CS142
Project 8 Section
ActiveRecord and Forms in Rails
Agenda

• Project 8 Overview
• ActiveRecord
  • CRUD
• Forms
  • Helpers
• Handling
• Uploading Files
Add a Login Page
Handle bad input
Update pics/user/id page
Add photo upload page
Add page for adding comments
Add a Comment

1 error prohibited this comment from being saved

There were problems with the following fields:

- Comment can't be blank

Handle bad input
Project 8 Overview

• Your layout doesn’t have to look exactly the same

• Be creative!
Project 8 Overview

• Questions?
ActiveRecord

- Object Relational Model (ORM)

```
SELECT * FROM students;
+----+-----------+------------+------+------+
| id | name      | birth      | gpa  | grad |
+----+-----------+------------+------+------+
|  1 | Anderson  | 1987-10-22 |  3.9 | 2009 |
|  2 | Jones     | 1990-04-16 |  2.4 | 2012 |
```

Relation \(\iff\) Object

```
class Student {
  int id;
  string name;
  Date birth;
  float gpa;
  int grad;
}
```

```
Student anderson = new Student();
anderson.setName(“Anderson”);
anderson.setGpa(3.9);
...```
ActiveRecord

- CRUD
  - Create
  - Read
  - Update
  - Destroy
ActiveRecord

- CRUD: Create

```ruby
student = Student.new
student.name = "Williams"
student.birth = "1989-11-16"
student.gpa = 2.8
student.grad = 2012
student.save()
```
ActiveRecord

- CRUD : Read

```ruby
student = Student.find(187)
student = Student.find_by_name("Hernandez")
student = Student.find_by_name_and_grad("Hernandez", 2012);
smarties = Student.find {:all,:conditions => "gpa >= 3.0"};
smarties = Student.find(:all, :limit => 10, :order => "gpa DESC");
```
ActiveRecord

• CRUD : Update

```ruby
student = Student.find(187)
student.gpa = 4.0
student.save()
```
ActiveRecord

- CRUD : Destroy

```ruby
student = Student.find(187)
student.destroy()
```
ActiveRecord

• Questions?
Forms

Controller

```ruby
#login id was saved in the session array
@student = Student.find(session["login_id"])
```

View

```html
<p>Update student:</p>
<% form_for(:student, :url => {:action => :modify, :id => @student.id}) do |form| %>
  Student name: <%= form.text_field(:name) %>
  <br />
  <%= submit_tag "Add student" %>
<% end %>
```
Forms

- Handling form submissions
  - Use redirect after post to avoid history issues

Controller

```ruby
def search
  @student = Student.find(params[:id])
  if @student.update_attributes(params[:student]) then
    redirect_to(:action => :show)
  else
    render(:action => :edit)
  end
end
```
Forms

• Other helper tags
  • check_box
  • file_field
  • hidden_field
  • password_field
  • radio_button
  • text_area
  • text_field
Forms

• Validation

  • Use built-in validators for checking input

Model

class Student < ActiveRecord::Base
  validates_length_of :name, :maximum=>30
  validates_presence_of :name, :birthday, :gpa
end
Forms

• File Upload

View

<% form_for(:video, :url => {:action => :videoPost},
   :html => { :multipart => true } ) do |form| %>
  <%= form.file_field :video_field %>
  <br />
  <%= submit_tag "Upload Video" %>
<% end %>
Forms

- File Upload

Controller

```ruby
# Assume a video model exists
def videoPost
  video_to_upload = params[:video][:video_field]
  video = Video.new
  video.save!
  video.file_name = "video" + video.id + File.extname(video_to_upload.original_filename())
  File.open("public/videos/" + video.file_name, "wb")
  do |f|
    f.write(video_to_upload.read())
  end
  video.save!
  redirect_to(:controller => :videos, :action => :list)
end
```
Forms

• Questions?