

Code Comments Specification

Code comments is a platform to help you take any repository and add comments to it, which can be used for code reviews, feedback, or any other purpose. It is designed to be simple and easy to use, with a focus on clear and consistent commenting so you can help your students improve their code quality.

Core Features

- **Manager Interface:** A simple interface for teachers to manage their projects and comments. This includes the ability to view, edit, and delete comments.
- **Commenting:** Teachers can add comments in various ways:
 - **Singleline Comments:** Teachers can leave comments on individual lines of code.
 - **Multiline Comments:** Teachers can leave comments on multiple lines of code.
 - **File Comments:** Teachers can leave comments on entire files. This means that teachers can provide feedback on the overall structure or content of a file.
 - **Folder Comments:** Teachers can leave comments on entire folders. This may allow teachers to provide feedback on the whole project organization or structure.
- **Editor-like Feel:** The commenting interface should feel like an editor, allowing teachers to easily navigate and comment on code.
- **Comment Extraction:** Comments should be extractable to a separate file, such as a Markdown file or a JSON file for easy sharing and review (raw comments).

User Roles & Permissions

- **Teachers:** Add comments, edit or delete their comments, and manage their projects (full access). They are the primary users of the platform and have full control over the comments they leave.
- **Students:** They are provided a special link by the teacher where they can view comments on their code, but cannot edit or delete them (read-only access). No registration is required for students to view the comments.

Workflow

Teacher Workflow Example

1. A teacher receives a link to a Github repository from a student.
2. The teacher opens the repository (providing the repo url and the branch he wants to comment on) in the Code Comments platform.
3. The teacher navigates through the codebase, viewing files and folders.
4. The teacher adds comments on specific lines of code, files, or folders as needed.
5. The teacher generates a link to share with the student, which includes all the comments made.

Student Workflow Example

1. A student receives a link to their commented code from their teacher.
 - The link is a read-only view of the comments made by the teacher.
2. The student views the comments on their code, mainly using the summary page where they are provided in a structured format.
3. The student can navigate to the specific lines of code where the comments were made and see the context of the comments.

Potential Additional Features

- **User Accounts:** Allow users to create accounts to manage their comments and projects.
 - I think this this should be added for the teachers, so they can manage their comments and projects more easily.
- **Threaded Commenting:** Allow multiple replies to a comment for better discussion. Goes hand in hand with user accounts.