Testing Flask Project – Zoo Inventory

1. Run through a mock process of gathering requirements, generating user stories to use as a guide for building test cases and development, and write out test cases for what you intend to test and how you expect them to go.
2. App Functionality: We need a Flask app connected to an ElephantSQL database. We need to have two tables created, inserted into, and queried from.
    a. Enclosure Table – This will have two data points
        i. Id (this can be serially generated as enclosures are added
        ii. Enclosure names
    b. Animal Table – This will have 4 – 5 data points
        i. Id (again, this can be serially generated)
        ii. Animal Name
        iii. Qty of Animal
        iv. Enclosure name (this can be explicit or accessed referentially from the enclosure table)
        v. Enclosure_id (foreign key for enclosure table)
3. Because we're focusing on TDD (test driven development) we want to build out our tests prior to building out app functionality.

    Example Tests:

    a. Test DB connection

    b. Test file paths of logging/csv/report files

    c. Testing specific functionality, such as inputs being a particular type

    d. Setting up fixtures or parametrizing with dummy data for lists/tuples of data

4. Once tests are completed and passed (be sure to grab a report of the unit tests and attempt to make them more verbose manually) you can begin to build out the actual functionality and create the variables of SQL statements and endpoints.
5. After DB connection is established and endpoints confirmed to work. Iterate through several test runs with Postman trying different data points and adding in bulk and things of that nature.
6. Write up a test report detailing the unit tests as well as endpoint testing with Postman. Use the Testing – Soft Skills document as a reference throughout the process.