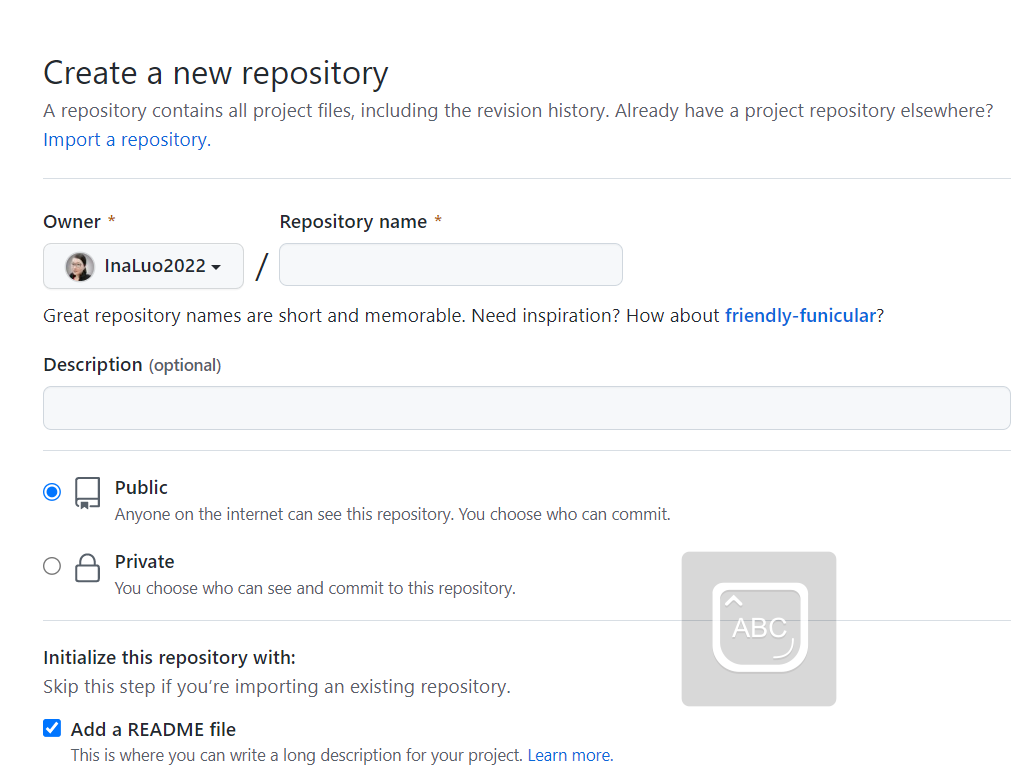
**Before You Begin**

1. Create a new repository for this project called sqlalchemy-challenge. **Do not add this assignment to an existing repository**.

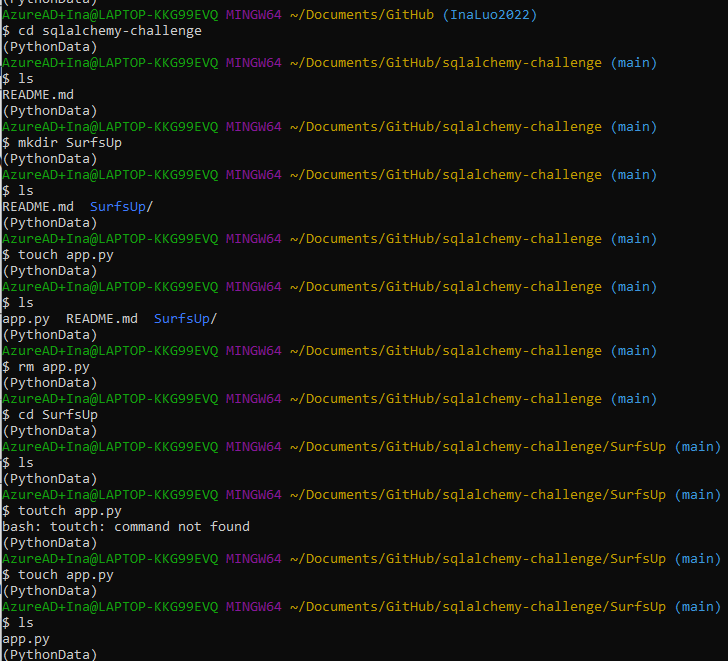


1. Clone the new repository to your computer.

Graphical user interface, text, application, email

Description automatically generated

1. Inside your local Git repository, create a directory for this Challenge. Use a folder name that corresponds to the Challenge, such as SurfsUp.
2. Add your Jupyter notebook and app.py to this folder. They’ll contain the main scripts to run for analysis.
3. Push the changes to GitHub or GitLab.



Text

Description automatically generated

### Files

Download the following files to help you get started:

C:\Users\Ina\Documents\GitHub\sqlalchemy-challenge\Starter\_Code

### Instructions



Congratulations! You've decided to treat yourself to a long holiday vacation in Honolulu, Hawaii. To help with your trip planning, you decide to do a climate analysis of the area. The following sections outline the steps that you need to take to accomplish this task.

#### Part 1: Analyse and Explore the Climate Data

In this section, you’ll use Python and SQLAlchemy to do a basic climate analysis and data exploration of your climate database. Specifically, you’ll use SQLAlchemy ORM queries, Pandas, and Matplotlib. To do so, complete the following steps:

1. Note that you’ll use the provided files (climate\_starter.ipynb and hawaii.sqlite) to complete your climate analysis and data exploration.
2. Use the SQLAlchemy create\_engine() function to connect to your SQLite database.

* create\_engine -> engine.execute(SELECT \* FROM table in sqlite) -> from record in data: print(record)

**show all data but without heading column**

* create\_engine -> conn = engine.connect() -> pd.read\_sql(SELECT \* FROM table in sqlite, conn) -> pd.head()

**show all data in pd dataframe with heading column**

1. Use the SQLAlchemy automap\_base() function to reflect your tables into classes, and then save references to the classes named station and measurement.
2. Link Python to the database by creating a SQLAlchemy session.