Essential System Environments to Deploy Project

- Python Script:
    - Required Libraries:
        - pandas: A data manipulation library used for handling data in tabular form.
        - Faker: A library for generating fake data such as names, addresses, and other information.
        - sqlalchemy: A SQL toolkit and Object-Relational Mapping library for Python.
- Bool app
    - IPhone or other IOS device
- Database
    - SQLite 3
- SQL Queries
    - Kaggle

Schema for Database

```
CREATE TABLE IF NOT EXISTS "Friends" (
    "FriendshipID" VARCHAR PRIMARY KEY,
    "Status" BIGINT,
    "Username" VARCHAR REFERENCES UserProfile(Username),
    "TargetedUsername" VARCHAR REFERENCES UserProfile(Username),
    "StartTime" VARCHAR
);
CREATE TABLE IF NOT EXISTS "Post" (
    "PostID" TEXT PRIMARY KEY,
    "Username" VARCHAR REFERENCES UserProfile(Username),
    "Comments" TEXT,
    "Description" TEXT,
    "PostCreatedAt" DATETIME,
    "PostLikes" TEXT
);
CREATE TABLE IF NOT EXISTS "Product" (
    "ProductID" VARCHAR PRIMARY KEY,
    "ProductName" VARCHAR,
    "ProdCategory" VARCHAR,
    "PoductDescription" TEXT,
    "Price" FLOAT
);
CREATE TABLE IF NOT EXISTS "ProductReview" (
    "ProductReviewID" VARCHAR PRIMARY KEY,
    "Username" VARCHAR REFERENCES UserProfile(Username),
    "Comments" TEXT,
    "PRDescription" TEXT,
    "PRCreatedAt" TIMESTAMP,
    "PicorVidID" VARCHAR,
    "PRLikes" INTEGER
);
CREATE TABLE IF NOT EXISTS "UserProfile" (
    "Username" VARCHAR PRIMARY KEY,
    "FName" VARCHAR,
    "LName" VARCHAR,
    "Email" VARCHAR,
    "Phone_number" VARCHAR,
    "Address" VARCHAR,
    "DOB" DATE,
    "PermLevel" INTEGER,
    "Password" VARCHAR,
    "Gender" VARCHAR,
    "Biography" TEXT
);
CREATE TABLE IF NOT EXISTS "UserTimeLine" (
    "UserTimeLineID" VARCHAR PRIMARY KEY,
    "PostID" VARCHAR REFERENCES Post(PostID),
    "VenReviewID" VARCHAR REFERENCES VenueReview(VenReviewID),
    "ProductReviewID" VARCHAR REFERENCES ProductReview(ProductReviewID),
    "Username" VARCHAR REFERENCES UserProfile(Username)
);
CREATE TABLE IF NOT EXISTS "VenueReview" (
    "VenReviewID" VARCHAR PRIMARY KEY,
    "Username" VARCHAR REFERENCES UserProfile(Username),
    "ReviewText" TEXT,
    "PRDescription" TEXT,
    "VRCreatedAt" DATE,
    "PicorVidID" VARCHAR,
    "Likes" INTEGER,
    "Rating" INTEGER,
    "Ven_Name" VARCHAR REFERENCES Venue(VenName),
    "WaitTimeReported" TEXT
);
CREATE TABLE IF NOT EXISTS "Venue" (
    "VenName" VARCHAR,
    "Ven_Phone_number" VARCHAR,
    "Location" VARCHAR,
    "Category" VARCHAR
);
```

https://drive.google.com/file/d/1drO10BMmZq-Lrc_KL6IZJBp0tWOVIOD1/view?usp=sharing

https://drive.google.com/file/d/1AK7Ij2miJgaXm6CtSN12wGpubG6fowIb/view?usp=sharing

We created this database for future use in a side project that a few friends, Aaryan, and I have been working on for a few months. It is an app called Bool where you can view pictures and videos taken at a given venue. At the moment, we only have bars that are in State College and are planning to add the stadium and other restaurants. There will be various columns in our database that will not be necessarily handled with in the database such as the password and wait time. The passwords will be handled using Apple’s own user authentication and the wait time is pulled from an algorithm that calculates the average speed of a line. It does this by taking GPS coordinates and calculating the speed of the line. When an image or video is taken, the image is physically stored on a server then an ID is given back that references that image. From there, all we need to do is pull ID to get the desired image or video.