

데이터베이스 교육실제연구

Practical Study on Teaching Database

- REPORT -

가계부 앱 사용설명서



강 의 시 간	금 5~7 교시
지 도 교 사	한동윤 교수님
학과(전공)	컴퓨터교육과
학 번	201927016
성 명	정송헌

가) 개발목적

나) 구현환경

다) 컴포넌트 구조

라) 프로그램 설치과정

마) 기능

1) 수입/지출 입력

2) 수입/지출 내역 수정

3) 수입/지출 내역 삭제

바) 프로그램 동작 과정

사) 데이터 베이스 설계

가 개발 목적

실생활에서 가계부를 수기로 작성하던 불편함을 없애는 동시에 분실방지 및 관리의 용이성을 위해 최소한의 기능만 가지고 있는 앱을 제작하였다. 이 앱을 통해 사용자가 수입/지출이 생겼을 때 원하는 장소에서 언제든지 작성할 수 있고 내역을 스마트폰 앱 또는 컴퓨터를 이용하여 쉽게 확인 가능하며 수입/지출 내역을 쉽게 관리할 수 있다.

나 구현 환경

Windows 11

Yarn

React

MySQL

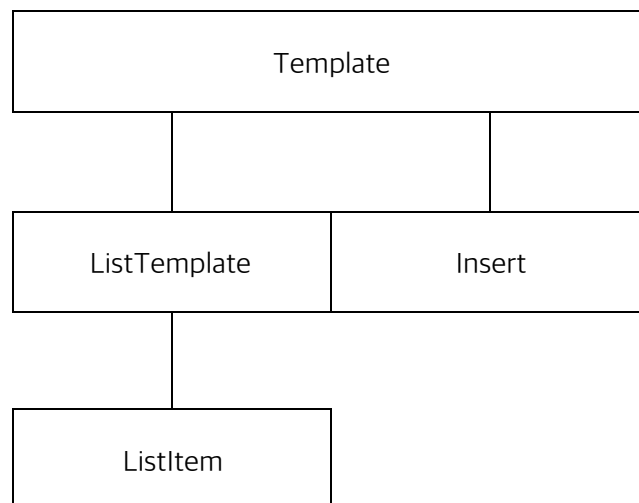
Dbeaver(Maria DB)

FastAPI

* 테스트 환경 : Chrome Browser

다 컴포넌트 구조

1. 컴포넌트 트리



다 컴포넌트 구조

2. 컴포넌트 설명

- Template : 전체 앱 레이아웃. 화면 상단 AppTitle 레이어와 하단 AppContent로 구성.
- ListTemplate : 수입/지출내역 출력부분. 상단 ListTitle에 제목이 표시되고 그 아래로 ListItem이 지출내역으로 출력됨.
- ListItem : 수입/지출내역이 최신순으로 정렬되어 출력됨. 수입/지출 내역을 확인할 수 있으며 해당 내역을 수정, 삭제 가능함.
- Insert : 오른쪽 하단에 고정되어있는 버튼을 누르면 나타나는 입력폼 컴포넌트. 수입/지출 내역을 등록 가능함.

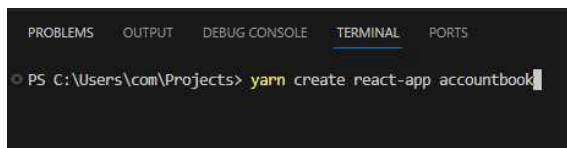
라 프로그램 설치과정

1. 설치 전 준비사항

- Node.js : <https://nodejs.org/download/release/v16.20.1/>
- yarn : 터미널에서 `npm install --global yarn`
- MySQL : <https://dev.mysql.com/downloads/>
- Python : <https://www.python.org>

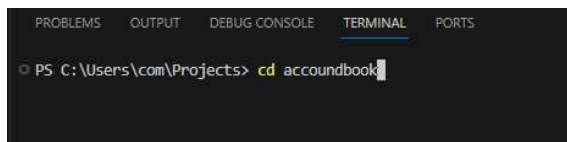
2. 새 React프로젝트 생성

- 1) 터미널에서 `yarn create react-app 프로젝트이름(accountbook)` 을 입력하여 서버를 구동할 프로젝트를 생성합니다.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\com\Projects> yarn create react-app accountbook
```

- 2) 이후 `cd` 명령어를 통해 생성한 프로젝트로 이동합니다.



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\com\Projects> cd accountbook
```

3. Python 가상환경 만들기

- 1) 터미널에서 가상 환경을 만들고자 하는 디렉토리로 이동한 후, 다음 명령을 수행하여 가상 환경의 홈 디렉토리를 만들고 그 디렉토리로 이동합니다.

2) 가상 환경을 만들 수 있게 다음 명령어를 실행합니다. (가상 환경의 이름은 myapi, 다른 이름도 상관없습니다.)

```
mkdir venv
```

```
cd venv
```

3) 명령 프롬프트 창을 엽니다. 그리고 venv 폴더로 이동합니다.

python --m venv myapi 명령어를 입력합니다.

4) 가상 환경에 진입하기

만들어진 가상 환경 내에 있는 Scripts 디렉토리로 이동합니다.

```
cd myapi
```

```
cd Scripts
```

5) 가상 환경을 활성화시키기 위해 activate 명령을 실행하면 가상 환경에 진입하고 이는 프롬프트에 (myapi) 라는 구문이 나타나는 것을 확인하면 됩니다.

6) FastAPI 설치하고 새 프로젝트 폴더 만들기

앞에서 만든 가상 환경에 진입합니다. 다음의 명령을 실행하여 FastAPI를 설치합니다.

```
pip install fastapi
```

7) 만약 “To update, run: python.exe -m pip install --upgrade pip”라는 메시지가 나오면 pip를 최신 버전으로 업그레이드하라는 의미이므로 다음 명령을 실행하여 pip를 업그레이드합니다

```
python -m pip install --upgrade pip
```

8) 백엔드에서 처리된 메시지를 확인하기 위해 비동기 방식의 웹서버가 필요한데 이런 방식으로 자주 사용되는 서버가 uvicorn입니다. uvicorn을 다음과 같이 설치합니다.

```
pip install uvicorn
```

9) 프로젝트 폴더로 이동하여 DB-main 이라는 이름의 폴더를 만들어서 해당 폴더로 이동합니다.

```
cd C:\projects
```

```
mkdir accountbookdb
```

```
cd accountbookdb
```










10) 제공된 DB확인

제공된 DB-main 폴더의 파일들을 알맞은 경로에 복사합니다.

 _pycache_	2023-12-17 오전 8:21	파일 폴더	
 domain	2023-12-17 오전 8:21	파일 폴더	
 database.py	2023-12-17 오전 8:25	Python 원본 파일	1KB
 main.py	2023-12-16 오후 2:47	Python 원본 파일	1KB
 models.py	2023-12-16 오후 2:47	Python 원본 파일	1KB

11) 제공된 프론트엔드 파일 확인

제공된 accountbook의 파일들을 알맞은 경로에 복사합니다.

 components	2023-12-17 오전 8:21	파일 폴더	
 App.css	2023-12-16 오후 2:43	CSS 스타일시트 ...	1KB
 App.js	2023-12-17 오전 8:21	JavaScript 원본 파...	3KB
 App.test.js	2023-12-17 오전 8:21	JavaScript 원본 파...	1KB
 index.css	2023-12-16 오후 2:43	CSS 스타일시트 ...	1KB
 index.js	2023-12-17 오전 8:21	JavaScript 원본 파...	1KB
 logo.svg	2023-12-16 오후 2:43	Microsoft Edge H...	3KB
 reportWebVitals.js	2023-12-17 오전 8:21	JavaScript 원본 파...	1KB
 setupTests.js	2023-12-17 오전 8:21	JavaScript 원본 파...	1KB

12) 데이터베이스 생성

아이디: dbuser

패스워드: db12

포트: 3306

데이터베이스명 sqldb

15) DB 테이블 생성

앱과 연결할 테이블을 생성하여 줍니다.

```
create table accountbook (
  id int auto_increment,
  date date,
  divi varchar(10),
  list varchar(100),
  pay int,
  primary key (id)
)

insert into accountbook(date, divi, list, pay) values('2021-12-14', '수입', '용돈', '500000');
insert into accountbook(date, divi, list, pay) values('2022-12-14', '지출', '커피', '5000');
insert into accountbook(date, divi, list, pay) values('2023-12-14', '지출', '맛식스', '2200');
```

16) 그리고 명령 프롬프트의 (myapi)가상 환경에서 다음 명령을 실행하여 uvicorn 웹 서버를 실행합니다.(백엔드 웹 서버)

uvicorn main:app --reload

17) yarn 실행



Visual Studio Code에서 이동한 프로젝트 터미널에서 yarn start를 실행합니다.


1. 수입/지출내역 확인

사용자가 앱을 실행하였을 때 가계부 DB에 입력되어있는 내역을 확인 할 수 있다. 목록은 최신순으로 정렬되어 가장 최근에 소비한 내역부터 오래된 내역 순으로 확인할 수 있다. 사용자가 입력한 내용이 지출인지 수입인지 한눈에 확인하기 위해서 지출내역은 빨간 배경에 빨간색 글자로 지출이라고 나타나고 수입 내역은 초록 배경에 파란색 글자로 지출임을 나타내도록 구현했다.




숫자는 천 단위로 자동으로 쉼표를 표시하도록 구현되어 있다.

2. 수입/지출내역 등록


화면 오른쪽 하단의  버튼을 선택하여 지출내용 또는 수입내용을 입력 할 수 있다.  버튼을 선택하면 바로 위에 수입/지출내역 입력양식이 나타난다.

- 날짜는 키보드를 직접 입력하거나 입력창 오른쪽 끝의 달력 아이콘을 눌러 선택할 수 있으며 공백으로 둔 채 입력 버튼을 누른 경우 날짜를 입력하라는 경고 메시지와 함께 등록이 중단된다.
- 구분은 선택상자에서 수입과 지출 중 선택할 수 있으며 선택하지 않고 공백으로 둔 채 입력 버튼을 누른 경우 구분을 선택하라는 경고 메시지와 함께 등록이 중단 된다.
- 내용은 입력상자에 수입/지출 내용을 기록할 수 있도록 구성되어 있으며 공백으로 둔 채 입력 버튼을 누른 경우 내용을 입력하라는 경고 메시지와 함께 등록이 중단 된다.
- 금액은 입력상자에 수입/지출 금액을 기록할 수 있도록 구성되어 있으며 숫자 이 외에 기타 문자, 특수문자, 공백문자는 입력이 불가능하도록 처리되어 있다. 또한 공백으로 둔 채 입력 버튼을 누른 경우 내용을 입력하라는 경고 메시지와 함께 등록이 중단된다.
- 양식을 모두 입력한 후 입력버튼을 누르면 방금 입력한 수입/지출 현황이 목록의 제일 위에 표시된다.
- 입력 버튼을 선택하여 수입/지출 내역 입력이 끝나면 빈 양식으로 표시되어 다른 내역을 입력하기 쉽게 구현하였다.
- 입력을 취소할 경우 닫기 또는 오른쪽 하단의  버튼을 한번 더 눌러 취소할 수 있다.

3. 수입/지출내역 수정

- 수입/지출 항목 오른쪽의  아이콘을 선택하여 내역 수정이 가능하다. 만약 수정을 취소하고 싶다면  버튼을 누르면 된다.
- 수정 중 하나 이상의 항목을 공백으로 둔 경우 경고창과 함께 수정이 중단된다.
- 금액은 숫자 이외의 문자, 특수문자는 입력이 불가능 하도록 구현 되어있다.
- 입력을 완료한 경우  버튼을 눌러 수정을 완료한다.
- 수정이 완료되면 수정 양식이 없어지고 수정된 사항은 즉시 화면에 반영되며 데이터베이스에도 업데이트된다.

4. 수입/지출내역 삭제

- 수입/지출 항목 오른쪽의  아이콘을 선택하여 내역 삭제가 가능하다.
- 삭제된 사항은 즉시 화면에 반영되며 데이터베이스에도 업데이트된다.

바 프로그램 동작

1) 초기화면

- 최근 수입/지출 현황을 최신 순으로 확인 가능



- 하단의 **+**을 선택하여 새 수입/지출 현황 입력 가능

날짜: 연도-월-일

구분:

내용: 지출내용을 입력해주세요.

금액: 금액을 입력해주세요.

입력 닫기

+

- 수입/지출 항목 오른쪽의 **✎**을 선택하여 수정 가능

지출

2023-12-04

집 구매

2,000,000원

✎ -

[선택 전]

지출

2023-12-04

집 구매

2000000

✓ ✕

[선택 후]

- 수입/지출 항목 오른쪽의 **-**을 선택하여 삭제 가능

최근 수입/지출 현황

지출

2023-12-04

집 구매

2,000,000원

✎ -

수입

2023-12-01

로또

200,000,000원

✎ -

지출

2022-02-02

람보르기니

1,000,000원

✎ -

수입

2021-01-01

용돈

100,000,000원

✎ -

+

최근 수입/지출 현황

수입

2023-12-01

로또

200,000,000원

✎ -

지출

2022-02-02

람보르기니

1,000,000원

✎ -

수입

2021-01-01

용돈

100,000,000원

✎ -

[삭제 전, 후 현황목록]

1. 테이블 구조

- Database명: sqldb
- Table명: accountbook
- 속성

컬럼명	데이터 형식	Not Null	Auto Increment	Key
id	int	O	O	Primary
date	date	O		
divi	varchar(10)	O		
list	varchar(100)	O		
pay	int	O		

2. 설계 시 고려사항

- 입력된 내역 중 id값을 중복으로 가지는 경우를 방지하기 위해 id 속성은 auto_increment 옵션을 사용하였고 primary key로 설정하였다.
- date 속성은 시간까지 입력해야 할 필요성을 발견하지 못하여 date형식으로 설정하였다.
- divi 속성은 수입/지출 내역 구분 속성으로 초기에 Boolean으로 설정하려 했으나 차후 버전 업데이트나 데이터베이스에서 데이터 조회 시 가독성을 높이기 위해서 varchar로 설정하였다.
- list 속성은 수입/지출 상세내역 기록용 속성으로 화면에서 표시 될 때 가능한 한 여러줄로 표시되는 것을 방지하고자 varchar(100)으로 설정하였고 프론트엔드에서 최대글자 길이를 제한하여 입력 범위를 벗어나지 않도록 구현하였다.
- pay 속성은 수입/지출 금액을 기록하는 속성으로 int 형식으로 구현하였고 목록에서 소수점으로 표현하는 것은 프론트엔드에서 구현하였다.
- 프론트엔드 스크립트에서 공백입력이 방지되도록 처리하였으나 만약의 경우를 대비하여 Null값을 포함한 레코드가 입력되는 것을 방지하기 위해 모든 컬럼에 Not Null을 설정하였다.

