DMIT2503 – ADVANCED WEB CONCEPTS

CODEIGNITER MVC PHP FRAMEWORK – DAY 2

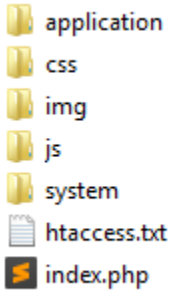
Overview: This lesson is continued from the CodeIgniter Lessons – Day 1, so we assume that you have CI installed, configured, and Bootstrapped. If not, please refer to that doc before moving on here.

OUR APP TEMPLATE - CONTINUED

Ok. Last time we looked at adding a simple header and footer as we load views in any Controller. Let's start today by adding some folders to our project root so we can start adding style sheets, javascript, images and all that other good web stuff.

Please add the following folders to your project root:

- css
- js
- img



Now, let's look at adding a custom style sheet to our app. One problem we will face is getting any relative links to work as we start using CI's segment based navigation rolling.
Our solution? Yes, let's not use relative pathing; lets use absolute pathing. Just like we have done in PHP class (dmit2025), we will use a PHP constant to hold the root (or "base") URL and use that to resolve all paths.

First, let's create a simple custom style sheet with a custom rule that we can see.

Create a file called styles.css in your /css/ folder, and add something that you can see either working…or not working. A simple test.

```
/* css/styles.css */
body{
	background-color: red;
}
```

Now, in order to create an absolute path to our stylesheet, we will use our base URL.

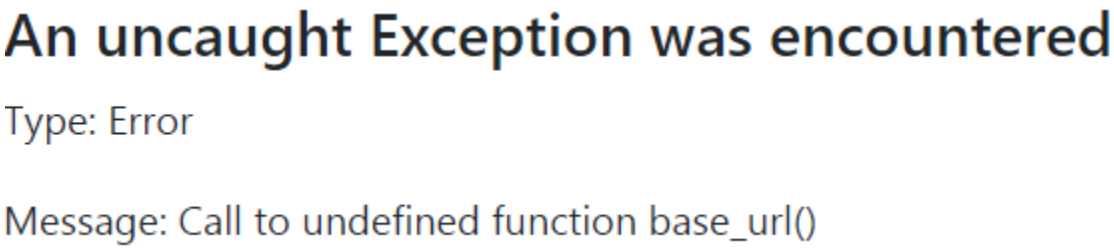
DMIT2503 – ADVANCED WEB CONCEPTS

In your ***application/views/includes/header.php ,*** add a link to your style sheet like so:

```
// application/views/includes/header.php

<link href="<?php echo base_url(); ?>css/styles.css" rel="stylesheet">
```

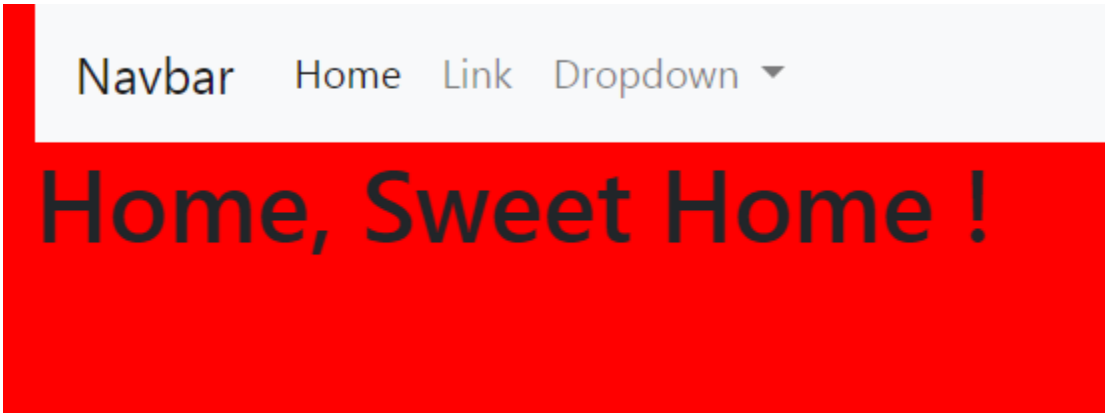
Here we are using a function called base_url() to write in the correct URL to the root of your app. Save, upload, and…



Whoops! Since we are using part of the CI framework that extends the core functionality, we need to load a ***helper***.

HELPERS & BASE_URL

- Open one of your config files called **application/config/autoload.php:**
- Seek out the line of helpers and add a text string for 'url': **$autoload['helper'] = array('url');**
- Save, upload, and see if your stylesheet is working now.



Note: Helpers (and Libraries) can be loaded globally through the config/autoload.php file. Whichever helper or library you load here is available throughout the entire app. The drawback is that too many may slow down your app.

We can also load helpers (or libraries) locally in controllers so that they are only available in the place you need them. Since we will be using the base_url() function on every page load, it needs to be loaded globally.

DMIT2503 – ADVANCED WEB CONCEPTS

You can get rid of the background color now. Just a test.

Remember: On dmitstudent, you may have to force-refresh (Shift + F5 in Chrome) to see CSS changes.

As we add links to any resource files (css, js, images, etc.) we will always use the base_url() to resolve this path.

- Links to segments (controllers/functions): <a href="<?php echo base_url()?>birds">Birds Main
- Links to images: <img src="<?php echo base_url()?>img/picture.jpg">
- Links to JS files: <script src="<?php echo base_url(); ?>js/jquery.js"></script>

Remember: In Day 1, we chatted about the importance of the **application/config/config.php** file and setting the base_url(): **$config['base_url'] = 'http://username.dmitstudent.ca/dmit2503/ci-2020/';**

Note: If any of these do not work, then View Page Source, and see if the path is correct. You may have to add a trailing slash to either the **$config['base_url']** or modify your links.

CONTROLLERS, FUNCTIONS, AND SEGMENT BASED NAVIGATION

In Day 1, we took a look at how we can view our Home controller , and even left off with another function (test). Note that the default function in Home controller is called index(). What does index do? It loads by default.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Home extends CI_Controller {
	public function index()
	{
		$this->load->view('includes/header');
		$this->load->view('home_view');
		$this->load->view('includes/footer');
	}
	public function test()
	{
		$this->load->view('includes/header');
		$this->load->view('test_view');
		$this->load->view('includes/footer');
		//echo "<h1>This is TEST function in Home controller</h1>";
	}
}
```

So, if we want to navigate to this Controllers index function, we can just called the Controller by it's name in the first segment of our nav links.

- So, we go to **http://username.dmitstudent.ca/dmit2503/ci-2020/home**
- For our function test(), we can add a 2nd segment to our URL: **http://username.dmitstudent.ca/dmit2503/ci-2020/home/test/**

Well, since our Home controller is the default (we set that in config/routes.php) and test() was just a test, then let's make a new controller with new functions, and create links to them.

PRETTY BIRD…PRETTY BIRD

Open your Controller called Home, and save it as a new file called Birds.php (in the controllers folder please).

First thing we change (because I always forget) is the class name. Then, add a few functions and some simple echo's to test our nav.

```
<?php
//controllers/Birds.php
defined('BASEPATH') OR exit('No direct script access allowed');

class Birds extends CI_Controller {


	public function index()
	{
		echo "Birds main";
	}
	public function loon()
	{
		echo "Birds - Loon";
	}
	public function sparrow()
	{
		echo "Birds - Sparrow";
	}
}
```

Now, let's save, upload, and test to see if we can navigate to these. Manually type in the segments to each:

- **http://username.dmitstudent.ca/dmit2503/ci-2020/birds**
- **http://username.dmitstudent.ca/dmit2503/ci-2020/birds/loon**
- **http://username.dmitstudent.ca/dmit2503/ci-2020/birds/sparrow**

Ok, if this works, let's create some links to each. Don't worry, we will do some actual bird stuff with these pages. In your **views/includes/header.php,** create links to each and remember to use your base_url() to create absolute paths.

```
// views/includes/header.php

<div class="dropdown-menu" aria-labelledby="navbarDropdown">
        <a class="dropdown-item" href="<?php echo base_url()?>birds">Birds Main</a>
        <a class="dropdown-item" href="<?php echo base_url()?>birds/loon">Loon</a>
        <a class="dropdown-item" href="<?php echo base_url()?>birds/sparrow">Sparrow</a>
    </div>
```

Here, I have used a Bootstrap 4 dropdown in my Navbar. Feel free to add links wherever you want if you prefer something else.

CONTROLLERS & VIEWS – GO TOGETHER LIKE PEAS & CARROTS

One of the things we have to deal with in MVC is how we move data between the 3 files we may need to create a single page view for the user. In CI, we have to create an array of data to move from the Controller to the View.

https://codeigniter.com/user_guide/general/views.html and scroll down to **Adding Dynamic Data to the View**

So, we create and populate an array of data in the controller like so.

```
//controllers/Birds.php

public function index()
{
        $data['heading'] = "The Birds of Alberta";
        $data['picture'] = "owl.jpg";
        $data['message'] = "<p>Birds (class Aves or clade Avialae) are feathered, winged, two-legged, warm-blooded, egg-laying vertebrates. Aves ranks as the tetrapod class with the most living species, approximately ten thousand. </p>
        <p>Extant birds belong to the subclass Neornithes, living worldwide and ranging in size from the 5 cm (2 in) Bee Hummingbird to the 2.75 m (9 ft) Ostrich. The fossil record indicates that birds emerged within the theropod dinosaurs during the Jurassic period, around 150 million years ago. <br>Most researchers agree that modern-day birds are the only living members of the Dinosauria clade.</p>";

        echo "<pre>";
        print_r($data);
        echo "</pre>";
}
```

The print_r() is just to visualize the array. If you don't like arrays….then you're not a web dev.

Next, let's see how a view will handle this. Create a new file in views called **bird_view.php.**

```
<!-- views/bird_view.php -->

<h2><?php echo $heading; ?></h2> <!--  each array item in the controller become a PHP variable in the view-->
<div class="whatever"><?php echo $message; ?></div>
```

Then, change your Controllers index function to remove the testing print_r(), and load your views.

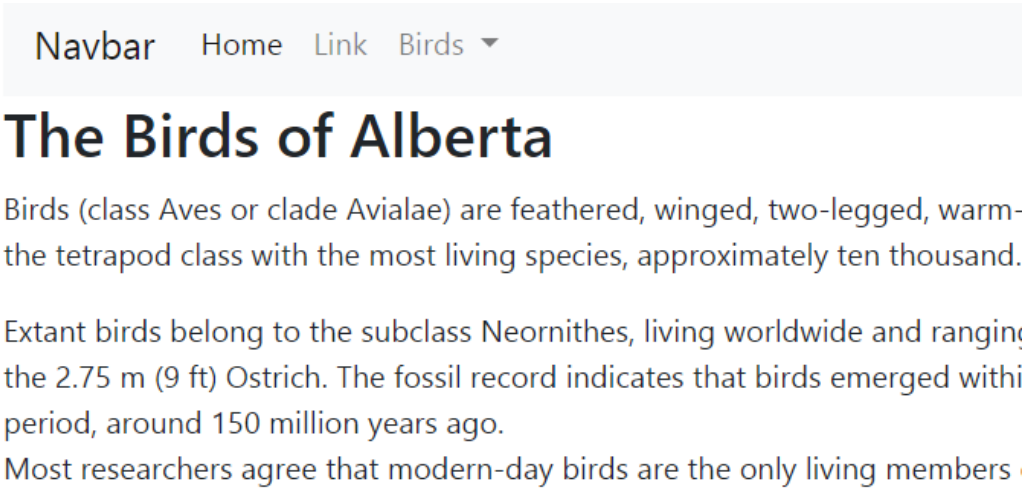
See how we need to pass the $data array to the view when we load it.

```
//controllers/Birds.php

public function index()
        {
                $data['heading'] = "The Birds of Alberta";
                $data['picture'] = "owl.jpg";
                $data['message'] = "<p>Birds (class Aves or clade Avialae) are feathered, winged, two-legged, warm-blooded, egg-laying vertebrates. Aves ranks as the tetrapod class with the most living species, approximately ten thousand. </p>

                <p>Extant birds belong to the subclass Neornithes, living worldwide and ranging in size from the 5 cm (2 in) Bee Hummingbird to the 2.75 m (9 ft) Ostrich. The fossil record indicates that birds emerged within the theropod dinosaurs during the Jurassic period, around 150 million years ago. <br>Most researchers agree that modern-day birds are the only living members of the Dinosauria clade.</p>";


                $this->load->view('includes/header');
                $this->load->view('bird_view',$data); // we need to pass the array to the view
                $this->load->view('includes/footer');
        }
```



Oooooh. Very pretty indeed! Let's do the same with our Loon function:

```
//controllers/Birds.php

public function Loon(){
```

```
                $data['heading'] = "The Loon";
                $data['picture'] = "loon.jpg";
                $data['message'] = "<p>The loon, the size of a large duck or small goose, resembles these birds in shape when swimming. Like ducks and geese but unlike coots (which are Rallidae) and grebes (Podicipedidae), the loon's toes are connected by webbing. The bird may be confused with cormorants (Phalacrocoracidae), not too distant relatives of divers and like them are heavy set birds whose bellies - unlike those of ducks and geese - are submerged when swimming.</p>
                <p> Flying loons resemble a plump goose with a seagull's wings, relatively small in proportion to the bulky body. The bird holds its head pointing slightly upwards during swimming, but less so than cormorants do. In flight the head droops more than in similar aquatic birds.</p>";

                $this->load->view('includes/header');
                $this->load->view('bird_view',$data);
                $this->load->view('includes/footer');
        }
```

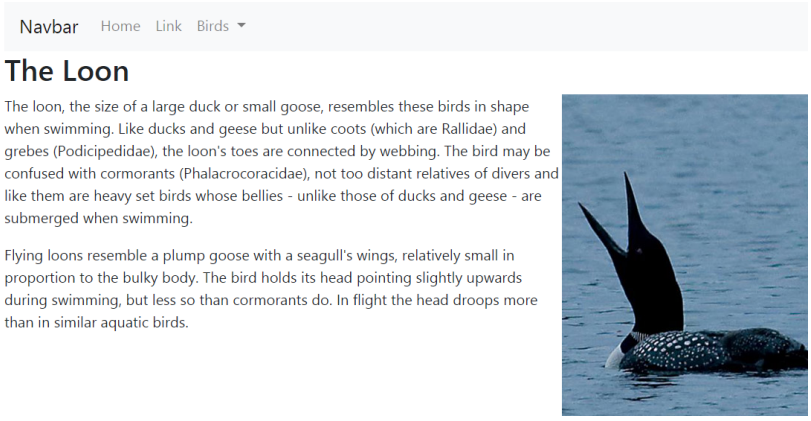
Notes:

- *Each array item populated in the Controller becomes a PHP variable in the view.*
- *We can re-use the same view and populate it with different data (this is best practice).*

But hey, what about the $data['picture'] = "loon.jpg"; ???

ACTIVITY – ON YOUR OWN…

- In bird_view, create an <img> tag and path it to a subfolder in /img/ called birds (img/birds/).
- On Moodle, the instructor has uploaded a folder called Birds. These have some pics for you to put into your /img/birds/ folder.
- Next, in your <img src=””>, use PHP blocks to both path to the base_url() first, then echo the picture variable. When this doesn't work first time, View >> Page Source and figure out the correct path.
- Add any CSS or Bootstrap classes to align your pictures. I'm using a float cuz I'm old-school ☺

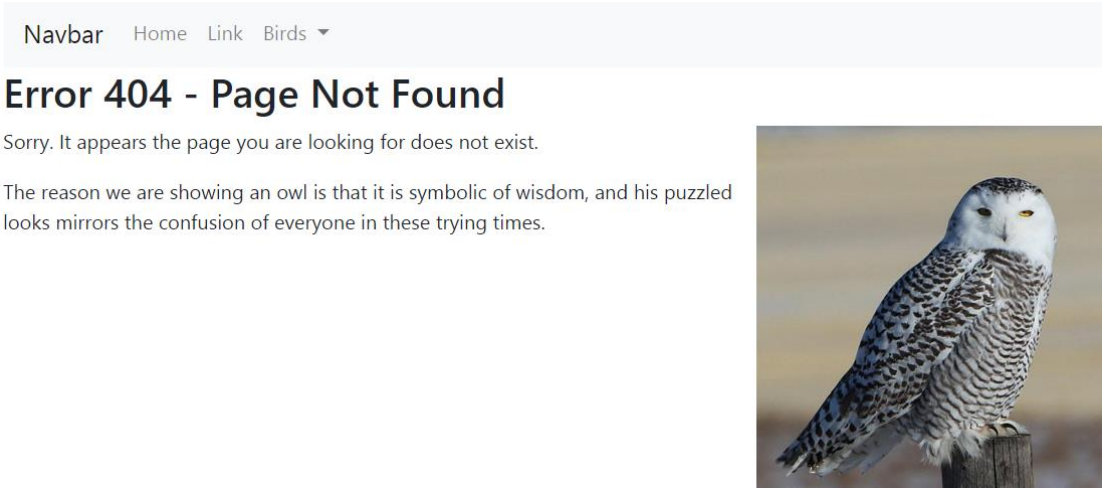


- Once you have that, go ahead and create more content for the Sparrow function. Feel free to raid Wikipedia for some content about sparrows.
- Note there is also a picture of a Falcon. What to do?
    - Create a new function in Birds controller for the Falcon.
    - Create a new nav link for this as well.
- Since we also have an Owl picture, we're using that for the birds main.

CUSTOM 404 NOT FOUND

Ok. Here's a little task you can do on your own. No code from me. ☺

- Create a new Controller called Error404.php. You can copy/paste the Home or Birds controller. We only need the index() function.
- You can either create a new view, or just re-use the bird_view like I did.
- The text content should be something like "404 – Page Not Found."
- In config/routes, change the line of the 4040 override: **$route['404_override'] = 'error404';**
- Save, upload, then test by typing in some gibberish after your base URL: **http://username.dmitstudent.ca/dmit2503/ci-2020/birds/dsadadassdasdsa**
- Voila: You have a custom 404 error.



LOOKING FORWARD TO NEXT CLASS

For next class, please go to the online help for CodeIgniter (https://codeigniter.com/user_guide/index.html)and read up on the following areas:

- Forms
- Validation
- Database

See you then ☺