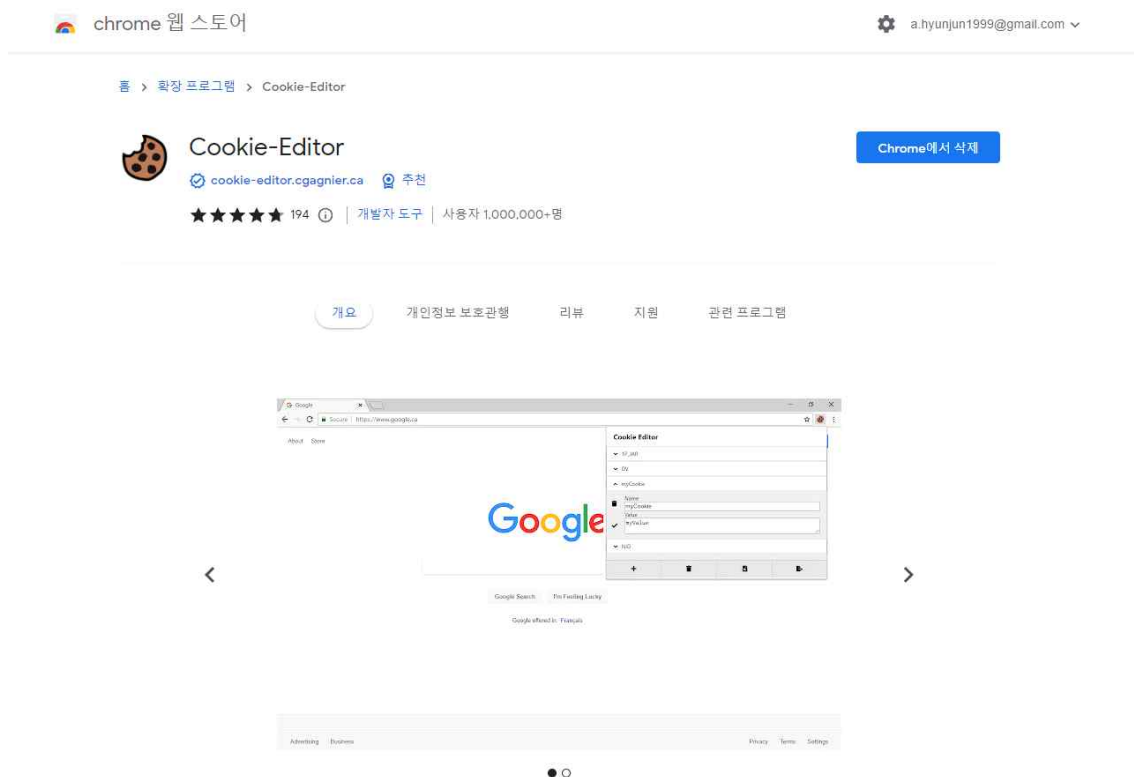


## Interface Test: 브라우저 쿠키 설정 관련 기능

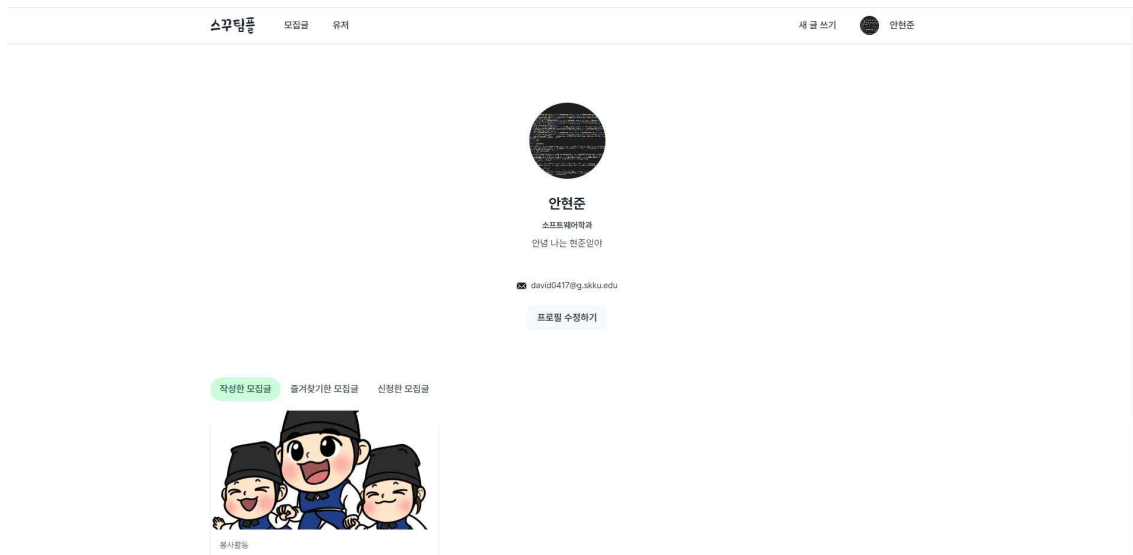
### 1. 쿠키 생성 및 삭제 테스트

본 시스템 어플리케이션의 기능에는 개인 인증이 필수적으로 필요한 서비스가 있다. 이러한 기능을 사용하기 위해서 로그인 후 토큰이 제대로 생성되는지 검증할 필요가 있다. 이를 테스트하기 위해서 크롬 브라우저의 개발자 확장 어플리케이션인 Cookie-Editor 프로그램을 사용했다. 해당 프로그램은 쿠키를 수정하고, 삭제하고 조회할 수 있는 기능을 제공한다. 본 테스트 케이스에서는 로그인 후 쿠키가 생성되고 유효한 쿠키로 마이페이지 진입이 되는지 확인하고, 로그아웃 후 쿠키가 제거되는지 확인하는 테스트를 진행했다.

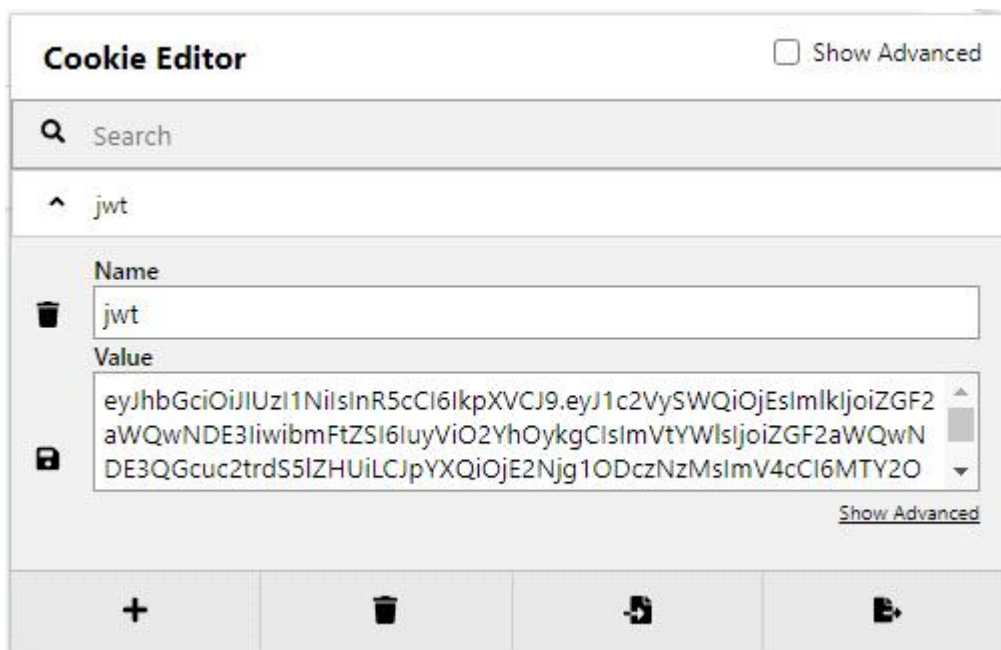


[그림 1] Cookie-Editor 프로그램 설치 페이지

우선 테스트를 위해 로그인을 진행했다. 그 후 Cookie-Editor 프로그램을 통해서 토큰이 정상적으로 생성되었는지 검사했다. 쿠키 토큰이 정상적인 토큰인지 확인하기 위해서 마이페이지 접속을 진행했다.



[그림 2] 로그인 후 마이페이지 접속을 진행. 진행 결과 상태에서 토큰의 상태는 아래 그림과 같음.

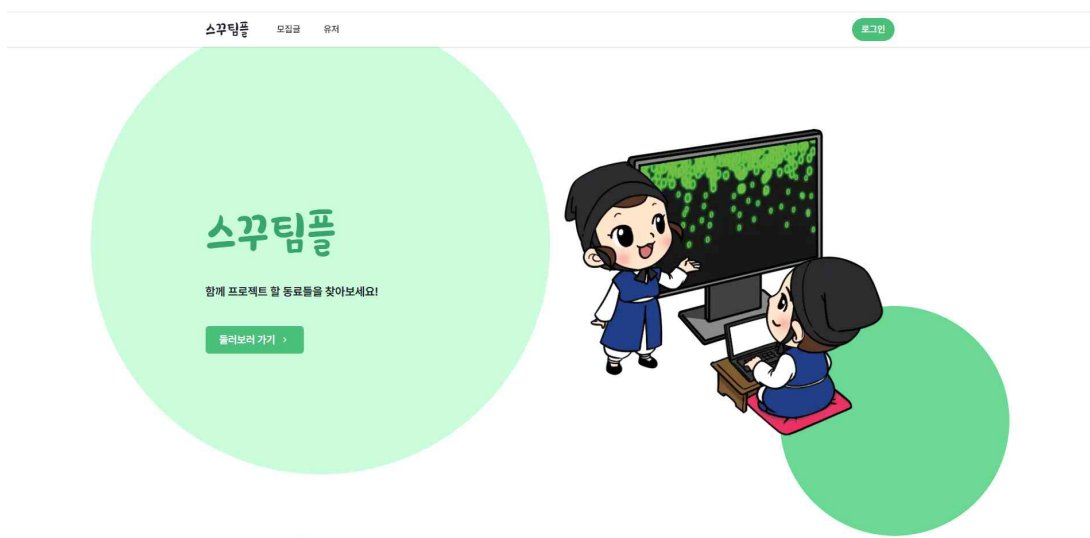


[그림 3] 생성한 쿠키를 확인

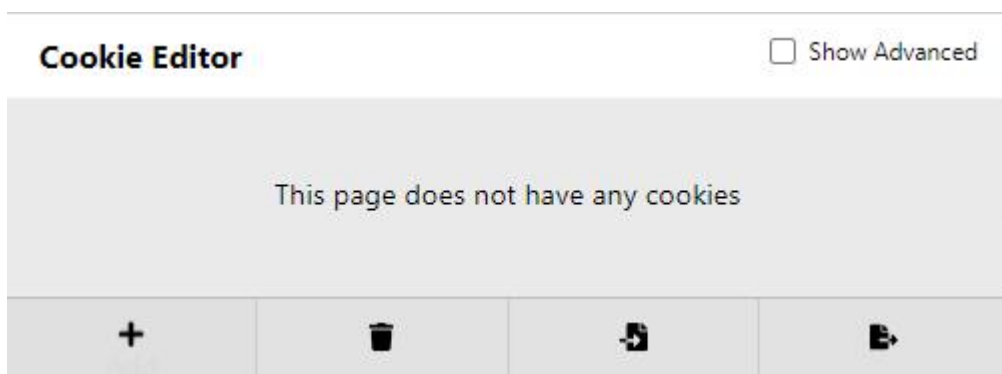
다음으로 로그아웃을 한 경우 쿠키가 삭제되는지 살펴 보았다.



[그림 4] 로그아웃 시도 전



[그림 5] 로그아웃 후 메인페이지로 돌아오며 로그인이 풀린 상태를 확인 가능



[그림 6] 로그아웃 후 쿠키 사라짐

## 2. 유효하지 않은 쿠키에 대한 테스트

본 테스트에서는 두 가지 경우의 유효하지 않은 쿠키 관리 테스트를 진행했다. 첫 번째로 쿠키의 유효기간이 지난 경우를 테스트했고, 두 번째로 쿠키의 값을 유효하지 않은 값으로 수정한 경우를 테스트했다.

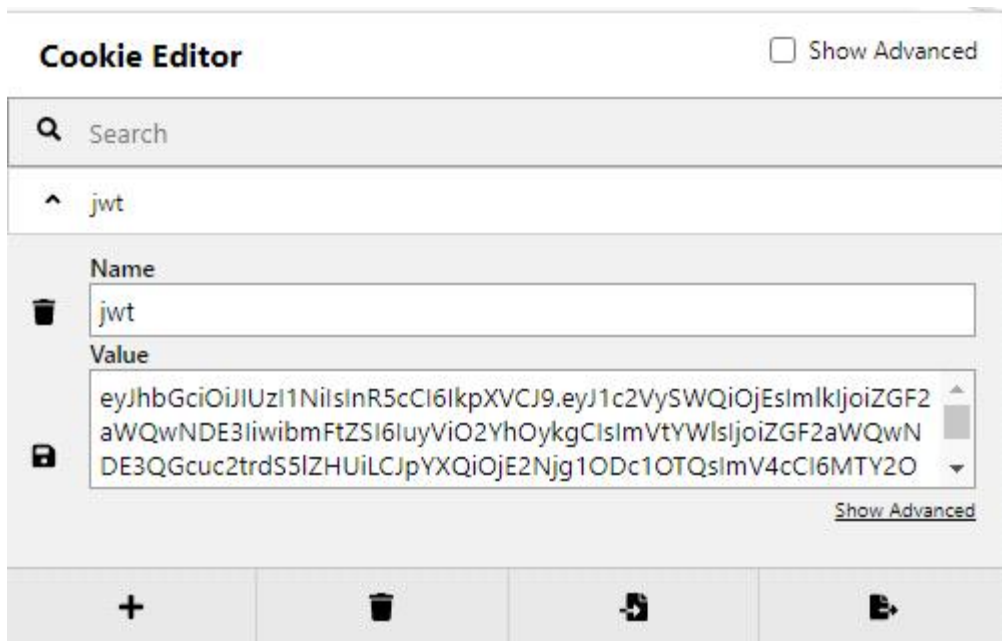
## 2.1) 쿠키 만료시간이 지난 경우

우선 쿠키의 만료시간을 1분으로 설정한 후 토큰을 생성해서 토큰이 1분 이후, 기능하지 못하도록 본 시스템 어플리케이션 코드 일부를 수정했다.

```
process.env.JWT_SECRET,  
{  
  subject: "skku_team_builder",  
  expiresIn: "1m",  
  issuer: "swe42_team12"  
});
```

[그림 7] 코드 수정(쿠키 만료시간 1분으로 설정)

위와 같은 코드로 생성된 토큰은 쿠키가 존재하더라도 제대로 작동하지 못하는 모습을 보여주어야 한다. 테스트 결과 1분이 지난 시점, 본 시스템 어플리케이션의 로그인 상태가 풀리는 모습을 관찰할 수 있었다. 특히, Cookie-Editor로 본 결과, 쿠키는 그대로 있었지만, 제대로 쿠키가 작동하지 않음을 알 수 있었다.



[그림 8] Cookie-Editor에 따르면 쿠키는 여전히 존재함

```

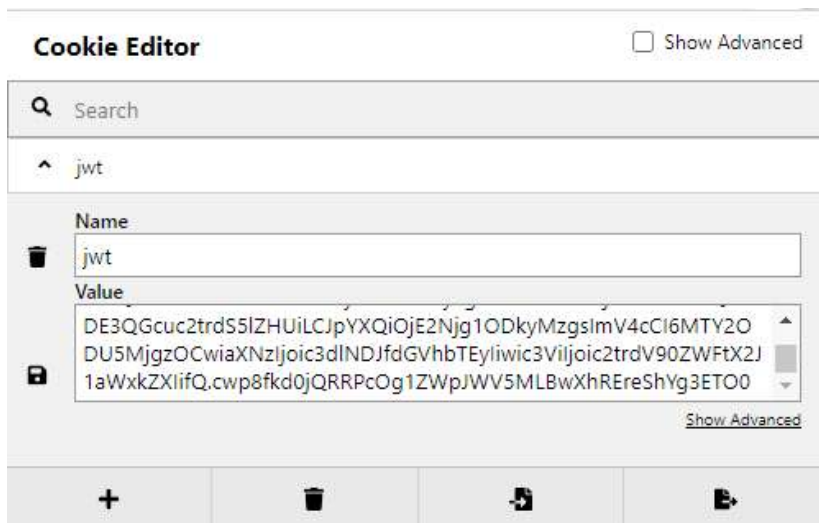
me
me
me
me
Executing (default): SELECT count(*) AS 'count' FROM 'project' AS 'Project' WHERE 'Project'.leader = '1';
Executing (default): SELECT count(*) AS 'count' FROM 'star' AS 'Star' WHERE 'Star'.user = '1';
Executing (default): SELECT 'projectId', 'name', 'topic', 'contact', 'stacks', 'required', 'current', 'stars', 'startTime', 'endTime', 'image', 'message', 'lastModified', 'createdAt', 'leader' FROM 'project' AS 'Project' WHERE 'Project'.leader = '1' ORDER BY 'Project'.lastModified DESC LIMIT 0, 9;
Executing (default): SELECT count(*) AS 'count' FROM 'participate' AS 'Participate' WHERE 'Participate'.user = '1';
Executing (default): SELECT 'userId', 'id', 'password', 'name', 'major', 'email', 'link', 'stacks', 'interest', 'message', 'image' FROM 'user' AS 'User' WHERE 'User'.userId = '1';
Executing (default): SELECT 'userId', 'id', 'password', 'name', 'major', 'email', 'link', 'stacks', 'interest', 'message', 'image' FROM 'user' AS 'User' WHERE 'User'.userId = 1;
Executing (default): SELECT count(*) AS 'count' FROM 'participate' AS 'Participate' WHERE 'Participate'.project = 1;
Executing (default): SELECT 'id', 'user', 'project' FROM 'star' AS 'Star' WHERE 'Star'.project = 1 AND 'Star'.user = '1' LIMIT 1;

```

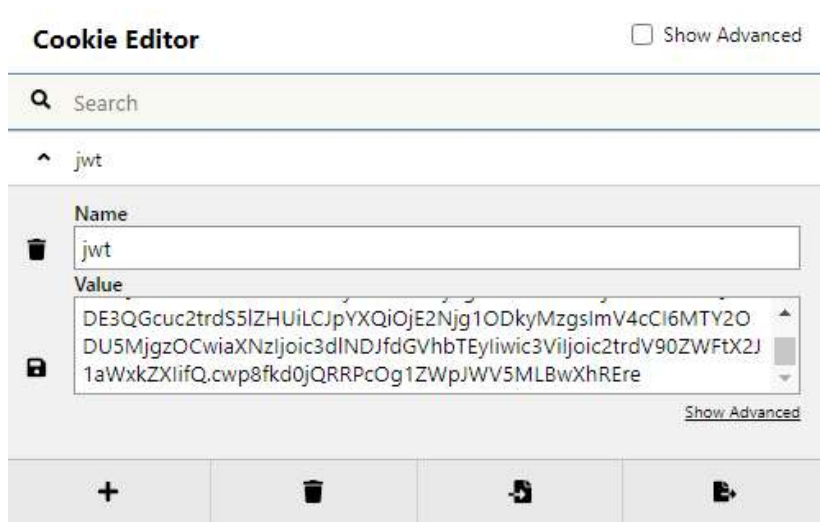
[그림 9] 마이페이지는 계정 권한 확인하는 기능으로 계속해서 반복해서 마이페이지 요청을 보내는 모습을 관찰할 수 있었음

## 2.2) 쿠키 토큰의 값을 인위적으로 변경한 경우(토큰의 integrity 침해)

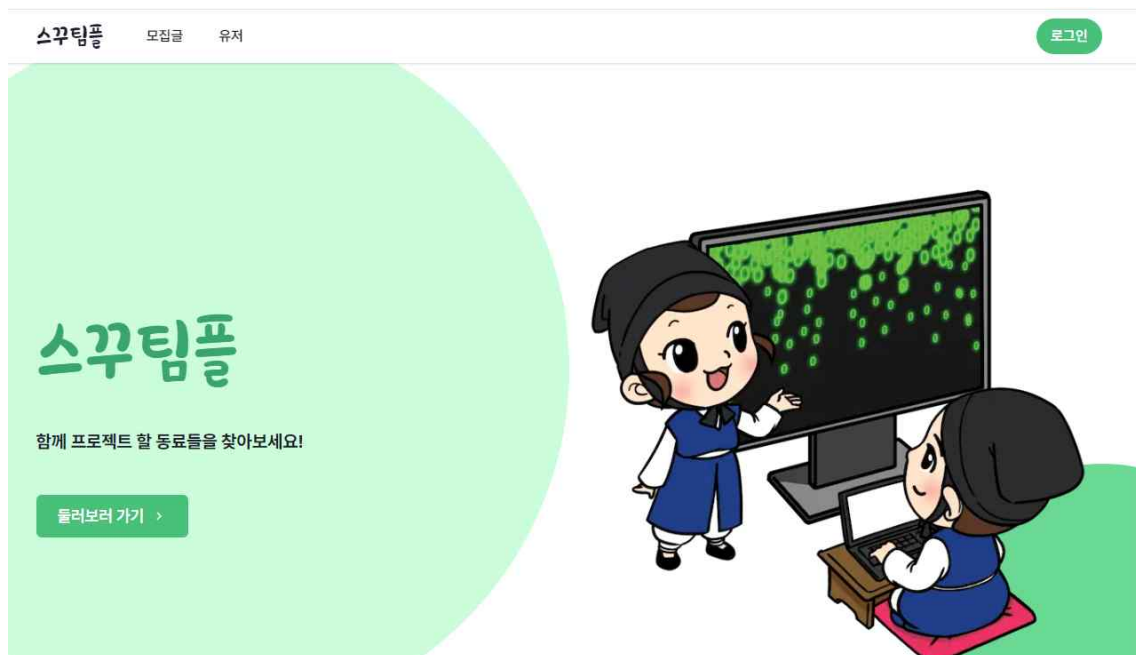
본 테스트 케이스는 Cookie-Editor를 이용해서 인위적으로 쿠키를 변경해서 토큰의 값이 비정상적인 값이 되었을 때 일어나는 상황을 검증하고자 진행한 테스트이다. 테스트 결과 쿠키가 유효하지 않은 값으로 변경될 경우 로그인이 풀리는 결과를 볼 수 있었다.



[그림 10] 로그인 직후 쿠키에 저장된 토큰 값



[그림 11] 쿠키에 저장된 토큰값의 뒤 자리 9자리 삭제



[그림 12] 토큰의 인위적 수정 후 로그인 폴림