EXP NO: 6	Building anonymous FTP Scanner using ftplib module

Aim

Check a list of FTP servers for whether anonymous login is allowed (useful for authorized security assessments and hardening).

Procedure / Algorithm

1. For each host in the input list: attempt to connect on port 21.
2. Try ftp.login('anonymous','anonymous@domain').
3. On success, record host as anonymous allowed, optionally list root directory.
4. On failure or timeout, record not allowed or unreachable.

Code (Python, ftplib)

```
import ftplib
import socket

def check_anonymous(host, timeout=5):
    try:
        ftp = ftplib.FTP()
        ftp.connect(host, 21, timeout=timeout)
        ftp.login('anonymous', 'anonymous@example.com')
        try:
            listing = ftp.nlst()[:5]  # show up to 5 items
        except Exception:
            listing = []
        ftp.quit()
        return True, listing
    except (ftplib.error_perm, ftplib.error_reply, socket.error, OSError) as e:
        return False, str(e)

if __name__ == '__main__':
    hosts = ['ftp.debian.org', 'test.rebex.net']  # example
    for h in hosts:
        ok, info = check_anonymous(h)
        print(f'{h}: anonymous={ok}, info={info}')
```

Output:

```
ftp.debian.org: anonymous=False, info=[Errno 101] Network is unreachable
test.rebex.net: anonymous=True, info=['pub', 'readme.txt']
```

Result:

Building an anonymous FTP Scanner using ftplib module is successfully implemented.