

2023 전공기초프로젝트2 초기 기획서 수정 1판

## 토익 단어 공부 프로그램:

워드 팡팡! *WORD PANGPANG!*

### 3204 A10팀

202211389 최준규

202211309 성은우

202211384 차민우

202211334 이동훈

202211246 강민우

# 목차

목차.....	2
1. 기획서 변경 내역.....	2
2. 용어.....	3
3. 기본사항.....	4
a. 작동 환경.....	4
b. 프로그램 구성.....	4
c. 프로그램 설치 및 실행.....	4
4. 프로그램 사용 흐름도.....	5
5. 데이터 요소.....	6
a. 영어 단어.....	6
b. 한글 뜻.....	7
c. 유저 아이디.....	8
d. 예문.....	8
e. 예문 한글 해석.....	9
6. 데이터 파일.....	10
a. 문법 규칙.....	11
i. 기본 규칙.....	11
ii. 사용자 정보 데이터 파일 규칙 및 예시.....	11
iii. 단어장 파일 규칙 및 예시.....	11
b. 의미 규칙.....	12
c. 무결성 확인 및 처리.....	13
7. 주 기능.....	13
a. 로그인.....	13
b. 메인메뉴.....	15
c. 단어장 열람/검색.....	15
d. 단어 퀴즈.....	16
e. 미니 게임.....	18

# 1. 기획서 변경 내역

- 2절 용어 “개행”, “Json 포맷”, “Local Storage” 추가
- 3.b절 프로그램 구성 csv -> json으로 변경
- 4절 사용 흐름도 로그인 화면에서 유저 아이디가 문법 규칙과 맞지 않을 때의 흐름도 추가
- 5절 오타와 json으로 바꾸면서 문법규칙에서 탭을 막을 필요가 없어졌으므로 사용자의 자유도를 위해 수정
- 6절 json으로 데이터 파일 저장구조를 바꾸면서 변경된 내용 전부 수정
- 6.b절 단어 퀴즈를 위해 단어가 10개 이상이 되어야한다는 의미 규칙 추가
- 7.c절 단어 열람 출력 방식이 기획서에 쓰인 대로 구현 시 기획서 예시 사진과 기획 의도와 다른 부분이 있어 구현 방법 수정

# 2. 용어

- **프로그램** : 이 문서에서 기획하고 있는 대상인 “토익 단어 공부” 웹사이트 프로그램
- **사용자** : 이 프로그램을 사용하려는 사람
- **데이터 요소** : 영어 단어와 한글 뜻, 유저아이디를 모두 포함하는 객체
- **데이터 파일** : 데이터 요소들이 저장되어 있는 파일
- **탭** : Tab문자를 시각적 표현을 위해 ‘→’로 표시하며 U+0009를 의미
- **개행** : 줄바꿈 문자를 시각적 표현을 위해 ‘↵’로 표시하며 각 운영체제에 따라 U+000A, U+000D을 의미
- **공백류** : 공백 문자, 탭, 줄 바꿈 등을 포함할 수 있으며 동일 단어는 동일 공백류를 가짐
- **유저 아이디** : 유저를 구분하기 위한 문자열
- **단어 정보** : 영어 단어, 한글 뜻, 예문, 예문 한글 해석을 포함하는 객체
- **영어 단어** : 단어와 구로 이루어진 뜻이 있는 영어 문자열

- **한글 뜻** : 속해 있는 단어 정보의 영어 단어를 한글로 해석한 문자열
- **예문** : 속해 있는 단어 정보의 영어 단어를 포함하는 영어 문장
- **예문 한글 해석** : 속해 있는 단어 정보의 예문을 한글로 해석한 문자열
- **최고 점수** : 사용자가 미니 게임에서 달성한 점수 중 가장 높은 점수가 저장되는 정수형 변수
- **입력 필드** : 사용자가 입력한 문자열을 받을 수 있는 칸
- **사용자 정보** : 사용자의 아이디, 최고 점수로 이루어진 객체
- **단어장** : 단어 정보들을 Row로 구분해 사전 순으로 정리한 형식
- **타이머** : 사전에 설정한 값에서 매 초 1 씩 줄어들며 갱신되는 기능
- **Json 포맷** : JavaScript 객체 문법으로 구조화된 데이터를 표현하기 위한 문자 기반의 표준 포맷
- **Local Storage** : 브라우저 상에 데이터를 저장할 수 있는 웹 스토리지

## 3. 기본사항

### a. 작동 환경

- 웹 사이트를 열 수 있는 웹 브라우저 환경이 필요합니다. 이에는 Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari 등이 포함됩니다.
- 웹 브라우저를 실행 할 수 있는 모든 운영체제 ( MS Windows, macOS, Linux 등 등 )에서 정상 작동합니다.
- 사용자의 웹 브라우저에서 JavaScript가 활성화되어야 합니다. 대부분의 웹 브라우저에서 JavaScript가 기본적으로 활성화 되어있으므로 사용자 설정을 변경할 필요는 없습니다.

## b. 프로그램 구성

- 프로그램이 배포될 때에는, 웹 사이트의 HTML파일, CSS파일, JavaScript파일, 이미지 및 미디어 파일과 ToeicWords.esvjson, Players.esv 파일들이 전부 압축되어 배포됩니다.
- 프로그램 스스로는 다른 파일을 더 생성하지 않습니다.

## c. 프로그램 설치 및 실행

기본적으로, 운영체제와 상관 없이:

- 아무 경로에나 배포된 압축파일의 압축을 풉니다.
- WordPangPang.html 파일을 더블클릭하거나, 웹 브라우저를 열고 주소 표시줄에 WordPangPang.html 파일의 로컬 파일 경로를 입력하여 실행할 수 있습니다.

## 4. 프로그램 사용 흐름도

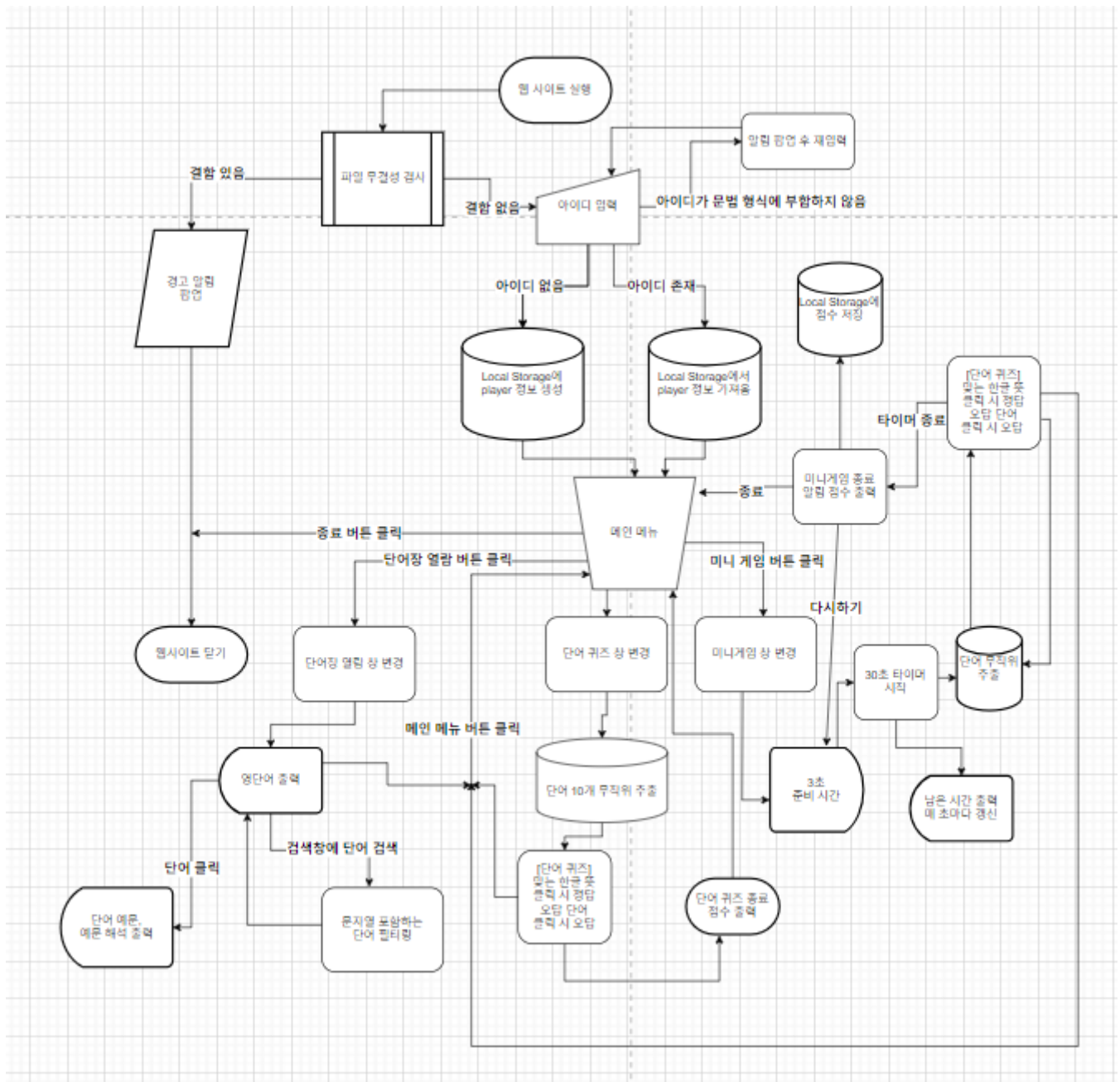


Figure 1. 프로그램 사용 흐름도

## 5. 데이터 요소

- 각 요소마다 동치 비교 규칙도 서로 다르고, ‘검색’ 검색어 합치 규칙도 서로 다르므로, 아래의 각 요소별 소절에서 해당 요소의 동치 비교 규칙, 검색어 합치 규칙을 각각 명시하겠습니다. 단, 정렬시의 순서 비교 방법은 모든 데이터 요소들이 공통적으로 ‘파일에 저장된 문자열 그대로, 사전식 순서로’ 비교됩니다.

### a. 영어 단어

- **문법 형식** : 문법적으로 올바른 영어 단어는 아래 조건들을 모두 만족시키는 문자열입니다:
  - 길이가 1 이상
  - 첫 문자와 마지막 문자는 실상 문자
  - 탭도 개행도 **이** 전혀 들어가지 않음
  - 영어로만 이루어짐
- 예를 들어, (따옴표 없이) “abolish”, “be subject to”, “A”는 모두 문법적으로 올바른 영어 단어입니다.
- **의미 규칙** : 영어 단어에는 아무런 추가적인 의미 규칙도 없습니다. (의도한 모든 규칙을 문법으로 전부 표현했습니다.)
- **동치 비교** : 두 영어 단어 간의 동치 비교를 할 때에는, 두 문자열 자체가 서로 대소문자만 구분하지 않고, 내부의 공백류들까지 완전히 일치해야만 같은 **어름은영어 단어**로 간주합니다.
- 예를 들어, “abolish” 와 “Abolish”, “aBoliSh” 셋 모두 같은 영어 단어이고, “be \_ subject \_ to”와 “be \_ \_ subject \_ to” 는 서로 다른 영어 단어입니다.
- **검색** : 단어 검색 필드의 인자 각각을 검색어로 삼아 파일에 저장된 영어 단어 중에서 찾을 때에는 :
  - 저장된 **어름영어 단어** 속의 공백류들을 전부 무시한 채로 (공백류 없이 딱 붙어있는 것으로 간주하고)
  - 대/소문자를 구분하지 않으면서
  - 검색어가 이름의 부분문자열이면 합치된 것으로 간주합니다.

- 예를 들어 검색어가 “organize”일 때, 파일에 저장된 영어 단어들 중 “organize” 뿐만 아니라 “reorganize”까지 이 검색어에 합치됩니다.
- 또한, 검색어가 “er”일 경우, 저장된 단어들 중 “order”와 “be ready to”도 이 검색어에 합치됩니다.

## b. 한글 뜻

- **문법 형식** : 문법적으로 올바른 한글 뜻은 아래 조건들에 모두 부합하는 문자열입니다.
  - 길이가 1 이상
  - 첫 문자와 마지막 문자는 실상 문자
  - 탭도 개행도 이 전혀 들어가지 않음
  - 한글과 기호로 이루어짐
- 예를 들어, (따옴표 없이) “폐지하다”, “~을 수용하다”, “정확한” 세 문자열 모두 문법적으로 올바른 한글 뜻입니다.
- **의미 규칙** : 한글 뜻에는 아무런 추가적인 의미 규칙도 없습니다.
- **동치 비교** : 두 한글 뜻 간의 동치 비교를 할 때에는, 두 문자열 전체가 (내부의 공백류들까지 포함해서) 서로 완전히 일치해야만 같은 이름으로 간주합니다.
- 예를 들어, (따옴표 없이) “진보, 전진”, “진보, 전진”, “진보, 전진”은 셋 모두 서로 다른 한글 뜻이고, “찬사, 호평”, “찬사, 칭찬”은 서로 다른 한글 뜻입니다.
- **검색** : 단어 검색 필드의 인자 각각을 검색어로 삼아 파일에 저장된 영어 단어 한글 뜻 중에서 찾을 때에는 :
  - 저장된 이름 한글 뜻 속의 공백류들을 전부 무시한 채로 (공백류 없이 딱 붙어있는 것으로 간주하고)
  - 검색어가 이름의 부분문자열이면 합치된 것으로 간주합니다.
- 예를 들어, 검색어가 “정하”일 때, 파일에 저장된 한글 뜻 중에서 “배정하다”와 “인정하다”가 이 검색어에 합치됩니다.



## c. 유저 아이디

- **문법 형식** : 문법적으로 올바른 유저 아이디는 아래 조건들에 모두 부합하는 문자열입니다.
  - a. 길이가 3 이상 16 이하
  - b. 첫 문자와 마지막 문자는 실상 문자
  - c. 탭도 개행도 전혀 들어가있지 않음
  - d. 영어와 숫자로만 이루어짐
- 예를 들어, (따옴표 없이) “wnsrb9890”, “nm1n0”, “d10gh0n” 세 문자열 모두 문법적으로 올바른 유저 아이디이지만, “\_wnsrb9890”, “wnsrb9890\_”, “wn\_srb9890”, “nm!n0”은 모두 문법적으로 올바르지 않은 유저 아이디입니다.
- **의미 규칙** : 유저 아이디에는 아무런 추가 의미 규칙도 없습니다.
- **동치 비교** : 두 유저 아이디간의 동치 비교를 할 때에는, 두 문자열 전체가 서로 완전히 일치해야만 같은 유저 아이디로 간주합니다.
- 예를 들어, (따옴표 없이) “nm1n0”과 “nm1n01”, “nm11n0”은 셋 모두 다른 유저아이디입니다.
- **검색** : 유저 아이디는 ‘검색’ 기능과 관련이 없습니다.

## d. 예문

- **문법 형식** : 문법적으로 올바른 예문은 아래 조건들에 모두 부합하는 문자열입니다.
  - a. 길이가 1 이상
  - b. 탭도 개행도 **아** 전혀 들어가있지 않음
  - c. 첫 문자와 마지막 문자는 실상 문자
  - d. 영어와 숫자, 기호로만 이루어짐
- 예를 들어, (따옴표 없이) “The \_government \_decided \_to \_abolish \_the \_outdated \_law”는 문법적으로 올바른 예문이지만, “The \_government ↵

decided \_to \_abolish \_the \_outdated \_law”,

“정부는 \_오래된 \_법률을 \_폐지하기로 \_결정했습니다.” 은 모두 문법적으로 올바르지 않은 예문입니다.

- **의미 규칙** : 예문에는 아무런 추가 의미 규칙도 없습니다.
- **동치 비교** : 예문은 동치 비교를 하는 경우가 없으므로, 생략합니다.
- **검색** : 예문은 ‘검색’ 기능과 관련이 없습니다.

## e. 예문 한글 해석

- **문법 형식** : 문법적으로 올바른 예문 한글 해석은 아래 조건들에 모두 부합하는 예문 한글 해석입니다.
  - a. 길이가 1 이상
  - b. ~~탐도~~개행도이 전혀 들어가있지 않음
  - c. 첫 문자와 마지막 문자는 실상 문자
  - d. 한글과 숫자, 기호로만 이루어짐
- 예를 들어, (따옴표 없이) “정부는 \_오래된 \_법률을 \_폐지하기로 \_결정했습니다.”는 문법적으로 올바른 예문 한글 해석이지만, “정부는 ◀ 오래된 \_법률을 \_폐지하기로 \_결정했습니다.”, “The \_government \_decided \_to \_abolish \_the \_outdated \_law” 는 모두 문법적으로 올바르지 않은 예문 한글 해석입니다.
- **의미 규칙** : 예문 한글 해석에는 아무런 추가 의미 규칙도 없습니다.
- **동치 비교** : 예문 한글 해석은 동치 비교를 하는 경우가 없으므로, 생략합니다.
- **검색** : 예문은 ‘검색’ 기능과 관련이 없습니다.

## 6. 데이터 파일

- 기본적으로, 이 프로그램은 실행 중에 사용자로부터 유저 아이디만을 TextField 입력으로 받아들이고 이 정보들을 프로그램 스스로 내부 기능을 통해 데이터 파일에 저장합니다.
- 프로그램 안에는 입력기능은 없지만, 사용자의 게임 기록들과 영어 단어, 한글 뜻, 예문, 예문 해석본이 데이터 파일에 저장되어 있습니다.
- 사용자가 메모장 등 **별도의 텍스트 편집기**를 이용해서 데이터 파일을 **직접 생성/편집/저장**하는 방식도 지원합니다. 단:
  - 프로그램이 실행되고 있는 중에 데이터 파일을 직접 편집할 경우, 정상 작동을 보장하지 않습니다.
  - 데이터 파일은 (BOM 없는) UTF-8 인코딩으로 저장되어야 합니다.
  - 운영체제마다 개행 문자들을 다루는 기본 방식이 서로 다르지만 (LF, CRLF 등) 이 프로그램은 모든 개행 방식을 똑같이 다룹니다. 즉, 데이터 파일을 어떤 운영체제에서 어떤 개행방식으로 작성/저장해도 상관 없습니다.
  - 이 절에서 정하는 문법과 의미 규칙을 준수하며 편집해야만 프로그램이 정상적으로 동작합니다.
  - 브라우저의 **LocalStorage**파일은 브라우저별로 설정된 경로와 방법으로 저장이 되므로, 일반적인 방법으론 직접 생성/편집/저장할 수 없습니다.
  - **LocalStorage**파일은 브라우저마다 다르기 때문에, 브라우저가 달라지면 **LocalStorage**에 저장된 파일의 내용도 달라집니다.
  - **LocalStorage**파일은 브라우저에서 캐시 취급이 되기 때문에, 캐시를 삭제하여 초기화 할 수 있습니다.
- 사용자는 문법에만 맞다면 얼마든지 데이터 파일을 수정해도 되지만, 데이터의 무결성을 보장할 수 없으므로 권장하지는 않습니다.
- 위 마지막 항목에 의거하여, 사용자가 데이터 파일을 올바르게 편집하고 어떤 상황에서 어떤 결과가 나올 지 정확히 예상할 수 있도록 데이터 파일의 문법(형식)과 의미 규칙(추가 조건)들을 명시합니다.

## a. 문법 규칙

### i. 기본 규칙

- 기본적으로 데이터 파일은 ~~Tab Delimited Value~~ **Json 포맷** 형식으로 작성되며, ~~Column은 Tab문자~~ **1개로, Row는 개행으로 구분합니다.** 로컬 Json파일의 경우, JavaScript에서 로드 할 수 있도록, 제일 앞에 “변수명 =”을 붙여주어야 합니다.
- 각각의 ~~Row는 각~~ **Json 객체들은** 각 Field와 ~~Tab문자를 번갈아가며 입력되며로 이루어 지며,~~ Field는 비어있을 수 없습니다.

### ii. 사용자 정보 데이터 파일 규칙 및 예시

- 사용자 정보 데이터 파일은 브라우저의 LocalStorage파일에 저장됩니다.
- 사용자 정보 데이터 파일에 저장될 Field들은 다음과 같습니다.
  - a. 유저 아이디
  - b. 최고 점수
- 각각의 항목들은 위 순서대로 Tab문자로 구분되어 한 줄로 객체로 이루어져 있으며, 파일은 이 객체들의 배열인 Json파일로 저장이 되므로 다음의 형식으로 구성되어야 합니다.
- [사용자 정보1, 사용자 정보2, 사용자 정보3 ... ]
- 각 사용자 정보들의 형식은 다음과 같습니다.
- { “ID” : “{유저 아이디}” , “Score” : “{최고 점수}” }
- 예를 들어, 아이디가 ‘wnsrb9890’이고, 최고 점수가 ‘10’인 사용자의 정보는 다음과 같이 변환되어 저장됩니다.
  - a. ~~wnsrb9890=10~~ { “ID” : “wnsrb9890” , “Score” : “10” }

### iii. 단어장 파일 규칙 및 예시

- 단어장 파일에 저장될 Field들은 다음과 같습니다.
  - a. 영어 단어

b. 한글 뜻

c. 예문

d. 예문 한글 해석

- 각각의 항목들은 위 순서대로 Tab문자로 구분되어 한 줄 Json 포맷 형식으로 구성되어야 합니다.
- 각 단어들은 객체로 저장되기 때문에, 단어장 파일은 객체들의 배열의 Json 파일이되며, 위에서 서술했듯, JavaScript가 이 Json파일을 읽기 위해선 다음의 형식으로 이루어져야 합니다.
- ToiecWord = { "Words" : [ 단어1, 단어2, 단어3 ..... ] }
- 각 단어들의 형식은 다음과 같습니다.
- { "English" : "{영어 단어}" "Korean" : "{한글 뜻}" "ExampleEng" : "{예문}" "ExampleKor" : "{예문 한글 해석}" }
- 예를 들어, 영어 단어가 'abolish'이고, 한글 뜻이 '폐지하다'이고, 예문이 'The government decided to abolish the outdated law', 예문 한글 해석이 '정부는 오래된 법률을 폐지하기로 결정했습니다.'라면 단어는 다음과 같이 변환되어 저장됩니다.
- {  
     "English": "abolish",  
     "Korean": "폐지하다",  
     "ExampleEng": "The government decided to abolish the outdated law.",  
     "ExampleKor": "정부는 오래된 법률을 폐지하기로 결정했습니다." }

## b. 의미 규칙

- 사용자 정보 데이터 파일 속의 서로 다른 두 Row들 속에는 서로 같은 '유저 아이디'가 존재할 수 없습니다.

- 단어장 파일속의 서로 다른 두 Row들 속에는 서로 같은 ‘영어 단어’가 존재할 수 없습니다.
- 단어의 개수는 10개 이상이어야 합니다.

## c. 무결성 확인 및 처리

- 프로그램 시작 시 파일을 읽어 사용자 정보와 단어장을 불러오는 데 실패한다면 실패 알림 팝업을 띄우고 프로그램을 종료합니다.

# 7. 주 기능

## a. 로그인

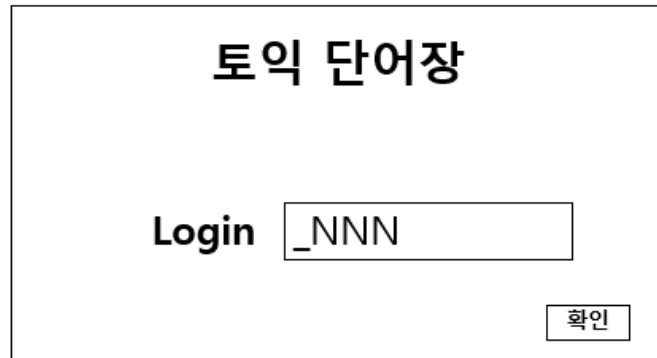
- 로그인은 기본적으로 별도의 회원가입 없이 진행됩니다.
- 프로그램 실행 시 로그인 화면에서 아이디 입력 필드에 아이디를 입력받습니다.

토익 단어장

Login

Figure 2. 로그인 화면 예시

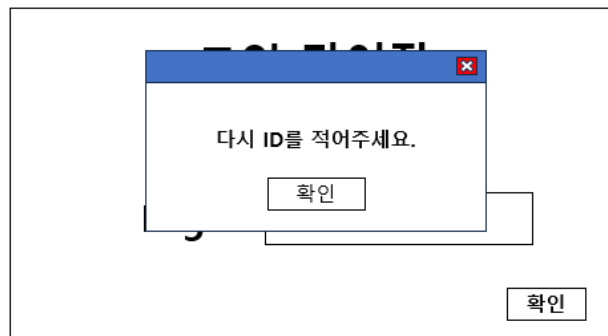
- 문법 형식에 어긋난 아이디라면 경고 문구 알림 팝업 후 아이디 입력 필드가 초기화됩니다.



토익 단어장

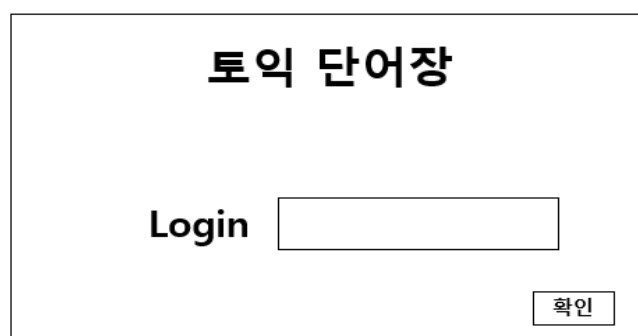
Login

Figure 3. 어긋난 아이디 예시\_1



다시 ID를 적어주세요.

Figure 4. 어긋난 아이디 예시\_2



토익 단어장

Login

Figure 5. 어긋난 아이디 예시\_3

- 입력된 아이디가 사용자 정보 데이터에 존재한다면, 그 사용자 정보를 불러오고, 존재하지 않는다면 사용자 정보 데이터에 입력된 아이디를 새로 추가한 후 불러옵니다. 사용자가 새로 추가될 때, 미니 게임 최고 점수는 0으로 저장됩니다.

## b. 메인메뉴

- 메뉴의 목록으로는 “단어 열람”, “단어 퀴즈”, “미니 게임”, “종료”가 있습니다.

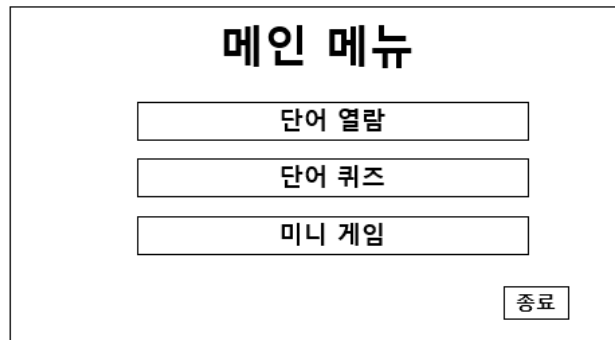


Figure 6. 메인 메뉴 예시

- 각 메뉴는 버튼을 클릭함으로써 실행됩니다.

## c. 단어장 열람/검색

- 데이터 파일에 저장되어 있는 단어 정보들을 (따옴표 없이) “영어 단어[공백] (+16자가 되도록 공백 추가 후)→한글 뜻”형식으로 영어 단어와 공백이 “16em” 만큼의 너비를 차지하게 보여줍니다. 각 단어 정보들은 행으로 구분되며, 사전 순으로 정렬되어 나타납니다.

●

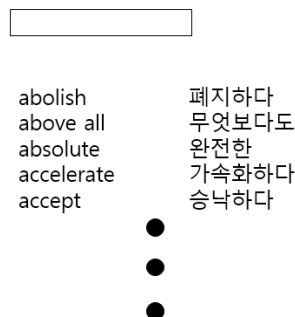


Figure 7. 단어장 열람 예시

- 사용자가 단어 정보의 영어 단어와 한글 뜻이 쓰여진 행을 클릭할 경우, 아래에 해당 영어 단어를 활용한 예문과 예문 해석이 출력됩니다. 다시 클릭하면 예문과 예문 해석은 사라집니다.



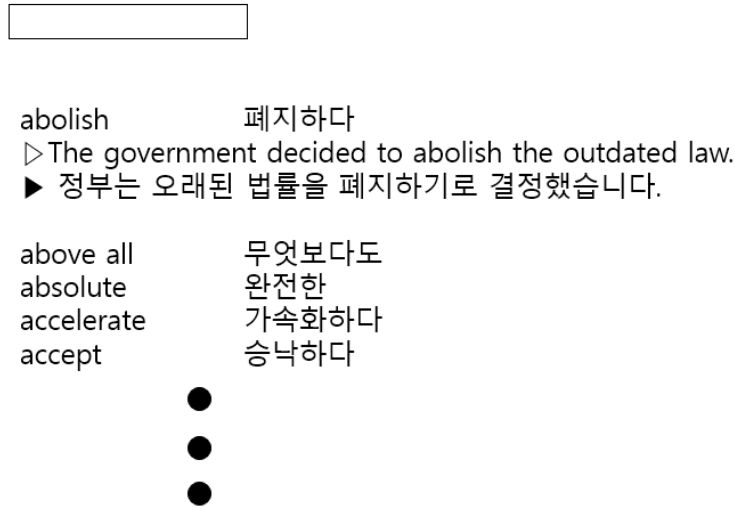


Figure 8. 단어장 예문 출력 예시

- 사용자가 검색창에 텍스트를 입력할 경우, 입력한 텍스트가 영어 단어와 한글 뜻 문자열에 포함되거나 일치하는 단어 정보들만을 보여줍니다. 보여지는 형식은 같습니다. **출력되던 예문들은 전부 사라집니다.**

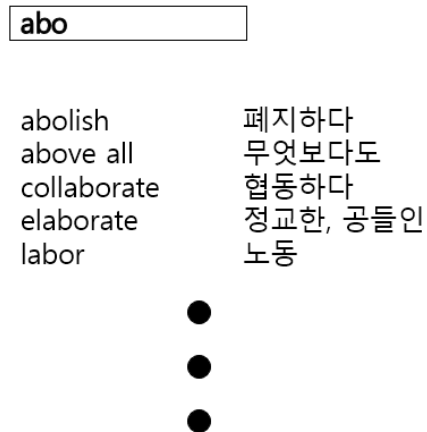


Figure 9. 단어장 검색 예시

## d. 단어 퀴즈

- 단어 퀴즈를 실행하면 바로 단어 퀴즈 창으로 넘어가 퀴즈가 시작됩니다.
- 데이터 파일에 저장되어 있는 단어 정보들 중 무작위로 단어 정보들을 10개 추출합니다.
- 추출한 단어 정보들의 “영어 단어”들을 차례로 보여주며 4개의 선지 중 적합한 “한글 뜻”을 선택하도록 사용자에게 문제를 냅니다. 모든 선택은 버튼으로 동작합니다.

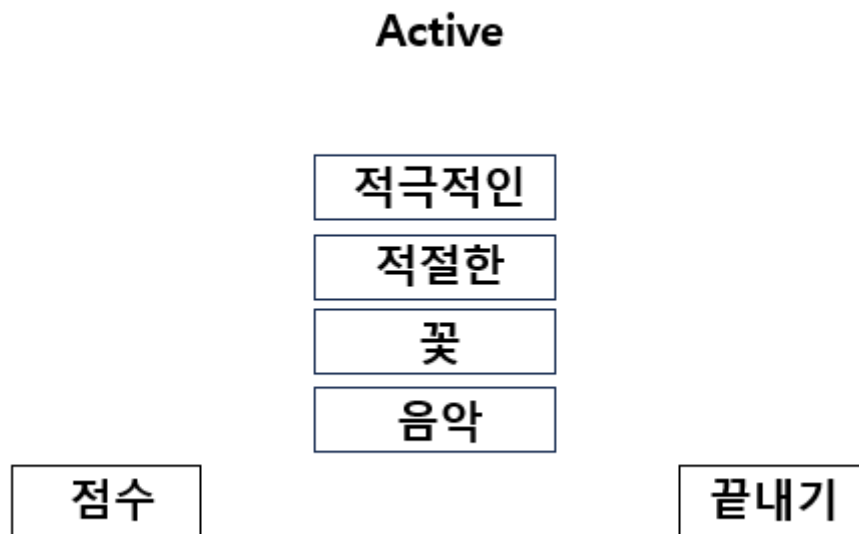


Figure 10. 단어 퀴즈 예시

- 적합한 “한글 뜻”을 선택했다면 점수를 10점씩 추가합니다(시작 점수는 0점입니다) . 적합함의 기준은 문제의 정답인 단어 정보의 “한글 뜻”의 “동치 비교” 방식을 기준으로 합니다.
- 올바른 선지를 정답으로 선택 시 해당 선지가 초록색으로 보이게 하고, 만약 틀린 선지를 고를 시에는 고른 선지를 빨간색으로 보이게 한 후, 정답을 초록색으로 보이도록 합니다.



Figure 12. 단어 퀴즈 정답,오답 예시

- 사용자가 끝내기 버튼을 누르거나, 10개의 문제들을 전부 풀면 퀴즈는 종료됩니다.
- 퀴즈가 종료된 후에는 점수를 화면에 팝업하고, 메인 메뉴로 돌아갑니다.

Score : 60

Figure 13. 점수 팝업 예시

## e. 미니 게임

- 미니 게임을 실행하면 게임 시작 버튼이 있는 화면으로 넘어가고, 게임 시작 버튼을 누르는 순간 3초의 기다림 후 미니 게임이 시작됩니다. 화면 중앙에 사용자의 미니 게임 최고 점수가 출력됩니다.

# Word PangPang

High score : 50

Game Start

Figure 14. 미니 게임 시작 화면 예시

- 게임의 형식은 퀴즈와 동일하지만 타이머 기능이 추가되어 주어진 시간(30초)동안 누가 더 많은 문제를 맞추는지를 겨룹니다. (따라서, 중복 없이 랜덤으로 단어 정보들이 계속 추출이 됩니다.)
- 만약 30초 안에 모든 단어 정보들의 문제들을 풀었다면, 미니게임은 종료됩니다. (따라서, 문제들의 최대 개수는 단어장 파일에 저장된 Row의 개수가 됩니다.)
- 게임이 시작되면 위에 화면 위에 남은 시간이 초 단위로 표시됩니다.

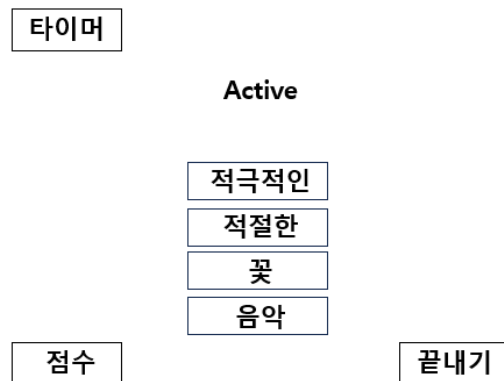


Figure 15. 미니 게임 화면 예시

- 점수는 정답 수 \* 10이며 이는 게임이 끝난 후 출력되고, 사용자들의 최고 점수가 함께 등수를 매겨져 상위 3명만 출력됩니다. (따라서 미니 게임의 최대 점수는 단어장 파일에 저장된 Row의 개수 \* 10 이 됩니다)

Rank 1 )	wnsrb9890	120
Rank 2 )	amdifj	100
Rank 3 )	mn139	80

My Score : 60  
My Hight Score : 60



Figure 16. 미니 게임 종료 예시

- 만약 동점자가 있을 경우에도 3명만 출력되기에, 유저 아이디의 사전 순 정렬을 통해 같은 순위에서도 순서가 정해지고 3위 이상의 점수라도 출력이 되지 않을 수 있습니다.

Rank 1 ) banana	120
Rank 1 ) cat134	120
Rank 1 ) delight12	120

My Score : 100  
My Hight Score : 100

다시하기

종료

Figure 17. 동점자 출력 예시

- 최종 점수가 사용자 정보 데이터에 사용자 ID의 최고 점수보다 높다면 사용자 ID의 최고 점수를 갱신하고 사용자 정보 데이터에 저장합니다.
- 하단 버튼을 통해 다시할 지 메인화면으로 이동할 지 정할 수 있습니다.
- 게임이 종료되지 않았을 때 메인화면으로 돌아가거나 비정상적으로 종료될 경우, 점수는 계산되지 않습니다. (즉, 최고 점수를 달성했더라도, 게임이 정상적으로 종료되지 않은 경우 점수가 갱신되지 않습니다.)