

Outputs for illustration

1. Seeding DB with mock data and retrieving the records

```
PS L:\Project> python db.py
Connection to MySQL established!
RECORDS INSERTED!!!!!!!
PS L:\Project> python retrieve.py
ID|Book Name|Author Name|Publication year
1|The C++ Programming Language|Bjarne Stroustrup|1985
2|Pride and Prejudice|Jane Austen|1813
3|The Alchemist|Bjarne Stroustrup|1988
4|Gone with the Wind| Margaret Mitchell|1936
5|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
```

2. Adding a new book by editing line 6-7 in file post.py and verifying by retrieving records.

```
1  import requests
2  import json
3
4  PORT=5000
5
6  book_name="Lord of the Rings"
7  author_name="J.R.R. Tolkien"
8  publication_year=1955
9
10
11 post_data={"bname":book_name,"aname":author_name,"py":publication_year}
12 json_data = json.dumps(post_data)
13 URL="http://localhost:"+str(PORT)+"/"
14 URL=URL+"api/books"
15 headers={'Content-Type':'application/json'}
16 response=requests.post(URL,data=json_data,headers=headers)
17 print(response.text)
```

```
PS L:\Project> python retrieve.py
ID|Book Name|Author Name|Publication year
1|The C++ Programming Language|Bjarne Stroustrup|1985
2|Pride and Prejudice|Jane Austen|1813
3|The Alchemist|Bjarne Stroustrup|1988
4|Gone with the Wind| Margaret Mitchell|1936
5|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
6|Lord of the Rings|J.R.R. Tolkien|1955
```

3. Reinserting same data again, we get the message that book already exists.

```
PS L:\Project> python post.py
Book was added to the database
PS L:\Project> python retrieve.py
ID|Book Name|Author Name|Publication year
1|The C++ Programming Language|Bjarne Stroustrup|1985
2|Pride and Prejudice|Jane Austen|1813
3|The Alchemist|Bjarne Stroustrup|1988
4|Gone with the Wind| Margaret Mitchell|1936
5|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
6|Lord of the Rings|J.R.R. Tolkien|1955
PS L:\Project> python post.py
Book already exists
```

4. Updating a book, mention book attributes on lines 6-7 in put.py file. We can observe modifications for book with id 3

```
1 import requests
2 import json
3
4 PORT=5000
5
6 bid=3
7 book_name="The Alchemist"
8 author_name="Paulo Coelho"
9 publication_year=2023
10
11 post_data={"bname":book_name,"aname":author_name,"py":publication_year}
12 json_data = json.dumps(post_data)
13 URL="http://localhost:"+str(PORT)+"/"
14 URL=URL+"api/books"
15 URL=URL+"/"+str(bid)
16 headers={'Content-Type':'application/json'}
17 response=requests.put(URL,data=json_data,headers=headers)
18 print(response.text)
```

```
PS L:\Project> python retrieve.py
ID|Book Name|Author Name|Publication year
1|The C++ Programming Language|Bjarne Stroustrup|1985
2|Pride and Prejudice|Jane Austen|1813
3|The Alchemist|Bjarne Stroustrup|1988
4|Gone with the Wind| Margaret Mitchell|1936
5|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
6|Lord of the Rings|J.R.R. Tolkien|1955
PS L:\Project> python put.py
Book was updated
PS L:\Project> python retrieve.py
ID|Book Name|Author Name|Publication year
1|The C++ Programming Language|Bjarne Stroustrup|1985
2|Pride and Prejudice|Jane Austen|1813
3|The Alchemist|Paulo Coelho|2023
4|Gone with the Wind| Margaret Mitchell|1936
5|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
6|Lord of the Rings|J.R.R. Tolkien|1955
```

5. Updating a book that does not exist, mention book id that does not exist in record. Then, you get the below message.

```
PS L:\Project> python put.py  
Book does not exist
```