

Book friend



목차

01 ● ● 개발 동기 및 기대 효과

04 ● ● 팀원 소개

02 ● ● 개발 일정 및 개발 문서

05 ● ● 기술 소개

03 ● ● 프로젝트 도구

06 ● ● 기능 시연

01

개발동기 및 기대효과

01. 개발 동기

“하루라도 책을 읽지 않으면
입안에 가시가 돋는다”

-안중근 의사-

01. 프로젝트 동기

성인 48% 1년동안
책 한 권도 안 읽어...
작년 독서량 6.1권



01. 프로젝트 동기



일대일 책 대여 서비스

01. 기대효과



독서율증가



경제적인
문화생활





02

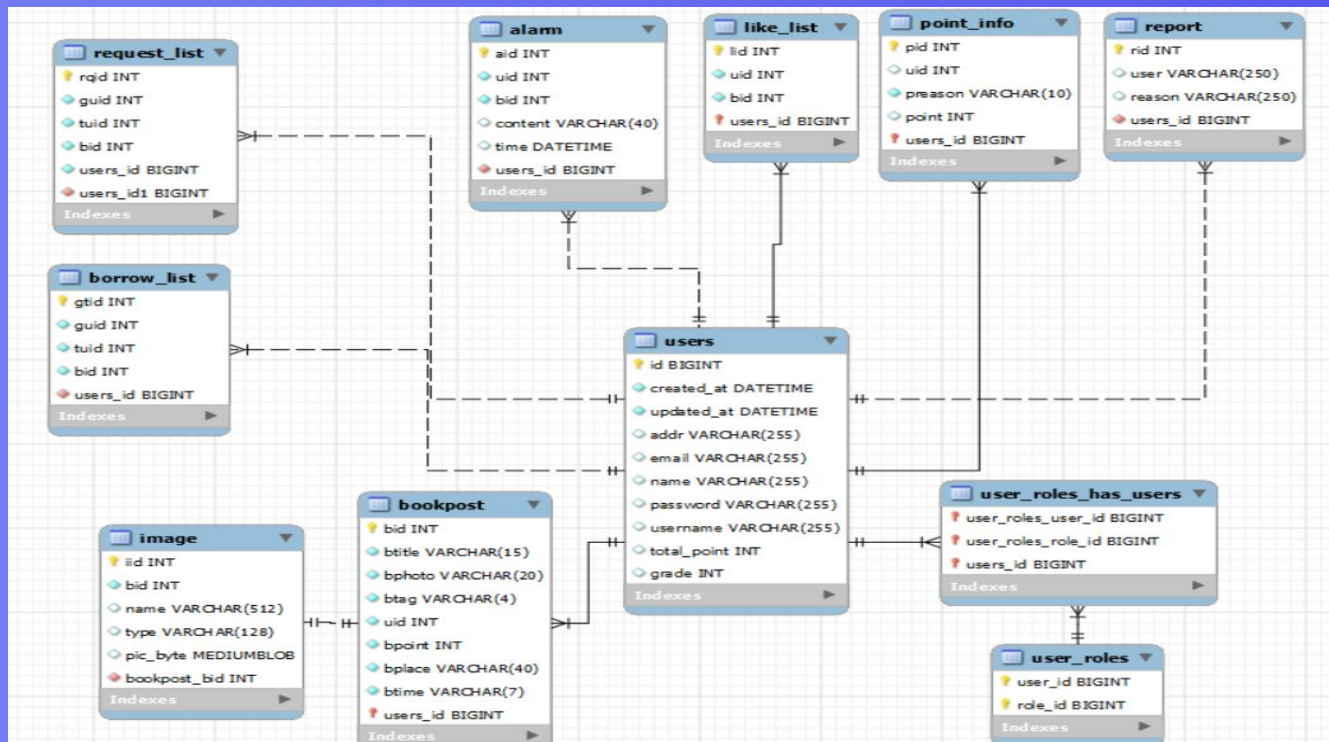
개발 일정 및 개발 문서



02. 시스템 흐름도



02. ERD



02. UI 스토리보드

스토리보드 - 책 대여
시스템

메인 페이지

일반 회원 가입 및 마이 페이지

마이페이지 기능 관련 페이지

설문조사 페이지

책 빌려주기 및 책
빌리기 페이지.

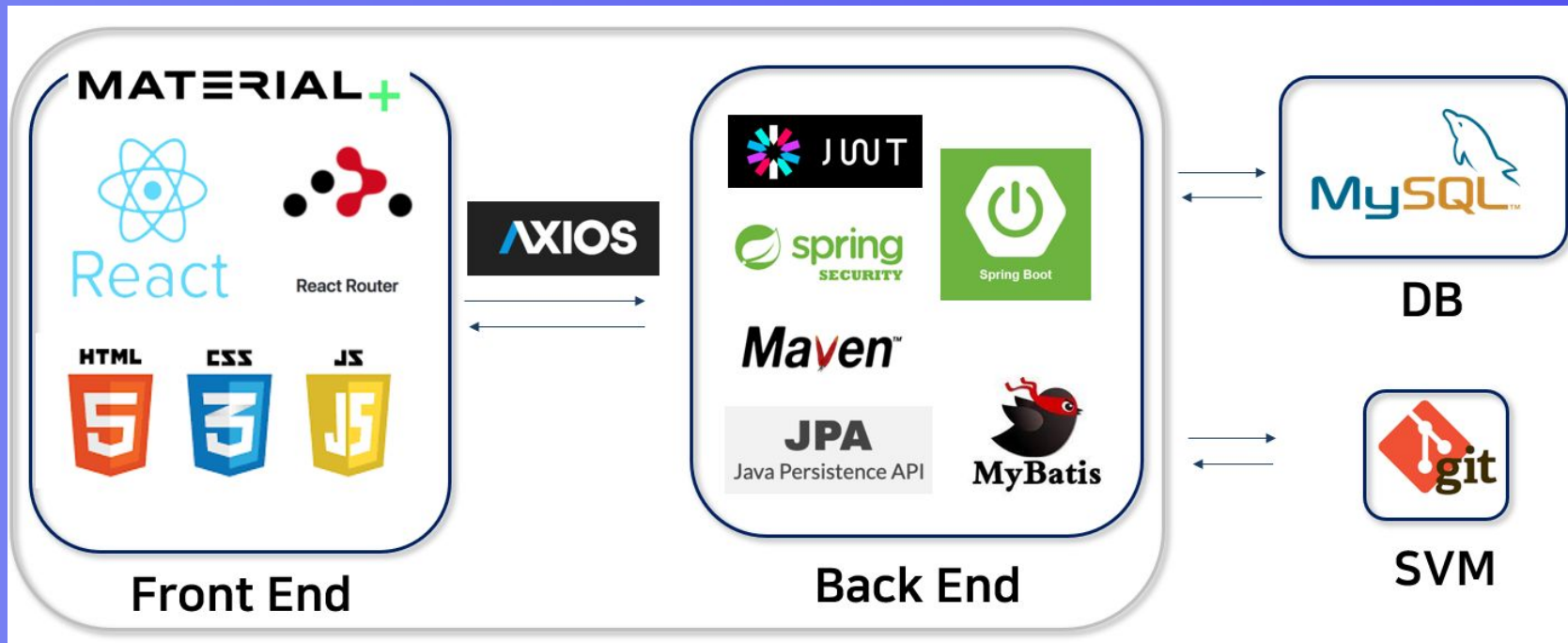


03

프로젝트 도구

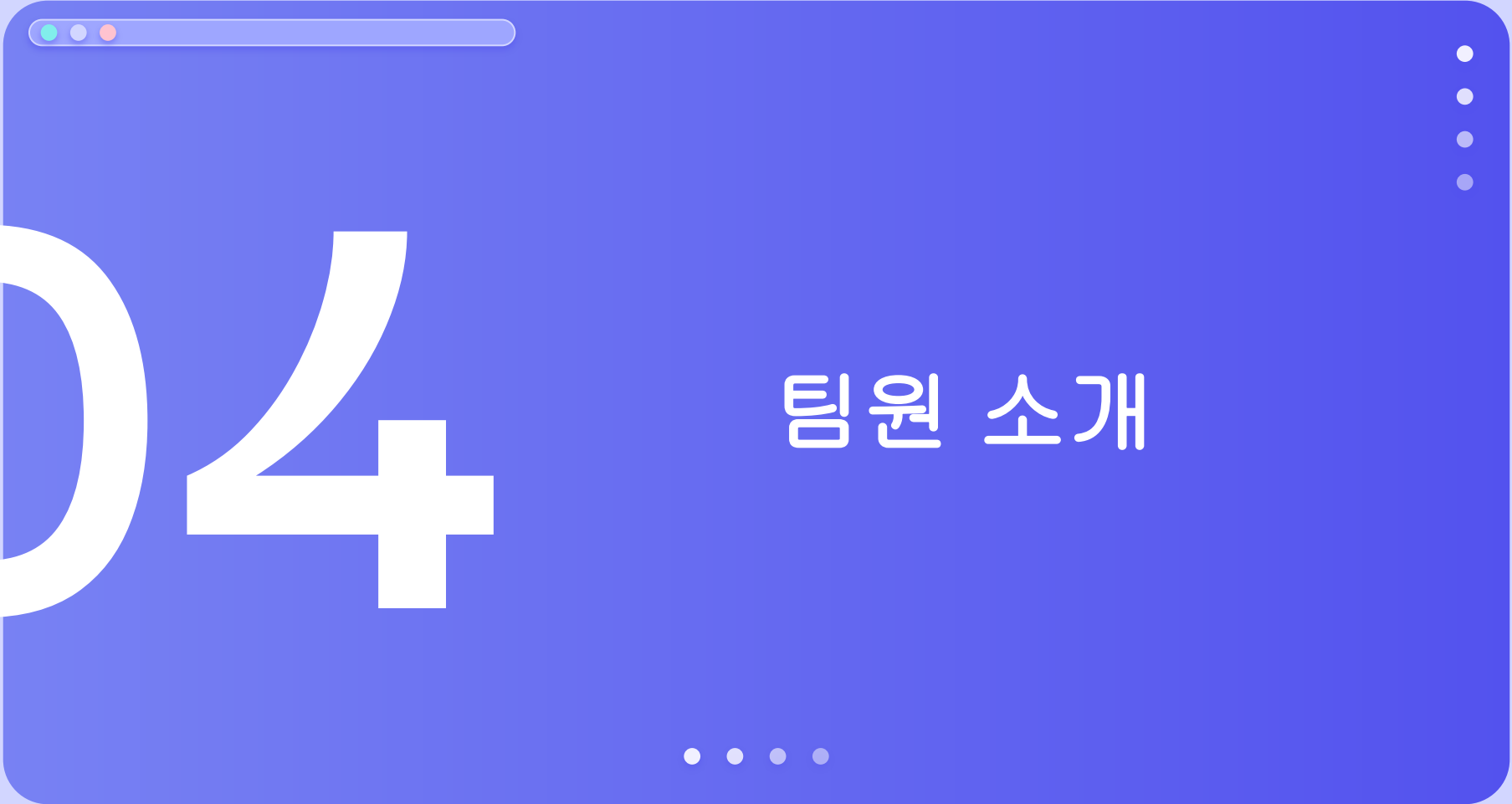


03. 프로젝트 도구



03. 프로젝트 도구





04

팀원 소개

04. 팀원 소개



김홍길

로그인, 회원가입,
수정,빌린책,
빌려준책
설문조사



김민선

책 빌려주기,
지도로보기,
신고,포인트



김주희

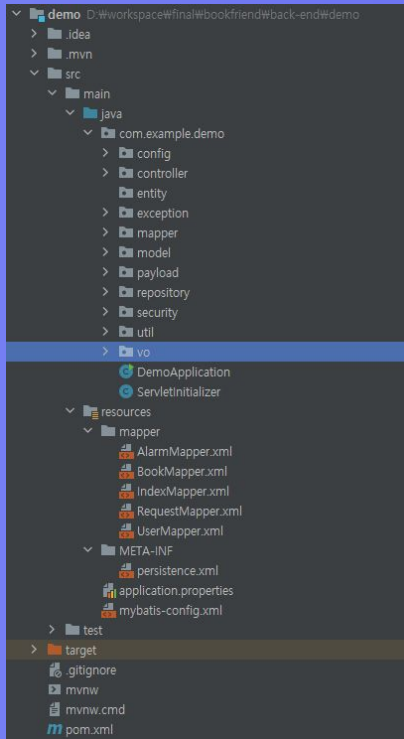
책리스트, 책상세,
마이페이지,
나의책, 찜목록,
요청,알림



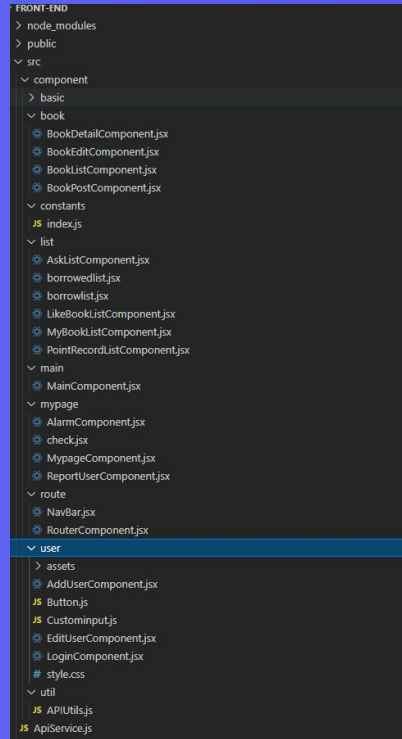
05

기술 소개

05. 프로젝트 구조

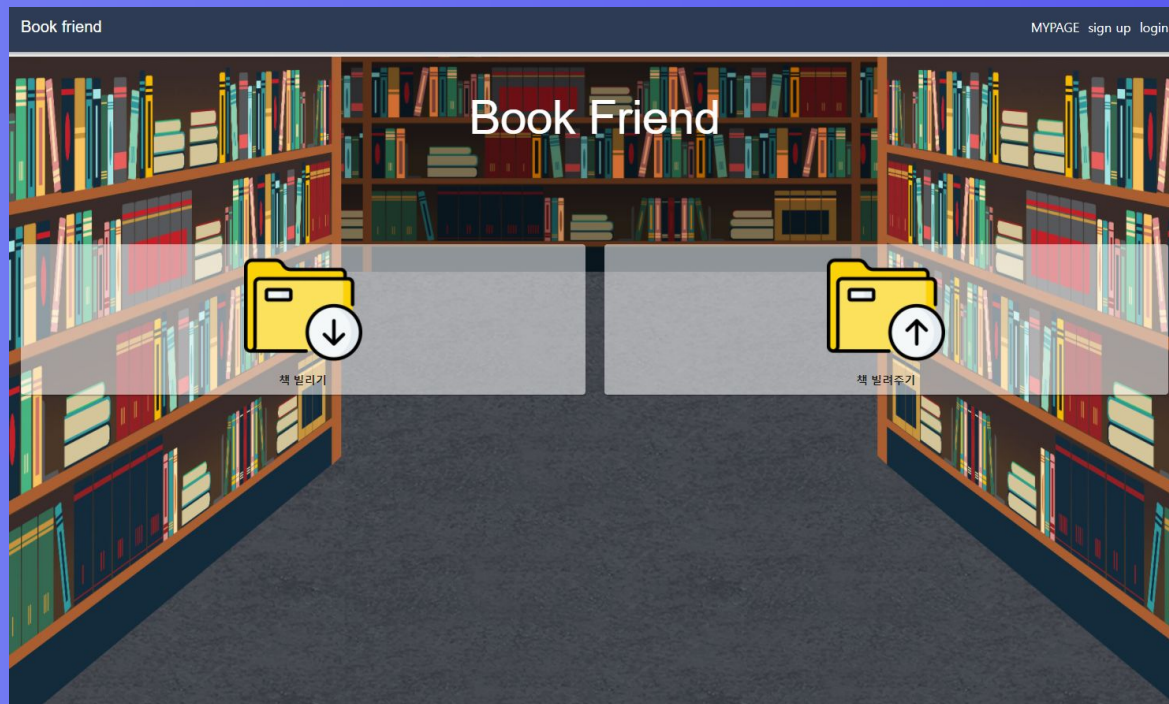


백엔드



프론트엔드

05. 기술 소개 - 메인 페이지



05.

```
saveUser = (e) => {
  e.preventDefault();

  let signupRequest = {
    name: this.state.name,
    email: this.state.email,
    username: this.state.username,
    password: this.state.password,
    addr: this.state.addr,
  }

  signup(signupRequest)
    .then( res => {
      this.setState({
        message: signupRequest.username + '님이 성공적으로 등록되었습니다.'
      })
      console.log(this.state.message);
      this.props.history.push('/login');
    })
    .catch( err => {
      console.log('saveUser() 에러', err);
    });
}

render(){
  return(
    <div>
      <Typography variant="h4" style={style}>Add User</Typography>
      <form style={formContainer}>

        <TextField placeholder="please input your nickname" name="name"
        fullWidth margin="normal" value={this.state.name} onChange={this.onChange}/>

        <TextField placeholder="please input your id" name="username"
        fullWidth margin="normal" value={this.state.username} onChange={this.onChange}/>
```

```

LoginUser = (e) => {
  e.preventDefault();
  console.log(this.state);
  let LoginRequest = {
    usernameOrEmail: this.state.usernameOrEmail,
    password: this.state.password
  }
  console.log(LoginRequest);
  login(LoginRequest)
    .then(response => {
      localStorage.setItem(ACCESS_TOKEN, response.accessToken);
      localStorage.setItem(isLoggedIn, true);
      localStorage.setItem(user_info, LoginRequest.usernameOrEmail);
      this.props.history.push('/');
      window.location.reload();
    })
    .catch(error => {
      if(error.status === 401) {
        alert('Your Username or Password is incorrect. Please try again!');
      } else {
        alert('something err!');
        console.log(error);
      }
    });
}

render() {
  return (
    <div className="App">
      <form className="form">
        <TextField placeholder="please input your id" name="usernameOrEmail"
fullWidth margin="normal" value={this.state.usernameOrEmail} onChange={this.onChange}/>
        <TextField type="password" placeholder="please input your password" name="password"
fullWidth margin="normal" value={this.state.password} onChange={this.onChange}/>
        <Button type="button" color="primary" className="form__custom-button" onClick={this.LoginUser}>
          Log in
        </Button>
      </form>
    </div>
  );
}
}

```

05. 기술 소개 - 책 빌려주기

Book friend

MYPAGE sign up login

Book Post

제목

포인트

시간

장소

파일 선택

선택된 파일 없음

UPLOAD

05. 기술 소개 - 책 빌려주기

```
let book = {  
  btitle : this.state.btitle,  
  bphoto : 1,  
  btag : "대출가능",  
  uid : this.state.uid,  
  bpoint : this.state.bpoint,  
  bplace : this.state.bplace,  
  btime : this.state.btime  
}  
  
ApiService.addBook(book)  
  .then( res=>{  
    this.props.history.push('/main');  
  })  
  .catch( err => {  
    console.log('handleUpload() addBook에러', err);  
  });
```

```
addBook(book){  
  return axios.post(USER_API_BASE_URL + '/books/book', book);  
}
```

Front

```
@PostMapping("/book")  
void insertBook(@RequestBody BookVO book) {  
  bookMapper.insertBook(book);  
  System.out.println("책 DB 저장 성공");  
}
```

Controller

```
void insertBook(BookVO book);  
void insertImage(ImageVO img);  
ImageVO fetchImageByID(int bid);
```

Mapper

```
<insert id="insertBook" parameterType="com.example.demo.vo.BookVO">  
  INSERT INTO bookpost(btitle,bphoto,btag,uid,bpoint,bplace,btime)  
  VALUES(#{btitle},#{bphoto},#{btag},#{uid},#{bpoint},#{bplace},#{btime})  
</insert>
```

Back

05. 기술 소개 - 책 리스트

Book friend

BookMapper.xml

```
reloadBookList = () => {  
  ApiService.fetchBooks()  
    .then( res => {  
      this.setState({  
        books: res.data  
      })  
    })  
    .catch(err => {  
      console.log('reloadBookList() Error!', err);  
    })  
}
```

```
6  
7  
8 <select id="bookList" resultType="com.example.demo.vo.BookVO">  
9   SELECT * FROM bookpost  
10  ORDER BY bid  
11 </select>
```

book1

100포인트

book2

100포인트

book5

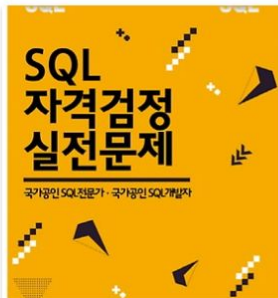
200포인트

book6

200포인트



대출가능



대출가능

05.

```
saveRequest = (e) => { //도서 대출 요청 db 저장
  e.preventDefault();
  alert("대출 요청 완료");
```

```
  let request = {
    guid:this.state.uid,
    tuid:window.localStorage.getItem("user_info"), //로그인 중인 user id
    bid:this.state.bid
  }
```

```
  ApiService.addrequest(request)
```

```
  let alarm = { //대여
    uid:this.state.uid, //
    bid:this.state.bid, //대출받은 도서 id
    content:this.state.btitle+' 대여 요청 '
  }
```

```
  ApiService.insertAlarm(alarm)
    .then( res => {
      this.props.history.push('/alarm')
    })
    .catch( err => {
      console.log('saveRequest error: ', err)
    });
}
```

PAGE LOGIN

```
<insert id="insertRequest" parameterType="com.example.demo.vo.BookVO">
  INSERT INTO request_list(guid, tuid, bid)
  SELECT #{guid}, id, #{bid}
  FROM users
  WHERE username=#{tuid}
</insert>
```

```
<insert id="insertAlarm" parameterType="com.example.demo.vo.AlarmVO">
  INSERT INTO alarm(uid, bid, content)
  VALUES(#{uid}, #{bid}, #{content})
</insert>
```

05. 기술 소개 - 마이페이지

```
saveBorrow = (request) => { //도서 대출 승인 db 저장
```

```
    alert("요청 승인 완료");
```

```
    let borrow = {
```

Book friend

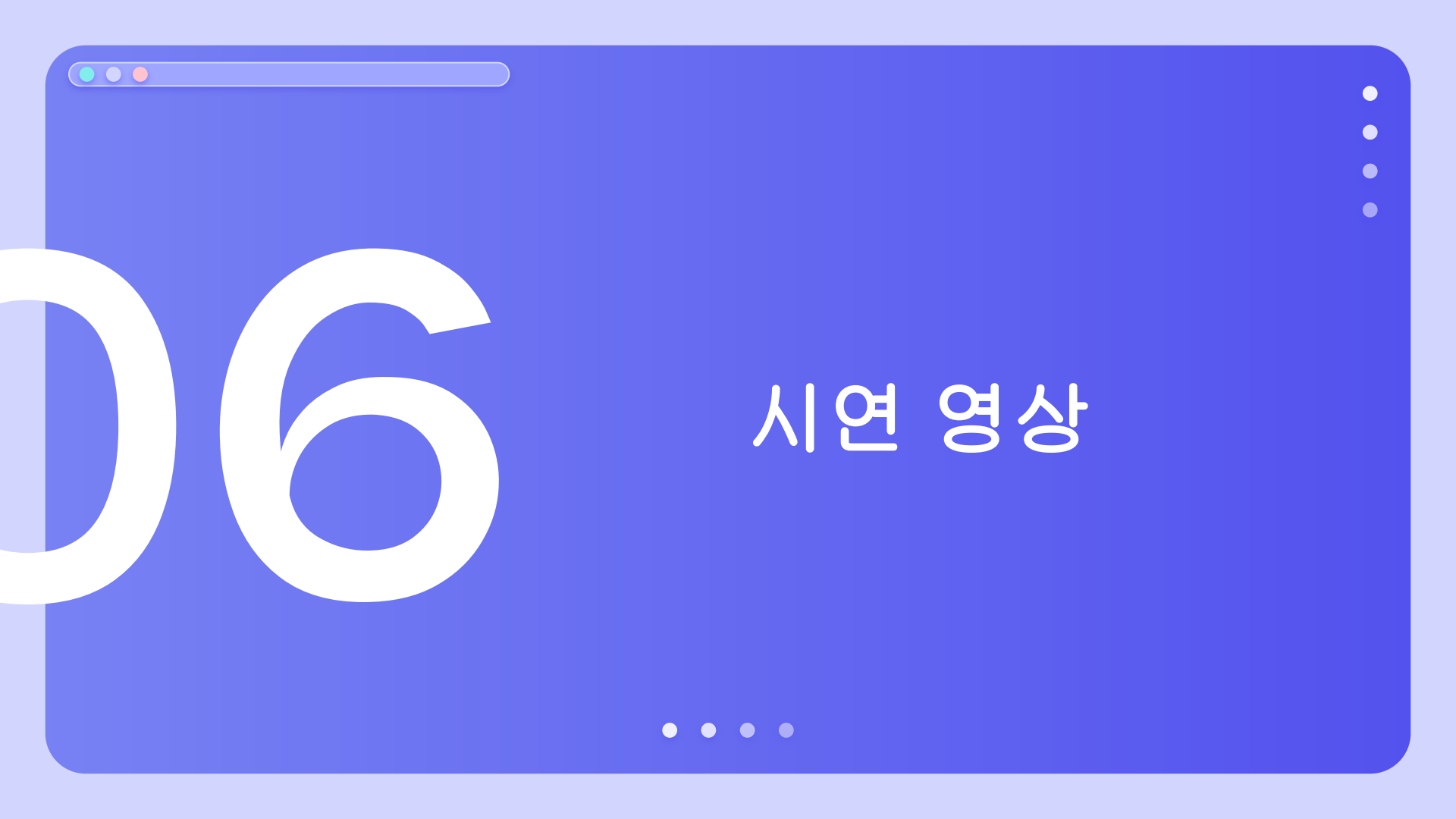
MYPAGE LOGIN

```
<select id="requestList" resultType="com.example.demo.vo.RequestlistVO">
    select rq.rqid, rq.bid, rq.tuid, rq.guid, bp.btitle, users.username
    from request_list AS rq
    JOIN bookpost AS bp ON bp.bid = rq.bid
    JOIN users ON users.id = rq.tuid
    where rq.tuid = #{tuid};
</select>
```

```
<delete id="deleteRequest" parameterType="int">
    DELETE FROM request_list
    WHERE rqid = #{rqid}
</delete>
```

```
<insert id="insertborrow" parameterType="com.example.demo.vo.borrowListVO">
    INSERT INTO borrow_list(guid, tuid, bid)
    VALUES(#{guid}, #{tuid}, #{bid})
</insert>
```

```
    }
}
```



06

시연 영상



감사합니다.