WAD Ruby + Sinatra Assignment 1

Stefan Rudvin 51549217

This is a wiki web application for tourist destinations in Scotland. It has an editable article that has information about the Dunnottar Castle in Stonehaven, Isle of Skye, Cairngorms National Wildlife Park and Glen Coe. The Wikipedia has multiple features listed below.

Note: Administrator username: ”Admin” password: “password”

Overview

Features:

- An article that can be edited by users with editing rights
- Word & Character Count
- Login page
- Account creation
- Send feedback to administrator
- Designed with CSS
- Custom font through Google’s font repositories
- Page visit counter

Administrator features:

- View log files (updated every time wiki is edited)
- Archive (Backup, View)
- Reset article
- Create, edit and delete users
- View feedback from users

To run this application, extract the .zip file and copy the ‘wiki’ folder into your preferred ruby on rails IDE which supports Sinatra (Cloud9 IDE and Chrome browser are recommended). Install the following gems if you do not have them installed yet: Sinatra, haml rspec, data_mapper, sqlite3, dm-sqlite-adapter.

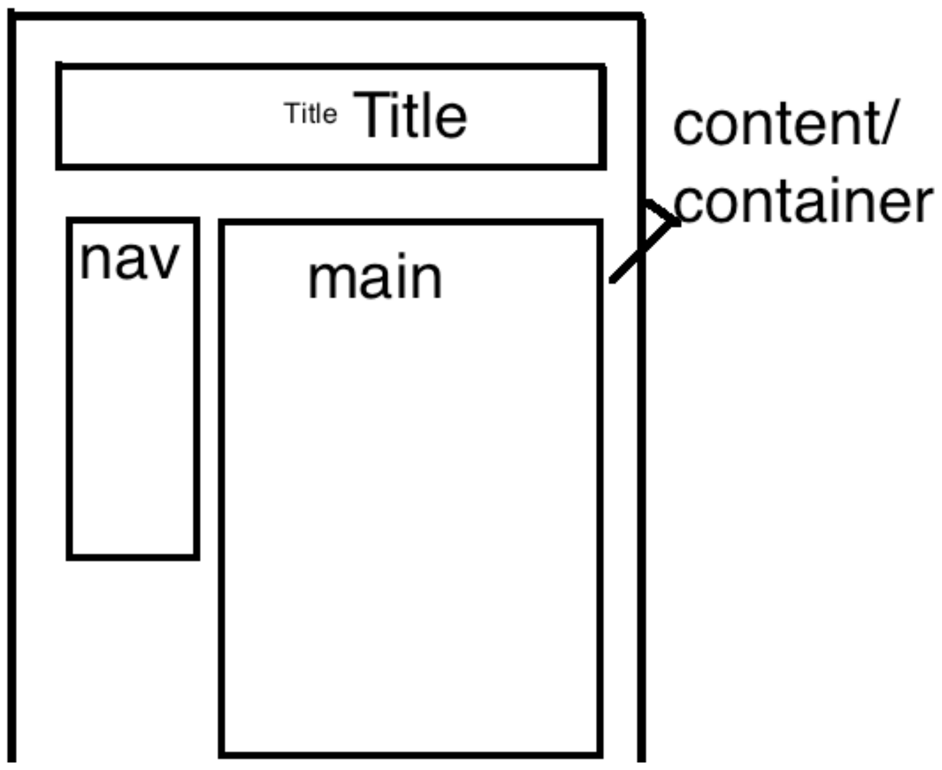
Navigate to the ‘/wiki’ folder and start the application with the command:

```
ruby wiki.rb -p $PORT -o $IP
```

Click on the link that appears to begin.

Appendices at the end of the report show screenshots of code and the website.

Design sketch



Features

Editable page

I created an editable page by writing text onto a .txt file that can be changed and opened anywhere within the application. This text file is manipulated when the article is edited and simultaneously copied to a log file.

Character and Word count

The characters were counted using the len.to_s and the word count was found with the .split.size function. These could then be displayed at the bottom of the article.

Page visit counter

I create a visit counter by creating a counter.txt file which appends a single letter 'h' every time the home page is called. The home page then displays the number of characters in the file, effectively showing how many times it has been visited.

Administrator Features

a. Override changes to the details made by a user

The administrator has the power to remove edit privileges from any user and edit the articles him/herself. He can also reset or look at the archives to find information that has been changed.

b. Archive the details to a separate text file

I created a separate archive txt. File to which I could send a copy of the wiki at any time. I used the append method so that previous entries in the archives would not be overwritten.

c. Add/edit/delete additional users and change their access privileges to edit details

These functions were created using the credentials method, were only the administrator can enter the admincontrols section where changes to users can be made. The administrator can change the edit privileges of users or delete them altogether. However, guests can create accounts for themselves but they won't get edit priviledges.

d. Enable the wiki to be reset by the administrator to its initial default details

To make this, I have a backup for the wiki article available so the reset function essentially removes the current article data and overwrites it with the backup file.

A link to a sign-in page which allows a user to register to edit the wiki

I created a page to create a user, but the administrator is required to gain edit rights. This was created using the DataMapper feature where usernames and passwords are handled.

A log text file which is appended each time the wiki is updated with details of changes made

For this application I created two log files: One that shows the username and time when the article was changes and another that shows the whole article so changes can be seen. I created separate methods for each, mostly using the file.open and file.puts functions.

A stylish layout displaying the wiki application

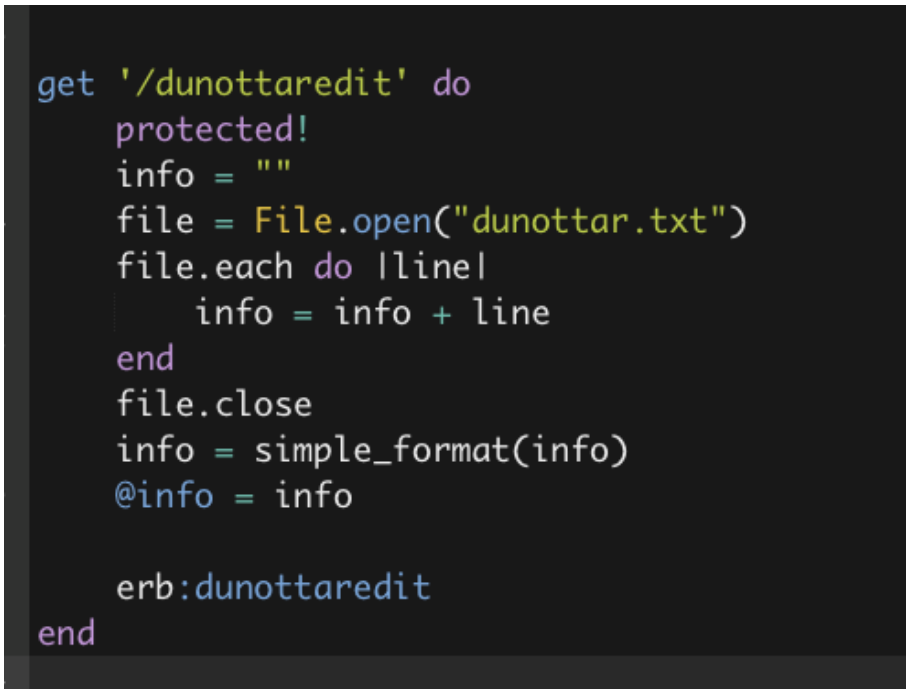
To create a stylish layout I used CSS with a background image, shadows, padding and margins. I also used the Roboto font from Google’s repositories.

Contact Administrator

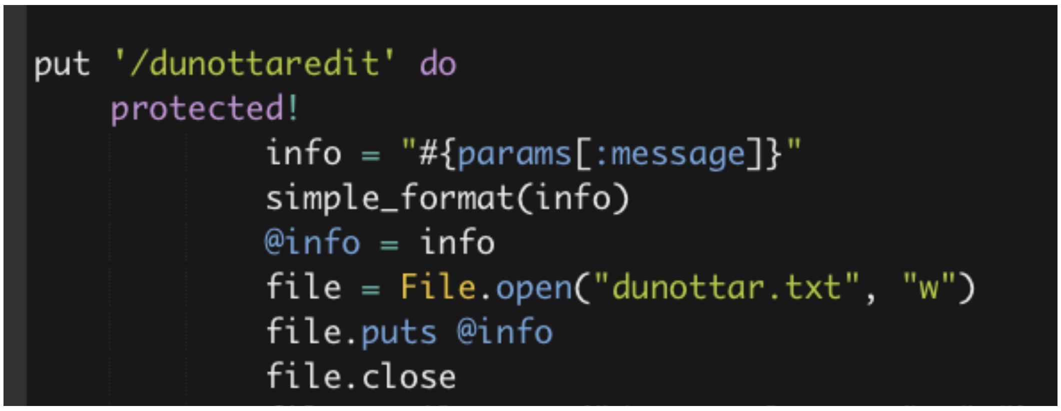
Lastly I wanted to create a feature where users (or guests) can contact the administrator for queries or wishes to gain edit permissions. These are linked in the administrator controls page.

Appendices

1. Editable page

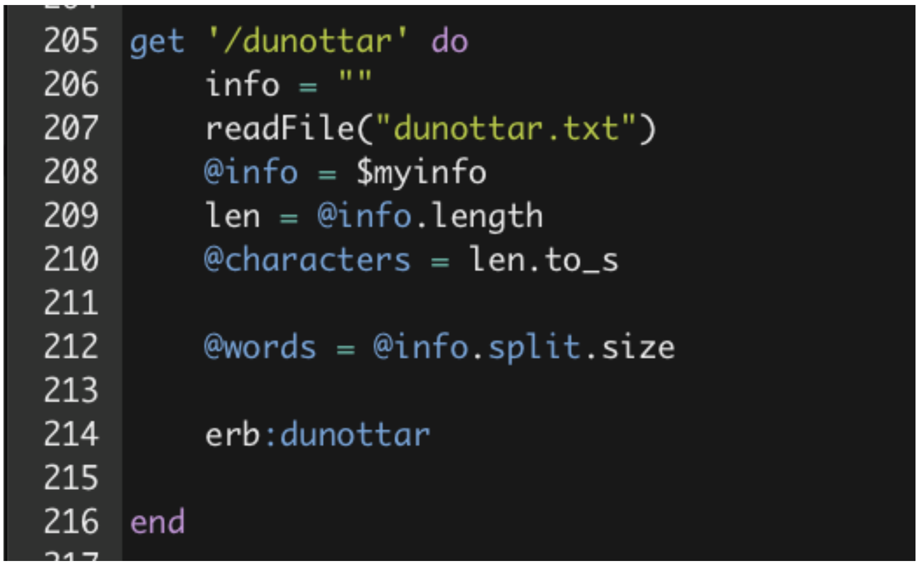


```
get '/dunottaredit' do
    protected!
    info = ""
    file = File.open("dunottar.txt")
    file.each do |line|
        info = info + line
    end
    file.close
    info = simple_format(info)
    @info = info

    erb:dunottaredit
end
```



```
put '/dunottaredit' do
    protected!
            info = "#{params[:message]}"
            simple_format(info)
            @info = info
            file = File.open("dunottar.txt", "w")
            file.puts @info
            file.close
```

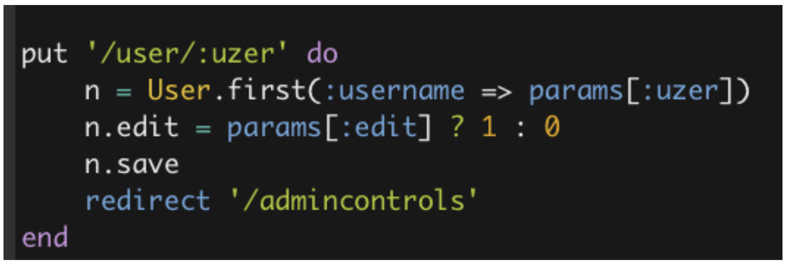
2. Character and Word count



```
get '/dunottar' do
    info = ""
    readFile("dunottar.txt")
    @info = $myinfo
    len = @info.length
    @characters = len.to_s

    @words = @info.split.size

    erb:dunottar

end
```

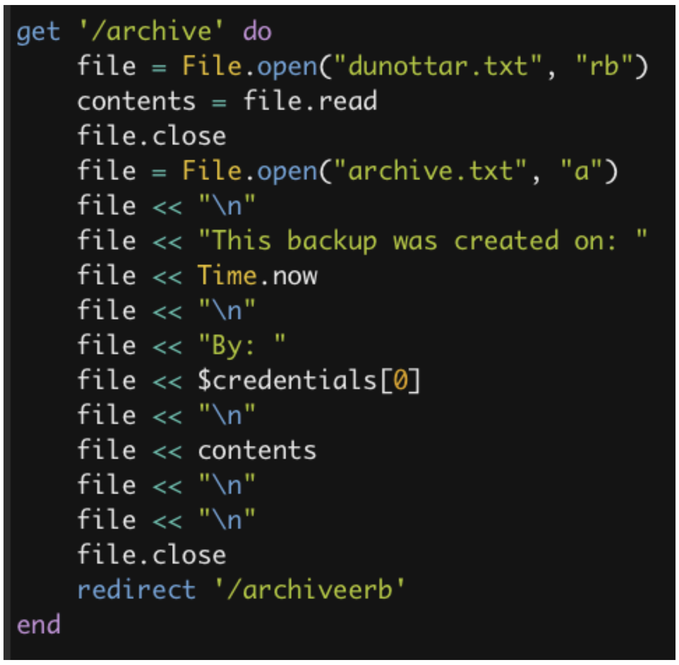
3. Administrator Features

a. override changes to the details made by a user



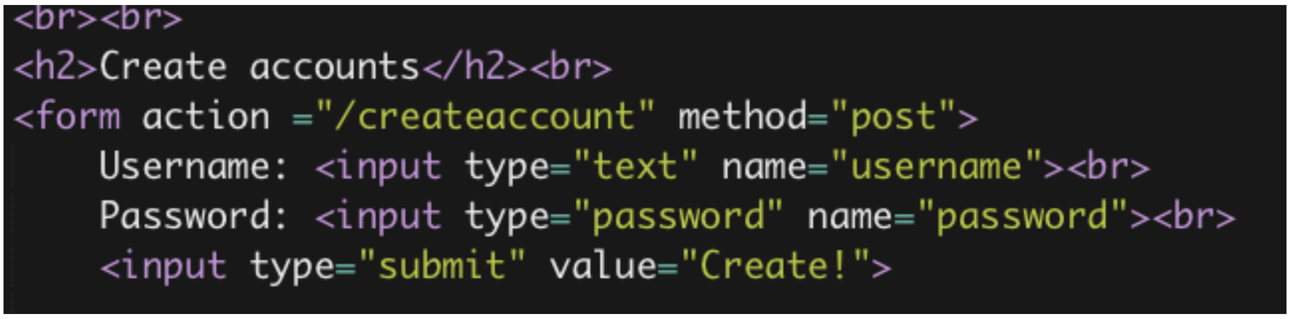
```
put '/user/:uzer' do
    n = User.first(:username => params[:uzer])
    n.edit = params[:edit] ? 1 : 0
    n.save
    redirect '/admincontrols'
end
```

b. archive the details to a separate text file

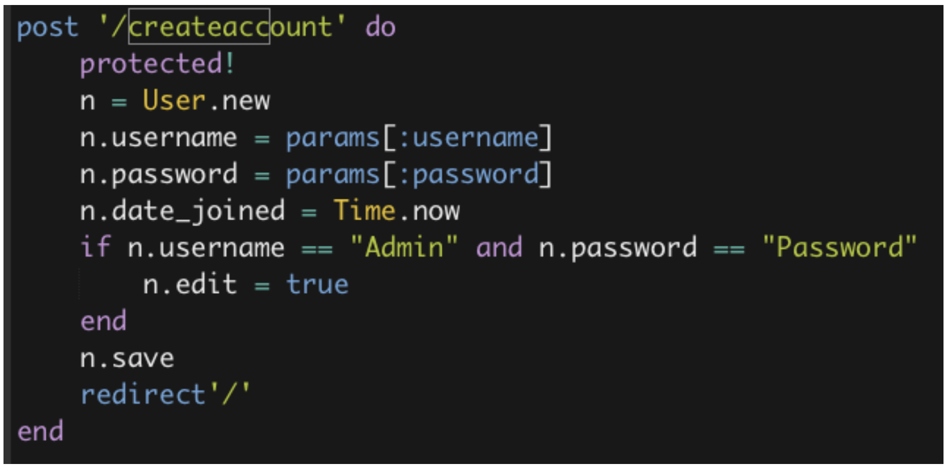


```ruby
get '/archive' do
    file = File.open("dunottar.txt", "rb")
    contents = file.read
    file.close
    file = File.open("archive.txt", "a")
    file << "\n"
    file << "This backup was created on: "
    file << Time.now
    file << "\n"
    file << "By: "
    file << $credentials[0]
    file << "\n"
    file << contents
    file << "\n"
    file << "\n"
    file.close
    redirect '/archiveerb'
end
```

c. add/edit/delete additional users and change their access privileges to edit details

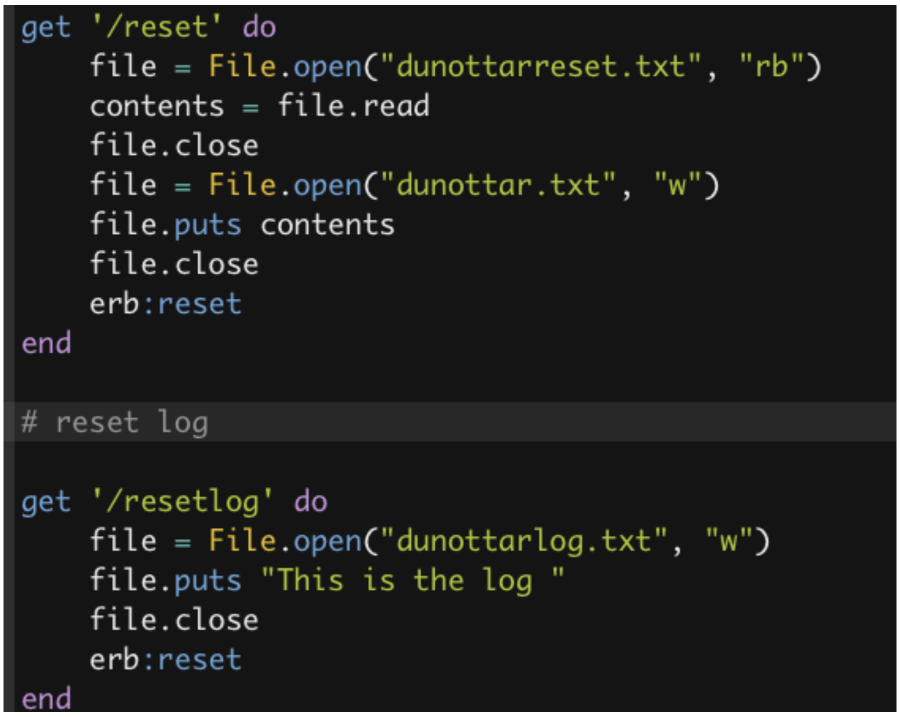


```html
<br><br>
<h2>Create accounts</h2><br>
<form action ="/createaccount" method="post">
    Username: <input type="text" name="username"><br>
    Password: <input type="password" name="password"><br>
    <input type="submit" value="Create!">
```



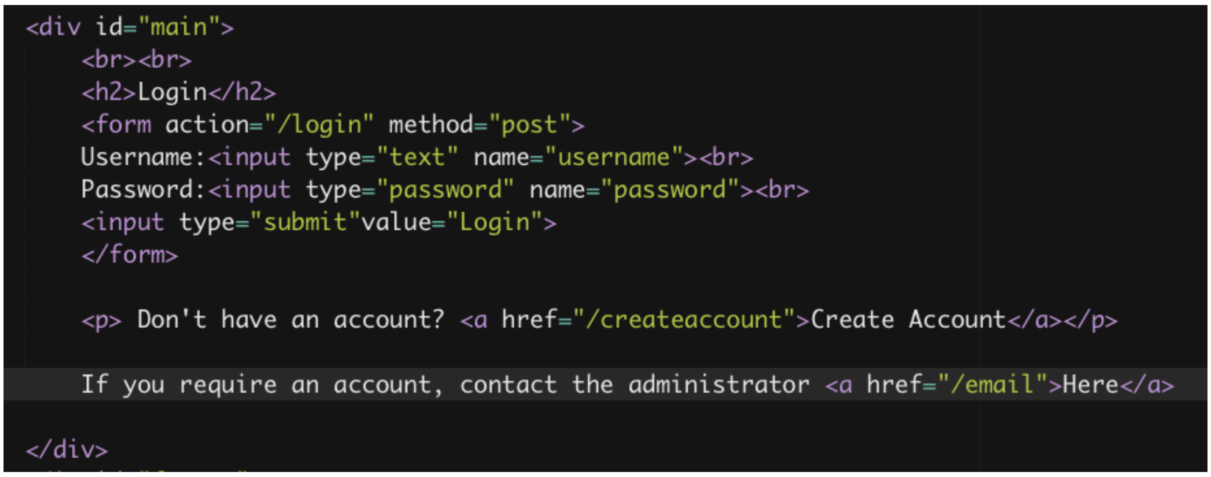
```ruby
post '/createaccount' do
    protected!
    n = User.new
    n.username = params[:username]
    n.password = params[:password]
    n.date_joined = Time.now
    if n.username == "Admin" and n.password == "Password"
        n.edit = true
    end
    n.save
    redirect'/'
end
```

d. enable the wiki to be reset by the administrator to its initial default details. Also code for resetting the log file.

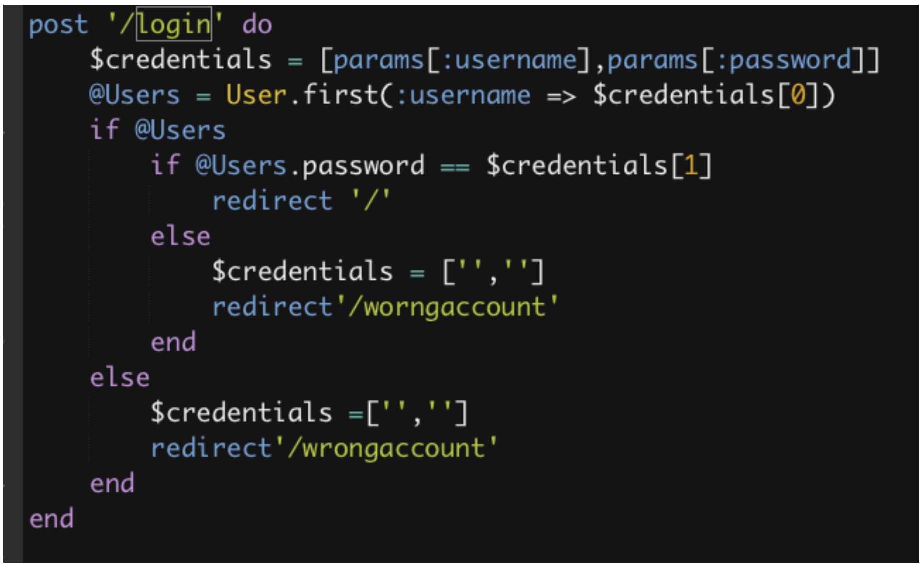


```ruby
get '/reset' do
    file = File.open("dunottarreset.txt", "rb")
    contents = file.read
    file.close
    file = File.open("dunottar.txt", "w")
    file.puts contents
    file.close
    erb:reset
end

# reset log

get '/resetlog' do
    file = File.open("dunottarlog.txt", "w")
    file.puts "This is the log "
    file.close
    erb:reset
end
```

A link to a sign-in page which allows a user to register to edit the wiki

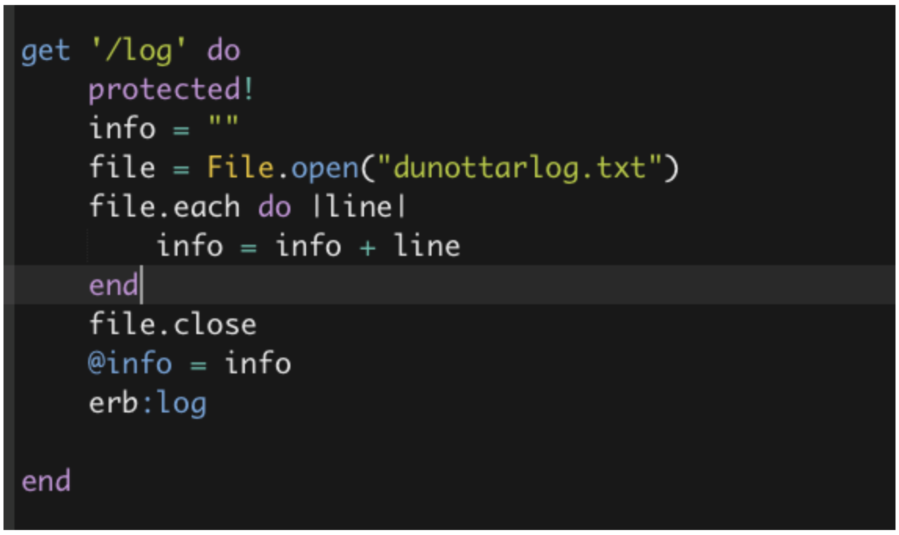


```html
<div id="main">
    <br><br>
    <h2>Login</h2>
    <form action="/login" method="post">
    Username:<input type="text" name="username"><br>
    Password:<input type="password" name="password"><br>
    <input type="submit"value="Login">
    </form>

    <p> Don't have an account? <a href="/createaccount">Create Account</a></p>

    If you require an account, contact the administrator <a href="/email">Here</a>

</div>
```

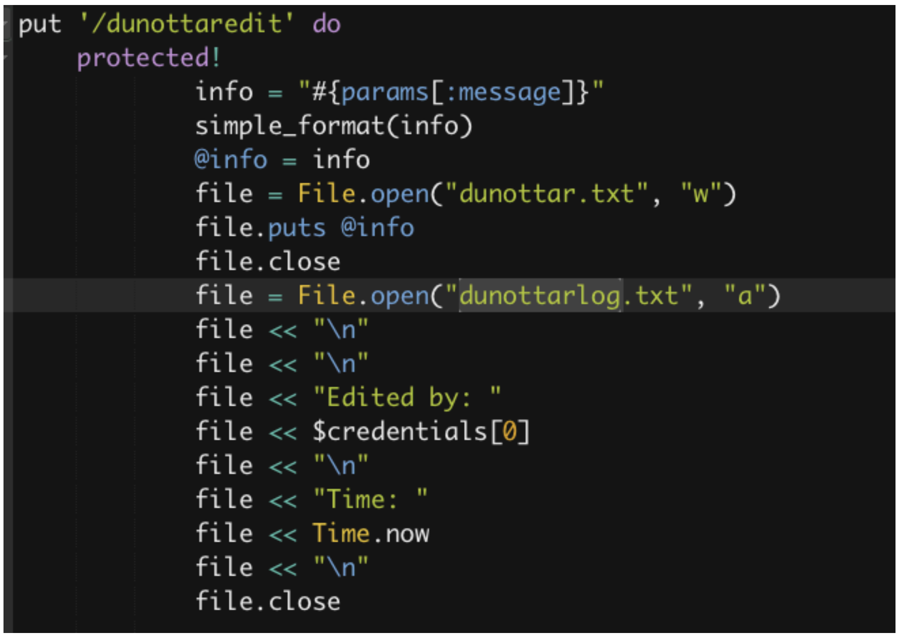


```ruby
post '/login' do
    $credentials = [params[:username],params[:password]]
    @Users = User.first(:username => $credentials[0])
    if @Users
        if @Users.password == $credentials[1]
            redirect '/'
        else
            $credentials = ['','']
            redirect'/worngaccount'
        end
    else
        $credentials =['','']
        redirect'/wrongaccount'
    end
end
```

A log text file which is appended each time the wiki is updated with details of changes made

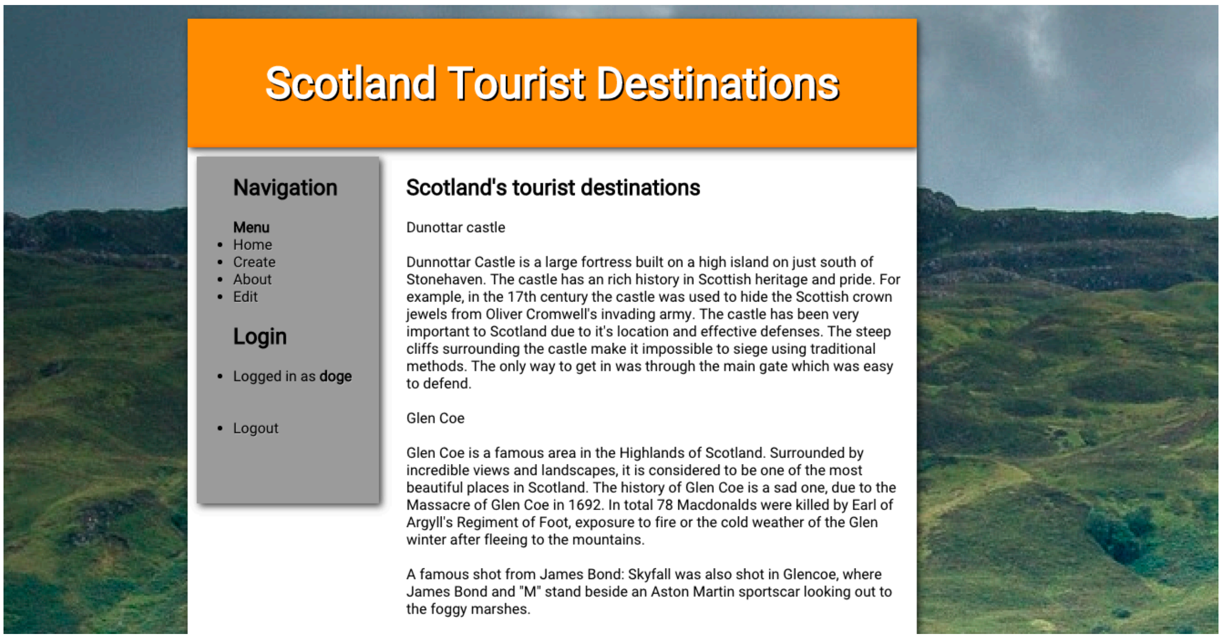


```ruby
get '/log' do
    protected!
    info = ""
    file = File.open("dunottarlog.txt")
    file.each do |line|
        info = info + line
    end
    file.close
    @info = info
    erb:log

end
```



```
put '/dunottaredit' do
    protected!
            info = "#{params[:message]}"
            simple_format(info)
            @info = info
            file = File.open("dunottar.txt", "w")
            file.puts @info
            file.close
            file = File.open("dunottarlog.txt", "a")
            file << "\n"
            file << "\n"
            file << "Edited by: "
            file << $credentials[0]
            file << "\n"
            file << "Time: "
            file << Time.now
            file << "\n"
            file.close
```

A stylish layout displaying the wiki application



Scotland Tourist Destinations

Navigation

Menu
- Home
- Create
- About
- Edit

Login

- Logged in as **doge**
- Logout

Scotland's tourist destinations

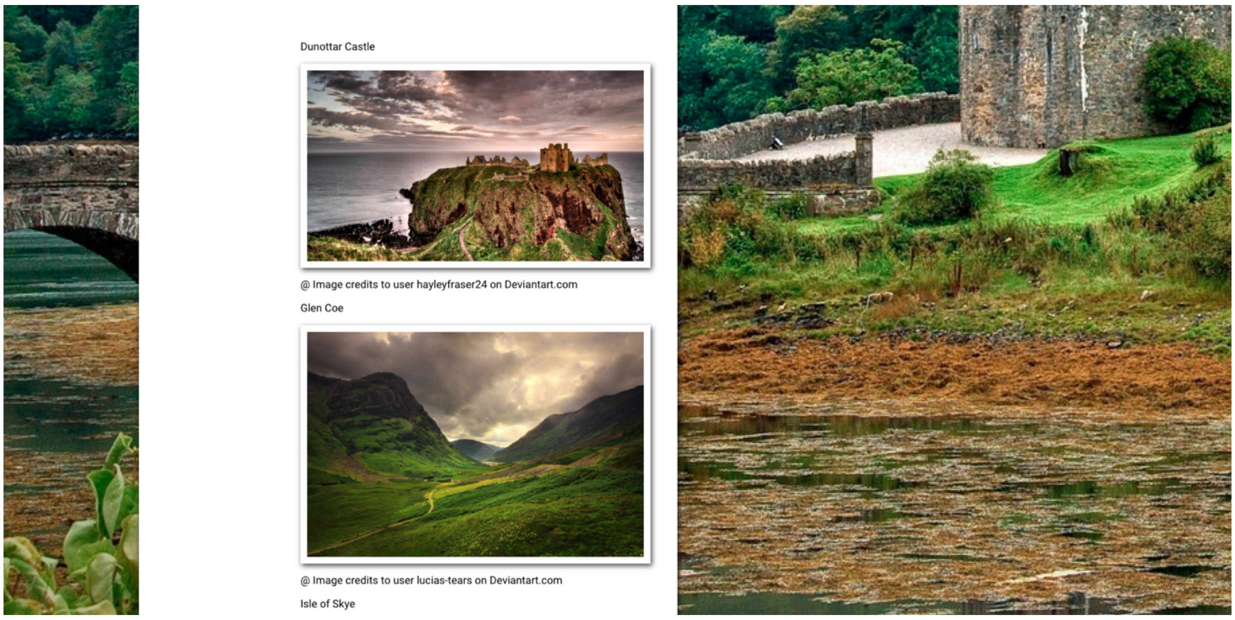
Dunottar castle

Dunnottar Castle is a large fortress built on a high island on just south of Stonehaven. The castle has an rich history in Scottish heritage and pride. For example, in the 17th century the castle was used to hide the Scottish crown jewels from Oliver Cromwell's invading army. The castle has been very important to Scotland due to it's location and effective defenses. The steep cliffs surrounding the castle make it impossible to siege using traditional methods. The only way to get in was through the main gate which was easy to defend.

Glen Coe

Glen Coe is a famous area in the Highlands of Scotland. Surrounded by incredible views and landscapes, it is considered to be one of the most beautiful places in Scotland. The history of Glen Coe is a sad one, due to the Massacre of Glen Coe in 1692. In total 78 Macdonalds were killed by Earl of Argyll's Regiment of Foot, exposure to fire or the cold weather of the Glen winter after fleeing to the mountains.

A famous shot from James Bond: Skyfall was also shot in Glencoe, where James Bond and "M" stand beside an Aston Martin sportscar looking out to the foggy marshes.



Dunottar Castle

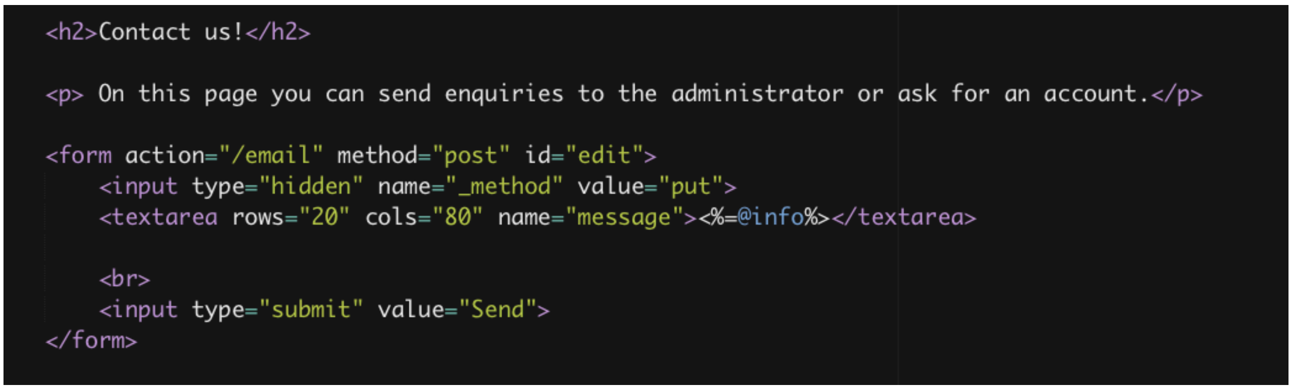
@ Image credits to user hayleyfraser24 on Deviantart.com

Glen Coe

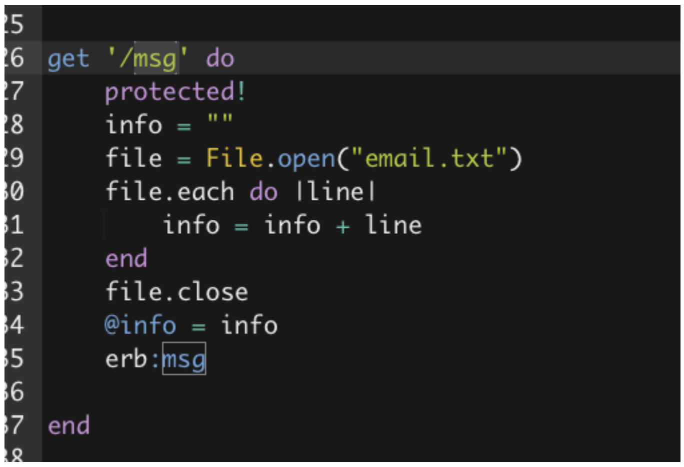
@ Image credits to user lucias-tears on Deviantart.com

Isle of Skye

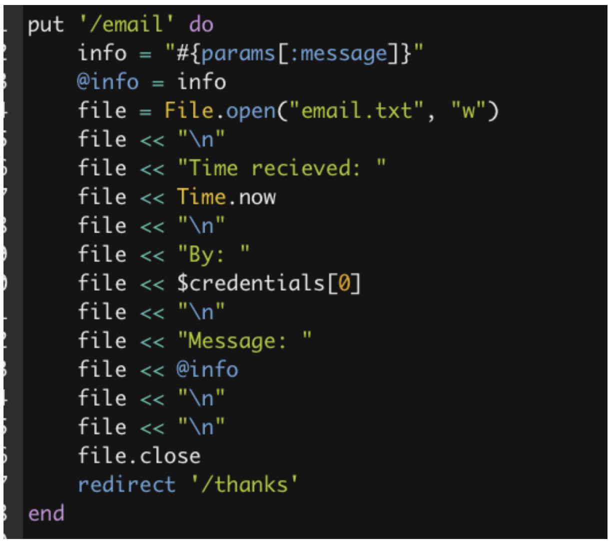
Contact Administrator



```
<h2>Contact us!</h2>

<p> On this page you can send enquiries to the administrator or ask for an account.</p>

<form action="/email" method="post" id="edit">
    <input type="hidden" name="_method" value="put">
    <textarea rows="20" cols="80" name="message"><%=@info%></textarea>

    <br>
    <input type="submit" value="Send">
</form>
```



```

get '/msg' do
    protected!
    info = ""
    file = File.open("email.txt")
    file.each do |line|
        info = info + line
    end
    file.close
    @info = info
    erb:msg

end
```



```
put '/email' do
    info = "#{params[:message]}"
    @info = info
    file = File.open("email.txt", "w")
    file << "\n"
    file << "Time recieved: "
    file << Time.now
    file << "\n"
    file << "By: "
    file << $credentials[0]
    file << "\n"
    file << "Message: "
    file << @info
    file << "\n"
    file << "\n"
    file.close
    redirect '/thanks'
end
```