Master of Science Project

Rubric-Feedback Generator

A Course Management System Module with a Focus on Assessment

Jennifer Jean Schiffer

Montclair State University

Computer Science Department

Faculty Advisor: Dr. Emily Hill

**Acknowledgments**

None of this would have been possible without my advisor, Dr. Emily Hill, the rest of the faculty of Montclair State University's Computer Science Department, the Dean's Office of the College of Science and Mathematics, and the irreplaceable J. Francis.

This project was a long time coming. It has become a symbol of my individuality and longstanding belief in personal freedom.

Table of Contents

I. Abstract 4

II. Introduction 5

Purpose 5

Scope 5

Definitions 5

III. Software Development Process 7

Requirements & Specifications 7

Stakeholder Requirements 7

Functional Requirements 7

Non-functional Requirements 8

Architecture & Design 8

Database Architecture 8

Implementation 12

Programming 12

Database Installation 12

Deployment 14

IV. Using Rubric-Feedback Generator 15

Login 15

Register 16

User Admin 19

Edit User Information 19

Create New Class 21

Edit Class 23

Delete Class 25

View Assignments 26

Create New Assignment 27

Edit Assignment 30

Delete Assignment 32

Grade Assignment/Create New Grade 34

View Grades 36

Edit Grade 37

Delete Grade 40

Email Grade 41

Batch Email Grades 43

Create New Rubric 46

Edit Rubric 52

Delete Rubric 54

Import Rubric 56

V. Source Code 62

Bibliography 63

# I. Abstract

As Montclair State University, including the Computer Science department, adapts to the new trends of online education, a new challenge the faculty faces is bridging the gap between the different technologies available for course and student management for both online and face-to-face students. Course management systems and online communication applications currently used in online courses collectively provide solutions to posting assignments, announcements, course slides, and video conferencing. What these systems lack is the functionality needed to create rubrics for assignments and allow faculty to easily provide feedback to students using them. These activities are otherwise done on paper, in separate spreadsheets, or email, which is difficult and impractical for instructors and students participating in online courses.

I have proposed the design and implementation of a Web-based assignment rubric and feedback generating system that allows users to easily create rubrics for their assignments, complete those rubrics, and automatically generate and send grades and feedback to their students, all electronically.

Rubric-Feedback Generator is built on a similar infrastructure to the Computer Science Department’s Teaching Information System (TIS), using MySQL as the relational database management system and PHP for server-side scripting as a means of communicating with the database in a secure and efficient way. Being Web-based provided that any online or face-to-face students and staff, with an Internet connection and proper credentials, will be able to use the system. Rubric-Feedback Generator will serve the needs of managing student feedback in an online environment, with the ability to integrate into already-implemented systems like TIS. The system provides structure and consistency in terms of rubric formatting, and efficiency in providing and sending feedback to students. The project has a potential impact beyond the Computer Science department and Montclair State University, as it can be useful in any environment that involves offering feedback to users based upon a rubric.

# II. Introduction

Assessment is an important part of any space in education, and it has become common to use course management systems to provide content to students, collect assignments, and post grades and feedback of those assignments. Blackboard is used by Montclair State University for these purposes, but its focus on assessment is shallow. Rubric-Feedback Generator was born out of the Computer Science Department’s need for a course management system tool that focuses on assessment, along with my vested interest in assessment at a university departmental level. I wanted to design and build a Web-based application that allowed an instructor to build a rubric as a web form, complete the rubric form, and disseminate the results of grading and feedback to students. The data from those rubrics, as well as the rubrics themselves, would be saved in a database for later use, such as assessment analytics and study of trends.

## Purpose

The purpose of this project is to design and develop a web application that would allow a course instructor to manage rubrics and assignments for a class, grade assignments, and disseminate grades and feedback to students.

## Scope

Rubric-Feedback Generator will give users the ability to create classes, build rubrics, add assignments for a class using a specific rubric, grade assignments using rubrics, and send grades and feedback to a student via email. There is only one role, “Instructor”, which would be fulfilled by faculty, adjuncts, graduate assistants, and administrators, and has authorization to perform any functionality of Rubric-Feedback Generator.

## Definitions

**Answer:** Values entered by a User within a Criteria object of a Rubric when creating a Grade. A User submits Answers as content of an Email to a Student.

**Assignment**: An assessment given to individuals of a single Class by a User.

**Class**: A grouping of individuals, typically students, in a single course section lead by an instructor.

**Criteria:** An object, Textbox or Radio, within a Rubric that a User submits a value for when creating a Grade.

**Email:** The electronic dissemination of a Grade by a User to an individual part of a Class.

**Grade**: A collection of data from a completed Rubric, representing the Assignment feedback by a User to an individual part of a Class.

**Instructor**: Authorized User; can complete any functionality of Rubric-Feedback Generator.

**LAMP:** Abbreviation for Linux-Apache-MySQL-PHP. A stack of technologies for running web applications on a Linux server running an Apache server with a MySQL RDBMS. PHP is installed and used for connecting to the database at the application level.

**MAMP:** Abbreviation for Mac-Apache-MySQL-PHP. A stack of technologies for running web applications on a Mac server running an Apache server with a MySQL RDBMS. PHP is installed and used for connecting to the database at the application level.

**RDBMS**: Abbreviation for “Relational Database Management System.” A database management system based on a relational model.

**Rubric**: A Web-based form containing Criteria for creating a Grade of an Assignment for an individual part of a Class.

**Teaching Information System:** The scheduler and department information system currently used by the Department of Computer Science at Montclair State University.

**TIS:** Abbreviation for “Teaching Information System.”

# III. Software Development Process

My advisor, Dr. Emily Hill, and I followed an agile development process due to the time restrictions and the nature of the scope of the project. Since Rubric-Feedback Generator was to contain different modules, i.e. Classes, Rubrics, Assignments, and Grades, it was more efficient to go through the development cycle for each single module incrementally. This allowed the efficient and development of Rubric-Feedback Generator as a whole as we made changes and ran into problems with earlier ideas for the project.

## Requirements & Specifications

### Stakeholder Requirements

In the scope of this project, the stakeholder is the Department of Computer Science at Montclair State University, represented by my project advisor, Dr. Emily Hill. As a faculty member in the Computer Science Department, Dr. Hill is an instructor for multiple courses and uses static rubrics for assignments. Typically in the lower level courses, graduate assistants use Dr. Hill’s rubrics to grade the students. Dr. Hill wanted an application that allowed the Web-based creation of rubrics and grading of assignments, and had an output that was user friendly for graduate assistants, students, and other instructors.

### Functional Requirements

Rubric-Feedback Generator provides a space for a User to create a Rubric. The Rubric is a web form that can be imported from a tab-delimited file or created within Rubric-Feedback Generator. Rubrics are saved so that they can be used in any Assignments or that the User can edit them at any time. Rubrics can only be deleted if there are no Assignments associated with them.

Users can create and edit a Class. Classes can only be deleted if there are no Assignments associated with them.

Within a Class, the User can create and edit an Assignment. Assignments belong to a single Class and require an already existing Rubric that the User specifies is to be used to create Grades for an individual Assignment. Assignments can only be deleted if there are no Grades associated with them.

Given an Assignment, a User can create a Grade for a Student by filling out and submitting the Assignment’s Rubric. The User can view, edit, and delete grades without constraints similar to the other objects. The User can send the given Student of a Grade the content of a Grade via email. If there are multiple Grades for an Assignment, the User may choose to send a batch of grades via email at once.

A User can edit their account information, i.e. their email, display name, and password.

### Non-functional Requirements

Rubric-Feedback Generator has a Web-based interface and can be accessed over a network, intranet or Internet. The User needs only the ability to connect to the network on which it is installed and know how to use a web browser. It is assumed that individuals part of a Class have access to the email account they provide to the User outside the functionality of Rubric-Feedback Generator.

The application runs on a server with the MySQL relational database management system. Rubric-Feedback Generator can be used on devices of varying form factors, such as Desktop, Laptop, tablet, and smartphone devices. The device being used by the User must have a standard web browser installed, such as Firefox, Chrome, Safari, or Internet Explorer, and access to the network the application is installed on.

For security, authorization via log in form is required for access to the User Admin page. A combination of unique username and password is required for authorization. A cookie is saved to the User’s device, therefore the web browser employed by the User must allow cookies.

## Architecture & Design

### Database Architecture

Data is stored in a relational database as it is beyond the complexity of a flat file and the use of the MySQL RDBMS allows future expansion and current compatibility with the Computer Science Department’s current TIS application. The database contains tables for holding data of each module that makes up the functionality of Rubric-Feedback Generator. Figure 1 shows the entity-relationship model for these modules, along with the fields that are found within their tables.

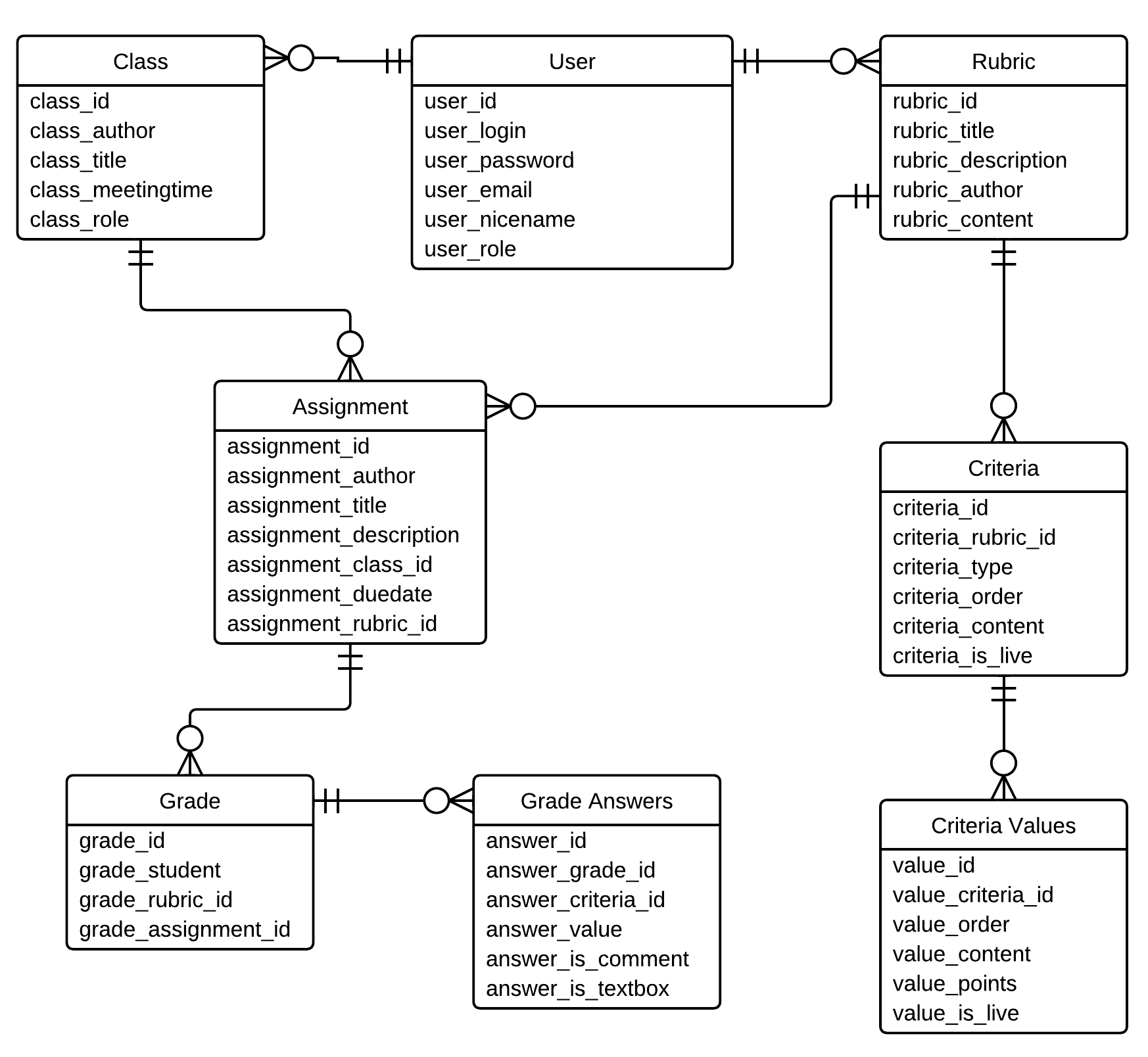


Figure 1: Entity-Relationship model of Rubric-Feedback Generator

The tables within Rubric-Feedback Generator database have a prefix of rubric\_. In the case of the Computer Science Department at Montclair State University, these tables would be created within the same database holding tables for TIS, so the prefix of rubric\_ will set Rubric-Feedback-Generator apart from other systems’ tables.

#### rubric\_assignment

The rubric\_assignment table holds data about individual Assignments. It contains the following fields:

assignment\_id: unique, integer; the identifier of this Assignment.

assignment\_author: string; the user\_login of the User that created this Assignment.

assignment\_title: string; the title of an Assignment.

assignment\_description: string; the description of this Assignment.

assignment\_class\_id: integer; the class\_id of this Assignment.

assignment\_duedate: date; the date that this Assignment is due.

assignment\_rubric\_id: integer; the rubric\_id of a Rubric chosen by a User to be used in grading this Assignment.

#### rubric\_class

The rubric\_class table holds data about individual Classes. It contains the following fields:

class\_id: unique, integer; the identifier of this Class.

class\_author: string; the user\_login of the User that created this Class.

class\_title: string; the title of this Class.

class\_meetingtime: string; the semester and year in that this Class runs.

class\_notes: string; the notes, or description, of this Class.

#### rubric\_criteria

The rubric\_criteria table holds data about individual Criteria of all Rubrics. It contains the following fields:

criteria\_id: unique, integer; the identifier of this Criteria.

criteria\_rubric\_id: integer; the rubric\_id of a Rubric this Criteria belongs to.

criteria\_type: string; the type of this Criteria.

criteria\_order: integer; the order number of this Criteria within a Rubric.

criteria\_content: string; the content of this Criteria.

criteria\_is\_live: integer; the switch for whether this Criteria is live (1) or not (0).

#### rubric\_criteria\_option

The rubric\_criteria\_option table holds data about individual Options of all Criteria. It contains the following fields:

option\_id: unique, integer; the identifier of this Option.

option\_criteria\_id: integer; the criteria\_id of a Criteria this Option belongs to.

option\_order: integer; the order number of this Option within a Criteria.

option\_content: string; the content of this Option.

option\_points: integer; the points attributed to this Option.

option\_is\_live: integer; the switch for whether this Option is live (1) or not (0).

#### rubric\_form

The rubric\_form table holds data about individual Rubrics. It contains the following fields:

rubric\_id: unique, integer; the identifier of this Rubric.

rubric\_title: string; the title of this Rubric.

rubric\_description: string; the description of this Rubric.

rubric\_author: string; the user\_login of the User that created this Rubric.

rubric\_content: string; the content of this Rubric.

#### rubric\_grade

The rubric\_grade table holds data about individual Grades. It contains the following fields:

grade\_id: unique, integer; identifier of this Grade.

grade\_student: string; the email address of a Student this Grade is assigned to.

grade\_rubric\_id: integer; the rubric\_id of a Rubric this Grade was created from.

grade\_assignment\_id: integer; the assignment\_id of an Assignment this Grade belongs to.

#### rubric\_grade\_answer

The rubric\_grade\_answer table holds data about individual Answer values of all Grades. It contains the following fields:

answer\_id: unique, integer; identifier of this Answer.

answer\_grade\_id: integer; the grade\_id of a Grade this Answer belongs to.

answer\_criteria\_id: integer; the criteria\_id of the Criteria this Answer is a response to.

answer\_value: string; the content of the Answer; if answer\_is\_comment value is 0 and answer\_is\_textbox value is 0, then the value is the option\_id of the Option the User chose when creating this Grade.

answer\_is\_comment: integer; the switch for whether this Answer is a comment box value (1) or now (0).

answer\_is\_textbox: integer; the switch for whether this Answer is a textbox value (1) or not (0).

#### rubric\_user

The rubric\_user table holds data about individual Users. It contains the following fields:

user\_id: unique, integer; identification of a User

user\_login: string; the username of a User used for authorization

user\_password: string; the md5 hash of the password of a User used for authorization

user\_email: string; the email address of a User

user\_nicename: string; the display name of a User

user\_role: integer; the role of a User, current default is “Instructor”

## Implementation

### Programming

As the data is stored in a MySQL database, and Rubric-Feedback Generator is Web-based, the programming language used is PHP. HTML and CSS are used for the front-end view’s content and styling. The jQuery library of JavaScript is used for validation of the forms within the application, as well with the dynamic addition and removal of objects within Rubrics.

### Database Installation

The following are SQL commands used to create a database and the tables that go within it for Rubric-Feedback Generator.

CREATE DATABASE `rubric\_creator`;

USE `rubric\_creator`;

CREATE TABLE IF NOT EXISTS `rubric\_assignment` (

`assignment\_id` int(20) NOT NULL AUTO\_INCREMENT,

`assignment\_author` varchar(35) NOT NULL,

`assignment\_title` varchar(80) NOT NULL,

`assignment\_description` varchar(140) NOT NULL,

`assignment\_class\_id` int(20) NOT NULL,

`assignment\_duedate` date NOT NULL,

`assignment\_rubric\_id` int(20) NOT NULL,

PRIMARY KEY (`assignment\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_class` (

`class\_id` int(20) NOT NULL AUTO\_INCREMENT,

`class\_author` varchar(35) NOT NULL,

`class\_title` varchar(80) NOT NULL,

`class\_meetingtime` varchar(80) NOT NULL,

`class\_notes` text NOT NULL,

PRIMARY KEY (`class\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_criteria` (

`criteria\_id` int(20) NOT NULL AUTO\_INCREMENT,

`criteria\_rubric\_id` int(20) NOT NULL,

`criteria\_type` varchar(80) NOT NULL,

`criteria\_order` int(11) NOT NULL,

`criteria\_content` text NOT NULL,

`criteria\_is\_live` int(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`criteria\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_criteria\_option` (

`option\_id` int(20) NOT NULL AUTO\_INCREMENT,

`option\_criteria\_id` int(20) NOT NULL,

`option\_order` int(11) NOT NULL,

`option\_content` text NOT NULL,

`option\_points` int(11) NOT NULL,

`option\_alive` int(1) NOT NULL DEFAULT '1',

PRIMARY KEY (`option\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_form` (

`rubric\_id` int(20) NOT NULL AUTO\_INCREMENT,

`rubric\_title` varchar(70) NOT NULL,

`rubric\_description` varchar(140) NOT NULL,

`rubric\_author` varchar(35) NOT NULL,

`rubric\_content` longtext NOT NULL,

PRIMARY KEY (`rubric\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_grade` (

`grade\_id` int(20) NOT NULL AUTO\_INCREMENT,

`grade\_student` varchar(80) NOT NULL,

`grade\_rubric\_id` int(20) NOT NULL,

`grade\_assignment\_id` int(20) NOT NULL,

PRIMARY KEY (`grade\_id`),

UNIQUE KEY `grade\_student` (`grade\_student`,`grade\_assignment\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_grade\_answer` (

`answer\_id` int(20) NOT NULL AUTO\_INCREMENT,

`answer\_grade\_id` int(20) NOT NULL,

`answer\_criteria\_id` int(20) NOT NULL,

`answer\_value` text NOT NULL,

`is\_comment` int(1) NOT NULL,

`is\_textbox` int(1) NOT NULL,

PRIMARY KEY (`answer\_id`)

) ENGINE=InnoDB;

CREATE TABLE IF NOT EXISTS `rubric\_user` (

`user\_id` int(20) NOT NULL AUTO\_INCREMENT,

`user\_login` varchar(35) NOT NULL,

`user\_password` varchar(32) NOT NULL,

`user\_email` varchar(100) NOT NULL,

`user\_nicename` varchar(80) NOT NULL,

`user\_role` varchar(20) NOT NULL,

PRIMARY KEY (`user\_id`),

UNIQUE KEY `user\_login` (`user\_login`)

) ENGINE=InnoDB;

If an already existing database is to be used for holding Rubric-Feedback Generator tables, then the CREATE DATABASE command is not necessary, and the USE command should call the already existing database instead.

## Deployment

After the database setup of Rubric-Feedback Generator, all of the application files are to be included in the same directory on any compatible server.

To connect Rubric-Feedback Generator to the database, which is necessary for it to run, open up the file functions-connect.php in the functions directory and update the code with the host name, database name, database username, and database password.

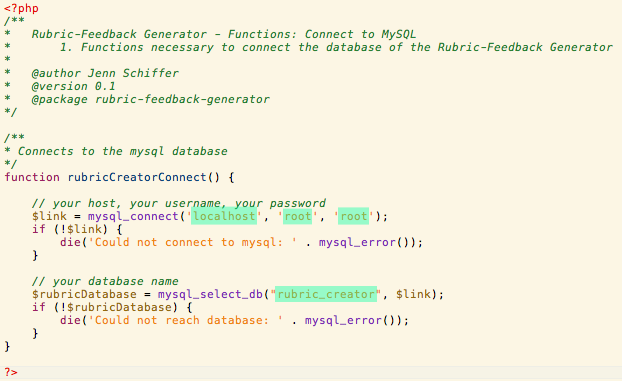


Figure 2: Connecting to the database – Open functions/functions-connect.php in a text editor and replace the database's host, username, password, and database name with the correct content

Rubric-Feedback Generator ran locally during the development process on a MAMP server setup on my 13” MacBook Pro. The application is currently deployed onto a server on a LAMP stack for testing before integration with TIS.

Future deployment plans include the integration of Rubric-Feedback Generator with the Computer Science Department’s TIS application.

# IV. Using Rubric-Feedback Generator

This section is the manual for using Rubric-Feedback Generator. It is a collection of all the user views and instructions of use in each view.

## Login

The index page for Rubric-Feedback Generator is the Login page. The correct username and password combination of a User is required for authorization into the rest of Rubric-Feedback Generator’s pages.

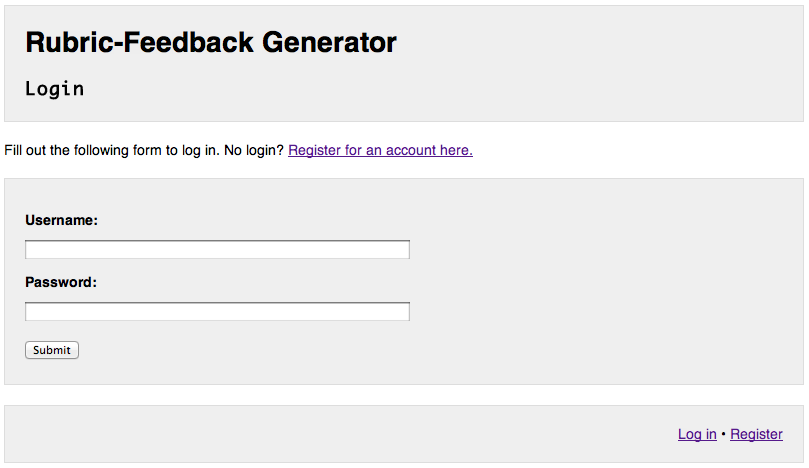


Figure 3: Login page

Form validation requires that the “Username” and “Password” fields are not blank at submission. If one or both are, the User is warned.

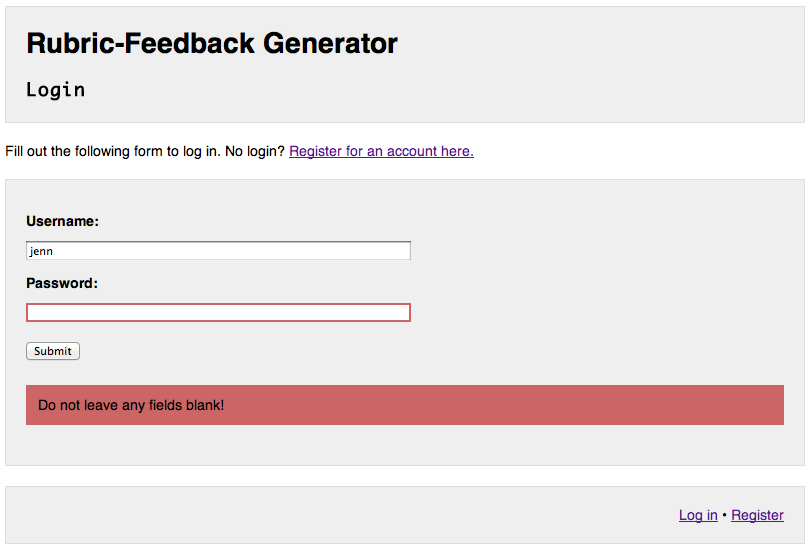


Figure 4: Login page – Warning User of blank fields after submitting form

If a User submits a correct username and password combination, they are sent to the User Admin page. Otherwise, the User is told to try again.



Figure 5: Login page – Warning User that username and password combination is incorrect

## Register

If a User does not have a Rubric-Feedback Generator account, they can register one by clicking the link on the Login page.

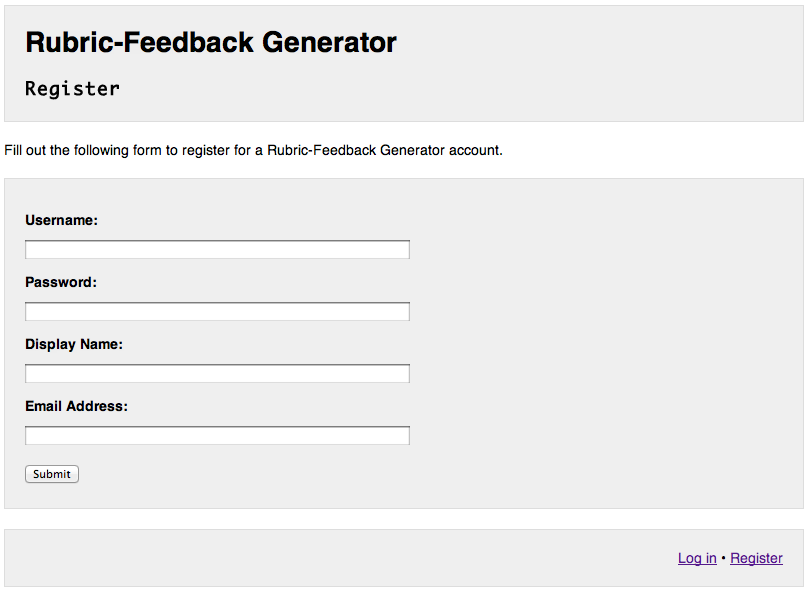


Figure 6: Register page

A User’s username and password are used on the Login page for authorization into the rest of Rubric-Feedback Generator. A User’s display name is used in Emails. A User’s email address is used as the from-address in Emails.

Form validation requires that all fields be filled. If one or both are, the User is warned.

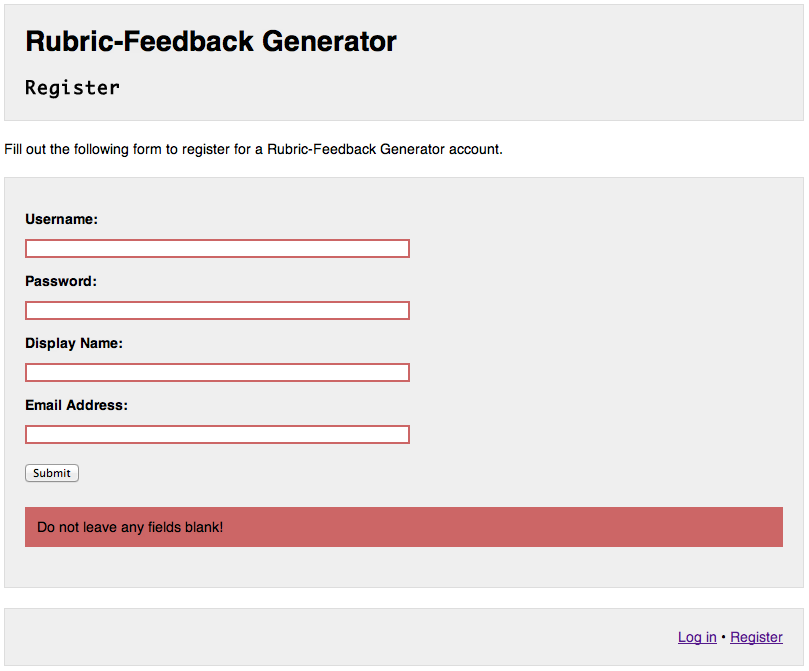


Figure 7: Register page - Warning User of blank fields after submitting form

Form validation requires a valid email address, and will warn the user if the “Email Address” field value is invalid after clicking “Submit.” A valid email address to the Rubric-Feedback Generator is any string of non-whitespace characters followed by “@”, another string of non-whitespace characters, a “.”, and a final string of non-whitespace characters. For example, “schifferj@mail.montclair.edu” is a valid email address. “Jenn” is not a valid email address.

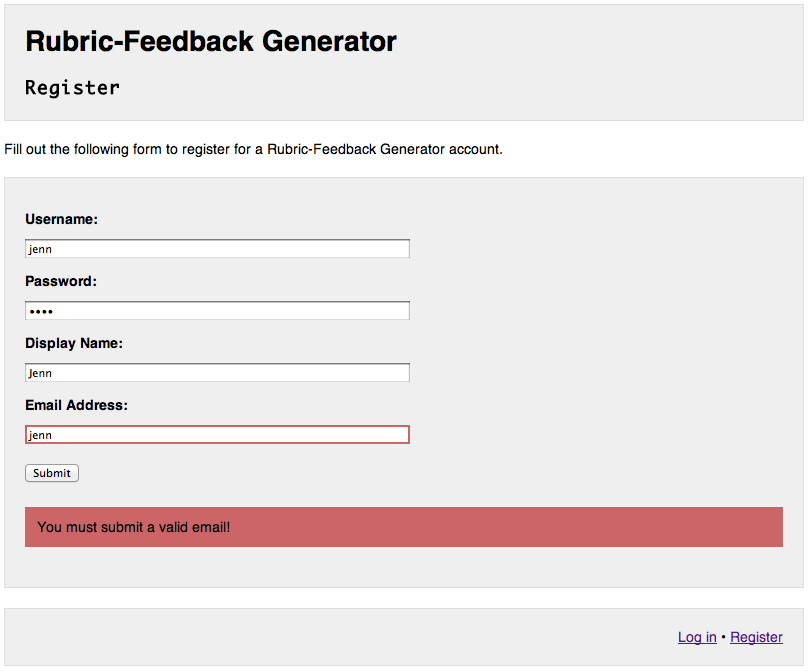


Figure 8: Register page – Warning user of invalid email after submitting form

All account usernames are unique. If a User submits the form and the “Username” field value they submitted is already used by another User, they will be warned that the username already exists and be told to try again.

## Macintosh HD:Users:jenn:Dropbox:Masters:masters screencaps:register-usernameexists.png

Figure 9: Register page - Warning User that the username submitted already exists

Upon successful creation of a new Rubric-Feedback Generator account, the User will be given a link to the Login page to log in using their new account.

## 

Figure 10: Register page - User has successfully created an account

## 

## User Admin

Once a User has successfully logged in at the Login page, they are sent to the User Admin page. The User Admin page contains menus for a User’s Classes, Rubrics, and User Info management.

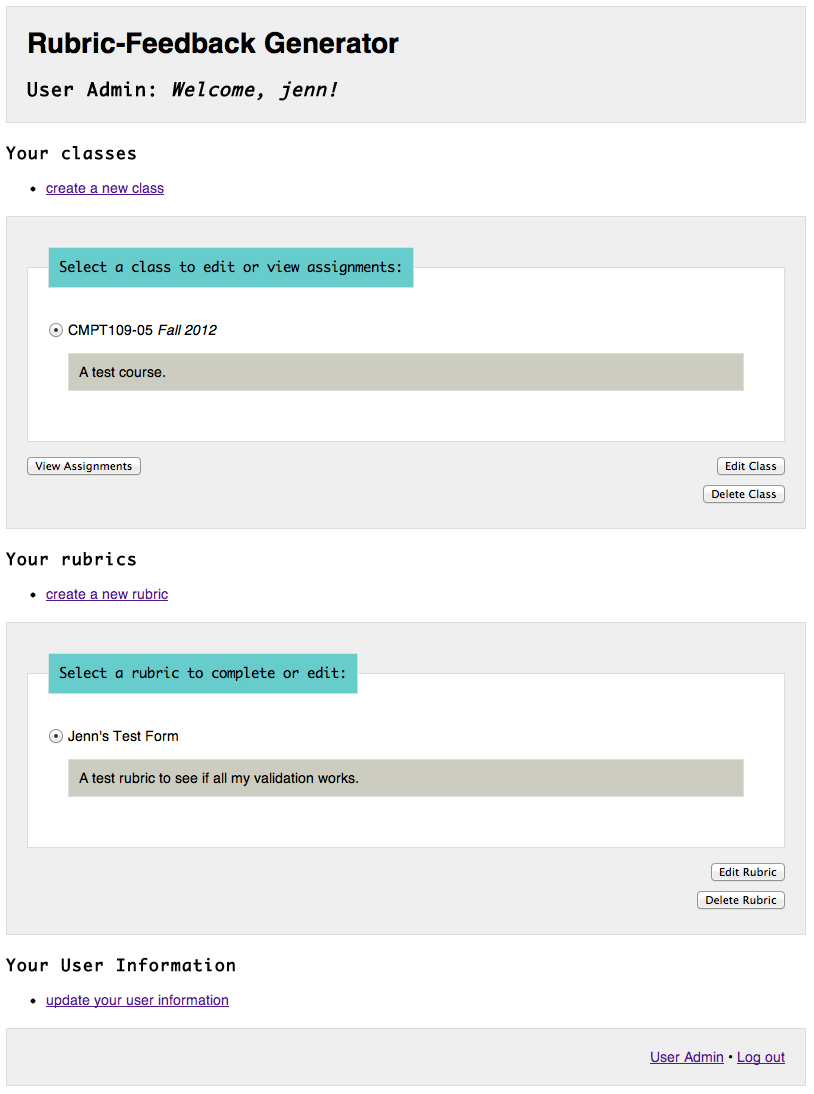


Figure 11: User Admin page - User with username "jenn" successfully logged in

## Edit User Information

When a User is logged in, they can edit their own Rubric-Feedback Generator user information by clicking “Update Your User Info” on the User Admin page.

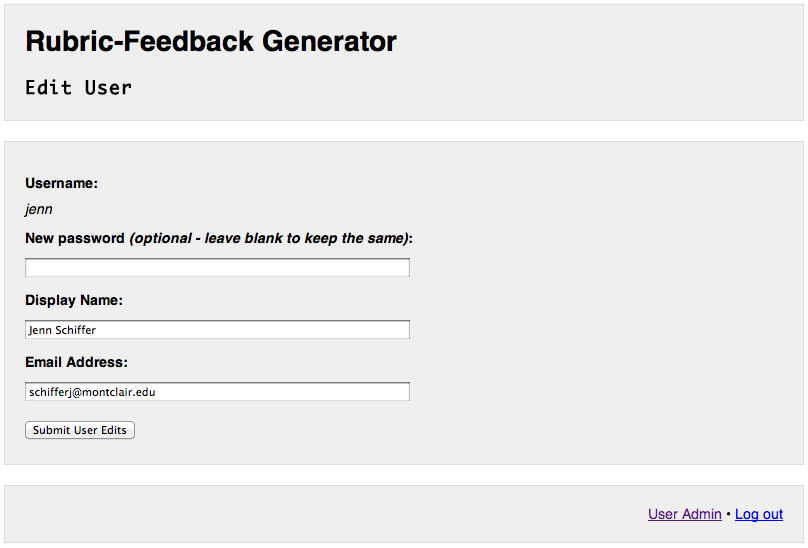


Figure 12: Edit User Information page

The username cannot be changed. The password is change be changed by entering the new password into the “New password” field. If the “New password” field is left blank, the password will not be changed at all after clicking “Submit.”

“Display Name” and “Email Address” fields must be filled. User will be warned if one or both of these fields are blank after clicking “Submit.”

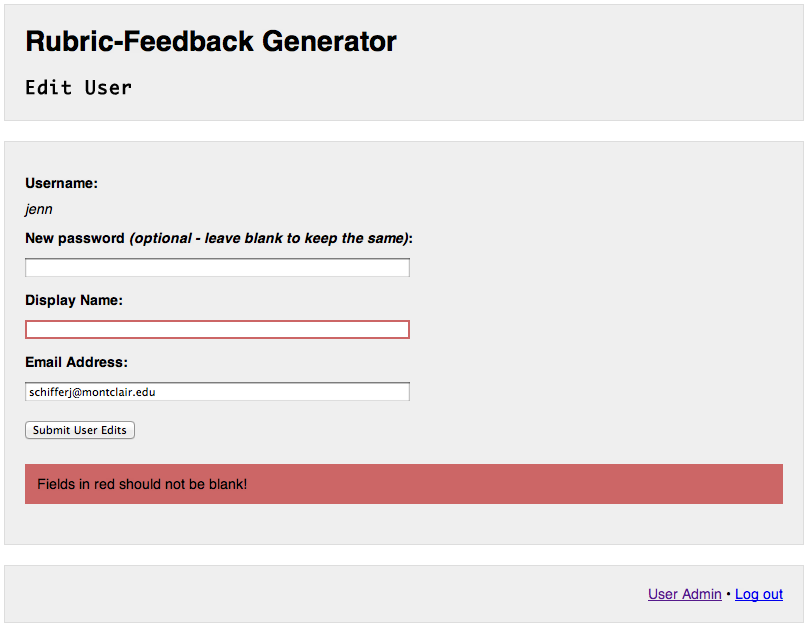


Figure 13: Edit User Information page - Warning User of blank fields after submitting form

Form validation requires a valid email address, and will warn the user if the “Email Address” field value is invalid after clicking “Submit.” A valid email address to the Rubric-Feedback Generator is any string of non-whitespace characters followed by “@”, another string of non-whitespace characters, a “.”, and a final string of non-whitespace characters. For example, “schifferj@mail.montclair.edu” is a valid email address. “Jenn” is not a valid email address.

## Macintosh HD:Users:jenn:Dropbox:Masters:masters screencaps:account-invalidemail.png

Figure 14: Edit User Information page - Warning User of invalid email after submitting form

Upon successful submission of the User Information, the User will be given a link back to the User Admin page.

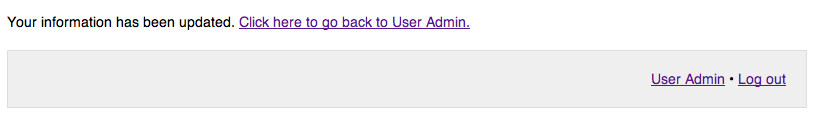


Figure 15: Edit User Information page - User successfully edited their User Information

## Create New Class

A User can create a new Class by clicking “Create a New Class” on the User Admin page.

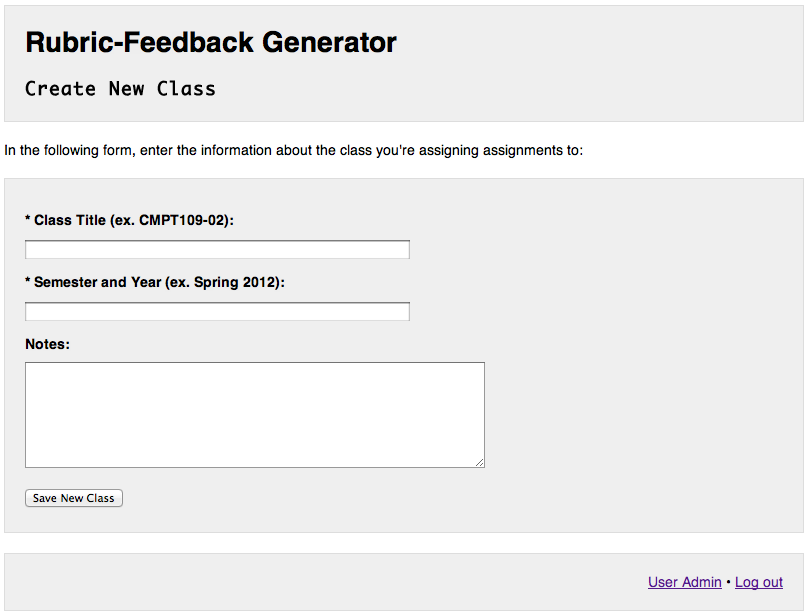


Figure 16: Create New Class page

The Class title is a string specifying a course title or code for the User’s identification. The Class title will also be displayed in Grades sent via Email. The semester and year is a string specifying when the Class is running. Notes of a Class are optional and can be used for a User’s notes or description of the Class.

“Class Title” and “Semester and Year” fields must be filled. User will be warned if one or both of these fields are blank after clicking “Save New Class.”

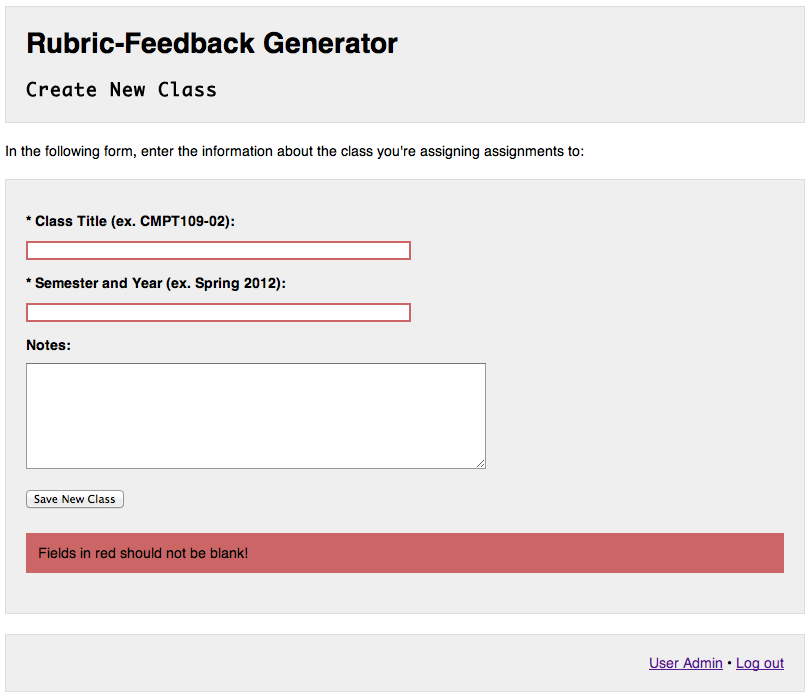


Figure 17: Create New Class page - Warning User of blank fields after submitting form

Upon successful submission of the new Class, the User will be given a link back to the User Admin page.

## Macintosh HD:Users:jenn:Dropbox:Masters:masters screencaps:class-new-success.png

Figure 18: Create New Class page - User successfully created a new Class

## Edit Class

A User can edit already existing Classes by selecting the Class from the “Your Classes” menu on the User Admin page and clicking the “Edit Class” button.

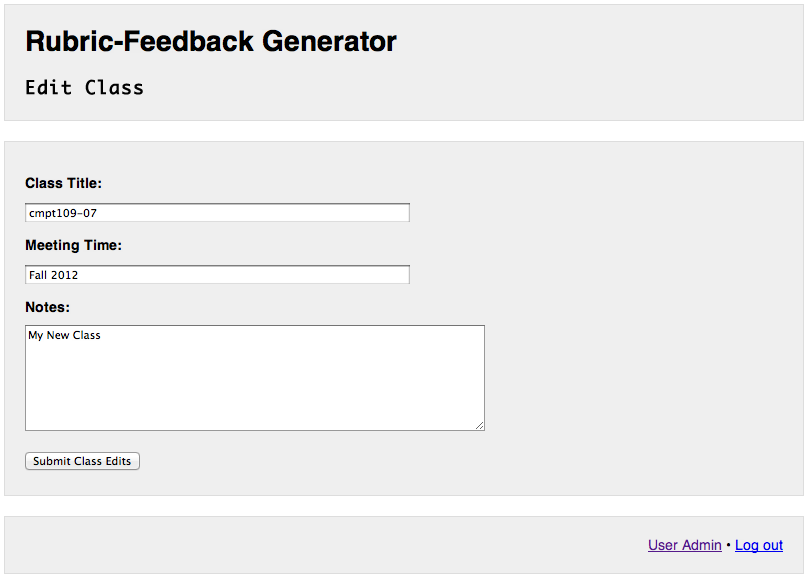


Figure 19: Edit Class page

Similar to the New Class page, “Class Title” and “Semester and Year” fields must be filled. User will be warned if one or both of these fields are blank after clicking “Submit Class Edits.”

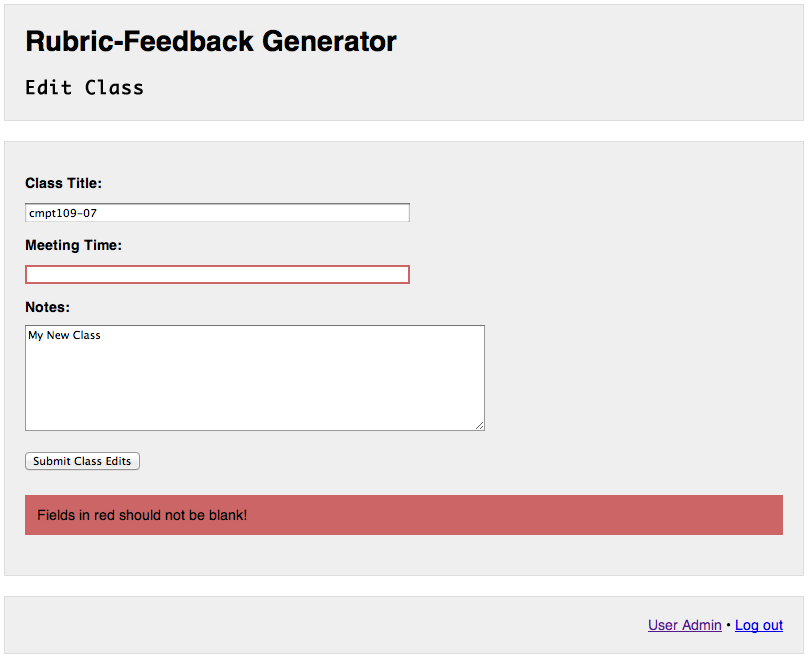


Figure 20: Edit Class page - Warning User of blank fields after submitting form

Upon successful submission of the Class edits, the User will be given a link back to the User Admin page.

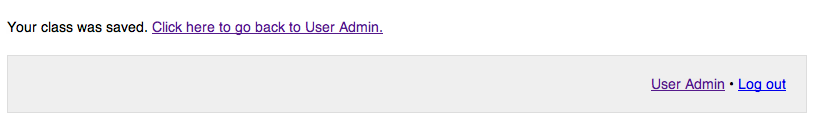


Figure 21: Edit Class page - User successfully submitted edited Class

## Delete Class

A User can delete an already existing Class by selecting the Class from the “Your Classes” menu on the User Admin page and clicking the “Delete Class” button.

A warning prompt will appear below the “Your Classes” menu. Clicking “Cancel” will cancel the delete request and hide the warning prompt. Clicking “Yes, Delete Class” will submit the request to delete the selected Class.

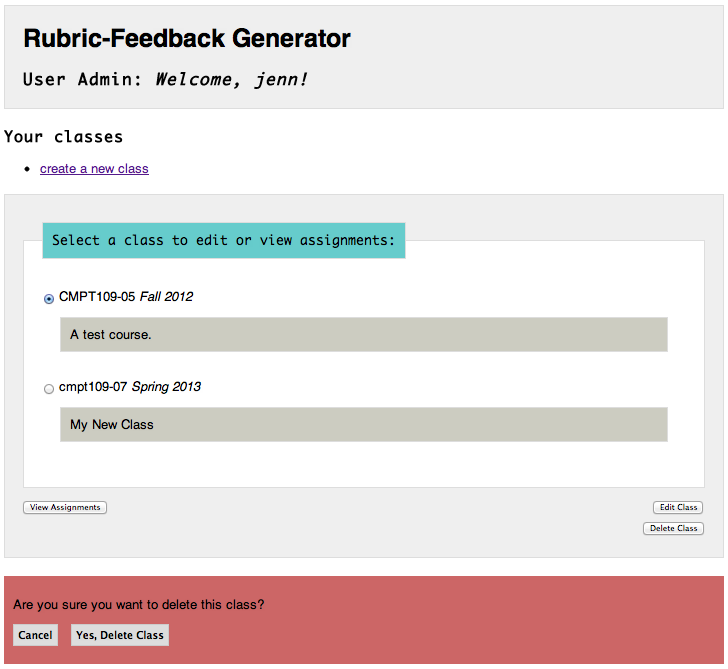


Figure 22: Deleting a Class

A Class can only be deleted if there are no Assignments already existing for this Class. If Assignments for the Class being deleted exist, the User is warned that the Class cannot be deleted. The User needs to delete all Assignments of a Class before deleting that Class successfully.

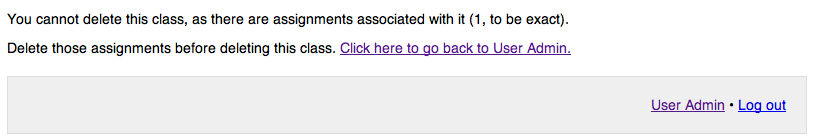


Figure 23 Deleting a Class - Warning User that the Class has Assignments and therefore cannot be deleted

Upon successful deletion of the Class, the User will be given a link back to the User Admin page.

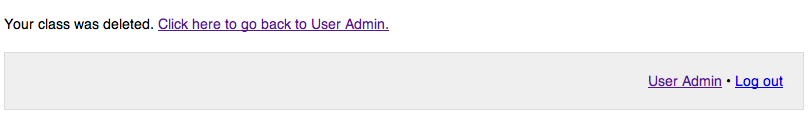


Figure 24 Deleting a Class - User successfully deleted a Class

## View Assignments

A User can view Assignments of an already existing Class by selecting the Class from the “Your Classes” menu on the User Admin page and clicking the “View Assignments” button.

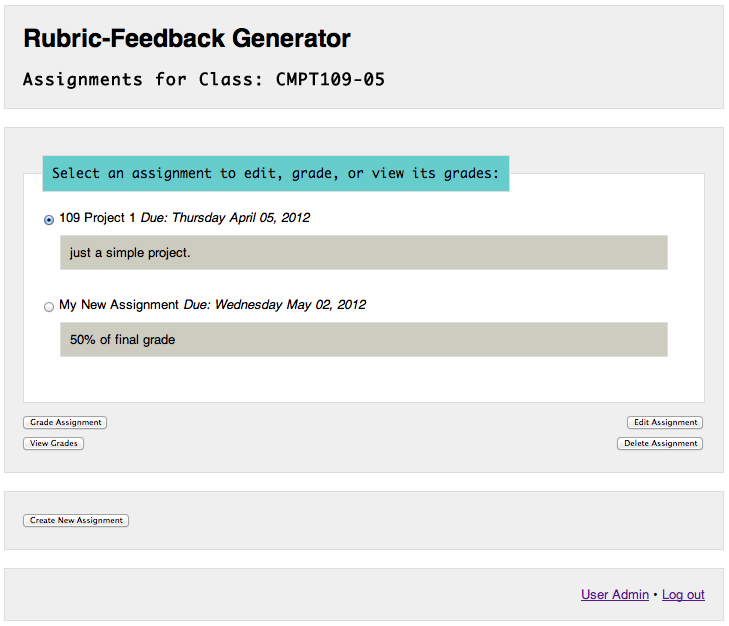


Figure 25: View Assignments page

If the Class has Assignments, they will be listed in this View Assignments page. If no Assignments exist, then only the “Create New Assignment” button appears.

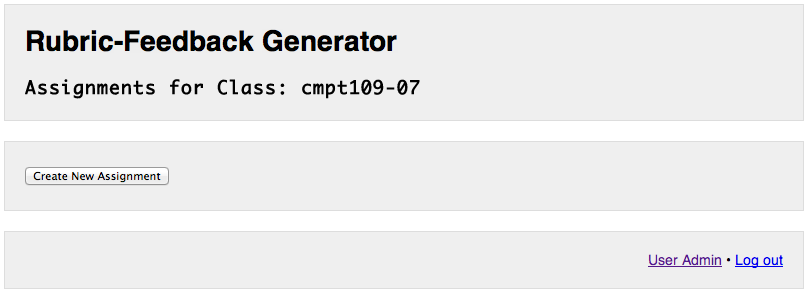


Figure 26 View Assignments page - No assignments for this Class exist

## Create New Assignment

To create a new Assignment, a User can click the “Create New Assignment” button on the View Assignments page.

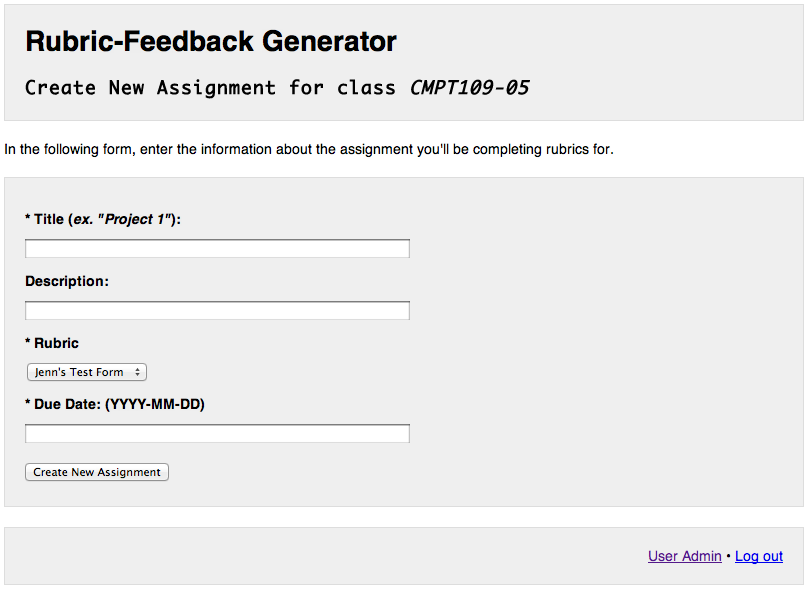


Figure 27: Create New Assignment page

The Assignment title is a string specifying a assignment title for the User’s identification. The Assignment title will also be displayed in Grades sent via Email. The description is a string for notes or description of the assignment. The Rubric selected is the Rubric that will be used to create Grades for this Assignment. The due date is a date specifying when the Assignment is due.

“Assignment Title,” “Rubric,” and “Due Date” fields must be filled. User will be warned if one or both of these fields are blank after clicking “Create New Assignment.”

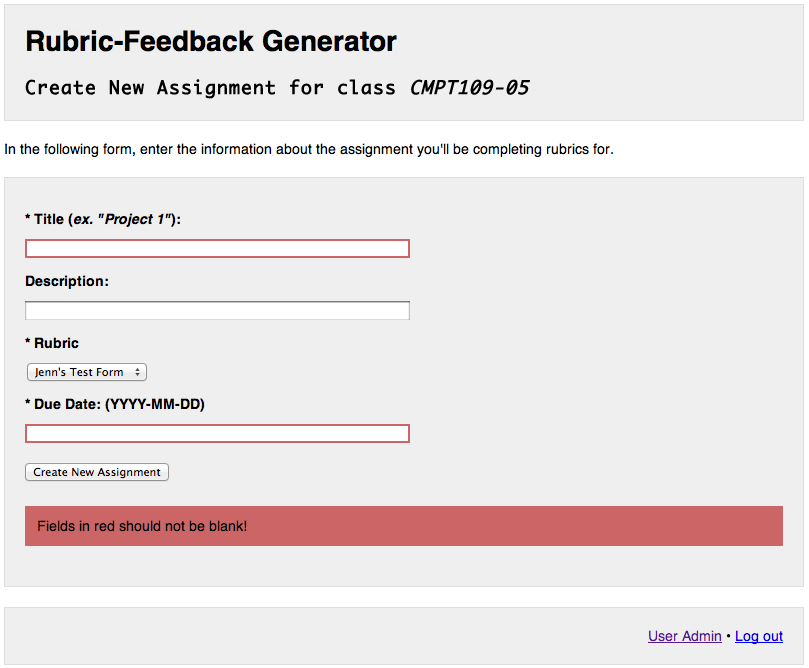


Figure 28: Create New Assignment page - Warning User of blank fields after submitting form

If the User has not created any Rubrics yet, then an Assignment cannot be created, as this would leave the required “Rubric” field blank.

If a User submits the form without any existing Rubrics, the User will be warned that it is not possible to create the Assignment and be given a link to the Create New Rubric page.

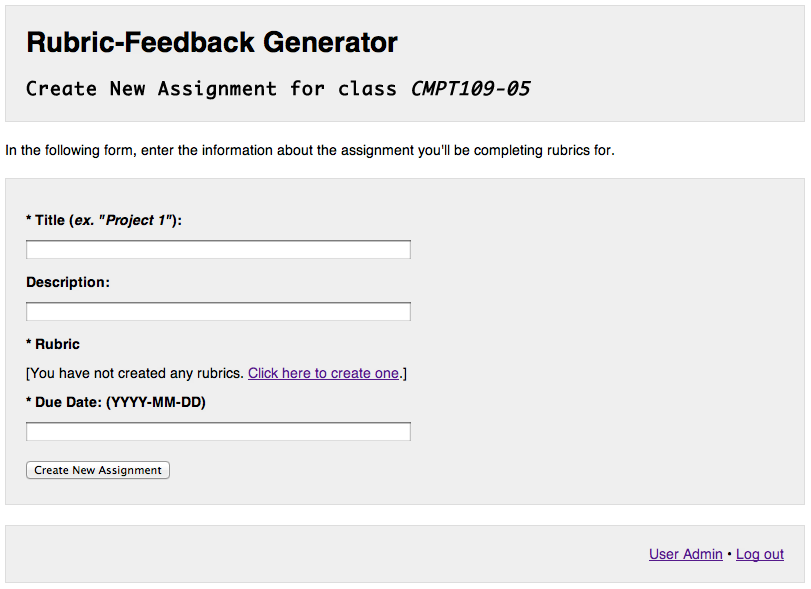


Figure 29: Create New Assignment page - "Rubric" selector is missing because there are no Rubrics to choose

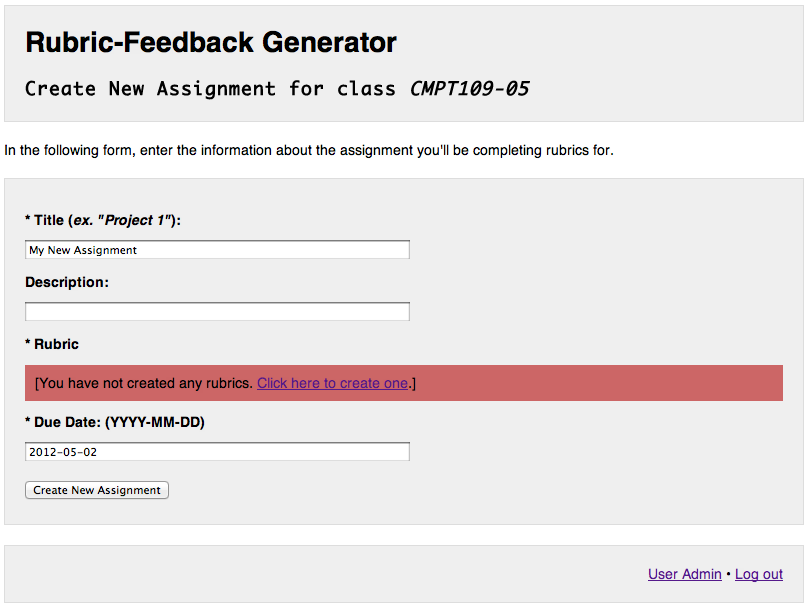


Figure 30: Create New Assignment page - Warning User that there are no Rubrics to choose after submitting form

The due date should be typed in YYYY-MM-DD format, as hinted on the page, where years starting with 20 and 19 are valid. For example, 2012-05-15 is a valid date. 2-2-2 is an invalid date. If a User submits an invalid date, they will be warned after clicking “Create New Assignment.”

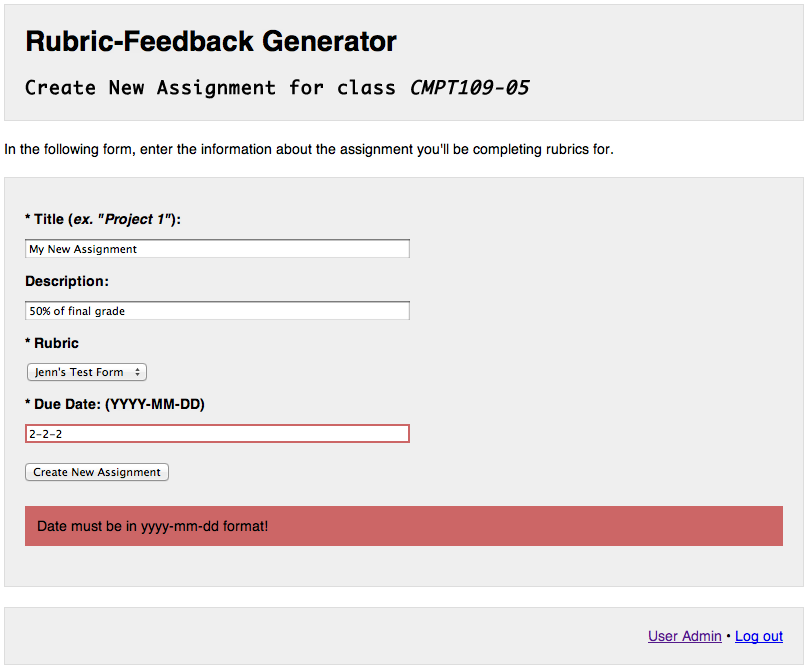


Figure 31: Create New Assignment page - Warning User of invalid date after submitting form

Upon successful creation of the Assignment, the User will be given a link back to the User Admin page.

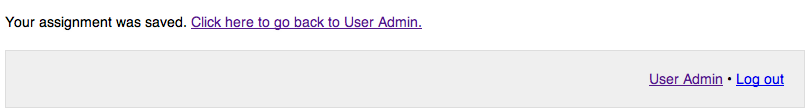


Figure 32: Create New Assignment page - User successfully created a new Assignment

## Edit Assignment

To edit an already existing Assignment, a User can select the Assignment from the View Assignments page and click the “Edit Assignment” button.

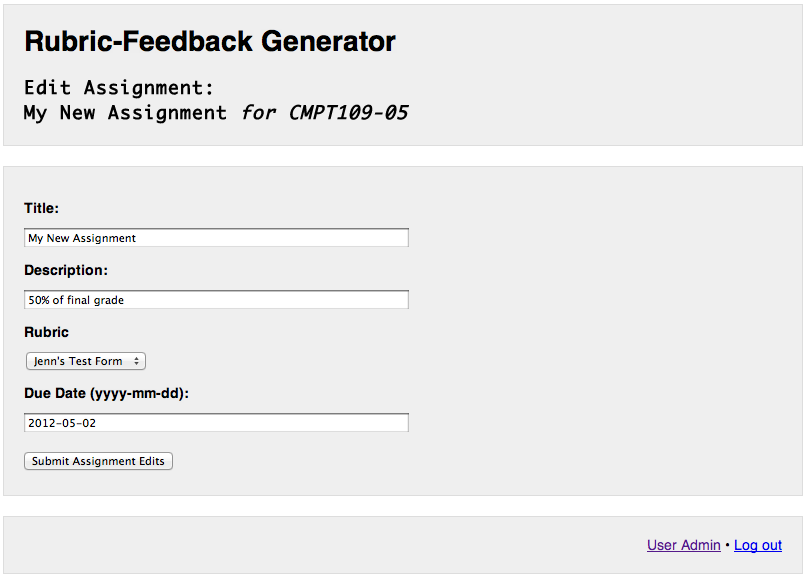


Figure 33: Edit Assignment page

Similar to the Create New Assignment page, “Assignment Title” and “Due Date” fields must be filled. User will be warned if one or both of these fields are blank after clicking “Create New Assignment.”

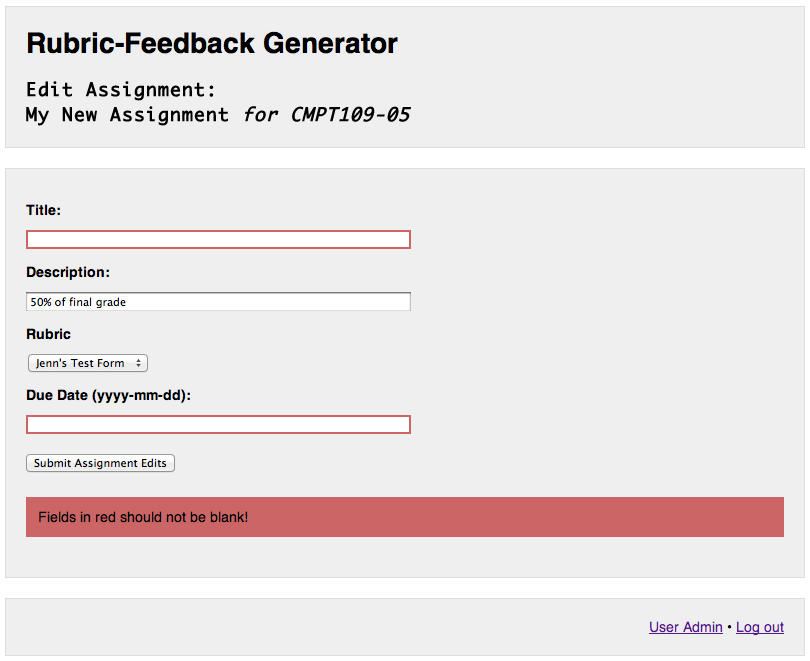


Figure 34: Edit Assignment page - Warning User of blank fields after submitting form

If an Assignment already has Grades created for it, the Rubric associated with the Assignment cannot be changed.

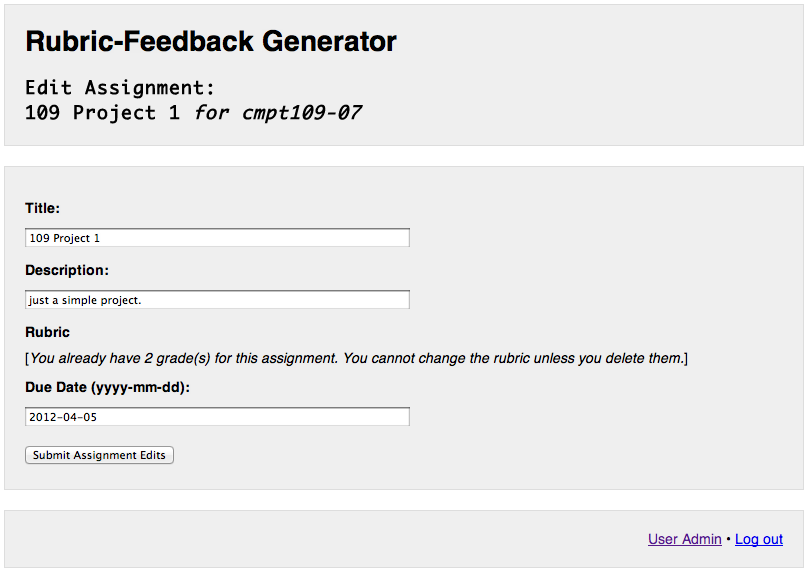


Figure 35: Edit Assignment page - Rubrics cannot be changed if this Assignment already has existing Grades

Upon successful submission of the Assignment edits, the User will be given a link back to the User Admin page.

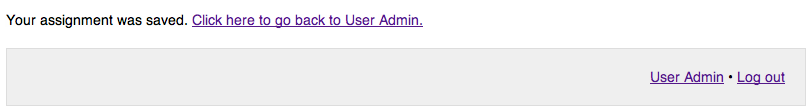


Figure 36: Edit Assignment page – User successfully edited the Assignment

## Delete Assignment

A User can delete an already existing Assignment by selecting the Assignment from the list on the View Assignments page and clicking the “Delete Assignment” button.

A warning prompt will appear below list of the Assignments. Clicking “Cancel” will cancel the delete request and hide the warning prompt. Clicking “Yes, Delete Assignment” will submit the request to delete the selected Assignment.

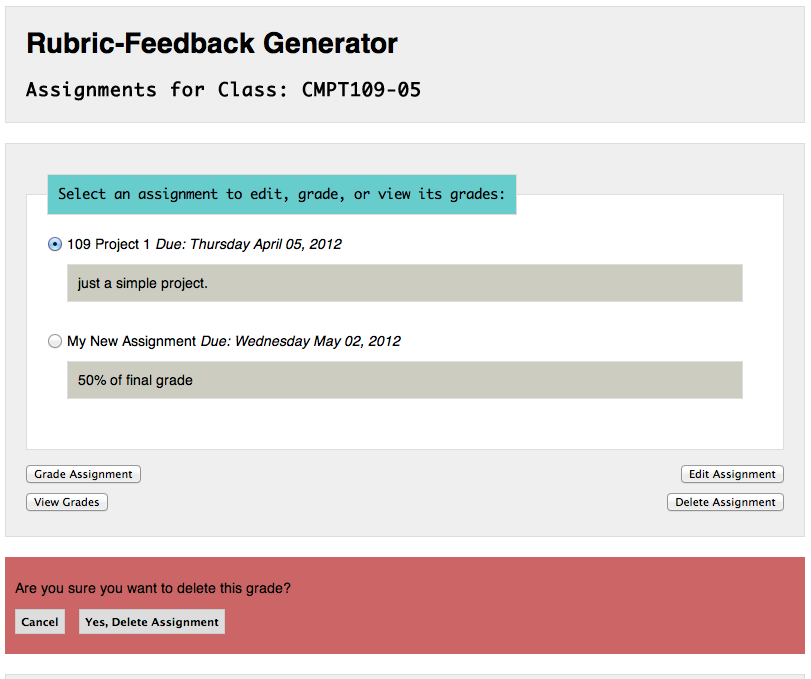


Figure 37: Deleting an Assignment

If Grades for the Assignment being deleted exist, the User is warned that the Assignment cannot be deleted. The User needs to delete all Grades of an Assignment before deleting that Assignment successfully.

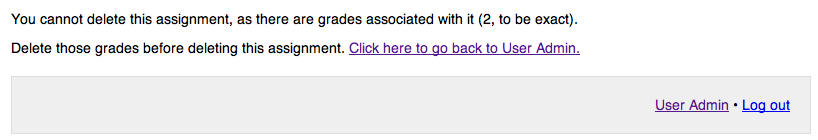


Figure 38 Deleting an Assignment - Warning User that the Assignment has Grades and therefore cannot be deleted

Upon successful deletion of the Assignment, the User will be given a link back to the User Admin page.

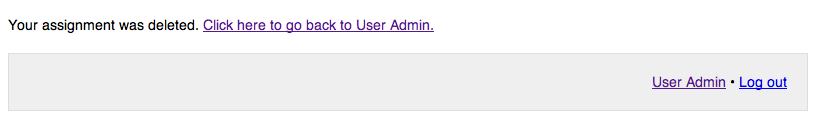


Figure 39 Deleting an Assignment - User successfully deleted an Assignment

## Grade Assignment/Create New Grade

A User can create a Grade for an already existing Assignment by selecting the Assignment from the list on the View Assignments page and clicking the “Grade This Assignment” button.

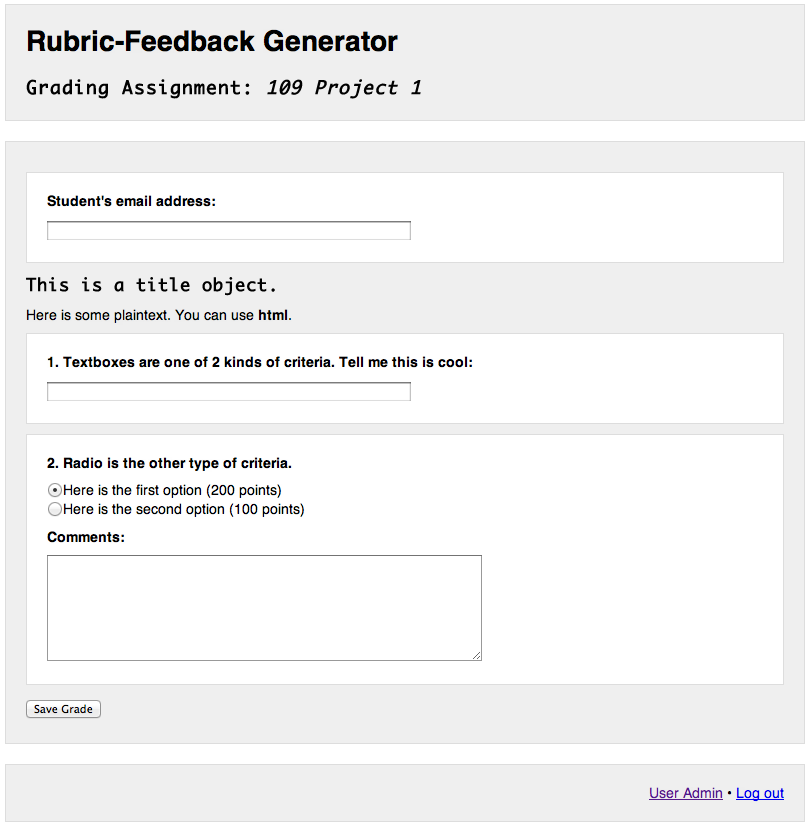


Figure 40 Grading Assignment page

The Grading Assignment page is a web form created from the Rubric selected when creating the Assignment being graded.

The only added field that does not belong to the Rubric of this Assignment is the “Student’s Email Address” field. This value is the “To-address” of any Email sent of a Grade. The rest of the fields belong to the Rubric selected when the Assignment was created.

All fields in the form must be filled or selected. User will be warned if one or both of these fields are blank after clicking “Save Grade.”

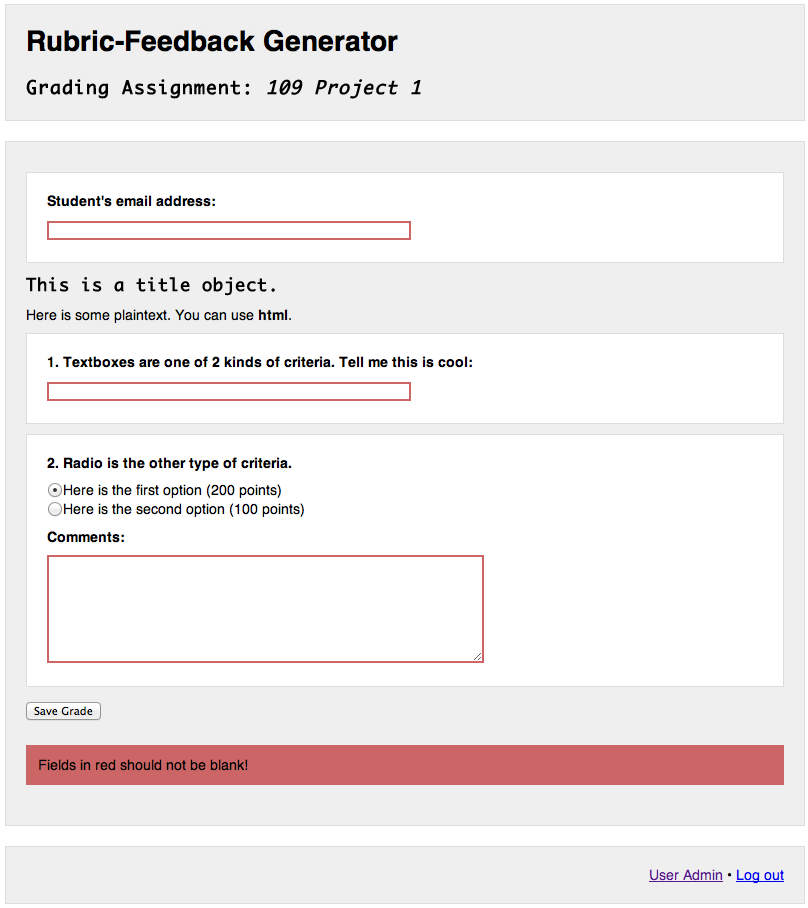


Figure 41 Grading Assignment page - Warning user of blank fields after submitting form

Form validation requires a valid email address, and will warn the user if the “Student’s Email Address” field value is invalid after clicking “Submit.” A valid email address to the Rubric-Feedback Generator is any string of non-whitespace characters followed by “@”, another string of non-whitespace characters, a “.”, and a final string of non-whitespace characters. For example, “schifferj@mail.montclair.edu” is a valid email address. “Jenn” is not a valid email address.

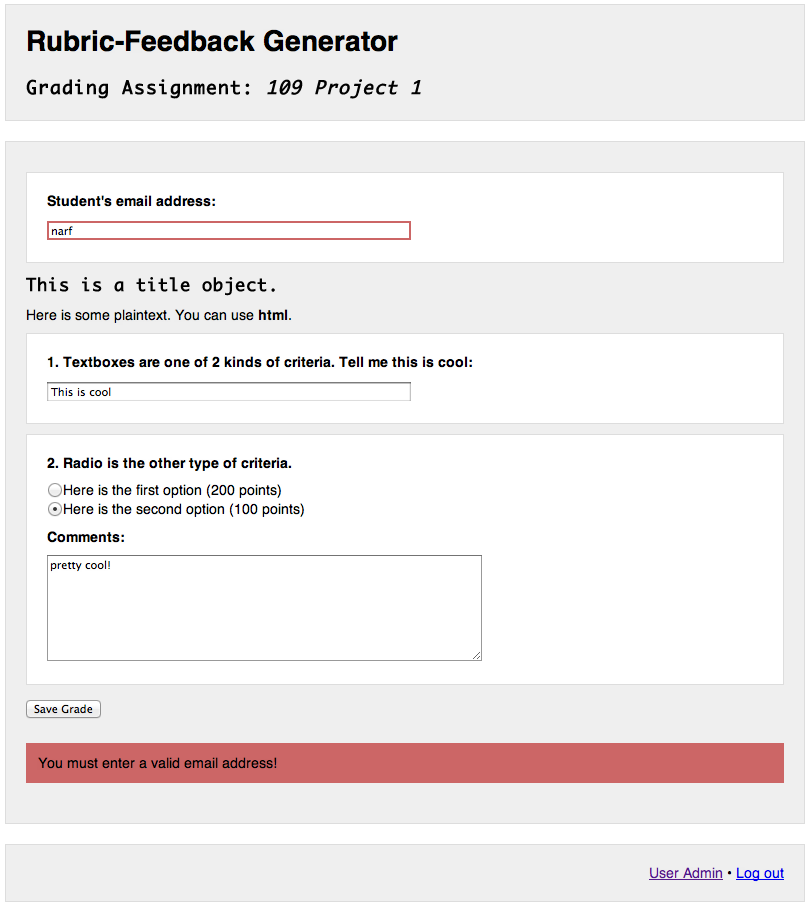


Figure 42 Grading Assignment page - Warning User of invalid email after submitting form

Upon successful creation of the Assignment’s Grade, the User will be given a link back to the User Admin page.

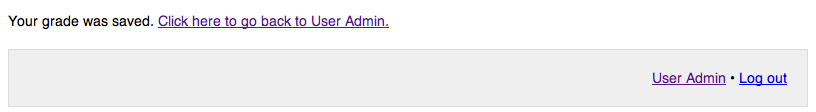


Figure 43 Grading Assignment page - User successfully created Grade for the Assignment

## View Grades

A User can view a list of Grades for an already existing Assignment by selecting the Assignment from the list on the View Assignments page and clicking the “View Grades for This Assignment” button.

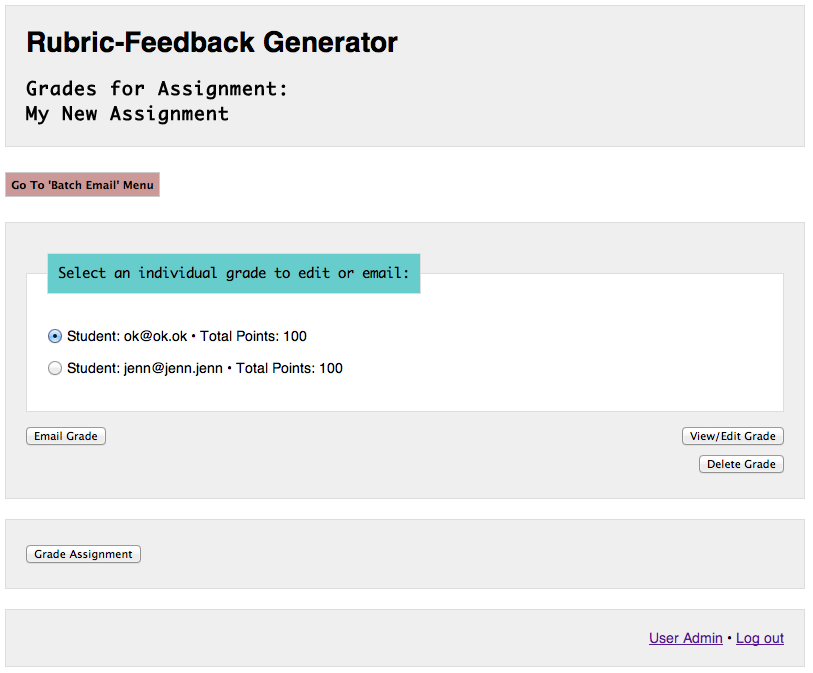


Figure 44: View Grades page

If the Assignment has Grades, they will be listed in this View Grades page, otherwise the User is told no Grades exist.

## 

Figure 45 View Grades page - No Grades for this Assignment exist

## Edit Grade

A User can edit an already existing Grade by selecting the Grade from the list on the View GraDes page and clicking the “Edit Grade” button.

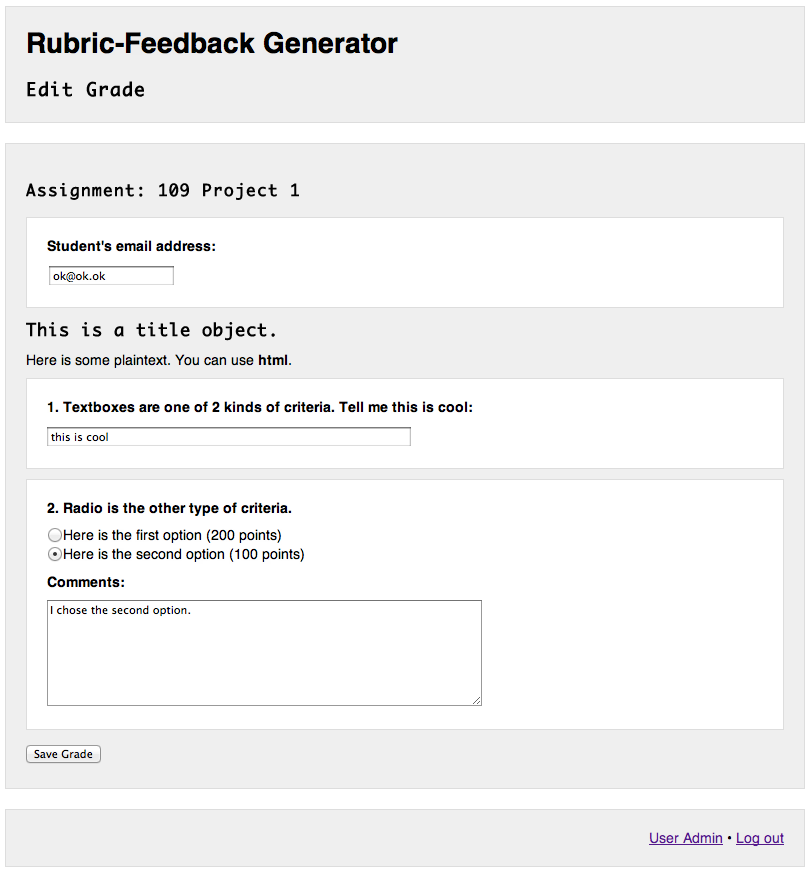


Figure 46: Edit Grade page

The Edit Grade page is a web form created from the Rubric selected when creating the Assignment being graded. All the fields are repopulated with values that were set when the Grade was last created or saved.

All fields in the form must be filled or selected. User will be warned if one or both of these fields are blank after clicking “Save Grade.”

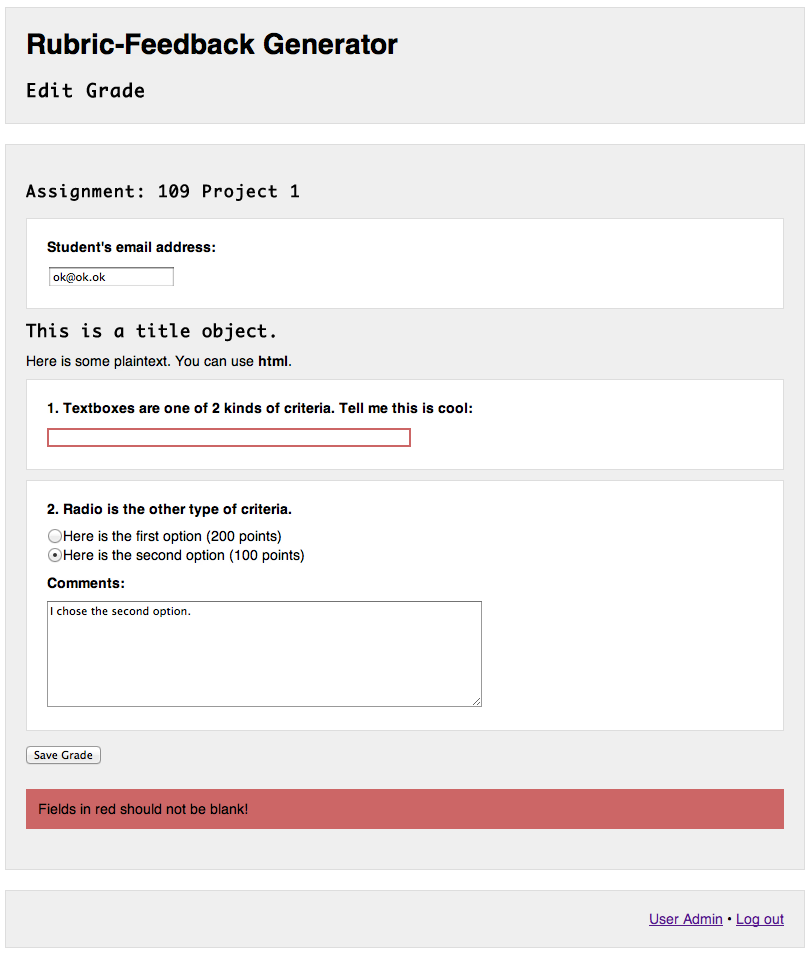


Figure 47: Edit Grade page - Warning User of blank fields after submitting form

Form validation requires a valid email address, and will warn the user if the “Student’s Email Address” field value is invalid after clicking “Submit.” A valid email address to the Rubric-Feedback Generator is any string of non-whitespace characters followed by “@”, another string of non-whitespace characters, a “.”, and a final string of non-whitespace characters. For example, “schifferj@mail.montclair.edu” is a valid email address. “Jenn” is not a valid email address.

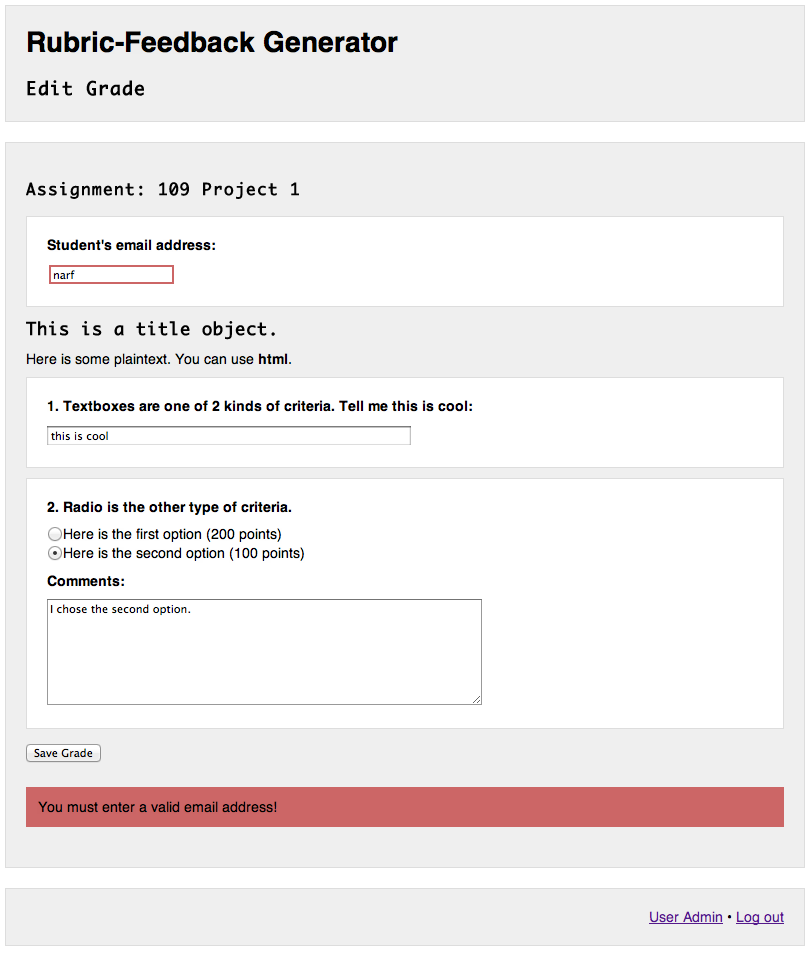


Figure 48: Edit Grade page - Warning User of invalid email after submitting form

Upon successful edit of the Grade, the User will be given a link back to the User Admin page.

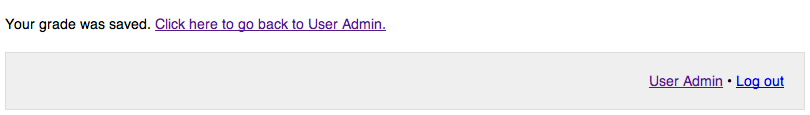


Figure 49: Edit Grade page - User successfully edited the Grade

## Delete Grade

A User can delete an already existing Grade by selecting the Grade from the list on the View Grades page and clicking the “Delete Grade” button.

A warning prompt will appear below list of the Grades. Clicking “Cancel” will cancel the delete request and hide the warning prompt. Clicking “Yes, Delete Grade” will submit the request to delete the selected Grade.

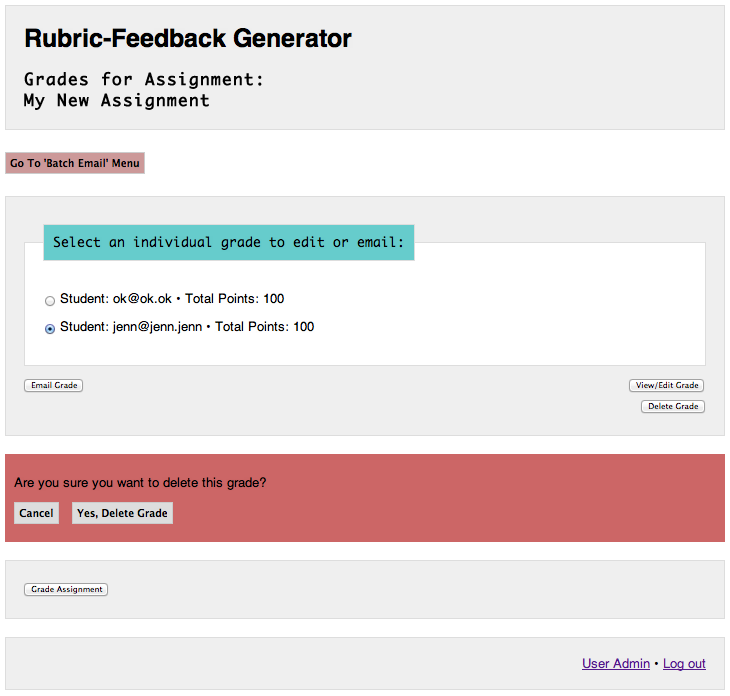


Figure 50: Deleting a Grade

Upon successful deletion of the Grade, the User will be given a link back to the User Admin page.

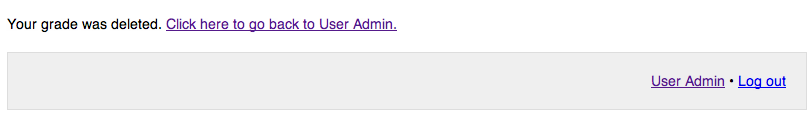


Figure 51: Delete a Grade - User successfully deleted Grade

## Email Grade

A User can send an Email of an already existing Grade by selecting the Grade from the list on the View Grades page and clicking the “Email Grade” button.

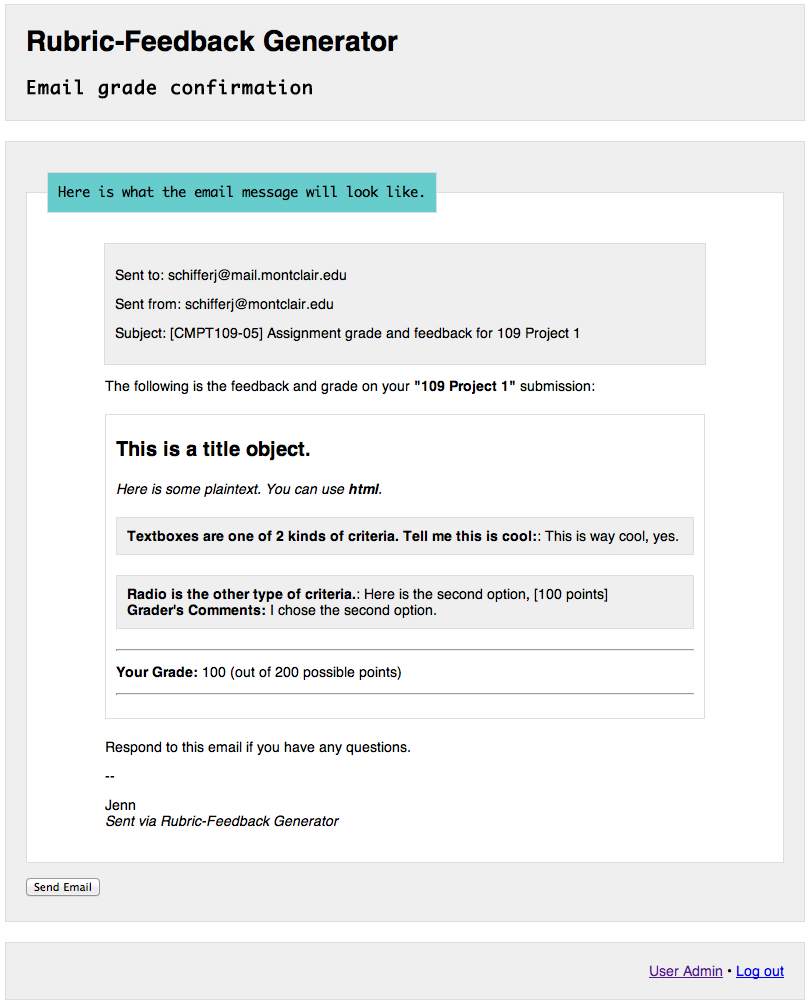


Figure 52: Email Grade page

The User can confirm the Email being sent before it goes out. The Email will be sent to the email address value of the Grade’s “Student’s Email Address” field, which is given when the Grade was created or last edited. The “From-address” is the email address of the User, which is given when the User account was created or last edited. The content of the email is generated from the Rubric of the Assignment and values of the Grade being sent.

Upon successful Email sending of the Grade, the User will be given a link back to the User Admin page.

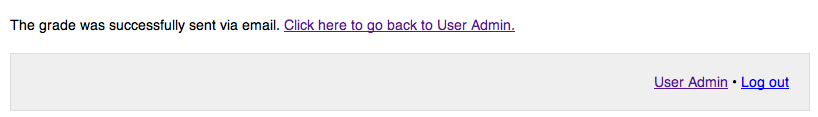


Figure 53: Email Grade page - User successfully sent Email of a Grade

## Batch Email Grades

A User can send multiple (a batch of) Emails of an already existing Grade by selecting the Grade from the list on the Batch Email menu of the View Grades page. Clicking the “Go to ‘Batch Email’ Menu button above the list of Grades accesses the Batch Email menu.

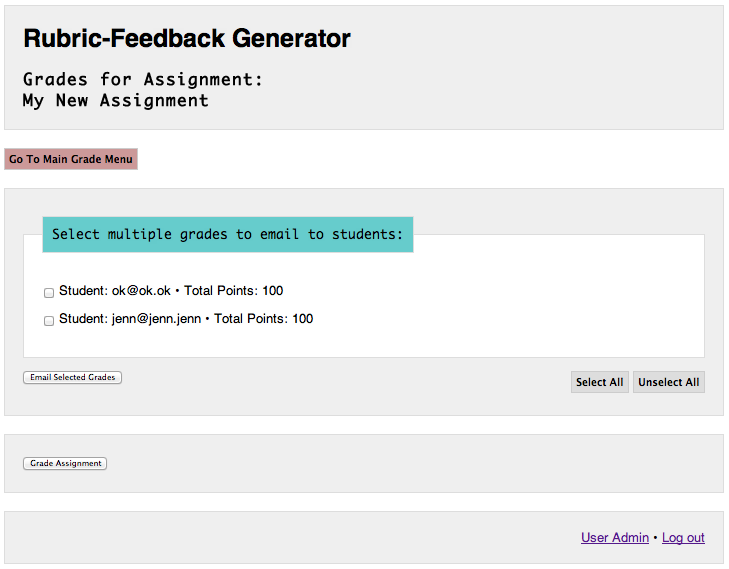


Figure 54: Batch Email menu

The Batch Email menu lists the same Grades as in the “Main Grade Menu,” except there are checkboxes that allow the User to select one or more Grades to send. User can click “Select All” button to check all the boxes, or “Unselect All” to unselect all boxes.

The User will be warned if they submit the form without checking any of the checkboxes.

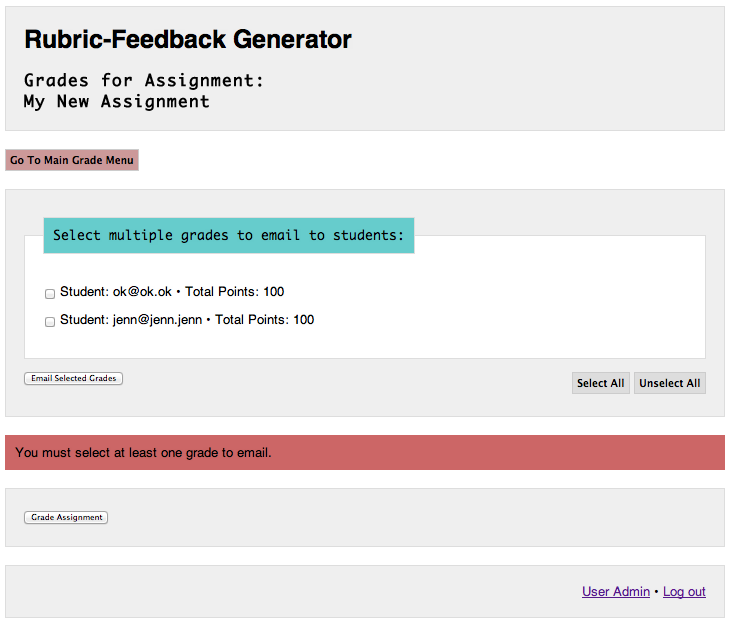


Figure 55: Batch Email - Warning User of no Emails selected after submitting form

The User can confirm the Email being sent before it goes out. An Email will be sent to each of the email addresses listed on the Email Grade Confirmation page. The “From-address” is the email address of the User, which is given when the User account was created or last edited. The content of the email is generated from the Rubric of the Assignment and values of the Grade being sent. Only one Email example is shown as a template of what will be sent.

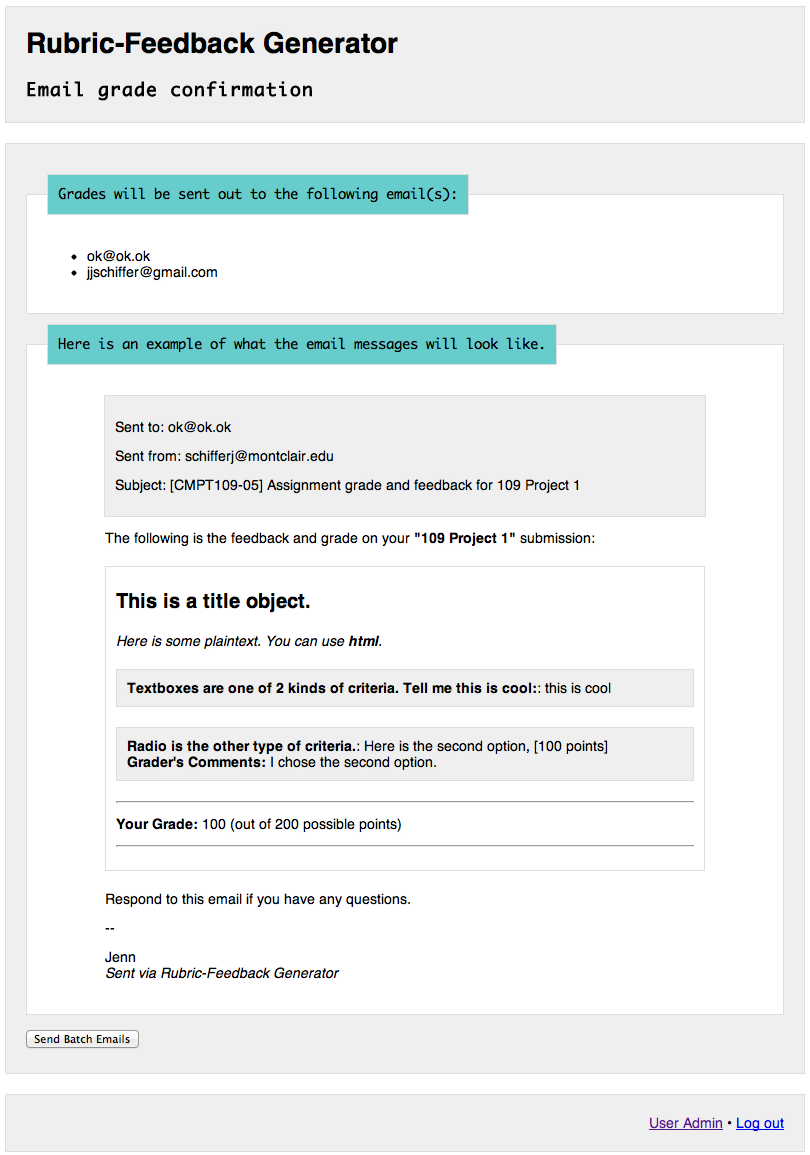


Figure 56: Batch Email - Confirming list of Emails and Email content

Upon successful Email sending of the Grade(s), the User will be given a link back to the User Admin page.

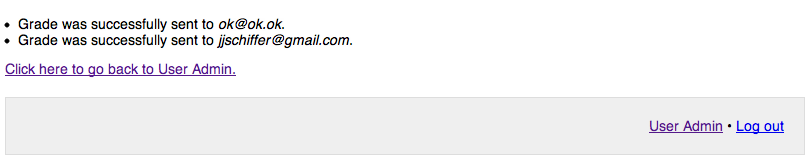


Figure 57: Batch Email page - User successfully sent one or more Emails of Grades

## Create New Rubric

A User can create a new Rubric by clicking “Create a New Rubric” on the User Admin page.

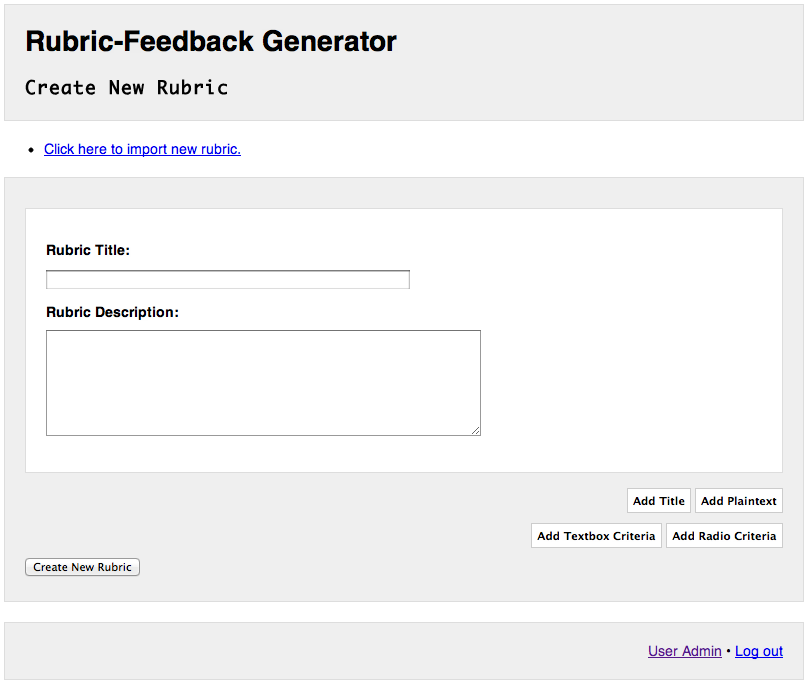


Figure 58: Create New Rubric page

The base fields of a Rubric are the “Rubric Title” and “Rubric Description.” Both have values of strings used to describe and identify a Rubric for the User.

There are four objects that make up the assessment part of the Rubric: Title, Plaintext, Textbox Criteria, and Radio Criteria:

**Title:**  Text that shows in boldface in a Rubric, typically for showing titles in a Rubric. In an Email, Title content is generated within <h1> and </h1> tags.

**Plaintext:** Text that shows without any type changes, typically for describing criteria or providing context to the receiver of an Email Grade. In an Email, Plaintext content is generated within <p> and </p>.

**Textbox Criteria:** A label given by a User for a textbox input.

**Radio Criteria:**  A label given by a User for a radio button selector. User can add multiple radio button options for Radio Criteria. Each Radio Criteria Option has a point value attributed to it. Radio Criteria Option point values are added to create the total score of a Grade.

The User adds the different objects to the Rubric by clicking their respective “Add” buttons.

Clicking “Add Title” adds the Title object. The “Order” of a Title is a number representing the position of this object in the Rubric. An Order of “1” means the object is first, for example. The “Title Content” of a Title is the text that shows when the Title appears in a Rubric.

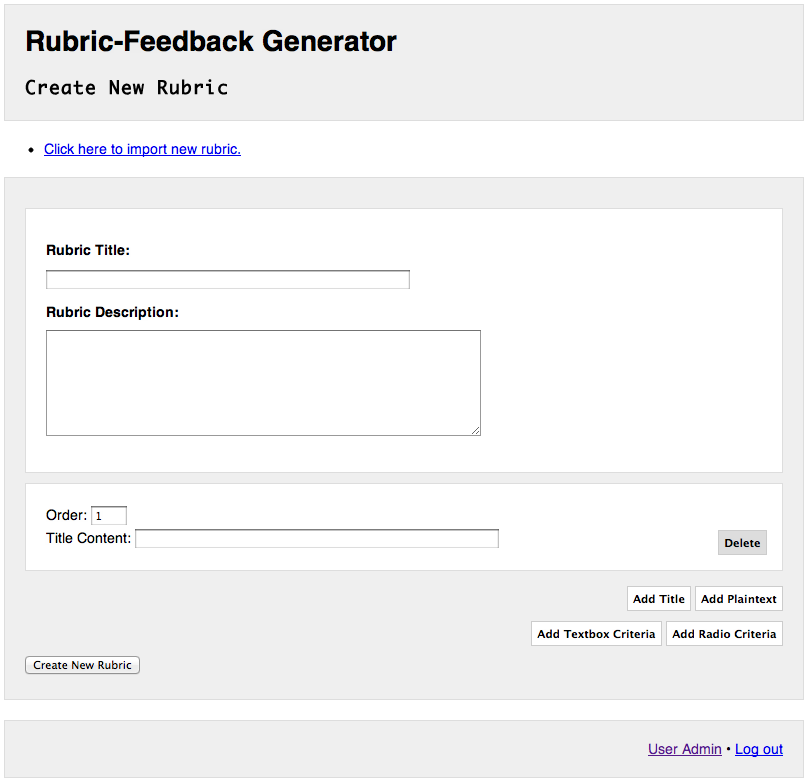


Figure 59: Create New Rubric page - Adding a Title to the Rubric

Clicking “Add Plaintext” adds the Plaintext object. The “Order” of a Plaintext, just like the Title, is a number representing the position of this object in the Rubric. The “Plaintext Content” of a Plaintext is the text that shows when the Plaintext appears in a Rubric.

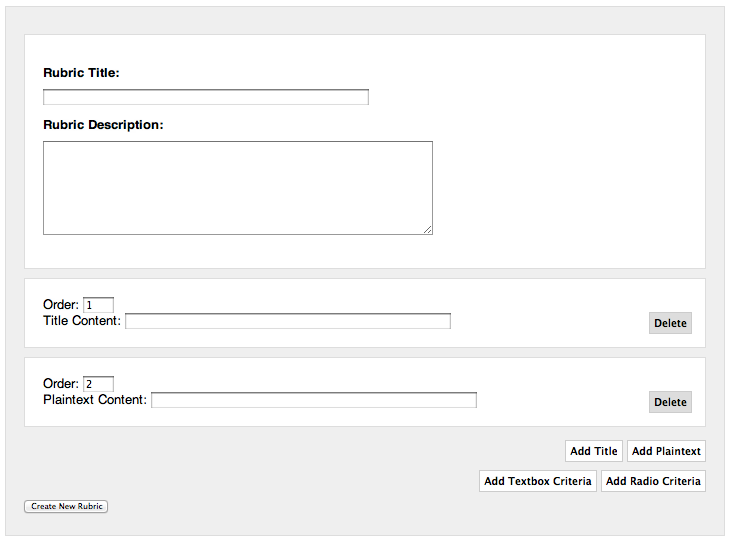


Figure 60: Create New Rubric page - Adding Plaintext to a Rubric

Clicking “Add Textbox Criteria” adds the Textbox Criteria object. The “Order” of a Textbox Criteria, just like the other objects, is a number representing the position of this object in the Rubric. The “Textbox Label” of a Textbox is the label of a textbox input that will be generated in the Rubric.

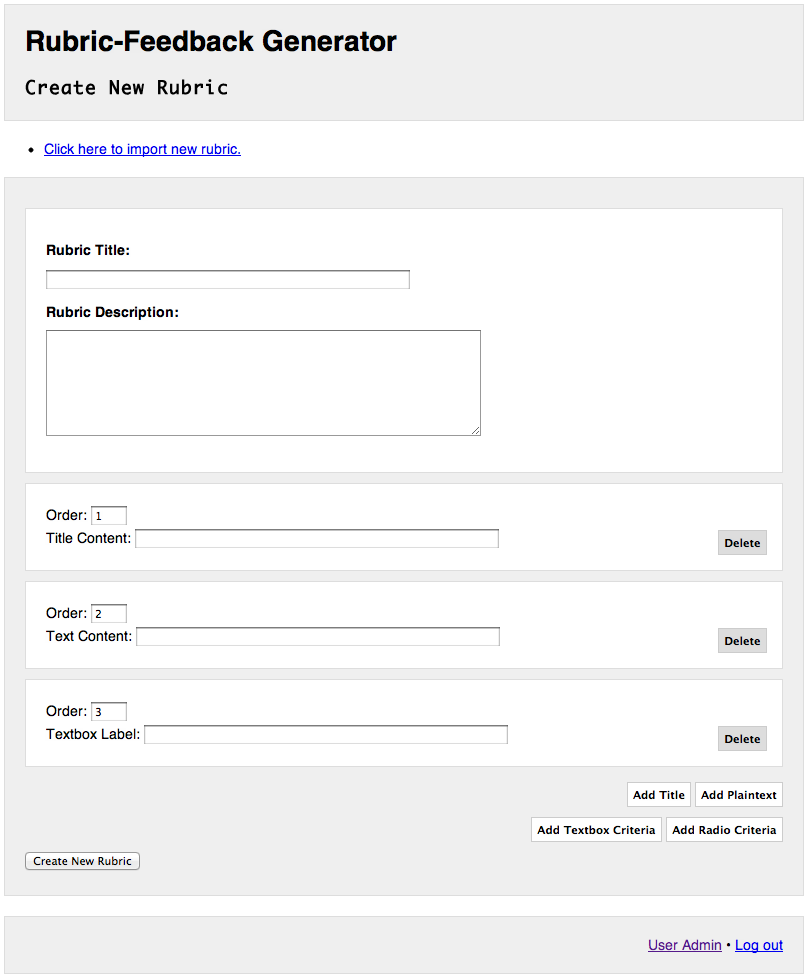


Figure 61: Create New Rubric page - Adding a Textbox Criteria to a Rubric

Clicking “Add Radio Criteria” adds the Radio Criteria object. The “Order” of a Radio Criteria, just like the other objects, is a number representing the position of this object in the Rubric. The “Radio Label” of a Radio Criteria is the label for a radio button fieldset that will be generated in the Rubric.

A newly added Radio Criteria object by default has one Option. An Option for a Radio Criteria has fields for “Option Order,” which represents the order placement of that option within the Radio Criteria fieldset it is a part of. The “Option Label” of an Option is the text label of that option. The “Option Points” is an integer value of how many points that option is worth if selected by the User when creating a Grade of an Assignment.

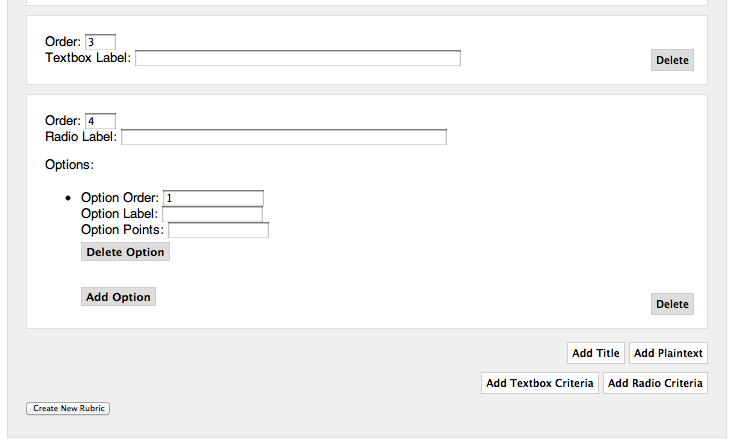


Figure 62: Create New Rubric page - Adding a Radio Criteria to a Rubric

Clicking “Add Option” within a Radio Criteria fieldset adds a new Option to that fieldset.

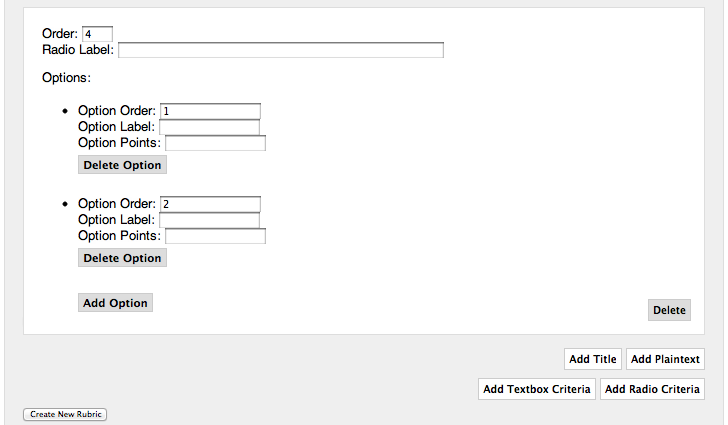


Figure 63: Create New Rubric - Adding an Option to Radio Criteria of a Rubric

Objects of a Rubric can be deleted by clicking their respective “Delete” buttons. To undelete, click the “Undelete” buttons. Options of Radio Criteria can be deleted by clicking “Delete this Option” and undeleted by clicking “Undelete this Option.”

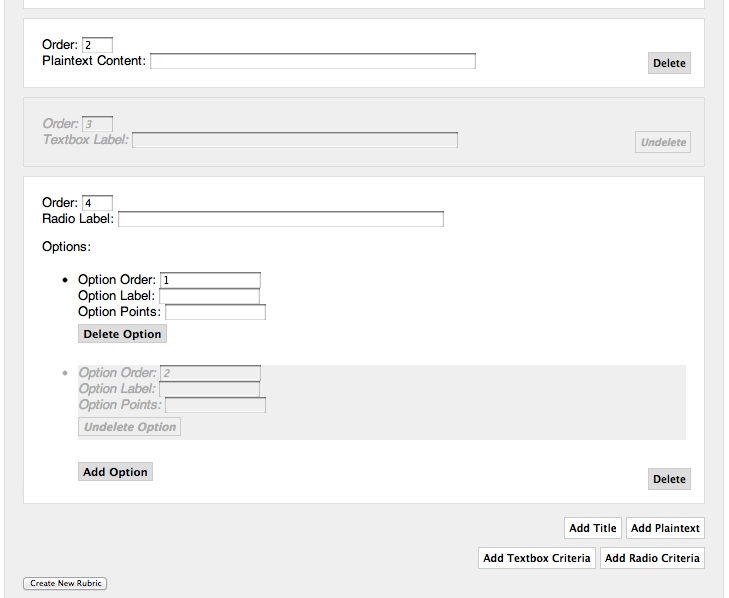


Figure 64: Create New Rubric - Deleting Values and Objects of a Rubric

All fields in the form must be filled or selected. User will be warned if one or both of these fields are blank after clicking “Create New Rubric.”

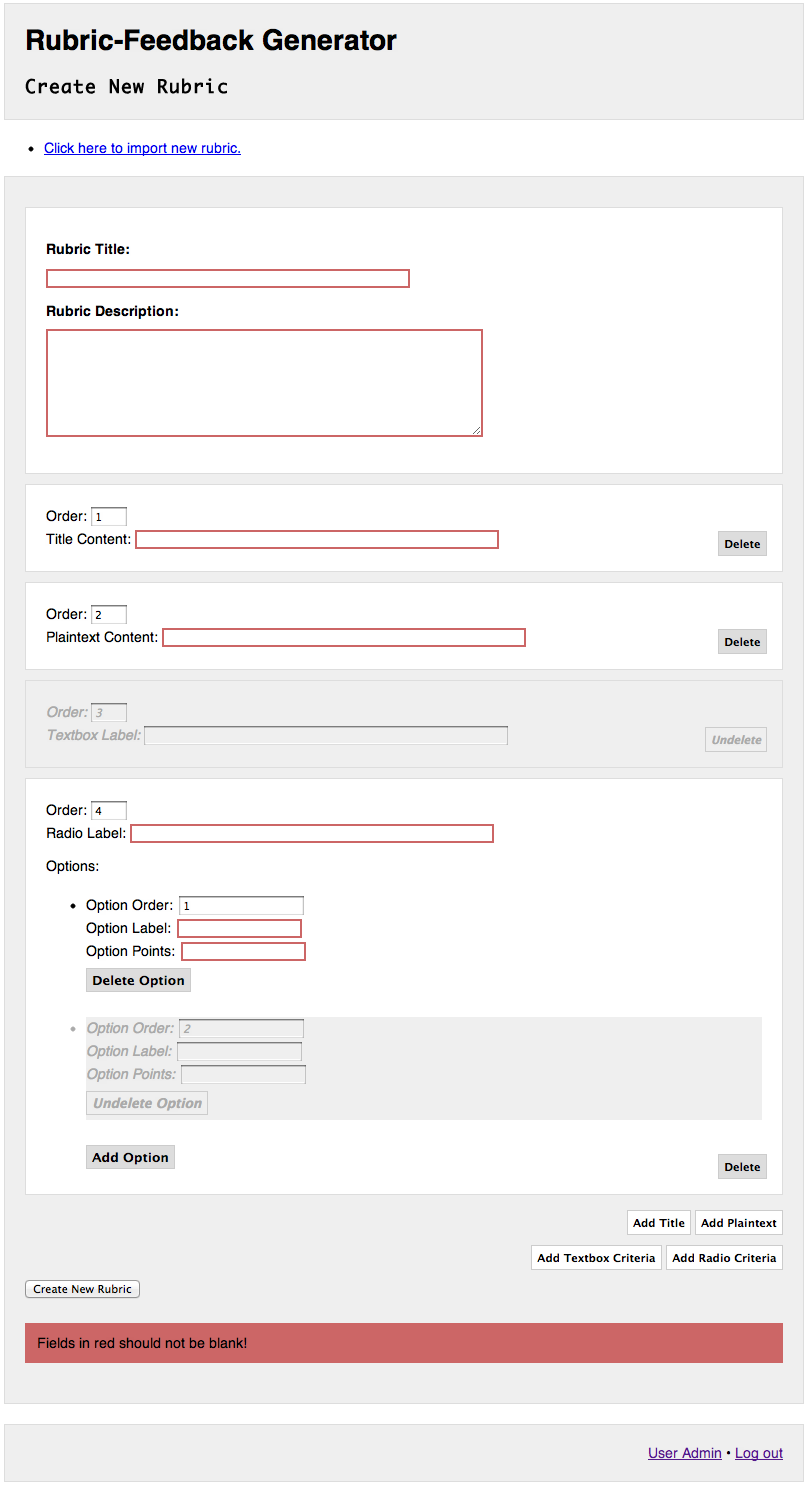


Figure 65: Create New Rubric - Warning User of blank fields after submitting a form

“Option Points” values must be an integer, either negative, zero, or positive. If an invalid value is submitted, the User will be warned.

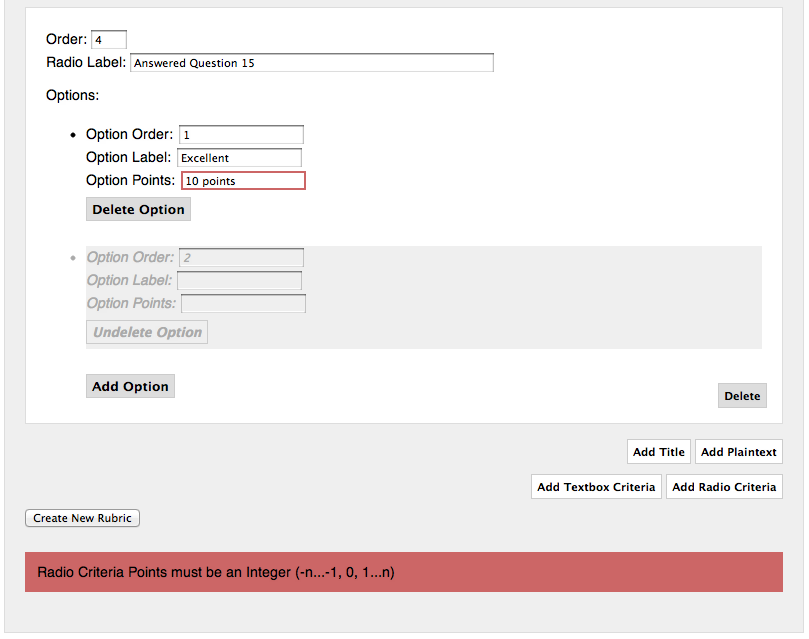


Figure 66: Create New Rubric page - Warning User of invalid point values after submitting a form

Upon successful creation of the Rubric, the User will be given a link back to the User Admin page.

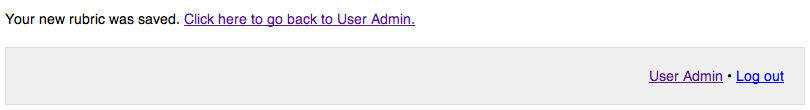


Figure 67 Create New Rubric page - User successfully created a new Rubric

## Edit Rubric

A User can edit already existing Rubrics by selecting the Rubric from the “Your Rubrics” menu on the User Admin page and clicking the “Edit Rubric” button.

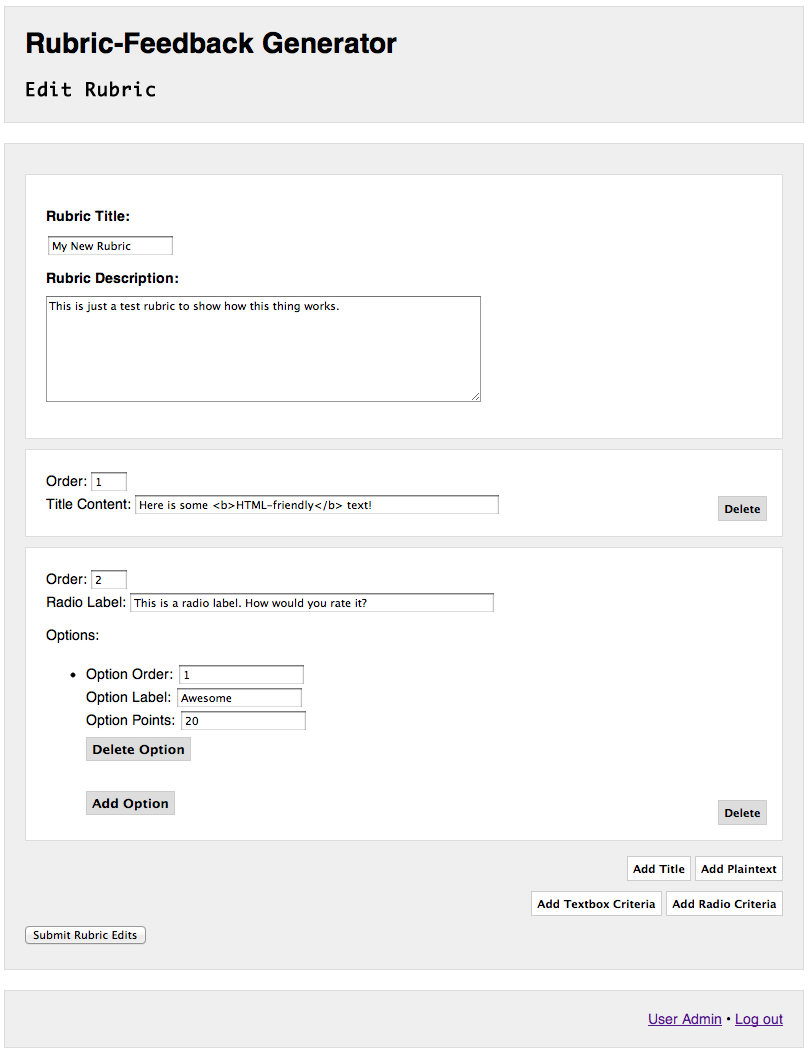


Figure 68: Edit Rubric page

All the same directions and validation rules apply for editing a Rubric as creating a new Rubric.

If Assignments exist that use the Rubric, a warning prompt appears above the Rubric, letting the User know that making changes to the Rubric may affect any Grades created for those Assignments.

It is recommended that the User goes to any Grades created with edited Rubrics and make any necessary changes on the Edit Grade page.

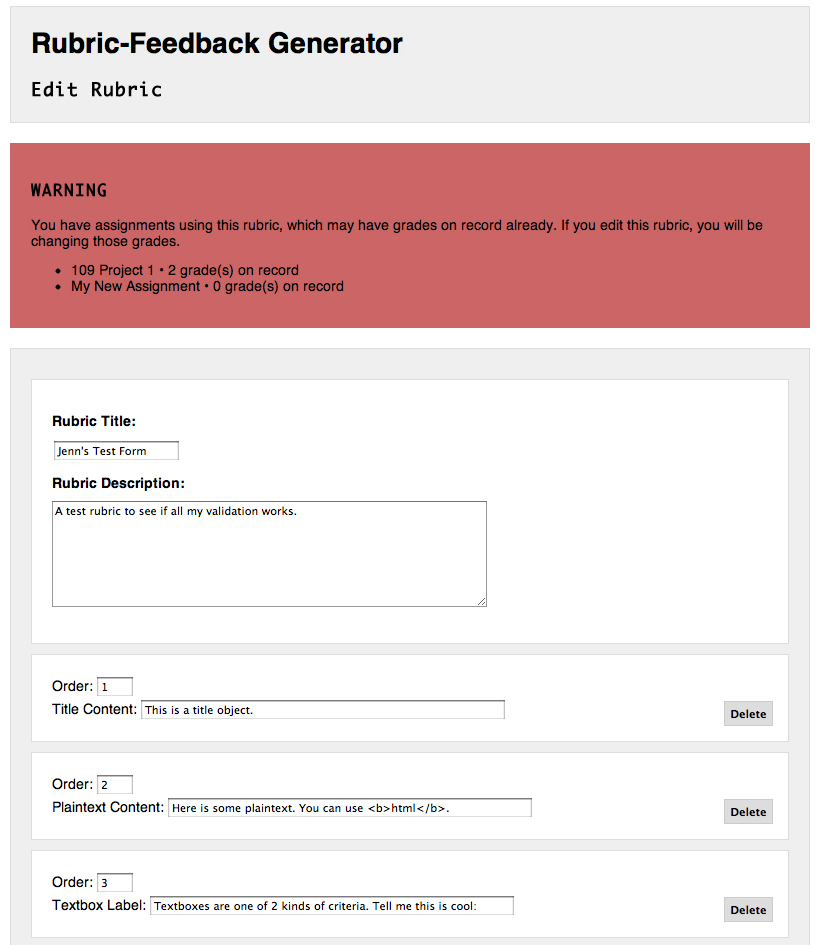


Figure 69: Edit Rubric Page - Warning User that Assignments using this Rubric exist

Upon successful deletion of the Rubric, the User will be given a link back to the User Admin page.

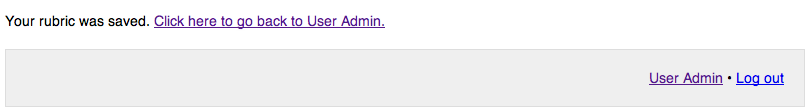


Figure 70: Edit Rubric page - User successfully edited the Rubric

## 

## Delete Rubric

A User can delete an already existing Rubric by selecting the Rubric from the “Your Rubrics” menu on the User Admin page and clicking the “Delete Rubric” button.

A warning prompt will appear below the “Your Rubric” menu. Clicking “Cancel” will cancel the delete request and hide the warning prompt. Clicking “Yes, Delete Rubric” will submit the request to delete the selected Rubric.



Figure 71: Deleting a Rubric

A Rubric can only be deleted if there are no Assignments that have the Rubric assigned to it. If any Assignments using the Rubric exist, the User is warned that the Rubric cannot be deleted. The User needs to delete all Assignments using the Rubric before deleting that Rubric successfully.

## Macintosh HD:Users:jenn:Dropbox:Masters:masters screencaps:rubric-delete-assignments.png

Figure 72: Deleting a Rubric - Warning User that Assignments use the Rubric and

therefore it cannot be deleted

Upon successful deletion of the Rubric, the User will be given a link back to the User Admin page.

## Macintosh HD:Users:jenn:Dropbox:Masters:masters screencaps:rubric-delete-success.png

Figure 73: Deleting a Rubric - User has successfully deleted the Rubric

## Import Rubric

A User can create a new Rubric by importing the content by clicking “Click here to import new rubric” on the Create New Rubric page.

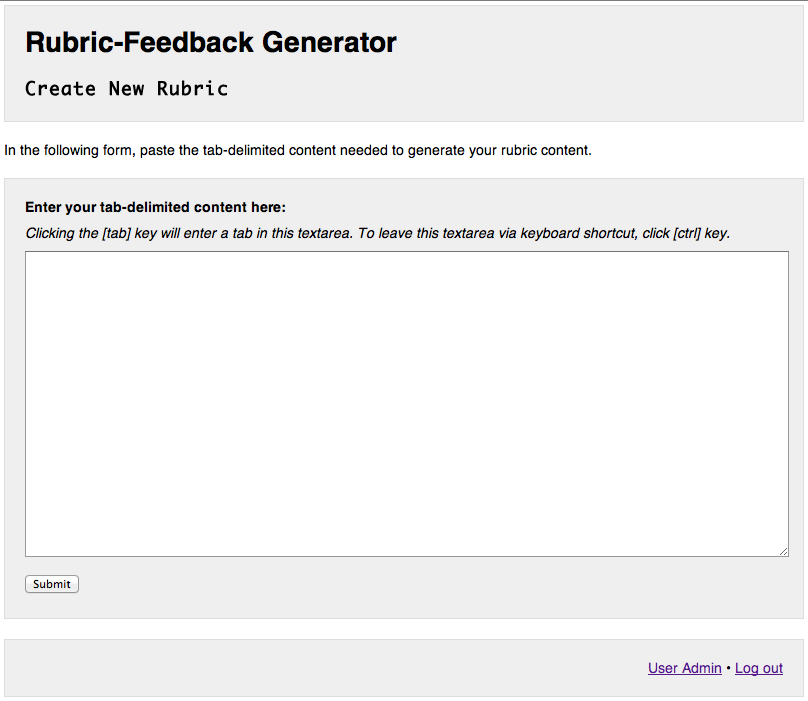


Figure 74: Create New Rubric via Import page

The content used to create a Rubric via import is tab-delimited, meaning tabs separate the different fields. This content can be created on this Create New Rubric via Import page, or in any text editor or spreadsheet software.

Each row, or record, of the content represents one Rubric object: a Title, Plaintext, or Criteria. Title and Plaintext only have one field, its content.



Figure 75: Importing Rubric - Example of a Title object and Plaintext object in a tab-delimited spreadsheet

The Criteria object has variable fields. The second field is the type of Criteria: Textbox or Radio. The third field is the Criteria label. Following the type and label are the Option labels followed by Point values for that Option before it.

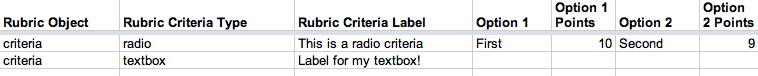


Figure 76: Importing Rubric – Example of a Radio Criteria and Textbox Criteria object in a tab-delimited spreadsheet

There is no limit to the number of options a Radio Criteria can have. The User must keep the order of Label-then-Points in order for the Option to show up in the new Rubric.

The User can add this tab-delimited content by typing directly in the Import textarea, or copying and pasting the content from a text editor or spreadsheet.

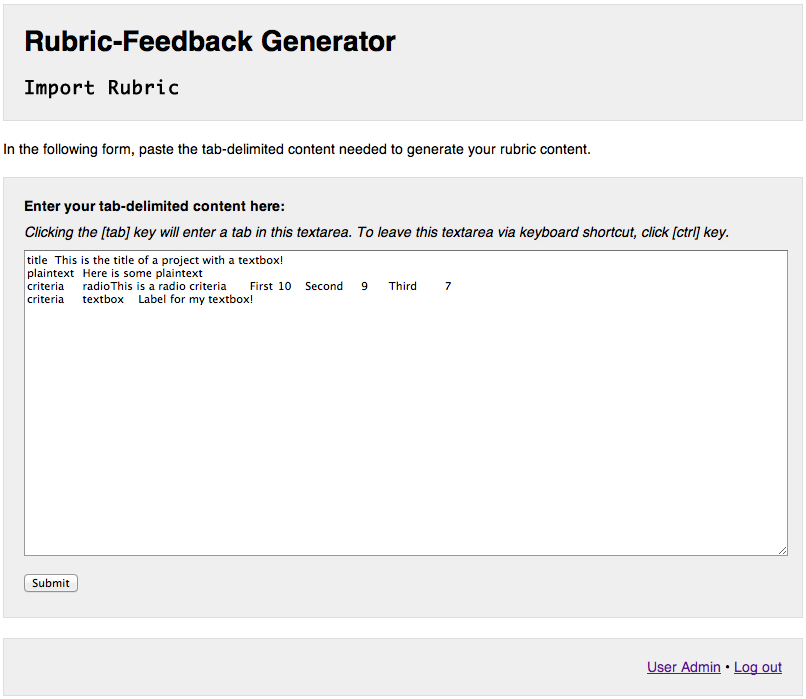


Figure 77: Importing Rubric - User enters tab-delimited Rubric content

The tab-delimited content textarea is required when click “Submit.” The User will be warned if this textarea is blank after submitting the form.

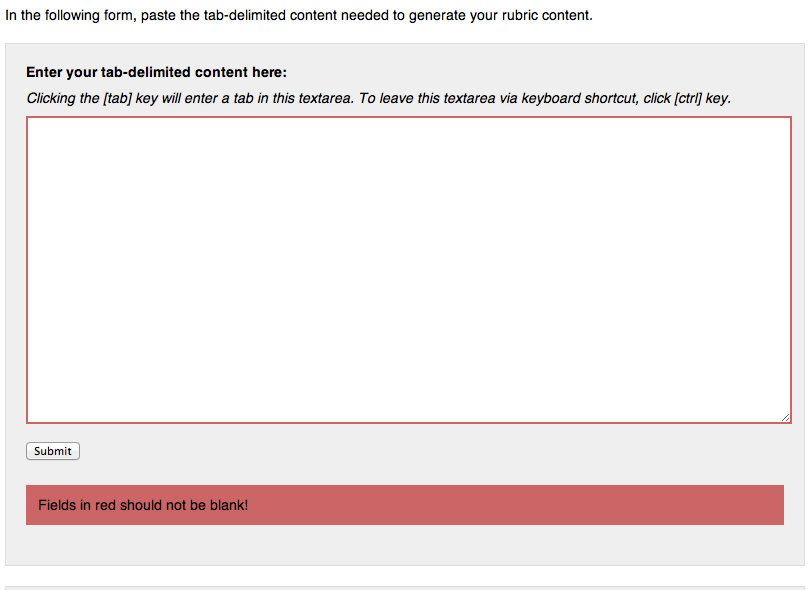


Figure 78: Importing Rubric - Warning User of blank textarea after submitting form

After User clicks “Submit,” they will be sent to the Verify Rubric Import page. This page gives the User a view of what the Rubric they imported looks like. There is also another textarea to allow the User to edit the tab-delimited content and resubmit changes.

Additionally, a Rubric has two fields for identification: “Rubric Title” and “Rubric Description.”

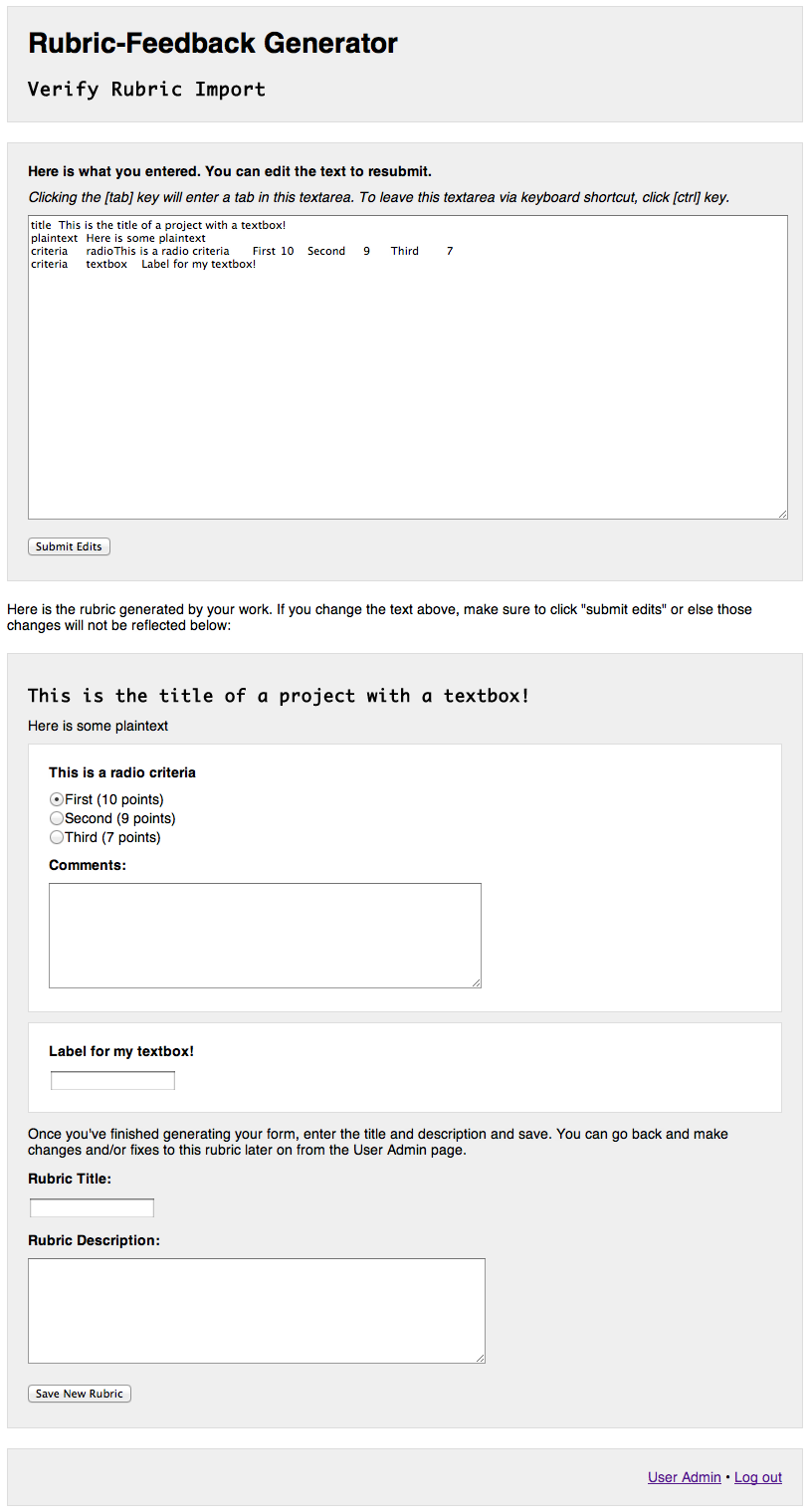


Figure 79: Verify Rubric Import page

If the tab-delimited content is invalid, the User will be warned. Rows with invalid content will be ignored if they exist when the User clicks the “Save New Rubric” button.

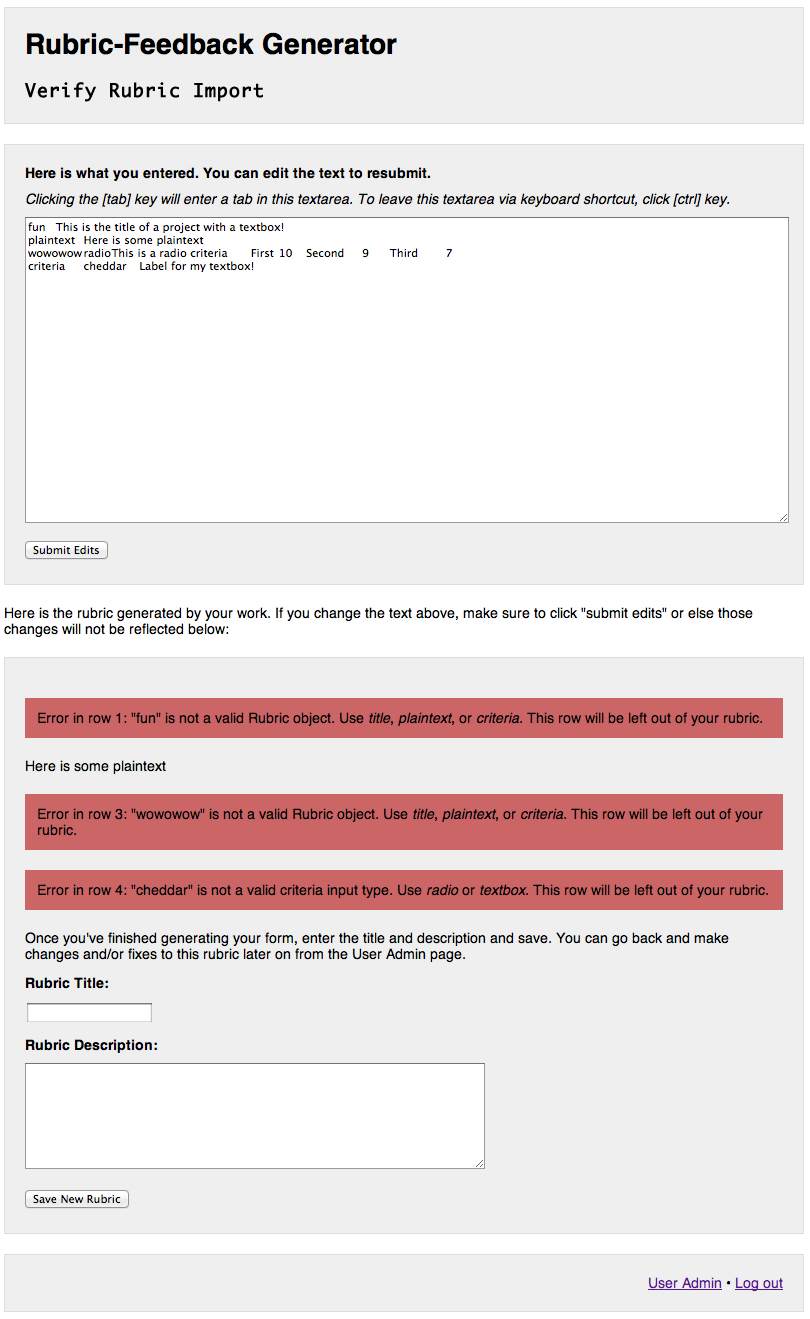


Figure 80: Verify Rubric Import page - Warning User of tab-delimited rows with invalid content

The Rubric Title and Rubric Description are required to create the new Rubric. User will be warned if one or both fields are blank after clicking “Save New Rubric” button.

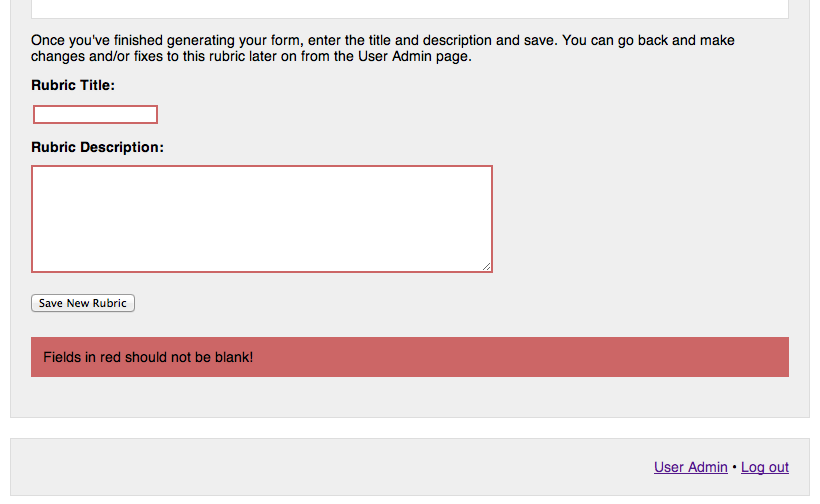


Figure 81: Verify Rubric Import page - Warning User of blank fields after submitting form

Upon successful import of a new Rubric, User will be given a link back to the User Admin page.

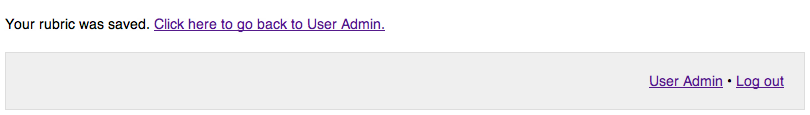


Figure 82: Verify Rubric Import page - User successfully imported a new Rubric

# V. Source Code

A CD, which contains all of the source code for this project, is included with this report.

You can also find the source code online:

https://github.com/jennschiffer/Rubric-Feedback-Generator

# Bibliography

(PHP Cookbook)

(jQuery Docs)

(Essential JavaScript)

(MySQL Guide)