# Book Inventory Guide

Book inventory webpage is mainly built to show case basic CRUD operations with DB using an User Interface. In our example we have used mainly Python 3.7 as backend server to handle CRUD operations, SQlite3 as Database and Html with AngularJS for Frontend.

Book inventory database consists of two tables- “Books” and “User”

Books table has below schema:-

book\_id = Column(my.INTEGER(unsigned=**True**), primary\_key=**True**, autoincrement=**True**,nullable=**False**)  
book\_author = Column(my.VARCHAR(128),nullable=**False**)  
book\_title = Column(my.VARCHAR(128),nullable=**False**)  
book\_published\_date = Column(my.VARCHAR(128),nullable=**False**)  
book\_added\_date = Column(my.DATETIME)  
book\_updated\_date = Column(my.DATETIME)

Objective of Books table is to capture above information. It can be further extended to add new schema parameters like link to download etc.,

User table has below schema:-

user\_id = Column(my.INTEGER(unsigned=**True**), primary\_key=**True**, autoincrement=**True**,nullable=**False**)  
user\_name = Column(my.VARCHAR(64),nullable=**False**)  
user\_email = Column(my.VARCHAR(64),nullable=**False**)  
user\_password = Column(my.VARCHAR(64),nullable=**False**)  
is\_admin = Column(my.BOOLEAN,nullable=**False**)  
is\_super\_user = Column(my.BOOLEAN,nullable=**False**)  
is\_active = Column(my.BOOLEAN)  
last\_login = Column(my.DATETIME)  
date\_joined = Column(my.DATETIME)

User table’s main objective is to handle user rights, capture updates to the data by the admin. In future we can extend the UI capability to delete users or modify the access/content/data they can look at

In this example we’ve used:-

Python3.7

Bottle web framework

AngularJS

Backend runs on port localhost:3031

Demo

Go to localhost:3031

Graphical user interface, website

Description automatically generated

I’ve created a login for user ‘admin’ password ‘admin’

Click login

Graphical user interface, application, website

Description automatically generated

Admin’s User interface looks a below

Graphical user interface, website

Description automatically generated with medium confidence

User can add a book detail and click ‘add’ button

Graphical user interface, website

Description automatically generated with medium confidence

Success response if bok details added

Graphical user interface, website

Description automatically generated

There is no back button- as I wanted to make this web app as simple as such I didn’t added that functionality. So the admin has to login again to see the update

Table

Description automatically generated with low confidence

Similarly Admin can delete/Update the record

Loging in as a user

Graphical user interface, application, website

Description automatically generated

An user can just see the detals of the books in the inventory, we can further implement downloadable link to download the book

Graphical user interface

Description automatically generated

A new user can register

Click Register- button

Graphical user interface, application, website

Description automatically generated

A success response looks as below

Graphical user interface, website

Description automatically generated