Jifty Now!

♨

Audrey Tang

Jifty Now!

♨

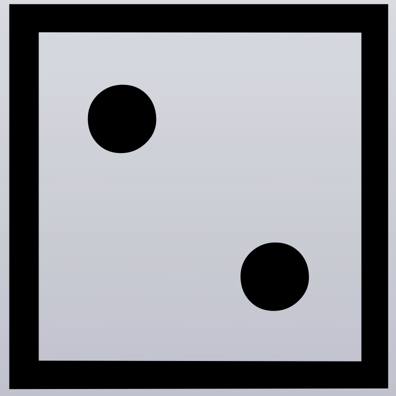
Audrey Tang

Jesse Vincent

Jifty Now!

♨

Pretending to be audreyt



April 1, 2007

April 1, 2007

- Going to Japan tomorrow

April 1, 2007

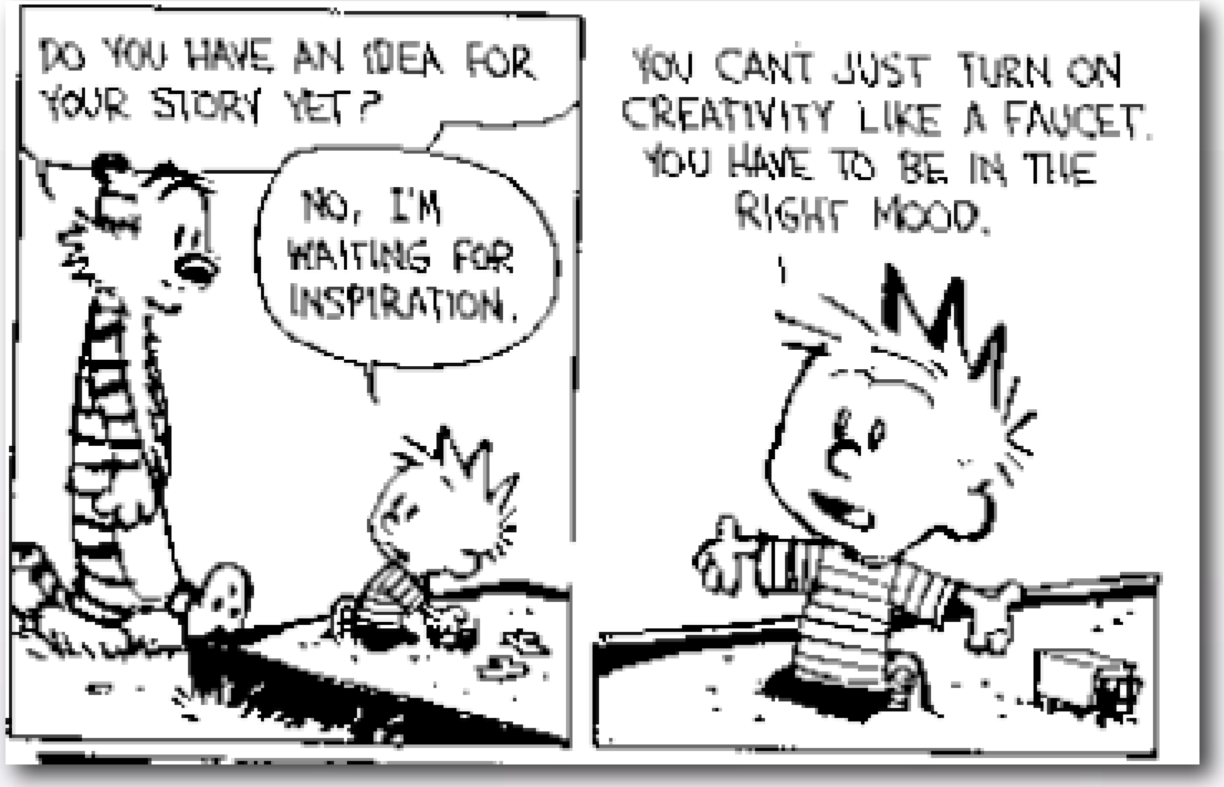
- Going to Japan tomorrow
- Two talks at YAPC::Asia

April 1, 2007

- Going to Japan tomorrow
- Two talks at YAPC::Asia
- Panic: No slides!

JIT

JIT


DO YOU HAVE AN IDEA FOR YOUR STORY YET?
NO, I'M WAITING FOR INSPIRATION.
YOU CAN'T JUST TURN ON CREATIVITY LIKE A FAUCET. YOU HAVE TO BE IN THE RIGHT MOOD.

JIT

DO YOU HAVE AN IDEA FOR YOUR STORY YET?
NO, I'M WAITING FOR INSPIRATION.
YOU CAN'T JUST TURN ON CREATIVITY LIKE A FAUCET. YOU HAVE TO BE IN THE RIGHT MOOD.
WHAT MOOD IS THAT?
LAST-MINUTE PANIC.

Dilemma

Dilemma

- ***Two* talks to work on**

Dilemma

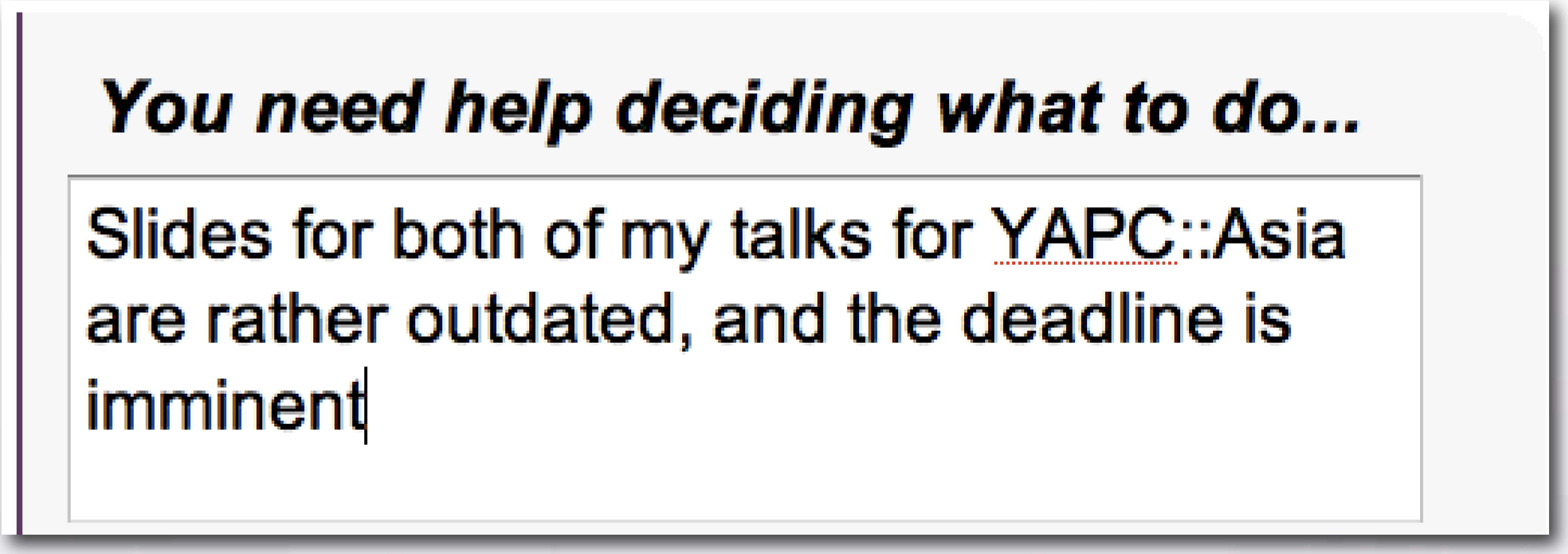
- *Two* talks to work on
- Which one first?

Dilemma

- *Two* talks to work on
- Which one first?
- Panic: Can’t decide!

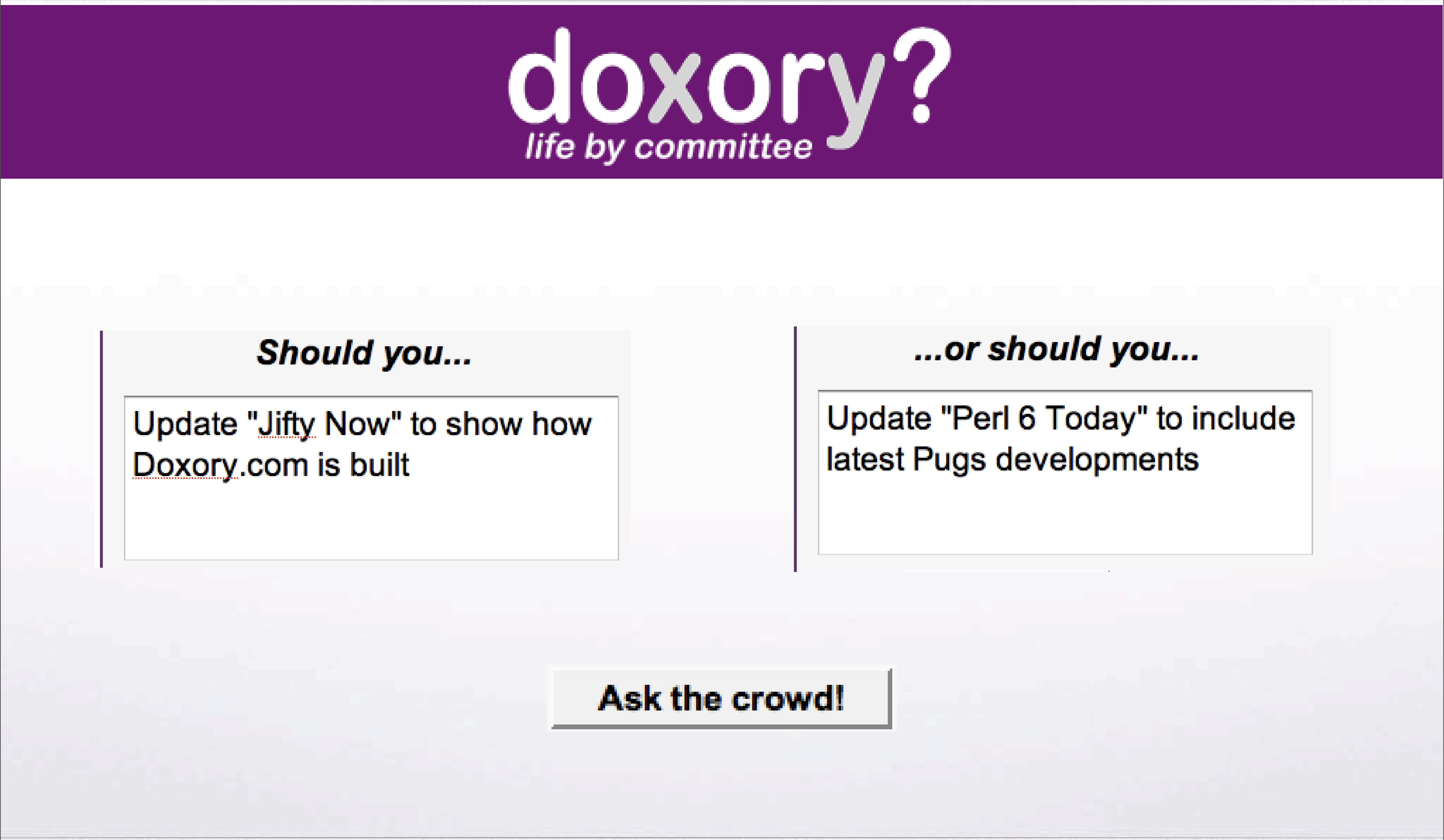
doxory?

life by committee



You need help deciding what to do...

Slides for both of my talks for YAPC::Asia are rather outdated, and the deadline is imminent

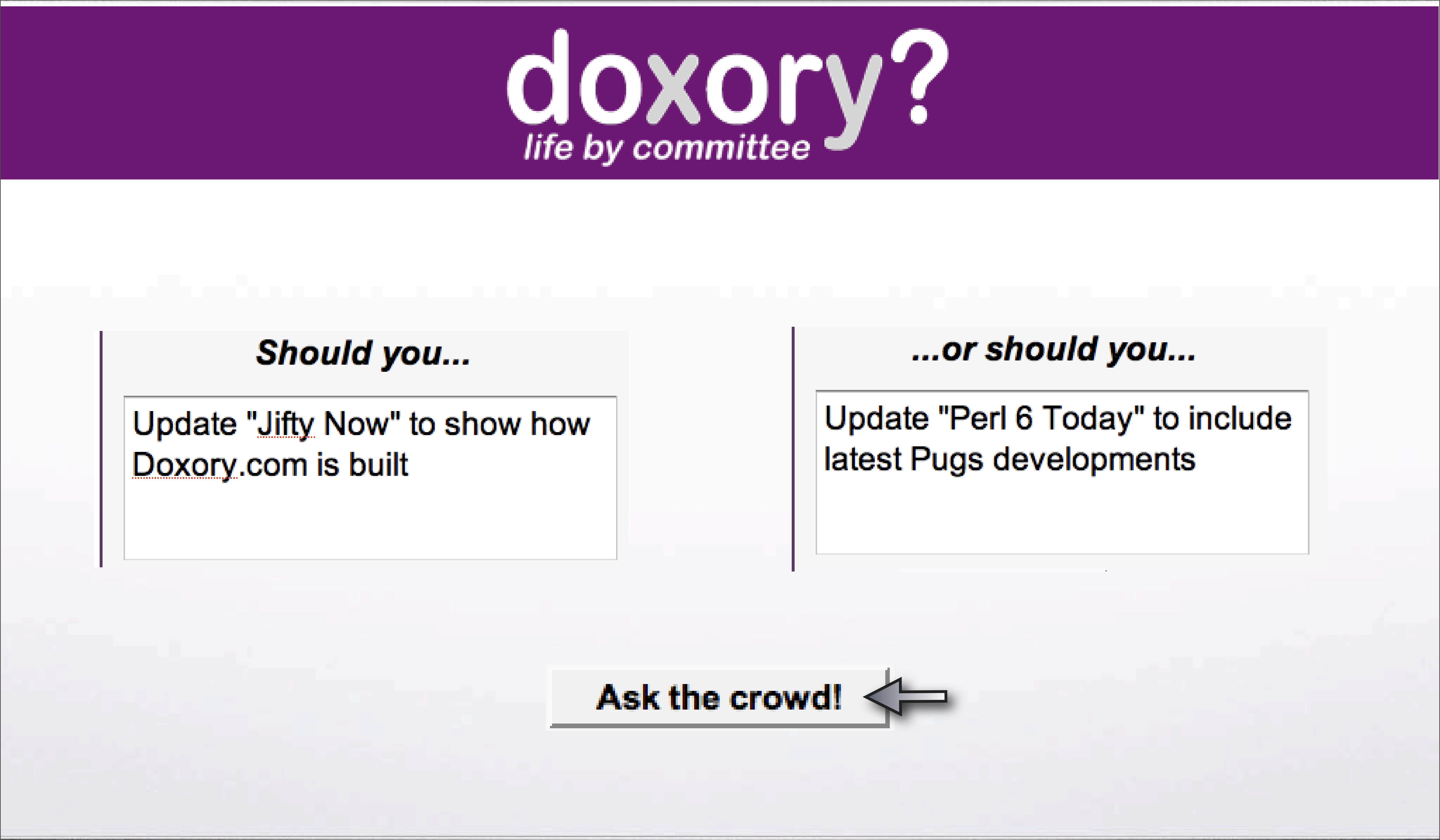


Should you...

Update "Jifty Now" to show how Doxory.com is built

...or should you...

Update "Perl 6 Today" to include latest Pugs developments



doxory?

life by committee

Should you...

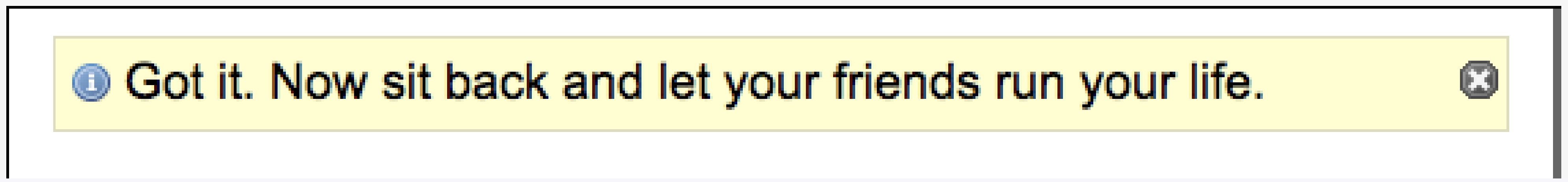
Update "Jifty Now" to show how Doxory.com is built

...or should you...

Update "Perl 6 Today" to include latest Pugs developments

doxory?

life by committee



Got it. Now sit back and let your friends run your life.

Feeling *much* Better

Feeling *much* Better

- **Waiting for a decision**

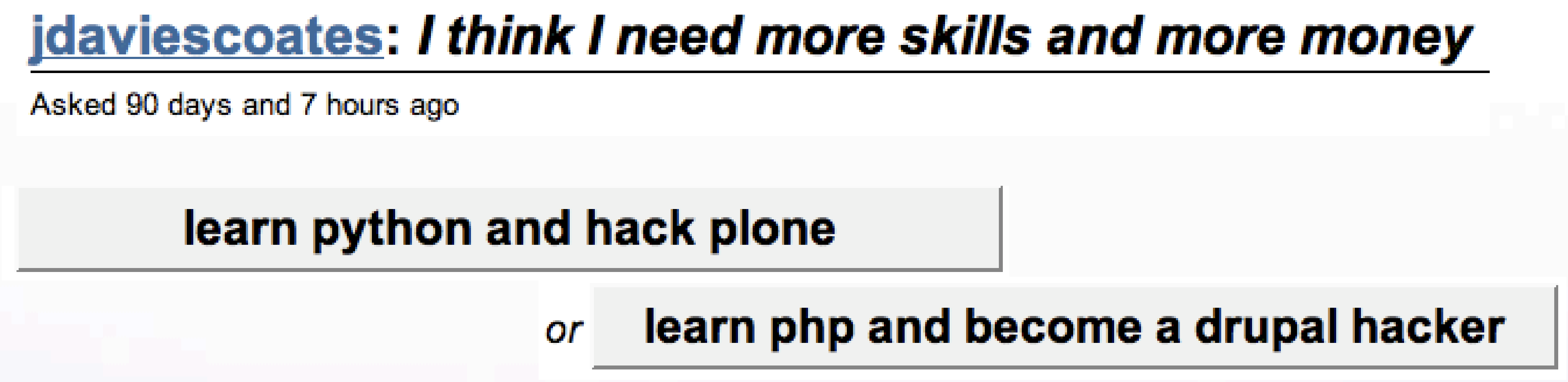
Feeling *much* Better

- Waiting for a decision
- No need to work just now

Feeling *much* Better

- Waiting for a decision
- No need to work just now
- So I have time to help people!

Pick!

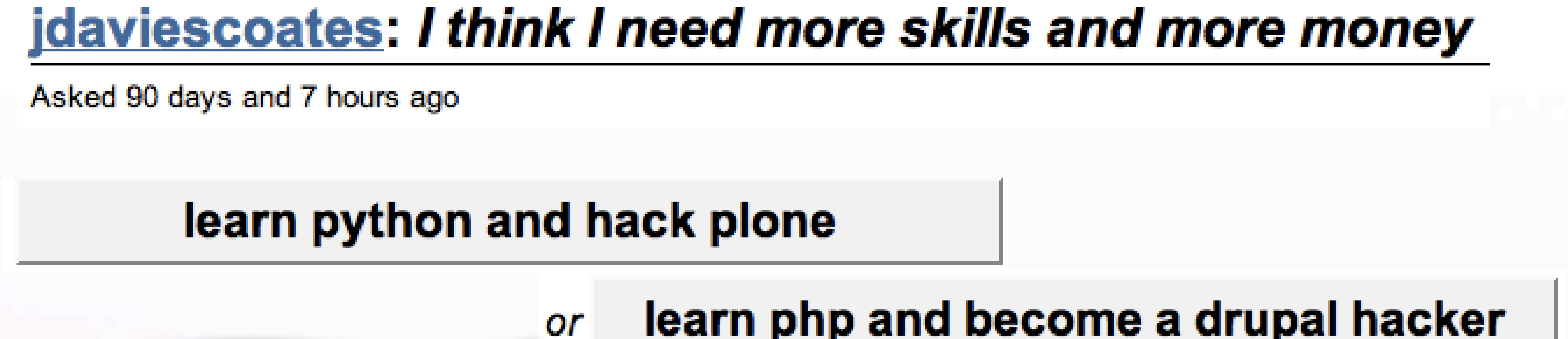


jdaviescoates: *I think I need more skills and more money*

Asked 90 days and 7 hours ago

learn python and hack plone
or
learn php and become a drupal hacker

Pick!



jdaviescoates: *I think I need more skills and more money*

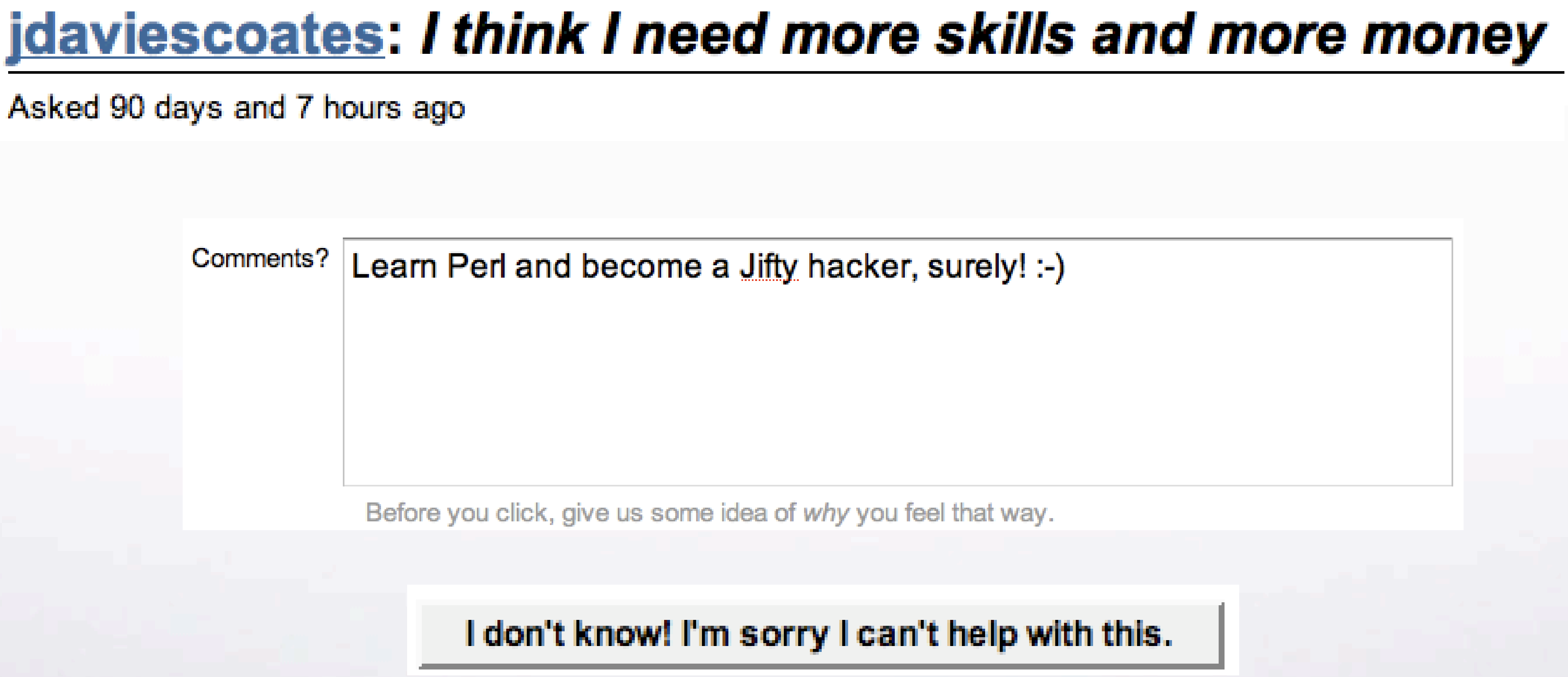
Asked 90 days and 7 hours ago

learn python and hack plone

or learn php and become a drupal hacker

“False Dilemma”

Pick!



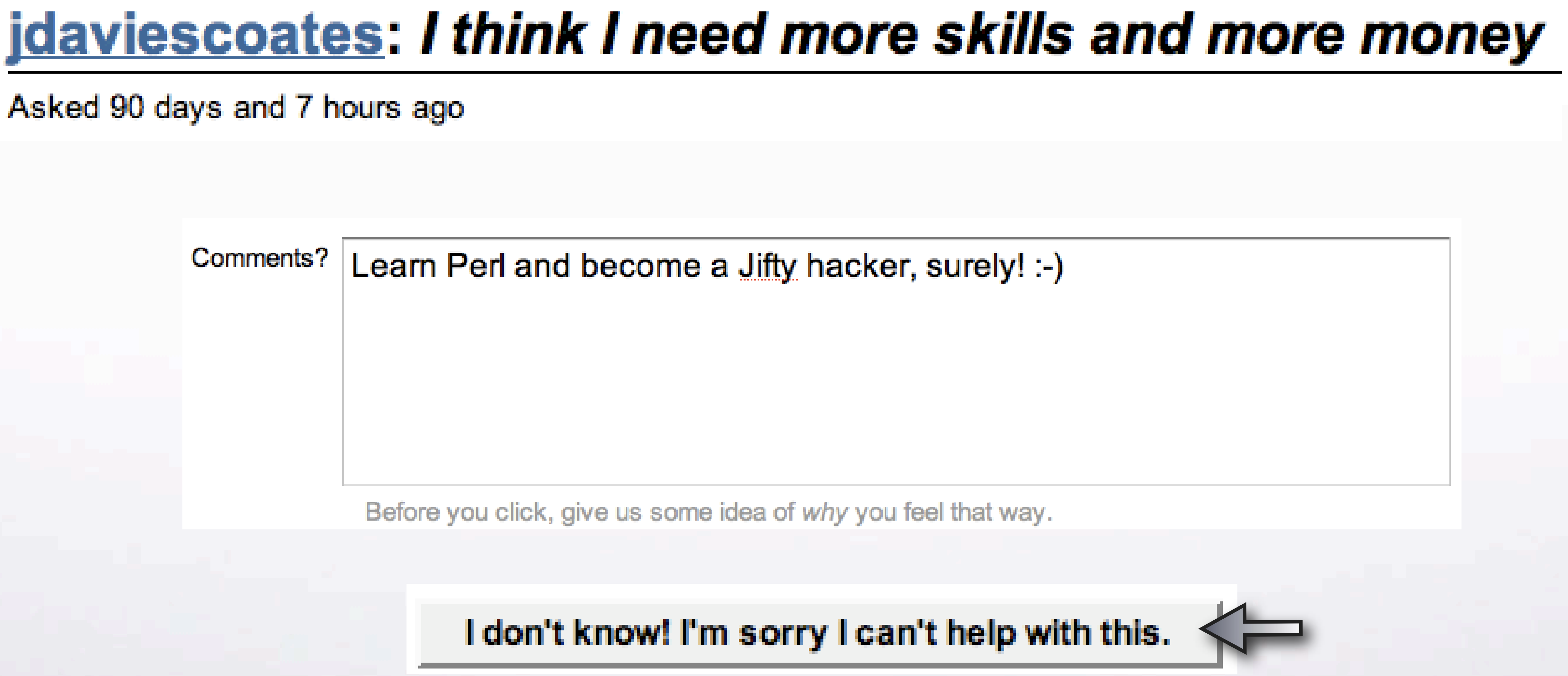
jdaviescoates: *I think I need more skills and more money*

Asked 90 days and 7 hours ago

Comments? Learn Perl and become a Jifty hacker, surely! :-)

Before you click, give us some idea of *why* you feel that way.

Pick!



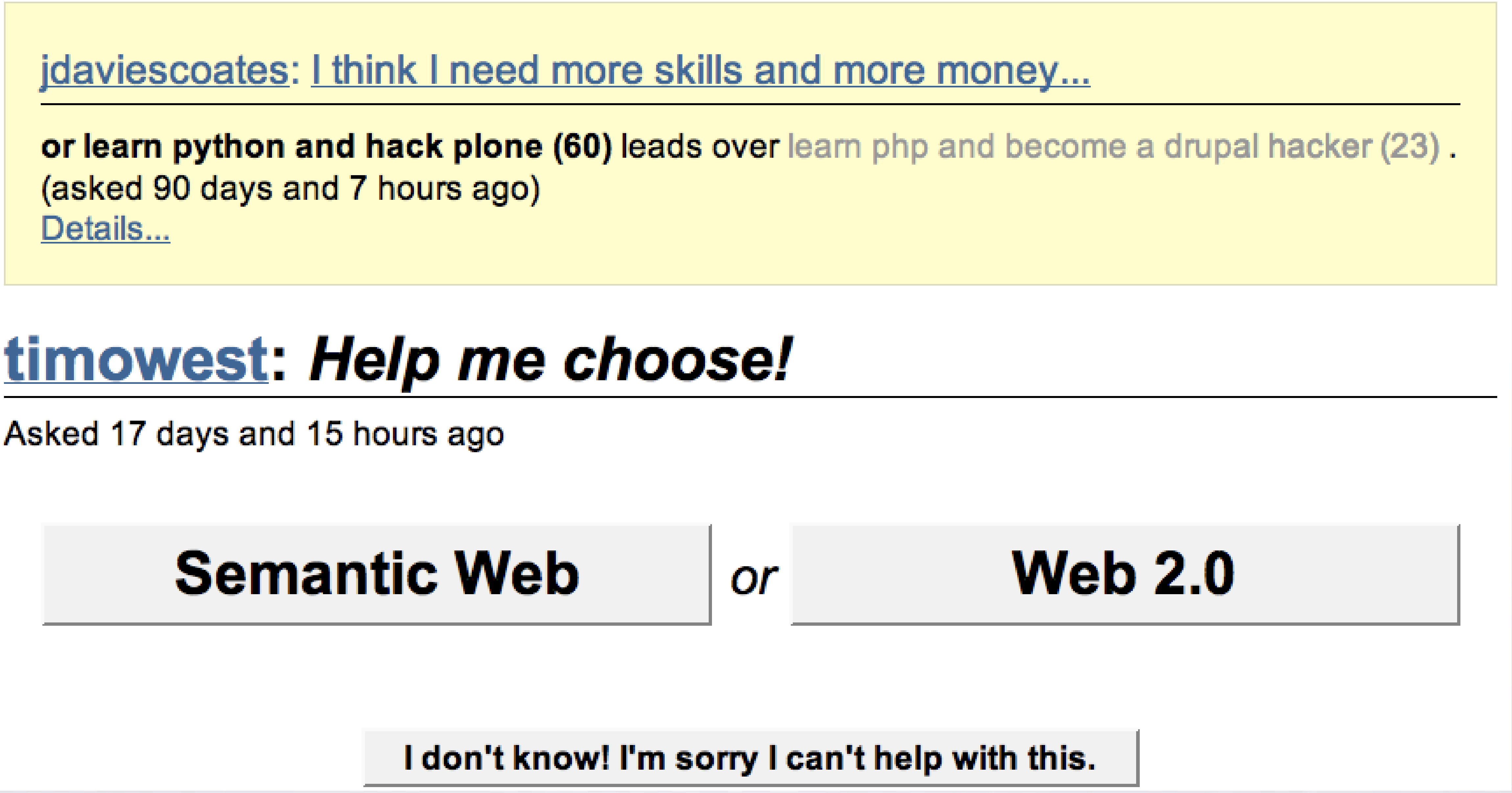
jdaviescoates: *I think I need more skills and more money*

Asked 90 days and 7 hours ago

Comments? Learn Perl and become a Jifty hacker, surely! :-)

Before you click, give us some idea of *why* you feel that way.

Pick!



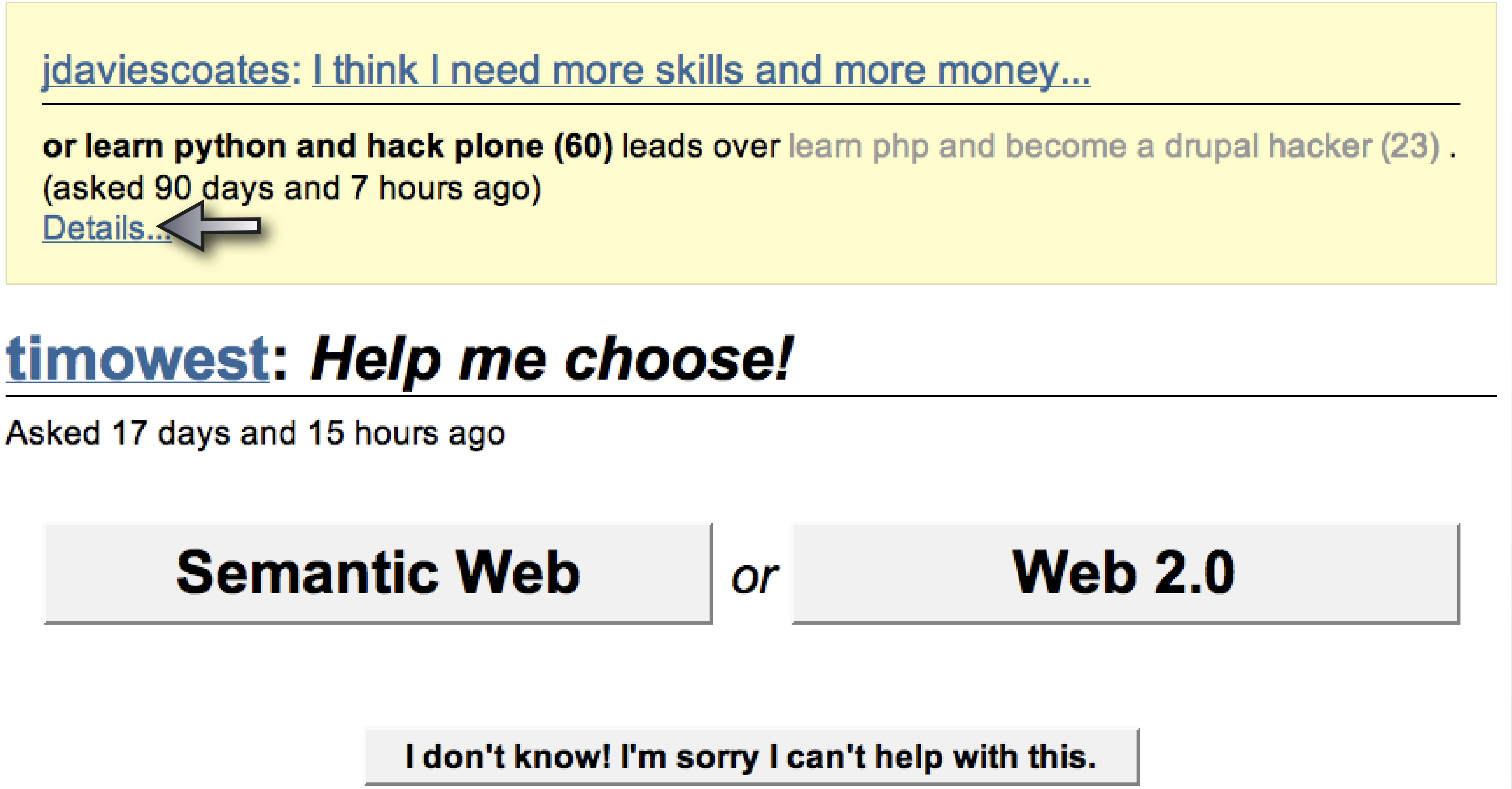
jdaviescoates: I think I need more skills and more money...

or learn python and hack plone (60) leads over learn php and become a drupal hacker (23) .
(asked 90 days and 7 hours ago)
Details...

timowest: *Help me choose!*

Asked 17 days and 15 hours ago

Pick!



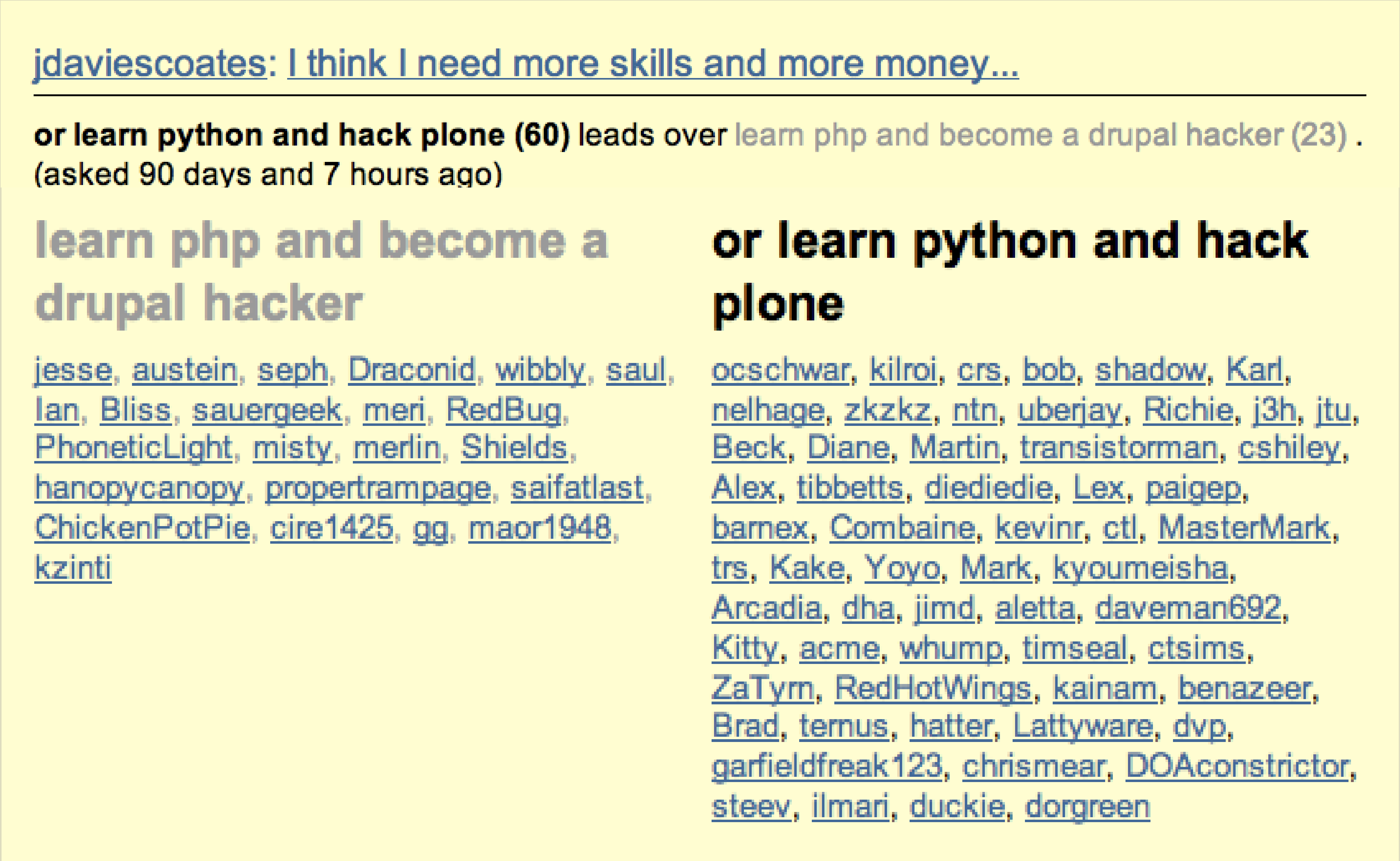
jdaviescoates: I think I need more skills and more money...

or learn python and hack plone (60) leads over learn php and become a drupal hacker (23) .
(asked 90 days and 7 hours ago)
Details...

timowest: ***Help me choose!***

Asked 17 days and 15 hours ago

Pick!



jdaviescoates: I think I need more skills and more money...

or learn python and hack plone (60) leads over learn php and become a drupal hacker (23) . (asked 90 days and 7 hours ago)

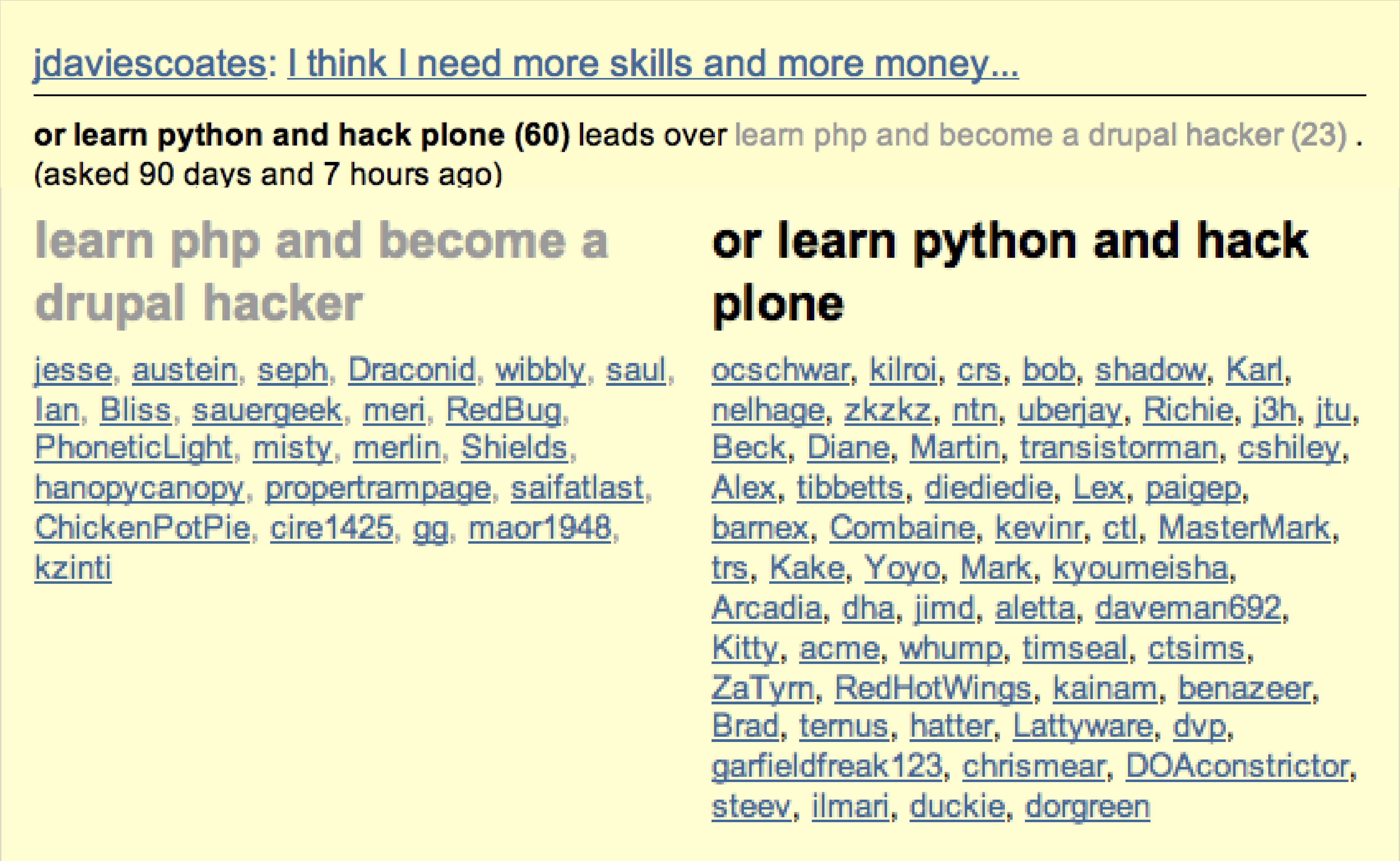
learn php and become a drupal hacker

jesse, austein, seph, Draconid, wibbly, saul, Ian, Bliss, sauergeek, meri, RedBug, PhoneticLight, misty, merlin, Shields, hanopycanopy, propertrampage, saifatlast, ChickenPotPie, cire1425, gg, maor1948, kzinti

or learn python and hack plone

ocschwar, kilroi, crs, bob, shadow, Karl, nelhage, zkzkz, ntn, uberjay, Richie, j3h, jtu, Beck, Diane, Martin, transistorman, cshiley, Alex, tibbetts, diediedie, Lex, paigep, barnex, Combaine, kevinr, ctl, MasterMark, trs, Kake, Yoyo, Mark, kyoumeisha, Arcadia, dha, jimd, aletta, daveman692, Kitty, acme, whump, timseal, ctsims, ZaTyrn, RedHotWings, kainam, benazeer, Brad, ternus, hatter, Lattyware, dvp, garfieldfreak123, chrismear, DOAconstrictor, steev, ilmari, duckie, dorgreen

Pick!



jdaviescoates: I think I need more skills and more money...

or learn python and hack plone (60) leads over learn php and become a drupal hacker (23) .
(asked 90 days and 7 hours ago)

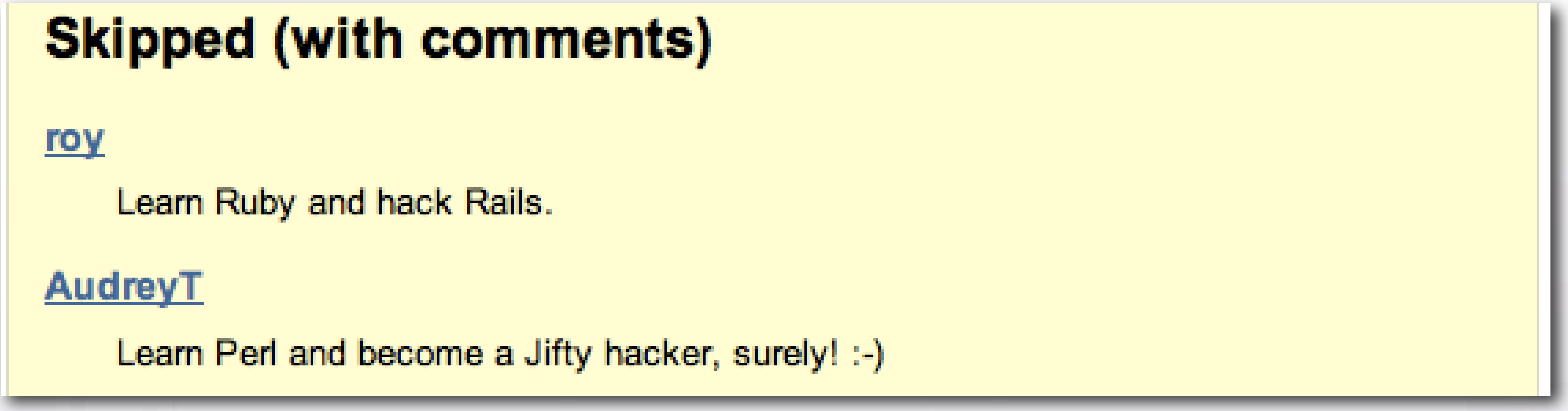
learn php and become a drupal hacker

jesse, austein, seph, Draconid, wibbly, saul, Ian, Bliss, sauergeek, meri, RedBug, PhoneticLight, misty, merlin, Shields, hanopycanopy, propertrampage, saifatlast, ChickenPotPie, cire1425, gg, maor1948, kzinti

or learn python and hack plone

ocschwar, kilroi, crs, bob, shadow, Karl, nelhage, zkzkz, ntn, uberjay, Richie, j3h, jtu, Beck, Diane, Martin, transistorman, cshiley, Alex, tibbetts, diediedie, Lex, paigep, barnex, Combaine, kevinr, ctl, MasterMark, trs, Kake, Yoyo, Mark, kyoumeisha, Arcadia, dha, jimd, aletta, daveman692, Kitty, acme, whump, timseal, ctsims, ZaTyrn, RedHotWings, kainam, benazeer, Brad, ternus, hatter, Lattyware, dvp, garfieldfreak123, chrismear, DOAconstrictor, steev, ilmari, duckie, dorgreen

Pick!



Skipped (with comments)

roy

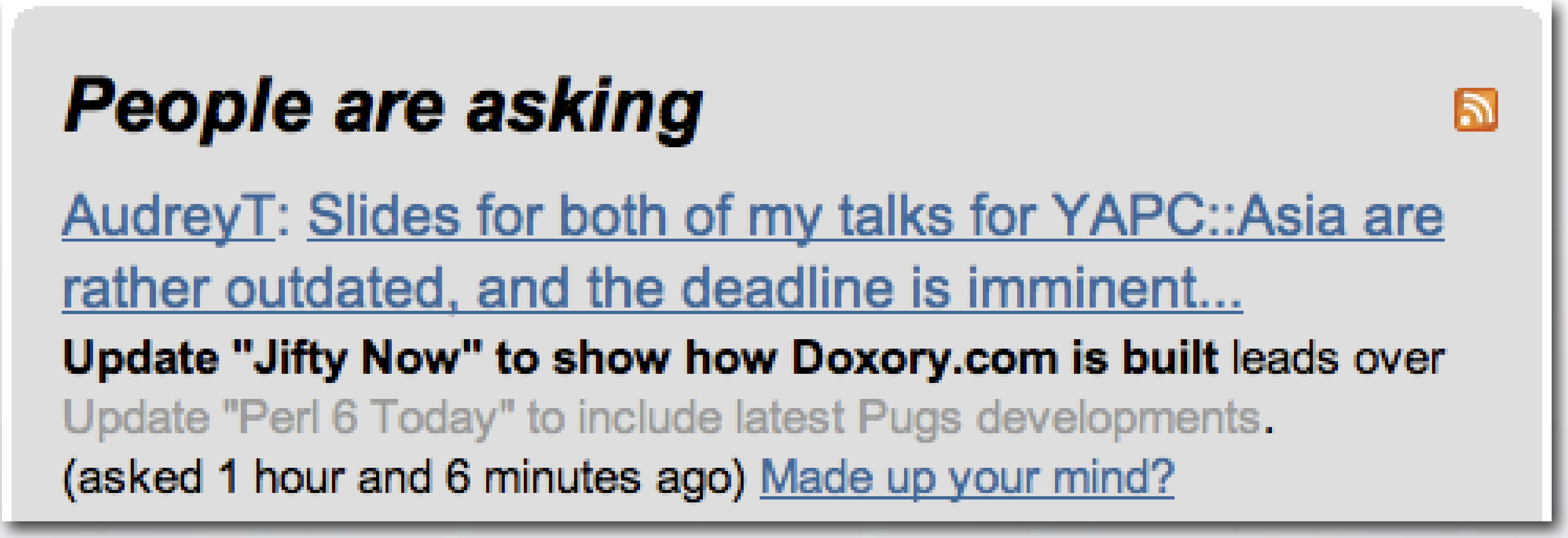
Learn Ruby and hack Rails.

AudreyT

Learn Perl and become a Jifty hacker, surely! :-)

An hour passes...

...we have a decision!



People are asking

AudreyT: Slides for both of my talks for YAPC::Asia are rather outdated, and the deadline is imminent...

Update "Jifty Now" to show how Doxory.com is built leads over Update "Perl 6 Today" to include latest Pugs developments.

(asked 1 hour and 6 minutes ago) Made up your mind?

Jifty

Jifty

- ♨ Web framework

Jifty

- ♨ Web framework
- ♨ Easy things easy

Jifty

♨ Web framework

♨ Easy things easy

♨ Impossible things *easy*

Jifty = JFDI

Jifty = JFDI

♨ Just Do It

Jifty = JFDI

- ♨ Just Do It
- ♨ Lets you JFDI

Jifty = JFDI

♨ Just Do It

♨ Lets you JFDI

♨ So let’s JFDI!

```
$ cpan
$ cpan
cpan> notest install Jifty
```

```
$ cpan
cpan> notest install Jifty
```

“notest” saves ~60 minutes here.

```
$ jifty app --name Doxory
$ jifty app --name Doxory
Creating new application Doxory
Creating directory lib
Creating directory lib/Doxory
Creating directory bin
Creating directory etc
Creating directory doc
Creating directory log
Creating directory var
Creating directory var/mason
Creating directory share
Creating directory share/po
Creating directory share/web
Creating directory share/web/templates
Creating directory share/web/static
Creating directory lib/Doxory/Model
Creating directory lib/Doxory/Action
Creating directory t
Creating configuration file Doxory/etc/config.yml
```

3 Files

3 Files

♨ `bin/jifty`

3 Files

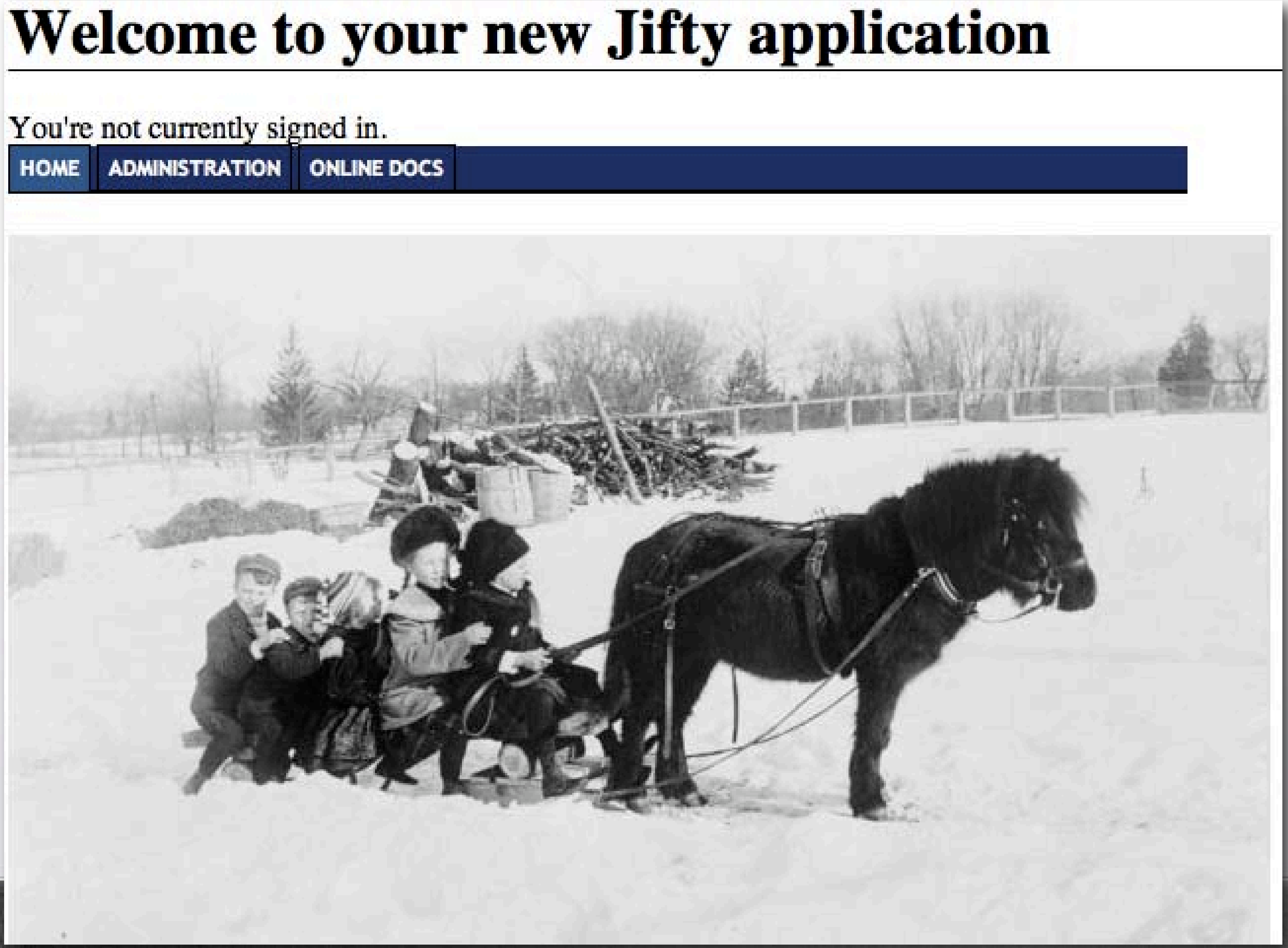
♨ `bin/jifty`

♨ `etc/config.yml`

3 Files

- bin/jifty
- etc/config.yml
- Makefile.PL

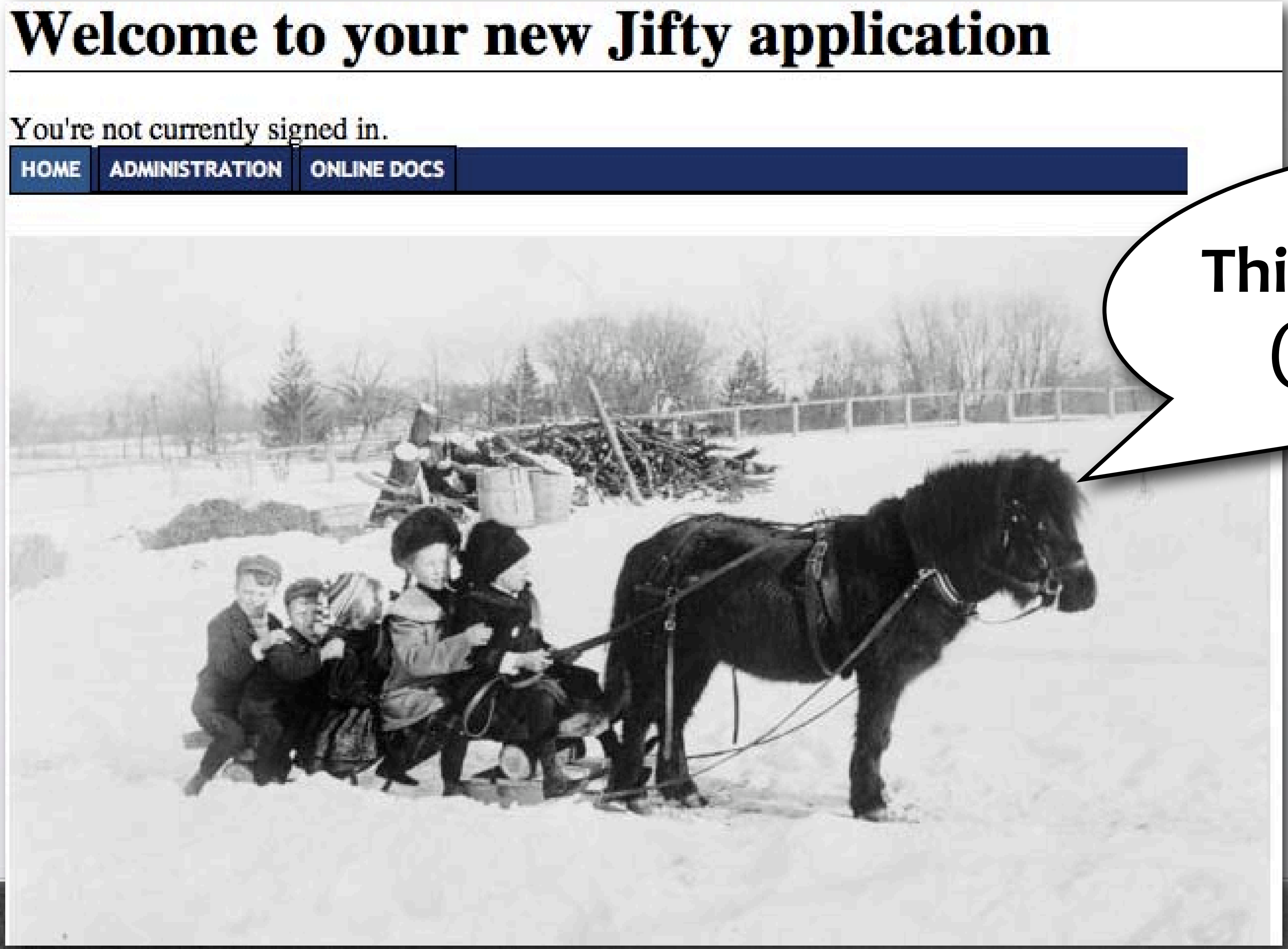
It runs!



Welcome to your new Jifty application

You're not currently signed in.

It runs!



Welcome to your new Jifty application

You're not currently signed in.

This is not really Doxory.
(Doxory has Users.)

```
$ cd Doxory
$ cd Doxory

$ jifty model --name User
$ cd Doxory

$ jifty model --name User
Writing file t/00-model-User.t
Writing file lib/Doxory/Model/User.pm
```

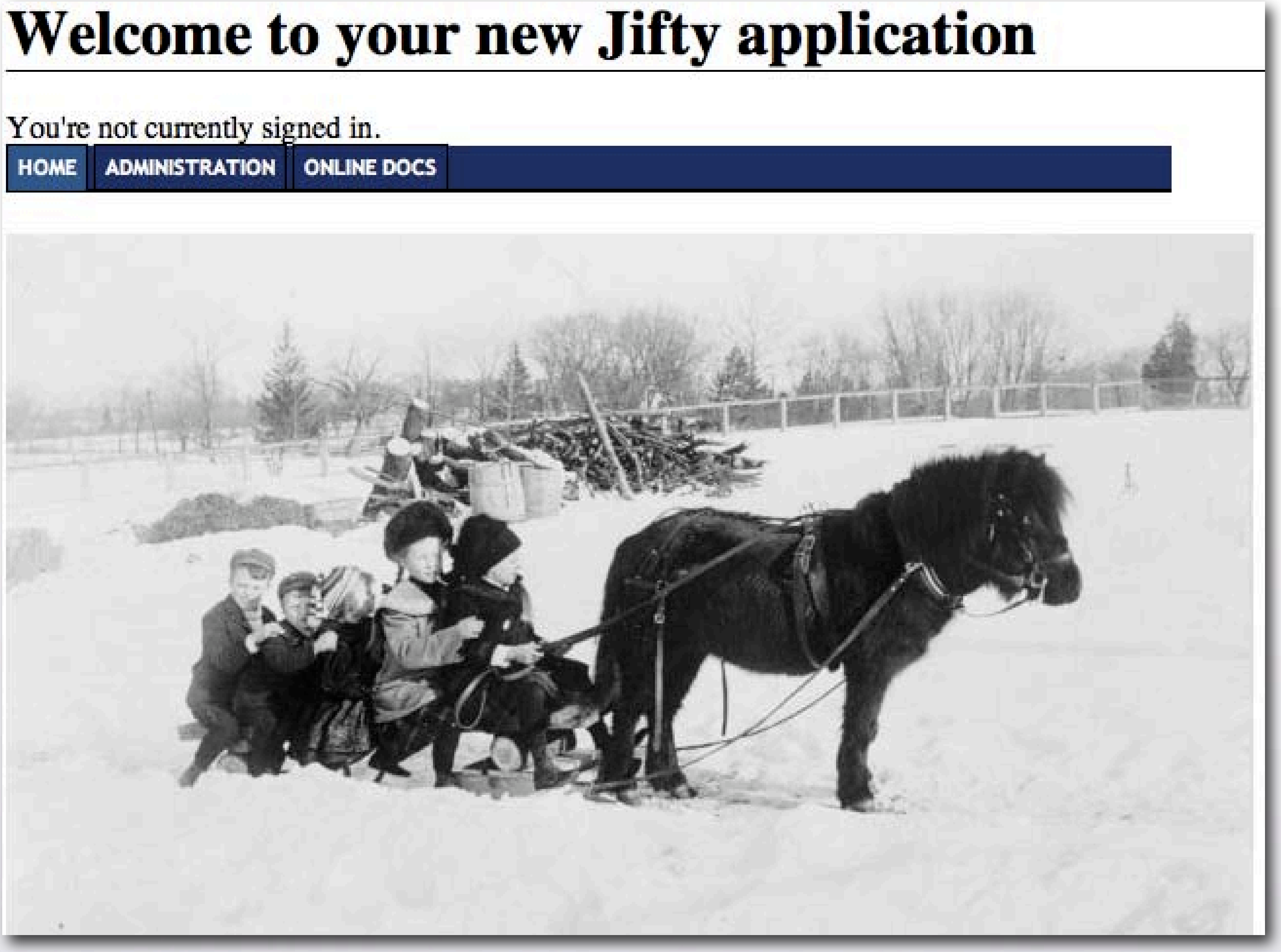
```
$ cd Doxory

$ jifty model --name User
Writing file t/00-model-User.t
Writing file lib/Doxory/Model/User.pm

$ vim lib/Doxory/Model/User.pm
```

```
package Doxory::Model::User;
use Jifty::DBI::Schema;

use Doxory::Record schema {






};
```

```
package Doxory::Model::User;
use Jifty::DBI::Schema;

use Doxory::Record schema {

column name =>
    label is 'Username',
    hints is 'Other people see this when you ask questions.',
    is distinct,
    is mandatory;

column email =>
    label is 'Email',
    hints is 'We will not sell it to anybody. Honest.',
    default is '',
    is distinct,
    is immutable;

};
```

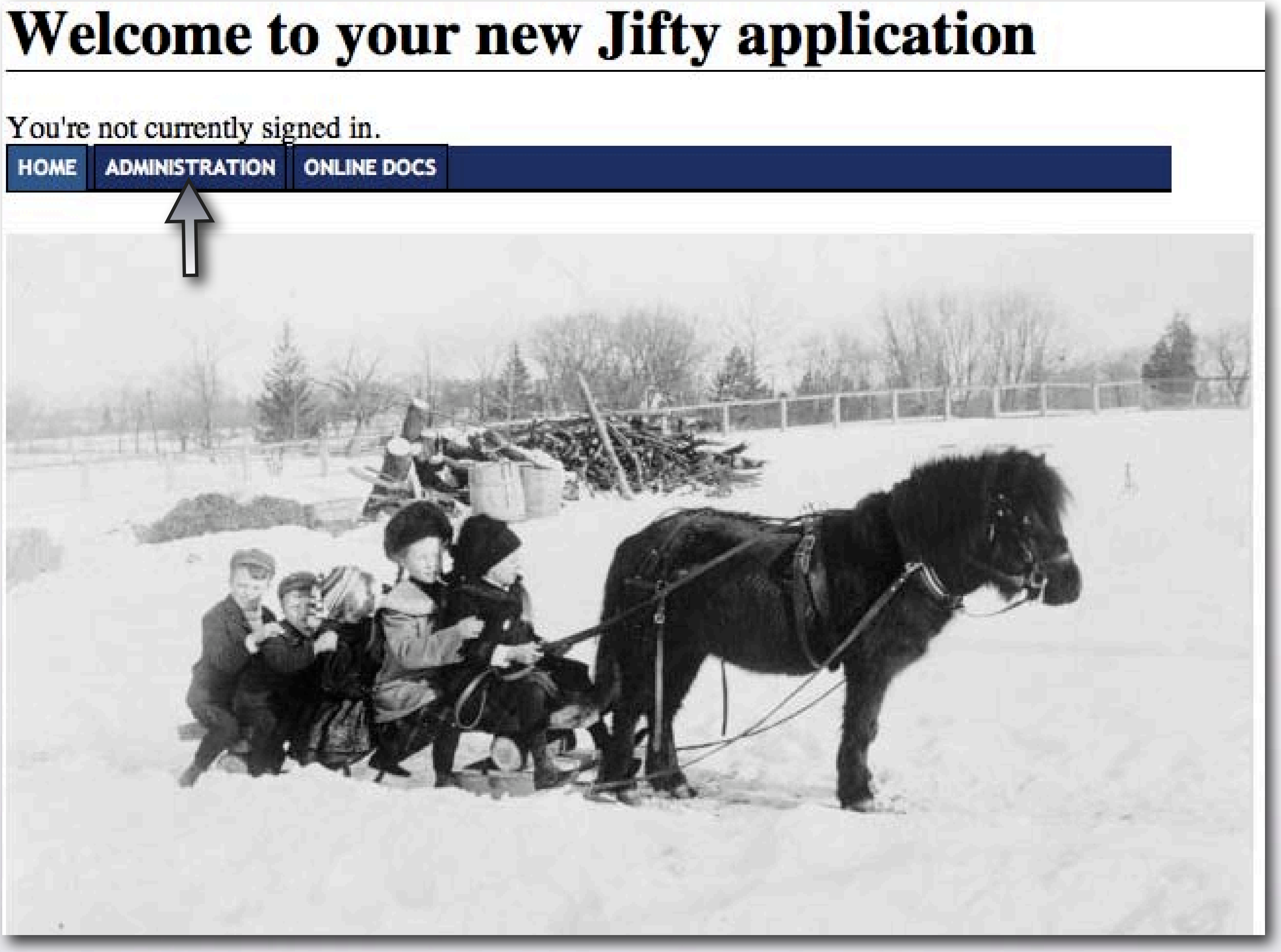
```
$ jifty schema
$ jifty schema
INFO - Generating SQL for application Doxory...
INFO - Using Doxory::Model::User
INFO - Using Jifty::Model::Session
INFO - Using Jifty::Model::Metadata
INFO - Set up version v0.0.1, jifty version 0.701170
```

```
$ jifty schema
INFO - Generating SQL for application Doxory...
INFO - Using Doxory::Model::User
INFO - Using Jifty::Model::Session
INFO - Using Jifty::Model::Metadata
INFO - Set up version v0.0.1, jifty version 0.701170

$ jifty server
$ jifty schema
INFO - Generating SQL for application Doxory...
INFO - Using Doxory::Model::User
INFO - Using Jifty::Model::Session
INFO - Using Jifty::Model::Metadata
INFO - Set up version v0.0.1, jifty version 0.701170

$ jifty server
INFO - You can connect to your server at
       http://localhost:8888/
```



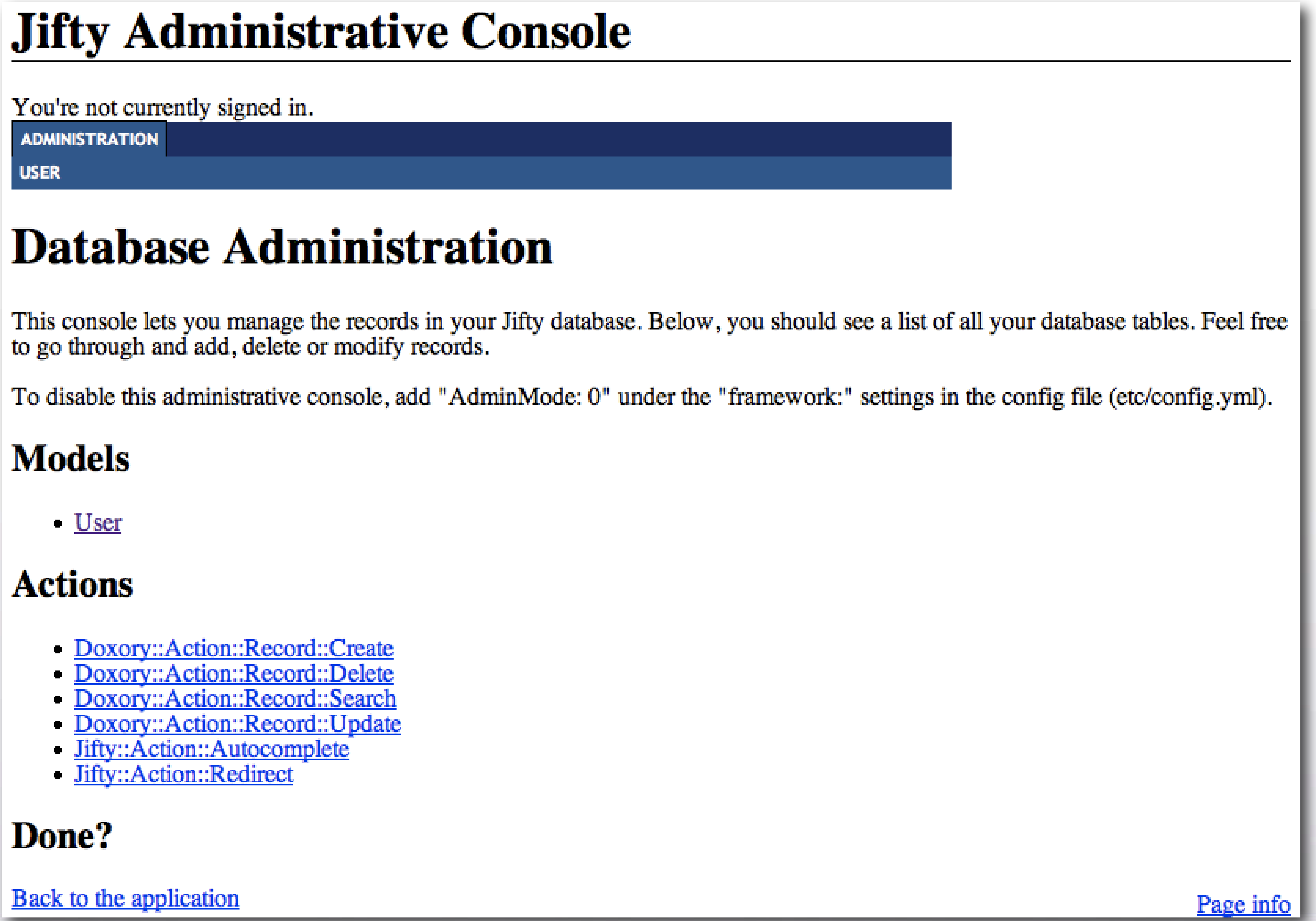
Welcome to your new Jifty application

You're not currently signed in.



Welcome to your new Jifty application

You're not currently signed in.



Jifty Administrative Console

You're not currently signed in.

Database Administration

This console lets you manage the records in your Jifty database. Below, you should see a list of all your database tables. Feel free to go through and add, delete or modify records.

To disable this administrative console, add "AdminMode: 0" under the "framework:" settings in the config file (etc/config.yml).

Models

- User

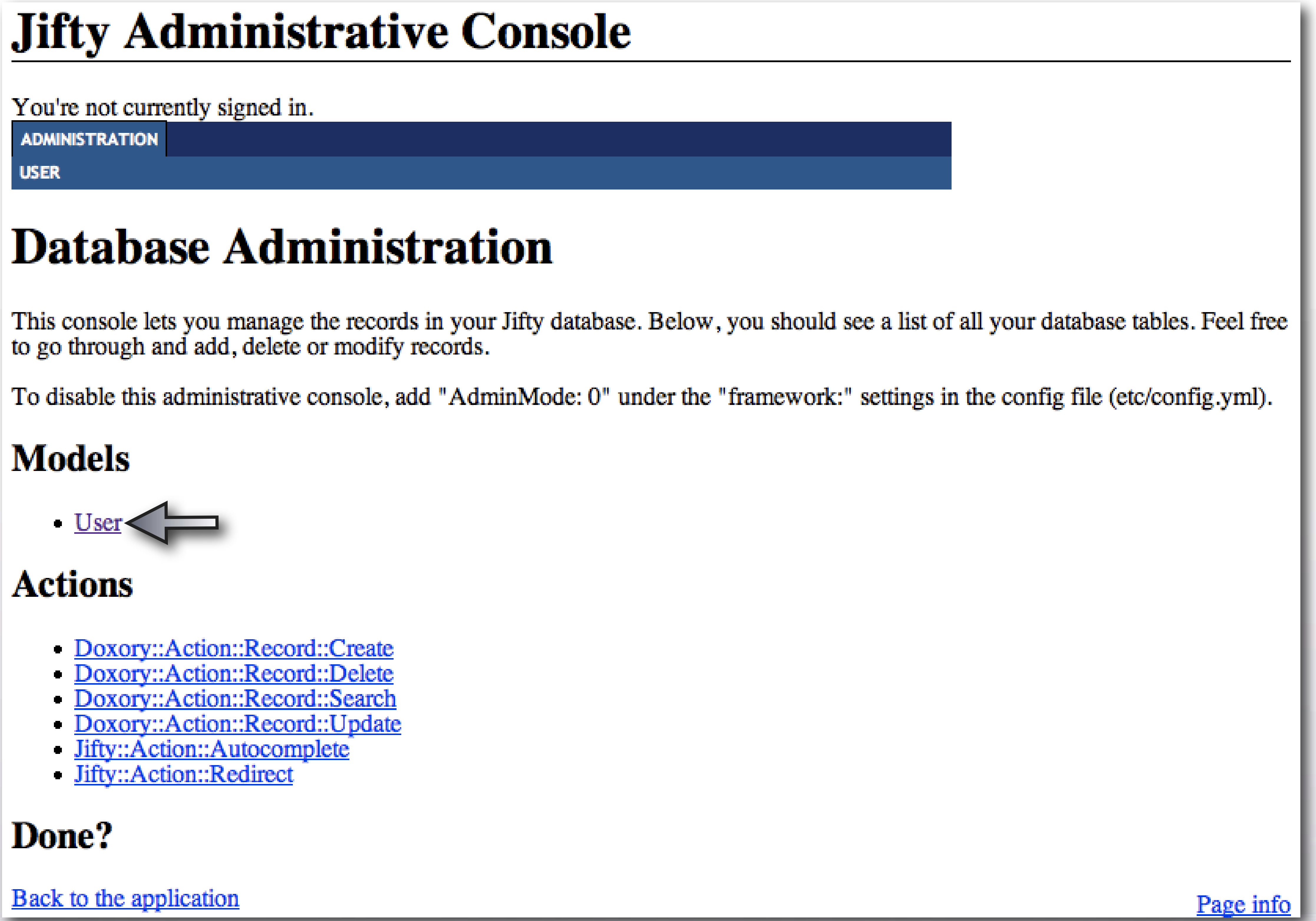
Actions

- Doxory::Action::Record::Create
- Doxory::Action::Record::Delete
- Doxory::Action::Record::Search
- Doxory::Action::Record::Update
- Jifty::Action::Autocomplete
- Jifty::Action::Redirect

Done?

Back to the application

Page info



Jifty Administrative Console

You're not currently signed in.

Database Administration

This console lets you manage the records in your Jifty database. Below, you should see a list of all your database tables. Feel free to go through and add, delete or modify records.

To disable this administrative console, add "AdminMode: 0" under the "framework:" settings in the config file (etc/config.yml).

Models

- User

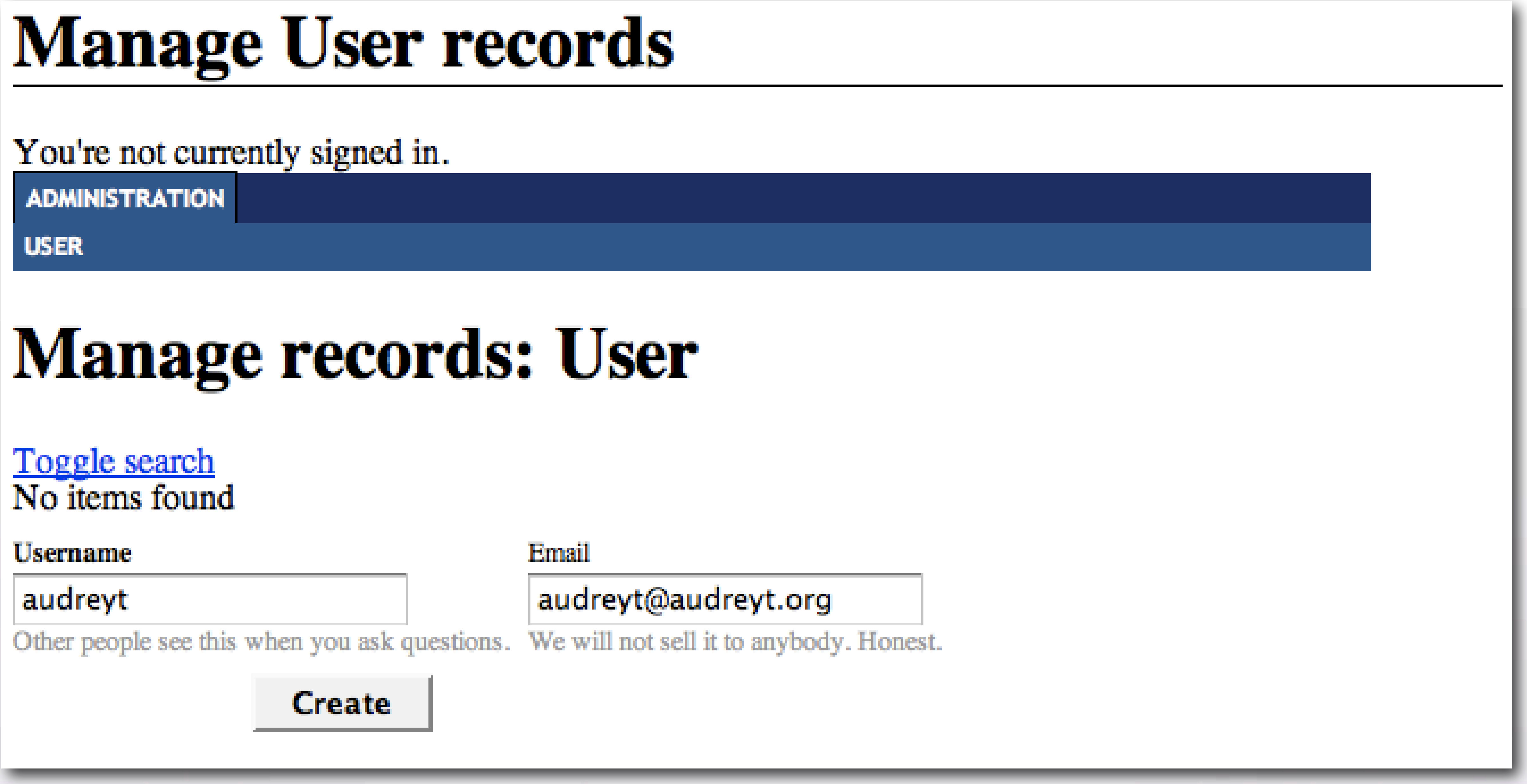
Actions

- Doxory::Action::Record::Create
- Doxory::Action::Record::Delete
- Doxory::Action::Record::Search
- Doxory::Action::Record::Update
- Jifty::Action::Autocomplete
- Jifty::Action::Redirect

Done?

Back to the application

Page info



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

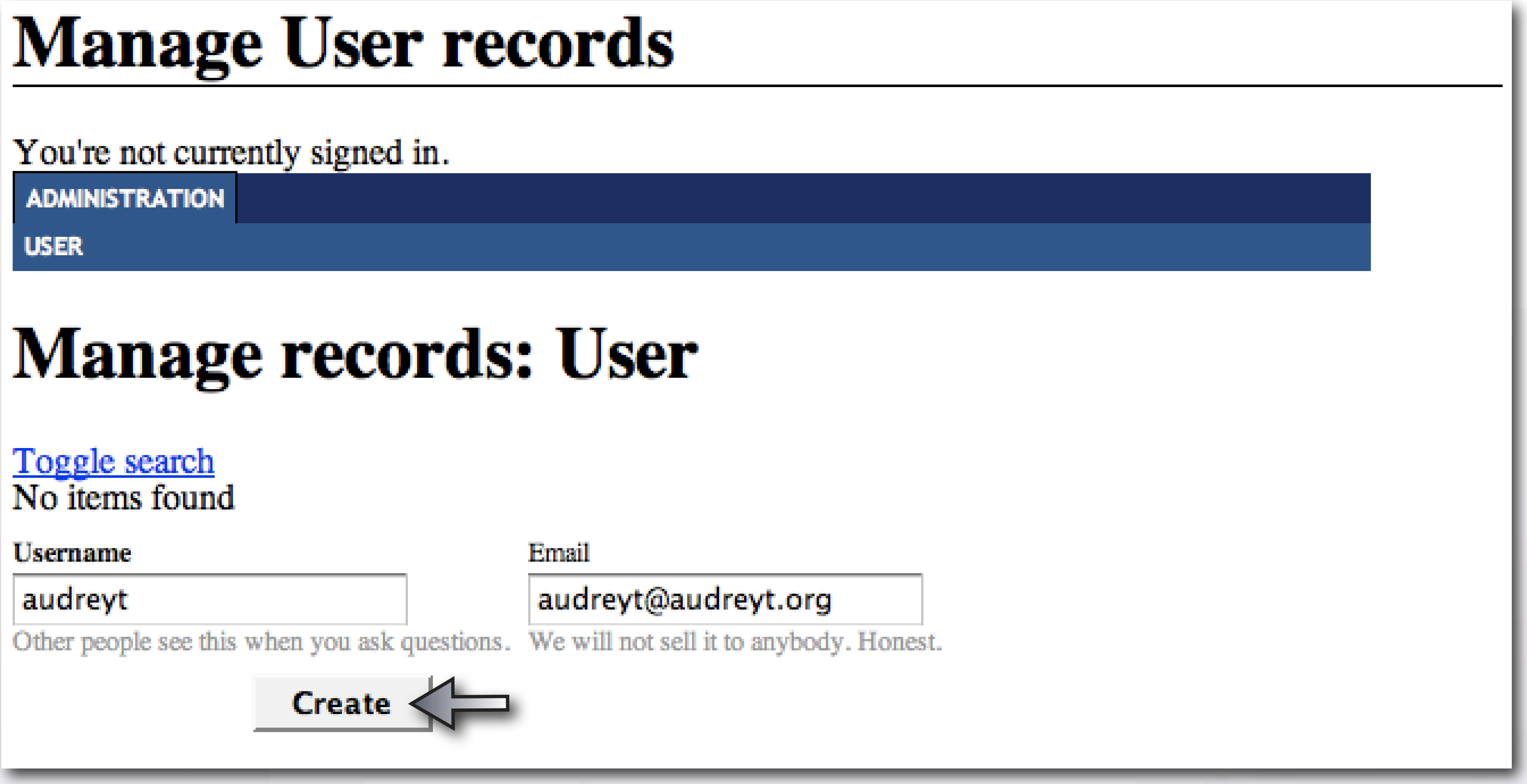
No items found

Username: audreyt

Other people see this when you ask questions.

Email: audreyt@audreyt.org

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

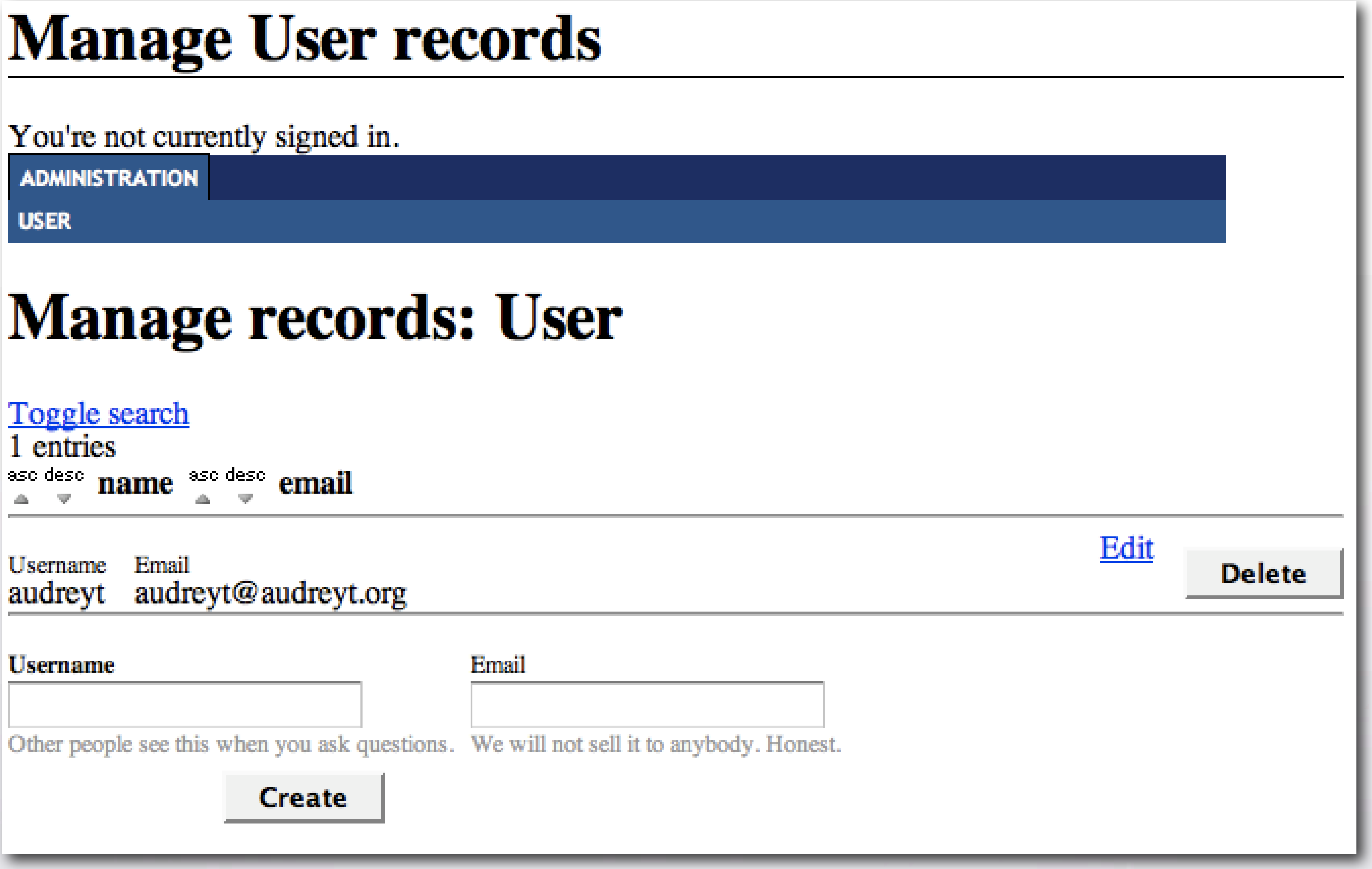
No items found

Username: audreyt

Other people see this when you ask questions.

Email: audreyt@audreyt.org

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

1 entries

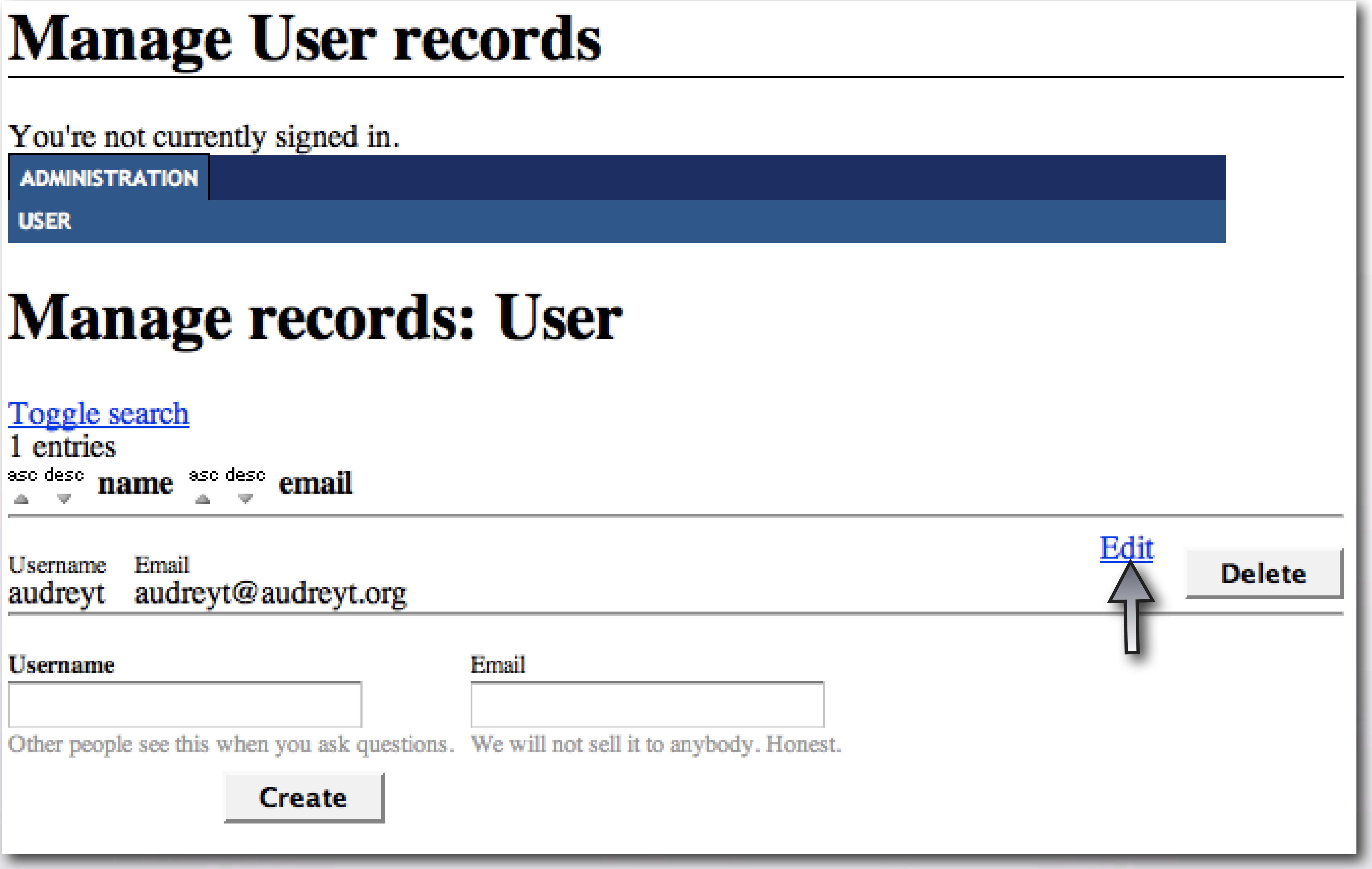
Username	Email
audreyt	audreyt@audreyt.org

Username

Other people see this when you ask questions.

Email

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

1 entries

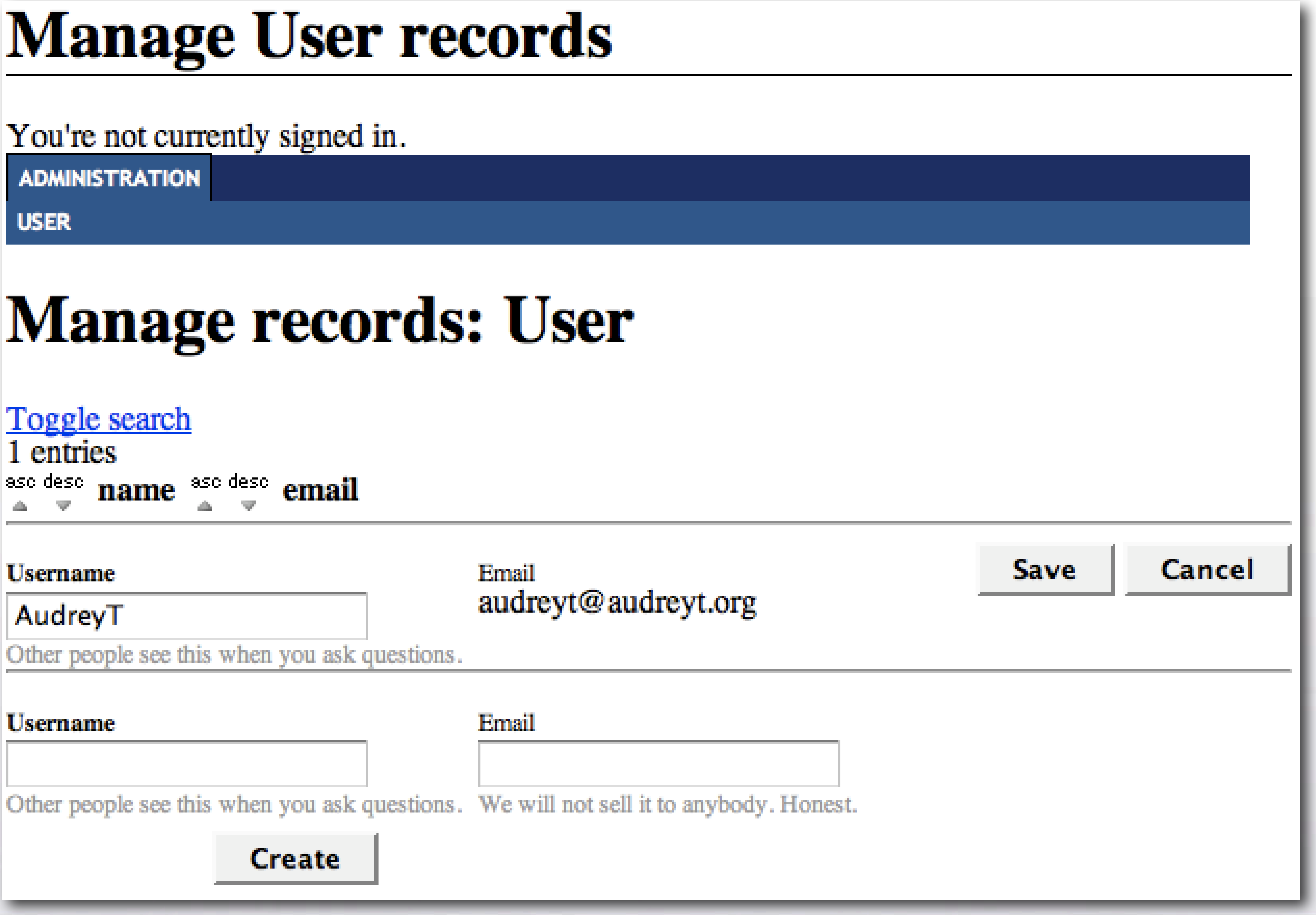
Username	Email
audreyt	audreyt@audreyt.org

Username

Other people see this when you ask questions.

Email

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

1 entries

Username: AudreyT

Other people see this when you ask questions.

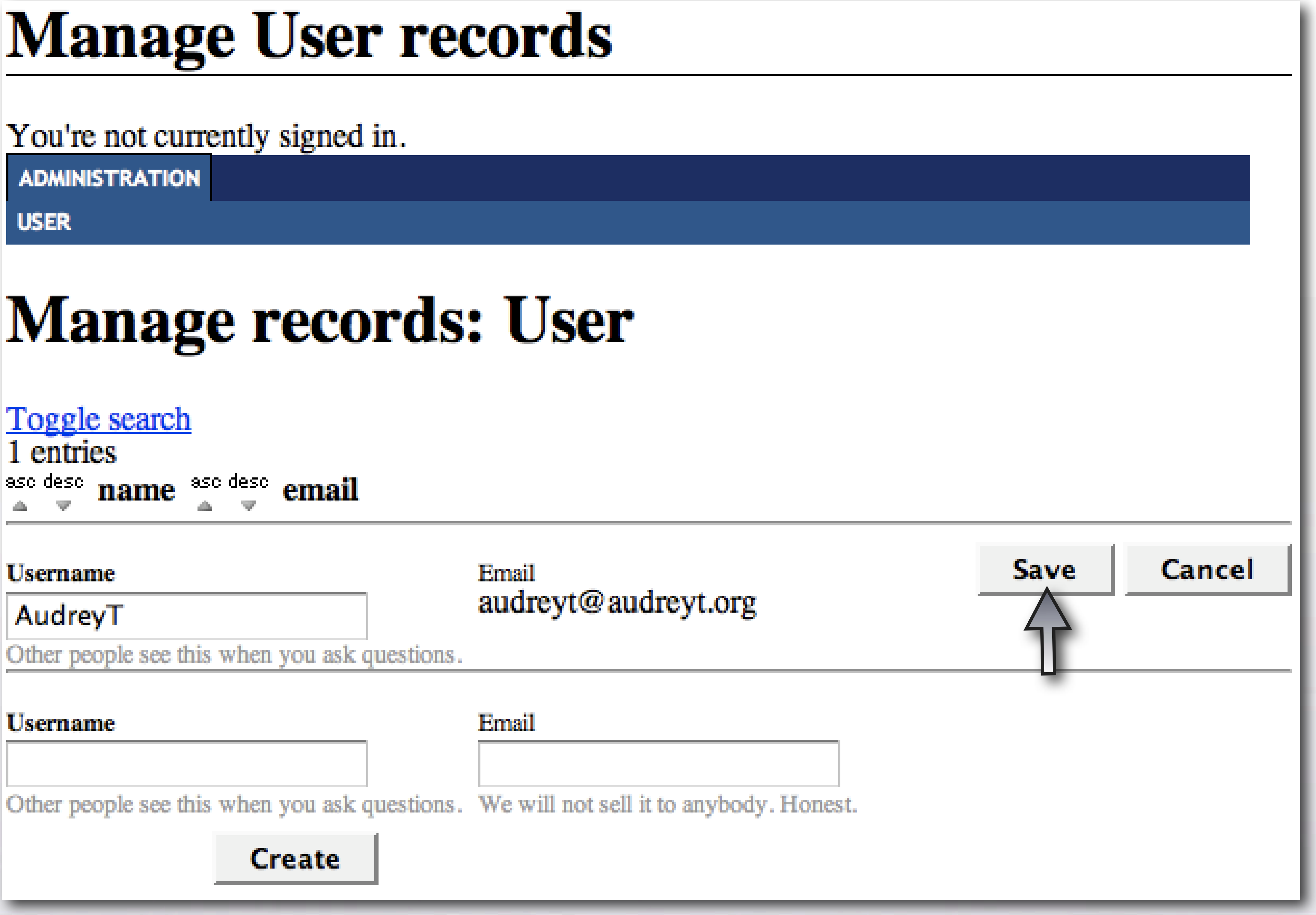
Email: audreyt@audreyt.org

Username:

Other people see this when you ask questions.

Email:

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

1 entries

Username

AudreyT

Other people see this when you ask questions.

Email

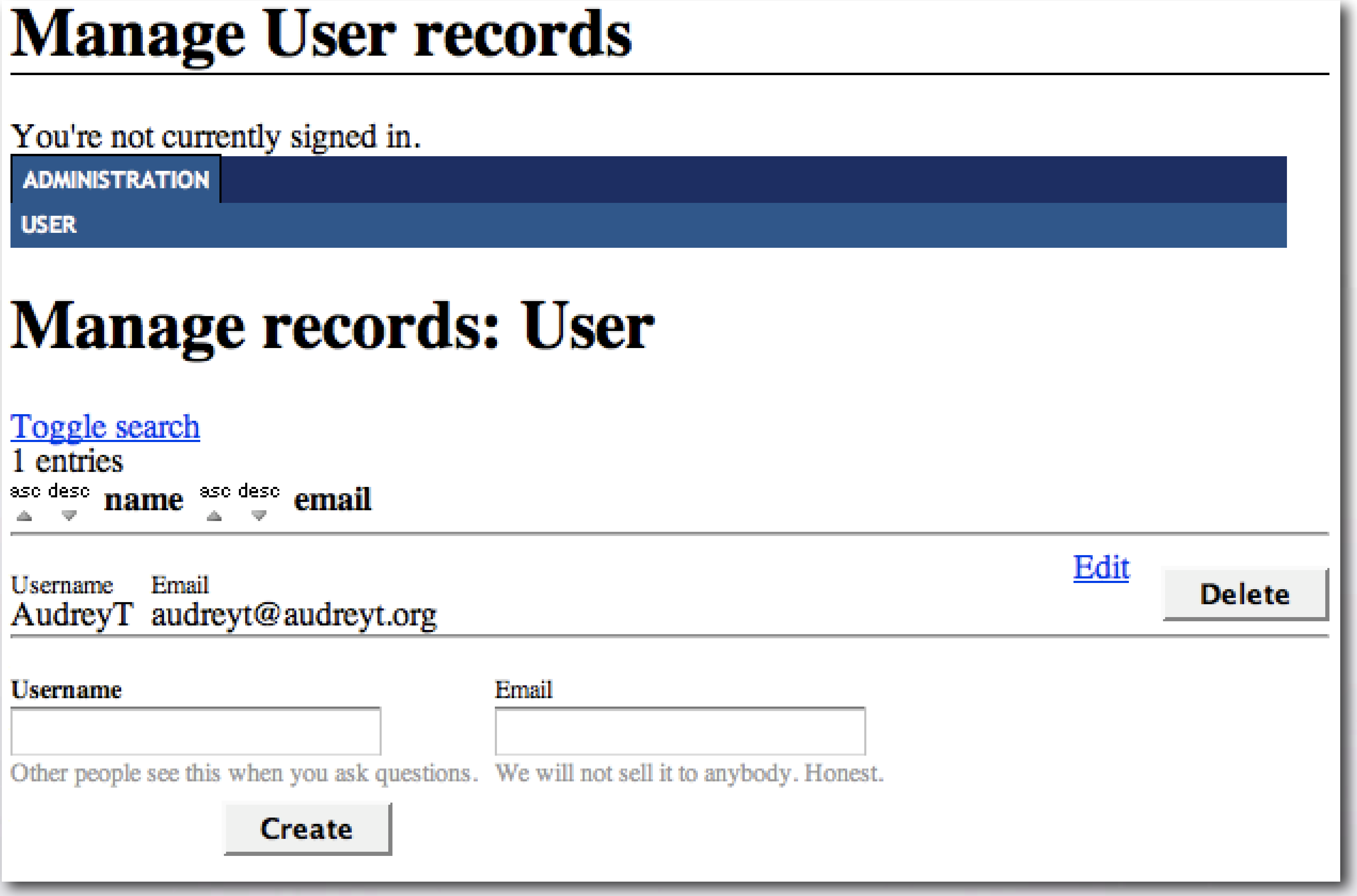
audreyt@audreyt.org

Username

Other people see this when you ask questions.

Email

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

Toggle search

1 entries

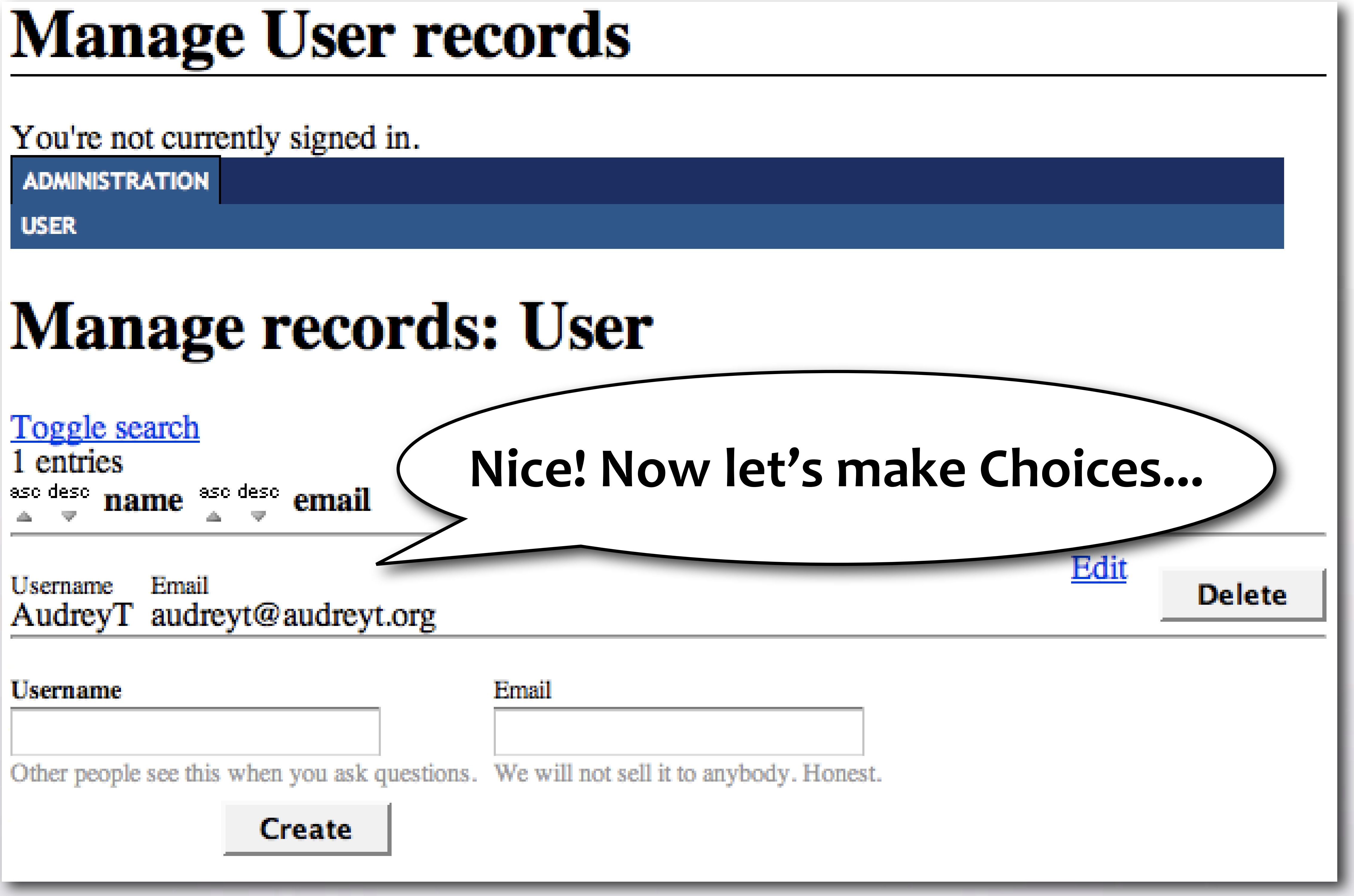
Username	Email
AudreyT	audreyt@audreyt.org

Username

Other people see this when you ask questions.

Email

We will not sell it to anybody. Honest.



Manage User records

You're not currently signed in.

Manage records: User

1 entries

Username	Email
AudreyT	audreyt@audreyt.org

Username

Other people see this when you ask questions.

Email

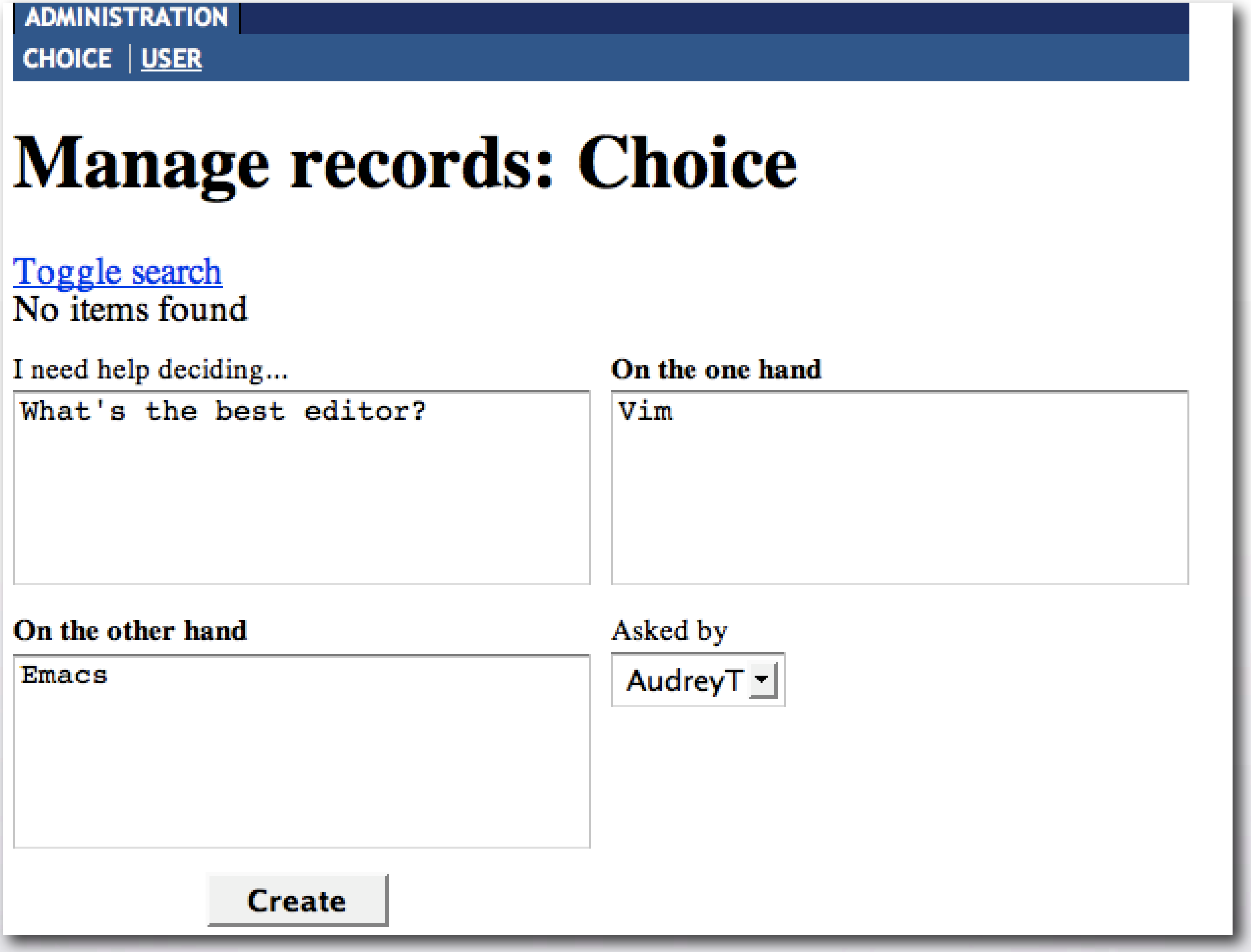
We will not sell it to anybody. Honest.

Nice! Now let’s make Choices...

```
$ jifty model --name Choice
$ jifty model --name Choice
Writing file t/00-model-Choice.t
Writing file lib/Doxory/Model/Choice.pm
```

```
$ jifty model --name Choice
Writing file t/00-model-Choice.t
Writing file lib/Doxory/Model/Choice.pm

$ vim lib/Doxory/Model/Choice.pm
```

```
package Doxory::Model::Choice;
use Jifty::DBI::Schema;
use Doxory::Record schema {

column name =>
    label is 'I need help deciding...',
    render as 'textarea';

column a =>
    label is 'On the one hand',
    render as 'textarea', is mandatory;

column b =>
    label is 'On the other hand',
    render as 'textarea', is mandatory;

column asked_by =>
    label is 'Asked by',
    default is defer { Jifty->web->current_user->id },
    references Doxory::Model::User;

};
```



Manage records: Choice

Toggle search
No items found

I need help deciding...

What's the best editor?

On the one hand

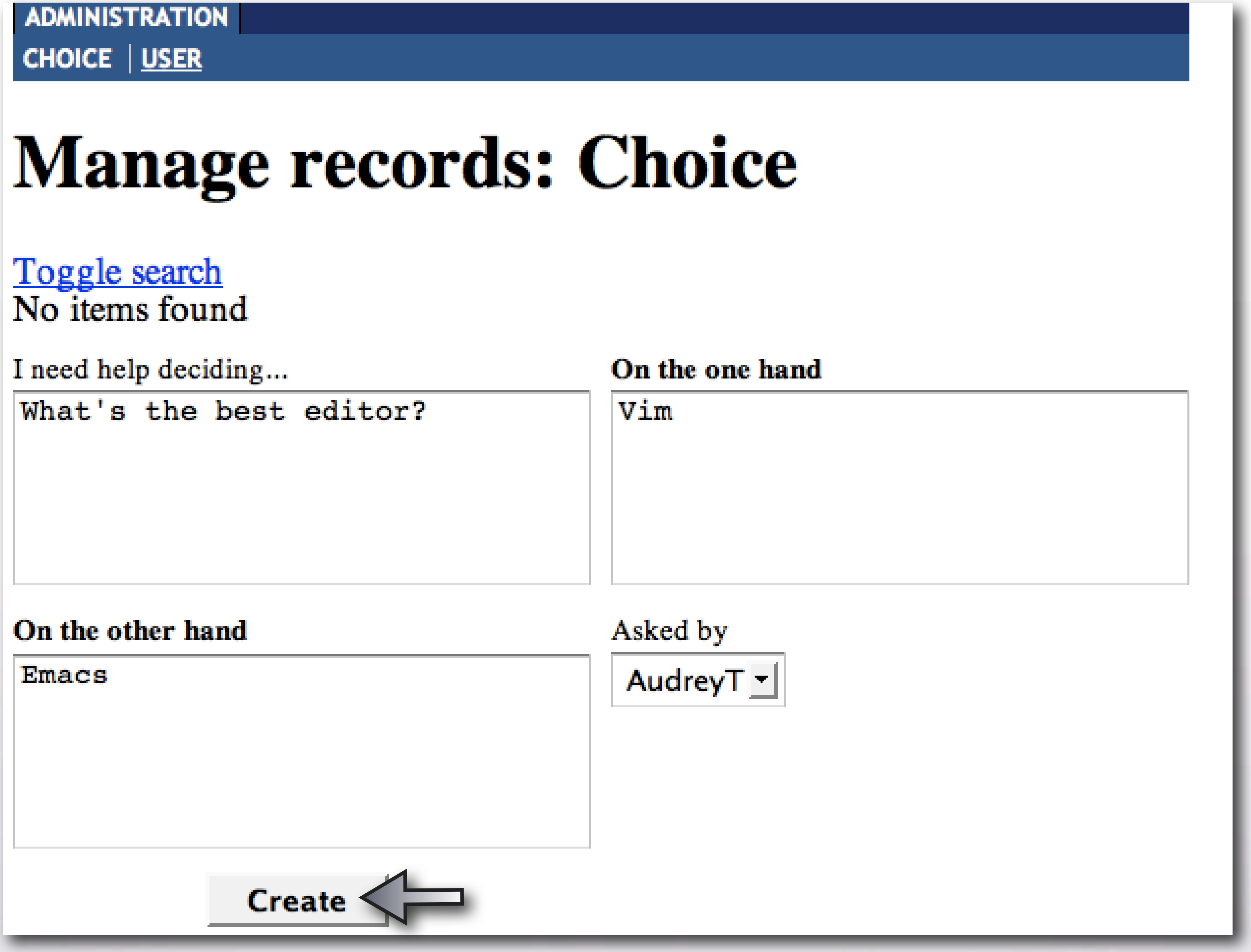
Vim

On the other hand

Emacs

Asked by

AudreyT



Manage records: Choice

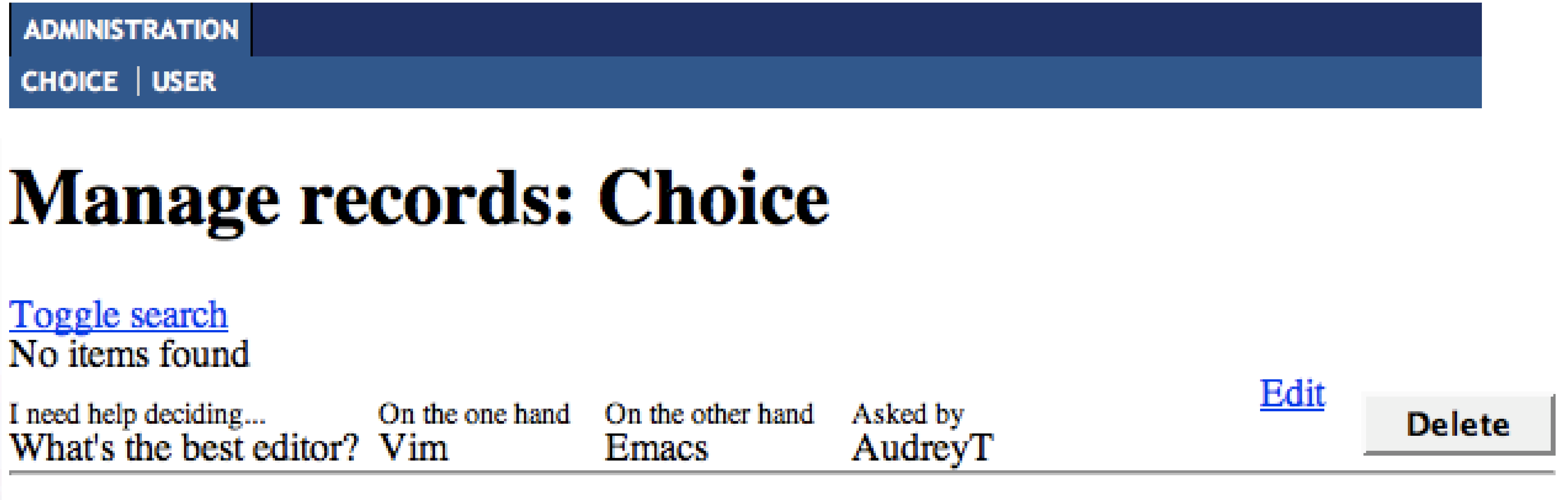
No items found

I need help deciding...: What's the best editor?

On the one hand: Vim

On the other hand: Emacs

Asked by: AudreyT



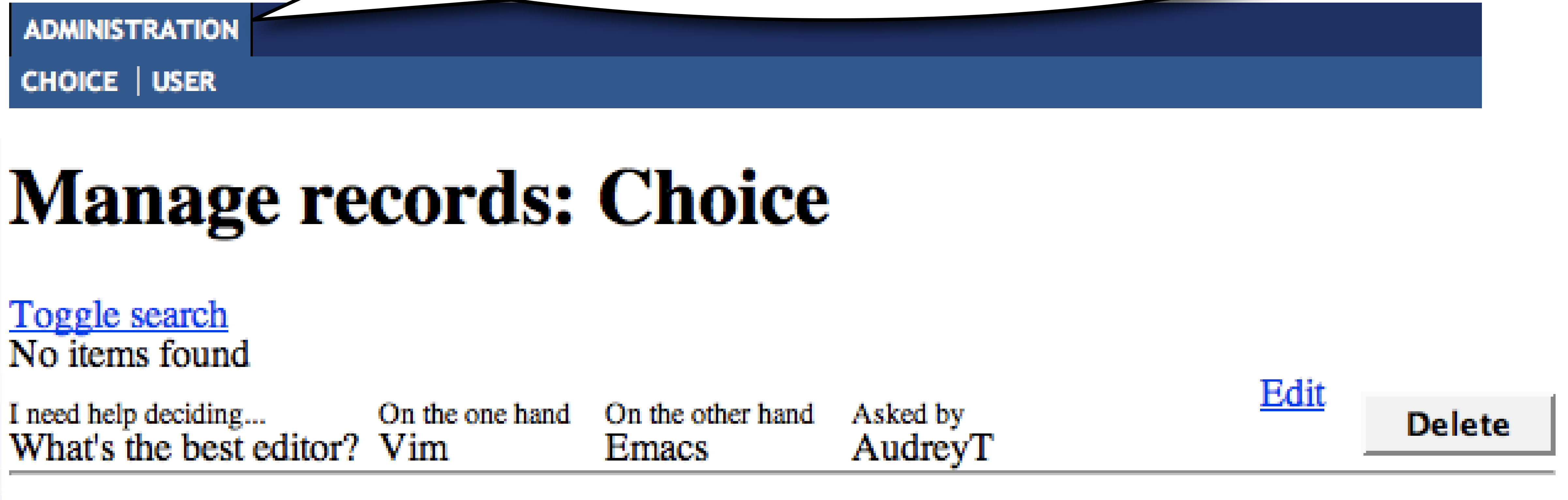
Manage records: Choice

Toggle search

No items found

I need help deciding...	On the one hand	On the other hand	Asked by
What's the best editor?	Vim	Emacs	AudreyT

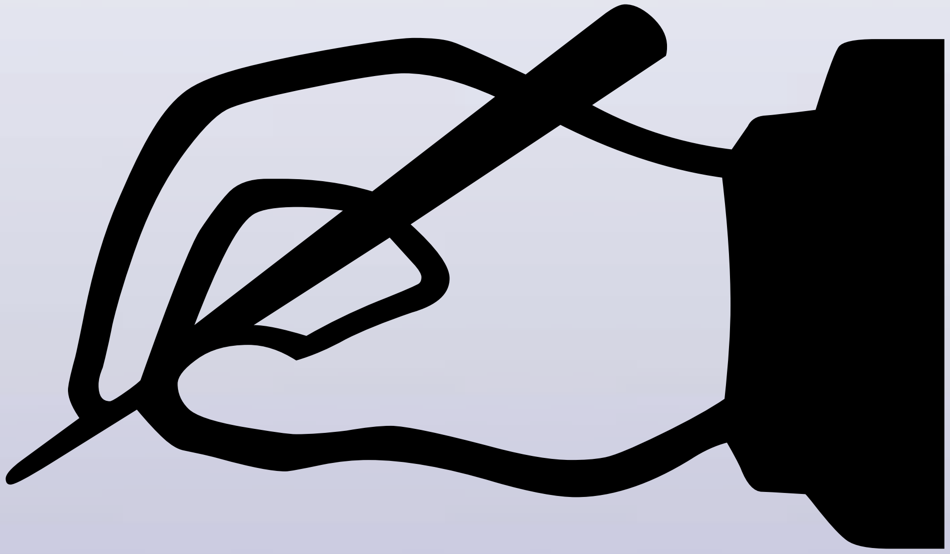
Done!
(But how do users actually sign in?)



Manage records: Choice

Toggle search
No items found

I need help deciding...	On the one hand	On the other hand	Asked by		
What's the best editor?	Vim	Emacs	AudreyT	Edit	Delete



```
$ vim etc/config.yml
$ vim etc/config.yml
```

We need to add two lines here.

```yaml
---
framework:
  Plugins:
    - SkeletonApp: {}
    - REST: {}
    - Halo: {}
    - ErrorTemplates: {}
    - CompressedCSSandJS: {}
```

```
---
framework:
  Plugins:
    - SkeletonApp: {}
    - REST: {}
    - Halo: {}
    - ErrorTemplates: {}
    - CompressedCSSandJS: {}
    - Authentication::Password: {}
  SkipAccessControl: 1
```

```
---
framework:
  Plugins:
    - SkeletonApp: {}
    - REST: {}
    - Halo: {}
    - ErrorTemplates: {}
    - CompressedCSSandJS: {}
    - Authentication::Password: {}
  SkipAccessControl: 1
```

Too lazy to define ACLs yet.

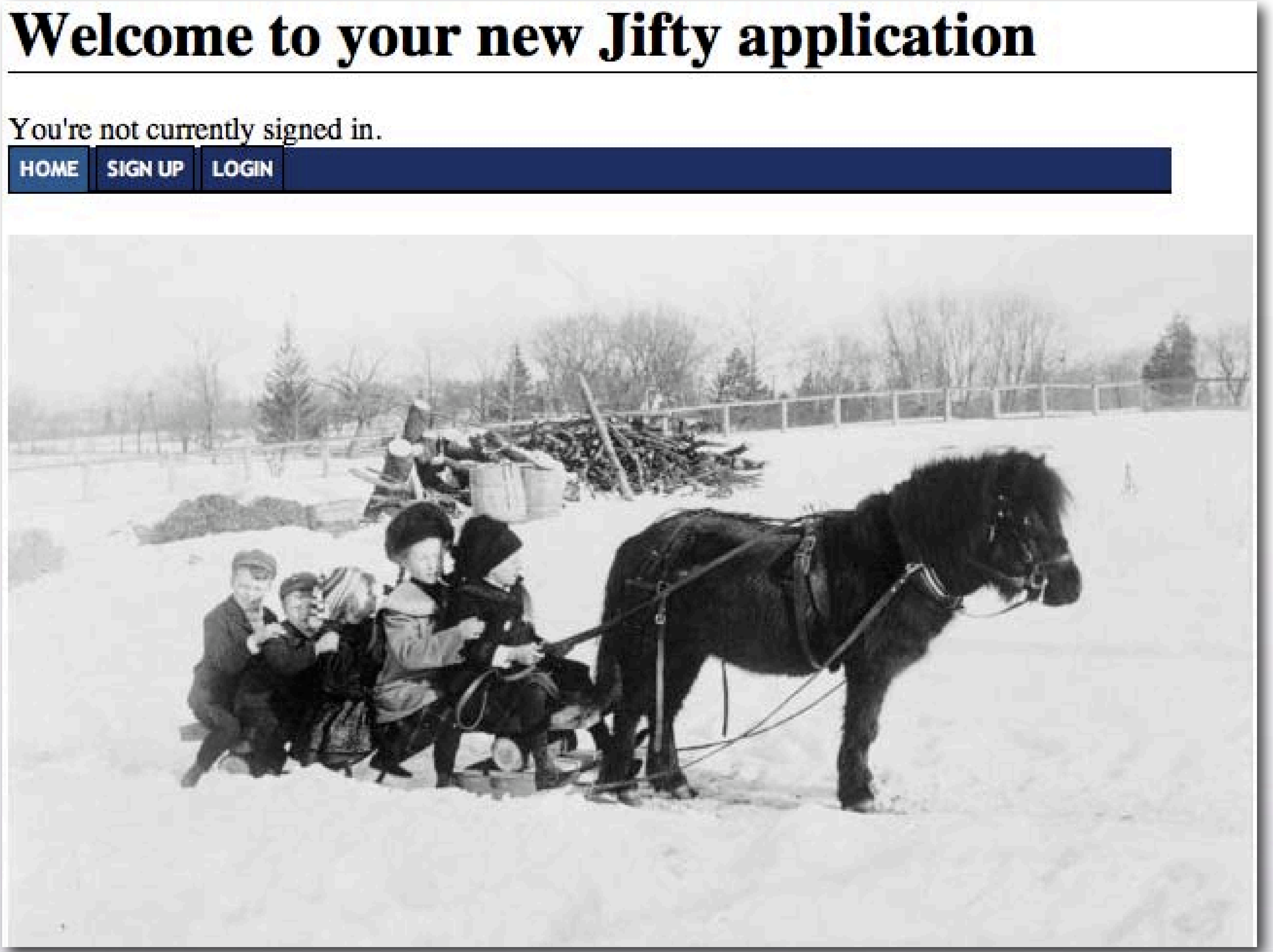
```
$ vim lib/Doxory/Model/User.pm
$ vim lib/Doxory/Model/User.pm
```

Then we add two mixins here.

```
package Doxory::Model::User;
use Jifty::DBI::Schema;
use Doxory::Record schema {

column name =>
    label is 'Username',
    hints is 'Other people see this when you ask questions.',
    is distinct,
    is mandatory;

column email =>
    label is 'Email',
    hints is 'We will not sell it to anybody. Honest.',
    default is '',
    is distinct,
    is immutable;

};
```

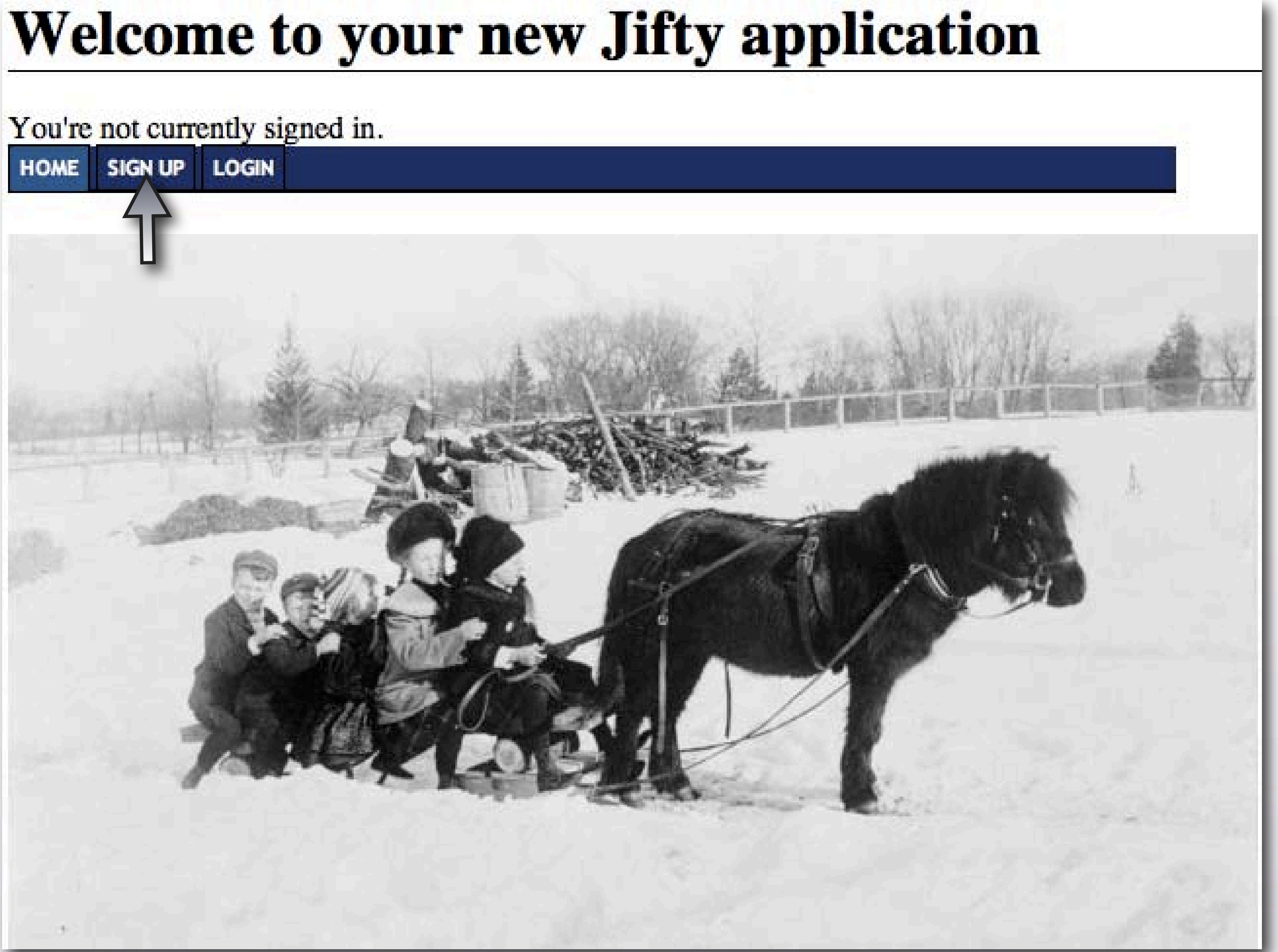
```
package Doxory::Model::User;
use Jifty::DBI::Schema;
use Doxory::Record schema {

column name =>
    label is 'Username',
    hints is 'Other people see this when you ask questions.',
    is distinct,
    is mandatory;

column email =>
    label is 'Email',
    hints is 'We will not sell it to anybody. Honest.',
    default is '',
    is distinct,
    is immutable;

};
```

```
use Jifty::Plugin::User::Mixin::Model::User;
use Jifty::Plugin::Authentication::Password::Mixin::Model::User;
```



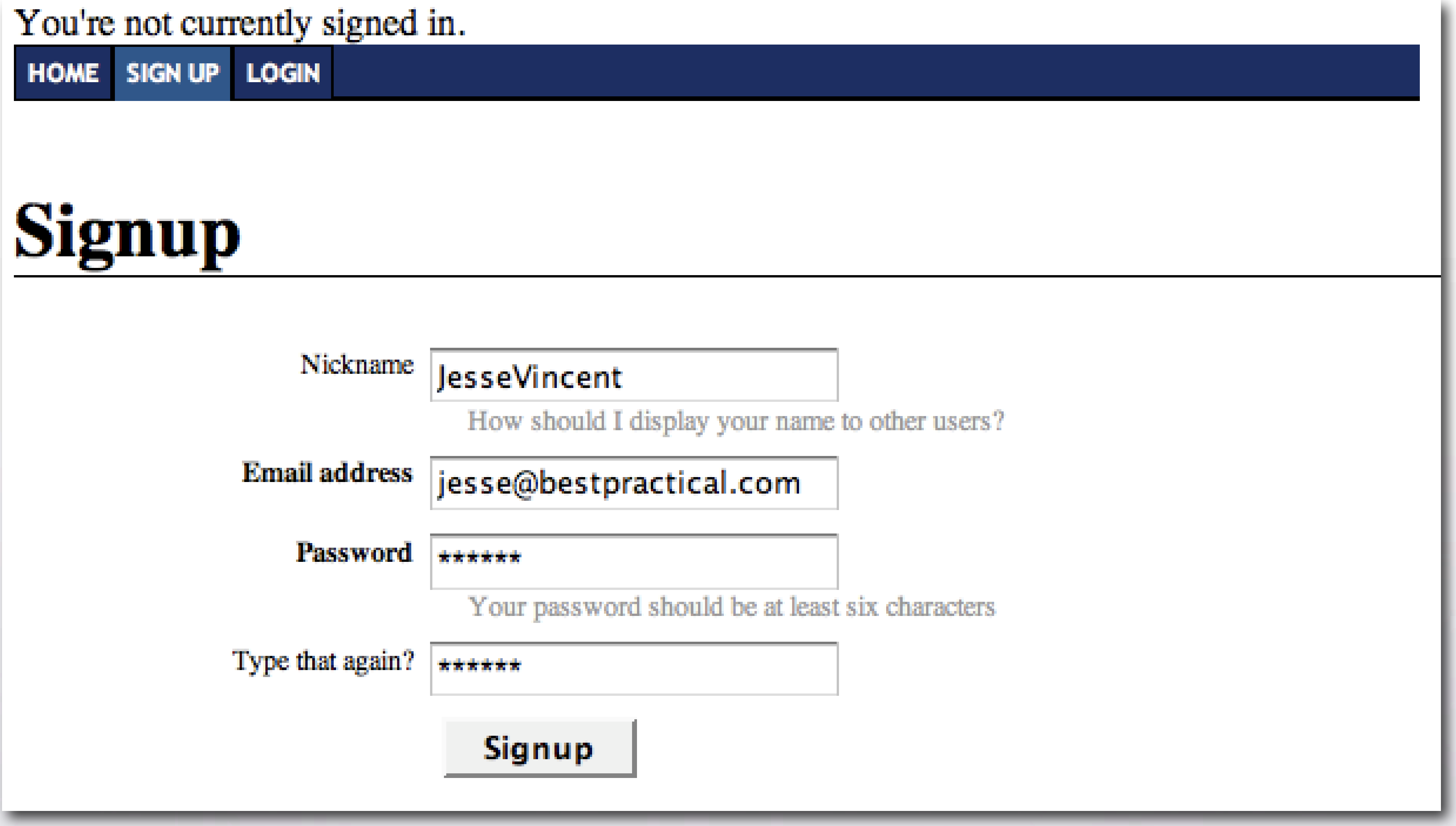
Welcome to your new Jifty application

You're not currently signed in.



Welcome to your new Jifty application

You're not currently signed in.



You're not currently signed in.

Signup

Nickname: JesseVincent

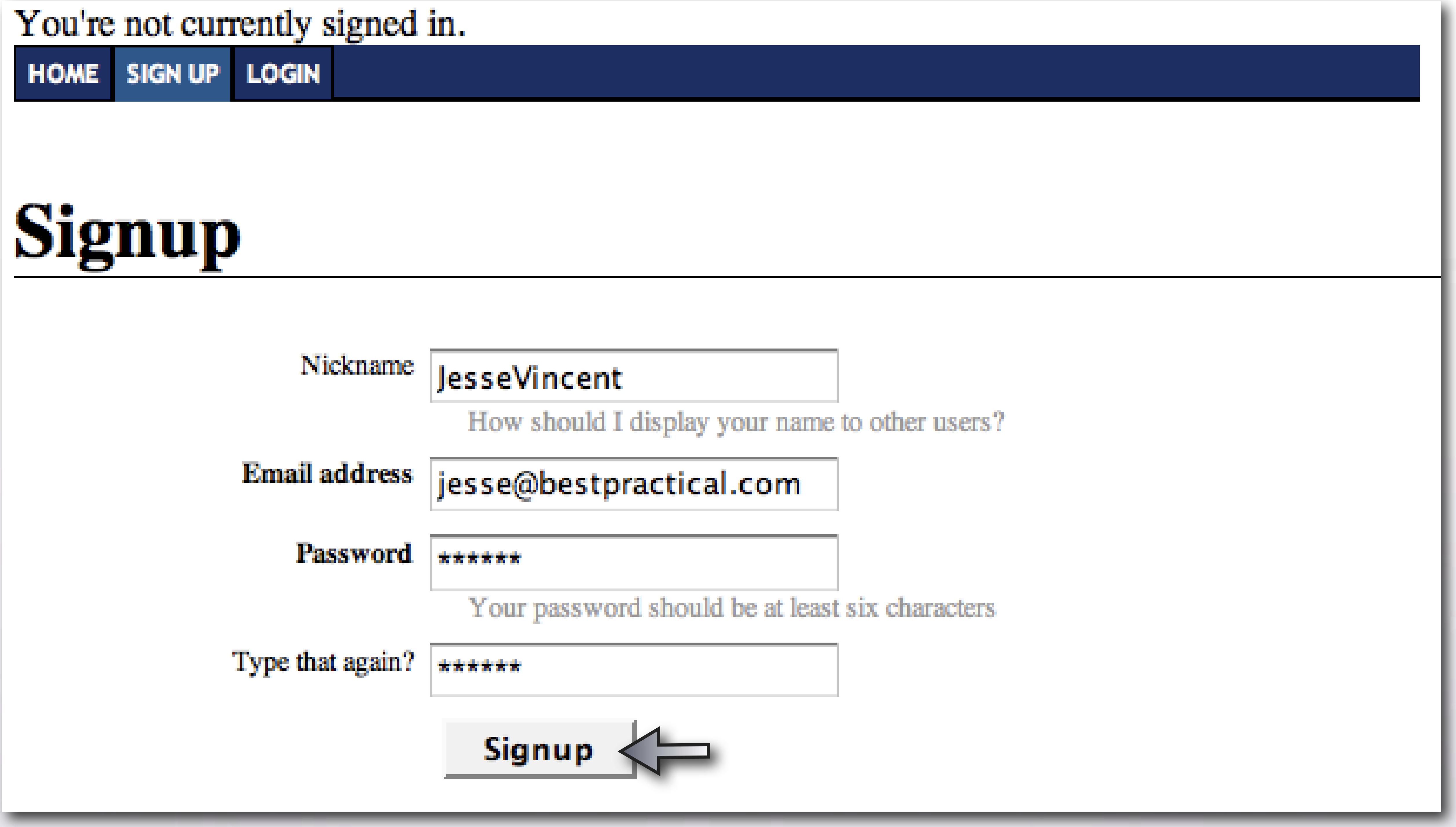
How should I display your name to other users?

Email address: jesse@bestpractical.com

Password: ******

Your password should be at least six characters

Type that again?: ******



You're not currently signed in.

Signup

Nickname: JesseVincent

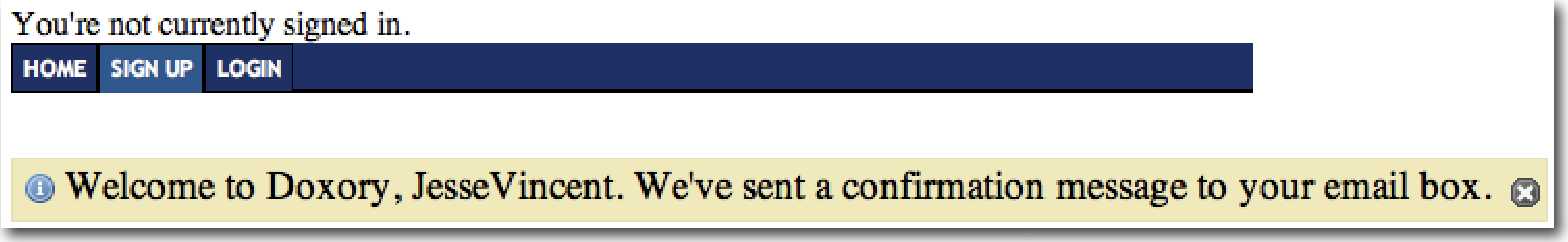
How should I display your name to other users?

Email address: jesse@bestpractical.com

Password: ******

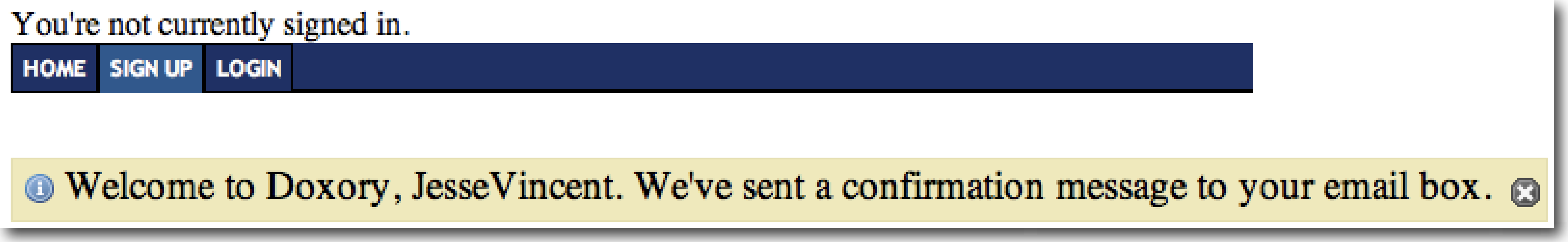
Your password should be at least six characters

Type that again?: ******



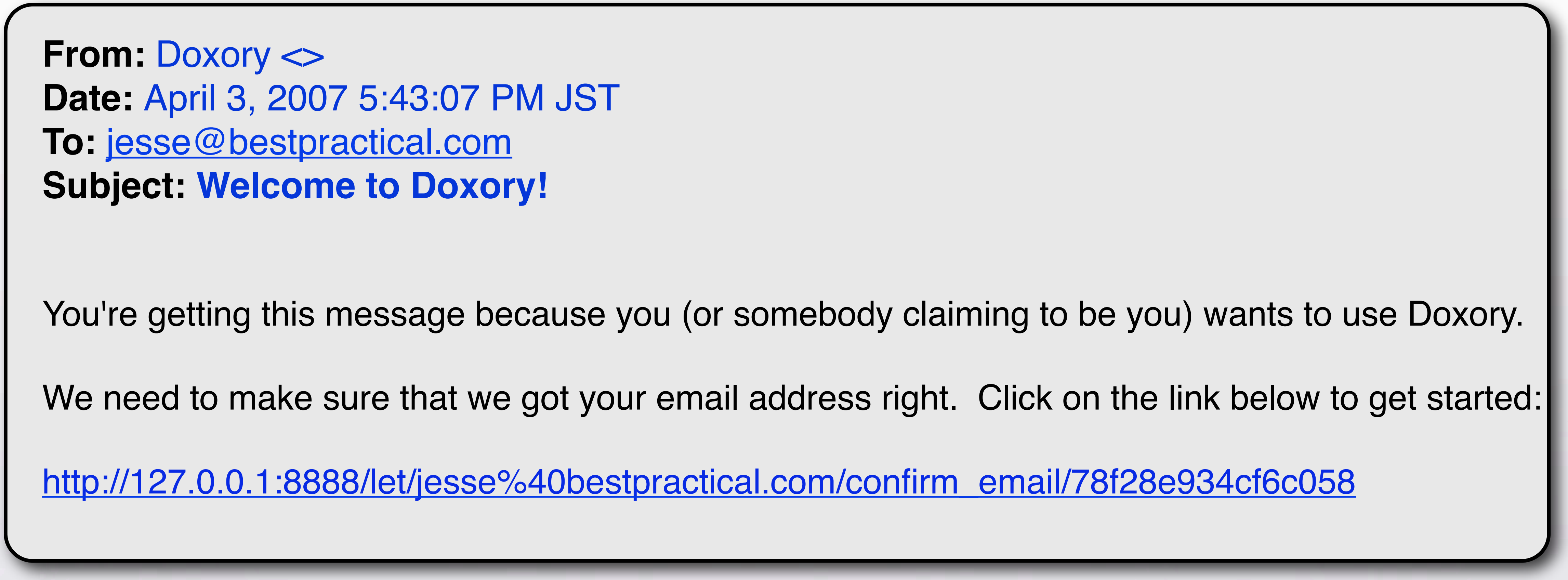
You're not currently signed in.

Welcome to Doxory, JesseVincent. We've sent a confirmation message to your email box.



You're not currently signed in.

Welcome to Doxory, JesseVincent. We've sent a confirmation message to your email box.

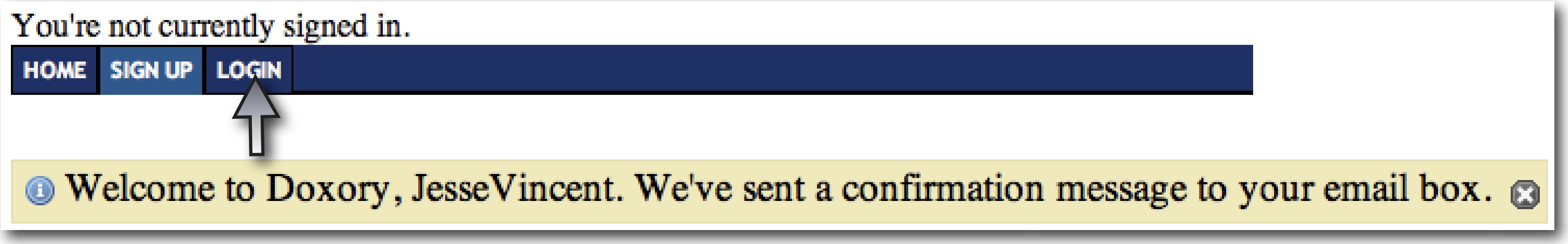


From: Doxory <>
Date: April 3, 2007 5:43:07 PM JST
To: jesse@bestpractical.com
Subject: **Welcome to Doxory!**

You're getting this message because you (or somebody claiming to be you) wants to use Doxory.

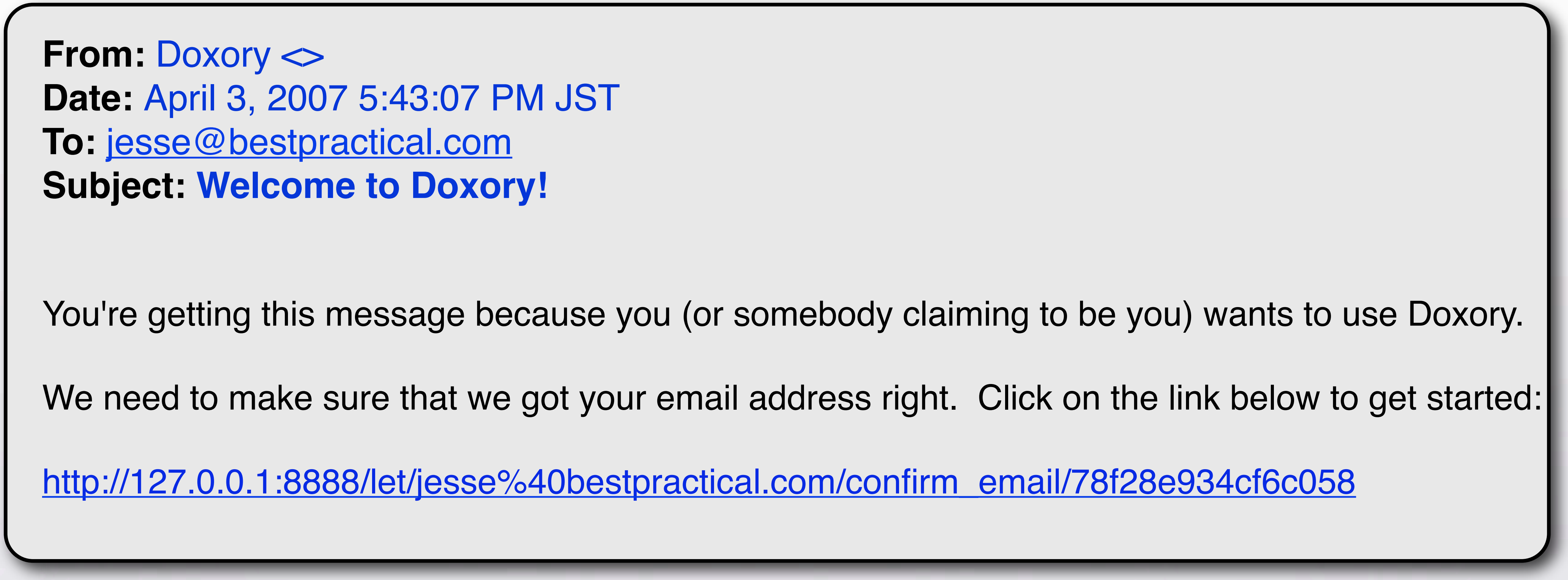
We need to make sure that we got your email address right. Click on the link below to get started:

http://127.0.0.1:8888/let/jesse%40bestpractical.com/confirm_email/78f28e934cf6c058



You're not currently signed in.

Welcome to Doxory, JesseVincent. We've sent a confirmation message to your email box.

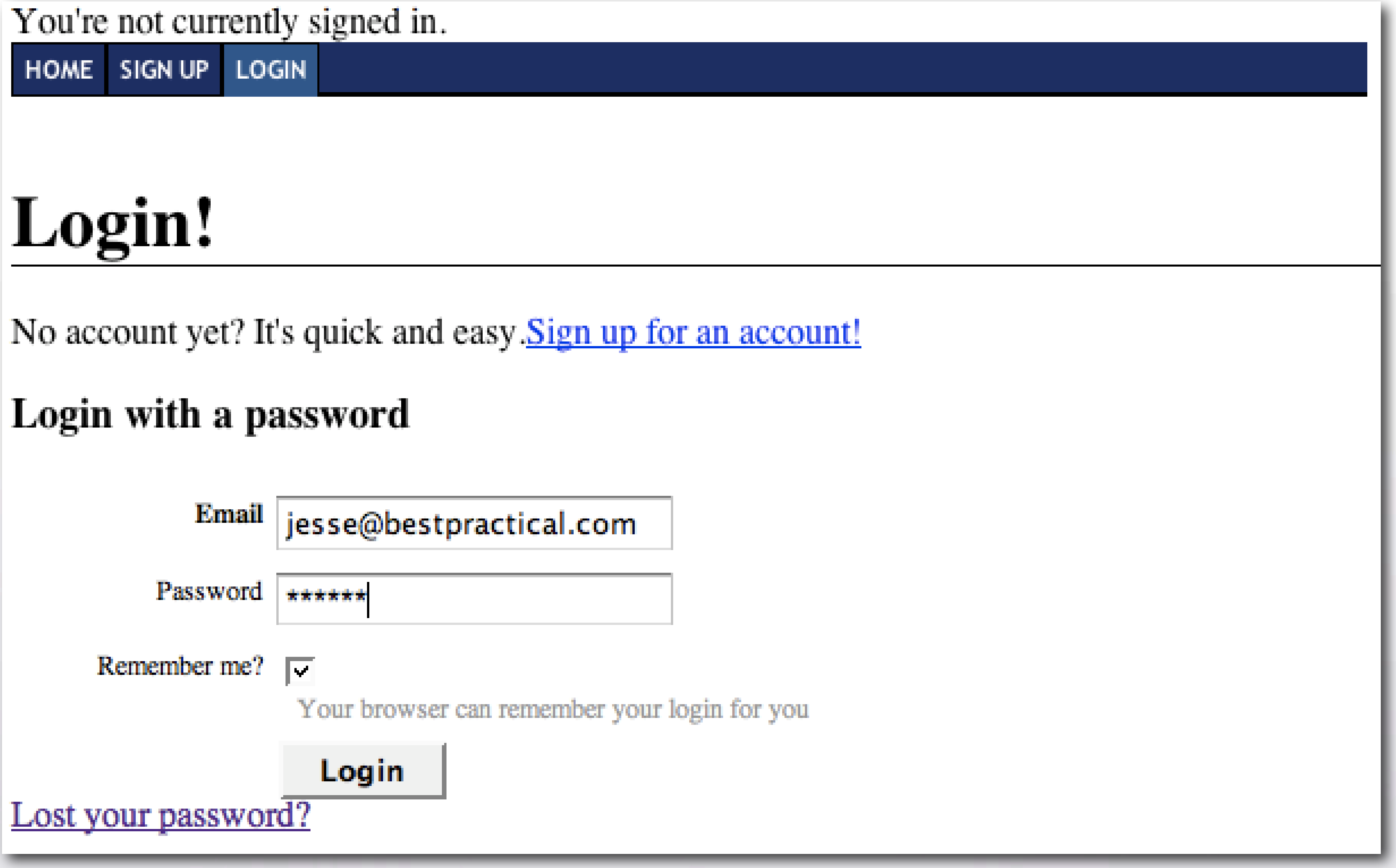


From: Doxory <>
Date: April 3, 2007 5:43:07 PM JST
To: jesse@bestpractical.com
Subject: **Welcome to Doxory!**

You're getting this message because you (or somebody claiming to be you) wants to use Doxory.

We need to make sure that we got your email address right. Click on the link below to get started:

http://127.0.0.1:8888/let/jesse%40bestpractical.com/confirm_email/78f28e934cf6c058



You're not currently signed in.

Login!

No account yet? It's quick and easy.Sign up for an account!

Login with a password

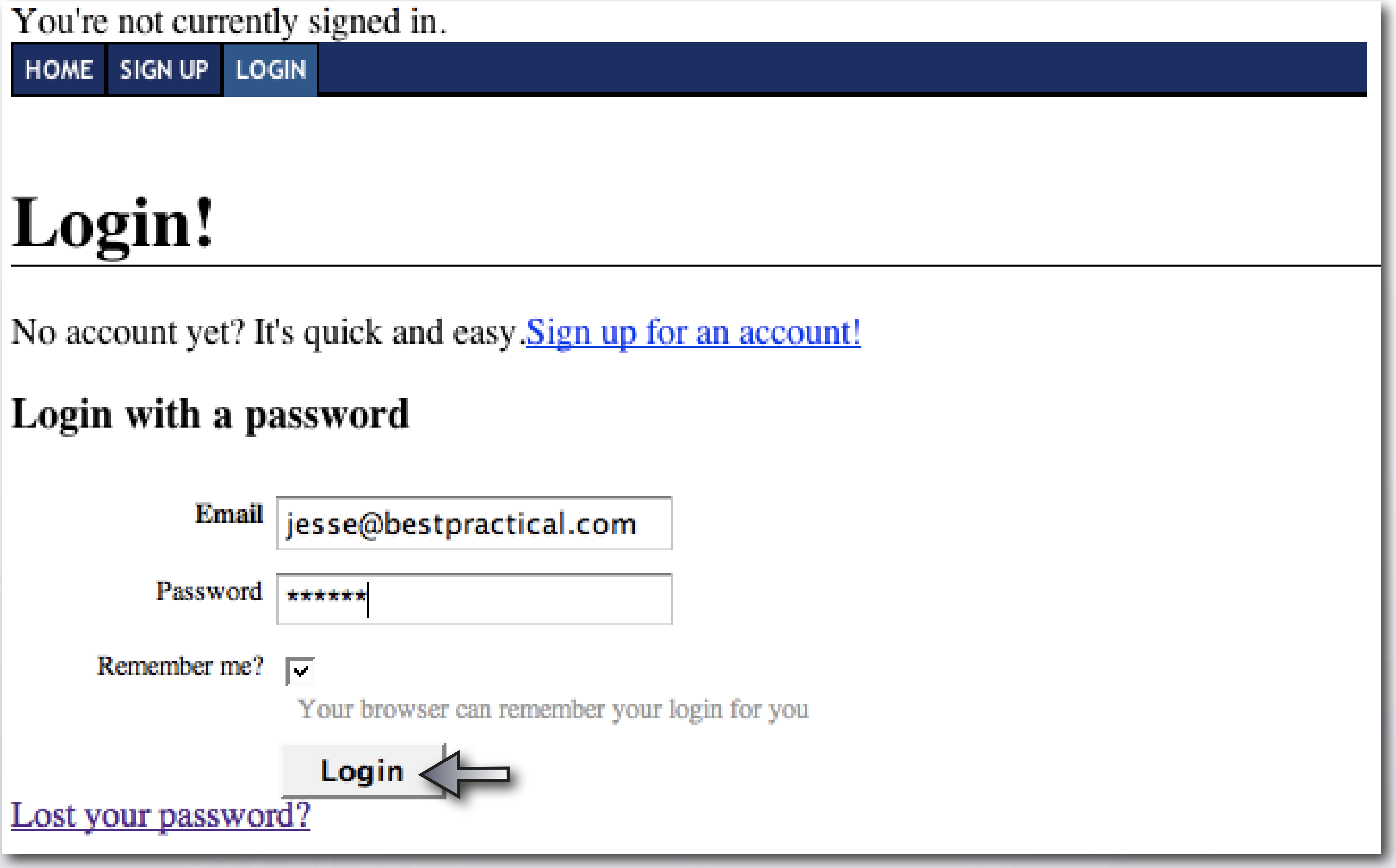
Email: jesse@bestpractical.com

Password: ******

Remember me? ☑

Your browser can remember your login for you

Lost your password?



You're not currently signed in.

Login!

No account yet? It's quick and easy.Sign up for an account!

Login with a password

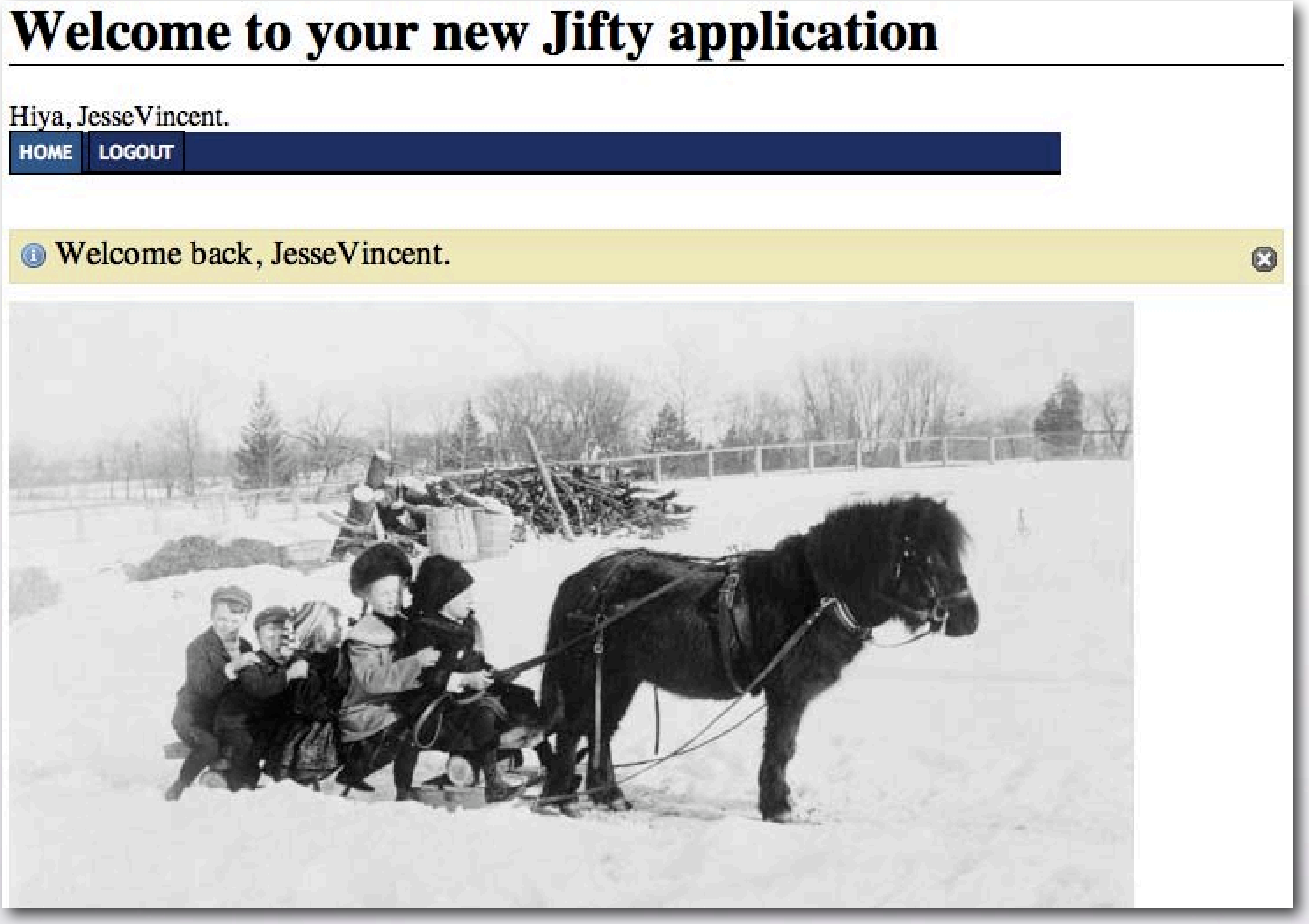
Email jesse@bestpractical.com

Password ******

Remember me? ☑

Your browser can remember your login for you

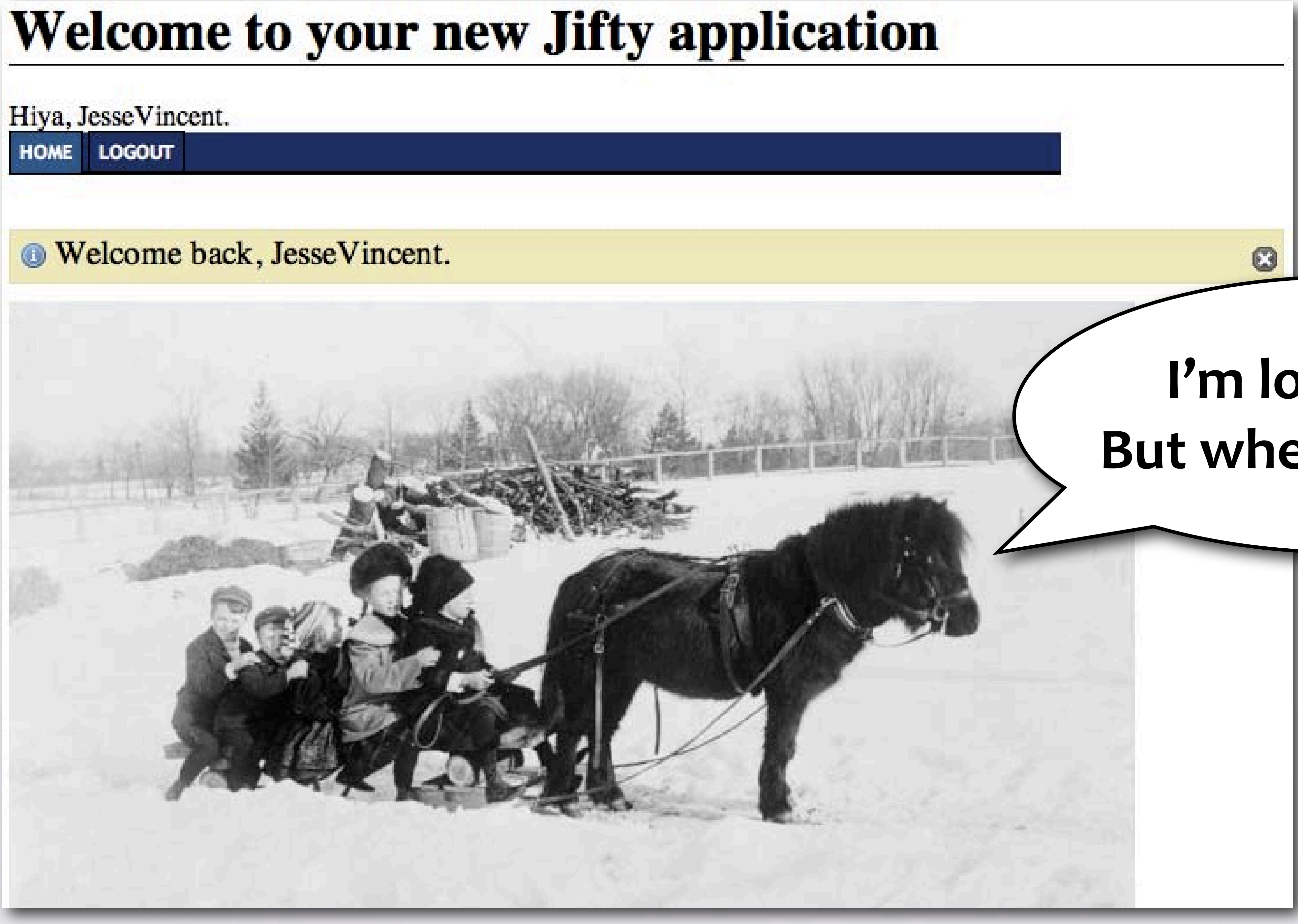
Lost your password?



Welcome to your new Jifty application

Hiya, JesseVincent.

Welcome back, JesseVincent.

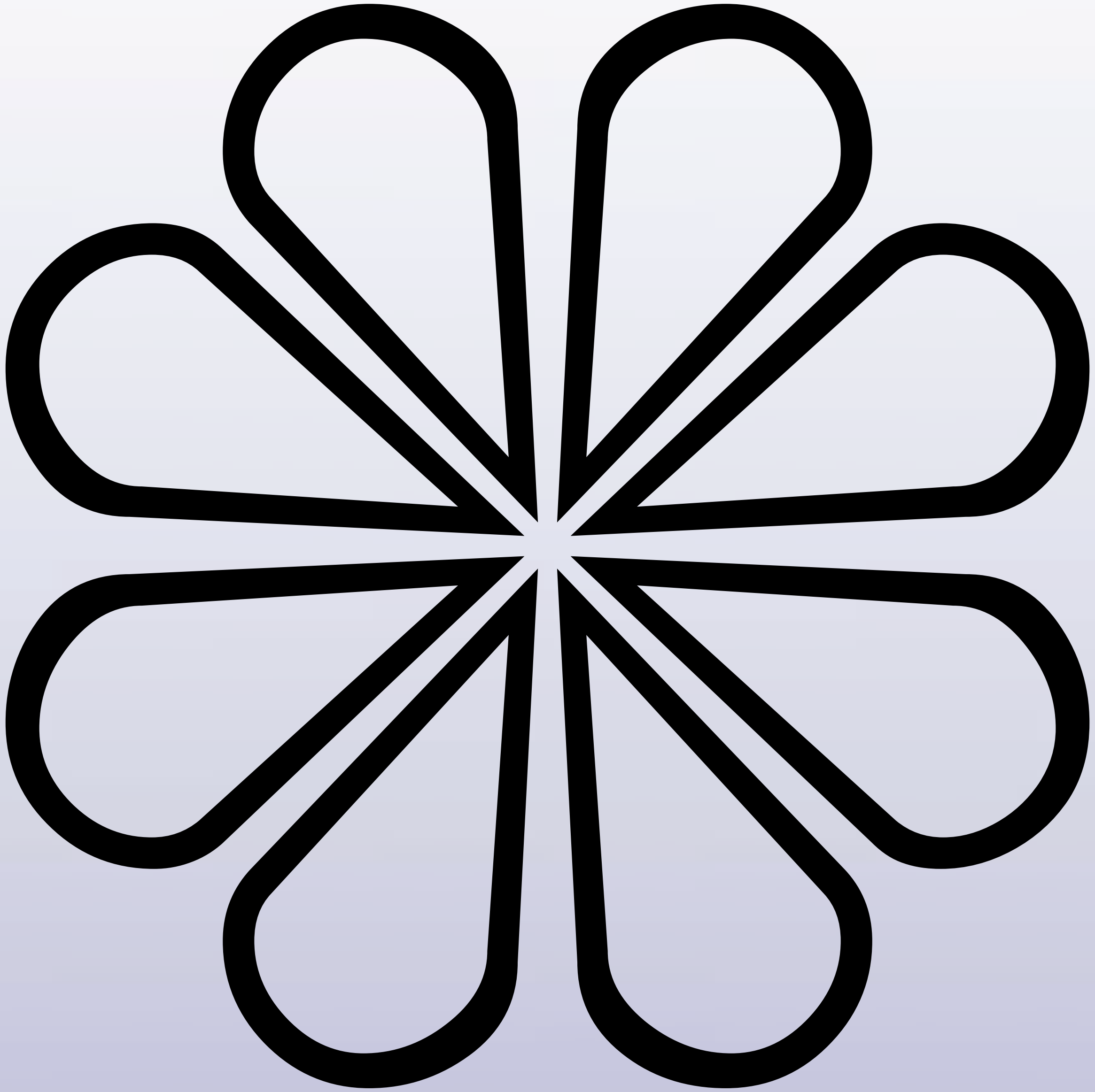


Welcome to your new Jifty application

Hiya, JesseVincent.

Welcome back, JesseVincent.

I’m logged in…
But where is the UI?



```
$ vim lib/Doxory/Dispatcher.pm
$ vim lib/Doxory/Dispatcher.pm
```

Dispatcher maps URLs to Views.

```perl
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
    }
```

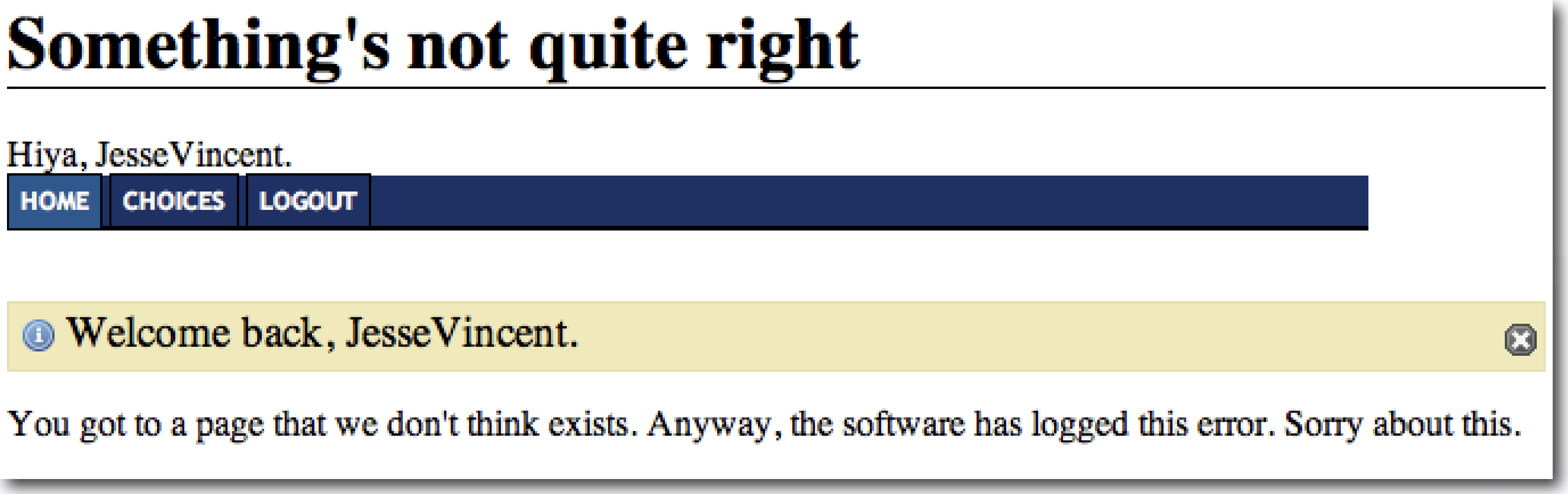
```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
    }
    elsif ($1 !~ /^login|^signup/) {
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
    }
    elsif ($1 !~ /^login|^signup/) {
        tangent 'login';
    }
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
    }
    elsif ($1 !~ /^login|^signup/) {
        tangent 'login';
    }
};
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
    }
    elsif ($1 !~ /^login|^signup/) {
        tangent 'login';
    }
};

# on '/'         => show '/';
```

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;
        $top->child( 'Choices' => url => '/choices' );
    }
    elsif ($1 !~ /^login|^signup/) {
        tangent 'login';
    }
};

# on '/'         => show '/';
# on '/choices'  => show 'choices';
```



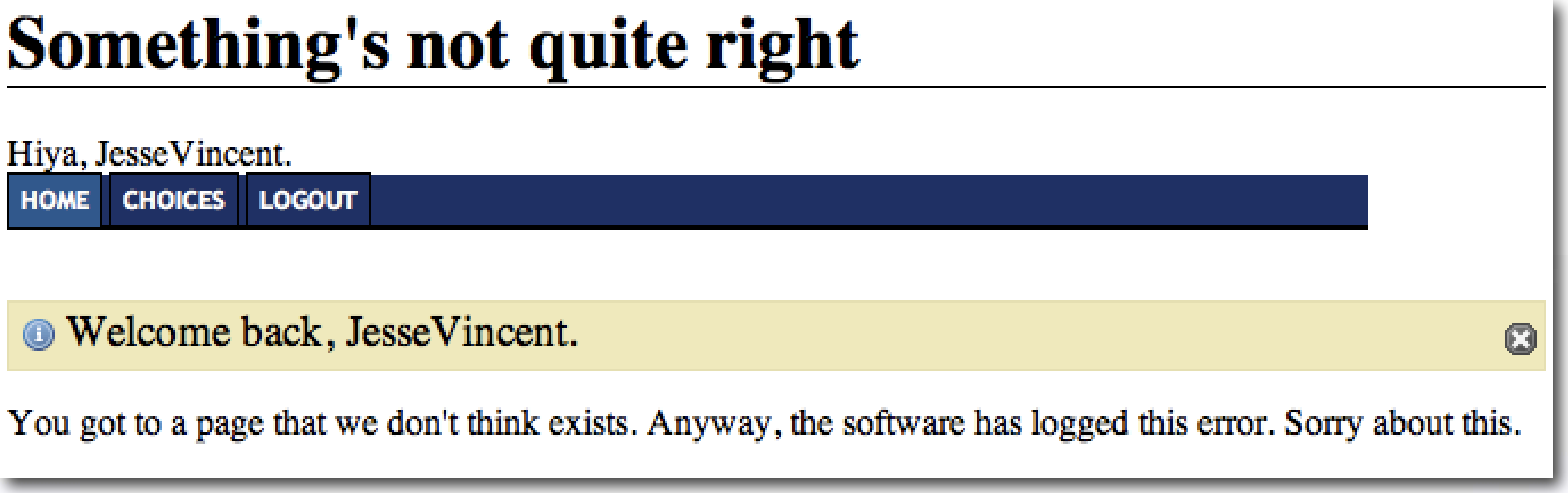
Something's not quite right

Hiya, JesseVincent.

Welcome back, JesseVincent.

You got to a page that we don't think exists. Anyway, the software has logged this error. Sorry about this.

```
ERROR - 404: User tried to get to /
```



Something's not quite right

Hiya, JesseVincent.

Welcome back, JesseVincent.

You got to a page that we don't think exists. Anyway, the software has logged this error. Sorry about this.

```
ERROR - 404: User tried to get to /
```

Now we should fill in the "/" view.

```
$ vim lib/Doxory/View.pm
$ vim lib/Doxory/View.pm
```

No TT2!
No Mason!
♡

```
package Doxory::View;
use Jifty::View::Declare -base;
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};
```

```perl
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};

private template new_choice => sub {
    div { form {
```

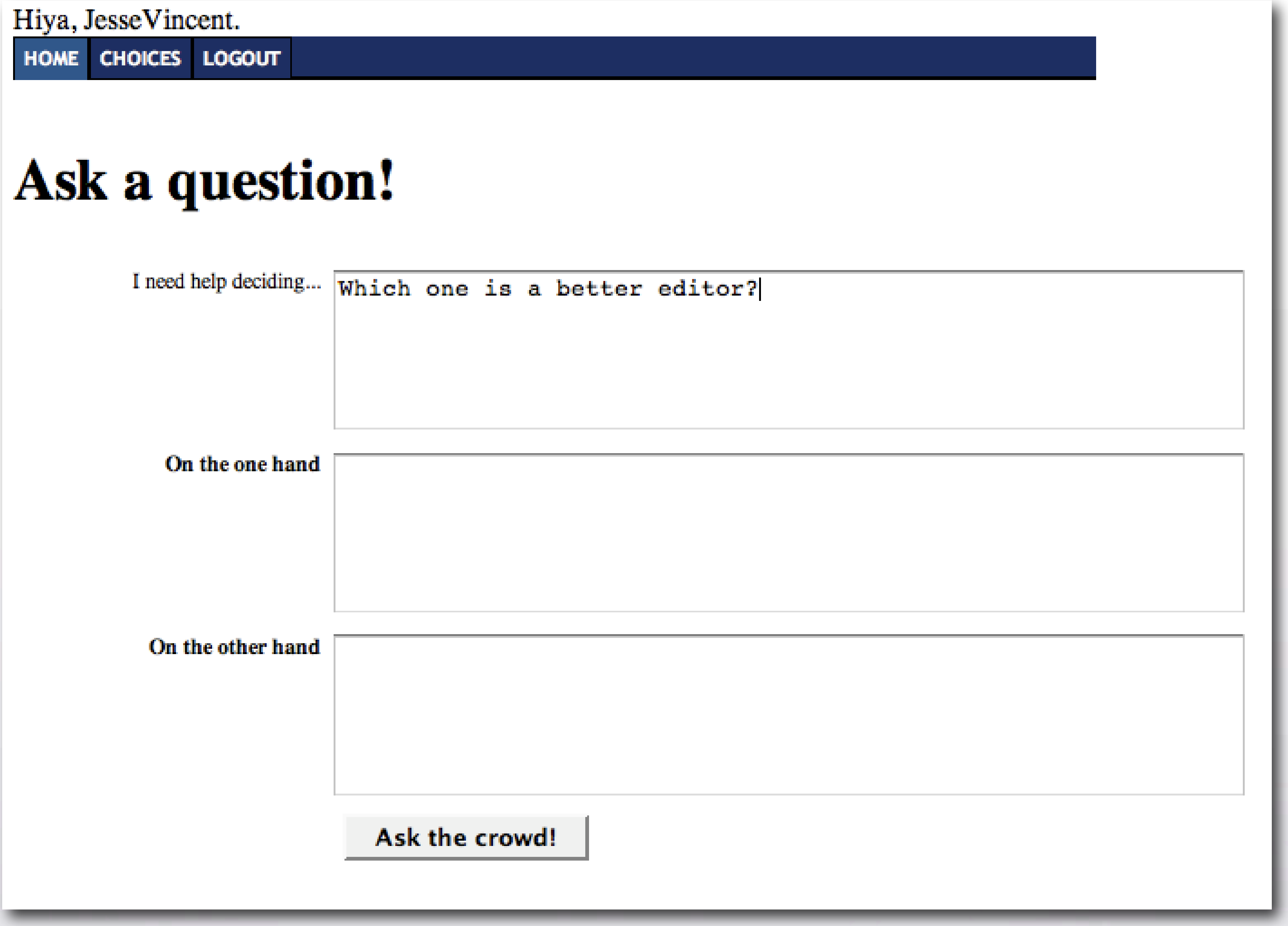
```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};

private template new_choice => sub {
    div { form {
        my $action = new_action( class => 'CreateChoice' );
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};

private template new_choice => sub {
    div { form {
        my $action = new_action( class => 'CreateChoice' );
        render_action( $action => ['name', 'a', 'b'] );
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};

private template new_choice => sub {
    div { form {
        my $action = new_action( class => 'CreateChoice' );
        render_action( $action => ['name', 'a', 'b'] );
        form_submit( label => 'Ask the Crowd!' );
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};

private template new_choice => sub {
    div { form {
        my $action = new_action( class => 'CreateChoice' );
        render_action( $action => ['name', 'a', 'b'] );
        form_submit( label => 'Ask the Crowd!' );
    } }
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page {
    h1 { 'Ask a question!' }
    show 'new_choice';
};

private template new_choice => sub {
    div { form {
        my $action = new_action( class => 'CreateChoice' );
        render_action( $action => ['name', 'a', 'b'] );
        form_submit( label => 'Ask the Crowd!' );
    } }
};
```



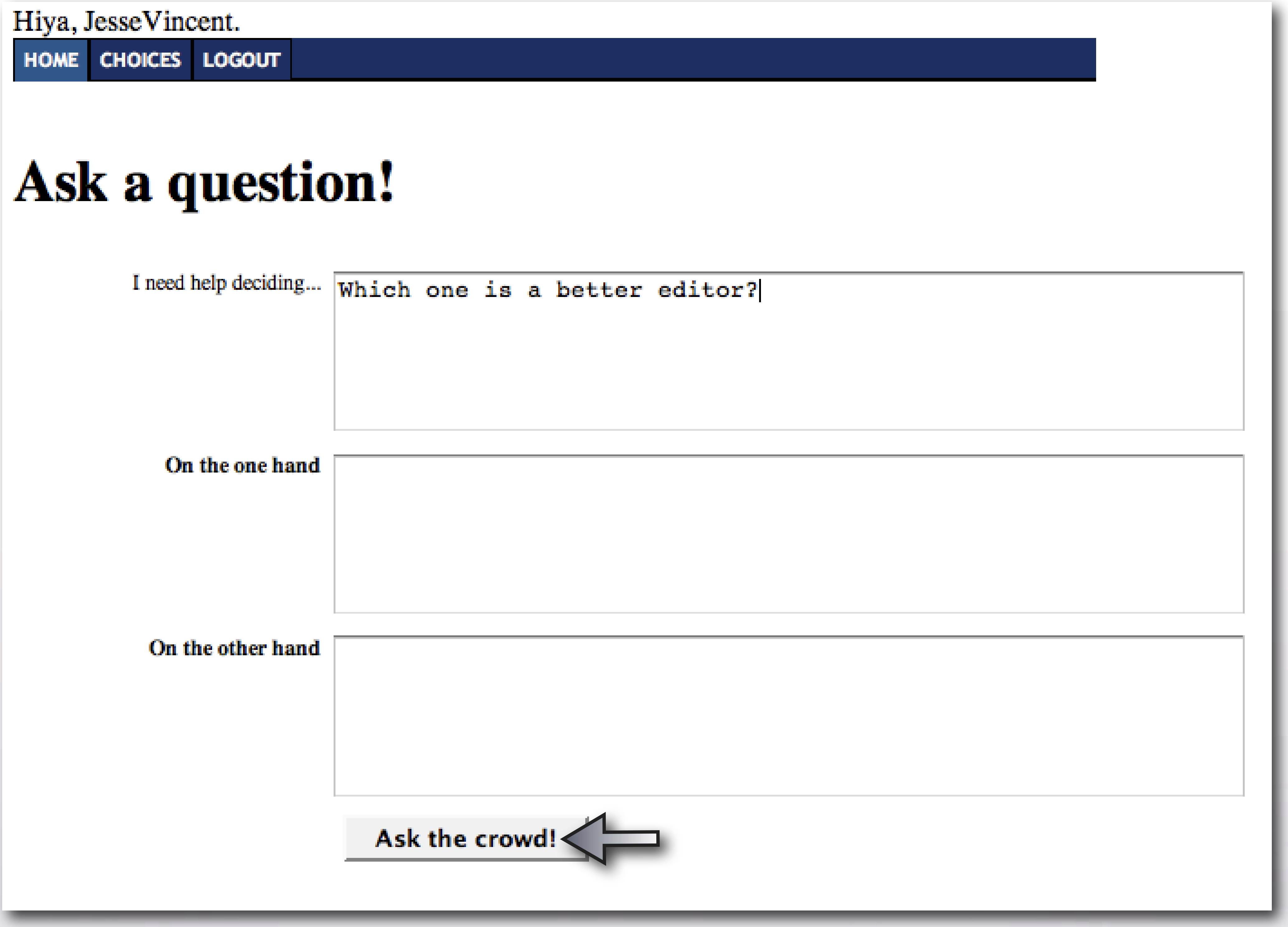
Hiya, JesseVincent.

Ask a question!

I need help deciding... Which one is a better editor?

On the one hand

On the other hand



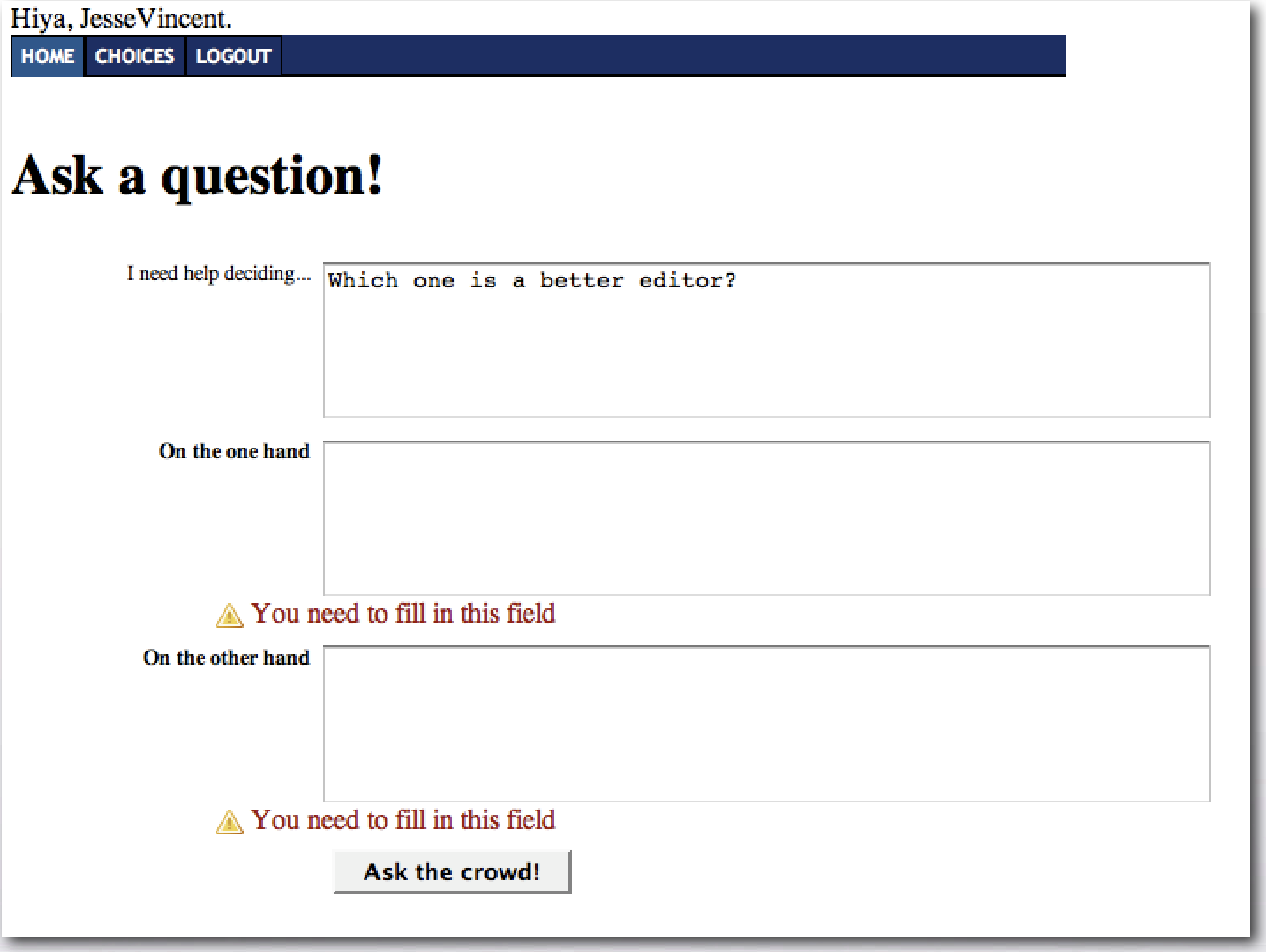
Hiya, JesseVincent.

Ask a question!

I need help deciding... Which one is a better editor?

On the one hand

On the other hand



Ask a question!

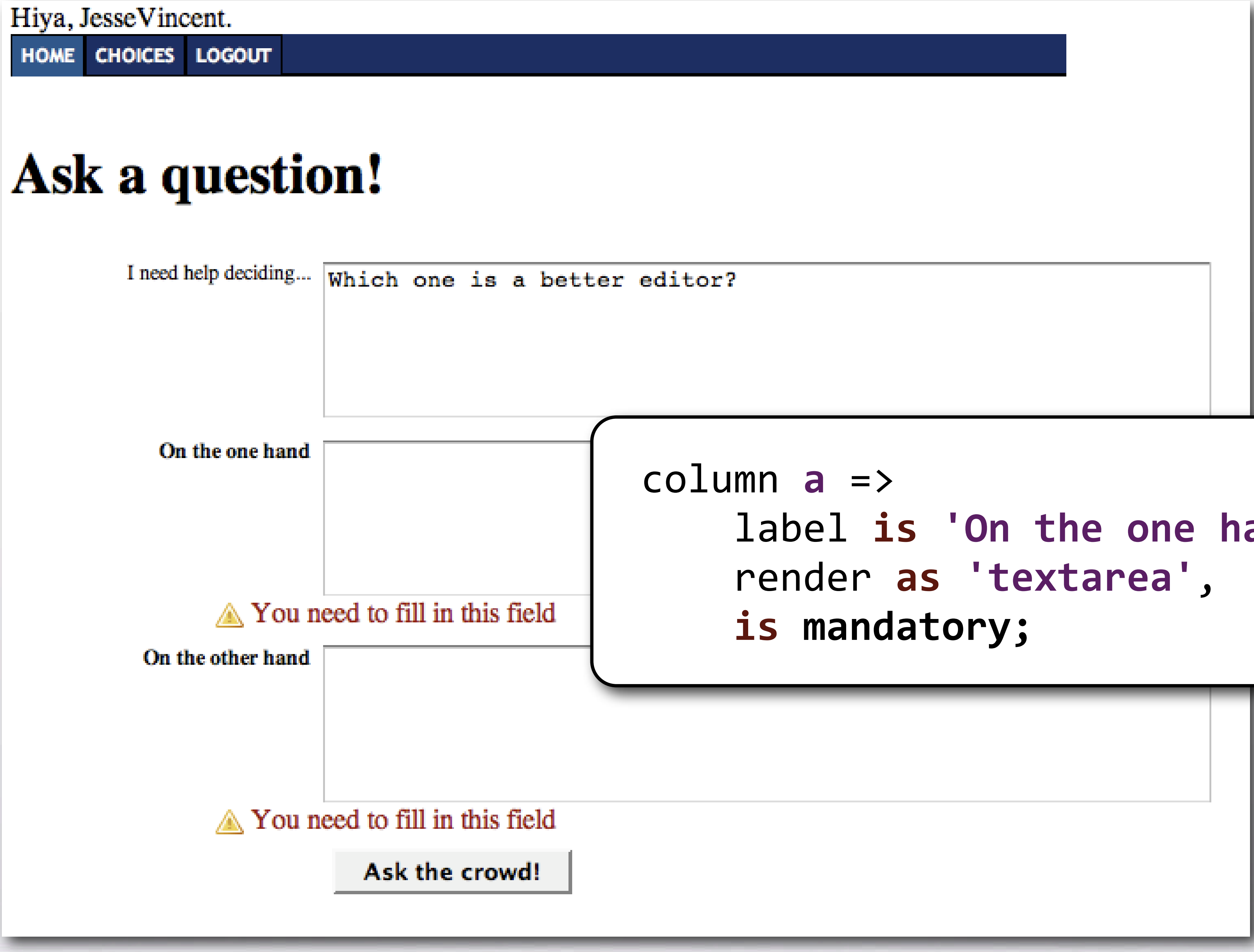
I need help deciding... Which one is a better editor?

On the one hand

⚠ You need to fill in this field

On the other hand

⚠ You need to fill in this field



Hiya, JesseVincent.

Ask a question!

I need help deciding... Which one is a better editor?

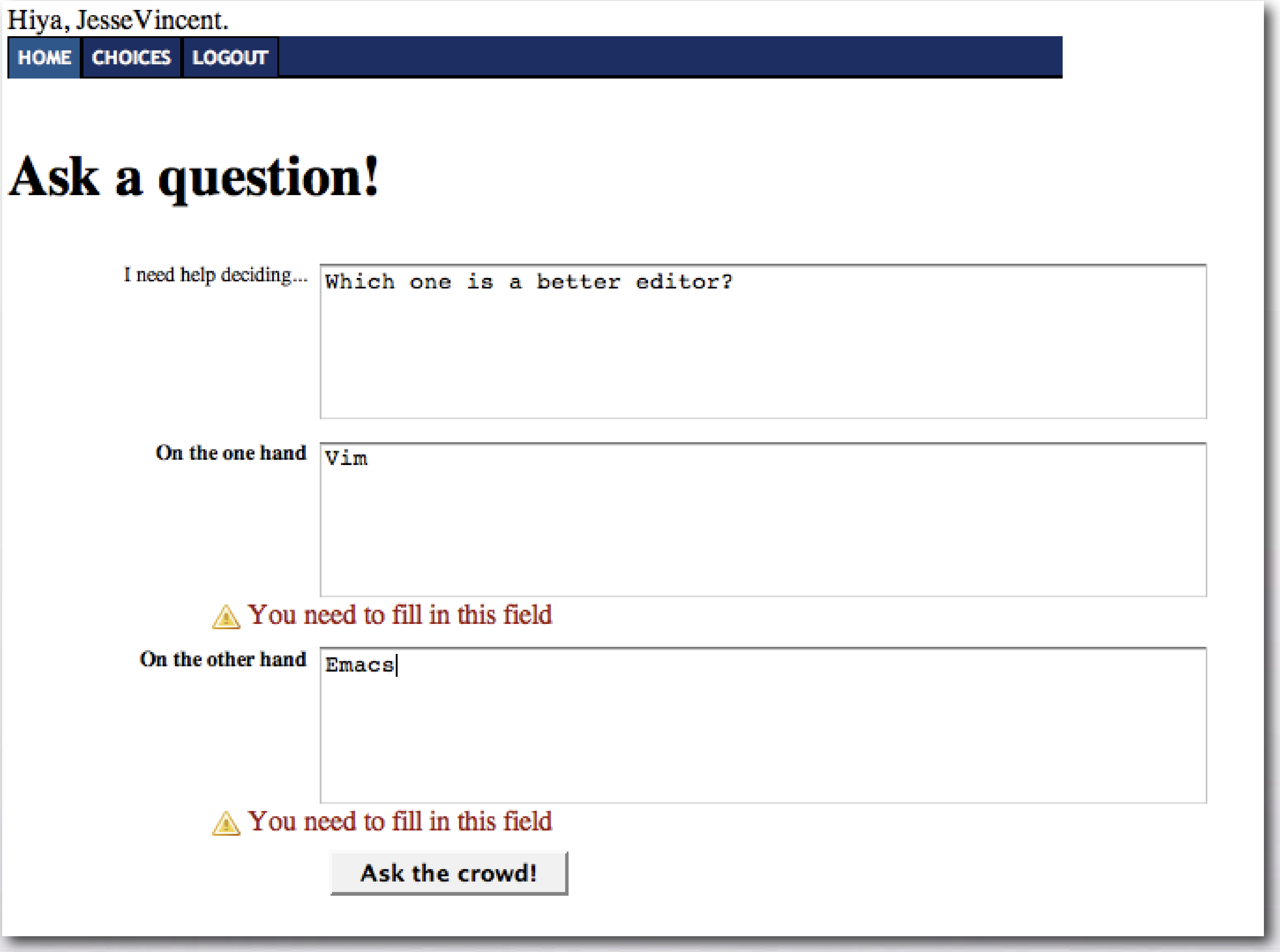
On the one hand

⚠ You need to fill in this field

On the other hand

⚠ You need to fill in this field

```
column a =>
    label is 'On the one hand',
    render as 'textarea',
    is mandatory;
```



Hiya, JesseVincent.

Ask a question!

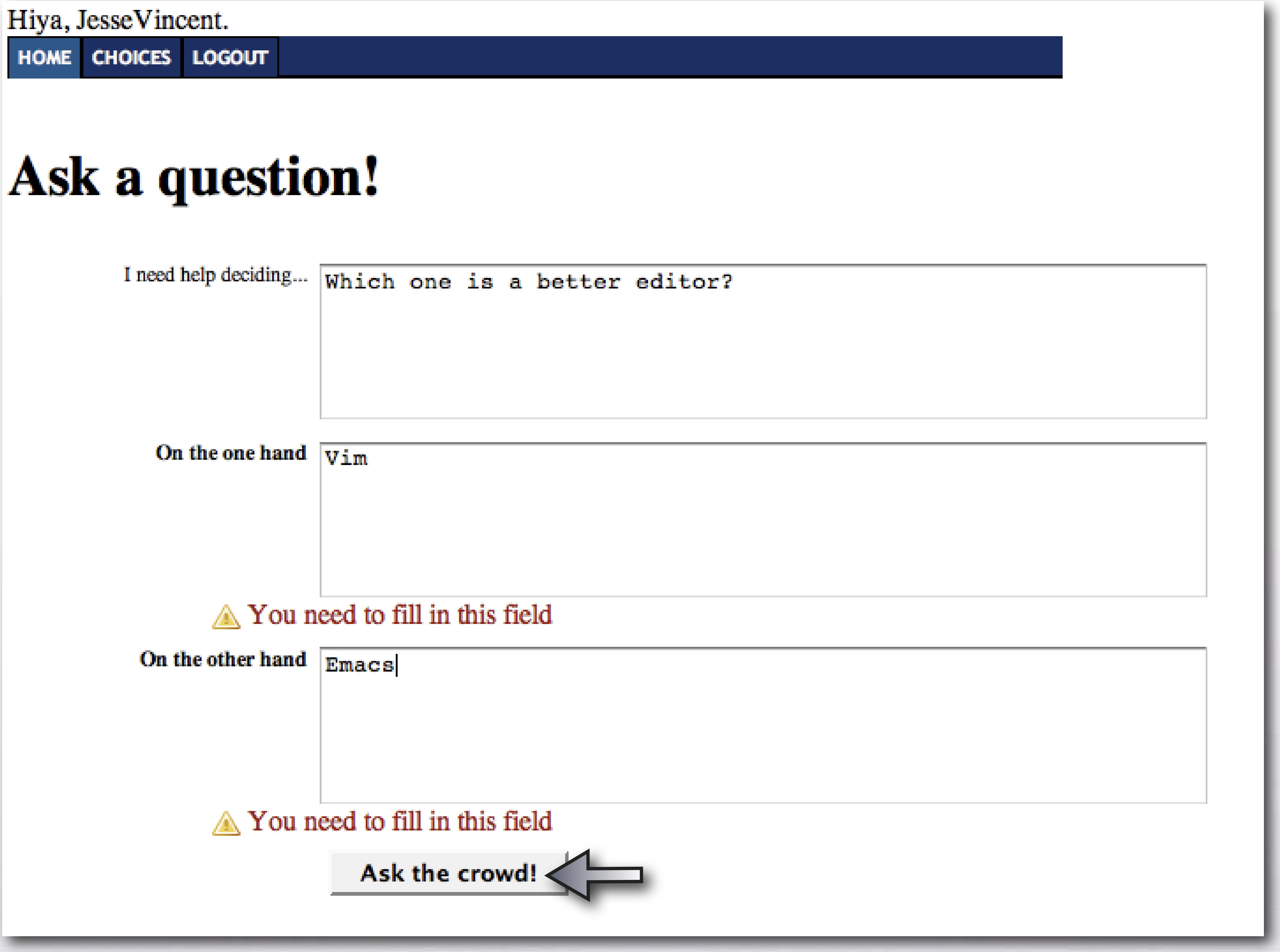
I need help deciding... Which one is a better editor?

On the one hand Vim

⚠ You need to fill in this field

On the other hand Emacs

⚠ You need to fill in this field



Ask a question!

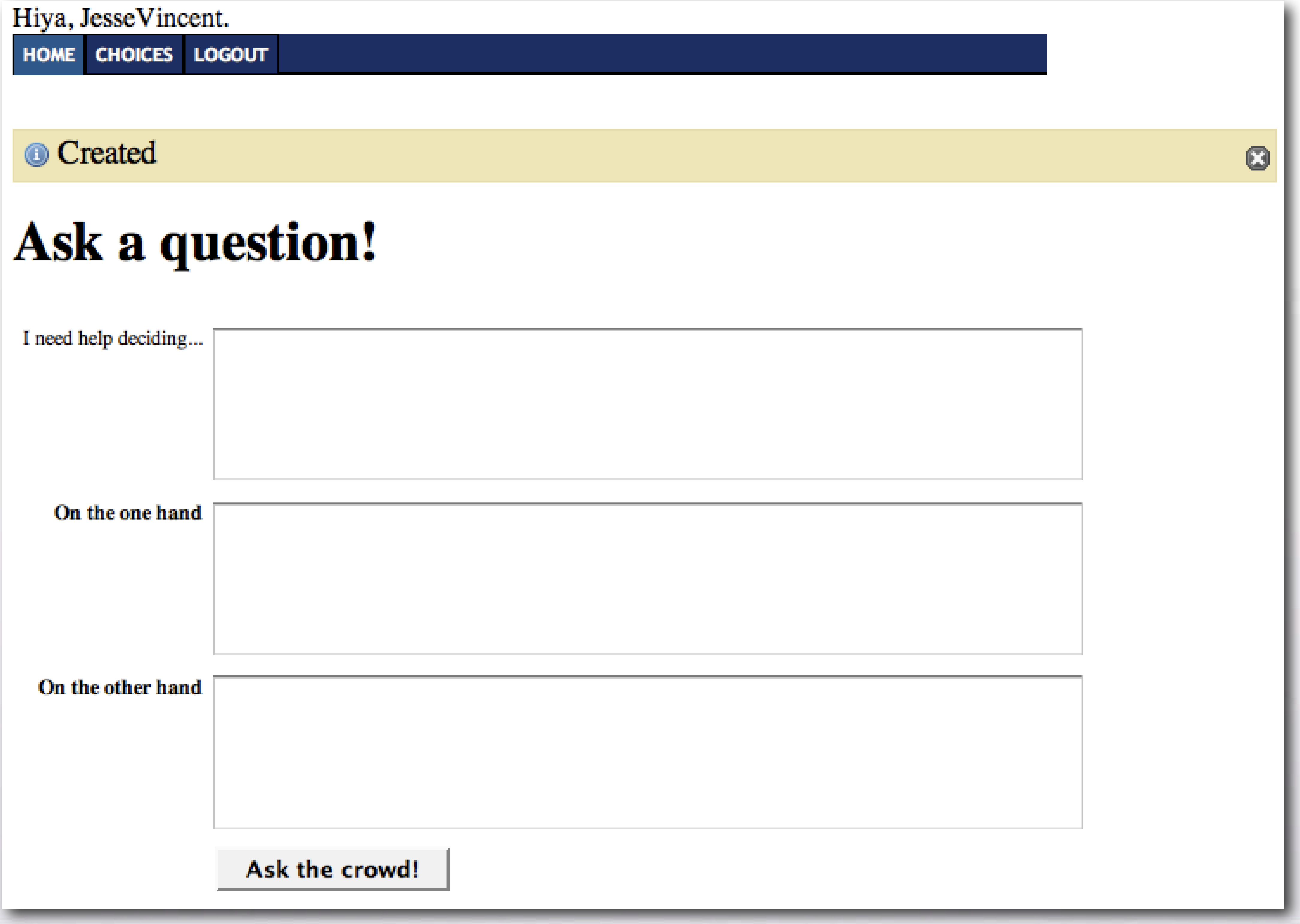
I need help deciding... Which one is a better editor?

On the one hand Vim

⚠ You need to fill in this field

On the other hand Emacs

⚠ You need to fill in this field



Hiya, JesseVincent.

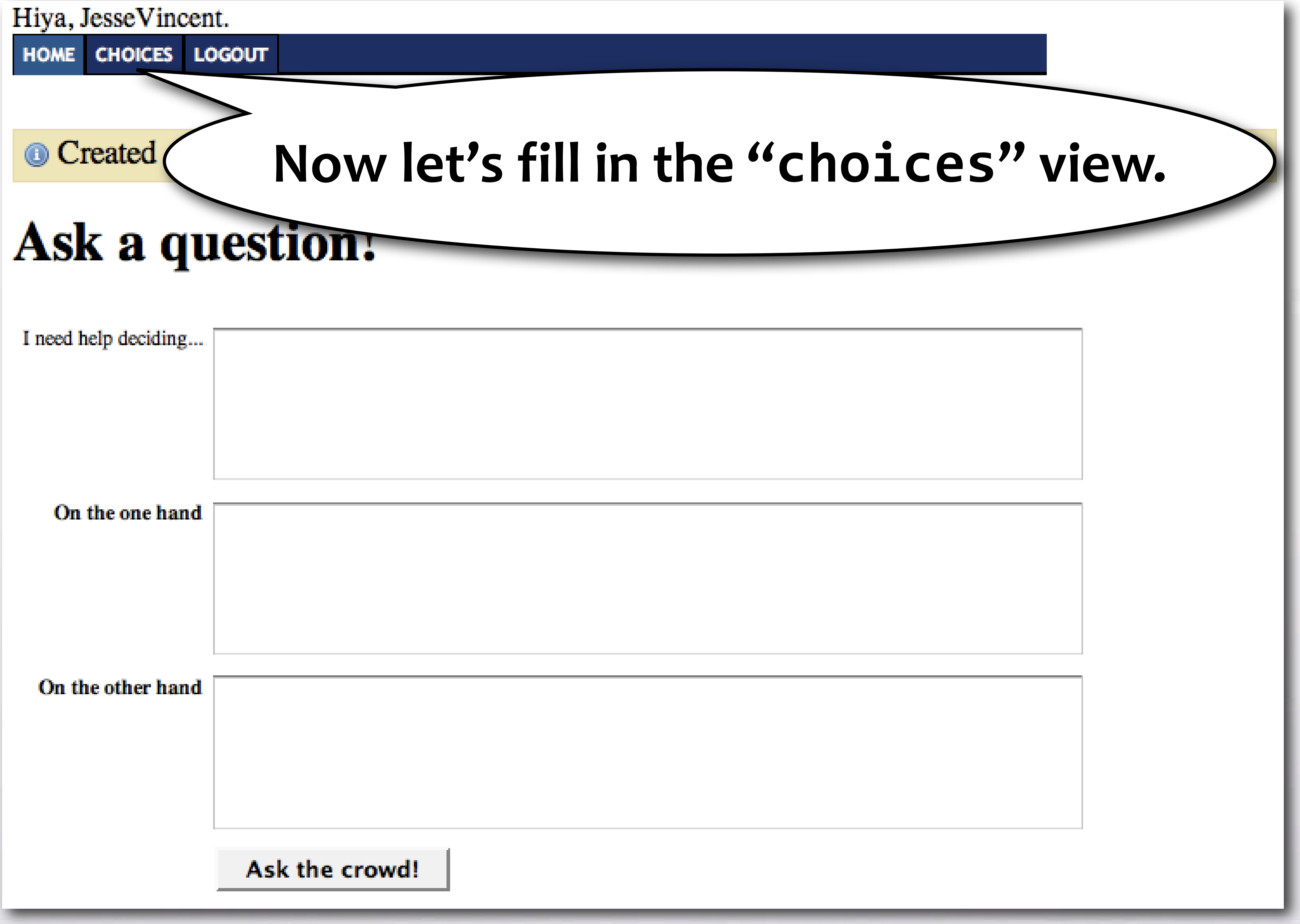
ⓘ Created

Ask a question!

I need help deciding...

On the one hand

On the other hand



Hiya, JesseVincent.

Created

Now let’s fill in the “choices” view.

Ask a question!

I need help deciding...

On the one hand

On the other hand

```
package Doxory::View;
use Jifty::View::Declare -base;
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
```

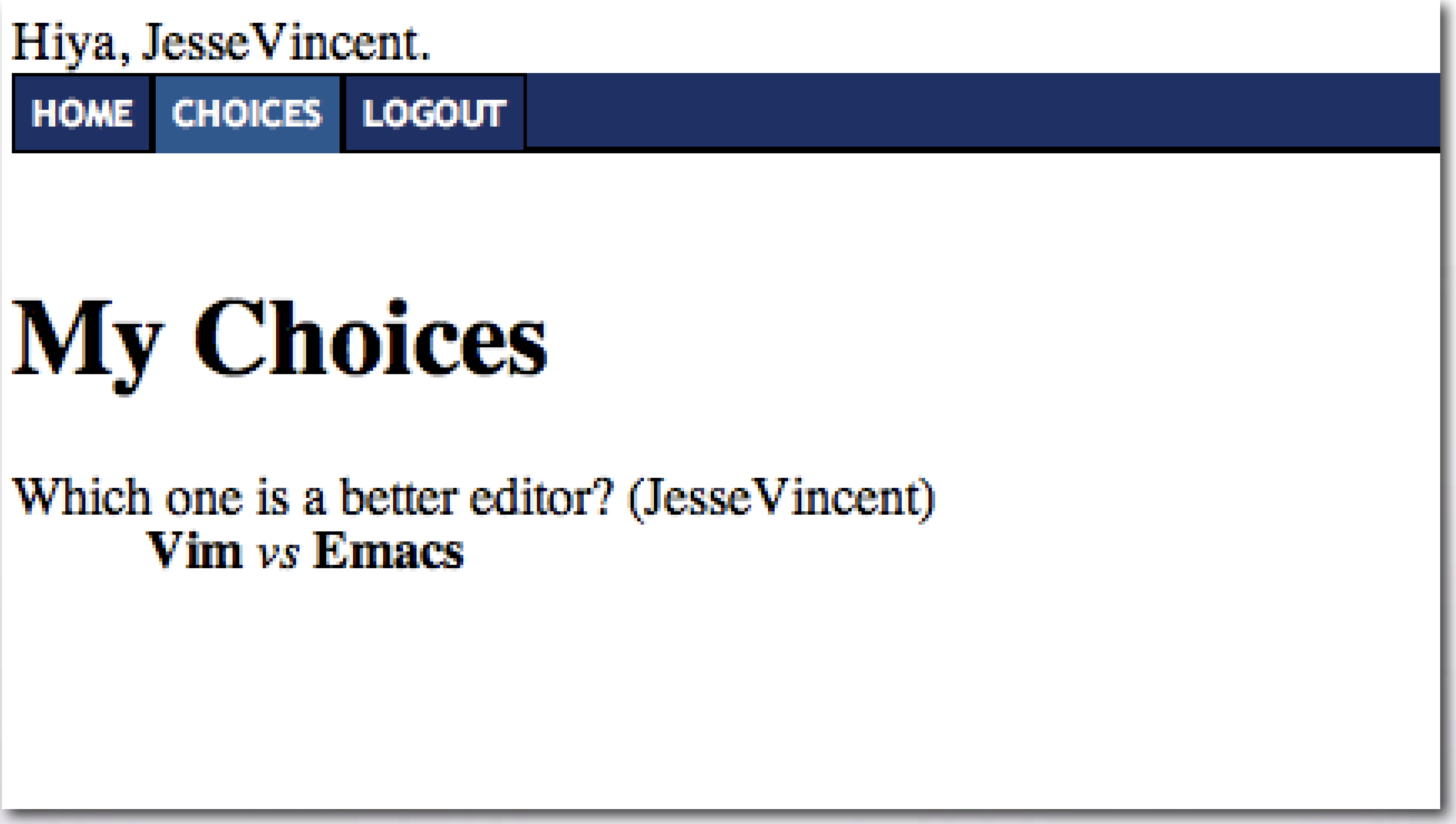
```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd { b { $c->a } em { 'vs' } b { $c->b } }
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd { b { $c->a } em { 'vs' } b { $c->b } }
        }
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd { b { $c->a } em { 'vs' } b { $c->b } }
        }
    }
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '/' => page { ... };
private template new_choice => sub { ... };

template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd { b { $c->a } em { 'vs' } b { $c->b } }
        }
    }
};
```

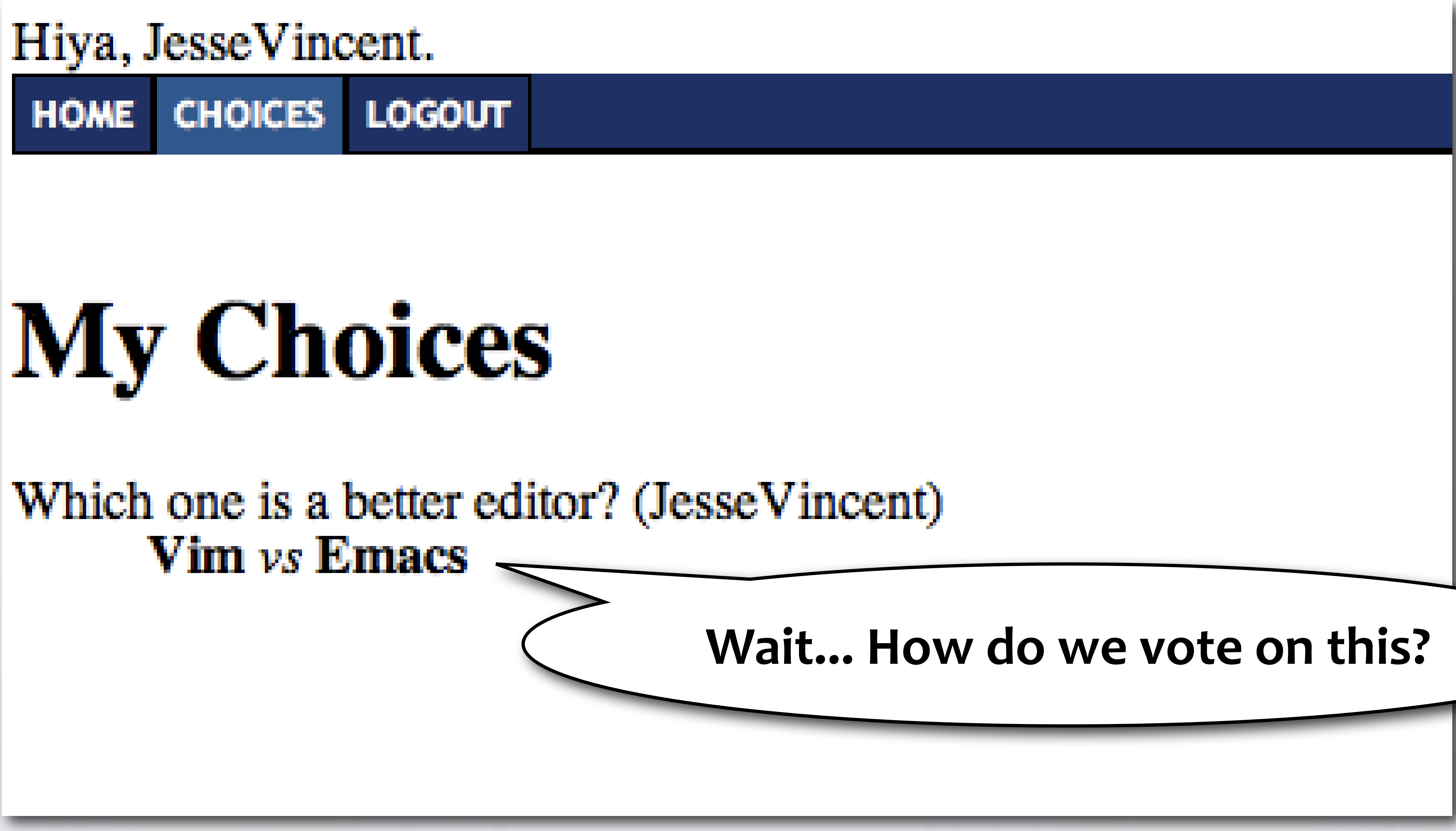


Hiya, JesseVincent.

My Choices

Which one is a better editor? (JesseVincent)

Vim *vs* **Emacs**



Hiya, JesseVincent.

My Choices

Which one is a better editor? (JesseVincent)

Vim *vs* **Emacs**

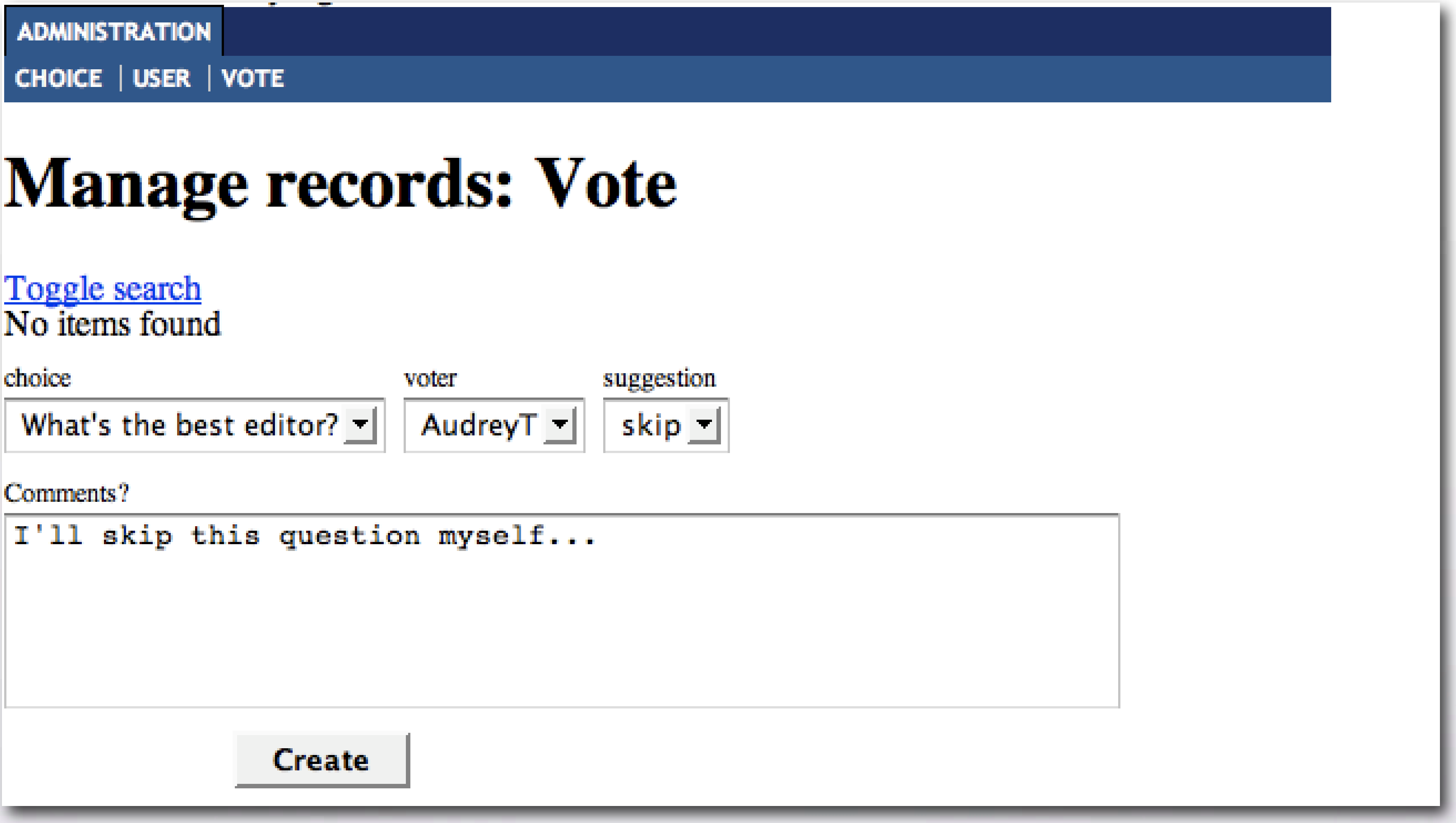
Wait... How do we vote on this?

☑

```
$ jifty model --name Vote
$ jifty model --name Vote
Writing file t/00-model-Vote.t
Writing file lib/Doxory/Model/Vote.pm
```

```
$ jifty model --name Vote
Writing file t/00-model-Vote.t
Writing file lib/Doxory/Model/Vote.pm

$ vim lib/Doxory/Model/Vote.pm
```

```perl
package Doxory::Model::Vote;
use Jifty::DBI::Schema;

use Doxory::Record schema {

column choice =>
    references Doxory::Model::Choice;

column voter =>
    references Doxory::Model::User;

column suggestion =>
    valid are qw( a b skip );

column comments =>
    label is 'Comments?',
    render as 'textarea’;

};
```



Manage records: Vote

No items found

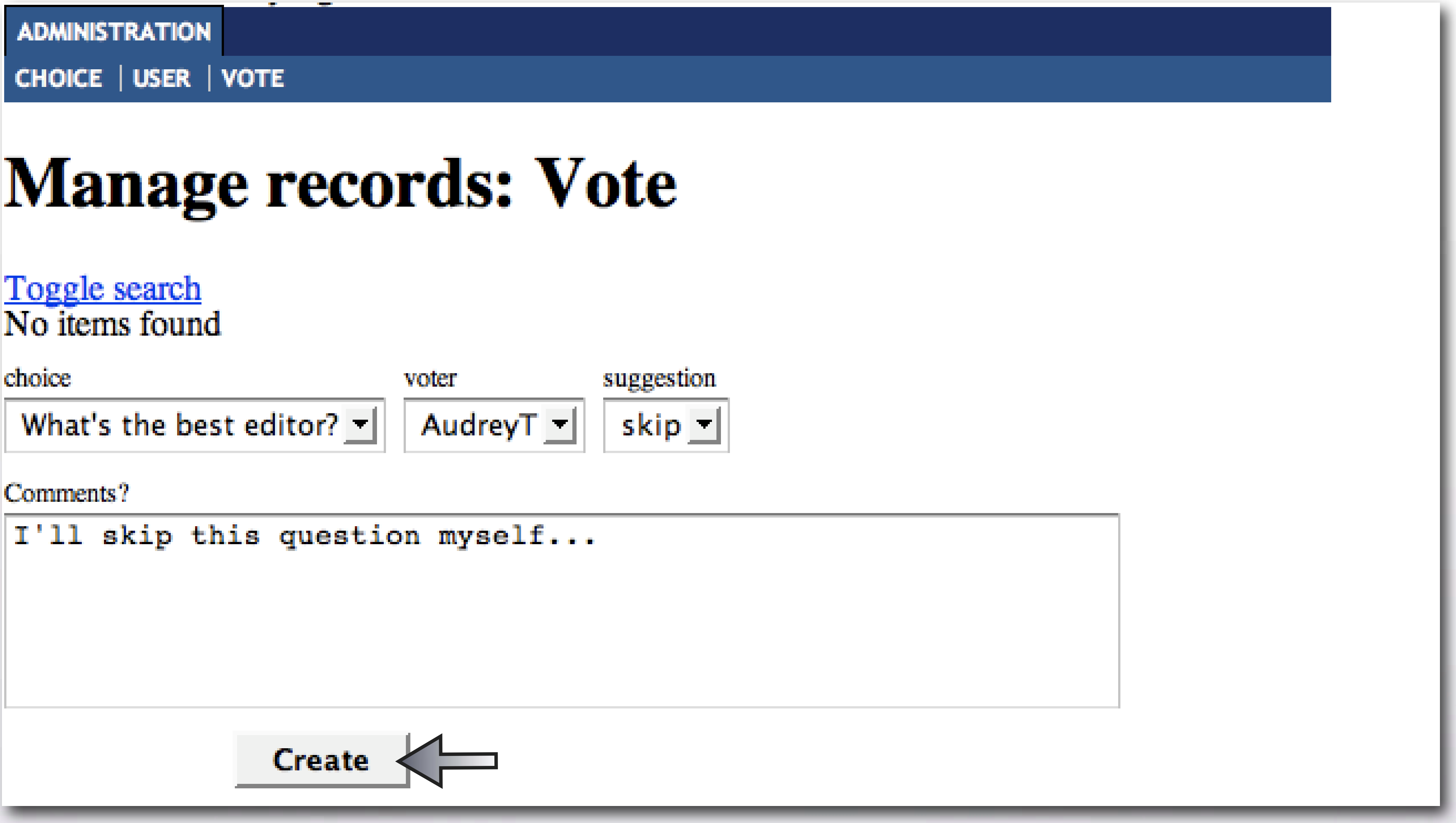
choice: What's the best editor?

voter: AudreyT

suggestion: skip

Comments?

I'll skip this question myself...



Manage records: Vote

No items found

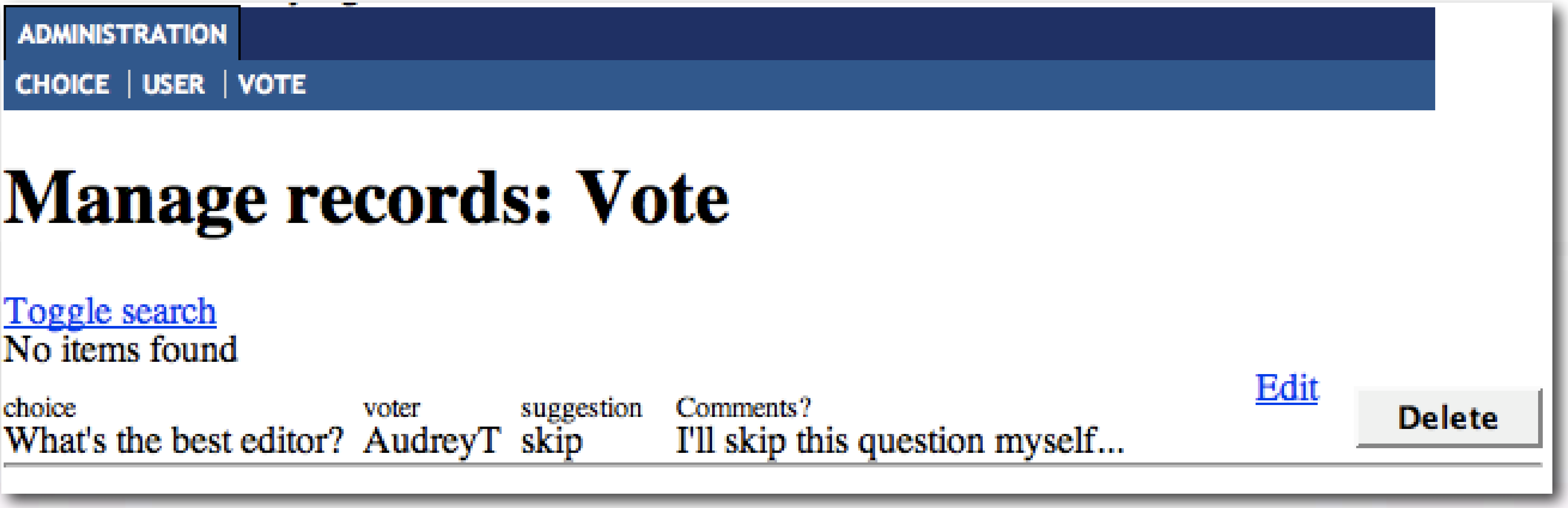
choice: What's the best editor?

voter: AudreyT

suggestion: skip

Comments?

```
I'll skip this question myself...
```

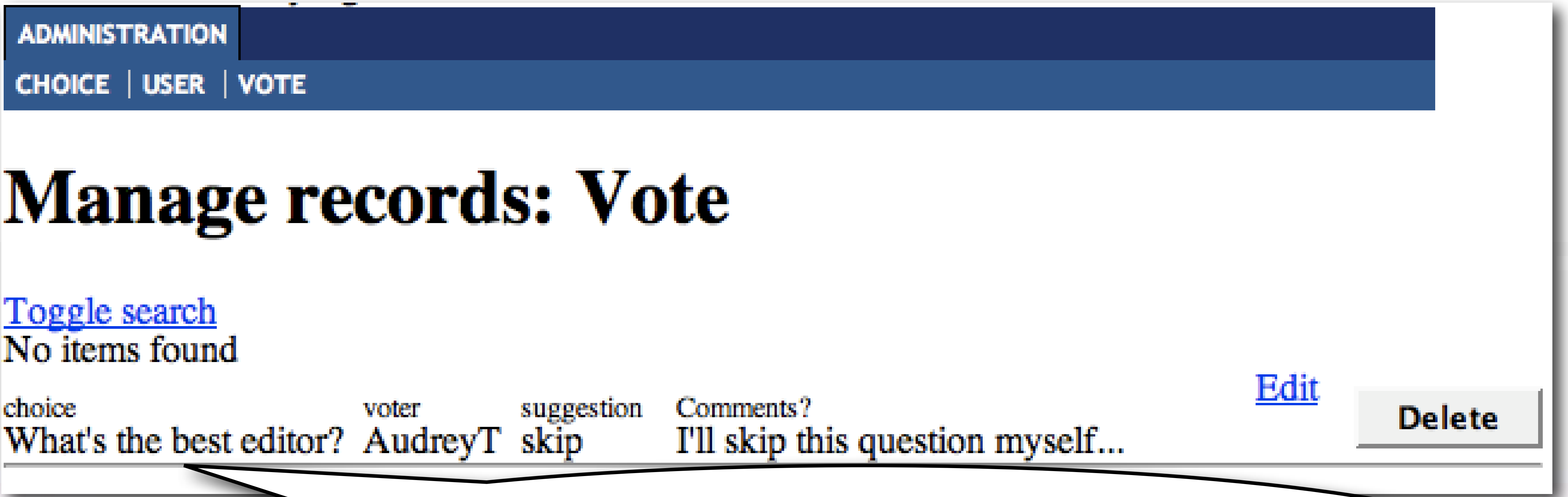


Manage records: Vote

Toggle search

No items found

choice	voter	suggestion	Comments?
What's the best editor?	AudreyT	skip	I'll skip this question myself...



Manage records: Vote

Toggle search
No items found

choice	voter	suggestion	Comments?
What's the best editor?	AudreyT	skip	I'll skip this question myself...

Now let’s implement “pick”.

```
package Doxory::Dispatcher;
use Jifty::Dispatcher -base;

before '*' => run {
    if (Jifty->web->current_user->id) {
        my $top = Jifty->web->navigation;

        $top->child( 'Pick!' => url => '/pick' );

        $top->child( 'Choices' => url => '/choices' );
    }
    elsif ($1 !~ /^login|^signup/) {
        tangent 'login';
    }
};

on '/' => show 'new_choice';
```

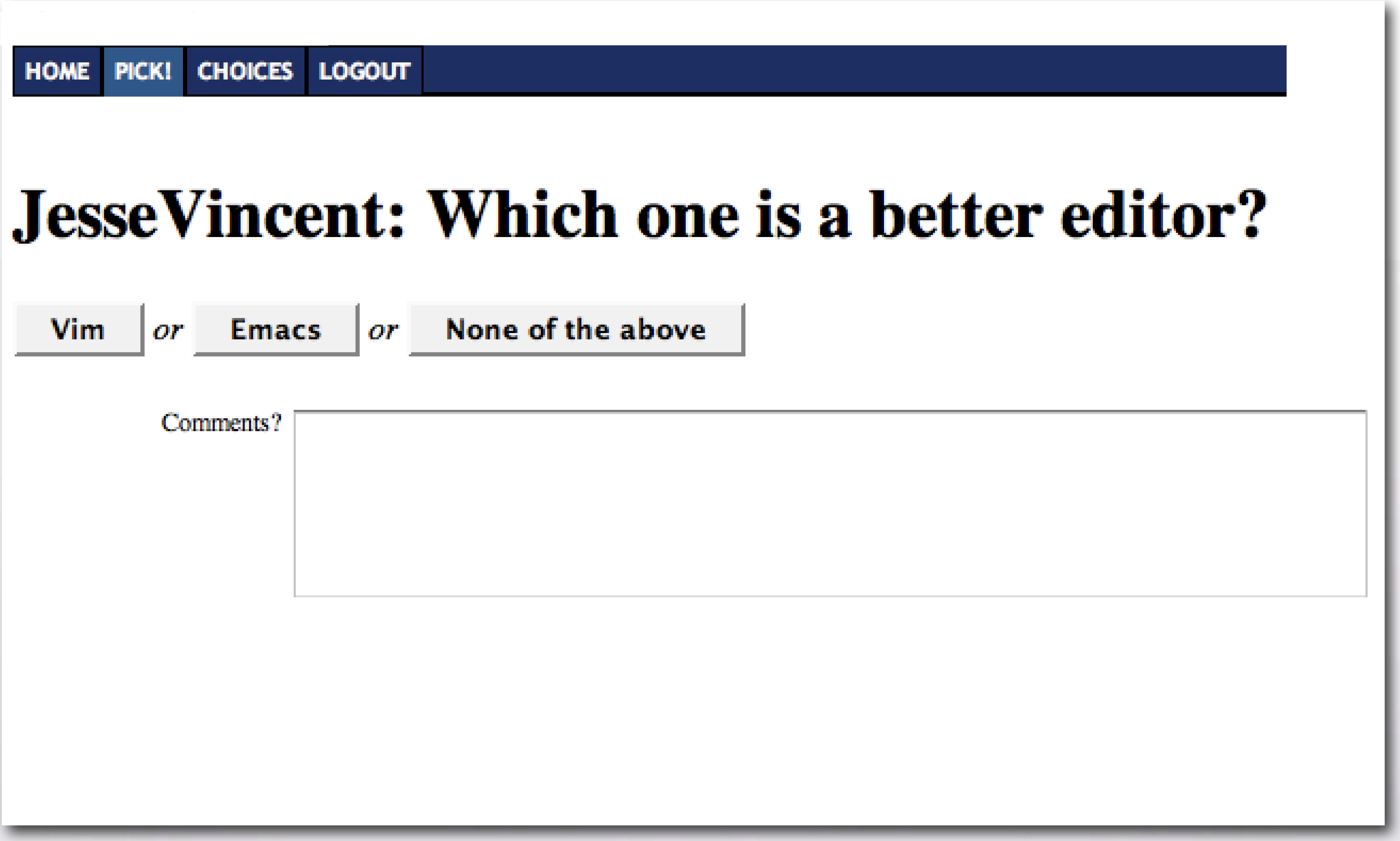
```
package Doxory::Dispatcher;
...
on pick => run {
    my $choices = Doxory::Model::ChoiceCollection->new;
    my $votes   = $choices->join(
        type    => 'left',
        alias1  => 'main',  column1 => 'id',
        table2  => 'votes', column2 => 'choice',
    );
    $choices->limit(
        leftjoin => $votes, column  => 'voter',
        value    => Jifty->web->current_user->id,
    );
    $choices->limit(
        alias    => $votes, column  => 'voter',
        operator => 'IS',   value   => 'NULL',
    );
    if (my $c = $choices->first) {
        set choice => $c;
    }
    else {
        show 'nothing_to_pick';
    }
};
```

```
package Doxory::Dispatcher;
...
on pick => run {
    my $choices = Doxory::Model::ChoiceCollection->new;
    my $votes   = $choices->join(
        type    => 'left',
        alias1  => 'main',  column1 => 'id',
        table2  => 'votes', column2 => 'choice',
    );
    $choices->limit(
        leftjoin => $votes, column  => 'voter',
        value    => Jifty->web->current_user->id,
    );
    $choices->limit(
        alias    => $votes, column  => 'voter',
        operator => 'IS',   value   => 'NULL',
    );
    if (my $c = $choices->first) {
        set choice => $c;
    }
    else {
        show 'nothing_to_pick';
    }
};
```

“1st choice I’ve not voted on”

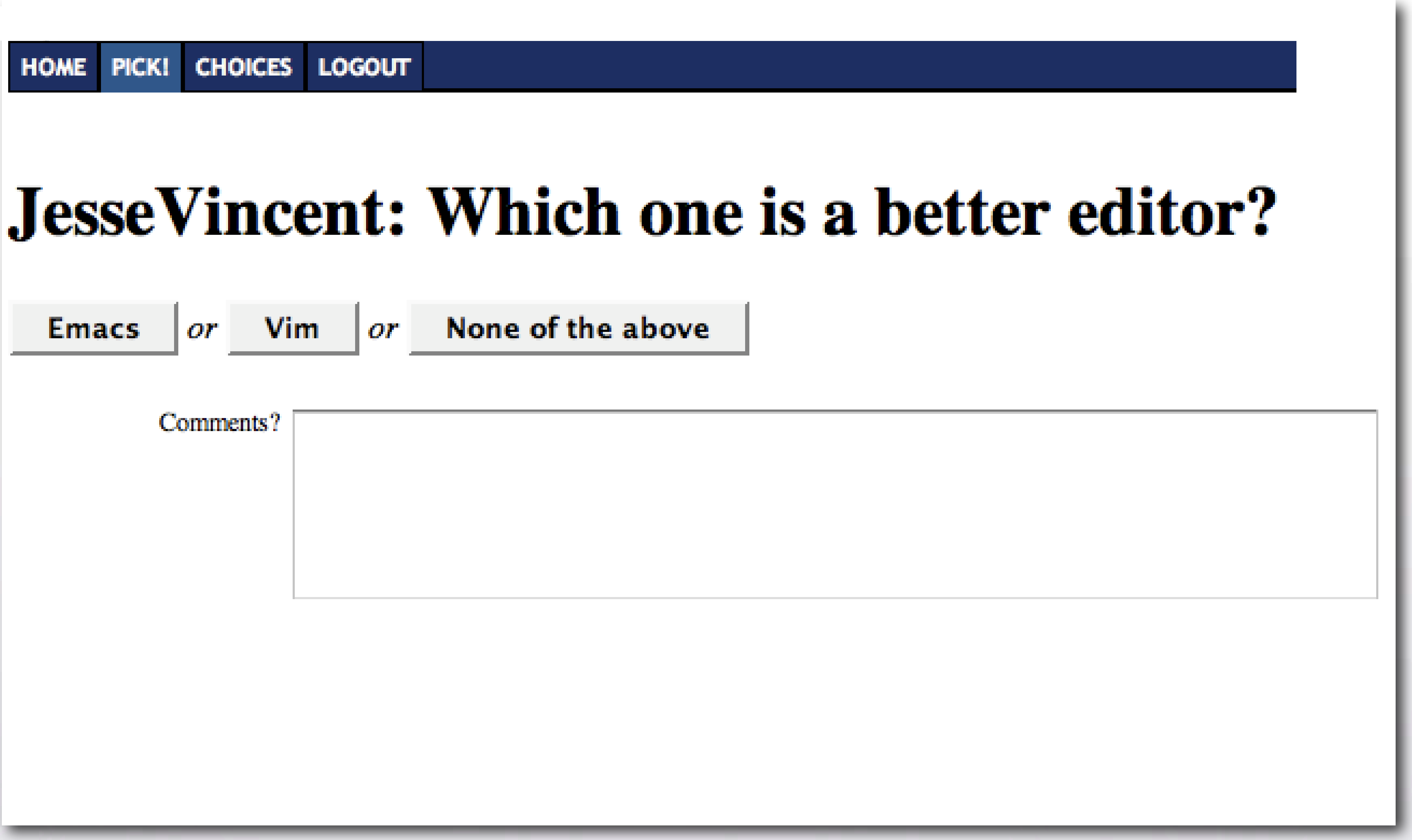
```
package Doxory::View;
use Jifty::View::Declare -base;

template '' => page { ... };
private template choices => sub { ... };

template pick => page {
    my $choice = get('choice');
    my $action = new_action( class => 'CreateVote' );
    my $redir  = new_action(
        class     => "Jifty::Action::Redirect",
        arguments => { url => '/pick' },
    );
    my %args   = (
        choice => $choice->id,
        voter  => Jifty->web->current_user->id,
    );
    h1 { $choice->asked_by->name, ': ', $choice->name }
    div { form {
        my ($x, $y) = map { $action->button(
            submit      => [ $action, $redir ],
            label       => $choice->$_,
            arguments   => { suggestion => $_, %args },
        ) } ((rand > 0.5) ? ('a', 'b') : ('b', 'a'));

        span { $x } em { 'or' } span { $y } em { 'or' } span {
            $action->button(
                submit      => [ $action, $redir ],
                label       => 'None of the above',
                arguments   => { suggestion => 'skip', %args },
            );
        }

        p { render_param( $action => 'comments' ) };
    } }
};
```

```
package Doxory::View;
use Jifty::View::Declare -base;

template '' => page { ... };
private template new_choice => sub { ... };
template choices => page { ... };
template pick => page { ... };

template nothing_to_pick => page {
    h1 { "There's nothing for you to pick." }

    p { "No one you know is angsting about anything.
         Everybody knows where they're going to dinner,
         what to do on their next date and whether to
         drop that class.  You have such lovely and
         well adjusted friends." }

    h2 { "Maybe it's time to ask for some advice..." };

    show 'new_choice';
};
```

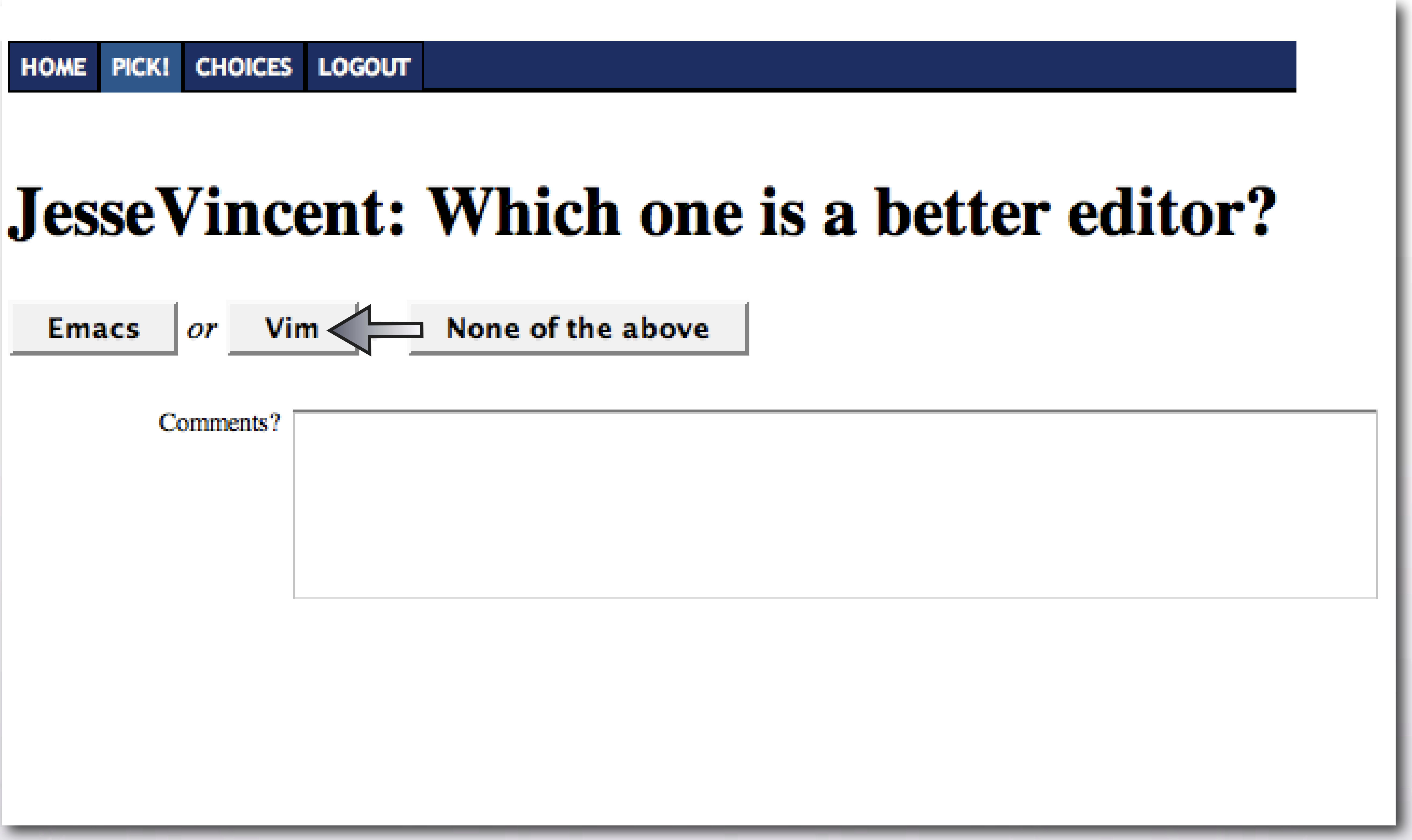


JesseVincent: Which one is a better editor?

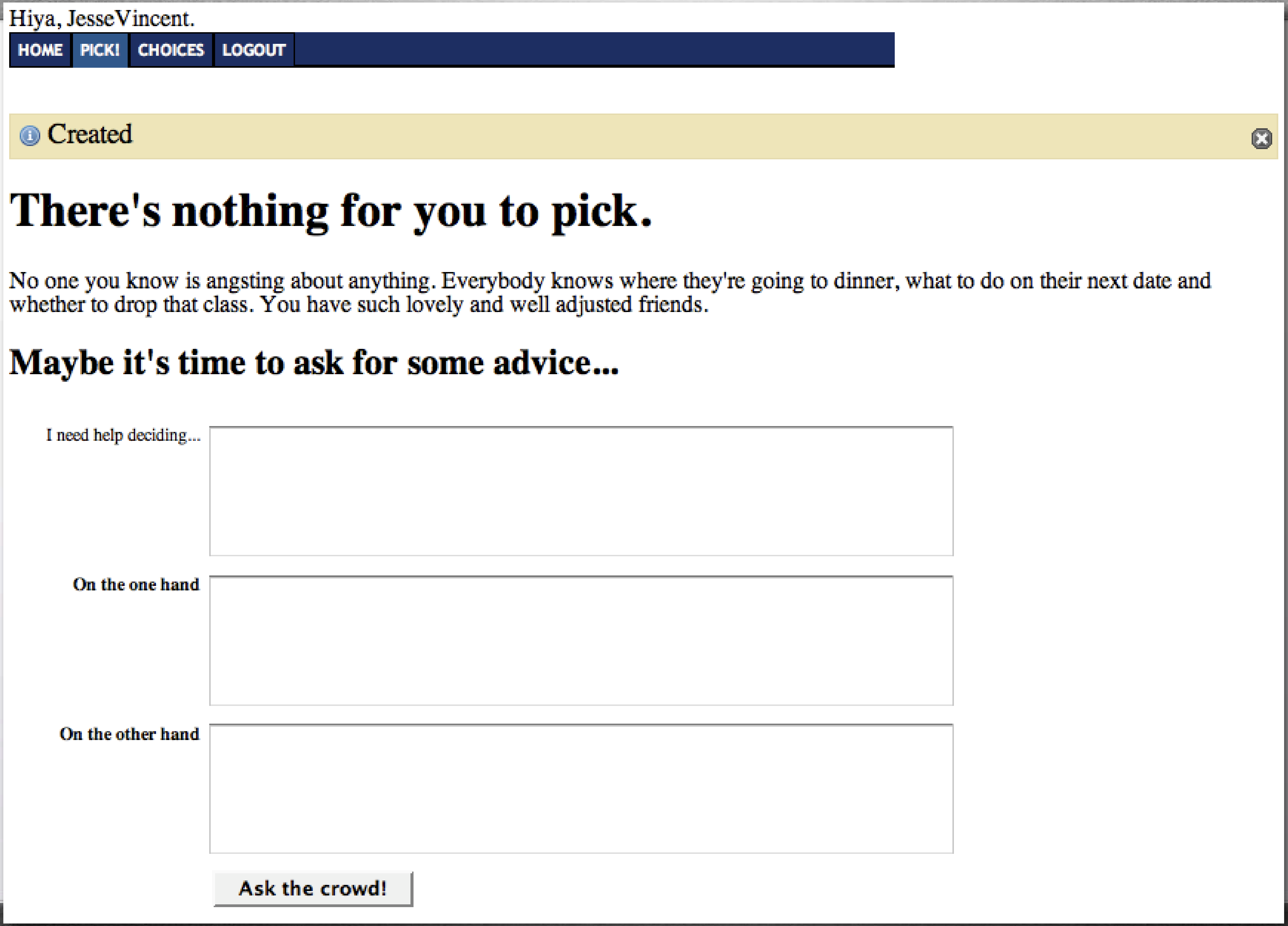
Comments?



JesseVincent: Which one is a better editor?



JesseVincent: Which one is a better editor?



Hiya, JesseVincent.

Created

There's nothing for you to pick.

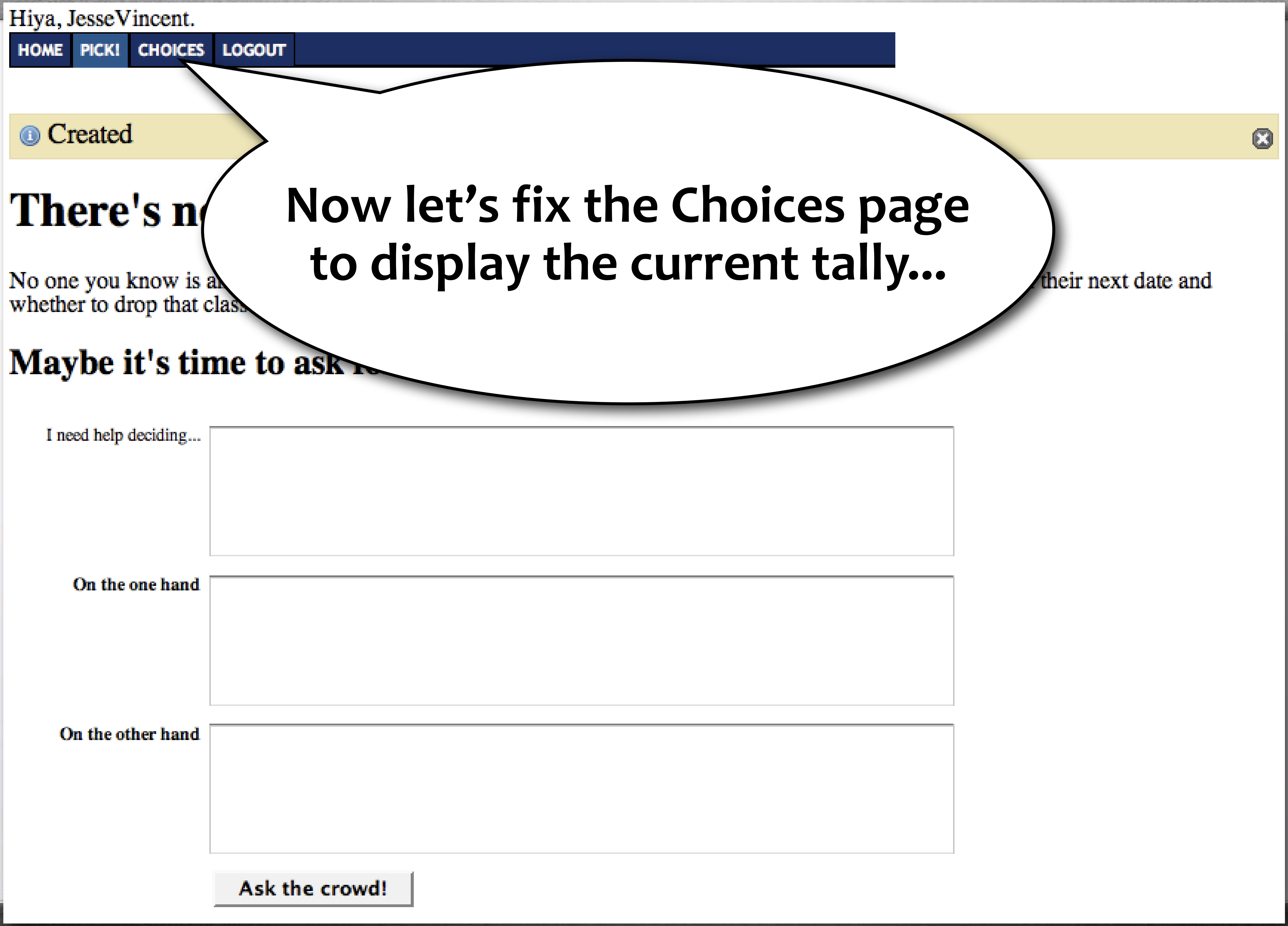
No one you know is angsting about anything. Everybody knows where they're going to dinner, what to do on their next date and whether to drop that class. You have such lovely and well adjusted friends.

Maybe it's time to ask for some advice...

I need help deciding...

On the one hand

On the other hand



Hiya, JesseVincent.

Created

There's n

No one you know is a … their next date and whether to drop that clas…

Maybe it's time to ask …

I need help deciding...

On the one hand

On the other hand

Now let's fix the Choices page to display the current tally...

```
package Doxory::Model::Choice;
use Jifty::DBI::Schema;
use Doxory::Record schema { ... }

sub in_favor_of_a {
    my $self = shift;
    $self->in_favor_of('a');
}

sub in_favor_of_b {
    my $self = shift;
    $self->in_favor_of('b');
}

sub in_favor_of {
    my ($self, $x) = @_;
    my $votes = Doxory::Model::VoteCollection->new;
    $votes->limit(column => 'choice',     value => $self->id);
    $votes->limit(column => 'suggestion', value => $x);
    return $votes;
}
```

```
package Doxory::View;
use Jifty::View::Declare -base;
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
```

```perl
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {

                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
```

```
                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
                b { $c->b, ' (', $c->in_favor_of_b->count, ')' }
```

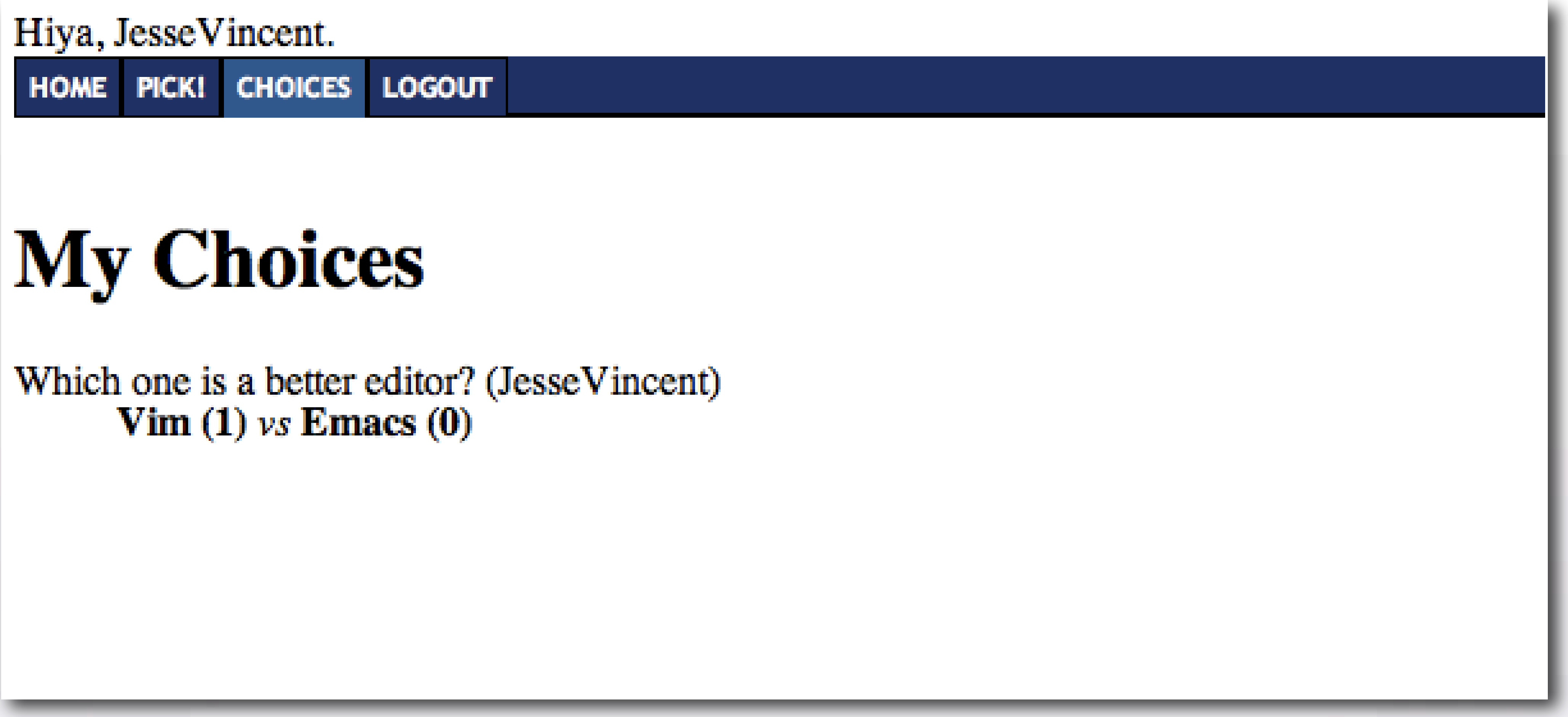
```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
                b { $c->b, ' (', $c->in_favor_of_b->count, ')' }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
                b { $c->b, ' (', $c->in_favor_of_b->count, ')' }
            }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
                b { $c->b, ' (', $c->in_favor_of_b->count, ')' }
            }
        }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {

                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
                b { $c->b, ' (', $c->in_favor_of_b->count, ')' }

            }
        }
    }
```

```
package Doxory::View;
use Jifty::View::Declare -base;
...
template choices => page {
    h1 { 'My Choices' }
    dl {
        my $choices = Doxory::Model::ChoiceCollection->new;
        $choices->limit(
            column => 'asked_by',
            value  => Jifty->web->current_user->id,
        );
        while (my $c = $choices->next) {
            dt { $c->name, ' (', $c->asked_by->name, ')' }
            dd {
                b { $c->a, ' (', $c->in_favor_of_a->count, ')' }
                em { 'vs' }
                b { $c->b, ' (', $c->in_favor_of_b->count, ')' }
            }
        }
    }
};
```

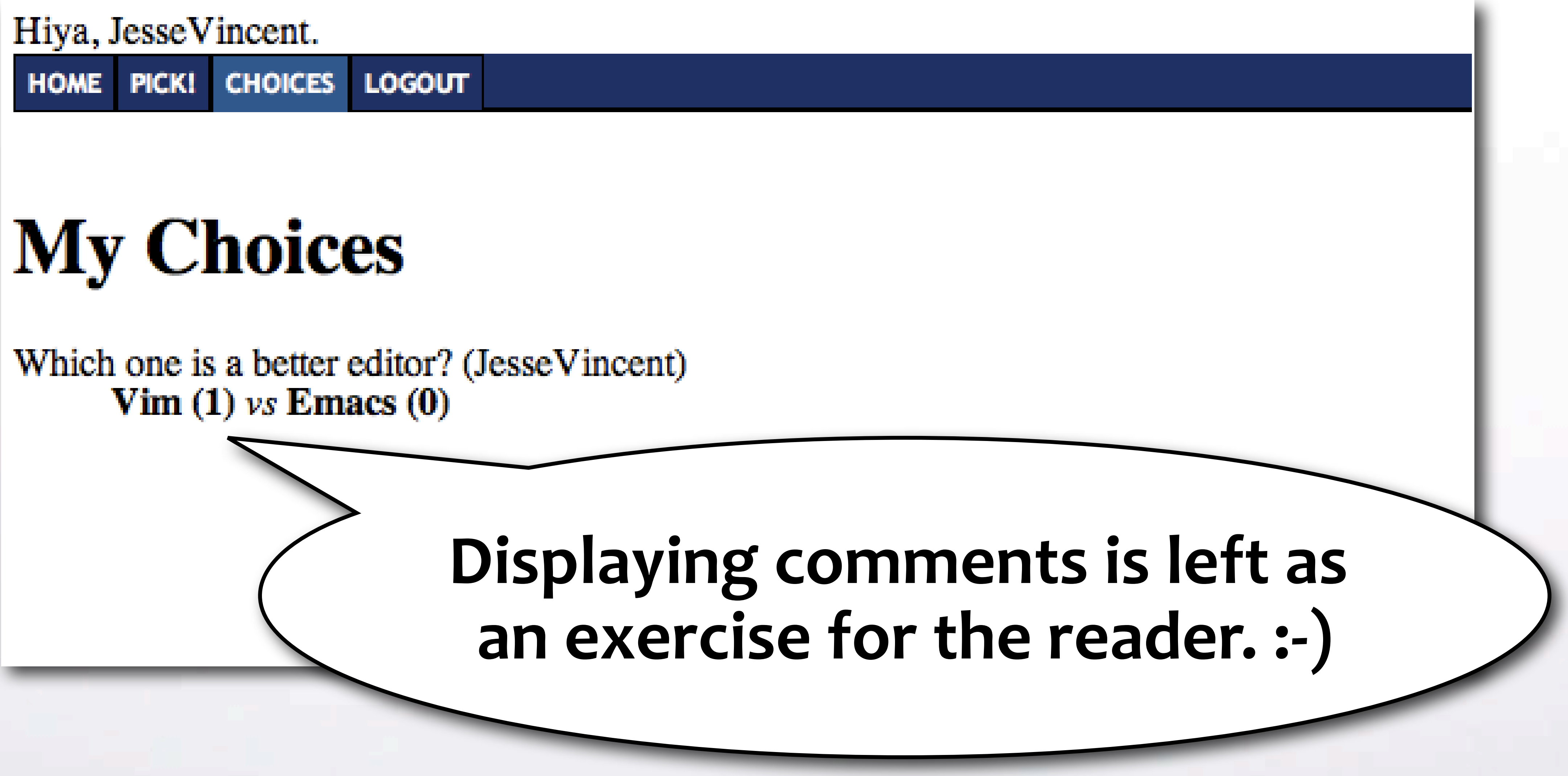


Hiya, JesseVincent.

My Choices

Which one is a better editor? (JesseVincent)

Vim (1) *vs* **Emacs (0)**



Hiya, JesseVincent.

My Choices

Which one is a better editor? (JesseVincent)
Vim (1) *vs* **Emacs (0)**

Displaying comments is left as an exercise for the reader. :-)

All Done!

All Done!

♨ `lib/Doxory/Model/User.pm`

All Done!

- `lib/Doxory/Model/User.pm`
- `lib/Doxory/Model/Choice.pm`

All Done!

- lib/Doxory/Model/User.pm
- lib/Doxory/Model/Choice.pm
- lib/Doxory/Model/Vote.pm

All Done!

♨ `lib/Doxory/Model/User.pm`

♨ `lib/Doxory/Model/Choice.pm`

♨ `lib/Doxory/Model/Vote.pm`

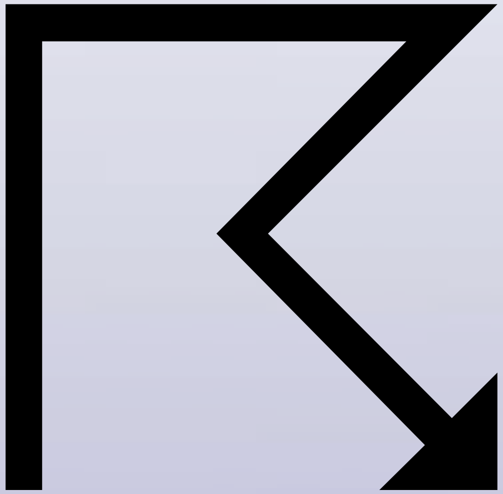
♨ `lib/Doxory/Dispatcher.pm`

All Done!

- ♨ `lib/Doxory/Model/User.pm`
- ♨ `lib/Doxory/Model/Choice.pm`
- ♨ `lib/Doxory/Model/Vote.pm`
- ♨ `lib/Doxory/Dispatcher.pm`
- ♨ `lib/Doxory/View.pm`

All Done!

- ♨ `lib/Doxory/Model/User.pm`
- ♨ `lib/Doxory/Model/Choice.pm`
- ♨ `lib/Doxory/Model/Vote.pm`
- ♨ `lib/Doxory/Dispatcher.pm`
- ♨ `lib/Doxory/View.pm`



New U.S. Law

New U.S. Law

- Banning profanity

New U.S. Law

- Banning profanity
- To protect the children

New U.S. Law

- Banning profanity
- To protect the children
- …from the internet

Doxory Censorship

Doxory Censorship

- No problem

Doxory Censorship

- No problem
- Back to the code!

Doxory Censorship

- No problem
- Back to the code!
- `lib/Doxory/Model/Choice.pm`

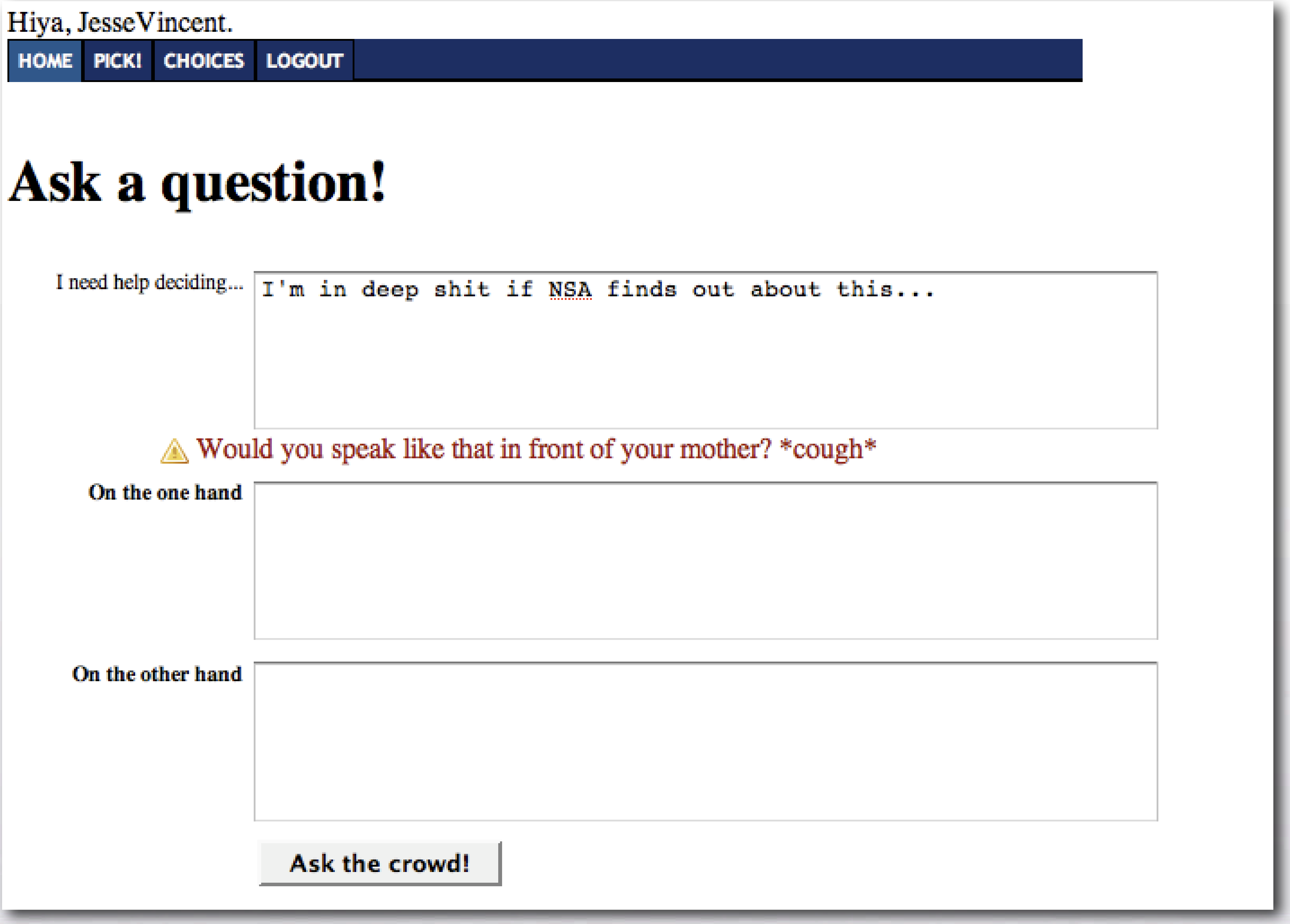
```
package Doxory::Model::Choice;
```

```perl
package Doxory::Model::Choice;

use Regexp::Common 'profanity_us';
```

```
package Doxory::Model::Choice;

use Regexp::Common 'profanity_us';

sub validate_name {
   my ($self, $name) = @_;
```

```
package Doxory::Model::Choice;

use Regexp::Common 'profanity_us';

sub validate_name {
   my ($self, $name) = @_;

   if ($name =~ /$RE{profanity}/i) {
       return (
           0, 'Would you speak like that '.
              'in front of your mother? *cough*'
       );
   }
```

```
package Doxory::Model::Choice;

use Regexp::Common 'profanity_us';

sub validate_name {
   my ($self, $name) = @_;

   if ($name =~ /$RE{profanity}/i) {
       return (
           0, 'Would you speak like that '.
              'in front of your mother? *cough*'
       );
   }

   return 1;
}
```



Hiya, JesseVincent.

Ask a question!

I need help deciding... I'm in deep shit if NSA finds out about this...

⚠ Would you speak like that in front of your mother? *cough*

On the one hand

On the other hand

Validator

Validator

- Not good enough

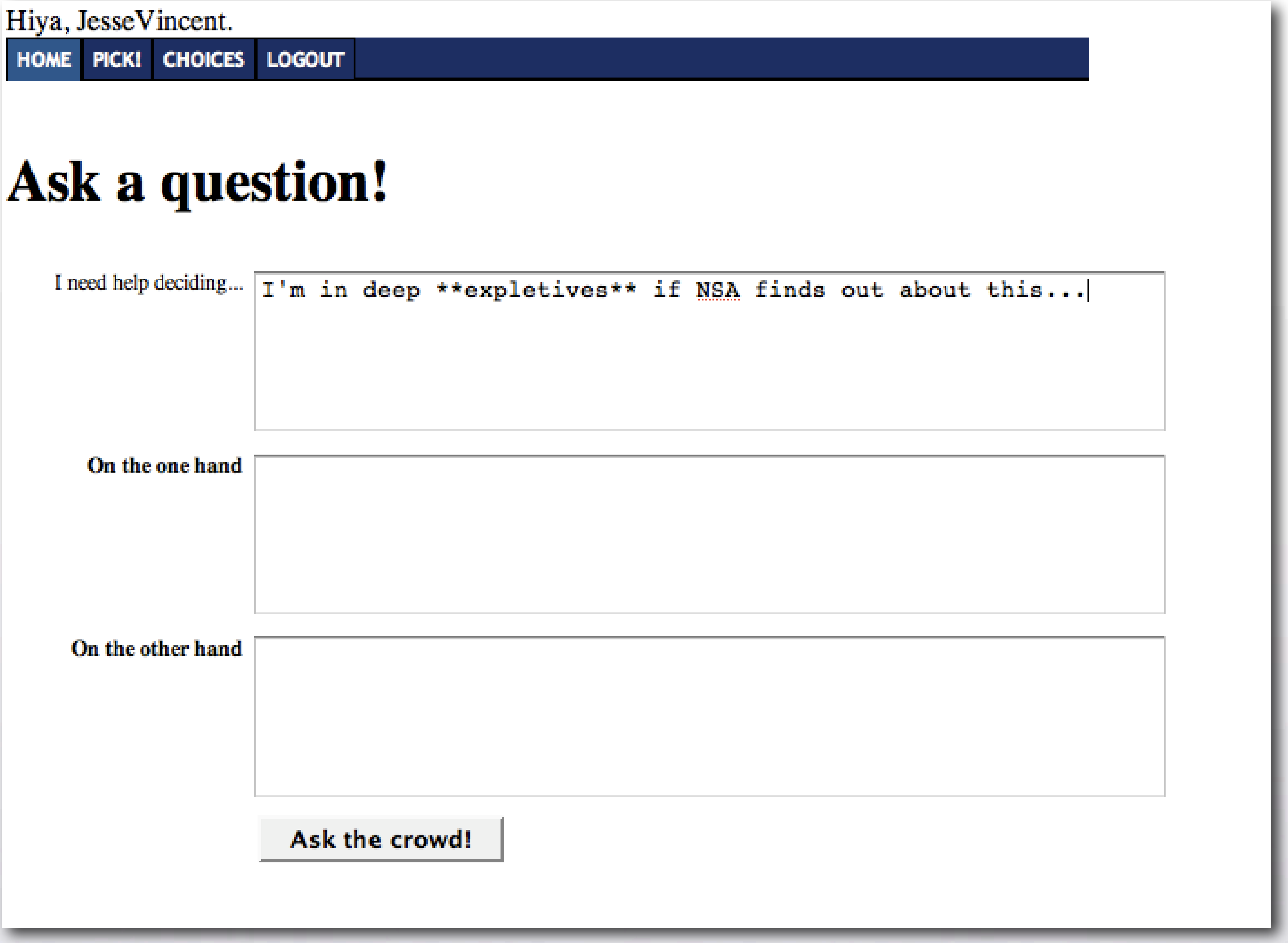
Validator

- Not good enough
- Stops people asking questions

Validator

- Not good enough
- Stops people asking questions
- Let’s clean it up directly!

```
package Doxory::Model::Choice;
use Regexp::Common 'profanity_us';

sub canonicalize_name {
   my ($self, $name) = @_;

   $name =~ s/$RE{profanity}/**expletives**/gi;
   return $name;
}
```



Hiya, JesseVincent.

Ask a question!

I need help deciding... I'm in deep **expletives** if NSA finds out about this...

On the one hand

On the other hand

Canonicalizer

Canonicalizer

- Part of the model

Canonicalizer

- Part of the model
- Same for autocompleters

Canonicalizer

- Part of the model
- Same for autocompleters
- *Now we're done!*

∀

Development Tools

Development Tools

∀ Standalone Server

Development Tools

∀ Standalone Server

∀ `Module::Install`

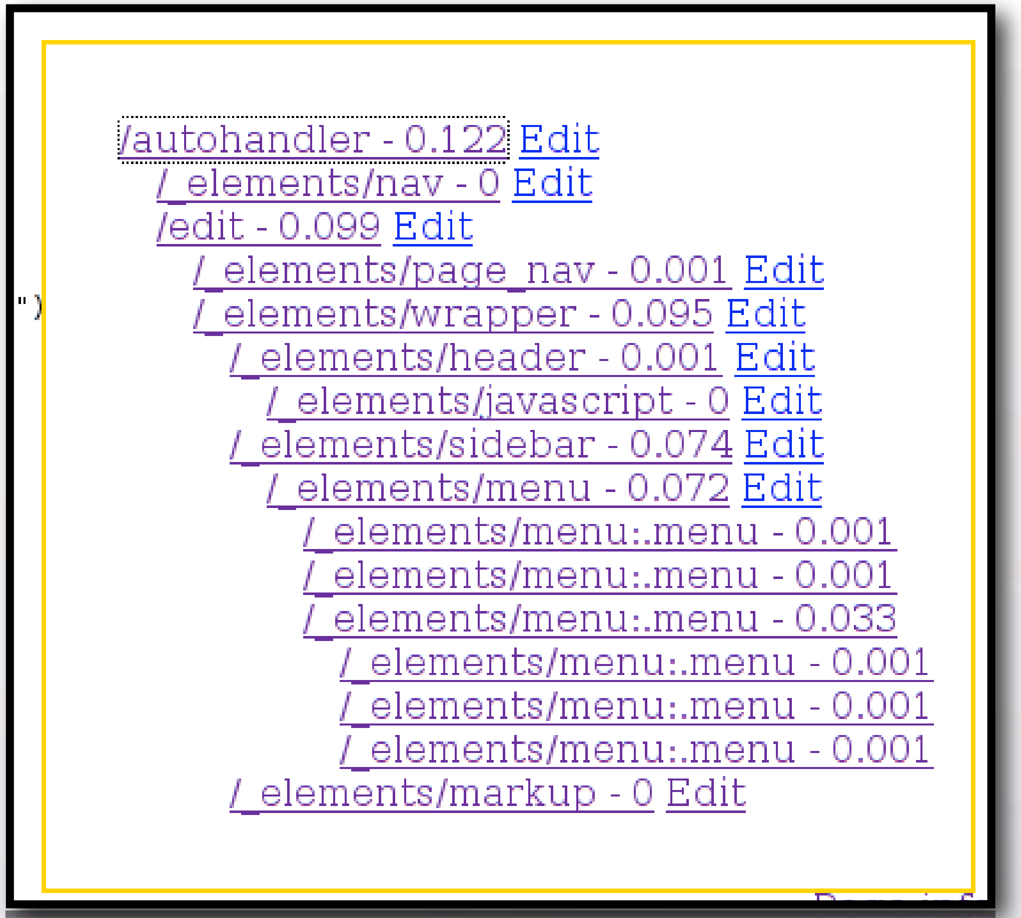
Development Tools

∀ Standalone Server

∀ `Module::Install`

∀ PAR Distribution

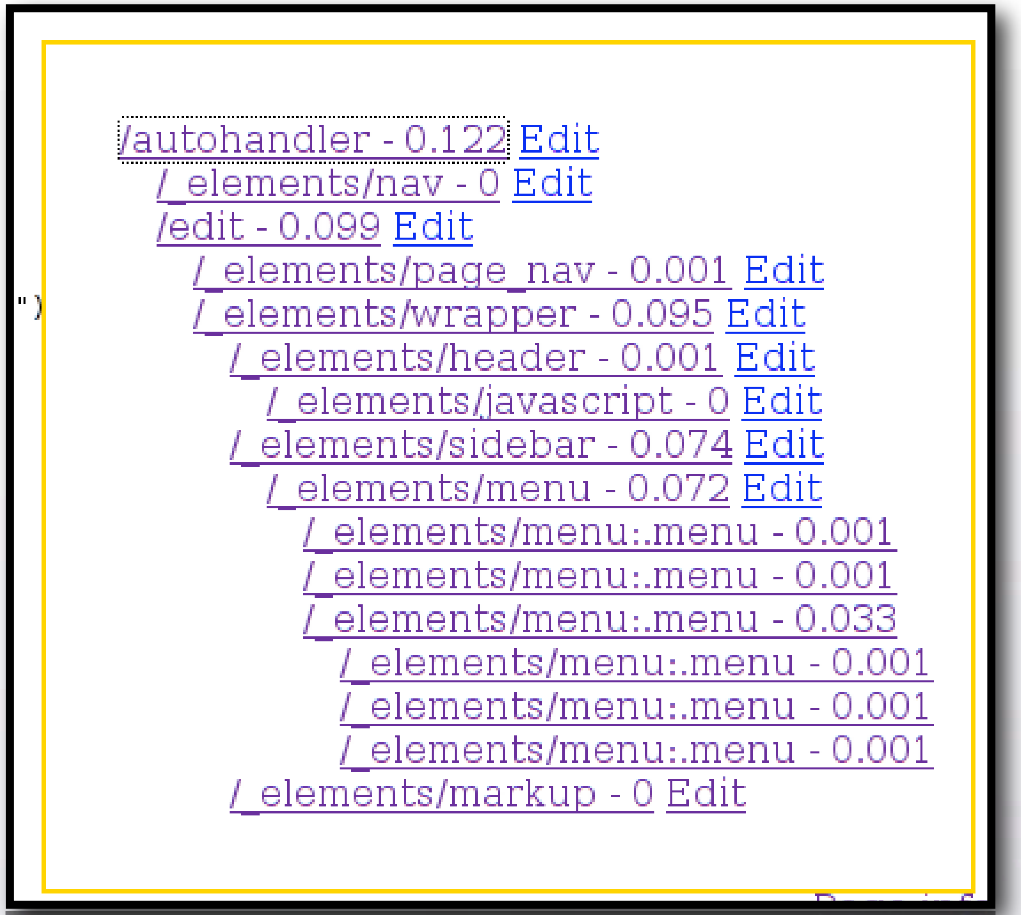
Halos



```
/autohandler - 0.122 Edit
  /_elements/nav - 0 Edit
  /edit - 0.099 Edit
    /_elements/page_nav - 0.001 Edit
    /_elements/wrapper - 0.095 Edit
      /_elements/header - 0.001 Edit
        /_elements/javascript - 0 Edit
      /_elements/sidebar - 0.074 Edit
        /_elements/menu - 0.072 Edit
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.033
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
      /_elements/markup - 0 Edit
```

Halos

∀ **View Tracing**

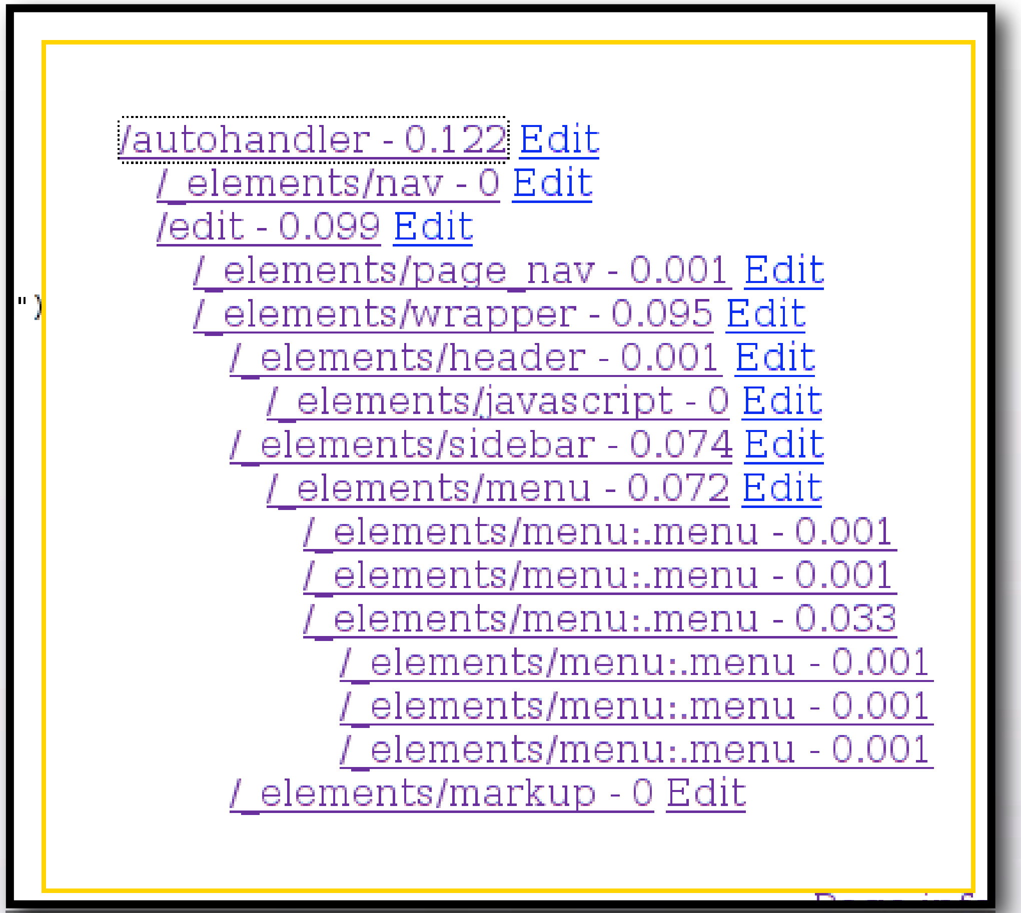


```
/autohandler - 0.122 Edit
  /_elements/nav - 0 Edit
  /edit - 0.099 Edit
    /_elements/page_nav - 0.001 Edit
    /_elements/wrapper - 0.095 Edit
      /_elements/header - 0.001 Edit
        /_elements/javascript - 0 Edit
      /_elements/sidebar - 0.074 Edit
        /_elements/menu - 0.072 Edit
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.033
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
      /_elements/markup - 0 Edit
```

Halos

∀ **View Tracing**

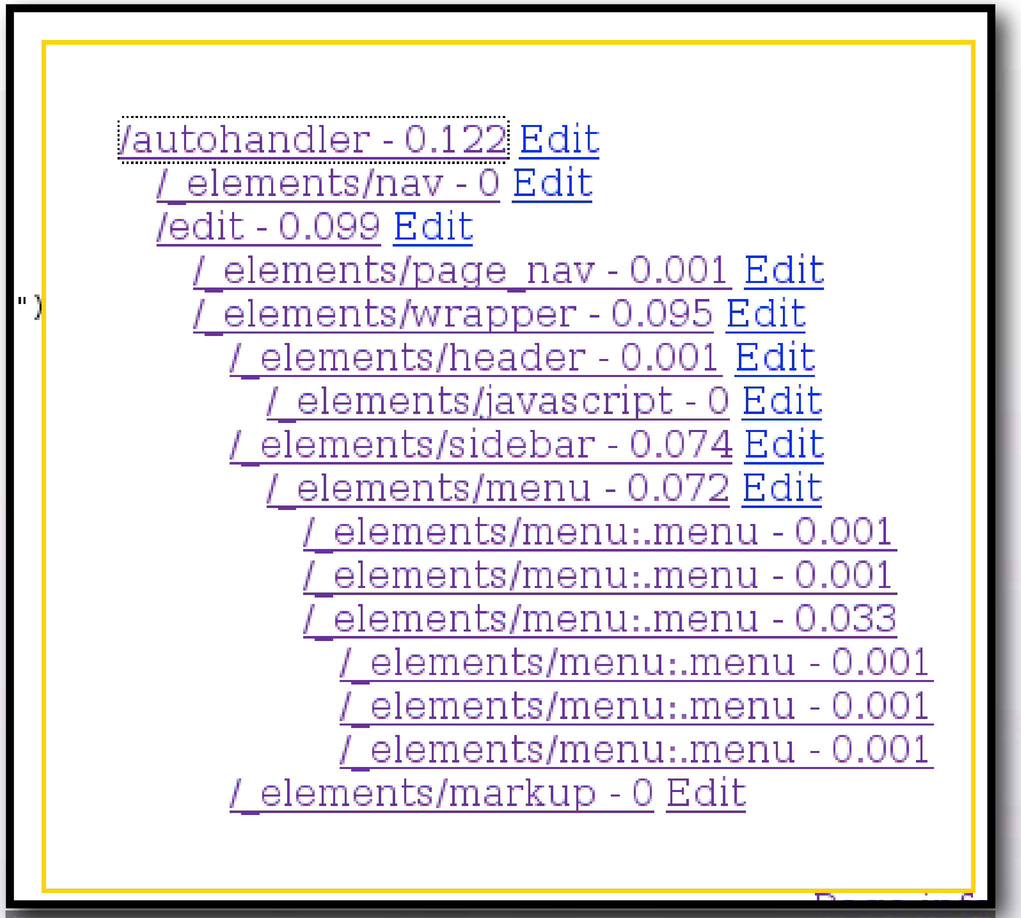
∀ **Online Editing**



```
/autohandler - 0.122 Edit
  /_elements/nav - 0 Edit
  /edit - 0.099 Edit
    /_elements/page_nav - 0.001 Edit
    /_elements/wrapper - 0.095 Edit
      /_elements/header - 0.001 Edit
        /_elements/javascript - 0 Edit
      /_elements/sidebar - 0.074 Edit
        /_elements/menu - 0.072 Edit
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.033
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
      /_elements/markup - 0 Edit
```

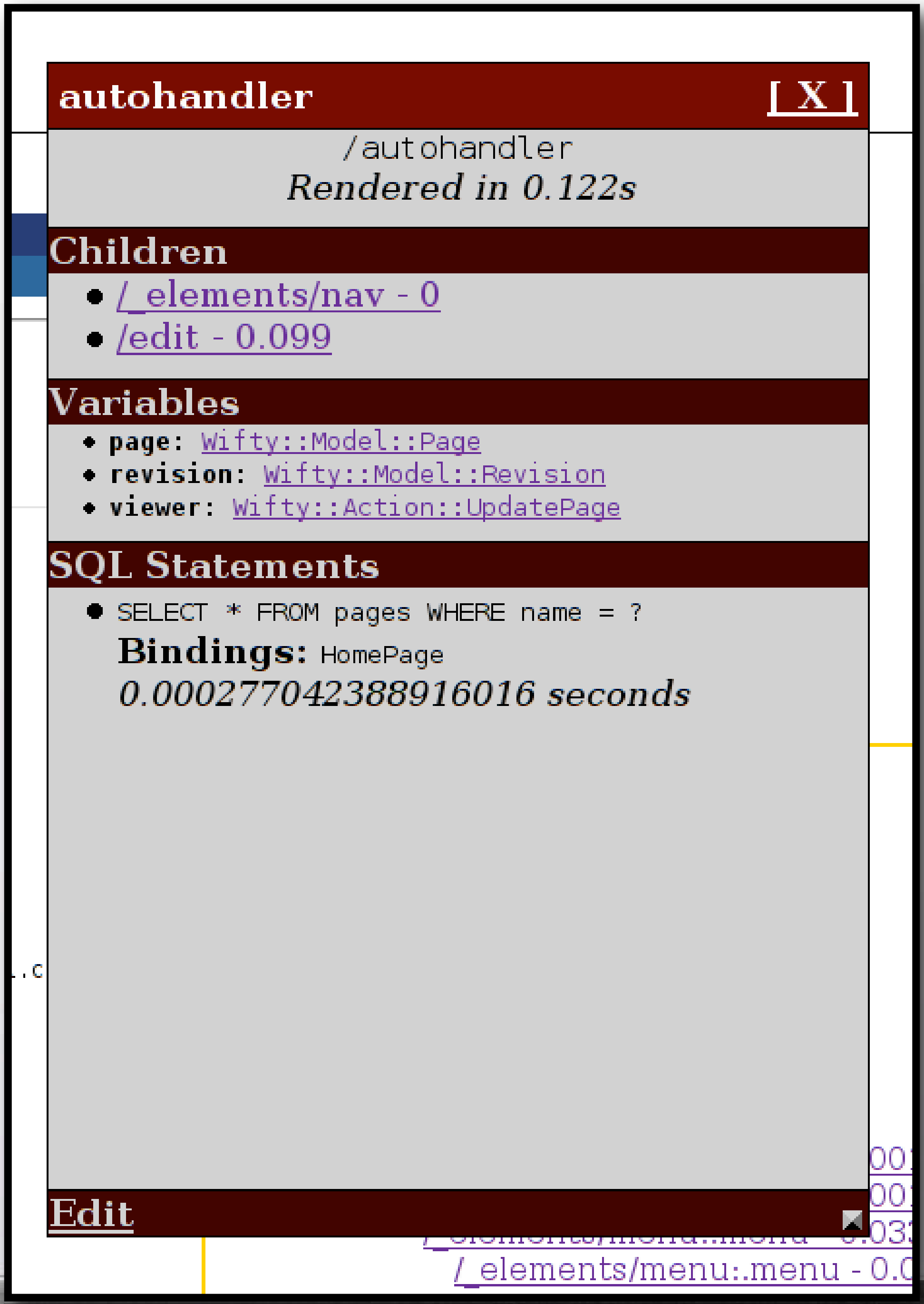
Halos

- ∀ View Tracing
- ∀ Online Editing
- ∀ ...and more!



```
/autohandler - 0.122 Edit
  /_elements/nav - 0 Edit
  /edit - 0.099 Edit
    /_elements/page_nav - 0.001 Edit
    /_elements/wrapper - 0.095 Edit
      /_elements/header - 0.001 Edit
        /_elements/javascript - 0 Edit
      /_elements/sidebar - 0.074 Edit
        /_elements/menu - 0.072 Edit
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.001
          /_elements/menu:.menu - 0.033
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
            /_elements/menu:.menu - 0.001
      /_elements/markup - 0 Edit
```

Halos



autohandler [X]

/autohandler

Rendered in 0.122s

Children

- [/_elements/nav - 0]
- [/edit - 0.099]

Variables

- **page:** Wifty::Model::Page
- **revision:** Wifty::Model::Revision
- **viewer:** Wifty::Action::UpdatePage

SQL Statements

- SELECT * FROM pages WHERE name = ?

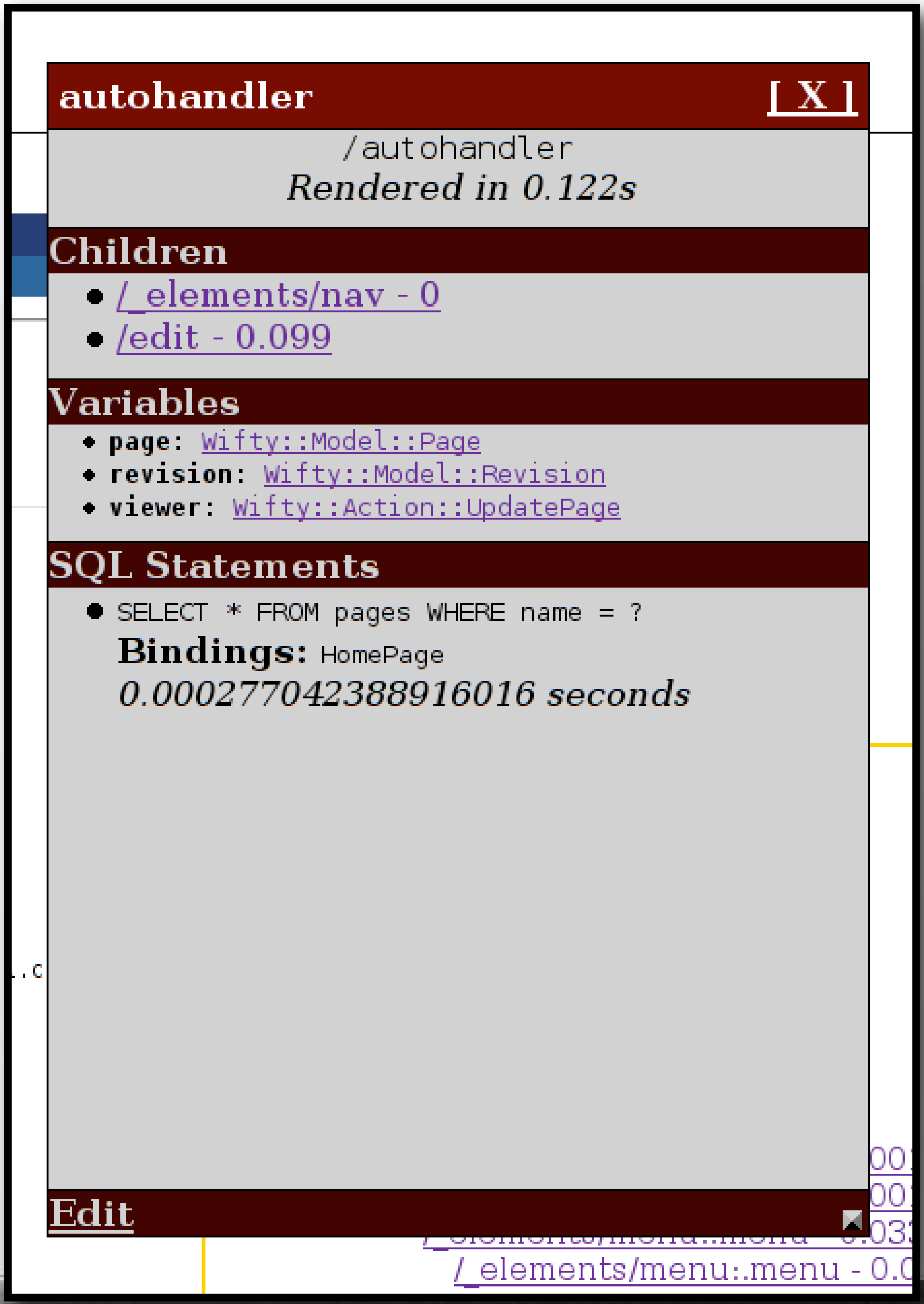
  Bindings: HomePage

  0.000277042388916016 seconds

Edit

Halos

- ∀ SQL Logging



autohandler [X]

/autohandler

Rendered in 0.122s

Children

- /_elements/nav - 0
- /edit - 0.099

Variables

- **page:** Wifty::Model::Page
- **revision:** Wifty::Model::Revision
- **viewer:** Wifty::Action::UpdatePage

SQL Statements

- `SELECT * FROM pages WHERE name = ?`

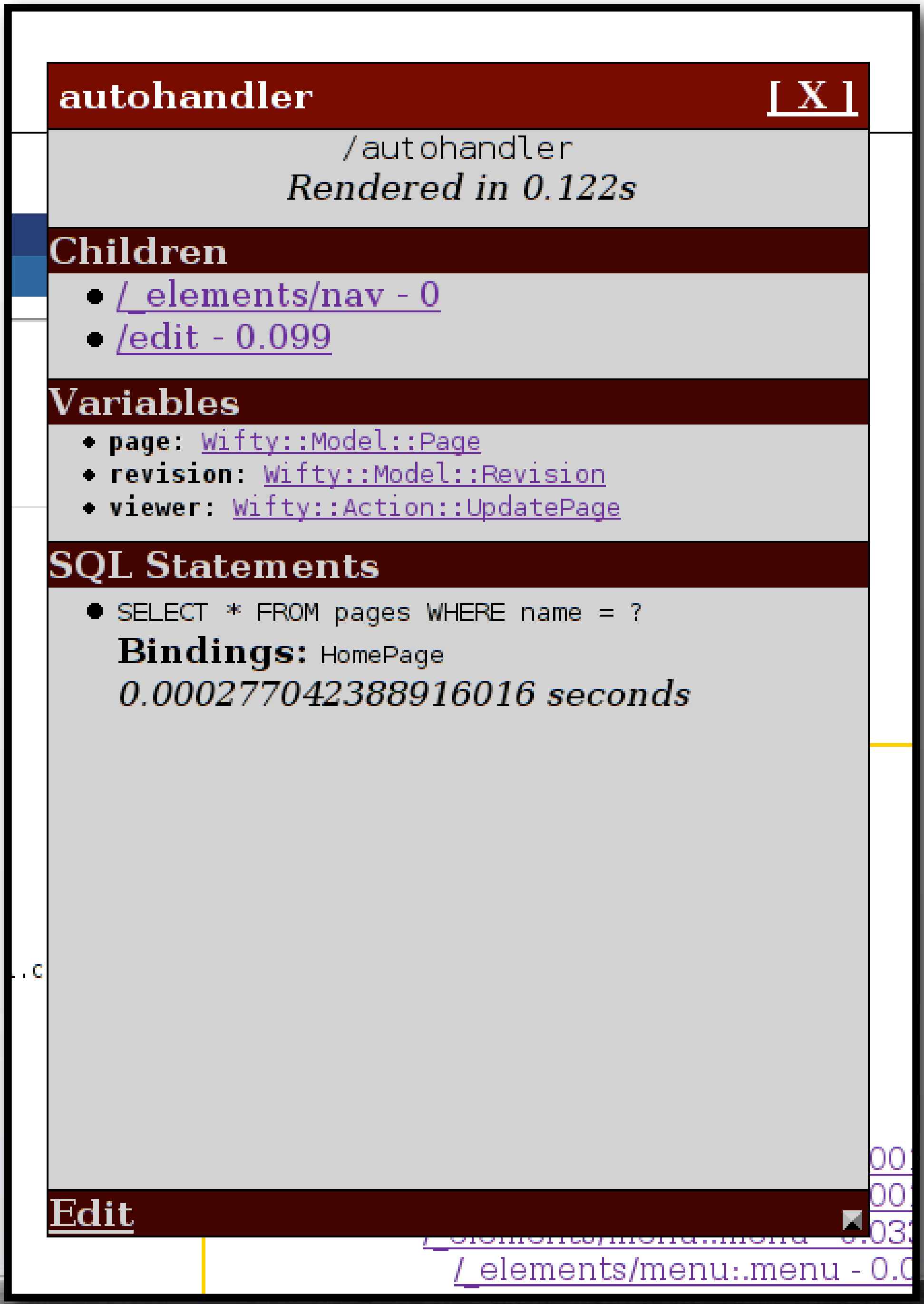
  Bindings: HomePage

  0.000277042388916016 seconds

Edit

Halos

- ∀ SQL Logging
- ∀ SQL Profiling



autohandler [X]

/autohandler
Rendered in 0.122s

Children

- /_elements/nav - 0
- /edit - 0.099

Variables

- **page:** Wifty::Model::Page
- **revision:** Wifty::Model::Revision
- **viewer:** Wifty::Action::UpdatePage

SQL Statements

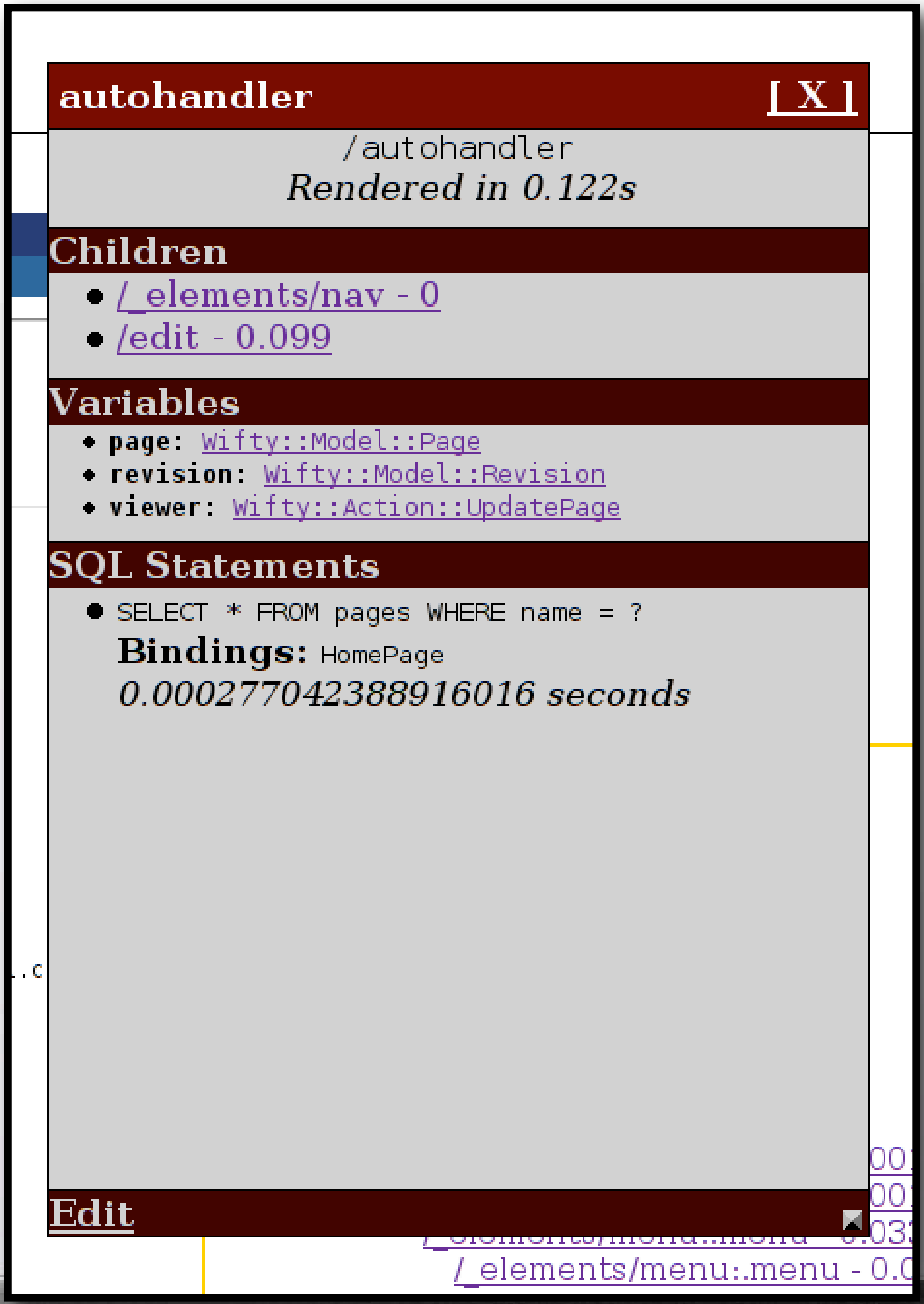
- SELECT * FROM pages WHERE name = ?
  Bindings: HomePage
  0.000277042388916016 seconds

Edit

/_elements/menu::menu - 0.0

Halos

- ∀ SQL Logging
- ∀ SQL Profiling
- ∀ Template Profiling



autohandler [X]

/autohandler
Rendered in 0.122s

Children

- /_elements/nav - 0
- /edit - 0.099

Variables

- **page:** Wifty::Model::Page
- **revision:** Wifty::Model::Revision
- **viewer:** Wifty::Action::UpdatePage

SQL Statements

- SELECT * FROM pages WHERE name = ?
  Bindings: HomePage
  0.000277042388916016 seconds

Edit

/_elements/menu:.menu - 0.0

Devel Mode

Devel Mode

∀ **Module::Refresh**

Devel Mode

∀ Module::Refresh

∀ Tracing with Log4Perl

Devel Mode

∀ Module::Refresh

∀ Tracing with Log4Perl

∀ Administrative Console

Deployment

Deployment

∀ Lighttpd / Apache

Deployment

∀ Lighttpd / Apache

∀ Fast Static Server

Deployment

- ∀ Lighttpd / Apache
- ∀ Fast Static Server
- ∀ CSS::Squish, JS::Squish

More Goodies

More Goodies

∀ Comprehensive I18N

More Goodies

∀ Comprehensive I18N

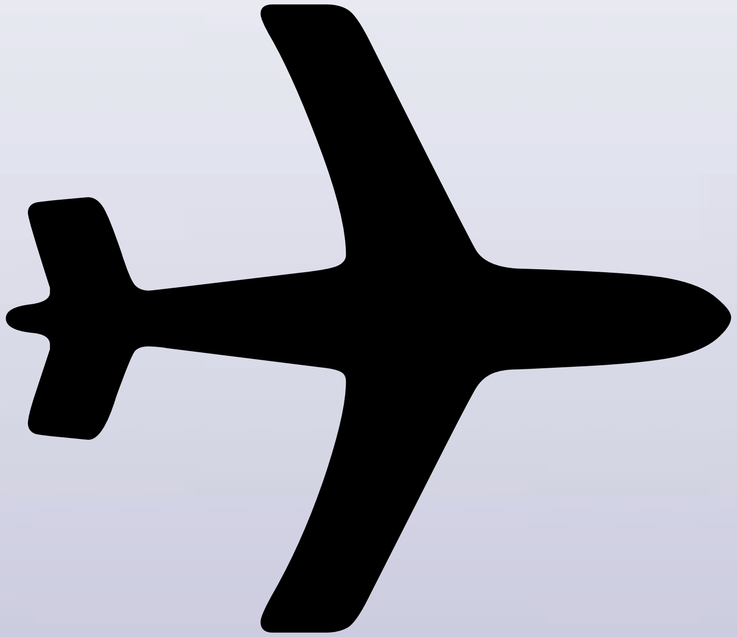
∀ Client-Side Sessions

More Goodies

∀ Comprehensive I18N

∀ Client-Side Sessions

∀ Events and Subscriptions



Future Plans

Future Plans

- Wiki-style UI Editor

Future Plans

- ✈ Wiki-style UI Editor
- ✈ Even less code

Future Plans

- Wiki-style UI Editor
- Even less code
- “Jifty in a Jiffy” MTVs

Sample Applications

Sample Applications

- Wifty

Sample Applications

- ✈ Wifty
- ✈ CommitBit

Sample Applications

- Wifty
- CommitBit
- Babelbee

Sample Applications

- ✈ Wifty
- ✈ CommitBit
- ✈ Babelbee
- ✈ Ping / Chat / Clock

Get Jifty

Get Jifty

- ✈ `cpan Jifty`

Get Jifty

- `cpan Jifty`
- svn.jifty.org/svn/jifty.org

Get Jifty

- `cpan Jifty`
- svn.jifty.org/svn/jifty.org
- http://jifty.org/

Getting Started

Getting Started

- ✈ Jifty::Manual::Tutorial

Getting Started

- ✈ Jifty::Manual::Tutorial
- ✈ irc.freenode.net #jifty

Getting Started

✈ Jifty::Manual::Tutorial

✈ irc.freenode.net #jifty

✈ jifty-devel@lists.jifty.org

That Demo App..

That Demo App..

- ✈ http://fsck.com/~audreyt/DoxoryDemo.tgz

Thank you!

♨

Bonus Slides

☺

Anatomy of a Jifty App

Anatomy of a Jifty App

☺ Model : Columns

Anatomy of a Jifty App

- ☺ Model : Columns
- ☺ Action : Parameters

Anatomy of a Jifty App

☺ Model : Columns

☺ Action : Parameters

☺ Template : Fragments

More Components

More Components

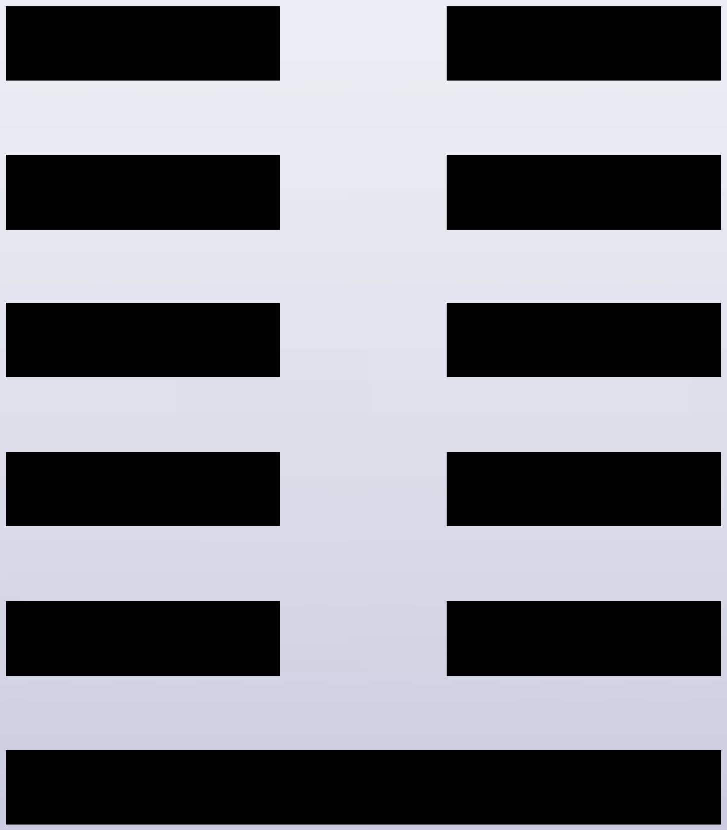
- ☺ Dispatcher

More Components

- ☺ Dispatcher
- ☺ ClassLoader

More Components

- ☺ Dispatcher
- ☺ ClassLoader
- ☺ Continuations



Jifty::DBI

Jifty::DBI

∅ CREATE TABLE

Jifty::DBI

∅ CREATE TABLE

∅ ALTER TABLE

Jifty::DBI

- ∅ CREATE TABLE
- ∅ ALTER TABLE
- ∅ CREATE FOREIGN KEY

Object Relational Mapper

Object Relational Mapper

∅ SELECT

Object Relational Mapper

- ∅ SELECT
- ∅ INSERT

Object Relational Mapper

∅ SELECT

∅ INSERT

∅ UPDATE

Object Relational Mapper

∅ SELECT

∅ UPDATE

∅ INSERT

∅ DELETE

Jifty::DBI::Handle

Jifty::DBI::Handle

- SQLite

Jifty::DBI::Handle

- SQLite
- Postgres

Jifty::DBI::Handle

- SQLite
- Postgres
- MySQL

Jifty::DBI::Handle

- SQLite
- Postgres
- MySQL
- Oracle

Jifty::DBI::Collection

Jifty::DBI::Collection

- Joins and Limits

Jifty::DBI::Collection

- Joins and Limits
- Paging = Data::Page

Jifty::DBI::Collection

- Joins and Limits
- Paging = `Data::Page`
- Caching = `Memcached`

Jifty::DBI::Schema

Jifty::DBI::Schema

- Declarative Columns

Jifty::DBI::Schema

- Declarative Columns
- Remember our Doxory?

Jifty::DBI::Schema

- Declarative Columns
- Remember our Doxory?
- Time for 0.0.2!

```
package Doxory::Model::Comment;
use Jifty::DBI::Schema;
use Doxory::Record schema {

since '0.0.2';
```

```
package Doxory::Model::Comment;
use Jifty::DBI::Schema;
use Doxory::Record schema {

since '0.0.2';

column User   => refers to Doxory::Model:User;
column author => refers to Doxory::Model::User;
```

```
package Doxory::Model::Comment;
use Jifty::DBI::Schema;
use Doxory::Record schema {

since '0.0.2';

column User  => refers to Doxory::Model:User;
column author => refers to Doxory::Model::User;

column body =>
   label   is 'Content',
   type    is 'Text',
   render  as 'Textarea',
   default is 'First Post!';
```

```
package Doxory::Model::Comment;
use Jifty::DBI::Schema;
use Doxory::Record schema {

since '0.0.2';

column User  => refers to Doxory::Model:User;
column author => refers to Doxory::Model::User;

column body =>
   label   is 'Content',
   type    is 'Text',
   render  as 'Textarea',
   default is 'First Post!';

column title =>
   is mandatory,
   label   is 'Title',
   length  is 40,
   hints   is '40 bytes max';
```

```
package Doxory::Model::Comment;
use Jifty::DBI::Schema;
use Doxory::Record schema {

since '0.0.2';

column User  => refers to Doxory::Model:User;
column author => refers to Doxory::Model::User;

column body =>
   label   is 'Content',
   type    is 'Text',
   render  as 'Textarea',
   default is 'First Post!';

column title =>
   is mandatory,
   label   is 'Title',
   length  is 40,
   hints   is '40 bytes max';

};
```

Database Upgrade

Database Upgrade

- Used to be painful

Database Upgrade

- Used to be painful
- Not anymore

Database Upgrade

- Used to be painful
- Not anymore
- Jifty fixed it!

```
$ vim etc/config.yml
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ jifty server
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ jifty server
Application schema version in database (v0.0.1)
doesn't match application schema version (0.0.2)
Please run `bin/jifty schema --setup` to upgrade
the database.
```

```
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ jifty server
Application schema version in database (v0.0.1)
doesn't match application schema version (0.0.2)
Please run `bin/jifty schema --setup` to upgrade
the database.

$ jifty schema --setup
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ jifty server
Application schema version in database (v0.0.1)
doesn't match application schema version (0.0.2)
Please run `bin/jifty schema --setup` to upgrade
the database.

$ jifty schema --setup
Jifty version 0.605070 up to date.
INFO - Generating SQL to upgrade Doxory
       v0.0.1 database to v0.0.2
INFO - Upgraded to version v0.0.2
```

```
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ jifty server
Application schema version in database (v0.0.1)
doesn't match application schema version (0.0.2)
Please run `bin/jifty schema --setup` to upgrade
the database.

$ jifty schema --setup
Jifty version 0.605070 up to date.
INFO - Generating SQL to upgrade Doxory
       v0.0.1 database to v0.0.2
INFO - Upgraded to version v0.0.2

$ jifty server
$ vim etc/config.yml
```

(Set version to 0.0.2)

```
$ jifty server
Application schema version in database (v0.0.1)
doesn't match application schema version (0.0.2)
Please run `bin/jifty schema --setup` to upgrade
the database.

$ jifty schema --setup
Jifty version 0.605070 up to date.
INFO - Generating SQL to upgrade Doxory
       v0.0.1 database to v0.0.2
INFO - Upgraded to version v0.0.2

$ jifty server
INFO - You can connect to your server at
       http://localhost:8888/
```

```
column tags =>
   since '0.0.3',
   refers to Doxory::Model::TagCollection by 'User';
```

```
column tags =>
   since '0.0.3',
   refers to Doxory::Model::TagCollection by 'User';

column mood =>
   since '0.0.4',
   default is 'happy',
   valid are
       { display => 'Sad', value => 'sad' },
       { display => 'Happy', value => 'happy' };
```

```
column tags =>
   since '0.0.3',
   refers to Doxory::Model::TagCollection by 'User';

column mood =>
   since '0.0.4',
   default is 'happy',
   valid are
       { display => 'Sad', value => 'sad' },
       { display => 'Happy', value => 'happy' };

column created =>
   since '0.0.5',
   is immutable,
   type is 'timestamp',
   default is literal 'now()',
   filters are 'Jifty::DBI::Filter::DateTime';
```

```
$ vim etc/config.yml
$ vim etc/config.yml
```

(Set version to 0.0.5)

```
$ vim etc/config.yml
```

(Set version to 0.0.5)

```
$ jifty schema --setup
$ vim etc/config.yml
```

(Set version to 0.0.5)

```
$ jifty schema --setup
Jifty version 0.605070 up to date.
INFO - Generating SQL to upgrade Doxory
       v0.0.2 database to v0.0.5
INFO - Upgraded to version v0.0.5
```

```
$ vim etc/config.yml
```

(Set version to 0.0.5)

```
$ jifty schema --setup
Jifty version 0.605070 up to date.
INFO - Generating SQL to upgrade Doxory
       v0.0.2 database to v0.0.5
INFO - Upgraded to version v0.0.5

$ jifty server
$ vim etc/config.yml
```

(Set version to 0.0.5)

```
$ jifty schema --setup
Jifty version 0.605070 up to date.
INFO - Generating SQL to upgrade Doxory
       v0.0.2 database to v0.0.5
INFO - Upgraded to version v0.0.5

$ jifty server
INFO - You can connect to your server at
       http://localhost:8888/
```

$$\Rrightarrow$$

Action = APIs

Action = APIs

⭆ **Method Calls**

Action = APIs

- ⇛ Method Calls
- ⇛ Typed named arguments

Action = APIs

⇛ Method Calls

⇛ Typed named arguments

⇛ Returns results or errors

Action ≠ URLs

Action ≠ URLs

⇛ **Many actions on one page**

Action ≠ URLs

- ⇛ Many actions on one page
- ⇛ Called with web forms

Action ≠ URLs

- ⇛ Many actions on one page
- ⇛ Called with web forms
- ⇛ Sounds like RPC?

Action = RPC

Action = RPC

⇛ **Web form encoding**

Action = RPC

- ⇛ Web form encoding
- ⇛ XML/YAML encoding

Action = RPC

⇛ Web form encoding

⇛ XML/YAML encoding

⇛ Web services for free!

Model ⇛ Action

Model ⇛ Action

⇛ Doxory::Action::CreateUser

Model ⇛ Action

- ⇛ Doxory::Action::CreateUser
- ⇛ Doxory::Action::SearchUser

Model ⇛ Action

⇛ Doxory::Action::CreateUser

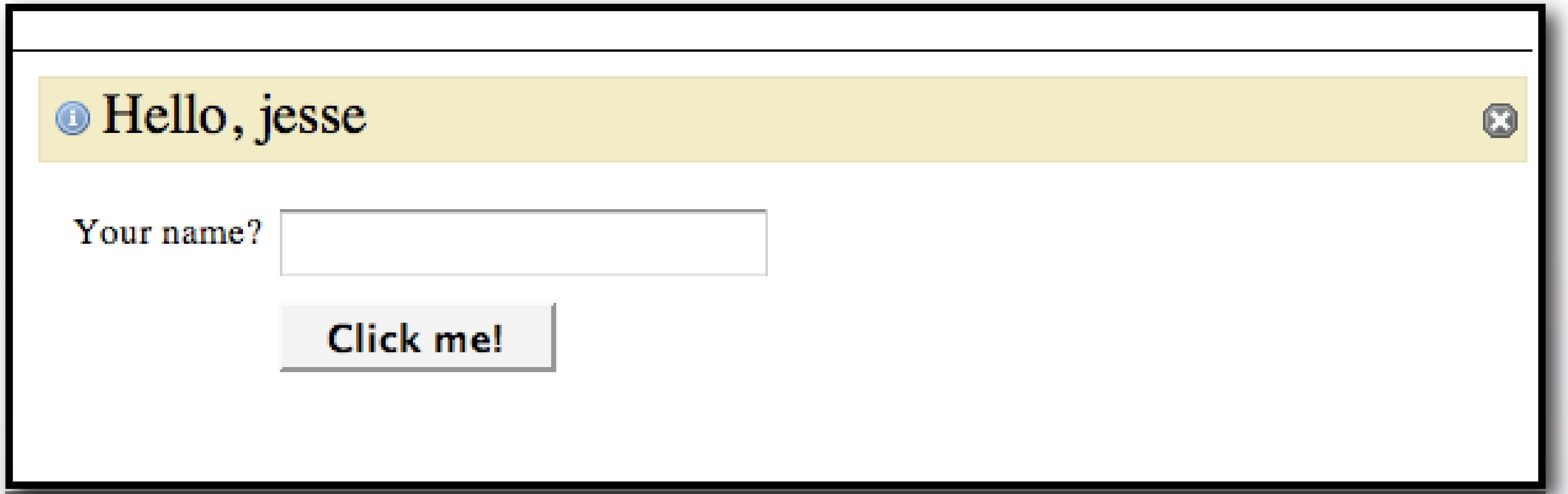
⇛ Doxory::Action::SearchUser

⇛ Doxory::Action::UpdateUser

Model ⇛ Action

- ⇛ Doxory::Action::CreateUser
- ⇛ Doxory::Action::SearchUser
- ⇛ Doxory::Action::UpdateUser
- ⇛ Doxory::Action::DeleteUser

Make your own Actions

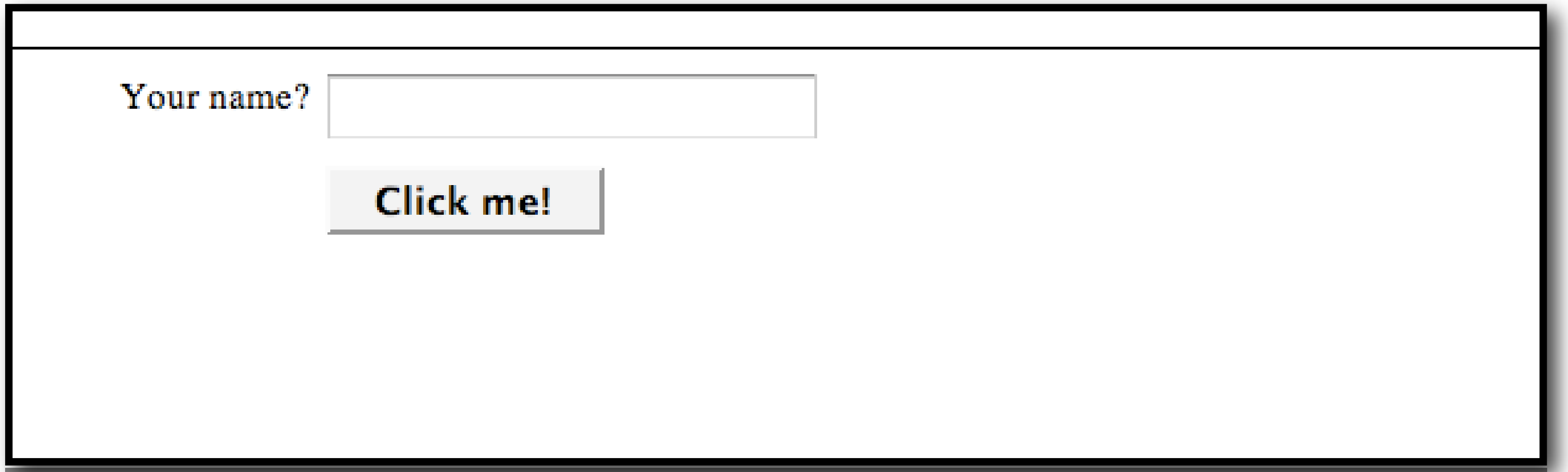


Hello, jesse

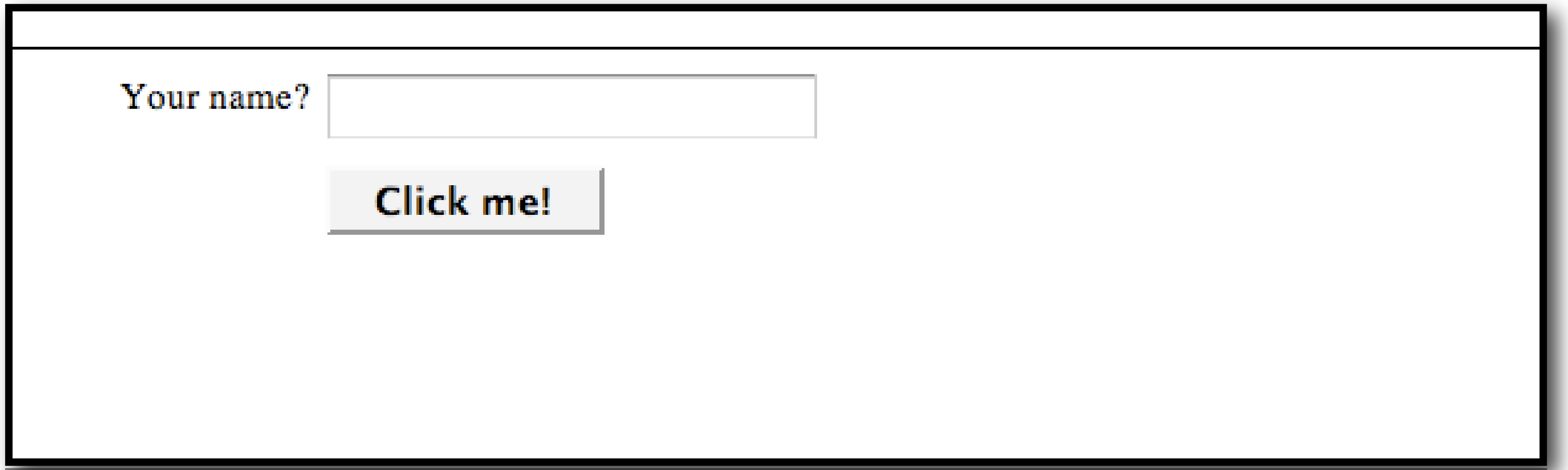
Your name?

```
package Hello::Action::SayHi;
use Jifty::Param::Schema;
use Jifty::Action schema {

param name =>
   label is 'Your name?';

};
```

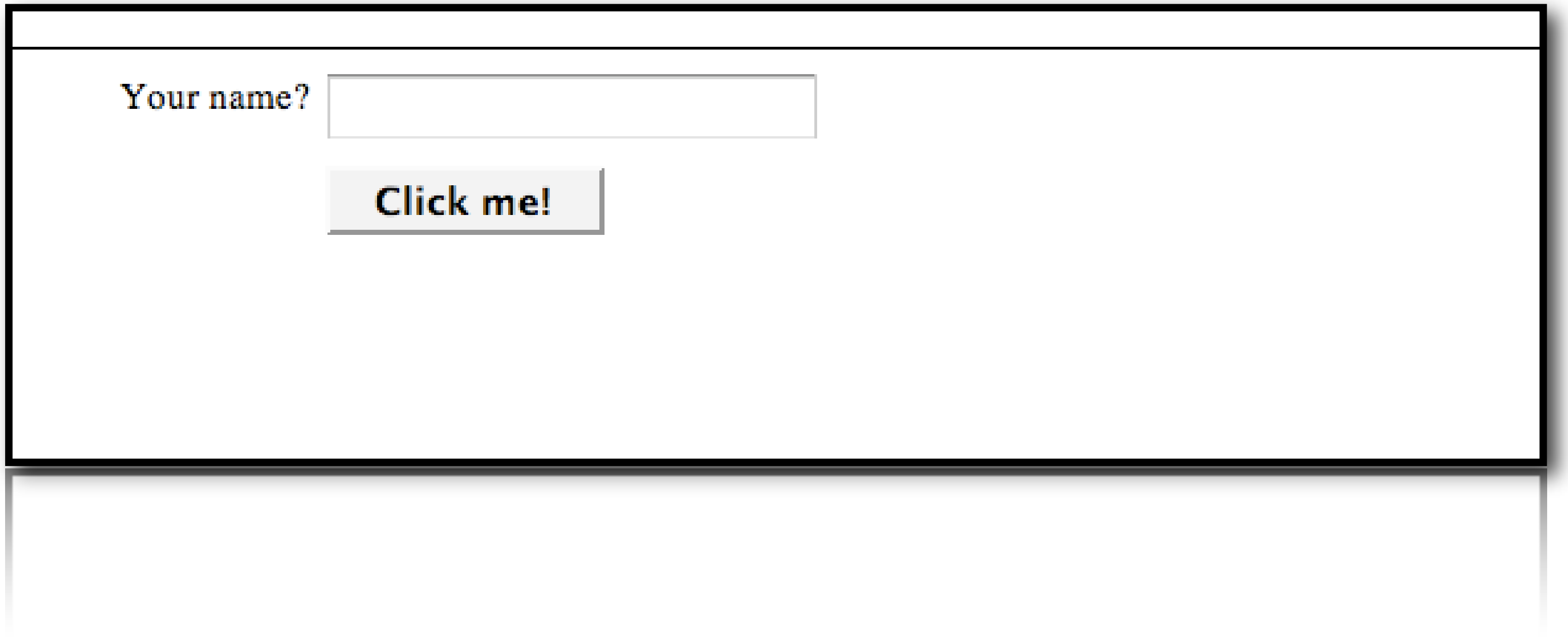
```
package Hello::Action::SayHi;
use Jifty::Param::Schema;
use Jifty::Action schema {

param name =>
   label is 'Your name?';

};

sub take_action {
   my $self = shift;
   my $name = $self->argument_value('name');
   $self->result->message( "Hello, $name" );
}
```


Your name?
Click me!

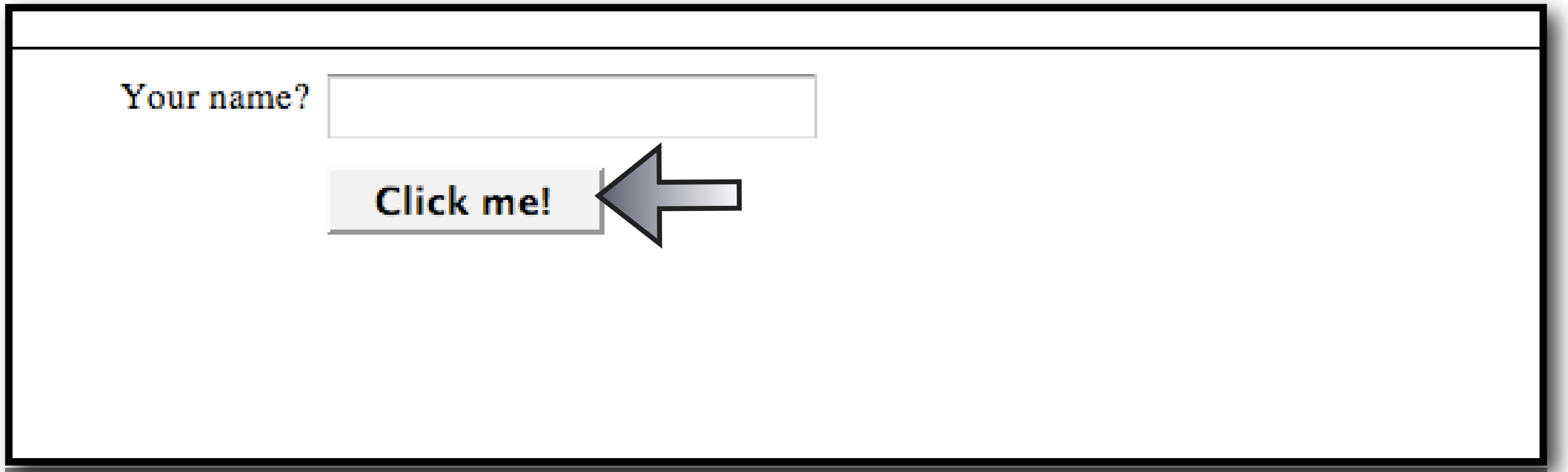
```
<%init>
my $action = Jifty->web->new_action(
    class => 'SayHi'
);
</%init>
```



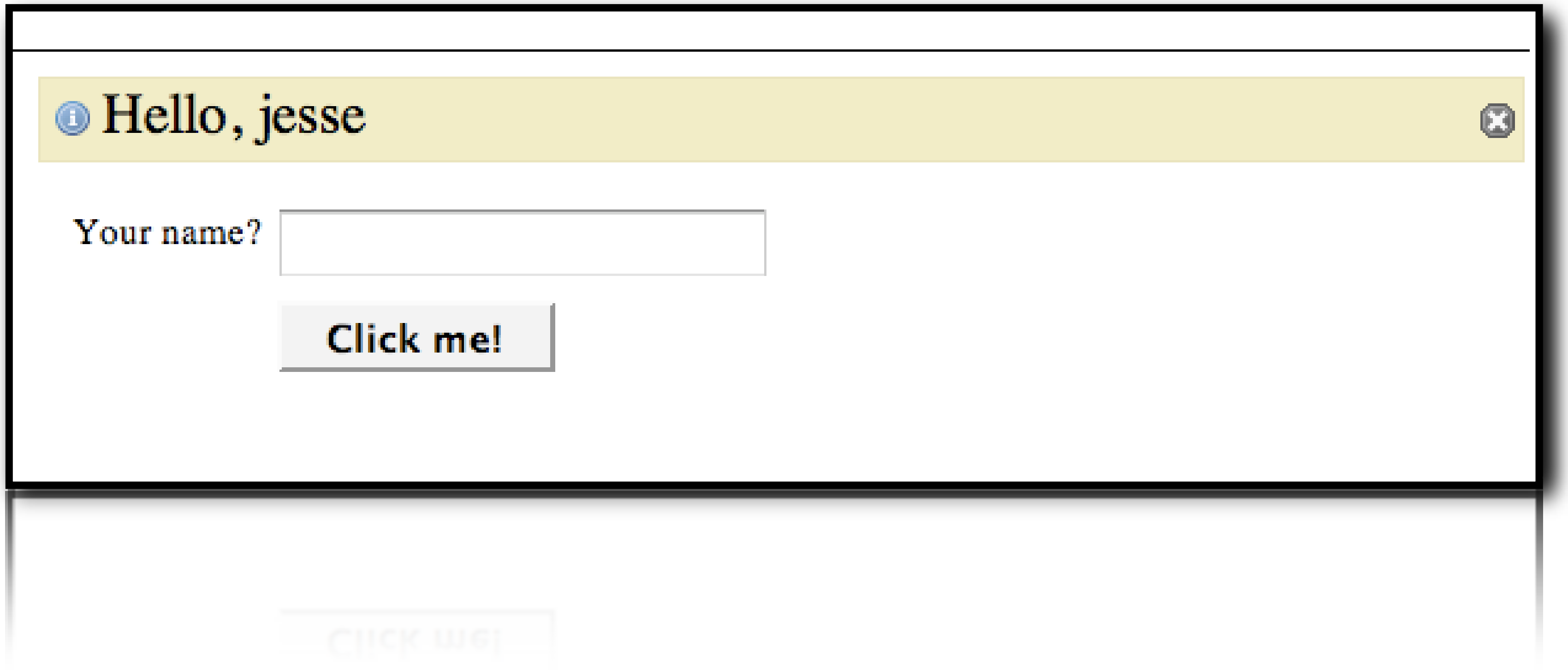
```
<%init>
my $action = Jifty->web->new_action(
    class => 'SayHi'
);
</%init>

<&|/_elements/wrapper&>
<% Jifty->web->form->start %>
<% $action->form_field('name') %>
<% Jifty->web->form->submit %>
<% Jifty->web->form->end %>
</&>
```



```
<%init>
my $action = Jifty->web->new_action(
    class => 'SayHi'
);
</%init>

<&|/_elements/wrapper&>
<% Jifty->web->form->start %>
<% $action->form_field('name') %>
<% Jifty->web->form->submit %>
<% Jifty->web->form->end %>
</&>
```



```
<%init>
my $action = Jifty->web->new_action(
    class => 'SayHi'
);
</%init>

<&|/_elements/wrapper&>
<% Jifty->web->form->start %>
<% $action->form_field('name') %>
<% Jifty->web->form->submit %>
<% Jifty->web->form->end %>
</&>
```



Hello, jesse

Your name?

```
$ curl \
  http://localhost:8888/__jifty/webservices/yaml?\
  J:A-1=SayHi;J:A:F-name-1=Jesse
```

```
$ curl \
  http://localhost:8888/__jifty/webservices/yaml?\
  J:A-1=SayHi;J:A:F-name-1=Jesse

1: !!perl/hash:Jifty::Result
  _content: {}
  action_class: Hello::Action::SayHi
  failure: 0
  field_errors:
    name: ~
  field_warnings:
    name: ~
  message: 'Hello, Jesse'
```

Jifty::Dispatcher

Dispatcher = Kernel

Dispatcher = Kernel

- Simple

Dispatcher = Kernel

- Simple
- Declarative

Dispatcher = Kernel

- Simple
- Declarative
- Encapsulates view logic

Three Stages

Three Stages

- **Before: Sets up state**

Three Stages

- Before: Sets up state
- On: Shows your pages

Three Stages

- Before: Sets up state
- On: Shows your pages
- After: Cleans up state

Jifty::ClassLoader

Scaffolding Bad

Scaffolding Bad

- Ugly

Scaffolding Bad

- Ugly
- Redundant

Scaffolding Bad

- Ugly
- Redundant
- Fragile

Scaffolding Bad

- Ugly
- Redundant
- Fragile
- Annoying

Scaffolding Bad

- Ugly
- Redundant
- Fragile

- Annoying
- Redundant

Scaffolding Bad

- Ugly
- Redundant
- Fragile
- Annoying
- Redundant
- Redundant

Why *Empty* Classes?

Why *Empty* Classes?

- Jifty::ClassLoader

Why *Empty* Classes?

- Jifty::ClassLoader
- Auto-creates classes

Why *Empty* Classes?

- Jifty::ClassLoader
- Auto-creates classes
- Never writes them to disk

Auto-created Classes

Auto-created Classes

- Doxory

Auto-created Classes

- Doxory
- Doxory::Record

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser
- Doxory::Model::UserCollection

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser
- Doxory::Model::UserCollection
- Doxory::Action::CreateUser

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser
- Doxory::Model::UserCollection
- Doxory::Action::CreateUser
- Doxory::Action::SearchUser

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser
- Doxory::Model::UserCollection
- Doxory::Action::CreateUser
- Doxory::Action::SearchUser
- Doxory::Action::UpdateUser

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser
- Doxory::Model::UserCollection
- Doxory::Action::CreateUser
- Doxory::Action::SearchUser
- Doxory::Action::UpdateUser
- Doxory::Action::DeleteUser

Auto-created Classes

- Doxory
- Doxory::Record
- Doxory::Collection
- Doxory::Bootstrap
- Doxory::Upgrade
- Doxory::Dispatcher
- Doxory::CurrentUser
- Doxory::Model::UserCollection
- Doxory::Action::CreateUser
- Doxory::Action::SearchUser
- Doxory::Action::UpdateUser
- Doxory::Action::DeleteUser
- *...et cetera*

Continuation = Job Control

Continuation = Job Control

- ☯ Continuations in Perl?

Continuation = Job Control

- Continuations in Perl?
- Not Full Continuations

Continuation = Job Control

- Continuations in Perl?
- Not Full Continuations
- Delimited Continuations

Full Continuations

Full Continuations

- Serializes everything

Full Continuations

☯ Serializes everything

☯ %ENV, STDIN, STDOUT

Full Continuations

- Serializes everything
- %ENV, STDIN, STDOUT
- DBI Handle, Log Handle

Full Continuations

- Serializes everything
- %ENV, STDIN, STDOUT
- DBI Handle, Log Handle
- Phase of the moon

Jifty Continuations

Jifty Continuations

- ☯ Only the application state

Jifty Continuations

- Only the application state
- Requests and Actions

Jifty Continuations

- Only the application state
- Requests and Actions
- That’s all we need

Why Continuations?

Why Continuations?

- The web is built on GOTO

Why Continuations?

- The web is built on GOTO
- Do you *like* GOTO?

Why Continuations?

- The web is built on GOTO
- Do you *like* GOTO?
- Continuations are GOSUB

Why Continuations?

- The web is built on GOTO
- Do you *like* GOTO?
- Continuations are GOSUB
- Welcome to Civilization!

A Brief Tangent...

A Brief Tangent...

- Back to our Doxory

A Brief Tangent...

- ☯ Back to our Doxory
- ☯ Requiring Login

A Brief Tangent...

- Back to our Doxory
- Requiring Login
- `lib/Doxory/Dispatcher.pm`

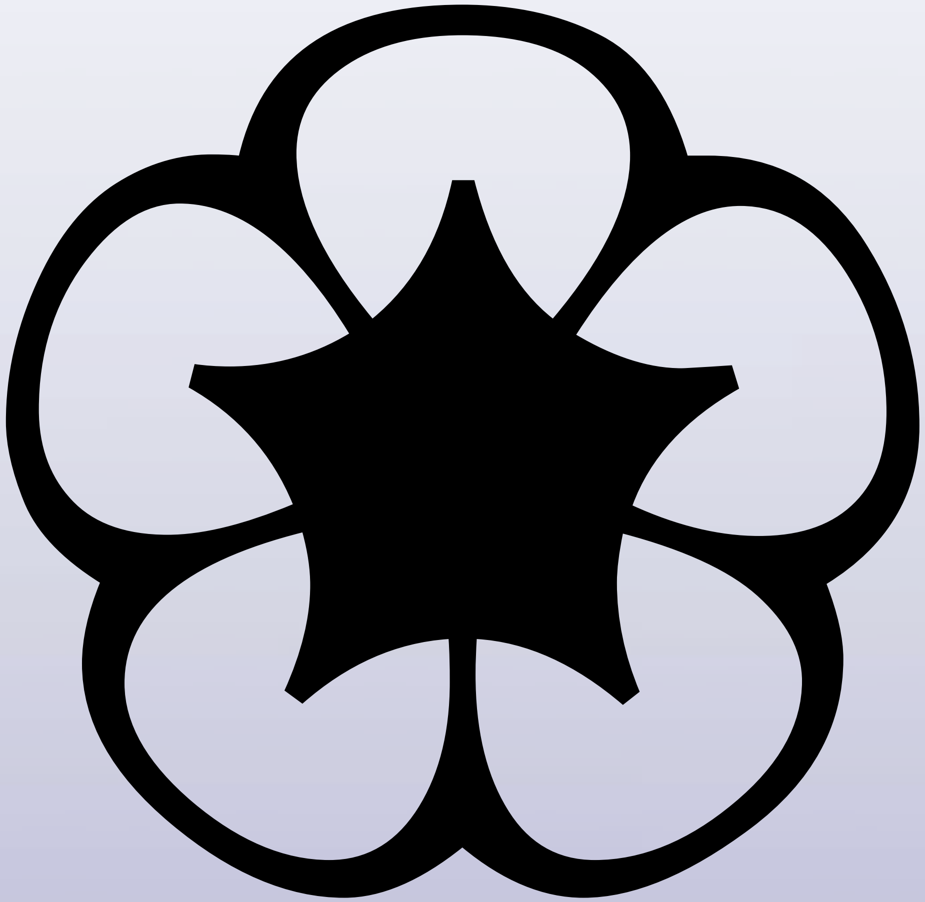
```
package Doxory::Action::Login;
use Jifty::Param::Schema;
use Jifty::Action schema {

param address =>
   is mandatory,
   ajax validates,
   label is 'Email address';

param password =>
   is mandatory,
   type  is 'Password',
   label is 'Password';

};
```

```
package Doxory::Action::Login;
use Jifty::Param::Schema;
use Jifty::Action schema {

param address =>
   is mandatory,
   ajax validates,
   label is 'Email address';

param password =>
   is mandatory,
   type  is 'Password',
   label is 'Password';

};

sub take_action {
   my $self = shift;
   if ( $self->validate_login ) {
       $self->login_as_user( $self->user );
       return 1;
   }
   return 0;
}
```

```
package Doxory::Dispatcher;

## Auth and login
before '*' => run {
   if (not Jifty->web->current_user->id ) {
       tangent '/login';
   }
};
```

```
%# share/web/templates/login
<%init>
my $action = Jifty->web->new_action(
    class   => 'Login',
);
</%init>
```

```
%# share/web/templates/login
<%init>
my $action = Jifty->web->new_action(
    class   => 'Login',
);
</%init>

<&|/_elements/wrapper, title => 'Login' &>
<h2>Login</h2>
   <% Jifty->web->form->start %>
       <% $action->form_field('address') %>
       <% $action->form_field('password') %>
       <% Jifty->web->return( label => 'Login') %>
   <% Jifty->web->form->end %>
</&>
```



Templates = View

Templates = View

- HTML::Mason

Templates = View

- HTML::Mason
- Mason has Components

Templates = View

- HTML::Mason
- Mason has Components
- Server-side inclusion

Fragments

Fragments

- Components 2.0

Fragments

- Components 2.0
- Client-side for AJAX

Fragments

- Components 2.0
- Client-side for AJAX
- Server-side for Lynx

```
<% Jifty->web->region(
   name => "myweDoxory-posts",
   path => "/fragments/page_of_posts"
) %>
```

Chat

Chat

- Server-push (Comet)

Chat

- Server-push (Comet)
- Refreshed fragments

Chat

- Server-push (Comet)
- Refreshed fragments
- Live Demo!

Thank you!

♨