|  |
| --- |
| 프로젝트 결과 보고서 |

|  |  |
| --- | --- |
| 프로젝트 명 | 냉장고를 부탁해 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 성명 | 소속 | 학년 | 학번 | 연락처 | e-mail |
| 신용준 | 컴퓨터공학과 | 2 | 21101195 | 01095176889 | 4rest528@naver.com |
| 정세영 | 컴퓨터공학과 | 2 | 21101227 | 01093139489 | jso8831@naver.com |

|  |  |
| --- | --- |
| Project 개요 및 예상결과물 |  |
| * 개요: 사용자가 보유하고 있는 식재료를 기반으로 요리할 수 있는 음식을 추천해 주는 웹 사이트 * 목표: 기본 식재료(돼지고기, 고등어, 감자, …) 목록에서 현재 가지고 있는 식재료를 선택하면 아래에 추천하는 메뉴의 목록이 나열된다. 사용자가 메뉴를 선택하면 메뉴의 간단한 레시피, 추천영상, 필요한 재료, 등을 포함하는 음식 페이지로 이동한다. | |

|  |  |
| --- | --- |
| Project 진행방법 및 절차 |  |
| * 예상 디자인  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | (메인 페이지) (오늘의 메뉴) (최근 본 메뉴) (메뉴 추가하기)  **냉장고를 부탁해**   |  | | --- | | 먹고 싶은 음식을 입력해주세요. (검색 창) |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 돼지고기 | 고등어 | 감자 | 당근 | 양파 | … |  |  |  |  |  | | --- | --- | --- | --- | | 돼지고기김치찜 | 고등어조림 | 감자볶음 | 김치볶음밥 | | 애호박나물 | 비빔밥 | 계란찜 | … |   저작권 표시 |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | (음식 페이지) 홈으로   |  |  |  | | --- | --- | --- | | 음식 사진 | 음식 이름 | | | 재료 | | | 레시피 | | 요리 영상 | |  * 가지고 있는 재료를 기반으로 음식을 추천해주고 레시피와 함께 조리영상까지 보여줌으로써 많은 사람들에게 쉽게 메뉴를 선택할 수 있는 편리함을 제공한다. * 예상업무  1. 식재료와 음식의 개수와 목록 선정하기 2. 데이터 수집하기 3. 웹 사이트 메인 페이지 제작하기 (HTML, CSS) 4. 음식페이지 제작하기 (HTML, CSS) 5. 페이지끼리 연결하기 6. 추천 알고리즘 짜기 (JAVASCRIPT) 7. 옵션(부가기능) 추가하기  * 업무분담  |  |  | | --- | --- | | 공동업무 | 1. 식재료와 음식의 개수와 목록 선정하기  2. 데이터 수집하기  3. 웹 사이트 메인 페이지 제작하기 (HTML, CSS)  7. 옵션(부가기능) 추가하기 | | 신용준 | 6. 추천 알고리즘 짜기 (JAVASCRIPT) | | 정세영 | 4. 음식페이지 제작하기 (HTML, CSS)  5. 페이지끼리 연결하기 |  * 일정  |  |  | | --- | --- | | 11.04 ~ 11.10 | 1. 식재료와 음식의 개수와 목록 선정하기 | | 11.11 ~ 11.17 | 2. 데이터 수집하기 | | 11.18 ~ 11.24 | 3. 웹 사이트 메인 페이지 제작하기 (HTML, CSS)  4. 음식페이지 제작하기 (HTML, CSS) | | 11.25 ~ 12.01 | 5. 페이지끼리 연결하기  6. 추천 알고리즘 짜기 (JAVASCRIPT) | | 12.02 ~ 12.08 | 7. 옵션(부가기능) 추가하기 | | 12.09 ~ | 마지막 점검 | | |

|  |  |
| --- | --- |
| Project  설계 |  |
| * 요구 다이어그램   텍스트, 도표, 포스트잇 노트, 스크린샷이(가) 표시된 사진  자동 생성된 설명  사이트에서 필요한 요구사항은 크게 두가지로 나눌 수 있습니다. 사용자에게 제공하는 기능(서비스)와 기능을 데이터의 관리입니다. ‘냉장고를 부탁해’에서 제공하는 서비스는 메뉴정보 열람, 오늘의 메뉴 추천, 최근 본 메뉴 열람, 메뉴 추가하기까지 현재 총 네 가지가 있습니다. 메뉴 정보 열람은 별도의 html로 분리해서 구현하였고, 메인 페이지에서 나머지 기능을 구현하였습니다.   * + 오늘의 메뉴는 날짜를 해싱하여 난수를 생성하여 메뉴를 추천합니다. 이를 통해 사용자는 매일 새로운 메뉴를 추천 받을 수 있습니다.   + 최근 본 메뉴에서는 로컬 스토리지를 활용하여 최대 n개의 메뉴를 표시합니다. n개에 메뉴가 이미 저장되어 있다면 열람한지 가장 오래된 메뉴를 리스트에서 제거합니다.   + 메뉴 추가하기에서는 form을 이용하여 사용자에게 메뉴이름, 레시피, 영상URL, 등을 입력 받습니다. 이를 food의 형태로 로컬 스토리지에 저장하여 새로운 메뉴를 기존 메뉴와 함께 표시합니다.   데이터 관리는 json과 로컬 스토리지를 사용합니다. json으로는 기존에 추가해 놓은 음식과 식재료의 데이터를 관리하고, 로컬 스토리지는 새로운 음식 데이터, 최근 본 메뉴, 등에 활용합니다.   * 파일 구조   도표, 라인, 평면도, 기술 도면이(가) 표시된 사진  자동 생성된 설명  프로젝트는 총 7개의 파일로 구성되어 있습니다. 메인 페이지인 index.html은 script.js와 style.css를 이용하여 동적 웹을 구성합니다. 음식의 상세 정보를 표시하는 food.html에서는 style표준을 맞추기 위하여 메인페이지와 동일한 css파일을 사용하고, 페이지 내의 콘텐츠 표시와 동적 컨텐츠를 위하여 food.js를 활용합니다.  또한 음식 데이터와 식재료 데이터를 json파일로 분리하여 구성하였기 때문에 언제든지 json파일에 새로운 데이터만 추가한다면 새로운 음식과 식재료를 기반으로 웹이 동작할 수 있습니다.   * 클래스 구조     식재료인 ingredients는 식재료 이름과 이미지파일 정보를 가지고 있습니다. 음식을 나타내는 food는 음식 이름, 레시피, 재료, 이미지와 동영상파일 정보를 가지고 있고, ingredients항목에서 어떤 식재료를 포함하고 있는지 따로 저장하고 있습니다. | |

|  |  |
| --- | --- |
| Project  진행내용 |  |
| * 1주차: 재료 및 요리 선정   대부분의 냉장고에 주로 남아있는 메인 재료를 우선적으로 골라 카테고리를 만들었다. 이후 대표 재료에서 만들 수 있는 음식들을 선정해 음식을 만들 때 공통된 메인 재료들이 있는지 확인해 정리했다.       * 2주차: 음식 재료와 메뉴를 보여줄 수 있는 메인 페이지를 제작했다.   메인 페이지에는 식재료를 선택하면 해당 식재료를 포함하는 음식을 골라주는 필터링 기능을 구현했고, 음식 이름을 검색하면 해당 음식을 보여주는 기능을 구현했다. 또한 식재료 선택 란에 식재료 이미지를 넣어 한눈에 재료들을 확인할 수 있게 만들었다.  텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진  자동 생성된 설명  [식재료 필터링 기능]  텍스트, 스크린샷, 소프트웨어, 컴퓨터 아이콘이(가) 표시된 사진  자동 생성된 설명   * 3주차: 음식 페이지를 제작하고, 메인 페이지의 음식 카드를 누르면 해당 음식페이지로 이동하도록 JSON을 통해 구현했다.   [음식 상세 페이지]     * 4주차: 메인 페이지 음식 카드에 음식 이미지와, 음식 이름이 이미지 위에 뜨도록 구현했다. 또한 3주차 때 만든 음식 상세페이지에 음식에 대한 정보가 나타나도록 데이터를 추가했다. 추가로 왼쪽 상단 사이트 로고를 누르면 메인 페이지로 이동하도록 만들었다.        * 5주차: 메인 페이지의 header영역에 신선해 보이는 식재료 배경 이미지를 넣어 메인 페이지에 깨끗하고 정돈된 느낌을 만들었다. 또한 네비게이션 바에 하루에 3개씩 랜덤으로 변하는 오늘의 메뉴, 최근에 봤던 음식들을 확인할 수 있는 최근 본 메뉴를 만들었다.   [헤더 영역]  [오늘의 메뉴]    [최근 본 메뉴]     * 6주차: 네비게이션 바에 메뉴 추가하기를 만들어서, 개인이 등록하고 싶은 음식, 음식 이미지, 재료, 요리방법, 영상을 추가할 수 있게 만들었다. 등록된 음식은 현재 로컬 저장소에 저장되어 음식을 등록한 사람만 볼 수 있다.   [메뉴 추가하기] | |

|  |
| --- |
| Project  수행 결과 |
| * 최종 프로젝트 결과:   [메인 페이지]    메인 페이지의 네비게이션 탭에서는 오늘의 메뉴, 최근 본 메뉴, 메뉴 추가하기 기능을 사용할 수 있다. 또한 식재료의 이름을 기반으로 검색하는 기능과 식재료가 포함되었는지 버튼을 눌러 간단하게 필터링할 수 있는 기능을 사용할 수 있다. 검색 또는 필터링의 결과로 나열되어 있는 음식 카드를 클릭하면 해당 음식의 정보를 url로 전달하여 상세 페이지에서 열람할 수 있다.  [상세 페이지]    상세페이지에서는 음식의 사진, 이름, 재료, 요리법, 요리영상을 열람할 수 있다. 홈으로 또는 왼쪽 위의 로고를 클릭하면 메인 페이지로 복귀할 수 있다. | |

|  |
| --- |
| 설계 요소  평가 |
| * **UI/UX 디자인**   + 검색 기능 및 재료 필터링이 사용자 친화적이며 직관적인 인터페이스 제공.   + 반응형 웹 디자인으로 다양한 디바이스에서 원활한 사용 가능.   + 음식 카드 디자인은 간결하면서도 시각적으로 매력적임. * **성능**   + **장점**:     - JavaScript 기반 비동기 처리로 빠른 검색과 데이터 표시.     - 로컬 스토리지를 활용한 데이터 저장으로 클라이언트에서 즉각적인 반응 제공.   + **단점**:     - 대규모 데이터를 처리하거나 동기화하는 데 한계 존재.     - 음식 데이터가 적어 필터링 결과가 제한적일 수 있음. * **안전성**   + **장점**:     - 클라이언트-사이드 로컬 스토리지 사용으로 데이터 무결성 유지.   + **단점**:     - 보안에 민감한 데이터가 저장될 경우 노출 가능성. * **보안성**   + **장점**:     - 데이터 접근이 클라이언트-사이드로 제한되어 간단한 인증이 필요 없음.   + **단점**:     - XSS 공격 등에 취약할 수 있음. 예를 들어, 사용자 입력 값에 대한 검증 부족. * **구현 단가**   + 기본 HTML/CSS와 JavaScript를 활용하여 추가적인 라이선스 비용이 발생하지 않음. | |

|  |
| --- |
| 추후 프로젝트  발전 방향 |
| * **기능확장**   + **음식 데이터 추가:**     - 음식을 카테고리별(한식, 양식, 디저트 등)로 분류하고, 다양한 레시피를 포함한 방대한 데이터를 확보.     - 사용자 메뉴 추가 기능을 통해 수집된 데이터를 분석하여 음식 목록에 자동 반영.   + **추천 시스템:**     - 최근 본 메뉴와 검색 패턴을 기반으로 한 개인화된 추천 기능 추가.   + **검색 기능 개선:**     - 실시간 검색 자동완성 및 카테고리별 필터링 추가.     - 여러 재료를 조합한 고급 검색 옵션 제공. * **성능 및 데이터 처리**   + 정적 JSON 데이터를 클라우드 기반 데이터베이스(API)로 전환하여 음식 데이터 실시간 동기화   + Lazy Loading을 도입해 대규모 데이터 로딩 시 성능 최적화. * **보안 강화**   + 사용자 입력 데이터에 대한 XSS 방지 처리.   + HTTPS 프로토콜을 사용해 데이터 전송 보호. * **확장 가능성**   + 여러 사용자 계정을 지원하기 위해 인증 및 권한 관리 기능 추가.   + 다국어 지원을 통해 사용자 기반 확장. | |

|  |
| --- |
| 종합  토의 |
| 이번 프로젝트는 사용자 중심의 음식 검색 및 관리 플랫폼으로, 제한된 데이터에서도 직관적이고 편리한 UI/UX를 통해 사용자 경험을 극대화했습니다. 특히, 검색 기능과 메뉴 추가 기능은 사용자에게 높은 활용성을 제공했습니다.  다만, 현재 음식 데이터의 개수가 적어 사용자에게 다양한 옵션을 제공하지 못하는 한계가 있습니다. 데이터 확장 및 추천 시스템 도입이 이루어진다면 사용자 참여를 높이고, 플랫폼의 실용성을 더욱 강화할 수 있습니다.  또한, 보안 강화를 통해 더 많은 사용자 데이터를 안정적으로 처리할 수 있도록 개선이 필요합니다. 이는 정적 로컬 스토리지 구조에서 클라우드 데이터베이스로 전환하는 과정에서 자연스럽게 이루어질 것입니다.  **결론**  본 프로젝트는 적은 비용과 제한된 자원으로 성공적으로 구현되었으며, 데이터 확장 및 성능 개선을 통해 음식 관리 플랫폼으로서 더 큰 잠재력을 발휘할 수 있습니다. | |

|  |
| --- |
| 첨부 |
| * 프로젝트에 관련된 자료는 모두 본 프로젝트의 레포지토리에서 확인할 수 있다.   https://github.com/Y0ngjun/Fridge\_Please | |