

Recy Buddies

controllers

app/controllers/application_controller.rb

- Created a class called ApplicationController which inherits from ActionController::Base.
- Defined a before_action callback which calls authenticate_user! method before every action, except index.
- Included Pundit module in the ApplicationController class.
- Defined an after_action callback which calls verify_authorized method after every action, except index, unless the skip_pundit? method returns true.
- Defined another after_action callback which calls verify_policy_scoped method only after the index action, unless skip_pundit? returns true.
- Defined a private method called skip_pundit? which returns true if the current controller is a devise_controller or if the controller's name matches certain regex patterns.

app/controllers/packages_controller.rb

- index method retrieves all Package records and sets them to the @packages instance variable.
- show method authorizes the @package instance variable before rendering the view.
- new method initializes a new Package object and authorizes it before rendering the view.
- create method creates a new Package object with the permitted parameters from the params hash. It then sets the current user as the package's owner and authorizes it before saving it to the database. If the save is successful, it redirects to the show view of the newly created package with a success message. Otherwise, it renders the new view again.
- destroy method authorizes and destroys the @package instance variable, then redirects to the index view with a success message.
- list_my_picked_up_packages method retrieves all Package records and sets them to the @packages instance variable. It then filters only the packages where picked_up attribute is true and the user_id attribute is equal to the current user's ID, and assigns them to @my_packages.
- set_package method sets the @package instance variable to the Package object with the ID specified in the params.
- packages_params method permits only the address and content parameters from the params hash.

Recy Buddies

controllers

app/controllers/pages_controller.rb

- Created a PagesController class.
- Inherited from the ApplicationController.
- Defined an index method within the PagesController class.
- The index method will render the index view by default.

app/controllers/pick_ups_controller.rb

- Define a PickUpsController class that inherits from ApplicationController.
- Set up before actions to set the user and package for certain controller actions.
- Define an index action that retrieves all pick ups for the current user and all packages.
- Define a new action that creates a new Pickup instance for a given package if the current user is not the package owner.
- Set up a private method to set a Pickup instance for show and destroy actions.

Recy Buddies

models

app/models/package.rb

- Created a Ruby on Rails model called "Package".
- Inherited from "ApplicationRecord", the default base class for Rails models.
- Defined a "belongs_to" association to the "User" model, indicating that a "Package" object belongs to a single "User" object.
- The association is based on a foreign key "user_id" that should be present in the "packages" database table.
- This code allows for easy retrieval of a user's packages and their association with the user.

app/models/pick_up.rb

- Created a new class called "PickUp" that extends (inherits from) the "ApplicationRecord" class.
- Defined a "belongs_to" association for "PickUp" that connects it to the "Package" model, meaning that a "PickUp" instance belongs to a single "Package" instance.
- Also defined a "belongs_to" association for "PickUp" that connects it to the "User" model, meaning that a "PickUp" instance belongs to a single "User" instance.

app/models/user.rb

- Included default devise modules by adding them as arguments to devise method in the User model class definition.
- Declared a has_many association from User to Package models, with dependent: :destroy option to destroy all associated packages when user is destroyed.
- Added devise authentication modules: :database_authenticatable, :registerable, :recoverable, :rememberable, and :validatable to the User model. These modules provide functionalities such as authentication, registration, password recovery, session management, and validation of user input data.

Recy Buddies

views

app/views/packages/_form.html.erb

- Created a form using `simple_form_for` method passing in a variable `@package`
- Added error notification messages if there are errors in the form
- Created two form input fields for the address and content using `f.input` method
- Created a form action with a submit button using `f.button` method
- Rendered the form using `erb` template tags

app/views/packages/index.html.erb

- Created a page to display a list of recycling packages.
- Used Bootstrap classes to style the page.
- Checked if the user is signed in and displayed a welcome message with the user's email.
- Listed all the packages in the database using the `@packages` instance variable.
- Checked if the package has been picked up, and skipped it if so.
- Displayed the package's address, content, user's email, and a link to the package's details.
- Checked if the current user is not the owner of the package and the package has not been picked up, and displayed a button to pick up the package, which sends a POST request to the server.

app/views/packages/list_my_picked_up_packages.html.erb

- Accessed an instance variable `@my_packages` in the view file.
- Used a `each` loop to iterate over `@my_packages` array.
- For each `my_package`, created an HTML list item with class `list-group-item` and `d-flex`.
- Created two divs inside the list item, one with class `ms-2 me-auto` and another with class `ms-2 me-auto`.
- In the first div, displayed the address of the package using ERB syntax.
- In the second div, displayed the content of the package using ERB syntax.

Recy Buddies

views

app/views/packages/new.html.erb

- Created a view file for a new recycling package
- Added a meta viewport tag to enable responsiveness on different devices
- Defined a div with the class "adjust" to adjust the view of the recycling package
- Defined a div with the class "recycling_container" to contain the form for the recycling package
- Defined a div with classes "col-12 col-md-10 col-lg-8" to define the width of the container div on different devices
- Added a h1 tag with the text "New Recycling Package"
- Rendered a form partial with a local variable "package" assigned to an instance variable "@package"
- Defined a div with the class "button justify-content-center" to contain the link to the "Back" button
- Created a link to the packages path with the text "Back" and the class "btn btn-primary"

app/views/packages/show.html.erb

- Initialized a container to display the package information.
- Created a row to center the container.
- Added a notice section to display any notifications.
- Displayed the package's address.
- Displayed the package's content.
- Retrieved the email of the user who created the package by finding the user based on the package's user ID and splitting the email to display only the username.
- Added a "Back" button to return to the packages page.
- Checked if the user has permission to delete the package.
- If the user has permission to delete, added a "Delete" button that sends a DELETE request to delete the package and shows a confirmation dialog before deleting.

Recy Buddies

views

/app/views/pick_ups/index.html.erb

- Render a view for the user to see their pick-up packages.
- Retrieve all packages and pick-ups from the database.
- For each package and pick-up, check if the current user has a pick-up for that package.
- If there is a pick-up, display the package's address, content, and whether it has been picked up.
- Provide a link for the user to cancel the pick-up, which will call the `user_pick_up_path` with the pick-up's ID and the user's ID and use the delete HTTP method.
- Add social media links to the footer.

app/views/pick_ups/new.html.erb

- Created an file which includes a meta tag to set the viewport width and initial scale for the page.
- Added a heading element with a style attribute to display the address of a pick-up package stored in an instance variable called `@pick_up_package`.
- Added another heading element with a style attribute to display the content of the pick-up package stored in the `@pick_up_package` instance variable.

Recy Buddies

views

app/views/shared/_flashes.html.erb

- Created an ERB template which uses conditional statements to display alerts based on the presence of notice and alert variables.
- If the notice variable is present, it displays an alert-info styled div with a close button and the value of the notice variable.
- If the alert variable is present, it displays an alert-warning styled div with a close button and the value of the alert variable.

app/views/shared/_footer.html.erb

- Created an HTML div element with a class attribute set to "footer".
- Created another HTML div element inside the footer div with a class attribute set to "footer-links".
- Added five anchor elements inside the footer-links div with href attributes pointing to various social media websites and font awesome icons as children.
- Created another HTML div element inside the footer div with a class attribute set to "footer-copyright".
- Added a text message with a heart icon as a child of the footer-copyright div.
- Concluded with a comment indicating that the footer was made at Le Wagon.

app/views/shared/_navbar.html.erb

- Created an HTML file with a meta tag to set the viewport width and initial scale for the page.
- Created a div element with several classes including "navbar", "navbar-expand-sm", "navbar-light", and "navbar-lewagon".
- Created another div element inside the navbar div with a class attribute set to "container-fluid".
- Added a link_to helper method inside the container-fluid div to create a logo image with a URL set to the root_path.
- Added a button with a class attribute set to "navbar-toggler" and data-bs-toggle, data-bs-target attributes set to control the expansion of the navbar.
- Created a div element with a class attribute set to "collapse navbar-collapse" to contain a navbar menu list.
- Added a navbar list with several list items including links to packages_path and new_package_path based on whether the user is signed in or not.
- Added a dropdown list item containing a user avatar and a dropdown-menu-end div containing links to pick_ups_path, list_my_picked_up_packages_path, and destroy_user_session_path based on whether the user is signed in or not.
- Concluded the HTML file with comments indicating the purpose of some of the HTML elements.

Recy Buddies

views

app/views/layouts/application.html.erb

- Used the render method to insert the contents of the _navbar.html.erb, _footer.html.erb, and _flashes.html.erb partial files into the page-container div.
- Added a main tag with an id attribute set to "main-content" to contain the main content of the page.
- Used the yield method to include the contents of the view file that is being rendered.

app/views/pages/index.html.erb

- Created a carousel using Bootstrap's carousel component to display multiple images.
- Used the carousel's indicators to show which slide is currently being viewed.
- Populated each carousel slide with an image and a caption.
- Added previous and next buttons to navigate through the carousel.
- Created two cards with headings and text to describe the platform's purpose and target audience.