

# Bakester

---

## Necessary info for marking

### Links

#### **Links**

---

Deployed Website: <http://bakester.herokuapp.com>

---

Github: <https://github.com/leighpoulier/bakester>

---

Trello: <https://trello.com/b/Uayp6SHv/bakester>

### Seeding the database.

A series of json files are read by the seeds file, and their contents inserted into the database. Images are also attached to their relevant entities and uploaded to AWS S3 as part of the seeds process.

The json files have been modified to include fields that enable relation to other entities without use of foreign keys. This is because the primary keys are probably different to what they were when the json data was taken from the database. To avoid problems with holes in the sequence of primary keys, the original primary keys are ignored. New primary keys are generated as part of the insertion. For entities that contain foreign keys another field, either "name" or the timestamp contained in "created\_at" fields, is used to ascertain the correct related entity relationship and rebuild new foreign keys.

### User accounts

## About Bakester

Bakester is a Ruby on Rails web app where people can buy "bakes" from home bakers. "Bakes" might include cakes, cupcakes, slices, pies, pastries, tarts, desserts etc. On Bakester a home baker can advertise baked items they are able and willing to produce in exchange for payment, and any other user can place an order for a specific bake. The baker receives the order, produces the bake, and delivers it to the customer.

### The Problem



"Car Cake" by Anne (source: BBC - The Great British Bake Off: An Extra Slice)

Many people, particularly those with young kids, may feel pressured to provide an impressive celebration cake for an upcoming event such as a kid's birthday. The pressures of social media and the demands of the honourary child place the parent in a difficult position, particularly for those without the skills to bake at home.

Non-bakers may look into sourcing a cake from a local supplier, however professionally made cakes can be very expensive, and are oriented more towards large celebrations such as weddings. Such cakes are outside the budget of most people planning a smaller event, and may be considered overkill. The aim this website aims to fill the gap by providing access to affordable home baked celebration cakes and other bakes.

The flipside of this situation is someone who DOES have the skills and time to bake at home, and is interested in making a bit of money from those skills. The home baker can post a listing called a "bake" on Bakester, to offer home baked celebration cakes or other items to the general community in exchange for payment.

## Target Audience

This web app aims to provide a marketplace solution to the above problems for two classes of users.

User Type	Definition
Customers	People who want to buy a home made cake or other baked item rather than make it themselves, due to lack of ability, time etc.
Bakers	People who want to bake at home on behalf of others in exchange for payment.

A third type of administrative user called an "Admin" is also part of the design for this web app, although this user does not necessarily participate in the marketplace.

## The Bakester Web App

Bakester was developed to provide a place for people to purchase home made bakes from other people. The main use case for the site is the need for a celebration cake, such as a birthday cake for a child. However, any type of home made baked item can be listed and ordered on the site.

## Features

The website implements the following features:

### General

- Neopolitan icecream colour scheme (chocolate, strawberry and vanilla) to add to the sweetness.
- Responsive design which adapts to mobile, tablet and desktop views.

### Customers

- Card based layout to view available bakes
- Search / filter / sort function for the list of bakes.
- Cart where bakes may be placed in order to prepare a list for checkout.
- Checkout function which finalises the cart into an order.
- Past orders page to keep track of orders and their progress
- Upgrade a "customer" account into a "baker" account which changes the UI

### Bakers

- Add and edit bake listings
- Upload an image on a bake listing. Images are stored in an Amazon S3 bucket.
- View my bakes, and manage their visibility to customers on the site.
- View my jobs, which are elements of customer orders containing a single bake.
- Receive a notification of new jobs at login
- Acknowledge receipt of a job, and update the status as it is produced and shipped.

### Admin

- Manage many aspects of the site as if they were the owner. For example
  - Create new categories
  - Moderate bakes for unacceptable user generated content. Edit bakes, remove images, or delete bakes completely if they are unreferenced in the database.
  - View all bakes, even if they are hidden.
  - View any user's cart and past orders.
  - View any baker's list of bake jobs, and edit their status.

## Tech Stack

Bakester uses a Ruby on Rails application framework. All of the programming code is written in Ruby, and the individual web pages are written in html with embedded Ruby, preprocessed by the puma server and delivered to the browser as standard HTML. Pages are styled using SCSS.

The backend database is managed by PostgreSQL.

Bakester makes use the following additional components:

- Devise - a rails engine for login authorisation and user, session, and password management.
- Bootstrap - for styling and standardised features such as buttons, modal popup dialog boxes and tooltips. Many of the standard bootstrap styles are customised and extended, in particular colours.
- JQuery - for a small amount of required client side scripting, including modal popups and tooltips.
- MiniMagick - a ruby gem for image manipulation - which is used to reduce the size of uploaded images before sending them to the Amazon S3 bucket.

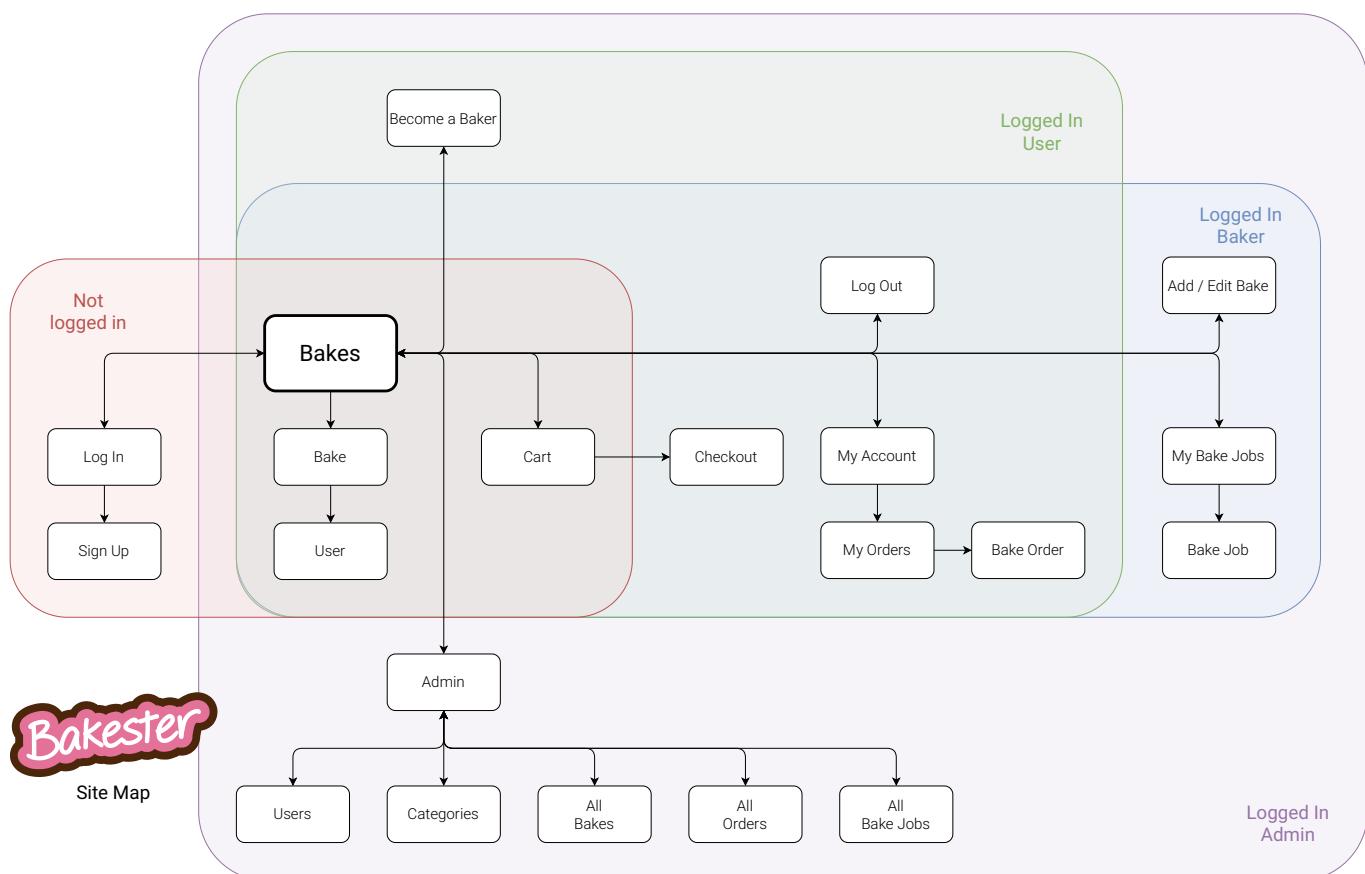
Bakester is deployed to a Heroku on a free "Hobby Dev" level account.

## Third Party Services

Bakester makes use of Amazon Web Services S3 buckets for its image storage. Two different buckets are setup, one for the development environment, and one for the production. This is so that the development bucket can be cleared easily, without affecting the online Heroku hosted website.

## Sitemap

The following diagram represents a map of the main areas of the Bakester site and the typical flow of navigation between them. The bold "Bakes" node is the root of the site, and all initial access to the site starts on this page, which includes a listing of all the bakes available for order.



There are four shaded regions on the sitemap, which represent the three classes of user accessing the site and a fourth state where the user is not logged in.

## Walkthrough / Screenshots

Different parts of the site are available according to the users' status as follows:

## **Not Logged in**

A user who is not logged in can browse bakes available on the site, and add them to their cart, but cannot checkout without signing in. They can also view other user profiles.

## **Logged in as User**

When logged in as a standard user, the above features are available, and checkout functionality is added. A user can also manage one's own account and view one's own past orders. There is also a function available to upgrade one's account to "baker" status.

## **Logged in as Baker**

After upgrading their account to a baker, the baker can access all of the above functions with the addition of creating and editing bake listings, and managing their own bake jobs.

## **Logged in as Admin**

An admin can access all of the above, with the addition of the ability to manage all users, categories, bakes, orders and bake jobs.

However, an admin is not automatically a baker, they too will need to upgrade their account to baker if they wish to create bakes and manage bakejobs in their own name.

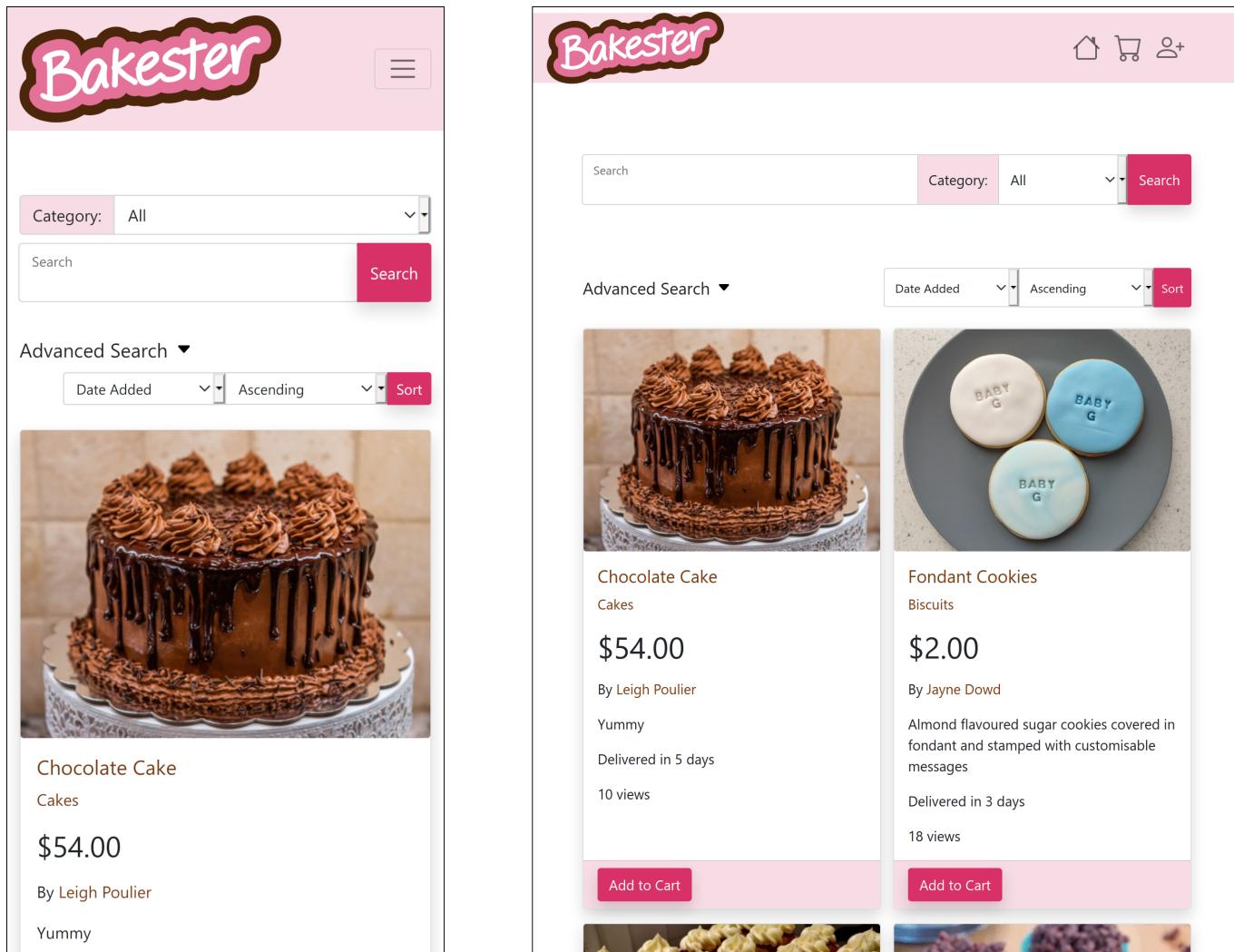
## Walkthrough / Screenshots

We will follow a typical user journey as they move through the site, create an account, use the features available to a user, upgrade their account to baker, and use the features available to a baker. Following this the features available to an Admin will be presented.

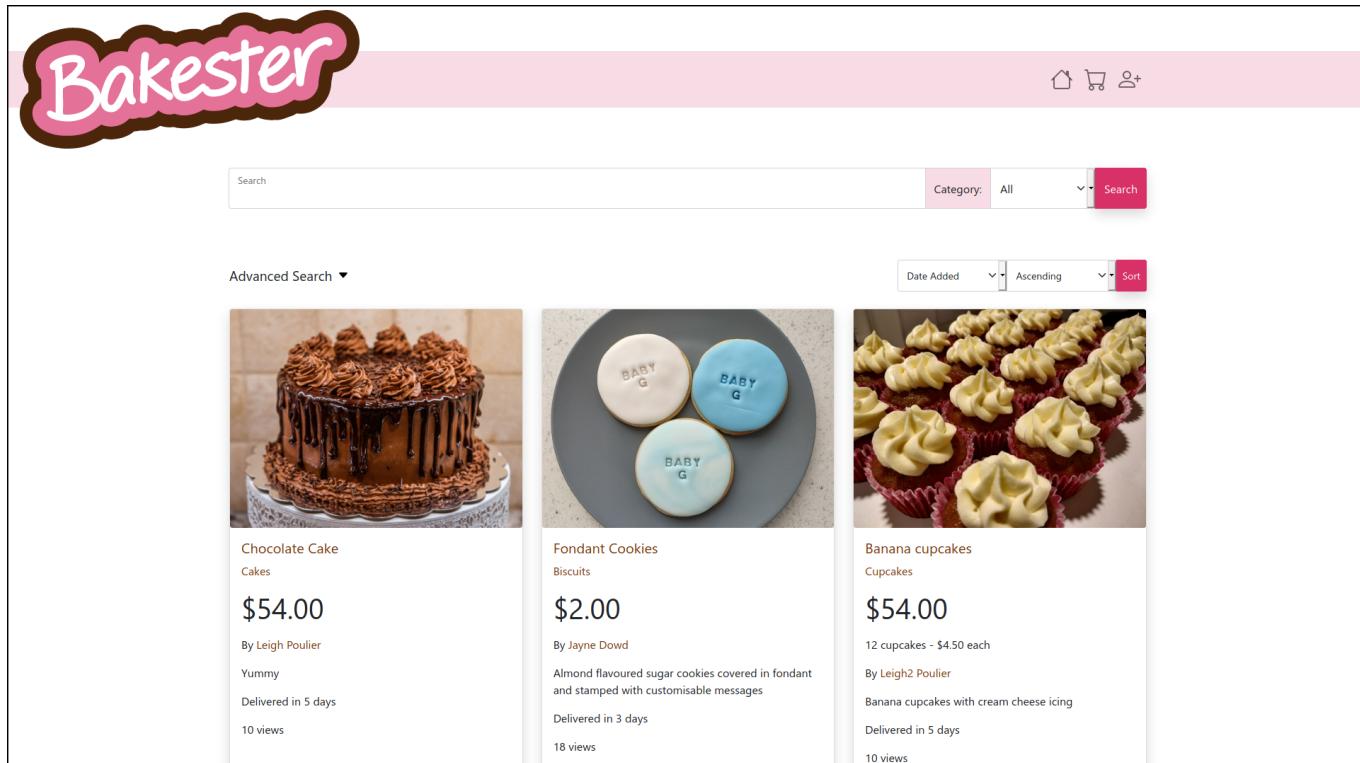
### **New user first contact**

When a new user first accesses the site, they are presented with the bakes list. This is a vertical scrollable list of bakes presented on cards, which responsively adapts by expanding up to 3 columns wide.

On the mobile view, a hamburger menu replaces the navigation bar icons. Pressing the hamburger button opens a drawer containing the navigation links



Bakester main page on mobile and tablet.

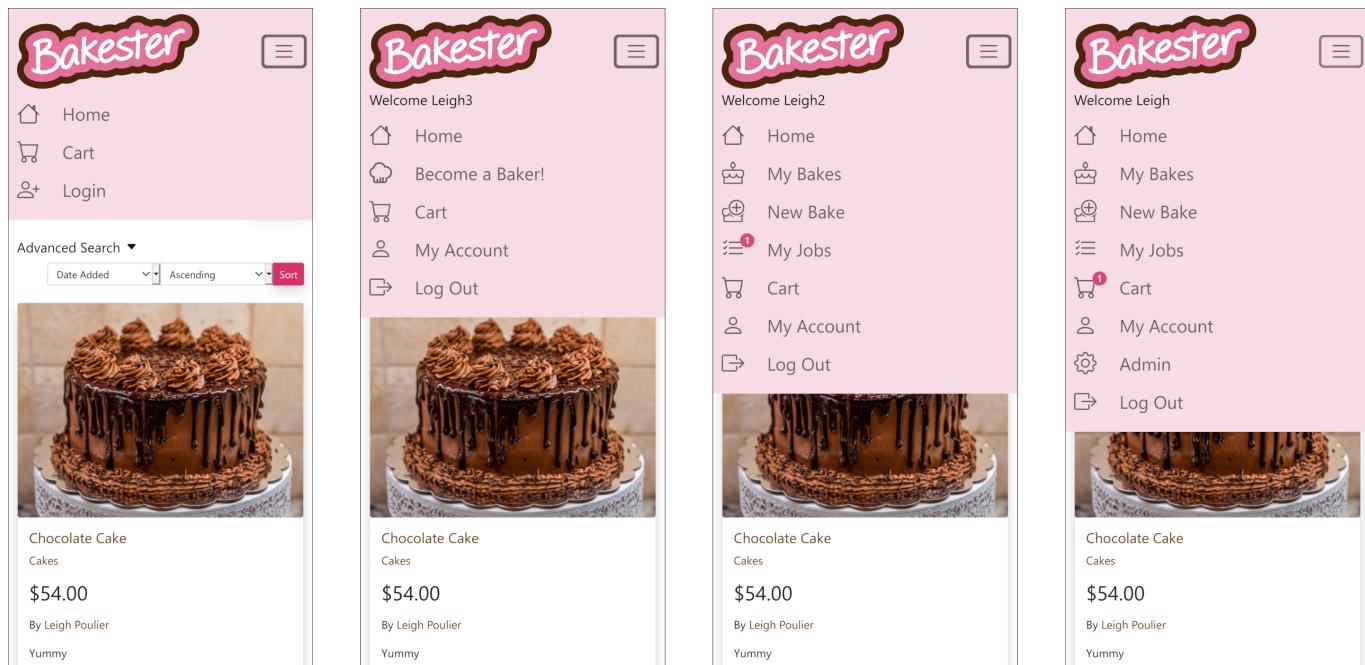


Bakester main page on desktop.

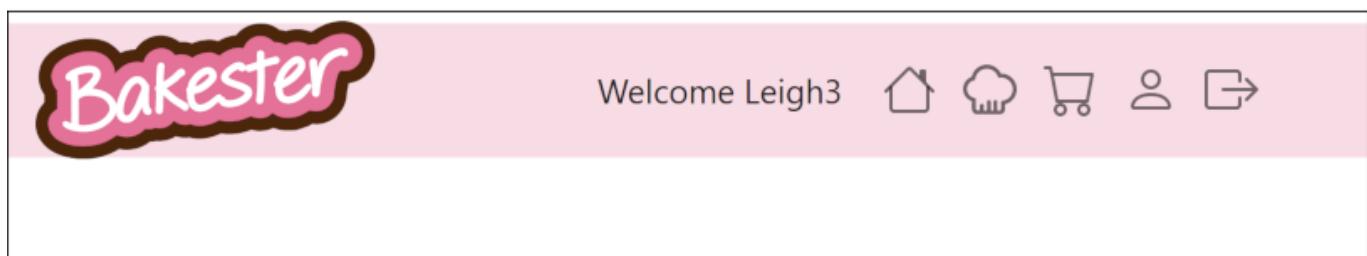
## Navigation

On mobile view, the navigation menu expands downward once the hamburger is pressed, and displays additional options for a logged in user, a baker, and an admin (who is also a baker in this case).

Also visible in the below image are the badges added on the "cart" icon and "my jobs" icon, to indicate that there are items in the cart, or new jobs respectively.



On a tablet and desktop, the hamburger button disappears and is replaced by a horizontal list of icons. The following image displays tablet view, but desktop is the same except for its width.



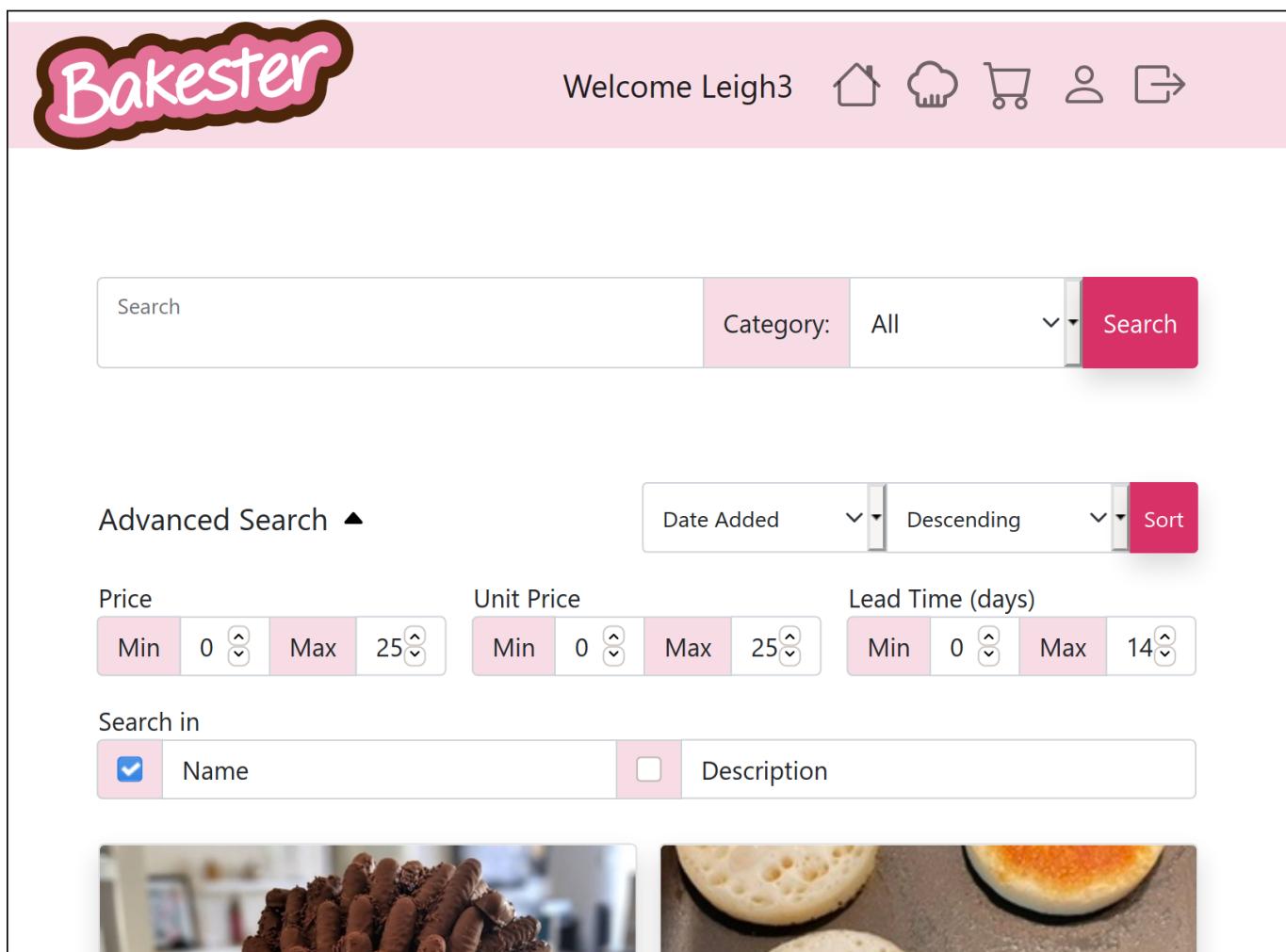
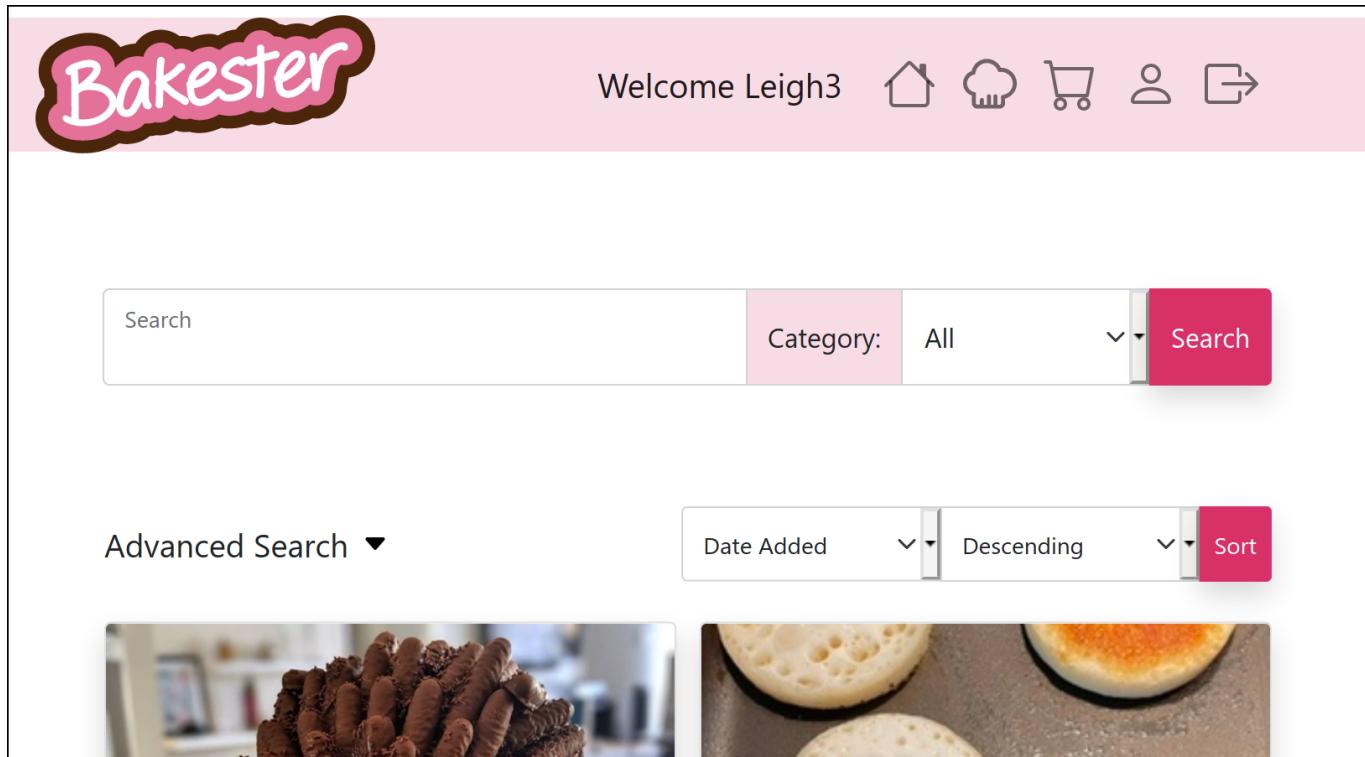
## Search

On the bake listing page, a search form is available where the user can search by text, filter various properties of the bakes, and sort by any property in ascending or descending direction. By default only the search text field, the category select box and the sort select boxes are visible, but clicking on the "Advanced Search" text expands the rest of the search form to show all available search and filter options.

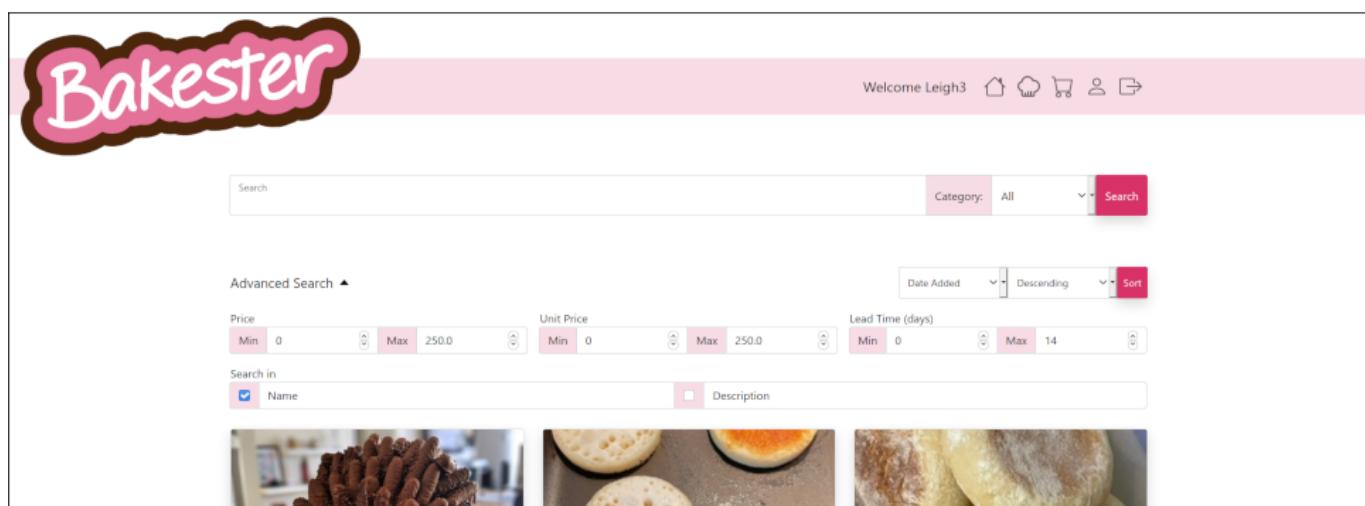
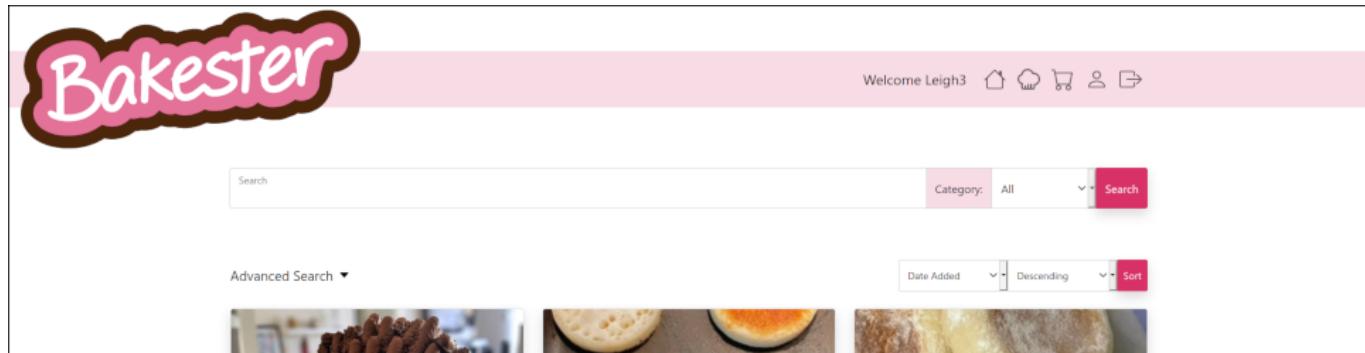
The screenshot shows the Bakester mobile search interface. At the top is the 'Bakester' logo. Below it is a search bar with a dropdown menu for 'Category' set to 'All'. A search input field and a pink 'Search' button are also present. An 'Advanced Search' section includes dropdowns for 'Date Added' (set to 'Ascending') and a 'Sort' button. A large image of a chocolate cake with chocolate frosting and drips is displayed. Below the image, the product details are listed: 'Chocolate Cake', 'Cakes', '\$54.00', 'By Leigh Poulier', and 'Yummy'.

The screenshot shows the Bakester mobile search interface with an expanded 'Advanced Search' section. It includes filters for 'Price' (Min: 0, Max: 250.0), 'Unit Price' (Min: 0, Max: 250.0), and 'Lead Time (days)' (Min: 0, Max: 14). There is also a 'Search in' section with checkboxes for 'Name' (checked) and 'Description'. A preview image of a chocolate cake is shown at the bottom.

Bake search form on mobile.



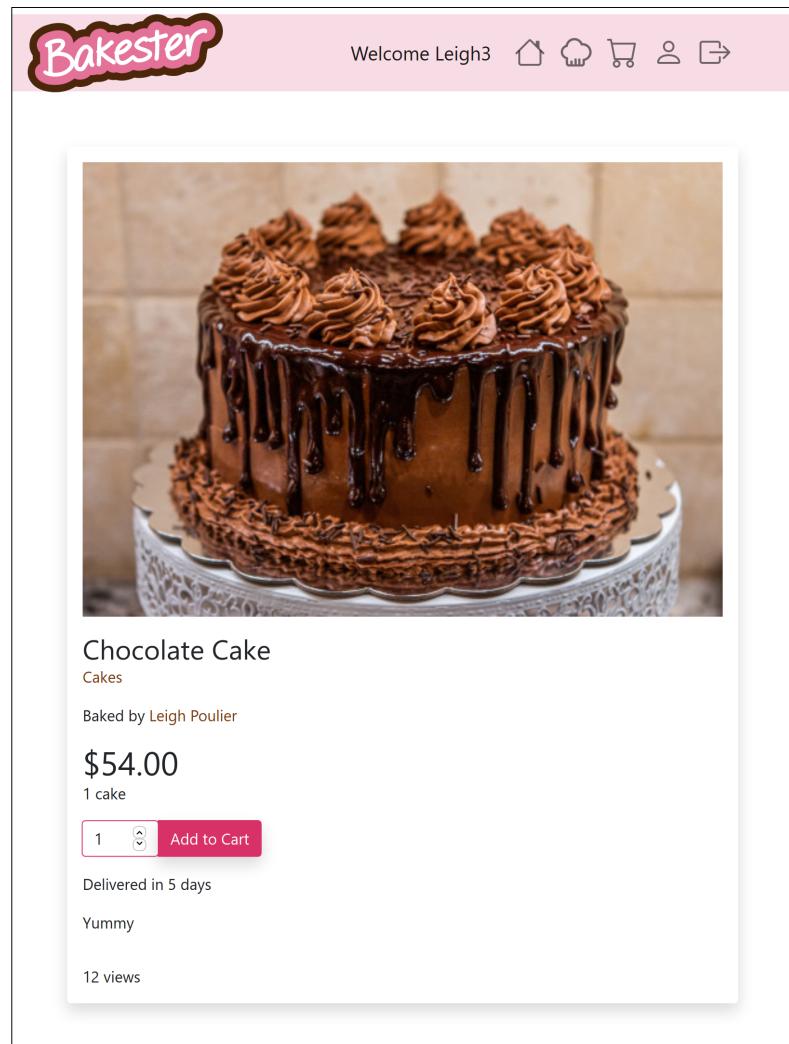
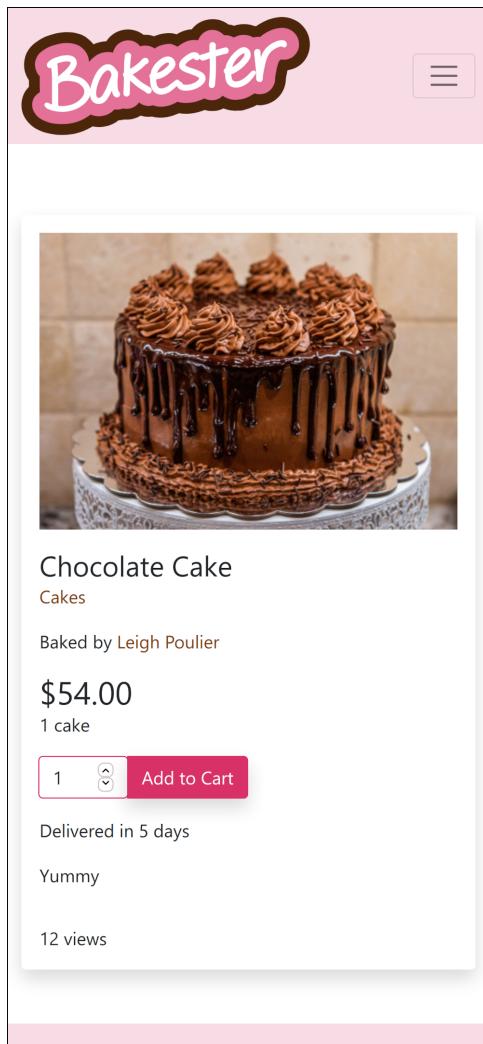
Bake search form on tablet.



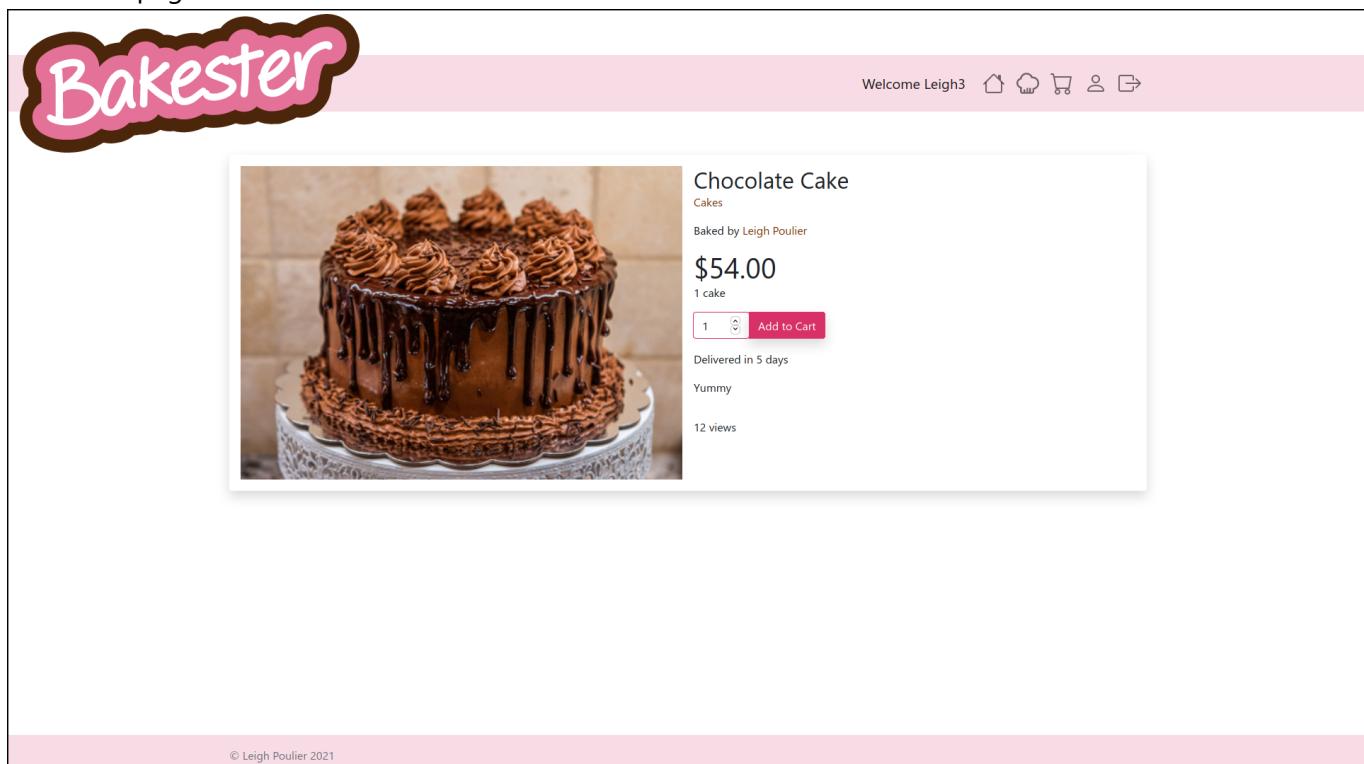
Bake search form on desktop.

## View Bake

The user can click on a bake in the main list, and be taken to it's detail page to read the full description and view a larger image



Bake view page on mobile and tablet.



Bake view page on desktop.

**View Cart**

Once the user clicks "Add to Cart", the bake is added to their cart. If they are not logged in this is handled by session storage, and for a logged in user it is stored in the database. After clicking on the navigation link to view their cart they will be presented with the view cart page.

The image displays two side-by-side screenshots of the Bakester website's cart page. Both screens have a pink header with the 'Bakester' logo and a navigation bar with icons for home, menu, shopping cart (with a notification of 1), user profile, and a share icon.

**Left Screen (Mobile/Tablet View):**

- Header:** 'Cart'.
- Table:** A table showing the item details. The columns are 'Bake', 'Qty', 'Unit Price', and 'Line Total'.

Bake	Qty	Unit Price	Line Total
Chocolate Cake (1 cake)	1	\$54.00	\$54.00
- Buttons:** '+', '-' (quantity adjustment), and 'x' (remove) buttons next to the quantity input field.
- Total:** 'Total \$54.00'.
- Actions:** 'Empty Cart' (dark button) and 'Checkout' (pink button).
- Footer:** © Leigh Poulier 2021.

**Right Screen (Tablet/Full Web View):**

- Header:** 'Cart'.
- Table:** A table showing the item details. The columns are 'Bake', 'Qty', 'Unit Price', and 'Line Total'.

Bake	Qty	Unit Price	Line Total
Chocolate Cake (1 cake)	1	\$54.00	\$54.00
- Buttons:** '+', '-' (quantity adjustment), and 'x' (remove) buttons next to the quantity input field.
- Total:** 'Total \$54.00'.
- Actions:** 'Empty Cart' (dark button) and 'Checkout' (pink button).
- Footer:** © Leigh Poulier 2021.

Cart view page on mobile and tablet.

The screenshot shows the 'Cart' page of the Bakester website. At the top left is the 'Bakester' logo. To the right of the logo is a pink header bar with the text 'Welcome Leigh3' and icons for home, search, shopping cart (with a red notification dot), user profile, and a share icon. Below the header is a light gray navigation bar with the word 'Cart'. The main content area has a white background and displays a table of items in the cart. The table has columns for 'Bake', 'Qty', 'Unit Price', and 'Line Total'. A single item, 'Chocolate Cake (1 cake)', is listed with a quantity of 1, a unit price of \$54.00, and a line total of \$54.00. To the right of the table are three small dark gray buttons with white symbols: a plus sign, a minus sign, and an 'X' for quantity adjustment. Below the table is a horizontal line. At the bottom left of the page is a small text footer: '© Leigh Poulier 2021'.

Cart view page on desktop.

## Check Out

The user can then press the "Checkout" button, which after confirmation transforms the cart into an order and sends bake jobs to bakers. After checking out, the user sees the new order confirmation.

Bakester

## Order

<b>Order No</b>	30				
<b>Submitted</b>	26 Nov 2021				
Status	Baker	Bake	Qty	Unit Price	Line Total
Received	Leigh Poulier	Chocolate Cake	1	\$54.00	\$54.00
<b>Total</b>			\$54.00		

**Back**

Welcome Leigh3

## Order

<b>Order No</b>	30				
<b>Submitted</b>	26 Nov 2021				
Status	Baker	Bake	Qty	Unit Price	Line Total
Received	Leigh Poulier	Chocolate Cake	1	\$54.00	\$54.00
<b>Total</b>			\$54.00		

**Back**

Completed order page on mobile and tablet.

Bakester

## Order

<b>Order No</b>	30				
<b>Submitted</b>	26 Nov 2021				
Status	Baker	Bake	Qty	Unit Price	Line Total
Received	Leigh Poulier	Chocolate Cake	1	\$54.00	\$54.00
<b>Total</b>			\$54.00		

**Back**

Completed order page on desktop.

## My Account

Links to upgrade to baker are provided on the navigation bar (see above). The link can also be accessed from the "My Account" page, along with options to edit username, email and password.

Bakester

## My Account

First name  
Leigh3

Last name  
Poulier

Email  
leigh3@test.com

Current password (we need your current password to confirm your changes)

Update

[Change Password](#)

[View My Orders](#)

Bakester

Email  
leigh3@test.com

Current password (we need your current password to confirm your changes)

Update

[Change Password](#)

[View My Orders](#)

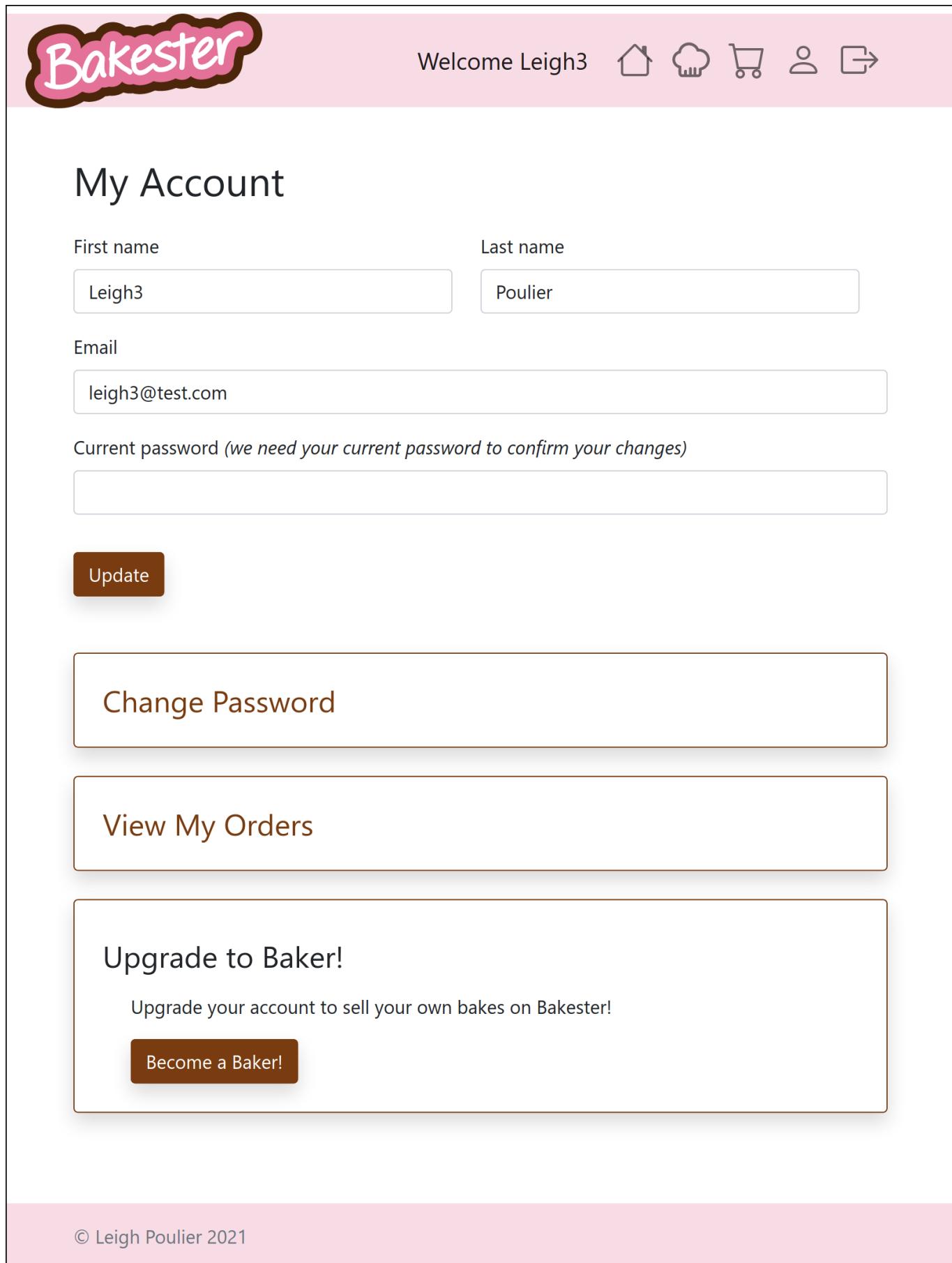
**Upgrade to Baker!**

Upgrade your account to sell your own bakes on Bakester!

[Become a Baker!](#)

© Leigh Poulier 2021

My account page on mobile.



The screenshot shows a mobile application interface for 'Bakester'. At the top, there's a pink header bar with the 'Bakester' logo on the left, a 'Welcome Leigh3' message, and icons for home, cloud, cart, user profile, and a right-pointing arrow. Below the header, the title 'My Account' is displayed in a large, dark font.

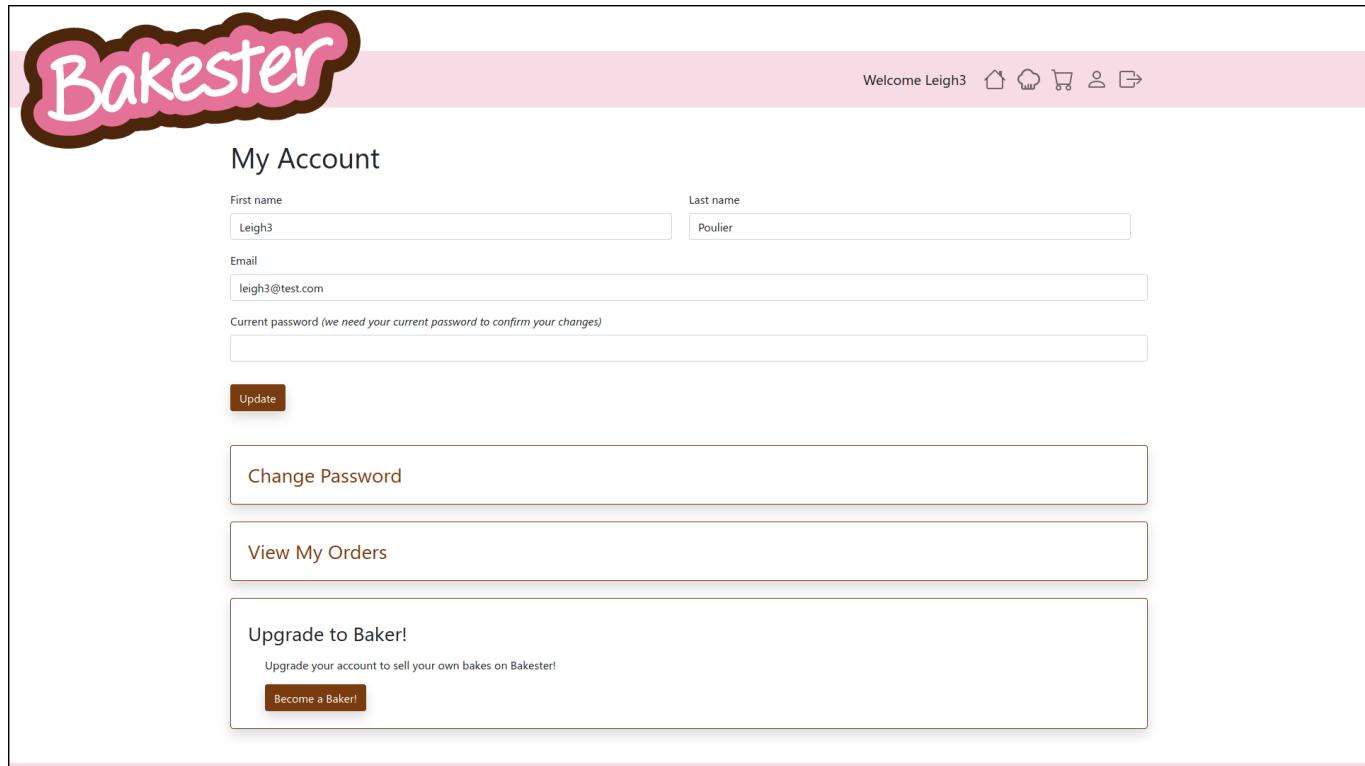
The main form area contains fields for 'First name' (Leigh3) and 'Last name' (Poulier), both in light blue input fields. An 'Email' field below contains 'leigh3@test.com'. A note in italicized text says 'Current password (we need your current password to confirm your changes)' followed by an empty password input field.

A brown rectangular button labeled 'Update' is positioned below the email field. To the right of the update button are three rounded rectangular boxes with brown borders:

- The first box contains the text 'Change Password'.
- The second box contains the text 'View My Orders'.
- The third box contains the text 'Upgrade to Baker!' followed by the subtext 'Upgrade your account to sell your own bakes on Bakester!', and a brown button labeled 'Become a Baker!'.

At the bottom of the screen, a pink footer bar contains the copyright notice '© Leigh Poulier 2021'.

My account page on mobile.



My account page on mobile.

## My Bakes

After upgrading their account to a baker the user will be able to add a new bake using the "new bake" option in the navigation menu. They will then be presented with the form for creating or editing bakes.

The image displays two side-by-side screenshots of a mobile application interface for adding a new bake. The app has a pink header with the word "Bakester" in white, stylized font.

**Left Screenshot (Portrait Mode):**

- Name:** An input field for entering the name of the bake.
- Category:** A dropdown menu labeled "Select Category".
- Lead Time (days):** An input field with up and down arrows for selecting time in days.
- Description:** A large text area for describing the bake.
- Price:** An input field with up and down arrows. Below it, the text "per" is followed by another input field with up and down arrows.
- Count:** An input field showing the value "1" with up and down arrows.
- Singular Item (eg: cake, cookie):** An input field for specifying the singular item type.
- Add Image:** A section with a "Browse..." button and a message "No file selected."
- Visibility:** Radio buttons for "Active" (selected) and "Hidden".

**Right Screenshot (Landscape Mode):**

- Category:** A dropdown menu labeled "Select Category".
- Lead Time (days):** An input field with up and down arrows for selecting time in days.
- Description:** A large text area for describing the bake.
- Price:** An input field with up and down arrows. Below it, the text "per" is followed by another input field with up and down arrows.
- Count:** An input field showing the value "1" with up and down arrows.
- Singular Item (eg: cake, cookie):** An input field for specifying the singular item type.
- Add Image:** A section with a "Browse..." button and a message "No file selected."
- Visibility:** Radio buttons for "Active" (selected) and "Hidden".
- Buttons:** "Create Bake" (pink background) and "Cancel" (brown background).

**Footer:** © Leigh Poulier 2021

Bake form on mobile.

The screenshot shows the 'Bakester' application interface. At the top, there's a pink header bar with the 'Bakester' logo on the left, followed by a 'Welcome Leigh2' message and several icons: a house, a crown, a plus sign, a list with a red notification badge (containing the number 1), a shopping cart, a person, and a right-pointing arrow.

The main content area is titled 'Add a new bake'. It contains the following fields:

- Name:** An input field for the name of the bake.
- Category:** A dropdown menu labeled 'Select Category' with up and down arrows.
- Lead Time (days):** An input field with up and down arrows for selecting days.
- Description:** A large text area for describing the bake.
- Price:** An input field with up and down arrows, followed by the text 'per'.
- Count:** An input field with up and down arrows containing the value '1'.
- Singular Item (eg: cake, cookie):** An input field.
- Add Image:** A button to add an image, with a 'Browse...' button next to it. The message 'No file selected.' is displayed.
- Visibility:** Radio buttons for 'Active' (selected) and 'Hidden'.
- Create Bake:** A pink button to create the bake.
- Cancel:** A brown button to cancel.

At the bottom of the screen, there's a pink footer bar with the copyright notice '© Leigh Poulier 2021'.

Bake form on tablet.

The screenshot shows the 'Add a new bake' form. At the top right, there's a navigation bar with icons for home, account, and shopping cart. Below the header, the title 'Bakester' is displayed in a large, stylized font. The main form area has several input fields: 'Name' (text input), 'Category' (dropdown menu with 'Select Category'), 'Lead Time (days)' (text input), 'Description' (text area), 'Price' (text input), 'Count' (text input with value '1'), and 'Singular Item (eg: cake, cookie)' (text input). There are also buttons for 'Add Image' (with a 'Browse...' button and a note 'No file selected.'), 'Visibility' (radio buttons for 'Active' and 'Hidden'), and 'Create Bake' (pink button) or 'Cancel' (dark button).

Bake form on desktop.

## Edit a Bake

After the creating a new bake, it will appear with an "Edit" button instead of "Add to Cart" in all views. For example in desktop view it can be seen that the first of these three bakes is owned by the currently logged in baker and therefore can be edited:

The screenshot displays three baked goods cards. Each card includes a thumbnail image, the name, price, a brief description, the baker's name, and a 'Views' count. The first card, 'Banana cupcakes', has an 'Edit' button at the bottom. The other two cards have 'Add to Cart' buttons.

Bake Type	Description	Price	Baker	Views	Action
Banana cupcakes	Cupcakes	\$54.00	Leigh2 Poulier	11	Edit
Fondant Cookies	Biscuits	\$2.00	Jayne Dowd	18	Add to Cart
Chocolate Cake	Cakes	\$54.00	Leigh Poulier	12	Add to Cart

Edit button on bake card on desktop.

Once the edit button is clicked the same form will be populated with that bake and can be edited.

# Bakester

## Edit Bake

Name  
Banana cupcakes

Category  
Cupcakes

Lead Time (days)  
5

Description  
Banana cupcakes with cream cheese icing

Price  
54.0

Count  
12

Singular Item (eg:  
cake, cookie)  
cupcakes

per

Image  


# Bakester

## Edit Bake

Price  
54.0

Count  
12

Singular Item (eg:  
cake, cookie)  
cupcakes

per

Image  


Replace Image  
Browse... No file selected.

Delete Image

Visibility:  Active  Hidden

Update Bake Delete Bake Cancel

© Leigh Poulier 2021

Bake edit form on mobile.

**Bakester**

Welcome Leigh2

### Edit Bake

Name	Category	Lead Time (days)
Banana cupcakes	Cupcakes	5
Description		
Banana cupcakes with cream cheese icing		
Price Count Singular Item (eg: cake, cookie)		
54.0	per 12	cupcakes
Image		
		

**Bakester**

Welcome Leigh2

Price	Count	Singular Item (eg: cake, cookie)
54.0	12 per	cupcakes
Image		
		
Replace Image <input type="button" value="Browse..."/> No file selected. <input type="button" value="Delete Image"/>		
Visibility: <input checked="" type="radio"/> Active <input type="radio"/> Hidden		
<input type="button" value="Update Bake"/> <input type="button" value="Delete Bake"/> <input type="button" value="Cancel"/>		

© Leigh Poulier 2021

Bake edit form on tablet.

**Bakester**

Welcome Leigh2

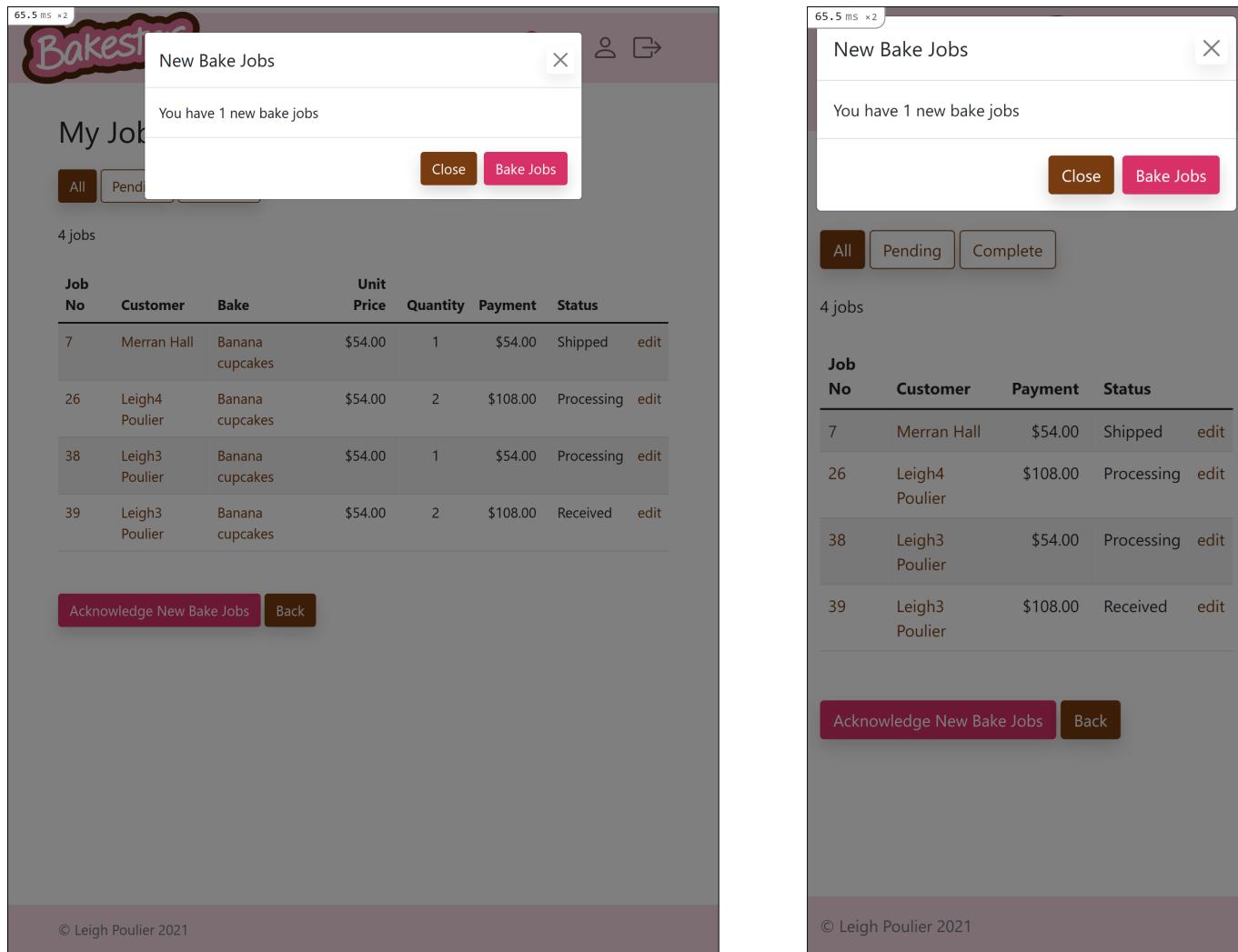
### Edit Bake

Name	Category	Lead Time (days)
Banana cupcakes	Cupcakes	5
Description		
Banana cupcakes with cream cheese icing		
Price Count Singular Item (eg: cake, cookie)		
54.0	per 12	cupcakes
Image		
		
Replace Image <input type="button" value="Browse..."/> No file selected. <input type="button" value="Delete Image"/>		

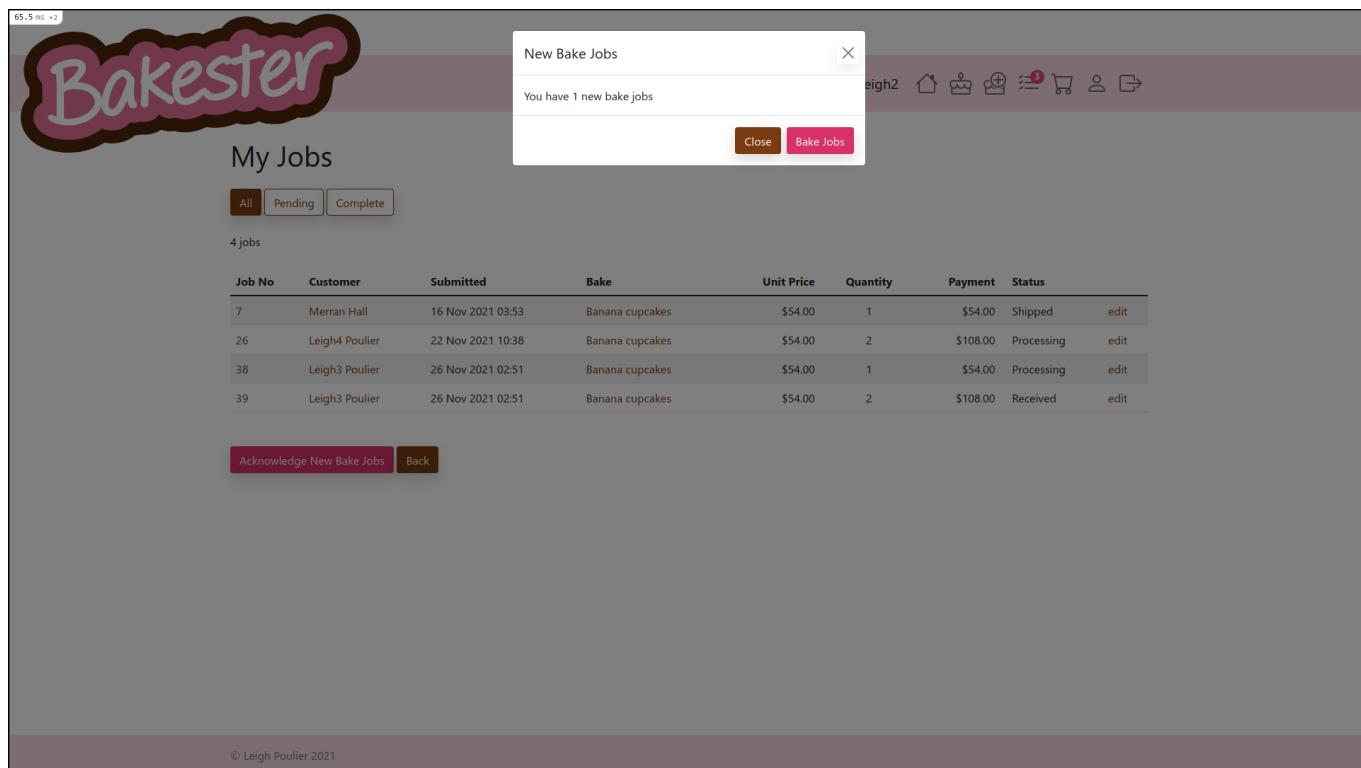
Bake edit form on desktop.

## Receiving a new job

If another user makes an order for a bake, a modal popup will appear to indicate this to the baker.



New Bake Job modal on mobile and tablet.



New Bake Job modal on desktop.

## Managing Bake Jobs

This modal prompt will provide a link to the Bake Jobs page, where the baker can see the status of all bake jobs and filter them by their completed status.

My Jobs

All Pending Complete

4 jobs

Job No	Customer	Payment	Status
7	Merran Hall	\$54.00	Shipped
26	Leigh4 Poulier	\$108.00	Processing
38	Leigh3 Poulier	\$54.00	Processing
39	Leigh3 Poulier	\$108.00	Processing

Back

© Leigh Poulier 2021

Welcome Leigh2

Bakester

My Jobs

All Pending Complete

4 jobs

Job No	Customer	Bake	Unit Price	Quantity	Payment	Status
7	Merran Hall	Banana cupcakes	\$54.00	1	\$54.00	Shipped
26	Leigh4 Poulier	Banana cupcakes	\$54.00	2	\$108.00	Processing
38	Leigh3 Poulier	Banana cupcakes	\$54.00	1	\$54.00	Processing
39	Leigh3 Poulier	Banana cupcakes	\$54.00	2	\$108.00	Processing

Back

© Leigh Poulier 2021

My bake jobs page on mobile and tablet.

Welcome Leigh2

Bakester

My Jobs

All Pending Complete

4 jobs

Job No	Customer	Submitted	Bake	Unit Price	Quantity	Payment	Status
7	Merran Hall	16 Nov 2021 03:53	Banana cupcakes	\$54.00	1	\$54.00	Shipped
26	Leigh4 Poulier	22 Nov 2021 10:38	Banana cupcakes	\$54.00	2	\$108.00	Processing
38	Leigh3 Poulier	26 Nov 2021 02:51	Banana cupcakes	\$54.00	1	\$54.00	Processing
39	Leigh3 Poulier	26 Nov 2021 02:51	Banana cupcakes	\$54.00	2	\$108.00	Processing

Back

© Leigh Poulier 2021

My bake jobs page on desktop.

## Updating Bake Job Status

If the baker clicks on link on the Job Number, or the "edit" link to the right, the status of the bake job can be updated to "Processing", "Shipped", "Delivered"

52.0 ms

### Edit Bake Job

Bake Job No	7
Order No	11
Customer	Merran Hall
Baker	Leigh2 Poulier
Submitted	16 Nov 2021
Bake	Banana cupcakes
Quantity	1
Unit Price	\$54.00
Total	\$54.00
Status	Processing ▾

**Update Bake job**

**Back**

© Leigh Poulier 2021

52.0 ms

### Edit Bake Job

Bake Job No	7
Order No	11
Customer	Merran Hall
Baker	Leigh2 Poulier
Submitted	16 Nov 2021
Bake	Banana cupcakes
Quantity	1
Unit Price	\$54.00
Total	\$54.00
Status	Processing ▾

**Update Bake job**

**Back**

Welcome Leigh2

© Leigh Poulier 2021

Edit bake job status on mobile and tablet.

The screenshot shows a web browser window for the 'Bakester' application. The title bar indicates a connection speed of 52.0 ms. The header features the 'Bakester' logo in a pink, bubbly font, a 'Welcome Leigh2' message, and several navigation icons. The main content area is titled 'Edit Bake Job'. It contains a table with the following data:

Bake Job No	7
Order No	11
Customer	Merran Hall
Baker	Leigh2 Poulier
Submitted	16 Nov 2021
Bake	Banana cupcakes
Quantity	1
Unit Price	\$54.00
Total	\$54.00
Status	Processing

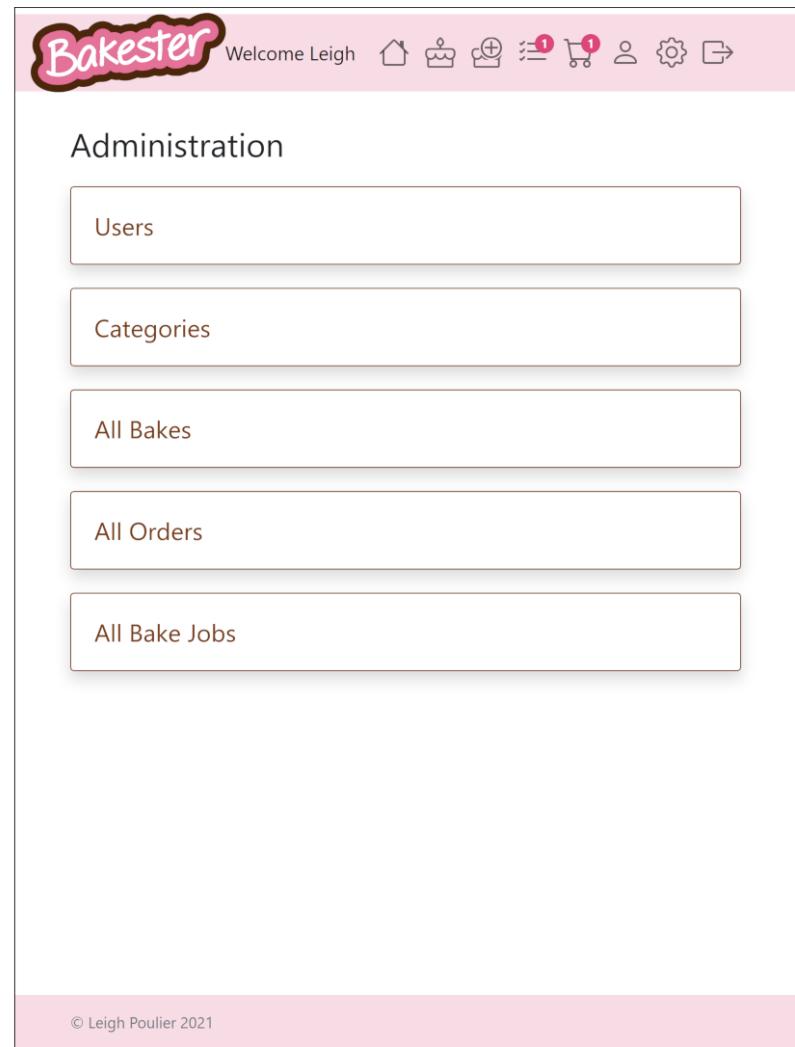
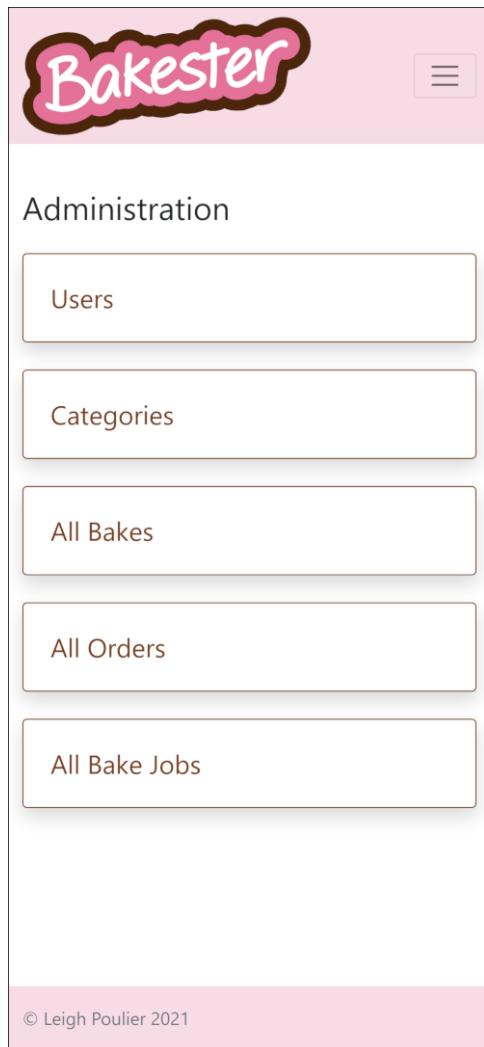
Below the table are two buttons: 'Update Bake job' (highlighted with a red box) and 'Back'.

At the bottom of the page, a footer bar contains the text '© Leigh Poulier 2021'.

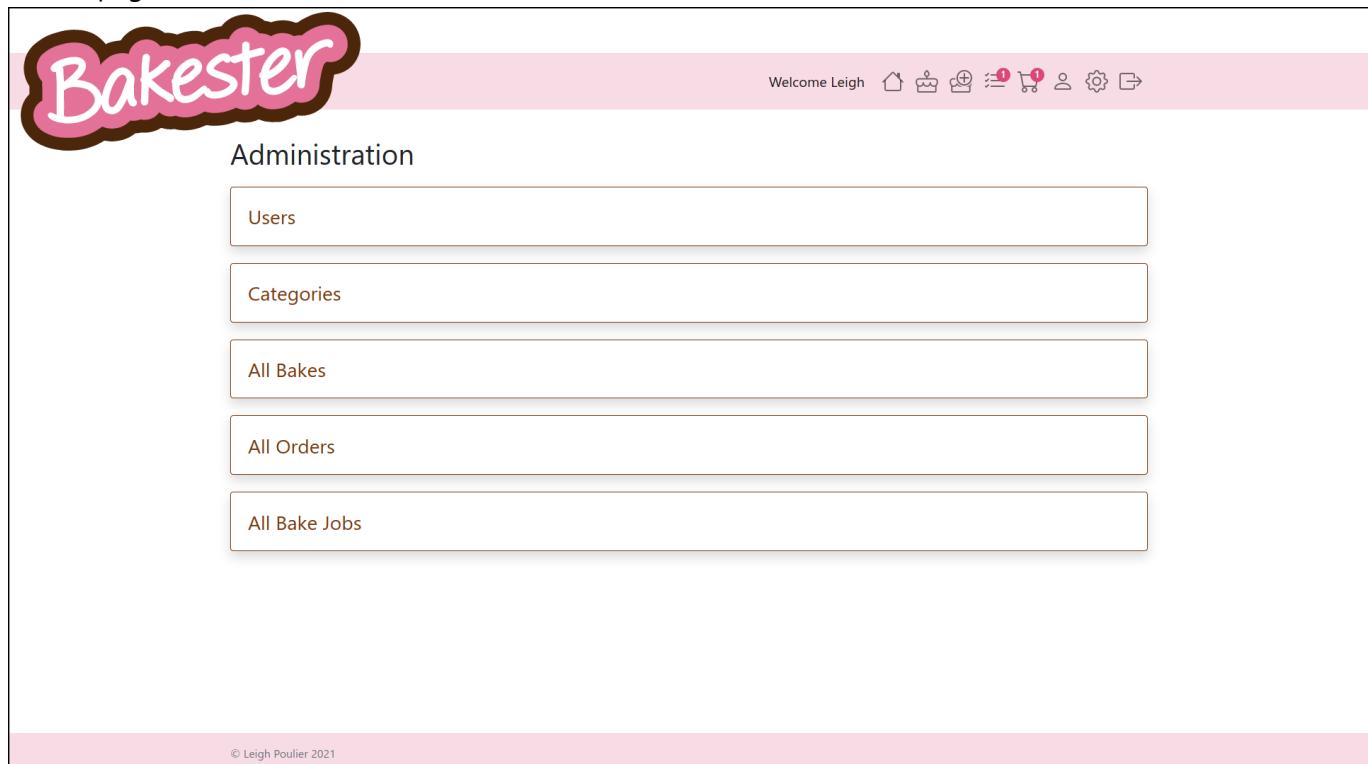
Edit bake job status on desktop.

## Admin page

An admin user can access an admin dashboard, from where they can manage all entities of the web app, including categories, users, bakes, orders, and bake jobs.



Admin page on mobile and tablet.



Admin page on desktop.

## All Users listing

Clicking on the "Users" link opens the admin-only Users index page which lists all users, and provides links to view their carts, orders, bake\_jobs, and profiles.

Name	Profile	Edit	Cart	Orders	Bake Jobs
Gen Boyd	Profile	Edit	Cart	Orders	Bake Jobs
Guapito Chaylian	Profile	Edit	Cart	Orders	Bake Jobs
Luis Chaylian	Profile	Edit	Cart	Orders	Bake Jobs
Jayne Dowd	Profile	Edit	Cart	Orders	Bake Jobs
Merran Hall	Profile	Edit	Cart	Orders	Bake Jobs
Kat Lee	Profile	Edit	Cart	Orders	Bake Jobs
Mel lim	Profile	Edit	Cart	Orders	Bake Jobs
Leigh Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh2 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh3 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh4 Poulier	Profile	Edit	Cart	Orders	Bake Jobs

Name	Profile	Edit	Cart	Orders	Bake Jobs
Gen Boyd	Profile	Edit	Cart	Orders	Bake Jobs
Guapito Chaylian	Profile	Edit	Cart	Orders	Bake Jobs
Luis Chaylian	Profile	Edit	Cart	Orders	Bake Jobs
Jayne Dowd	Profile	Edit	Cart	Orders	Bake Jobs
Merran Hall	Profile	Edit	Cart	Orders	Bake Jobs
Kat Lee	Profile	Edit	Cart	Orders	Bake Jobs
Mel lim	Profile	Edit	Cart	Orders	Bake Jobs
Leigh Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh2 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh3 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh4 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Wendy Poulier	Profile	Edit	Cart	Orders	Bake Jobs

[Back](#)

© Leigh Poulier 2021

Users list on mobile and tablet.

Name	Profile	Edit	Cart	Orders	Bake Jobs
Gen Boyd	Profile	Edit	Cart	Orders	Bake Jobs
Guapito Chaylian	Profile	Edit	Cart	Orders	Bake Jobs
Luis Chaylian	Profile	Edit	Cart	Orders	Bake Jobs
Jayne Dowd	Profile	Edit	Cart	Orders	Bake Jobs
Merran Hall	Profile	Edit	Cart	Orders	Bake Jobs
Kat Lee	Profile	Edit	Cart	Orders	Bake Jobs
Mel lim	Profile	Edit	Cart	Orders	Bake Jobs
Leigh Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh2 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh3 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Leigh4 Poulier	Profile	Edit	Cart	Orders	Bake Jobs
Wendy Poulier	Profile	Edit	Cart	Orders	Bake Jobs

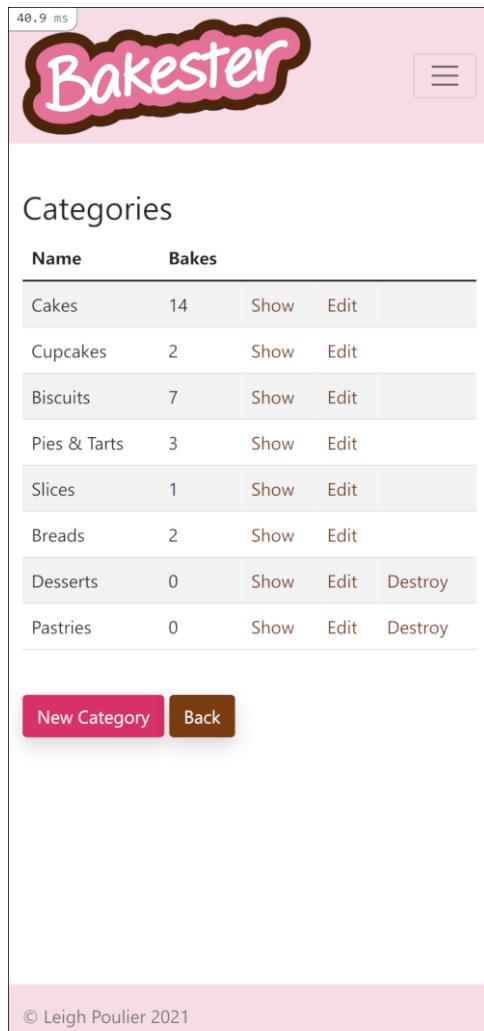
[Back](#)

© Leigh Poulier 2021

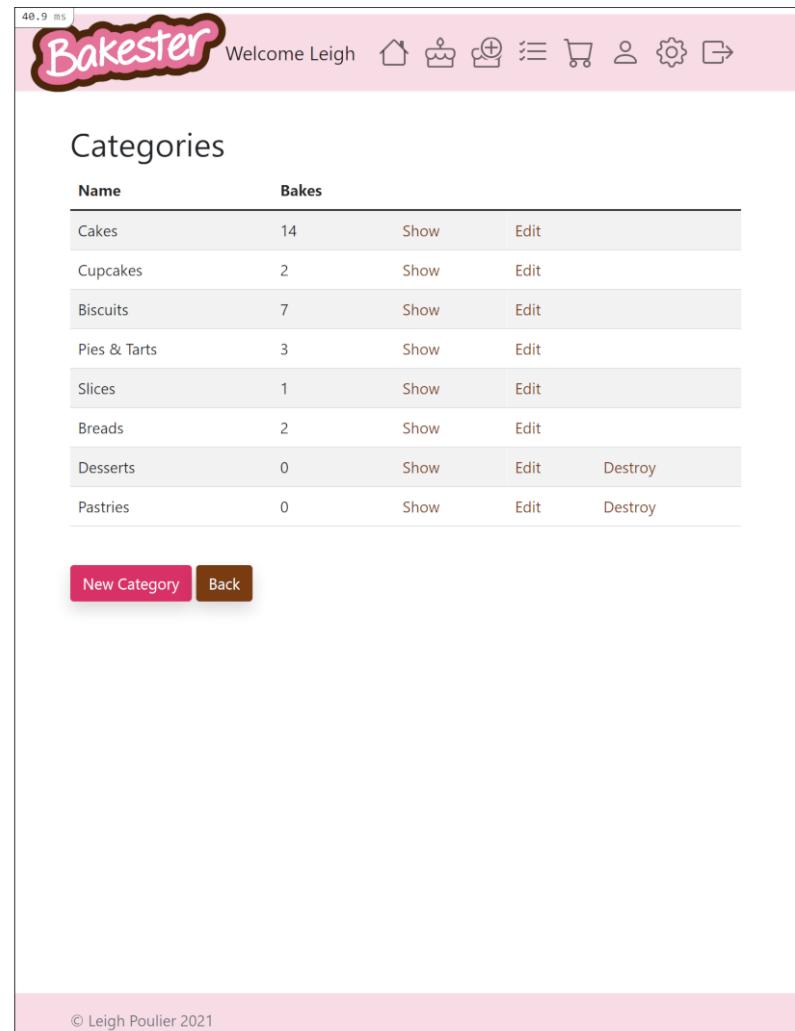
Users list on desktop.

## Categories List

From the Admin page, clicking on the "Categories" link opens the categories list, from which categories can be edited and deleted if they are empty.



A mobile screenshot of the 'Categories' list. At the top, there's a navigation bar with the 'Bakester' logo and a menu icon. Below it, the title 'Categories' is displayed. A table lists eight categories: Cakes (14 bakes), Cupcakes (2 bakes), Biscuits (7 bakes), Pies & Tarts (3 bakes), Slices (1 bake), Breads (2 bakes), Desserts (0 bakes), and Pastries (0 bakes). Each row has 'Show' and 'Edit' buttons. The 'Desserts' row also includes a 'Destroy' button. At the bottom, there are 'New Category' and 'Back' buttons, and a copyright notice: © Leigh Poulier 2021.



A tablet screenshot of the 'Categories' list. At the top, there's a navigation bar with the 'Bakester' logo, user name 'Welcome Leigh', and various icons. Below it, the title 'Categories' is displayed. A table lists the same eight categories as the mobile version. Each row has 'Show', 'Edit', and 'Destroy' buttons. The 'Desserts' row also includes an 'Edit' button. At the bottom, there are 'New Category' and 'Back' buttons, and a copyright notice: © Leigh Poulier 2021.

Categories list on mobile and tablet.

The screenshot shows a web application interface for 'Bakester'. At the top, there's a navigation bar with icons for home, orders, products, categories, users, settings, and help. The title 'Bakester' is prominently displayed in a large, pink, bubbly font. Below the title, the word 'Categories' is centered. A table lists eight categories: Cakes, Cupcakes, Biscuits, Pies & Tarts, Slices, Breads, Desserts, and Pastries. Each row includes a 'Name' column, a 'Bakes' column (containing the count), and 'Show', 'Edit', and 'Destroy' buttons. At the bottom left are 'New Category' and 'Back' buttons.

Name	Bakes	Show	Edit	Destroy
Cakes	14	Show	Edit	
Cupcakes	2	Show	Edit	
Biscuits	7	Show	Edit	
Pies & Tarts	3	Show	Edit	
Slices	1	Show	Edit	
Breads	2	Show	Edit	
Desserts	0	Show	Edit	Destroy
Pastries	0	Show	Edit	Destroy

Categories list on desktop.

## All Bakes

The "All Bakes" link takes the administrator to the previously discussed Bakes list, but includes some admin only search form parameters - enabling the inclusion of hidden bakes.

## All Orders

227.4 ms

Order No	User	Submitted?	Bakes	Order Total
1	Leigh Poulier	Y	1	\$300.00
2	Leigh Poulier	Y	1	\$2.00
3	Leigh Poulier	Y	1	\$2.00
4	Leigh3 Poulier	Y	3	\$77.00
5	Wendy Poulier	Y	1	\$10.00
6	Leigh2 Poulier	Y	1	\$30.00
7	Jayne Dowd	Y	3	\$114.00
8	Leigh3 Poulier	Y	6	\$60.00
9	Leigh2	Y	1	\$305.00

227.4 ms

Order No	User	Submitted?	Submitted	Bakes	Order Total
1	Leigh Poulier	Y	11 Nov 2021	1	\$300.00
2	Leigh Poulier	Y	11 Nov 2021	1	\$2.00
3	Leigh Poulier	Y	11 Nov 2021	1	\$2.00
4	Leigh3 Poulier	Y	14 Nov 2021	3	\$77.00
5	Wendy Poulier	Y	14 Nov 2021	1	\$10.00
6	Leigh2 Poulier	Y	14 Nov 2021	1	\$30.00
7	Jayne Dowd	Y	14 Nov 2021	3	\$114.00
8	Leigh3 Poulier	Y	14 Nov 2021	6	\$60.00
9	Leigh2 Poulier	Y	14 Nov 2021	1	\$305.00
10	Leigh Poulier	Y	14 Nov 2021	2	\$1,196.00
11	Merran Hall	Y	16 Nov 2021	3	\$228.00
12	Leigh2 Poulier	Y	16 Nov 2021	5	\$465.00
13	Leigh3 Poulier	Y	16 Nov 2021	1	\$250.00
14	Leigh3 Poulier	Y	16 Nov 2021	2	\$500.00
15	Guapito Chaylian	N		2	
16	Leigh4 Poulier	Y	22 Nov 2021	7	\$98.00
17	Leigh4 Poulier	Y	22 Nov 2021	2	\$108.00
18	Kat Lee	Y	24 Nov 2021	2	\$60.00
19	Mai Liisa	Y	24 Nov 2021	1	\$25.00

All Orders list on mobile and tablet.

227.4 ms

Order No	User	Submitted?	Submitted	Bakes	Includes	Order Total
1	Leigh Poulier	Y	11 Nov 2021	1	tres leches	\$300.00
2	Leigh Poulier	Y	11 Nov 2021	1	Fondant Cookies	\$2.00
3	Leigh Poulier	Y	11 Nov 2021	1	Fondant Cookies	\$2.00
4	Leigh3 Poulier	Y	14 Nov 2021	3	Number Birthday Cake, Fondant Cookies, Special Occasion Cookies	\$77.00
5	Wendy Poulier	Y	14 Nov 2021	1	Special Occasion Cookies	\$10.00
6	Leigh2 Poulier	Y	14 Nov 2021	1	Lemon Meringue Pie	\$30.00
7	Jayne Dowd	Y	14 Nov 2021	3	Lemon Meringue Pie, Chocolate Cake	\$114.00
8	Leigh3 Poulier	Y	14 Nov 2021	6	Special Occasion Cookies	\$60.00
9	Leigh2 Poulier	Y	14 Nov 2021	1	Baby	\$305.00
10	Leigh Poulier	Y	14 Nov 2021	2	L'amour, Baby	\$1,196.00
11	Merran Hall	Y	16 Nov 2021	3	Banana cupcakes, Chocolate Crackles, Basketball cake	\$228.00
12	Leigh2 Poulier	Y	16 Nov 2021	5	Gingerbread House, Number Birthday Cake, Lemon Meringue Pie, and 2 more	\$465.00
13	Leigh3 Poulier	Y	16 Nov 2021	1	Batman Kids Birthday Cake	\$250.00
14	Leigh3 Poulier	Y	16 Nov 2021	2	Batman Kids Birthday Cake	\$500.00
15	Guapito Chaylian	N		2	Banana cupcakes, Basketball cake	
16	Leigh4 Poulier	Y	22 Nov 2021	7	Arcoíris, Fondant Cookies	\$98.00
17	Leigh4 Poulier	Y	22 Nov 2021	2	Banana cupcakes	\$108.00

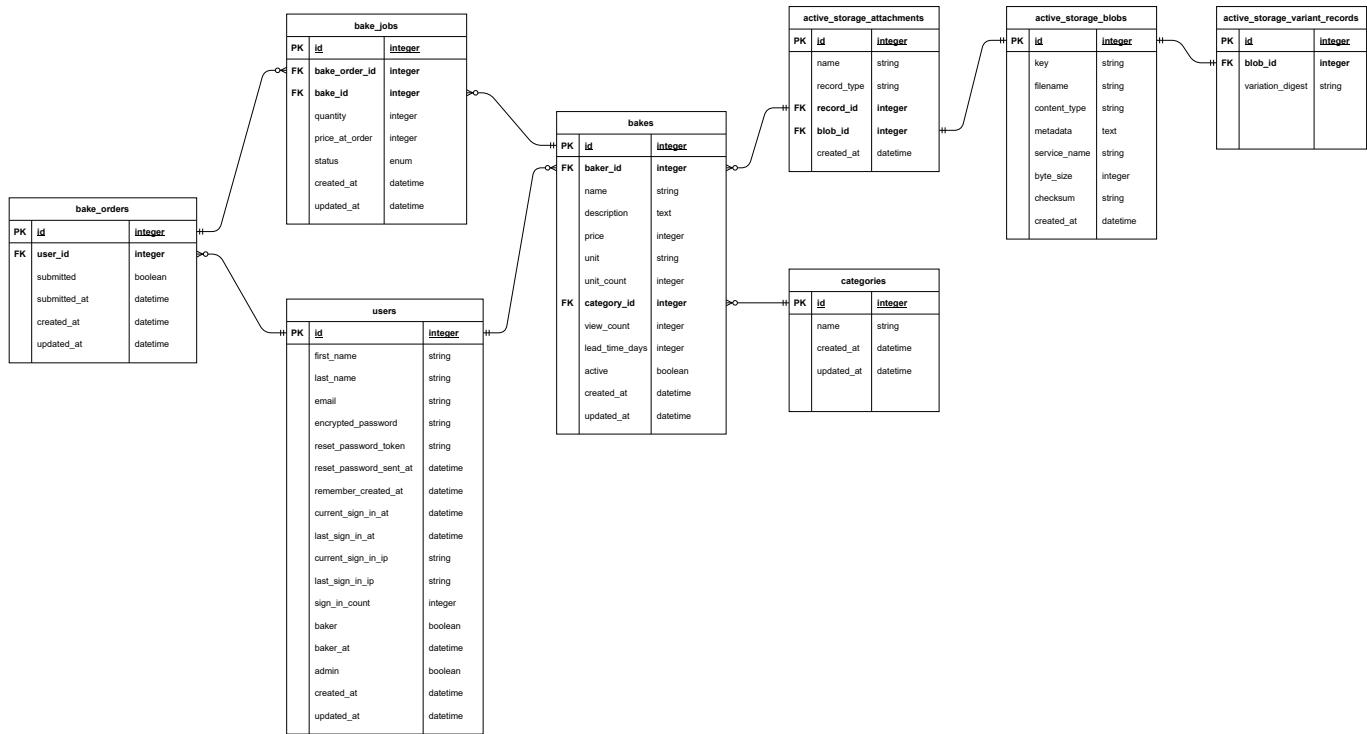
All Orders list on desktop.

Under the hood

## Database

The database underneath Bakester is provided by PostgreSQL. The main abstractions stored in the database are **users**, **categories**, **bakes**, **bake\_orders** and **bake\_jobs**. Additional tables "active\_storage\_attachments", "active\_storage\_blobs" and "active\_storage\_variant\_records" were automatically added for the storage of information related to the images of bakes, which are handled as ActiveStorage attachments.

## Entity Relationship Diagram (ERD)



[Bakester Entity Relationship Diagram - Click here for a larger version](#)

## Users

A user is a person who uses the site and holds an account. Information relating to the user is used to manage the sign up and login process and identify who is responsible for the various other entities of the app.

The users table stores information relating to the users of the database. Most of the fields in the users table were created by the Rails engine "Devise", although some extra fields have been added manually, such as the "admin" boolean and the "baker" boolean, which define the users status in those regards. The "boolean\_at" field keeps a record of the date and time that the user became a baker. Fields for "first\_name" and "last\_name" were created in addition to the default email so that more personalisation was possible in the app's views.

Foreign Keys links to the users "id" column are used in the orders table, to define the owner of an order, and the bakes table, to define the "baker" of a bake. Each user has many orders, and is the baker of many bakes.

## Categories

All bakes are placed in to a category when they are created. The list of categories is available for selection in the bake create/update form, but only admin users are able to create, update, or delete categories. Only the "name" field is used throughout the app's views.

A Foreign Key field linking to the categories "id" column are present on the bakes table, which defines which category a bake belongs to, and defines a one-to-many relationship. Each category has many bakes.

## Bakes

A bake is the product of the web app. It is a baked item that is offered for sale by a baker for other users to order.

The bakes table holds the records of all the bakes that bakers have uploaded to the site for sale. It includes many fields to describe the bake, such as "name", "description" and "price".

Some bakes may include multiple items, for example a batch of 12 cupcakes, and so additional fields are used to store the "unit\_count" (12) and "unit", (the name of the basic unit of the bake, eg "cupcake"). Additional tooltips are provided in the bake create/edit form to explain these fields, as experience has shown them not to be intuitive.

A "view\_count" field keeps track of how many times a bake has been viewed, so that it's popularity can be gauged by the baker or customers. The field "lead\_time\_days" records how many days a baker requires in order to create a bake, and gives the customer an idea of how much time needs to be allowed following an order. It also allows the customer to filter out bakes that cannot be provided in the available time before an event.

The "baker\_id" field is a foreign\_key field which relates to the "users" table, and defines the "baker" (user) of this bake. It is used to define who may edit the bake, and where bake jobs containing this bake are sent. The relationship is one-to-many. One baker (user) can have many bakes.

The "category\_id" field defines which category the bake belongs to, and makes a "one-to-many" relationship. Each bake has one category, but each category has many bakes.

A Foreign Key field linking to the bakes "id" column is used in the bake\_jobs table to define which bake is being requested as part of a bake job. This defines a one-to-many relationship, where each bake\_job relates to one bake.

## Bake Orders

A "Bake Order", or an order(\*) is created by a user when they create a cart, and proceed through checkout. It contains a list of bake\_jobs, which are the tasks that are sent to bakers for production.

A cart is a special unfinshed bake order, of which there should only be one per user if any. The field "submitted" defines whether a cart has been transformed into a bake order, although both are stored in the same "bake\_orders" table. A "submitted\_at" field records a time stamp of when this happened.

This restriction is only enforced in the logic of the [BakeOrder](#) controller, and i could not find a way to ensure only one cart existed per user inside the database, because the same table is used to store multiple past orders.

The "user\_id" field is a foreign key referring to the "users" table, and relates the bake order to a user through a one-to-many relationship. One user can have many bake orders.

A Foreign key field to the bake orders "id" column is used in the "bake jobs" table, which defines which bake\_order is the "parent" of the bake job. This defines a one-to-many relationship. All bake\_jobs are related to one order, but each bake\_order contains many bake\_jobs.

- Initially this table and the model was name "order" but it was found that it conflicted with the built-in Rails ActiveRecord query method "order" which is used to define the sort column and direction of an ActiveRecord relation.

## Bake Jobs

A bake\_job is a line item of a Bake Order. Each bake order can contain many different bakes each provided by a different baker. A bake\_job encapsulates the information required to be sent to a baker to fulfil an order, which bake, how many etc.

A "bake\_job" defines the details of a particular transaction, and is a join table between the "bake\_orders" and "bakes" table. Each bake\_job contains foreign key fields for a bake\_order (one-to\_many) and a bake (one\_to\_many). Through these two relationships there is therefore a many-to-many relationship between bakes and bake orders. Each bake\_order can contain many bakes, and each bake may appear in many bake orders.

The user checks out an entire order at once, which may consist of many bake\_orders, each of which has its own baker and status to be managed by that baker. The "status" field is updated by bakers when they manage their bake jobs, and the customer can view that bake job's status as part of their order. A field "quantity" tracks if multiple of a particular bake were ordered in one go, and the field "price\_at\_order" is used to keep track of the price of the bake at the time of the transaction, which differ from the current price of the bake.

## Active Storage

Each bake has an image, but these are not stored in the bakes table.

The "active\_storage\_blobs" table stores information about attached files, including size, a checksum, content type, original filename and the name of the service storing the attachment.

The table "active\_storage\_attachments" records information related to the relationship between the bake and the "storage\_blob". It is related to the bakes table through a polymorphic join. The "record\_type" and "record\_id" strings define the relationship, with the "record\_type" equal to "Bake" for this database, and the "record\_id" being a foreign key field defining a one to many relationship with bakes. Each bake could have multiple attached images, although in my implementation there is only one.

The "active\_storage\_attachments" table also contains a foreign key field for "blob\_id" which creates a one-to-many relationship to the "active\_storage\_blobs" table. This actually makes the "active\_storage\_attachments" table a join table, and the relationship from bakes to active\_storage\_blobs through active\_storage\_attachments is many\_to\_many. This means that each bake can refer to many blobs, and each blob can be related to many bakes, which helps to ensure that there is no need for duplication of attached files.

I have no idea what the "active\_storage\_variant\_records" table is for. It is empty in my database, but it also refers to the blobs table with a foreign key "blob\_id", so I assume it is some sort of record of variation or

variants to each image.

## Models

The same 5 main abstractions (users, categories, bakes, bake orders and bake\_jobs) are implemented as models in the Rails MVC (model, view, controller) paradigm.

Although the foundation of the associations between them is as discussed above in the database relationships, there are many additional details implemented in each model definition.

### User model

The User model, although created automatically with devise, has been customised to add additional functionality required by the Bakester app.

Mirroring the database relationships, the User model has many bakes, and many bake\_orders.

A "bake\_jobs" association is defined which relates a user to bake\_jobs through the bake\_orders table. This means that it is possible to quickly determine who was the origin of a bake\_job.

In addition an association called "carts" is defined, related to the "BakeOrder" model. It returns only bake orders which are not submitted, related to a particular user. There is nothing stopping an error in logic inserting multiple carts for one user into the bake\_orders table, although there SHOULD be only one. When ever this scope is used in practice, it is chained with .first, so that only one cart is returned. The "carts" association also employs eager loading to include references to the component bake\_jobs and bakes as part of the order.

A "bakers" scope filters the users table to only those users which are bakers (as defined by the "baker" boolean column).

An "eager\_loading" scope aims to speed up query of the users table, and includes bakes, categories, and image attachments. When viewing a user profile, all their bakes are displayed and this scope ensures limits the number of queries required to build that view.

Additional helper methods are required which join the users first\_name and last\_name columns into a "full\_name", and return the first cart (from the carts scope), and return the size of the user's cart.

### Category model

The category model is comparitively simple, and defines the expected "has many" association with the bakes model. This means each category relates to many bakes.

One helper method "filter\_options" builds the list of categories that is displayed in various select boxes within the app, and prepends an "All" option.

### Bake model

The Bake is rather more involved.

Some constants are defined which govern the behaviour of the image resizing feature, and also hard cord the fields by which the bakes listing may be sorted.

Each Bake belongs to a user in two distinct ways. A User may be associated to a Bake through the BakeJobs and BakeOrders model, as "a user who has ordered this bake". A User can also be related directly to the User model, as "a user who owns this bake". Because user is ambiguous, the term "baker" is used to define the owner of a bake, and the association is defined in the model as "belongs\_to :baker, class\_name: "User" to avoid this potential conflict.

Each Bake is associated to a Category with a "belongs\_to" association. This means each bake "has one" category

The Bake model has "has\_many" associations on the BakeJob model, and also the BakeOrder model through the BakeJob model, as detailed above. It also "has\_one\_attached" :image, although a future feature may be to be able to add multiple images.

Many scopes are defined to implement the search / filter / sort feature of the bakes index page. A few helper scopes are defined, which each implement one aspect of the feature, such as text searching, filtering on price, category, or lead time. The text search scope actually splits a multi-word search up into an array of strings so that the words can be AND'd, which means that the words don't have to appear in the entered order.

A very large and complicated scope builds up an ActiveRecord relationship with all these helper scopes, which returns the data set requested in the form.

## Bake Order model

The BakeOrder model implements a "belongs\_to" association with the User model, which means each BakeOrder has one user who created it.

There is a "has\_many" association in the BakeOrder model with the BakeJob model, because each bake\_order contains many bake\_jobs. If a bake\_order is destroyed (or a cart is emptied), related bake\_job model objects will be deleted also, due to the "dependent destroy" addition.

Each Bake Order also "has\_many" bakes, through the "bake\_jobs" join table. This means it is easy to access the list of bakes contained within the bake\_jobs which are themselves contained within a bake\_order.

Each Bake Order also "has\_many" bakers, as they are associated with their bakes. This is a double "through" "many-to-many" relationship, where the BakeOrder model is associated to the User model (as baker) through the BakeJob model and the Bake model.

An eager loading scope is used to improve load times for views accessing a bake order, and allows the related User, Baker and BakeJob information to be loaded in one query.

Scopes "submitted" and "carts" are mutually exclusive, and restrict the returned set of bake\_orders to those which are carts (not submitted) or submitted bake orders.

## Bake Job model

The BakeJob is a joining model, with "belongs\_to" associations to both the BakeOrder model and Bake model. Each BakeJob refers to one Bake and one BakeOrder, which implements the many-to-many relationship between bake\_orders and bakes.

The BakeJob model also defines an association to the User model, through the Bake model. This relates the baker of a bake to the bake\_job. The BakeJob could also define an association to the User model through

BakeOrder, as the customer who created the job, but this was removed as it wasn't actually required for the functionality of the app.

An enum defines the available values of the "status" field, relating a hash of values with integers which are stored in the database. This gives many helper methods to the model, including one for each enumerated value. A few additional scopes are defined to enable more general filtering of bake\_jobs based on their status, which can be seen on the Bake Job index view.

Eager loading is also implemented on the model, so that queries may include related fields from the BakeOrder model, the Bake model and its related User model (as baker).

## Design Process

### User Stories

When designing the website and its functions, the following user stories were used to clarify the required functionality of the website. In these definitions, an "order" is what the customer creates, and a "job" is the task that the baker receives.

### Customers

As a **Customer**...

I want to...	So that i can...
Browse all bakes	see what bakes are available
Search and filter bakes	drill down to the types of bakes i am looking for
View a bake	see details about the bake and its description
Sign up for a new account	make orders for bakes
Log in to my account	see details of my past orders
Change my password	keep my account secure
Edit my account details	correct/update my name or email
Add a bake to my cart	build up a basket of bakes to order
Checkout	place an order with a baker
See my orders	keep track of the progress of my orders
View a User	see details of a user, in particular a baker's bakes
Become a baker	sell my home bakes on Bakester

### Bakers

As a **Baker**...

I want to...	So that i can...
--------------	------------------

I want to...	So that i can...
Do everything a customer can do	still act as a customer
Add a bake listing	start to receive customer orders for that bake
Add an image to a bake	make the bake more appealing to customers
Edit bake	update or correct details of my bakes
Hide a bake	temporarily prevent customers from placing an order
Delete a bake	permanently remove a bake
View my bakes	keep track of what bakes i am offering
Be notified of new jobs	i am aware of new jobs as soon as customers checkout
View my jobs	keep track of the customer orders that i need to fulfil
Update Job status	Track which jobs i have completed, and which are still pending

## Admin

As an **Admin**...

I want to...	So that i can...
View categories	know what categories are provided for bakers to classify their bakes
Add a category	bakers can use it to classify their bakes
Edit a category	correct / update a category name
Delete a category	permanently remove a category from the site
Edit a bake	moderate content provided by bakers
Delete a bake	permanently remove a bake
See all users	know about all the users signed up on the site
Edit a user	change a user's name or email
Delete a user	close a user's account
See all orders	know about all the orders placed on the site
See the orders from a customer	investigate problems on behalf of a customer
See all jobs	know about all the jobs assigned to bakers
See all jobs for a particular baker	investigate problems on behalf of a baker

## Project Management

To manage the creation of this site, a [trello board](#) was implemented. Each user story above and many other design tasks were moved through four stages:

Stage	Description
Backlog	A place for all tasks until they are prioritised
Current	Tasks that are prioritised to be completed in the short term
Work in Progress	The tasks that are currently being worked on
Complete	Tasks that are finished and active on the site

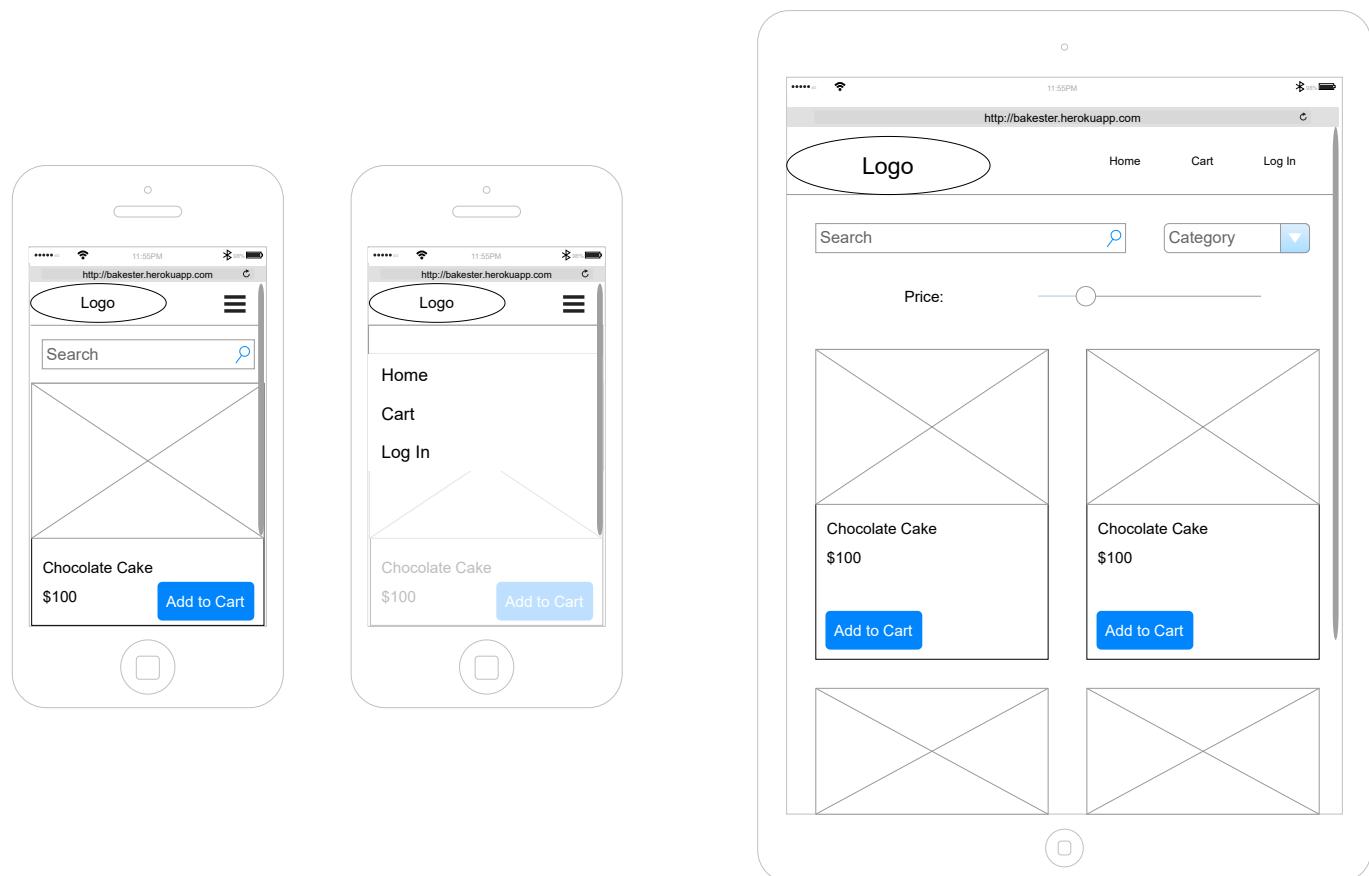
Some features are not as yet implemented, and remain on the backlog, such as comments and ratings for users and bakes.

## Wireframes

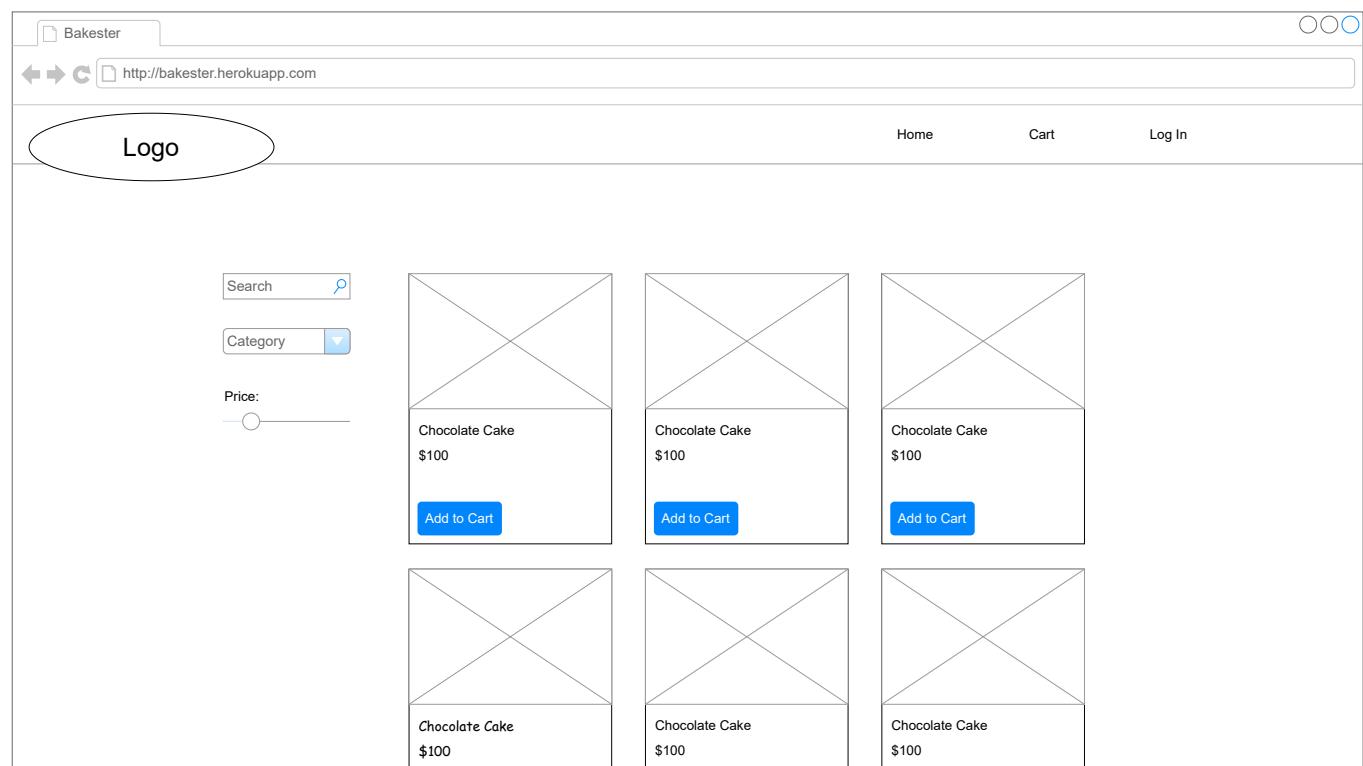
The following wireframes were created to crystalise my vision of the layout of various core pages of the web site. We will follow the journey of a new user through first access, signup, use of the site as a customer, and then use of the site as a baker.

## Main page

When the user first accesses the site, they will be presented with a screen containing all of the bakes that are available on the site. Sign up or log in is not required to view this list. A grid of cards is presented, each linking to a bake's own page. A search / filter / sort form is provided to customise the listing.



Main bakes page - mobile and tablet

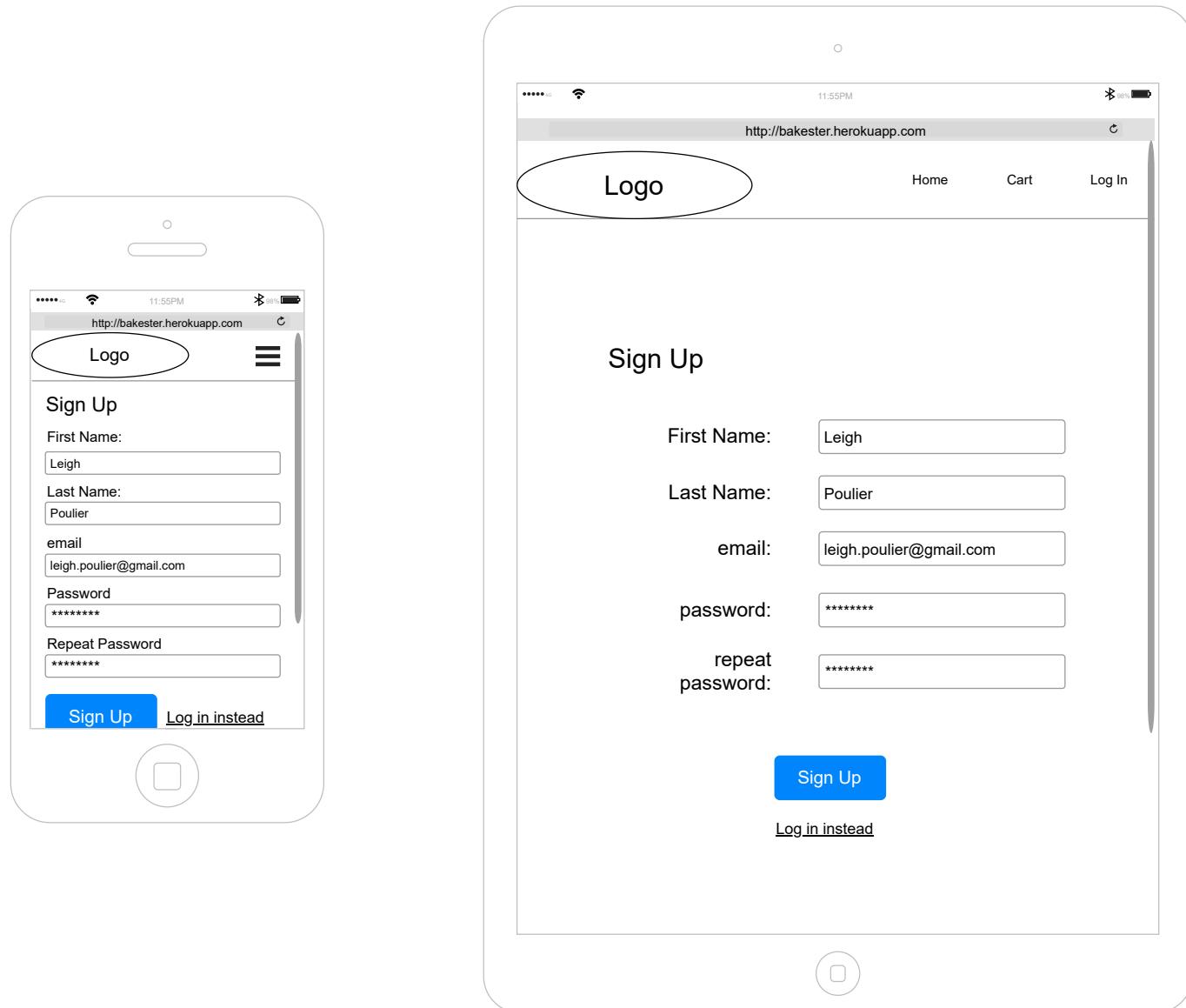


Main bakes page - desktop

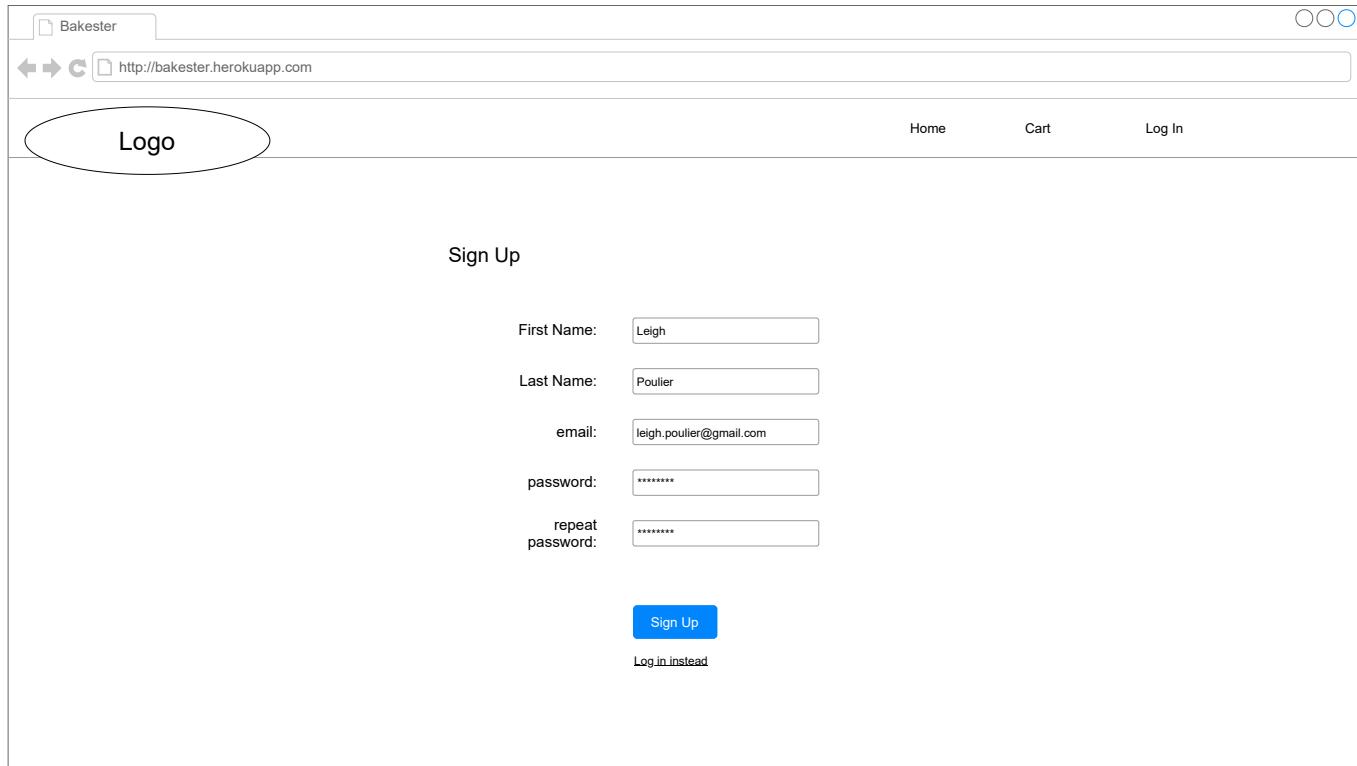
To navigate the site on desktop and tablet, a navigation bar is provided at the top of the page with a horizontal list of links. In the mobile view, the navigation bar is compacted and a hamburger button replaces the horizontal list, and when activated displays a drawer containing the same links.

## Sign Up

After deciding to become an active user of the site, the user can sign up



Main bakes page - mobile and tablet



Main bakes page - desktop

## View Bake

For more detail on a particular bake, a user can click on it and access a detail page

The diagram illustrates the "View Bake" feature across two devices: a mobile phone and a tablet. Both devices display the same detailed view of a "Chocolate Cake" listing.

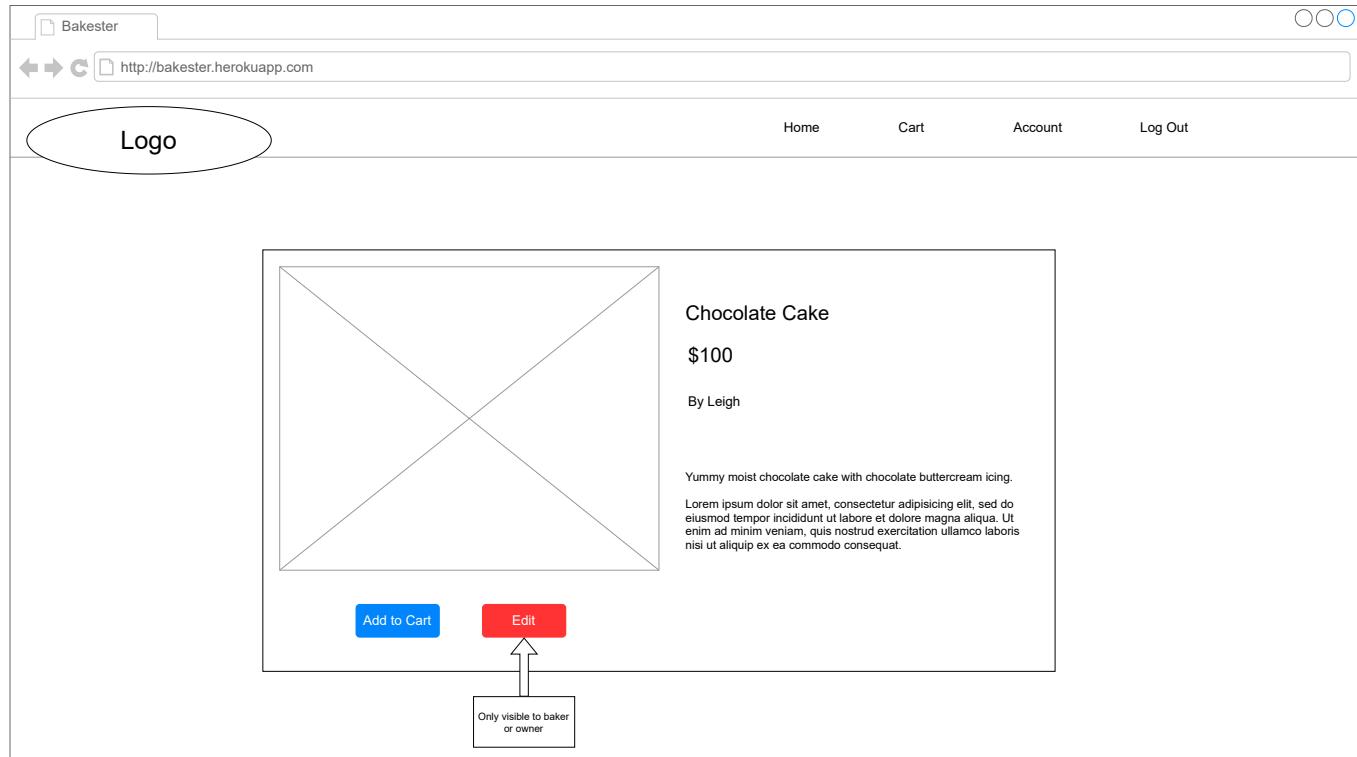
**Mobile Phone View:**

- The top status bar shows signal strength, battery level, and time (11:55PM).
- The browser header shows the URL <http://bakester.herokuapp.com>.
- The main content area features a large "X" placeholder image for the cake image.
- The listing includes the name "Chocolate Cake", price "\$100", author "By Leigh", and a blue "Add to Cart" button.
- A descriptive text box states: "Yummy moist chocolate cake with chocolate buttercream icing."

**Tablet View:**

- The top status bar shows signal strength, battery level, and time (11:55PM).
- The browser header shows the URL <http://bakester.herokuapp.com>.
- The main content area features a large "X" placeholder image for the cake image.
- The listing includes the name "Chocolate Cake", price "\$100", author "By Leigh", and a blue "Add to Cart" button.
- A descriptive text box states: "Yummy moist chocolate cake with chocolate buttercream icing."
- An additional red "Edit" button is present below the listing.
- A callout box indicates that the "Edit" button is "Only visible to baker or owner".

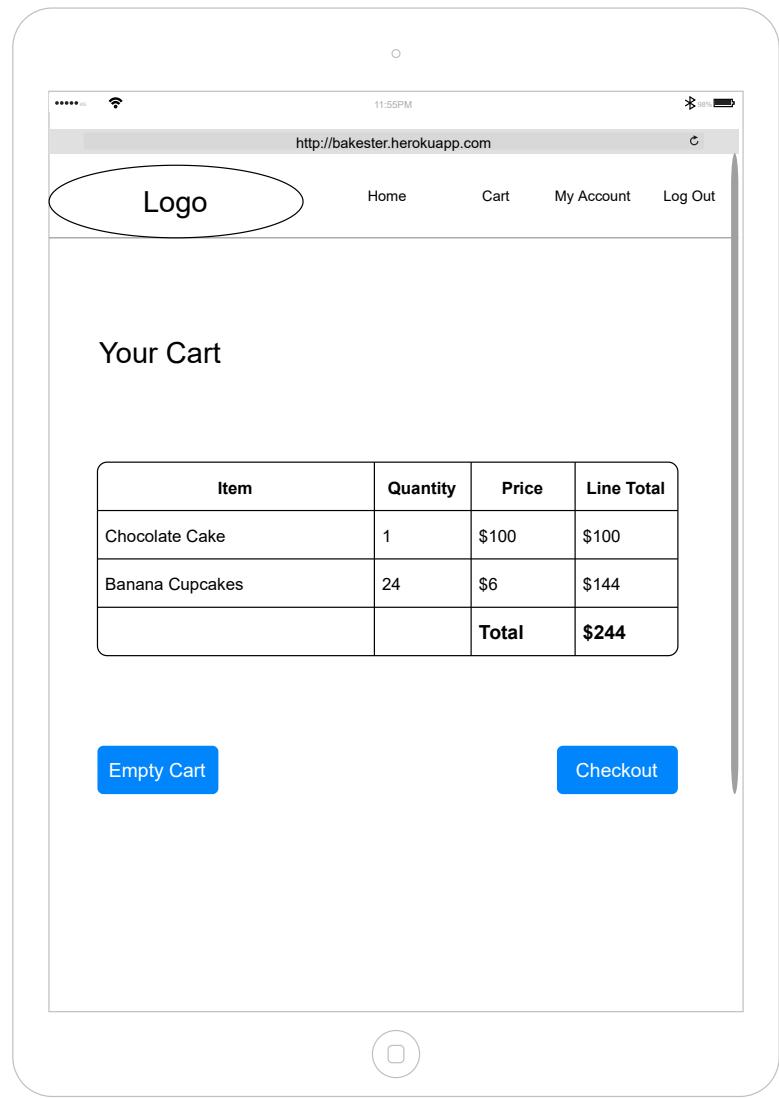
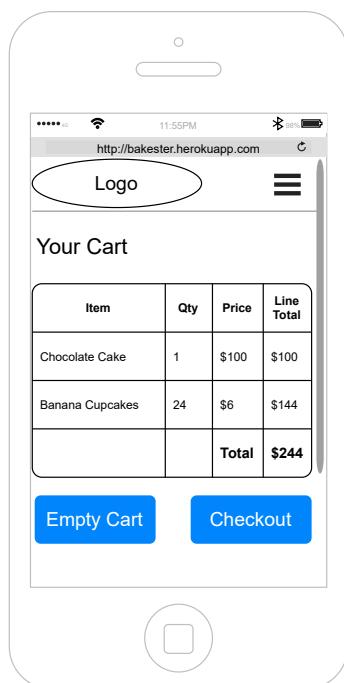
Bake page - mobile and tablet



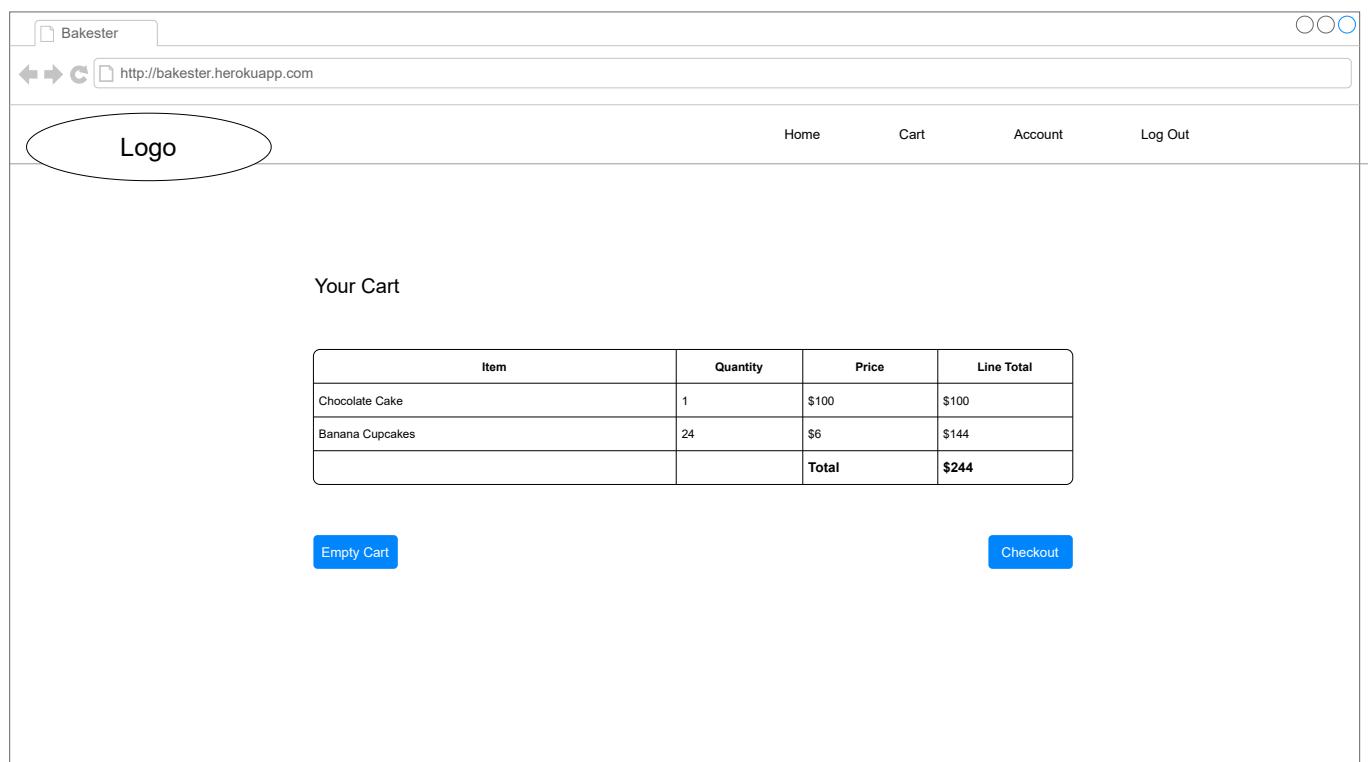
Bake page - desktop

## Cart

A bake can be added to cart from either the card on the bakes list, or a bake detail page. The cart page shows the bakes currently in the users cart.



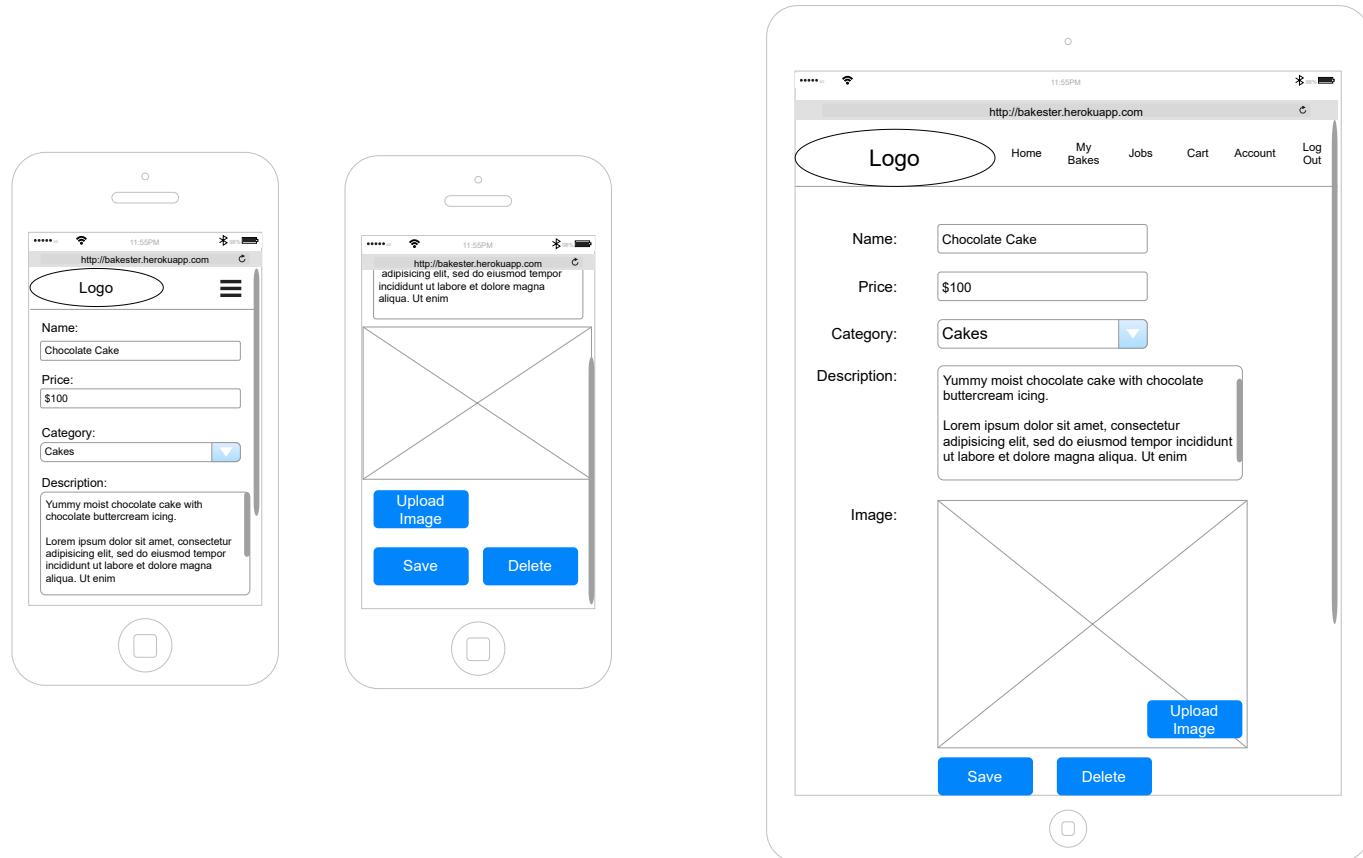
Cart page - mobile and tablet



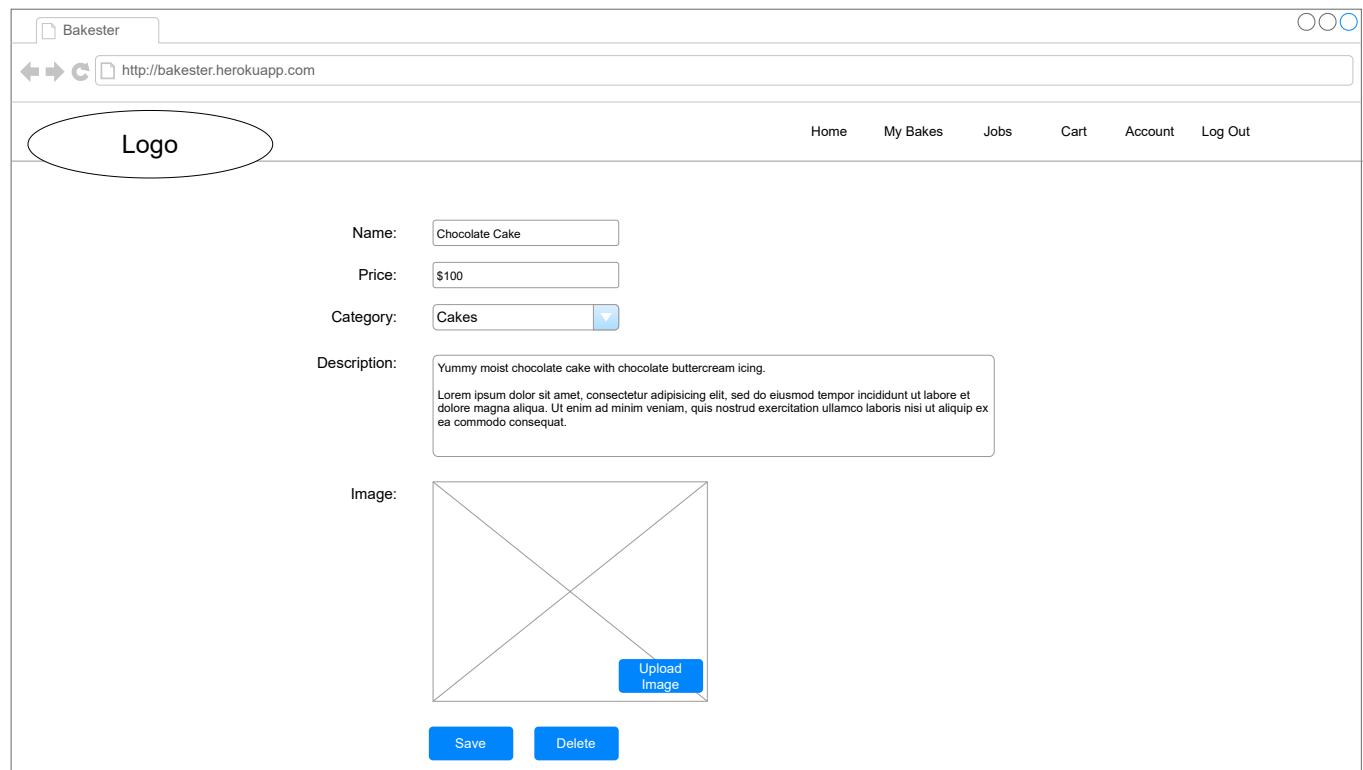
Cart page - desktop

## Edit Bake

After upgrading their account to a baker, the user is now classed a "baker" and is able to create or edit their own bakes. A form is provided including functionality to upload an image.



Edit Bake page - mobile and tablet



Edit Bake page - desktop

The navigation menu has changed and more options are available to a baker to manage their bakes listed on the site

## Bake Jobs

Once a user has made an order for a baker's bake, the baker is notified they have a new bake job. The baker can view the list of their bake jobs to manage their tasks.

**Mobile View (iPhone):**

Logo

My Jobs

Customer	Bake	Status	Total
Fred Flintstone	Chocolate Cake	Received <a href="#">edit</a>	\$100
Scrooge McDuck	Banana Cupcakes	Processing <a href="#">edit</a>	\$72
Spongebob Squarepants	Choc Chip Cookies	Delivered <a href="#">edit</a>	\$48

[Back](#)

**Tablet View (Tablet):**

Logo

Home My Bakes Jobs Cart Account Log Out

My Jobs

Customer	Submitted	Bake	Status	Unit Price	Quantity	Total
Fred Flintstone	20 Nov 2021	Chocolate Cake	Received <a href="#">edit</a>	\$100	1	\$100
Scrooge McDuck	21 Nov 2021	Banana Cupcakes	Processing <a href="#">edit</a>	\$6	12	\$72
Spongebob Squarepants	24 Nov 2021	Choc Chip Cookies	Delivered <a href="#">edit</a>	\$4	12	\$48

[Back](#)

Bake Jobs page - mobile and tablet

Logo

Home My Bakes Jobs Cart Account Log Out

My Jobs

Customer	Submitted	Bake	Status	Unit Price	Quantity	Total
Fred Flintstone	20 Nov 2021	Chocolate Cake	Received	\$100	1	\$100
Scrooge McDuck	21 Nov 2021	Banana Cupcakes	Processing	\$6	12	\$72
Spongebob Squarepants	24 Nov 2021	Choc Chip Cookies	Delivered	\$4	12	\$48

Back

Bake Jobs page - desktop

## Edit Bake Job

A baker can edit the status of their bake jobs to keep track of what is pending, and also keep the customer updated.

Logo

Home My Bakes Jobs Cart Account Log Out

Edit Bake Job

Bake Job No	1
Order No	2
Customer	Fred Flintstone
Baker	Leigh Poulier
Submitted	20 Nov 2021
Bake	Chocolate Cake
Quantity	1
Unit Price	\$100
Total	\$100
Status	Processing

Update

Logo

Home My Bakes Jobs Cart Account Log Out

Edit Bake Job

Bake Job No	1
Order No	2
Customer	Fred Flintstone
Baker	Leigh Poulier
Submitted	20 Nov 2021
Bake	Chocolate Cake
Quantity	1
Unit Price	\$100
Total	\$100
Status	Processing

Update

Edit Bake Job page - mobile and tablet

The screenshot shows a web application interface for managing bake jobs. At the top, there's a header with a logo (an oval containing the word "Logo"), navigation links for "Home", "My Bakes", "Jobs", "Cart", "Account", and "Log Out", and a URL bar showing "http://bakester.herokuapp.com". Below the header is a title "Edit Bake Job". The main content is a form with the following data:

Bake Job No	1
Order No	2
Customer	Fred Flintstone
Baker	Leigh Poulier
Submitted	20 Nov 2021
Bake	Chocolate Cake
Quantity	1
Unit Price	\$100
Total	\$100
Status	Processing

At the bottom right of the form is a blue "Update" button.

Edit Bake Job page - desktop

# README

---

This README would normally document whatever steps are necessary to get the application up and running.

Things you may want to cover:

- Ruby version
- System dependencies
- Configuration
- Database creation
- Database initialization
- How to run the test suite
- Services (job queues, cache servers, search engines, etc.)
- Deployment instructions
- ...

Images may be attached to bakes by bakers when they create or update a bake. The Rails controller action which handles this first resizes this image to maximum of 500x500 pixels, to ensure that large images do not get uploaded to the amazon servers, and to ensure fast download speeds when images are requested by browsers. The resizing is handled by a ruby gem called "MiniMagick".