# Finals Guide

1. Create a new app; gitignore; commit

* cp ../helps/.gitignore .

1. Create vendor scaffold; migrate; adjust index and edit; seeds.1.rb; faker; remove new vendor link on index; commit

* rails g scaffold vendor name:string phone:string email:string website:string address:string description:string content:text
* cp ../helps/seeds.1.rb db/seeds.rb

1. Add search by keywords feature; controller; model; partial view; add highlights; commit

* <form method="get">  
   <input name="keywords" value="<%= params[:keywords] %>" placeholder="keywo  
   <button>Search</button>  
  </form>

1. Create school resource; nest routes; root route; add school\_id to vendors; seeds.2.rb; adjust school index; redirect on show; adjust vendor controller to load @school and load vendors from @school; adjust vendor index/show/edit links; commit

* rails g scaffold school name:string
* rails g migration add\_school\_id\_to\_vendor school\_id:integer:index
* cp ../helps/seeds.2.rb db/seeds.rb
* redirect\_to school\_vendors\_path(@school)
* before\_action :set\_school, only: [:index]
* @school = School.find(params[:school\_id]) if params[:school\_id]
* @vendors = @school.vendors
* <%= link\_to 'Back', school\_vendors\_path(@school) %>
* <%= link\_to 'Vendors', school\_vendors\_path(@vendor.school) %>
* <%= link\_to 'School', schools\_path %>

1. Add comment feature via scaffold; add association to vendor and comment; add index and form to vendor show; redirect create comment to vendor; hide vendor\_id on form; commit

* rails g scaffold comment like:boolean content:string vendor\_id:integer:index
* <% @vendor.comments.each do |comment| %>
* <%= render "comments/form", comment: @vendor.comments.new %>

1. Style page to add water.css; remove scaffold.css.scss; commit

* <link rel="stylesheet" href="https://cdn.jsdelivr.net/gh/kognise/water.css@latest/dist/light.min.css">

1. Clean up controllers; remove unnecessary actions; remove unnecessary formats; commit
2. Add pagination via pagy gem; include Pagy::Backend to vendors controller and Pagy::Frontend to application helper; load @pagy on vendors index action; set items to 5; update vendors index view; commit

* @pagy, @vendors = pagy(@vendors, items: 5)
* <%== pagy\_info @pagy %>  
  <%== pagy\_nav @pagy %>

1. Add geosearch via geocoder; add config; add redis gem; update geocoder initializer with redis cache; add long lat to vendor model; add yaml\_db gem; load data with long lat; update vendors controller with geo search via near; add location field on search box; commit

* rails generate geocoder:config
* cache: Redis.new,
* rails g migration add\_long\_lat\_to\_vendors longitude:float latitude:float
* rm db/seeds.rb
* rake db:reset
* cp ../helps/data.yml db/data.yml
* rake db:data:load
* <input name="location" value="<%= params[:location] %>" placeholder="locat
* @vendors = @vendors.near(params[:location], 100) if params[:location].present?

1. Add authentication via devise; setup config; update development.rb; generate user; add login links on top of page via application.html with a partial in app/views/application dir; separate flash messages into a partial in same dir; remove doubled flash messages from scaffold views; commit

* rails generate devise:install
* config.action\_mailer.default\_url\_options = { host: 'localhost', port: 3000 }
* rails generate devise user
* <%= render "messages" %>
* <%= render "login" %>
* <% if user\_signed\_in? %>  
   <%= link\_to "Sign out", destroy\_user\_session\_path, method: :delete %>  
   <%= current\_user.email %>  
  else %>  
   <%= link\_to "Home", root\_path %> |  
   <%= link\_to "Sign in", new\_user\_session\_path %>  
   <%= link\_to "Sign up", new\_user\_registration\_path %>  
  <% end %>
* <% flash.each do |type, msg| %>  
   <p><%= msg %></p>  
  <% end %>

1. Add claim vendor feature; add user\_id to vendor and add association on model; add claim action on vendor controller; add claim action on routes; add claim button; commit

* rails g migration add\_user\_id\_to\_vendor user\_id:integer:index
* def claim  
   if @vendor.update(user: current\_user)  
   redirect\_to @vendor, notice: 'Vendor was successfully claimed.'  
   else  
   render :show  
   end  
  end
* resources :vendors do  
   get :claim, on: :member  
  end
* <%= link\_to 'Claim', claim\_vendor\_path(@vendor) if user\_signed\_in? %> |

1. Add authorization via cancancan; generate ability class; add permission rules to allow reading everything by everyone except vendor edit–only allowed by user who owns vendor; update vendors controller to use load\_and\_authorize\_resource; commit

* rails g cancan:ability
* rescue\_from CanCan::AccessDenied do |exception|  
   redirect\_to root\_url, :alert => exception.message  
  end

1. Add icons via fontawesome; update vendor likes to icons; commit

* <script src="https://kit.fontawesome.com/0d41050561.js"></script>
* <% if comment.like %>  
   <i class="fa fa-heart" style="color:red;"></i>  
  <% else %>  
   <i class="fa fa-heart-broken" style="color:#ccc;"></i>  
  <% end %>

1. Deploy to Heroku; run data load

* git push heroku master
* heroku run rake db:reset
* heroku run rake db:data:load