**ForestSat3**

**code**

# ======= ======= ======= application.html.erb ======= ======= =======

# ======= ======= ======= application.html.erb ======= ======= =======

# ======= ======= ======= application.html.erb ======= ======= =======

<!DOCTYPE html>

<html>

<head>

<title>||| ForestSat |||</title>

<%= csrf\_meta\_tags %>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

<%= stylesheet\_link\_tag 'application', media: 'all', 'data-turbolinks-track': 'reload' %>

<%= javascript\_include\_tag 'application', 'data-turbolinks-track': 'reload' %>

</head>

<!-- ======= ======= ======= body ======= ======= ======= -->

<body>

<div id="pageWrap">

<!-- ======= menu ======= -->

<div id="menuBox">

<div class="menuRow">

<div id="signinItems" class="menuGroup">

<p class="menuLabel">signin</p>

<%= link\_to "Home", home\_path, class: "menuItem" %>

<% if !current\_user %>

<%= link\_to "Sign up", new\_user\_registration\_path, class: "menuItem" %>

<%= link\_to "Sign in", new\_user\_session\_path, class: "menuItem" %>

<% else %>

<%= link\_to "Sign out", destroy\_user\_session\_path, :method => :delete, class: "menuItem" %>

<a href="#" class="menuItem filler"></a>

</div>

<div id="signinFiller" class="menuGroup">

<p class="menuLabel">current user</p>

<p id="userFullName" class="filler">

<%= current\_user[:firstname] + " " + current\_user[:lastname] %>

</p>

</div>

</div>

<div class="menuRow">

<div id="userItems" class="menuGroup">

<p class="menuLabel">users</p>

<%= link\_to "My Profile", user\_path(current\_user), class: "menuItem" %>

</div>

<div id="abstractItems" class="menuGroup">

<p class="menuLabel">abstracts</p>

<%= link\_to "My Abstracts", "/my\_abstracts", class: "menuItem" %>

<%= link\_to "All Abstracts", abstracts\_path, class: "menuItem" %>

<%= link\_to "New Abstract", new\_abstract\_path, class: "menuItem" %>

</div>

<div id="reviewItems" class="menuGroup">

<p class="menuLabel">reviewers</p>

<% if current\_user.reviewer? %>

<% if @reviewer\_abstracts.length > 0 %>

<%= link\_to "Review Abstracts", "/review\_abstracts", class: "menuItem" %>

<%= link\_to "Review Status", new\_presentation\_path, class: "menuItem" %>

<% else %>

<a href="#" class="menuItem filler"></a>

<a href="#" class="menuItem filler"></a>

<% end %>

<% else %>

<a href="#" class="menuItem filler"></a>

<a href="#" class="menuItem filler"></a>

<% end %>

</div>

</div>

<% if current\_user.admin? %>

<div class="menuRow">

<div id="authorItems" class="menuGroup">

<p class="menuLabel">authors</p>

<%= link\_to "All Authors", authors\_path, class: "menuItem" %>

</div>

<div id="adminItems" class="menuGroup">

<p class="menuLabel">select/assign reviewers</p>

<%= link\_to 'Select Reviewers', "/select\_reviewers", class: "menuItem" %>

<%= link\_to "Assign Reviewers", "/assign\_reviewers", class: "menuItem" %>

<%= link\_to "Select Abstracts", "/select\_abstracts", class: "menuItem" %>

</div>

<div id="sessionItems" class="menuGroup">

<p class="menuLabel">sessions</p>

<%= link\_to "List View", rooms\_path, class: "menuItem" %>

<%= link\_to "Chart View", new\_room\_path, class: "menuItem" %>

</div>

</div>

<% end %>

<% end %>

</div>

<!-- ======= messages ======= -->

<div id="messages">

<p id="notice"><%= notice %></p>

</div>

<!-- ======= yield ======= -->

<div id="yield" class="section" data-state="default">

<%= yield %>

</div>

</div>

</body>

</html>

# ======= ======= ======= users\_controller.rb ======= ======= =======

# ======= ======= ======= users\_controller.rb ======= ======= =======

# ======= ======= ======= users\_controller.rb ======= ======= =======

class UsersController < ApplicationController

before\_action :set\_user, only: [:show]

before\_action :check\_env\_vars, only: [:home]

# ======= home =======

def home

flash\_msg =""

@users = User.all

@rooms = Room.all

@authors = Author.all

@abstracts = Abstract.all

@affiliations = Affiliation.all

if @users == []

puts "+++++++ NO USERS +++++++"

flash\_msg = flash\_msg + " NO USERS "

end

if @rooms == []

puts "+++++++ NO ROOMS +++++++"

flash\_msg = flash\_msg + " NO ROOMS "

end

if @authors == []

puts "+++++++ NO AUTHORS +++++++"

flash\_msg = flash\_msg + " NO AUTHORS "

end

if @abstracts == []

puts "+++++++ NO ABSTRACTS +++++++"

flash\_msg = flash\_msg + " NO ABSTRACTS "

end

if @affiliations == []

puts "+++++++ NO AFFILIATIONS +++++++"

flash\_msg = flash\_msg + " NO AFFILIATIONS "

end

flash.now[:notice] = flash\_msg

end

# ======= index =======

# lists user admins only

def index

@users = User.where(:admin => true)

end

# ======= select\_reviewers =======

# admin screen for selecting reviewers

def select\_reviewers

@users = User.all.order("lastname")

end

# ======= save\_select =======

# save selected reviewers

def save\_select

current\_reviewer\_ids = User.where(:reviewer => true).pluck(:id)

former\_reviewer\_ids = []

# == change un-checked reviewer status to false

if params[:user]

ok\_params = reviewer\_status\_params

current\_reviewer\_ids.each do |reviewer\_id|

if !ok\_params[:reviewer\_ids].include? reviewer\_id.to\_s

former\_reviewer\_ids << reviewer\_id

former\_reviewer = User.find(reviewer\_id)

former\_reviewer.update(:reviewer => false)

end

end

# == change newly selected reviewer status to true

ok\_params[:reviewer\_ids].each do |reviewer\_id|

user = User.find(reviewer\_id)

user.update(:reviewer => true)

end

else

current\_reviewer\_ids.each do |reviewer\_id|

former\_reviewer\_ids << reviewer\_id

former\_reviewer = User.find(reviewer\_id)

former\_reviewer.update(:reviewer => false)

end

end

# == remove former reviewers from abstracts previously assigned

former\_reviewer\_ids.each do |reviewer\_id|

abstracts1 = Abstract.where(:reviewer1\_id => reviewer\_id)

abstracts2 = Abstract.where(:reviewer2\_id => reviewer\_id)

abstracts1.each do |abstract|

abstract.update(:reviewer1\_id => nil)

end

abstracts2.each do |abstract|

abstract.update(:reviewer2\_id => nil)

end

end

redirect\_to :select\_reviewers

end

# ======= assign\_reviewers =======

# admin screen for assigning reviewers to abstract

def assign\_reviewers

@reviewers = User.where(:reviewer => true).order("lastname")

@abstracts = Abstract.all.order("abs\_title")

end

# ======= save\_assign =======

# save assign for each abstract (single save disabled)

def save\_assign

ok\_params = reviewer\_params

# == save all abstract updates

ok\_params[:abstract\_ids].each\_with\_index do |abstract\_id, index|

reviewer1\_id = ok\_params[:reviewer1\_ids][index]

reviewer2\_id = ok\_params[:reviewer2\_ids][index]

@abstract = Abstract.find(abstract\_id)

@abstract.update(:reviewer1\_id => reviewer1\_id, :reviewer2\_id => reviewer2\_id)

end

flash[:notice] = 'All reviewer changes updated successfully.'

redirect\_to :assign\_reviewers

end

# ======= show =======

def show

puts "\n\*\*\*\*\*\*\* show \*\*\*\*\*\*\*"

end

# ======= check\_env\_vars =======

def check\_env\_vars

puts "\n\*\*\*\*\*\*\* check\_env\_vars \*\*\*\*\*\*\*"

puts "SENDMAIL\_USERNAME: #{SENDMAIL\_USERNAME}"

puts "SENDMAIL\_PASSWORD: #{SENDMAIL\_PASSWORD}"

puts "MAIL\_HOST: #{MAIL\_HOST}"

end

# ======= my\_abstracts =======

def my\_abstracts

# == get author record for current\_user

author = Author.where(:firstname => current\_user.firstname, :lastname => current\_user.lastname).first

# == get abstracts where current\_user is primary or secondary author

abstracts = author.abstracts

# == get abstracts where current\_user is corresponding or presenting author

corr\_abstracts = Abstract.where("corr\_author\_id = " + current\_user[:id].to\_s + " OR pres\_author\_id = " + current\_user[:id].to\_s)

# == merge both selections withour duplication

@abstracts = abstracts + corr\_abstracts

@title = "My Abstracts"

render :template => 'abstracts/index'

end

private

def set\_user

@user = User.find(params[:id])

end

def reviewer\_status\_params

params.require(:user).permit(:reviewer\_ids => [])

end

def reviewer\_params

params.require(:abstract).permit(:abstract\_ids => [], :reviewer1\_ids => [], :reviewer2\_ids => [], :abstract\_ids => [])

end

end

# ======= ======= ======= abstracts\_controller.rb ======= ======= =======

# ======= ======= ======= abstracts\_controller.rb ======= ======= =======

# ======= ======= ======= abstracts\_controller.rb ======= ======= =======

class AbstractsController < ApplicationController

before\_action :set\_abstract, only: [:show, :edit, :update, :destroy]

# ======= edit =======

def edit

@users = User.all

# == get author data from authors and abstract\_authors

sql = "SELECT authors.id, authors.firstname, authors.lastname, abstract\_authors.author\_type FROM authors INNER JOIN abstract\_authors ON authors.id = abstract\_authors.author\_id WHERE abstract\_authors.abstract\_id = '" + @abstract[:id].to\_s + "';"

@authors\_data = ActiveRecord::Base.connection.execute(sql)

@keywords\_1 = Keyword.where(:keyword\_category => 1)

@keywords\_2 = Keyword.where(:keyword\_category => 2)

@keywords\_3 = Keyword.where(:keyword\_category => 3)

end

# ======= update =======

def update

ok\_params = update\_params

respond\_to do |format|

if @abstract.update(ok\_params)

format.html { redirect\_to @abstract, notice: 'Abstract was successfully updated.' }

else

format.html { render :edit }

end

end

end

# ======= review\_abstracts =======

# == reviewer selects abstracts to review

def review\_abstracts

@users = User.all.order("lastname")

@abstracts = Abstract.where("reviewer1\_id = ? OR reviewer2\_id = ?", current\_user[:id], current\_user[:id]).order("abs\_title")

end

# ======= save\_all\_reviews =======

# == reviewer selects abstracts to review

def save\_all\_reviews

ok\_params = rec\_grade\_params

ok\_params[:abstract\_ids].each\_with\_index do |abstract\_id, index|

reviewer\_rec = ok\_params[:reviewer\_recs][index]

reviewer\_grade = ok\_params[:reviewer\_grades][index]

@abstract = Abstract.find(abstract\_id)

if ok\_params[:which\_reviewer] == "reviewer1"

@abstract.update(:reviewer1\_rec => reviewer\_rec, :reviewer1\_grade => reviewer\_grade)

else

@abstract.update(:reviewer2\_rec => reviewer\_rec, :reviewer2\_grade => reviewer\_grade)

end

end

flash[:notice] = 'All abstract changes updated successfully.'

redirect\_to :select\_abstracts

end

# ======= filter\_by\_keyword =======

# == any combination of three keywords can be used for filtering

def filter\_by\_keyword

ok\_params = filter\_params

query\_array = []

value\_array = []

keyword\_array = []

query\_string = ""

value\_string = ""

keyword\_string = ""

if params[:keyword\_1].length > 0

keyword\_1 = Keyword.select(:keyword\_name).find(params[:keyword\_1])

keyword\_array << keyword\_1

value\_array << params[:keyword\_1]

query\_array << "keyword\_1 = ?"

end

if params[:keyword\_2].length > 0

keyword\_2 = Keyword.select(:keyword\_name).find(params[:keyword\_2])

keyword\_array << keyword\_2

value\_array << params[:keyword\_2]

query\_array << "keyword\_2 = ?"

end

if params[:keyword\_3].length > 0

keyword\_3 = Keyword.select(:keyword\_name).find(params[:keyword\_3])

keyword\_array << keyword\_3

value\_array << params[:keyword\_3]

query\_array << "keyword\_3 = ?"

end

# == build query based on non-nil keyword values

if keyword\_array.length > 0

keyword\_array.each\_with\_index do |keyword, index|

keyword\_string += keyword[:keyword\_name]

query\_string += query\_array[index]

if index < (keyword\_array.length - 1)

keyword\_string += " OR "

query\_string += " OR "

end

end

end

value\_array = value\_array.unshift(query\_string)

@abstracts = Abstract.where(value\_array)

# == get current keyword lists by type

@users = User.all.order("lastname")

@keywords\_1 = Keyword.where(:keyword\_category => 1)

@keywords\_2 = Keyword.where(:keyword\_category => 2)

@keywords\_3 = Keyword.where(:keyword\_category => 3)

flash.now[:notice] = "Selection by keywords: " + keyword\_string

respond\_to do |format|

format.html { render :select\_abstracts }

end

end

# ======= select\_abstracts =======

# == create admin screen for selecting/rejecting abstracts

def select\_abstracts

@users = User.all.order("lastname")

@keywords\_1 = Keyword.where(:keyword\_category => 1)

@keywords\_2 = Keyword.where(:keyword\_category => 2)

@keywords\_3 = Keyword.where(:keyword\_category => 3)

@abstracts = Abstract.all.order("abs\_title")

end

# ======= save\_selection =======

# == save admin selection ()

def save\_selection

ok\_params = selection\_params

if ok\_params[:abstract\_id] == "all"

ok\_params[:abstract\_ids].each\_with\_index do |abstract\_id, index|

admin\_final = ok\_params[:admin\_finals][index]

@abstract = Abstract.find(abstract\_id)

@abstract.update(:admin\_final => admin\_final)

end

flash[:notice] = 'All abstract changes updated successfully.'

redirect\_to :select\_abstracts

else

flash[:notice] = 'Selected abstract status updated successfully.'

@abstract = Abstract.find(ok\_params[:abstract\_id])

@abstract.update(:admin\_final => ok\_params[:admin\_final])

redirect\_to :select\_abstracts

end

end

# ======= signup\_abstract =======

def signup\_abstract

@abstracts = Abstract.all

end

# ======= index =======

def index

@abstracts = Abstract.all

end

# ======= new =======

def new

@users = User.all.order("lastname")

@keywords1 = Keyword.where(:keyword\_category => "1")

@keywords2 = Keyword.where(:keyword\_category => "2")

@keywords3 = Keyword.where(:keyword\_category => "3")

@affiliations = Affiliation.all

@abstract\_author = AbstractAuthor.new

@affiliation = Affiliation.new

@abstract = Abstract.new

@author = Author.new

if current\_user[:admin] == true

@admin\_flag = true

else

@admin\_flag = false

end

end

# ======= create =======

def create

# == extract abstract/author/affiliation values from params

ok\_params = abstract\_params

abs\_params = ok\_params[0]

author\_params = ok\_params[1]

affiliation\_params = ok\_params[2]

author\_type\_params = ok\_params[3]

pres\_author\_index = abs\_params[:pres\_author\_id]

if abs\_params[:new\_keyword\_1] != ""

@keyword1 = Keyword.create(:keyword\_name => abs\_params[:new\_keyword\_1], :keyword\_category => "1")

if @keyword1

abs\_params[:keyword\_1] = @keyword1[:id]

else

flash.now[:notice] = "There was a problem with the new keyword\_1."

end

end

if abs\_params[:new\_keyword\_2] != ""

@keyword2 = Keyword.create(:keyword\_name => abs\_params[:new\_keyword\_2], :keyword\_category => "2")

if @keyword2

abs\_params[:keyword\_2] = @keyword2[:id]

else

flash.now[:notice] = "There was a problem with the new keyword\_2."

end

end

if abs\_params[:new\_keyword\_3] != ""

@keyword3 = Keyword.create(:keyword\_name => abs\_params[:new\_keyword\_3], :keyword\_category => "3")

if @keyword3

abs\_params[:keyword\_3] = @keyword3[:id]

else

flash.now[:notice] = "There was a problem with the new keyword\_3."

end

end

abs\_params = abs\_params.except(:new\_keyword\_1)

abs\_params = abs\_params.except(:new\_keyword\_2)

abs\_params = abs\_params.except(:new\_keyword\_3)

@abstract = Abstract.new(abs\_params)

# == save abstract

respond\_to do |format|

if @abstract.save

# == loop through author params arrays (firstnames/lastnames/author\_types)

author\_params[:firstnames].each\_with\_index do |firstname, index|

lastname = author\_params[:lastnames][index]

institution = affiliation\_params[:institutions][index]

department = affiliation\_params[:departments][index]

author\_type = author\_type\_params[:author\_types][index]

@author = Author.where(:firstname => firstname, :lastname => lastname).first

current\_user\_fname = current\_user[:firstname]

current\_user\_lname = current\_user[:lastname]

# == check if current\_user is one of the authors

if current\_user\_fname == firstname && current\_user\_lname == lastname

user\_id = current\_user[:id]

else

user\_id = nil

end

# == author not in database; create new author

if !@author

@author = Author.new(:firstname => firstname, :lastname => lastname, :user\_id => user\_id)

# == save author

if @author.save

@author = Author.order("created\_at").last

# == check if author is selected as presenting author

if index == pres\_author\_index.to\_i

@abstract.update(:pres\_author\_id => @author[:id])

@pres\_author = Author.find(@author[:id])

end

# == create join table abstract/author link with author\_type

@abstract\_author = AbstractAuthor.new(:abstract\_id => @abstract[:id].to\_i, :author\_id => @author[:id].to\_i, :author\_type => author\_type)

if @abstract\_author.save

puts "++++++ abstract\_author NEW OK +++++++"

else

flash[:notice] = 'Join table save failed. Please try again.'

end

# == save affiliation

@affiliation = Affiliation.where(:institution => institution, :department => department).first

if !@affiliation

@affiliation = Affiliation.new(:author\_id => @author[:id], :institution => institution, :department => department)

if @affiliation.save

puts "++++++ affiliation NEW OK +++++++"

else

flash[:notice] = 'Affiliation save failed. Please try again.'

end

end

end

# == author already in database; create join table abstract/author link with author\_type

else

if index == pres\_author\_index.to\_i

@abstract.update(:pres\_author\_id => @author[:id])

@pres\_author = Author.find(@author[:id])

end

@abstract\_author = AbstractAuthor.new(:abstract\_id => @abstract[:id].to\_i, :author\_id => @author[:id].to\_i, :author\_type => author\_type)

if @abstract\_author.save

puts "++++++ abstract\_author NEW OK +++++++"

else

flash[:notice] = 'Join table save failed. Please try again.'

end

end

end

abstract\_authors = @abstract.authors

format.html { redirect\_to @abstract, notice: 'Abstract/author/join all successfully created.' }

else

@users = User.all.order("lastname")

@affiliations = Affiliation.all

@affiliation = Affiliation.new

@abstract = Abstract.new

@author = Author.new

flash[:notice] = 'There was a problem saving your abstract. Please try again.'

format.html { render :new }

end

end

end

# ======= show =======

def show

@edit\_flag = false

@authors\_array = @abstract.authors.pluck(:id)

if @authors\_array.include? current\_user[:id]

puts "\n+++++++ author included +++++++"

end

if current\_user[:admin] == true

@edit\_flag = true

end

# == get author data from authors and abstract\_authors

sql = "SELECT authors.firstname, authors.lastname, abstract\_authors.author\_type FROM authors INNER JOIN abstract\_authors ON authors.id = abstract\_authors.author\_id WHERE abstract\_authors.abstract\_id = '" + @abstract[:id].to\_s + "';"

@authors\_data = ActiveRecord::Base.connection.execute(sql)

corr\_author = User.find(@abstract[:corr\_author\_id])

@corr\_author\_name = corr\_author[:firstname] + " " + corr\_author[:lastname]

pres\_author = Author.find(@abstract[:pres\_author\_id])

@pres\_author\_name = pres\_author[:firstname] + " " + pres\_author[:lastname]

if @abstract[:reviewer1\_id]

reviewer1 = User.find(@abstract[:reviewer1\_id])

@reviewer1\_name = reviewer1[:firstname] + " " + reviewer1[:lastname]

else

@reviewer1\_name = "unassigned"

end

if @abstract[:reviewer1\_id]

reviewer2 = User.find(@abstract[:reviewer2\_id])

@reviewer2\_name = reviewer2[:firstname] + " " + reviewer2[:lastname]

else

@reviewer2\_name = "unassigned"

end

if @abstract[:present\_id]

present = Presentation.find(@abstract[:present\_id])

@present = present[:session\_title]

else

@present = "unassigned"

end

end

# ======= destroy =======

def destroy

@abstract.destroy

respond\_to do |format|

format.html { redirect\_to abstracts\_url, notice: 'Abstract was successfully destroyed.' }

end

end

private

def set\_abstract

@abstract = Abstract.find(params[:id])

end

def update\_params

params.require(:abstract).permit(:id, :abs\_title, :abs\_text, :corr\_author\_id, :pres\_author\_id, :present\_request, :keyword\_1, :keyword\_2, :keyword\_3)

end

def filter\_params

params.permit(:keyword\_1, :keyword\_2, :keyword\_3)

end

def rec\_grade\_params

params.require(:abstract).permit(:which\_reviewer, :abstract\_ids => [], :reviewer\_recs => [], :reviewer\_grades => [])

end

def selection\_params

params.require(:abstract).permit(:abstract\_id, :admin\_final, :admin\_finals => [], :abstract\_ids => [])

end

def abstract\_params

ok\_params = []

ok\_params << params.require(:abstract).permit(:corr\_author\_id, :pres\_author\_id, :reviewer1\_id, :reviewer2\_id, :presentation\_id, :session\_sequence, :abs\_title, :abs\_text, :keyword\_1, :keyword\_2, :keyword\_3, :new\_keyword\_1, :new\_keyword\_2, :new\_keyword\_3, :present\_request, :present\_assigned, :reviewer1\_rec, :reviewer2\_rec, :accepted, :invited, :paper)

ok\_params << params.require(:author).permit(:firstnames => [], :lastnames => [])

ok\_params << params.require(:affiliation).permit(:institutions => [], :departments => [])

ok\_params << params.require(:abstract\_author).permit(:author\_types => [])

return ok\_params

end

end

# corr\_author\_id, pres\_author\_id, reviewer1\_id, reviewer2\_id, presentation\_id, session\_sequence, abs\_title, abs\_text, keyword\_1, keyword\_2, keyword\_3, present\_request, present\_assigned, reviewer1\_rec, reviewer2\_rec, accepted, invited, paper

# ======= ======= ======= users/select\_reviewers.html.erb ======= ======= =======

# ======= ======= ======= users/select\_reviewers.html.erb ======= ======= =======

# ======= ======= ======= users/select\_reviewers.html.erb ======= ======= =======

<h2>Select Reviewers</h2>

<%= form\_for("save\_select", url: "/save\_select", :html => {:id => "save\_select"}) do %>

<table>

<thead>

<tr>

<th>name</th>

<th>affiliation</th>

<th>reviewer</th>

</tr>

</thead>

<tbody>

<% @users.each do |user| %>

<% author = Author.where(:user\_id => user[:id]).first %>

<% affiliation = author.affiliation %>

<% puts "affiliation: #{affiliation.inspect}" %>

<tr>

<% fullname = user[:firstname].to\_s + " " + user[:lastname].to\_s %>

<td class="main-col"><%= link\_to fullname, user %></td>

<td><%= affiliation[:institution] + ", " + affiliation[:department] %></td>

<td></td>

<% if current\_user.admin? %>

<% if user.reviewer? %>

<td>

<%= check\_box\_tag "user[reviewer\_ids][]", user[:id], :checked %>

</td>

<% else %>

<td>

<%= check\_box\_tag "user[reviewer\_ids][]", user[:id] %>

</td>

<% end %>

<% else %>

<% if user.reviewer? %>

<td>reviewer</td>

<% else %>

<td></td>

<% end %>

<% end %>

</tr>

<% end %>

<tr>

<td colspan="2"></td>

<td><%= button\_tag "submit" %></td>

</tr>

</tbody>

</table>

<% end %>

# ======= ======= ======= users/assign\_reviewers.html.erb ======= ======= =======

# ======= ======= ======= users/assign\_reviewers.html.erb ======= ======= =======

# ======= ======= ======= users/assign\_reviewers.html.erb ======= ======= =======

<h1>Assign Reviewers</h1>

<!-- ======= reviewers selectbox (users filtered for reviewer => true) ======= -->

<% reviewer\_name\_ids = [nil] %>

<% @reviewers.each\_with\_index do |reviewer, index| %>

<% fullname = reviewer[:firstname] + " " + reviewer[:lastname] %>

<% reviewer\_name\_ids << [fullname, reviewer[:id]] %>

<% end %>

<%= form\_for("save\_assign", url: "/save\_assign", :html => {:id => "save\_assign"}) do %>

<%= fields\_for :abstract do |a| %>

<table class="abstractAdmin">

<tr>

<th>title</th>

<th>primary</th>

<th>reviewer1</th>

<th>reviewer2</th>

<th></th>

<th>last update</th>

</tr>

<!-- ======= list abstracts/reviewers ======= -->

<% @abstracts.each do |next\_abstract| %>

<!-- ======= reviewer initialize ======= -->

<% selected\_reviewer\_id1 = nil %>

<% selected\_reviewer\_id2 = nil %>

<% if next\_abstract[:reviewer1\_id] %>

<% selected\_reviewer\_id1 = next\_abstract[:reviewer1\_id] %>

<% end %>

<% if next\_abstract[:reviewer2\_id] %>

<% selected\_reviewer\_id2 = next\_abstract[:reviewer2\_id] %>

<% end %>

<% sql = "SELECT authors.firstname, authors.lastname, abstract\_authors.author\_type FROM authors INNER JOIN abstract\_authors ON authors.id = abstract\_authors.author\_id WHERE abstract\_authors.abstract\_id = '" + next\_abstract[:id].to\_s + "' AND abstract\_authors.author\_type = 'primary';" %>

<% author\_data = ActiveRecord::Base.connection.execute(sql) %>

<% primary\_author = author\_data[0]["firstname"].to\_s + " " + author\_data[0]["lastname"].to\_s %>

<tr>

<td><%= link\_to next\_abstract[:abs\_title], abstract\_path(next\_abstract) %></td>

<td><%= primary\_author %></td>

<td><%= select\_tag("abstract[reviewer1\_ids][]", options\_for\_select(reviewer\_name\_ids, selected\_reviewer\_id1), :id => "abstract\_reviewer1\_ids\_" + next\_abstract[:id].to\_s) %></td>

<td><%= select\_tag("abstract[reviewer2\_ids][]", options\_for\_select(reviewer\_name\_ids, selected\_reviewer\_id2), :id => "abstract\_reviewer2\_ids\_" + next\_abstract[:id].to\_s) %></td>

<td>

<%= hidden\_field\_tag "abstract[abstract\_ids][]", next\_abstract[:id].to\_s %>

</td>

<% formatted\_date = next\_abstract[:updated\_at].in\_time\_zone("Eastern Time (US & Canada)").strftime('%a %b %d %l:%M') %>

<td>updated: <%= formatted\_date %> </td>

</tr>

<% end %>

<!-- ======= save all changes option ======= -->

<tr>

<td colspan="3"></td>

<td colspan="3">

<button type="submit">Save all changes</button>

</td>

</tr>

</table>

<% end %>

<% end %>

# ======= ======= ======= abstracts/new.html.erb ======= ======= =======

# ======= ======= ======= abstracts/new.html.erb ======= ======= =======

# ======= ======= ======= abstracts/new.html.erb ======= ======= =======

<%= form\_for(@abstract, url: "/signup\_abstract") do |f| %>

<!-- ======= reviewer initialize ======= -->

<% selected\_keyword1 = nil %>

<% selected\_keyword2 = nil %>

<% selected\_keyword3 = nil %>

<% if @abstract[:keyword1\_id] %>

<% selected\_keyword1 = @abstract[:keyword1\_id] %>

<% end %>

<% if @abstract[:keyword2\_id] %>

<% selected\_keyword2 = @abstract[:keyword2\_id] %>

<% end %>

<% if @abstract[:keyword3\_id] %>

<% selected\_keyword3 = @abstract[:keyword3\_id] %>

<% end %>

<!-- ======= keywords selectboxes ======= -->

<% keywords1 = [nil] %>

<% @keywords1.each do |keyword1| %>

<% keywords1 << [keyword1[:keyword\_name], keyword1[:id]] %>

<% end %>

<% keywords2 = [nil] %>

<% @keywords2.each do |keyword2| %>

<% keywords2 << [keyword2[:keyword\_name], keyword2[:id]] %>

<% end %>

<% keywords3 = [nil] %>

<% @keywords3.each do |keyword3| %>

<% keywords3 << [keyword3[:keyword\_name], keyword3[:id]] %>

<% end %>

<% puts "keywords3: #{keywords3.inspect}" %>

<!-- ======= reviewer initialize ======= -->

<% selected\_reviewer1 = nil %>

<% selected\_reviewer2 = nil %>

<% if @abstract[:reviewer1\_id] %>

<% selected\_reviewer1 = @abstract[:reviewer1\_id] %>

<% end %>

<% if @abstract[:reviewer2\_id] %>

<% selected\_reviewer2 = @abstract[:reviewer2\_id] %>

<% end %>

<!-- ======= users selectbox ======= -->

<% user\_name\_ids = [] %>

<% @users.each\_with\_index do |user, index| %>

<% fullname = user[:firstname] + " " + user[:lastname] %>

<% user\_name\_ids << [fullname, user[:id]] %>

<% end %>

<!-- ======= entry screens ======= -->

<div class="abstractEntryBox">

<!-- ======= abstract ======= -->

<%= fields\_for :abstract do |abstract| %>

<div class="abstractRow\_1 toggle\_input">

<p class="abstractLabel\_1">abstract title</p>

<%= abstract.text\_field :abs\_title, placeholder: "Abstract Title", class: "title" %>

</div>

<div class="abstractRow\_2 toggle\_input">

<p class="abstractLabel\_2">abstract text</p>

<%= abstract.text\_area :abs\_text, placeholder: "Abstract Text", class: "text", cols: "30", rows: "6" %>

</div>

<!-- ======= buttons ======= -->

<div class="abstractRow\_3 toggle\_input">

<button id="reviewBtn" class="abstractBtn" type="button">reviewers</button>

<button id="nextBtn\_1" class="abstractBtn" type="button">next</button>

</div>

<%= abstract.hidden\_field :corr\_author\_id, value: current\_user[:id].to\_s %>

<%= hidden\_field\_tag "admin\_flag", @admin\_flag.to\_s %>

<% end %>

<!-- ======= authors ======= -->

<div class="authorRow\_1 toggle\_input">

<div id="primary" class="authorEntryGrp">

<p class="authorLabel\_1">primary author</p>

<%= fields\_for :author do |author| %>

<%= text\_field\_tag "author[firstnames][]", "Bill", id: "primary\_firstname", class: "authorName entry firstname" %>

<%= text\_field\_tag "author[lastnames][]", "Clinton", id: "primary\_lastname", class: "authorName entry lastname" %>

<% end %>

<%= fields\_for :abstract\_author do |abstract\_author| %>

<%= hidden\_field\_tag "abstract\_author[author\_types][]", "primary", class: "author author\_type" %>

<% end %>

<div class="radio">

<%= fields\_for :abstract do |abstract| %>

<span><%= abstract.radio\_button :pres\_author\_id, 0, class: "presenting" %></span>

<% end %>

<span>presenting</span>

</div>

<%= fields\_for :affiliation do |affiliation| %>

<%= text\_field\_tag "affiliation[institutions][]", "Yale", id: "primary\_institution", class: "affiliation entry institution" %>

<%= text\_field\_tag "affiliation[departments][]", "Law", id: "primary\_department", class: "affiliation entry department" %>

<% end %>

</div>

<p class="authorLabel\_2">secondary authors</p>

</div>

<div id="authorRow\_2" class="authorRow\_2 toggle\_input">

<div id="secondary\_" class="authorEntryGrp">

<%= fields\_for :author do |author| %>

<%= text\_field\_tag "author[firstnames][]", "George", id: "firstname\_1", class: "authorName entry firstname" %>

<%= text\_field\_tag "author[lastnames][]", "Bush", id: "lastname\_1", class: "authorName entry lastname" %>

<% end %>

<%= fields\_for :abstract\_author do |abstract\_author| %>

<%= hidden\_field\_tag "abstract\_author[author\_types][]", "secondary", class: "author author\_type" %>

<% end %>

<div class="radio">

<%= fields\_for :abstract do |abstract| %>

<span><%= abstract.radio\_button :pres\_author\_id, 1, class: "presenting" %></span>

<% end %>

<span>presenting</span>

</div>

<%= fields\_for :affiliation do |affiliation| %>

<%= text\_field\_tag "affiliation[institutions][]", "Texas Tech", id: "primary\_institution", class: " affiliation entry institution" %>

<%= text\_field\_tag "affiliation[departments][]", "Fratboyism", id: "primary\_department", class: " affiliation entry department" %>

<% end %>

</div>

</div>

<!-- ======= buttons ======= -->

<div class="authorRow\_3 toggle\_input">

<button id="prevBtn\_2" class="abstractBtn" type="button">prev</button>

<button id="authorBtn" class="abstractBtn" type="button">add author</button>

<button id="nextBtn\_2" class="abstractBtn" type="button">next</button>

</div>

<!-- ======= keywords ======= -->

<%= fields\_for :abstract do |abstract| %>

<div class="keyRow\_1 toggle\_input">

<div id="primary" class="keyGrp\_1">

<p class="keyLabel\_1">

<span class="keywordLabel">keyword 1</span>

<span class="keywordLabel">keyword 2</span>

<span class="keywordLabel">keyword 3</span>

</p>

</div>

<p class="reviewerInfo">select 0 to 3 keywords</p>

<%= abstract.select(:keyword\_1, options\_for\_select(keywords1, selected\_keyword1), {}, { :class => "keyword entry" }) %>

<%= abstract.select(:keyword\_2, options\_for\_select(keywords2, selected\_keyword2), {}, { :class => "keyword entry" }) %>

<%= abstract.select(:keyword\_3, options\_for\_select(keywords3, selected\_keyword3), {}, { :class => "keyword entry" }) %>

</div>

<div class="keyRow\_2 toggle\_input">

<p class="reviewerInfo adminInfo">enter new keywords here</p>

<%= text\_field\_tag "abstract[new\_keyword\_1]", "", id: "new\_keyword\_1", class: "keyword entry" %>

<%= text\_field\_tag "abstract[new\_keyword\_2]", "", id: "new\_keyword\_2", class: "keyword entry" %>

<%= text\_field\_tag "abstract[new\_keyword\_3]", "", id: "new\_keyword\_3", class: "keyword entry" %>

</div>

<!-- ======= buttons ======= -->

<div class="keyRow\_3 toggle\_input">

<button id="prevBtn\_3" class="abstractBtn" type="button">prev</button>

<button id="nextBtn\_3" class="abstractBtn" type="button">next</button>

</div>

<% end %>

</div>

<% end %>

# ======= ======= ======= abstracts/review\_abstracts.html.erb ======= ======= =======

# ======= ======= ======= abstracts/review\_abstracts.html.erb ======= ======= =======

# ======= ======= ======= abstracts/review\_abstracts.html.erb ======= ======= =======

<% if @title %>

<h1><%= @title %></h1>

<% else %>

<h1>Review Abstract</h1>

<% end %>

<% grade\_select\_options = [nil, "A", "B", "C", "D", "F"] %>

<% rec\_select\_options = [nil, "oral", "poster", "rejected"] %>

<%= form\_for("save\_all\_reviews", url: "/save\_all\_reviews", :html => {:id => "save\_all\_reviews"}) do |f| %>

<table>

<thead>

<tr>

<th>title</th>

<th>corresponding author</th>

<th>presenting author</th>

<th>primary author</th>

<th>recommendation</th>

<th>grade</th>

</tr>

</thead>

<tbody>

<% whichReviewer = nil %>

<!-- ======= loop through all abstracts ======= -->

<% @abstracts.each do |abstract| %>

<% puts "abstract[:reviewer1\_rec]: #{abstract[:reviewer1\_rec].inspect}" %>

<% puts "abstract[:reviewer1\_grade]: #{abstract[:reviewer1\_grade].inspect}" %>

<!-- == get primary author -->

<% prim\_author\_id = nil %>

<% prim\_author\_name = nil %>

<% sql = "SELECT authors.id, authors.firstname, authors.lastname, abstract\_authors.author\_type FROM authors INNER JOIN abstract\_authors ON authors.id = abstract\_authors.author\_id WHERE abstract\_authors.abstract\_id = '" + abstract[:id].to\_s + "' AND abstract\_authors.author\_type = 'primary';" %>

<% author\_data = ActiveRecord::Base.connection.execute(sql) %>

<% author\_data.each do |author| %>

<% prim\_author\_name = author["firstname"] + " " + author["lastname"] %>

<% prim\_author\_id = author["id"] %>

<% end %>

<!-- ======= loop through reviewer abstracts ======= -->

<% if User.exists?(abstract[:corr\_author\_id]) %>

<!-- == corresponding/presenting authors == -->

<% corr\_author = User.find(abstract[:corr\_author\_id]) %>

<% corr\_author\_name = corr\_author[:firstname].to\_s + " " + corr\_author[:lastname].to\_s %>

<% pres\_author = Author.find(abstract[:pres\_author\_id]) %>

<% pres\_author\_name = pres\_author[:firstname].to\_s + " " + pres\_author[:lastname].to\_s %>

<!-- ======= abstract row ======= -->

<tr>

<td class="main-col"><%= link\_to abstract[:abs\_title], abstract\_path(abstract) %></td>

<td><%= link\_to corr\_author\_name, user\_path(abstract[:corr\_author\_id]) %></td>

<td><%= link\_to pres\_author\_name, author\_path(abstract[:pres\_author\_id]) %></td>

<td><%= link\_to prim\_author\_name, author\_path(prim\_author\_id.to\_s) %></td>

<!-- == recommendations and grades -->

<% if current\_user[:id] == abstract[:reviewer1\_id] %>

<% whichReviewer = "reviewer1" %>

<td><%= select\_tag("abstract[reviewer\_recs][]", options\_for\_select(rec\_select\_options, abstract[:reviewer1\_rec]), :id => "abstract\_reviewer\_rec\_" + abstract[:id].to\_s) %></td>

<td><%= select\_tag("abstract[reviewer\_grades][]", options\_for\_select(grade\_select\_options, abstract[:reviewer1\_grade]), :id => "abstract\_reviewer\_grade\_" + abstract[:id].to\_s) %></td>

<% else %>

<% whichReviewer = "reviewer2" %>

<td><%= select\_tag("abstract[reviewer\_recs][]", options\_for\_select(rec\_select\_options, abstract[:reviewer2\_rec]), :id => "abstract\_reviewer\_rec\_" + abstract[:id].to\_s) %></td>

<td><%= select\_tag("abstract[reviewer\_grades][]", options\_for\_select(grade\_select\_options, abstract[:reviewer2\_grade]), :id => "abstract\_reviewer\_grade\_" + abstract[:id].to\_s) %></td>

<% end %>

<%= hidden\_field\_tag "abstract[abstract\_ids][]", abstract[:id], :id => "abstract\_abstract\_ids\_" + abstract[:id].to\_s %>

</tr>

<% end %>

<% end %>

<!-- ======= save all changes option ======= -->

<tr>

<td colspan="3"></td>

<%= hidden\_field\_tag "abstract[which\_reviewer]", whichReviewer %>

<% puts "whichReviewer: #{whichReviewer}" %>

<td colspan="3"><%= f.button "Save all changes", type: "submit" %></td>

</tr>

</tbody>

</table>

<% end %>

# ======= ======= ======= abstracts/select\_abstracts.html.erb ======= ======= =======

# ======= ======= ======= abstracts/select\_abstracts.html.erb ======= ======= =======

# ======= ======= ======= abstracts/select\_abstracts.html.erb ======= ======= =======

<h1>Select Abstracts</h1>

<!-- ======= users selectbox ======= -->

<% user\_name\_ids = [] %>

<% @users.each\_with\_index do |user, index| %>

<% fullname = user[:firstname] + " " + user[:lastname] %>

<% user\_name\_ids << [fullname, user[:id]] %>

<% end %>

<% keywords1 = [nil] %>

<% @keywords\_1.each\_with\_index do |keyword, index| %>

<% keywords1 << [keyword[:keyword\_name], keyword[:id]] %>

<% end %>

<% keywords2 = [nil] %>

<% @keywords\_2.each\_with\_index do |keyword, index| %>

<% keywords2 << [keyword[:keyword\_name], keyword[:id]] %>

<% end %>

<% keywords3 = [nil] %>

<% @keywords\_3.each\_with\_index do |keyword, index| %>

<% keywords3 << [keyword[:keyword\_name], keyword[:id]] %>

<% end %>

<!-- ======= admin\_select selectbox ======= -->

<% admin\_select\_options = [nil, "oral", "poster", "rejected"] %>

<!-- ======= filter by keyword ======= -->

<%= form\_for("filter\_by\_keyword", url: "/filter\_by\_keyword", :html => {:id => "filter\_by\_keyword"}) do %>

<div id="filters">

<%= select\_tag("keyword\_1", options\_for\_select(keywords1), :id => "keywords1") %>

<%= select\_tag("keyword\_2", options\_for\_select(keywords2), :id => "keywords2") %>

<%= select\_tag("keyword\_3", options\_for\_select(keywords3), :id => "keywords3") %>

<%= button\_tag "Filter by Keyword", type: "submit", id: "filter\_selections" %>

<%= button\_tag "Get all abstracts", type: "button", id: "all\_selections" %>

</div>

<% end %>

<%= form\_for("save\_selection", url: "/save\_selection", :html => {:id => "save\_selection"}) do %>

<!-- == note: form submitted via javascript -->

<%= fields\_for :abstract do |a| %>

<!-- == for one-at-a-time abstract save (value via javascript) -->

<%= a.hidden\_field "abstract\_id" %>

<%= a.hidden\_field "admin\_final" %>

<table class="abstractAdmin">

<tr>

<th>title</th>

<th>primary</th>

<th>reviewers</th>

<th>recommendations</th>

<th>grades</th>

<th>final</th>

<th></th>

</tr>

<!-- ======= list abstracts/reviewers ======= -->

<% @abstracts.each do |next\_abstract| %>

<!-- == get reviewers via abstracgt reviewer ids== -->

<% if next\_abstract[:reviewer1\_id] %>

<% selected\_reviewer1 = User.find(next\_abstract[:reviewer1\_id]) %>

<% selected\_reviewer\_id1 = selected\_reviewer1[:id] %>

<% reviewer1 = User.find(next\_abstract[:reviewer1\_id]) %>

<% reviewer\_name1 = reviewer1[:firstname] + " " + reviewer1[:lastname]%>

<% end %>

<% if next\_abstract[:reviewer2\_id] %>

<% selected\_reviewer2 = User.find(next\_abstract[:reviewer2\_id]) %>

<% selected\_reviewer\_id2 = selected\_reviewer2[:id] %>

<% reviewer2 = User.find(next\_abstract[:reviewer2\_id]) %>

<% reviewer\_name2 = reviewer2[:firstname] + " " + reviewer2[:lastname]%>

<% end %>

<!-- == get abstract authors via join table == -->

<% sql = "SELECT authors.firstname, authors.lastname, abstract\_authors.author\_type FROM authors INNER JOIN abstract\_authors ON authors.id = abstract\_authors.author\_id WHERE abstract\_authors.abstract\_id = '" + next\_abstract[:id].to\_s + "' AND abstract\_authors.author\_type = 'primary';" %>

<% author\_data = ActiveRecord::Base.connection.execute(sql) %>

<% primary\_author = author\_data[0]["firstname"].to\_s + " " + author\_data[0]["lastname"].to\_s %>

<!-- ======= abstract identity/reviewer1 rec/grade ======= -->

<tr>

<td><%= link\_to next\_abstract[:abs\_title], abstract\_path(next\_abstract) %></td>

<td><%= primary\_author %></td>

<td><%= reviewer\_name1 %></td>

<td><%= next\_abstract[:reviewer1\_rec] %></td>

<td><%= next\_abstract[:reviewer1\_grade] %></td>

<td><%= select\_tag("abstract[admin\_finals][]", options\_for\_select(admin\_select\_options, next\_abstract[:admin\_final]), :id => "abstract\_admin\_finals\_" + next\_abstract[:id].to\_s) %></td>

<td>

<%= hidden\_field\_tag "abstract[abstract\_ids][]", next\_abstract[:id].to\_s %>

</td>

<% formatted\_date = next\_abstract[:updated\_at].in\_time\_zone("Eastern Time (US & Canada)").strftime('%a %b %d %l:%M') %>

<td>updated: <%= formatted\_date %> </td>

</tr>

<!-- ======= reviewer2 rec/grade ======= -->

<tr>

<td></td>

<td></td>

<td><%= reviewer\_name2 %></td>

<td><%= next\_abstract[:reviewer2\_rec] %></td>

<td><%= next\_abstract[:reviewer2\_grade] %></td>

<td></td>

</tr>

<tr>

<td colspan="6"></td>

</tr>

<% end %>

<!-- ======= save all changes option ======= -->

<tr>

<td colspan="3"></td>

<td colspan="3"><%= a.button "Save all changes", type: "button", id: "save\_all\_selections" %></td>

</tr>

</table>

<% end %>

<% end %>