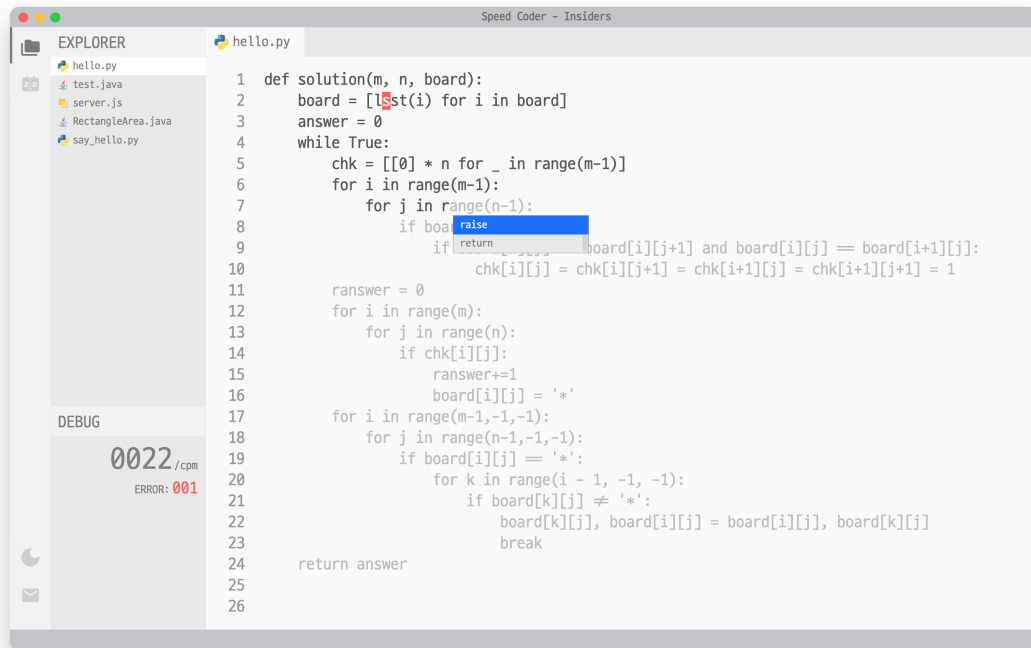


2022 GUI 프로그래밍 기말과제 보고서

분반:	1	학번:	20190602	이름:	설진영
-----	---	-----	----------	-----	-----

Speed Coder - 개발자를 위한 타자연습



요 약

개발자들을 위한 타자연습 사이트이다. 사용자는 여러 종류의 코드들을 따라 입력하며 자신의 타자속도를 측정하고, 리더보드에 자신의 타자속도를 등록하여 다른 사용자와 경쟁할 수 있다. 사용자 경험 향상을 위해 직관적이고 깔끔한 UI를 구성하려 노력하였으며 이를 위해 Adobe XD 프로토타이핑 툴을 사용해 디자인 및 인터페이스 설계를 진행하였다. 불필요한 렌더링을 최소화하는 방식으로 최적화를 진행해 낮은 사양의 컴퓨터에서도 불편함 없이 이용할 수 있도록 노력했다. 사용자의 입력을 적절히 처리하여 틀린 글자를 표시하고, 자동완성 문구를 추천한다. 사용자의 입력을 어떻게 처리할 것인지가 이번 프로젝트의 주된 고민거리였는데, 화면에 나타나지 않는 숨겨진 TextArea를 생성하고 여기로 들어온 입력을 적절히 처리하는 방식으로 구현하였다. 사용자 입력 길이를 지속적으로 검사해 타자속도를 실시간으로 확인할 수 있도록 하였다. 사용자의 입력이 완료되면(코드를 모두 작성하면) 자신의 타자속도와 함께 이름과 메시지를 입력할 수 있는 창이 생성되고, Push 버튼을 눌러 리더보드에 등록할 수 있다. 사이드바에서 리더보드 버튼을 클릭하면 기존 Explorer창과 Debug창이 리더보드 창으로 바뀌며, 여러 사용자들이 등록한 타자속도를 순위별로(타자속도 내림차순) 확인할 수 있다. 많은 양의 콘텐츠를 표현하는 방법에는 페이지네이션과 무한 스크롤이 있는데 리더보드에는 무한 스크롤 방식을 적용하는게 알맞다고 판단하였고, 마지막 요소가 노출되는 즉시 다음 페이지의 순위를 요청하는 API를 호출하도록 하였다. 해당 사이트는 Oracle Cloud를 통해 호스팅하였고, 다음 링크를 통해 접속할 수 있다. 사이트 주소: <http://144.24.64.251/>

1. 개발배경

1.1 개발 동기

우리 같은 개발자들은 키보드와 매우 가깝게 지냅니다. 키보드는 컴퓨터와 소통하는 좋은 수단 중 하나고, 우리는 키보드를 통해 코드를 짜며 자신의 상상을 현실로 만듭니다. 그럼 개발자는 타자속도가 빨라야 할까요? 결론부터 말하자면, 꼭 그렇진 않습니다. 개발자에게는 프로그램 설계나 디버깅 등 다양한 업무가 있고 코드를 작성하는 시간은 생각보다 많지 않기 때문입니다. 하지만 개발자가 타자속도가 빠르다면 장점이 많습니다. 타자속도가 빠르다는 것은 자신이 말하고자 하는 내용을 빠르고 정확하게 컴퓨터에 전달한다는 것과 같습니다. 타자가 빠른 개발자는 다른 개발자에 비해 빠르게 이해한 내용을 실습할 수 있어 다양한 기술을 학습하는 데 있어 유리합니다. 또한, 머릿속에 번뜩 떠오른 아이디어를 빠르게 코드로 풀어낼 수도 있습니다.

타자속도의 중요성을 알아봤다면 이제 타속을 늘리기 위해 타자연습을 해야 합니다. 프로그래밍 코드는 대부분 영문으로 이루어져 있으므로 영문 타자를 연습하면 됩니다. 하지만 프로그래밍 코드에는 알파벳뿐만 아니라 다양한 숫자 및 특수기호가 존재하고, 언어별로 제각각의 코딩 규칙(컨벤션)들 또한 존재합니다. 저는 이러한 프로그래밍 코드들의 특성을 기존에 존재하는 타자연습 프로그램에 반영하고자 하였습니다. 다양한 프로그래밍 언어로 이루어진 잘 짜인 코드들을 따라 쳐보며 타자연습을 하고, 실시간으로 자신의 타자속도를 확인할 수 있도록 하고자 합니다. 오직 개발자들만을 위한 타자연습 사이트를 만드는 것입니다.

1.2 유사 프로그램 소개 (또는 관련 기술)

관련 기술 소개

본 프로젝트는 웹 개발을 통해 구현하려고 합니다. 이유는 다음과 같습니다. 우선 현존하는 거의 모든 브라우저가 웹 표준을 준수하고 있기 때문에, 각기 다른 운영체제와 브라우저에서도 동일한 화면과 기능을 나타낼 수 있습니다. (크로스 플랫폼 지원) 또한, 별도의 프로그램 설치나 설정이 필요하지 않기 때문에 빠르게 여러 사용자에게 배포할 수 있습니다.

웹 프론트엔드 개발도 역시 GUI 프로그래밍의 일종이라 할 수 있습니다. Javascript를 통해 동적 웹 사이트를 개발한다는 전제하에 윈도우를 생성하고 이벤트를 처리한다는 기본 원리를 공유합니다. 또한, Javascript 역시 사건 기반 프로그래밍(Event-Driven Programming) 방식의 프로그래밍 언어입니다. 일반적인 동적 웹 사이트 개발에서 웹 페이지 전체를 하나의 윈도우로 보거나 html 태그의 일정단위를 하나의 윈도우로 볼 수 있습니다. Javascript의 `addEventListener` 와 같은 함수를 통해 특정 태그(윈도우)에서 발생하는 이벤트를 처리할 수 있습니다.

이번 프로젝트에서는 프론트엔드 웹 프레임워크 중 하나인 React를 사용합니다. React는 사용자 인터페이스 구성을 위한 Javascript 웹 프레임워크입니다. React에서는 컴포넌트(Component)라는 하나의 렌더링 단위를 통해 웹 페이지를 구성하게 되는데, 이 컴포넌트를 윈도우 프로그래밍 관점에서 하나의 윈도우로 볼 수 있습니다. 각각의 컴포넌트는 함수를 통해 정의합니다. 함수의 반환 값은 생성할 윈도우에 대한 html태그들로 구성되며, 함수의 내용으로는 이 윈도우에 대한 이벤트 처리를 담당하는 각종 Hooks와 처리 함수들이 정의됩니다.

웹 백엔드는 Node.js를 통해 구성하려고 합니다. Node.js는 Chrome V8 Javascript 엔진으로 빌드된 Javascript 런타임입니다. Javascript 런타임이란 Javascript가 구동되는 환경을 말합니다. 일반적으로 Javascript는 Chrome과 같은 웹 브라우저에서 구동되는데, 이를 브라우저를 사용하지 않고 실행할 수 있도록 한 것이 Node.js 입니다. 이번 프로젝트에서는 Node.js 환경에서 Express.js 라는 웹 프레임워크를 활용해 백엔드를 구성할 예정입니다. 리더보드를 위한 데이터베이스로는 MongoDB 시스템을 사용할 예정입니다. 백엔드 시스템과 데이터베이스는 단순히 사용자의 점수를 리더보드에 등록하고 조회하기 위한 정도로만 구성합니다.

프론트엔드 개발에 초점을 맞춘 프로젝트이기 때문에, 백엔드 기술과 관련된 자세한 설명은 생략하도록 하겠습니다.

유사 프로그램

타자연습을 할 수 있는 프로그램이나 웹 사이트는 많습니다. 대표적으로 typing.works와 한컴 타자연습 두 사이트가 있습니다.

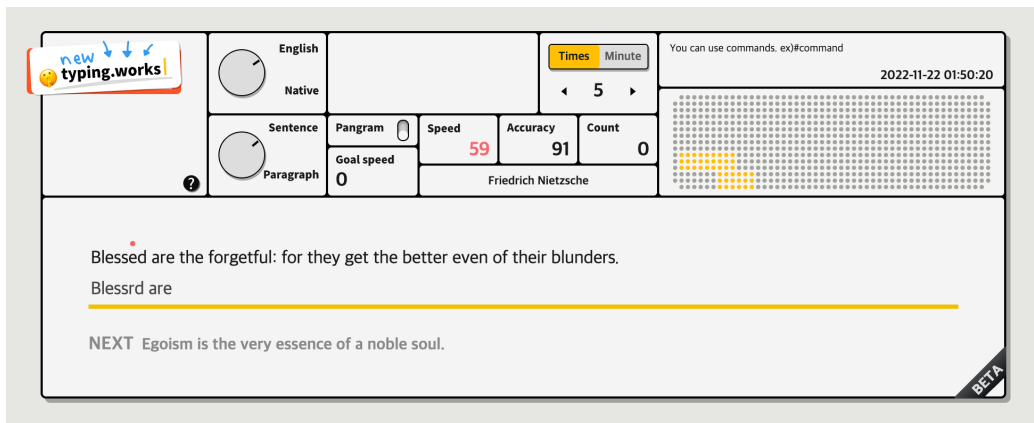
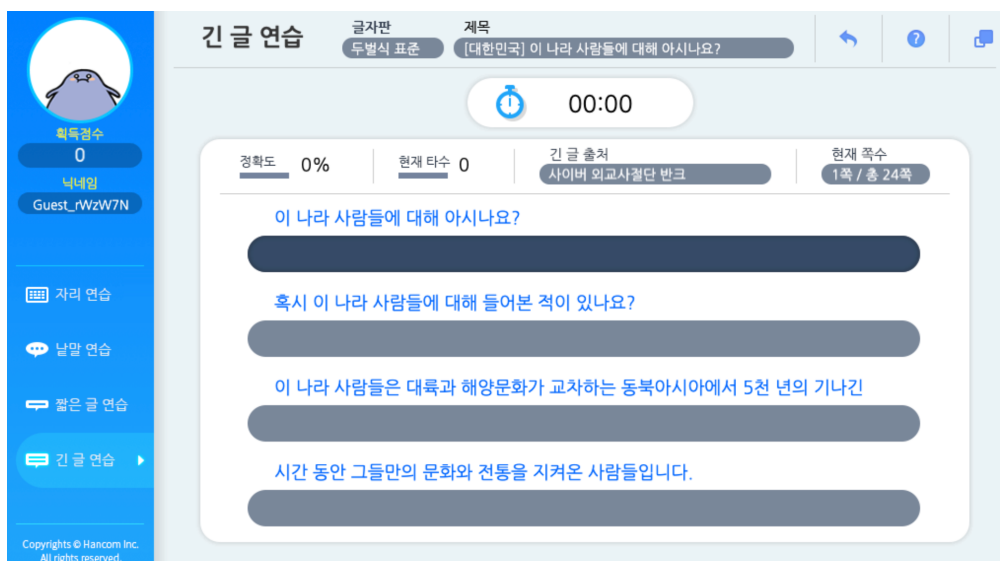


그림 2 typing.works 사이트 화면

typing.works는 깔끔한 UI를 바탕으로 문장 단위로 타자연습을 할 수 있는 기능을 제공합니다. 타자속도와 정확도를 실시간으로 확인할 수 있으며, 글자가 틀렸을 경우, 빨간색 점을 통해 틀렸음을 표시합니다.



한컴 타자연습은 비교적 친숙한 UI를 통해 자리 연습, 긴 글 연습 등 단계적으로 사용자의 수준에 맞는 타자연습 기능을 제공합니다. 이러한 특징 덕에, 주로 타자를 처음 접하는 사용자들을 교육하는데 사용됩니다. 두 프로그램 모두 한글 및 영문 타자연습을 지원하고, 타자속도와 정확도를 실시간으로 표시해 주는 것을 볼 수 있습니다. 또한, 여러 종류의 문장과 글을 바꿔가면서 연습할 수 있습니다.

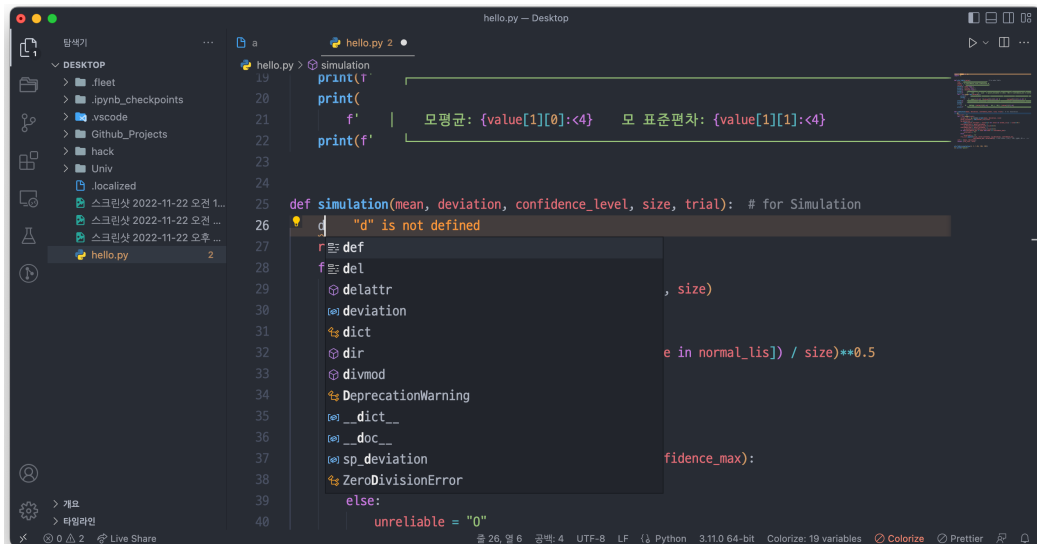


그림 4 Visual Studio Code 화면

Visual Studio Code는 개발자들이 가장 많이 사용하는 텍스트 에디터 중 하나입니다. 좌측 사이드바에 파일들의 목록을 출력하고, 오른쪽 화면에서 소스코드를 편집할 수 있습니다. 소스코드를 입력할 때 소스코드의 확장자에 맞게 해당하는 자동완성 단어를 추천해 주는 창을 띄워 주는 것을 볼 수 있습니다.

이번 프로젝트에서는 개발자들에게 친숙한 텍스트 에디터 및 IDE를 모티브로 하여 이들과 유사한 사용자 경험을 할 수 있는 새로운 타자연습 프로그램을 만들려고 합니다. 좌측 사이드바에서는 자신이 타자연습을 진행할 소스코드 파일을 고를 수 있고, 자신의 타자속도 및 정확도를 확인할 수 있으며, 자신의 타자속도가 어느 정도 수준인지를 리더보드를 통해 나타낼 수 있도록 하려 합니다.

2. 개발목표 및 내용

2.1 개발목표

이번 프로젝트의 목표는 개발자만을 위한 타자 연습 사이트를 제작하는 것입니다. 여러 종류의 코드들을 따라 입력하면서 자신의 타자속도를 측정하고, 결과를 리더보드에 등록할 수 있도록 합니다. 사용자 인터페이스는 앞서 언급한 것과 같이 개발자들에게 친숙한 IDE 혹은 텍스트 에디터의 모습을 바탕으로 합니다. 완성도 높은 프로젝트를 위해 프로젝트 핵심 요소 두 가지를 정의해 보았습니다.

첫번째, 써 보고 싶게 만드는 것입니다. 직관적이고 깔끔한 UI를 적용하고, 사용자가 기대할 수 있는 적절한 상호작용을 제공함으로써 사용자가 해당 프로그램을 사용해보고 싶도록 해야 합니다. 사용자가 불편한 경험을 하지 않도록 성능 또한 개선해야 합니다.

두 번째로는, 최적화를 통해 성능을 향상시키는 것입니다. React는 컴포넌트(윈도우) 단위로 값의 변경을 추적해

화면을 새로 그립니다. 변수값 변경이 잦은 (ex. 실시간 타자속도 표시, 사용자 입력 부분) 프로젝트 특성상, 컴포넌트(윈도우)를 제대로 나누지 않거나 불필요한 변수 업데이트 등의 이유로 프로그램이 성능이 하락할 가능성이 있습니다. 프로그램 구현 시 이러한 점을 고려해야 합니다.

2.2 개발내용

개발할 내용은 과정과 순서를 명확하게 하기 위해 개조식으로 작성하였습니다.

개발 내용		설명	관련 기술
UI 디자인 설계 및 개발	사이드바의 작업 막대	탐색기 버튼, 리더보드 버튼, 다크모드 버튼 등	html, css, javascript. React
	사이드바의 탐색기	연습할 소스코드 목록 및 타자속도와 정확도를 표시하는 화면 리더보드 등을 배치, 사이드바를 통해 전환 가능	
	편집기 화면	실제로 입력한 소스코드가 나타나는 화면, 이미 입력한 글자와 틀린 글자를 확인할 수 있도록 표시 해야함	
	결과 화면	입력이 완료되었을 시 나타나는 화면, 리더보드에 등록할 사용자 이름과 메시지를 입력받는 입력창이 있어야 함	
	자동완성 화면	사용자 입력을 기반으로 자동 완성 단어를 나타내는 화면, 단어가 존재 시 커서 바로 옆에 위치됨	
	다크모드 지원	다크모드 지원을 위한 별도 css코드를 작성해야함	
기능 설계 및 개발	기본 사용자 입력 처리	방법: 숨겨져 있는 textarea에서 사용자 입력을 받고, 이를 적절히 처리하여 편집기 화면에 표시함	javascript, React
	타자속도 및 정확도 실시간 계산	(입력된 글의 수 - 틀린글자수)/시간 을 통해 실시간으로 타자속도를 계산하여 표시함	
	코드 자동완성 기능	단어 단위로 해당 소스코드에 대응되는 언어의 키워드를 비교해 적절한 자동완성 단어를 추출해 해당 화면에 표시함. Enter나 Tab, 방향키로 단어 선택 및 입력이 가능하도록 함	
	다크모드 기능	작업막대의 다크모드 버튼을 통해 일반모드와 다크모드를 토글 할 수 있도록 함	javascript, css
	리더보드에 결과 등록 및 조회 기능	연습이 완료되어 나타난 결과 화면에서 버튼을 누르면 사용자입력과 메시지, 타자속도 등을 데이터베이스에 저장하도록 하는 기능	React, Express.js MongoDB
	웹 호스팅	실제로 배포를 하는 과정,	Express.js,

		서버는 Oracle Cloud의 인스턴스 이용, Docker 및 Github Actions를 통해 배포 자동화	Oracle CI, Docker, Github actions,
추가 개발 사항	다양한 소스코드 파일 제작	연습할 수 있는 다양한 소스코드 파일 제작	javascript
	코드 하이라이팅	실제 IDE처럼 해당하는 키워드의 색을 변경하는 기능	React

3. 개발결과



그림 5 초기화면

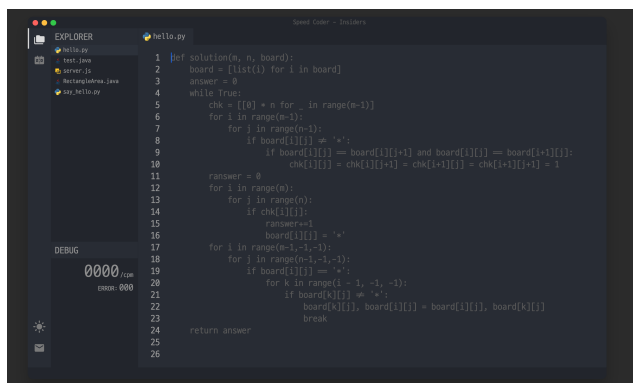


그림 6 다크모드 화면

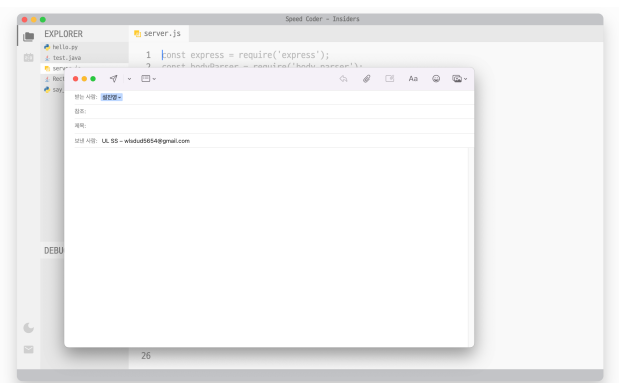


그림 7 피드백 버튼

[그림 5]는 처음 사이트에 접속하면 나타나는 화면이다. 코드 구역을 마우스로 클릭해 코드를 입력할 수 있다. 좌측 사이드바에는 차례대로 탐색기 버튼, 랭킹(리더보드) 버튼, 다크모드 버튼, 피드백 버튼이 존재한다. 창은 마우스로 끌어 위치를 움직일 수 있도록 개발하였으며, 다크모드 버튼을 클릭하면 [그림 6]과 같은 화면으로 변경되며, 피드백 버튼을 클릭하면 사이트에 대한 피드백을 전달할 수 있도록 개발하였다.



그림 8 hello.py 화면



그림 9 server.js 화면

탐색기 창(EXPLORER)에서는 타자연습을 수행 할 코드파일을 선택할 수 있도록 개발하였다. [그림 8]과 [그림 9]를 통해 코드가 변경된 것을 확인할 수 있다.



그림 10 코드 입력 화면

[그림 10]은 코드를 입력하면 나타나는 화면이다. 코드가 정상적으로 입력되었을 경우 진하게 표시되며, 오타인 경우는 빨간색으로 강조하여 오타임을 알린다. 만약 입력한 단어와 비슷한 예약어가 존재한다면 그림과 같이 자동완성 키워드를 추천한다. Enter혹은 Tab을 통해 즉시 입력할 수 있으며, 방향키를 통해 키워드를 선택할 수 있다.

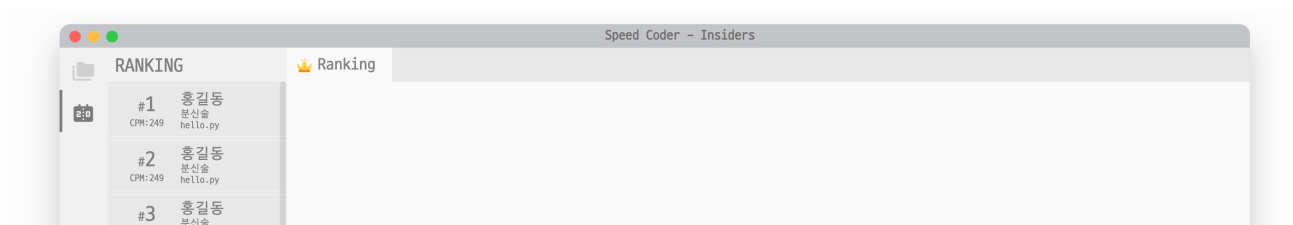


그림 11 랭킹 화면

[그림 11]은 랭킹 버튼을 눌렀을 시 나타나는 랭킹 화면이다. 사용자들이 기록한 타자속도를 내림차순으로 순위를 매겨 보여준다. 무한스크롤 방식으로, 스크롤을 내리면 위 내용은 삭제하고 아래 페이지 내용을 불러온다.



그림 12 DEBUG 화면

그림 13 결과창 화면

[그림 12]는 DEBUG 화면으로 사용자가 실시간으로 자신의 타자속도와 오타수를 확인할 수 있다. [그림 13]은 결과창 화면으로, 자신의 타자속도를 보여주며 이름과 메시지를 사용자로부터 입력받고, Push버튼을 클릭하거나 Enter를 입력하면 리더보드에 기록된다.