

종합설계프로젝트 최종보고서

2016년 2학기

제출일: 2016.12.06

| | |
|---------|---|
| 과제명 | 홈페이지 재능 기부 프로젝트 - 모두의 할인 |
| 프로젝트 주제 | 웹 사이트를 이용한 단체 할인 티켓 구매 시스템 및 이를 이용한 단체 할인 티켓 구매 방법 (Purchasing System of Group's Discount Ticket by Using Web-Site and Method thereof) |
| 프로젝트 기간 | 2016.09.01. ~ 2016.12.23 |
| 지도교수 | 송동호 교수님 |

| | | | | | | |
|----|--------------------------|-----|----|------------|-----|---------------|
| 팀원 | 이름 | 이민석 | 학번 | 2011122190 | 연락처 | 010-5694-2282 |
| | 기획 및 디자인, 프론트엔드 및 백엔드 개발 | | | | | |

| | |
|-----------|--|
| 구분 | 내용 |
| 개발목적 및 동기 | 지인의 특허(웹 사이트를 이용한 단체 할인 티켓 구매 시스템 및 이를 이용한 단체 할인 티켓 구매 방법) 출원으로 인한 아이디어 구체화 및 웹 어플리케이션 프로토타입 제작 |
| 개발환경 | Mac OS, Sublime Text2, MEAN STACK(MongoDB, ExpressJS, AngularJS, NodeJS), HTML5, CSS3, Javascripts, JQuery, NPM, Bower, Yo Framework |
| 우수성/창의성 | 일반적으로 고궁, 놀이공원, 철도와 같은 기관에서는 단체 30 명이 일시에 이용하는 경우 단체 할인을 적용하고 있으나 일반적으로 구성원 수가 부족하여 단체 할인을 받는 경우가 극히 드문 것이 현실이다. 따라서 소수가 회원으로 가입하여 단체를 형성함으로써 티켓 구매 시에 단체 할인을 받도록 하는 것 |

차 례

| | | |
|--------|-----------------|----|
| I. 서 | 론 | 1 |
| 1. | 연구목표 | 2 |
| 2. | 연구필요성 | 2 |
| 3. | 연구범위 | 2 |
| 4. | 연구결과 | 2 |
| II. 방 | 법 | 3 |
| 1. | 당초 연구 계획 및 수정사항 | 3 |
| 2. | 연구 내용 | 5 |
| III. 결 | 과 | 6 |
| IV. 고 | 찰 | 14 |
| 1. | 연구 달성도 분석 | 14 |
| 2. | 기술 수준 고찰 | 15 |
| 3. | 우수성 및 독창성 분석 | 20 |
| 4. | 기대효과 | 20 |
| VI. 결 | 론 | 21 |
| 1. | 결 론 | 21 |
| 2. | 향후 연구 계획 | 21 |
| VII. 참 | 고 문 헌 | 22 |

I. 서 론

1. 연구 목표

- MEAN(MongoDB, ExpressJS, AngularJS, NodeJS) 스택을 활용하여 웹 사이트를 이용한 단체 할인 티켓 구매 시스템 및 이를 이용한 단체 할인 티켓 구매를 구현

2. 연구 필요성

- 일반적으로 고궁, 철도와 같은 기관에서는 단체 30 명이 일시에 이용하는 경우 단체 할인을 적용하고 있으나 일반적인 경우는 구성원 수가 부족하여 단체 할인을 받는 경우가 극히 드문 것이 현실이다. 따라서 소수가 회원으로 가입하여 단체를 형성함으로써 티켓 구매 시에 단체 할인을 받도록 하는 것이다.

3. 연구 범위

- 철도, 여행, 박물관, 미술관과 같이 일정 규모 이상의 단체 관람자들에게 요금 할인을 적용받을 수 있는 장소에서 단체를 형성하여 구매함으로써 요금할인을 받도록 하는 단체 할인 티켓 구매 시스템 구현한다.

- 단체에 대한 할인율이 정해진 철도청, 미술관과 같은 티켓 판매 사이트를 통한 티켓 구매 시 단체를 모집하여 단체를 기초로 티켓팅을 대리하고 구매한 티켓을 각 사람에게 전송하도록 함으로써 티켓 구매 시의 할인 혜택을 시스템 관리자 및 단체에 속한 각 회원과 공유한다.

4. 연구 결과

- 웹 사이트를 이용한 단체 할인 티켓 구매 시스템은 티켓 정보(이용시간 정보, 장소 정보, 좌석 정보, 요금 정보, 단체 할인요금 정보)를 가지고 있는 유저가 사람을 모으기 위해 글을 등록한다. 글이 등록되면 정보를 가지고 있는 유저뿐만 아니라 특정한 정보를 찾고 있는 유저들에게도 글이 공유된다. 이때 단체 할인에 참여할 의사가 있는 유저들은 글을 등록한 유저에게 직접적인 연락(전화 번호 또는 이메일) 또는 간접적인 연락(댓글)을 통하여 참여 의사를 밝힌다. 글을 등록한 사람의 주도적인 결정에 의해 단체 할인을 받기 위한 유저를 모을 수 있다. 이때 처음에 기획한 인원 모집이 완료되었을 경우에 추가적인 모집은 이루어지지 않는다. 또한 기획한 날짜가 지났을 경우에도 추가적인 모집은 이루어지지 않는다.

- 단체 할인을 분류하기 위해 크게 지역별, 분야별로 분류하였다. 먼저 지역별로 서울, 대구, 부산, 광주, 제주, 대전으로 분류하였다. 그리고 분야별로 문화(공연, 전시, 체험), 레저 및 스포츠(스포츠 경기, 헬스, 익스트림), 여행(투어, 항공, 숙박), 의료(진료, 상담, 시술)로 분류하였다.

II. 방 법

1. 당초 연구 계획

- 당초 연구 계획 및 수정사항

1) LAMP(Linux, Apache, Mysql, Php) 스택을 활용하여 단체 할인 티켓 구매 시스템 및 이를 이용한 단체 할인 티켓 구매를 구현 하고자 하였다. 하지만 프론트엔드 및 백엔드를 혼자서 풀스택으로 개발해야 하는 상황으로 인해 기존 웹 프로젝트 경험을 최대한 활용하기 위해 MEAN(MongoDB, ExpressJS, AngularJS, NodeJS) 스택으로 수정하여 개발을 진행하였다.

2) Sign up, Sign In 페이지, 메인 페이지(상품 소개 및 단체 예약), 싱글 페이지(상품 자세한 소개 및 현재 예약된 인원), 마이 페이지, 관리자 페이지, 결제 모듈, 반응형 버전과 같은 순서로 계획은 진행할 예정이었지만 HTML 및 CSS를 진행할 때 바로 반응형으로 퍼블리싱을 하면서 진행하였다.

| 연구 내용 | 기간 (주) | | | | | |
|------------------|--------|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 |
| Sign up, Sign In | | | | | | |
| 메인 페이지 | | | | | | |
| 싱글 페이지 | | | | | | |
| 마이 페이지 | | | | | | |
| Admin 페이지 | | | | | | |
| 결제 모듈 | | | | | | |

3) 결제 모듈(다날 및 카카오페이)을 연동하여 웹 사이트에서 단체 할인 티켓을 일정 수수료를 받고 결제까지 진행하도록 구현 하고자 하였다. 하지만 일정한 인원을 모아서 티켓을 구매하는 것뿐만 아니라 일정 시간에 동시에 입장해야하는 제약 조건 때문에 운영에 추가적인 인원이 필요하다는 판단 하에 결제를 연동하지 않기로 했다. 특히나 결제에 따른 환불 조건이 엄격해질 수밖에 없다는 판단을 하였다. 왜냐하면 한명의 단순한 변심으로 인해 환불을 받게 된다면 나머지 인원 모두 단체할인을 받을 수 없는 상황에 직면하기 때문이다. 따라서 결제 모듈을 제거하고 게시판 형식의 자유로운 형식으로 진행하기로 하였다. 하지만 단체 할인 정보를 제공하면서 글을 등록하는 유저는 글을 등록하는 것뿐만 아니라 유저들을 관리해야 한다는 부담이 작용하는 부작용이 있다. 이는 단체 할인을 받기 위해 글을 등록하는 유저에게는 할인된 금액에서 자신의 금액을 일정 감면해주는 식으로 진행할 예정이다.

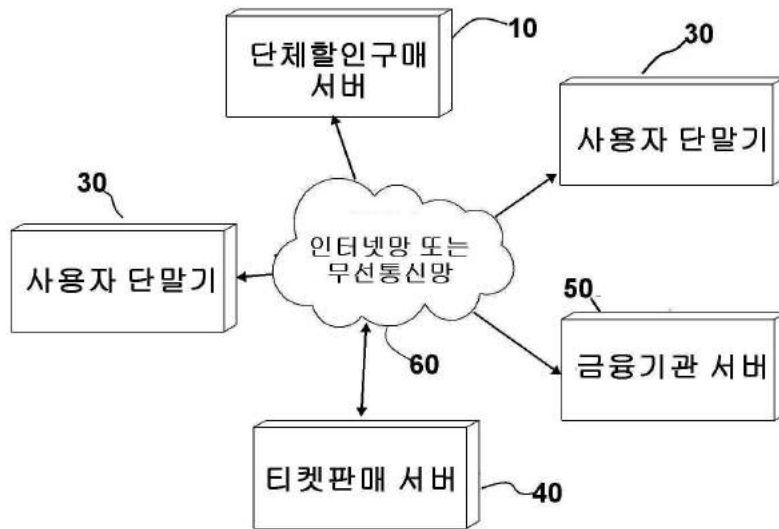


그림 1 결제모듈을 활용한 기존 연구 계획

4) 기존의 연구 계획에서 특정한 날짜, 시간, 인원만 있으면 단체할인을 받는 경우와 달리 KTX, 영화관과 같이 실시간으로 현재의 자석을 알 수 있는 데이터베이스의 정보가 필요한 경우가 존재할 때 이는 개발에서 제외하기로 하였다.

- 당초 연구원 편성

연구책임자

| 이민석 |
|---------------------------------------|
| 기획(와이어 프레임, DB), 백엔드 개발, AWS 관리 |

연구원

| 이수복 |
|----------------------------|
| 프론트엔드 개발(웹 페이지 퍼블리싱) |

2. 연구 내용(MEAN Stack)

- NodeJS

1) 노드는 구글의 V8 자바스크립트 엔진과 조이엔트의 libuv 라이브러리를 기반으로 만들어졌다. 노드는 자바스크립트를 서버 쪽 프로그래밍에서도 쓸 수 있는 대안으로 만들었다. 노드의 출현으로 인해 자바스크립트 개발자도 파일시스템에 접근하고, 네트워크 소켓을 열고, 자식 프로세스를 만들 수 있게 되었다.

2) 노드가 서버 쪽 자바스크립트 구현을 위한 첫 번째 시도는 아니지만, 현재까지 가장 성공한 제품이다. 노드는 월마트, 페이스북, 링크드인, 마이크로소프트와 같은 대기업에서 채택해왔다.

3) 노드는 이벤트 기반 비동기 방식으로 작동하므로 대규모 네트워크 프로그램 개발에 적합하다. 하지만 프로그램 진행에 문제가 발생하는 순간 프로그램 전체가 영향을 받는다.

4) <http://nodejs.org>

- MongoDB

1) 데이터 저장소를 다루는 주요 수단인 관계형 데이터베이스와 달리 SQL에 의존하지 않는 또 다른 데이터 저장소인 NoSQL로서 테이블 구조를 사용하지 않는다. NoSQL 데이터베이스는 키-값 쌍이나 문서와 같은 다양한 형식으로 데이터를 저장하며, 관계형 데이터베이스보다 덜 엄격하고 덜 구조적인 형태를 지원한다. 즉, 구조적인 제약이 덜하다는 특징은 단순한 프로토타입 제작이나 개발 편의성 측면에서 더 유리하다.

2) 문서 중심 데이터베이스로서, 정보를 BSON(Binary JSON) 문서 형태로 저장한다. JSON 계열의 포맷을 사용함으로써 몽고디비는 자바스크립트 코드로 손쉽게 객체를 읽고 쓸 수 있다.

- AngularJS

1) 기술의 발전과 함께 자바스크립트 역시 진화를 거듭했다. 웹 앱을 위한 주요 돌파구는 광범위한 브라우저 지원과 고속 인터넷 보급과 함께 찾아왔다. 이는 전체 페이지를 읽는 대신 배경 작업으로 요청을 처리하는 Ajax 앱을 위한 문을 열었다. Ajax 앱에서는 네트워크 성능이 핵심이었다. 연결이 느리면 페이지가 반응하지 않는 것처럼 보이게 만들기 때문이다. 앱은 점진적으로 페이지 읽기를 줄이고 Ajax 요청을 늘리는 쪽으로 이동했다. 궁극적으로 단일 페이지 앱(Single Page Application)이 탄생했다. 즉, SPA는 딱 한 페이지만 읽고, 나머지 모든 데이터는 Ajax 호출로 요청한다.

2) Angular는 웹 앱에 모델-뷰-컨트롤러(Model-View-Controller)를 적용했고, 주목할 만한 여러 기능을 제공한다. 먼저 뷰와 모델 사이에서 양방향 데이터 바인딩을 제공한다. 이는 개발 시간을 단축할 수 있게 해준다. 앵글러가 자동으로 모든 것을 동기화 한다.

3) Angular는 템플릿을 포함한 여러 작업을 강화된 HTML로 수행할 수 있다.

III. 결 과

- 완성된 웹 페이지 소개

1) 회원 가입 및 로그인 페이지

메인 페이지를 제외한 모든 페이지는 회원 가입과 로그인이 이루어지고 난 뒤에 사용할 수 있도록 구현하였다.



그림 2 회원 가입 페이지

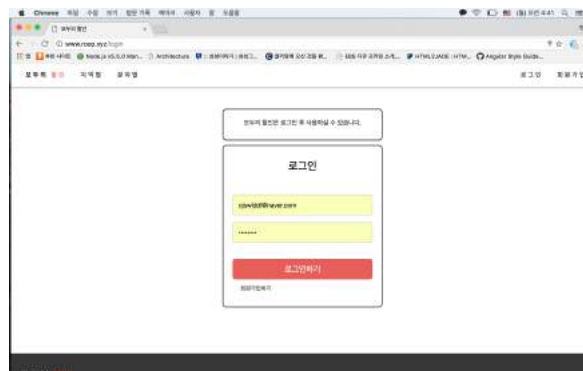


그림 3 로그인 페이지

2) 메인 페이지

메인 페이지에서는 모두의 할인 서비스 슬로건 및 지역별(서울, 대구, 부산, 대전, 제주, 광주) 분류 및 분야별(문화, 레저, 여행, 의료) 분류로 나누어서 유저가 원하는 정보를 쉽게 찾을 수 있도록 분류하였다. 네비게이션 바는 스크롤을 내리더라도 상단에 유지하도록 개발하였다. 그리고 각각의 분류를 마우스오버할 때 마다 이미지의 색깔이 변하거나 밑줄이 생기도록 이벤트를 구현하였다.



그림 4 메인페이지1(슬로건)

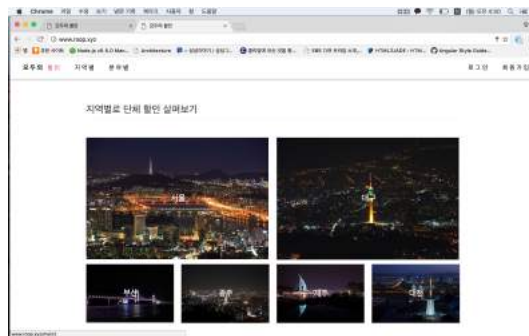


그림 5 메인페이지2(지역별 분류)

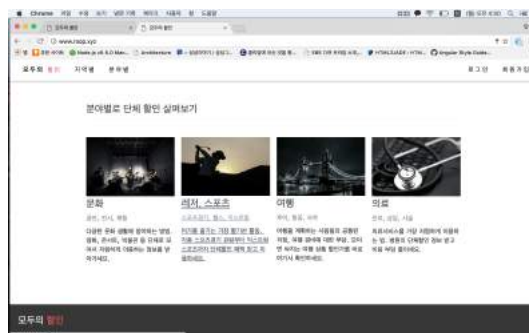


그림 6 메인페이지3(분야별 분류)

3) 지역별 페이지 및 분야별 페이지

지역별 페이지는 단체 할인을 위한 모임을 지역별로 분류한 페이지이다. 네비게이션 바에서 지역별 부분을 클릭하게 되면 전체 모임 정보를 나타낸다. 이때 서울, 대구, 부산, 제주, 광주, 대전 버튼을 누르게 되면 각 지역에 관한 정보만 볼 수 있도록 구현하였다. 특히나 글이 등록되어 있는 정보는 단순한 바둑판식 배열이 아닌 핀터레스트 형식의 배열로 구현하였다. 왜냐하면 각각의 유저들은 크기가 다른 사진을 가지고 있을텐데 바둑판식 배열로 각각의 사진을 표현하게 된다면 어떤 이미지는 가로가 늘어서 보이게 될 것이고 반대로 어떤 이미지는 가로가 작게 보이게 될 것이다. 따라서 이미지의 크기에 따라 글의 배열이 결정되도록 구현하였다.

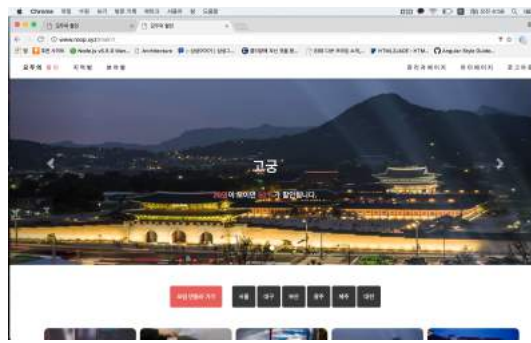


그림 7 지역별 페이지1
(광고 슬라이더 및 지역별 분류)

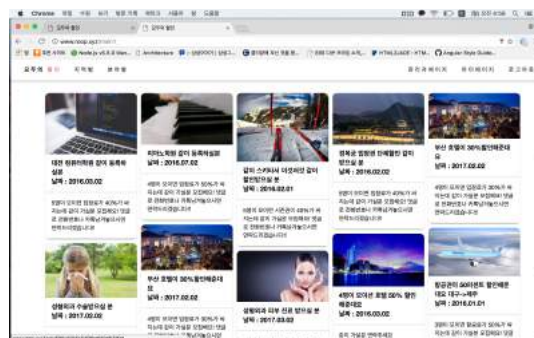


그림 8 지역별 페이지2(핀터레스트 배열)

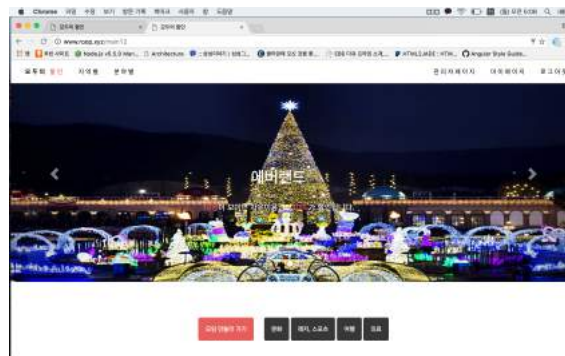


그림 9 분야별 페이지
(광고 슬라이더 및 분야별 분류)

4) 단체 할인 모임 등록 모달창

타이틀, 자세한 설명(모집 인원, 현재 인원, 할인율, 할인된 금액), 모임 날짜, 지역, 분야, 이미지를 선택하여 모임을 개설할 수 있는 모달창이다. 원하는 정보가 모두 채워졌을 때 글이 등록되도록 구현하였다.

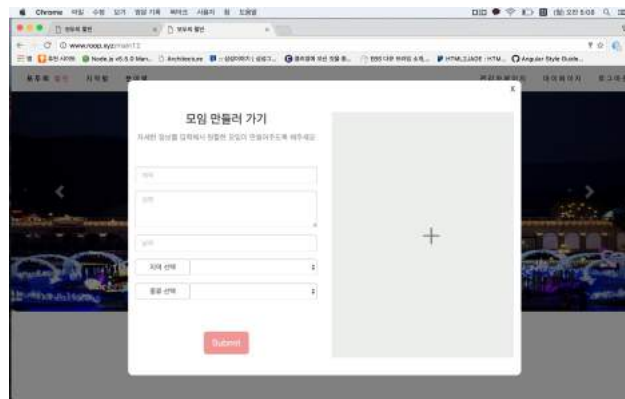


그림 10 단체 할인 모임 등록 모달창

5) 싱글페이지(단체 할인 모임)

핀터레스트 배열로 간략한 내용을 나타냈다면 싱글페이지는 자세한 내용을 알 수 있다. 그리고 댓글로 직접적인 참여가 이루어질 수 있도록 구현하였다. 특히나 SNS 성격을 주기 위해 좋아요 버튼, 현재까지 본 인원, 다양한 단체 할인 소개와 같이 추가적인 기능도 구현하였다.



그림 11 싱글페이지
(자세한 내용, 다양한 단체 할인 소개, 좋아요 버튼 등)

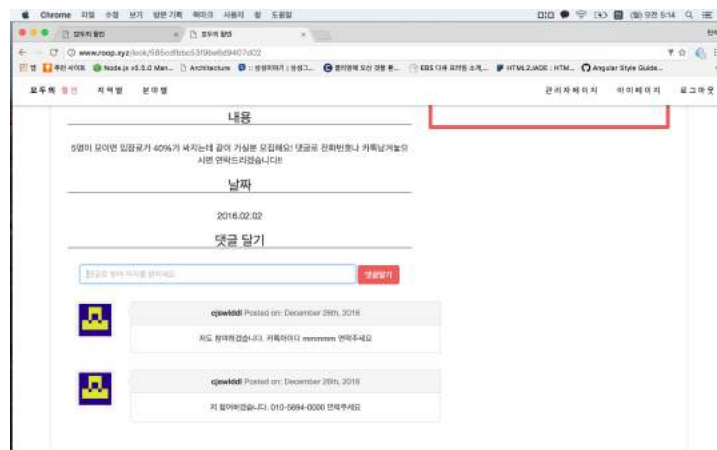


그림 12 댓글 달기
(전화 번호, 카톡 아이디 등을 남길 수 있다)

6) 마이 페이지
내가 등록한 모임을 볼 수 있다. 특히나 글 수정, 삭제, 좋아요 기능이 가능하다.



그림 13 마이 페이지(내가 등록한 단체 할인
모임)

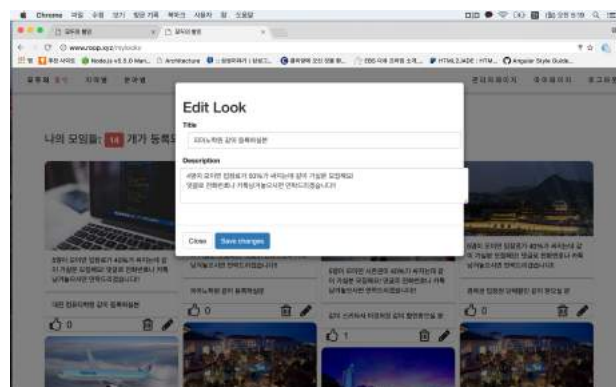


그림 14 마이 페이지(등록한 모임 수정 및 삭제)

7) 관리자 페이지

관리자 권한을 가진 계정에서는 네비게이션 바에 관리자 페이지 버튼이 보이게 된다. 관리자 페이지에서 모든 유저 및 모임의 정보를 볼 수 있다. 유저 및 모임의 정보를 삭제, 수정이 가능하도록 구현하였다.

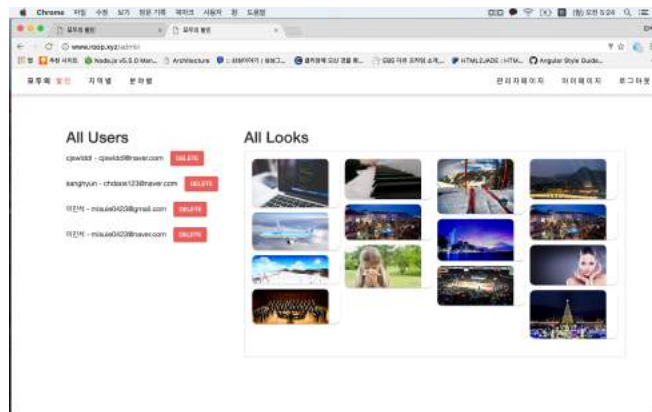


그림 15 관리자 페이지(모든 유저 및 모든 모임)

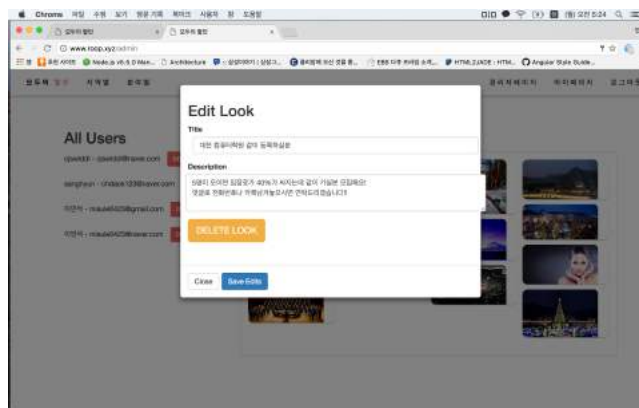


그림 16 관리자 페이지(모임 글 수정 및 삭제)

- 반응형 웹 페이지

웹 브라우저의 크기에 따라 이미지의 크기 및 배치가 반응형이 되도록 구현하였다. 메인 페이지에서의 이미지, 유저가 등록한 글의 이미지, 마이 페이지, 관리자 페이지 모든 페이지에서 반응형이 되도록 구현하였다.



그림 17 지역별, 분야별 소개 페이지 반응형
(웹 브라우저를 반으로 줄였을 때)



그림 18 지역별, 분야별 소개 페이지 반응형
(웹 브라우저는 최소로 줄였을 때)

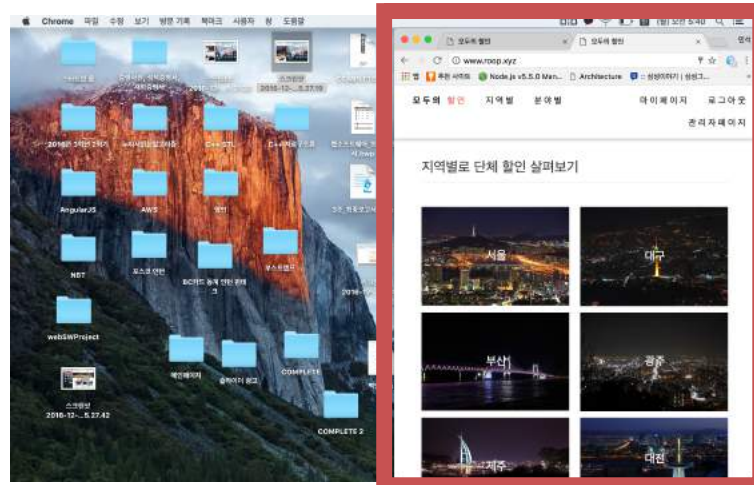


그림 19 메인 페이지 반응형
(웹 브라우저를 반으로 줄였을 때)

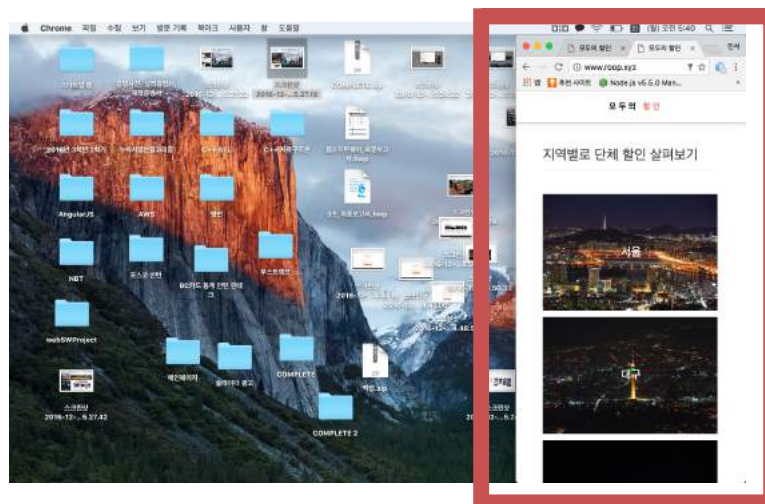


그림 20 메인 페이지 반응형
(웹 브라우저를 최소로 줄였을 때)

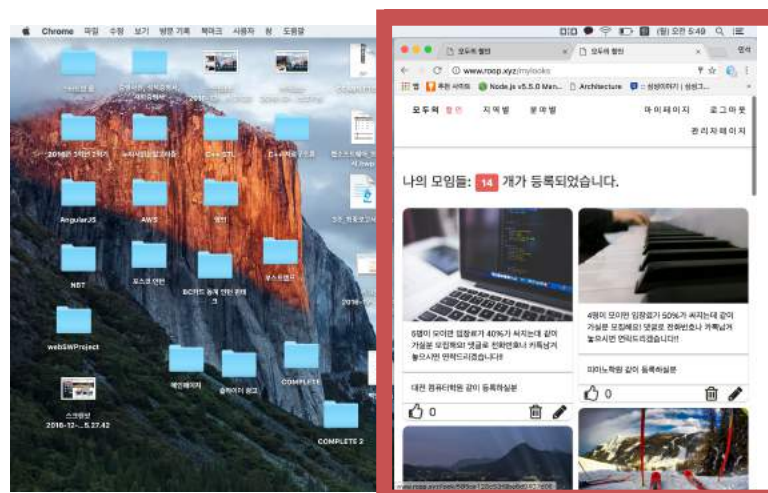


그림 21 마이 페이지 반응형
(웹 브라우저 반으로 줄였을 때)

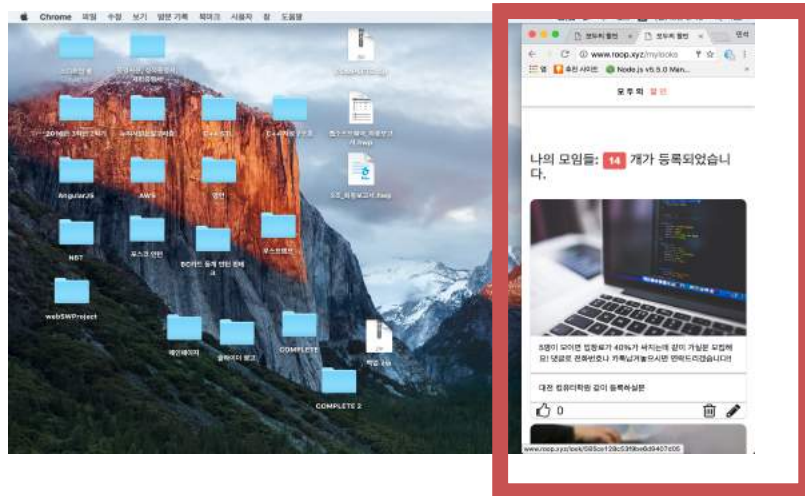


그림 22 마이 페이지 반응형
(웹 브라우저 최소로 줄였을 때)

IV. 고 찰

1. 연구 달성도 분석

- 기존에 생각한 기능 signup, login, main, single, mypage, admin 등 기본적인 페이지에 분야별, 지역별 분리를 통한 사용자가 더 편리하게 사용할 수 있도록 구현하였다.

- 차후 광고 배너를 활용하기 위해 메인 슬라이더를 지역별, 분야별로 분리해서 각각의 지역과 분야에 맞게 광고를 할 수 있는 부분을 추가하였다.

- 바둑판식 배열이 아닌 이미지 크기에 맞게 게시물 크기가 정해지는 핀터레스트 식 배열로 통해 조금 더 SNS처럼 보이도록 구현하였다. 이를 통해 가로가 긴 이미지가 세로로 늘려지는 불상사 또는 세로가 긴 이미지가 가로로 늘려지는 불상사를 막을 수 있다. 이는 반응형으로 가능하도록 구현하였다.

- 하지만 추가적인 운영 인원 없이 단체 할인을 개설하는 유저에게 글을 등록하고 유저를 모으고 따로따로 연락을 해야 하는 부담감이 크다.

- 특히나 단체 할인을 개설하는 유저 또는 단체 할인에 참여하는 유저의 무책임한 태도로 인해 함께 참여하는 모든 사람이 피해를 볼 수 있다.

2. 기술 수준 고찰

1) 세션

1-1) 회원 가입시 아래 소스코드와 같이 특정 시간 동안 세션을 유지하도록

록 개발하였다. 특히나 이러한 세션을 활용하여 전체적으로 클라이언트가 서비스를 이용할 때 편리하도록 개발을 진행하였다.

```
exports.create = function(req, res, next) {

    var getGravatar = gravatar.url(req.body.email, {
        s: 40,
        d: 'retro'
    });

    var newUser = new User(req.body);
    newUser.gravatar = getGravatar;
    newUser.provider = 'local';
    newUser.role = 'user';
    newUser.save(function(err, user) {
        if (err) return validationError(res, err);
        var token = jwt.sign({
            _id: user._id
        }, config.secrets.session, {
            expiresInMinutes: 60 * 5
        });
        res.json({
            token: token
        });
    });
};
```

2) 보안

2-1)회원 가입 또는 로그인시 이메일은 regex를 통해 이메일 형태만 입력받을 수 있으며 이메일 형태가 입력받지 못했을 경우 회원가입과 로그인을 시도하지 않는다. 아래 스크린샷과 같이 이메일을 입력하는 곳에 abc를 입력하였을 때 Doesn't look like a valid email이라고 출력되고 회원가입이 진행되지 않는다.

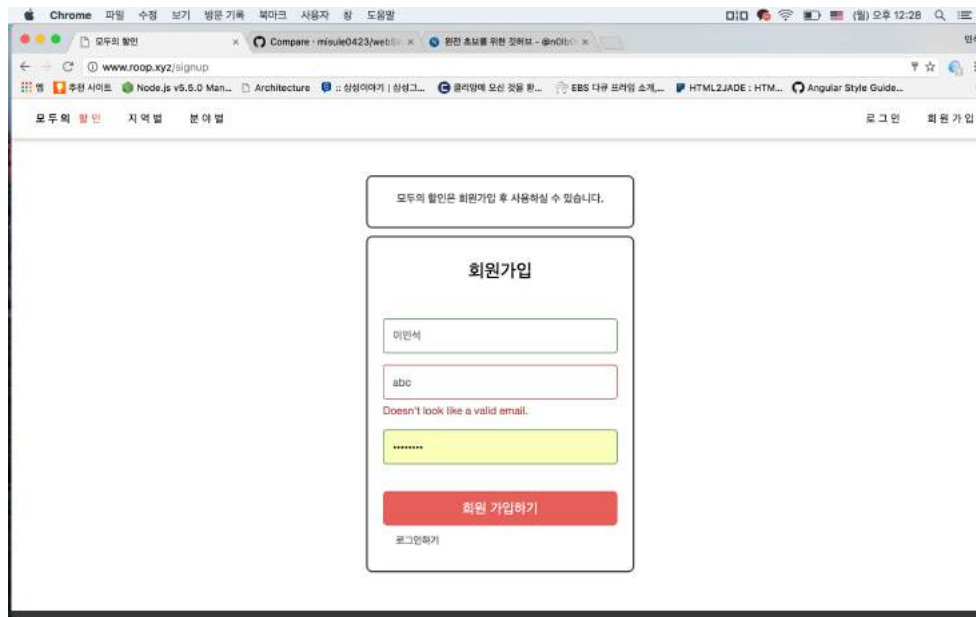


그림 23 이메일을 입력하는 곳에 abc를 입력하였을 때

2-2)아래와 같이 소스 코드로 회원가입시 비밀번호는 encryption 되어서 데이터베이스에 저장한다.

```
makeSalt: function() {
    return crypto.randomBytes(16).toString('base64');
}

encryptPassword: function(password) {
    if (!password || !this.salt) return "";
    var salt = new Buffer(this.salt, 'base64');
    return crypto.pbkdf2Sync(password, salt, 10000,
64).toString('base64');
}
```

```

[> db.users.find().pretty() * I CONTROL [initandlisten] db version v3.2.6
{
  "16-12-26T12:23:42.060+0900 I CONTROL [initandlisten] git version: 05552b562c7a0b3143a729aaa0838e558dc4
  9b25",
  "_id" : ObjectId("58538b35fcb48281337b086"),
  "provider" : "local",
  "gravatar" : "http://www.gravatar.com/avatar/b57dfc5a72a57723d67f4cfdd4a6fea9?s=40&d=retro",
  "name" : "이민석",
  "email" : "misule0423@gmail.com",
  "hashedPassword" : "dk55wBiBkkvHVXFA8dRlg70e936p76UUtWSHT2voJYSPXlidHrMctXhjquFwon7ZUSrsX+5yG5Ui
  1kLgRXz5Q==",
  "salt" : "+eGdMkr2NwMS/DnLmIdpWQ==",
  "rssUrls" : [],
  "role" : "admin",
  "likes" : [],
  "looks" : [],
  "v" : 0,
  "16-12-26T12:23:42.062+0900 I STORAGE [initandlisten] Recovering data from the last clean checkpoint.
  {
    "16-12-26T12:23:42.062+0900 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=4G,session
    {
      max=20000,eviction=(threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true
      ,path=),
      "16-12-26T12:23:42.062+0900 I STORAGE [initandlisten] ** WARNING: soft limits too low. Number of files
      is 256,
      "16-12-26T12:23:42.062+0900 I STORAGE [initandlisten] Initializing full-time diagnostic data capture with
      directory "/data/diagnostic.data",
      "16-12-26T12:23:42.062+0900 I NETWORK [initandlisten] waiting for connections on port 27017
      "16-12-26T12:23:42.062+0900 I FTDC [Ftdc] Unclean full-time diagnostic data capture shutdown detecte
      d, found
      "16-12-26T12:23:42.062+0900 I NETWORK [initandlisten] connection accepted from 127.0.0.1:63101 #1 (1 co
      nnection now open)
    }
  }
}
>

```

그림 24 encryption한 패스워드

3) 데이터베이스 스키마

3-1) 간략한 User 스키마를 통해 불필요하게 데이터베이스가 확대되는 것을 방지하였다.

```

var UserSchema = new Schema({
  name: String,
  gravatar: String,
  looks: [{
    type: Schema.ObjectId,
    ref: 'Look'
  }],
  likes: [{
    type: Schema.ObjectId,
    ref: 'Look'
  }],
  details: {
    website: String,
    bio: String,
    location: String,
    twitter: String,
    facebook: String,
    instagram: String,
    other: String
  }
});

```

```

    },
    email: {
        type: String,
        lowercase: true
    },
    role: {
        type: String,
        default: 'user'
    },
    hashedPassword: String,
    provider: String,
    salt: String,
    facebook: {},
    twitter: {},
    github: {},
    rssUrls: Array
});

```

3-2) 간략한 단체 할인 글 스키마를 통해 불필요한 데이터가 확대되는 것을 방지하였다.

```

var LookSchema = new Schema({
    image: String,
    linkURL: String,
    title: String,
    description: String,
    date: String,
    location: String,
    category: String,

    _creator: {
        type: Schema.ObjectId,
        ref: 'User'
    },
    email: String,
    userName: String,
    userGravatar: String,
    createTime: {
        type: Date,
        'default': Date.now
    },
    lookStatus: {

```

```

        type: Number,
        'default': 0
    },
    views: {
        type: Number,
        'default': 0
    },
    upVotes: {
        type: Number,
        'default': 0
    },
    ansBrandname: String,
    ansDescription: String,
    ansSize: String,
    ansSex: String,
    ansColor: String,
    ansPrice: Number,
    ansTag: String,
    ansLink: String
});

```

3-3) 간략한 댓글 스키마를 통해 불필요한 데이터가 확대되는 것을 방지하였다.

```

var CommentSchema = new Schema({
  author: {
    id: {
      type: Schema.ObjectId,
      ref: 'User'
    },
    name: {
      type: String
    },
    email: {
      type: String
    }
  },
  lookId: {
    type: Schema.ObjectId,
    ref: 'Look'
  },
  gravatar: {

```

```

        type: String
    },
    comment: {
        type: String,
        trim: true
    },
    createTime: {
        type: Date,
        'default': Date.now
    }
});

```

3. 우수성 및 독창성 분석

- 현재 단체 할인을 받기 위해 정보가 매우 제한적이다. 정보를 가지고 있는 사람은 소수이지만 막상 정보를 가지고 있더라도 인원을 모으는 것을 더 어렵다. 그래서 네이버 카페, 다음 카페와 같은 곳에서 유저들을 모으는 경우가 있다. 이러한 분리되어 있는 각각의 부분을 통합적으로 ‘모두의 할인’에서 대부분의 단체 할인 모임을 개설 및 유저들을 모을 수 있다.

- 단체에 대한 할인이 정해진 철도청, 미술관과 같은 티켓 판매 사이트를 통한 티켓 구매 시 단체를 모집하여 단체를 기초로 티켓팅을 대리하고 구매한 티켓을 각 사람에게 전송하도록 함으로써 티켓 구매 시의 할인 혜택을 시스템 관리자 및 단체에 속한 각 회원과 공유할 수 있다.

- 단순히 글을 쓰고 등록하고 댓글을 다는 게시판이 아니라 조금 더 크게 본다면 단순한 유저가 아니라 실제로 광고를 원하는 사람이 글을 등록하여 할인 혜택을 제공할 수 있을 수 있다. 이를 통해 모임을 개설하는 유저는 정보를 가지고 있는 유저뿐만 아니라 실제로 관련된 내용에 종사하는 유저일 수 있다.

4. 기대효과

- 웹 사이트를 이용한 단체 할인 티켓 구매 시스템 및 이를 이용한 단체 할인 티켓 구매 방법은 불특정 다수의 사람들이 단체를 구성하여 티켓 구매 시 단체 할인을 받을 수 있도록 하는 효과가 있는 것이다.

- 시스템 관리자와 티켓 구매자가 단체 할인 요금에 따른 이익금을 분배할 수 있는 효과가 있는 것이다.

- 공연 관람, 박물관, 미술관의 특별전 개최와 같은 문화 정보를 제공할 수 있다.

- 단체 할인 구매 서버의 웹 사이트를 통하여 철도청, 공연, 박물관 등의 홍보 등을 함께 제공할 수 있는 효과가 있는 것이다.

VI. 결 론

1. 결 론

한 학기 동안 웹 서비스를 기획 및 개발 그리고 보고서 제출까지 혼자서 진행하기에는 여러 어려움이 있었지만 그만큼 더더욱 애착이 가는 결과물로 만들고 싶었다. 처음에는 재능기부를 부탁한 사람의 개발 목적을 알기 위해 일주일에 두 번씩 모여서 그 사람이 생각하고 있는 생각을 구체화 시켰다. 단순히 아이디어 및 콘텐츠만 가지고 있는 상태에서 어떻게 웹 어플리케이션을 통해서 나타낼지 한 달 동안 구상하였다. 수많은 아이디어가 있었다. P2P 사이트와 같이 캐쉬를 충전하고 모임에 참여할 때 캐쉬가 빠지는 형식 또는 모임에 참여할 때 그제서야 결제가 이루어지는 형식 등 수많은 형식이 고려되었지만 앞으로의 확장성을 기준으로 결정을 내렸을 때는 게시판 형식이 가장 적합하다는 것으로 판단하였다. 그리고 처음 경험해 보는 UI 및 UX 부분에서 오랜 시간이 걸렸다. 예를 들어 디자인에서 내가 생각하는 부분이 처음 보는 다른 사람에게 잘 전달 될 수 있도록 노력하였다. 서비스의 설명, 간략한 슬로건, 전체적인 컬러, 포인트 컬러와 같이 유저를 기준으로 생각하였다. 그리고 유저 입장에서 불필요한 일을 더 수행할 일은 없는지 구상하였다. 기획 및 와이어프레임 제작이 끝나고 실질적으로 개발에 착수하였다. 개발을 진행할 때 풀 스택으로 개발하는 것이 정말 큰 부담감으로 작용하였다. 예를 들어 회원 가입 및 로그인을 하기 위해서 서버 코딩을 마치고 데이터베이스 스키마를 생각하는 것뿐만 아니라 반응형으로 웹 페이지를 퍼블리싱 그리고 이벤트 등록 등 수많은 일들이 끊임없이 있었다. 카페에서 끊임없이 밤을 새고 대구에 내려가서도 끊임없이 작업을 하였다. 하면할수록 부족한 부분이 계속해서 나타났다. 하나하나 점진적으로 개선하고자 노력하였다. 그리고 구현하였다. 이러한 풀 스택 경험이 학부 생활을 하는 동안 큰 프로젝트로 남을 수 있을 것 같다.

2. 향후 연구 계획

- 단체할인을 위해 글을 등록하는 유저를 조금 더 편리하게 할 수 있는 방안을 진행할 것이다. 가장 먼저 글 등록 및 관리를 참여 의사만 밝히면 카카오톡 단체 방을 만들어 추가적인 작업 없이 바로 연락이 가능하도록 만들 것이다. 이를 위해 카카오톡 로그인 및 회원가입을 구현할 것이다.

- 단체 할인 멤버가 확정 된 경우 결제 시스템 도입을 통해 단체할인 모임의 약속이 더욱 더 견고해 지도록 할 것이다. 단순한 변심으로 인해 끊임없이 단체 할인이 취소되는 경우 ‘모두의 할인’ 사이트는 신뢰도가 점점 떨어질 것이다. 따라서 결제라는 확실한 의사 결정 단계를 도입할 예정이다.

- 정보를 가지고 있는 단순한 유저뿐만 아니라 정보를 판매하고 있는 유저들을 통해 ‘모두의 할인’ 서비스가 활성화되길 기대하고 있다. 단순히 멤버를 모으는 것이 자신의 매출과 연관이 된다면 하나의 홍보 수단으로 이용할 수 있을 수 있다.

- ‘스택오버플로우’와 같이 유저들을 등급별로 나누어 글을 등록할 수 있는 단계, 댓글을 달 수 있는 단계를 세분화시켜 서비스의 신뢰도와 정확도를 높일 예정이다.

VII. 참 고 문 헌

