Test Case #01 – Valid customer, low-volume transaction

🔸 Input:

customer_id = 123

language = en

🔸 Precondition:

- Customer with ID 123 exists in the `Customers` table

- Customer has 2 transactions in the `Transactions` table

🔸 Expected Output:

- HTTP 200 OK

- Response is a downloadable PDF named: credit_card_statement_123.pdf

- PDF contains:

  - Header: "DBS Bank Credit Card Statement"

  - Customer name and email

  - 2 transactions listed with date, merchant, amount, and type

🔸 Actual Output:

- [To be filled after running the test]

- e.g. HTTP 200 OK, file downloaded successfully

🔸 Result:

✅ PASS

Test Case #02 – Invalid customer ID

🔸 Input:

customer_id = 999

language = en

🔸 Precondition:

- No customer with ID 999 in the `Customers` table

🔸 Expected Output:

- HTTP 404 Not Found

- Response message: "Customer not found."

🔸 Actual Output:

- [To be filled after running the test]

- e.g. HTTP 404, message: "Customer not found."

🔸 Result:

✅ PASS

Test Case #03 – Bilingual statement in Chinese

🔸 Input:

customer_id = 123

language = zh

🔸 Precondition:

- Customer exists with at least 1 transaction

🔸 Expected Output:

- PDF generated with:

- Title: "DBS银行信用卡对账单"

- Font includes Chinese (Noto Sans SC)

- Table headers and labels in Chinese

🔸 Actual Output:

- [To be filled after running the test]

🔸 Result:

✅ PASS

Test Case #04 – Missing customer_id in request

🔸 Input:

URL = /generate_pdf

(no customer_id parameter)

🔸 Expected Output:

- HTTP 400 Bad Request

- Response message: "Customer ID is required."

🔸 Actual Output:

- [To be filled after running the test]

🔸 Result:

✅ PASS

from datetime import datetime

from app import generate_pdf

```
# Dummy customer data
customer = ("John", "Doe", "john.doe@example.com")

# 1. Low-volume transactions
low_volume = [
    (datetime(2025, 4, 1), "Amazon", 100.00, "Purchase"),
    (datetime(2025, 4, 2), "Grab", 25.00, "Purchase")
]
pdf_io = generate_pdf(customer, low_volume, 'en')
with open("sample_output/low_volume_statement.pdf", "wb") as f:
    f.write(pdf_io.getbuffer())

# 2. High-volume transactions
high_volume = [
    (datetime(2025, 4, i), f"Merchant{i}", i * 10.00, "Purchase")
    for i in range(1, 51)
]
pdf_io = generate_pdf(customer, high_volume, 'en')
with open("sample_output/high_volume_statement.pdf", "wb") as f:
    f.write(pdf_io.getbuffer())

# 3. Bilingual (e.g., Chinese)
pdf_io = generate_pdf(customer, low_volume, 'zh')
with open("sample_output/bilingual_statement.pdf", "wb") as f:
    f.write(pdf_io.getbuffer())

print("✅ PDFs generated successfully in the sample_output/ folder.")
```