OTP를 활용한 로그인 방식

One-Time

INDEX

조원편성

이름	역할	공통 역활
노현철	팀장, 발표	코딩
강승구	OTP 메인 코딩	
김정윤	PPT	
장현수	PPT	
한소희	웹 페이지 코딩	

주제선정 및 동기

현재 중부대학교 웹 로그인은 로그인외에는 별다른 보호방법이 존재하지 않아서 정보보호에 취약한 상황이다.

웹 로그인시 2차적인 보호에는 무엇이 있을까 고민하여 “OTP(일회용비밀번호)”를 활용한다면 좋을 것 같다고 생각했다.

2차적인 보호방법이 있어 개인정보 유출이나 위험적인 측면이 감소한다면, 정보보호에 있어서 나아질 것이다.

프로젝트의 목표와 설계

이번 프로젝트에서 로그인 시스템에
사용할 수 있는 일회용 패스워드를 제작해
취약한 로그인 보안 문제를 개선하고

더 나아가 프로젝트 성공시 현재 단일 패스워드만을
사용하는 중부대학교 홈페이지의 로그인 시스템 또한
개선할 수 있는 효과를 기대할 수 있을거라 생각했다.

요새 많은 우리들이 사용하는 사이트들이 다들 OTP를
연계해서 사용한다.

개인정보 침해 피해상황도 늘어나는 지금, OTP를 활용
한 이중 보안의 효과는 이미 입증되어 있기에 이를 직
접 설계해보기로 하였다.

OTP-One Time Password

일회성으로 사용하는 비밀번호이다

OTP는 주로 높은 수준의 보안을 유지하며 사용자를 인증해야 할 필요가 있을 때 사용되며, 그 일반적인 경우는 아래와 같다.

- 온라인 뱅킹
- 온라인 게임
- 포털 사이트
- 기업 네트워크

도입배경

OTP 이외의 사용자 인증 수단으로는 사용자 아이디 및 암호를 이용한 방식, 추기 가지가 있는데,

대부분 한 번 인 전에는 해킹을 당하기

이를 보완하기 만, 이 역시 정해진 수십여 유출에 의한 위험성을 가

이에 따라, 한 번에 한해서만 유효한 OT

OTP원리


② OTP값 입력
사용자 OTP
① OTP값 생성
OTP값 = 암호함수(비밀키, 동기화 정보)
사용자 PC
③ OTP값 전송
(거래정보 포함)
④ OTP값 검증
OTP' = 암호함수(비밀키, 동기화 정보)
전송된 OTP값이 OTP' 와 동일한지 비교
금융기관 (OTP 인증서버)

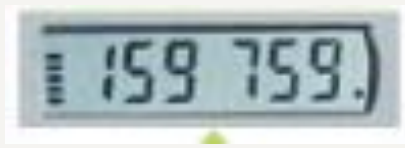
개발환경 및 OTP 동작방식

개발 환경 및 OTP 동작방식

TIME+SEED

OTP 생성 알고리즘

생성된 OTP



ID OTP

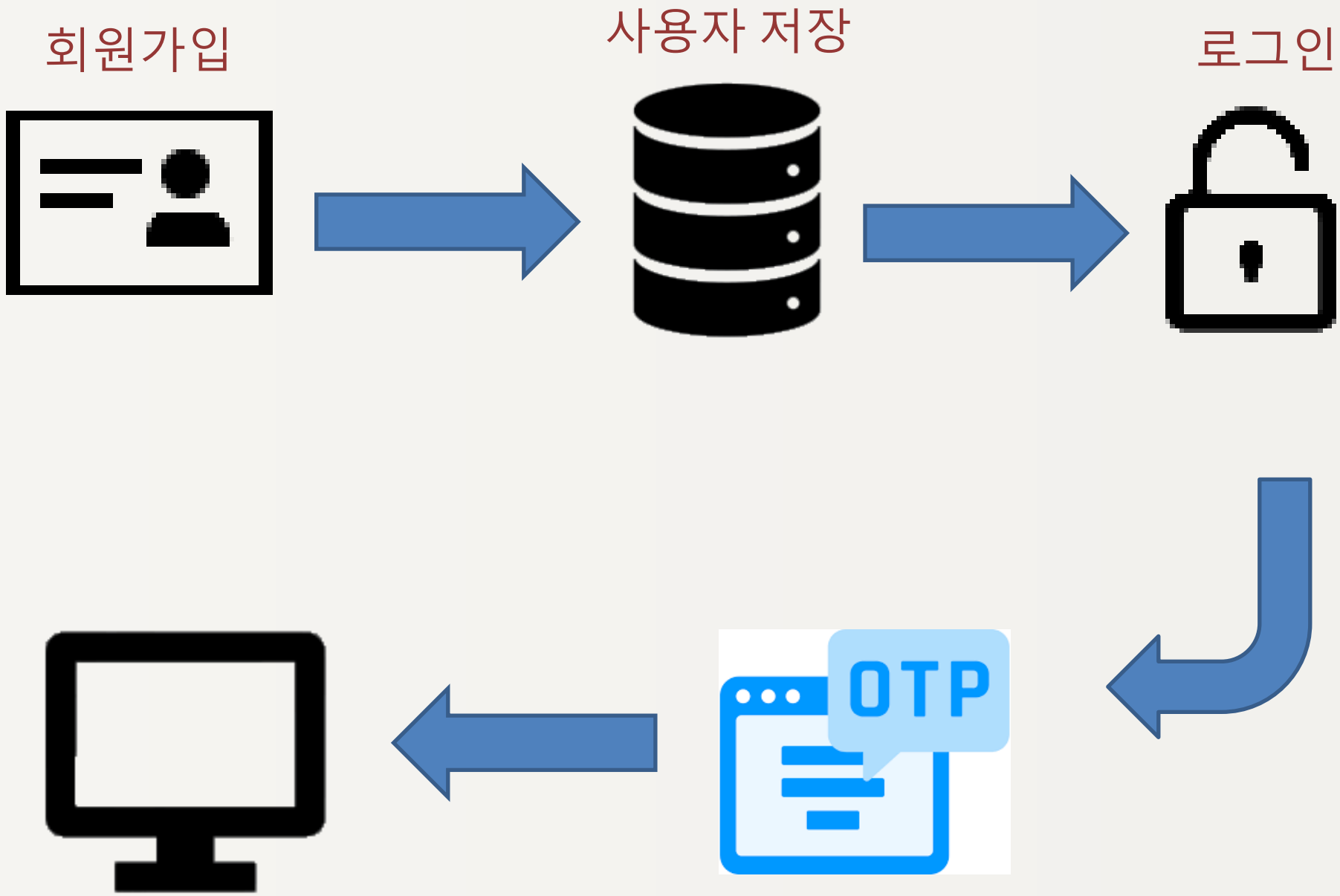
OTP확인

TIME+SEED

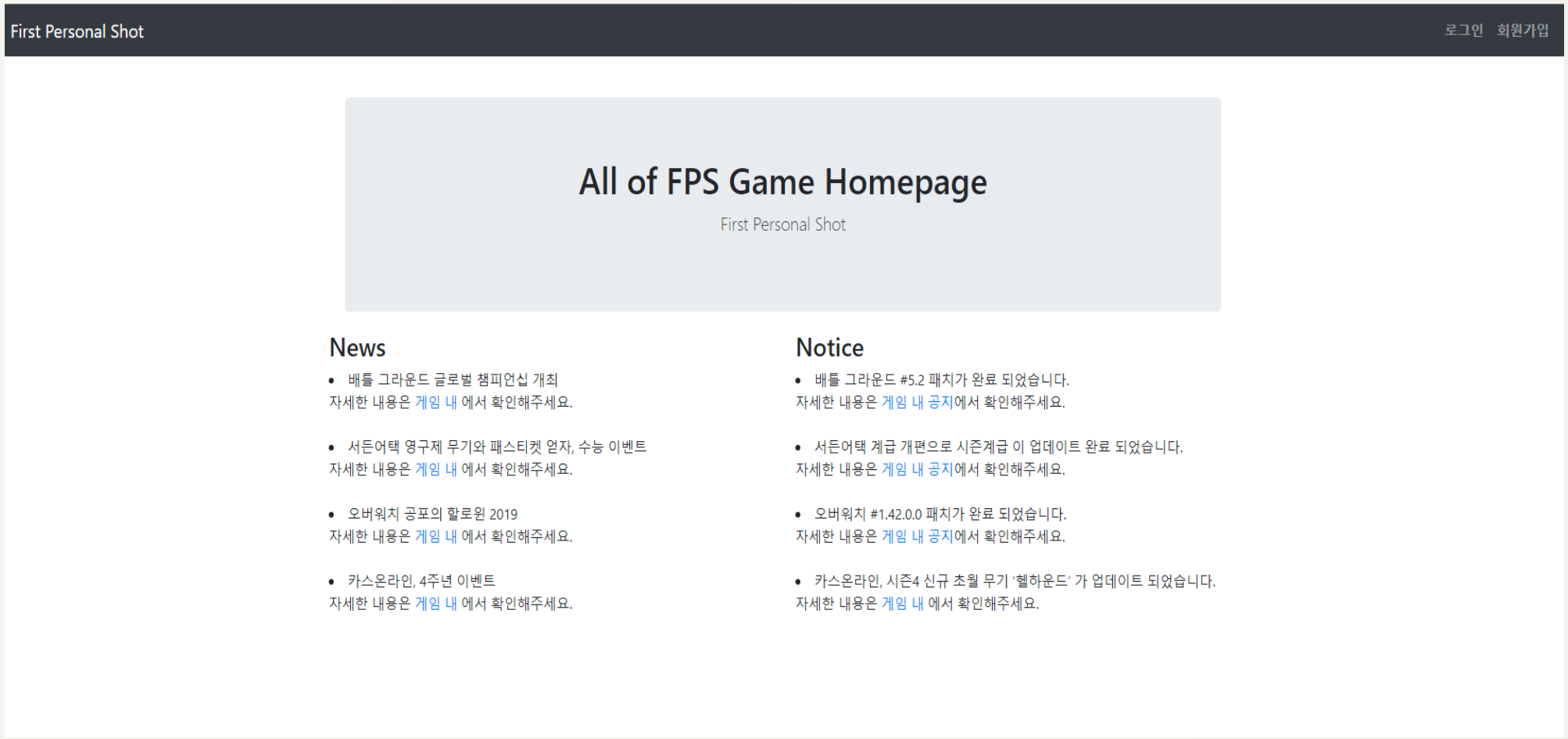
OTP 생성 알고리즘

생성된 OTP

OTP인증



메인 페이지



→ FPS 게임 의 뉴스및 공지등 각종소식을 한눈에 알아볼수있는 페이지

사용자 등록

→ 사용자등록시 사용자의 이름, ID, 닉네임, 즐겨하는 게임, 나이, Password 를 기입해서 사용자정보를 데이터베이스에 등록

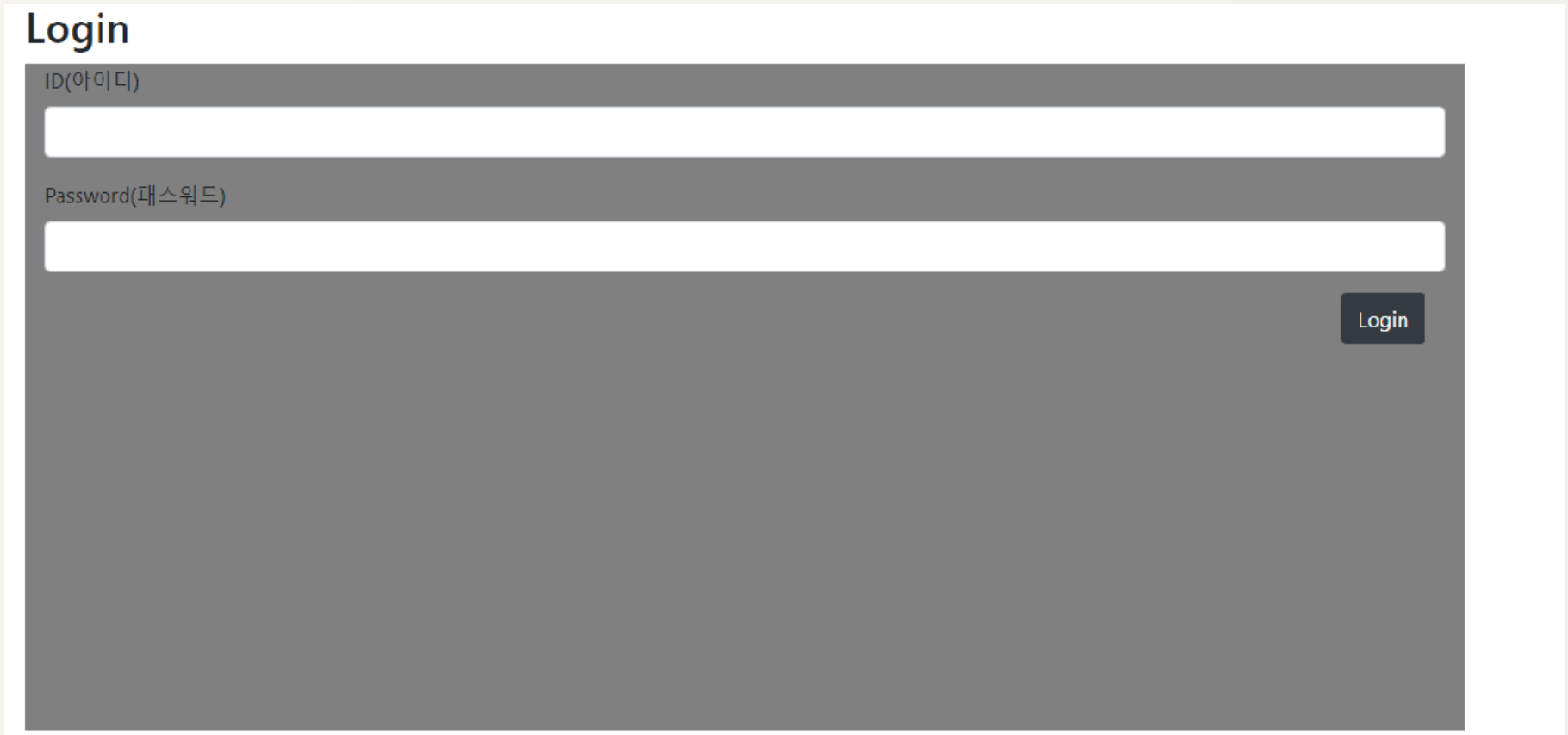
사용자 등록 코드

```
<h2 class="page header">사용자 등록</h2>
<form (ngSubmit)="onRegisterSubmit()">
  <div class="form-group">
    <label> Name 이름 </label>
    <input type="text" [(ngModel)]="name" name="name" class="form-control" />
  </div>
  <div class="form-group">
    <label> ID 아이디 </label>
    <input
      type="text"
      [(ngModel)]="username"
      name="username"
      class="form-control"
    />
  </div>
  <div class="form-group">
      <label> 닉네임 </label>
      <input type="text" [(ngModel)]="nickname" name="nickname" class="form-control" />
    </div>

<div class="form-group">
  <label> 즐겨하는 게임 </label>
  <input type="text" [(ngModel)]="email" name="email" class="form-control" />
</div>
<div class="form-group">
  <label> 나이 </label>
  <input type="number" [(ngModel)]="age" name="age" class="form-control" />
</div>
<div class="form-group">
  <label> Password 패스워드 </label>
  <input
    type="password"
    [(ngModel)]="password"
    name="password"
    class="form-control"
  />
```

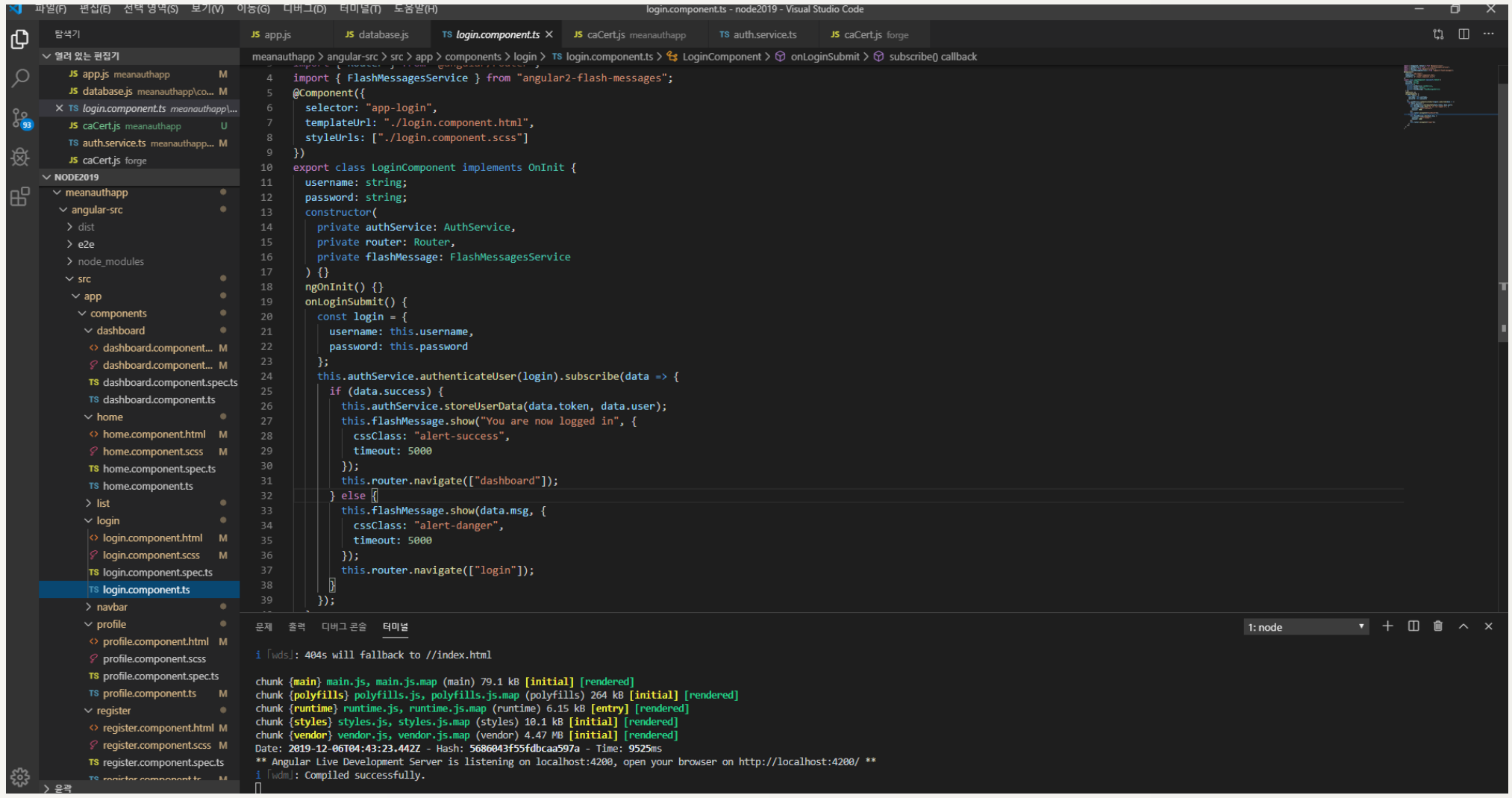
문제 출력 디버그 콘솔 터미널

로그인



→ 사용자 등록 시에 데이터베이스에 저장했던 정보로 로그인

로그인 코드


login.component.ts - node2019 - Visual Studio Code
meanauthapp > angular-src > src > app > components > login > login.component.ts > LoginComponent > onLoginSubmit > subscribe() callback
import { FlashMessagesService } from "angular2-flash-messages";
@Component({
selector: "app-login",
templateUrl: "./login.component.html",
styleUrls: ["./login.component.scss"]
})
export class LoginComponent implements OnInit {
username: string;
password: string;
constructor(
private authService: AuthService,
private router: Router,
private flashMessage: FlashMessagesService
) {}
ngOnInit() {}
onLoginSubmit() {
const login = {
username: this.username,
password: this.password
};
this.authService.authenticateUser(login).subscribe(data => {
if (data.success) {
this.authService.storeUserData(data.token, data.user);
this.flashMessage.show("You are now logged in", {
cssClass: "alert-success",
timeout: 5000
});
this.router.navigate(["dashboard"]);
} else {
this.flashMessage.show(data.msg, {
cssClass: "alert-danger",
timeout: 5000
});
this.router.navigate(["login"]);
}
});
chunk {main} main.js, main.js.map (main) 79.1 kB [initial] [rendered]
chunk {polyfills} polyfills.js, polyfills.js.map (polyfills) 264 kB [initial] [rendered]
chunk {runtime} runtime.js, runtime.js.map (runtime) 6.15 kB [entry] [rendered]
chunk {styles} styles.js, styles.js.map (styles) 10.1 kB [initial] [rendered]
chunk {vendor} vendor.js, vendor.js.map (vendor) 4.47 MB [initial] [rendered]
Date: 2019-12-06T04:43:23.442Z - Hash: 5686043f55fdbcaa597a - Time: 9525ms
** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **
i ｢wdm｣: Compiled successfully.

게임 리스트

FPS 게임 리스트

게임리스트 코드

```
<div class="container">
<div class="home_title_ctn">
  <div class="home_title">FPS 게임 리스트</div>
  <div class="home_line"></div>
</div>
<div class="home_topsellers_games_ctn" >
  <div class="a1">
<a href="http://pubg.daum.net/" >
  <img  src="https://steamcdn-a.akamaihd.net/steam/apps/578080/capsule_616x353.jpg?t=1572384051" >
</a>
</div>
<a href="https://store.steampowered.com/app/359550/Tom_Clancys_Rainbow_Six_Siege/?snr=1_4_autumnsale__104">
    <img  src="https://steamcdn-a.akamaihd.net/steam/apps/359550/capsule_616x353.jpg?t=1574941292">
</a>
</div>
<br>
<br>
<div class="home_topsellers_games_ctn1">
<a href="https://store.steampowered.com/app/550/Left_4_Dead_2/">
  <img  src="https://steamcdn-a.akamaihd.net/steam/bundles/233/38smshw5fjvh9vp4/capsule_616x353.jpg?t=1456861419">
</a>
<a href="http://sa.nexon.com">
<img src="http://www.nexon-gt.com/common/images/icon/game2_movie2.jpg" width="616px" height="353px"></a>
</div>
<br>
<br>
<div class="home_topsellers_games_ctn2">
    <a href="http://csonline.nexon.com/">
      <img  src="https://post-phinf.pstatic.net/MjAxODEyMDdfMjIz/MDAxNTQ0MTY1Njg2MTU5.969nTjJFfJm1NFhGQyOXD5TBmQSLMvMt0ArXvwdr_kwg.y2JiIZulK1cPknMcYPqntVpQQU6gd4HzDNOwUzIFbGgg.JPEG/%ED
    </a>
    <a href="https://playoverwatch.com/ko-kr/">
    <img src="http://img.tf.co.kr/article/home/2016/02/17/201637941455648532.jpg" width="616px" height="353px"></a>
    </div>
    <br>
<br>
</div>
```

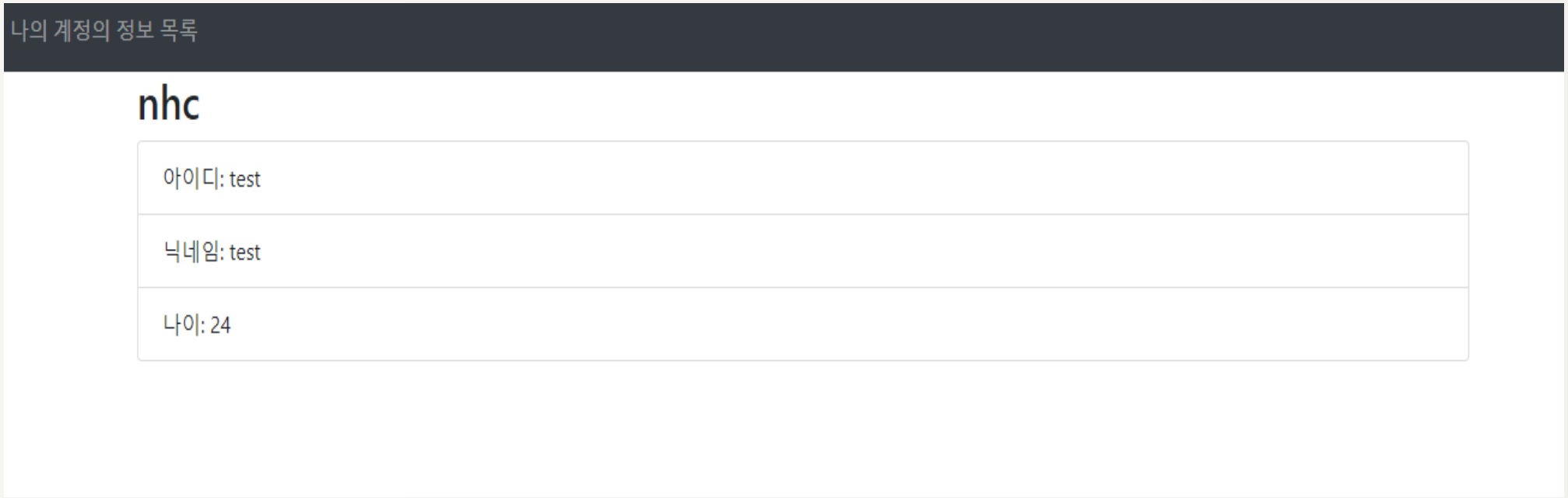
사용자 목록

→ 사용자의 즐겨하는 게임목록을 한눈에 알아볼 수 있다.

사용자 목록 코드

```html
<h2 class="page-header"> 사용자 목록</h2>
<table class="table table-bordered table-striped">
  <thead class="thead-dark">
    <tr>
      <th>이름</th>
      <th>아이디</th>
      <th>닉네임</th>
      <th>즐겨하는 게임</th>
      <th>나이</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let user of users">
      <td>{{ user.name }}</td>
      <td>{{ user.username }}</td>
      <td>{{ user.nickname }}</td>
      <td>{{ user.email }}</td>
      <td>{{ user.age }}</td>
    </tr>
  </tbody>
</table>
```

마이페이지



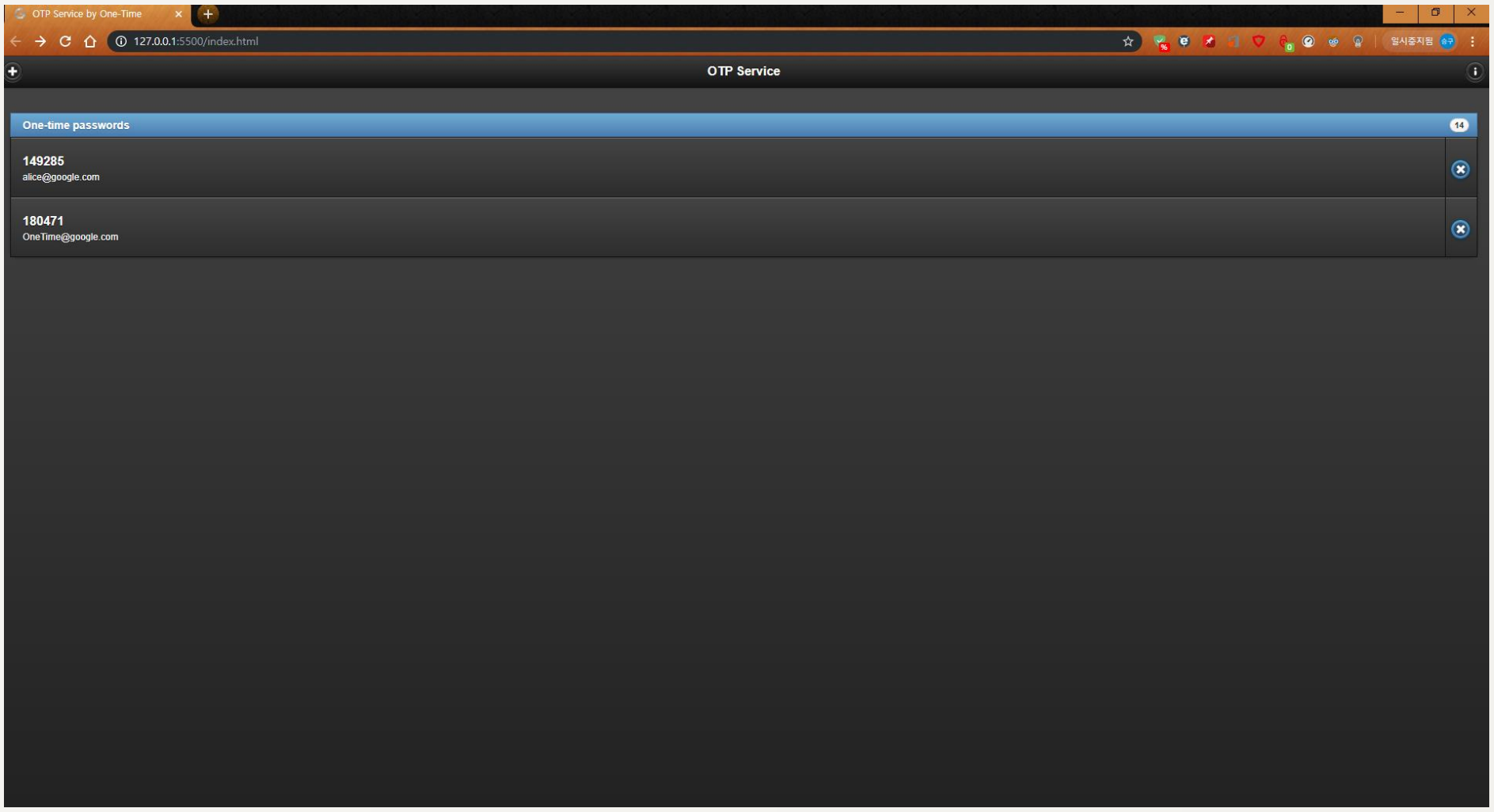
→ 사용자 등록 시 데이터베이스에 등록했던 정보를 로그인 했을 경우에만 마이 페이지 에서 볼 수 있다.

마이페이지 코드

```
<div *ngIf="name">
  <h2 class="page-header">{{ name }}</h2>
  <ul class="list-group">
    <li class="list-group-item">아이디: {{ username }}</li>
    <li class="list-group-item">닉네임: {{ nickname }}</li>
    <li class="list-group-item">나이: {{ age }}</li>
  </ul>
</div>

```

사용자 등록/인증



OTP 인증할 사용자를 데이터베이스에 등록할 수 있는 페이지 이다.

사용자 등록/인증 코드

OTP > index.html > html > head > script

```
<!DOCTYPE html public "Gerard Braad">
<html>
    <head>
        <title>OTP Service by One-Time</title>
        <meta charset="utf-8">
        <meta name="description" content="GAuth Authenticator">
        <meta name="HandheldFriendly" content="True">
        <meta http-equiv="cleartype" content="on">
        <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
        <!-- purposely at the top -->
        <script src="jquery-1.7.2.min.js"></script>
        <script src="init.js"></script>
        <script src="jquery.mobile-1.1.0.min.js"></script>
        <script src="sha.js"></script>
        <script src="code.js"></script>
        <link rel="stylesheet" href="jquery.mobile-1.1.0.min.css" />
        <link rel="stylesheet" href="styling.css" />
        <link rel="shortcut icon" href="./favicon.ico" />
        <link rel="apple-touch-icon" href="./images/icon-48.png" />
    </head>
    <body>
        <div data-role="page" id="main" data-theme="a">

            <div data-role="header">
                <h1>OTP Service</h1>
                <a href="#settings" data-role="button" data-rel="dialog" data-icon="plus" data-iconpos="notext">Settings</a>
                <a href="#about" data-role="button" data-rel="dialog" data-icon="info" data-iconpos="notext">About</a>
            </div>

            <div data-role="content">
                <ul data-role="listview" data-inset="true" data-theme="a"  data-split-theme="b" data-split-icon="delete" id="accounts">
                    <li id="accountsHeader" data-role="list-divider">One-time passwords<span class="ui-li-count" id='updatingIn'>..</span></li>
                </ul>
            </div>

            <div data-role="footer">
                <h1>@gbraad</h1>
```

결론 및 기대효과

OTP를 사용하여 로그인시 이중 보안절차를 거쳐 실제적인 보안능력 향상
DB를 해킹하여 사용자 아이디와 패스워드 등 로그인 정보를 알아내더라도 제공된 OTP 서비스로 인하여 본인이 아닌 사용자의 로그인이 불가

해당 OTP를 중부대학교 웹이나 app 로그인시 유용하게 활용될 것이다.
사용자들이 웹이나 APP에서의 OTP 사용을 하며
2차 보호방법인 이중 인증 기술에 대한 관심과 이해도를 높일 것이다.
추후 중부대 웹과 APP과의 연계를 통해 보안을 강화 할 수 있는 목표로 한다.

완성 후 성능 분석

2중 로그인을 통해 본인의 계정을 보다 안전하게 이용할 수 있게 되었다.

서버에서 1단계 로그인 후 OTP 번호를 받아
2단계 로그인을 통해 보다 안전한
로그인이 가능해졌다.

따라서 홈페이지 이용시 개인정보에 대한 보안 능력이 향상되었음을 확인하였다.

감사합니다

홈페이지 링크: https://limitless-journey-71231.herokuapp.com/