

개발 완료 보고서

제출일 : 2023. 02. 23

| | | | |
|-----------|--|--------------------------|------------------|
| 팀명 | | | |
| 참여인원 | 이수성 | | |
| 개발프로젝트 소개 | | | |
| 프로젝트 명 | 파이썬과 Mysql을 이용한 퀴즈프로그램 | | |
| 활동일시 | 2023. 02. 20 ~ 02. 23 | 장소 | 광주인력개발원 기초설계실 |
| 개발목적 | 파이썬 및 SQL 배운내용 복습 | | |
| 파트별 임무분담 | 이수성 | - DB 제작 및 관리 - 파이썬 코딩 | |
| 개발환경 | Window OS, Visual Studio Code, python, Mysql(HeidiSQL), Github | | |

필수

실습실 퇴근 시간 : 매일 09시 ~ 18시 00분

학습 일정표

| 분류 | 항목 | | 2-20 | 2-21 | 2-22 | 2-23 |
|----|----|-----------|------|------|------|------|
| 학습 | 구현 | 데이터베이스 설정 | | | | |
| | | 매뉴 | 코딩 ! | | | |
| | | 회원관리 | | | | |
| | | 퀴즈 | | | | |
| | | 랭킹 | | | | |

개인 일정

| | |
|-----|----------|
| 이수성 | 특이 사항 없음 |
|-----|----------|

일정표

| 요구분석서 | <div>요구사항 분석서</div> <table><tr><th>번호</th><th>유형</th><th>요약</th><th>요구사항</th><th>요구분석 내용</th></tr><tr><td>1</td><td>데이터베이스</td><td>퀴즈 및 회원의 정보 저장</td><td>퀴즈 및 회원의 정보 저장 파라미터를 활용해 DB 수정 및 추가</td><td>1. mysql에 quiz테이블 및 user 테이블 생성 2. 퀴즈 데이터 및 회원 데이터 입력 3. mysql 파라미터에 연결</td></tr><tr><td>2</td><td>회원관리</td><td>회원가입, 로그인, 로그아웃 기능 구현</td><td>회원가입 로그인 로그아웃</td><td>1. 아이디와 비밀번호를 입력받는다 2. 아이디 또는 비밀번호가 정해진 형식에 맞지 않으면 오류문을 출력하고 메뉴로 돌아간다 3. 중복된 아이디가 있는지 확인한다 4. 아이디와 비밀번호가 형식에 맞으면 데이터베이스에 저장하고 메뉴로 돌아간다 1. 로그인할 한 상황인지 확인한다 2. 아이디를 입력받고 데이터베이스에 일치하는 아이디가 있는지 확인한다 3. 비밀번호를 입력받고 데이터베이스에 입력된 아이디와 매칭되는 비밀번호를 확인한다 1. 로그아웃 한 상황인지 확인한다 2. 로그아웃을 하고 메뉴로 돌아간다</td></tr><tr><td>3</td><td>퀴즈</td><td>퀴즈 추가, 퀴즈 풀기</td><td>퀴즈 추가 퀴즈 풀기(시간제한 : 60초)</td><td>1. 퀴즈의 질문과 답을 입력받는다 2. 입력받은 내용을 데이터베이스에 저장한다. 1. 로그인할 한 상황인지 확인한다 2. 퀴즈를 하나 보여준다 3. 퀴즈의 답을 입력받고 정답의 유무와 남은 시간을 보여준다 4. 60초 동안 2-3을 반복한다 5. 모은 정답의 갯수와 게임시간을 데이터베이스에 누적한다 1. 데이터베이스에서 user정보를 가져온다 2. 점수를 기준으로 오름차순으로 정렬한다 3. 5위까지를 출력한다</td></tr><tr><td>4</td><td>랭킹</td><td>아이디, 점수, 플레이타임 출력 및 나의 정보 확인</td><td>점수를 기준으로 오름차순 정렬 내 정보 확인</td><td>1. 로그인할 한 상황인지 확인한다 2. 데이터베이스에 저장된 아이디, 점수, 게임시간을 보여준다</td></tr></table> | 번호 | 유형 | 요약 | 요구사항 | 요구분석 내용 | 1 | 데이터베이스 | 퀴즈 및 회원의 정보 저장 | 퀴즈 및 회원의 정보 저장 파라미터를 활용해 DB 수정 및 추가 | 1. mysql에 quiz테이블 및 user 테이블 생성 2. 퀴즈 데이터 및 회원 데이터 입력 3. mysql 파라미터에 연결 | 2 | 회원관리 | 회원가입, 로그인, 로그아웃 기능 구현 | 회원가입 로그인 로그아웃 | 1. 아이디와 비밀번호를 입력받는다 2. 아이디 또는 비밀번호가 정해진 형식에 맞지 않으면 오류문을 출력하고 메뉴로 돌아간다 3. 중복된 아이디가 있는지 확인한다 4. 아이디와 비밀번호가 형식에 맞으면 데이터베이스에 저장하고 메뉴로 돌아간다 1. 로그인할 한 상황인지 확인한다 2. 아이디를 입력받고 데이터베이스에 일치하는 아이디가 있는지 확인한다 3. 비밀번호를 입력받고 데이터베이스에 입력된 아이디와 매칭되는 비밀번호를 확인한다 1. 로그아웃 한 상황인지 확인한다 2. 로그아웃을 하고 메뉴로 돌아간다 | 3 | 퀴즈 | 퀴즈 추가, 퀴즈 풀기 | 퀴즈 추가 퀴즈 풀기(시간제한 : 60초) | 1. 퀴즈의 질문과 답을 입력받는다 2. 입력받은 내용을 데이터베이스에 저장한다. 1. 로그인할 한 상황인지 확인한다 2. 퀴즈를 하나 보여준다 3. 퀴즈의 답을 입력받고 정답의 유무와 남은 시간을 보여준다 4. 60초 동안 2-3을 반복한다 5. 모은 정답의 갯수와 게임시간을 데이터베이스에 누적한다 1. 데이터베이스에서 user정보를 가져온다 2. 점수를 기준으로 오름차순으로 정렬한다 3. 5위까지를 출력한다 | 4 | 랭킹 | 아이디, 점수, 플레이타임 출력 및 나의 정보 확인 | 점수를 기준으로 오름차순 정렬 내 정보 확인 | 1. 로그인할 한 상황인지 확인한다 2. 데이터베이스에 저장된 아이디, 점수, 게임시간을 보여준다 |
|-------|--|---------------------------------|---|--|------|---------|---|--------|----------------|--|---|---|------|-----------------------|---|--|---|----|--------------|--|--|---|----|---------------------------------|---|---|
| 번호 | 유형 | 요약 | 요구사항 | 요구분석 내용 | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 데이터베이스 | 퀴즈 및 회원의 정보 저장 | 퀴즈 및 회원의 정보 저장 파라미터를 활용해 DB 수정 및 추가 | 1. mysql에 quiz테이블 및 user 테이블 생성 2. 퀴즈 데이터 및 회원 데이터 입력 3. mysql 파라미터에 연결 | | | | | | | | | | | | | | | | | | | | | | |
| 2 | 회원관리 | 회원가입, 로그인, 로그아웃 기능 구현 | 회원가입 로그인 로그아웃 | 1. 아이디와 비밀번호를 입력받는다 2. 아이디 또는 비밀번호가 정해진 형식에 맞지 않으면 오류문을 출력하고 메뉴로 돌아간다 3. 중복된 아이디가 있는지 확인한다 4. 아이디와 비밀번호가 형식에 맞으면 데이터베이스에 저장하고 메뉴로 돌아간다 1. 로그인할 한 상황인지 확인한다 2. 아이디를 입력받고 데이터베이스에 일치하는 아이디가 있는지 확인한다 3. 비밀번호를 입력받고 데이터베이스에 입력된 아이디와 매칭되는 비밀번호를 확인한다 1. 로그아웃 한 상황인지 확인한다 2. 로그아웃을 하고 메뉴로 돌아간다 | | | | | | | | | | | | | | | | | | | | | | |
| 3 | 퀴즈 | 퀴즈 추가, 퀴즈 풀기 | 퀴즈 추가 퀴즈 풀기(시간제한 : 60초) | 1. 퀴즈의 질문과 답을 입력받는다 2. 입력받은 내용을 데이터베이스에 저장한다. 1. 로그인할 한 상황인지 확인한다 2. 퀴즈를 하나 보여준다 3. 퀴즈의 답을 입력받고 정답의 유무와 남은 시간을 보여준다 4. 60초 동안 2-3을 반복한다 5. 모은 정답의 갯수와 게임시간을 데이터베이스에 누적한다 1. 데이터베이스에서 user정보를 가져온다 2. 점수를 기준으로 오름차순으로 정렬한다 3. 5위까지를 출력한다 | | | | | | | | | | | | | | | | | | | | | | |
| 4 | 랭킹 | 아이디, 점수, 플레이타임 출력 및 나의 정보 확인 | 점수를 기준으로 오름차순 정렬 내 정보 확인 | 1. 로그인할 한 상황인지 확인한다 2. 데이터베이스에 저장된 아이디, 점수, 게임시간을 보여준다 | | | | | | | | | | | | | | | | | | | | | | |
| DB | <div>quiz</div> <div>quiz_number Auto_increment quiz_name quiz_answer</div> <div>user</div> <div>user_id Unique INDEX user_score user_pw playTime</div> | | | | | | | | | | | | | | | | | | | | | | | | | |
| 코드설명 | <div>sing_up()</div> <div>- 아이디와 비밀번호를 입력받아 user 테이블에 저장한다.</div> <div>def login()</div> <div>- 로그인 상태 여부를 확인한 후 로그인이 되어있지 않으면 DB에 저장된 아이디와 비밀번호가 일치한지 확인후 로그인한다.</div> <div>def logout()</div> <div>- 로그인 상태 여부를 확인 후 로그인이 되어 있으면 로그아웃 한다.</div> <div>def update_quiz()</div> <div>- 퀴즈의 내용과 정답을 입력받아 quiz 테이블에 저장한다.</div> | | | | | | | | | | | | | | | | | | | | | | | | | |

| | |
|------|--|
| | <pre>def play()</pre> <ul style="list-style-type: none"> - 로그인이 되어있으면 DB에 저장된 퀴즈를 랜덤으로 불러오고 정답을 입력한다. 정답이 일치하면 점수를 1씩 올려준다. 총 제한시간은 60초이고 매 정답을 입력할때마다 남은시간을 출력해준다. <pre>def update_score(score, time, id)</pre> <ul style="list-style-type: none"> - play함수를 실행하면 생기는 플레이타임과 점수를 기존 데이터베이스에 누적해 저장한다. <pre>def rank()</pre> <ul style="list-style-type: none"> - user 테이블에서 아이디와 점수를 불러와서 점수를 기준으로 5위까지의 순서를 출력한다. <pre>def info()</pre> <ul style="list-style-type: none"> - 로그인한 아이디의 점수와 플레이타임을 출력해서 보여준다. |
| 개인후기 | <p>배웠었던 기능을 최대한 써볼려고 노력을 했었다. 특히나 정규표현식과 예외처리를 다시 검색해보고 배우고 쓰게되면서 좋은 복습의 경험이 되었다. 또한 구현하고 싶은 기능을 각각의 함수로 만들고 테스트하며 취합하는 과정에서 사용했던 기능들의 개념을 다시 정리하면서 기억에 남게 해주었다.</p> |