Create a flash sales site where a customer saves more when they shop more.

## Users signup - sign-in:

Our site would have two types of users, customers & admins. Admin is a kind of customer with extra privileges. Admin can do anything a customer can do, and also can perform administrative tasks from the admin section of the site.

- Users can signup by providing their valid details
    - Name, email, password
    - On signup, the user will receive a verification email as other sites do
    - Forgot password and remember me functionality
    - Only verified users can access the system
- Admin can be created by a rake task “rake admin:new”, this rake task will prompt the user to enter required valid details from the console.
    - When admin is created no verification email is required. His account is verified by default.

## Admin section:

The admin section of the site is namespaced under /admin and will be accessible to admin users only.

### Deals:

- Admin should be able to create deals by specifying the title, description, price, discounted price, quantity, and publish date. He should also be able to upload deal images(use paperclip gem).
- While creating/updating a deal, the admin can upload multiple images. Use the nested form with the “Add More” link to add multiple images.
- By default, all deals will be unpublishable.
- A deal should be publishable if:
    - It has all the required attributes.
    - It has min 2 images
    - Max one other publishable deal is scheduled to publish on the same date, [Max two deals can be scheduled to be published on the same date]
    - Quantity is > 10
    -
    - .
- Admin should have an option to check the deal’s publishability from the deal’s details page
- A publishability check is also done whenever a deal is updated.
- The published date can not be changed within 24 hours of the deal going live
- Once a deal is over or published, its publish date can not be changed.
- Admin should be able to specify %tax applicable on each deal, by default null.
    - Can be 0 to 28 %
    - The deal should not be publishable if the tax is null/blank

### Scheduled Jobs

- The deal will be live for 24 hours only, and the next new deal will be published at 10 AM IST. There will be a scheduled job running which will run a rake task to publish new deals and unpublish old deals at 10AM everyday.

## Store Front:

- The home page will display the live deals for the day with details.
- The deal image should be shown as an image slideshow
- If by any chance there is no deal scheduled for the day, show the most recent two deals. Since these are past deals so can not be purchased
- Only logged-in users can buy a deals
- One user can buy one and only one qty of a deal
- Deals are sold at the discounted price
    - Additional loyalty discount also applies to returning customers, as:
        - If it is the user's first order, he gets no additional discount and buys at discounted price
        - If it is the user's second order, he should get an additional 1% discount over and above the discounted price
        - Similarly, for the third order, he should get 2% and so on. Max up to 5%.
- Checkout should be done by using stripe checkout
- While checkout we'll take the user's shipping addresses or the user can choose from previous addresses. The last used address is selected by default.
- Since there can max 2 deals published for a day, users can order both in one order or can place two orders one for each deal
- On successful checkout, the user should receive an email with order information
- If we run out of qty, show the deal as sold out. Make sure we don't oversell.
- There should be an option to browse past deals
- The user should be able to see his orders and order status under the my orders' section
- When the user leaves the page open, and a deal is not available for purchase anymore, then the page should be updated accordingly(Ajax polling)

## API

- Expose public JSON API for Live and Past deals
    - domain.com/api/deals/live
    - domain.com/api/deals/past
- Expose private API for MyOrders. Generate and make use of the User's auth_token.
    - Every user will have a unique API auth token.
        - Create a user verification
        - For existing users, make a rake task to generate a such token
        - The user can see his token from the profile page
    - localhost.com/api/myorders?token=fdsafdsjflkdsajf

## Admin section:

- Admin should be able to browse/update user records.
- Admin can browse orders, search orders by user’s email address
- Admin can mark an order as delivered or canceled, the user gets notified accordingly
    - Including payment info and refund info
    - When marking canceled, the amount should be refunded
- Few reports:
    - Total revenue generated by a deal
    - Potential of a Deal
        - Minimum
        - Maximum
    - Top spending customers
        - In a date range

----

## Additional functionality:

- Users can cancel an order while the deal is live and not expiring in the next x minutes. Qty in the order is back to the deal. Once a user cancels an order he can not place another order for the same deal.
    - Users can cancel orders from API also.