

CIT 597 – Project – Iteration 2

Iteration 2 Deliverables

For this iteration, there are four main deliverables/tasks.

Pivotal Tracker – User Stories, Point Values, Estimates, Velocity (10 points)

Plan your iteration in Tracker. You can update user stories/point values/velocity if needed. Keep it up to date as you work through this iteration.

Note: You shouldn't edit anything from a previous iteration.

Models and the Database (40 points)

Add in all the models that you need for your project. For every model, set up the appropriate associations, validations, and migrations.

Note: It's fine to use scaffolds here as well. Alternatively, you can use 'rails generate model ...' if you only want to generate the model, and not the view and controller.

Controllers and Routing (30 points)

Add in more functionality for your controller(s). Set up appropriate routes for them.

Note: Don't worry too much about the view for this iteration. That will be the focus for the last iteration. You can still use scaffolds for the views, if you'd like.

Cucumber (20 points)

Continue using Cucumber/Capybara. For scenarios/features that were failing in the previous iteration, ensure that they pass as you implement more functionality. For scenarios/features that were passing in the previous iteration, ensure that they still pass and you didn't break anything while adding new functionality.

For new feature that you implement in this iteration, add at least two new scenarios.

Grading Criteria

This assignment is worth 100 points (10% of the overall grade).

Note: For this assignment, late submissions are not allowed.

Submission Instructions

Please submit the following:

1. A text file with the link to your github and Tracker projects.
2. Generate a Cucumber HTML Report for your project. Run 'cucumber -f html > cucumber.html' from your Rails project folder. Upload the generated cucumber.html file.
3. A text file called README – this should explain your decision decisions for the models (Why are these your models? Why these associations? Why these validations?) and the controllers (Why this functionality in this controller? Why this functionality in this iteration versus the next iteration?)