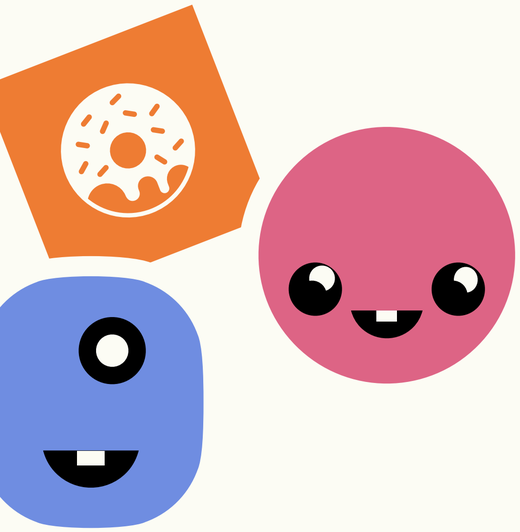
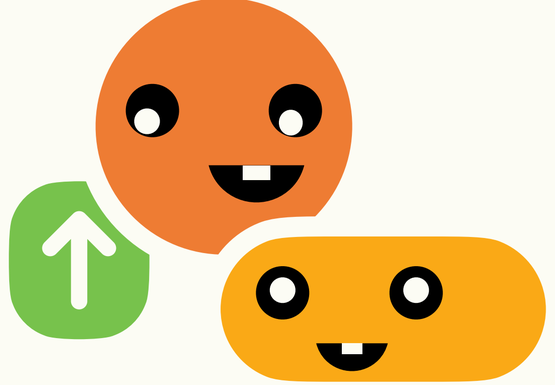
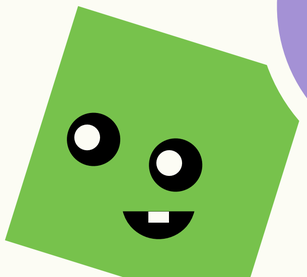


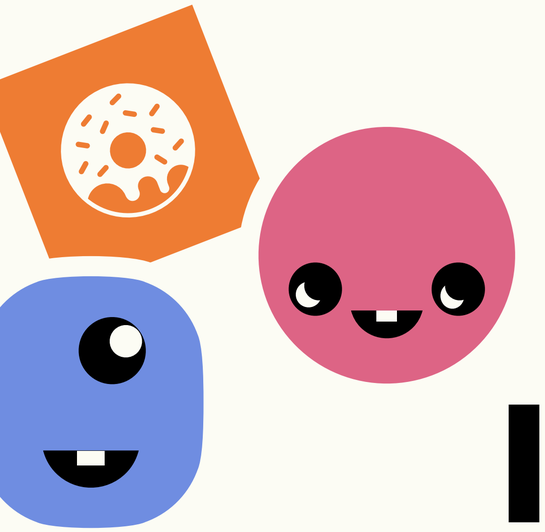


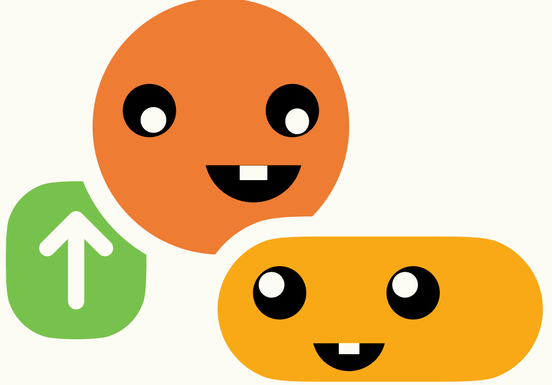
LEVELING UP FROM PLANNING TO PRODUCTION



@thomascountz

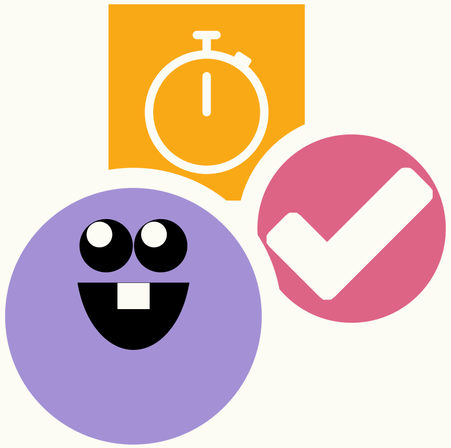

zendesk

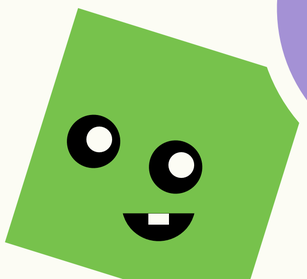




I'M AN INDIVIDUAL CONTRIBUTOR!

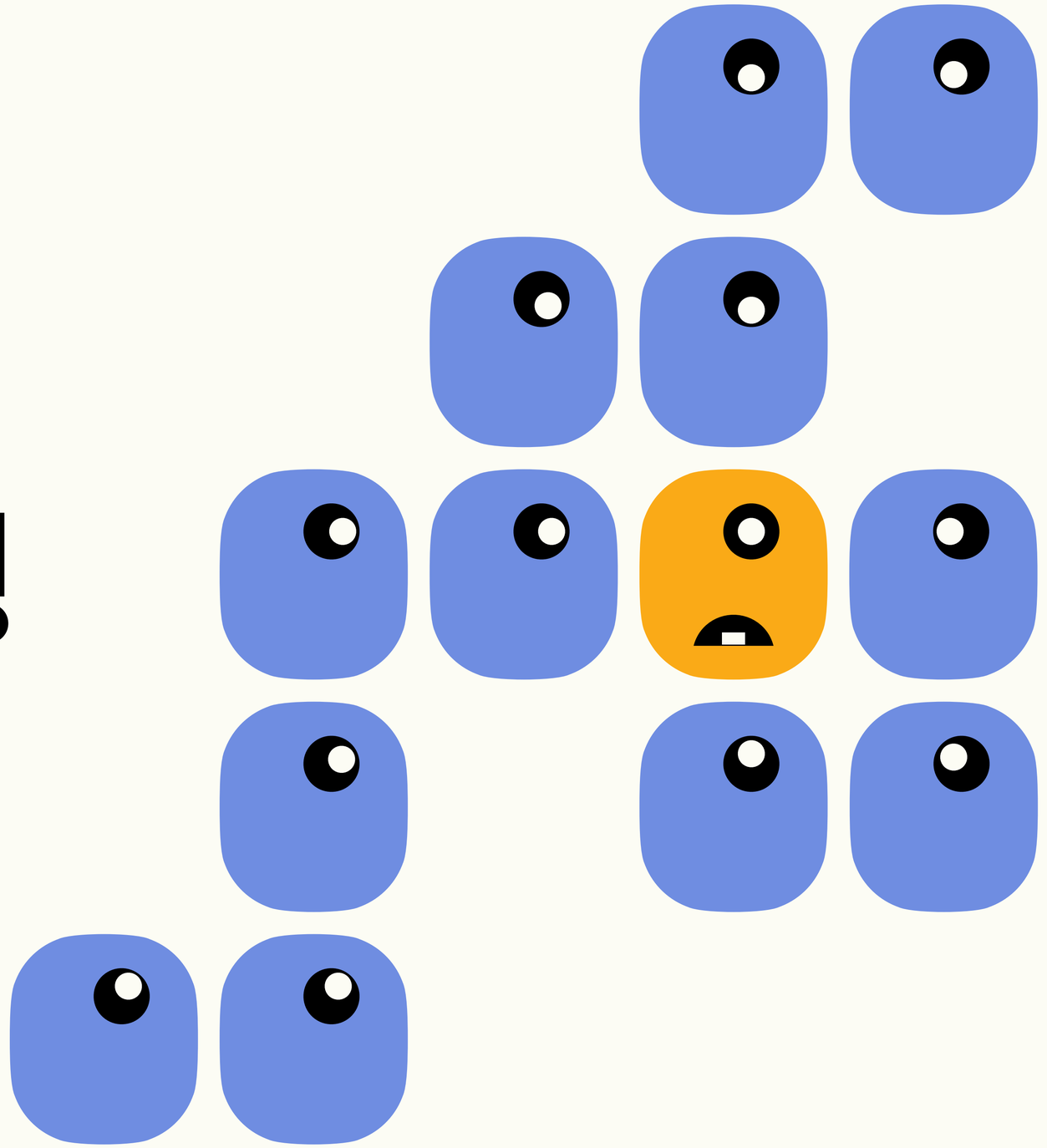
NOW WHAT?





@thomascountz

IMPOSTER!



BIG IDEA.

Give structure to **curiosity** and **collaboration** in order to get things done.

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

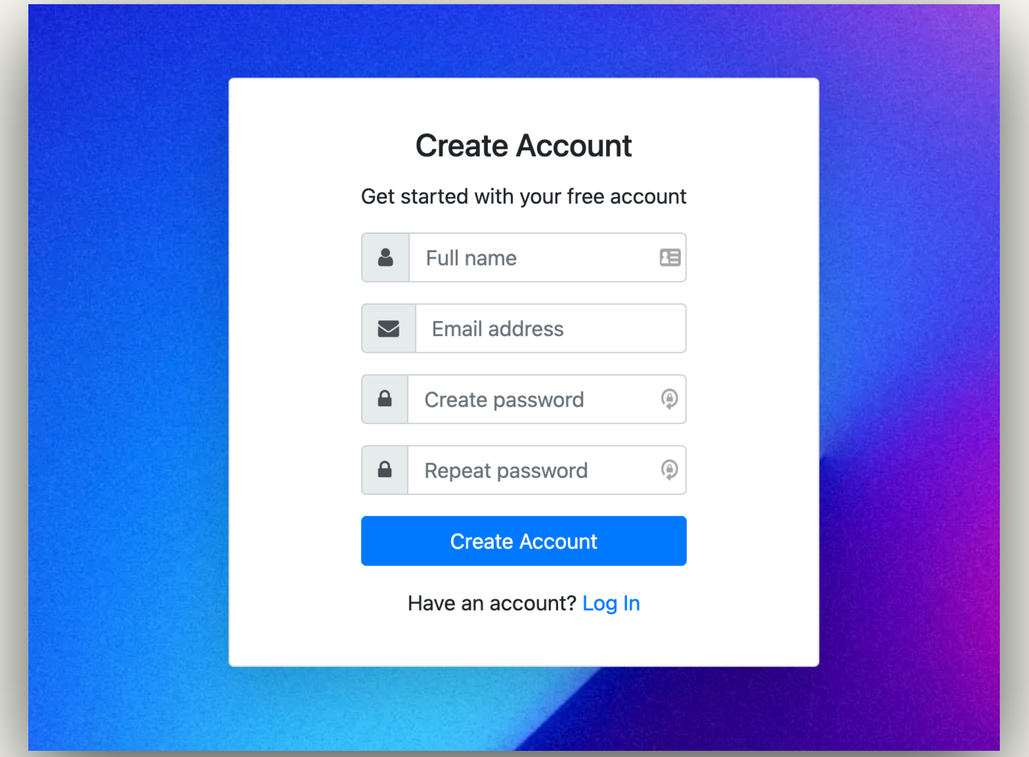
SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

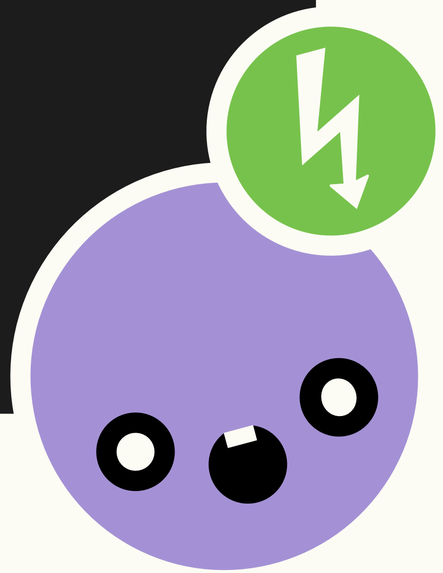
YOU HAVE A NEW TASK!

The legal team wants new users to be able to sign up with "First Name" & "Last Name."

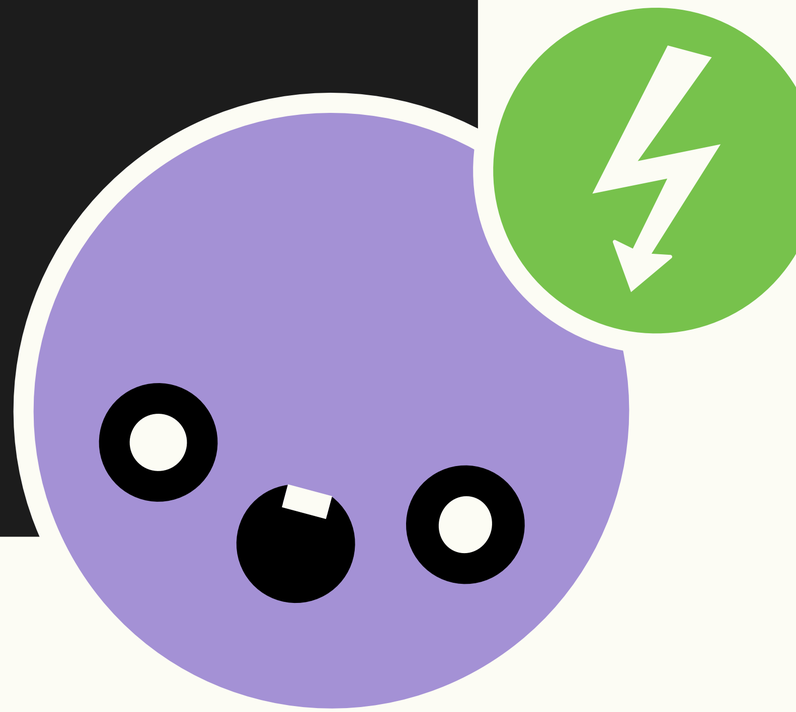


```
ActiveRecord::Schema.define do
  create_table "users", id: do |t|
    t.string "email"
    t.string "first_name"
    t.string "last_name"
  end
end
```

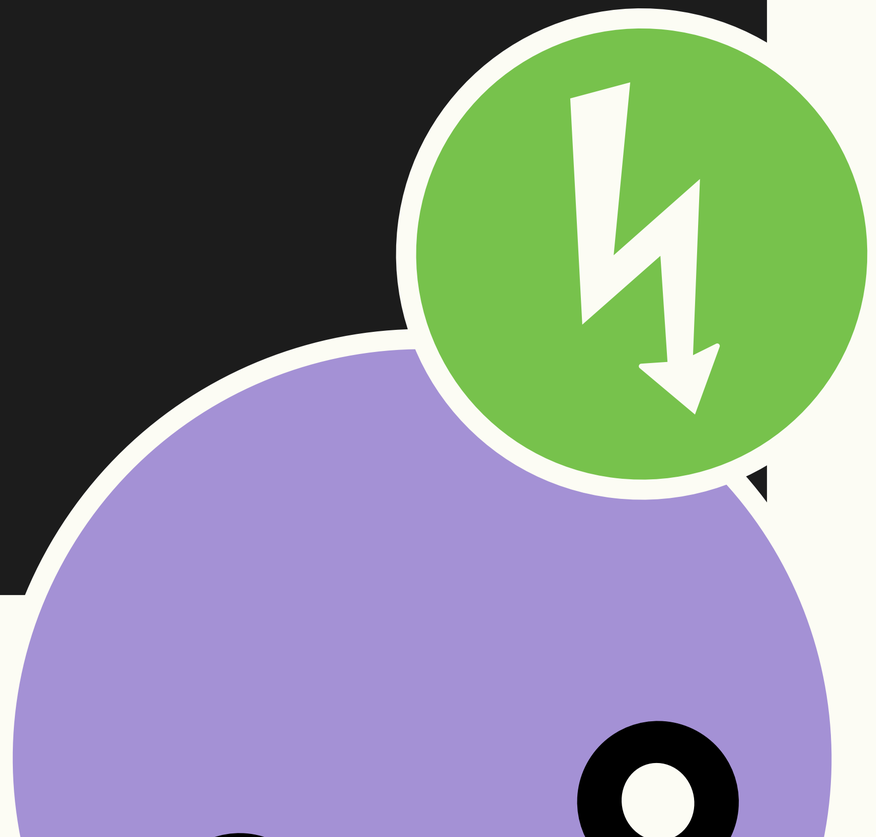
```
> user = User.first
⟹ #<User:0x00007f84c6c253b0
 id: 1,
 first_name: "Grace Richie",
 last_name: nil>
```



```
> user = User.second
⟹ #<User:0x00007f84a4b28b1e
 id: 2,
 first_name: "Mary",
 last_name: "Wayne">
```

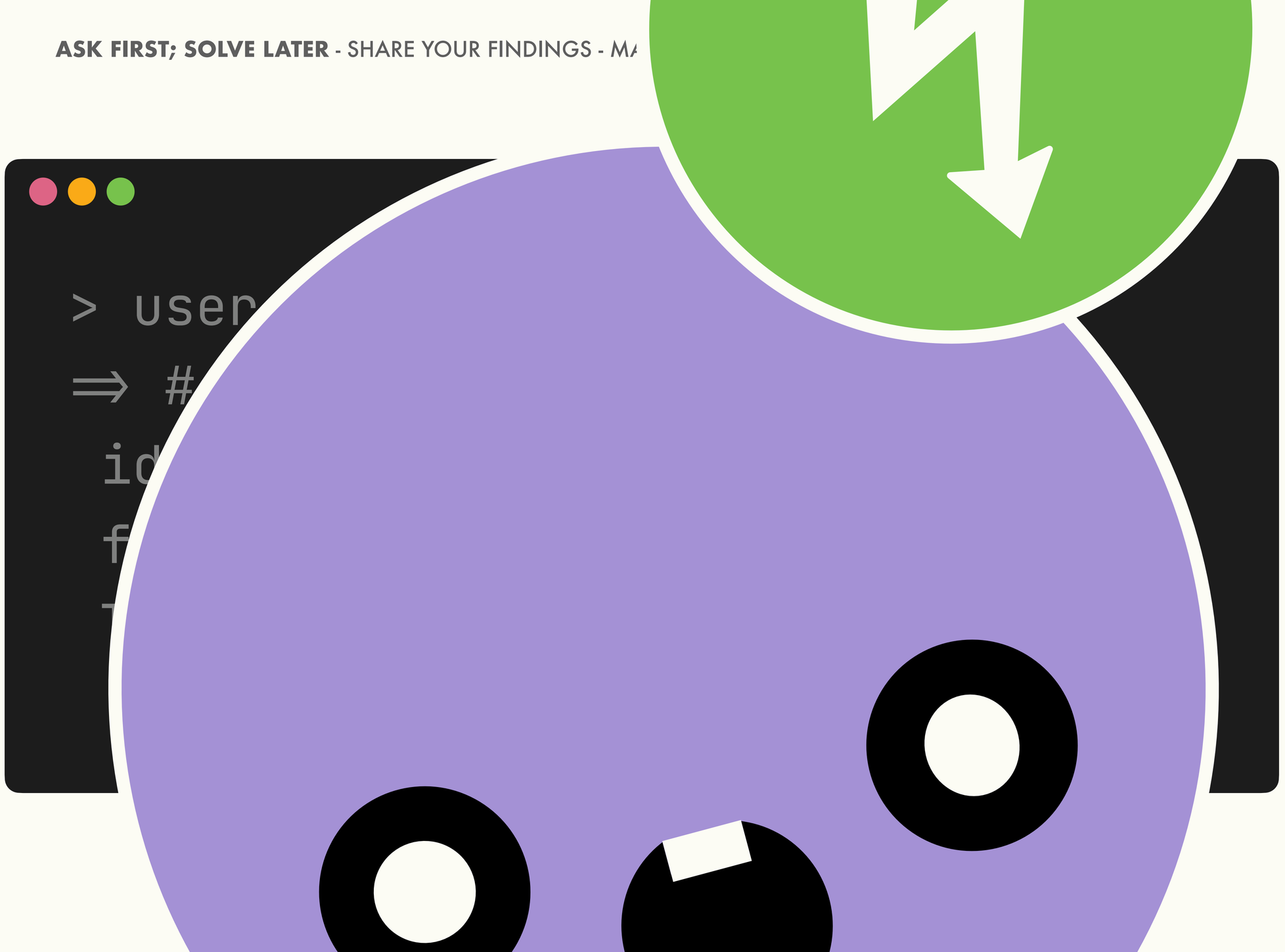


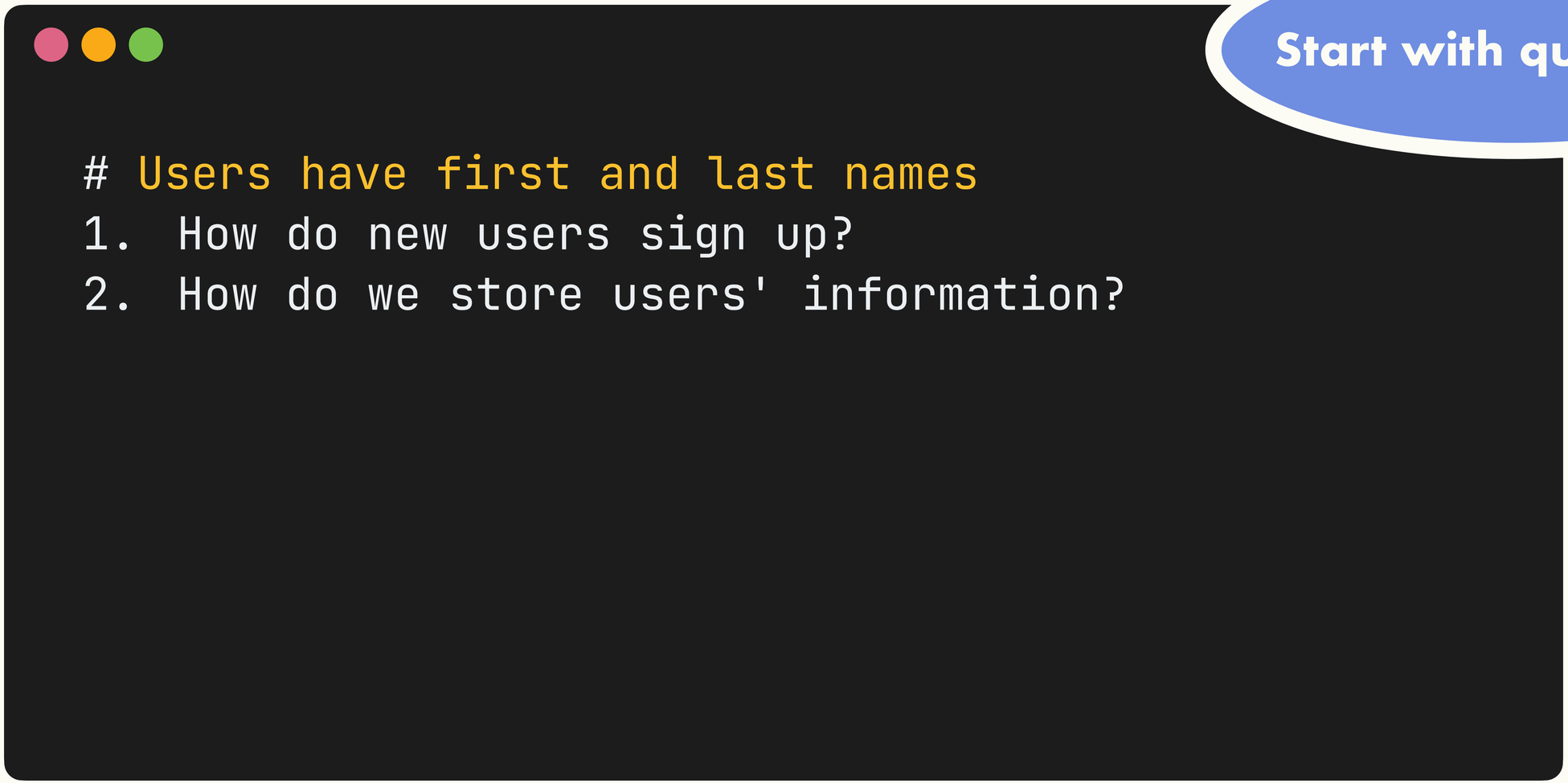
```
<%= form_for(@user) do |f| %>
  <%= f.text_field :first_name, placeholder: "Full name" %>
  # ...
  <%= f.submit "Create Account" %>
<% end %>
```



```
<%= form_for(@user) do |f| %>
  <%= f.text_field :first_name, placeholder: "First name" %>
  <%= f.text_field :last_name, placeholder: "Last name" %>
  # ...
 <%= f.submit "Create Account" %>
<% end %>
```

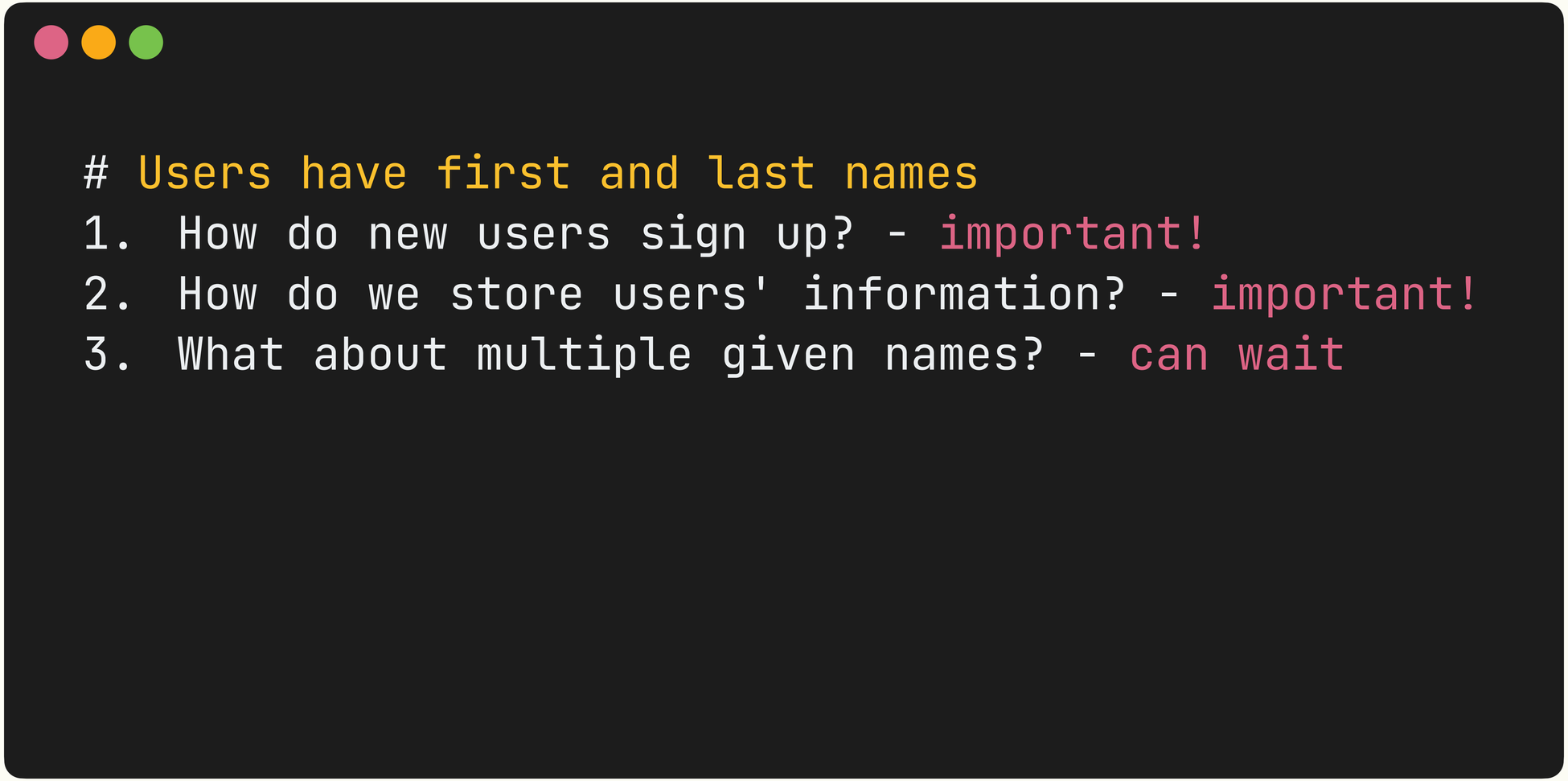
ASK FIRST; SOLVE LATER - SHARE YOUR FINDINGS - MA

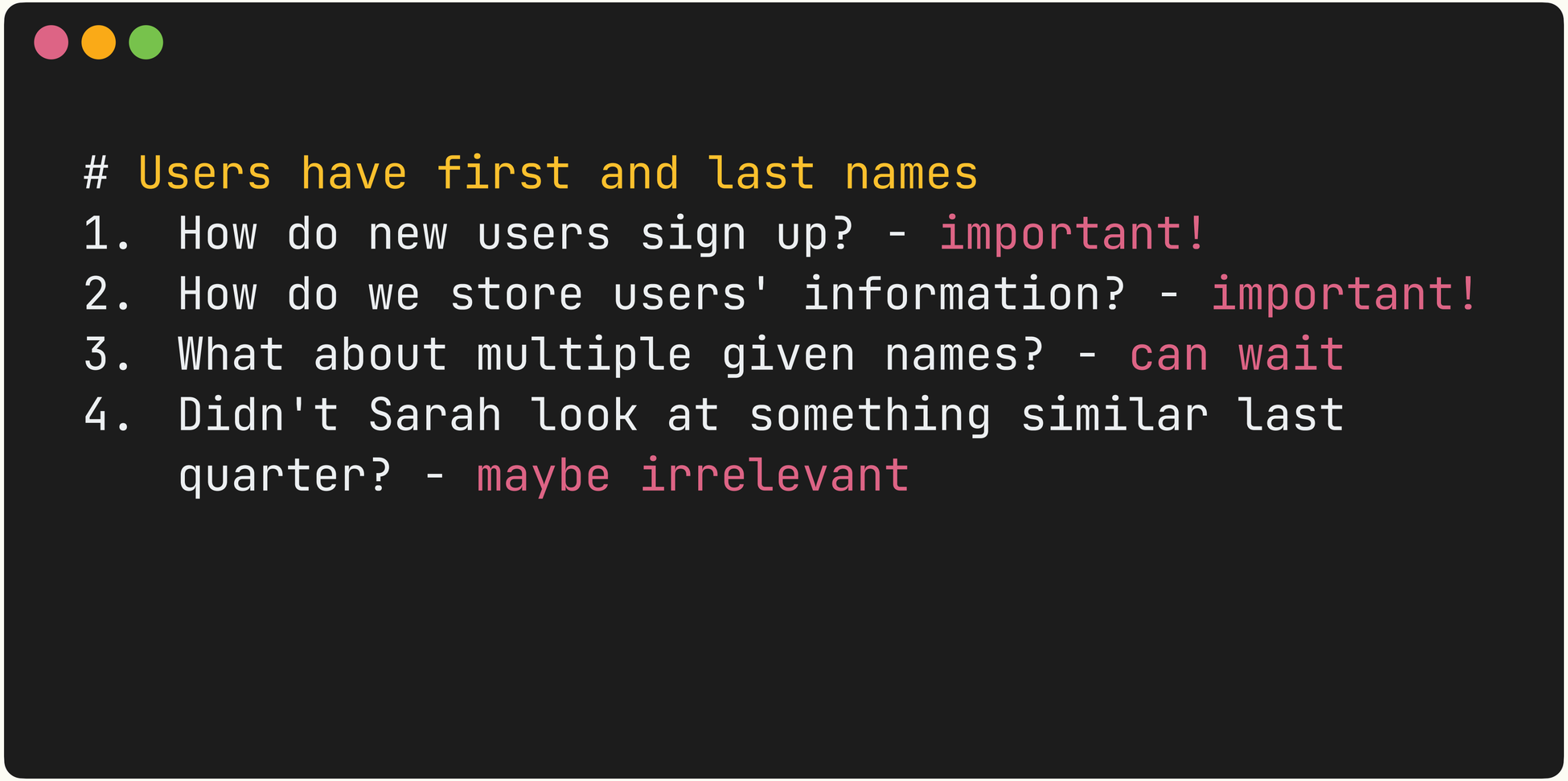



Start with questions.
Users have first and last names
1. How do new users sign up?
2. How do we store users' information?

```
# Users have first and last names
1.  How do new users sign up? - important!
2.  How do we store users' information? - important!
```

Is this blocking?

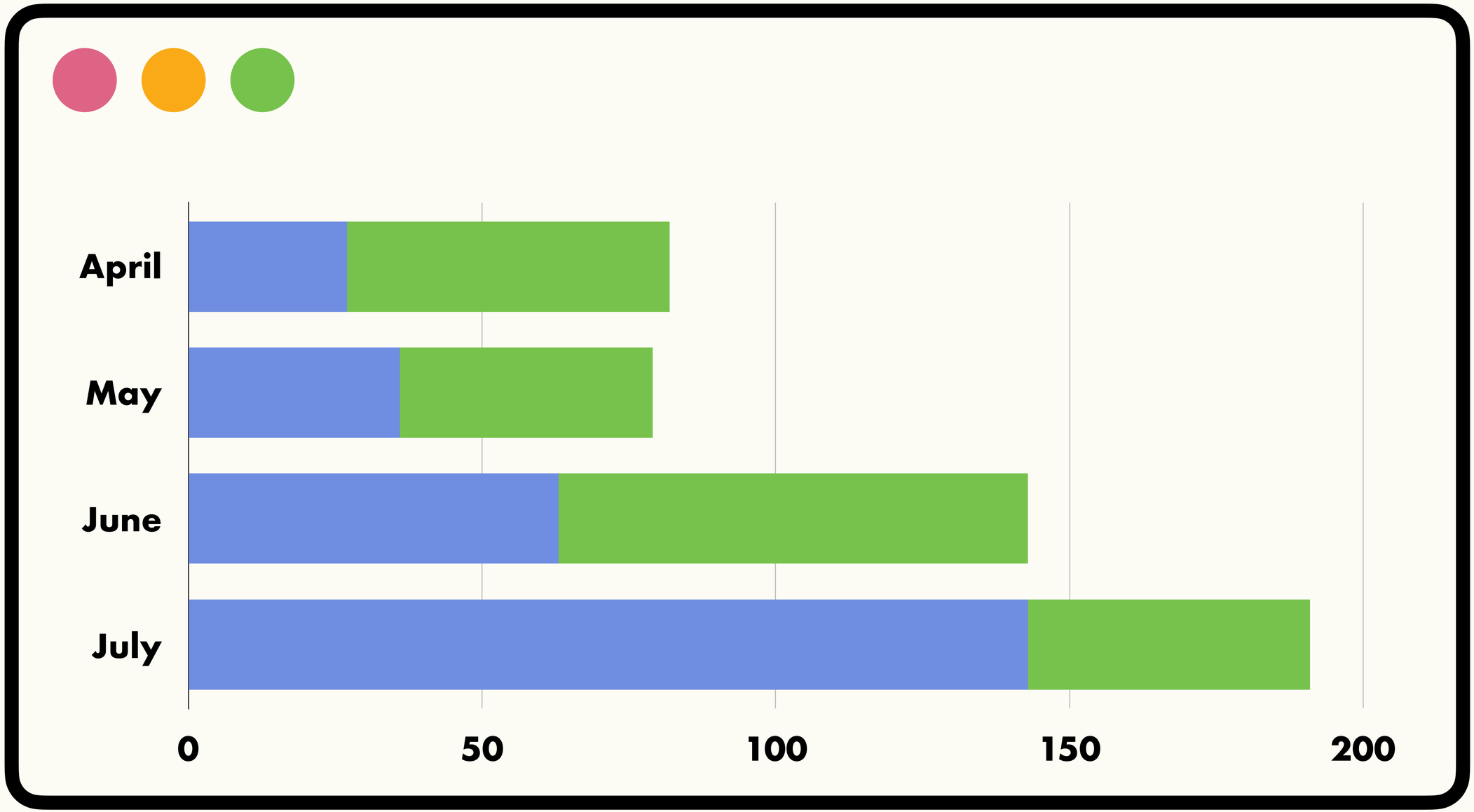

Users have first and last names
1. How do new users sign up? - important!
2. How do we store users' information? - important!
3. What about multiple given names? - can wait



```
# Users have first and last names
1. How do new users sign up? - important!
2. How do we store users' information? - important!
3. What about multiple given names? - can wait
4. Didn't Sarah look at something similar last
   quarter? - maybe irrelevant
```

2. How do we store users' information? - important!

```
ActiveRecord::Schema.define do
  create_table "users", id: do |t|
    t.string "email"
    t.string "first_name"
    t.string "last_name"
  end
end
```


April
May
June
July
0
50
100
150
200

ASK FIRST; SOLVE LATER

- Start by breaking down questions—not solutions
- No hidden work
- Asking questions is a valuable part of a programmer's job—encourage others!

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

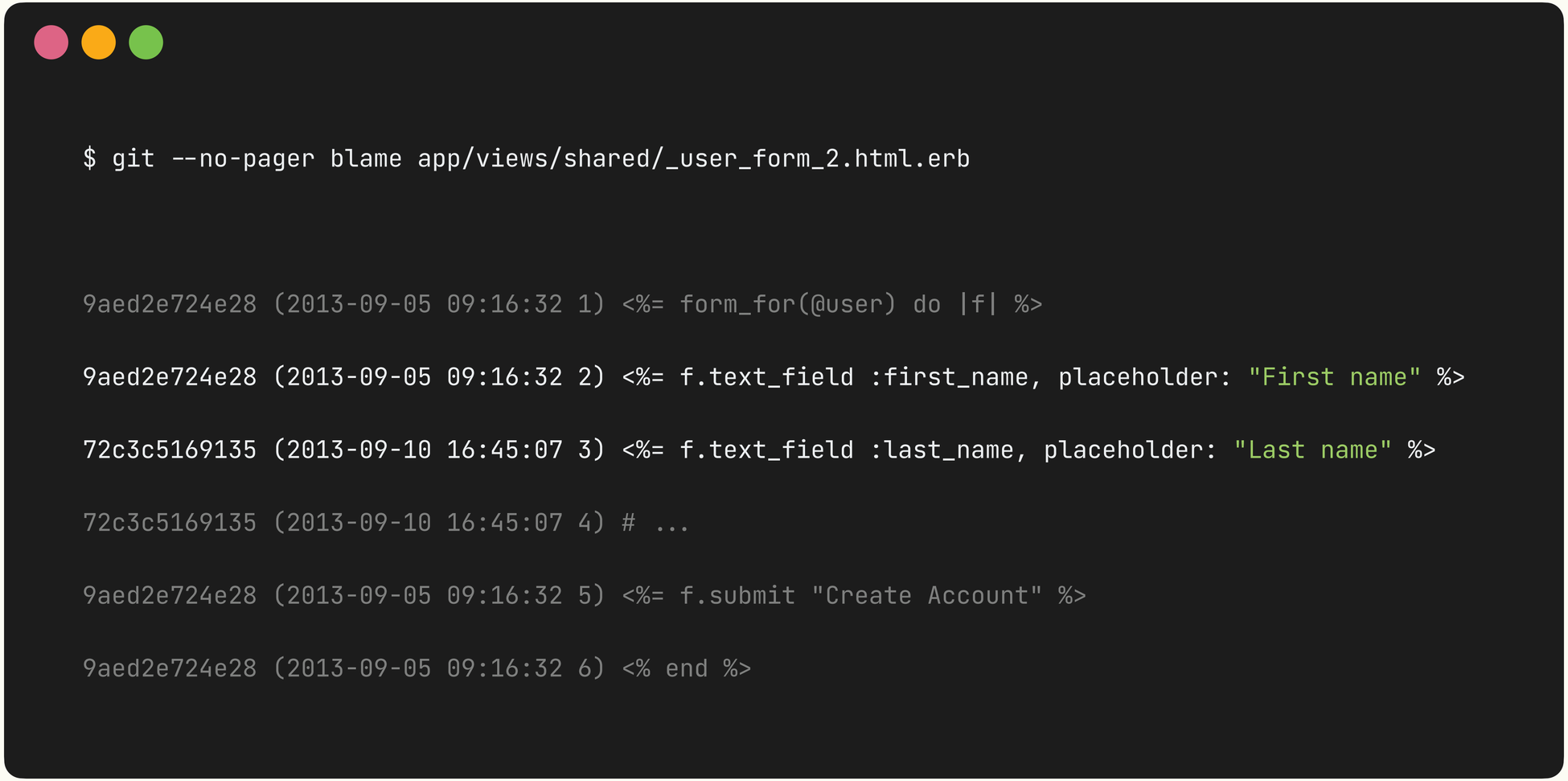
PREPARE FOR POST-DEPLOYMENT

```
> user = User.first
⇒ #<User:0x00007f84c6c253b0
 id: 1,
 first_name: "Grace Richie",
 last_name: nil>
```

```
<%= form_for(@user) do |f| %>
  <%= f.text_field :first_name, placeholder: "Full name" %>
 # ...
 <%= f.submit "Create Account" %>
<% end %>
```

```
> all_users = User.count
⇒ 5134712

> nil_last_name_users = User.where(last_name: nil).count
⇒ 5131694

> nil_last_name_users/all_users.to_f
⇒ 0.9994122357787545
```

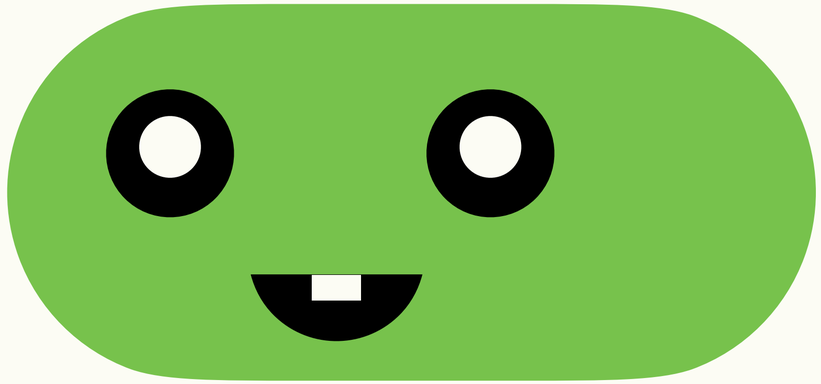
```
$ grep -r "shared/user_form" .

./app/views/users/new.html.erb:  render "shared/user_form"

./app/views/users/edit.html.erb:  render "shared/user_form"


$ grep -r "shared/user_form_2" .
```

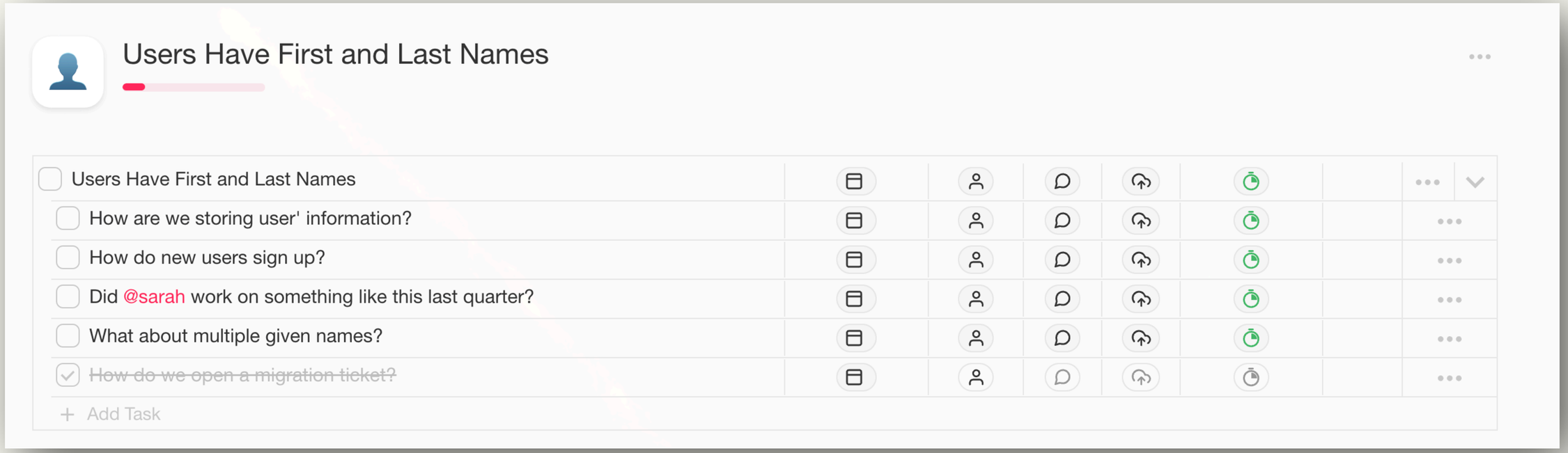


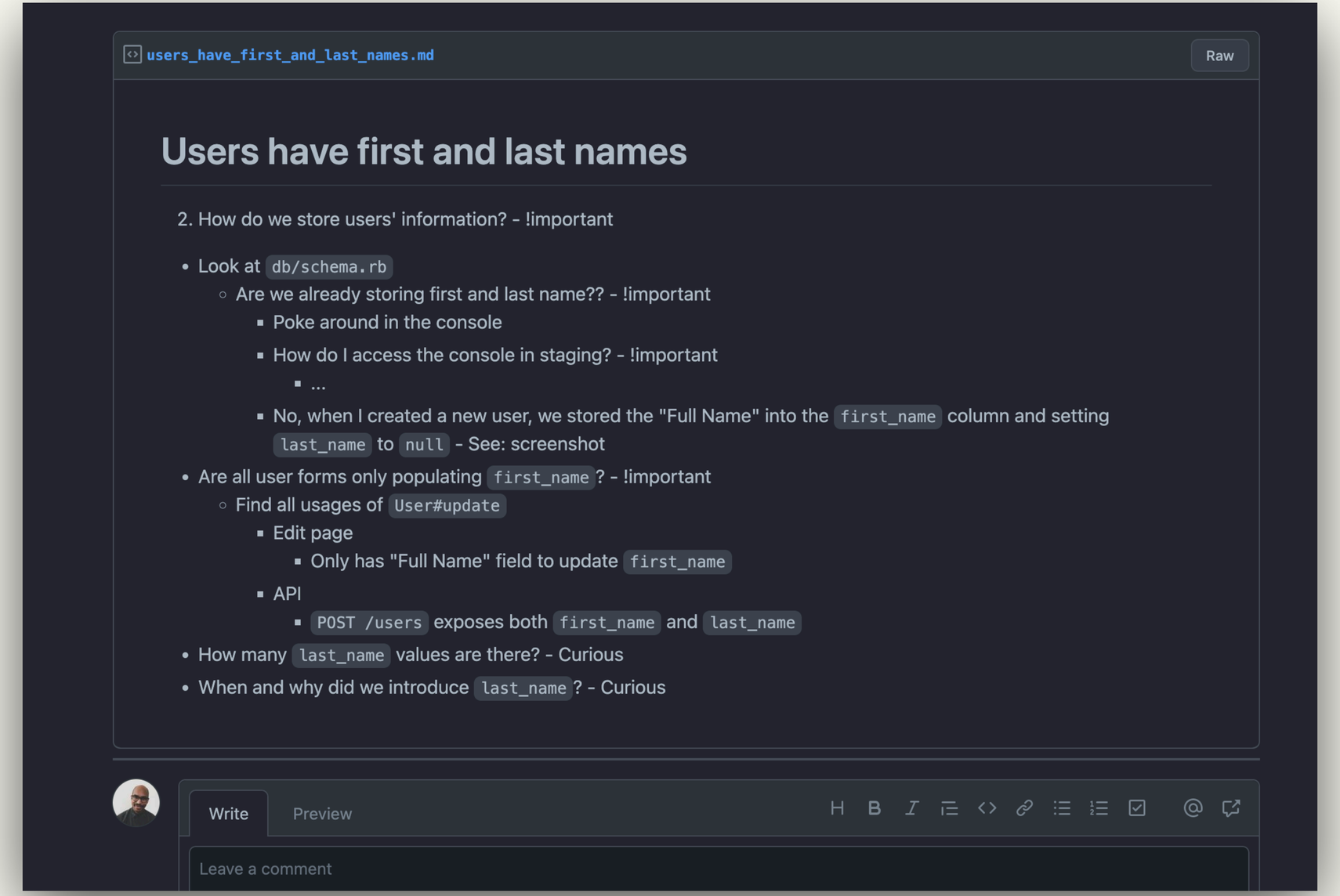
```
$ git --no-pager blame app/views/shared/_user_form_2.html.erb


9aed2e724e28 (2013-09-05 09:16:32 1) <%= form_for(@user) do |f| %>
9aed2e724e28 (2013-09-05 09:16:32 2) <%= f.text_field :first_name, placeholder: "First name" %>
72c3c5169135 (2013-09-10 16:45:07 3) <%= f.text_field :last_name, placeholder: "Last name" %>
72c3c5169135 (2013-09-10 16:45:07 4) # ...
9aed2e724e28 (2013-09-05 09:16:32 5) <%= f.submit "Create Account" %>
9aed2e724e28 (2013-09-05 09:16:32 6) <% end %>
```

I'm still investigating inconsistencies in the users table, but after two weeks, I finally figured out why we have two forms!

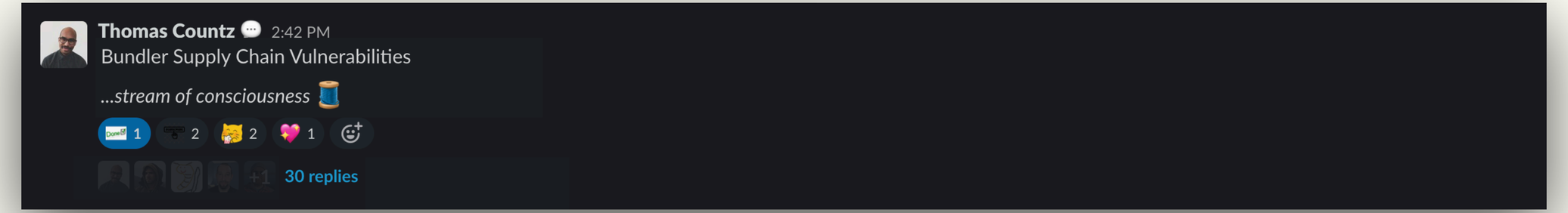
Oh, Zilphia and I added that form last year. It was for the Test team only. Let's delete the old form since they use an API now.



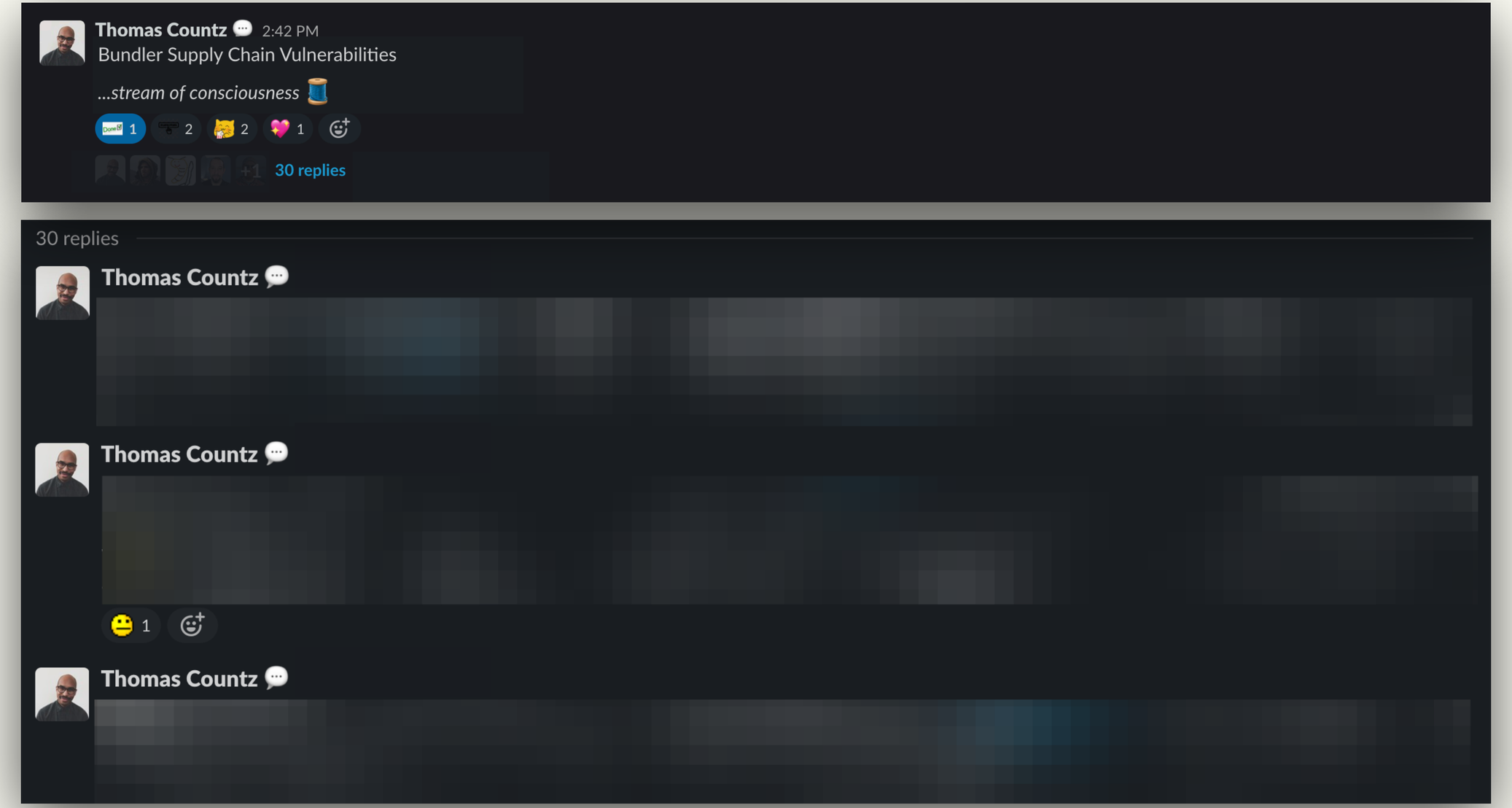

Users Have First and Last Names
Users Have First and Last Names
How are we storing user' information?
How do new users sign up?
Did @sarah work on something like this last quarter?
What about multiple given names?
How do we open a migration ticket?
Add Task


users_have_first_and_last_names.md
Raw
Users have first and last names
2. How do we store users' information? - !important
Look at db/schema.rb
Are we already storing first and last name?? - !important
Poke around in the console
How do I access the console in staging? - !important
...
No, when I created a new user, we stored the "Full Name" into the first_name column and setting last_name to null - See: screenshot
Are all user forms only populating first_name? - !important
Find all usages of User#update
Edit page
Only has "Full Name" field to update first_name
API
POST /users exposes both first_name and last_name
How many last_name values are there? - Curious
When and why did we introduce last_name? - Curious
Write
Preview
Leave a comment

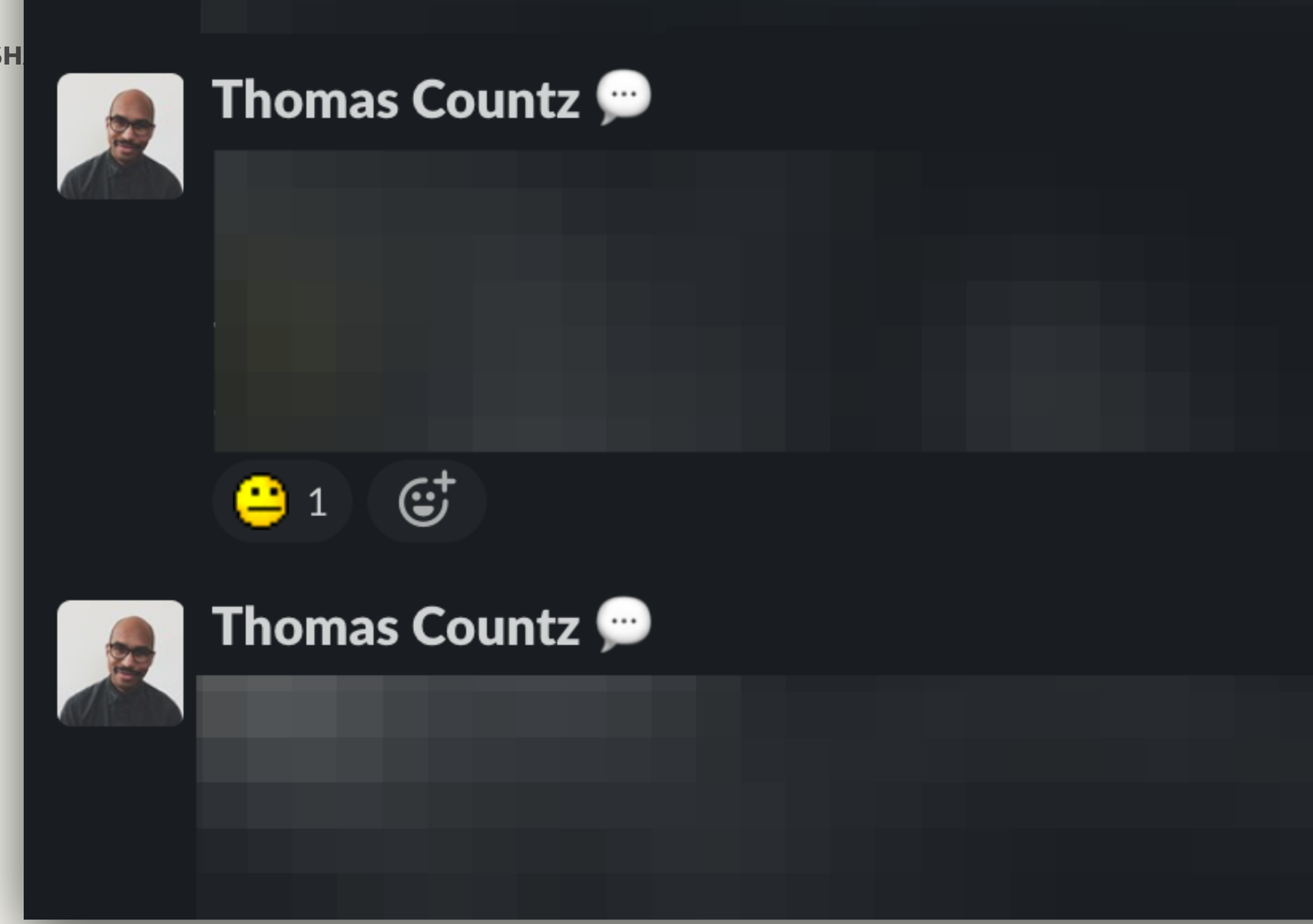
ASK FIRST; SOLVE LATER - **SHARE YOUR FINDINGS** - MAKE CODE EXECUTABLE - PREPARE FOR POST-DEPLOYMENT

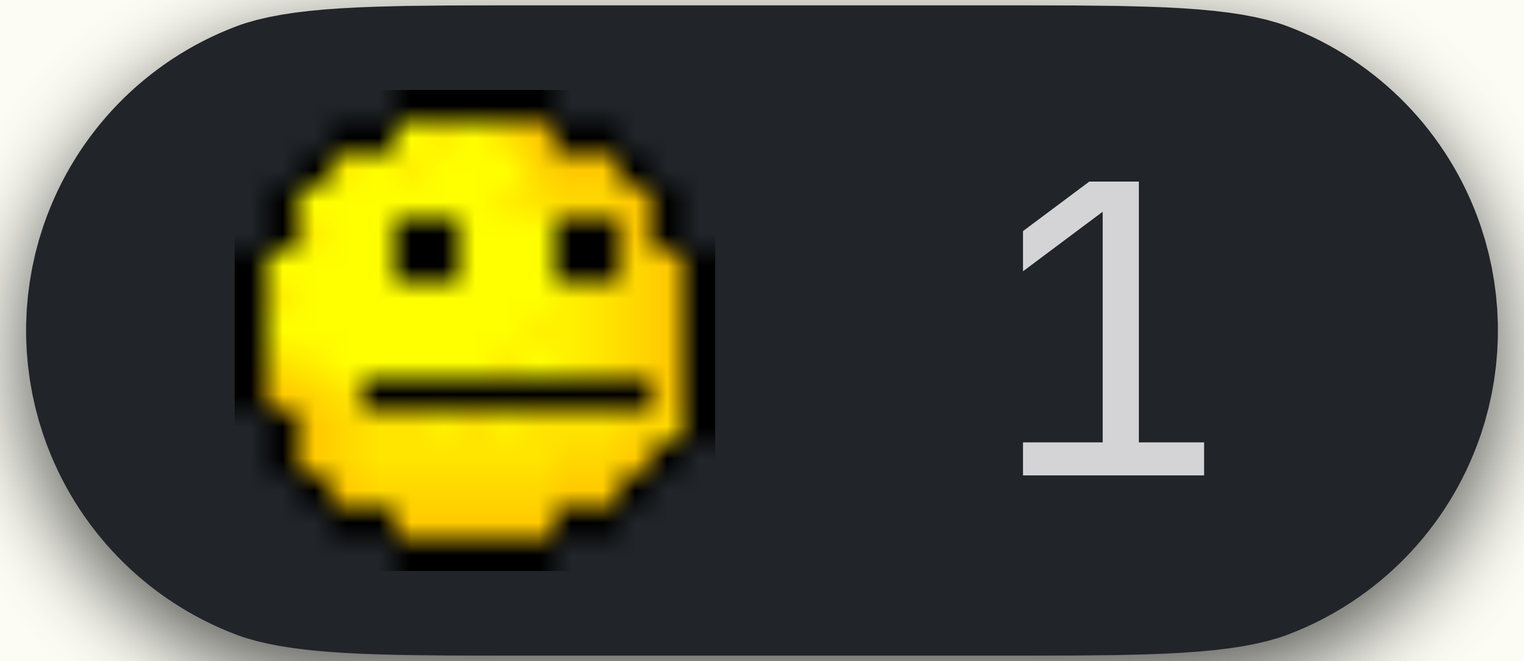


ASK FIRST; SOLVE LATER - **SHARE YOUR FINDINGS** - MAKE CODE EXECUTABLE - PREPARE FOR POST-DEPLOYMENT



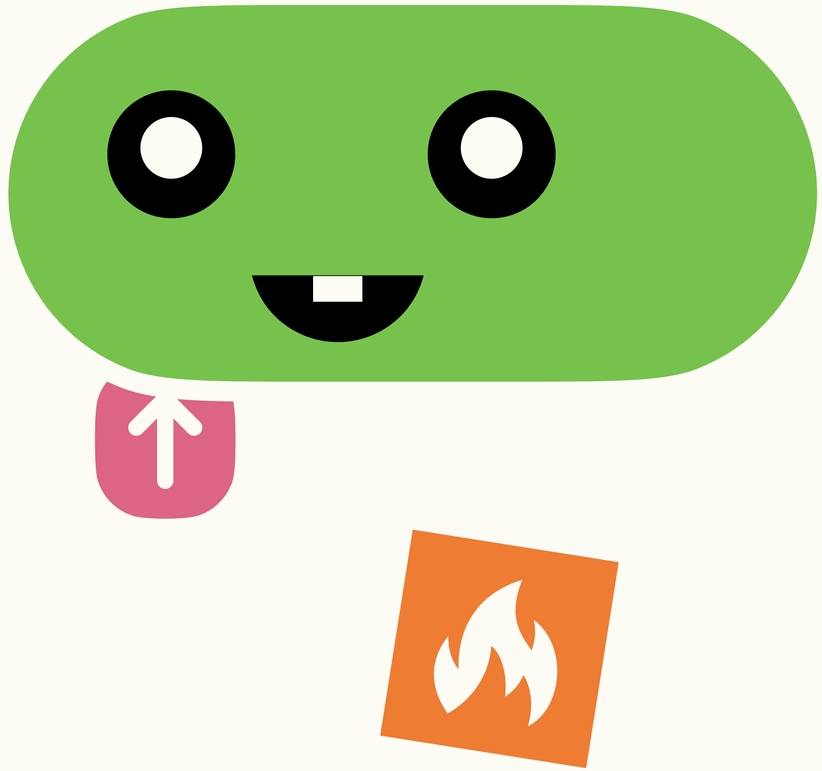
ASK FIRST; SOLVE LATER - **SH**




1

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

- Tap into your team's expertise and experience
- Organically document the system
- Turn knowledge into value
- Just get started—there's no wrong way



ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

```
> user = User.first
⟹ #<User:0x00007f84c6c253b0
 id: 1,
 first_name: "Grace Richie",
 last_name: nil>
```

```
> user = User.first
⟹ #<User:0x00007f84c6c253b0
 id: 1,
 first_name: "Grace",
 last_name: "Richie">
```

```ruby
def User < ActiveRecord::Base
  def split_names!
    return self if !last_name.nil?
    names = first_name.split(/\s+/)
    tap do |user|
      user.first_name = names[0]
      user.last_name = names[1]
    end
  end
end
```

```ruby
def User < ActiveRecord::Base
  def split_names!
    return self if !last_name.nil?
    names = first_name.split(/\s+/)
    tap do |user|
      user.first_name = names[0]
      user.last_name = names[1]
    end
  end
end
```

```
def User < ActiveRecord::Base
  def split_names!
    return self if !last_name.nil?
    names = first_name.split(/\s+/)
    tap do |user|
      user.first_name = names[0]
      user.last_name = names[1]
    end
  end
end
```

```ruby
def User < ActiveRecord::Base
  def split_names!
    return self if !last_name.nil?
    names = first_name.split(/\s+/)
    tap do |user|
      user.first_name = names[0]
      user.last_name = names[1]
    end
  end
end
```

```ruby
def User < ActiveRecord::Base
  def split_names!
    return self if !last_name.nil?
    names = first_name.split(/\s+/)
    tap do |user|
      user.first_name = names[0]
      user.last_name = names[1]
    end
  end
end
```

```ruby
def User < ActiveRecord::Base
  def split_names!
    return self if !last_name.nil?
    names = first_name.split(/\s+/)
    tap do |user|
      user.first_name = names[0]
      user.last_name = names[1]
    end
  end
end
```

```ruby
class User
  def initialize; end
  def split_names!; end
end

User.new(first_name: "Isaiah Lee").split_names!
⇒ #<User:0x00007fa92232bea0
 first_name: "Isaiah",
 last_name: "Lee">
```

```ruby
class User < ActiveRecord::Base
  def split_names!; end
end
```

```ruby
RSpec.describe User do
  describe "#split_names!" do
    it "splits the first_name into first_name & last_name"
  end
end
```

```ruby
#!/usr/bin/env ruby
puts "Hello, World!"
```

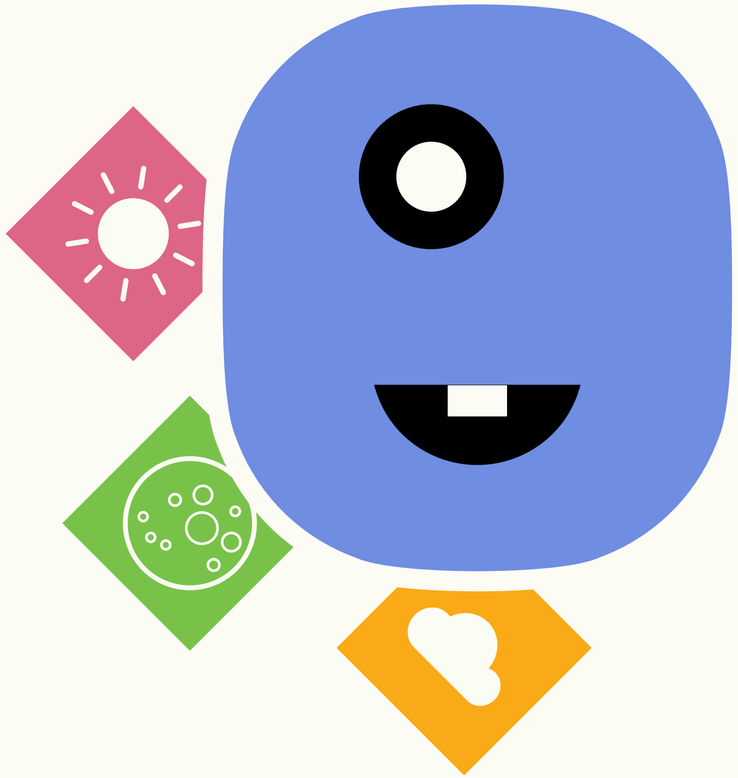
```
$ chmod +x hello.rb
$ ./hello.rb
"Hello, World"
```

```ruby
require 'bundler/inline'

gemfile do
  source 'https://rubygems.org'

  gem 'null_gem', require: true
end
```

```
ActiveRecord::Base.establish_connection(
  adapter: 'sqlite3',
  database: ':memory:'
)
```

https://bit.ly/3lhk0OS

```ruby
def split_names!
  return self if !last_name.nil?
  names = first_name.split(/\s+/)
  tap do |user|
    user.first_name = names[0]
    user.last_name = names[1]
  end
end
```



MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

- Rubyists love writing tests
- Executing code is less ambiguous
- The less others have to keep in their heads, the better their feedback can be

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

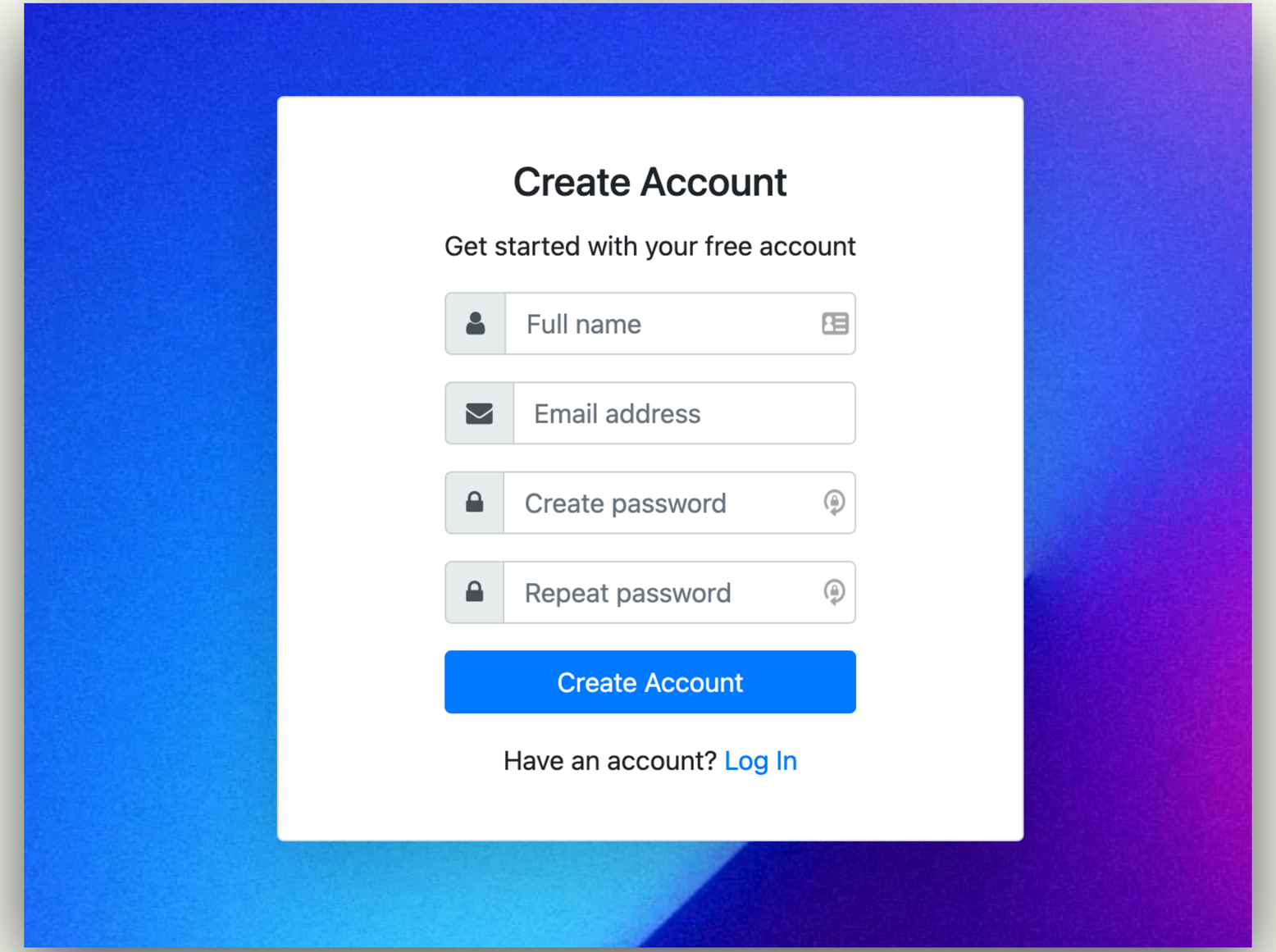
ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

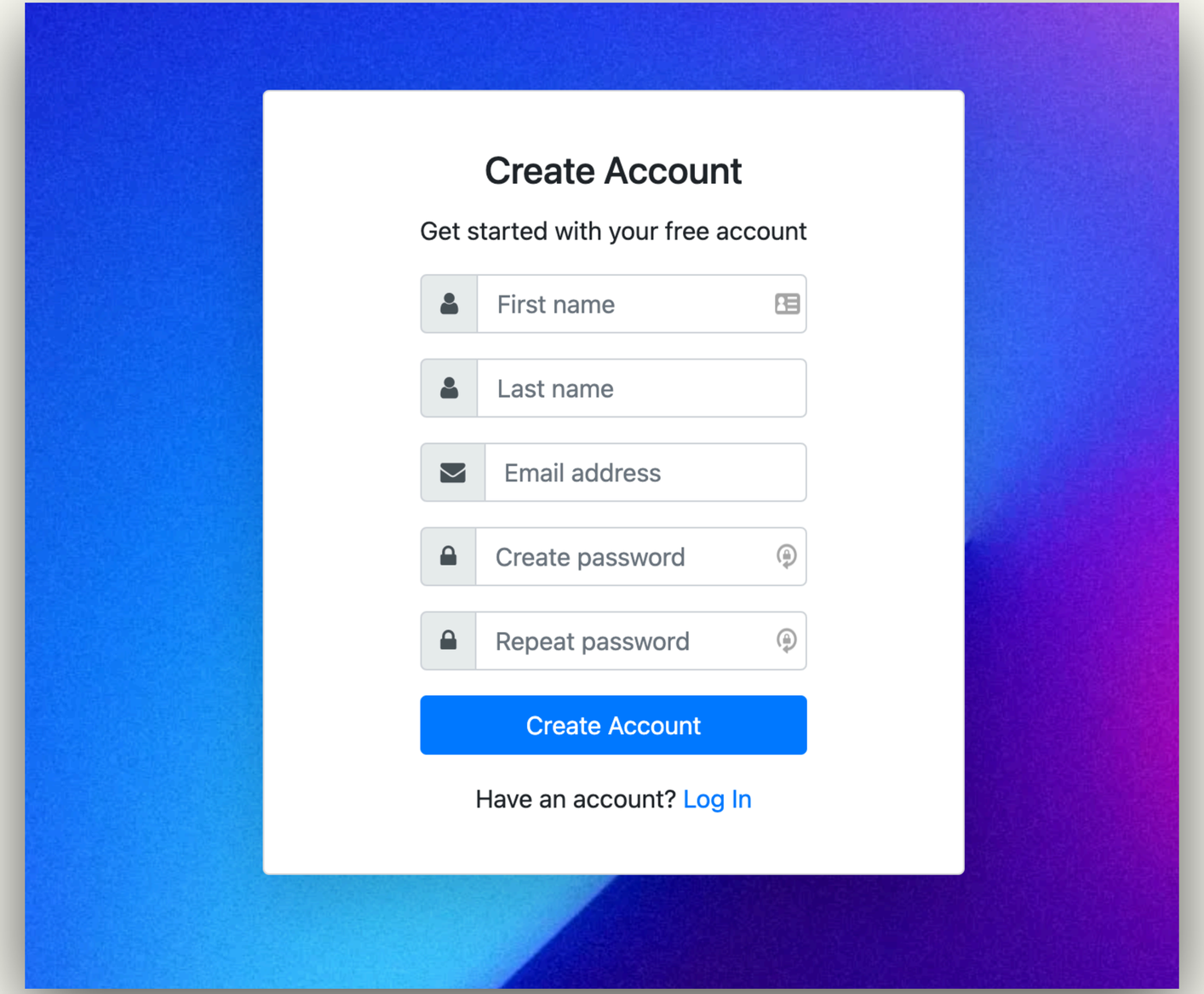
```
> user = User.first
⇒ #<User:0x00007f84c6c253b0
 id: 1,
 first_name: "Grace Richie",
 last_name: nil>
```


Create Account
Get started with your free account
Full name
Email address
Create password
Repeat password
Create Account
Have an account? Log In

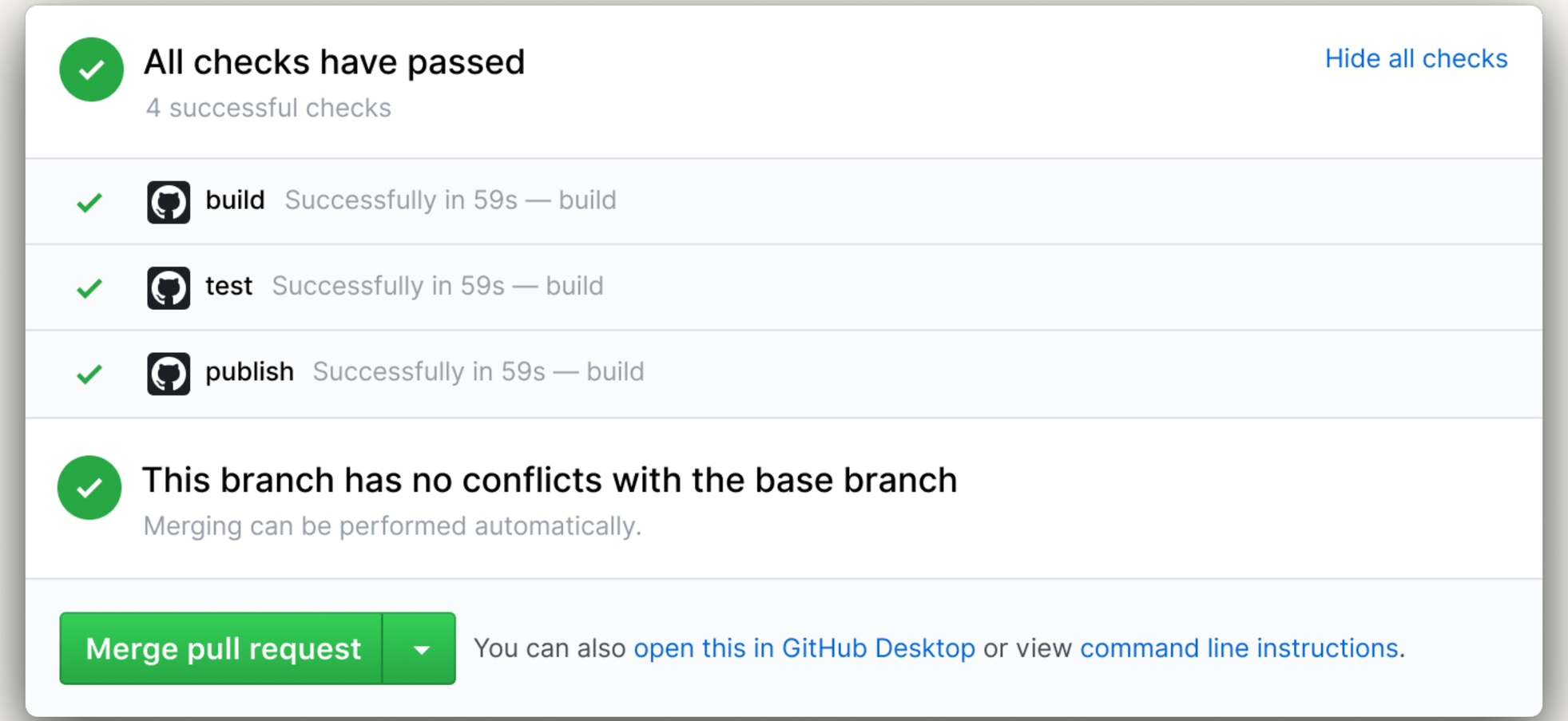
```
<%= form_for(@user) do |f| %>
  <%= f.text_field :first_name, placeholder: "Full name" %>
  # ...
  <%= f.submit "Create Account" %>
<% end %>
```

```
> user = User.last
⟹ #<User:0x00007f84c6c253b0
 id: 42,
 first_name: "Thomas Countz",
 last_name: nil>
```

```
<%= form_for(@user) do |f| %>
  <%= f.text_field :first_name, placeholder: "First name" %>
  <%= f.text_field :last_name, placeholder: "Last name" %>
  # ...
 <%= f.submit "Create Account" %>
<% end %>
```


Create Account
Get started with your free account
First name
Last name
Email address
Create password
Repeat password
Create Account
Have an account? Log In

```
> user = User.last
⟹ #<User:0x00007f84c201bbae
 id: 43,
 first_name: "Thomas",
 last_name: "Countz">
```


All checks have passed
4 successful checks
Hide all checks
build Successfully in 59s — build
test Successfully in 59s — build
publish Successfully in 59s — build
This branch has no conflicts with the base branch
Merging can be performed automatically.
Merge pull request
You can also open this in GitHub Desktop or view command line instructions.

ASK FIRST; SOLVE LATER - SHARE YOUR FINDINGS - MAKE CODE EXECUTABLE - **PREPARE FOR POST-DEPLOYMENT**

This pipeline thing is all green!!

	Prepare Container	Install Gems	Prepare Database	Rake	Security scan	Deploy
Average stage times: (Average full run time: ~14min 34s)	10min 14s	3min 41s	3s	1min 42s	25s	10ms
#20 Aug 04 16:31 — 1 commits	9min 9s	3min 32s	7s	1min 19s	24s	8ms
#19 Aug 04 16:10 — 1 commits	8min 58s	3min 31s	3s	1min 15s	24s	8ms
#18 Aug 04 15:36 — 1 commits	9min 6s	3min 35s	3s	2min 9s	24s	7ms
#17 Aug 04 15:30 — No Changes						

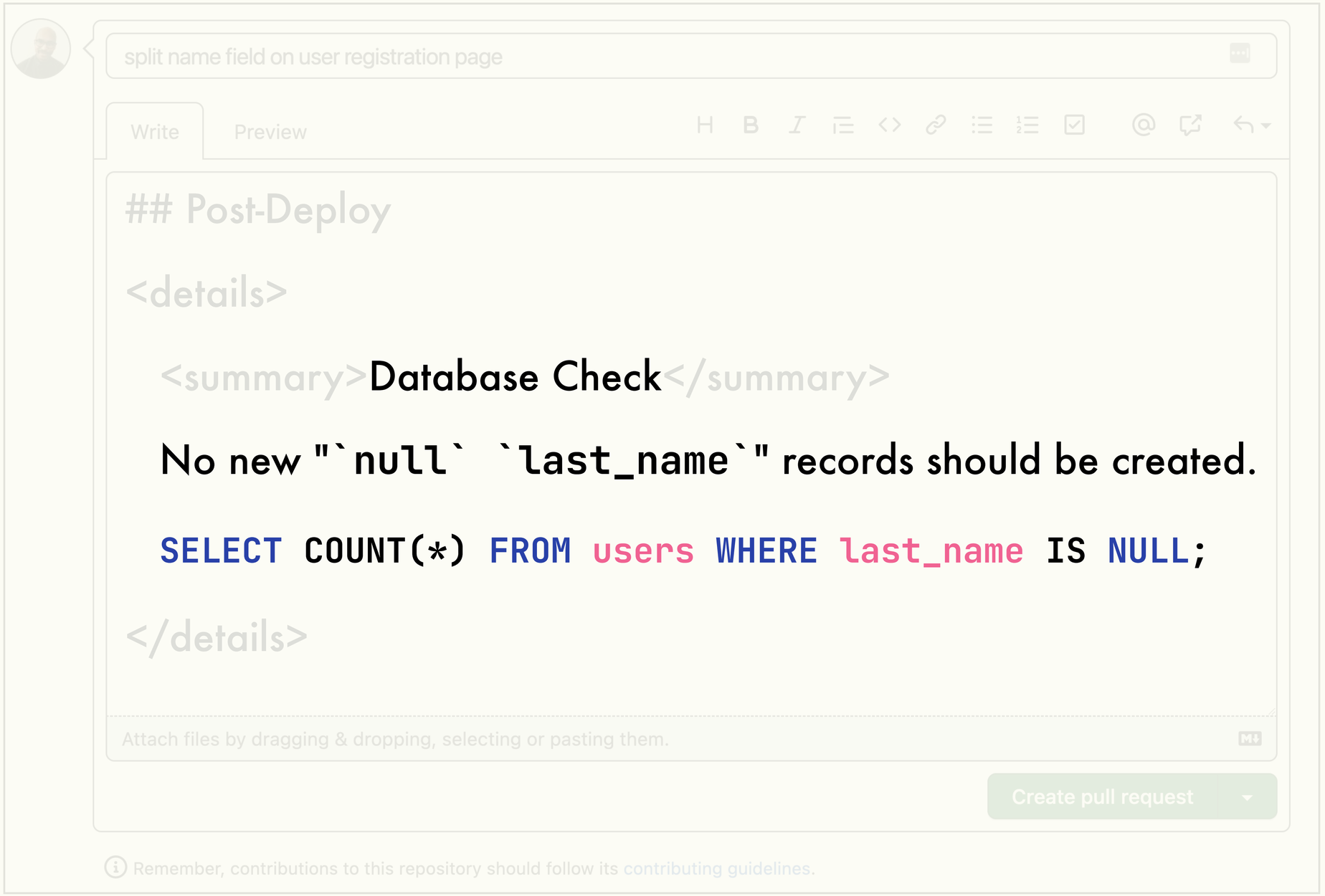
I'm sure the code is working just fine!

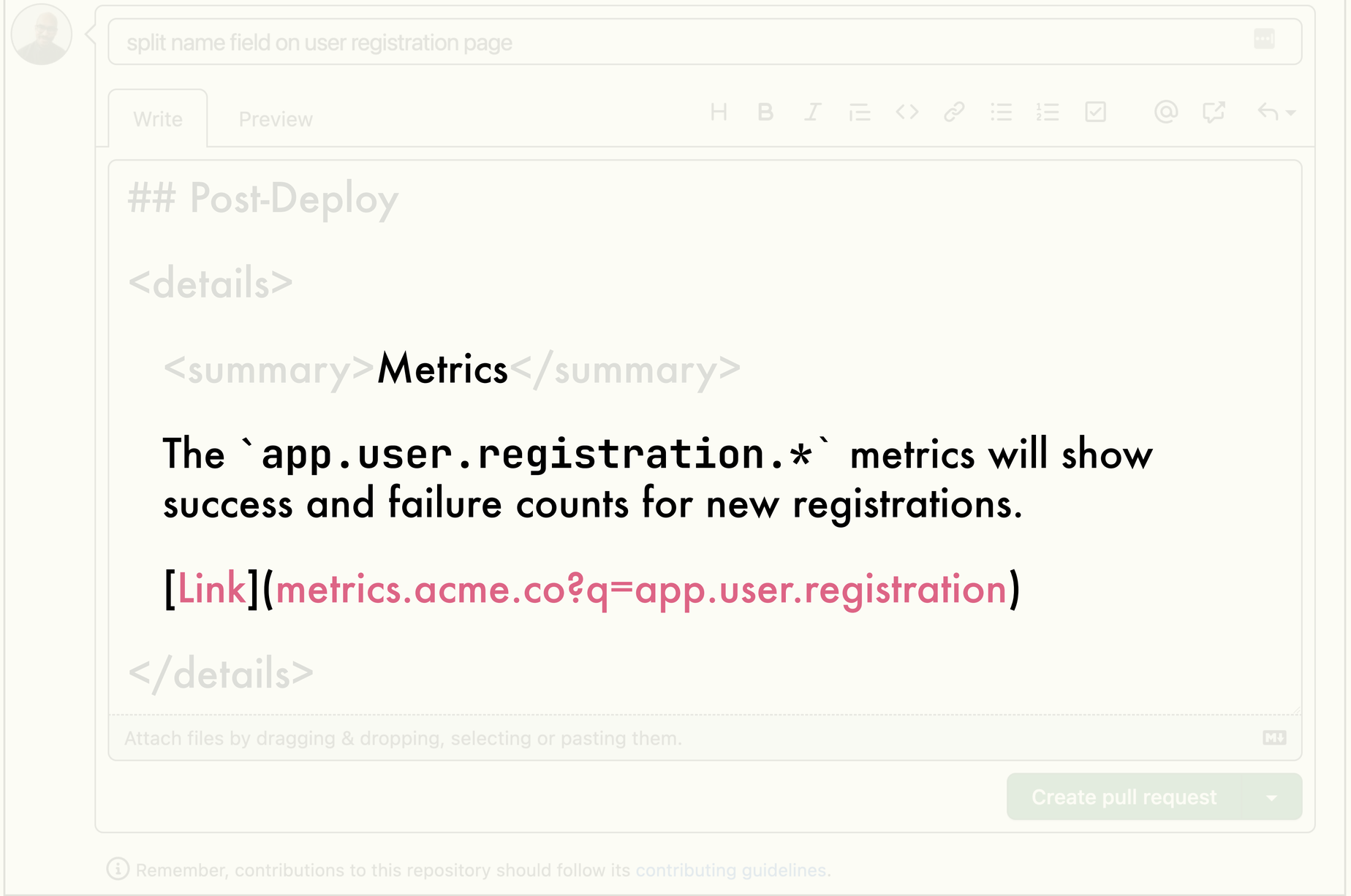
I don't think I see any issues...?

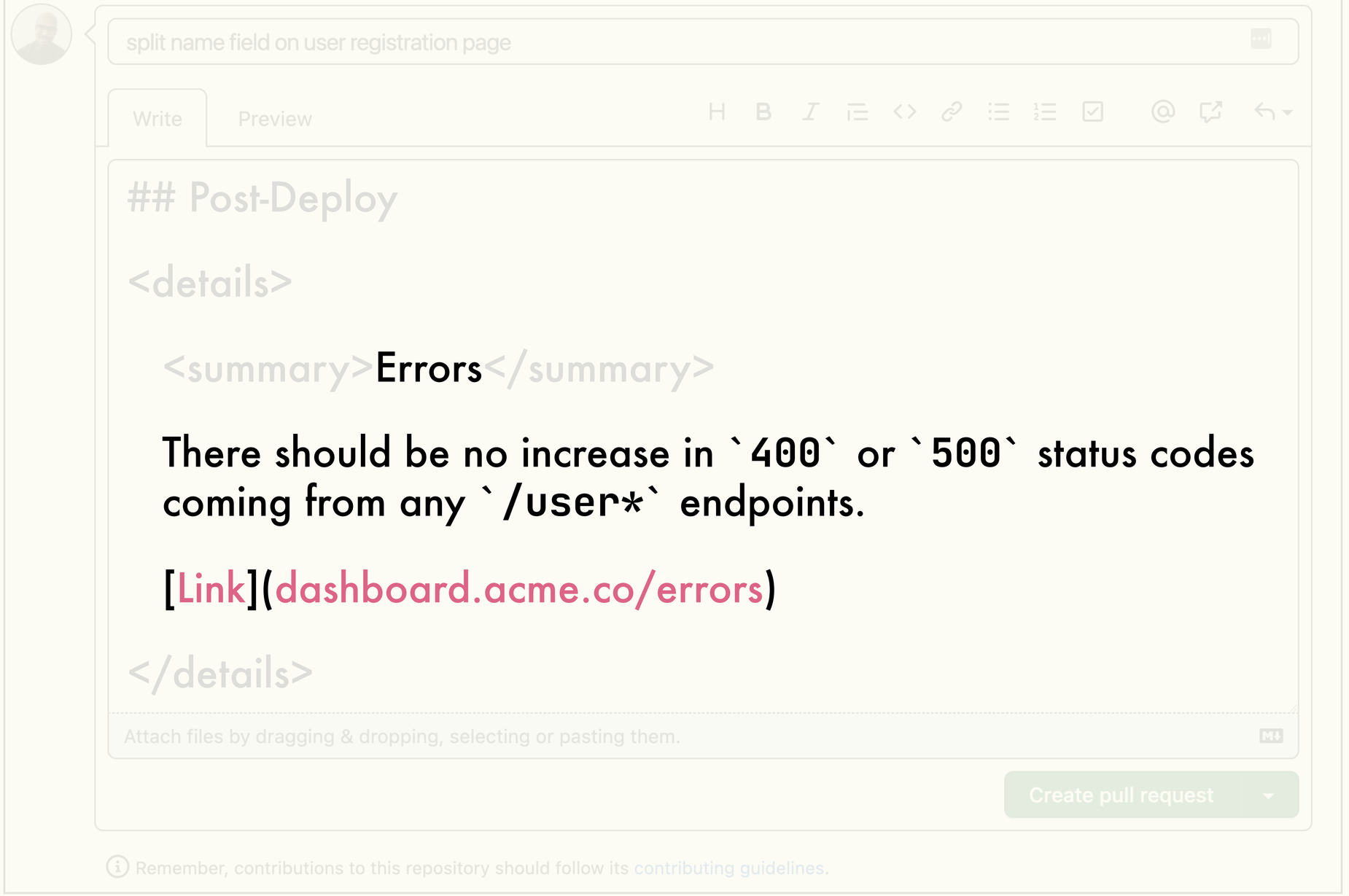
...I think it's working...?

HOW IS OUR CODE DOING NOW THAT IT'S IN PRODUCTION?


split name field on user registration page
Write
Preview
Post-Deploy
<details>
<summary>Browser Test</summary>
Registering a new user on the `/registration` page should require "First name" and "Last name."
</details>
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Remember, contributing to this repository should follow its contributing guidelines.


split name field on user registration page
Write
Preview
Post-Deploy
<details>
<summary>Database Check</summary>
No new "`null` `last_name`" records should be created.
SELECT COUNT(*) FROM users WHERE last_name IS NULL;
</details>
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Remember, contributions to this repository should follow its contributing guidelines.


split name field on user registration page
Write
Preview
Post-Deploy
<details>
<summary>Metrics</summary>
The `app.user.registration.*` metrics will show success and failure counts for new registrations.
[Link](metrics.acme.co?q=app.user.registration)
</details>
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Remember, contributions to this repository should follow its contributing guidelines.


split name field on user registration page
Write
Preview
Post-Deploy
<details>
<summary>Errors</summary>
There should be no increase in `400` or `500` status codes coming from any `/user*` endpoints.
[Link](dashboard.acme.co/errors)
</details>
Attach files by dragging & dropping, selecting or pasting them.
Create pull request
Remember, contributions to this repository should follow its contributing guidelines.

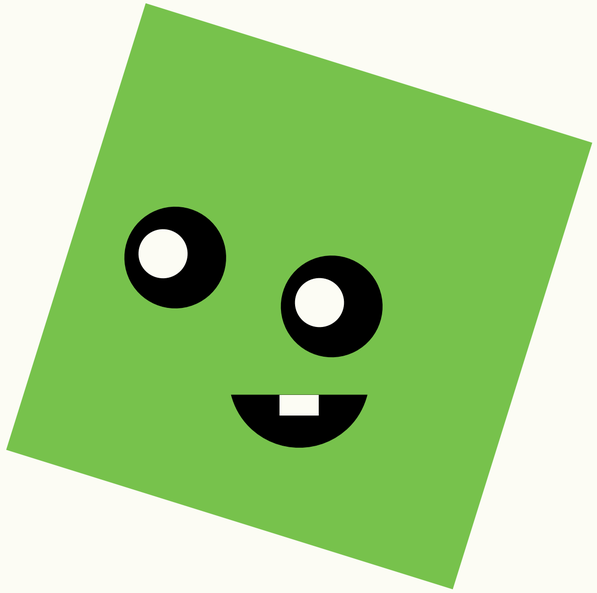
The input validations are working on the registration form.

All new users have a non-nil value for "last name" in the database.

The metrics show consistent counts of successful user registrations.

...And there are no increases in errors!

PREPARE FOR POST-DEPLOYMENT



- Your job doesn't end when the code is deployed
- Use feedback loops in production just like with test-driven development

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

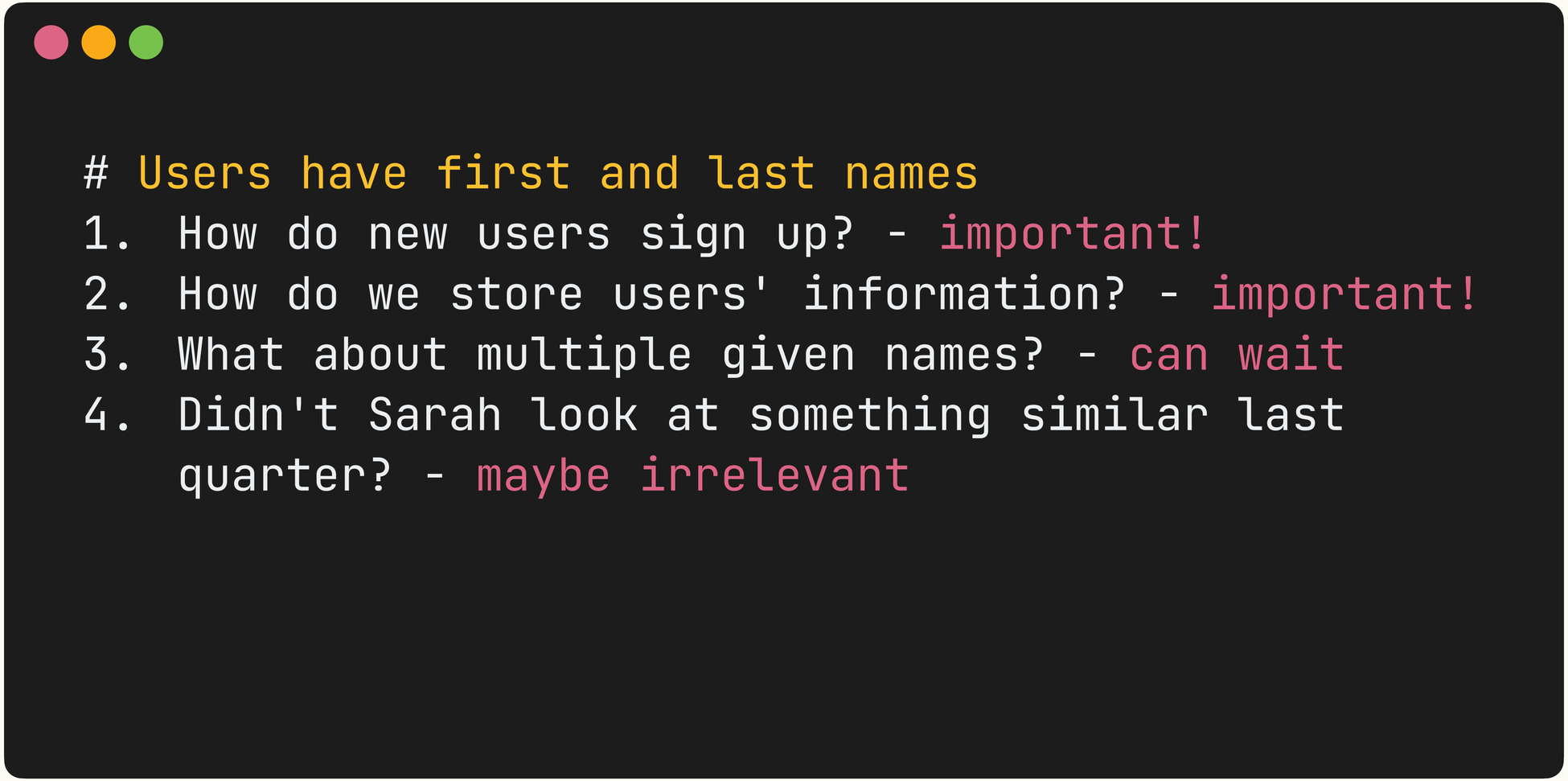
PREPARE FOR POST-DEPLOYMENT

ASK FIRST; SOLVE LATER

SHARE YOUR FINDINGS AS YOU DISCOVER THEM

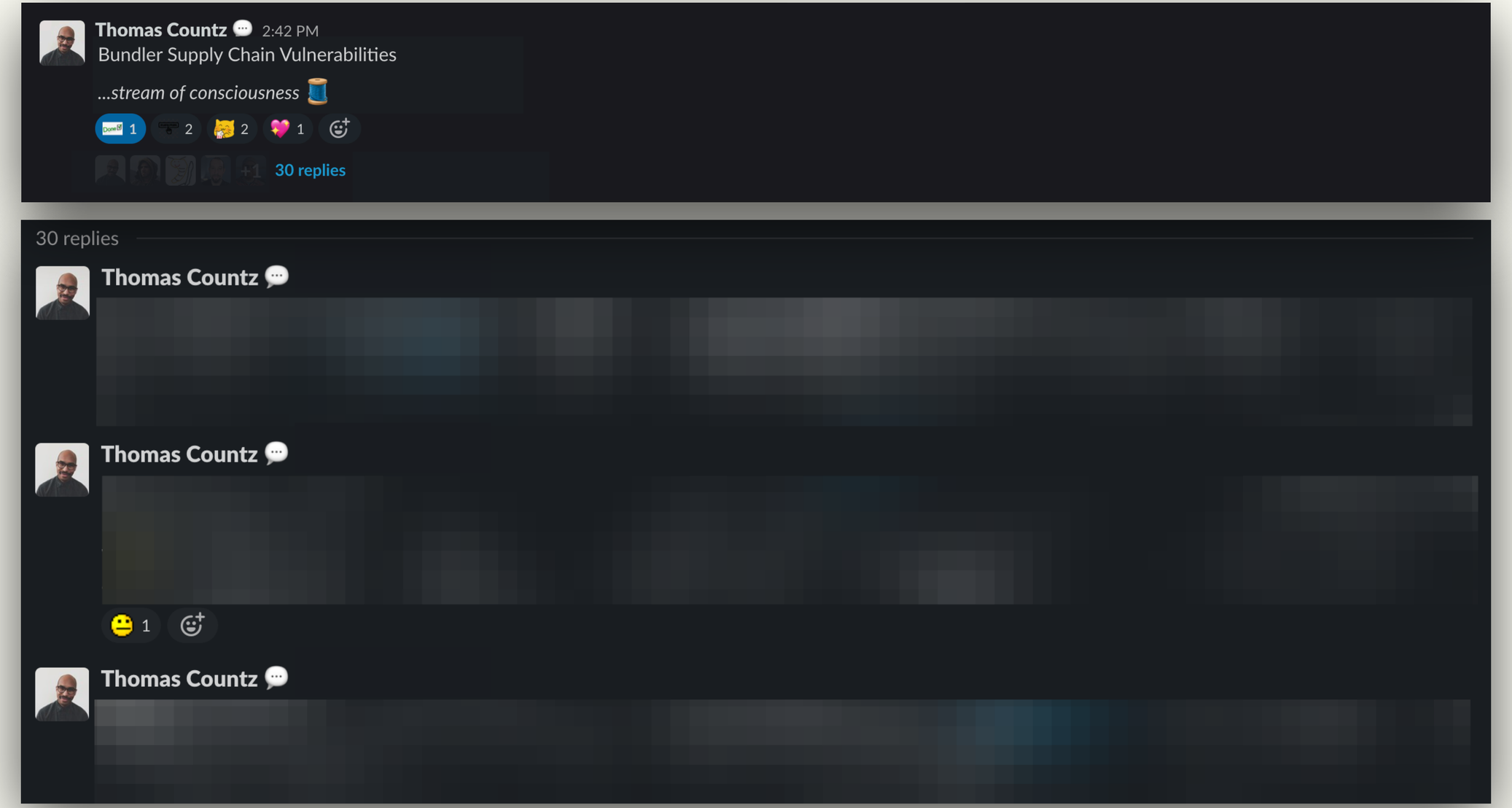
MAKE CODE EXECUTABLE FOR BETTER FEEDBACK

PREPARE FOR POST-DEPLOYMENT



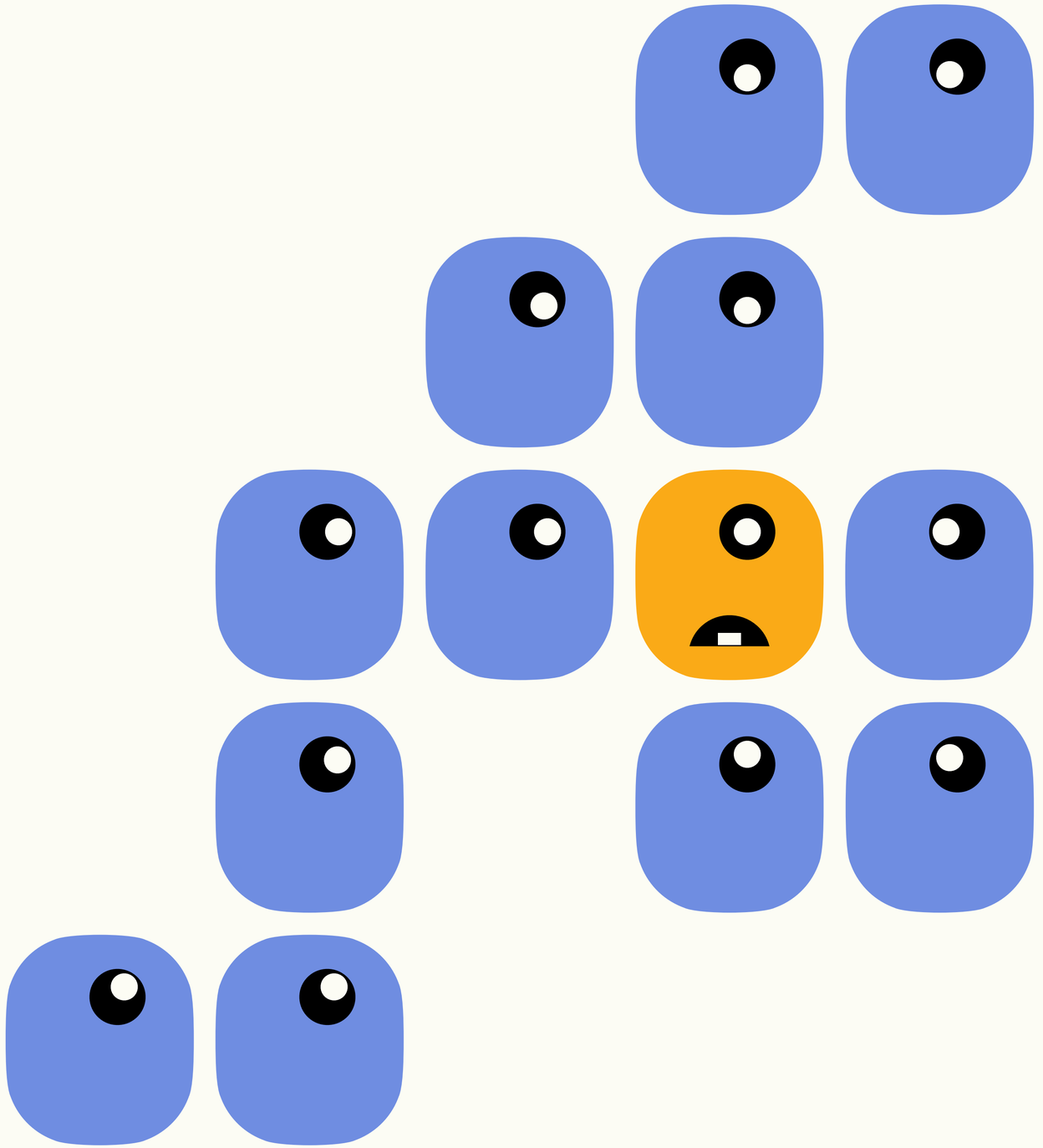
```
# Users have first and last names
1. How do new users sign up? - important!
2. How do we store users' information? - important!
3. What about multiple given names? - can wait
4. Didn't Sarah look at something similar last
   quarter? - maybe irrelevant
```

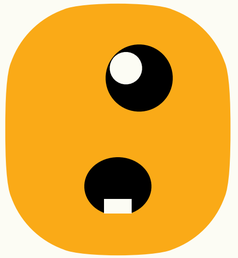
ASK FIRST; SOLVE LATER - **SHARE YOUR FINDINGS** - MAKE CODE EXECUTABLE - PREPARE FOR POST-DEPLOYMENT

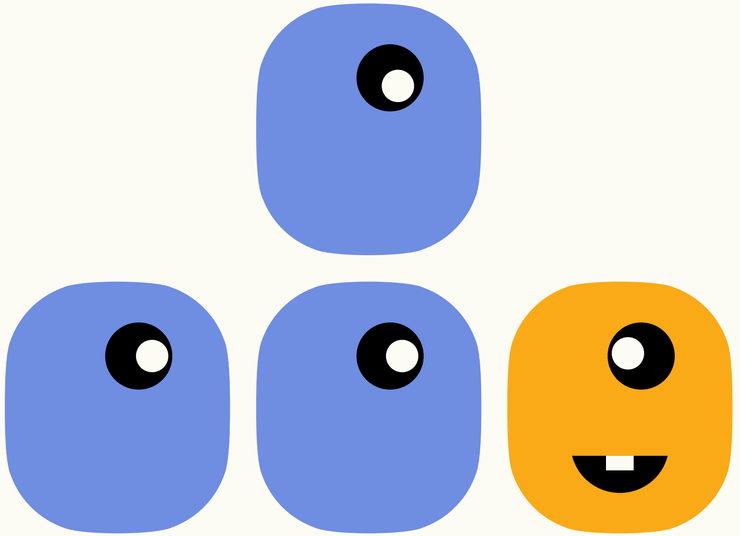


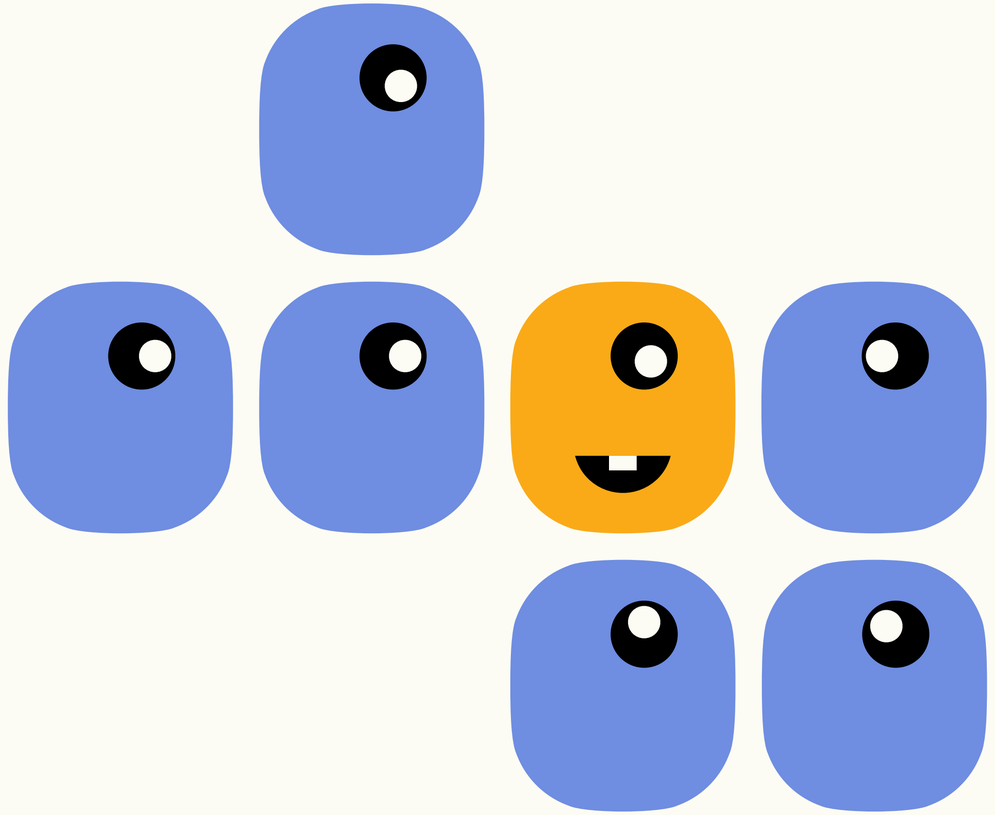
```ruby
class User < ActiveRecord::Base
  def split_names!; end
end
```

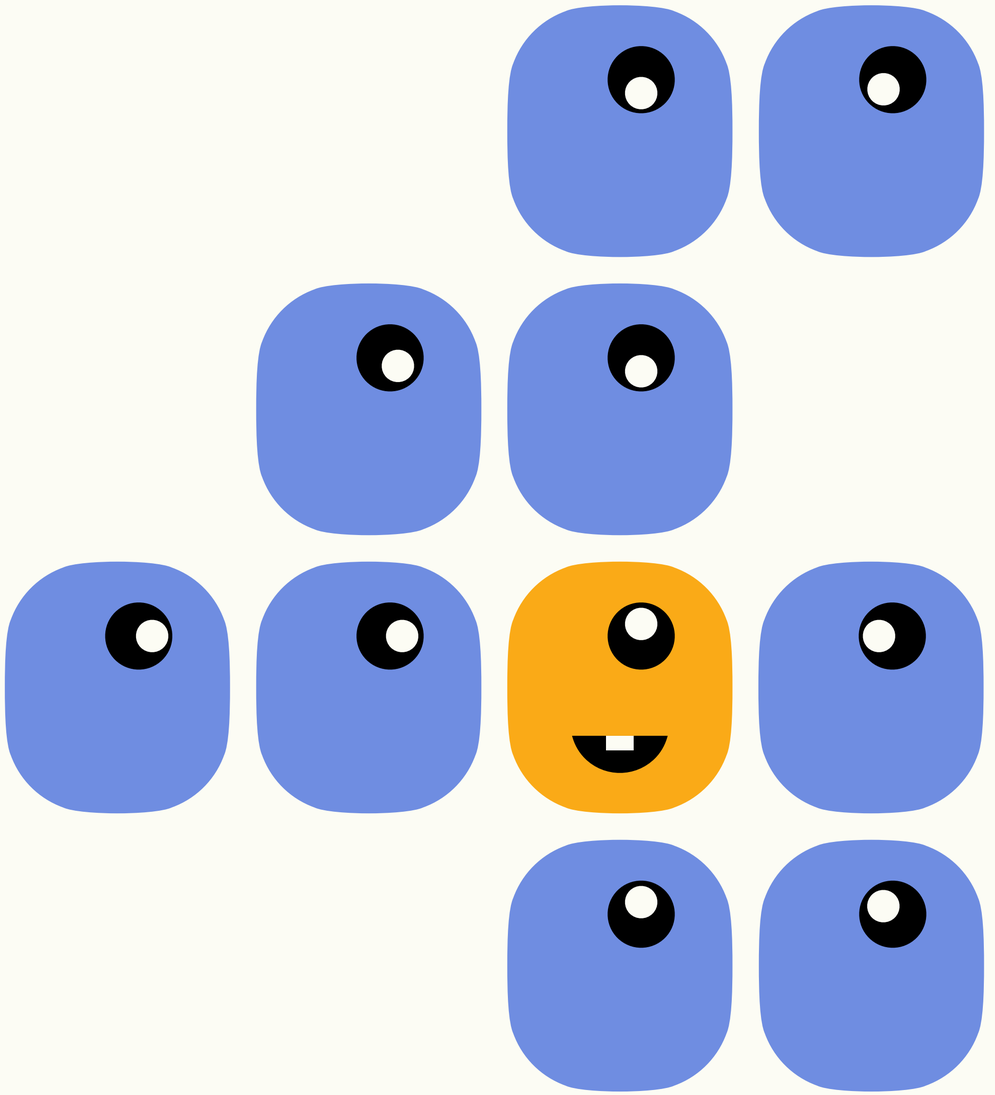
```ruby
RSpec.describe User do
  describe "#split_names!" do
    it "splits the first_name into first_name & last_name"
  end
end
```

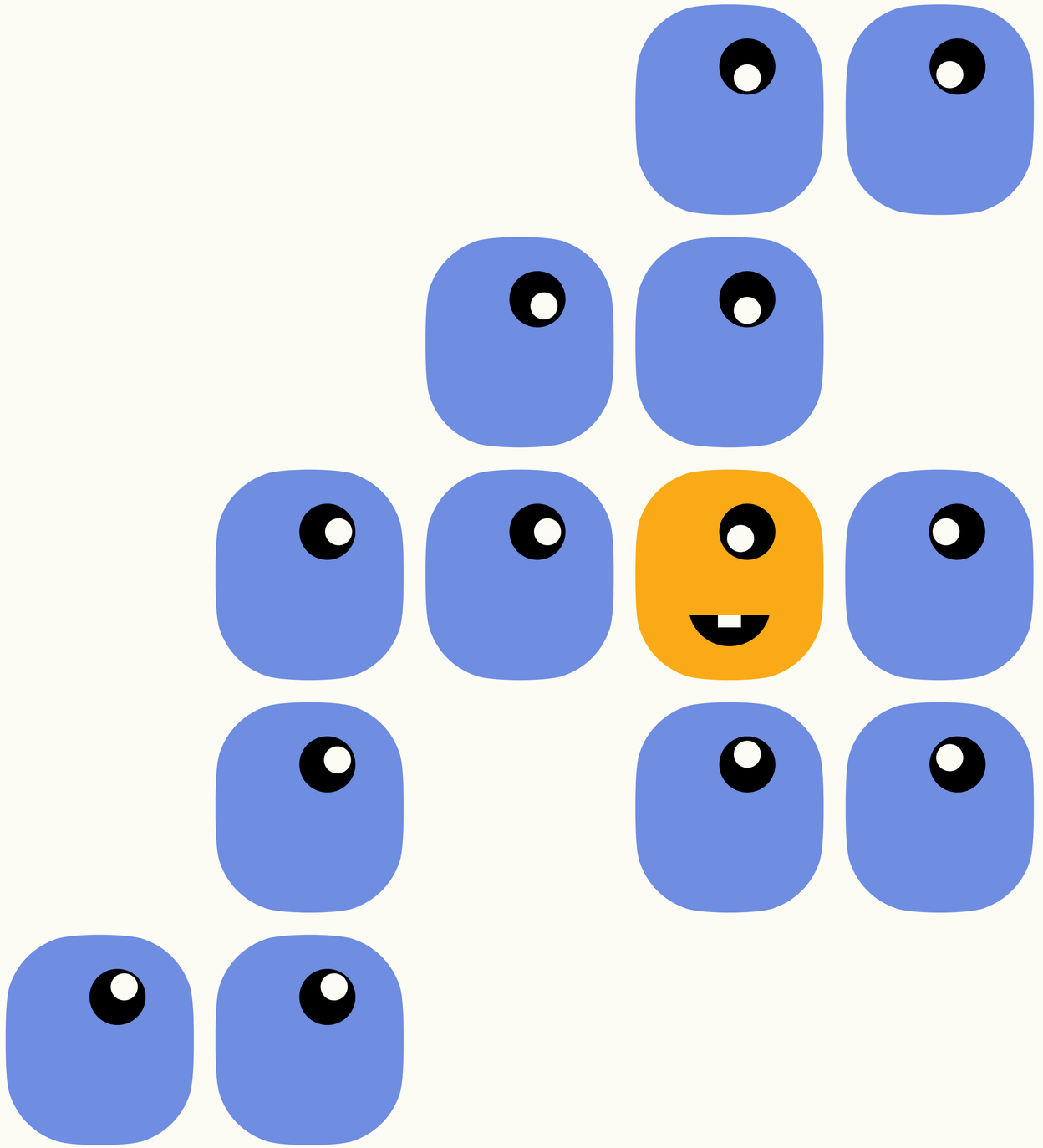








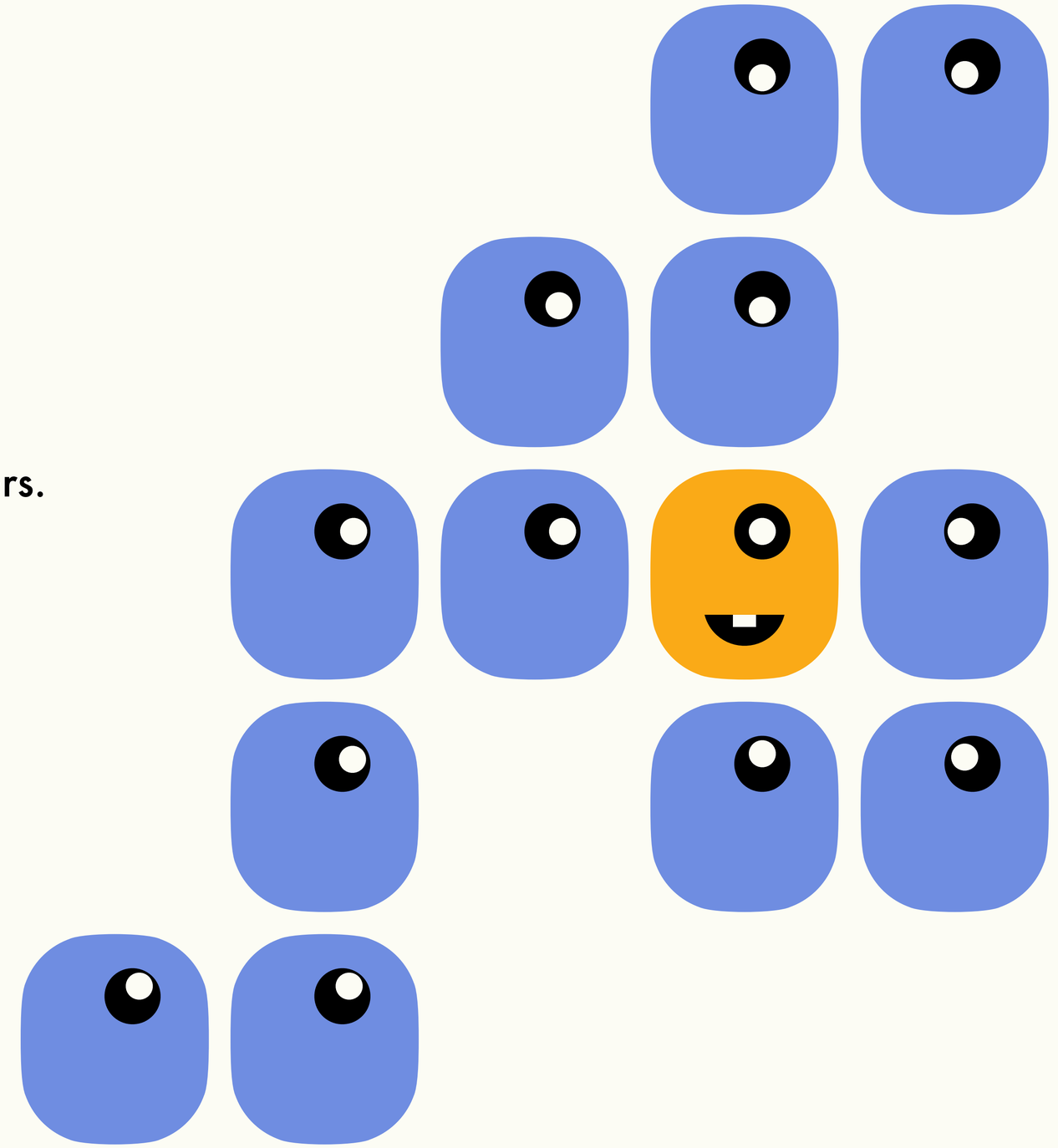




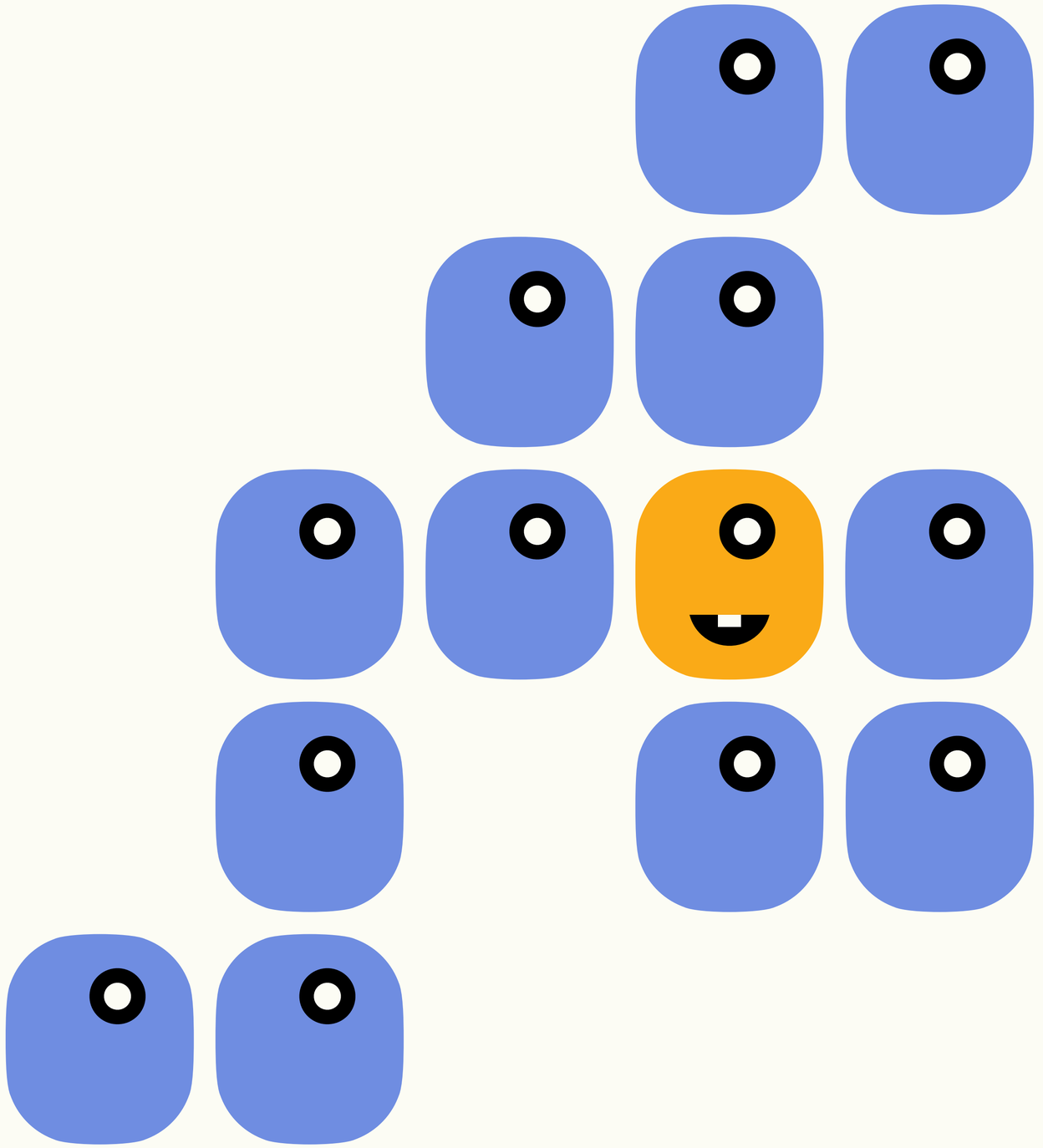
Software engineers never have all of the answers.

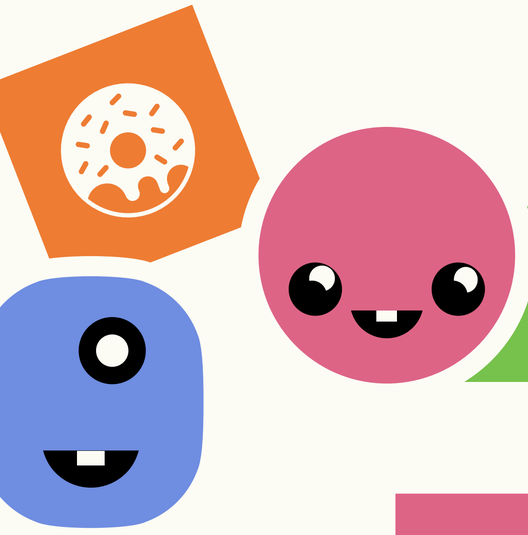
They're good at being curious.

I bet you're good at being curious, too!

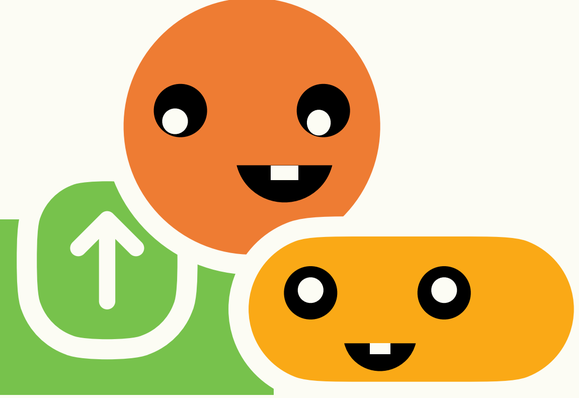


THANKS!



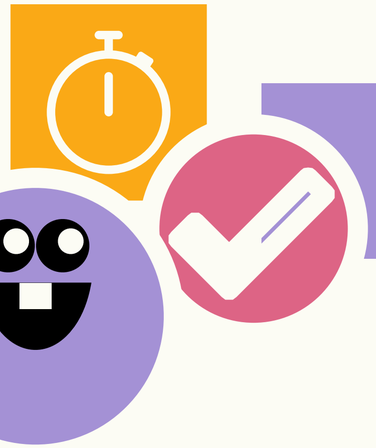


ASK FIRST; SOLVE LATER



SHARE YOUR FINDINGS AS YOU DISCOVER THEM

MAKE CODE EXECUTABLE FOR BETTER FEEDBACK



PREPARE FOR POST-DEPLOYMENT