Databases Programming Project: **Progress** **Report**

이름 (Name): 김윤하, 구지원

학과 (Department): 정보컴퓨터공학부

학번 (Student ID): 201924437, 202155507

1. **변경된 점과 이유**

|  |
| --- |
| **- User Table 수정**   * Table의 이름 변경 : ‘User’라는 이름의 테이블은 사용할 수 없었다. 따라서 ‘**Users’** 테이블로 이름을 변경하였다. * **UserPassword** **Column 추가** (VARCHAR(255)) : 로그인시 필요함. * **UserPassword를 SHA-256으로 암호화** : 보안 강화를 위해 비밀번호를 안전한 해시 함수로 암호화하였다. * **UserEmail Column 추가** (VARCHAR(255)) : **로그인시 ID**로 사용하기 위해 * **UserRole을 enum type**인 role\_enum type으로 새롭게 정의하였다. (원장(‘Principal’), 교사(‘Teacher’), 원아(‘Student’), 돌봄 주체(‘Guardian’), 기타 교직원(‘OtherSchoolStaff’), 원아 가족(‘StudentFamily‘)) |
| **- Comment Table에 CommenterID 추가**   * Commenter의 ID를 추가해야 댓글에 대한 delete 등의 권한을 부여할 수 있기 때문에, CommenterID를 추가하였다. (해당 foreign key도 추가) |

**2. 기능 관련 함수 정의**

|  |  |
| --- | --- |
| **함수 이름** | **설명** |
| **connect\_to\_database** | psycopg2를 사용하여 PostgreSQL 데이터베이스에 연결하고 연결 및 커서를 반환한다. |
| **close** | 주어진 데이터베이스 연결을 닫는다. |
| **validate\_password** | 비밀번호의 길이를 기반으로 비밀번호를 유효성 검사한다. |
| **register** | 데이터베이스에 새로운 사용자를 등록하고 비밀번호를 저장하기 전에 해시화한다. 성공 또는 오류 메시지를 반환한다. |
| **log\_in** | 입력된 비밀번호를 데이터베이스에 저장된 해시된 비밀번호와 비교하여 사용자를 로그인한다. 성공하면 사용자 ID를 반환하고, 그렇지 않으면 None을 반환한다. |
| **log\_out** | 로그아웃을 위해 데이터베이스 연결을 닫는다. 성공 메시지를 반환한다. |
| **view\_user\_info** | 주어진 사용자 ID를 기반으로 사용자 정보 (UserName, UserRole, StudentID, UserEmail)를 가져온다. 결과 또는 오류 메시지를 반환한다. |
| **view\_student\_info** | 주어진 사용자 ID를 기반으로 학생에 대한 정보를 가져온다. 결과 또는 오류 메시지를 반환한다. |
| **manage\_student\_info** | 주어진 사용자 ID를 기반으로 학생 정보 (Attendance, HealthStatus, Address)를 업데이트한다. 성공 또는 오류 메시지를 반환한다. |
| **view\_chat** | 수신자 ID 및 사용자 역할 (Principal 또는 Teacher)을 기반으로 채팅 메시지를 가져온다. 결과 또는 오류 메시지를 반환한다. |
| **insert\_chat** | 데이터베이스에 새로운 채팅 메시지를 삽입한다. 성공 또는 오류 메시지를 반환한다. |
| **set\_schedule** | 데이터베이스에 새로운 일정을 등록하고 여러 학생 ID와 연관시킨다. 성공 또는 오류 메시지를 반환한다. |
| **view\_schedule** | 주어진 날짜 또는 모든 일정이 있는 경우 모든 일정을 기반으로 일정 정보를 가져온다. 결과 또는 오류 메시지를 반환한다. |
| **guardian\_select** | 보호자를 학생과 연결한다. 성공 또는 오류 메시지를 반환한다. |
| **post\_free\_board** | 자유 게시판에 제목, 내용 및 선택적 이미지를 사용하여 새 게시물을 게시한다.. 생성된 게시물의 PostID를 반환한다. |
| **get\_comments\_by\_postid** | PostID를 기반으로 게시물과 관련된 댓글을 검색한다. 결과 또는 오류 메시지를 반환한다. |
| **view\_post** | PostID를 기반으로 게시물의 세부 정보 및 댓글을 검색한다. 결과 또는 오류 메시지를 반환한다. |
| **write\_post\_comment** | 특정 게시물에 새로운 댓글을 작성한다. 성공 또는 오류 메시지를 반환한다. |
| **view\_free\_board** | 자유 게시판의 게시물 PostID 및 제목을 검색한다. 결과 또는 오류 메시지를 반환한다. |
| **delete\_post\_free\_board** | PostID를 기반으로 게시물 및 관련된 댓글을 삭제한다. 성공 또는 오류 메시지를 반환한다. |
| **delete\_comment** | CommentID를 기반으로 댓글을 삭제한다. 성공 또는 오류 메시지를 반환한다. |
| **register\_meal** | 특정 날짜의 식단 계획을 등록하고 식사 세부 정보를 저장한다. 성공 또는 오류 메시지를 반환한다. |
| **view\_todays\_meal** | 현재 날짜의 식단 계획을 데이터베이스에서 가져온다. 결과 또는 오류 메시지를 반환한다. |
| **view\_other\_days\_meal** | 특정 날짜의 식단 계획을 데이터베이스에서 가져온다. 결과 또는 오류 메시지를 반환한다. |

**3. UI 구현 - Flask**

|  |  |
| --- | --- |
| **작업 공간 구성** | 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명  test\_all.py : 기능 관련 함수 정의  templates : \*.html 파일 |
| **Index.html** | 텍스트, 스크린샷, 노랑, 폰트이(가) 표시된 사진  자동 생성된 설명  웹 실행 시 초기 화면.  로그인 또는 회원가입 페이지로 이동할 수 있다. |
| **login.html** | 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  자동 생성된 설명  UserEmail 및 Password 입력을 통해 로그인하는 페이지이다. 로그인 실패시 ‘Invalid credentials. Please try again.’라는 메세지가 나타난다. |
| **registering.html** | 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진  자동 생성된 설명  새로운 사용자는 Username, UserEmail, Password, Role, Student ID를 입력하여 회원가입 할 수 있다.  여기서 입력한 패스워드는 암호화되어 DB에 저장된다. |
| **dashboard.html** | 텍스트, 스크린샷, 폰트, 노랑이(가) 표시된 사진  자동 생성된 설명  로그인 성공 시 나타나는 화면이다.  게시판 / 식단표 / 알림장 / 로그아웃 버튼을 통해 원하는 기능을 수행시킬 수 있다. |
| **board.html** | 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명  Dashboard에서 ‘게시판’ 버튼 클릭 시, 게시판에 작성된 각 Post의 id 및 title만 출력한다.  CSS 추가를 통해 더욱 가독성 있는 자유게시판 구현이 필요하다. |
| **post\_detail.html** | 텍스트, 스크린샷, 폰트이(가) 표시된 사진  자동 생성된 설명  board.html에서 상세한 내용을 보기 원하는 Title을 클릭 시, 해당 포스트의 내용과 작성된 시간 및 등록된 이미지를 출력해준다. 또한 해당 포스트에 작성된 댓글도 출력해준다.  댓글 쓰기 기능을 추가해야 하고, CSS 추가를 통해 마찬가지로 더욱 가독성있는 자유게시판 구현이 필요하다. |
| **new\_free\_board.html** | 텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진  자동 생성된 설명  board.html에서 Create New Post 버튼 클릭 시 보여지는 화면이다.  Title, Content, Image(선택 사항)을 등록하고, Submit 버튼을 클릭해 자유게시판에 글을 등록할 수 있다.  CSS 추가를 통해 마찬가지로 더욱 가독성있는 자유게시판 글쓰기 구현이 필요하다. |
| **meal.html** | Dashboard에서 ‘식단표 보기’ 버튼 클릭 시, 오늘의 식단을 열람할 수 있다.  원하는 날짜의 식단을 보고싶을 경우, 달력에서 해당 날짜를 선택하여 그 날짜에 등록된 식단을 출력해준다.  (구현 필요) |
| **notification.html** | 알림장 기능으로, User의 Student id가 받은 알림장 내용만 출력해준다.  (구현 필요) |

**4. 개발 일정**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **11/28-12/1** | | | | **12/2-12/5** | | | | **12/6-12/9** | | | | **12/10-12/12** | | |
| **11/28** | **11/29** | **11/30** | **12/1** | **12/2** | **12/3** | **12/4** | **12/5** | **12/6** | **12/7** | **12/8** | **12/9** | **12/10** | **12/11** | **12/12** |
| **진행 보고서**  **작성** | |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | **User Role에 따른**  **권한 부여** | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | **FreeBoardQA, Comment**  **구현** | | |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | **MealPlan, Guardian Selection**  **구현** | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  | **Notification 구현**  **(알림장)** | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  | **원아 정보 관리**  **구현** | | |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  | **UI 수정 및**  **마무리** | | |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  | **최종 보고서**  **작성** | | |