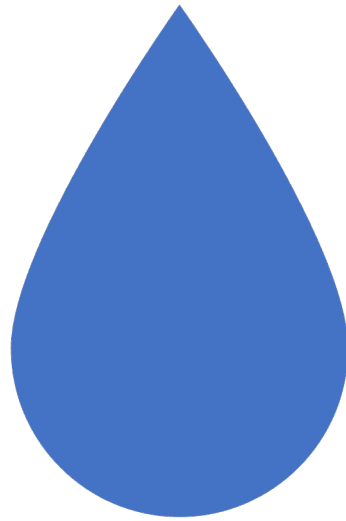


[모두의 일퀘]_20203033_김대성

웹 클라이언트 컴퓨팅



목차

1. 서비스 개요
2. 기획 의도
3. 구현하지 못한 기능
4. 추가 구현 기능
5. 구현화면
6. 느낀점
7. 참고문헌

1. 서비스 개요

- 매일 퀘스트를 수행하면 매일 환급받는 서비스
- "모두의 일퀘 서비스는 공통의 목표를 가진 유저들이 모여 참가비를 지불하고 목표를 완수하는 날마다 참가비의 총합을 남은목표일수와 사람수로 나누어 돌려주는 서비스입니다. 여러 사람이 다같이 참여하여 다른 참여자들을 보며 자극을 받을 수 있고, 실패시 금전적 손해를 볼 수 있다는 점에서 유저에게 동기부여를 해줍니다. 매일 목표를 성취하며 받는 보상으로 유저들에게 자기 성장을 경험할 수 있도록 도와줍니다.

2. 기획 의도

- 공부,운동 등 성장하기 위해서는 "매일" 꾸준한 노력을 해야합니다. 하지만 현실에서 이러한 것들은 즉각적인 큰 변화가 있지않아 노력을 지속하기가 어렵습니다. 그래서 저는 게임에서 몬스터를잡으면 바로 경험치가 오르는거나 아이템을 얻는것처럼 즉각적인 보상이 있으면 매일 하기 쉽겠다 생각하여 "모두의 일퀘" 서비스를 기획하게되었습니다.

3. 구현하지 못한 기능

- 할일 완수 기능
- 썸네일 업로드 기능
- 커뮤니티 기능

4. 추가 구현기능

- 캐시 충전 기능
- 챌린지 생성 기능
- 로그인,회원가입, 페이지네이션 기능
- 중도포기 기능

*. 필요한 라이브러리

*package.json이 있어
npm install 명령어로
전부 설치가능

express

npm install express

mongodb

npm install mongodb@5

method-override

npm install method-override

passport

npm install express-session passport passport-local

bcrypt

npm install bcrypt

5.구현화면

커버페이지



5. 구현화면

회원가입

```
app.post('/signup', async(request,response)=>{  
  let hash = await bcrypt.hash(request.body.password,10)  
  let compare = await db.collection('user').findOne({username : request.body.username})  
  if(compare) return response.send("<script>alert('이미 존재하는 아이디입니다. ');window.location.replace('/signup')</script>");  
  if(request.body.password != request.body.password_check) return response.send("<script>alert('비밀번호가 다릅니다. ');window.location.replace('/signup')</script>");  
  await db.collection('user').insertOne(  
    {  
      username : request.body.username,  
      password : hash,  
      cash : 0,  
      participating_in_challenge : []  
    }  
  )  
  response.redirect('/main')  
})
```

회원가입

사용자 이름

비밀번호

비밀번호 확인

회원가입

```
_id: ObjectId('6645ab874faa5136bf404a14')  
username : "시험용"  
password : "$2b$10$j5.LK9Vh6oS9tIOHl/U8f.lf  
cash : 0  
participating_in_challenge : Array (empty)
```

로그인

로그인

사용자 이름

비밀번호

로그인

회원가입

```
// 제출한 아이디/ 비밀번호 검사
passport.use(new LocalStrategy(async (입력한아이디, 입력한비번, cb) => {
  let result = await db.collection('user').findOne({ username : 입력한아이디})
  console.log(result)
  if (!result) {
    return cb(null, false, { message: '아이디 DB에 없음' })
  }
  if (await bcrypt.compare(입력한비번.toString(), result.password)) {
    return cb(null, result)
  } else {
    return cb(null, false, { message: '비번불일치' });
  }
}))

// 로그인시 세션 만들기, 쿠키를 유저에게 보내주기
passport.serializeUser((user, done) => {
  process.nextTick(() => { // <-- 내부코드를 비동기적으로 처리해줌
    done(null, { id : user._id , username : user.username })
  })
})

// 유저가 보낸 쿠키분석
passport.deserializeUser(async(user, done) => {
  let result = await db.collection('user').findOne({_id : new ObjectId(user.id)})
  delete result.password
  process.nextTick(() => {
    done(null, result)
  })
})

// 이제 request.user 하면 현재 로그인한 유저 정보 불러오기 가능
```

```
app.post('/login', async(request, response, next) =>{
  passport.authenticate('local', (error, user, info) =>{
    if(error) return response.status(500).json(error)
    if(!user) return response.status(401).json(info.message)
    request.login(user, (err) =>{
      if(err) return next(err)
      response.redirect('/main')
    })
  })(request, response, next)
})
```

메인페이지

DB에서 챌린지, 유저정보를 가져와 main.ejs파일에 전달

```
<section class="challenge-list">
  <h1 style="text-align: center;">챌린지 리스트</h1>

  <% for(let i = 0; i < 챌린지.length; i++) { %>
    <div class="challenge-card">
      <h2><%= 챌린지[i].title %></h2>
      <p>참여인원 : <%= 챌린지[i].participants %>명</p>
      <p>참가비 : <%= 챌린지[i].fee %>원</p>
      <p>챌린지 종료 : <%= 챌린지[i].deadline %></p>
      <button onClick="location.href='/challenge_detail/<%= 챌린지[i]._id %>'>참여하기</button>
    </div>
  <% } %>
```

```
// 메인페이지 접속
app.get(['/main', LoginCheck, async(request, response) => {
  let result = await db.collection('card').find().skip(10*(request.params.num-1)).limit(10*request.params.num).toArray()
  let user = await db.collection('user').findOne({username : request.user.username})
  response.render('main.ejs', {챌린지 : result , 유저 : user})
})
```

DB에서 받은 챌린지들 나열

메인페이지



홈페이지 인기 챌린지 커뮤니티 로그아웃 My Page My Cash : 25000 Cash+ 로그인된 아이디 : hello

챌린지 리스트

3주 미라클 모닝

참여인원 : 3명

참가비 : 30000원

챌린지 종료 : Thu Jun 06 2024 14:53:37 GMT+0900 (대한민국 표준시)

참여하기

2주동안 6시 기상

참여인원 : 2명

참가비 : 30000원

챌린지 종료 : Thu May 30 2024 14:55:55 GMT+0900 (대한민국 표준시)

참여하기

백준 하루 한문제(30일)

참여인원 : 1명

참가비 : 30000원

챌린지 종료 : Sat Jun 15 2024 14:56:47 GMT+0900 (대한민국 표준시)

참여하기

매일 공부인증(50일)

참여인원 : 1명

참가비 : 50000원

챌린지 종료 : Fri Jul 05 2024 14:57:23 GMT+0900 (대한민국 표준시)

참여하기

매일 원편맨 운동(100일)

참여인원 : 1명

참가비 : 100000원

챌린지 종료 : Sat Aug 24 2024 14:57:53 GMT+0900 (대한민국 표준시)

참여하기

1일 1영화감상

참여인원 : 1명

참가비 : 50000원

챌린지 종료 : Wed Jun 05 2024 14:58:20 GMT+0900 (대한민국 표준시)

참여하기

매일 청소하기

참여인원 : 1명

참가비 : 30000원

챌린지 종료 : Mon Jun 10 2024 14:58:36 GMT+0900 (대한민국 표준시)

참여하기

매일 직접 음식 만들기

참여인원 : 1명

참가비 : 30000원

챌린지 종료 : Sun May 26 2024 14:58:49 GMT+0900 (대한민국 표준시)

참여하기

강아지 산책시키기

참여인원 : 1명

참가비 : 50000원

챌린지 종료 : Sat Jun 01 2024 14:59:09 GMT+0900 (대한민국 표준시)

참여하기

1일 1애니 감상

참여인원 : 1명

참가비 : 50000원

챌린지 종료 : Fri May 31 2024 14:59:21 GMT+0900 (대한민국 표준시)

참여하기

노래연습

참여인원 : 1명

참가비 : 50000원

챌린지 종료 : Wed Jun 12 2024 14:59:50 GMT+0900 (대한민국 표준시)

참여하기

매일 등교하기

참여인원 : 1명

참가비 : 50000원

챌린지 종료 : Sat Jun 15 2024 15:01:14 GMT+0900 (대한민국 표준시)

참여하기

<- 1 2 3 4 5 ->

+

© 2024 모두의 일케. All rights reserved.

네비게이션 바



홈페이지	Main 페이지로 이동
인기챌린지	참여인원이 많은 상위 5개의 챌린지를 보여줌
커뮤니티,MyPage	미구현
MyCash	로그인한 유저의 Cash 잔액을 보여줌
Cash+	Cash충전 결제창으로 이동

챌린지 생성

챌린지 생성

챌린지 이름:

최소 참가비:

3만원

▼

목표 일수:

예: 30

챌린지 생성

최소 참가비:

✓ 3만원

5만원

10만원

캐시충전

참가비 결제

이름:

이메일:

카드 번호:

유효 기간:

CVV:

지불 금액:

결제하기

유저 DB업데이트

```
// 캐시충전
router.post('/pay', async(request, response) => {
  let payment = parseInt(request.body.payment);
  await db.collection('user').updateOne(
    {username : request.user.username},
    {$inc : {cash : payment}});

  response.redirect('/main')
})
```

챌린지 생성

백준 하루 한문제(30일)

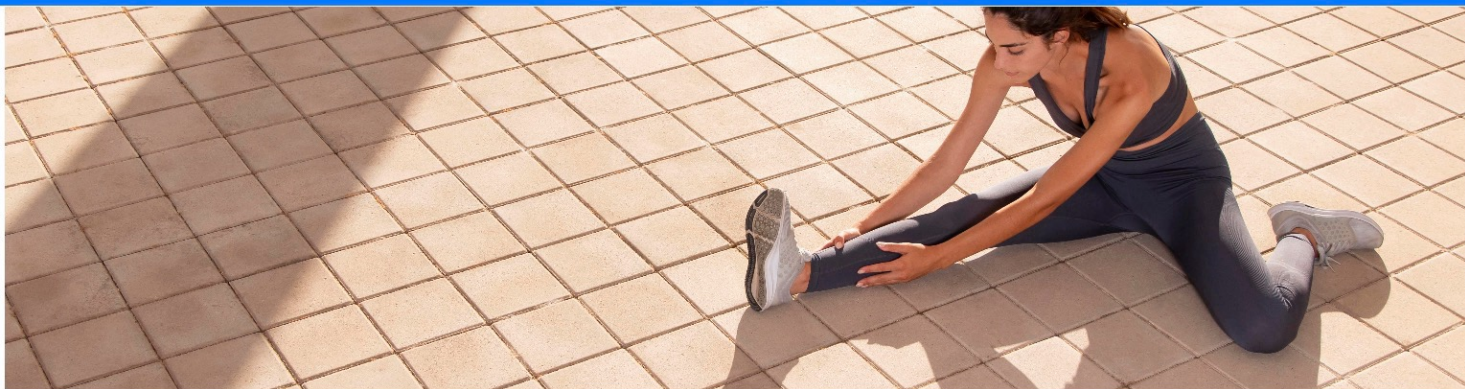
참여인원 : 2명

참가비 : 30000원

챌린지 종료 : Sat Jun 15 2024
14:56:47 GMT+0900 (대한민국 표준시)

참여하기

챌린지 상세페이지

[홈페이지](#)[인기 챌린지](#)[커뮤니티](#)[로그아웃](#)[My Page](#)[My Cash : 25000](#)[Cash+](#)[로그인된 아이디 : hello](#)

참여 인원 3명

챌린지 주인 : hello

hello

test1

test2

참가비용 : 30000

쌓인 상금 : 90000

나의 예상 환급금액 : 30000

[페이지 상태 최신화](#)[챌린지 중도표기](#)[챌린지 끝내기](#)[챌린지 삭제](#)

챌린지 상세페이지

챌린지 상세 페이지

쌓인 상금 += 참가비용

환급금액 = 쌓인상금 / 참여인원수

참여 인원 3명

챌린지 주인 : hello

hello

test1

test2

참가비용 : 30000

쌓인 상금 : 90000

나의 예상 환급금액 : 30000

페이지 상태 최신화

챌린지 중도포기

챌린지 끝내기

챌린지 삭제

중도포기 하는 사람이 많아질 수록 환급금액UP

챌린지 끝내기

My Cash : 25000

Cash+

로그인된 아이디 : hello

나의 예상 환급금액 : 30000

페이지 상태 최신화

챌린지 중도포기

챌린지 끝내기

챌린지 삭제

My Cash : 55000

Cash+

로그인된 아이디 : hello

챌린지를 끝내면
참여한 인원들에게
환급금액이 돌아감

챌린지 끝내기

유저의 cash 업데이트 & 챌린지 삭제

```
// 챌린지에 참여한 모든 유저의 캐시를 업데이트합니다.
```

```
await db.collection('user').updateMany(  
  { username: { $in: card.in_user_ID } },  
  { $inc: { cash: individual_prize } }  
);
```

```
// 챌린지 정보를 삭제합니다.
```

```
await db.collection('card').deleteOne({ _id: new ObjectId(request.params.id) });
```

6. 느낀점

- 처음하다보니 프로젝트 관리가 잘 되지않음
- 체계적으로 개발하지 못한것같다.
- 특정 기능을 마감기한을 두고 개발 하고 했으면 좋았을텐데 떠오르는데로 개발하니 놓치는것도 생기고 시간관리도 효율적이지 못했던것같다.

7. 참고문헌

- 코딩애플 2시간만에 웹서버 개발하기(Node.js +MongoDB)

https://www.youtube.com/playlist?list=PLfLgtT94nNq1qmsvlii_CAxFID7tvB5NE

- 구글 검색, 각종 블로그