In the Coding phase, different modules specified in the design document are coded according to the module specification. The major purpose of the coding phase is to use a high-level language to code from the design document created during the design phase, and then to unit test this code.

Code standards are a well-defined and standard style of coding that good software development companies expect their programmers to follow. They frequently create their own coding standards and rules based on what works best for their company and the types of software they create. Maintaining coding standards is critical for programmers; else, code will be rejected during code review.

Purpose of Having Coding Standards:

- A coding standard offers the programs created by different engineers.
- It increases the code's readability and maintainability while simultaneously reducing its complexity.
- It aids in the reuse of code and the detection of errors.
- It encourages good programming habits and boosts programmers' productivity.

Some of the coding standards are given below:

1. **Limited use of global:**

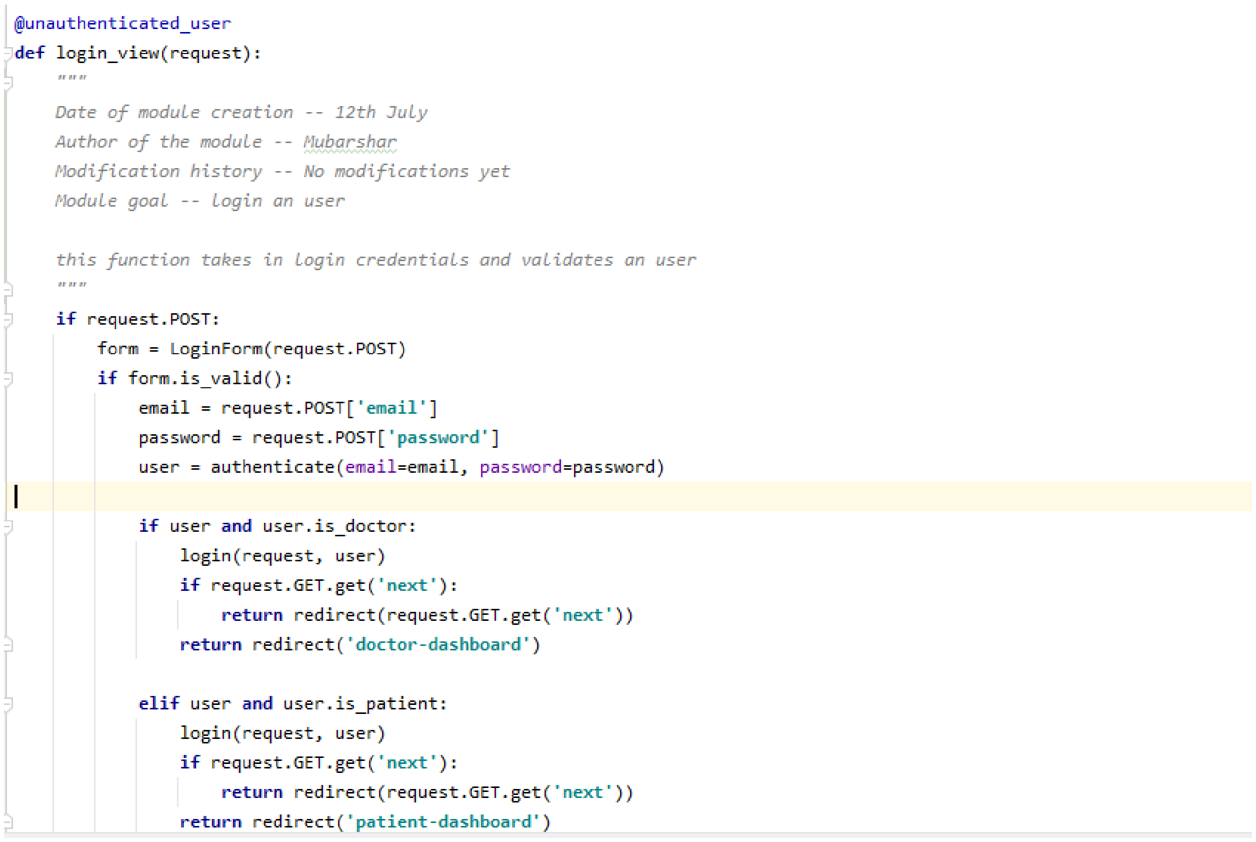
   These rules specify which types of data are allowed to be declared global and which are not.

2. **Standard headers for different modules:**

   The headers of distinct modules should follow a common format and information for better understanding and maintenance of the code. The header format must have the following items, which are often utilized in different companies:

   - The module's name
   - Module creation date
   - The module's creator
   - History of changes
   - The module's synopsis describes what the module performs.
   - The module supports a number of functions, each with their own set of input and output parameters.
   - Module-accessible or modifiable global variables.

Example:-

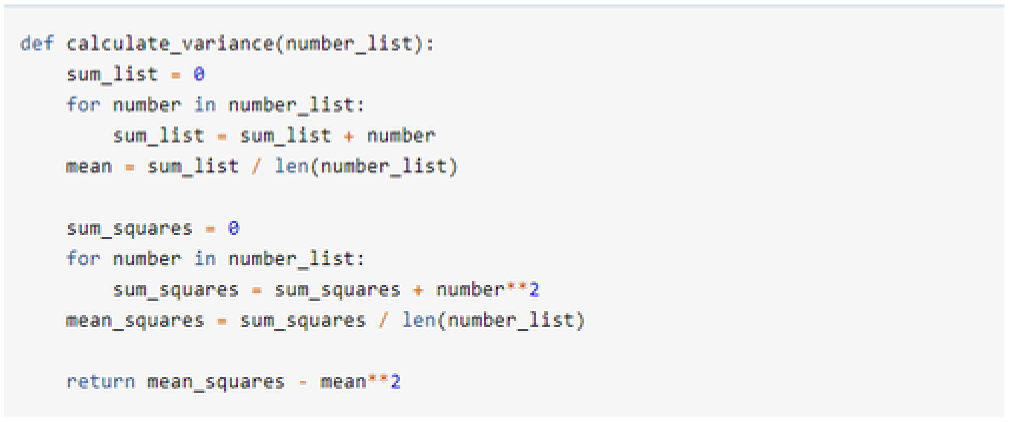


```
@unauthenticated_user
def login_view(request):
    """
    Date of module creation -- 12th July
    Author of the module -- Mubarshar
    Modification history -- No modifications yet
    Module goal -- login an user

    this function takes in login credentials and validates an user
    """
    if request.POST:
        form = LoginForm(request.POST)
        if form.is_valid():
            email = request.POST['email']
            password = request.POST['password']
            user = authenticate(email=email, password=password)

            if user and user.is_doctor:
                login(request, user)
                if request.GET.get('next'):
                    return redirect(request.GET.get('next'))
                return redirect('doctor-dashboard')

            elif user and user.is_patient:
                login(request, user)
                if request.GET.get('next'):
                    return redirect(request.GET.get('next'))
                return redirect('patient-dashboard')
```

3. **Naming conventions for local variables, global variables, constants and functions:**

The following are some of the naming conventions:

- The use of variables with meaningful and intelligible names makes it easier for everyone to understand why they are being used.
- Local variables will be like ("name" if it is single word, "user_name" if the variable sonsists of two words) Global variables will be like ("User_Name")
- It's preferable to avoid using digits in variable names.
- The function names should begin with small letters
- The function's name must clearly and succinctly describe why the function is being used.



```python
def calculate_variance(number_list):
    sum_list = 0
    for number in number_list:
        sum_list = sum_list + number
    mean = sum_list / len(number_list)

    sum_squares = 0
    for number in number_list:
        sum_squares = sum_squares + number**2
    mean_squares = sum_squares / len(number_list)

    return mean_squares - mean**2
```

4. **Indentation:**

   To improve the readability of the code, proper indentation is critical. White spaces should be used properly by programmers to make their code readable. The following are some examples of spacing conventions:

   - After a comma, there must be a space between two function arguments.
   - Each nested block should be indented and spaced properly.
   - Each block in the program should have proper indentation at the beginning and finish.
   - All braces should begin on a new line, and the code following the braces should begin on a new line as well.

5. **Avoid using a coding style that is too difficult to understand:**

   The code should be simple to comprehend. Maintenance and debugging are difficult and expensive due to the sophisticated code.

6. **Code should be well documented (easy to read):**

   The code should be well-commented so that it is easy to understand. The code is made more understandable by adding comments to the statements.

7. **Length of functions should not be very large:**

   Long functions are quite tough to comprehend. As a result, functions should be tiny enough to perform minor tasks, and larger functions should be split down into smaller ones to complete minor tasks.