

## Summer 2022

### Code Review (Thursday, July 21<sup>st</sup>, 11pm PST)

The deliverable for this milestone is automated and manual code reviews. You will:

- Perform automated code review of both front-end and back-end of **your own** project.
- Perform manual code review of both front-end and back-end of **your peer group project**.
- Analyze and fix issues identified by your own and your peer group reviews.

Make sure to meet your peer group (online or in person) way before the deadline!

**Submission.** The submission for this milestone is a single **pdf** file that must include the following information:

#### **PART I.** Project overview.

1. The up-to-date description of your project, its requirements, and its design (as in M3-M6). Please include the full report; there are no space limitations and we need to have this information for context. You do not need to include test design at this stage **[3 points]**.
  - NOTE: This part will not be regraded but must be included for context.
2. Explicitly list the changes you made from previous reports for your project scope, requirements, and design, if any **[5 points]**.
  - If no changes were made, state “None”.
  - If there are changes, specify the reason for the change and provide a description of the new artifact as required in the previous milestones. You can either duplicate the artifact or point to its description in item 1.

#### **PART II.** Automated Code Review.

3. Set up Codacy (<https://www.codacy.com/>) – an automated code review tool – to work on your repository **[5 points]**
  - NOTE: Codacy is free for public repositories only; if you prefer to keep your repository private, you can use the 14-day free trial and pay \$15/month afterwards.
    - You can reach out to the course staff through a private Piazza post if you have a strong reason to keep the repository private and need help financing the monthly fee.
  - You can use the following instructions to get you started with Codacy: (<https://docs.codacy.com/getting-started/codacy-quickstart/>)
  - In the “Built-in tools” section of the “Code patterns” page, enable the code checking tools listed below and disable all other tools:

- Checkstyle for Java
- ESLint 8.18.0 for JavaScript and Typescript
- PMD for Java and JavaScript
- TSLint for TypeScript (only if your project uses TypeScript)
- Download *M7ConfigFiles.zip* from the Canvas Files section, unzip it, and push all individual files (.eslintrc.json, checkstyle.xml, ruleset.xml, tslint.json) to the **root** directory of the **main** branch.
- Configure Codacy to use the selected configuration files, following the instructions here: <https://docs.codacy.com/repositories-configure/configuring-code-patterns/#using-your-own-tool-configuration-files>.
- In your report, list the link to the Codacy “Repository Dashboard” page of your project repository .

**NOTE:** Do not alter the configuration files or Codacy setup in any way. Specifically, do not disable any Codacy checks, do not exclude any files, do not use inline code comment to suppress issues, etc. **Altering the setup will cause losing all marks for the entire milestone.**

4. Run Codacy in the main branch on the latest release of your project **[15 points]**.
  - This will likely be your MVP version, but you can also push a more advanced version into the main branch, if you have one available.
  - In your report, list the commit SHA that Codacy runs on.
  - Report the number of identified issues in each of the following categories: code style, security, error prone, performance, compatibility, code complexity, documentation, and unused code. For simplicity, you can copy Codacy’s “Issues Breakdown” table into your report.
  - Report the number of issues for each of the identified issue patterns (again, you can copy it directly from Codacy). For each issue pattern, give one concrete example from your code.
5. Fix the identified issues and push the fixes to the main branch. Run Codacy again on the main branch after the issues were fixed **[30 points]**.
  - In your report, list the commit SHA that Codacy runs on.
  - Report the number of identified issues in each category and the number of identified issues for each pattern after the fix, as in item 4.
  - To receive a full mark, there should be either 0 issues left or every issue that is left needs to be thoroughly justified. That is, there should be a well-formed and well-documented reason not to fix Codacy issues, with appropriate justifications and with citations to reputable sources.
    - Marks will be deducted justifications that are opinion-based (e.g., a opinion on StackOverflow).
    - In reality, it is often easier to fix the issues than to provide sufficient evidence for not fixing it; points will be deducted for remaining issues that do not have an adequate justification.

### **PART III. Manual Code Review.**

6. Meet with your peer group and review their front-end and back-end code. Ask them to review your front-end and back-end code.
7. Report two major issues you found in **your peer group code [20 points]**.
  - Major issues are those that typically cannot be found by automated tools and relate to security, inefficiency in algorithms, etc. They should not be minor issues such as comments.
  - Yes, you will find some major issues – there is no code without issues.
  - If you report that you cannot find any major issues in your peer group code and then TAs do, you will lose marks. However, if TAs do not find any issues either, you will get the full mark.
8. Report two major issues that your peer group found in **your code [5 points]**.
  - These two issues should match the issues reported by your peer group, verbatim. I.e., agree on and use the same text in both reports.
9. Fix the issues identified by your peer group **[15 points]**.
  - In your report, specify how you addressed these issues, e.g., by showing screenshots of the corresponding code before and after the change and providing 2-3 sentence explanations about each fix.
  - If your peer group did not find any major issues in your code, ask your teammates working on the back-end to review front-end code and vice-versa. Report on and fix the identified issues.
  - If you report that neither your peer group nor your own group member can find major issues in your code and then TAs do, you both will lose marks. However, if the TAs do not find any issues in your code, you will get the full mark.
10. Report on the specific contribution of each group member to the work done for this milestone, in 1-2 sentences per member. **[2 points]**

Good luck!