

COMP41100

EXPLORING PROGRAMMING IN RUBY

PRACTICAL NINE

22nd November 2013

Andrew Doyle
Student Number: 12252388

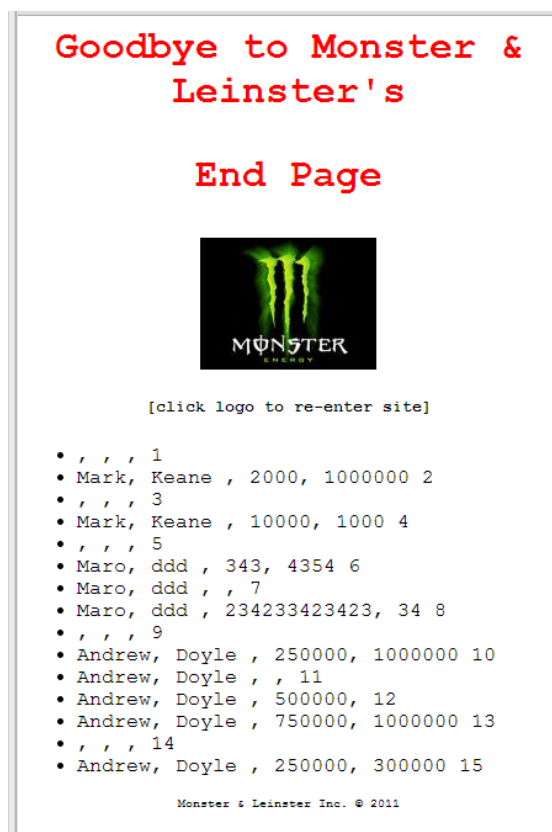
Q1

Task:

Go back to the hello program that does the mortgage stuff and get a better idea of how the different records are being created as you go along. Have a close look at the methods in the Other Controller (esp. index, change and quote) and describe how they are interacting with the database. They could be better, no? So, modify each method to make it interact with the db in a better way.

A major problem with the hello program is that records are duplicated; for example, when editing a record, a new record will be created (with the previous version remaining – as shown in Figure 1.1).

Figure 1.1 – Duplicated information



Instead of creating a new entry each time, the entries should be updated. The HTML file should retrieve an ID for a record which will be retrieved by the method. This would allow the update of the method instead of duplication.

Proposed changes are outlined below, with comments to outline changes:

Index Method

```
def index(record_id) #parameter added to identify record (give <li> tag an id)
  @id = record_id
  @person = Entry.find_by_id(id) #instead of entry.find use entry.find_by_id
  @fname = @person.first_name
  @lname = @person.last_name
  @person.update_attributes({:address => params[:st_name],
    :salary => params[:salary], :loan => params[:loan],
    :loan_reason => params[:reason]})
  if !@person.address.nil? then render "show" end
end
```

Change Method

```
def change(record_id)
  @id = record_id
  @person = Entry.find_by_id(id) #instead of entry.find use entry.find_by_id
  @fname = @person.first_name
  @lname = @person.last_name
  @entry = Entry.create({:first_name => @fname, :last_name => @lname,
                        :salary => params[:salary], :loan => params[:loan]})
end
```

Quote Method

```
def quote(record_id)
  @id = record_id
  @person = Entry.find_by_id(id) # changed to use entry.find_by_id
  if !@person.salary.nil? then
    if (@person.salary * 3) < @person.loan
      then @message = "You goddam pauper, you asked us for a loan that is:
                       more than three times your salary. \n Please go back
                       and re-enter your correct salary."
    else @message = "Sure thing baby..."
    end
  else @message = "Seem to have an empty record??"
  end
end
```

Q2

Task:

Create your own rails app that has, at least, 3 separate pages and 3 controllers. Do not worry too much about layout, but just get each page to do something different and design it so that you can move from one page to the next by clicking on links.

The homepage provides clickable image links to each dog's page; tom embeds a youtube video and tammy displays an image slideshow.

Figure 2.1 - Homepage:

Andrew Doyle`s Dogs

A dog's love seems to have no end
When humans want to hate
a dog, simply wants to mend



[click Tom`s picture to enter]



[click Tammy`s picture to enter]

Andrew Doyle 12252388 Practical 9 Inc. © 2013

Figure 2.2 – Tom the Dog

Andrew Doyle`s Dogs!!



Welcome to the page of Tom the Doggy!. That`s me below!



Check out this awesome song that I love below:



Tom and Tammy the doggies take no responsibility for you falling in love with them. Find yourself your own doggy.

Figure 2.3 - Tammy page:

Andrew Doyle`s Dogs!!



Welcome to the page of Tammy the Doggy!. That's me below!



Here is Tammy
doing some
study



Tammy in the
sitting room



Tammy chilling
on the kitchen
floor



Tammy watchin
tv (Tom in the
background)

Q3

Task:

Finally, is it possible to move between pages without having an explicit link on which a user clicks? If so, how?

Yes; for example, by using [automatic rendering](#) or by using the rails [autolink](#) gem. The gem was previous built into rails, but was removed due to security concerns (the gem is provided for those who require the functionality).