1. DBMS Software:

   (a) The database manage software we have chosen is mySQL integrated with Heroku PostgreSQL.

2. Scripts to create the two tables we will be using: Politicians & Users.

```
---Politician Table creation script---
CREATE TABLE Politicians (
     Politician_id serial NOT NULL PRIMARY KEY,
     Name  varchar(250)  NOT NULL,
     Chamber varchar(100) NOT NULL,
     Party  varchar(20) NOT NULL,
     State varchar(50)  NOT NULL,
     Phone_num varchar(15) NOT NULL,
     Twitter varchar(15) NOT NULL
)

--Users Table creation scripts---
CREATE TABLE Users (
     User_ID  serial NOT NULL PRIMARY KEY,
     Email  varchar(150) NOT NULL,
     Encrypted_Password  varchar(256) NOT NULL
)
```

3. Scripts to populate database script with API integration

```
import requests
import os
import json
import psycopg2

def doInsert(conn, name, chamber, party, state, phone, twitter):
    cur = conn.cursor()
    insert = "INSERT INTO politicians (Name, Chamber, Party, State, Phone_num,
        Twitter) VALUES ('%s', '%s', '%s', '%s', '%s', '%s')" %(name, chamber,
        party, state, phone, twitter)
    cur.execute(insert)
    conn.commit()

def populateDB():
    key = os.environ['ProPublicAPIKey']
    senateURL = "https://api.propublica.org/congress/v1/115/
        senate/members.json"
    houseURL = "https://api.propublica.org/congress/v1/115/
```

```
    house/members.json"
senateRaw = requests.get(senateURL, headers = {"X-API-Key": key})
houseRaw = requests.get(houseURL, headers = {"X-API-Key": key})
myconn = psycopg2.connect( host=os.environ['HostName'],
    user=os.environ['UserName'], password=os.environ['password'],
    dbname=os.environ['DataBase'])

if senateRaw.status_code == 200 and houseRaw.status_code == 200:
    senate = json.loads(senateRaw.text)
    chamber = "Senate"
    for politician in senate['results'][0]['members']:
        firstName = politician['first_name']
        lastName = politician['last_name']
        party = politician['party']
        state = politician['state']
        twitter = politician['twitter_account'] # May be None
        phone = politician['phone']

        if twitter == None:
            # Or any other filler for null entries
            twitter = "NULL"
        # Replace with a SQL insert
        fullname = firstName + " " + lastName
        doInsert(myconn, fullname, chamber, party, state, phone, twitter)

    chamber = "House"
    house = json.loads(houseRaw.text)
    for politician in house['results'][0]['members']:
        firstName = politician['first_name']
        lastName = politician['last_name']
        party = politician['party']
        state = politician['state']
        twitter = politician['twitter_account'] # May be None
        phone = politician['phone']
        # Sometimes the district is
        "At-Large", sometimes a district number
        # district = politician['district']

        if twitter == None:
            # Or any other filler for null entries
            twitter = "NULL"
        # Replace with a SQL insert
        fullname = firstName + " " + lastName
        if(fullname == 'Tom O\'Halleran'):
            fullname = 'Tom O Halleran'
```

```
elif(fullname == 'Beto O\'Rourke'):
    fullname = 'Beto O Rourke'
doInsert(myconn, fullname, chamber, party, state, phone, twitter)
```

4. Database Design Diagram

(a)