COM2025 Web Applications Development
Project

Handout Date:	Wednesday 14th October 2020
Deadline:	Monday 7th December 2020 by 4:00pm via SurreyLearn
Written Feedback:	Friday 8th January 2020
Module Weight:	Assignment worth 50% of the module

Academic Misconduct: Coursework will be routinely checked for academic misconduct. Your submission must be your own work. Please read the Student Handbook to ensure that you know what this means. Do not give your code to anyone else, either before or after the coursework deadline.

1. Purpose

The purpose of this assignment is for you to demonstrate your ability to:

- Develop an interactive web app.
- Use Ruby on Rails.

This will involve you building a web app using Rails, which will include Ruby classes and a database that will produce HTML, CSS and jQuery that will be used as the user interface to your app via standard web browsers.

2. Problem Definition

Your first task is to select an application that you wish to develop. We are not constraining your choice so that you have an interest in what you are developing, and scope to make it your own.

To help with your selection we are expecting your application at a minimum to include the following elements:

- A home page.
- A contact page which provides a form for you to contact the site owners via email.
- Resource pages linked to at least three models which are contained in a database. You must show resource associations in your models.

Some example sites you might consider:

1. Car park space bookings: a system to allow you to book a parking space in advance. The site could report the available spaces, location of cars and costs. We are not expecting any payment mechanism.
2. Events management: a system to allow you to manage a series of events. Each event will have a number of tasks associated with it all linked to a calendar and deadlines.

3. Module option selections: a system which will allow students to select modules that they wish to take form a list of optional modules. The system must ensure that students can only take modules which add up to the required number of credits.
4. A football league system: a system used to track football teams in a league, record points and display the current league table and fixtures. Feel free to pick any sport if you do not like football.
5. Music collection: a system to keep track of your music collection, including album and track information, plus your own notes or favourites.

Feel free to pick your own idea, but make sure it has enough complexity to meet our requirements, while not being too ambitious in scope. If in doubt, please ask.

3. Requirements (marks)

We are expecting to see a fully working Rails application which includes:

1. Views (30%):
    a. A home page.
    b. A contact page with contact form.
    c. Resource pages for all of your appropriate model resources. Only provide routes and actions to all of your resource paths if they are needed.
    d. All views conforming to best practice guidelines for HTML5 and CSS, for example as through boilerplate templates, with styles used to present content in a clear and appropriate way.
    e. Partials used to ensure that each page has a standardised header and footer.
    f. jQuery used where appropriate to assist in providing a good user experience (for example, checking that form fields are not empty prior to submission, if they are required).
2. Controllers (20%):
    a. A controller to handle your home and contact page.
    b. Controllers for each of your model resources with an appropriate use of strong parameters.
    c. Only actions for those routes which are exposed.
    d. Appropriate tests for each of your controller actions, which all pass. You should also test basic view content within your controller tests.
3. Mailer (10%):
    a. A mailer to handle your contact request. This does not need to be linked to an SMTP server.
    b. Appropriate previews and tests for the mailer, which all pass.
4. Models (30%):
    a. A series of database migrations that you have used to build your application, with the schema including appropriate fields, data types, defaults, nullability and indices.

   b. At least three model classes which provide appropriate validations, associations and scopes.
   c. Appropriate tests for each of your models, which all pass.
5. Configuration (10%):
   a. Appropriate environment configuration for your application, including gems and environment settings.
   b. Only the required routes needed for your application.
   c. The use of localised strings throughout your application.
   d. A git repository showing your development through regular commits.
   e. Appropriate use of comments and code style throughout.

Refer to the attached marking scheme for the range of marks and expectations. The marking scheme is based on the University Grade descriptors from the Code of Practice. To get the highest bands in each category there, you will need to go above and beyond the lecture material to more advanced material. Please bear this in mind if you want to get the top marks. **Before attempting more advanced features, make sure that you have the basics working!**

4. Advanced Features

If you would like to include additional items in your project, feel free to do so, but please ensure that you meet the minimum requirements for marking, and that your optional extras **do not** break your code. Do not try and attempt everything here.

Some things you might consider:

- Authentication using Devise, perhaps with additional support for permissions and control using Pundit or Active Admin.
- Support for file uploads using Paperclip or CarrierWave, perhaps with ImageMagick support.
- The use of AJAX to provide in-place updates, or live streaming.
- Use of GitHub as a remote repository or deployment of your app to Heroku.
- Integration of other web technologies (websockets, react, etc)

For any gems, images or other sources you use please ensure that you do not infringe any copyright (check image permissions) and only use gems for which the license permits (check the license).

5. What to submit?

You can use whatever development tool you like to develop your application, including RubyMine or the command line. However, your app must be built using Ruby on Rails 5.2.* and Ruby 2.3 or above. Your code and test cases should run in the lab environment. Some variation in the Ruby version is anticipated. Please try an use the Rails 5 framework unless you have good reason not to and then ask in advance of starting work.

- Upload an archive containing the whole of your Rails app.
  - Your archive should include all of your source code and git repository.

    - Call the file username_com2025_project.tar (or .zip), where username is your username.
    - Upload your file to SurreyLearn.
- We will run the development version to mark your app using “rails server”.
- I will be testing using the Chrome browser unless otherwise specified. To reduce compatibility issues, please test your web app on the Chrome browser before submitting.

Please use the SurreyLearn assignment Discussion Board to ask any questions.