

ByteNote - ByteMe

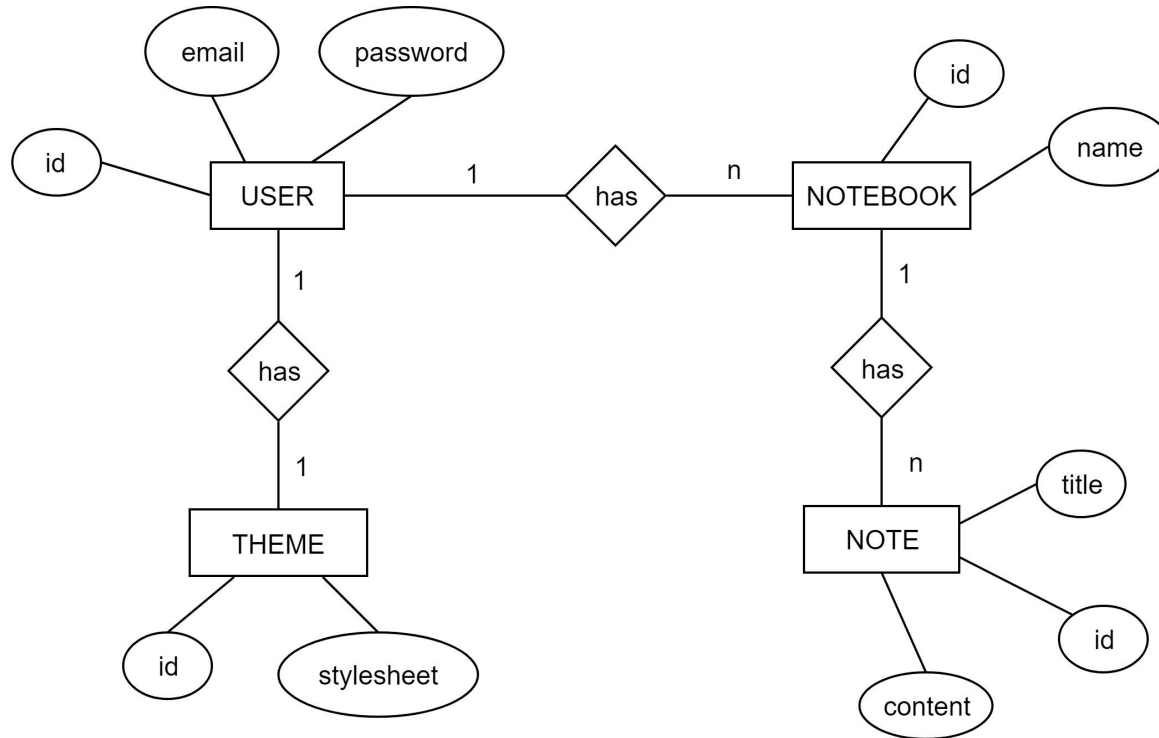
**Carter Brown, Ally Chitwood, Will Greenway,
Dan Greer, Nathan Simpson**

Introduction

- What?
 - ▷ Cloud-based note taking
- Why?
 - ▷ Simple and streamlined
 - ▷ Multiple platforms



Entity Relationship - Database Diagram



REST Diagram

REpresentation State Transfer

Nathan Simpson



Welcome to ByteNote!

Returning user?

[Log In](#)

New user?

[Sign Up](#)

Join ByteNote

Email

Password (6 characters minimum)

Password confirmation

[Sign up](#)

[Already a user?](#)

[Log in](#)

[Didn't receive confirmation instructions?](#)

Log in

Email

Password

☐ Remember me

[Log in](#)

[Forgot your password?](#)

New User?

[Sign up](#)

[Didn't receive confirmation instructions?](#)

Profile Notebooks Games Logout

Notebooks

[New Notebook](#)

Profile Notebooks Games Logout

New Notebook

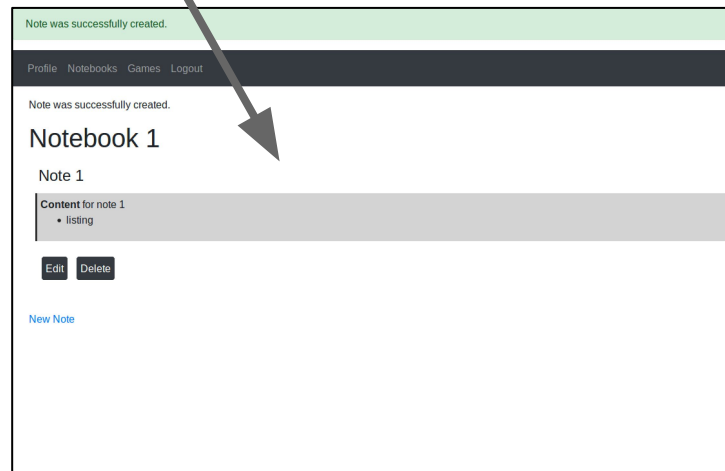
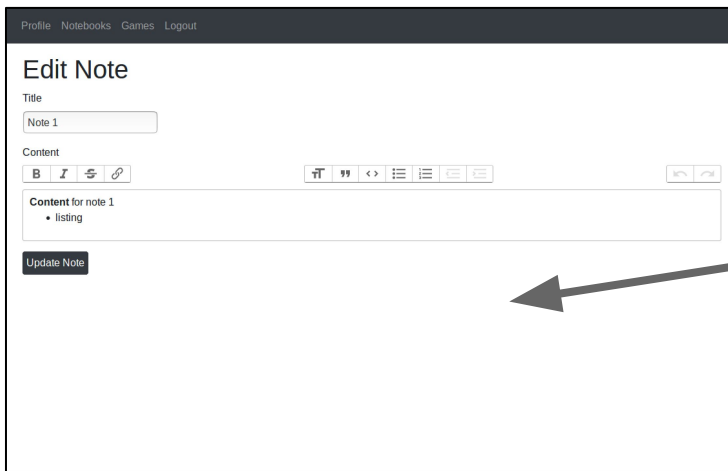
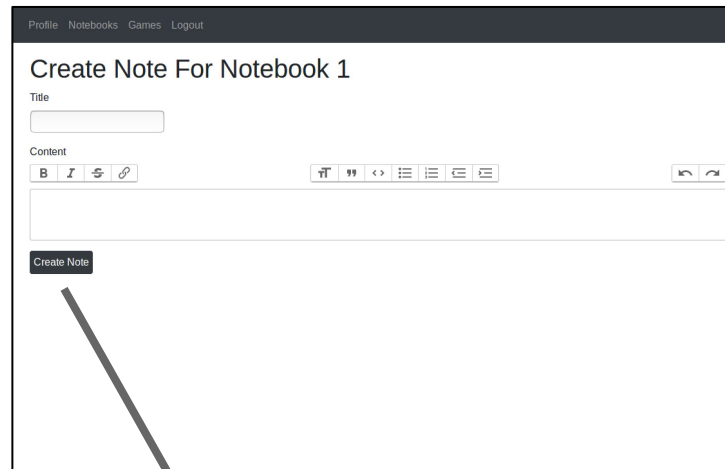
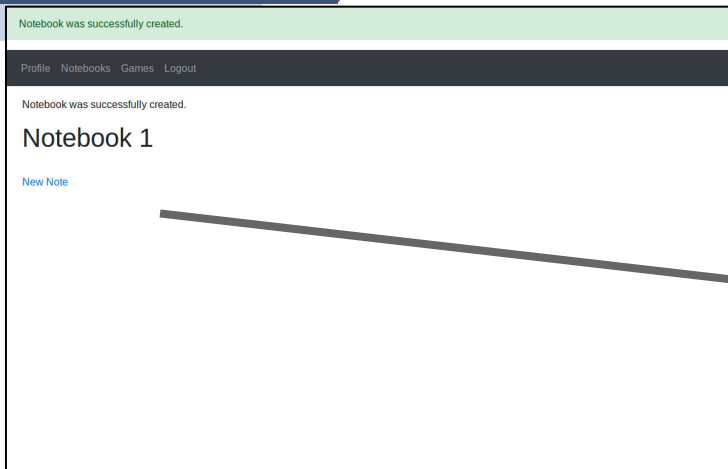
Name

[Create Notebook](#)

[Back](#)



Nathan Simpson



Routes



Routes

```
Rails.application.routes.draw do
  get '/games', to: 'games#index'

  get '/themes', to: 'themes#index'
  get '/set_stylesheet', to: 'themes#index'

  devise_for :users
  resources :notebooks
  resources :notes, :except => [:index]
  resources :greetings, :only => [:index]
  root 'greetings#index'
end
```




Notebooks Controller

```
def index
  @notebooks = current_user.notebooks
end

def create
  @notebook = Notebook.new(notebook_params)
  @notebook.user_id = current_user.id
  respond_to do |format|
    if @notebook.save
      format.html { redirect_to @notebook, notice: 'Notebook was successfully created.' }
    else
      format.html { render :new }
    end
  end
end

def destroy
  @notebook.notes.destroy_all
  @notebook.destroy
  respond_to do |format|
    format.html { redirect_to notebooks_url, notice: 'Notebook was successfully destroyed.' }
  end
end
```



Notes Controller

```
def new
  @note = Note.new
  @notebook = Notebook.find(params[:notebook_id])
end

def create
  @notebook = Notebook.find(params[:notebook_id])
  @note = Note.new(note_params)
  @note.notebook_id = @notebook.id
  respond_to do |format|
    if @note.save
      format.html { redirect_to @notebook, notice: 'Note was successfully created.' }
    else
      format.html { render :new }
    end
  end
end
```



Notes Controller Table

| HTTP Verb | Path | Controller#Action | Purpose |
|-----------|-----------------|-------------------|---|
| GET | /notes | notes#index | display all notes in current notebook |
| POST | /notes | notes#create | create a new note |
| GET | /notes/new | notes#new | return an HTML form to create a new note |
| GET | /notes/:id | notes#show | display a specific note |
| GET | /notes/:id/edit | notes#edit | return an HTML form for editing a specific note |
| PUT | /notes/:id | notes#update | update a specific note |
| DELETE | /notes/:id | notes#destroy | destroy a specific note |



Themes Controller

```
class ThemesController < ApplicationController
  before_action :authenticate_user!
  before_action :set_stylesheet

  def index
  end

  def show
    @stylesheet = User.find(params[:id])

    respond_to(:html, :css) if stale?(@stylesheet)
  end

  def set_stylesheet
    @stylesheet = params[:id]
  end
end
```

devise

Trix

B Bootstrap



Rspec

Technologies



Bootstrap

```
<h1>Notebooks</h1>
<hr />
<div class="container-fluid">
  <% @notebooks.each with_index do |notebook, i| %>
    <div class="row">
      <div class="col-3">
        <h2 class="display_object nb_name"><%= notebook.name %></h2>
      </div>
      <div class="col">
        <%= link_to 'Notes', notebook, class: "btn btn-dark spaced" %>
        <%= link_to 'Edit', edit_notebook_path(notebook), class: "btn btn-dark spaced" %>
        <%= link_to 'Delete', notebook, method: :delete, data: { confirm: 'Are you sure?' }, class:
"btn btn-dark spaced" %>
      </div>
    </div>
    <% if i < @notebooks.size-1 %>
      <hr />
    <% end %>
  <% end %>
</div>
<br />
<%= link_to 'New Notebook', new_notebook_path, class: "btn btn-dark" %>
```



Bootstrap

```
<nav class="navbar navbar-expand-sm bg-dark navbar-dark">
  <ul class="navbar-nav">
    <li class="nav-item">
      <%= link_to "Profile", edit_user_registration_path, :class => 'nav-link' %>
    </li>
    <li class="nav-item">
      <%= link_to "Notebooks", notebooks_path, :class => 'nav-link' %>
    </li>
    <li class="nav-item">
      <%= link_to "Games", games_path, :class => 'nav-link' %>
    </li>
    <li class="nav-item">
      <%= link_to "Logout", destroy_user_session_path, method: :delete, :class => 'nav-link' %>
    </li>
  </ul>
</nav>
```



Devise

- Log in
- Sign up
- Log out
- Extra verification



Welcome Page

- Routing issues
- Rooting issues
- How do we redirect Devise logins to the correct page?



Greetings Controller

```
1 class GreetingsController < ApplicationController
2
3   # GET /greetings
4   def index
5   end
6
7 end
```



New Welcome Page Routing

```
8 resources :greetings, :only => [:index]  
9 root 'greetings#index'
```



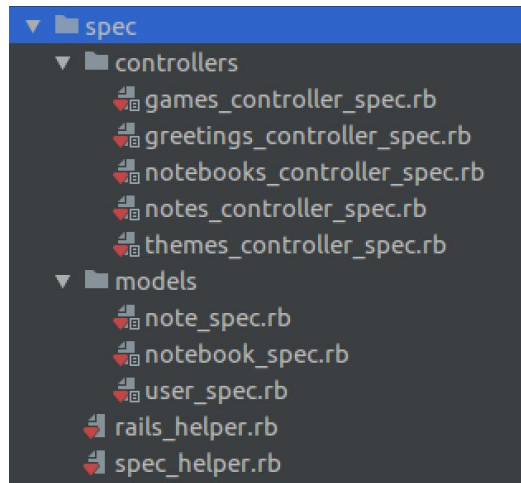
Welcome Page Redirecting

```
1 <% if user_signed_in? %>
2   <!-- Redirect -->
3   <% controller.redirect_to notebooks_path %>
4 <% end %>
5
6 <div class="jumbotron jumbotron-fluid">
7   <div class="container">
8     <h1> Welcome to Bytenote! </h1>
9   </div>
10 </div>
11 <br/>
12
13 <h2> Returning user? </h2>
14 <%= link_to('Log In', new_user_session_path, class: "btn btn-dark") %>
15 <br/>
16
17 <h2> New user? </h2>
18 <%= link_to('Sign Up', new_user_registration_path, class: "btn btn-dark") %>
```



Rspec

- Testing keywords
 - ▷ describe
 - ▷ it
 - ▷ expect



```
allychitwood@allychitwood-VirtualBox:~/byteme-Project-6/project_6$ rspec spec
```

```
.....
```

```
Finished in 0.5422 seconds (files took 7.43 seconds to load)  
39 examples, 0 failures
```



Rspec Example - Notebook Model

```
RSpec.describe Notebook, type: :model do
  user = User.new(email: 'user@domain.com', password: 'password')
  notebook = Notebook.new(name: 'Notebook Name', user: user)

  describe "Validations" do
    it "is valid with valid attributes" do
      expect(notebook).to be_valid
    end
  end
end
```



Trix

- “A Rich Text Editor for Everyday Writing”
- Notes more than just string
- code inside form partial

Content

B

I

Text

- **bold**
 - [hyperlink](#)



Trix - Form Partial

```
<%= form_for @note do |f|%>
  <p>
    <%= f.label :title %><br>
    <%= f.text_field(:title, {:class => 'form_control'}) %>
  </p>
  <p>
    <%= f.label :content %><br>
    <%= f.hidden_field :content, id: :content_body %>
    <trix-editor input="content_body"></trix-editor>
  </p>
  <p>
    <%= hidden_field_tag :notebook_id, params[:notebook_id] %>
    <%= f.submit({:class => 'btn btn-dark'}) %>
  </p>
<%end %>
```




JQuery and JQuery UI

```
1 <head>
2   <link rel="stylesheet" href="/theme_sheets/default_theme.css" type="text/css">
3   <script>
4     $( function() {
5       $( "input" ).checkboxradio();
6       $( "fieldset" ).controlgroup();
7     } );
8   </script>
9 </head>
10 <h1>Themes</h1>
11 <div class="widget">
12   <fieldset id="set">
13     <legend>Select a Theme: </legend>
14     <label for="radio-1">Default</label>
15     <input value="/theme_sheets/default_theme.css" type="radio" name="radio-1" id="radio-1">
16     <label for="radio-2">Green</label>
17     <input value="/theme_sheets/green_stylesheet.css" type="radio" name="radio-1" id="radio-2">
18     <label for="radio-3">Blue</label>
19     <input value="/theme_sheets/blue_stylesheet.css" type="radio" name="radio-1" id="radio-3">
20   </fieldset>
21 </div>
22 <script>
23   $(document).ready(function(){
24     $("#set").change(function(){
25       var sheet=$(this).val();
26       $("link").attr("href",sheet);
27     });
28   });
29 </script>
30
```

The background features a light blue gradient with a dark blue diagonal band. In the bottom right corner, there are overlapping geometric shapes in light blue and orange.

Demo

ByteNote