

Group 2

Programming using Linters and Formatters

GROUP 2 MEMBERS:

- Ninah Mutesi
- Birungi Faimah
- Ashley Masajjage
- Laura Apiyo
- Billy Bahirwe
- Felicity Akol
- Veronica Amooti
- Shallon Ayebare

CONTENT:

- An overview of what linters and code formatters are.
- The benefits of using them in software development projects.
- A presentation on ESLINT
- The rules and guidelines enforced by each tool
- Demo on setting up and configure the ESLINT tool and PRETTIER

Overview of Linters and Code Formatters

- **Linters:** A tool that analyzes source code to detect errors, enforce coding standards, and highlight potential issues before execution.
- **Code Formatter:** A tool that automatically restructures code to maintain a consistent style by adjusting indentation, spacing, and other formatting rules.

Benefits

- **Improves Code Quality** – Detects errors early.
- **Ensures Consistency** – Unified coding style.
- **Saves Time** – Reduces manual code reviews.
- **Enhances Collaboration** – Easy-to-read, structured code.
- **Reduces Bugs** – Prevents common mistakes.
- **Facilitates Maintenance** – Clean, organized code.

3. ESLint: A JavaScript Linter

Key Features

- Custom rules & plugins.
- Detects syntax issues & best practices.
- Integrates with IDEs (VS Code, WebStorm, etc.).

Rules Enforced

1. **Stylistic Issues** – Indentation, quotes, semicolons.
2. **Best Practices** – Unused variables, consistent returns.
3. **Possible Errors** – Undefined variables, unreachable code.
4. **ES6+ Features** – `const` vs `var`, arrow functions.
5. **Plugin Support** – Airbnb, Prettier integration.

4. Demo: Setting Up ESLint & Prettier

Step 1: Install ESLint

```
npm init -y  
npm install eslint --save-dev
```

Step 2: Initialize ESLint

```
npx eslint --init
```

- Select environment & style guide.

Step 3: Install Prettier

```
npm install --save-dev prettier eslint-config-prettier eslint-plugin-prettier
```

Step 4: Configure ESLint & Prettier

Modify `.eslintrc.json`:

```
{
  "extends": ["eslint:recommended", "plugin:prettier/recommended"],
  "rules": {
    "prettier/prettier": ["error", { "singleQuote": true, "semi": false }]
  }
}
```

Step 5: Run ESLint & Prettier

`npx eslint . --fix` # Auto-fix linting issues

`npx prettier --write .` # Format code

Conclusion

ESLint and Prettier automate code quality and formatting, ensuring consistency, readability, and efficiency in development.