Section 1:
Name:Chris Jakubowski
Student ID: 20388138
Submission Date: November 10, 2025

Section 2:
A stub is essentially creating a stand-in actor that is given designated values for specific functions. It allows incomplete/not usable functions to still give a value for the sake of testing, by having the programmer designate what the outcome would/should be.
Mocks basically pretend to be the real function instead of just spitting out a value. It is an object that holds methods, and also records the interactions it has.

Section 3:
Creating the virtual enviroment:
```
python -m venv venv
```

activating the venv in git bash:
```
source venv/Scripts/activate
```

running all tests in tests/:
```
pytest tests/
```

running tests with coverage report for services:
```
pytest --cov=services --cov-report=html --cov-report=term
```

Section 4:

Function Name	Purpose	Mocks	
test_pay_late_fees_success	Successful payment	Mock(spec=PaymentGateway)	
test_pay_late_fees_declined_by_gateway	Declined by gateway	Mock(spec=PaymentGateway)	
test_pay_late_fees_invalid_patron_id	Had invalid patron id	Mock(spec=PaymentGateway)	
test_pay_late_fees_zero_late_fees	Had zero late fees	Mock(spec=PaymentGateway)	

test_pay_late_fees_network_error_exception	Had network error exception	Mock(spec=PaymentGateway)	
test_refund_late_fee_payment_success	Refund success	Mock(spec=PaymentGateway)	
test_refund_late_fee_invalid_transaction_id	Invalid transaction id	Mock(spec=PaymentGateway)	
test_refund_late_fee_invalid_amount_zero	Late fee was zero	Mock(spec=PaymentGateway)	
test_refund_late_fee_invalid_amount_negative	Late fee was negative	Mock(spec=PaymentGateway)	
test_refund_late_fee_invalid_amount_exceeds_maximum	Late fee was greater than max	Mock(spec=PaymentGateway)	

Section 5:

Near initial (could not see far back enough in terminal for exact initial):
Some coverage was in error/misleading and not actually covered and had to be replaced

```
Name                          Stmts   Miss  Cover
-------------------------------------------------
services\__init__.py              0      0   100%
services\library_service.py     147     23    84%
services\payment_service.py      29     22    24%
-------------------------------------------------
TOTAL                           176     45    74%
Coverage HTML written to dir htmlcov
=========================================================== short test summary info ============================================================
FAILED tests/test_add_book.py::test_add_book_valid_input - assert False == True
FAILED tests/test_add_book.py::test_add_book_negative_copies - AssertionError: assert 'need positive integer' in 'Total copies must be a positive integer.'
FAILED tests/test_add_book.py::test_add_book_no_author - AssertionError: assert 'need author' in 'Author is required.'
FAILED tests/test_add_book.py::test_add_book_letters_in_isbn - AssertionError: assert 'invalid isbn' in 'A book with this ISBN already exists.'
FAILED tests/test_borrow_book.py::test_borrow_book_invalid_patron_id_too_short - TypeError: argument of type 'builtin_function_or_method' is not iterable
FAILED tests/test_borrow_book.py::test_borrow_book_invalid_type_book_id - AssertionError: assert 'invalid book id' in 'book not found.'
FAILED tests/test_borrow_book.py::test_borrow_book_negative_book_id - AssertionError: assert 'invalid book id' in 'book not found.'
FAILED tests/test_patron_status.py::test_patron_with_books - TypeError: 'dict' object is not callable
FAILED tests/test_patron_status.py::test_patron_with_invalid_id - ValueError: too many values to unpack (expected 2)
FAILED tests/test_return_book.py::test_return_book_valid_input - AssertionError: assert 'successfully return' in 'book returned on time'
========================================================== 10 failed, 26 passed in 0.47s ===========================================================
```

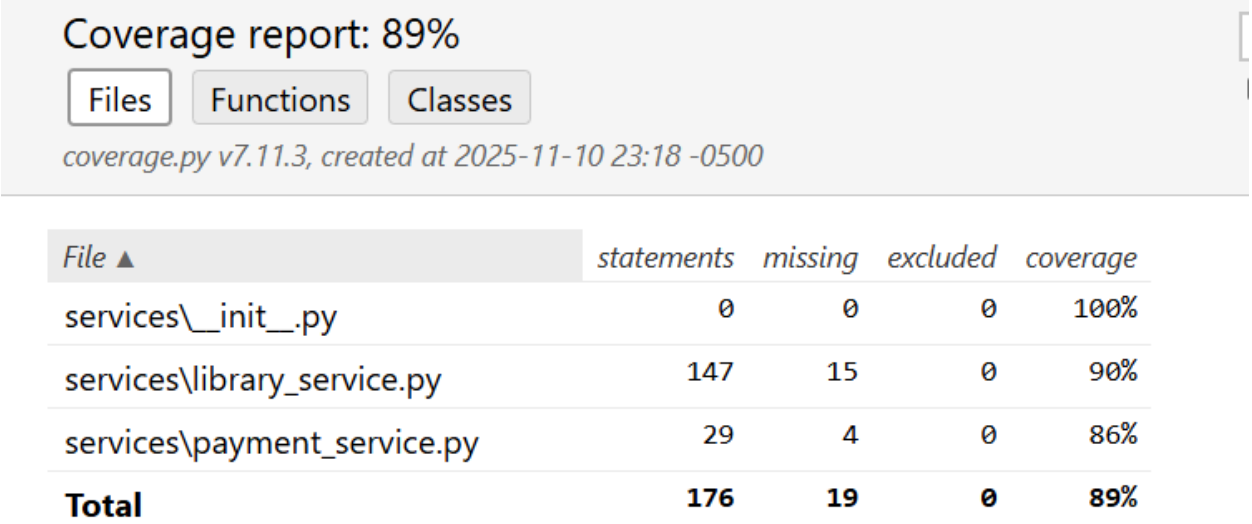
Final:

```
============================= test session starts =============================
platform win32 -- Python 3.13.9, pytest-9.0.0, pluggy-1.6.0
rootdir: C:\Users\laugh\Documents\CISC327\CISC327-CMPE327-F25-main
plugins: cov-7.0.0, mock-3.15.1
collected 49 items

tests\test_add_book.py .........                                         [ 18%]
tests\test_borrow_book.py ......                                         [ 30%]
tests\test_calculate_late_fee.py ....                                    [ 38%]
tests\test_patron_status.py ..                                           [ 42%]
tests\test_pay_late_fees.py .....                                        [ 53%]
tests\test_payment_service.py .........                                  [ 71%]
tests\test_refund_late_fee_payment.py .....                              [ 81%]
tests\test_return_book.py .....                                          [ 91%]
tests\test_search_catalog.py ....                                        [100%]

=============================== tests coverage ================================
_____________________ coverage: platform win32, python 3.13.9-final-0 _____________________

Name                          Stmts   Miss  Cover
-------------------------------------------------
services\__init__.py              0      0   100%
services\library_service.py     147     15    90%
services\payment_service.py      29      4    86%
-------------------------------------------------
TOTAL                           176     19    89%
Coverage HTML written to dir htmlcov
============================== 49 passed in 5.02s ==============================
```



File ▲	statements	missing	excluded	coverage
services__init__.py	0	0	0	100%
services\library_service.py	147	15	0	90%
services\payment_service.py	29	4	0	86%
Total	**176**	**19**	**0**	**89%**

Section 6:
The main challenge I faced was figuring out exactly how to use stubbing/mocking features in my code. I have gained insight on how/when to use this, and it appears to be a straight forward and effective feature.

Section 7:

```
tests\test_add_book.py .........
tests\test_borrow_book.py ......
tests\test_calculate_late_fee.py ....
tests\test_patron_status.py ..
tests\test_pay_late_fees.py .....
tests\test_calculate_late_fee.py ....
tests\test_patron_status.py ..
tests\test_pay_late_fees.py .....
tests\test_patron_status.py ..
tests\test_pay_late_fees.py .....
tests\test_pay_late_fees.py .....
tests\test_payment_service.py .........
tests\test_refund_late_fee_payment.py .....
tests\test_return_book.py .....
tests\test_search_catalog.py ....
```