VIDYAVARDHINI'S COLLEGE OF ENGINEERING AND TECHNOLOGY **Vasai, India**	
Subject: CSL405	
Assistant Professor: Raunak Joshi	
Semester: IV	**Branches:** AI & DS
Deadline: 12th March 2025	**Academic Year:** 2024-25
Module 2: Advanced Python	

Course Outcome 2 - Use features of files, directories and regular expression in python for file manipulation.

CO2 - Apply Level

1. You are given a large log file containing various system events. Each line in the log file follows this format:

```
[YYYY-MM-DD HH:MM:SS] [LOG_LEVEL] [MODULE] Message
```

where:

- **YYYY-MM-DD HH:MM:SS** is a timestamp.
- **LOG_LEVEL** can be `INFO`, `WARN`, `ERROR`, or `DEBUG`.
- **MODULE** represents the system module name (alphanumeric, can contain underscores).
- **Message** is the actual log message (it may contain any characters).

Your Task

Write a function `extract_critical_errors(log_data: str) -> list[tuple]` that takes a multiline string `log_data` (containing log entries) and returns a list of tuples containing:

1. The timestamp
2. The module name
3. The error message

BUT only if:

- The **LOG_LEVEL** is `ERROR`.
- The message contains at least one **IP address** in IPv4 format (`xxx.xxx.xxx.xxx`, where `xxx` is in the range `0-255`).
- The message contains a **hexadecimal error code**, formatted as `0x` followed by exactly 8 hexadecimal digits (`0-9, A-F`).

Example Input

```
[2025-02-10 14:23:01] [INFO] [Auth_Module] User login successful.
[2025-02-10 15:45:32] [ERROR] [Net_Module] Connection timeout from
192.168.1.10. Error Code: 0xAB12CD34
[2025-02-10 16:01:10] [WARN] [Disk_Module] Low disk space warning.
[2025-02-10 17:12:05] [ERROR] [Security_Module] Unauthorized access detected
from 10.0.0.5. Error Code: 0xDEADBEEF
```

Expected Output

```
[
('2025-02-10 15:45:32', 'Net_Module', 'Connection timeout from 192.168.1.10.
Error Code: 0xAB12CD34'),
('2025-02-10 17:12:05', 'Security_Module', 'Unauthorized access detected from
10.0.0.5. Error Code: 0xDEADBEEF')
]
```

Constraints

- Your function **must** use **one single regex pattern** to extract the required information.
- You **cannot** use multiple regex calls; the full extraction must be done in **one pass** using `re.findall()` or `re.finditer()`.
- Assume `log_data` contains multiple lines.
- Make your regex IP-matching strict, ensuring that invalid IPs (e.g., `256.100.10.10`) are not mistakenly matched. (Optional)

Name= Bhgyashree Sutar

Roll no.=75

ASSIGNMENT 2

```
import re
```

Expression to match the required pattern

```
def extract_critical_errors(log_data: str) -> list[tuple]:
    # Regular expression to match the required pattern
    log_pattern = re.compile(
        r'\[(?P<timestamp>\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2})\] '  # 
Timestamp
        r'\[ERROR\] '                                               # 
Error Level
        r'\[(?P<module>[\w_]+)\] '                                  # 
Module Name
        r'(?P<message>.*)'                                          # 
Error Message
    )

    # Extract relevant log entries
    results = []
    for match in log_pattern.finditer(log_data):
        results.append((match.group('timestamp'), 
match.group('module'), match.group('message')))

    return results
```

Example log data for testing

```
test_log_data = '''
[2025-02-24 10:12:45] [ERROR] [Database_Module] Failed query from 
192.168.0.
[2025-02-24 11:30:22] [INFO] [User_Module] User profile updated.
[2025-02-24 12:45:00] [ERROR] [Payment_Module] Payment declined from 
10.0.0.
[2025-02-24 13:15:30] [ERROR] [Auth_Module] Invalid token access from 
172.16
[2025-02-24 14:00:00] [WARN] [Network_Module] Slow response detected.
'''
```

print Output

```
output = extract_critical_errors(test_log_data)
print(output)
```

```
[('2025-02-24 10:12:45', 'Database_Module', 'Failed query from
192.168.0. '), ('2025-02-24 12:45:00', 'Payment_Module', 'Payment
declined from 10.0.0. '), ('2025-02-24 13:15:30', 'Auth_Module',
'Invalid token access from 172.16 ')]
```