

Init visual inspection of the User_Registration.py:

Init(self)

- Uses dictionary to store sensitive information as plain text (passwords email etc in) -> No hashing, db storing etc.

Register(self, email, password, confirm_password)

- Handels everything as plain text, no db (compare hash – hash e.g. in emails) or hashed/encrypted. Just simple checks / comparators

Is_valid_email(self, email)

- Checks only whether the input contains @ somewhere in the string and contains at least one '.' After the @ -> ([1] from the @)
- So the input can be pretty much anything

Is_strong_password

- No case sensitive testing, still using plain text

First run with Pylint:

```
***** Module User_Registration
User_Registration.py:5:0: C0301: Line too long (126/100) (line-too-long)
User_Registration.py:13:0: C0301: Line too long (141/100) (line-too-long)
User_Registration.py:20:0: C0301: Line too long (144/100) (line-too-long)
User_Registration.py:21:0: C0301: Line too long (102/100) (line-too-long)
User_Registration.py:30:0: C0301: Line too long (122/100) (line-too-long)
User_Registration.py:34:0: C0301: Line too long (118/100) (line-too-long)
User_Registration.py:36:0: C0301: Line too long (118/100) (line-too-long)
User_Registration.py:38:0: C0301: Line too long (125/100) (line-too-long)
User_Registration.py:40:0: C0301: Line too long (130/100) (line-too-long)
User_Registration.py:49:0: C0301: Line too long (101/100) (line-too-long)
User_Registration.py:62:0: C0301: Line too long (134/100) (line-too-long)
User_Registration.py:70:0: C0301: Line too long (111/100) (line-too-long)
User_Registration.py:87:0: C0301: Line too long (103/100) (line-too-long)
User_Registration.py:91:0: C0301: Line too long (126/100) (line-too-long)
User_Registration.py:96:0: C0301: Line too long (104/100) (line-too-long)
User_Registration.py:100:0: C0301: Line too long (103/100) (line-too-long)
User_Registration.py:105:0: C0301: Line too long (102/100) (line-too-long)
User_Registration.py:109:0: C0301: Line too long (105/100) (line-too-long)
User_Registration.py:118:0: C0301: Line too long (112/100) (line-too-long)
User_Registration.py:123:0: C0301: Line too long (110/100) (line-too-long)
User_Registration.py:125:0: C0301: Line too long (104/100) (line-too-long)
User_Registration.py:127:0: C0301: Line too long (116/100) (line-too-long)
User_Registration.py:128:0: C0301: Line too long (107/100) (line-too-long)
User_Registration.py:1:0: C0114: Missing module docstring (missing-module-docstring)
User_Registration.py:1:0: C0103: Module name "User_Registration" doesn't conform to snake_case naming style (invalid-name)
User_Registration.py:1:0: C0115: Missing class docstring (missing-class-docstring)
User_Registration.py:73:0: C0413: Import "import unittest" should be placed at the top of the module (wrong-import-position)
User_Registration.py:75:0: C0115: Missing class docstring (missing-class-docstring)

-----
Your code has been rated at 3.78/10 (previous run: 3.78/10, +0.00)
```

Naming style violation, missing class, import not at the top of the file, code lines too long on multiple rows etc.

First run Bandit / SonarQube: (security and deeper issues)

No significant issues detected.

```
[main] INFO    profile include tests: None
[main] INFO    profile exclude tests: None
[main] INFO    cli include tests: None
[main] INFO    cli exclude tests: None
[main] INFO    running on Python 3.13.6
Run started:2025-10-09 15:19:41.659091
```

Test results:

No issues identified.

Code scanned:

Total lines of code: 108
Total lines skipped (#nosec): 0

Run metrics:

Total issues (by severity):
Undefined: 0
Low: 0
Medium: 0
High: 0
Total issues (by confidence):
Undefined: 0
Low: 0
Medium: 0
High: 0

Files skipped (0):

ToDo based on static testing:

1. Iteration:

- Correct long lines
 - o Test and verify

```
***** Module User_Registration
User_Registration.py:46:52: C0303: Trailing whitespace (trailing-whitespace)
User_Registration.py:49:0: C0303: Trailing whitespace (trailing-whitespace)
User_Registration.py:84:0: C0301: Line too long (111/100) (line-too-long)
User_Registration.py:1:0: C0114: Missing module docstring (missing-module-docstring)
User_Registration.py:1:0: C0103: Module name "User_Registration" doesn't conform to snake_case naming style (invalid-name)
User_Registration.py:1:0: C0115: Missing class docstring (missing-class-docstring)
User_Registration.py:87:0: C0413: Import "import unittest" should be placed at the top of the module (wrong-import-position)
User_Registration.py:89:0: C0115: Missing class docstring (missing-class-docstring)

-----
Your code has been rated at 8.22/10 (previous run: 3.78/10, +4.44)
```

2. Iteration:

- Naming style violations
 - o Samesame
- Fix class
 - o Samesame
- Relocate import to top of the file
 - o Samesame

```
-----
Your code has been rated at 10.00/10 (previous run: 9.78/10, +0.22)
```

Dynamic tests via pytest:

```
===== test session starts =====
platform win32 -- Python 3.13.6, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\testaus-repo
collected 5 items

user_registration.py ..... [100%]

===== 5 passed in 0.02s =====
```

Quick visual inspection of the Payment_Processing.py:

Validate_credit_card(self, details)

- Very straight forward and simple card validation (length only)

First run with Pylint:

```
***** Module Payment_Processing
Payment_Processing.py:7:0: C0301: Line too long (110/100) (line-too-long)
Payment_Processing.py:10:0: C0301: Line too long (107/100) (line-too-long)
Payment_Processing.py:24:0: C0301: Line too long (113/100) (line-too-long)
Payment_Processing.py:30:0: C0301: Line too long (101/100) (line-too-long)
Payment_Processing.py:67:0: C0301: Line too long (115/100) (line-too-long)
Payment_Processing.py:80:0: C0303: Trailing whitespace (trailing-whitespace)
Payment_Processing.py:82:0: C0301: Line too long (112/100) (line-too-long)
Payment_Processing.py:117:0: C0301: Line too long (106/100) (line-too-long)
Payment_Processing.py:127:0: C0301: Line too long (105/100) (line-too-long)
Payment_Processing.py:144:0: C0301: Line too long (106/100) (line-too-long)
Payment_Processing.py:146:0: C0301: Line too long (118/100) (line-too-long)
Payment_Processing.py:152:0: C0301: Line too long (102/100) (line-too-long)
Payment_Processing.py:158:0: C0301: Line too long (116/100) (line-too-long)
Payment_Processing.py:167:0: C0301: Line too long (128/100) (line-too-long)
Payment_Processing.py:170:0: C0301: Line too long (116/100) (line-too-long)
Payment_Processing.py:1:0: C0114: Missing module docstring (missing-module-docstring)
Payment_Processing.py:1:0: C0103: Module name "Payment_Processing" doesn't conform to snake_case naming style (invalid-name)
Payment_Processing.py:55:8: W0612: Unused variable 'expiry_date' (unused-variable)
Payment_Processing.py:90:15: W0718: Catching too general exception Exception (broad-exception-caught)
Payment_Processing.py:85:12: R1705: Unnecessary "else" after "return", remove the "else" and de-indent the code inside it (no-else-return)
Payment_Processing.py:94:52: W0613: Unused argument 'amount' (unused-argument)

-----
Your code has been rated at 6.87/10 (previous run: 6.87/10, +0.00)
```

Line too longs, missing module docsting, snake_case naming style violation, unused variable and argument, exception “misuse”, unnecessary statements

First run Bandit:

No significant issues detected.

```
[main] INFO    profile include tests: None
[main] INFO    profile exclude tests: None
[main] INFO    cli include tests: None
[main] INFO    cli exclude tests: None
[main] INFO    running on Python 3.13.6
Run started:2025-10-09 17:22:20.983352
```

```
Test results:
    No issues identified.
```

```
Code scanned:
    Total lines of code: 135
    Total lines skipped (#nosec): 0
```

```
Run metrics:
    Total issues (by severity):
        Undefined: 0
        Low: 0
        Medium: 0
        High: 0
    Total issues (by confidence):
        Undefined: 0
        Low: 0
        Medium: 0
        High: 0
```

```
Files skipped (0):
```

ToDo based on static tests:

1. Iteration

- Line too longs
 - Test and verify
- Everything else
 - Samesame

```
***** Module payment_processing
Payment_Processing.py:104:15: W0718: Catching too general exception Exception (broad-exception-caught)

-----
Your code has been rated at 9.85/10 (previous run: 9.71/10, +0.15)
```

2. Iteration

```
PS C:\Users\ptk\SharedRepos\testaus-repo> pylint .\payment_processing.py

-----
Your code has been rated at 10.00/10 (previous run: 9.85/10, +0.15)
```

Dynamic tests on pytest:

```
----- test session starts -----
platform win32 -- Python 3.13.6, pytest-8.4.2, pluggy-1.6.0
rootdir: C:\Users\ptk\SharedRepos\testaus-repo
collected 6 items

payment_processing.py ..... [100%]

----- 6 passed in 0.00s -----
```