Contents

[**Creating a Python Web Application Hosted in Azure** 1](#_Toc5879887)

[MS SQL Access From Python 2](#_Toc5879888)

[Create New Azure Machine Learning Studio Project 7](#_Toc5879889)

[REST Api and Using it to serve data for d3.js 10](#_Toc5879890)

[**Sprint Planning (10 of 20)** 14](#_Toc5879891)

Due on the 15th April - this is worth 20% of the final year results - It is exclusively an Individual Project

It is to deliver in two elements

**Technical Proof of Concept (10 of 20)**

Complete the following tutorials from the There are a number of technical skills to demonstrate which need to be trialled for the development of the Group Project.  We have discussed these elements in this Group Project Handbook.

You are required to individually demonstrate to me the following

1. A Python Web Application using the Django web framework.
2. Using the skills acquired accessing a Microsoft database using Python, this web application must have three views accessible from the menu which do the following
   1. Select - list data from a table containing 2 or more records in a tabular format from table TBL\_XXXXX where XXXX is your student id.
   2. Insert - provide a page/view which will insert a new record into this table.
   3. Delete - provide a page/view which will delete a new record into this table
3. Using the skills in connecting to an Azure Machine learning project (deployed as a web service and connecting from your Python Django web application), create an experiment which uses an R Script and data from the table TBL\_XXXXX and returns a summary on the data (for example the average age by role).
4. The final part is to display this input (data from TBL\_XXXXX) and output (summary from R Script) using d3.js.  The visualisation needs to be a simple charting visualisation, for example, a bar chart of role showing average age.

# **Creating a Python Web Application Hosted in Azure**

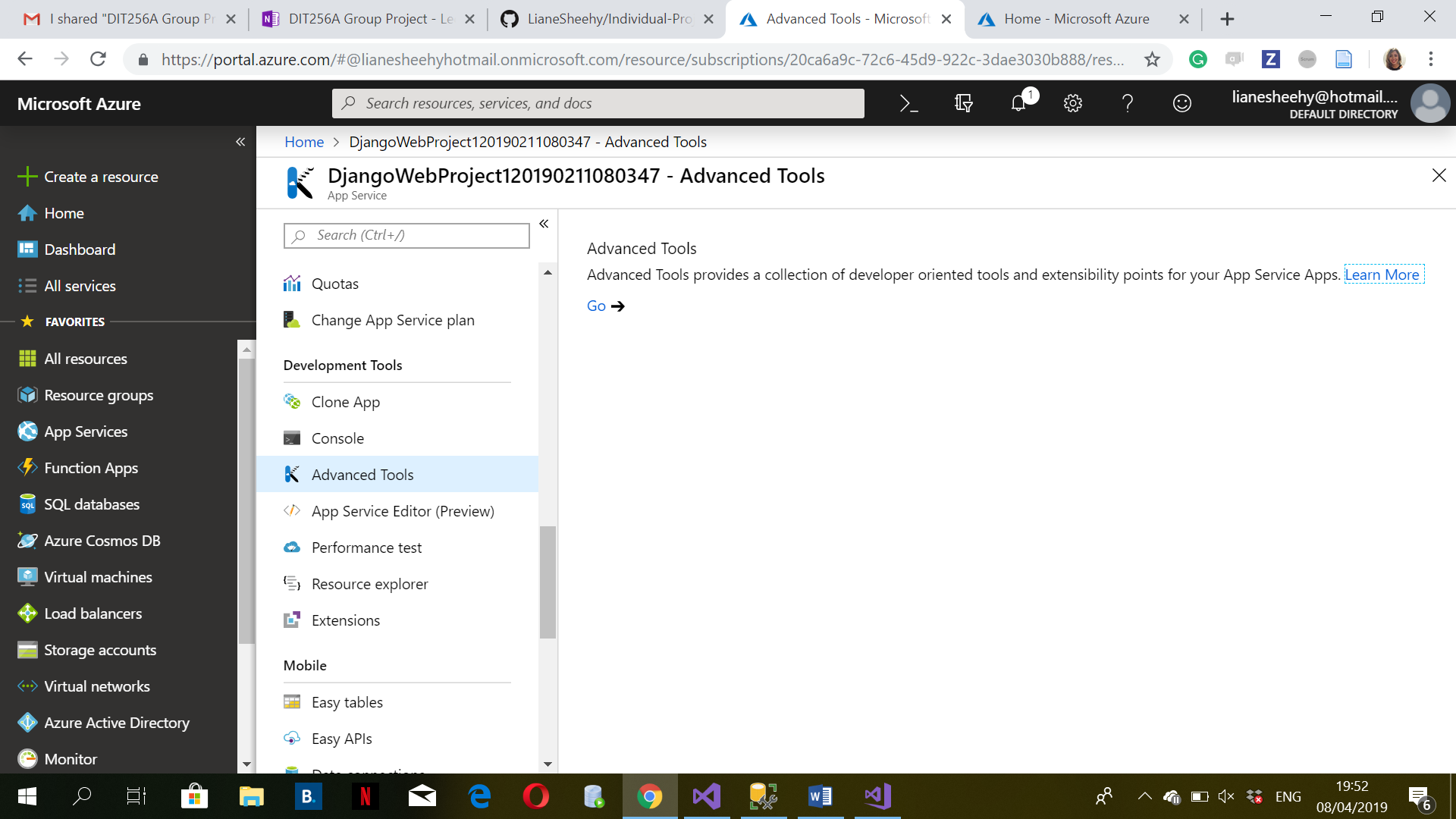
create a Python Web Application using the Django web framework. Done

publish it to Azure Done

Create app Service Done

Publish to App Service - Visual Studio 2017 Done

 Use the Kudu console to upgrade any packages listed in your app's requirements.txt file: navigate to the same Python folder that's used in web.config, such as /home/python361x64, and run the following command as described in the [Kudu console](https://docs.microsoft.com/en-us/visualstudio/python/managing-python-on-azure-app-service?view=vs-2017#azure-app-service-kudu-console) section:



Done

Python Extension Added. Done

Update the web.config . Done

It was including static\web.config which fixes the static files loading problem.  Done

# MS SQL Access From Python

The second tutorial will be accessing a Microsoft database from MS SQL Management Console and Python. Select "Database Engine" and the following dialog will appear. Here are credentials for an Azure hosted database .

|  |  |  |
| --- | --- | --- |
| Server Name |  | natheandemos.database.windows.net |
| Database Name |  | RianProjectDB1 |
| Login Name |  | DIT265A\_USR |
| Password |  | 479M1LZ5Uk |

**Create a table**

Use the following script to create and populate a table with some records.  Replace the **XXXXX** with your own student ID as it's a shared database.

CREATE TABLE TBL\_XXXXX

(

ID INT NOT NULL,

NAME NVARCHAR(50) NOT NULL,

AGE INT NOT NULL,

ROLE NVARCHAR(20) NOT NULL,

NOTES NVARCHAR(200) NULL,

CONSTRAINT PK\_XXXXX PRIMARY KEY (ID)

)

INSERT INTO TBL\_XXXXX

(ID,NAME, AGE, ROLE, NOTES)

VALUES(1, 'John', 50, 'Office Boy',null)

INSERT INTO TBL\_XXXXX

(ID,NAME, AGE, ROLE, NOTES)

VALUES(2, 'Mark', 26, 'Manager',null)

INSERT INTO TBL\_XXXXX

(ID,NAME, AGE, ROLE, NOTES)

VALUES(3, 'Alison', 43, 'Owner',null)

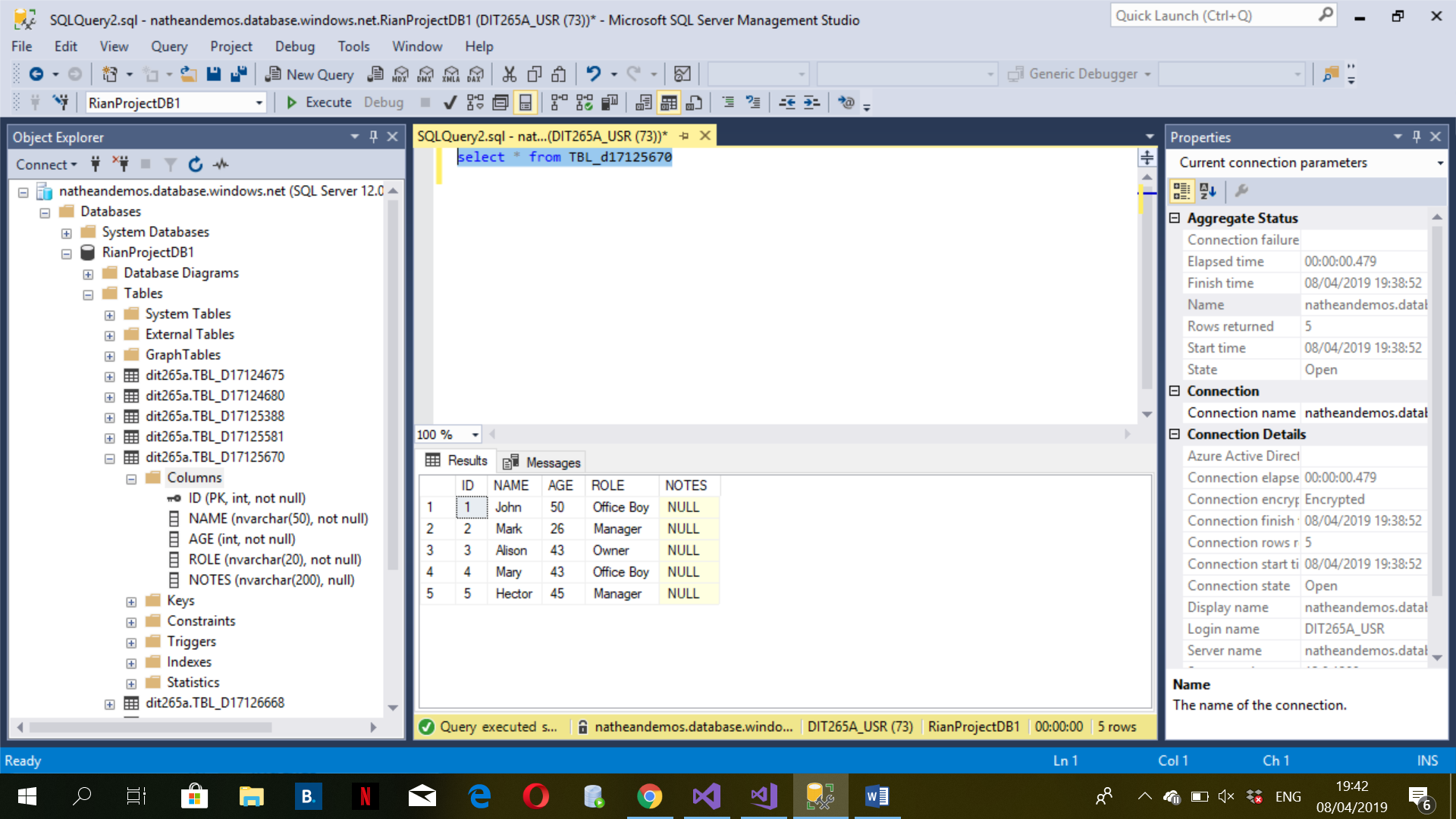
INSERT INTO TBL\_XXXXX

(ID,NAME, AGE, ROLE, NOTES)

VALUES(4, 'Mary', 43, 'Office Boy',null)

INSERT INTO TBL\_XXXXX

(ID,NAME, AGE, ROLE, NOTES) VALUES(5, 'Hector', 45, 'Manager',null)



**Connecting from Python.**  Done

To install the package ‘pyodbc’. Done

Using the skills acquired accessing a Microsoft database using Python, this web application must have three views accessible from the menu which do the following

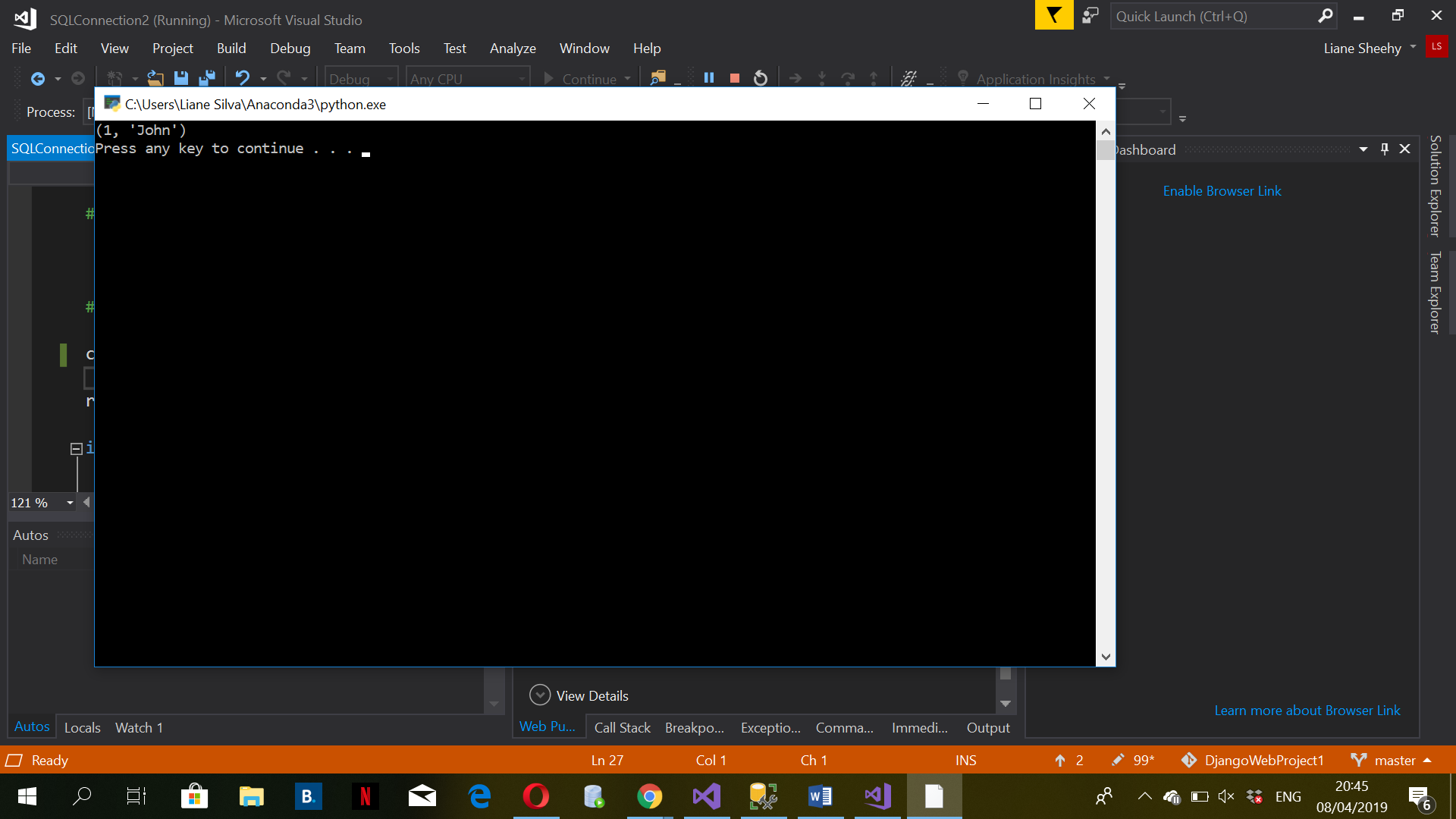
1. Select - list data from a table containing 2 or more records in a tabular format from table TBL\_XXXXX where XXXX is your student id.
2. Insert - provide a page/view which will insert a new record into this table.
3. Delete - provide a page/view which will delete a new record into this table

cursor.execute("select id, name from tbl\_D17125670")

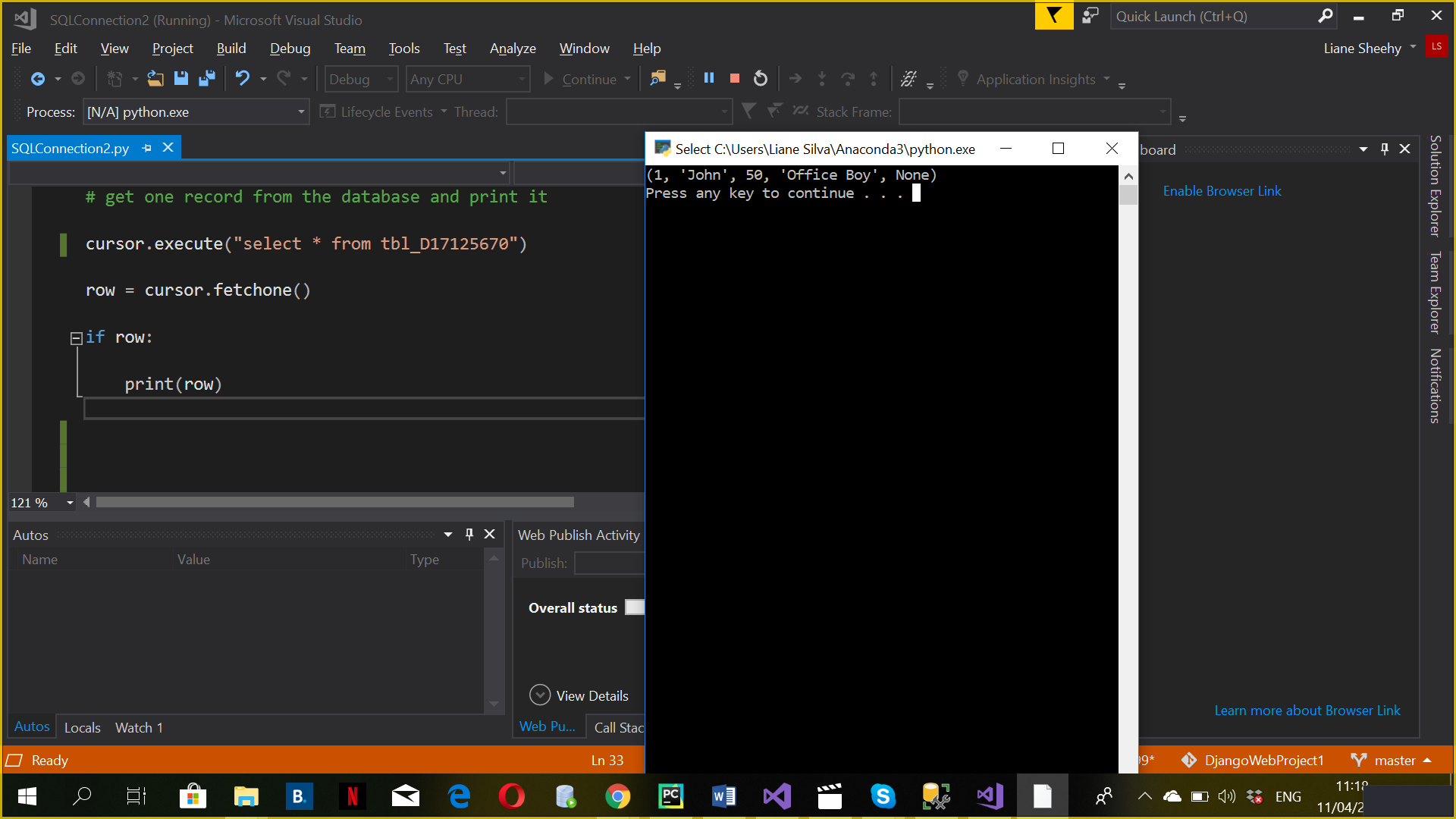
row = cursor.fetchone()

if row:

print(row)



A Select - list data from a table containing 2 or more records in a tabular format from table



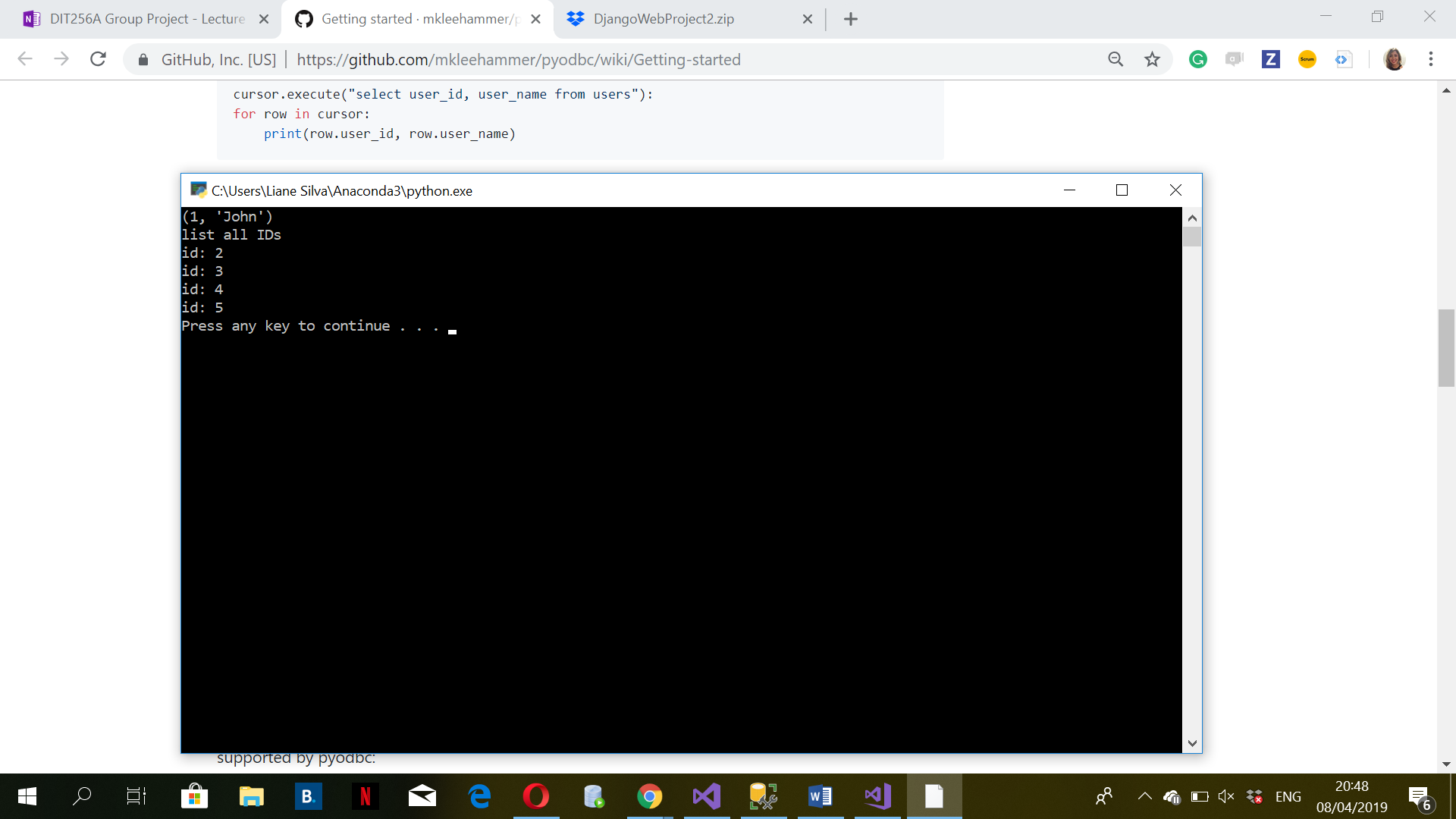
while True:

row = cursor.fetchone()

if not row:

break

print('id:', row.id)



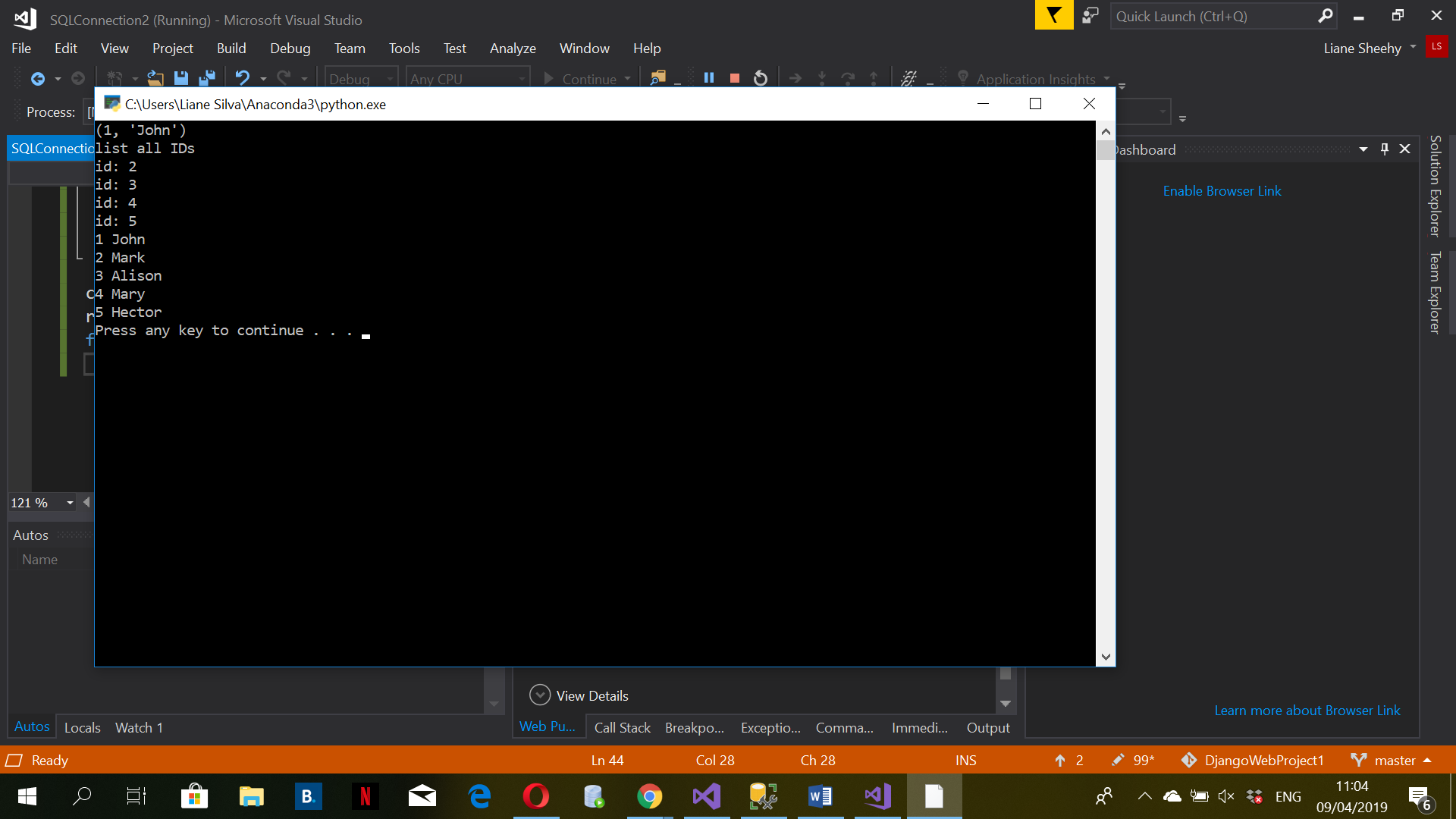
The fetchall() function returns all remaining rows in a list. Bear in mind those rows will all be stored in memory so if there a lot of rows, you may run out of memory. If there are no rows, an empty list is returned.

cursor.execute("select id, name from tbl\_D17125670")

rows = cursor.fetchall()

for row in rows:

print(row.id, row.name)



**INSERTING DATA**

insert into tbl\_D17125670

(ID,NAME, AGE, ROLE, NOTES)

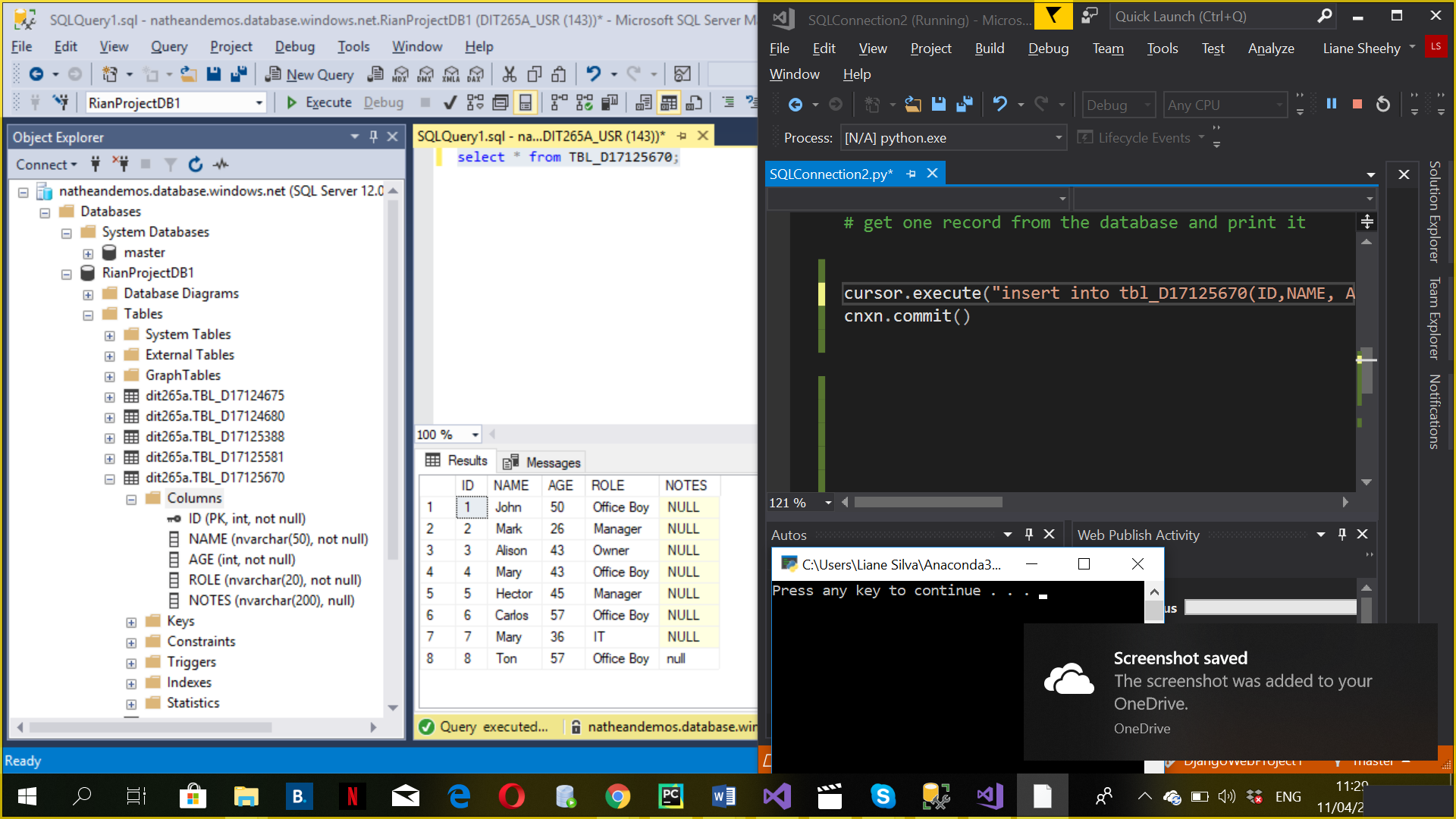
VALUES(6, 'Carlos', 57, 'Office Boy',null)

insert into tbl\_D17125670

(ID,NAME, AGE, ROLE, NOTES)

VALUES(7, 'Mary', 36, 'IT',null)

B - Insert - provide a page/view which will insert a new record into this table.



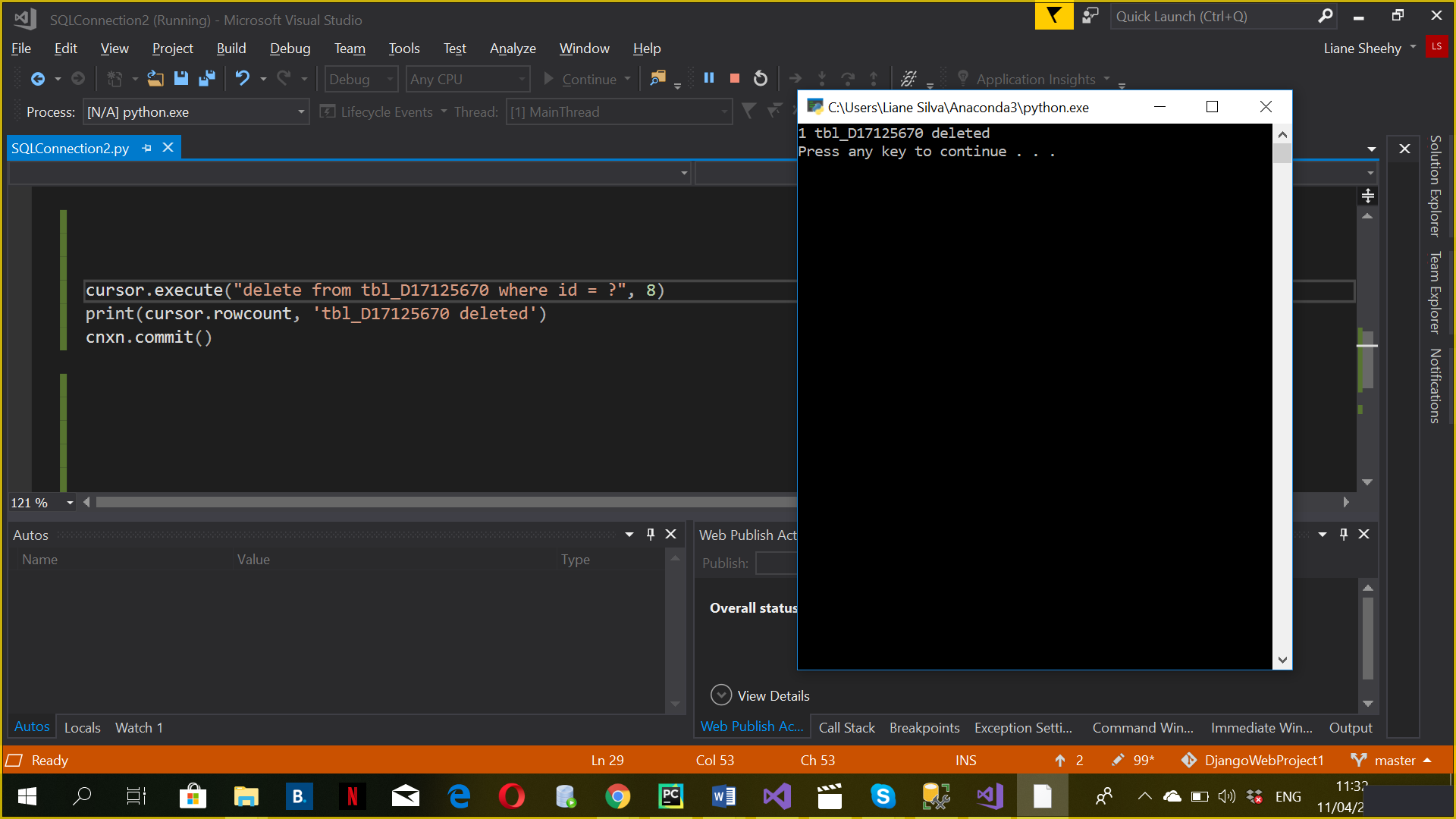
UPDATING AND DELETING

cursor.execute("delete from products where id <> ?", 'pyodbc')

print(cursor.rowcount, 'products deleted')

cnxn.commit()

C - Delete - provide a page/view which will delete a new record into this table

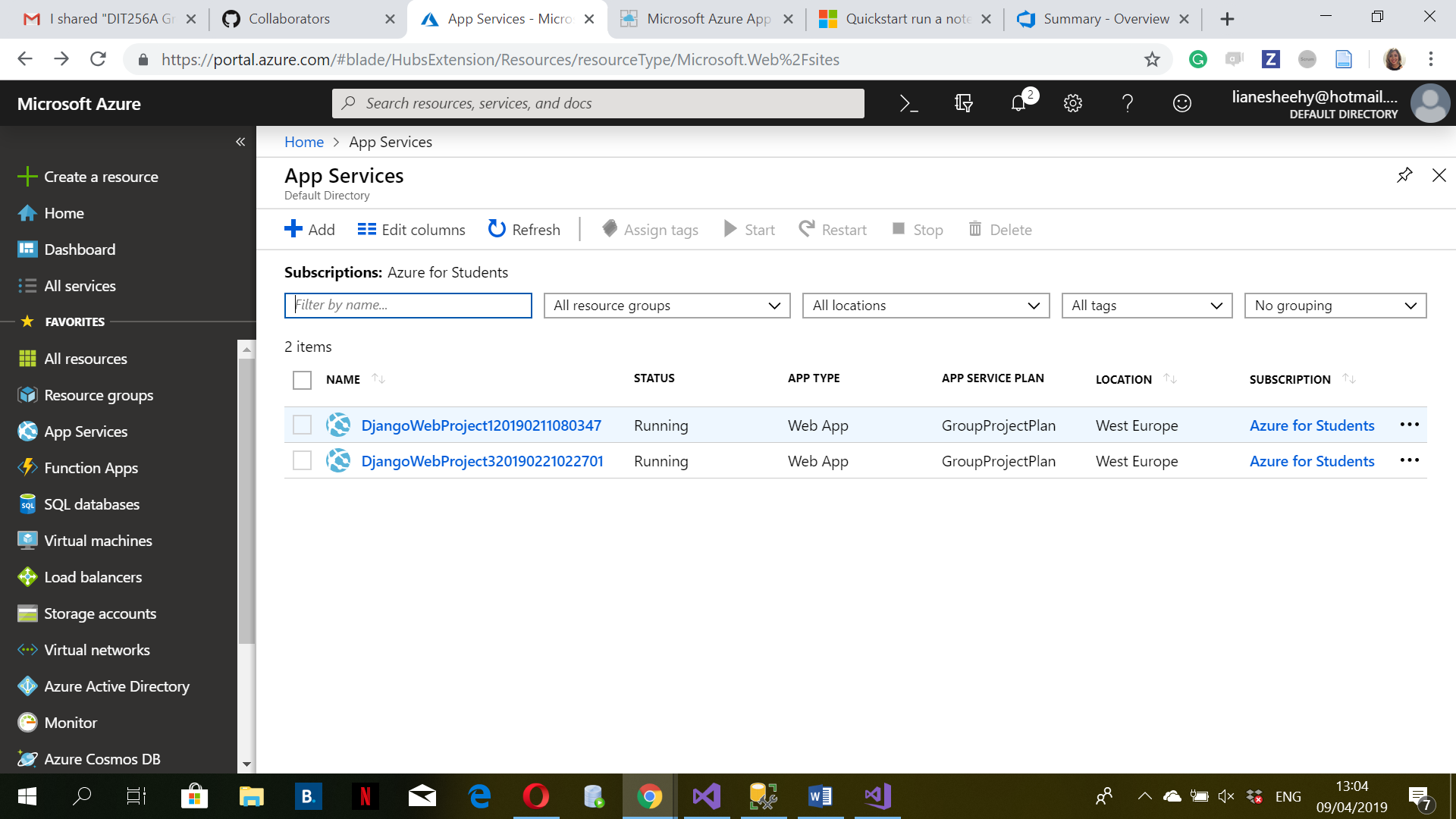


# Create New Azure Machine Learning Studio Project

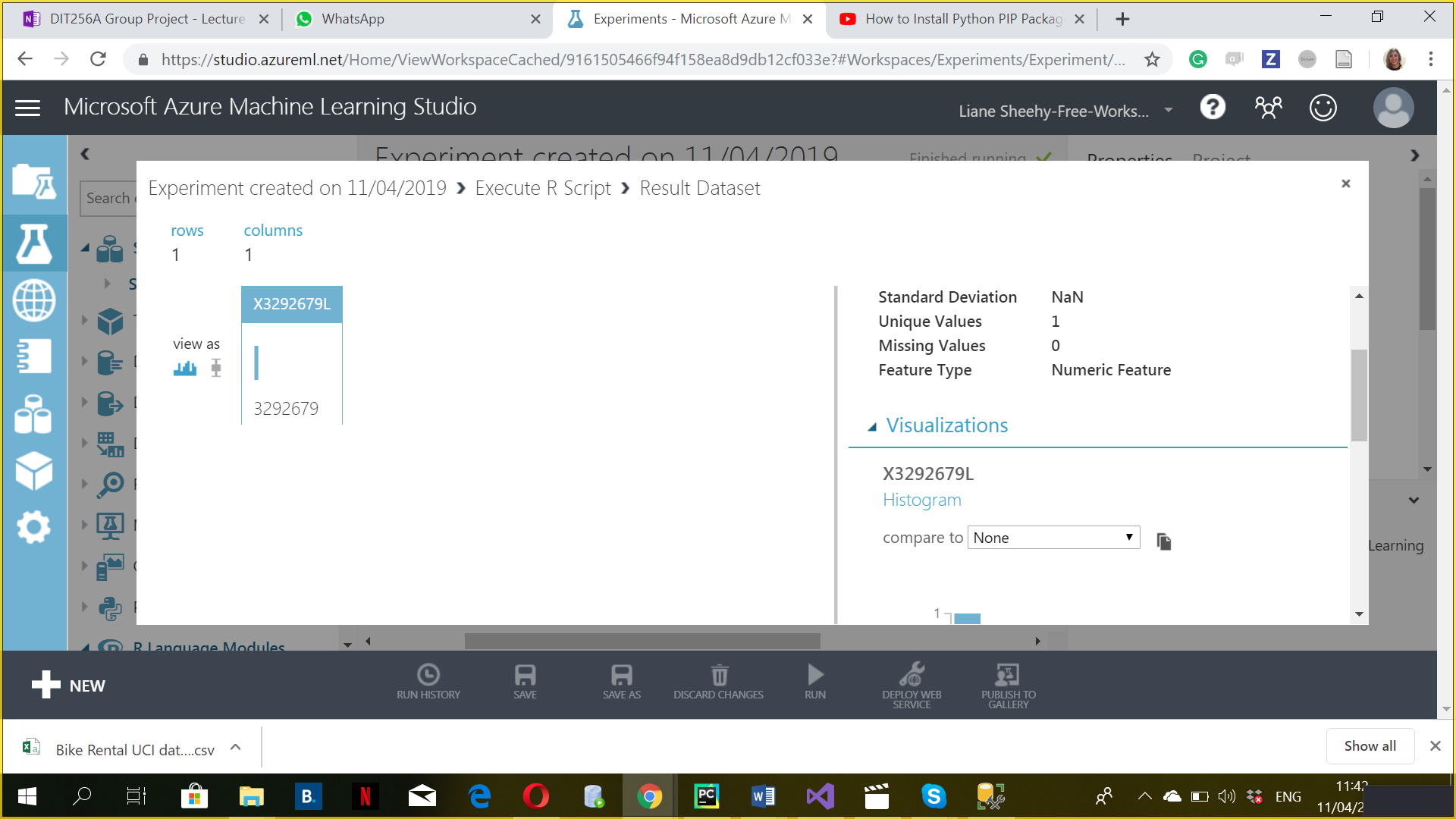
Done

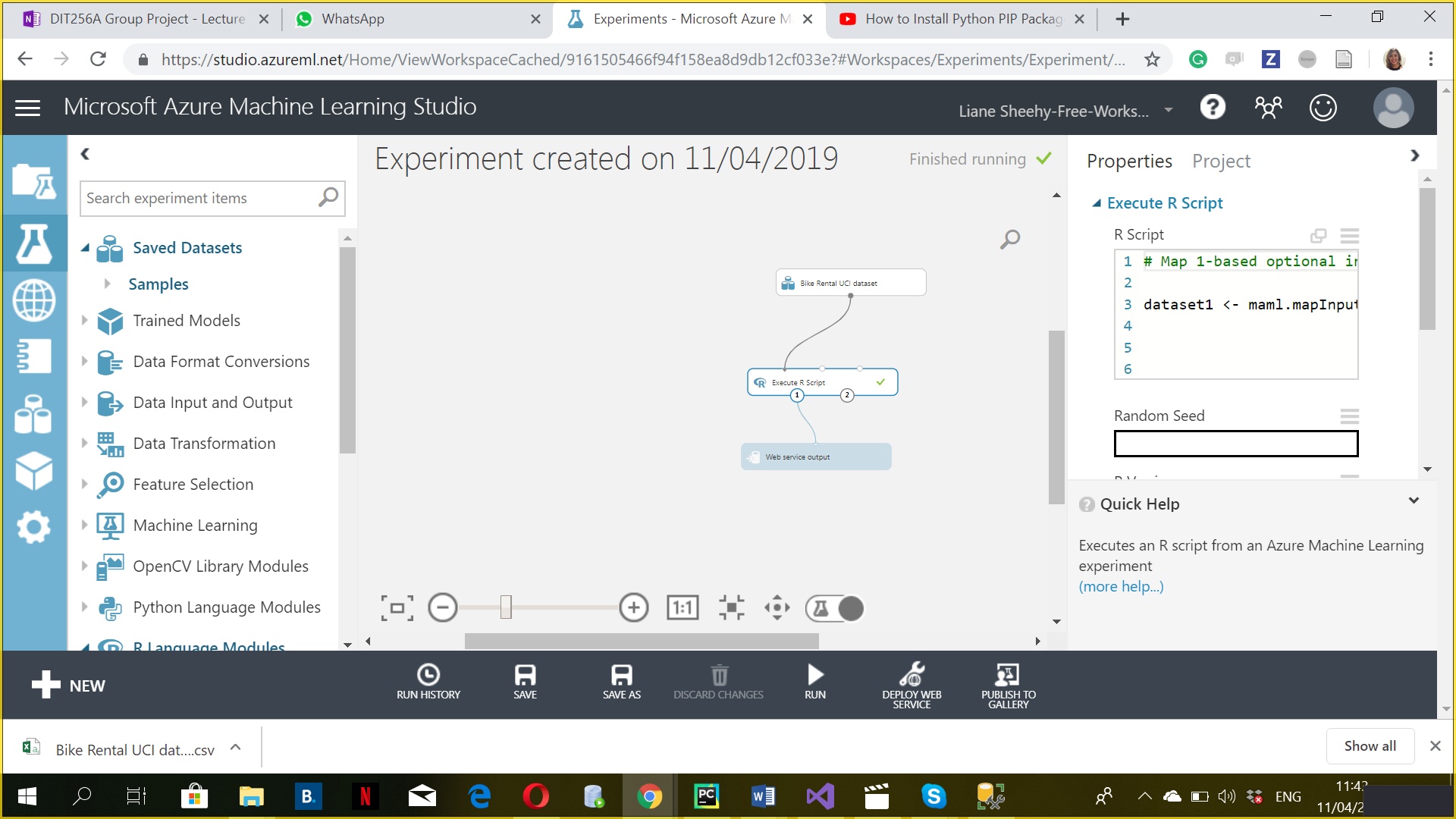
Project created.





Experiments Done.





Add a Web service output to the experiment and publishing it Done.

Converting to a web service. Done

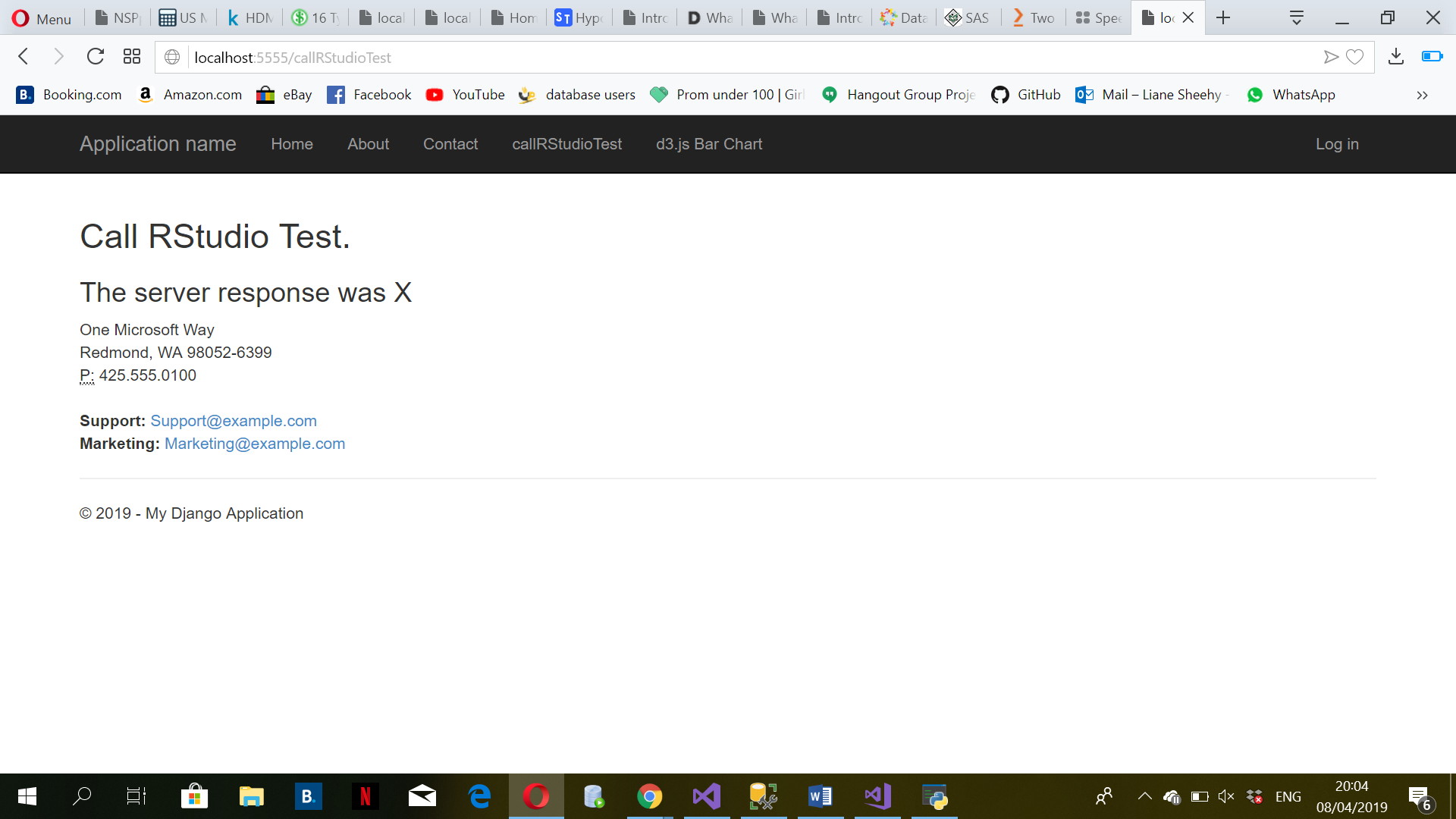
Plugging into Web application Done

Create a new html file in \template\app\ called CallRStudioTest.html  Done

**Create a new function in views.py** Done

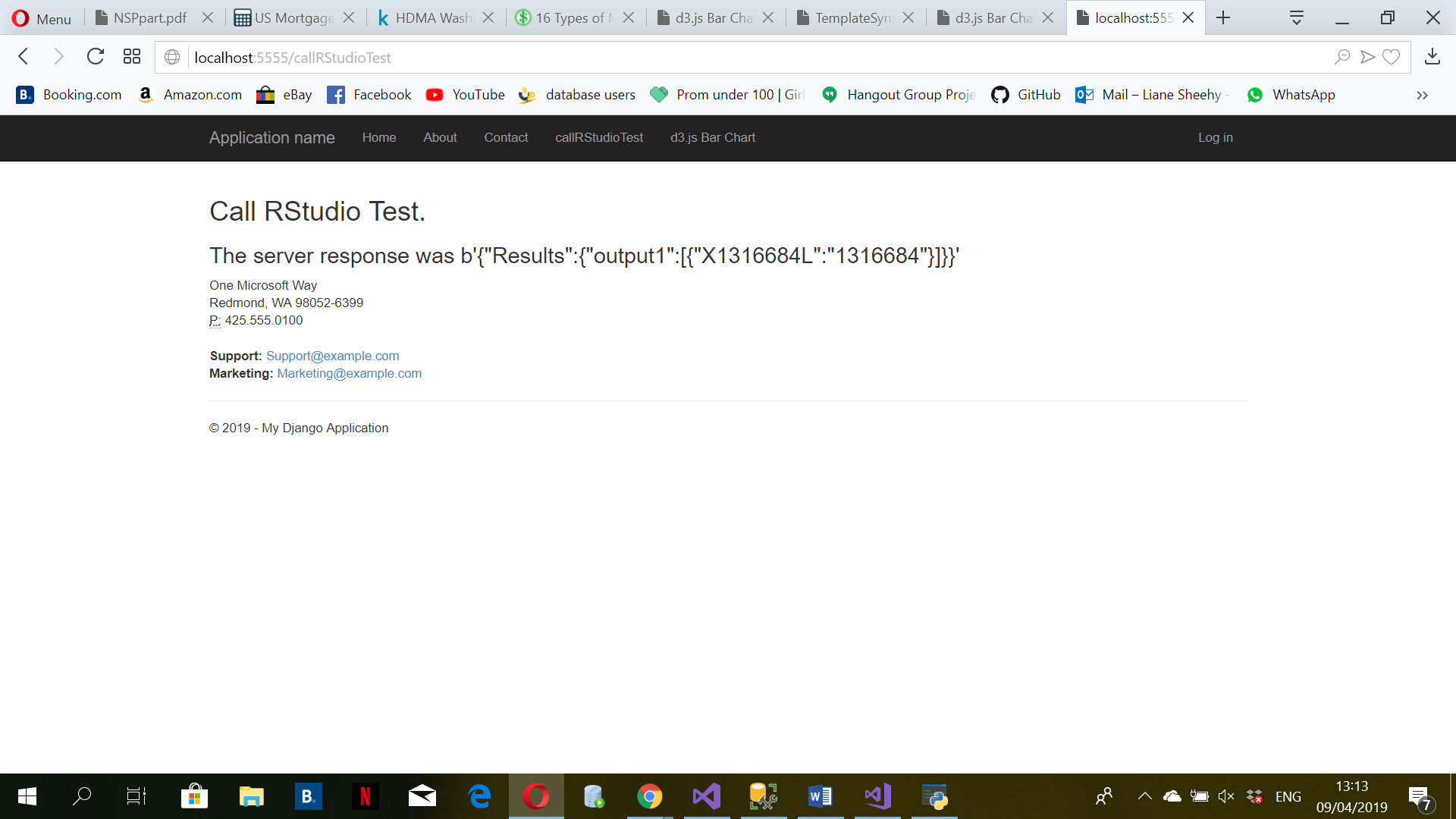
**Add a new item in the layout menu (layout.html)** Done

**Add new url patterm to urls.py** Done



Now connect to out experiment in Rstudio

Replace the view function with this  Done



# REST Api and Using it to serve data for d3.js

**Create a new menu option for displaying our d3.js bar chart**

pip install djangorestframework   
pip install markdown       # Markdown support for the browsable API.   
pip install django-filter  # Filtering support  Done

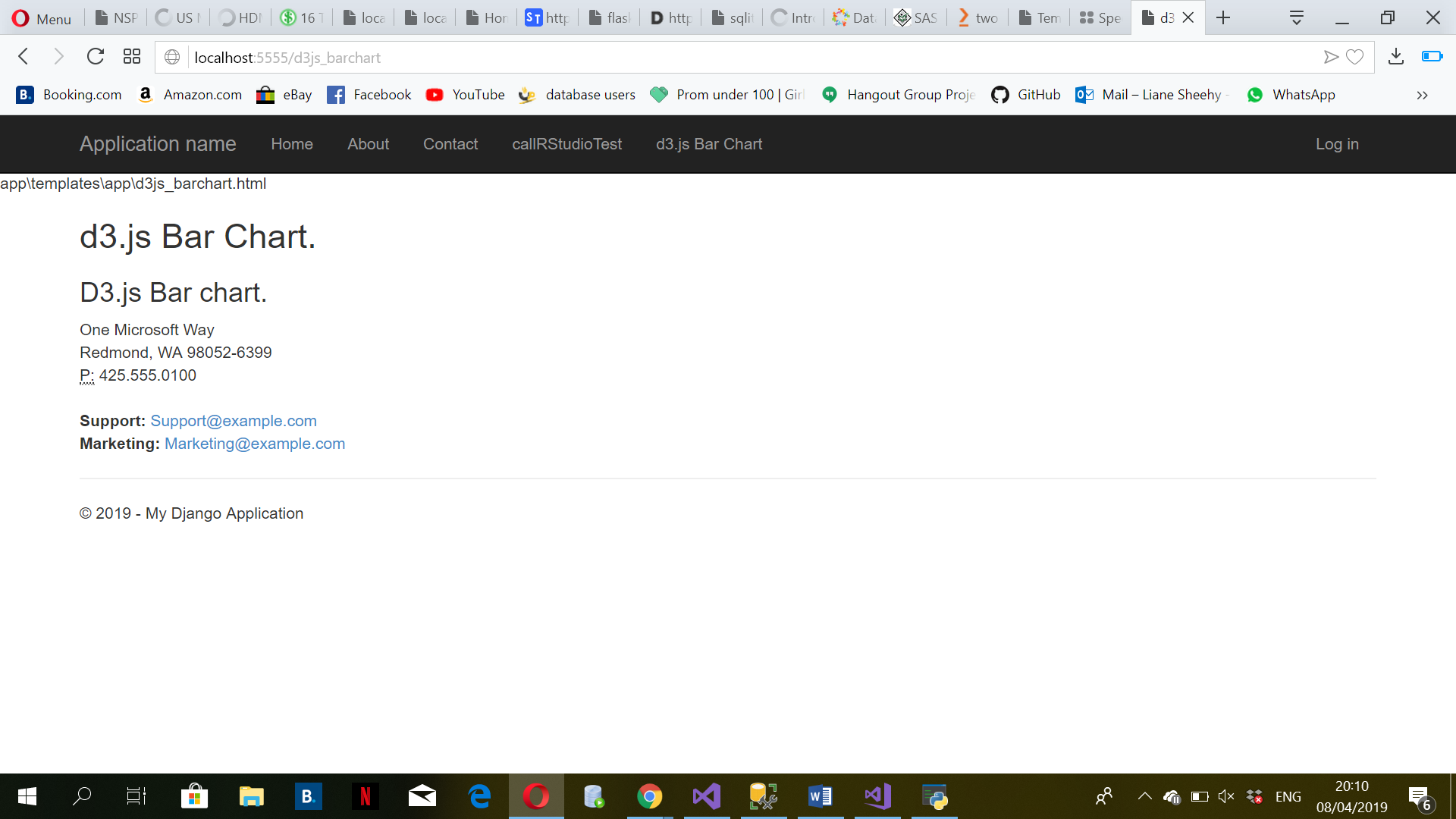
Code Added to views.py Done

**Now create a HTML Template file** Done

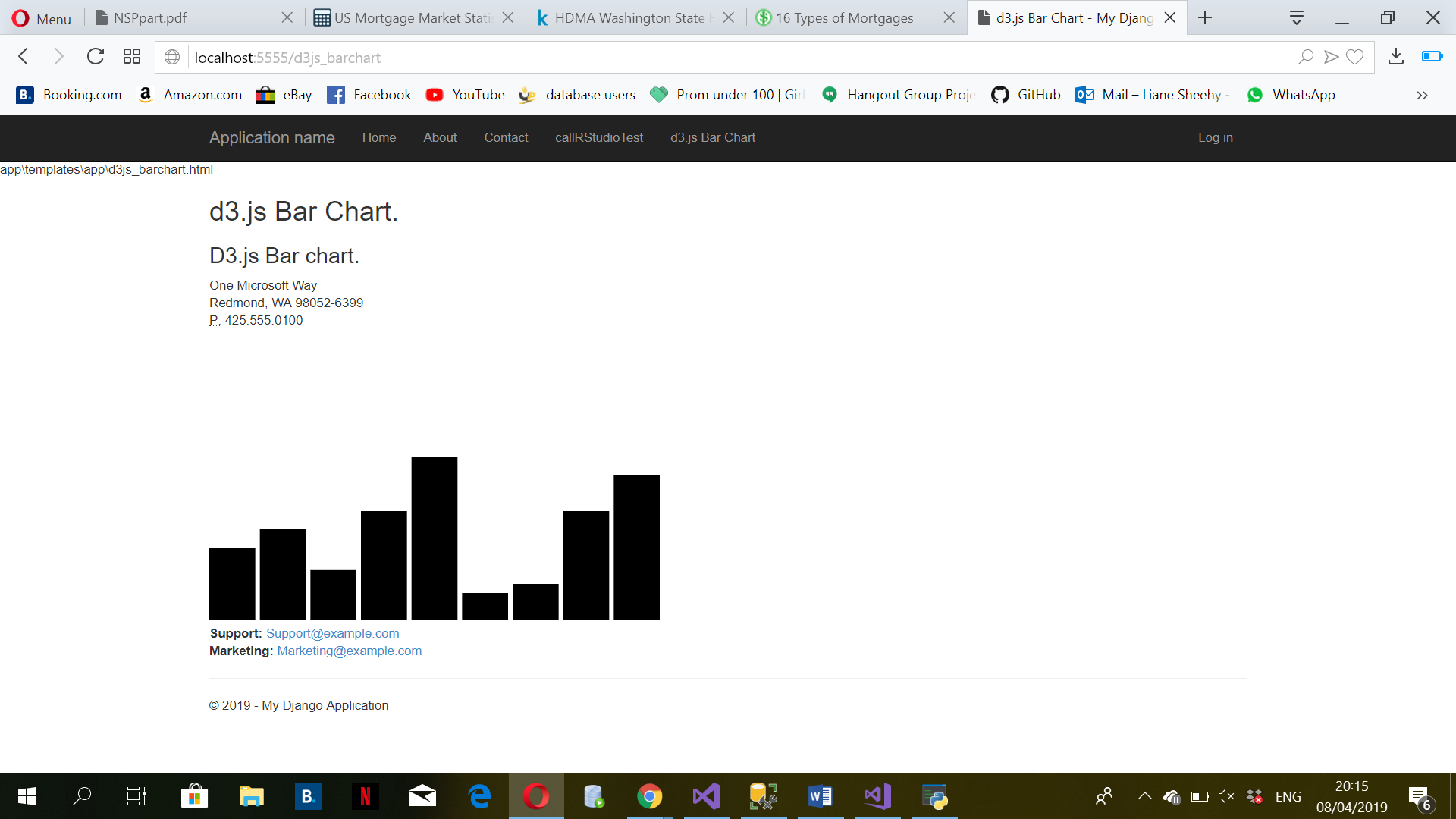
Code Added to **urls.py** Done

**Create a new menu option for displaying our d3.js bar chart**

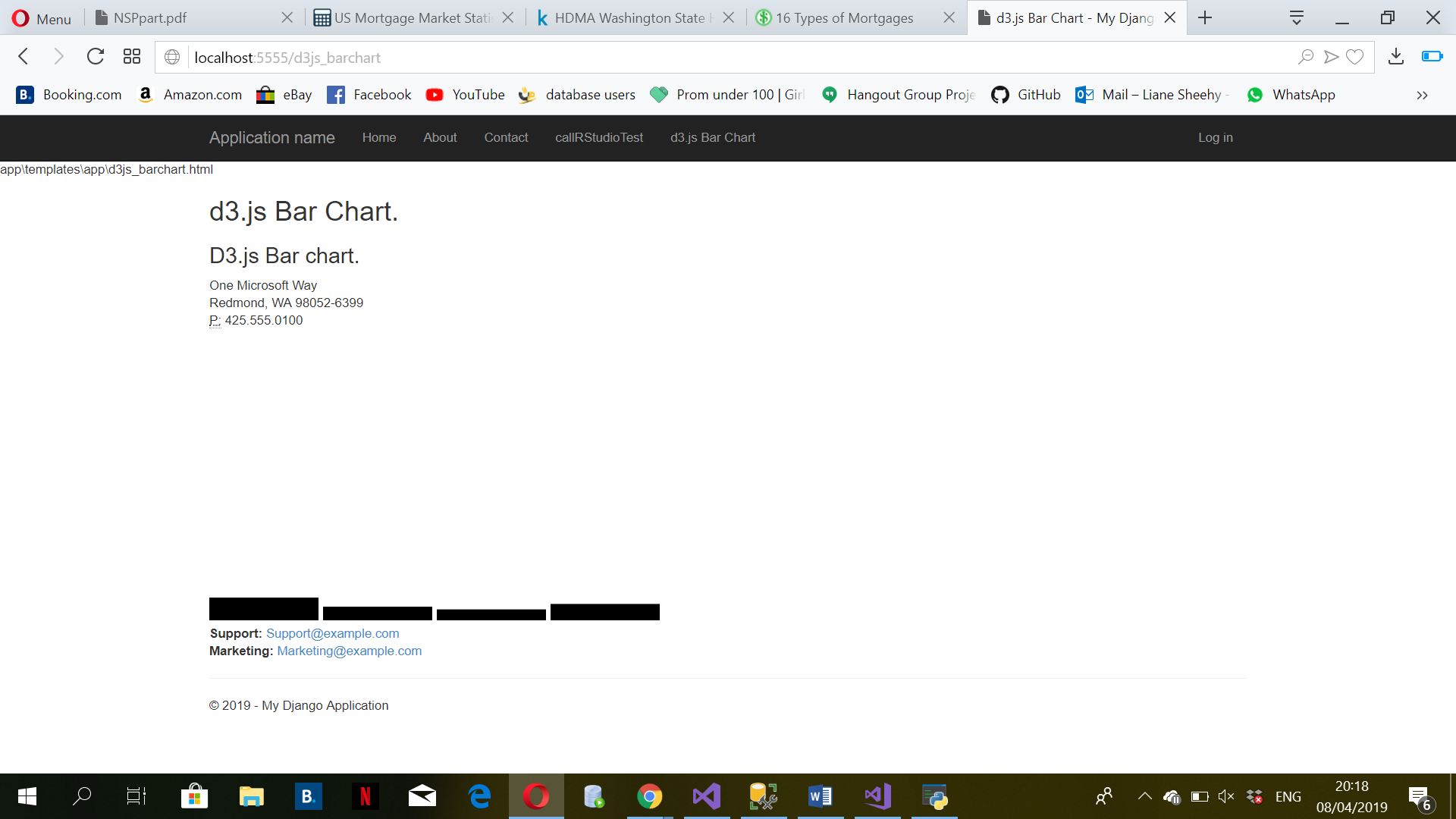
**Now create a HTML Template file**

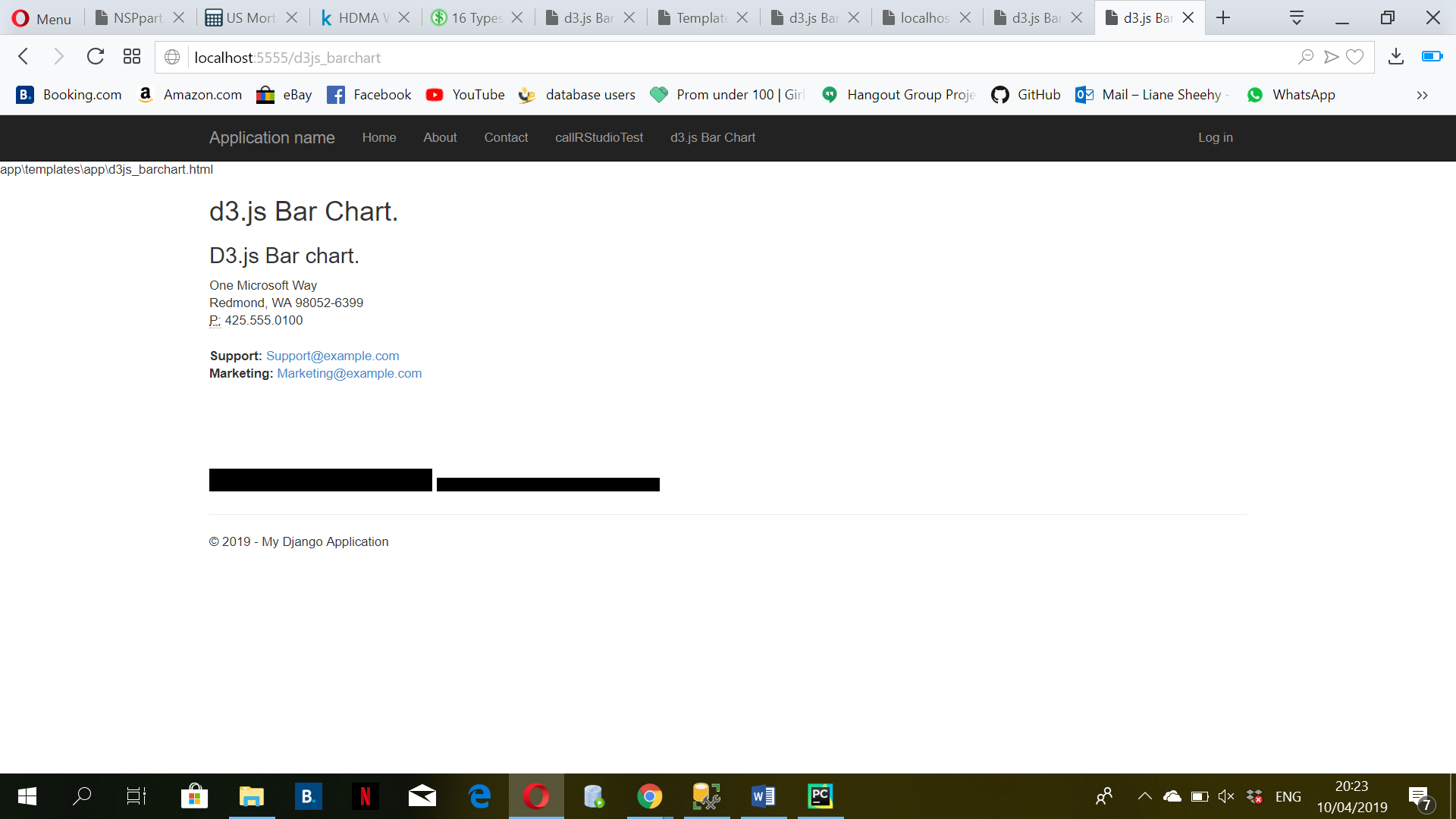


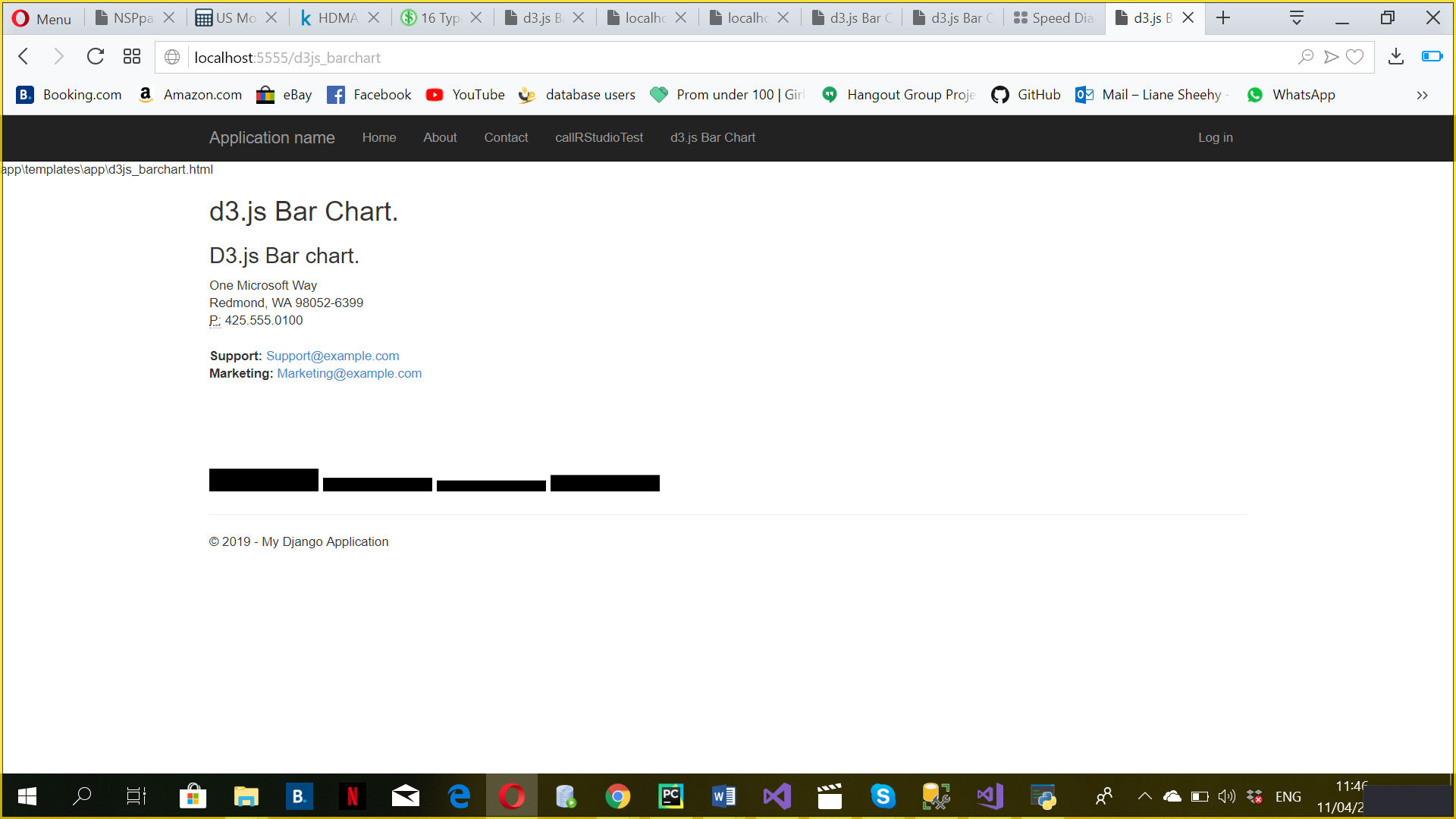
**Now add in barchart d3.js code to our template html page**



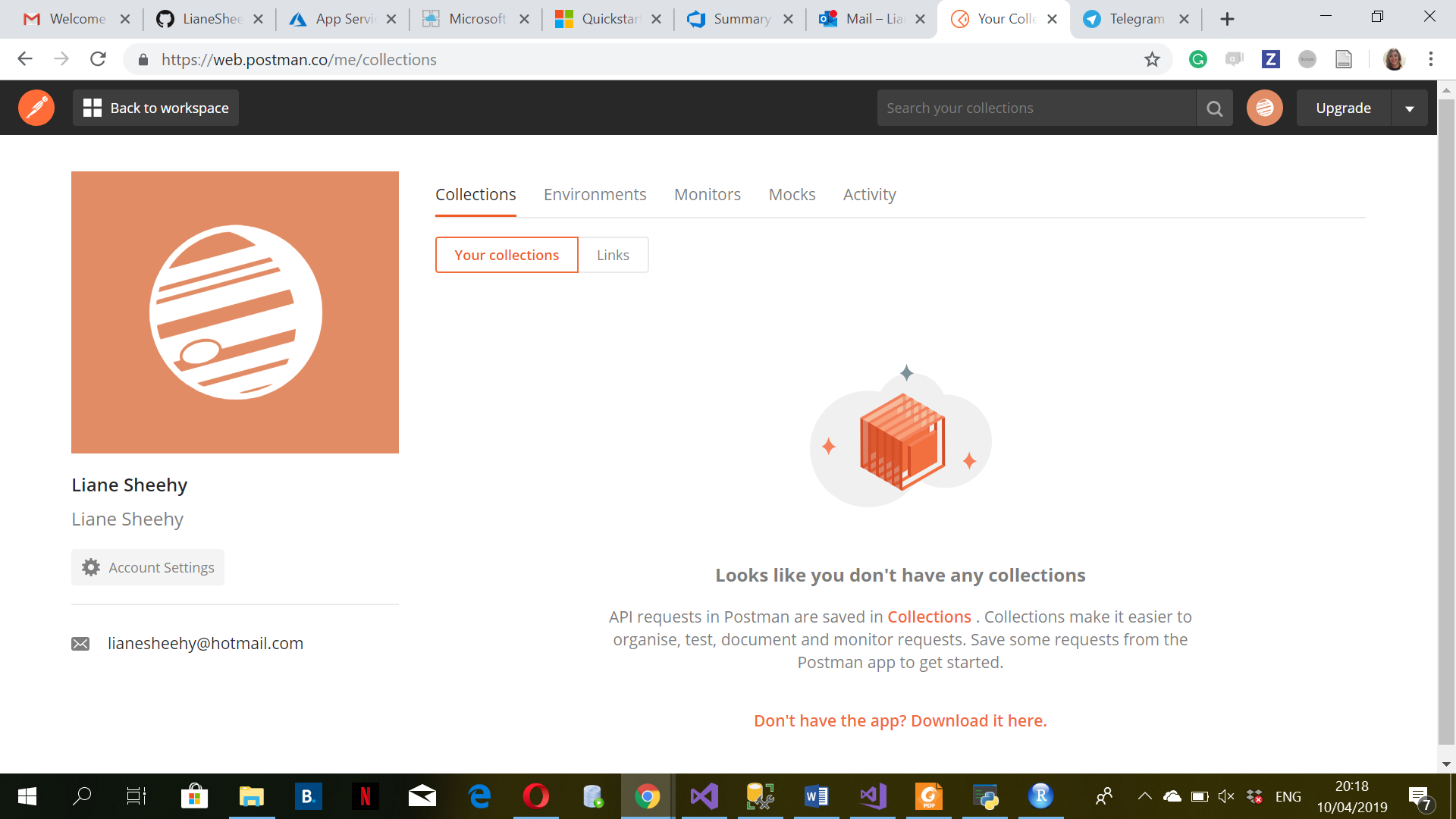
Use an object with multiple fields for the dataset







Post man account Done



# **Sprint Planning (10 of 20)**

Using the POC as the sprint goal, you need to provide a sprint plan hosted on line in one of the devops tools which outlines the following elements

* Sprint Goal: This refers to what can be delivered during the sprint.
* Sprint Backlog: The list of tasks to be completed during the sprint to achieve that goal.

The tasks should relate to the POC

We'll be using US Census data as it's available and detailed.

There are a number of tasks in relation to using this census data

1. Understand the structure of the files provided.  They are summarised data but specific criteria so for example you can find the number of individuals in Texas in the  45 to 49 years age bracket separately the number of those of Korean race but not necessarily a summary of those of Korean race between the ages of 45 and 49.  The item codes, eg dpsf0080011 represents Korean race.  You can however get the each of these by State which will provide you the ability to display results by State in a Choropeth visualisation of the data.
2. Identify the elements you require for your analysis.
3. Outline the objective of the R Script you wish to apply to the data.
4. Create routines for extracting this data from these formats into a MS SQL database.  I like to use Excel generating SQL insert statements.
5. WEB UI Design - create mock-up screen layouts which would be used for the Web UI to
   1. filter the data
   2. selecting the R Script you wish to apply
   3. Identify the coding requirements for implementing the server side execution for the request
   4. which will finally render the results in a US map using the Choropeth d3.js assets.

For the purpose of your Assignment, you need to break each of these steps down to actionable tasks.  Using a Devops tool of your choice,

1. Create a new sprint
2. Create items to fill the Sprint Backlog for each of the tasks for each of the items above.
3. Consider the tasks which support the implementation of a web based application which deliver this functionality (POC elements).
4. On each task you should provide sufficient detail that another developer could pick up the task without needing to come back to you for more information - pretend that the developer works on the other side of the world.
5. Provide an estimate for development time on each task

The objective is to identify as many tasks as are needed to implement the above functionality.

**Completed**

Devops tools: GitHub and ZenHub.

Dataset: from US Census – Texas

1 - The purpose of this study is to evaluate the average Texas population by age and sex.

General Population and Housing Characteristics (Population, Age, Sex, Race, Households and Housing, ...)

2 – Sex and Age.

Population, Age, Sex, Race, Households and Housing.

3- done

**4- done**

**5- done**

<https://github.com/LianeSheehy/Individual-Project>

<https://github.com/LianeSheehy/Individual-Project#workspaces/individual-project---us-census-5c8d714aa7f08c1a3af978ac/board?filterLogic=any&repos=173936070>

