

Proposal on the Improvement Direction of Web App Development lecture for Non-IT majors

Koono Kim*

*Professor, Global Leadership School, Handong Global University, Pohang, Korea

[Abstract]

In this paper, I analyze the difficulties of web service development education for non-IT majors through the Q&A of students posted in the k-mooc lecture, and propose methods to improve them. Through Q&A analysis, it was confirmed that non-majors mainly had difficulties in using unfamiliar tools, cost of cloud service, setting up server environment, and writing code while taking web service development courses. To solve this problem, I propose a method to reduce the server cost problem and the complexity of the server environment by using BaaS(Backend as a Service), which is one of the cloud service models. It also shows that it is possible to reduce the length of code that needs to be written at once by using the React library to modularize long code into smaller units. Finally, I propose an improvement plan that even non-IT majors can easily learn by implementing a web application that works by using the design output obtained by using Figma.

▶ **Key words:** Coding education, Web service development, Cloud service, Web framework, Javascript

[요 약]

본 연구에서는 비전공자들을 대상으로 한 웹서비스 개발 교육의 어려움을 k-mooc 강의에 게시된 수강생들의 질의응답을 통해 분석하고 이를 개선하기 위한 방법들을 제안한다. 질의 응답 분석을 통해 비전공자들은 웹서비스 개발 강좌를 수강하면서 주로 익숙하지 않은 도구 사용, 클라우드 서비스의 비용 문제, 서버 환경 설정, 코드 작성에서 어려움을 보이는 것으로 확인되었다. 이를 해결하기 위해 클라우드 서비스의 모델 중 하나인 BaaS (Backend as a Service)를 활용해 서버 비용에 대한 문제와 서버 환경의 복잡도를 낮출 수 있는 방법을 제안한다. 또한 리액트 라이브러리를 사용하여 긴 코드를 작은 단위로 모듈화하여 한번에 작성해야 할 코드 길이를 줄일 수 있다는 것을 보였다. 마지막으로 피그마를 활용해 얻은 디자인 산출물을 활용하여 동작하는 웹 애플리케이션을 구현함으로써 IT 비전공자들도 쉽게 학습할 수 있는 개선안을 제안한다.

▶ **주제어:** 코딩교육, 웹서비스 개발, 클라우드 서비스, 웹 프레임 워크, 자바스크립트

• First Author: Koono Kim, Corresponding Author: Koono Kim
*Koono Kim (kok@handong.edu), Global Leadership School, Handong Global University
• Received: 2022. 03. 30, Revised: 2022. 04. 19, Accepted: 2022. 04. 19.

I. Introduction

1. Background

최근 소프트웨어 개발자의 처우 개선과 IT 기업들의 개발자 수요 증가의 이유로 코딩을 배우려는 비전공자의 수요가 높아지고 있다. IT 관련 직종이 아니더라도 다양한 업무 분야에서 소프트웨어 개발 및 활용 수요는 계속 증가하는 추세이다. 4차 산업혁명 시대와 함께 인터넷과 모바일 앱 사용이 증가하면서 소프트웨어 개발의 여러 분야 중에도 특히 웹서비스 개발 분야는 소프트웨어 개발자의 공급과 수요 양면에서 주목받는 분야이다. 웹서비스 개발의 모든 부분을 담당할 수 있는 풀 스택 엔지니어는 LinkedIn의 2020 Emerging Jobs Report[1]에서 2015년 이후 35% 이상의 고용 성장을 기록한 네 번째로 빠르게 성장하는 직업으로 선정되기도 하였다.

반면 웹서비스 개발은 그 분야가 매우 방대하여 전문적인 소프트웨어 교육받지 않은 비전공자가 접근하기에 쉽지 않은 분야이다. 코딩 경험이 없는 비전공자가 스스로 앱을 개발해 출시 또는 사용할 수 있는 수준까지 개발하는 데는 서버 구축, 프론트 앤드 개발, 백 앤드 개발, 앱 패키징에 이르는 포괄적인 주제를 다루어야 하므로 어려움이 많다. 웹서비스 개발의 단편적인 주제를 다루는 온 오프라인 강좌들은 쉽게 찾아볼 수 있지만 비전공자가 이해할 수 있을 만큼 쉬우면서 동시에 웹서비스 개발의 전반적인 부분을 다루어 볼 수 있는 강좌들은 쉽게 찾아보기 힘들다.

이러한 산업 동향과 비전공자들의 코딩 교육 수요를 충족하기 위하여 한국형 온라인 MooC 플랫폼인 K-MooC에 “HTML에서 웹앱까지”라는 강좌를 개발하여 운영하였다. Table 1은 5학기 운영하는 동안 2,000명 이상의 학생들이 수강 및 청강한 데이터이고 수료율은 타 온라인 강좌 대비 높은 수료율을 보인다[2]. 이러한 실적을 바탕으로 2021년에는 K-MooC+ 사업에 선정되었지만, 빠르게 변화하는 개발 트렌드와 수강생들의 피드백을 반영하여 비전공자들에게 쉽게 다가갈 수 있는 강좌가 되기 위해 개선할 필요가 생겨났다.

본 연구에서는 K-MooC 플랫폼에서 운영 중인 “HTML에서 웹앱까지” 과목을 5학기 동안 운영하며 얻어진 수강생 피드백을 바탕으로 비전공자 대상 웹서비스 개발 관련 과목교육의 어려운 부분을 분석하고 이를 해결하기 위한 강의 개선방안을 제안한다.

Table 1. Number of students enrolled in “From HTML To Webapp” course

semester	number of enrolled	number of auditors	number of graduates	total number
2019-2	245	136	0	381
2020-1	250	172	51	422
2020-2	227	177	44	404
2021-1	391	143	52	534
2021-2	246	63	44	309

2. Related works

비전공자를 위한 코딩 교육 역시 다방면으로 연구되고 있다. 먼저, 알고리즘 씽킹 기반으로 설계된 SW 교과 과정의 효과성을 분석하기 위해 논리적 사고력, 창의적 잠재력, 논리적 잠재력의 3가지 분야를 측정하고 교육 효과를 분석한 연구[3]와 여러 대학의 기존 SW 교육 커리큘럼을 분석하고 IT 전공 학생과 비전공 학생 간의 격차를 해소하기 위한 새로운 교육 커리큘럼을 제안한 연구가 수행되었다[4]. 또한 코딩에 어려움을 느끼는 학생들의 문제점을 파악하기 위해 학생들의 독서에 대한 습관과 태도를 측정하기 위한 조사를 수행하고 그 관련성을 분석하는 연구도 진행되었고[5] 기초 프로그래밍 언어 학습자의 개념적 이해를 돕기 위해 메모리 디버거를 이용함으로써 논리적 오류를 줄이고 프로그램 작성 능력을 향상할 수 있다는 것이 증명되었다[6]. 그 외에도 비전공자들의 코딩 교육을 위한 학습자의 시각적 문해력과 사고 유형에 따른 연구[7]와 디자인 씽킹 기반의 코딩 교육 효과 분석 연구[8], 그리고 프로그램의 시각적 시뮬레이션을 활용하는 기초 프로그래밍 학습자를 위한 교육적 기법을 공식화하고 평가하기 위한 다양한 시도도 있었다[9].

이러한 연구들에 기초하여 본 연구에서는 웹서비스 개발이라는 특정 교육 목표를 가지고 실제 학습자들의 어려움을 질의응답을 통해 분석하고 구체적인 교과목의 구성을 제안하였다는 점에서 다른 연구들과는 차별점을 두었다.

II. Problem Description

1. Composition of existing subject contents

“HTML에서 웹앱까지” 과목은 프로그래밍 교육을 받은 적이 없거나 컴퓨터 공학을 전공하지 않은 학생들을 대상으로 웹 서버와 HTML과 같은 웹 프로그래밍의 기본적인 부분부터 시작하여 실제 동작하는 예제를 제작함으로써

웹 프로그래밍의 전반적인 개발 프로세스를 이해하고 나아가 스스로 하나의 서비스를 개발할 수 있는 단계를 그 교육 목표로 하는 과목이다.

단순한 따라 하기에 그치지 않고 웹 애플리케이션이 동작하는 원리와 개발 흐름을 익힐 수 있도록 웹의 표준 통신 프로토콜인 HTTP에 대한 소개부터 프로그래밍 언어를 이용한 웹 프론트 앤드 및 백 앤드 서비스를 개발까지 웹 서비스 개발의 전반적인 부분을 강의에 담고자 하였다. 비전공자에게 여러 가지 언어를 사용해 개발하는 것은 매우 큰 부담이 되므로 프론트 앤드, 백 앤드, 하이브리드 앱 개발에 모두 사용할 수 있는 자바스크립트를 주요 학습 언어로 선정하였다. 실제로 자바스크립트는 전 세계의 실무 개발자들이 가장 많이 이용하는 커뮤니티인 스택 오버플로우에서 2021년에 실시한 설문조사에서 개발자들이 가장 좋아하는 언어 1위에 선정되기도 하였다[10]. 또한 개발자들이 가장 원하는 프레임워크 상위 1위부터 3위까지가 모두 자바스크립트 기반으로 만들어진 프레임워크인 것으로 조사되었다.

강좌의 전반부에서는 웹앱의 기본 개념과 AWS (Amazon Web Server)를 활용해 실습할 수 있는 웹 서버를 설치하는 과정을 다룬다. 이후 웹 서버에 접속하기 위한 ssh, scp 툴 사용법을 학습한다. 그 후 HTML과 CSS와 같이 브라우저에서 렌더링되는 페이지를 만들기 위한 기술들을 학습하고 AWS에 구성해 놓은 서버에 호스팅한다. 강의 중반부에서는 동적인 웹 페이지를 만들기 위해 꼭 필요하고 후반부에 배우게 될 백 앤드 서비스 개발에 사용하게 될 프로그래밍 언어인 자바스크립트에 대해 집중적으로 학습한다. 강의 후반부에서는 백 앤드 서비스를 제작하기 위한 기술들과 하이브리드 웹앱을 만들기 위한 기술인 Cordova 프레임워크에 대해 학습하는 것으로 강의안이 만들어졌다.

2. Problem Domain

본 연구에서는 비전공자 대상 웹서비스 개발 프로그래밍 교육의 어려움과 문제점을 분석하기 위해 “HTML부터 웹앱까지” 과목의 전체 질의응답 게시글을 기타(과제, 이수 등) 질문을 제외한 툴 사용, 서버 환경, 서버 비용, 코드 관련 4가지 클래스로 분류하고 그 비율을 조사하였다. Table 2는 이를 조사한 결과이다. 이를 바탕으로 이 과목을 수강하는 학습자의 관점에서 문제점을 3가지 측면으로 분석하였다.

Table 2. Question and answer rate by semester

semester	tool use	server environment	server cost	code related
2019-2	0%	75.0%	25.0%	0%
2020-1	19.3%	40.3%	14.5%	25.8%
2020-2	16.6%	38.8%	11.1%	33.3%
2021-1	6.2%	50.0%	37.5%	6.2%
2021-2	22.2%	44.4%	22.2%	11.1%

2.1 Usage of tools

“HTML부터 웹앱까지” 과목에서는 AWS에 직접 리눅스 인스턴스를 실행하여 과제를 수행하도록 하였다. 이를 위해 SSH 접속을 위한 putty 프로그램과 ftp, scp 등의 프로토콜로 파일을 전송하기 위한 winscp 프로그램을 사용하였다. 툴 사용 범주의 질문의 내용 중 윈도즈 환경이 아닌 애플 mac 등을 사용하는 수강생들이 해당 프로그램을 사용할 수 없다는 것이었다. 물론 이러한 툴들을 사용하지 않고도 ssh 나 scp 연결을 수립할 수 없는 것은 아니고, 다른 운영체제를 위한 여러 가지 툴들이 인터넷상에 공개되어 있긴 하지만 비전공자 입장에서는 강의 내용을 그대로 따라 하지 못하는 상황이 왔을 때 몹시 어려워하고 강의 내용 자체에 불만을 가질 수 있다는 것을 알 수 있었다. 또한 어떤 툴들은 강의를 제작하던 시점의 버전과 학기를 진행하면서 수강생이 사용하고 있는 버전이 상이하여 어려움을 호소하는 수강생의 질문들도 다수 등록되었다.

2.2 Server environment and cost

클라우드 서비스는 값비싼 하드웨어를 구매하고 관리할 필요를 없애주고 컴퓨팅 파워와 같은 리소스들을 필요할 때 바로 제공할 수 있다는 점에서 최근 개발 추세에서는 빠질 수 없는 요소가 되었다. 초기 비용이 없고 리소스를 대여한 시점부터 반납한 시점까지의 시간에 대해서만 과금이 되기 때문에 웹서비스 개발 시 필요한 자본금을 크게 줄일 수 있다. 그중 가장 높은 점유율을 가지고 있는 서비스는 아마존 웹서비스(AWS)이다[11]. AWS는 가입 후 1년 동안 프리티어 한도 내에서의 리소스를 무료로 사용할 수 있지만 프리티어 한도를 초과한 비용에 대해서는 과금이 이루어져야 하므로 AWS는 최초 가입 시 해외 결제가 가능한 카드 정보를 입력받는다. 스스로 리소스 사용량을 잘 관리하지 않으면 큰 비용이 청구될 수 있으므로 서버 비용과 관련된 질문이 상당수 등록되었다. AWS에서는 사용자가 명시적으로 리소스를 반납하지 않으면 비용이 결제되지 않더라도 즉시 리소스를 사용 불가 상태로 전환되지 않도록 하고 있다. 이는 서비스 중단이 큰 비즈니스적 손실

이 될 수 있는 프로덕션 환경에서는 적합한 정책이지만 교육용으로 사용할 때는 수강생의 주의를 꼭 필요하다 보니 클라우드 서비스에 익숙하지 않은 비전공자들에게는 큰 어려움이 있을 수 있다.

AWS 또한 강의를 제작한 시점과는 그 기능과 UI/UX 측면에서 많은 변경과 개선이 있었다. 리소스를 대여하고 사용한다는 큰 기능 요건들의 변화가 있는 것은 아니지만 이러한 변경이 클라우드 용어나 개념에 익숙하지 않은 수강생들에게는 많은 혼란을 줄 수 있다는 것도 확인되었다. 실제로 가장 큰 기능 요건의 변화는 컴퓨팅 리소스를 대여할 수 있는 EC2 서비스의 가상화 OS 이미지 중 AWS에서 공식적으로 제공하는 이미지인 Amazon AMI 가 버전 2로 변경되었다는 점이다. 이에 따라 기존에 강의 내용에 있던 이전 버전의 Amazon AMI를 사용할 수 없게 되었다. Amazon AMI 2를 사용해도 강좌의 실습을 진행하는 데 큰 무리가 없지만, 수강생들에게 어느 정도 혼동이 될 수 있는 변경이었다.

IaaS(Infrastructure as a Service)는 인터넷을 통해 가상화 OS 등의 컴퓨팅 리소스를 엔드 유저에게 제공하는 클라우드 서비스이다. 가상화 환경에서는 대부분 수강생들에게 익숙한 윈도우 환경보다는 리눅스 환경이 주로 사용된다. 리눅스 환경은 윈도우 환경과 달리 라이선스 비용이 필요 없고 최근에 주목받고 있는 컨테이너 기술을 통해 더욱더 편리하게 호스트 OS와 격리된 가상화된 컨테이너를 구축할 수 있다는 큰 장점이 있다. 하지만 리눅스 환경 자체가 수강생들이 실습을 진행하는데 진입장벽이 될 수 있다는 것은 당연한 사실이다. GUI(Graphic user interface) 환경에 익숙한 수강생들은 CLI(Command-line Interface) 환경에서 복잡한 명령어를 입력하고 명령 수행 결과를 확인하는데, 어려움을 겪는다. 그 결과가 사용자 친화적으로 즉각 표시되는 GUI 환경과 달리 비전공자 수강생들은 CLI 환경에서 실습을 진행하는 도중 명령어에 오타자에 의한 에러나 apt, yum과 같은 패키지 라이브러리 등의 변화로 인해 발생할 수 있는 오류를 대처할 수 없었다. 실제로 질문 중 가장 높은 비율을 차지한 부분이 서버 환경과 관련된 부분이고 이를 해결해 주기 위해 질의응답 게시판만으로는 부족한 경우가 많아 직접 수강생의 인스턴스에 접속하여 문제의 원인을 파악하고 해결해 주어야 하는 경우가 자주 발생하였다.

2.3 Code related

강좌의 주된 프로그래밍 언어로 사용한 자바스크립트는 프로그래밍 언어의 분류 중 고수준 언어에 속한다. 고수준 언어에 가까울수록 사람이 이해하기 쉽고 사용 중 메모리

나 스택과 관련된 에러 같은, 운영체제 레벨에서 발생할 수 있는 오류가 발생할 확률이 저수준 언어에 비해 현저히 낮다. 그렇다고 해서 코딩을 처음 배우는 수강생 관점에서 코딩은 그렇게 쉽게 접근할 수 있는 분야는 아니다. 파이썬이나 자바스크립트와 같은 스크립트 언어 하나만으로도 한 권의 책이나 강좌로 다루어야 할 만큼 다루어야 할 내용이 적지 않은 분야이다. 그러다 보니 강좌 내용을 복잡한 컴퓨터 알고리즘이나 로직보다는 배열의 맵, 필터, 리듀스 연산과 같은 값 조작 부분에 관한 내용을 주로 담고자 하였다. 실제로 웹서비스 개발에서는 컴퓨터 알고리즘보다는 데이터를 처리하고 클라이언트에게 적절한 형태로 변환하여 반환하는 작업이 많은 부분을 차지한다. 코드 관련으로 분류한 질의응답 중에서 언어와 문법에 관한 질문은 거의 없었던 것으로 보아 이러한 의도는 수강생들이 강의를 진행하는데 잘 맞았다고 볼 수 있었다.

코드 관련 질문은 주로 강좌 후반부에 실전 예제 프로그램을 제작해보는 실습을 다루는 부분에서 많이 등록되었다. 이는 온전히 동작하는 프로그램을 예제로 사용하다 보니 자연스럽게 코드가 길어지게 되고 온라인 강좌의 특성상 한 화면에 긴 코드를 담을 수 없다는 데서 비롯되었다. 수강생들의 질문에서는 긴 코드의 흐름을 따라가지 못해 놓치는 부분이 생겨 문제가 발생한 경우가 다수 있었다. 또한 코드가 프런트 앤드와 백 앤드로 나뉘어 있다 보니 전체적인 애플리케이션에 동작 흐름과 작성한 코드의 동작 순서를 혼동하는 예도 자주 발생하였다.

프런트 앤드 실습에서는 애플리케이션의 여러 컴포넌트가 하나의 HTML과 자바스크립트 파일에 함께 존재하여 영상 강의를 실시간으로 들으며 따라가기에는 어려움이 있다는 의견이 있었다. 이를 해결하기 위해 지문으로 된 교재를 제작하여 수강생들에게 제공하였지만, 전체 코드를 지문에 실을 수 없고 긴 코드의 여러 부분을 나누어 설명할 수밖에 없어 한계가 존재함을 알 수 있었다.

III. The Proposed Methods

II 장에서 기술한 문제점들을 해결하기 위해 수강생들이 공통으로 어려움을 보였던 클라우드 서버 환경과 리눅스를 활용하여 웹 서버 등의 패키지를 설치하는 과정을 생략하고 BaaS(Back-end as a Service)와 같이 백 앤드 애플리케이션 개발을 단순화해주는 클라우드 서비스를 활용한 강좌 구성을 제안한다. 또한 React library를 활용해 실습 예제를 컴포넌트 단위로 구성하여 강좌 진행을 컴포

먼트 수준에 맞추어 Top-down 방식으로 재구성하고 수 강생들이 집중해서 봐야 할 코드의 범위를 줄이고자 한다.

3.1 BaaS (Back-end as a Service)

클라우드 서비스는 컴퓨팅 리소스, 스토리지, 네트워크 등의 서버 인프라들을 가상화하여 사용자에게 대여해 주는 방식의 IaaS(Infrastructure as a Service) 방식으로 시작하여 최근에는 미리 준비된 컴퓨팅 리소스, 네트워크 설정들과 함께 애플리케이션에 특화된 런타임 환경을 제공하여 웹 애플리케이션 워크로드를 즉시 실행할 수 있는 플랫폼을 대여해 주는 PaaS(Platform as a Service) 클라우드 서비스로 발전하였다. 더 나아가서 인증, 도큐먼트 데이터베이스, 클라우드 메시징 등 대부분의 웹 애플리케이션 워크로드에서 공통으로 사용하는 백 앤드 서비스들을 API(Application Programming Interface) 혹은 SDK(Standard Development Kit) 형태로 제공하는 BaaS(Back-end as a Service) 형태의 클라우드 서비스가 각광을 받고 있다[12].

AWS Amplify, Google Firebase와 같은 BaaS 서비스를 이용하면 풀 스택 애플리케이션을 쉽고 빠르게 구축하고 확장할 수 있는 형태로 배포할 수 있다[13, 14]. BaaS 서비스는 프로덕션 레벨에서도 적극적으로 사용되는 추세이지만 코딩을 전문적으로 배우지 않은 비전공자 수강생들이 인증, 데이터 영속화 등의 구현이 어렵고 복잡한 시스템 설계를 요구하는 기능들을 몇 줄의 코드로 간단하게 구현하여 실습할 수 있다. 기존 “HTML부터 웹앱까지” 과목의 전반부에 구성하였던 클라우드 서버 환경 구축 및 사용에 해당하는 부분을 삭제하고 정적 웹 사이트 제작을 다루는 실습 예제를 Firebase Hosting을 이용하여 배포하도록 하여 복잡성을 크게 줄일 수 있다. 또한 NodeJS를 활용한 백 앤드 API 제작 관련 부분도 모두 BaaS에서 제공하는 기능을 활용하는 것으로 대체하고 나머지 부분들을 기존 강좌에서 그 난이도와 강좌 시간의 제약 때문에 넣지 못했던 인증 및 데이터 영속화 부분을 추가함으로써 실습 예제의 완성도를 높일 수 있다.

3.2 React Library

“HTML부터 웹앱까지” 과목에서는 프론트 앤드 실습 예제를 순수한 HTML 및 CSS와 jQuery 자바스크립트 라이브러리를 사용하여 작성하였다. 이러한 개발 스택은 웹 프론트 앤드 프로젝트를 모듈 단위로 나누기 힘들어 코드 길이가 강의 자료 및 영상에 담기에 너무 길어진다는 단점이 있었다. 이를 해결하기 위해 웹 프론트 앤드 프레임워크

들을 도입하기로 하고 React, Angular, Vue, Svelte 네 가지 자바스크립트 라이브러리 및 프레임워크들을 후보군으로 비교하였다[15, 16, 17, 18]. 일반적으로 생산성, 개발 커뮤니티 활성화 정도 등을 고려하여 개발 스택을 선정하지만, 교육 목적의 기술 스택에서는 개발 난이도가 쉬워야 하고 자바스크립트 언어 기능을 벗어나는 템플릿 문법 등이 없이도 개발 가능한가를 가장 중요하게 고려하였다. React 라이브러리는 이 두 가지 요건을 모두 충족하고 현업에서도 활발히 사용되고 있어 기존 강좌의 개선방안으로 프론트 앤드 기술 스택을 제안하였다.

리액트(React) 라이브러리의 함수형 컴포넌트(Functional component)는 UI 구성요소를 간편하게 모듈화할 방법을 제공한다. UI 구성요소의 페이지, 콘텐츠 영역, 헤더, 푸터와 같은 큰 단위의 레이아웃 구성요소뿐만 아니라 리스트, 카드, 버튼과 같은 작은 단위의 디자인 컴포넌트들도 리액트 라이브러리의 함수형 컴포넌트로 모듈화하면 코드를 작성하는 학습자에게 중요하지 않은 부분을 감출 수 있고 UI 구성요소의 한 부분만 독립적으로 개발할 수 있으므로 강의 자료화면에 담아야 할 코드의 양을 줄일 수 있고 수강생들에게 강의 흐름에 더 집중할 수 있도록 하는 효과를 얻을 수 있다. Fig. 1은 리액트를 사용하지 않고 순수한 HTML 코드만으로 이루어진 예제 코드와 리액트를 사용하여 UI 요소를 모듈화한 코드를 비교한 것이다. (a)의 경우 코드의 길이가 매우 길어 강의 슬라이드 한 장에 넣기 힘들고 수강생들이 화면을 보고 실습을 진행하기에 어려움이 있다. 반면 예제 코드 (b)는 같은 기능을 하는 코드를 리액트 컴포넌트를 이용해 구조화하여 (a)와 비교했을 때 코드의 길이가 크게 줄어들고 각 컴포넌트가 어떤 기능을 하게 될지 쉽게 예상할 수 있다. 이처럼 리액트를 이용해 UI 컴포넌트를 모듈화하면 수강생들에게 할 일 목록(To-do list)이나 필터와 같은 세부적인 사항을 감추어 두고 실습 예제 애플리케이션의 메인 페이지에 집중하여 코드를 작성할 수 있도록 해 준다.

```

<section>
  <div class="hello">
    <h2>Hi Stepanie.</h2>
    <p>Good Morning</p>
  </div>
  <ul class="group-tasks">
    <li>
      <div class="card">
        <p>13 tasks</p>
        <h3>TATTOO Project Make Icon</h3>
      </div>
      <div class="participants">
        
        
        <span>+3</span>
      </div>
    </li>
    <li>
      <div class="card">
        <p>13 tasks</p>
        <h3>TATTOO Project Make Icon</h3>
      </div>
      <div class="participants">
        
        
        <span>+3</span>
      </div>
    </li>
  </ul>
</section>
<section>
  <div class="filter">
    <div class="active">Task List</div>
    <div>Completed</div>
  </div>
</section>

```

(a)

```

<Container>
  <Hello>
    <h2>Hi Stepanie.</h2>
    <p>Good Morning</p>
  </Hello>
  <GroupTasks />
  <TaskFilters />
  <Tasks />
</Container>

```

(b)

Fig. 1. Comparison of (a) example code written in pure HTML and (b) structured UI code as a functional component

또한 리액트 라이브러리는 UI 컴포넌트를 선언적으로 개발할 수 있도록 해 준다. 이는 자바스크립트 객체와 리액트 라이브러리의 상태 관리 API를 이용하면 애플리케이션의 상태를 결정하는 데이터만 관리해 주면 DOM (Document object model)을 직접 제어하여 UI를 업데이트해 줄 필요가 없어 코드가 더욱 간결해지고 UI 컴포넌트를 데이터 주도적으로 개발할 수 있도록 해 준다. 핵심 비즈니스 로직과 관계없는 DOM 요소를 선택하고 업데이트 하는 코드를 작성하지 않아도 되기 때문에 수강생들에게 애플리케이션의 핵심 비즈니스 로직에 대한 코드에만 집중할 수 있도록 해 주어 학습 효과를 높일 수 있다. Fig. 2

에서 볼 수 있듯이 DOM 조작 코드 없이도 리액트의 함수형 컴포넌트의 인자로 전달된 리액트 프롭스가 변경되면 UI 업데이트는 리액트에서 자동으로 처리된다.

```

export const Task: React.FC<{ title: string, info: string }> = ({ title, info }) => {
  return <TaskItem>
    <div>
      
      <div>
        <label>{ title }</label>
        <p>{ info }</p>
      </div>
    </div>
    </img>
  </TaskItem>
}

```

Fig. 2. Removed DOM manipulation code by using declarative UI

3.3 Using Figma

파편화 되어 있던 웹 기술 표준이 웹 2.0과 함께 정리되고 HTML5의 등장으로 다양한 OS 네이티브 기술들을 웹 브라우저에서 활용할 수 있게 되면서 웹서비스의 활용 분야는 점점 넓어지고 있다. 이에 따라 웹서비스 개발에 필요한 기술은 기술 스택의 각 분야에서 점점 고도화 되어가고 한 사람의 개발자가 다루기에는 그 분야가 점점 넓어지고 있다. 웹서비스 개발에 여러 분야의 전문가들이 필요하게 되면서 각 분야의 전문가들의 원활한 소통도 프로젝트의 성공을 가를 수 있는 요소 중 하나로 평가되고 있다. 웹서비스 개발 프로세스에서 개발자는 주로 웹 디자이너의 작업 결과물을 받아 작업을 진행하게 된다. 제플린, 피그마 같은 도구를 활용하면 디자인 산출물을 추가적인 작업 없이도 웹상에 배포하여 개발자가 활용할 수 있도록 해준다. 이를 통해 수강생들은 여백, 글꼴 등과 같은 애플리케이션의 디자인 요소를 쉽게 가져다 사용할 수 있다. 또한 실습 예제의 디자인적 완성도를 높여 실습 예제를 제작하는 수강생들의 흥미를 끌어 올릴 수 있는 효과도 기대할 수 있다.

3.4 Implementation and experimentation

강좌 구성과 함께 다수의 작은 예제로 나누어져 있던 실습 예제를 하나의 일관되고 완성도 있는 풀 스택 웹 애플리케이션으로 구현하여 대체하였다. 이를 통해 수강생들이 강좌를 진행하며 웹 개발 기술 스택이 하나의 웹서비스에서 어느 곳에 사용되는지 자연스럽게 깨우칠 수 있도록 하였고 하나의 예제를 점진적으로 발전시켜가면서 학습 방향의 통일성을 유지할 수 있도록 하였다.

실습 예제로 사용한 애플리케이션은 사용자 인증과 협업 기능이 추가된 태스크 매니지먼트 애플리케이션이다. Fig. 3은 구현한 완성된 실습 애플리케이션을 스크린 캡처한 그

림이다. BaaS 서비스를 이용해 복잡도 및 난이도를 조절하면서도 인증, 문서화된 데이터베이스를 통한 데이터 영속화와 같은 기존 강의에서 다루지 못했던 기능들을 활용하여 실습 예제를 실전 프로젝트에 가깝게 구성할 수 있었다.

실습 예제는 일반적인 태스크 매니지먼트 애플리케이션과 비슷하게 태스크에 대한 제목, 기한, 설명과 함께 태스크를 생성할 수 있다. 이에 더해 이 태스크와 관련된 다른 사용자를 참여자로 등록할 수 있다. 참여자로 등록된 사용자는 생성된 태스크를 확인할 수 있고 태스크의 상태를 공동으로 업데이트할 수 있다. 또한 태스크들을 그룹화 하여 하나의 프로젝트로 묶을 수 있는 기능도 제공한다. 이러한 기능들을 통해 수강생은 데이터 영속화에서 엔티티의 개념과 엔티티 간의 관계에 대해서 학습할 수 있다.

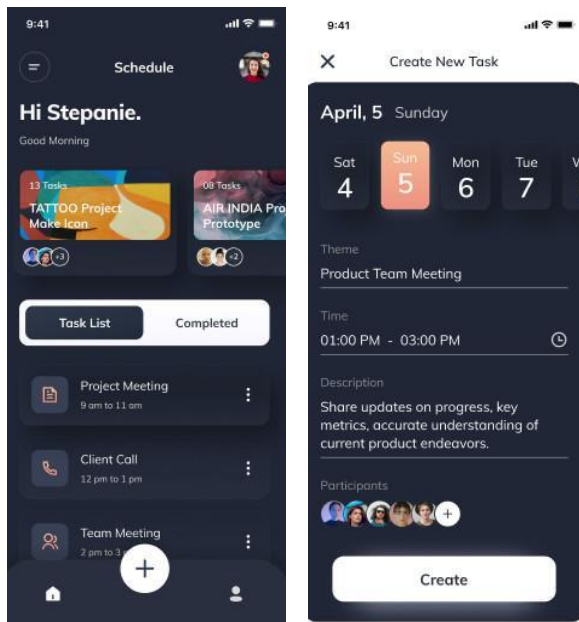


Fig. 3. Screen capture of hands-on example application

수강생은 모든 내용을 학습한 뒤 한 번에 완성된 실습 예제 애플리케이션을 프로그래밍하는 것이 아니라 강의를 진행하면서 점진적으로 제작해 볼 수 있도록 강의를 구성하였다. 강의 초반부에서는 시맨틱 HTML 코드를 통해 애플리케이션의 구성요소를 그 의미에 맞는 태그를 활용하여 구성하고 Firebase 호스팅을 통해 배포한다. CSS를 학습한 뒤에는 의미 구조에 색상, 크기와 같은 단순한 디자인 요소들을 적용해 볼 수 있다. 레이아웃과 관련된 스타일 어트리뷰트를 학습한 후에는 디자인적으로 완성된 애플리케이션을 제작해 볼 수 있다. 자바스크립트 기본 문법과 여러 가지 자료형에 대해 학습한 후 애플리케이션에 필요한 상태를 자바스크립트의 자료형을 이용해 정의하고 상

태에 따라 UI를 업데이트하여 동적인 웹 애플리케이션을 제작해 볼 수 있다. 리액트를 통해 UI 컴포넌트를 구조화하고 상태에 따른 UI 업데이트를 리액트를 통해 처리하는 방법에 대해 학습하게 된다.

마지막으로 Firebase의 다른 기능인 인증과 Firestore를 이용하여 상태를 영속화하고 로그인된 사용자에게 따라서 적절한 태스크를 조회하는 방법에 대해 배우고 실습함으로써 실습 예제 애플리케이션을 완성하게 된다.

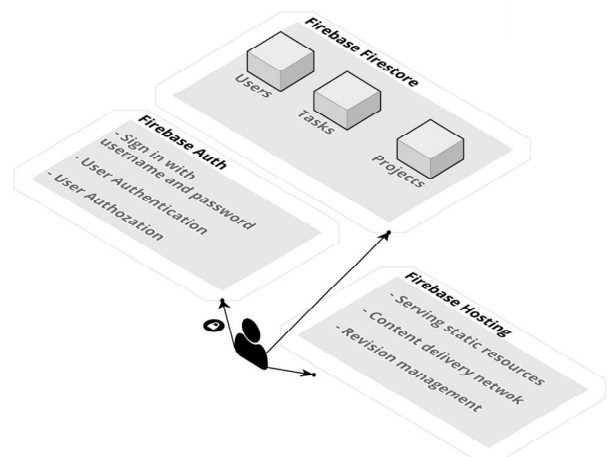


Fig. 4. System architectures of the implemented application

Fig 4 는 실습 예제 애플리케이션의 시스템 아키텍처를 도식화 한 것이다. 애플리케이션은 정적인 웹 리소스의 형태로 Firebase Hosting에 배포된다. 수강생은 리눅스 환경이나 웹 서버에 대한 지식이 없이도 애플리케이션을 전세계에서 접근할 수 있도록 손쉽게 배포할 수 있다. Firebase Hosting은 배포된 애플리케이션의 리버전을 관리하여 언제든지 이전 버전으로 배포를 롤백할 수 있도록 도와준다. 사용자는 Firebase Auth를 통해 아이디와 비밀번호로 인증하고 자신이 등록한 태스크에 대한 접근 및 조작 권한을 애플리케이션에 허용한다. Firebase Auth를 통해 발급된 인증 토큰은 Firebase Firestore에 작성된 ACL(Access control list)을 통해 허가된 리소스에만 접근할 수 있다. Firebase Firestore는 문서화된 데이터베이스이고 컬렉션을 통해 다수의 문서를 저장하고 쿼리할 수 있는 기능을 제공한다.

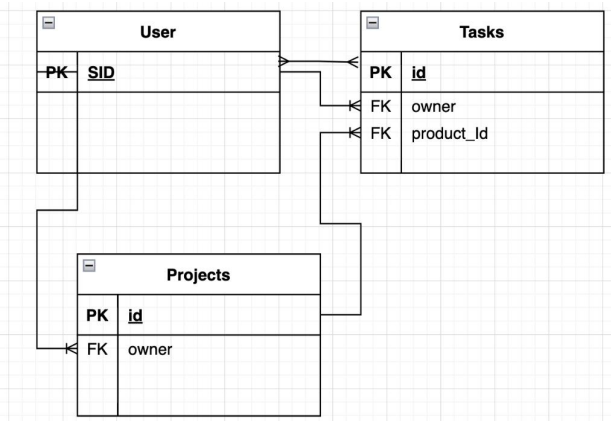


Fig. 5. Entity relationship of the implemented application

Fig. 5는 실습 예제 프로그램의 엔티티 다이어그램이다. 사용자는 다수의 할 일 목록을 생성하고 관리할 수 있다. 할 일 목록은 다수의 사용자를 참가자로 등록할 수 있다. 또한 사용자는 프로젝트를 만들어 다수의 할 일 목록을 등록하고 관리할 수 있다.

IV. Conclusions

본 연구에서는 2021년도 k-mooc+ 사업에 연속 선정된 “HTML부터 웹앱까지” 과목을 기본 모델로 하여 비전공자가 웹서비스 개발을 학습할 때 어려움으로 느끼는 점들을 Q&A 내용들을 기준으로 분석하였다. 그리고 비전공자의 관점에서 분석된 자료를 기반으로 어려운 강의 부분들을 좀 더 쉽게 학습할 수 있도록 강의 전반적인 내용을 수정하는 방안들을 제안하였다. 수강생들이 어려움을 느끼는 AWS 사용과 리눅스 서버 환경을 다루는 내용은 과감히 삭제하였고 최신 트렌드에 맞는 프론트 앤드 라이브러리를 이용해 웹서비스의 작은 컴포넌트들을 Top-down 방식으로 개발하며, 강의 내용에서 한 번에 다루는 코드의 양을 줄이고 지금 개발하고 있는 부분에 초점을 맞출 수 있도록 개선하도록 제안하였다. 또한 이렇게 개선된 내용을 통해 조금 더 실무와 가까운 예제를 구현할 수 있게 되었고, 비전공자에게 맞지 않는 수준의 난이도 때문에 다룰 수 없었던 인증, 데이터 영속화, 디자이너-개발자 협업 툴 등의 분야도 다룰 수 있게 되어 교양과목으로 강좌를 수강하는 수강생들뿐만 아니라 전문 개발자를 목표로 하는 비전공자 수강생들에게 필요한 내용도 충분히 포함하도록 하였다.

본 연구를 진행한 시점에서는 아직 강좌가 운영되지 않아 수강생들의 직접적인 피드백은 얻을 수 없었기 때문에 본 연구에서 제안한 내용을 정량적으로 평가할 수 없었다.

향후 연구에서는 웹서비스 개발의 성취도를 정량적으로 평가할 수 있는 평가 기준을 만들고 본 연구에서 제안한 내용을 바탕으로 강의를 운영하면서 얻어진 피드백을 바탕으로 개선한 교육 과정을 평가하고 비전공자에게 적합한 웹서비스 개발 강의 모델을 제시하고자 한다.

REFERENCES

- [1] 2020 EmergingJobs Report, https://business.linkedin.com/content/dam/me/business/en-us/talent-solutions/emerging-jobs-report/Emerging_Jobs_Report_U.S_FINAL.pdf
- [2] YeSeon Bae, "Study on MOOC status and improvement plan", *Journal of The Korean Society of Internet Information*, Vol. 17, No. 2, pp. 27-33, Dec. 2016.
- [3] Hyejin Jeong, "Analysis of Algorithm Thinking based on Basic Software Education Effectiveness for non-major students in SW college", *The Korean Association of General Education*, Vol 15, No 2, 175-187, Aug. 2021.
- [4] Moonseong Kim, Hyunchul Park and Woochan-Lee, "Advanced Software Education Curriculum for Non-IT Major University Students", *Journal of the Korea Society of Digital Industry and Information Management*, Vol. 15, no. 3, Sep. 2019.
- [5] Daeyoung Na and Koono Kim, "Coding Education Academic Achievement Analysis According to Reference Book and Type of Reading", *Journal of The Korea Society of Computer and Information*, Vol. 26, No. 12, pp. 323-330, Dec. 2021
- [6] M. Nirgude, "Debugger Tool to Improve Conceptual Understanding of Programming Language of Novice Learners", *IEEE Fifth International Conference on Technology for Education*, pp. 69-72, Dec. 2013.
- [7] Chan Jung Park, Jung Suk Hyun, "Analysis of the Effects of Learners' Visual Literacy and Thinking Patterns on Program Understanding and Writing in Basic Coding Education for Computer Non-majors", *The Journal of Korean Association of Computer Education*, Vol 23, No. 2, pp. 1-11, Mar. 2020.
- [8] Yoonee Shin, Hyojung Jung, Suh Eungkyo, "Effect of Coding Education Program based on Design Thinking for Non-engineering students", *The Journal of Learner-Centered Curriculum and Instruction*, Vol 19, No. 10, pp. 351-373, May. 2019.
- [9] J. Sorva, "Visual Program Simulation in Introductory Programming Education", *Doctoral Dissertation, Aalto University, Helsinki*, Feb. 2012.
- [10] 2021 Developer Survey, <https://insights.stackoverflow.com/survey/2021#most-popular-technologies-language>
- [11] Digital today, <https://www.digitaltoday.co.kr/news/articleView.html?idxno=423104>
- [12] J. Y. Lee, "An Analysis on Cloud Mobile BaaS Market

- Perspective", *Electronic and Telecommunications*, Vol. 30, No. 4, pp. 191-202, Aug. 2015.
- [13] Nader Dabit, "Full-Stack Cloud Development with AWS Amplify", *CODE Magazine*, Vol. 22, No. 1, pp. 38-46, Jan. 2021.
- [14] Housseem Yahiaoui, "Firebase Cookbook", Birmingham, UK:Packt Publishing, 2017.
- [15] Bonnie Eisenman, "React native", *Insight*, 2016.
- [16] B.O. Lee, "Angular 2programming : with bootstrap", LBO, 2016.
- [17] Y.H. Kim, D.W. Moon, "Vue.JS to end with a cup of coffee: Building real-world applications with Vue.JS", *BJpublic*, 2019.
- [18] Mark Volkmann, "Svelt and Sapper in Action", Hanbit Media Inc, 2021.

Authors



Koono Kim received the B.S., M.S. and Ph.D. candidate degrees in Computer Science and Engineering from Hannam University and Keimyung university, Korea, in 1993, 1995 and 1999, respectively.

Prof. Kim joined the faculty of the Global Leadership School at Handong University, Pohang, Korea, in 2000. He is currently a associate professor in the same university. He is interested in common knowledge, Metalogic programming, Multi-agent system.