# Invoice Tracker

## It is recommended that you use python 3.6 virtualenv to run this application.

## Clone the git repository name InvoiceTracker. Activate the virtualenv and use the following command to install the requirements

## pip install -r requirements.txt

## In addition to that you will need to download the latest version of the postgres database.

## Once database is installed and running, open the psql CLI or pgAdmin web client to create a database called invoicedb.

## I have used the database with userid = postgres and password =tiger in this application, so the connection string for the database instance is:

## 'postgresql+psycopg2://postgres:tiger@localhost/invoicedb'

## If you have used different credentials, you may set the connection string in the \_\_init\_\_.py file in the invoice folder to match your credentials.

## The next step is to create database objects, namely two tables, invoices and invoice\_items.

## For that you will need to open the Python shell from project folder (InvoiceTracker)

## In the python REPL run the following commands to create the two tables

## >>>from invoice.models import Base

## >>> Base.metadata.create\_all(bind=db.engine)

## >>

## Once done, type quit() to exit the shell. You can make sure whether table are created by going PGAdmin or psql CLI.

## If you are running the app on windows system run the following command to start the application

## 

## set FALSK\_APP=invoice

## flask run

## In Linux environment run

## export FALSK\_APP=invoice

## flask run

## If everything goes well you will get some warnings and a link like the following to run the application on the browser

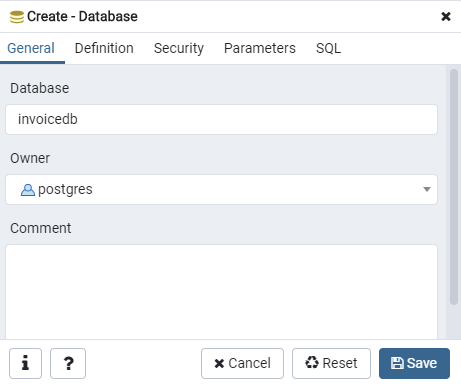
## Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)

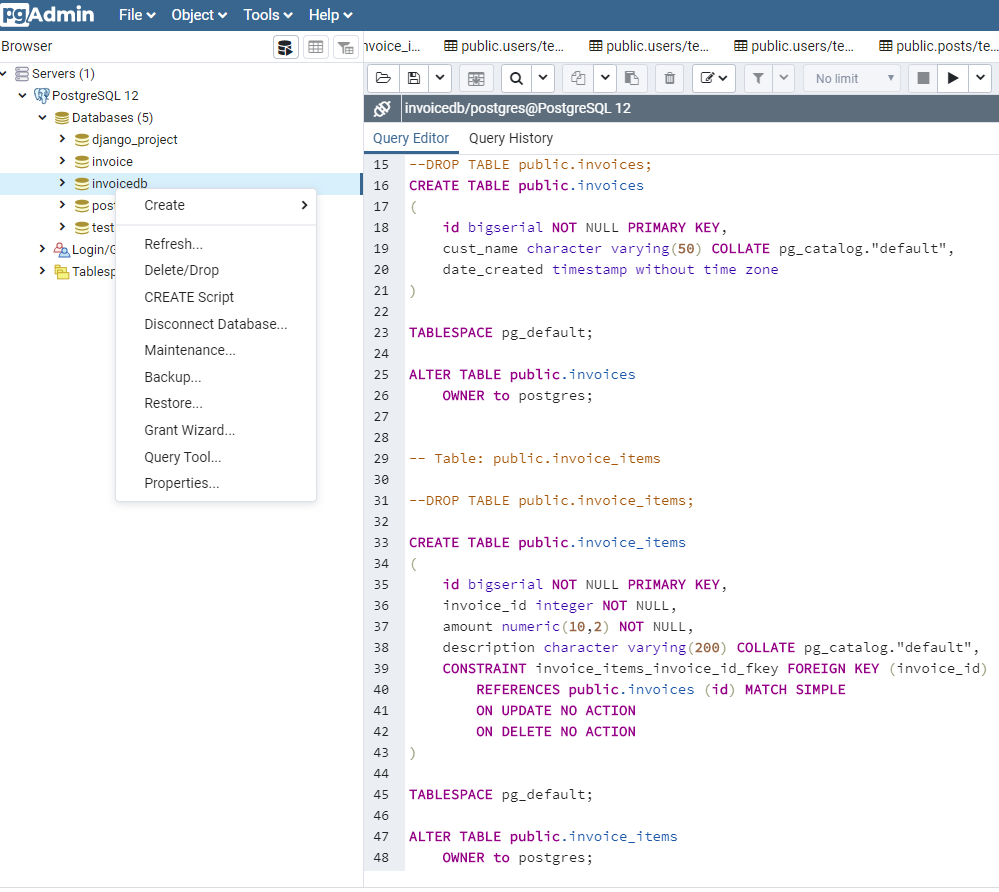
## You can go to invoice page with this link <http://127.0.0.1:5000/invoice/new> to create new invoices. To add any invoice items to the newly create invoice click on the name.

## Last 3 screenshots show what you see when you run the application.

## In case if there is an issue with creating invoiced database and the 2 tables, please use the Postgresql admin web client to create the database and the tables.

## Open the pgAdmin web tool and right click on PostgreSQL 12. Select Create and then Database. On the dialog box enter database name as invoicedb and select owner as postgres. Click on save. Once done, you will see invoicedb database is created under the Databases node. Right click on the newly created invoicedb database and select Query Tool. Copy and paste the following script on to the Query Editor, run the query to create the tables. Two screenshots below show the steps described above.





# Scripts to create invoice and invoice\_item tables

## ====================================================

-- DROP DATABASE invoicedb;

CREATE DATABASE invoicedb

WITH

OWNER = postgres

ENCODING = 'UTF8'

LC\_COLLATE = 'English\_United States.1252'

LC\_CTYPE = 'English\_United States.1252'

TABLESPACE = pg\_default

CONNECTION LIMIT = -1;

-- Table: public.invoices

--DROP TABLE public.invoices;

CREATE TABLE public.invoices

(

id bigserial NOT NULL PRIMARY KEY,

cust\_name character varying(50) COLLATE pg\_catalog."default",

date\_created timestamp without time zone

)

TABLESPACE pg\_default;

ALTER TABLE public.invoices

OWNER to postgres;

-- Table: public.invoice\_items

--DROP TABLE public.invoice\_items;

CREATE TABLE public.invoice\_items

(

id bigserial NOT NULL PRIMARY KEY,

invoice\_id integer NOT NULL,

amount numeric(10,2) NOT NULL,

description character varying(200) COLLATE pg\_catalog."default",

CONSTRAINT invoice\_items\_invoice\_id\_fkey FOREIGN KEY (invoice\_id)

REFERENCES public.invoices (id) MATCH SIMPLE

ON UPDATE NO ACTION

ON DELETE NO ACTION

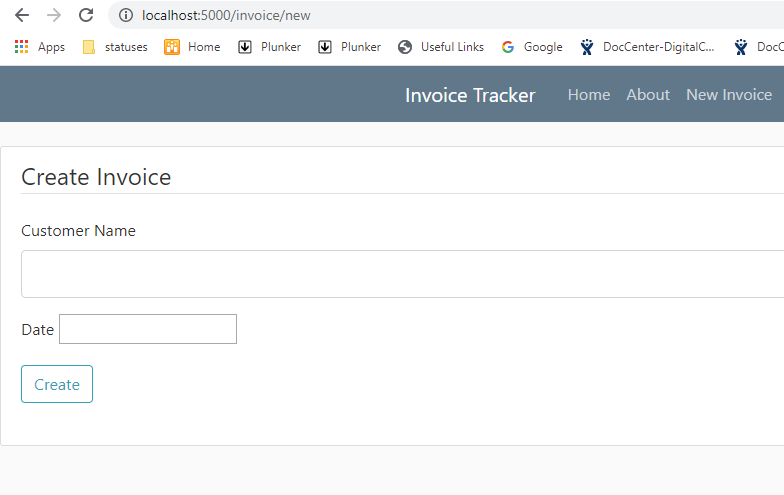
)

TABLESPACE pg\_default;

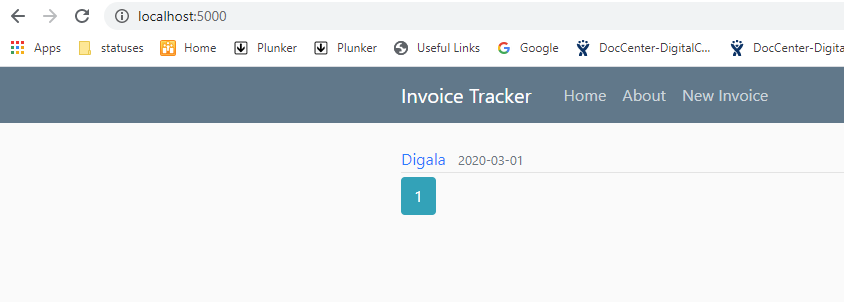
ALTER TABLE public.invoice\_items

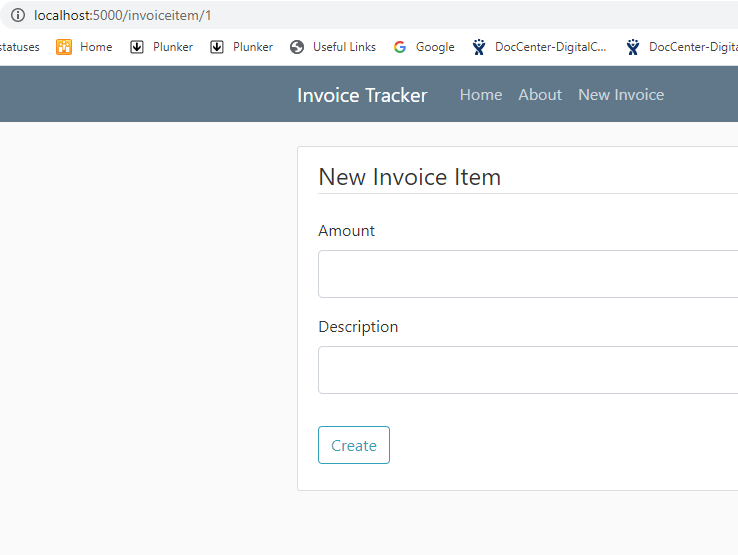
OWNER to postgres;

# Following Screenshots show what you see when you run the application



## Click on the name to navigate to add invoice item





## Thanks,

## Digala