

산학연계프로젝트 멘토링보고서

※ 멘토링 실시 후 팀별로 제출바랍니다.

(1page)

| 과 제 수 행 원 현 황 | | | | | |
|---------------|--|--------------|---------|-------------|---------------------|
| 수행 학기 | <input type="checkbox"/> 2023년07월~2023년08월(여름학기) | | | | |
| 프로젝트명 | 글로벌 물류 O2O 건적 웹플랫폼 개발 | | | | |
| 팀명 | 햄빠 (HaPPPPy) | | | | |
| | 학과 | 학번 | 성명 | 연락처 | E-mail |
| 팀장 | 정보통신공학과 | 2020112140 | 윤영서 | 01053864057 | dbsys1120@naver.com |
| 팀원 | 정보통신공학과 | 2020112107 | 김태연 | 01089031349 | xodus1349@naver.com |
| | 정보통신공학과 | 2020112095 | 안정민 | 01045247651 | 102chimee@gmail.com |
| | 정보통신공학과 | 2020112096 | 전현정 | 01033978358 | j10123@naver.com |
| | | | | | |
| 교과목 명 | 산학연계프로젝트 | | | | |
| 산업체 멘토 | 기업명 | 모바일 앱개발 협동조합 | | | |
| | 멘토직위 | 대표 | 멘토이름/서명 | 최원석 (서명) | |

| | |
|-----------|---------|
| 멘 토 링 내 용 | (2page) |
|-----------|---------|

| | |
|--------|--|
| 멘토링 일시 | 2023년 7월 29일 17 : 00 ~ 19 : 00 (총 2시간) |
|--------|--|

| | |
|--------|------|
| 멘토링 장소 | 오프라인 |
|--------|------|

| | |
|--|-----------------------------|
| | * 멘토가 지도한 내용 (보완 및 추가 요청사항) |
|--|-----------------------------|

- 유사 프로젝트를 통한 기술 스택 멘토링

● 프론트

| | |
|-----------------|-----------------|
| <p>Figure 4</p> | <p>Figure 5</p> |
|-----------------|-----------------|



| | | |
|--------|--|---|
| 멘토링 내용 | <div data-bbox="371 736 873 880"> <p>Register.js - checkid</p> <div data-bbox="402 665 545 846"> <p>1. 아이디 그룹</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <div data-bbox="576 672 825 831"> <p>1. Radio Group</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <p>관리자와 사용자를 구분하는 버튼 관리자 정보와 사용자 정보 입력 on/off 기능</p> <p>아이디의 중복을 체크하는 버튼 유효성 감시를 위한 아이디 중복 검사</p> <p>카카오, 다음 등에서 사용하는 우편번호 API</p> <p>입력된 사용자 또는 관리자 정보를 Admin DB에 전달하기 위한 버튼</p> </div> | <div data-bbox="880 736 1383 880"> <p>Register.js - checkid</p> <div data-bbox="900 665 1043 846"> <p>1. 아이디 그룹</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <div data-bbox="1074 672 1323 831"> <p>1. Radio Group</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <p>관리자와 사용자를 구분하는 버튼 관리자 정보와 사용자 정보 입력 on/off 기능</p> <p>아이디의 중복을 체크하는 버튼 유효성 감시를 위한 아이디 중복 검사</p> <p>카카오, 다음 등에서 사용하는 우편번호 API</p> <p>입력된 사용자 또는 관리자 정보를 Admin DB에 전달하기 위한 버튼</p> </div> |
| 멘토링 내용 | <div data-bbox="371 880 873 1149"> <p>Register.js - DaumPostCode</p> <div data-bbox="402 943 545 1077"> <p>1. 아이디 그룹</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <div data-bbox="576 949 825 1077"> <p>1. Radio Group</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <p>관리자와 사용자를 구분하는 버튼 관리자 정보와 사용자 정보 입력 on/off 기능</p> <p>아이디의 중복을 체크하는 버튼 유효성 감시를 위한 아이디 중복 검사</p> <p>카카오, 다음 등에서 사용하는 우편번호 API</p> <p>입력된 사용자 또는 관리자 정보를 Admin DB에 전달하기 위한 버튼</p> </div> | <div data-bbox="880 880 1383 1149"> <p>Register.js - registerUser - 실행결과</p> <div data-bbox="900 943 1043 1077"> <p>1. 아이디 그룹</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <div data-bbox="1074 949 1323 1077"> <p>1. Radio Group</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <p>관리자와 사용자를 구분하는 버튼 관리자 정보와 사용자 정보 입력 on/off 기능</p> <p>아이디의 중복을 체크하는 버튼 유효성 감시를 위한 아이디 중복 검사</p> <p>카카오, 다음 등에서 사용하는 우편번호 API</p> <p>입력된 사용자 또는 관리자 정보를 Admin DB에 전달하기 위한 버튼</p> </div> |
| 멘토링 내용 | <div data-bbox="371 1149 873 1323"> <p>LoginPage2.js</p> <div data-bbox="402 1236 545 1348"> <p>1. 아이디 그룹</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <div data-bbox="576 1245 825 1348"> <p>1. Radio Group</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <p>관리자와 사용자를 구분하는 버튼 관리자 정보와 사용자 정보 입력 on/off 기능</p> <p>아이디의 중복을 체크하는 버튼 유효성 감시를 위한 아이디 중복 검사</p> <p>카카오, 다음 등에서 사용하는 우편번호 API</p> <p>입력된 사용자 또는 관리자 정보를 Admin DB에 전달하기 위한 버튼</p> </div> | <div data-bbox="880 1149 1383 1323"> <p>LoginPage2.js</p> <div data-bbox="900 1236 1043 1348"> <p>1. 아이디 그룹</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <div data-bbox="1074 1245 1323 1348"> <p>1. Radio Group</p> <p>2. 아이디 그룹</p> <p>3. 아이디 그룹</p> <p>4. 아이디 그룹</p> <p>5. 아이디 그룹</p> </div> <p>관리자와 사용자를 구분하는 버튼 관리자 정보와 사용자 정보 입력 on/off 기능</p> <p>아이디의 중복을 체크하는 버튼 유효성 감시를 위한 아이디 중복 검사</p> <p>카카오, 다음 등에서 사용하는 우편번호 API</p> <p>입력된 사용자 또는 관리자 정보를 Admin DB에 전달하기 위한 버튼</p> </div> |

● 백엔드

The diagram illustrates the package structure of a Spring Boot & MySQL application. It is divided into two main sections: 'package' and 'class'.

package section: This section shows a tree view of the project structure. The root package is 'com.example.springboot'. It contains sub-packages for 'controller', 'service', 'repository', 'entity', 'dto', 'exception', 'config', 'security', 'utils', and 'mysql'. The 'mysql' package is further divided into 'mysql' and 'mysql.entity'.

class section: This section lists the classes and their package names. The classes are organized into groups based on their package structure. The 'com.example.springboot' package contains 'Application' and 'ApplicationTest'. The 'com.example.springboot.controller' package contains 'AdminController', 'AuthController', 'UserController', and 'WebMvcConfig'. The 'com.example.springboot.service' package contains 'AuthService', 'UserService', and 'WebMvcConfig'. The 'com.example.springboot.repository' package contains 'AuthServiceRepository', 'UserRepository', and 'WebMvcConfig'. The 'com.example.springboot.entity' package contains 'AuthServiceEntity', 'UserEntity', and 'WebMvcConfig'. The 'com.example.springboot.mysql' package contains 'AuthServiceRepository', 'UserRepository', and 'WebMvcConfig'. The 'com.example.springboot.mysql.entity' package contains 'AuthServiceEntity', 'UserEntity', and 'WebMvcConfig'.

| | | | | | | |
|---------------------|---|-----|-----|---|--|--|
| | <div> <div> <div>Part 4</div> <div>Spring Boot & MySQL</div> </div> <div> <div>REST API</div> <div> <div>- CRUD Operation # Create(POST), Read(GET), Update(PUT) 구현 완료</div> <div> <div> <pre> 1 // Create a new book 2 @PostMapping("/books") 3 public ResponseEntity<Book> createBook(@RequestBody Book book) { 4 return ResponseEntity.status(HttpStatus.CREATED).body(bookRepository.save(book)); 5 } </pre> </div> <div> <pre> 1 // Read all books 2 @GetMapping("/books") 3 public ResponseEntity<List<Book>> getAllBooks() { 4 return ResponseEntity.ok(bookRepository.findAll()); 5 } </pre> </div> </div> </div> <div> <div>- POST : 클라이언트는 서버에 데이터 전송</div> <div> <div> <pre> 1 // Update a book 2 @PutMapping("/books/{id}") 3 public ResponseEntity<Book> updateBook(@PathVariable Long id, @RequestBody Book book) { 4 return ResponseEntity.ok(bookRepository.findById(id).map(b -> b.update(book)).orElseThrow()); 5 } </pre> </div> <div> <pre> 1 // Delete a book 2 @DeleteMapping("/books/{id}") 3 public ResponseEntity<Book> deleteBook(@PathVariable Long id) { 4 return ResponseEntity.ok(bookRepository.findById(id).map(b -> b.delete()).orElseThrow()); 5 } </pre> </div> </div> </div> <div> <div>- PUT : 클라이언트가 서버의 기존 리소스를 업데이트</div> <div>- GET : 클라이언트가 서버의 리소스에 액세스</div> </div> </div> </div> | | | | | |
| 참석자 (이름기입) | 최원서 | 윤영서 | 김태연 | | | |
| | 멘토님 포함 총 (3) 명 참석 | | | | | |
| 사진 증빙첨부 2장 첨부 |  | | |  | | |