(슬라이드 1)   
안녕하십니까 오늘 세션과 쿠키에 대해 발표하게 된 소프트웨어학과 22학번 이이슬입니다.   
세션과 쿠키에 대해 설명하기 전 알아야 할 개념들을 보고 들어가겠습니다.

(슬라이드 2)   
먼저 인증이란 유저가 누구인지 확인하는 절차이고, 로그인 즉 유저의 identification (신분증)을 확인하는 절차입니다. 그리고 인가는 유저의 권한을 허락하는 것입니다.  
오늘 설명할 주제는 인증을 위한 방식이기 때문에 차이점은 토큰에 대해서 공부하게 된다면 참고하시길 바라고 넘어가겠습니다.

(슬라이드 3)   
다음으로는 HTTP프로토콜의 특성이자 약점인데요.  
첫번째로는 connectionless 프로토콜로 비연결 지향하는 것입니다. 이는 클라이언트가 서버에 요청을 했을 때, 그 요청에 맞는 응답을 보낸 후 연결을 끊는 처리 방법인데요. 이를 통해서 수만명의 웹 서비스 유저를 동시에 받더라도 접속유지는 최소한으로 하기 때문에 더 많은 유저의 요청을 처리할 수 있습니다.  
다음으로는 stateless 프로토콜로 상태정보를 유지 안 하는 것입니다. 이는 클라이언트의 상태 정보를 가지지 않는 서버처리 방식이며, 위의 connectionless로부터 파생되는 특징으로 클라이언트의 이전 상태 정보를 알 수 없다면 계속해서 로그정보를 유지할 수가 없습니다. 왜냐하면 클라이언트와 첫번째 통신에서 데이터를 주고받았더라도, 두번째 연결에서는 첫번째 데이터를 유지하지 않기 때문이죠.

(슬라이드 4)   
다음으로 쿠키에 대해 알아보겠습니다. 쿠키는 클라이언트 즉 브라우저 로컬에 저장되는 키와 값이 들어있는 작은 데이터 파일입니다. 그리고 사용자 인증이 유효한 시간을 명시 가능하며 유효시간이 정해지면 브라우저가 종료되어도 유지되고,   
사용자가 따로 요청하지 않아도 브라우저가 자동으로 HTTP 헤더에 포함하여 서버에 전송합니다.  
예시로는 로그인 정보 저장, 쇼핑몰 장바구니 기능 등이 있습니다.  
쿠키의 장점으로는 서버가 아닌 클라이언트에서 데이터를 저장하기 때문에 서버의 부하를 줄일 수 있으며, 사용자별로 개인화된 정보를 저장하고 이를 기반으로 사용자 경험을 개선할 수 있습니다.  
쿠키의 단점으로는 보안성이 낮으며, 클라이언트 측에서 삭제하거나 변경할 수 있기 때문에 민감한 데이터를 저장하기에 부적합하며, 사용자 브라우저의 저장공간이 한정적이기 때문에 큰 데이터를 저장하기 어렵습니다.

(슬라이드 5)   
다음으로는 세션인데요. 세션이란 쿠키를 기반으로 하지만 사용자 파일 정보를 브라우저에서 저장하는 것이 아닌 서버 측에서 관리하는 것을 의미합니다.  
클라이언트와 서버간 네트워크 연결이 지속되는 동안 사용자의 상태를 유지하기 위한 방법이며,  
서버에서는 클라이언트를 구분하기 위해서 세션 ID값을 부여하고, 서버에 접속해 있다면 인증상태를 유지합니다.  
세션의 장점으로는 상태유지, 사용자의 정보와 설정 등을 유지할 수 있고, 세션 정보는 서버에 저장되므로, 클라이언트에서 직접적으로 접근하거나 수정하는 것이 어려우며 대용량의 데이터를 저장할 수 있어 대용량 데이터와 보안상의 장점이 있습니다.  
단점으로는 서버에 저장되어 많은 사용자가 접속하면 서버의 메모리를 많이 차지하게 되어 서버 부하의 단점이 있고, 사용자가 브라우저를 변경하거나 다른 기기로 이동하면 세션을 유지할 수 없습니다. 그리고 마지막으로 세션 ID가 공격자에게 노출되면, 공격자가 세션ID를 이용해 사용자를 가장할 수 있는 단점이 있고 이를 세션 하이재킹이라 합니다.

(슬라이드 6)   
다음으로 쿠키와 세션에 대해 비교하며 정리해 보겠습니다.  
먼저 공통점으로는 웹사이트가 사용자의 브라우저를 통해 사용자의 상태를 기억하고 추적하는데 사용되는 기술이라는 것이고, HTTP의 stateless한 특성을 보완하기 위해 사용되며, 웹사이트와 사용자의 상호작용을 개선하는데 도움이 된다는 것입니다.  
차이점으로는 쿠키는 텍스트 형식으로 클라이언트에 저장되는 반면, 세션은 객체 형태로 서버에 저장되고, 쿠키는 직접 삭제 전까지 유지되는 반면 세션은 서버와의 연결 종료까지 유지됩니다.

(슬라이드 7)  
이제 실습을 한번 해보겠습니다. 먼저 react-app으로 폴더 만들어 주시고 cookie와 session 라이브러리를 설치해주시면 됩니다.  
다음으로 App.js 코드를 복사 붙여넣기 해주시고, Cookie.js, Session.js 파일을 만들고 슬라이드 8페이지, 9페이지에 있는 코드도 복사 붙여넣기 해주시면 됩니다.

(슬라이드 8)   
먼저 쿠키 코드를 보자면 useCookies([“쿠키이름”]) 훅을 사용하여 기존 쿠키 저장을 하고,  
setCookies (“쿠키이름”, 쿠키에 넣을 값, 옵션)을 사용하여 쿠키에 값을 넣고, 체크가 해제될 시에는 removeCookies(“쿠키이름”, 옵션)를 사용하여 쿠키를 삭제하고,  
cookies.쿠키이름을 사용하여 쿠키를 불러오는 코드입니다.  
이제 실행시켜 보겠습니다.  
제가 수업 전에 한번 쿠키에 값을 넣었기 때문에 지금 쿠키 불러오기를 하면 제가 이전에 저장했던 값이 나오죠 이는 쿠키의 생명주기가 직접 삭제 전까지 이기 때문에 유지된 것을 볼 수 있습니다. 그리고 이제 다시 쿠키에 넣을 값을 입력하고, 체크하여 저장한 뒤 불러오면 값이 나오는 것을 확인할 수 있고 다시 체크를 해제하여 쿠키를 삭제한 뒤 불러보면 삭제된 것을 확인할 수 있습니다.

(슬라이드 9)   
다음으로 세션 코드를 보자면 set함수를 사용해 값을 저장하고 get함수를 사용해 불러오는 코드입니다.  
이제 실행시켜 보겠습니다  
저장하지 않고 그냥 읽기를 눌러보면 저장하지 않아 비어 있는 것을 확인할 수 있죠. 이제 저장하고 읽기를 눌러보면 숫자와 멤버 값이 저장되어 불러오는 것을 볼 수 있습니다.

(슬라이드 10) 이상으로 발표를 마치겠습니다.