

데이터로 일하는 개발자

박홍도/HONGDOPARK

1994.10.05/경기 김포시

TEL.010-9134-3270

Email. gkgkghdeh21@naver.com

GRADUATION

2022 세민 정보고등학교

조리과학과 졸업

간단한 소개



생각해요

다듬어진 길이있어도 더좋은길이
있지않을까라라고 되물어요



질문해요

모르는것을 숨기지 않고 질문해요



발전해요

공부하고 내것으로 만들어요

목차

1. 보유스킬
2. 교육과정
3. 프로젝트
4. 후기 및 방향
5. 마치며

1. Skill

DB



Language



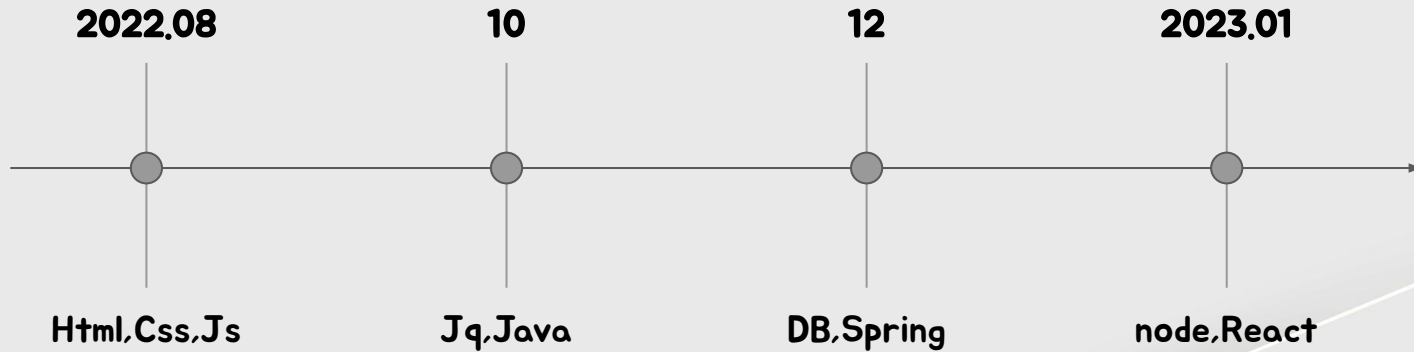
Tools



and



2. Curriculum



3. Project

- Spring Project
- Html Project

Spring Project

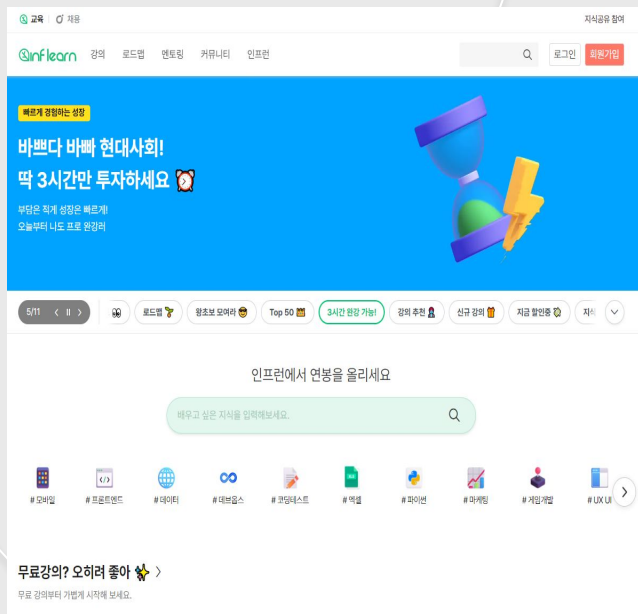
온라인 강의 사이트

**온라인 강의 관련 콘텐츠, 멘토링 서비스,
커머스가 결합된 홈페이지**

2022.12.4~2023.01.06

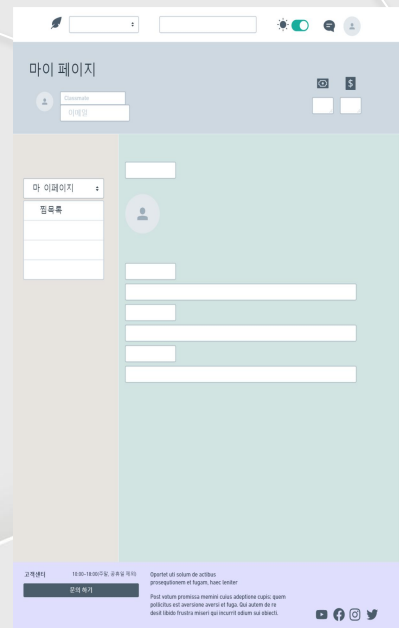
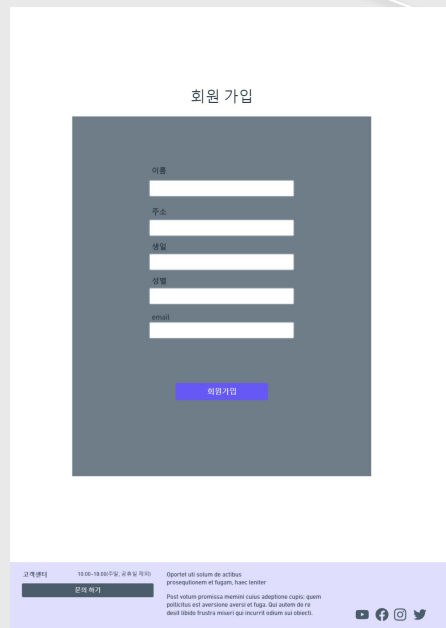
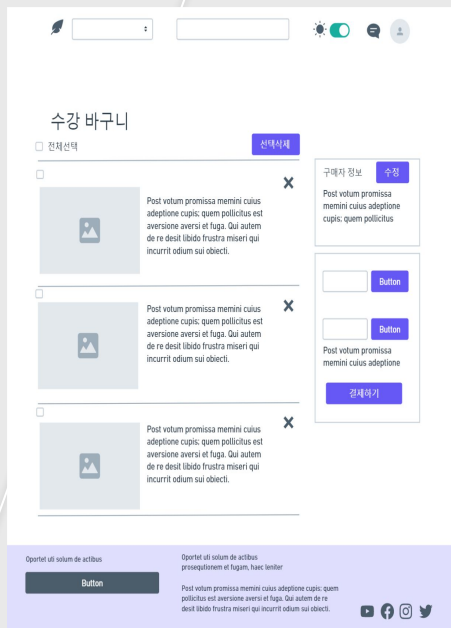
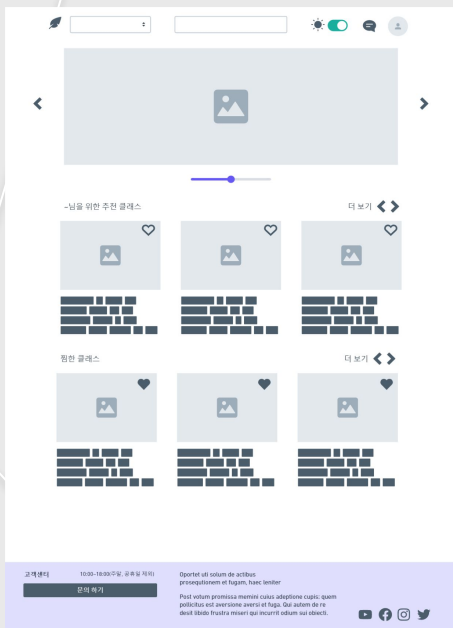
사이트 분석

참고사이트



1. 구현할 기능 탐색
2. 수익구조
3. 유저의 편의성 고려

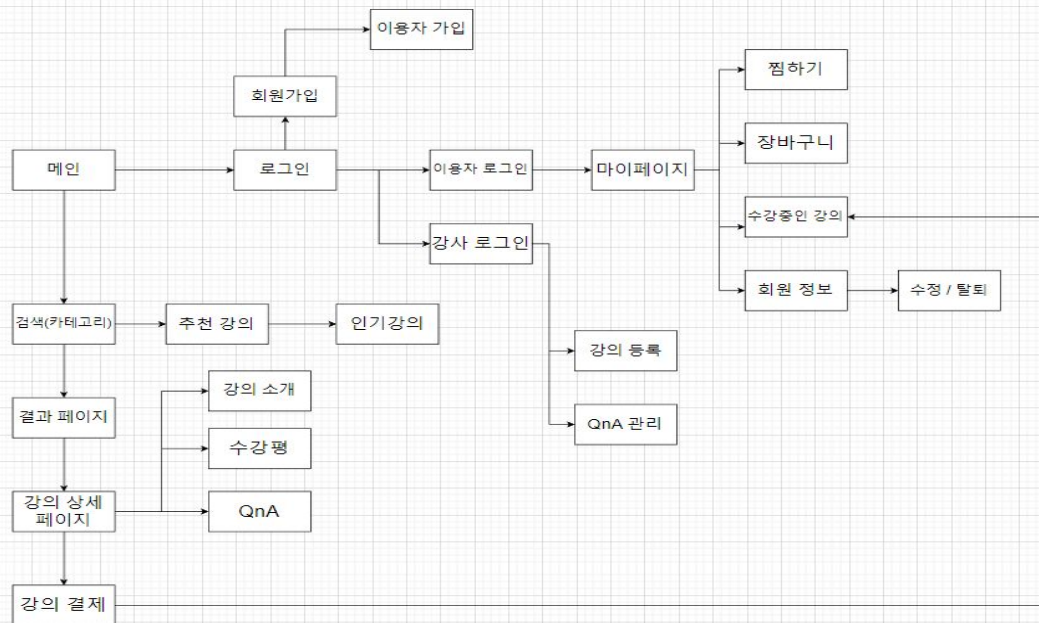
와이어 프레임 제작



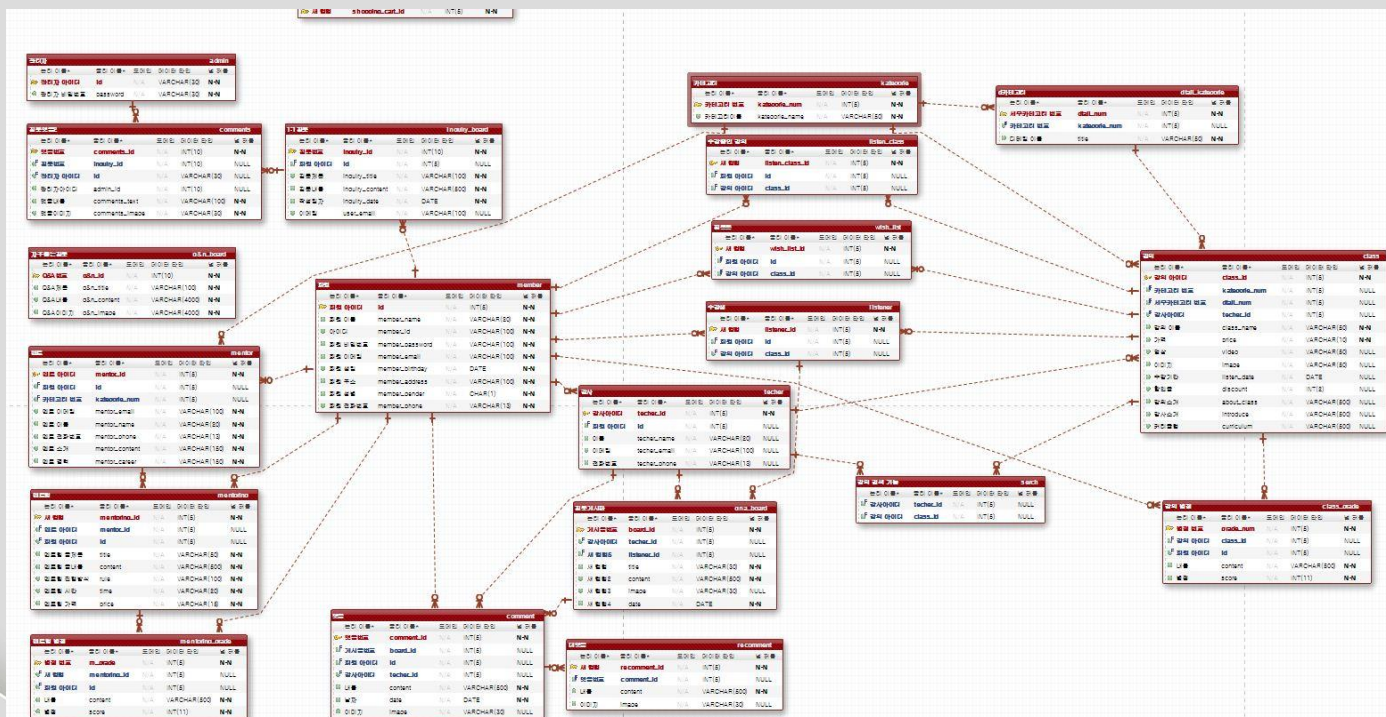
개발 환경 선정



플로우 차트








DB



주요기능(회원)

```
// 회원가입  
@GetMapping("/regist")  
public String regist(Member member, Model model) {  
    log.info("member >>>>>>>>>> " + member);  
    log.info("model >>>>>>>>>> " + model);  
  
    String encodedPassword = bCryptPasswordEncoder.encode(member.getPassword());  
    member.setPassword(encodedPassword);  
    memberService.registMember(member);  
    model.addAttribute("result", "success");  
    return "home";  
}
```

Result Grid		 Filter Rows: <input type="text"/>	Edit: 	Export/Import:  	Wrap Cell Contents: 		
id	name	password	email	birthday	address	gender	phone
abb	장복순	abb	abb@abb.com	2002-03-11	경기	F	010-1111-2222
abc	김가동	abc	abc@abc.com	2000-12-11	서울	M	010-0000-0000
acc	이복순	acc	acc@acc.com	2004-06-11	경기	F	010-2222-3333
add	최복순	add	add@add.com	2006-09-11	경기	M	010-4444-5555
ee	ee	\$as10\$wCwrl8d0tMGWfjdTo7DF.QLUskhn...	ee@ee	2023-01-04		M	
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
// 비밀번호찾기
@GetMapping("/findPassword")
public void findPassword() {
}

@PostMapping("/emailDuplication")
public @ResponseBody String emailDuplication(String email) {
    System.out.println(email);

    Member member = memberService.findByEmail(email);
    System.out.println(member);
    if (member != null) {
        return "yes";
    } else {
        return "no";
    }
}

@PostMapping("/findPassword")
public String findPassword(@RequestParam("email") String email, Model model) {
    MailDTO dto = memberService.createMailAndChangePassword(email);

    return "/member/loginForm";
}
```

Bcrypt 암호와 처리

분실시
가입이메일로
임시비밀번호 처리

주요기능(예외처리)

```
@Override
public boolean preHandle(HttpServletRequest request, HttpServletResponse response,
    Object handler) throws Exception {
    HttpSession session = request.getSession(false);
    if (session != null) {
        Object auth = session.getAttribute("auth");
        System.out.println("auth >>>>>>>>> " + auth);
        if (auth != null) {
            return true;
        }
    }
    response.sendRedirect(request.getContextPath() + "/member/loginForm");
    return false;
}
```

```
@Override
public void addInterceptors(InterceptorRegistry registry) {
    registry.addInterceptor(authCheckInterceptor())
        .addPathPatterns("/member/myPage")
        .addPathPatterns("/member/modify")
        .addPathPatterns("/mypage/*")
        ;
}
```

```
<!-- 로그인 - 회원가입 버튼들 -->
<c:if test="${empty auth}">
    <a role="button" class="button space-inset-4 signin"
        href="${pageContext.request.contextPath }/member/loginForm">로그인</a>
    <a class="e-signup button space-inset-4 is-primary "
        href="${pageContext.request.contextPath }/member/registForm">회원가입</a>
</c:if>
<c:if test="${!empty auth}">
    <!-- <span>${auth.name}님, 즐 하루!!!</span> --%>
    <a role="button" class="button space-inset-4 signin"
        href="${pageContext.request.contextPath }mypage/viewmypage " >마이페이지</a>
    <a class="e-signup button space-inset-4 is-primary "
        href="${pageContext.request.contextPath }/member/Logout">로그아웃</a>
</c:if>
</div>
```

예외처리

session.jsp
2중처리

주요기능(업로드,썸네일)

```
@Override
public void addResourceHandlers(ResourceHandlerRegistry registry) {
    registry.addResourceHandler("/resources/**").addResourceLocations("/resources/");
    //registry.addResourceHandler("/upload/**").addResourceLocations("/upload/");
    registry.addResourceHandler("/upload/**").addResourceLocations("file:///c:/upload/");
}

@RequestMapping("/thumbnails.do")
protected void thumbnails(@RequestParam("fileName") String fileName,
                           @RequestParam("lecture_id") String lecture_id,
                           HttpServletResponse response) throws Exception {
    // 소켓 클라와 서버에 데이터를 주고 받기
    OutputStream out = response.getOutputStream();
    // 파일 경로 파라미터로 받아서 설정
    String filePath=Curr_IMAGE_REPO_PATH+"\\\\"+lecture_id+"\\\\"+fileName;
    File image=new File(filePath);
    // 파일이 있다면 썸네일 만들어라
    if (image.exists()) {
        Thumbnails.of(image).size(1200,781).outputFormat("png").toOutputStream(out);
    }
    //버퍼 생성
    byte[] buffer = new byte[1024 * 8];
    out.write(buffer);
    out.close();
}


```

업로드파일 저장

파라미터로 받아온
값 썸네일 처리

주요기능(좋아요)

```
function like_func(id){  
  
    var lecture_id = id;  
    $.ajax({  
        url: '${contextPath }/mypage/like',  
        type: "GET",  
        data: 'lecture_id=' + lecture_id  
        ,  
        success: function(result) {  
            if(result == "like"){  
                document.getElementById("like_img"+lecture_id).src =  
                    '${contextPath }/resources/img/heart.png';  
            }else{  
                document.getElementById("like_img"+lecture_id).src =  
                    '${contextPath }/resources/img/love.png';  
            }  
        }  
    },  
  
    // 좋아요 추가 삭제  
    @GetMapping("/like")  
    public @ResponseBody String like(@RequestParam("lecture_id")  
    int lecture_id ,HttpSession session) {  
        System.out.println("lecture_id>>>" + lecture_id);  
        String id = ((Member) session.getAttribute("auth")).getId();  
        Map<String, Object> map = new HashMap<String , Object>();  
        map.put("lecture_id", lecture_id);  
        map.put("id", id);  
        int check = myPageService.checkLike(map);  
        System.out.println("check =====>" + check);  
  
        if(check == 1) {  
            int deletecheck = myPageService.deleteLike(map);  
            System.out.println("deletecheck>>>" + deletecheck);  
            return "hate";  
        }else{  
            int insertcheck = myPageService.insertLike(map);  
            System.out.println("insertcheck>>>" + insertcheck);  
            return "like";  
        }  
    }  
}
```

Ajax 활용 데이터
전달

DB select 후 if
처리

추후 구현 예정



결제페이지

결제 API 연결 상업페이지
실현



동영상 업로드

내가 보던 동영상 이어보기
업로딩



채팅 고객센터

게시판 방식이아닌
1:1챗봇 상담

프로젝트 후기

- 팀원과 협업의 중요성을 알수있는 계기가 되었다.
- 초기 설계 등이 미흡한점
(초기 구상 와이어프레임과 다른 결과물,DB설계,공통변수)
- 만들수록 기능적으로 필요한 부분이 눈에 보이고 못만든게 아쉽다.
- 실무에서의 협업 및 작업에 대한 궁금증

마무리

I Want to Be..

1. 언어나 문법을 단순 암기만 하는것이 아닌 이해하는 개발자
2. 오래된 기술 혹은 내가 할 줄 아는 것만 고집하는 않는 개발자
3. 퇴근 시간이 아닌 내가 맡은 미션의 끝일 기다리는 개발자
4. 내가 속한 부분이 아닌 전체를 생각하고 이해하는 개발자
5. 급변하는 세상과 기술에 대한 대응이 항상 준비되어 있는 개발자

이런 개발자가 되도록 노력할 것 입니다

감사합니다!
열심히 하겠습니다!

-by 신입 개발자 박홍도

2022~2023 PORTFOLIO

CONTACT

gkgkghdeh21@naver.com

010 9134 3270