DB Seeds:

```ruby
User.create!(name:  "Mika",
             last_name: "Williams",
             dob: "11/11/1111",
             gender: "Other",
             email: "test@test.org",
             password:            "password",
             password_confirmation: "password",
             admin: true)
User.create!(name:  "Mr",
             email: "teacher1@teacher.org",
             last_name: "Teacher",
             dob: "11/11/1111",
             gender: "Male",
             teacher_qualifications: "Finsished Highschool",
             password:            "password",
             password_confirmation: "password",
             teacher: true)
User.create!(name:  "Ms",
             email: "teacher2@teacher.org",
             last_name: "Teacher",
             dob: "11/11/1111",
             gender: "Female",
             teacher_qualifications: "Finsished Highschool",
             password:            "password",
             password_confirmation: "password",
             teacher: true)
20.times do |n|
  name  = Faker::Name.first_name
  last_name = Faker::Name.last_name
  dob = "11/11/1111"
  gender = "Other"
  email = "example-#{n+1}@student.org"
  password = "password"
  User.create!(name:  name,
               email: email,
               last_name: last_name,
               dob: dob,
               gender: gender,
               password:            password,
               password_confirmation: password)
end
```

Figure 1. Database Seeds used for Release 1

The above seeds (Figure 1) were used to populate the database with users. This was completed so that the other group members could view how the website works and also be able to navigate the website without having to create their own account. I create a seed for one admin named Mika, two seeds for teachers and then a random 20 seeds for students. I believe this was a decent enough variation in seeds to display the website. Additionally, the 20 student used the faker gem to obtain sudo random names.
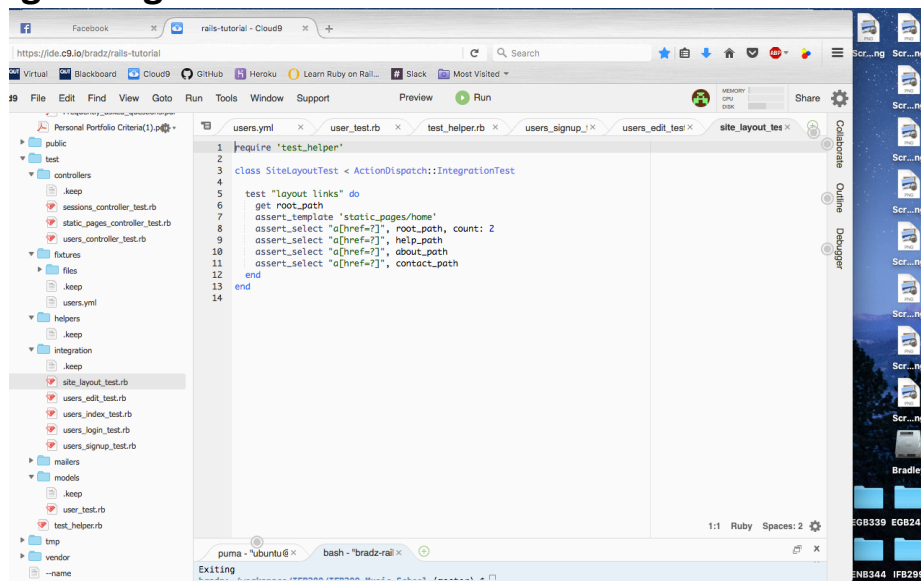
Static Page Integration Test:

Figure 2. Static Page Integration Test used for Release 1

Figure 2 shows an example of an integration test used in the project for release 1. This specific test is used to make sure that all links to the other pages can be found in the header of each page. If any path name was changed, test would display an error and then the test files would be changed to reflect these changes. Many of the integration test files were changed throughout the production of release 1.

Test Fixtures:

Figure 3. Test Fixtures used for Release 1

Figure 4. Test Fixtures used for Release 1 continued

Above (Figure 3 and 4) show the test fixtures used to test the controllers on the website. These fixtures remain very close to the fixtures shown in the tutorial by Michael Hart (So much so that one fixture is named after him). Some fixtures were changed to include fields added and those that were not were tested to make sure that they were unable to sign up as a user. From playing with these fixtures and database files it was found that these fixtures actually get stored in an SQLite file. This file was also downloaded and viewed to see which users were allowed to sign up.
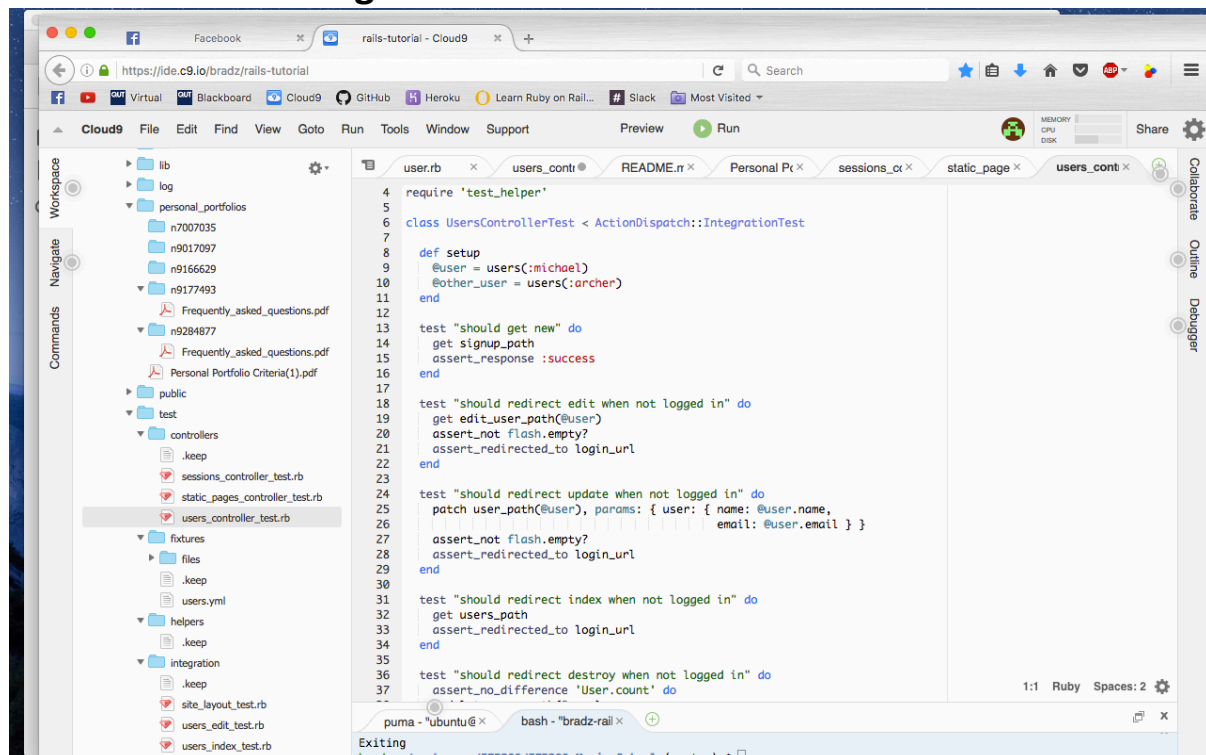
Controller Test using Fixtures:

Figure 4. Test Fixtures used in controller test for Release 1

Above (Figure 4) shows the fixtures being used as controller tests. They were used to test things such as if the user was able to login in with the given parameters and making sure that the admin did have the power to delete users. Once again this followed closely with the tutorial and some addition features were not added as other members will be going through the website and testing to make sure acceptance criteria is met.