

24.01.12

| | |
|---------|-----------------------|
| 🕒 작성 일시 | @2024년 1월 23일 오후 1:11 |
| ☰ 태그 | |

SignChack.js

192.168.213.142.0000 네 등.

약관에 동의하세요

이용약관

확인

제1조(목적) 이 약관은 업체 회사(전자상거래 사업자)가 운영하는 업체 사이버 물(이하 "물"이라 한다)에서 제공하는 인터넷 관련 서비스(이하 "서비스"라 한다)를 이용함에 있어 사이버 물과 이용자의 권리·의무 및 책임사항을 규정함을 목적으로 합니다. ※「PC통신, 무선 등을 이용하는 전자상거래에 대해서도 그 성질에 반하지 않는 한 이 약관을 준용합니다.」 제2조(정의) ① "물"이란 업체 회사가 재화 또는 용역(이하 "재화 등"이라 함)을 이용자에게 제공하기 위하여 컴퓨터 등 정보통신설비를 이용하여 재화 등을 거래할 수 있도록 설정한 가상의 영업장을 말하며, 아울러 사이버물을 운영하는 사업자의 의미로도 사용합니다. ② "이용자"란 "물"에 접속하여 이 약관에 따라 "물"이 제공하는 서비스를 받는 회원 및 비회원을 말합니다. ③ '회원'이라 함은 "물"에 회원등록을 한 자로서, 계속적으로 "물"이 제공하는 서비스를 이용할 수 있는 자를 말합니다. ④ '비회원'이라 함은 회원에 가입하지 않고 "물"이 제공하는 서비스를 이용하는 자를 말합니다. 제3조 (약관 등의 명시와 설명 및 개정) ① "물"은 이 약관의 내

서비스 이용 약관에 동의 하시겠습니까? ☒

개인정보 수집 및 이용 동의 (필수)

개인정보처리방침 [차례] 1. 총칙 2. 개인정보 수집에 대한 동의 3. 개인정보의 수집 및 이용목적 4. 수집하는 개인정보 항목 5. 개인정보 자동수집 장치의 설치, 운영 및 그 거부에 관한 사항 6. 목적 외 사용 및 제3자에 대한 제공 7. 개인정보의 열람 및 정정 8. 개인정보 수집, 이용, 제공에 대한 동의철회 9. 개인정보의 보유 및 이용기간 10. 개인정보의 파기절차 및 방법 11. 아동의 개인정보 보호 12. 개인정보 보호를 위한 기술적 대책 13. 개인정보의 위탁처리 14. 의견수렴 및 불만처리 15. 부 칙(시행일) 1. 총칙 본 업체 사이트는 회원의 개인정보 보호를 소중하게 생각하고, 회원의 개인정보를 보호하기 위하여 항상 최선을 다해 노력하고 있습니다. 1) 회사는 「정보통신망 이용촉진 및 정보보호 등에 관한 법률」을 비롯한 모든 개인정보보호 관련 법률규정을 준수하고 있으며, 관련 법령에 의거한 개인정보처리방침을 정하여 이용자 권익 보호에 최선을 다하고 있습니다. 2) 회사는 「개인정보처리방침」을 제정하여 이를 준수하고 있으며, 이를 인터넷사이트 및 모바일 어플리케이션에 공개하여 이용자가 언제나

개인정보 수집 및 이용 약관에 동의 하시겠습니까? ☐

회원가입

```

<script>

export default {
  methods: {
    nextPage() {
      const acc = document.getElementsByClassName("acc-or");
      // acc.checked = true
      if(acc[0].checked == true && acc[1].checked == true ){
        location.href = "SignUp"
      }else{alert("약관에 동의하세요");
        return;
      }
    },
  },
};
</script>

```

SignUp.js

회원가입

이름*

최혁

아이디*

ge

중복 확인

비밀번호*

192.168.219.142:8080 내용:

사용 불가능한 아이디 입니다.

확인

```

async checkUserId(id){
const target = document.getElementById('sign_up_btn');
  try{
    await axios.post(`http://localhost:5000/user/${id}`, {
      UserId: this.userId,
    }).then(function(res){
      console.log(id);
      console.log(res.data[0]);
      if(id == ""){
        alert("아이디를 입력해주세요");
        return;
      }
      if(res.data[0] == undefined){
        alert("사용 가능한 아이디 입니다.");
        target.disabled = false;
      }else{
        alert("사용 불가능한 아이디 입니다.");
      }
    }).catch(function(err){
      console.log(err);
    });
    // this.userId = "";
    // this.$router.push("/");
  }catch(err){
    console.log(err);
  }
},

```

BACK_SERVER 포트

```

app.listen(5000, () => console.log('Server running at http://localhost:5000')

```

아이디 중복 검사

router 경로

```
router.post('/user/:id', checkUserById);
```

Controller

```
export const checkUserById = (req, res) => {  
  const id = req.params.id;  
  console.log(req.body);  
  getUserById(id, (err, results) => {  
    if (err){  
      res.send(err);  
    }else{  
      res.json(results);  
    }  
  });  
}
```

Model

```
// Get Single User  
export const getUserById = (id, result) => {  
  db.query("SELECT * FROM user WHERE UserId like ?", [id], (err, results) => {  
    if(err) {  
      console.log(err);  
      result(err, null);  
    }else {  
      // console.log(results)  
      result(null, results);  
    }  
  });  
}
```

회원가입.js

```

async createUser() {
  try {
    await axios.post("http://localhost:5000/user", {
      UserNa: this.userName,
      UserId: this.userId,
      UserPassword: this.userPw,
      UserPhon: this.userPhon,
      UserEmail: this.userEmail,
    });
    console.log(this.userName[0].value);
    this.userName = "";
    this.userId = "";
    this.userPw = "";
    this.userPhon = "";
    this.userEmail = "";
    this.$router.push("/login");
    console.log("성공");
  } catch (err) {
    console.log(err);
    console.log("실패");
  }
},

```

router 경로

```
router.post('/user', createUser);
```

Controller

```
export const createUser = (req, res) => {
  const data = req.body;
  insertUser(data, (err, results) => {
    if (err){
      res.send(err);
    }else{
      res.json(results);
    }
  });
}
```

Model

```
export const insertUser = (data, result) => {
  db.query("INSERT INTO user SET ?", [data], (err, results) => {
    if(err) {
      console.log(err);
      result(err, null);
    } else {
      result(null, results);
    }
  });
};
```

유효성 검사

```

    signUpButton(){
        let testid = document.getElementsByClassName("testid");
        // console.log(testid);
        if (testid[0].value == ""){
            alert("아이디를 입력해주세요")
        } else{
            this.pwCheck();
        }
    },
    pwCheck() {
        let pw = document.getElementsByClassName("signup_pw");
        let pw_check = document.getElementsByClassName("signup_pw_check");
        if (pw[0].value == pw_check[0].value && pw[0].value.length >= 8) {
            this.phonCheck()
        } else if (pw[0].value.length <= 8 || pw_check[0].value.length <= 8) {
            alert("8자리 이상 입력하세요");
        } else if (pw[0].value !== pw_check[0].value) {
            alert("비밀번호가 일치하지 않습니다.");
        }
        return
    },
    phonCheck(){
        let Pdata= document.getElementsByClassName("PhonData");
        // console.log(Pdata[0].value)
        if (Pdata[0].value == ""){
            alert("전화번호를 입력해주세요")
        }else if(11<= Pdata[0].value.length <= 13){
            this.emailCheck()
        }else {alert("형식에 맞게 작성해주세요")}
    },
    emailCheck(){
        let Edata = document.getElementsByClassName('EmailData');
        if(Edata[0].value == ""){
            alert("Email을 입력해 주세요");
        }else{
            this.createUser();
            alert("가입 완료");
        }
    }
},

```