```
# 2024 Â© Idan Hazay

import re

# Begin validation checking related functions
class Validation:
    """
    Provides validation methods for email, username, password, and input strings.
    """
    def __init__(self):
        self.illegal_chars = {'\'', '"', '>', '<', '~', '`', '|', '\\','/', '}', '{', '[', ']', '+', '=', ';', '(', ')'}
# Set of illegal characters

    @staticmethod
    def is_valid_email(email):
        """
        Validate an email address using a regular expression.
        """
        email_regex = r'^[a-zA-Z0-9_.+-]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-.]+$'
        return re.match(email_regex, email) is not None

    @staticmethod
    def is_valid_username(username):
        """
        Validate a username ensuring it is at least 4 characters long and alphanumeric.
        """
        return len(username) >= 4 and username.isalnum()

    @staticmethod
    def is_valid_password(password):
        """
        Validate a password ensuring it is at least 8 characters long, contains uppercase letters, and numbers.
        """
        return len(password) >= 8 and any(char.isupper() for char in password) and any(char.isdigit() for char in
password)

    @staticmethod
    def is_empty(list):
        """
        Check if any string in a list is empty.
        """
        return any(item == "" for item in list)

    def has_illegal_chars(self, input_str):
        """
        Check if a string contains any illegal characters.
        """
        return any(char in self.illegal_chars for char in input_str)

    def check_illegal_chars(self, string_list):
        """
        Check if any string in a list contains illegal characters.
        """
        return any(self.has_illegal_chars(s) for s in string_list)
```