

# My Bookshelf

Vivianna Tilander

## 1. Johdanto ja sovelluksen kuvaus

My Bookshelf on opiskelijaprojektina toteutettu harjoitustyö Helsingin yliopiston tarjoamaa Tietokantasovellus-kurssia varten.

My Bookself on web-sovellus, jonka avulla sovelluksen käyttäjä voi pitää kirjaa oman kirjahyllynsä sisällöstä.

Kirautumattomlle käyttäjällä sovellus tarjoaa mahdollisuuden selata sovelluksen tietokantaan lisättyjä teoksia, kategorioita ja kirjailijoita. Lisäksi käyttäjä voi selata tiettyyn kategoriaan kuuluvia tai tietyn kirjailijan teoksia.

Rekisteröitynyt tavallinen käyttäjä voi siirtyä omaan "kirjahyllyynsä", jossa hänen on mahdollista tarkastella sinne lisäämiään painoksia eli versioita teoksista (siis yksittäisiä kappaleita kirjoja, sarjakuvia, mangaa jne.). Lisäksi rekisteröityneen käyttäjän on mahdollista lisätä ja poistaa painoksia hyllystään sekä määrittää ja muuttaa niiden tietoja. Kirjautuminen sovellukseen tapahtuu käyttäjätunnuksen ja salasanan avulla. Rekisteröityneellä käyttäjällä on myös oikeus tarkastella, muuttaa ja poistaa omat tietonsa.

Rekisteröityneellä tavallisella käyttäjällä on myös mahdollisuus lisätä puuttuvia teoksia, kategorioita ja kirjailijoita. Sekä lisätä kategorioita teoksiin sekä teoksia kirjailijoille.

Järjestelmällä on lisäksi adminiteitä/ylläpitäjiä, joilla on oikeus muuttaa sellaisia tietoja, jotka voivat vaikuttaa/vaikuttavat esimerkiksi muiden käyttäjien painosten tietoihin. Esimerkkinä tällaisesta ovat kategoroiden/kirjailijoiden/teosten tietojen muokkaus sekä kategorioiden/kirjailijoiden/teosten poistaminen tietokannasta. Admin voi myös lisätä uusia formaatteja, antaa toisille käyttäjille admin-oikeudet sekä tarkastella listaa admin-roolillisista käyttäjistä.

Sovellus on kirjoitettu Pythonilla (Python 3) ja se hyödyntää Flaskia sovelluskehiksenä. Käytössä on myös SQLAlchemy tietokannan käsittelyssä. Sivujen tyylissä apuna on käytetty Bootstrap:ia ja CSS:sää.

Sovellus on tarkasteltavissa Herokussa: [Sovellus Herokussa](#).

Herokussa vain adminiteille sallitun sisällön testaus onnistuu kirjautumalla käyttäjätunnuksella Plopper ja salasanalla testplopper.

Tietokannan hallinnassa sovellus käyttää Herokussa PostgreSQL:lää ja paikallisesti käynnistettynä SQLitea.

## 2. Sovelluksen käyttöohje

[Sovellus on Herokussa](#). Sovellus avautuu aloitussivulle, jossa kerrotaan tietoa sovelluksen käyttäjien lukumäärästä ja "kirjahyllyihin" lisättyjen painosten yhteismäärästä. Aloitussivun

ylälaidassa, kuten sovelluksen jokaisen sivun ylälaidassa on navigointipalkki, jonka avulla sivulta toiselle suunnistuksen olisi tarkoitus olla intuitiivista ja helppoa. Palkissa tarjolla olevien linkkien määrä riippuu käyttäjän roolista.

Esimerkki navigointipalkin linkki "Works" vie sivulle, jolla listataan kaikki tietokannassa olevat teokset. Listattujen teoksien nimet toimivat linkkeinä, jotka vievät kyseisen teoksen tietoihin. Vastaavasti navigointipalkin linkki "Authors" vie sivulle, jossa listataan kaikki kirjailijat.

Jotta käyttäjä voisi lisätä painoksia (Editions) omaan "kirjahyllyynsä" tulee käyttäjän olla kirjautunut. Kirjautuminen onnistuu navigointipalkin Register-linkistä ja jatkossa kirjautuminen onnistuu Log in -linkkiä painamalla ja kirjautumalla omalla käyttäjätunnuksella ja salasanalla.

HUOM! Testausta varten Herokussa olevaan sovellukseen ([Sovellus Herokussa](#)) on lisätty testi admin tunnukset. Kirjautuminen adminina onnistuu klikkaamalla navigointipalkin Log in -linkkiä ja syöttämällä Username-kenttään Plopper ja Password-kenttään testplopper. Pyydän ettet admin-oikeuksia testatessasi lisää tuntemattomia käyttäjiä adminiksi tai poista olennaista tietoa sovelluksesta. Toivon, että luot oman sisältösi joita sitten voit vapaasti poistaa ja muokata.

### 3. Sovelluksen asennusohje

Sovelluksen käynnistämiseksi paikallisesti tulee osoitteessa <https://github.com/Viannaiv/My-Bookshelf> painaa vihreää nappia "Clone or download" ja valita "Download ZIP". Pura zip-paketti uuteen tiedostoon.

Huomaa, että sinulla tulee olla Python 3 asennettuna koneellesi.

Siirry terminaalissa juuri purettuun tiedostoon (siis uuteen kansioon, joka luotiin purettaessa zip) ja aja komento `python -m venv venv`. Tämän jälkeen aja komento `source venv/bin/activate`. Huom! Windowsilla vastaava komento on `venv\Scripts\activate.bat`.

Jotta sovellus toimisi tulee seuraavaksi asentaa tiedostossa requirements.txt sijaitsevat vaatimukset. Tämä onnistuu komennolla `pip install -r requirements.txt`.

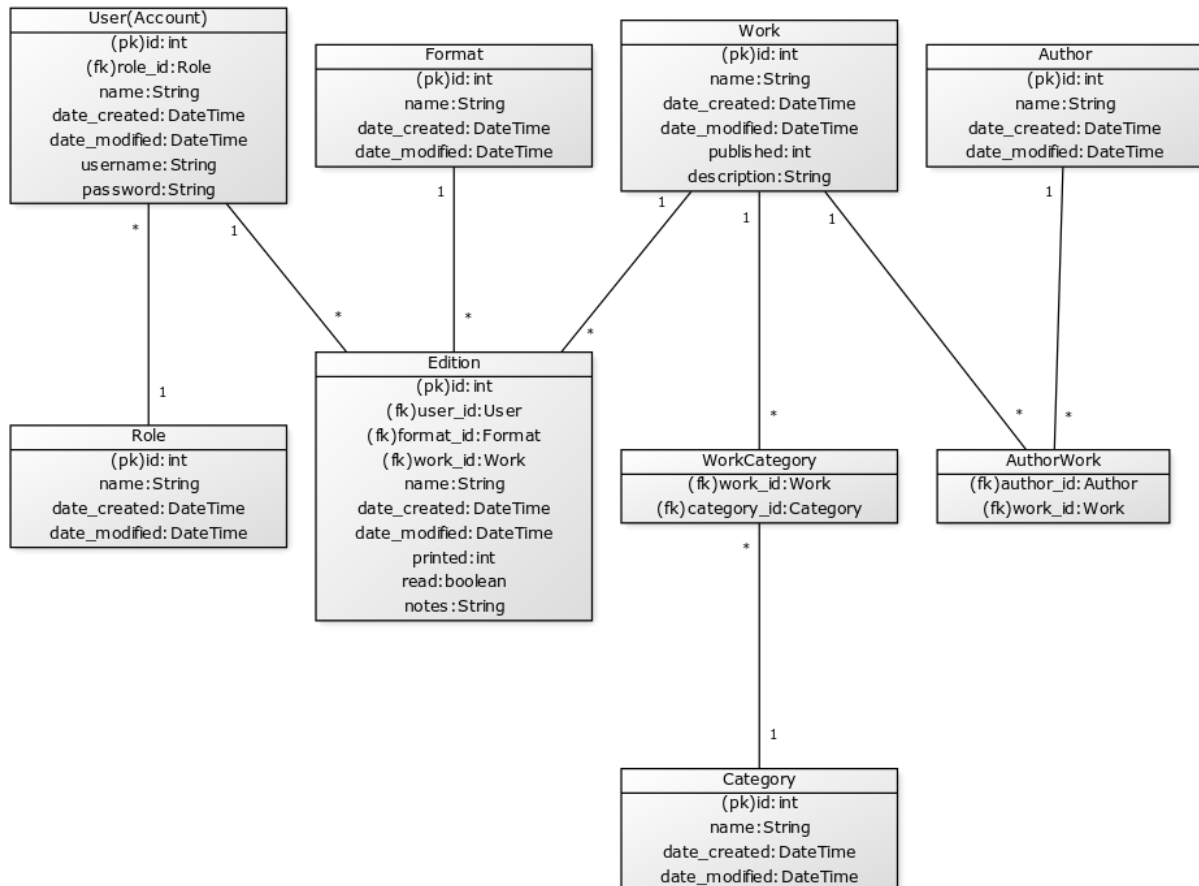
Tämän jälkeen sovelluksen käynnistys onnistuu komennolla `python run.py` tai `python3 run.py`. Sovellus on nyt käynnissä ja tarkasteltavissa selaimessa osoitteessa : <http://localhost:5000/>

### 4. Sovelluksen rajoitteet ja puutteet

Olisi hyvä, jos sovelluksessa esitetyt listat olisivat aakkostettuja, nyt vain osa on. Lisäksi olisin halunnut lisätä sovellukseen haku mahdollisuuden, jolla teoksia, painoksia jne. olisi

voinut hakea esimerkiksi nimen tai julkaisu- /painovuoden perusteella, mutta aika loppui kesken. Lisäksi sovelluksen ulkonäköä olisi voinut siistiä piilottamalla tyhjäksi jäävät taulukot. Listojen tulisi myös olla sivutettuja.

## 5. Tietokantakaavio



## 6. CREATE TABLE -lauseet

```

CREATE TABLE role (
id INTEGER NOT NULL,
name VARCHAR(150) NOT NULL,
date_created DATETIME,
date_modified DATETIME,
PRIMARY KEY (id)
);
    
```

```

CREATE TABLE format (
id INTEGER NOT NULL,
name VARCHAR(150) NOT NULL,
    
```

```
date_created DATETIME,  
date_modified DATETIME,  
PRIMARY KEY (id)  
);
```

```
CREATE TABLE category (  
id INTEGER NOT NULL,  
name VARCHAR(150) NOT NULL,  
date_created DATETIME,  
date_modified DATETIME,  
PRIMARY KEY (id)  
);
```

```
CREATE TABLE author (  
id INTEGER NOT NULL,  
name VARCHAR(150) NOT NULL,  
date_created DATETIME,  
date_modified DATETIME,  
PRIMARY KEY (id)  
);
```

```
CREATE TABLE work (  
id INTEGER NOT NULL,  
name VARCHAR(150) NOT NULL,  
date_created DATETIME,  
date_modified DATETIME,  
published INTEGER NOT NULL,  
description VARCHAR(600) NOT NULL,  
PRIMARY KEY (id)  
);
```

```
CREATE TABLE account (  
id INTEGER NOT NULL,  
name VARCHAR(150) NOT NULL,  
date_created DATETIME,  
date_modified DATETIME,  
username VARCHAR(50) NOT NULL,  
password VARCHAR(20) NOT NULL,  
role_id INTEGER NOT NULL,  
PRIMARY KEY (id),  
FOREIGN KEY(role_id) REFERENCES role (id)  
);
```

```
CREATE TABLE IF NOT EXISTS "WorkCategory" (  
work_id INTEGER NOT NULL,  
category_id INTEGER NOT NULL,
```

```
PRIMARY KEY (work_id, category_id),  
FOREIGN KEY(work_id) REFERENCES work (id),  
FOREIGN KEY(category_id) REFERENCES category (id)  
);
```

```
CREATE TABLE IF NOT EXISTS "AuthorWork" (  
work_id INTEGER NOT NULL,  
author_id INTEGER NOT NULL,  
PRIMARY KEY (work_id, author_id),  
FOREIGN KEY(work_id) REFERENCES work (id),  
FOREIGN KEY(author_id) REFERENCES author (id)  
);
```

```
CREATE TABLE edition (  
id INTEGER NOT NULL,  
name VARCHAR(150) NOT NULL,  
date_created DATETIME,  
date_modified DATETIME,  
printed INTEGER NOT NULL,  
publisher VARCHAR(100) NOT NULL,  
language VARCHAR(50) NOT NULL,  
read BOOLEAN NOT NULL,  
notes VARCHAR(600) NOT NULL,  
account_id INTEGER NOT NULL,  
format_id INTEGER NOT NULL,  
work_id INTEGER NOT NULL,  
PRIMARY KEY (id),  
CHECK (read IN (0, 1)),  
FOREIGN KEY(account_id) REFERENCES account (id),  
FOREIGN KEY(format_id) REFERENCES format (id),  
FOREIGN KEY(work_id) REFERENCES work (id)  
);
```

## 7. Käyttötapaukset

Käyttäjärooleja sovelluksessa ovat rekisteröitymätön käyttäjä, rekisteröitynyt käyttäjä ja ylläpitäjä. Käydään läpi käyttötapaksia:

Rekisteröitymätön käyttäjä:

- Rekisteröitymättömänä käyttäjänä voin selata teoksia, ja tarkailla jonkun tietyn teoksen tietoja.
- Rekisteröitymättömänä käyttäjänä voin selata kategorioita sekä niihin kuuluvia teoksia.

- Rekisteröitymättömänä käyttäjänä voin selata kirjailijoita sekä näiden teoksia.

```
SELECT * FROM Author;
```

Kun Author.id = 1;

```
SELECT Work.id, Work.name, Work.published FROM Work
```

```
LEFT JOIN "AuthorWork" ON "AuthorWork".work_id = Work.id
```

```
LEFT JOIN Author ON "AuthorWork".author_id = Author.id
```

```
WHERE Author.id = 1
```

```
ORDER BY Work.published
```

- Rekisteröitymättömänä käyttäjänä voin rekisteröityä.

Tavallinen rekisteröitynyt käyttäjä:

- Käyttäjänä voin tehdä kaiken, minkä rekisteröitymätön käyttäjä voi tehdä.
- Käyttäjänä voin lisätä painoksia omaan "Kirjahyllyyni". Lisäksi voin myös poistaa niitä sieltä sekä muokata niiden tietoja, jotta voin helposti pitää inventaariota oikean kirjahyllyn sisällöstä.
- Käyttäjänä voin lisätä tietokantaan teoksia, kategorioita ja kirjailijoita.
- Käyttäjänä voin liittää teoksia kirjailijoihin ja kategorioita teoksiin.
- Käyttäjänä voin tarkastella hyllyyni lisäämiäni painoksia sovelluksessa.
- Käyttäjänä voin muokata oman tilini tietoja ja poistaa tilini.

Ylläpitäjä:

- Ylläpitäjänä voin tehdä kaiken mitä tavallinen käyttäjä voi tehdä.
- Ylläpitäjänä voin käsitellä ja muokata tietokannassa olevien teosten, kategorioiden ja kirjailijoiden tietoja. Lisäksi voin poistaa niitä tietokannasta.
- Ylläpitäjänä voin lisätä muita käyttäjiä ylläpitäjiksi.
- Ylläpitäjänä voin lisätä uusia formaatteja.

## 8. Omat kokemukset

Suurin ongelma sovelluksen kanssa tuntui olevan sqlalchemy:n order\_by toiminnon toimimaan saaminen, sillä loppujen lopuksi, vaikka asiaa kuinka pyöritteli, en sitä saanut toimimaan ja nyt listat ovat edelleen aakkostamattomia.