Connect to database

```
student_id = "s223128143"
student_first_last_name = "Hoang Long Tran"
print(student_id, student_first_last_name)
```

```
s223128143 Hoang Long Tran
```

```
import firebase_admin
from firebase_admin import credentials

databaseURL = "https://week-5-63814-default-rtdb.firebaseio.com"
cred = credentials.Certificate("credentials.json")
firebase_admin.initialize_app(cred,
                              {'databaseURL':databaseURL})
```

```
<firebase_admin.App at 0x251cfb95cd0>
```

Write to the database

```
from firebase_admin import db

# reference to the root of the DB
ref = db.reference("/")

# json format data (key/value pair)
data = {  # Outer {} contains inner data structure
    "Book1":
    {
        "Title": "The Fellowship of the Ring",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book2":
    {
        "Title": "The Two Towers",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book3":
    {
        "Title": "The Return of the King",
        "Author": "J.R.R. Tolkien",
        "Genre": "Epic fantasy",
        "Price": 100
    },
    "Book4":
    {
        "Title": "Brida",
        "Author": "Paulo Coelho",
        "Genre": "Fiction",
        "Price": 100
    }
}

# overwrite the the reference point at root node
ref.set(data)
```

Read the data

```
# query all data under ref
books = ref.get()
print(books)
print(type(books))
```

```
{'Book1': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The
<class 'dict'>
```

```
# print each item separately
for key, value in books.items():
    print(f"{key}: {value}")
```

```
Book1: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Fell
Book2: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Two
Book3: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Retu
Book4: {'Author': 'Paulo Coelho', 'Genre': 'Fiction', 'Price': 100, 'Title': 'Brida'}
```

```
# Query /Book1
ref = db.reference("/Book1")
books = ref.get()
print(books)
```

```
{'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Fellowship
```

Write the database using `push()`

This method ensures that if multiple writes are being performed under the same key, they do not overwrite each other. The method uses unique keys for each new child added.

```
# Note that a set() is called on top of a push()
ref = db.reference("/")
ref.set({
    "Books":
    {
        "Best_Sellers": -1
    }
})

ref = db.reference("/Books/Best_Sellers")

for key, value in data.items():
    ref.push().set(value)
```

Update data

J.R.R. Tolkien books are reduced to 80 dollars to offer a discount

```
ref = db.reference("/Books/Best_Sellers/")
best_sellers = ref.get()
print(best_sellers)
```

```
{'-O3UyIWByTLMQkYqVjOR': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100,
```

```
for key, value in best_sellers.items():
    if(value["Author"] == "J.R.R. Tolkien"):
        ref.child(key).update({"Price":80})
```

Delete data

Delete all best seller books with J.R.R. Tolkien as the author

```
ref = db.reference("/Books/Best_Sellers")

for key, value in best_sellers.items():
    if(value["Author"] == "J.R.R. Tolkien"):
        ref.child(key).set({})


# If "ref.child" is not used, all the data will be removed. Like the code below
# ref = db.reference("/Books/Best_Sellers")
# ref.set({})
```