In [1]:
```python
# Fill in student ID and name
#
student_id = "221435713"
student_first_last_name = "Anish Bansal"
print(student_id, student_first_last_name)
```

```
221435713 Anish Bansal
```

In [2]:
```python
"""
        Firebase Realtime database demonstration.

    @Ahsan Habib
    School of IT,
    Deakin University, Australia.
"""

# Install libraris, if not yet.
! pip install firebase_admin pandas

import firebase_admin

databaseURL = 'https://sit225-f035c-default-rtdb.asia-southeast1.firebase
cred_obj = firebase_admin.credentials.Certificate(
    'sit225-f035c-firebase-adminsdk-ggf3h-7eb2488cd1.json'
)
default_app = firebase_admin.initialize_app(cred_obj, {
        'databaseURL':databaseURL
        })
```

Requirement already satisfied: firebase_admin in /Users/jazz/anaconda3/lib/python3.12/site-packages (6.5.0)
Requirement already satisfied: pandas in /Users/jazz/anaconda3/lib/python3.12/site-packages (2.2.2)
Requirement already satisfied: cachecontrol>=0.12.6 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from firebase_admin) (0.14.0)
Requirement already satisfied: google-api-python-client>=1.7.8 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from firebase_admin) (2.143.0)
Requirement already satisfied: google-cloud-storage>=1.37.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from firebase_admin) (2.18.2)
Requirement already satisfied: pyjwt>=2.5.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pyjwt[crypto]>=2.5.0->firebase_admin) (2.8.0)
Requirement already satisfied: google-api-core<3.0.0dev,>=1.22.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (2.19.2)
Requirement already satisfied: google-cloud-firestore>=2.9.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from firebase_admin) (2.18.0)
Requirement already satisfied: numpy>=1.26.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pandas) (1.26.4)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pandas) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pandas) (2023.3)
Requirement already satisfied: requests>=2.16.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from cachecontrol>=0.12.6->firebase_admin) (2.32.2)
Requirement already satisfied: msgpack<2.0.0,>=0.5.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from cachecontrol>=0.12.6->firebase_admin) (1.0.3)
Requirement already satisfied: googleapis-common-protos<2.0.dev0,>=1.56.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (1.65.0)
Requirement already satisfied: protobuf!=3.20.0,!=3.20.1,!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<6.0.0.dev0,>=3.19.5 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (5.28.0)
Requirement already satisfied: proto-plus<2.0.0dev,>=1.22.3 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (1.24.0)
Requirement already satisfied: google-auth<3.0.dev0,>=2.14.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (2.34.0)
Requirement already satisfied: grpcio<2.0dev,>=1.33.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (1.66.1)
Requirement already satisfied: grpcio-status<2.0.dev0,>=1.33.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (1.66.1)
Requirement already satisfied: httplib2<1.dev0,>=0.19.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-python-client>=1.7.8->firebase_admin) (0.22.0)

Requirement already satisfied: google-auth-httplib2<1.0.0,>=0.2.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-python-client>=1.7.8->firebase_admin) (0.2.0)
Requirement already satisfied: uritemplate<5,>=3.0.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-api-python-client>=1.7.8->firebase_admin) (4.1.1)
Requirement already satisfied: google-cloud-core<3.0.0dev,>=1.4.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-cloud-firestore>=2.9.1->firebase_admin) (2.4.1)
Requirement already satisfied: google-resumable-media>=2.7.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-cloud-storage>=1.37.1->firebase_admin) (2.7.2)
Requirement already satisfied: google-crc32c<2.0dev,>=1.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-cloud-storage>=1.37.1->firebase_admin) (1.6.0)
Requirement already satisfied: cryptography>=3.4.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pyjwt[crypto]>=2.5.0->firebase_admin) (42.0.5)
Requirement already satisfied: six>=1.5 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Requirement already satisfied: cffi>=1.12 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from cryptography>=3.4.0->pyjwt[crypto]>=2.5.0->firebase_admin) (1.16.0)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-auth<3.0.dev0,>=2.14.1->google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (5.3.3)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-auth<3.0.dev0,>=2.14.1->google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (0.2.8)
Requirement already satisfied: rsa<5,>=3.1.4 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from google-auth<3.0.dev0,>=2.14.1->google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (4.9)
Requirement already satisfied: pyparsing!=3.0.0,!=3.0.1,!=3.0.2,!=3.0.3,<4,>=2.4.2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from httplib2<1.dev0,>=0.19.0->google-api-python-client>=1.7.8->firebase_admin) (3.0.9)
Requirement already satisfied: charset-normalizer<4,>=2 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from requests>=2.16.0->cachecontrol>=0.12.6->firebase_admin) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from requests>=2.16.0->cachecontrol>=0.12.6->firebase_admin) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from requests>=2.16.0->cachecontrol>=0.12.6->firebase_admin) (2.2.2)
Requirement already satisfied: certifi>=2017.4.17 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from requests>=2.16.0->cachecontrol>=0.12.6->firebase_admin) (2024.6.2)
Requirement already satisfied: pycparser in /Users/jazz/anaconda3/lib/python3.12/site-packages (from cffi>=1.12->cryptography>=3.4.0->pyjwt[crypto]>=2.5.0->firebase_admin) (2.21)
Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /Users/jazz/anaconda3/lib/python3.12/site-packages (from pyasn1-modules>=0.2.1->google-auth<3.0.dev0,>=2.14.1->google-api-core<3.0.0dev,>=1.22.1->google-api-core[grpc]<3.0.0dev,>=1.22.1; platform_python_implementation != "PyPy"->firebase_admin) (0.4.8)

```python
In [3]:  from firebase_admin import db

         # A reference point is always needed to be set
         # before any operation is carried out on a database.
         #
         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
                 }
         }

         # JSON format data is set (overwritten) to the reference
         # point set at /, which is the root node.
         #
         ref.set(data)
```

```python
In [4]:  ref = db.reference("/")  # set ref point

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

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


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

```
books = ref.get()
print(books)
```

```
{'Book1': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 1
00, 'Title': 'The Fellowship of the Ring'}, 'Book2': {'Author': 'J.R.R. To
lkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Two Towers'},
'Book3': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 10
0, 'Title': 'The Return of the King'}, 'Book4': {'Author': 'Paulo Coelho',
'Genre': 'Fiction', 'Price': 100, 'Title': 'Brida'}}
<class 'dict'>
Book1: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100,
'Title': 'The Fellowship of the Ring'}
Book2: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100,
'Title': 'The Two Towers'}
Book3: {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100,
'Title': 'The Return of the King'}
Book4: {'Author': 'Paulo Coelho', 'Genre': 'Fiction', 'Price': 100, 'Titl
e': 'Brida'}
{'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Titl
e': 'The Fellowship of the Ring'}
```

In [5]:
```python
# Write using push() function
# Note that a set() is called on top of 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)
```

In [6]:
```python
# Update data
#
# Requirement: The price of the books by
# J. R. R. Tolkien is reduced to 80 units to
# offer a discount.
#
ref = db.reference("/Books/Best_Sellers/")
best_sellers = ref.get()
print(best_sellers)
for key, value in best_sellers.items():
        if(value["Author"] == "J.R.R. Tolkien"):
                value["Price"] = 90
                ref.child(key).update({"Price":80})
```

```
{'-O5sCPk45pRoShjTWYwZ': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fanta
sy', 'Price': 100, 'Title': 'The Fellowship of the Ring'}, '-O5sCPoFnhJLZi
l9Np_Y': {'Author': 'J.R.R. Tolkien', 'Genre': 'Epic fantasy', 'Price': 10
0, 'Title': 'The Two Towers'}, '-O5sCPsVrp0jpnKgAcSn': {'Author': 'J.R.R.
Tolkien', 'Genre': 'Epic fantasy', 'Price': 100, 'Title': 'The Return of t
he King'}, '-O5sCPwb4IRrzV8CEseE': {'Author': 'Paulo Coelho', 'Genre': 'Fi
ction', 'Price': 100, 'Title': 'Brida'}}
```

```python
In [7]:  # Let's 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({})
```

```python
In [8]:  # Delete all best_seller data.
         #
         ref = db.reference("/Books/Best_Sellers/")
         best_sellers = ref.get()
         print(best_sellers)
         print(type(best_sellers))
```

```
{'-O5sCPwb4IRrzV8CEseE': {'Author': 'Paulo Coelho', 'Genre': 'Fiction', 'P
rice': 100, 'Title': 'Brida'}}
<class 'dict'>
```

```python
In [9]:  ref = db.reference("/Books/Best_Sellers")
         ref.set({})
```