**문서종류 : 최종보고서**

**주제 : SoongPiler / Born to Coding**

**Born to Coding**

**최유성**

**김동환**

**김현석**

**박종빈**

목 차

**1절 프로젝트 개요 3**

1.1 SoongPiler 3

1.2 개발 배경 및 목적 3

1.2.1 개발 배경 3

1.2.2 시장 현황 4

1.3 추진체계 및 일정 5

1.3.1 추진 체계 5

1.3.2 일정 5

1.3.3 부문별 수업 업무 및 담당자 5

**2. 배경 지식 7**

2.1 RSS(Really Simple Syndication) 7

2.1.1 <내용 삽입>

**3. 시스템 구축 내용 15**

<해당 내용 삽입>

**4. 운영자 및 사용자 메뉴얼 61**

4.1 Web Sites 5

4.1.1 메인 홈페이지 5

4.1.2 JAVA 강의 웹페이지 5

4.1.3숭파일러 초기화면 5

4.1.4숭파일러 실행화면 5

**5. 결론 및 개선 방안 67**

5.1 결론 및 기대 효과 67

5.1.1 결론 67

5.1.2 기대 효과 68

5.2 개선 방안 68

1절 프로젝트 개요

* 1. SoongPiler

최근 언어 Trend에 지배적이고, 개발 입문자들의 난이도를 고려한 JAVA 언어를 활용하여 웹 상에서 컴파일이 가능하도록 하고자 하였다. 사용자들은 기초적인 내용을 학습한 후에는, 직접 실습할 컴파일 환경을 제공하는 것이 목적이다.

추가적으로 기본적인 개발 지식을 가지고 있는 사용자들에게는 다양한 언어를 이용한 알고리즘 교육 환경을 제공할 계획이다. 커뮤니티 기능을 활성화하여 자신이 풀었던 문제들을 정리하여 다른 사용자들과 공유, 비교할 수 있도록 하고, 알고리즘 문제를 해결하였을 시, 사용자가 작성한 소스코드의 시간 복잡도를 시각화하여 다른 사용자들의 코드와 비교, 확인할 수 있도록 버전을 높이면서 그 기능을 확장할 것이다.

1.2 개발 배경 및 목적

1.2.1 개발 배경

현 4차산업으로 인해 양질의 소프트웨어의 중요성이 대두되고 있는 사회 분위기에, 초/중/고등학생들의 교육과정에 코딩교육이 추가되면서 많은 사람들이 코딩에 관심을 가지기 시작했지만, 단순히 또 다른 교과목으로 인식되면서 학생들과 학부모들에게 부담을 가지고 있는 것 역시 사실이다. 따라서 코딩에 대한 플랫폼의 수요는 지속적으로 높아지지만, 기존의 교육 플랫폼은 입문자보다는 전공자들에게 초점이 맞춰져 있기 때문에 코딩을 배우기 위한 진입장벽은 크게 낮추면서, 코딩에 필요한 Editor와 필요한 시스템 환경변수 설정 없이 Web Compiler를 제공하고자 한다. 즉, 기초교육을 받는 학생들을 대상으로 하여 쉽게 컴퓨터 언어를 접하고 배울 수 있는 교육 플랫폼을 만들고자 하였다. 따라서 우리가 인터넷에서 배웠던 다양한 학습 페이지들과 플랫폼을 고려해, 교육의 핵심적인 내용은 Youtube의 “생활코딩” 채널의 내용을 통해 제공하고 있다. 또한, 커뮤니티 기능을 활성화해서, 입문자들에게 도움을 대화의 장을 통해 스스로 문제를 해결해 갈 수 있도록 지원하고 있다.

추가적으로, 현재 개발한 Compile 환경을 기반으로 심화 학습자를 위해 알고리즘 문제 및 채점 기능을 제공해, 시간 복잡도를 고려해 커뮤니티 이용자들 간의 코드를 비교하는 기능을 확장할 예정이다. 즉 현재는 입문자들의 교육 플랫폼에 초점을 맞추고, 추후 심화 학습자들을 위한 contents를 추가하면서 플랫폼을 지속적으로 확장하는 것이 우리 BTC팀의 목표이다.

1.2.2 시장 현황

앞서 언급했듯이 현재, 다양한 교육 플랫폼이 웹에서 제공되고 있다. 교육 플랫폼마다 고유한 특성을 가지고 있는데, 국내/ 해외를 기준으로 분석해 보았다.

우선 국내 코딩 교육 플랫폼에는 “프로그래머스”, ”구름”, ”소프트웨어야 놀자”, “생활코딩” 등이 존재하고, 해외 교육 플랫폼에는 “LeetCode”, “CS Academy”가 존재한다.

우선 국내 플랫폼인 “프로그래머스”와 “구름”의 경우는 국내에서 가장 잘 알려진 코딩 교육 서비스를 제공하고 있다. 기초적인 개발환경을 제공해 Web-Compiler기능을 갖추고 있으며, 다양한 로그인 방식으로 입문자들도 쉽게 접근할 수 있다는 우수한 접근성을 가지고 있다. 하지만, 전공생들의 고급 알고리즘 문제 Solution에 초점이 맞춰져 있어, 주로 제공하는 Main Contents는 입문자용이 아니라는 아쉬운 점이 존재했다. 또한 “소프트웨어야 놀자”와 “생활코딩”의 경우 입문자들도 쉽게 개발이란 무엇이지 배울 수 있는 기초적인 교육자료를 제공하고, 접근성 또한 우수하다는 장점을 가지고 있지만 배운 내용을 실습할 수 있는 Web-Compiler기능이 존재하지 않았다. 해외 플랫폼 “LeetCode”의 경우는, Web-Compiler 기능과, 추가적으로 User간의 Code를 공유 및 자신의 Code를 실행시간 측면에서 비교할 수 있는 기능을 제공하고 있지만 모든 작업이 영어 기반이라 사용 연령층이 낮은 국내의 사용자에겐 진입장벽이 높게 느껴질 수 있는 요소와, 기초적인 교육자료를 제공하지 않는 다는 단점이 존재한다.

따라서 우리 BTC팀은, 사용 연령층을 고려해 진입장벽을 낮춰 누구나 쉽게 우리의 사이트를 접근할 수 있도록 접근성을 높였으며, 주로 다루는 콘텐츠는 기초적인 교육 내용부터 시작해 복잡한 내용을 추가해 나가는 확장적인 구조를 가지고, 별도의 Editor나 환경변수 설정 없이 개발환경을 제공한다. 또한, 현재 초기버전은 Youtube를 통한 기초적인 교육자료를 제공하고 있지만, 이에 그치지 않고 추후 심화 사용자를 위한 내용을 추가하려고 한다. 그 내용으로는, 알고리즘과 다른 언어, 사용자간 소통이 가능한 커뮤니티 기능 등을 버전을 높여가면서 그 기능들을 확장할 계획이다.

1.3 추진 체계 및 일정

1.2.1 추진 체계

팀장 : 최유성

팀원 : 김동환, 김현석, 박종빈

1.3.2 일정

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **작업 내용** | **9월** | | | | **10월** | | | | **11월** | | | | **12월** | | | |
| **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| 1 | 주제 선정 및 탐색 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | 자료 수집 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | 시장 분석 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | 사례 연구 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | 관련 스터디 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | 프로젝트 설계 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | 세부 구현 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | 통합 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | 테스트 디버깅 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DESIGN | | | | |
| Task | Output | Schedule | Charger | Tools |
| Architectural  Design | System  Architecture | 19.09.19~10.13 | All developer | Draw.io |
| UI  Design | UI  Story board | 19.10.17~10.30 | 박종빈 | 3D Graphics |
| Logical  Design | Logical  UML Diagram | 19.11.02~11.17 | 김동환 | Draw.io |
| Physical  Design | Logical  System Structure | 19.10.31~11.25 | 최유성 | Draw.io |

1.3.3 부문별 수행업무 및 담당자

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| IMPLEMENTATION | | | | |
| Develop  Web Module | Web source  Compile source | 19.12.03~12.15 | All developer | Apache  HTML/JavaScript  Php/CSS |
| Develop  Server Module | Web Server  Compile Server | 19.12.15~12.23 | 김현석, 김동환 | Docker  Apache |
| Develop  Dynamic hosting | Web Server  Docker Container | 19.12.20~12.25 | 김현석, 최유성 | Docker  Bitnami |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TESTING | | | | |
| Unit Test | Unit test report | 19.12.25~12.26 | All Developer | Chrome  Atom |
| Integration  Test | Integration  Test report | 19.12.26~12.28 | All Developer | Chrome  Atom |
| Black/White Box  Test | SoongPiler  Ver 1.0 | 19.12.28~12.29 | All Developer |  |

2절 배경 지식

2.1 Docker

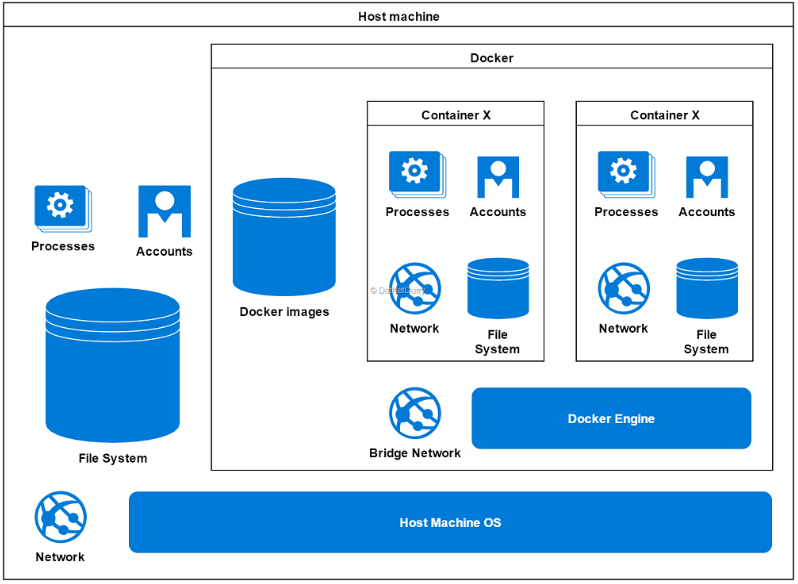
도커는 컨테이너를 이용한 가상 환경 플랫폼이다. 도커는 이미지 및 컨테이너로 구성되어 있으며 설정 된 값을 컨테이너(실행 환경)에서 실행할 수 있다.

그러므로 기존의 Virtual Machine과 비교시 월등한 실행 속도를 보장할 수 있고 하드웨어

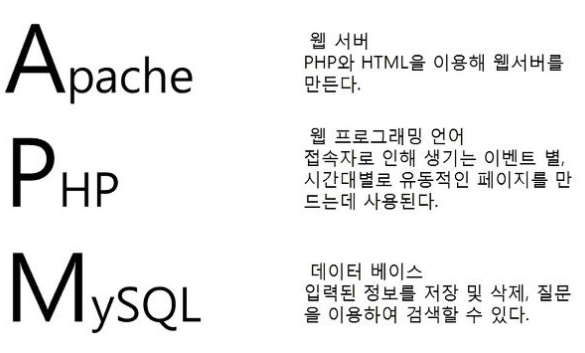
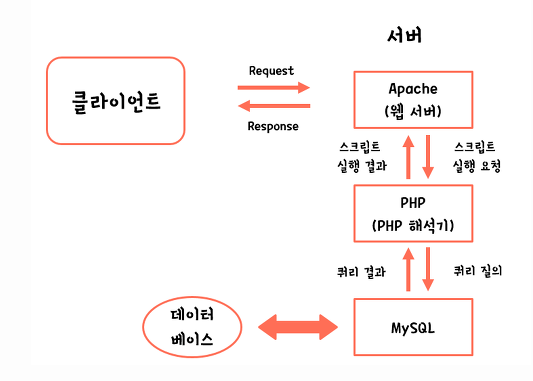
를 가상화 하지 않으므로 메모리 엑세스, 파일 시스템, 네트워크 실행 성능향상 등의

장점을 가져올 수 있다. 또한 사용 웹 컴파일러 사이트인 구름에서도 도커를 이용해

개발을 했음을 통해 기존 시장에서도 효용성을 인정받고 있음을 알 수 있다.



[Docker의 기술 적용 방식]

2.2 APM

APM은 Apache, PHP, MySQL을 통합한 웹 서버 개발 프레임워크이다. APM의 구동 방식은

다음과 같은 순서를 지킨다.

1. Client가 웹 브라우저를 통해 URL을 이볅하여 원하는 정보를 서버에 요청한다
2. 웹 서버인 Apache는 Client의 요청을 승인하고 정적인 페이지를 보여준다.
3. 만약 요청한 정보가 동적 웹페이지라면 PHP에 스크립트 실행을 요청한다.
4. PHP는 전달받은 소스를 해석해준다.
5. 이 때, 데이터베이스 요청이 들어오면 MySQL과의 상호 작용을 통해 관련 작업을 수행한다.
6. Apache 웹 서버는 PHP로부터 결과를 전달받고 이를 웹 브라우저를 통해 Client에게

출력한다.

2.3 Apache Web Server

Apache HTTP Server는 오픈 소스 소프트웨어 그룹인 아파치 소프트웨어 재단에서 개발한

웹 서버 프로그램이다. 팀 버너스 리가 만든 최초의 웹 서버 프로그램인 “CNCSA HTTPD”를 기반으로 만들어졌다.

현재 거의 모든 리눅스 배포판이 아파치를 지원하고 있다. 또한, 아파치는 확장성이 매우

좋아서 모듈이라는 개념으로 수 많은 기능을 덧붙일 수 있따. 이 모듈을 통해 다른 프로그램과의 연동도 가능하며 이 때문에 여러 가지 서버 사이드 프로그래밍 언어나 DBMS와도

궁합이 잘 맞는다.

2.4 PHP

Personal Home Page의 약자로, 통칭 PHP : Hypertext Preprocessor라고 부른다.

웹 프로그래밍 언어로 서버 사이드에서 실행되는 언어이다. HTML의 경우 웹 브라우저에서

소스를 해석해 보여주지만 PHP는 서버에서 해석하여 HTML 코드로 만들어 브라우저에게

전달하는 것이다.

PHP의 특성은 다음과 같다.

* PHP는 Apahce라는 웹서버 프로그램과 연동하여 동작한다.
* PHP는 .php파일을 처리하고 Apache는 HTML 파일을 처리한다.
* PHP를 활용하여 Dynamic Web Page를 만들 수 있다.
* 이 외에도 ASP, JSP 등이 사용되고 있다.
* 

[PHP의 예제 코드]

또한 PHP의 장점으로는 웹에 최적화된 언어이며 웹 개발에 필요한 수 많은 로직들이

함수의 형태로 제공된다. 게다가 거의 모든 데이터베이스를 제공하고 현재 가장 많은

공개 소프트웨어가 PHP로 만들어졌다. 하지만, 요즘은 객체 기반의 Node.js와 JavaScript 가 많이 사용되고 있지만. 여전히 PHP는 가장 기본적이면서도 전세계에서 많이 사용된다



[PHP, Node.js, JavaScript의 지난 5년간 검색량 비교]

2.5 MySQL

데이터베이스는 정보를 저장하는 애플리케이션이다. 정보를 파일에 저장하는 것도 좋은

방법이 될수 있지만 데이터베이스는 파일에 저장하는 것 보다 훨씬 많은 기능을 제공한다.

말하자면 정보를 저장하는데 특화된 시스템이라고 할 수 있다.

오늘날 거의 대부분의 데이터들이 데이터베이스에 저장되고 있다. 그 중에서도 MySQL은

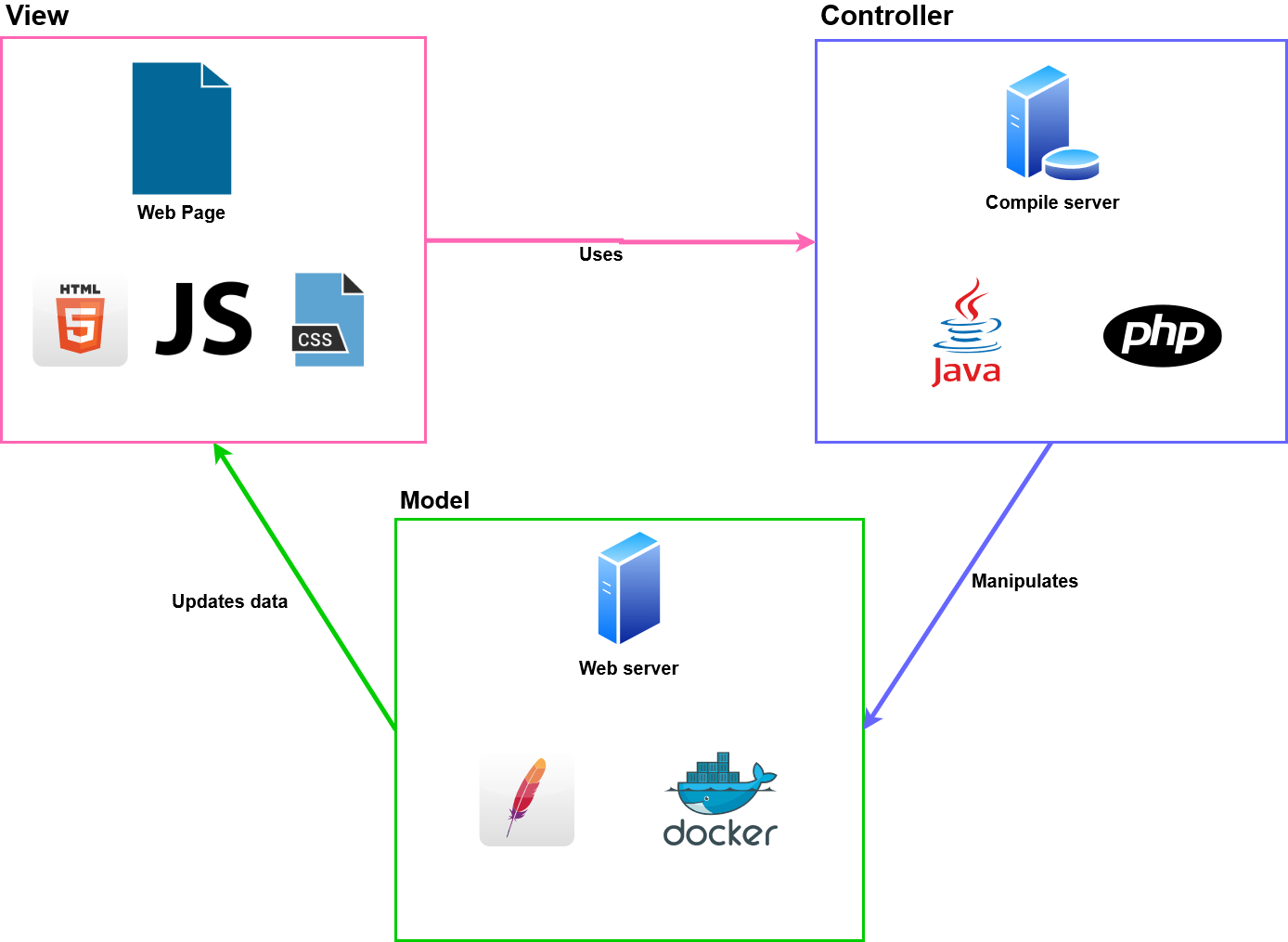
오픈소스이고 무료인 DBMS이다. 이런 이유로 많은 웹 어플리케이션이 MySQL을 기본

데이터베이스로 채택하고 있다. 우리도 MySQL을 이용해 웹 서버 구동에 필요한 게시판,

사진, 동영상 등의 데이터들을 저장하고 사용자에게 제공하기 위해서 MySQL을 사용할 것이다.

3절 시스템 구축 내용

3.1 시스템 구성도



**[ 시스템 구성도 ]**

시스템을 구성하기 위해 디자인 패턴 중 하나인 MVC 모델을 사용하였다. 그림의 좌측에 있는 것은 MVC 모델 중 View 모듈을 나타낸다. 사용자와 직접적으로 소통하고 동작하는 View 모듈의 구현을 위하여 우리는 HTML, JavaScript, CSS와 같은 기술을 채택하였다. 이를 통하여 직관적인 U/I를 갖는 ‘웹 서비스’를 만들어 낼 수 있다.

그림의 우측에 있는 것은 사용자로부터 전달받은 명령을 처리하는 Controller 모듈이다. Controller 모듈에서는 사용자로부터 전달 받은 명령(영상 재생, 코드 컴파일 등)을 직접 처리하는 역할을 맡는다. 본 프로젝트에서 제공하고자 하는 프로그래밍 언어는 Java이기 때문에 .Java 파일을 컴파일하기 위한 Java와 이를 기반으로 컴파일을 수행하고 관리하기 위하여 PHP를 사용하여 웹 페이지를 구현하였다.

마지막으로 그림의 아랫 부분은 Model 모듈을 나타낸다. 이 모듈은 서버 모듈로써 사용자에게 서비스 제공을 담당하는 영역이다. 웹 서버를 구현하기 위하여 Docker를 기반으로 하여 그 위에 Apache를 올려서 구현하고자 하였다. 일반적인 OS에서 웹 서버를 제공하지 않고 Docker를 사용하여 컨테이너화 된 웹 서버를 구현함으로써 확장성과 이동성을 보장하고자 하였다.

3.2 시스템 설계

3.2.1 세부 모듈

본 프로젝트는 사용자에게 Web Page를 제공하기 위한 웹 서버와 사용자로부터 입력 받은 코드를 컴파일하기 위한 컴파일 서버로 크게 나뉜다. 이에 본 프로젝트에서는 Docker를 활용하여 컨테이너에 Apache를 사용한 웹 서버와 Java를 사용한 컴파일 서버를 구현하고자 하였다. 이에 앞서 우선적으로 Bitnami를 활용한 웹 서버를 구현하여 정상적으로 Web 사이트가 작동하고 유저에게 직관적인 컴파일 환경을 제공할 수 있음을 확인하였다.

세부 모듈은 MVC 디자인 패턴을 적용하여 크게 3가지 모듈로 구분할 수 있다.

3.2.2 Web Server 모듈

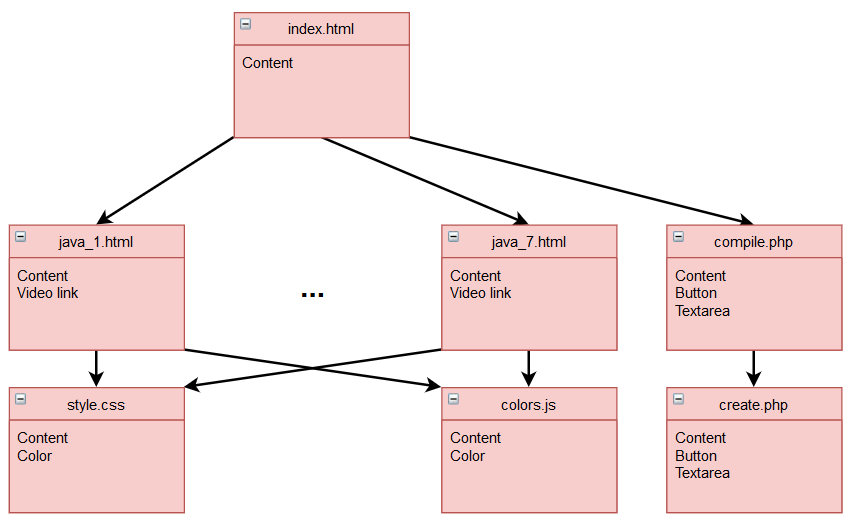
사용자에게 웹 서비스를 제공하기 위하여 Apache를 사용하여 웹 서버를 구축하고자 하였다. Apache를 통해 구축된 웹 서버에 사용자는 접속하여 Java와 관련된 동영상 강의를 접할 수 있고 Java 코드를 직접 작성해 볼 수 있는 에디터 환경을 활용할 수 있다. 웹 서버를 구축하는 방법에는 Docker를 활용해 컨테이너화 된 Apache를 사용하는 방법과 Bitnami를 통해 구축된 Apache를 사용하는 2가지 방법이 존재한다. 이에 본 프로젝트에서는 우선 Bitnami를 사용하는 방법을 통해 사용자에게서 전달 받은 Java 코드를 Web 환경을 통해 처리하고 컴파일 결과를 제공할 수 있음을 확인하고자 했다.

3.2.2 Compile Server 모듈

본 프로젝트에서 사용자에게 제공하고자 하는 프로그래밍 언어는 Java이므로 Java를 컴파일하기 위해 Java를 설치할 필요가 있었다. 이는 웹 서버와 마찬가지로 Docker를 통해 컨테이너화 된 Ubuntu위에 Java를 설치하는 방법과 기존에 설치된 Java를 사용하는 방법이 존재한다. 이에 본 프로젝트에서는 웹 상에서 사용자가 입력한 코드를 OS에 설치되어 있는 Java를 통해 컴파일하는 방식을 채택하여 사용자에게 직관적으로 컴파일 가능한 환경을 제공하고자 하는 궁극적인 목표의 가능성을 우선적으로 확인했다.

3.2.4 Web Solution Workflow

(1) Web Site Workflow



**[ Web Site Workflow ]**

본 프로젝트에서 제공하고자 하는 웹 서비스의 첫 page는 index.html이다. 해당 페이지에는 이 웹 사이트가 어떠한 목적으로 만들어졌고 어떠한 서비스를 제공하는지를 간략하게 설명해주는 텍스트와 사용자가 직접 컴파일 해볼 수 있도록 구축된 링크로 통하는 버튼과 강의 목록을 보여주는 List로 구성되어 있다. java\_1.html~ java\_7.html은 총 7 페이지로 구성된 Java 관련 강좌의 세부 목록들이다. 각 페이지들은 페이지를 구성하기 위한 속성들로 이루어진 style.css, colors.js 파일을 참조하여 페이지를 구성한다. 사용자는 각 페이지를 순차적으로 접근하며 강좌를 수강 할 수 있고 강좌 마지막에는 학습한 내용을 직접 실습할 수 있도록 컴파일 Page로 연결된 버튼으로 구성된다. 또한 DISQUS에서 제공하는 Open API를 활용하여 각 강좌 별로 Comments를 남길 수 있는 기능을 제공하여 유저 간의 원활한 소통을 장려하고자 했다. 사용자로부터 입력 받은 Java 코드를 컴파일하기 위한 모듈이 compile.php와 create.php이다. Compile.php는 사용자가 코드를 입력할 수 있는 text area와 예시 코드를 제공한다. Create.php에서는 작성된 코드로부터 도출된 “Solution.java” 파일을 javac 명령어를 통해 컴파일 후에 생성된 class 파일을 실행하고 그 결과를 화면 상에 출력하고 생성된 파일을 삭제하는 방식으로 컴파일 과정을 컨트롤 할 수 있다. 이러한 과정에서 유저는 별도의 과정 없이 Java 코드를 직접 작성해보고 컴파일 결과를 웹 상에서도 확인할 수 있게 된다.

(2) Web Site 클래스 명세

|  |  |
| --- | --- |
| **클 래 스** | **설 명** |
| Index.html | 초기 Main page |
| Style.css | 기본 Page 속성 중 css 관련 속성 설정 파일 |
| Color.js | 기본 Page 속성 중 script 관련 속성 설정 파일 |
| Java\_1~7.html | Java 관련 content와 text로 구성된 page |
| Compile.php | Java 코드를 입력받고 컴파일 하기 위한 초기 page |
| Create.php | 입력 받은 코드를 기반으로 컴파일/결과 출력 담당 page |

3.3 프로그램 세부 구현

3.3.1 웹 서버 모듈(Web Server Module)

본 프로젝트에서 모듈을 2가지로 나눈다면 웹 서버를 기반으로 구축된 웹 페이지를 통해 사용자에게 강의를 제공하는 Web Server 모듈과 사용자에게 입력 받은 Java 코드 기반으로 하여 컴파일 결과를 웹 상에 제공하는 Compile 모듈로 나눌 수 있다. 첫 번째 모듈, 웹 서버 모듈을 구성하기 위해 메인 page인 index.html 파일과 Java 관련 세부 강의 목록을 제공하는 파일, java\_1.html ~ java\_7.html을 살펴보면 다음과 같다.

먼저 index.html 파일의 세부 내용이다.

<!doctype html>

<html>

<head>

<title> SoongPiler-Welcome</title>

<meta charset="utf-8">

<link rel="stylesheet" href="style.css"> //작성해 둔 style.css 파일을 참조

<script src=

"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">

</script>

<script src ='colors.js'></script> //작성해 둔 colors.js 파일을 참조

</head>

<body>

<h1><a href="view/index.html" style="color: skyblue">SoongPiler</a></h1>

<div id="grid">

<ol>

<li><a href="view/1.html" style="color:blue"> JAVA학습 </a></li>

//Java 관련 강의를 제공하는 첫번째 파일로 link를 연결

<li> Python ..</li>

<li> Algorithm .. </li>

</ol>

<div id="article">

<h2> SoongPiler Project </h2>

<p>

….중략

</p>

<div id="compile">

<a href="../controller/compile.php" style="text-decoration:underline;">컴파일 해보기</a> //메인 page에서도 사용자가 컴파일 Page에 접근가능 하도록 링크를 생성

</div>

</div>

</body>

</html>

다음은 Java 관련 강의를 제공하는 첫 번째 파일, java\_1.html의 세부 내용이다.

<!doctype html>

<html>

<head>

<title> SoongPiler-JAVA</title>

<meta charset="utf-8">

<link rel="stylesheet" href="style.css"> //작성해 둔 style.css 파일을 참조

<script src=

"https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">

</script>

<script src ='colors.js'></script> //작성해 둔 colors.js 파일을 참조

</head>

<body>

<h1><a href="index.html" style="color: skyblue">SoongPiler-JAVA란</a></h1>

<div id="grid">

<ol> /\*웹 페이지에서 제공하는 총 7가지의 Java 관련 수강 목록 리스트,현재 접속 중인 챕터의 글자 색을 구분하여 현재 어느 페이지에 접속해 있는지 직관적으로 표현\*/

<li><a href="1.html" style="text-docoration:underline; color:blue;"> JAVA란</a> </li>

<li><a href="2.html"> 데이터와 연산</a> </li>

<li><a href="3.html"> 숫자와 연산</a> </li>

<li><a href="4.html"> 문자열 다루기</a> </li>

<li><a href="5.html"> 변수</a> </li>

<li><a href="6.html"> 입력과 출력</a> </li>

<li><a href="7.html"> 수업을 마치며</a> </li>

</ol>

<div id="article">

<h2> SoongPiler-JAVA란 </h2>

<iframe width="560" height="315" src="https://www.youtube.com/embed/-dPXqgWQBGE" frameborder="0" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>

/\*사용자가 접근한 챕터에 해당하는 Java 강의를 Youtube형식으로 웹 페이지 내에서 제공\*/

<p style="margin-top:0;">

….중략

</p>

<img src="html\_coding\_image.jpg" width="50%">

<p style="margin-top: 0px;">

<a href="../controller/compile.php" style="text-decoration:underline;">직접 도전해 보세요!</a>

/\*사용자가 위에서 학습한 내용을 Java 컴파일이 가능한 환경에서 실습해 볼 수 있도록 컴파일 page에 접근 가능한 link 생성 \*/

</p>

</div>

</div> /\* open API 중 하나인 DISQUS를 적용하여 사용자가 원하는 강의에 코멘트를 남길 수 있는 기능을 제공 \*/

<div id="disqus\_thread"></div>

<script>

(function() {

var d = document, s = d.createElement('script');

s.src = 'https://soongpiler.disqus.com/embed.js';

s.setAttribute('data-timestamp', +new Date());

(d.head || d.body).appendChild(s);

})();

</script>

<noscript>Please enable JavaScript to view the <a href="https://disqus.com/?ref\_noscript">comments powered by Disqus.</a></noscript>

</body>

</html>

이와 같이 Java에 관련된 강의 목록, 7개의 java\_1 ~ 7.html 파일이 구성된다.

3.3.1 컴파일 모듈(Compile Module)

사용자가 입력한 Java 코드를 기반으로 javac 명령어를 통해 Class 파일을 생성하고 해당하는 Class 파일의 출력 결과를 Web 상에 출력하는 부분이 컴파일 모듈이 담당하는 부분이다. 이러한 동작은 compile.php와 create.php 파일을 통해 컨트롤 된다. 그 세부 내용은 다음과 같다.

먼저 compile.php의 세부 내용이다.

<?php

function javac(){

if(isFileExist() == 1) {

$exec = "javac Solution.java";

shell\_exec($exec);

// echo $exec;

// java를 통해 생성된 class 파일 실행

$output = shell\_exec("java Solution");

echo $output;

}

}

function isFileExist(){

$location = "Solution.java";

if(file\_exists($location)) {

return true;

} else {

return false;

}

}

$default = "public class Solution

{

public static void main(String[] args) {

}

}"

?>

<head>

<meta charset="utf-8">

<title>Soongpiler</title>

<a href="../index.html"><h1>숭파일러</h1></a>

</head>

<body>

소스코드 입력(class명은 Solution)

<form action="create.php" method="post">

<p>

<input type="hidden" name="title" value="Solution.java"/>

</p>

<p>

<textarea name="code" placeholder="public class Solution

{

public static void main(String[] args) {

}

}"

style="width:100%;border:1;overflow:visible;text-overflow:ellipsis;" rows=30 ></textarea>

</p>

<p>

<input type="submit">

</p>

</form>

<예시 코드>

<br>

<img src="../model/example.PNG">

<?php

javac();

?>

</body>

</html>

4절 운영자 및 사용자 매뉴얼

4. 1 Web Sites

4.1.1 메인 홈페이지

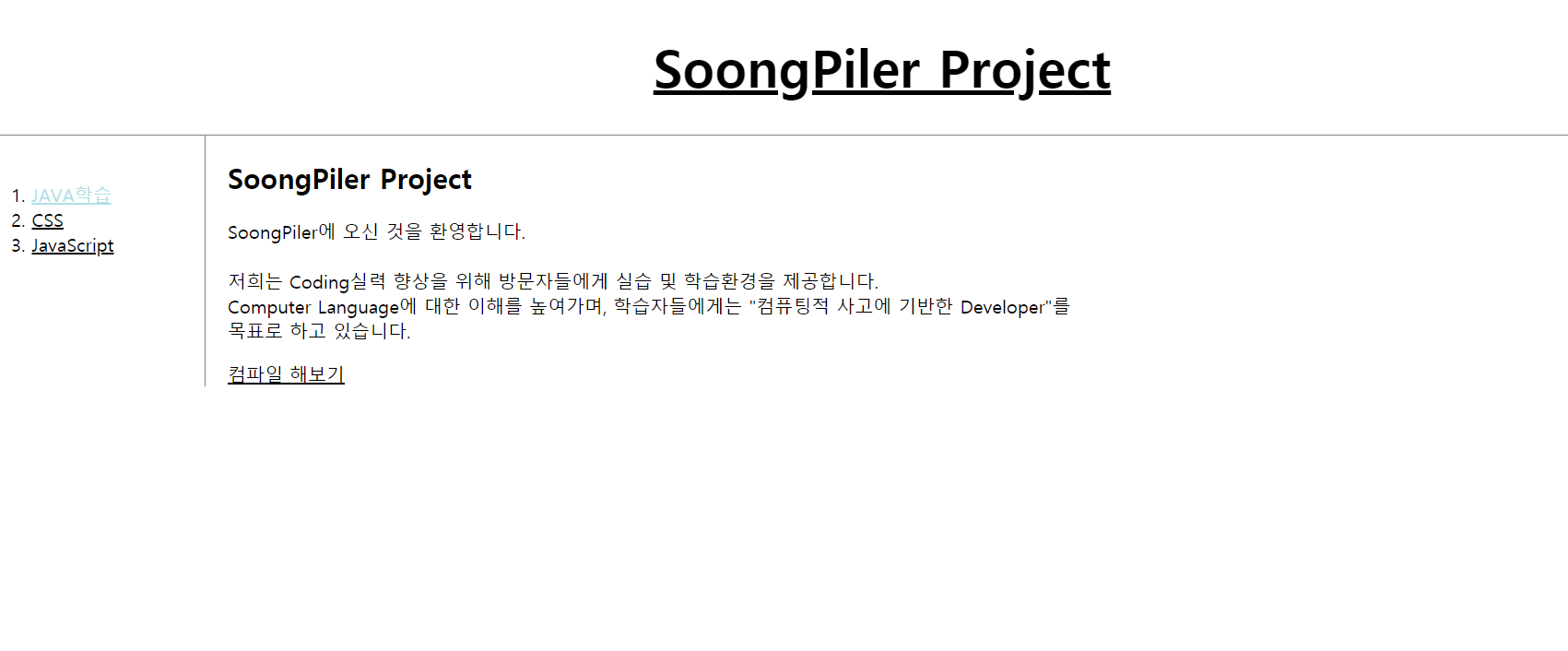


그림 메인 화면

방문 초기 화면에서 제목을 클릭하면 메인 화면으로 되돌아 갈 수 있다. 좌측에 있는 목차에서 사용자가 수강하길 원하는 과목을 선택하여 해당 웹페이지로 이동할 수 있다. 메인 내용에는 간략하게 Soongpiler에 대한 소개와 더불어 컴파일을 할 수 있는 웹페이지로 갈 수 있는 링크도 같이 포함되어 있다.

**4.1.2 JAVA 강의 웹페이지**

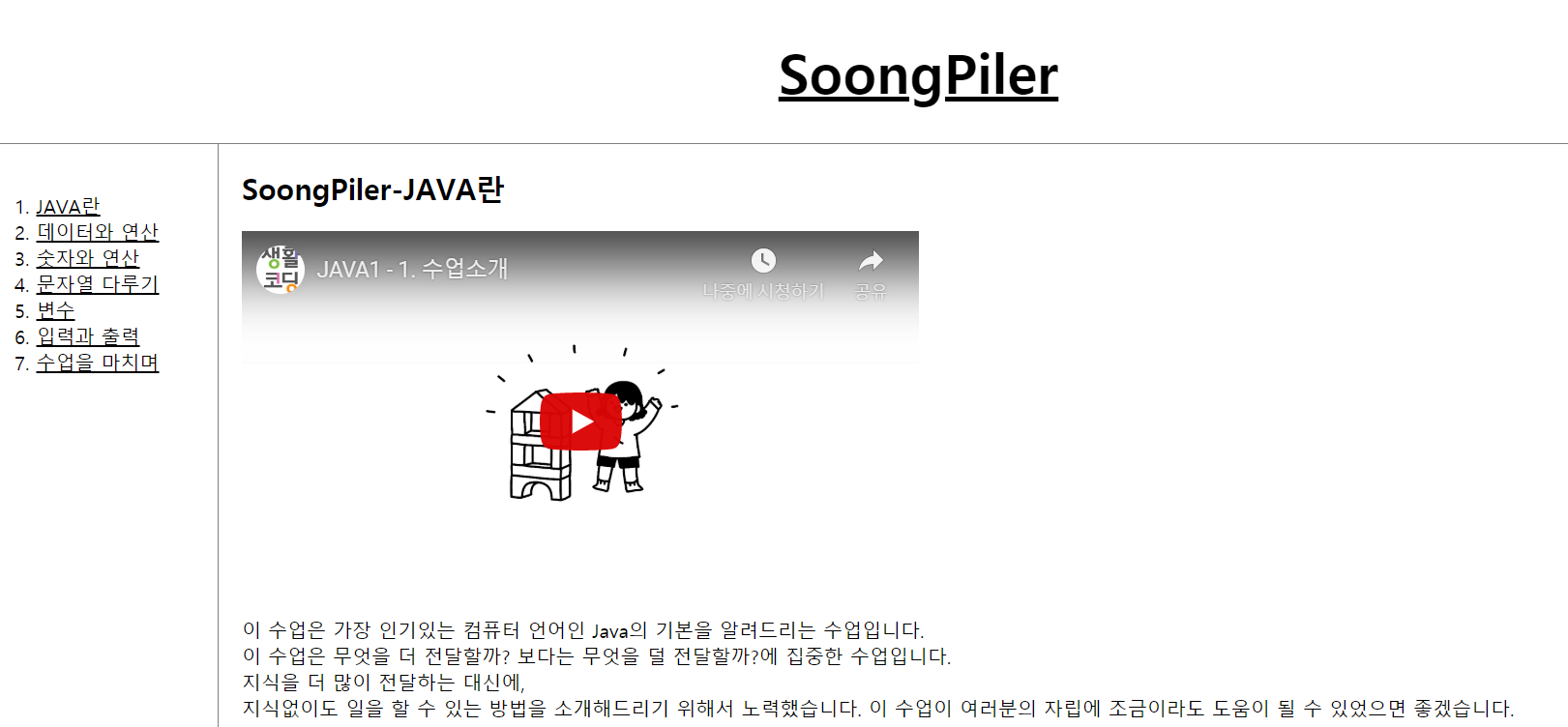


그림 JAVA 강의 페이지

메인 홈페이지에서 JAVA 강의를 선택한 이후의 처음 단계로 JAVA에 대해 소개하는 “JAVA란” 웹 페이지이다. 화면 상단에는 메인 홈페이지로 다시 돌아갈 수 있는 링크를 가진 제목이 있으며, 화면 좌측에는 수강 세부 항목들을 선택할 수 있는 리스트가 있다. 리스트는 수강 세부 과정에 대해 순차적으로 정렬되어 있으며 사용자가 언제든 원하는 과목을 먼저 수강할 수 있다.

수강 내용에 대해 사용자의 이해도를 높이기 위해 이미지와 관련된 Youtube 및 다양한 플랫폼에 있는 강의 영상을 첨부하여 수강 내용과 같이 설명하고 있다. “직접 도전해 보세요!”를 클릭하게 되면 배운 코드를 바로 컴파일하고 실행해볼 수 있는 숭파일러 웹페이지로 이동하게 된다.

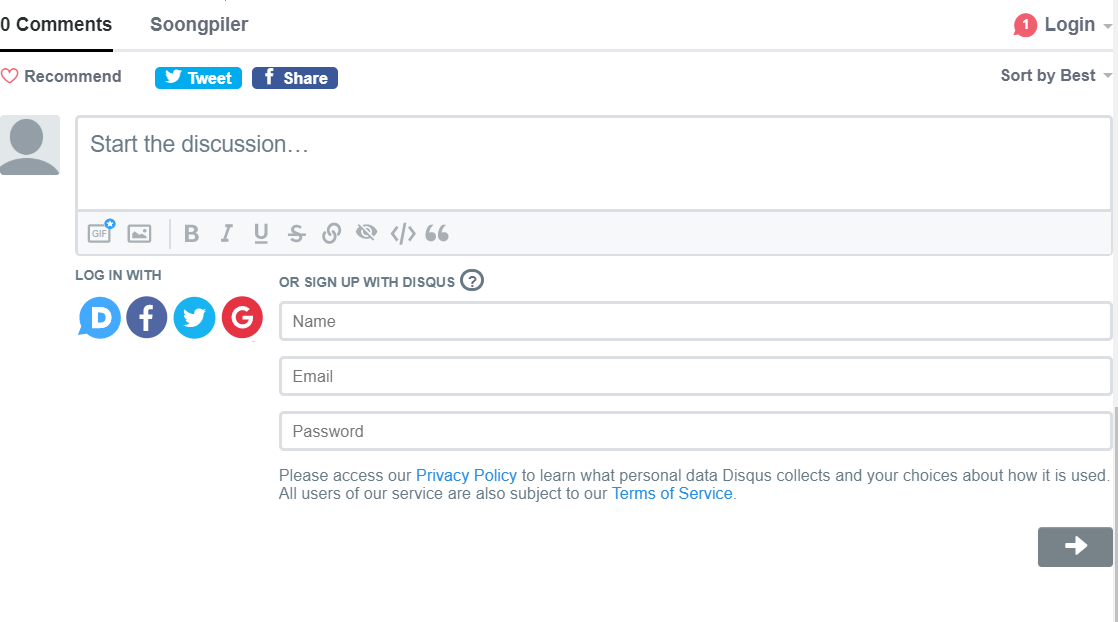


그림 숭파일러 댓글 화면

댓글 기능을 통해 해당 수강 항목과 관련된 내용을 Disqus를 통해 웹 운영자나 사용자들끼리 서로 소통이 가능하다. 댓글 기능을 사용하기 위해서는 Disqus나 Facebook, Twitter로 로그인이 필요하다.

**4.1.3 숭파일러 초기화면**





메인 홈페이지 내용에서 “컴파일 해보기” 또는 각 강의 내용 말미에서 “직접 도전해 보세요!”를 통해 들어오게 되면 나오는 숭파일러 웹페이지이다. 제목인 숭파일러는 메인 홈페이지로 링크되어 있다.

화면 중간에 나오는 텍스트 박스에 사용자가 강의를 통해 배운 내용이나 임의로 컴파일을 하고자 하는 코드를 입력한다. 클래스명은 Solution으로 예시 코드와 같은 형식으로 클래스를 설정하여 Main 내용에 코드를 추가하면 된다.

코드를 다 작성한 후 텍스트 박스 아래에 있는 제출하기 버튼을 클릭하면 코드를 java파일로 변환하여 서버로 전송하게 된다.

**4.1.4 숭파일러 실행화면**

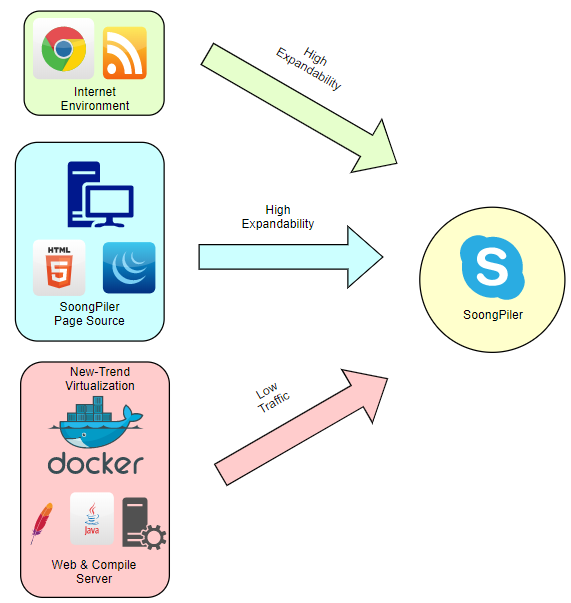


숭파일러 초기화면에서 제출 버튼을 누르면 새로운 웹페이지가 열리게 된다. 서버로 전송된 java 파일은 서버에서 컴파일하여 제출 버튼 아래에 java로 실행시킨 결과를 보여준다. 사용자는 원하는 만큼 코드를 재작성 하여 컴파일 해 볼 수 있으며, 실행을 다 완료하였으면, 돌아가기로 메인 홈페이지로 되돌아갈 수 있다.

**5. 결론 및 개선 방안**

5.1 결론 및 기대 효과

5.1.1 결론



소프트웨어 전문가를 육성하겠다는 現교과과정의 목표를 만족하기 위해, 수 많은 학생들 및 코딩 입문자들에게 부담을 덜어주고, 스스로 단계적 학습을 통해 컴퓨팅적 사고를 가진 Developer를 양성하는 것이 우리 BTC의 목표이다.

기존의 교육 플랫폼 “프로그래머스”나 “구름” 역시 사용자 연령층을 고려하며, 기초적인 내용을 포함하며 소프트웨어 코딩 교육에 대한 수요를 해결하고자 노력중임을 시장 조사하면서 알 수 있었다. 하지만, 주요 사용 대상이 여전히 심화 학습이라 초기 화면에서 코딩 입문자들이 복잡함을 느끼고 포기하는 경우가 많다는 것 역시 파악했다.

이러한 문제를 해결하기 위해, 정확한 코딩의 문법과 구조적인 형태를 강조하는 구시대적 교육이 아닌, 쉽지만 핵심적인 내용과, 짧지만 반드시 알아야 하는 내용을 다루는 개인이 운영하는 플랫폼이 많은 사용자들에게 좋은 평을 받고 있으며, 스스로 검색하는 힘을 길러 어떻게 공부하는지를 가르치는 “생활코딩”을 통해 해답을 찾았다. 하지만 우리는, 여전히 입문자들이 언어에 따른 다양한 Editor를 설치하는데 환경이 마땅치 않거나, 어려움을 겪는 사용자들의 고민을 해소하고자 Web에서도 개발이 가능하도록 노력했다. 그리고 SoongPiler가 탄생했다.

이러한 사용자들의 불편함을 해결하고자 우리 BTC팀에서 개발한 Web page SoongPiler는, 기초적인 교육자료를 통해 사용자가 영상을 통해 공부한 후, 실습할 수 있도록 Web-Compiler를 통해 작성한 Code에 대한 Feedback이 가능하도록 한다.

또한, Web-Server의 과부하를 방지하기 위해 최신 가상화 기술인 Docker를 적용해 Web-Server와 Compile-Server를 관리한다. 이로써 무겁고 비싼 Server없이 사용자에게 Web-Server를 제공해 Traffic을 최소화했다.

또한, 현재는 입문 단계의 개발자만을 고려한 contents만 다루고 있지만, 추가적으로 다른 언어(C언어/Python 등)와 알고리즘/자료구조 문제에 대한 내용을 추가하며 현재 SoongPiler를 지속적으로 확장해, 기존 사용자들에게 단계적 학습을 제공하고자 한다. 또한 커뮤니티 기능을 추가함으로써, 사용자들간 문제에 대한 Solution을 찾기 위한 지원을 하도록 제공할 것이다. 우리의 “SoongPiler”는, 개발을 시작하고 심화적인 학습을 하려는 사용자들의 불편함을 최대한 해소하고, 코딩을 두려워하지 않는 학습자와 스스로 심화학습을 학습해 숙련된 개발자를 양성하는 것이 최종 목표이다.

5.1.2 기대효과

- 사용자는 JAVA언어에 대한 기초적인 내용을 학습할 수 있다.

- 사용자는 학습한 JAVA언어를 토대로, 직접 Coding할 수 있다.

- 복잡한 Editor설치 과정없이, 인터넷만 된다면 JAVA Compile의 결과를 얻음

- 독립적인 Web-Server로, 접근하는 사용자들의 Traffic을 최소화.

- 최신 가상화 기술을 통해, 관리자의 Server관리 부담 최소화.

- 추가적인 버전 관리를 통해, 확장 가능한 웹 페이지 관리 가능.

- 지속적인 학습 컨텐츠를 추가해, 사용자들은 단계적 학습 가능.

- 커뮤니티 기능을 확장해, 사용자간 Solution이 가능하도록 지원.

5.2 개선방안

Docker는 생긴지 얼마 안된 가상화 Service이다. 본래 Docker는 Linux환경을 지원하고 있지만, 현재 우리들의 실제 개발환경인 Windows를 지원하긴 하지만 굉장히 제한적이라, 개발에 많은 장애요소를 겪었다. 따라서 우리는 Virtual Machine을 통해 Web Server와 Compile Server를 관리하고 있지만, 가상머신 위에서 동작해 여러모로 불안정성이 존재한다. 현재로선, Apache Server를 직접 설치해서 관리하는 방식을 2안으로 두고 있지만, 조금 더 Docker에 대한 이해를 통해 Windows환경에 최적화를 더 진행해야 한다.

만약 Windows환경에서 최적화된 Docker 서비스를 구현했을 경우, 더욱 안정성 있는 Web-Server를 사용자들에게 제공할 수 있었겠지만 현재로선 약간 아쉬운 부분이 있어 추가적인 보충 및 개선을 통해 더욱 안정된 Server를 제공할 수 있을 것이다. 또한, 현재 존재하는 양질의 교육 컨텐츠를 제공함과 동시에, 저작권에서 자유로울 영상이 한정적이라는 점이 아쉽지만, 한정적인 범위 안에서 양질의 컨텐츠를 찾기 위해 앞으로도 더욱 다채롭게 확장할 것을 약속한다. 앞으로 더욱 다채로운 내용의 컨텐츠와 다양한 언어 Compiler 확장, 알고리즘 문제 제공으로 현재 SoongPiler의 아쉬운 내용을 보충해 나가는 방향으로 버전 관리할 것이다.