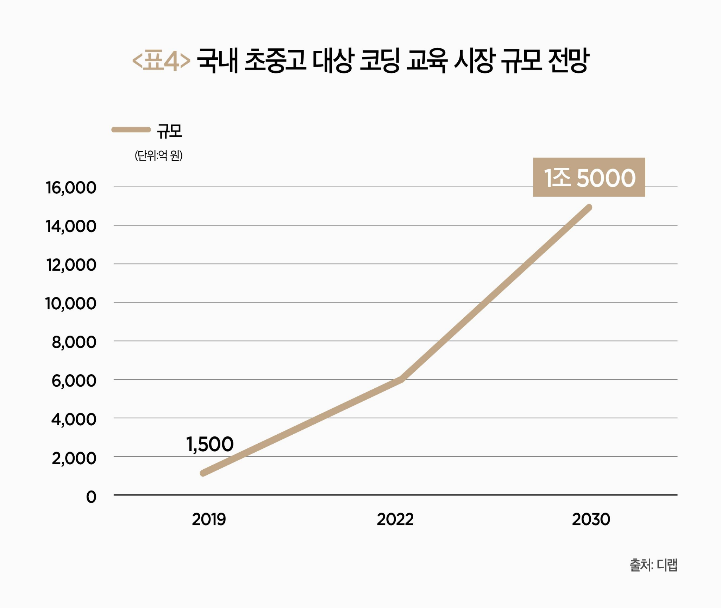
고급웹프로그래밍 텀프로젝트

< KNU Online Judge >

team OH!

|  |  |
| --- | --- |
| 박세찬 | 2018112379 |
| 박재완 | 2018116373 |
| 박정민 | 2019112264 |
| 황아영 | 2019112953 |

**1. 개발배경**

최근 몇 년간 IT 교육의 중요성이 확산되면서, 코딩 교육이 점차 활성화되고 있는 추세이다. 정부의 IT 교육 국비 지원과 같은 국내 IT 인력을 양성하기 위한 정책이 시행 중이며, 표에서 알 수 있듯이 학생들을 대상으로 하는 코딩 교육 또한 활발히 이루어지고 있다. 또한 IT 업계 개발자 연봉이 인상되면서, 많은 비전공자들 또한 IT 업계로 이직을 희망하는 경우도 많다. 이러한 실정에서 코딩을 공부하기 위한 시스템의 수요는 필연적일 것이다. 따라서 Online Judge page를 개발하여 사용자가 스스로 문제를 출제하고 풀이하며 공부할 수 있는 시스템을 구축해보고자 하였다.

**2. 프로젝트 구조**

(1) 개발환경

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| task | Week1  (5/17-5/23) | | Week2  (5/24-5/30) | | Week3  (5/31-6/6) | | Week4  (6/7-6/13) | |
| 계획 및 계획서 작성 |  |  |  |  |  |  |  |  |
| Back-end 개발 |  |  |  |  |  |  |  |  |
| Front-end 개발 |  |  |  |  |  |  |  |  |
| Front-Back 연동 |  |  |  |  |  |  |  |  |
| 보고서 작성 및 발표 준비 |  |  |  |  |  |  |  |  |

Front-end는 EJS, Node.js, CSS, 그리고 JavaScript를 사용하였으며, Back-end는 Node.js Express를 사용하여 개발하였다. 데이터베이스는 mongoDB를 이용하였다. 조원 별로 각각 Back-end와 Front-end의 역할과 개발할 페이지를 분담하였고, github을 이용하여 개발내용을 공유하며 협업하였다.

(2) 프로젝트 일정

위의 마일스톤과 같은 일정으로 2021년 5월 17일부터 2021년 6월 13일까지 약 한 달간 프로젝트를 진행하였다.

**3. 개발내용**

(1) 공통적인 레이아웃

start-bootstrap의 template(<https://startbootstrap.com/template/business-frontpage>)을 활용하였다. 모든 페이지의 공통적인 레이아웃을 views/layouts 폴더에서 관리하였다. footer.ejs, navbar.ejs, scripts.ejs 파일이 이에 해당한다. 공통적인 코드들을 각 파일에 작성하고, 페이지들에서 include하여 사용하였다.

(2) 서버에 데이터 전송하기

Jquery 문의 ajax 함수를 호출해 서버에 request를 보낸다. 데이터는 stringfy를 통해 json 파일로 변환해 전송한다.

// 어떻게 구현했는지 써주세요.

(3) 서버로부터 데이터 받아오기

index.js에서 axios.get()을 비동기적으로 호출하였으며, 함수 파라미터로 해당 서버 url과 권한을 얻기 위한 토큰을 전달하였다. 함수 호출의 결과로 서버로부터 반환된 데이터 중 필요한 부분을 render()의 파라미터로 해당 페이지에 전달하여 사용하였다.

(4) 홈 페이지

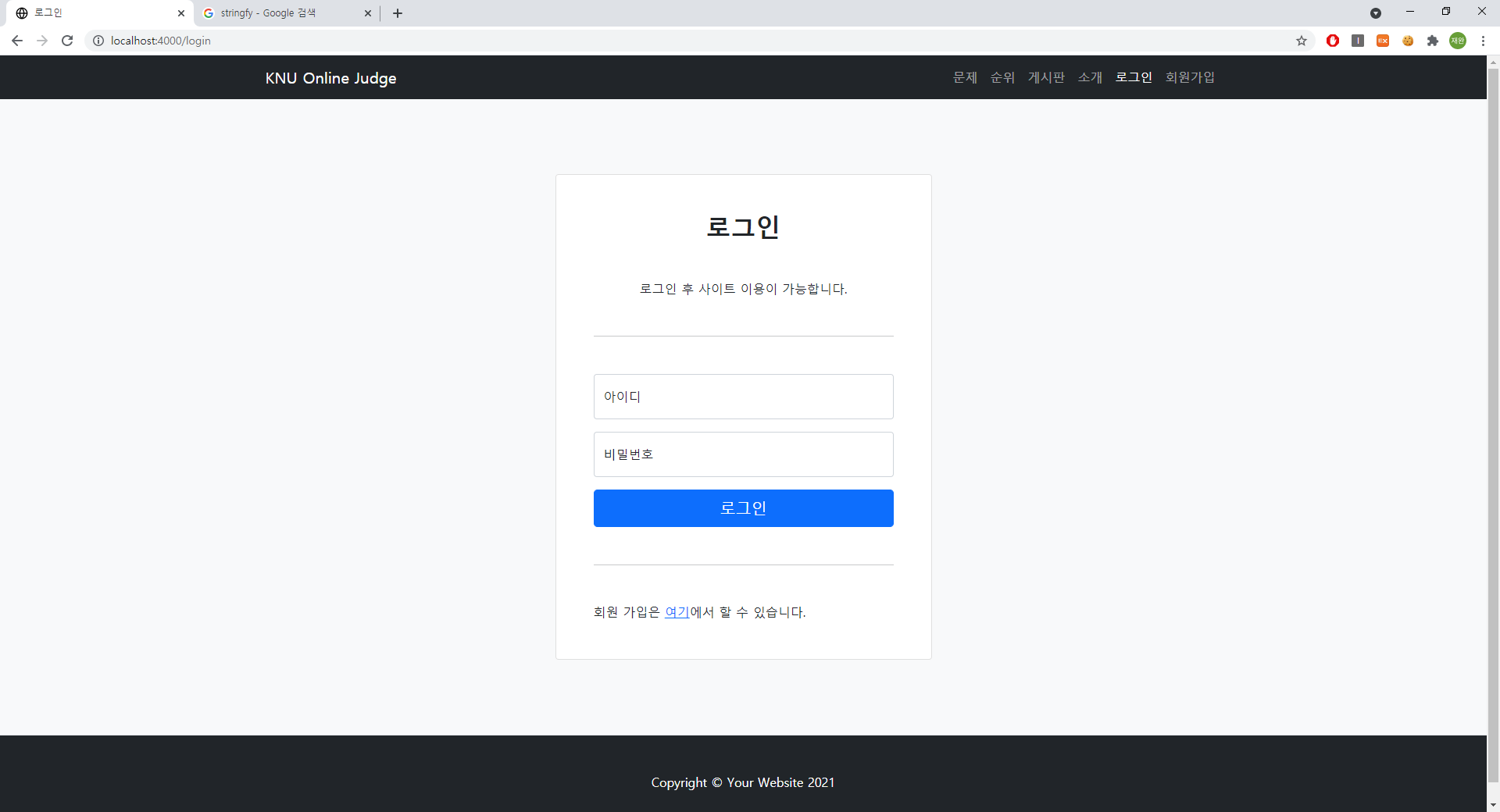
전체 문제, 전체 게시글, 전체 멤버 수를 counter up(<https://github.com/bfintal/Counter-Up>) 효과를 적용해 표시하였다.

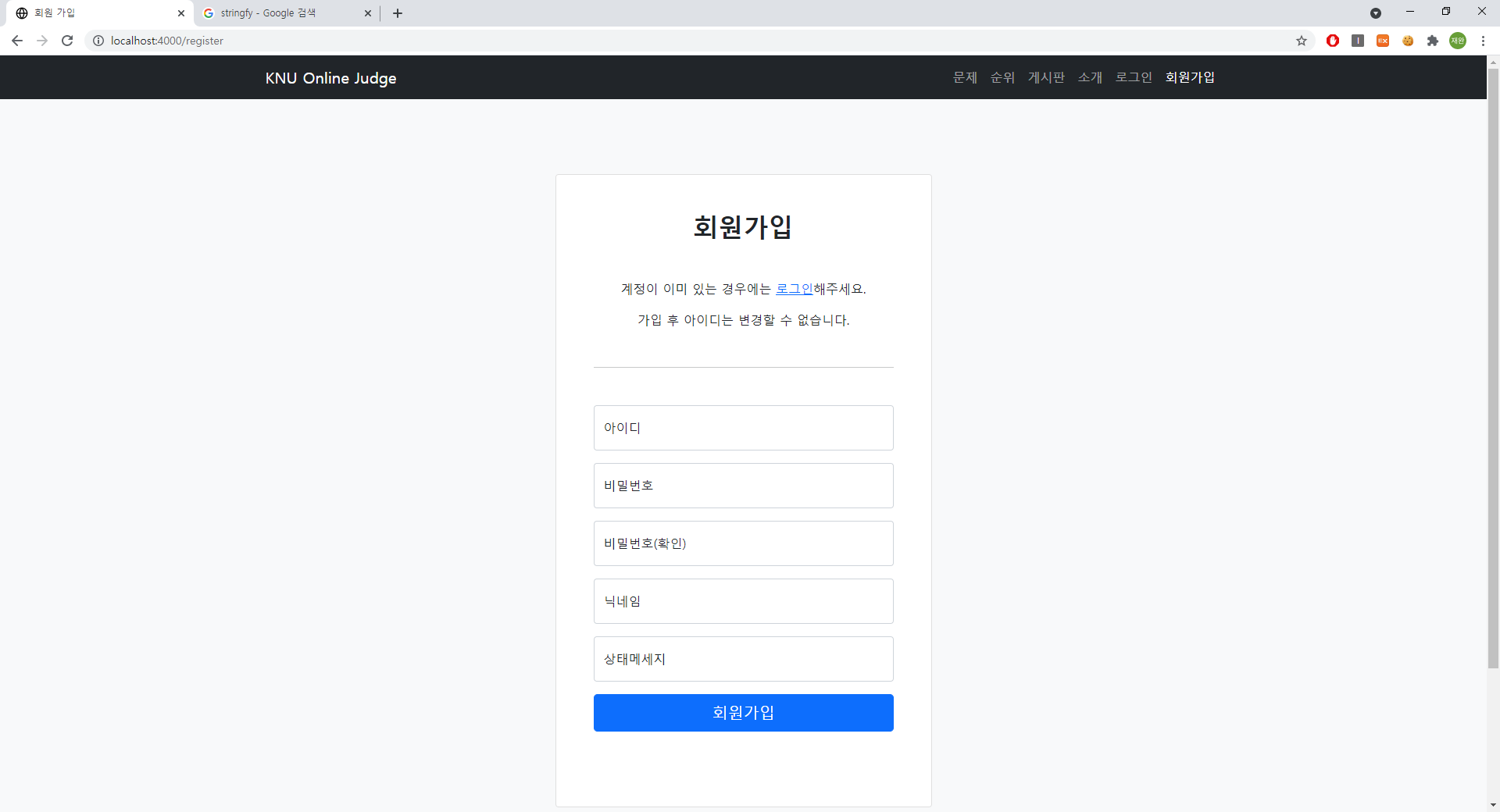
최근 문제, 최근 게시글, 상위순위 10개씩 표시

* (최근 문제, 최근 게시글, 상위순위) - 10개씩
* 로고, 배너, footer

(2) 회원가입 및 로그인 기능

회원가입과 로그인 기능은 로그인 되어있지 않은 상태에서 진행된다. 따라서 토큰 없이 user\_no\_auth/join과 user\_no\_auth/login 을 통해 작동한다.





(3) 문제 출제 및 풀이 기능

‘문제’ 페이지에서 ‘출제하기’ 버튼을 누르면 문제를 출제할 수 있는 페이지로 연결된다. 제목, 난이도, 문제내용, 입력케이스, 출력케이스, 알고리즘 분류를 작성할 수 있다. 해당 form을 채우고 “ADD” 버튼을 누르면 출제가 완료된다. form의 각 필드에 채워진 값들을 서버로 전송하여 DB에 저장되도록 하였다.

문제 출제 완료 후 “문제” 페이지에서 문제 리스트에 출제한 문제가 추가된 것을 확인할 수 있다. 문제 리스트에는 문제 번호, 문제 제목, 난이도, 분류, 맞은 사람 수, 제출 수, 그리고 정답 비율이 표시된다. 서버로부터 “문제”와 관련된 데이터를 가져와 DB에 저장된 문제 수만큼 각 정보를 표시하도록 하였다.

또한, 문제 제목을 클릭해 해당 문제의 구체적인 정보를 확인하고 문제 풀이 코드를 제출할 수 있다. <a> 태그에서 문제 id를 이용하여 페이지 경로를 지정하여 클릭한 문제의 정보를 표시하도록 하였다. // 쓰는 중

// 화면 캡쳐 첨부

(4) 코드 채점 기능

// 동작 과정 + 어떻게 구현했는지 써주세요.

// 화면 캡쳐 첨부

(5) 게시판 기능

“게시판” 페이지에서 사용자들이 작성한 질문 등의 게시글을 보거나, “글쓰기” 버튼을 통해 게시글을 작성할 수 있다. “글쓰기” 버튼을 눌러 form을 채우고 “ADD” 버튼을 누르면 “게시판” 페이지에서 게시글 리스트에 글이 추가된 것을 확인할 수 있다. 게시글 리스트에는 글 번호, 제목, 글쓴이, 작성일, 그리고 조회수가 표시된다. 서버로부터 “게시판”와 관련된 데이터를 가져와 DB에 저장된 게시글 수만큼 각 정보를 표시하도록 하였다.

게시글 리스트에서 글의 제목을 클릭하면 해당 글을 상세하게 확인할 수 있다. 글에는 제목과 내용이 있고, 글 아래에서 댓글을 작성할 수 있다. 작성 후, 작성 란 아래에 댓글의 내용과 작성자의 닉네임을 확인할 수 있다.

게시글 상세 페이지의 하단에는 댓글을 입력할 수 있고, 댓글을 확인할 수 있다.

댓글을 작성한 후 우측의 입력 버튼을 누르면 댓글이 작성된다. 작성한 후 게시물을 다시 확인해보면 이전에 쓴 댓글이 올라와 있음을 확인할 수 있다. 입력이라는 버튼이 클릭되는 순간 서버에 올려지는 request()함수가 실행되고, 댓글의 제목과 내용을 각각 분리해서 서버로 전송한다. 댓글을 불러올 때는 서버에 접근하여, 댓글의 제목, 내용, 작성자, 시간을 불러온 후 해당 칸에 각각 입력된다.

// 화면 캡쳐 첨부

(6) “순위” 페이지

“순위” 페이지에서 회원가입 한 사용자들의 순위를 확인할 수 있다. 사용자 별 맞춘 문제 수를 기준으로 정렬하여 순위를 정하였다. 순위 리스트에는 등수, 사용자 아이디, 상태 메시지, 맞은 문제 수, 제출 수, 정답 비율이 표시된다.

// 화면 캡쳐 첨부

(7) “소개” 페이지

“소개” 페이지에서 개발자 소개와 다양한 애니메이션 효과 등을 확인할 수 있다. 4개의 section으로 구성하였다. 첫 번째 section에서 text 애니메이션을 볼 수 있고, about “Oh!” 버튼을 눌러 두 번째 section으로 이동할 수 있다. 두 번째 section에서는 hover 속성을 이용해 개발자 사진에 마우스를 가져가면 각 정보를 볼 수 있도록 하였다. 세 번째 section에서는 사람이 달리는 것 같은 애니메이션 효과를 삽입하였다. 사람의 움직임은 관절 별로 애니메이션을 따로 설정해 구현되었고, <https://blog.naver.com/co-nam/222190411502>의 코드를 수정하여 사용하였다. 마지막 section에는 자동으로 움직이는 배너를 삽입하였다. overflow: hidden 속성을 이용해 한 번에 배너 하나만 보이도록 설정하였다. <https://blog.naver.com/co-nam/222198229342>의 코드를 수정하여 사용하였고, 이미지는 본 학부 홈페이지 배너의 이미지를 사용하였다.

// 화면 캡쳐 첨부

(8) 로그인 시 navbar 변경

Navbar에 있는 항목 중 마지막 두 항목인 로그인과 회원가입 항목은 로그인 시 로그아웃과 내 정보 페이지로 연결되는 아이디로 표시된다.





변경되는 코드는 logout.js 파일에 jquery문으로 작성되어 있다.

로그인 시 생성되는 cookie가 존재하면 로그인 버튼의 text를 로그아웃으로 바꾸고, onclick으로 logout()을 연결해 준다.

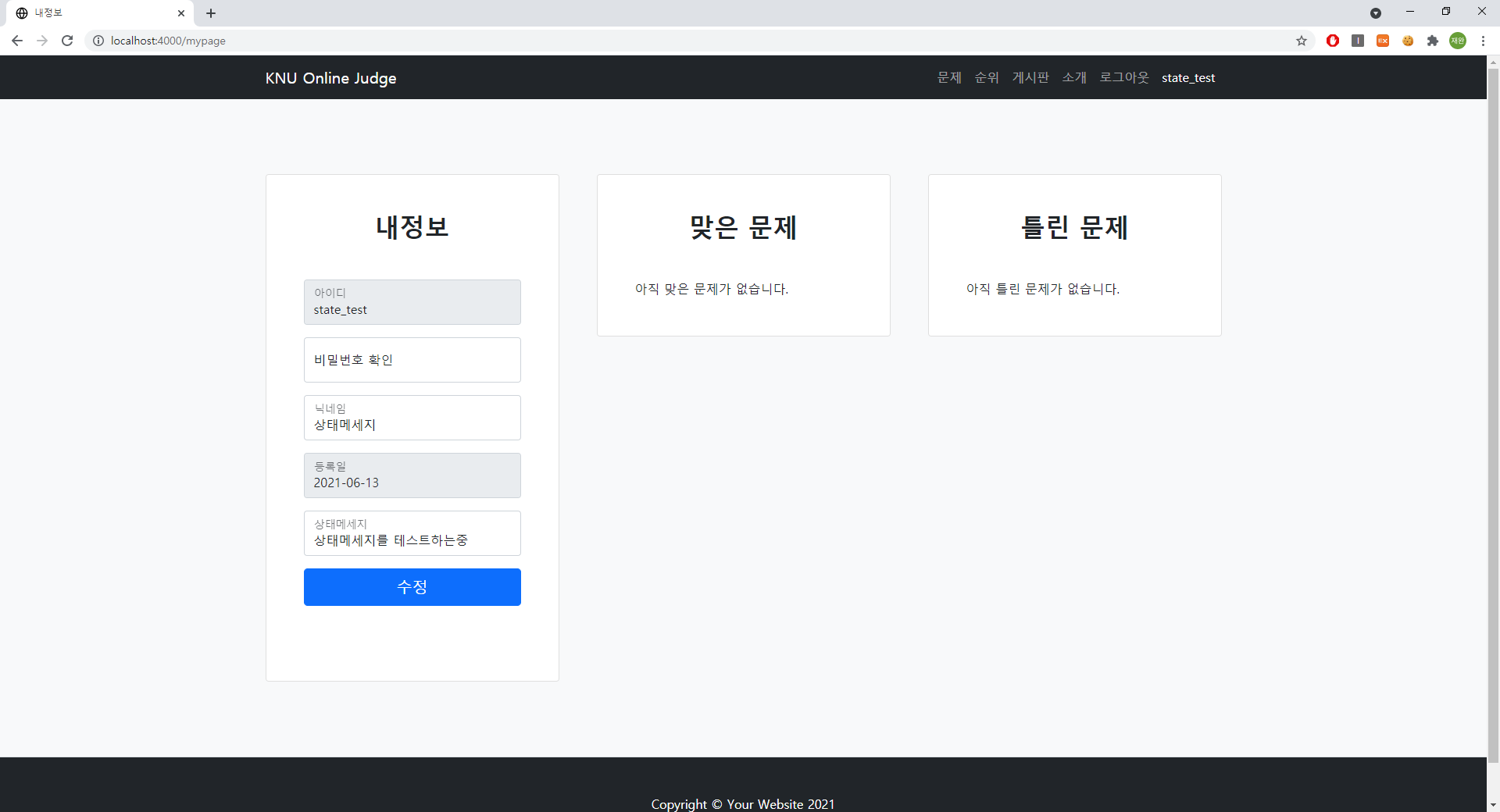
연결되는 logout() 함수는 쿠키의 만료기간을 과거로 설정해 쿠키를 삭제한다.

현재 로그인 되어있는 아이디를 표시하기 위해 토큰을 사용해 서버에서 id를 받아온다.

(9) 내 정보

DB에서 아이디, 닉네임, 등록일, 상태메세지를 가져와 표시한다. 아이디와 등록일은 수정할 수 없고, 비밀번호를 입력하면 닉네임과 상태메세지 수정이 가능하다.

맞은 문제 리스트와 틀린 문제 리스트를 표시한다.



**4. 기대효과**

본 웹 프로그램은 주된 채점 언어로 Python을 사용하기 때문에, 향후 코딩 미숙련자의 학습 보조 및 학교 서버로 활용될 수 있을 것이다. 또한, 게시판 기능을 통해 사용자 간 소통이 가능하고, 개발자들의 상호성장을 기대할 수 있을 것이다. 본 프로젝트가 국내 개발자 양성에 이바지하고, 우리나라를 IT 강국으로 만드는 데 기여할 수 있기를 바란다.

//reference