

블로그 만들기

1. 목표

- 데이터를 생성, 조회, 삭제, 수정할 수 있는 Web Application 제작
- sqlite3를 통한 데이터 조작

2. 준비 사항

1. (필수) Flask Framework 사용을 위한 환경 설정

- C9 개발 가능 환경

3. 요구 사항

1. 데이터베이스

- 데이터 베이스 파일의 이름은 `blog.db` 입니다.
- 테이블의 이름은 `articles` 이며, 다음과 같은 정보를 저장합니다.
- 필드명: 내용, 자료형
 - `title`: 제목, String
 - `content`: 내용, String
 - `created_at`: 작성일자, String
 - `author`: 글쓴이(이메일), String

2. 페이지

1. 글 목록

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/` 입니다.
2. (필수) 데이터베이스에 존재하는 모든 블로그 글의 목록이 표시 되며, 각 블로그 글의 `title` 이 표시됩니다.
3. (필수) `title` 을 클릭 시, 해당 `블로그 조회` 페이지로 이동합니다.
4. (필수) 글 목록 최상단에 `새 글 등록` 링크가 있으며, 클릭 시 `새 글 Form` 페이지로 이동합니다.

2. 새 글 생성 Form

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/new/` 입니다.
2. (필수) 글을 작성할 수 있는 Form이 표시 되며, 다음과 같은 input들을 가지고 있습니다.
 - `title`: input text
 - `content`: textarea
 - `author`: input text
3. (필수) Form에 작성된 정보는 Form Submit 버튼 클릭 시 `글 생성` 페이지로 생성 요청 (request)과 함께 전송됩니다.
4. `created_at` 작성 일자의 경우, 작성 당시 시간을 저장합니다. (python의 datetime 활용)

5. (선택) 요청을 보내는 방식(method)은 GET, POST 중 어느 것을 사용하여도 무관합니다.

3. 새 글 생성

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/create/` 입니다.

2. (필수) 이전 페이지로부터 전송 받은 데이터를 데이터베이스에 저장합니다.

3. (필수 - 택1)

- '새 글 생성이 완료되었습니다.' 메시지가 표시되며, 글 목록으로 이동하는 링크가 표시됩니다.
- 해당 페이지에서 저장한 글을 조회하는 글 상세 페이지로 Redirect 합니다.

4. 글 상세 페이지

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/1/`, `/articles/2/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 글 정보의 Primary Key가 들어갑니다.

2. (필수) 해당 Primary Key를 가진 글의 모든 정보가 표시됩니다.

3. (필수) 글 정보의 최하단에는 목록, 수정, 삭제 링크가 있으며, 클릭 시 각각 글 목록, 해당 글 수정 Form, 해당 글 삭제 페이지로 이동합니다.

5. 글 수정 Form

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/1/edit/`, `/articles/1/edit/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 글 정보의 Primary Key가 들어갑니다.

2. (필수) 해당 Primary Key를 가진 글 정보를 수정할 수 있는 Form이 표시되며, 정보가 입력된 채로 다음과 같은 input들을 가지고 있습니다.

- `title`: input text
- `content`: textarea
- `author`: input text

3. (필수) Form에 작성된 정보는 Form Submit 버튼 클릭 시 글 수정 페이지로 수정 요청(request)과 함께 전송됩니다.

4. `created_at` 작성 일자의 경우, 수정 당시 시간을 저장합니다. (python의 datetime 활용)

5. (선택) 요청을 보내는 방식(method)은 GET, POST 중 어느 것을 사용하여도 무관합니다.

6. 글 수정

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/1/update/`, `/articles/2/update/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 글 정보의 Primary Key가 들어갑니다.

2. (필수) 해당 Primary Key를 가진 글 정보를 이전 페이지로부터 전송 받은 데이터로 변경하여 데이터베이스에 저장합니다.

3. (필수 - 택1)

- '글 수정이 완료되었습니다.' 메시지가 표시되며, 글 목록으로 이동하는 링크가 표시됩니다.
- 해당 페이지에서 수정한 글을 조회하는 글 상세 페이지로 Redirect 합니다.

7. 글 삭제

1. (필수) 해당 페이지에 접근하는 URL은 `/articles/1/delete/`, `/articles/2/delete/` 등이며, 동적으로 할당되는 부분이 존재합니다. 동적으로 할당되는 부분에는 데이터베이스에 저장된 글 정보의 Primary Key가 들어갑니다.

2. (필수) 해당 Primary Key를 가진 글 정보를 데이터베이스에서 삭제합니다.

3. (필수 - 택1)

- '글 삭제가 완료되었습니다.' 메시지가 표시되며, 글 목록으로 이동하는 링크가 표시됩니다.
- 글 목록 페이지로 Redirect 합니다.

8. 심화 Bootstrap을 활용하여 꾸미기

1. Bootstrap을 활용하여 자유롭게 완성된 블로그를 꾸며 봅니다.

4. 결과 예시

Python Web Framework를 활용해 작성한 모든 파일을 `blog` 디렉토리에 위치하도록 합니다.

결과물은 반드시 `README.md` 로 활용 하였던 내용을 정리 해주세요.

```
blog/  
  ...  
  ...  
  ...  
  README.md
```