

Establishing Coding Standards for EduLink

Presented by: Asmin ara mim & Fahmida
afrin
Software Testing & Maintenance Lab
Course

The Necessity of Code Standards

Clarity and Consistency are Key: Without established guidelines, codebases quickly become inconsistent and disorganized, creating a confusing and fragmented project structure.

Maintenance and Debugging Efficiency: A lack of standards directly increases the difficulty of reading code, making it time-consuming to maintain and significantly complicating the process of identifying and fixing bugs.

Team Performance and Professionalism: The resulting inconsistencies hinder collaboration among team members and project handoffs, ultimately reducing the overall quality and professional standard of the software product..

Why Automate?

The Solution: Automation

Instead of relying on human memory, we should automate rule enforcement to eliminate the cognitive burden on developers. This shifts the focus from remembering tedious rules to concentrating on solving technical problems.

1. **ESLint: The Logic Enforcer** This tool acts as a static analysis linter, actively flagging potential logic errors, security vulnerabilities, and anti-patterns. It ensures the code not only looks good but also works reliably and adheres to best practices.
2. **Prettier: The Style Harmonizer** Prettier is a dedicated opinionated code formatter that automatically handles the visual appearance of your code. It ensures uniformity in elements like spacing, indentation, line wrapping, and semicolons, eliminating style debates and making all code look the same.

Defining Our Coding Policy

- **Core Standard:** We are adopting the **Airbnb JavaScript Style Guide** as our foundation.
- **Why Airbnb?** It is a popular, well-documented industry standard that fully supports both modern **JavaScript (ES6+)** and **React** development.
- **Customization:** We will add specific rules for **EduLink** to ensure alignment with our internal project needs.

Key Standards We Will Follow

Category	Rule	Example (for EduLink)
Variables/Functions	Use camelCase .	loadAnnouncement, sendNotification
React Components	Use PascalCase .	AnnouncementFeed, MessageCenter
Declaration	Must use const or let (Avoid var).	
Strings	Use single quotes (' ') and template literals (` `).	'New Email', $\text{\text{\`Welcome, \${user}\`}}$
Functions	Prefer arrow functions for concise callbacks.	const onClick=()⇒{ }
Modules	Use ES6 import/export exclusively.	import { LinkGeneratorModel } from '...'
Custom Rule	Role Identification: User-related object names must include the user's role.	Good: studentProfileLink / Bad: userProfile

Deployment Plan

1. Project Setup (Configuration)

- **Install Dependencies:** Add **ESLint**, **Prettier**, and the **Airbnb config** to the `package.json` file.
 - **Define Rules:** Create and customize the configuration files (`.eslintrc.json` and `.prettierrc`) to include the Airbnb standards and our specific **EduLink** rules.
-

2. Developer Workflow (Automation)

- **Editor Extensions:** Install the necessary **ESLint and Prettier extensions** in VS Code to get instant visual feedback while you code.
 - **Auto-Apply Rules:** Enable **'Format on Save'** in your editor settings. This ensures the rules are applied automatically every time you save a file, making style compliance effortless.
-

3. Quality Gate (Stretch Goal)

- **Pre-Commit Checks:** Implement **husky + lint-staged** to run all quality checks *before* code is officially committed to the repository. This guarantees that only clean, compliant code enters our main branch.

Final Action Plan

- **Decision:** The team officially agrees to use **ESLint + Prettier + Airbnb** standards.
- **Initial Setup:** One teammate will create a **Pull Request (PR)** containing all the new configuration files.
- **Team Review:** Everyone must **review and approve** the setup PR, then pull the updated branch.
- **Local Install:** Run `npm install` and set up the **ESLint & Prettier extensions** in your editor.
- **Discussion:** All further rule questions or feedback should be discussed in the project's **Discord channel**.

Q&A and Policy Validation

- Any immediate concerns with the new technical rules?
- What EduMail-specific rules should we add?
- Open Q&A and final feedback to ensure full team alignment.