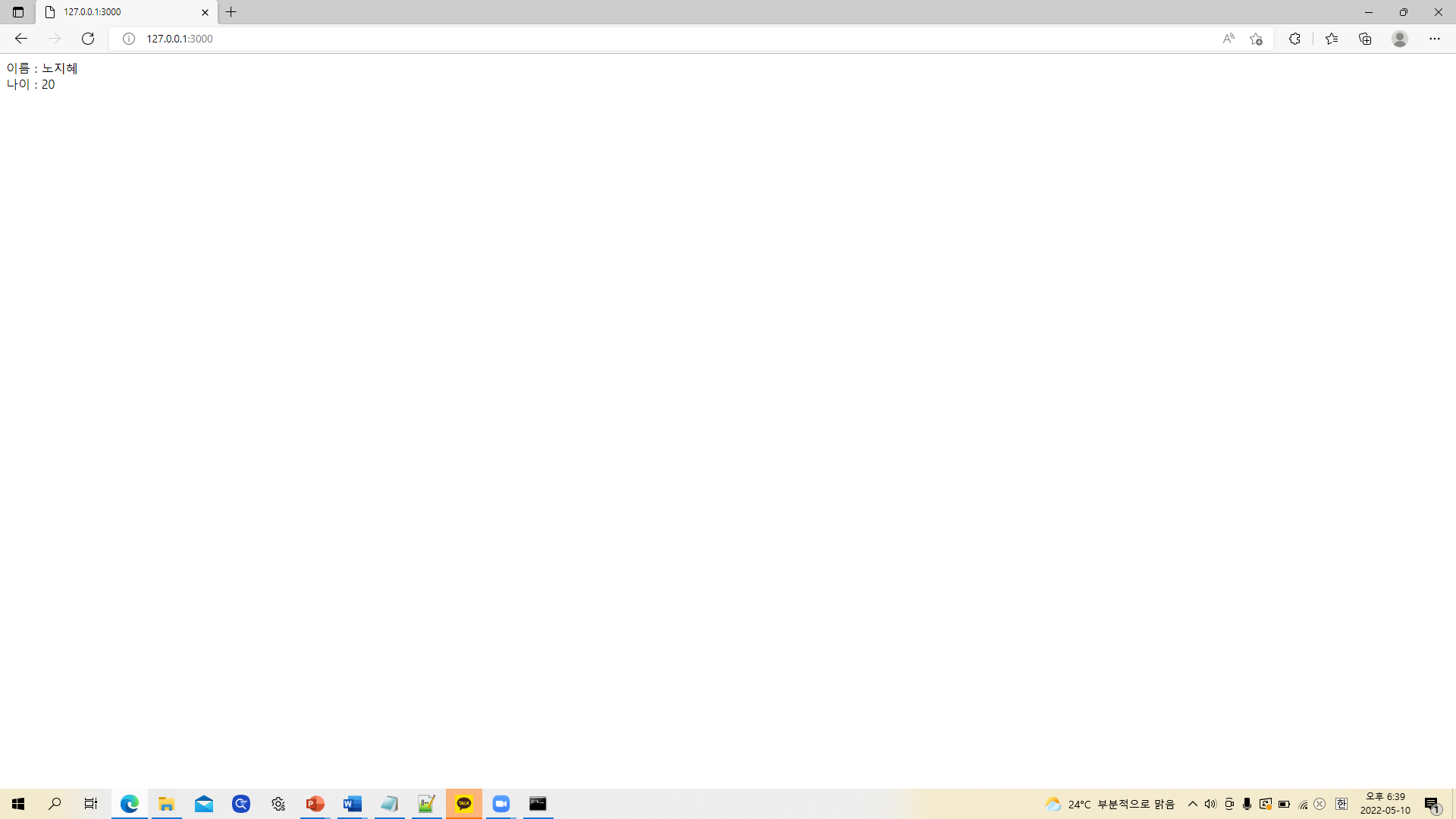
|  |
| --- |
| var express = require("express");  var app=express();  app.use(express.urlencoded({extended:true}));  app.use(express.json());  app.listen(3000,function(){  console.log("server on");  });  app.get("/",function(req,res){  res.sendFile(\_\_dirname+"/input.html");  });  app.post('/',function(req,res){  console.log(req.body.iname,req.body.iage);  var name=req.body.iname;  var age=req.body.iage;  res.send('이름 : '+name+'<br>'+'나이 : '+age);  }); |

input.html

|  |
| --- |
| <html>  <head> </head>  <body>  <h1> 2022 node.js study </h1>  <form method="POST" action="">  이름 <input type="text" name="iname"/>  나이 <input type="text" name="iage"/>  <input type="submit" value="제출"/>  </form>  </body>  </html> |

실행 결과





**2. Web Storage조사 및 실습**

Web storage

HTML5에서 웹 사이트의 데이터를 클라이언트에 저장할 수 있는 웹 스토리지 기능 도입

웹 스토리지의 특징

서버에 전송되지 않으므로 서버에 부담이 가지 않는다.

키와 값의 쌍 형태로 데이터를 저장한다. 키를 기반으로 데이터를 조회

대략 5MB까지의 데이터를 저장할 수 있다.

키와 값은 문자열로 저장하고 대소문자를 구분한다.

세션 스토리지 : 브라우저를 끄는 순간 데이터 모두 지워진다.

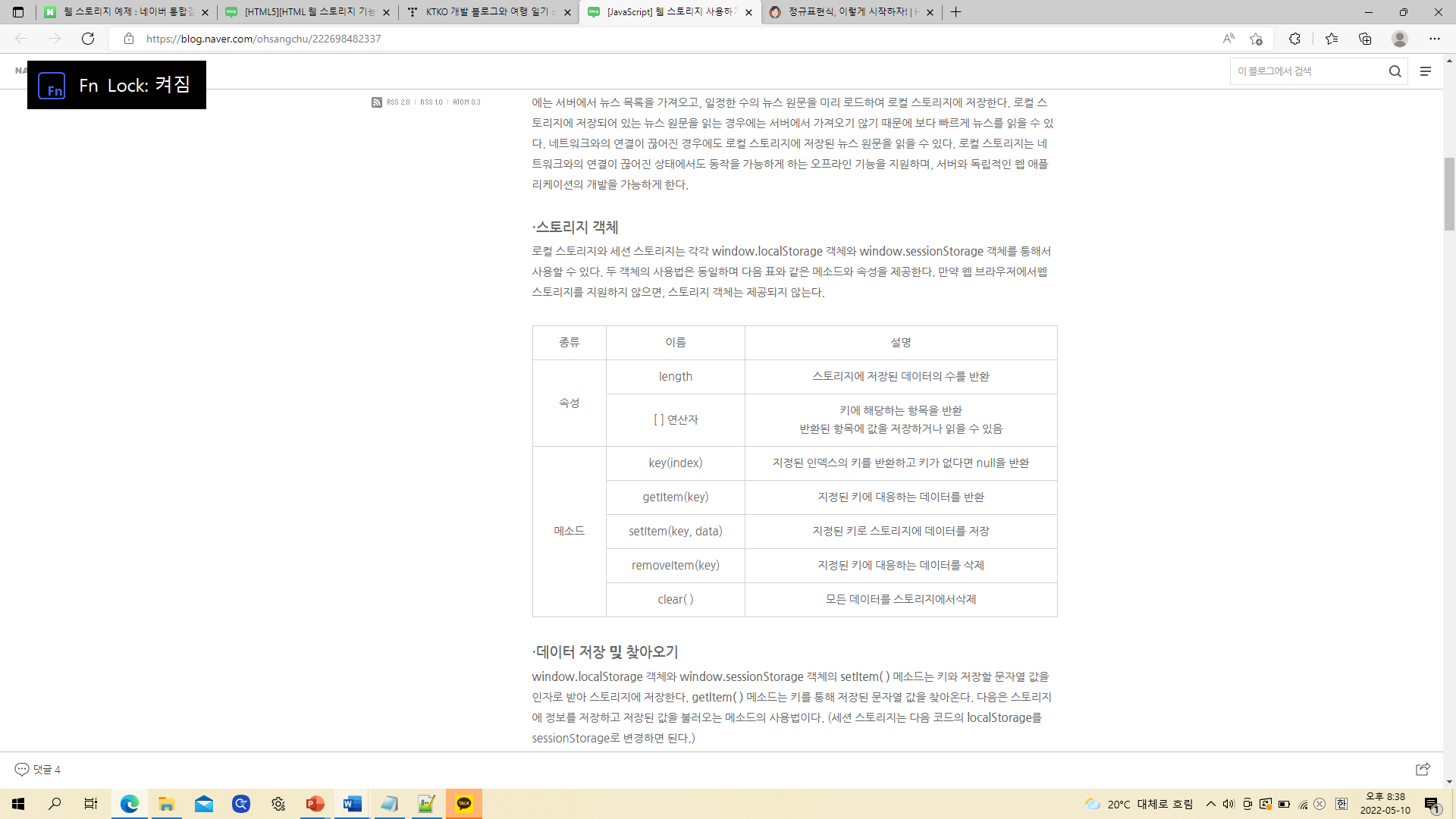
영구 보관하지 않는다. 탭마다 다른 세션 스토리지를 가진다.

로컬 스토리지 : 명시적으로 데이터 삭제과정을 거치기 전까지 지워지지 않음

브라우저가 종료되어도 지워지지 않음

웹 사이트를 다시 로드하면 이미 존재하는 로컬 스토리지를 사용한다.

동일한 컴퓨터, 동일한 브라우저 내에서만 유지된다.



**3. 정규표현식 조사 및 실습**

정규 표현식

일정한 패턴을 가진 문자열의 집합을 표현하기 위해 사용하는 형식 언어(패턴)

문자열을 대상으로 패턴 매칭 기능 제공 - 문자 검색, 문자 추출, 문자 대체 역할 수행

주석이나 공백을 허용하지 않고 여러 기호를 혼합해 가독성이 좋지 않다는 문제가 있다.

자바스크립트 정규식 생성

생성자 함수 방식 - RegExp 생성자 함수를 호출하여 사용할 수 있다.

const regexp1 = new RegExp(“^abc”);

리터럴 방식 – 정규표현식은 /로 감싸진 패턴을 리터럴로 사용한다.

const regexp1 = /^abc/;

재할당 – 사용 중인 정규식을 재할당할 수 있다. 단, 상수가 아닌 변수로 선언해야 한다.

let regexp1 = /ipsum/g;

regexp1 = /lorem/i;

자바스크립트 속성

|  |  |
| --- | --- |
| 속성 | 설명 |
| flags | 플래그 반환 |
| source | 표현식 반환 |
| global | 플래그 g 여부 반환 |
| ignoreCase | 플래그 i 여부 반환 |
| multiline | 플래그 m 여부 반환 |
| sticky | 플래그 y 여부 반환 |
| unicode | 플래그 u 여부 반환 |

플래그

|  |  |
| --- | --- |
| 플래그 | 설명 |
| g | 모든 문자와 여러 줄 일치 |
| i | 영어 대소문자 구분하지 않고 일치 |
| m | 여러 줄 일치 |
| u | 유니코드 |
| y | lastIndex 속성으로 지정된 인덱스에서만 1회 일치 |

메소드

|  |  |  |
| --- | --- | --- |
| 메소드 | 문법 | 설명 |
| exec | 정규식.exec(문자열) | 일치하는 하나의 정보 반환 |
| test | 정규식.test(문자열) | 일치여부 반환 |
| match | 문자열.match(정규식) | 일치하는 문자열의 배열 반환 |
| search | 문자열.search(정규식) | 일치하는 문자열의 인덱스 반환 |
| replace | 문자열.replace(정규식,대체문자) | 일치하는 문자열을 대체하고 대체된 문자열 반환 |
| split | 문자열.split(정규식) | 일치하는 문자열을 분할하여 배열로 반환 |
| toString | 생성자\_정규식.toString() | 생성자 함수 방식의 정규식을 리터럴 방식의 문자열로 반환 |

**4. Node.js와 npm, express 조사**

Node.js

Chrome의 V8 자바스크립트 엔진을 이용하여 자바스크립트를 서버에서도 사용할 수 있도록 만든 런타임( 프로그램을 실행할 수 있는 환경)이다. 웹서버와 같이 확장성 있는 네트워크 프로그램을 제작하기 위해 만들어졌다.

\*스크립트 언어는 특정한 프로그램 안에서 동작하는 프로그램이기 때문에 웹 브라우저 프로그램 안에서만 동작을 한다.

내장 HTTP 서버 라이브러리를 포함하고 있어 웹 서버에서 아파치 등의 별도 소프트웨어 없이 동작하는 것이 가능하다.

자바스크립트를 웹 브라우저에서 독립시킨 것으로 Node.js를 설치하게 되면 터미널 프로그램에서 Node.js를 입력하여 브라우저 없이 바로 실행할 수 있다.

Node.js를 이용하여 서버를 만들 수 있어 이전까지는 클라이언트 부분은 자바스크립트를 사용하고 서버는 Reby, Java 등 다른 언어를 사용해 만들었지만 한 가지 언어로 전체 웹 페이지를 만들 수 있게 되었다.

Non-blocking I/O와 단일 스레드 이벤트 루프를 통한 높은 처리 성능을 가지고 있다.

단일 스레드가 혼자서 일을 처리하지만 들어오는 요청 순서가 아닌 논블로킹 방식(함수 호출 시당장 실행하는 것이 아닌 어느 곳에 쌓아 놓고 동시에 요청을 처리하고 요청이 완료된 순서대로 처리하는 방식)으로 이전 작업이 완료될 때까지 대기하지 않고 다음 작업을 수행한다. \*스레드(프로세스 내에서 실행되는 흐름의 단위)

멀티 스레드 방식에 비해 적은 컴퓨터 자원을 사용한다.

I/O 작업이 많은 서버로 적합하다.

웹 서버가 내장되어 있어 별도의 웹서버를 설치할 필요가 없다

npm(Node Package Manager)

자바스크립트 프로그래밍 언어를 위한 패키지 관리자, Node.js의 기본 패키지 관리자

npm init 현재 디렉터리에 npm 기반으로 프로젝트를 생성 패키지 이름, 버전, 설명 등을 입력

pakage.json 파일 생성 (프로젝트의 정보와 프로젝트가 의존하고 있는 패키지에 대한 정보가 저장되어 있는 파일)

npm install <패키지명> npm을 이용하여 패키지를 설치

npm uninstall <패키지명> node\_modules 폴더, package.json의 “dependencies”객체에서 삭제

Express

Node.js 애플리케이션을 위한 일련의 기능을 제공하는 간결하고 유연한 웹 애플리케이션 프레임 워크이다. Express는 프레임 워크이므로 웹 애플리케이션을 만들기 위한 각종 라이브러리와 미들웨어 등이 내장되어 있어 개발하기 편하고 많은 개발자들에게 개발 규칙을 강제하여 코드 및 구조의 통일성을 향상시킬 수 있다.

사용법

const express = require('express')

const app = express()

**5. 게임 save/load 기능 추가**

**6. php 베스킨라빈스**

**7. node.js 스마트 출석 만들기**

소스코드

attend.js

|  |
| --- |
| var express = require("express");  var app=express();  app.use(express.urlencoded({extended:true}));  app.use(express.json());  app.listen(4000,function(){  console.log("server on");  });  app.get("/",function(req,res){  res.sendFile(\_\_dirname+"/check.html");  });  app.post('/',function(req,res){  console.log(req.body.num);  res.send('출석했습니다.');  }); |

check.html

|  |
| --- |
| <html>  <head> </head>  <body>  <h4> 인증번호 입력하세요. </h4>  <form method="POST" action="">  <input type="text" name="num"/> <br>  <input type="submit" value="출석"/>  </form>  </body>  </html> |

실행 결과

