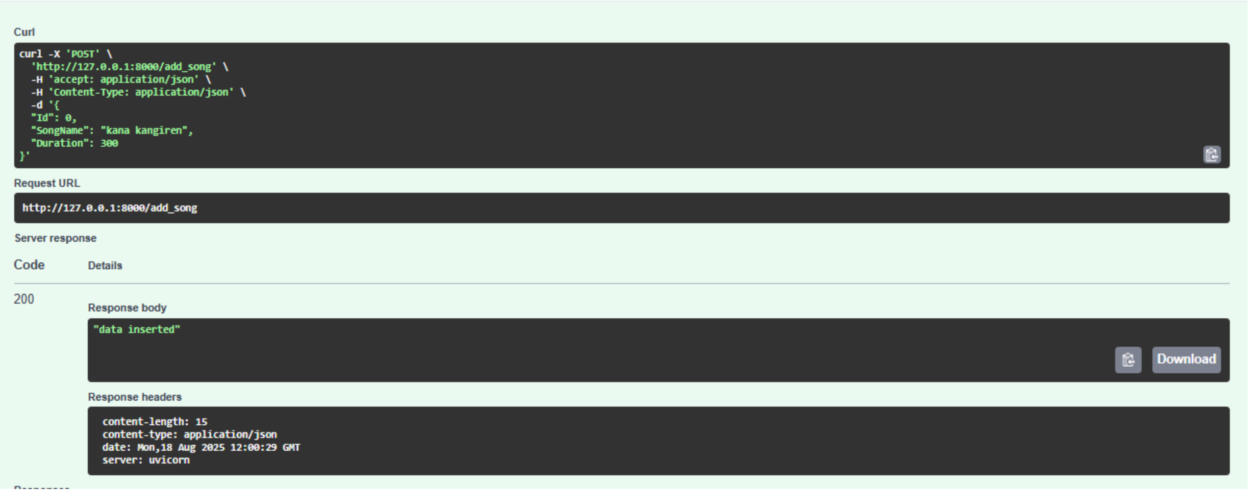
Day 10

Dharunya S

- Create Fast API, to store, retrive all and retrive specific song by Id.
- SQLite can be used as data store
- table can contain

day10 task.py

```
from fastapi.templating import Jinja2Templates
from pydantic import BaseModel
from sqlalchemy import Column, Integer, String, create_engine
from sqlalchemy.dialects.sqlite import *
from sqlalchemy.orm import sessionmaker,Session,declarative_base
import uvicorn
from typing import List
from fastapi.responses import HTMLResponse
templates = Jinja2Templates(directory="templates")
SQLALCHEMY_DATABASE_URL="sqlite:///./movie.db"

engine=create_engine(SQLALCHEMY_DATABASE_URL,connect_args={"check_same_thread":False})
session=sessionmaker(autocommit=False,autoflush=False,bind=engine)


base=declarative_base()
class Moviessong(base):
    __tablename__='movie'

    Id=Column(Integer,primary_key=True,index=True )

    SongName=Column(String(50),nullable=False)
    Duration=Column(Integer,nullable=False)
    base.metadata.create_all(bind=engine)



from fastapi import Depends, FastAPI,Request
app=FastAPI()

def get_db():
    db=session()
    try:
        yield db
    finally:
        db.close()



    # return db
class Movies(BaseModel):
    Id:int
    SongName:str
    Duration:int
    class config:
        orm_mode=True


@app.post('/add_song')
def add_song(mt1:Movies,db:Session=Depends(get_db)):
    # db=get_db()
    mt= Moviessong(SongName=mt1.SongName,Duration=mt1.Duration)
    db.add(mt)
    db.commit()
    db.refresh(mt)
    return "data inserted"



@app.get("/movies", response_class=HTMLResponse)
def get_movies(request: Request, db: Session = Depends(get_db)):
    movies = db.query(Moviessong).all()
    return templates.TemplateResponse("songs.html", {"request": request, "movies":
movies})


class movie_dto(BaseModel):
    SongName:str
    class config:
        orm_mode=True
@app.get("/movies/{mov_id}",response_model= movie_dto)
def get_movie_by_id(mov_id: int,db: Session = Depends(get_db)):
    mt= db.query(Moviessong).filter_by(Id=mov_id).first()
    mt_dto=movie_dto(SongName=mt.SongName)
    return mt_dto

@app.middleware("http")
async def addmiddleware(request:Request,call_next):
    print("Middleware has intercepted the calls!")
    response=await call_next(request)
    return response

if __name__=="__main__":
    uvicorn.run("day10task:app",host="127.0.0.1",port=8000,reload=True)
```



Curl

```
curl -X 'POST' \
  'http://127.0.0.1:8000/add_song' \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d '{
  "Id": 0,
  "SongName": "kana kangiren",
  "Duration": 300
}'
```

Request URL

```
http://127.0.0.1:8000/add_song
```

Server response

Code Details

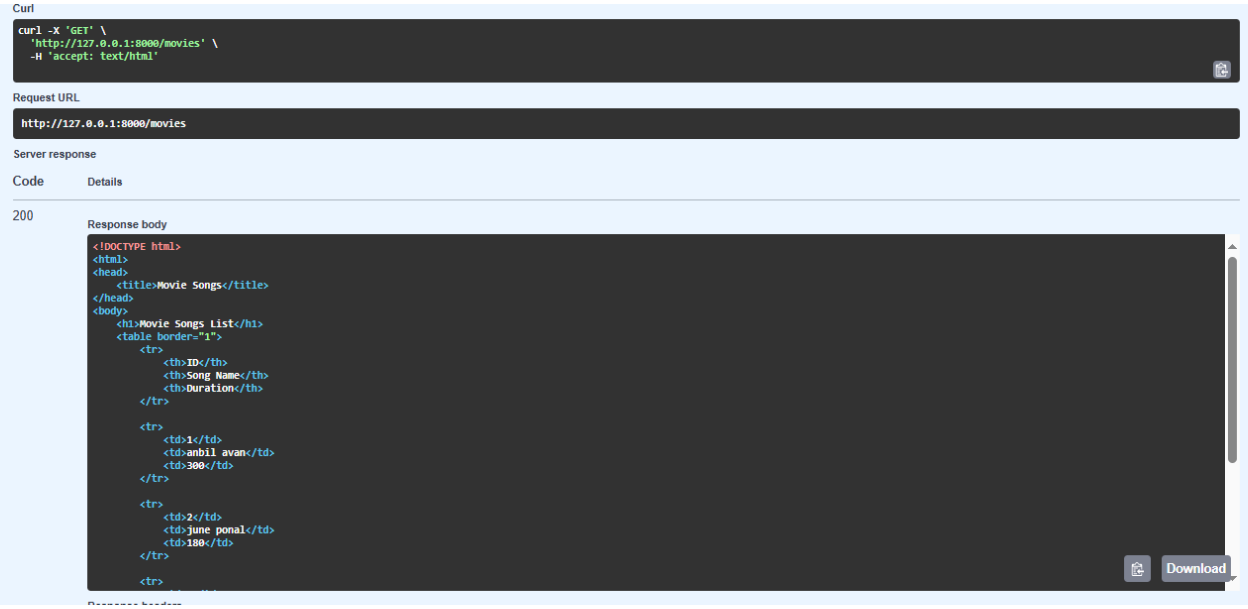
200

Response body

```
"data inserted"
```

Response headers

```
 content-length: 15
 content-type: application/json
 date: Mon,18 Aug 2025 12:00:29 GMT
 server: uvicorn
```



Curl

```
curl -X 'GET' \
  'http://127.0.0.1:8000/movies' \
  -H 'accept: text/html'
```

Request URL

```
http://127.0.0.1:8000/movies
```

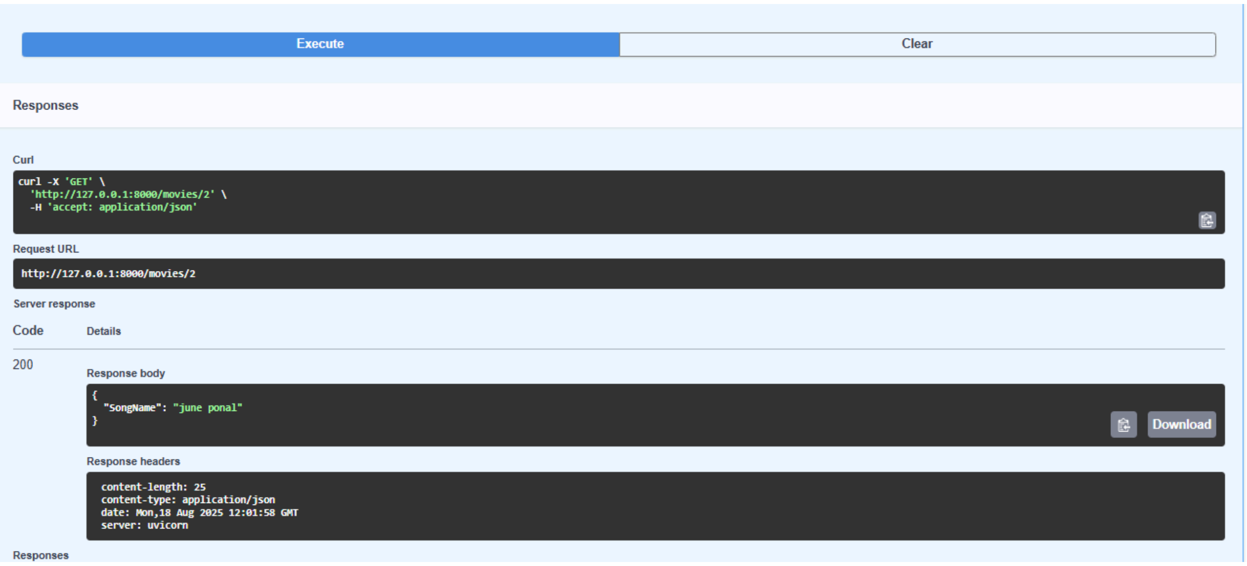
Server response

Code Details

200

Response body

```
<!DOCTYPE html>
<html>
<head>
    <title>Movie Songs</title>
</head>
<body>
    <h1>Movie Songs List</h1>
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Song Name</th>
            <th>Duration</th>
        </tr>

        <tr>
            <td>1</td>
            <td>anbil avan</td>
            <td>300</td>
        </tr>

        <tr>
            <td>2</td>
            <td>june ponal</td>
            <td>180</td>
        </tr>

        <tr>
```

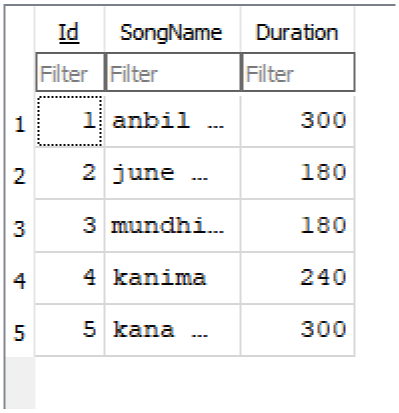


```
curl -X 'GET' \
  'http://127.0.0.1:8000/movies/2' \
  -H 'accept: application/json'
```

```
http://127.0.0.1:8000/movies/2
```

```
{
  "SongName": "june ponal"
}
```

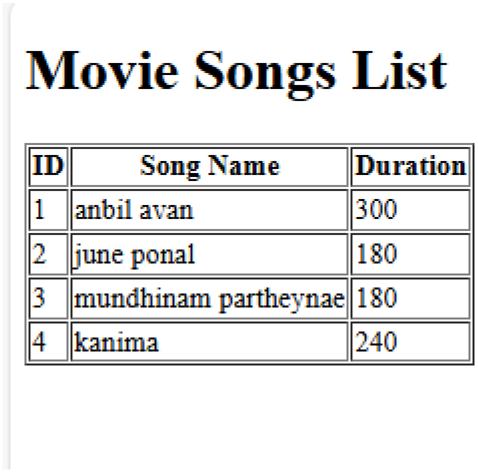
```
content-length: 25
content-type: application/json
date: Mon,18 Aug 2025 12:01:58 GMT
server: uvicorn
```



	Id	SongName	Duration
	Filter	Filter	Filter
1	1	anbil ...	300
2	2	june ...	180
3	3	mundhi...	180
4	4	kanima	240
5	5	kana ...	300

Songs.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Movie Songs</title>
</head>
<body>
    <h1>Movie Songs List</h1>
    <table border="1">
        <tr>
            <th>ID</th>
            <th>Song Name</th>
            <th>Duration</th>
        </tr>
        {% for song in movies %}
        <tr>
            <td>{{ song.Id }}</td>
            <td>{{ song.SongName }}</td>
            <td>{{ song.Duration }}</td>
        </tr>
        {% endfor %}
    </table>
</body>
</html>
```



Movie Songs List

ID	Song Name	Duration
1	anbil avan	300
2	june ponal	180
3	mundhinam partheynae	180
4	kanima	240