CRUD Operations with MongoDB and FastAPI

Group Members:

Rahul Govind Kumar-25PGAI0003

Ria Singh-25PGPAI0006

Anurag Sahu-25PGAI0143

Nishanth Pandey –25PGPAI0042

Mihir Kumar- 25PGPAI0046

GitHub Repository Link: https://github.com/rahulorihiki/FastAPI_MongoDB_CRUD_Application

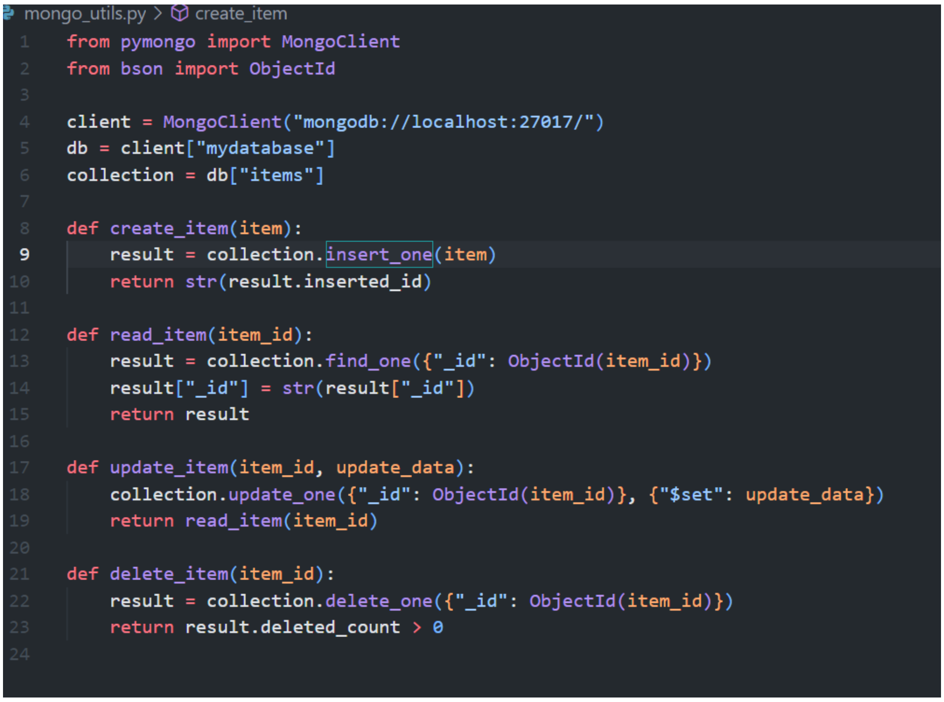
1. Setup MongoDB database and collection:

1. Download and install MongoDB

2. Create a MongoDB database and collection and run below commands:

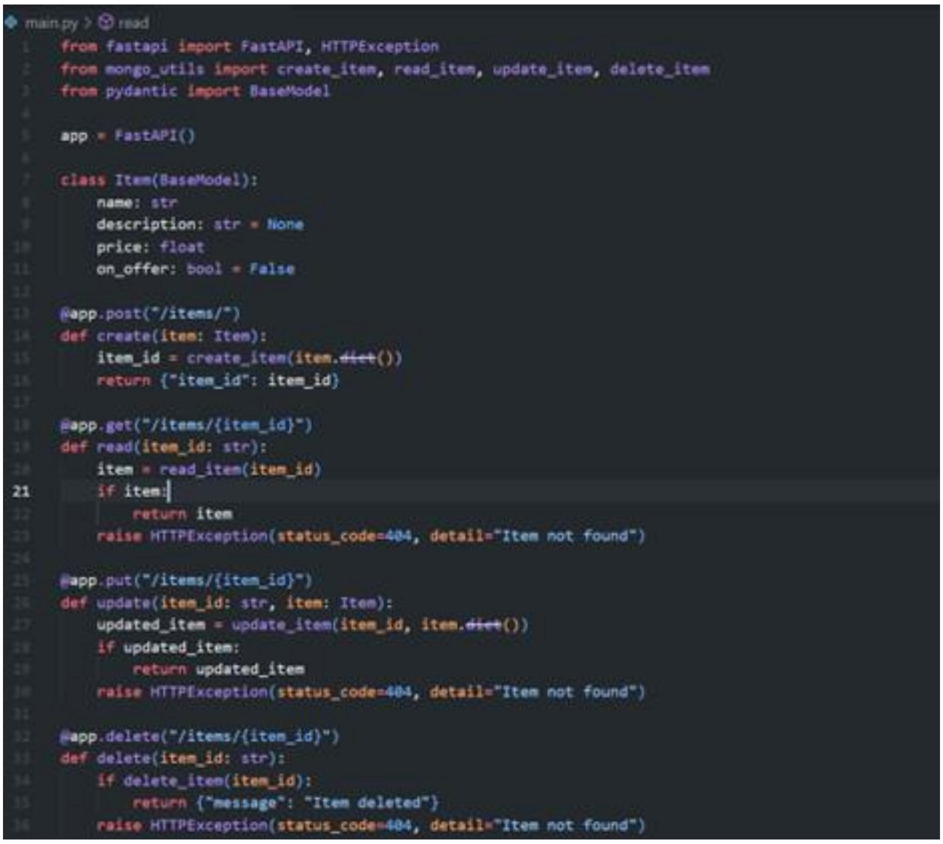
```
mongosh
use mydatabase
db.createCollection('items')
```

2. Implement CRUD Wrapper Functions:



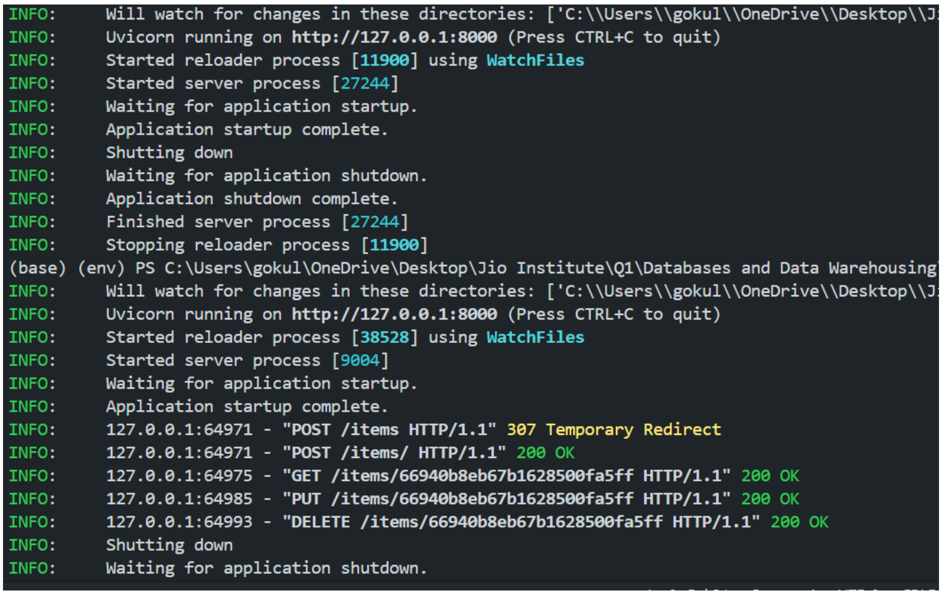
```python
from pymongo import MongoClient
from bson import ObjectId

client = MongoClient("mongodb://localhost:27017/")
db = client["mydatabase"]
collection = db["items"]

def create_item(item):
    result = collection.insert_one(item)
    return str(result.inserted_id)

def read_item(item_id):
    result = collection.find_one({"_id": ObjectId(item_id)})
    result["_id"] = str(result["_id"])
    return result

def update_item(item_id, update_data):
    collection.update_one({"_id": ObjectId(item_id)}, {"$set": update_data})
    return read_item(item_id)

def delete_item(item_id):
    result = collection.delete_one({"_id": ObjectId(item_id)})
    return result.deleted_count > 0

```

3. Develop RESTful APIs with FastAPI:



main.py > read

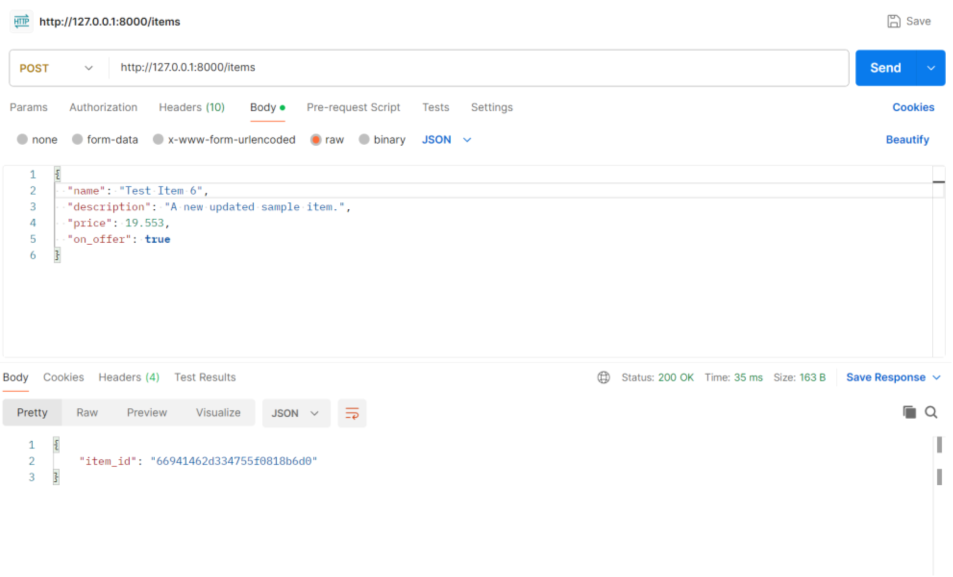
```python
from fastapi import FastAPI, HTTPException
from mongo_utils import create_item, read_item, update_item, delete_item
from pydantic import BaseModel

app = FastAPI()

class Item(BaseModel):
    name: str
    description: str = None
    price: float
    on_offer: bool = False

@app.post("/items/")
def create(item: Item):
    item_id = create_item(item.~~dict~~())
    return {"item_id": item_id}

@app.get("/items/{item_id}")
def read(item_id: str):
    item = read_item(item_id)
    if item:
        return item
    raise HTTPException(status_code=404, detail="Item not found")

@app.put("/items/{item_id}")
def update(item_id: str, item: Item):
    updated_item = update_item(item_id, item.~~dict~~())
    if updated_item:
        return updated_item
    raise HTTPException(status_code=404, detail="Item not found")

@app.delete("/items/{item_id}")
def delete(item_id: str):
    if delete_item(item_id):
        return {"message": "Item deleted"}
    raise HTTPException(status_code=404, detail="Item not found")
```



```
INFO:     Will watch for changes in these directories: ['C:\\Users\\gokul\\OneDrive\\Desktop\\J
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [11900] using WatchFiles
INFO:     Started server process [27244]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Shutting down
INFO:     Waiting for application shutdown.
INFO:     Application shutdown complete.
INFO:     Finished server process [27244]
INFO:     Stopping reloader process [11900]
(base) (env) PS C:\Users\gokul\OneDrive\Desktop\Jio Institute\Q1\Databases and Data Warehousing
INFO:     Will watch for changes in these directories: ['C:\\Users\\gokul\\OneDrive\\Desktop\\J
INFO:     Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)
INFO:     Started reloader process [38528] using WatchFiles
INFO:     Started server process [9004]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     127.0.0.1:64971 - "POST /items HTTP/1.1" 307 Temporary Redirect
INFO:     127.0.0.1:64971 - "POST /items/ HTTP/1.1" 200 OK
INFO:     127.0.0.1:64975 - "GET /items/66940b8eb67b1628500fa5ff HTTP/1.1" 200 OK
INFO:     127.0.0.1:64985 - "PUT /items/66940b8eb67b1628500fa5ff HTTP/1.1" 200 OK
INFO:     127.0.0.1:64993 - "DELETE /items/66940b8eb67b1628500fa5ff HTTP/1.1" 200 OK
INFO:     Shutting down
INFO:     Waiting for application shutdown.
```

4. Test Application using Postman

a. Create



http://127.0.0.1:8000/items

POST http://127.0.0.1:8000/items

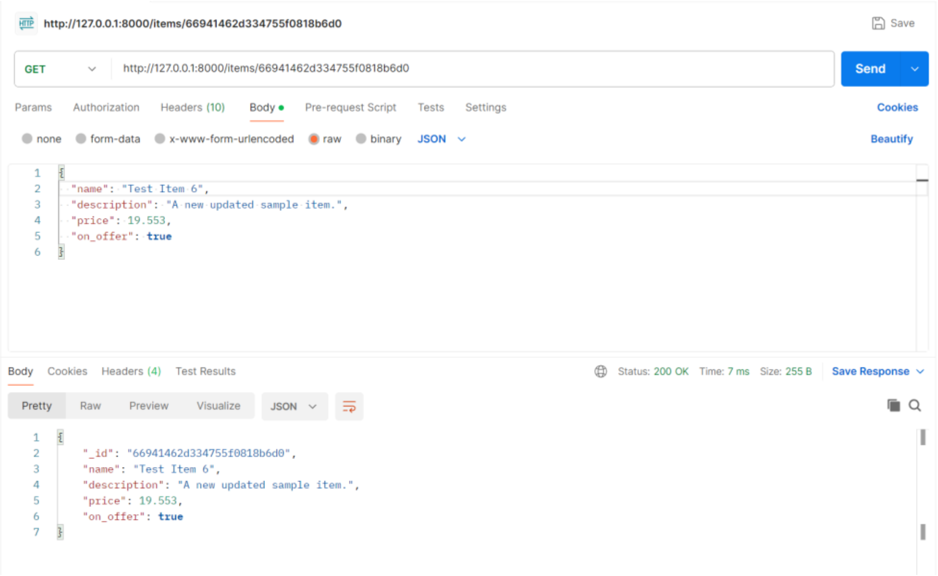
Body: raw JSON

```
{
  "name": "Test Item 6",
  "description": "A new updated sample item.",
  "price": 19.553,
  "on_offer": true
}
```

Status: 200 OK Time: 35 ms Size: 163 B

```
{
    "item_id": "66941462d334755f0818b6d0"
}
```

b. Read



http://127.0.0.1:8000/items/66941462d334755f0818b6d0

GET http://127.0.0.1:8000/items/66941462d334755f0818b6d0

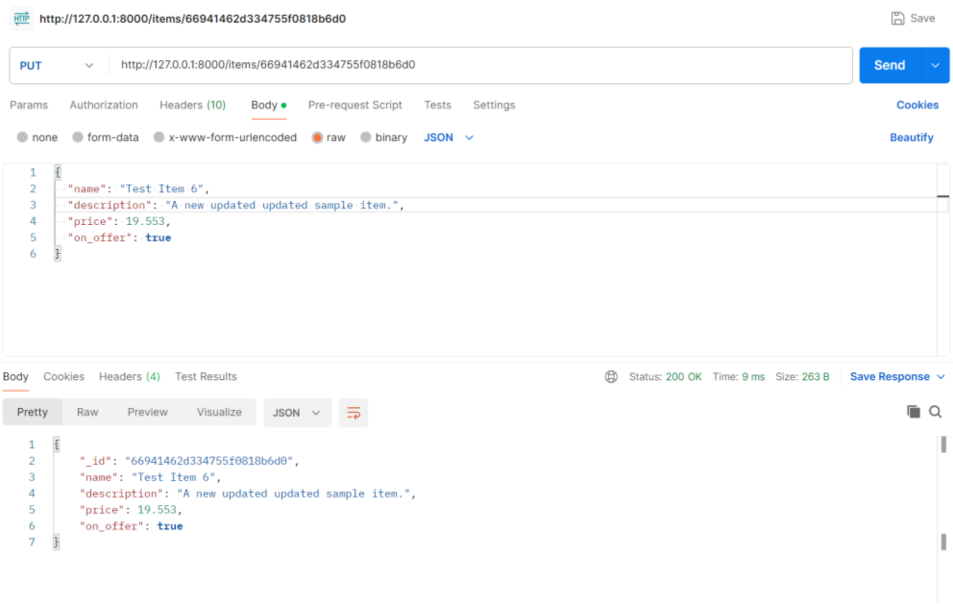
Body: raw JSON

```
{
  "name": "Test Item 6",
  "description": "A new updated sample item.",
  "price": 19.553,
  "on_offer": true
}
```

Status: 200 OK Time: 7 ms Size: 255 B

```
{
    "_id": "66941462d334755f0818b6d0",
    "name": "Test Item 6",
    "description": "A new updated sample item.",
    "price": 19.553,
    "on_offer": true
}
```

c. Update



http://127.0.0.1:8000/items/66941462d334755f0818b6d0

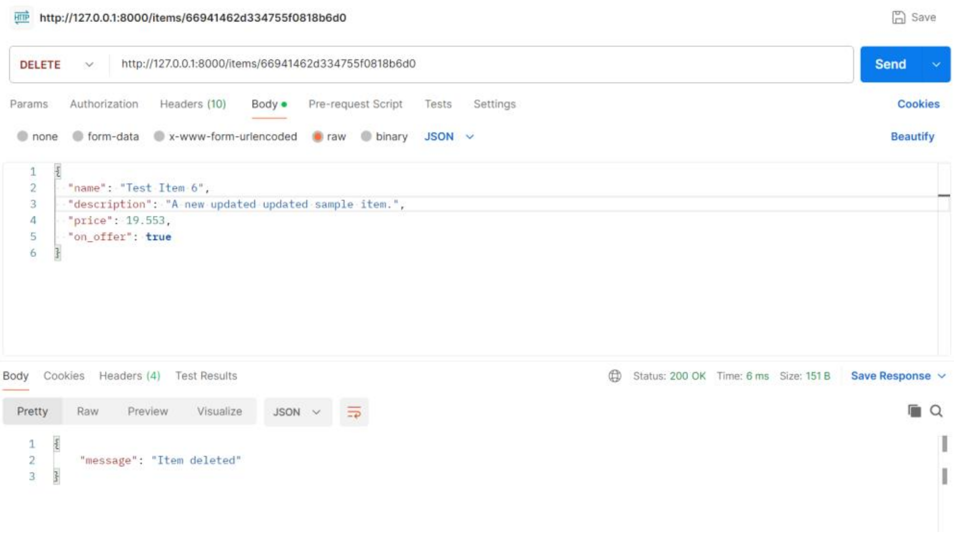
PUT http://127.0.0.1:8000/items/66941462d334755f0818b6d0

```
{
  "name": "Test Item 6",
  "description": "A new updated updated sample item.",
  "price": 19.553,
  "on_offer": true
}
```

Status: 200 OK Time: 9 ms Size: 263 B

```
{
    "_id": "66941462d334755f0818b6d0",
    "name": "Test Item 6",
    "description": "A new updated updated sample item.",
    "price": 19.553,
    "on_offer": true
}
```

4. Delete



http://127.0.0.1:8000/items/66941462d334755f0818b6d0

DELETE http://127.0.0.1:8000/items/66941462d334755f0818b6d0

```
{
  "name": "Test Item 6",
  "description": "A new updated updated sample item.",
  "price": 19.553,
  "on_offer": true
}
```

Status: 200 OK Time: 6 ms Size: 151 B

```
{
    "message": "Item deleted"
}
```