

Coding Standard 1

Project: University E-Voting System

Primary Language: Python (with support for JS, HTML/CSS)

Version: 1.0

Last Updated: 24 October 2025

Author: Tamim Ahmed Rijan (TAR)

1. General Guidelines

1. Code should be clean, consistent, and well documented.
2. Each file must have a header comment describing its purpose, author, and last modified date.
3. Use meaningful variable and function names, avoid one-letter variables except in loops (i, j).
4. Keep lines under 80-100 characters for better readability.
5. Always use 4 spaces for indentation (no tabs).
6. Comment logic, not the obvious (why, not what).
7. Remove unused imports, dead code, and print/debug statements before committing.

2. Python Coding Standards:

Element	Convention	Example
Variables	snake_case	total_votes, user_count
Functions	snake_case()	calculate_result(), get_voter_info()
Classes	PascalCase	ElectionManager, VoteCounter
Constants	ALL_CAPS	MAX_USERS, TOKEN_EXPIRY
Private Members	_leading_underscore	_connect_db()
File Names	lowercase_with_underscores	vote_handler.py, user_auth.py

Example:

```
app.py > ...
1  class Voter:
2      """Represents a registered voter in the e-voting system."""
3
4      MAX_ATTEMPTS = 3
5
6      def __init__(self, name: str, voter_id: str):
7          self.name = name
8          self.voter_id = voter_id
9          self.is_verified = False
10
11     def verify_voter(self, otp: str) -> bool:
12         """Verify voter using one-time password."""
13         if otp == "1234":
14             self.is_verified = True
15             return True
16         return False
17
```


Additional:

Element	Convention	Examples	Notes
Modules / Files	lowercase_with_underscores	data_loader.py, config_parser.py	Keep names short & descriptive.
Packages / Directories	lowercase	utilities, core	Avoid special characters.
Private Members	_leading_underscore	_helper_function, _internal_var	Indicates internal use only.
Strongly Private	__double_leading_underscore	__password	Prevents name conflicts (mangling).
Global Variables	snake_case	global_cache, app_version	Use only when necessary.
Instance Variables	snake_case	self.user_name	Same as normal variables.
Class Variables	snake_case	counter = 0	Declared inside class, outside methods.

3. JavaScript / Frontend Standards

Element	Convention	Example
Variables	camelCase	voteCount, userToken
Functions	camelCase()	handleVote(), fetchResults()
Constants	UPPER_CASE	MAX_RETRIES, BASE_URL
React Components	PascalCase	VoteButton, AdminDashboard

Example:

```
JS app.js >  castVote
1 function castVote(voterId, candidateId) {
2   if (!voterId || !candidateId) {
3     console.error("Invalid input: voterId and candidateId required.");
4     return false;
5   }
6
7   const vote = {
8     voterId,
9     candidateId,
10    time: new Date().toISOString(),
11  };
12
13  console.log(`Vote recorded for candidate ${candidateId}`);
14  return vote;
15 }
16
17
```

4. HTML / CSS Standards

4.1 HTML

- Use semantic HTML tags (<header>, <section>, <article>, <footer>).
- Maintain proper indentation (2 spaces per level).
- Close all tags properly.
- Use lowercase for all tags and attributes.
- Avoid inline styles; use external CSS or utility classes.

Example:

```
<> voter.html > ...
1   <form action="/vote" method="post">
2   |   <label for="voterId">Voter ID:</label>
3   |   <input type="text" id="voterId" name="voterId" required />
4   |
5   |   <label for="candidate">Select Candidate:</label>
6   |   <select id="candidate" name="candidate">
7   |       <option value="c1">Alice</option>
8   |       <option value="c2">Karim</option>
9   |   </select>
10  |
11  |   <button type="submit">Cast Vote</button>
12  </form>
13  |
```