

Source Code: AI Generated Report

Generated on: 2026-02-07 16:33:31

```
from fastapi import FastAPI, HTTPException

from pydantic import BaseModel, conlist

from sqlalchemy import create_engine, Column, String, Float

from sqlalchemy.orm import DeclarativeBase, Mapped, mapped_column, sessionmaker

import pandas as pd

from fastapi.testclient import TestClient


# Database setup

DATABASE_URL = "sqlite:///memory:"

engine = create_engine(DATABASE_URL)

SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)


class Base(DeclarativeBase):

    pass


class ProductModel(Base):

    __tablename__ = "products"

    id: Mapped[int] = mapped_column(primary_key=True)

    name: Mapped[str] = mapped_column(String)

    price: Mapped[float] = mapped_column(Float)


Base.metadata.create_all(bind=engine)
```

```
# FastAPI app

app = FastAPI()


class Product(BaseModel):

    name: str

    price: float


class BillRequest(BaseModel):

    products: conlist(Product)


@app.post("/calculate-bill")

def calculate_bill(bill_request: BillRequest):

    # Validate product prices

    for product in bill_request.products:

        if product.price < 0:

            raise HTTPException(status_code=400, detail="Product price cannot be
negative.")


    # Extract prices and calculate total

    prices = pd.Series([product.price for product in bill_request.products])

    total_price = prices.sum()

    gst = total_price * 0.12

    final_bill = total_price + gst


    return {

        "total_price": total_price,
```

```

        "gst": gst,

        "final_bill": final_bill
    }

# Test client

client = TestClient(app)

# Test case

def test_calculate_bill():

    response = client.post("/calculate-bill", json={

        "products": [

            {"name": "Product 1", "price": 100.0},

            {"name": "Product 2", "price": 200.0},

            {"name": "Product 3", "price": 300.0}

        ]

    })

    print(response.json())

    assert response.status_code == 200

    assert response.json() == {

        "total_price": 600.0,

        "gst": 72.0,

        "final_bill": 672.0

    }

# Run the test

test_calculate_bill()

```