

Lab Activity 4: Resources, Simple Form, Validations, Associations

The objectives of this lab is to implement a simple student record system using the Rails framework. The output should show how the resources routing, simple_form gem, validations and associations work in Rails.

1) Create a new project called studentmanagementsystem

2) Create the Student model with the following fields and validations:

Student

- first_name (string, required)
- last_name (string, required)
- middle_name (string, required)
- date_of_birth (date, required)
- address (text, required)
- student_number (string, required, unique)
- contact_number (string, required)
- email (string, required, unique)

Define a to_s method in the model that returns the student_number of the instance.

3) Create the Subject model with the following fields and validations:

Subject

- name (string, required, unique)
- code (string, required, unique)
- num_units (integer, required, numerical)

Define a to_s method in the model that returns the name of the instance

4) Create a model called StudentSubject with the following fields:

StudentSubject

- student_id (integer)
- subject_id (integer)
- date_enrolled (date, required)
- date_completed (date, required)
- letter_grade (string, required, possible values: [A, B+, B, C+, C, D, F])
- numeric_grade (decimal, required, numerical)

5) Create resource paths for all the models:

```
resources :students
resources :subjects
resources :student_subjects
```

6) Create the necessary associations for each model:

```
# student.rb
has_many :student_subjects
```

```
# subject.rb
has_many :student_subjects
```

```
# student_subject.rb
belongs_to :student
belongs_to :subject
```

7) Complete the controllers for each resourced routes. You should have the following files in your controller directory:

```
- students_controller.rb
- subjects_controller.rb
- student_subjects_controller.rb
```

8) Add the following to your Gemfile:

```
gem 'simple_form'
```

Run “bundle install” to include the library to your project.

In the forms part of your system, make sure you use simple_form for the user to enter values.

9) In the show view of a student, make sure it also displays a table with all the subjects that student took and its corresponding numerical grade.

10) In the show view of a student, create a custom route:

```
get "/students/:id/add_student_subject_record", to: "students#add_student_subject_record", as: :add_student_subject_record
get "/students/:id/edit_student_subject_record/:student_subject_record_id", to: "students#edit_student_subject_record", as: :edit_student_subject_record
```

The view for this method should display the new and edit pages of the student_subjects model and redirect to the student show page upon successful save.