기능 명세

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 세부 기능 설명 | 추가기능여부 |
| FR1 | 일정을 추가할 수 있습니다. | Y |
| FR2 | 일정을 삭제할 수 있습니다. | N |
| FR3 | 회원가입 후 해당 정보가 저장됩니다. | N |
| FR4 | 아이디를 중복 체크할 수 있습니다. | Y |
| FR5 | 비밀번호 입력 시, 조건에 맞는지 체크할 수 있습니다. | Y |

구현

구현 사항

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 완료여부  (O,X) | 세부사항 |
| FR1 | O | 일정을 추가할 수 있습니다. “일정을 추가하세요”라는 텍스트란을 표시해 주었습니다. 그 옆에 추가 버튼에 마우스를 가져다 대면 회색으로 색상이 변경됩니다. 일정을 추가하면 작성한 내용이 표시되며 삭제 버튼이 추가됩니다. |
| FR2 | O | 일정을 삭제할 수 있습니다. 삭제 버튼에 마우스를 가져다 대면 회색으로 색상이 변경됩니다. 삭제 버튼을 누르면 저장된 일정이 삭제가 됩니다. |
| FR3 | O | 회원가입 후 해당 정보가 저장됩니다. 가입 버튼을 클릭 시 “회원가입이 완료되었습니다” 라는 문구가 출력됩니다. |
| FR4 | O | 아이디를 중복 체크할 수 있습니다. 동일한 아이디를 입력하게 된다면 “사용자 이름이 이미 존재합니다” 라는 문구가 출력됩니다. |
| FR5 | O | 비밀번호 입력 시, 조건에 맞는지 체크할 수 있습니다. 비밀번호는 8자 이상이어야 하고 대문자, 소문자, 숫자가 최소 하나 이상을 포함되어야 합니다. 조건에 맞지 않는다면 비밀번호가 유효하지 않는다는 문구가 출력됩니다. |

웹 화면 구성

제목 : To-Do List & 회원가입

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | 1. To-Do List와 회원가입을 할 수 있는 웹페이지 입니다.  2. 일정을 추가/삭제할 수 있습니다.  3. 회원가입 하면 해당 정보가 저장됩니다.  4. 가입 시 작성한 아이디는 저장되어, 중복 체크가 가능합니다.  5. 비밀번호 입력 시 조건에 맞는지 체크할 수 있습니다. 올바르게 입력하지 않는다면 경고문이 뜹니다. |

구현 내용

HTML

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| index.html | To-Do List와 회원가입 기능을 구현한 웹 페이지를 보여줍니다. CSS 및 JavaScript 파일을 통해 페이지의 스타일과 동작을 나타낼 수 있습니다. |

CSS

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| style.css | 기본 페이지, 제목, 컨테이너, 입력 필드 등 화면의 전체적인 스타일을 구성합니다. |

JavaScript

파일 명 : script.js

|  |  |
| --- | --- |
| 함수 명 | 구현 내용 |
| function addTodo() | - 사용자가 입력한 To-Do 항목을 배열 todoList에 추가하는 역할을 한다.  - todoInput.value.trim()을 사용하여 입력 필드에 입력된 값을 가져온다. .trim()은 문자열 앞뒤의 공백을 제거하는 역할이다.  - 입력된 값이 존재하면 todoList 배열에 추가하고 입력 필드를 비웁니다.  - renderTodoList 함수를 호출하여 To-Do List 목록을 다시 그립니다. |
| function deleteTodo(index) | - 사용자가 To-Do 항목을 삭제할 때 호출되며, 지정된 인덱스의 항목을 todoList 배열에서 삭제합니다. |
| function renderTodoList() | - To-Do List 목록을 HTML로 렌더링합니다.  - for 루프를 사용하여 todoList 배열을 반복하면서 각 To-Do 항목을 처리합니다. |

파일 명 : signup.js

|  |  |
| --- | --- |
| 함수 명 | 구현 내용 |
| function isUsernameTaken(username) | - 사용자 이름이 이미 존재하는지 확인하는 역할을 합니다.  - **users** 배열 내에서 **username** 속성을 확인하여 동일한 이름이 있는지 여부를 판단합니다. |
| function isValidPassword(password) | - 사용자가 입력한 비밀번호가 유효한지 확인합니다.  - 비밀번호는 8자 이상이어야 하고 대문자, 소문자, 숫자가 각각 최소 하나 이상 포함되어야 합니다. |
| function signup() | - 사용자의 가입을 처리하는 역할을 합니다.  - 사용자 이름이 이미 존재하는지 확인하고, 존재한다면 알림 메시지를 표시합니다.  - 모든 확인 사항을 통과하면 사용자 정보를 **users** 배열에 추가하고 회원가입 완료 메시지를 표시합니다. |