Abdul-THU-5P 4 Assignment 3

Tutor - Abdul Obeid

Jackson Hall – N10504915 Justin Hughes – N10206191 Trần Vũ Nhất Nam – N10306234

Website

Website Access

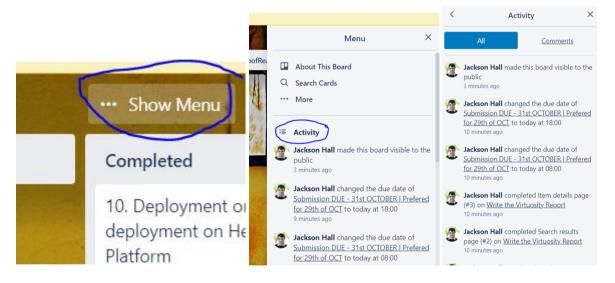
Link - https://virtuosity.herokuapp.com/

Trello

Trello Board Access

Link - https://trello.com/b/Sv4KnJvf/abdul-thu-5p-4-iab207-assignment-3-due-31st-october

Remember to check activities



GitHub

Github Access

Link - https://github.com/HappilyIgnorant/IFB207-Assignment3

Demonstration

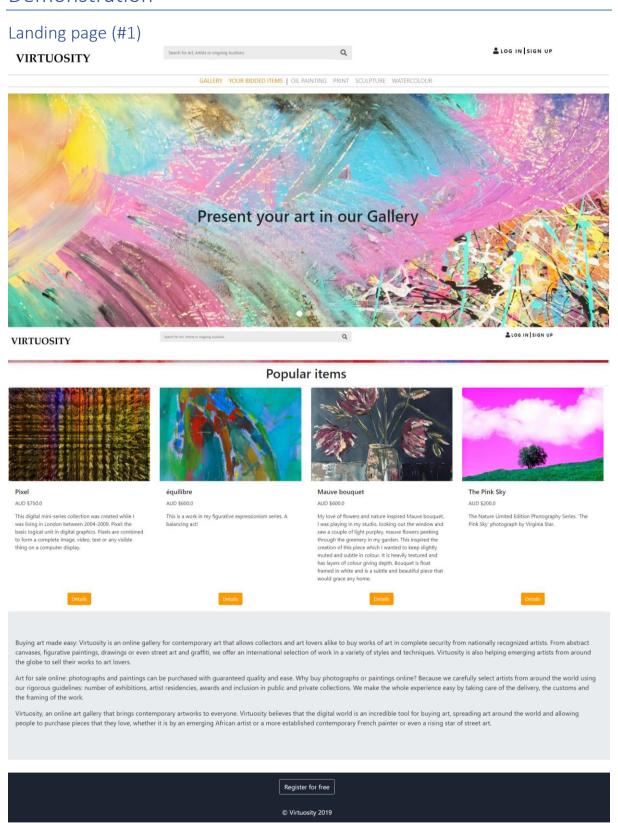


Figure 1 - Main page / Landing

Landing Page Description

This is the initial page that the user first encounters, the layout did change from the conceptual HTML in Assignment 2 to now as we incorporate elements required by the specification.

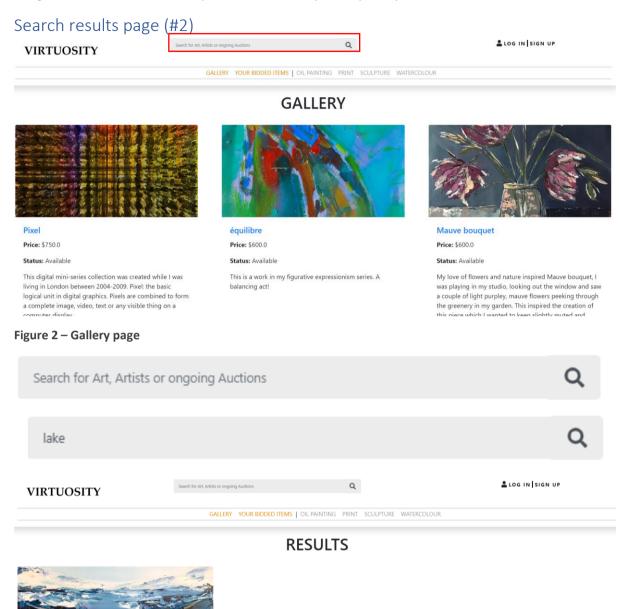


Figure 3 – Results page with Lake as result

Abstract lakes in the south west Australia. The sheer diversity of Western Australia's climate and geography has created some of the most unique rivers, lakes and wetlands on Earth havens to staggering variety of wildlife, and meccas for seekers of watercoort thrills and wildlarners expansioners.

Lake land

Status: SOLD

Search results description

From any page on the website, the search bar function can be used to find any item with a specific name such as Lake land by using either Lake or land in the search bar.

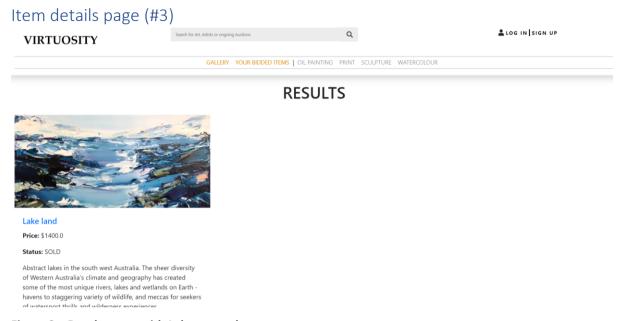


Figure 3 – Results page with Lake as result

RESULTS



Lake land

Price: \$1400.0

Status: SOLD

Abstract lakes in the south west Australia. The sheer diversity of Western Australia's climate and geography has created some of the most unique rivers, lakes and wetlands on Earth - havens to staggering variety of wildlife, and meccas for seekers

Figure 4 – Select the Lake Land link

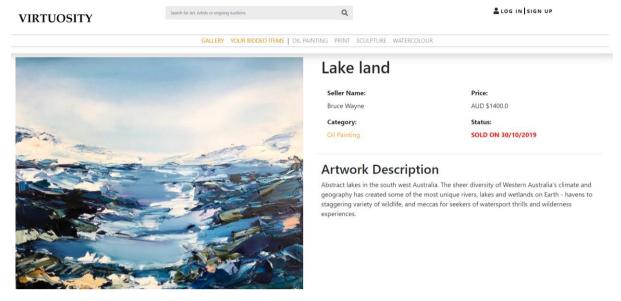


Figure 5 - Items details page Sold

Note this item had been sold on the 30/10/2019 and I am unable to place any bids on this item.

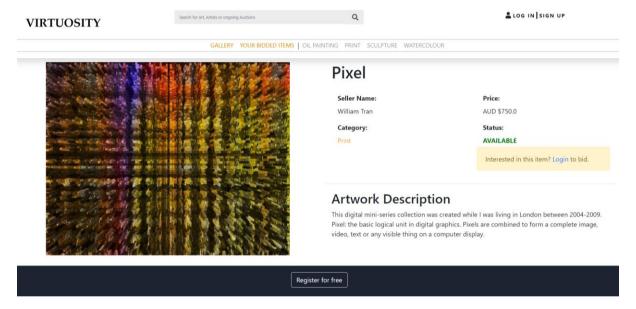


Figure 6 – Items details page Available

This item is currently available for \$750 AUD to be bid, yet the user must log in.

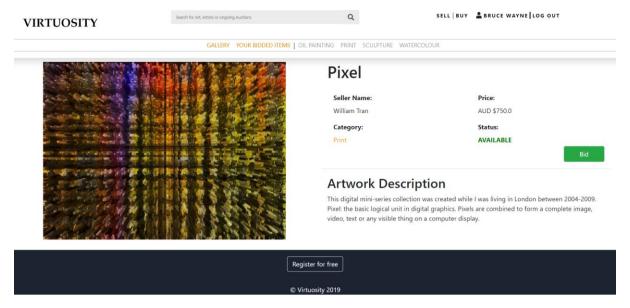


Figure 7 – Items details page Available after logging in

Now the user is able to place a bid, so they select bid and the option to bid changes to show that the user had placed a bid

Status:

AVAILABLE

Bid Placed

Items Details page description



Figure 9 - Any Page / Seller Nav Bar

From here you want to select "YOUR INVENTORY" to get to the seller's page full of items

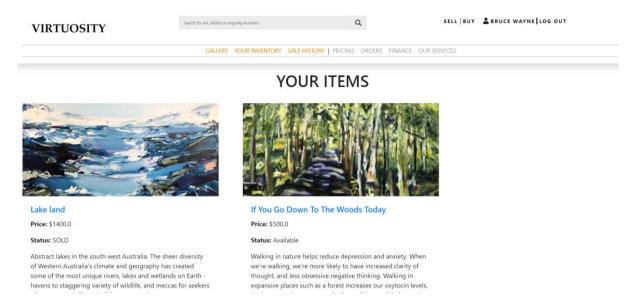


Figure 10 - Your items page

From here you can see two items, one had already been sold "Lake Land" this case we wish to look at "If You Go Down To The Woods Today"

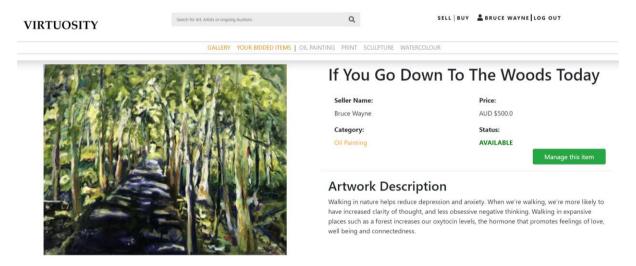


Figure 11 - Item Details for Seller Account

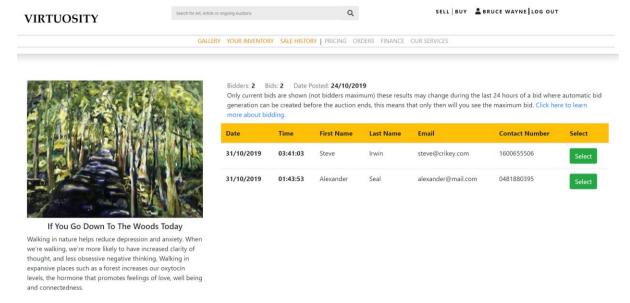


Figure 12 – Manage Item page

From here you can see the information about that product. We can see two bids had already been placed on the item both of which occurred on the same day, following the US time zone as that is where the server is hosted.

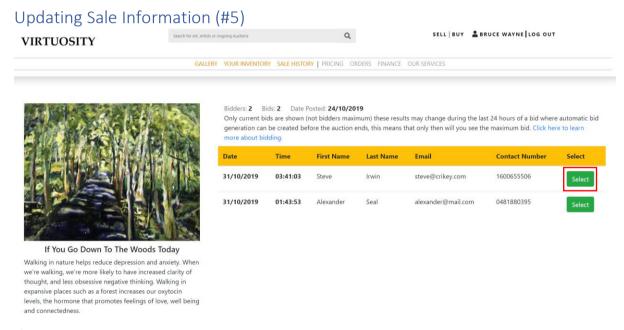


Figure 12 – Manage Item page

This case there is a second bid With Steve Irwin, and I decide he should receive the item due to his bid.

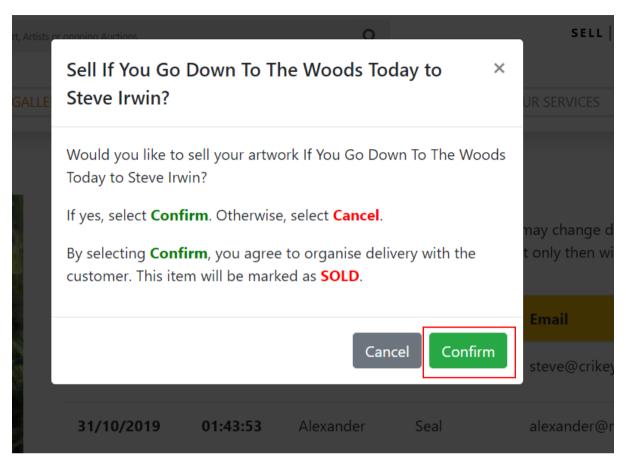


Figure 13 - Bid Confirmation popup

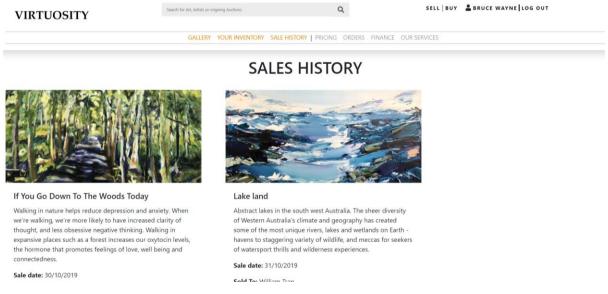


Figure 14 - Sales history

Once Confirming the bid, the user will be taken to the Sales history page as confirmation.

Item creation form (#6)

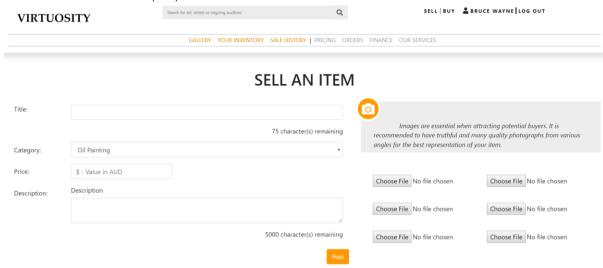


Figure 15 - Item Creation

From here the user is able to create a listing, changing the Title, Category, Listed Price, Description and even selecting a picture to use. Selecting Choose file will open up Windows Explorer.

Choose File No file chosen

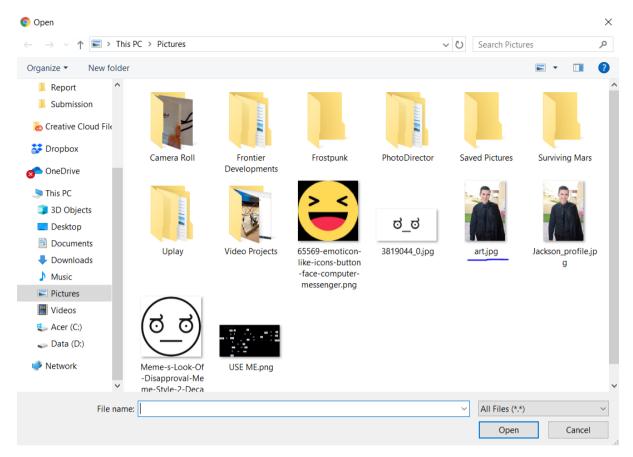


Figure 16 – Windows Explorer

From here I get to choose which image I will use as an example. Here I want to use Art.jpg

Choose File art.jpg

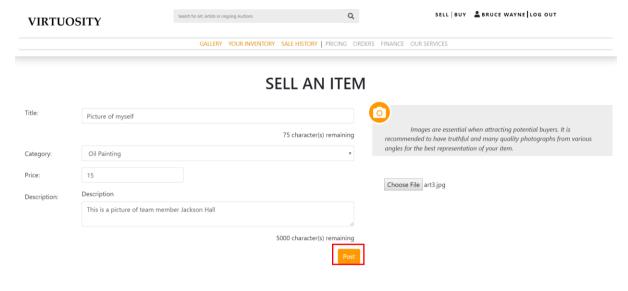


Figure 17 - Item Create

Selecting Post will create the listing.

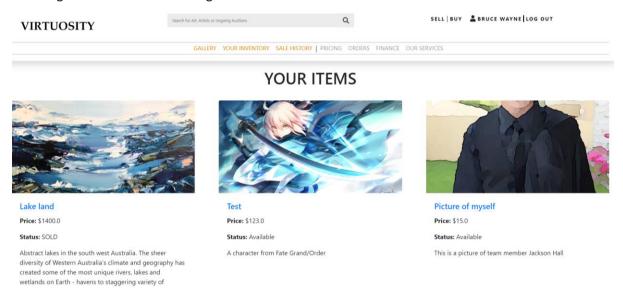
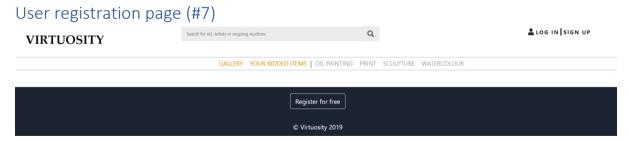


Figure 18 - Your Items page

Here is evidence that the item had been created after the user is redirected to their items page, they can see all of their items and the latest created item, this acts as an additional form of confirmation.



By Selecting SIGN UP on the navbar of every page, this allows the user to create their account. Alternatively using the registration at the footer of the page works as well.

VIRTUOSITY		Search for Art. Artists or ongoing Auctions		Q		♣ LOG I	N SIGN UP
F	Register: First Name Email Address			Last Name			
	Contact Number						
	Password						
	Confirm Password						
	NEXT						
VIRTUOSITY		Search for Art, Artists or ongoing Auctions		Q		≗ LOG	IN SIGN UP
Register:							
r				Last Name			
	First Name Jackson			Last Name Hall			
				Hall			
	Email Address						
	jackson@mail.com						
	Contact Number						
	0412345678911						
	Password						
	••••						
	Confirm Password						
	NEXT						
Register	•						
About you							
My name is Jacks	on and this is a sa	ample account					
							/

Figure 20 – Registration page

This page is where the user is first introduced when creating their account, after they fill in the fields they are taken to the next stage of the carousel is the description which is not mandatory.



password_hash=generate_password_hash(password, method='sha256'))

Figure 21 - Database and code

It must be noted that the password that the user creates on the Registration form is automatically hashed and stored within the database, this means it becomes unreadable to those with access to the database.

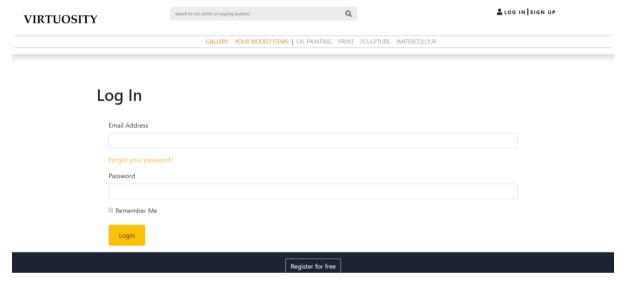


Figure 22 – Login page

After the user creates their account they are taken to the login page where they can then sign in

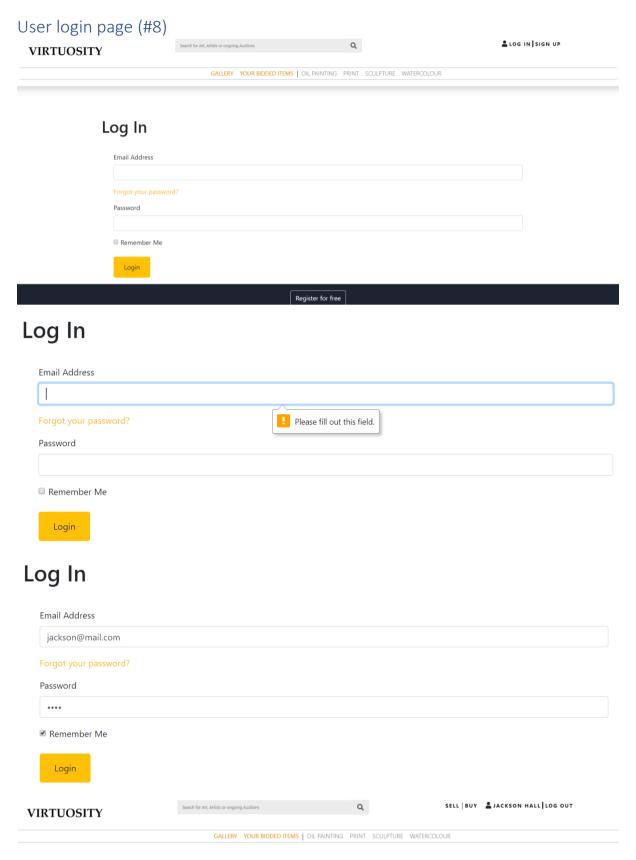


Figure 22 – Login page

From here after the user puts in their credentials, they should be able to log in, if they fail or miss a field they will be notified. When they successfully log in they will notice how their navigation bar changes to display their name and two new options of SELL and BUY

Seller past sales page (#9)

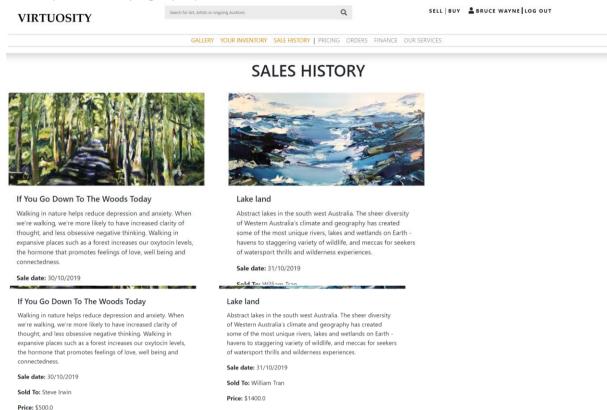


Figure 14 – Sales History

On this page we can see which items had been sold in the past to who and at what price at what date.

Error handling (#10)

```
@app.errorhandler(400)
def bad_request(e):
    return render_template("400.html"),400

@app.errorhandler(403)
def forbidden(e):
    return render_template("403.html"),403

@app.errorhandler(404)
def page_not_found(e):
    return render_template("404.html"),404

@app.errorhandler(410)
def page_deleted(e):
    return render_template("410.html"),410

@app.errorhandler(500)
def internal_error(e):
    return render_template("500.html"),500
```

Figure 23 - Error Handling code

Sample code of what was performed. To gain access to these pages you can /number.

Error 403

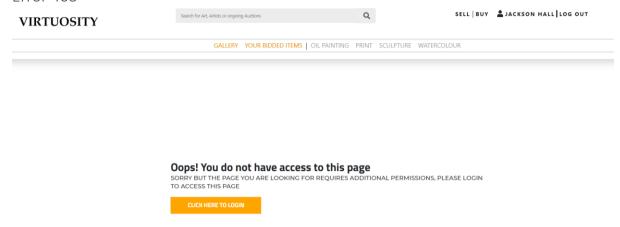


Figure 24 - Error 403

This error can be replicated by searching up any page that does not exist, such as "https://virtuosity.herokuapp.com/sell"

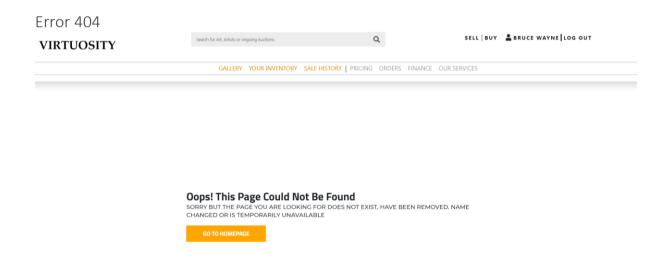


Figure 25 – Error 404

This error can be replicated by searching up any page that does not exist, such as "https://virtuosity.herokuapp.com/badstringfafsaefw23423r"

Error 410

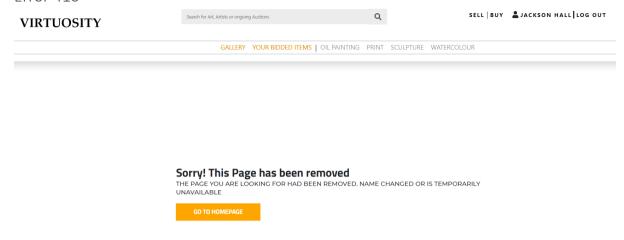


Figure 26 - Error 410

This page can be found when looking for a page that was recently removed

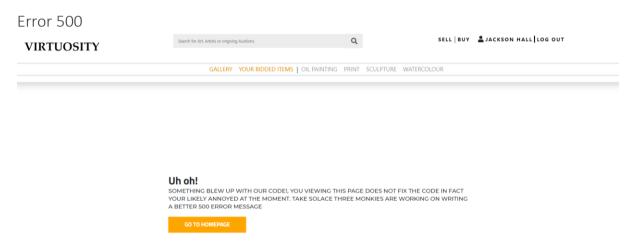


Figure 27 – Error 500

This Error can be replicated by a failure on our behalf.

Heroku Git Integration Setup

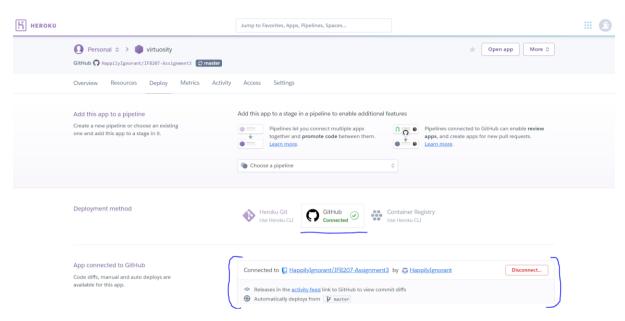


Figure 28 - Setting up the GitHub repository to work with Heroku

To get GitHub to send each commit to Heroku I had to authorize it first, and allow automatic development builds to be set up.

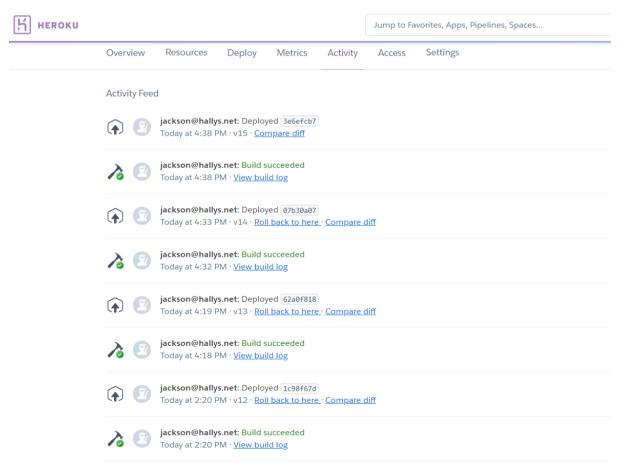


Figure 29

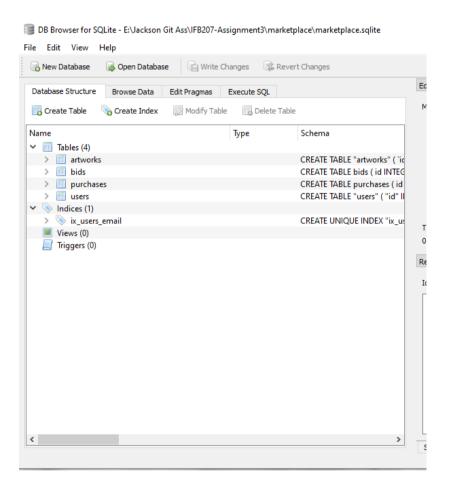
Data Bases and Data Management SQLITE and POSTGRESQL

SQLite

The initial database used during development was a SQLITE database, it was automatically generated by sqlalchemy.

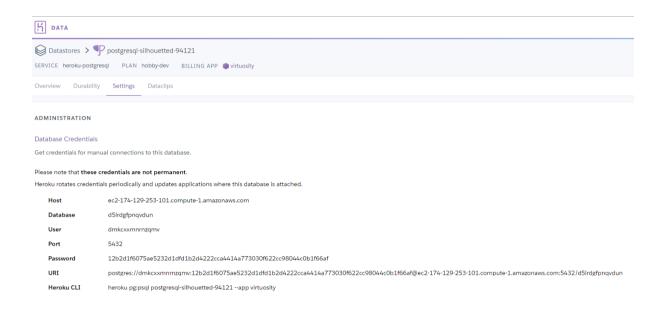
```
Python.exe
from marketplace import db, create_app
db.create all(app=create app())
```

It was very useful during the early stages as it deployed along with the git commits to Heroku



Heroku Postgresql

Once the application was mostly completed the database was migrated to Postgres, the Heroku Postgres database was configured for the app. The credentials for the postgres database were then used to replace the sqlite database in init .py





With the new database URI the database was re-created on Heroku the same way the sqlite database was created

```
Python.exe
from marketplace import db, create_app
db.create_all(app=create_app())
```

Then comes along the task of downloading and installing Postgresql from the main website

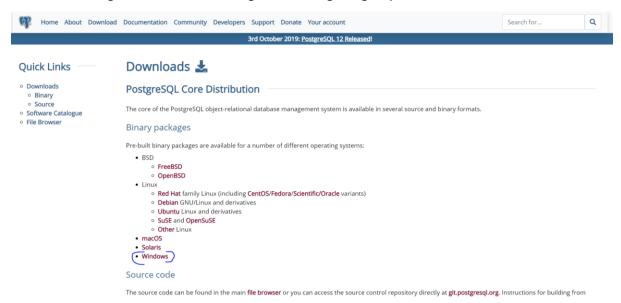


Figure 30 - Download

Here I simply downloaded the windows version of the Core distribution

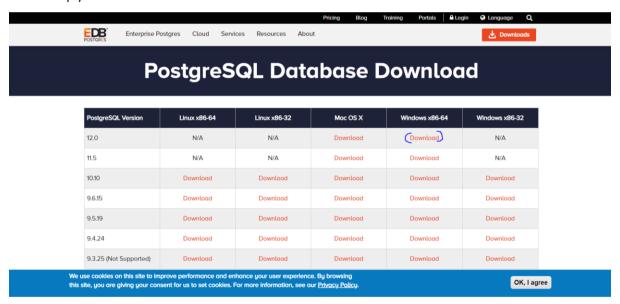


Figure 31

Downloading the latest version of PostfreSQL allows me to use the PGAdmin tool, a database manager for servers, like MySQL used in IBF-105 in Semester 1.

Management

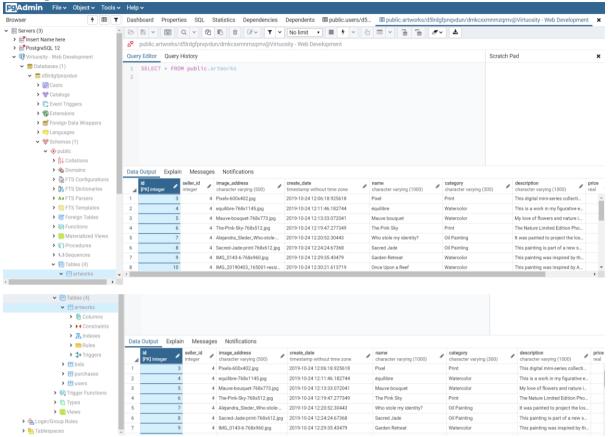


Figure 32 – Using PgAdmin

Utilizing PgAdmin the tools used with this database management tool are identical if not the same to using MySQL after connecting to the Heroku cloud with this software. By doing so I was able to extract the CSV file from SQL Lite and import that into this database.