

```

from fastapi import FastAPI, HTTPException
from sqlalchemy import create_engine, Column, Integer, String, ForeignKey
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import sessionmaker, relationship

# FastAPI setup
app = FastAPI()

# SQLite setup
SQLALCHEMY_DATABASE_URL = "sqlite:///./test.db"
engine = create_engine(SQLALCHEMY_DATABASE_URL)
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
Base = declarative_base()

# Database models
class Book(Base):
    __tablename__ = "books"

    id = Column(Integer, primary_key=True, index=True)
    title = Column(String, index=True)
    author = Column(String, index=True)
    publication_year = Column(Integer)

    reviews = relationship("Review", back_populates="book")

class Review(Base):
    __tablename__ = "reviews"

    id = Column(Integer, primary_key=True, index=True)
    text = Column(String)
    rating = Column(Integer)
    book_id = Column(Integer, ForeignKey("books.id"))

    book = relationship("Book", back_populates="reviews")

# Create tables in the database
Base.metadata.create_all(bind=engine)

# CRUD operations
def get_db():
    db = SessionLocal()
    try:
        yield db
    finally:
        db.close()

@app.post("/books/")
def add_book(book: BookCreate, db: Session = Depends(get_db)):
    db_book = Book(**book.dict())
    db.add(db_book)
    db.commit()
    db.refresh(db_book)
    return db_book

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

```