LX 공간정보 아카데미 12기

Java & SQL 프로젝트

6조 스켈레톤

고영준, 김준하, 박민호

프로젝트 발표 순서

01 프로젝트 개요

02 프로젝트 주요 기능

03 팀원 분담

04 개발 일정

05 개발 원칙

06 프로그램 구조도

07 프로그램 시연

08 배운 점, 소감

09 Q&A

프로젝트 개요

프로젝트 명	Java & SQL 프로젝트
개발 일정	07.17 ~ 07.21
특징	JDBC를 이용한 데이터베이스와 Java의 결합

프로젝트 주요 기능

검색 기능	여러 매개변수로 직원 정보 조회, 검색 기능	다이얼 시스템 활용 서비스 선택 기능
통계 출력 기능	여러 기준의 연봉 통계정보 출력 및 정렬 기능	다이얼 시스템 활용 서비스 선택 기능

팀원 분담

Java

고영준

MenuController class
DBMethod class
CrateList class

김준하

Custom Excetprion class
DBMethod class
통계자료 SQL 쿼리문 작성

SQL

박민호

DBMethod class
EMP 검색 SQL 쿼리문 작성
통계자료 SQL 쿼리문 작성

개발 일정

6월 17일 ~ 21일

17일	18일	19일	20일	21일

분석/설계

세부 기능 개발 및 단위 테스트

기능 통합 및 최적화

전체 시스템 테스트 및 피드백

개발 원칙

1. 명확한 클래스 기능 분담

2. 일관된 코딩 컨벤션: camelCase 표기법

3. 안정성 확보: 예외 처리 설계

4. 코드 간결화: 공통 기능 모듈화

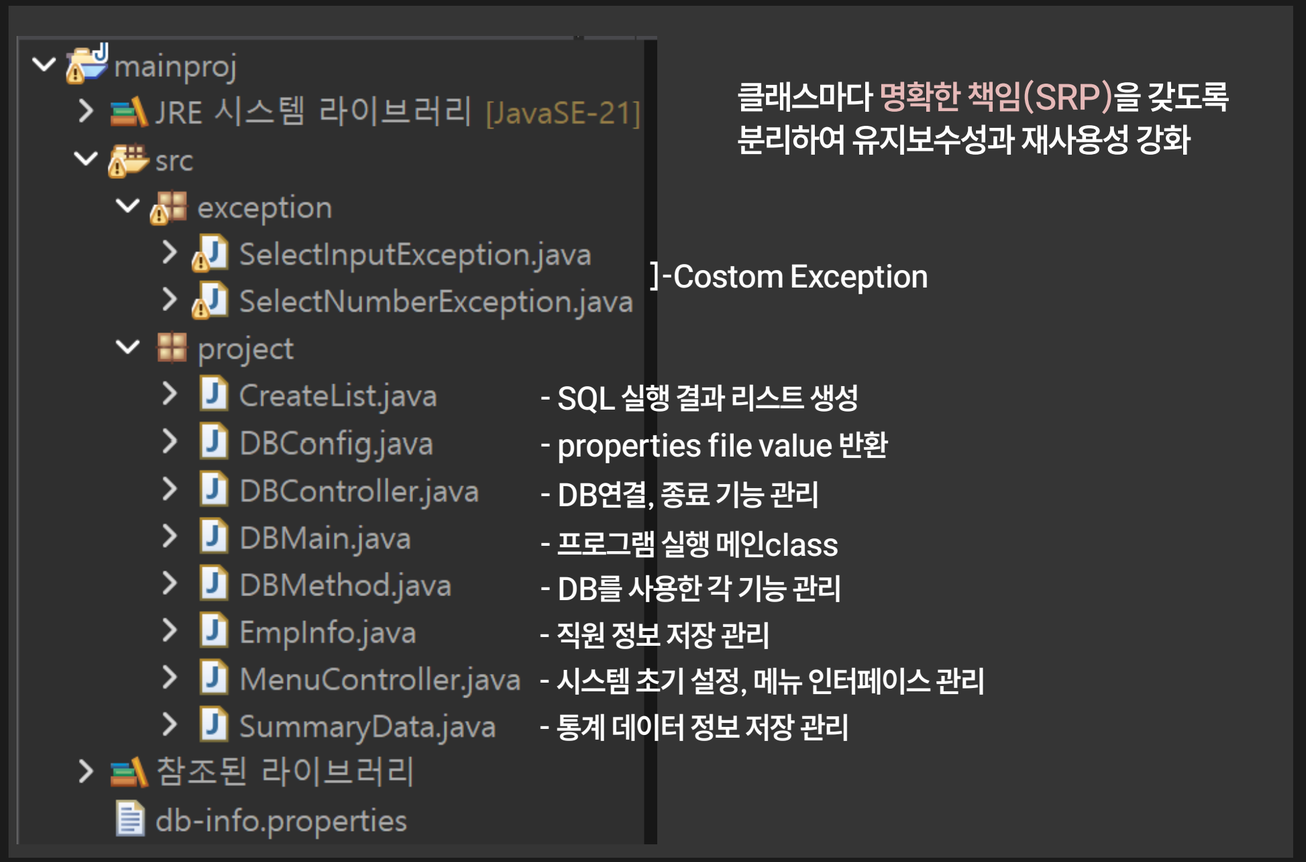
개발 원칙

1. 담당 기능에 따른 클래스 분리

2. 일관된 코딩 컨벤션: camelCase 표기법

3. 시스템 안정성 확보: 예외 처리 설계

4. 코드 간결화: 공통 기능 모듈화



개발 원칙

1. 담당 기능에 따른 클래스 분리

2. 일관된 코딩 컨벤션: camelCase 표기법

3. 시스템 안정성 확보: 예외 처리 설계

4. 코드 간결화: 공통 기능 모듈화

- 변수명 및 메서드명 모두 camelCase 적용
  ex) ‘findEmpInfoByEmpName()’, ‘executeQuery()’, ‘hireDate’’

-변수명은 해당 변수의 역할을 명확히 표현하고, 메서드명은 동작이나 목적이 드러나도록 작성

- camelCase를 사용하면 코드의 가독성이 높아지고, 협업 시 명명 규칙으로 인한 혼선을 줄일 수 있음

개발 원칙

1. 담당 기능에 따른 클래스 분리

2. 일관된 코딩 컨벤션: camelCase 표기법

3. 시스템 안정성 확보: 예외 처리 설계

4. 코드 간결화: 공통 기능 모듈화

- 주요 기능에 try-catch 구문을 적용하여 프로그램의 비정상 종료를 최소화하고, 해당 오류에 대한 원인을 화면에 출력

- 예외 발생 시 사용자 입장에서 오류 원인을 직관적으로 파악할 수 있도록 하여, 입력 실수나 시스템 문제 해결이 쉽게 이루어지도록 설계

- costom exception을 정의하여 오류 원인 추적을 더욱 용이하게 하고, 시스템의 안정성과 유지보수성을 높이고자 함.

개발 원칙

1. 담당 기능에 따른 클래스 분리

2. 일관된 코딩 컨벤션: camelCase 표기법

3. 시스템 안정성 확보: 예외 처리 설계

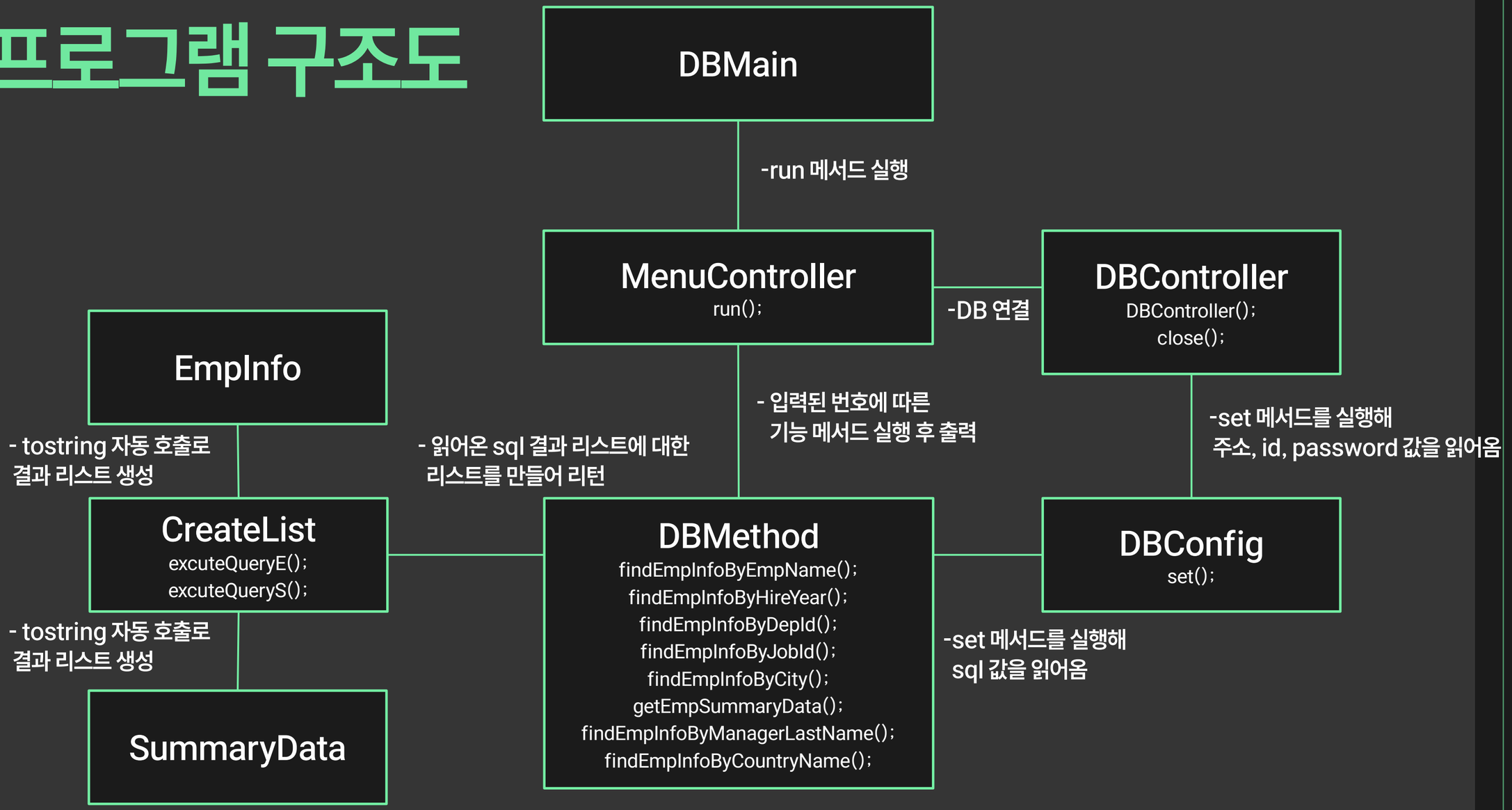
4. 코드 간결화: 공통 기능 모듈화

- 전체적으로 기능별 클래스를 분리하고 중복 코드를 메서드화함으로써, 코드 가독성과 유지보수 효율성을 높였음

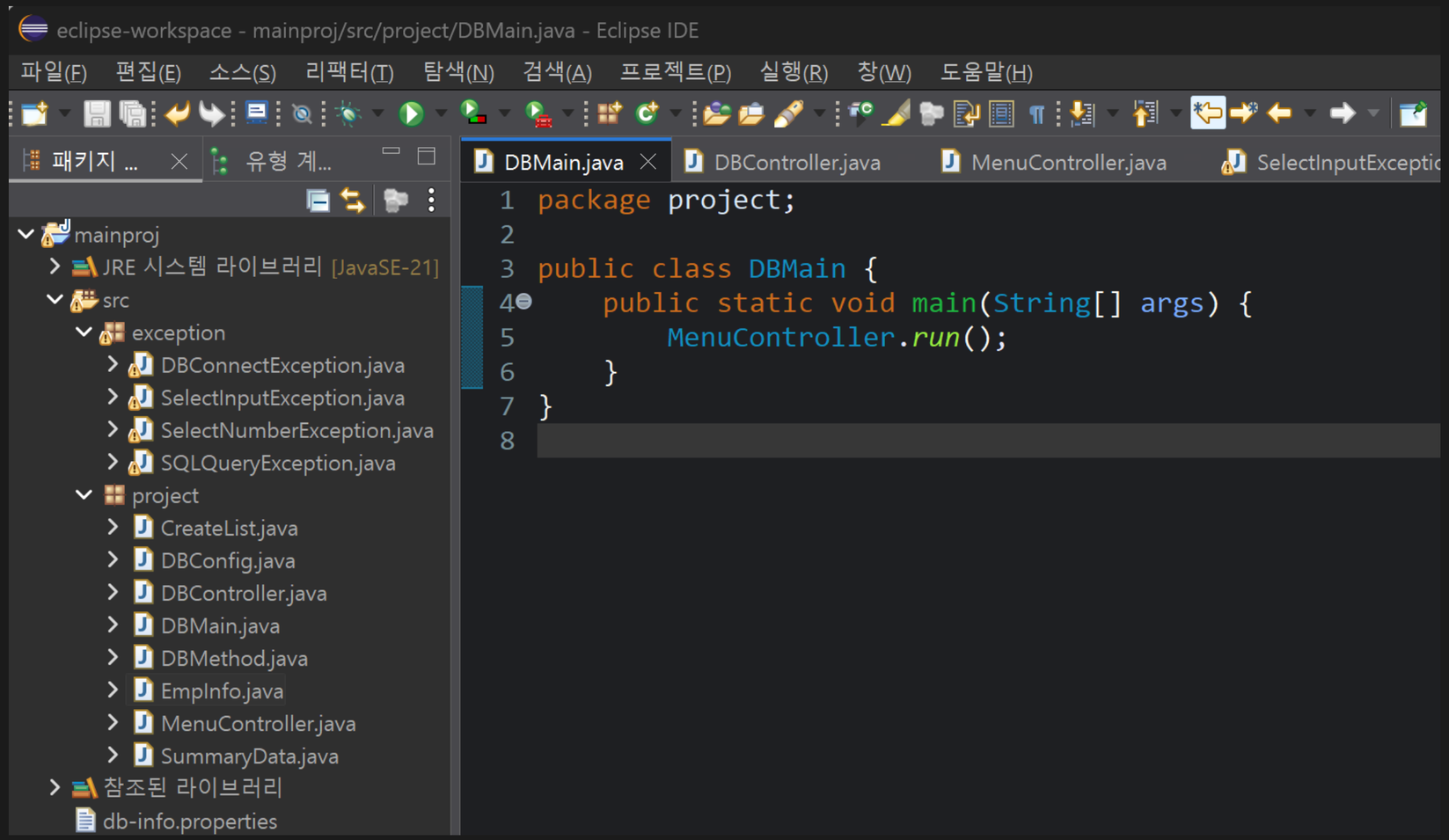
- 출력 메시지 및 조건문 처리도 하나의 패턴으로 구성함으로써, 각 기능 메서드마다 일관된 결과 처리가 가능하도록 구성
ex) SQL 쿼리 실행 및 결과 리스트 생성 기능을 메서드화

```
public class CreateList {

    static public List<EmpInfo> executeQueryE(String sql, Object params) {// int와 String을 모두 받음
        List<EmpInfo> empList = new ArrayList<>();
        try (PreparedStatement stmt = DBController.conn.prepareStatement(sql)) {
            stmt.setObject(1, params);
            ResultSet rs = stmt.executeQuery();
                // while문으로 직원정보를 리스트에 담음 -> empinfo클래스에 변수가 정의되어 있음
            while(rs.next()) {
                EmpInfo emp = new EmpInfo();
                emp.employeeId = rs.getInt("employee_id");
                emp.lastName = rs.getString("last_name");
                emp.salary = rs.getInt("salary");
                emp.email = rs.getString("email");
                emp.hireDate = rs.getString("hire_date");
                emp.firstName = rs.getString("first_name");
                emp.phoneNumber = rs.getString("phone_number");
                emp.jobId = rs.getString("job_id");
                emp.managerId = rs.getString("manager_id");
                emp.departmentId = rs.getString("department_id");
                emp.commissionPct = rs.getDouble("commission_pct");

                empList.add(emp);
            }
        } catch (SQLException e) { // 쿼리 실행에 오류 발생 시 실패 메세지 전송
            System.out.println(new SQLQueryException().getMessage());
            MenuController.run();
        }
        return empList;
        }
```

프로그램 구조도


DBMain
-run 메서드 실행
MenuController
run();
-DB 연결
DBController
DBController();
close();
EmpInfo
- 입력된 번호에 따른
기능 메서드 실행 후 출력
-set 메서드를 실행해
주소, id, password 값을 읽어옴
- tostring 자동 호출로
결과 리스트 생성
- 읽어온 sql 결과 리스트에 대한
리스트를 만들어 리턴
CreateList
excuteQueryE();
excuteQueryS();
DBMethod
findEmpInfoByEmpName();
findEmpInfoByHireYear();
findEmpInfoByDepId();
findEmpInfoByJobId();
findEmpInfoByCity();
getEmpSummaryData();
findEmpInfoByManagerLastName();
findEmpInfoByCountryName();
DBConfig
set();
- tostring 자동 호출로
결과 리스트 생성
-set 메서드를 실행해
sql 값을 읽어옴
SummaryData

프로그램 시연


eclipse-workspace - mainproj/src/project/DBMain.java - Eclipse IDE
파일(F) 편집(E) 소스(S) 리팩터(T) 탐색(N) 검색(A) 프로젝트(P) 실행(R) 창(W) 도움말(H)
패키지 ...
유형 계...
mainproj
JRE 시스템 라이브러리 [JavaSE-21]
src
exception
DBConnectException.java
SelectInputException.java
SelectNumberException.java
SQLQueryException.java
project
CreateList.java
DBConfig.java
DBController.java
DBMain.java
DBMethod.java
EmpInfo.java
MenuController.java
SummaryData.java
참조된 라이브러리
db-info.properties
DBMain.java
DBController.java
MenuController.java
SelectInputExceptio
1 package project;
2
3 public class DBMain {
4 public static void main(String[] args) {
5 MenuController.run();
6 }
7 }
8

프로젝트에서 배운 점

배운점

1. 수정사항 기록
- 코드를 수정하고 나서 이걸 왜 넣었는지 명시하지 않으면 나중에 볼 때 모른다. 이게 또 문법오류가 안나는 코드를 넣었다고 하면 나중에 보면서 지워버릴 수 있기 때문에 수정사항이 생기면 그걸 꼭 명시해줘야 한다.

2. 진행상황 공유
- 협업을 하면 본인의 진행상황을 공유해줘야 한다. 서로의 진행상황을 알고 있지 않다면, 이른바 “바퀴를 재발명하지 마라"라는 말에서 서로 같은 바퀴만을 만들고 있을 수 있다.

3. 프로젝트 계획 중요시하기
- 처음에 계획했던 기능들을 구현하는 것은 순조롭게 진행될 수 있다. 하지만 구현 도중에 기능을 추가하고자 하면 그때부터 머리가 아파진다. 처음에 구현 목표와 계획을 탄탄하게 짜야 더 좋은 프로그램을 만들 수 있다.

프로젝트 소감

고영준

자바와 SQL을 배우고 첫 팀 프로젝트를 해봤는데 재미있기도 했고, 한편으로는 지금 저의 자바실력에 한계를 느낀 시간이었다고 생각합니다. 시간이 좀 더 많았다면 또 아는게 더 많았다면 더 구현해보고 싶은데 라는 욕심이 들었습니다. 이런 욕심을 가지고 앞으로의 프로젝트에서는 제가 구현해 보고 싶은 기능을 다 구현할 수 있을 때까지 열심히 노력하고 배우겠습니다. 감사합니다.

김준하

Java를 3주동안 배우면서 개념적인 부분도 잘 이해가 되고, 과제는 응용을 더해 해볼 수 있었습니다. 그 덕에 자신감을 얻고 이번 프로젝트에서 팀장을 맡게 되었지만, 이번 프로젝트를 시작하면서 프로그래밍 공부의 부족함을 절실히 깨닫고 배우는 자세로 임하게 되었습니다. 이를 통해 이후의 것들을 학습할 때에도 어느정도 선에서 만족하고 정체되는 것이 아닌, “배움에는 끝이 없다"라는 마인드를 가지고 하나라도 더 학습하고 제 것으로 만들 수 있도록 노력하겠습니다. 감사합니다.

박민호

이번 JDBC 프로젝트를 통해 실무에서 활용 가능한 역량을 키울 수 있었고, 코딩 이해도가 높은 팀원들과 함께하며 많은 것을 배웠습니다. 팀 프로젝트에서는 꼼꼼함과 세심함이 중요하다는 것을 깨달았습니다. 예를 들면 합의된 변수명을 지키지 않거나 공동 모듈을 수정하는 등의 작은 실수가 팀 전체에 영향을 줄 수 있다는 점에서 항상 신중하게 임해야겠다고 느꼈습니다.

Q&A

자유롭게 질문해주세요!