```
from fastapi import FastAPI, HTTPException, Query
from pydantic import BaseModel
from typing import List, Optional

app = FastAPI()

# Pydantic models
class Book(BaseModel):
    title: str
    author: str
    publication_year: int

class Review(BaseModel):
    text: str
    rating: int

# In-memory storage for books and reviews (replace with database in production)
books_db = []
reviews_db = []

# Endpoints
@app.post("/books/")
def add_book(book: Book):
    books_db.append(book)
    return {"message": "Book added successfully"}

@app.post("/books/{book_id}/reviews/")
def submit_review(book_id: int, review: Review):
    if book_id < 0 or book_id >= len(books_db):
        raise HTTPException(status_code=404, detail="Book not found")
    reviews_db.append((book_id, review))
    return {"message": "Review submitted successfully"}

@app.get("/books/")
def get_books(author: Optional[str] = None, publication_year: Optional[int] = None):
    filtered_books = books_db
    if author:
        filtered_books = [book for book in filtered_books if book.author == author]
    if publication_year:
        filtered_books = [book for book in filtered_books if book.publication_year == publication_year]
    return filtered_books

@app.get("/books/{book_id}/reviews/")
def get_book_reviews(book_id: int):
    if book_id < 0 or book_id >= len(books_db):
        raise HTTPException(status_code=404, detail="Book not found")
    return [review for idx, review in reviews_db if idx == book_id]

# Error Handling
@app.exception_handler(HTTPException)
async def http_exception_handler(request, exc):
    return JSONResponse(
        status_code=exc.status_code,
        content={"message": exc.detail}
    )
```