**Advanced Web Development – “Unit and Functional Testing Lab”**

Points: 25

Due: Night (11:59 pm) before next class.

1. Clone the depot\_p\_for\_testing project from https://github.com/siddharthkaza/depot\_p\_for\_lab.git under the lab and import into the IDE.
   1. Run the application to make sure its running in the browser.

*Hint:* It’s always a good idea to run ‘bundle install’ and then ‘rake db:migrate’ and ‘rake db:seed’ first - when using an application from another location. This updates any gems needed and brings the DB to its newest version.

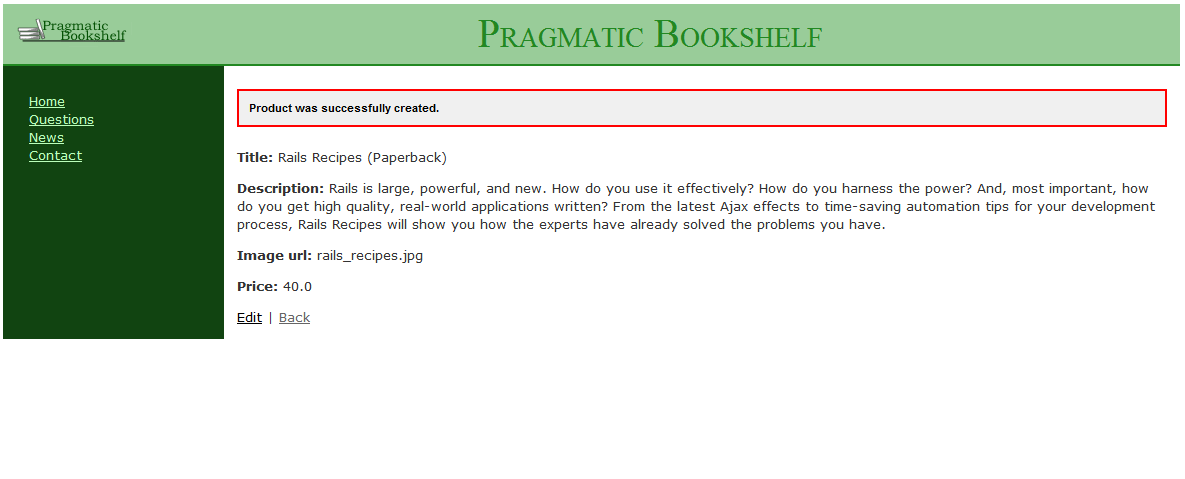
**The application ran fine!**

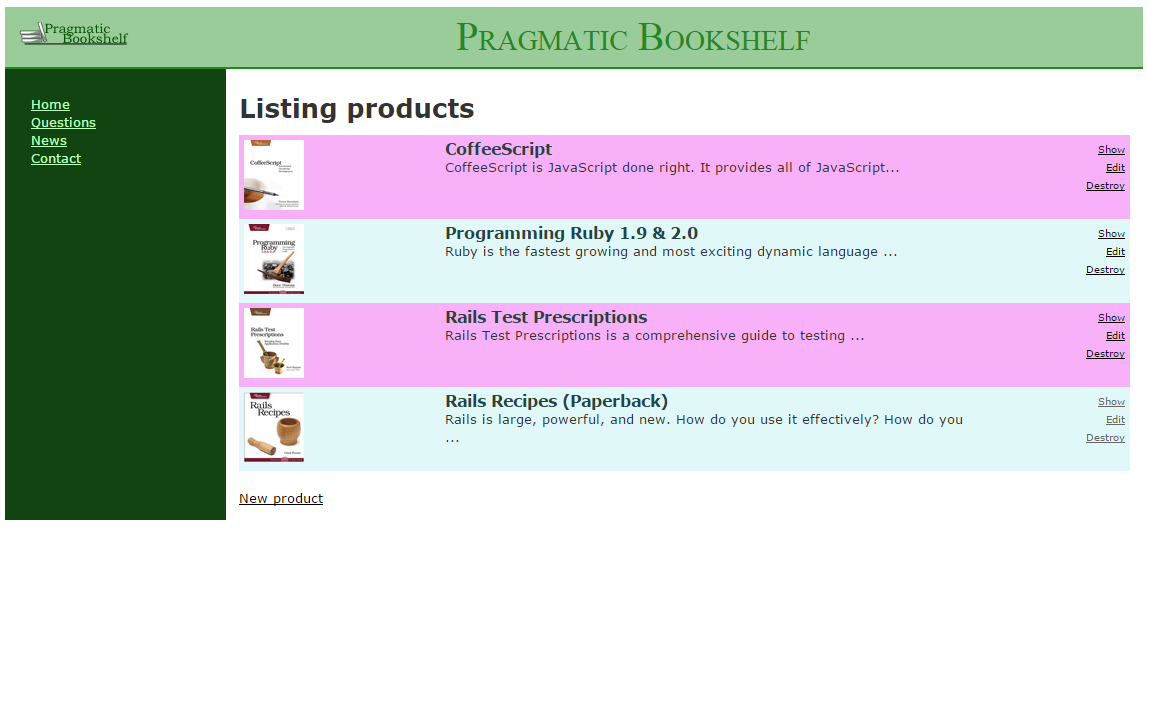
1. **Validations (5 points)**

Validations are methods that are placed in the model to validate data before records are entered in the database.

In the depot application, we will put in validations in the product model to prevent bad data from being entered in the database (for instance, ‘a product should always have a title’).

1. Run the depot application, enter a new product in the database (go to /products in the URL to get to the products listing).





1. What validations do you think the product model needs? List them here.

There needs to be validation for all the mandatory fields. For instance, the image url field is absolutely mandatory since we want to display and image of the book. The title field must not be blank, and it must be unique since we don’t want two books having the same name. We could also make sure the Description is not blank. The title must only be alphanumeric. The image url must end with an acceptable file extention. The price must be in decimal format and only numbers.

1. Enter the following statements in the product model (product.rb) one-by-one

validates :title, :description, :image\_url, presence: true

validates :price, numericality: {greater\_than\_or\_equal\_to: 0.01}

validates :title, uniqueness: true

validates :image\_url, allow\_blank: true, format: {

with: %r{\.(gif|jpg|png)\Z}i,

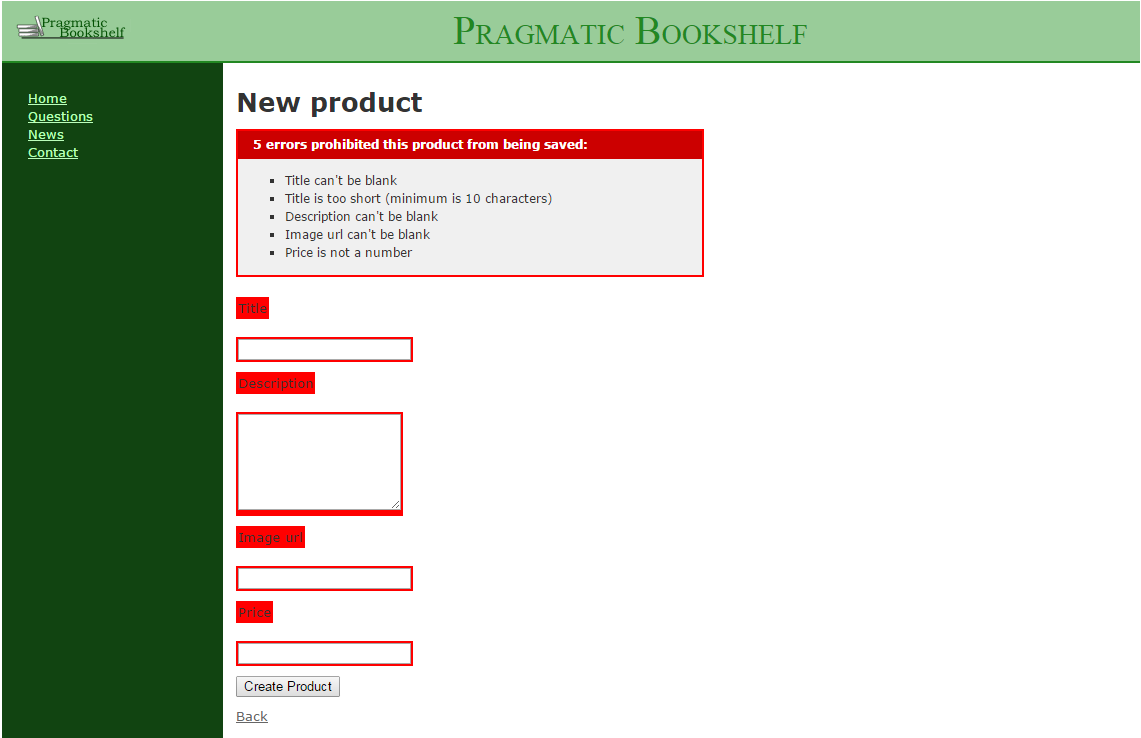
message: 'must be a URL for GIF, JPG or PNG image.'

}

validates :title, length: {minimum: 10}

1. Create another product that fails the validations (for e.g., leave the title empty). You notice that errors show up at the top of the page after you have. How does that work? What data structures are being used in the view to show the errors?

Each active record generates its own error messages. These can be accessed using the error attribute. Whenever a validation is set for a model any errors are stored in the error hash. The data will only be stored if the error hash is empty.



1. Are there any more validations that you want see in the products model? Think about how you would add them. Bring them up when we discuss the lab in the class.

We can also check to make sure that the title is title-cased such as ‘The Rails Guide’ and not ‘the rails guide’.

1. **Testing (10 points)**
   1. Study the model tests in the application. Run those model tests and see the results. The command to run all the model tests (also known as ‘unit’ tests) in your application is ‘rake test:units’

Do you think the application needs more unit tests for products? Explain here.

* 1. Study the functional tests in the application. Run the functional tests and see the results.
     1. You will notice that some functional tests fail. Why do they fail? Study the assert\_difference method for one of them. When one of the test fails you get a message:

Product.count didn’t change by -1

<2> expected but was

<4>.

What do the numbers <2> and <4> mean?

* + 1. Fix the code so that all failing tests pass.

1. Download the addressbook application from the blackboard (it’s under the slides>addressbook\_rails41.zip). Write two unit tests and two functional tests for the application. See the examples in depot\_p. You might have to write the validations in the model to write unit tests. (10 points)