# CSC309 Project Grading Rubrics

**Last Updated on November 10.**

**For Students' References Only, some minor revisions might be made.**

The Spreadsheet: https://www.panchen.ca/csc309-project-grading-tables

The purpose of this document is to provide information on how we are grading your project in a one-hour long grading interview. We've defined the following terms:

1. Grader: The TA who is assigned to your TA will be the grader for your team's project.
2. Your Team/The Team/Students: 1 – 4 students who work on the project together.
3. You/Student: About an individual team member.
4. The Project: the software your team is going to present.
5. Grading Interview: the one-hour long session when Your Team meets Grader.

As a summary, the project gradings will be done by 300 points, with 16 possible bonus points:

**The final grade for project will be MIN(300, pointsEarned + bonusEarned) / 300.**

| | |
|---|---|
| Basic Features | 190 points |
| Advanced Features / Improvements / Highlights | 50 points |
| Deployment and Reliability | 45 points |
| Code Quality and Organization | 15 points |
| Bonus | At most 16 points |

**On High-Level**, During the grading interview, it is you and your team's opportunity to clearly demonstrate and convince us during the grading interviews that Your Team does a good job from the following perspectives:

1. Functionality and Requirements Fulfillment & Improvement
2. Technical Understanding and Explanation
3. Code Quality and Organization
4. System Design and Architecture
5. Deployment and Reliability
6. Error Handling and Security Practices

7. Professionalism and Presentation Quality

# Functionality and Requirements Fulfillment

We encourage you to also read over CSC309 Project Information.docx for some general requirements including QR codes.

## How are we going to grade?

For each of the functionality and requirements mentioned in this section, we will grade based on the following criteria:

Note that: The Grader might apply a discount to any of these points if the students can not explain how they implemented each feature. For more details, please check <Technical Understanding and Explanation>.

### Aesthetics: 30%

- The user interface should be visually appealing;
- The website should offer a smooth user experience.
- The theme and design should be consistent across pages, and each page should be responsive, ensuring it looks good on both desktop and mobile devices.
- As the TAs are going through each web page you demonstrate, you can lose marks for UI issue, including, but not limited to, poor alignment of elements, counterintuitive navigation, broken links, poor color contrast (not accessible for color-blind users), hard-to-see content, e.g., tiny font sizes, clutter, i.e., too much information on one page, and lack of feedback when using interactive elements, e.g., buttons and inputs do not change style on hover or focus.

### Functionality: 50%

- The happy path of each page should work as described.
- For pages where it is reasonable, it is expected to allow a user to bookmark a specific search result.
- For pages where it is reasonable, it is expected to implement pagination and infinite scrolling.

### Error handling: 20%

- Errors are handled appropriately for each page, where applicable.
- When a page does not have error handling for a valid reason, the 20% will be shifted into Aesthetics and Functionality evenly.

## List of Basic Features (190 points )

- ❑ Landing Page (10 marks)

- Regular users (3 points):
  - Dashboard showing points balance and recent transactions.
- Cashiers (3 points):
  - Quick access to transaction creation and redemption processing.
- Managers & Superusers (4 points):
  - Overview of events, promotions, and user management.

- ❑ Accounts (60 marks)
  - Login (5 marks):
    - Users can log in with their credentials.
  - Registration (5 marks):
    - Cashiers (or higher) can create accounts for users.
  - Profile Management (5 marks):
    - Users can update their account information and passwords.
  - Password Reset (5 marks):
    - Users can reset their password if they have forgotten.
  - Interface Switching (15 marks):
    - Users can switch between different interfaces (e.g., cashier, event organizer, manager).
    - For example, a manager can switch to the regular user interface to accumulate points, while a regular user who is also an event organizer can switch to the organizer interface to manage events.
  - User Listing (15 marks): Managers can view a list of users.
  - User Management (10 marks): Mangers can verify a user and make a cashier either suspicious or not. Managers or higher can promote/demote users.
- ❑ Transactions (50 marks)
  - Purchase (10 marks):
    - Cashiers can create purchase transactions.
    - Cashiers can correctly apply promotions to transactions.
  - Redemption (10 marks):
    - Users can make a redemption request.
    - Cashiers can process the redemption request.
  - Adjustment (5 marks):
    - Managers can create adjustment transactions.
  - Transfer (10 marks):
    - Users can transfer points to another user.
  - Transaction Listing (15 marks) Users can see their past transactions. Managers can see ALL transactions.
- ❑ Events (45 marks)

- Event Management (15 marks):
  - Managers can create, update, and delete events.
  - Managers can add event organizers.
  - Event organizers can update events.
- Event Listing (15 marks):
  - All logged in users can view the list of events.
- RSVP and Attendance (10 marks):
  - Users can RSVP to events.
  - Managers and event organizers can add guests to an event.
  - Managers can remove guests from an event.
- Point Allocation (5 marks):
  - Managers and event organizers can award points to guests.

❑ Promotions (25 marks)

- Promotion Management (10 marks):
  - Managers can create, update, and delete promotions.
- Promotion Listing (15 marks):
  - All logged in users can view the list of promotions.

## Advanced/Improved Features (50 points )

You must mention the advanced/improved features explicitly. Otherwise we might consider it as part of the basic features. You can highlight features your team is most proud of.

Teams are encouraged to go beyond the basic requirements by developing **advanced or improved features** that demonstrate creativity, technical depth, or exceptional polish.

Each improvement can be worth **up to 50 points**, depending on **its impact, usefulness, complexity, and quality of implementation**.

- **Innovation:** Introduces new or non-trivial functionality not required by the basic spec.
- **Technical Depth:** Demonstrates a strong understanding of advanced concepts (e.g., caching, API integration, state management, or performance optimization).
- **User Value:** Improves usability, accessibility, or user experience in a meaningful way.
- **Polish:** Exceptional UI/UX quality or attention to details beyond expectations.
- **Integration:** Works smoothly with existing functionality without breaking core features.

**Examples (not ordered by how many points they are worth for):**

- Using a database other than SQLite3.
- Implementing live updates or real-time notifications using websockets.
- Adding search and filtering with pagination and sorting.
- Integrating third-party APIs (e.g., Google Maps, email services).
- Introducing data visualization of dashboards.
- Improving performance (lazy loading, caching, code optimization).
- Advanced accessibility or internationalization support.

**The following may not be considered as an improvement:**

- Display birthday also on a user's landing page.
- Give the cashier a subset of the manager's permissions (without a good reason).

**Note:**

- All **Aesthetics (30%)**, **Functionality (50%)**, and **Error Handling (20%)** requirements apply to these features as well.

Your team's Advanced Features / Improvements mark is based on the quality and impact of your enhancements, not the quantity.

- A single, well-designed improvement can earn full marks for this section.

- Conversely, implementing many small or low-impact features may still result in no additional marks.

If you are unsure whether an improvement is considered major or minor, you are encouraged to discuss it with your TA during tutorials before the grading interview.

# Code Quality and Organization (15 points)

The students need to show reasonable code quality and organization for their projects.

- Code quality (5 points)
- Organization(10 points)

# Deployment and Reliability (45 points)

## Public URL (10 points)

- The website must be deployed and accessible via a public URL.
- The application should be reachable on the internet without requiring local setup.
- The deployed version must reflect the same functionality as the version demonstrated during the grading interview.
- The URL and any necessary demo credentials must be clearly provided in your README or submission document.
- Full marks (10) will be awarded only if the grader can successfully access and interact with the deployed website.

## Security Practice (10 points)

The team must demonstrate awareness and application of basic security practices. Below are some examples:

- The website should use HTTPS to ensure secure data transmission.
- Sensitive data (such as passwords, tokens, or API keys) must not be exposed in the codebase or client-side storage.
- Inputs should be validated and sanitized to prevent common security issues such as SQL injection or XSS.

## Pre-populated Database (10 points)

A pre-populated database must be available before the scheduled grading interview.

- The database should include sample users, transactions, events, and promotions so that the grader can evaluate all key functionalities immediately.
- Credentials for different roles (e.g., manager, cashier, regular user) should be clearly provided in the documentation.

### Documentation (15 points)

The team must provide clear and complete deployment instructions in their documentation.

- Instructions should describe how to run the project **locally** and in **production**, including all setup steps, dependencies, and environment configurations.
- The documentation should list all available demo accounts, URLs, and any third-party services integrated.
- A short summary of the project's architecture and **technology stack** is recommended.

## Technical Understanding and Explanation ( 🚨 )

During the grading interview, the Grader will ask questions to assess each student's technical understanding of their project. Students are expected to use appropriate technical terms and concepts learned in the course to clearly explain how their system works.

If there is **insufficient evidence of understanding or explanation**, a **mark deduction (discount)** may be applied to the related component.

This deduction may apply to **the entire team** or to **an individual student** who was responsible for that part of the project, at the grader's discretion.

A deduction may result in a zero for the affected portion but will not reduce marks below zero.

## Bonus

### Show TA your progress and ask for feedback (up to 10 bonus points)

We strongly encourage every team to attend the final three tutorial and discuss your project progress with your TA (who is also your grader).

Your team can earn up to 10 bonus marks based on your participation and progress demonstration during these tutorials:

- Week 10 Tutorial: Meet your TA and discuss your project plan (2 marks)
- Week 11 Tutorial: Demonstrate reasonable progress and receive feedback (3 marks)

- Week 12 Tutorial: Present a nearly complete version of your project (5 marks)

Teams that attend all three tutorials and show meaningful progress each week will receive the full 10 bonus marks.

## Meaningful integration of third-party services (up to 5 bonus points)

Teams can earn up to 5 bonus marks for a meaningful integration of third-party services or APIs.

This bonus is awarded separately from the Advanced Features / Improvements / Highlights category.

*For example, if one of your improvements earns 20 marks and it also involves a meaningful use of a third-party API, your team will receive an additional 5 bonus marks, for a total of 25 marks for that improvement.*

To qualify, the integration must:

- Provide clear technical value (not trivial embedding or display).
- Be reliably functional and integrated smoothly into your system.
- Be demonstrated during the grading interview.

## Give your cool project a cool name! (up to 1 bonus point)

Give your project a cool, creative, and unique name to earn 1 bonus mark!

- The name should be original. If two or more teams choose the same name, the instructor will run a small Node.js script to generate a random number.
- Teams whose team number has the same parity (i.e., both even or both odd) as the generated number will receive the bonus mark.
- The name should appear clearly on your project's landing page and documentation.

# Tips

The Grader will complete all evaluations during the one-hour grading interview. This is your team's opportunity to demonstrate your project's strengths and convince the grader that your work meets (and exceeds) expectations.

- Highlight your best features: Focus on the functionalities and improvements that you are most proud of. Make sure they are easy to access and clearly demonstrated during the session.
- Be confident and professional: Remember — this is your project. You and your teammates understand it better than anyone else. Explain your design choices and implementation clearly, using the technical terms learned in class.
- Engage with your TA: Your TA is also your grader. Attending tutorials regularly helps them understand your progress and design decisions. The better your TA knows your project, the easier it will be for them to appreciate the depth and quality of your work during grading.

In short, treat the grading interview as a technical presentation — demonstrate functionality, explain your design, and show confidence in the work your team has built.

You will be doing great!