Custom Layouts Using Bootstrap

Traditional web site design allows for limited customization for individual users, while this makes sense for applications serving a wide variety of one time users, applications such as the LMS, where a registered group of repeat users provide most of the traffic, more customization can and should be afforded to individual users. The purpose of our extension is to allow LMS users to pick custom layouts configured with Bootstrap. Each of these layouts is defined in the static/css folder and represented in the database. The corresponding session variables, database definitions, and controller-view pairs are explored below.

**SESSION VARIABLES:**

current\_theme: Tracks which theme should be used to render the current session. Defaults to the “Default” theme

preview\_theme: Reference to the preview image requested to be displayed to the user.

previous\_theme: Reference to the previous value of “current\_theme”. Used to allow users to revert changes.

**DATABASE:**

*theme*

name: Name of the theme, used as the user friendly identifier

URL: Relative path to bootstrap css file defining this theme

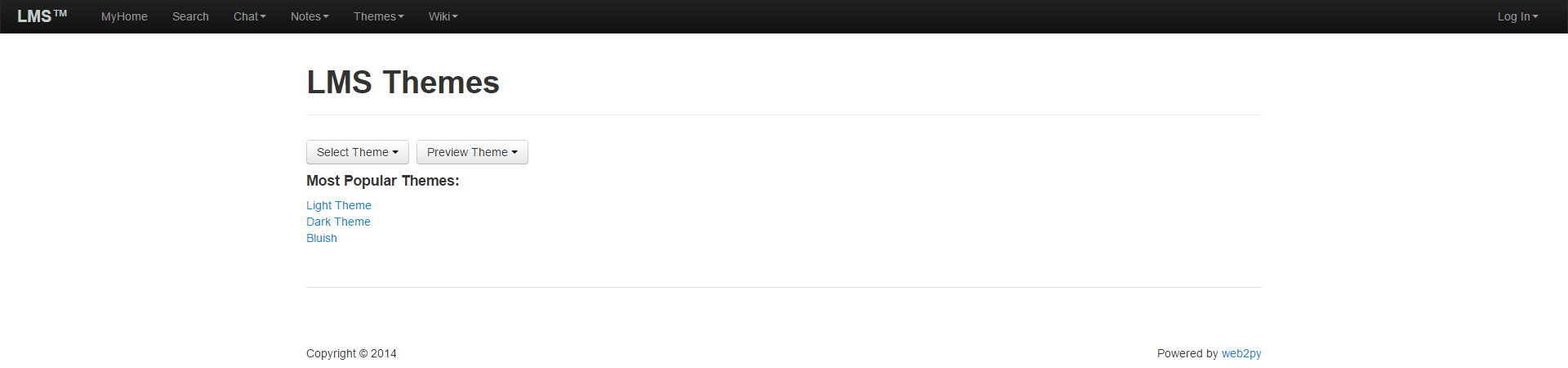
image\_URL: Relative path to the preview image for this theme

**CONTROLLER FUNCTIONS AND VIEWS**

*index*

Controller: Returns a dictionary of available themes.

View: Renders two dropdown menus of links to each theme. One list sets the theme selected (*theme\_picked*) while the other previews the theme (*preview*) without changing the current theme. A list of links to pick each of the 3 most popular themes (*theme\_picked*).

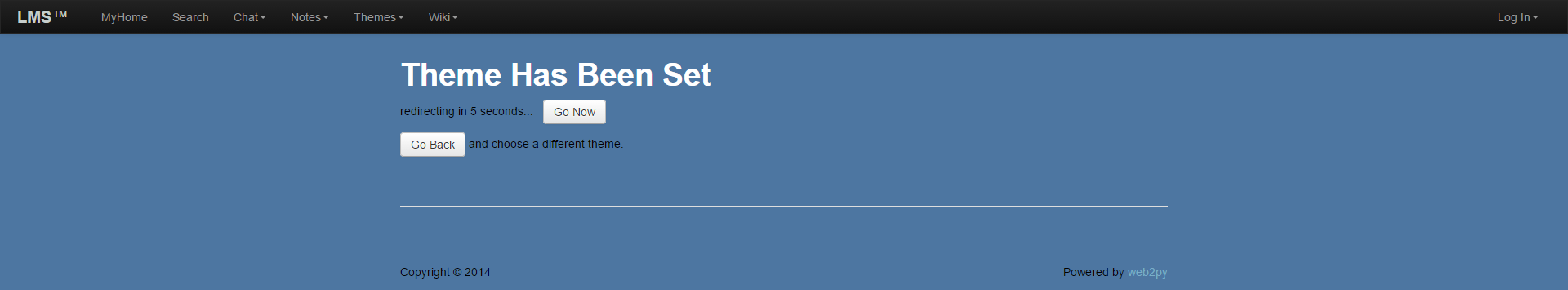


*theme\_picked:*

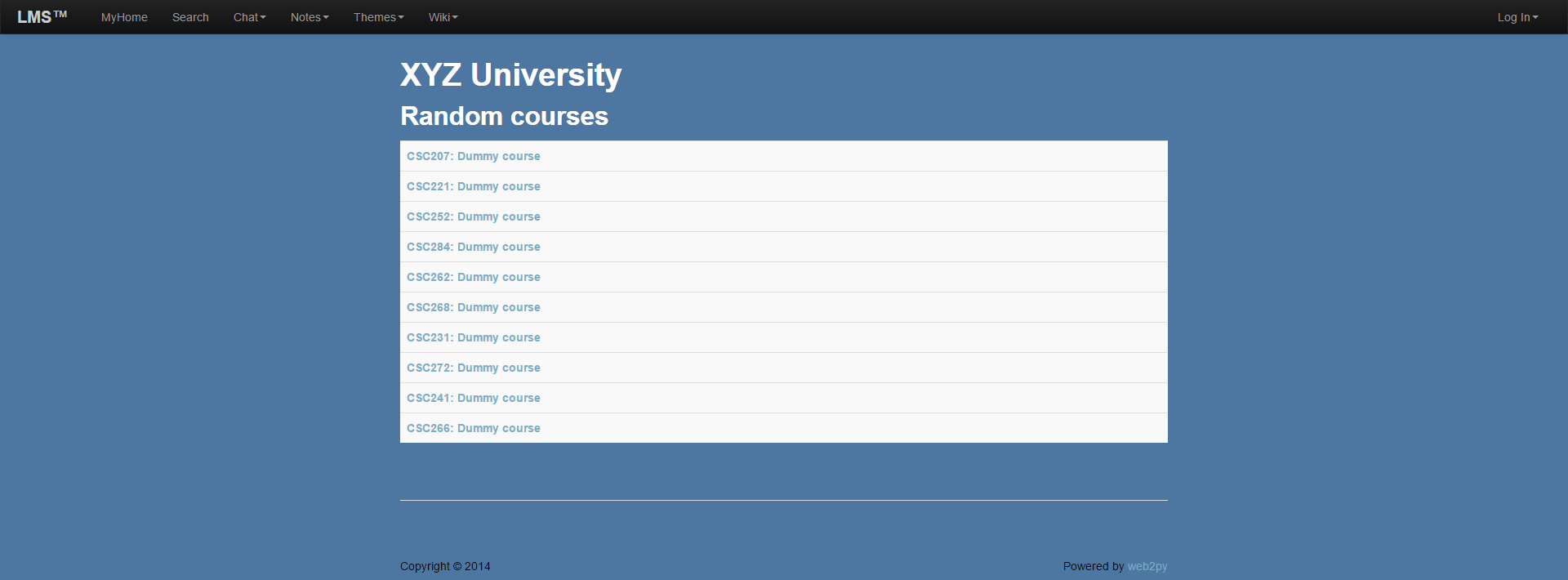
Controller: Expects a subfolder and filename for the requested css theme. Sets the previous theme to the current theme, sets the current session theme to the requested theme, increments to use count on the selected theme in the database, and sends the browser to the “Countdown” view (via the *countdown* controller).

View (countdown): Page rendered using the selected theme (a). JavaScript on the page counts down from 5 seconds. If the countdown expires or the user clicks on “Go Now” the browser is redirected to the home page rendered with the theme of the user’s choice (b). If the user clicks “Go Back” the browser is redirected to the index page with the theme reverted to the previous value (*theme\_back*).

a – countdown landing page



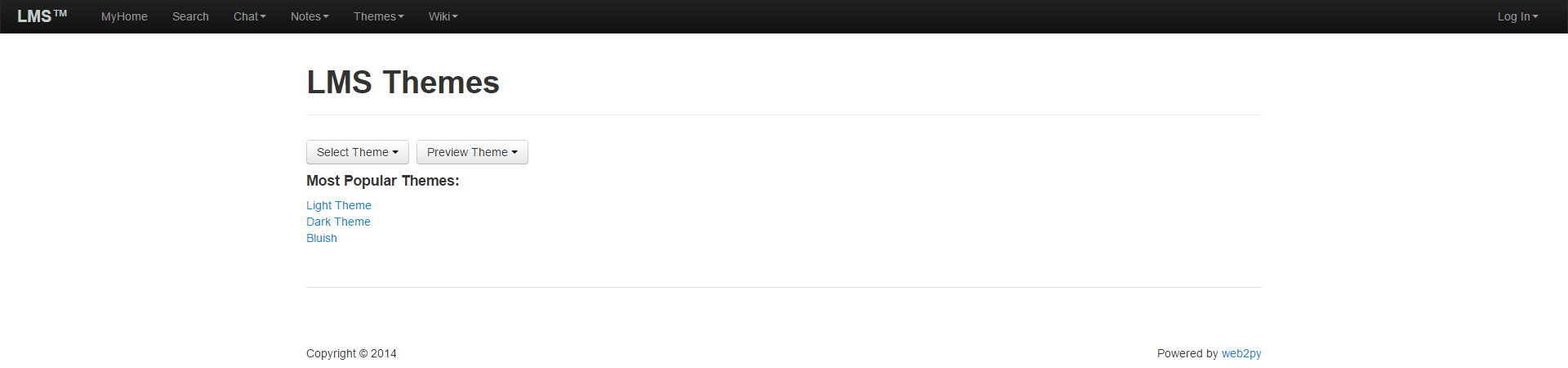
b – home page with selected theme



*theme\_back:*

Controller: Sets the current theme to the previous theme, decreases the use count of the previous theme, and sets the previous theme value to an empty string. Redirects the user back to the index page rendered with the previously selected theme.

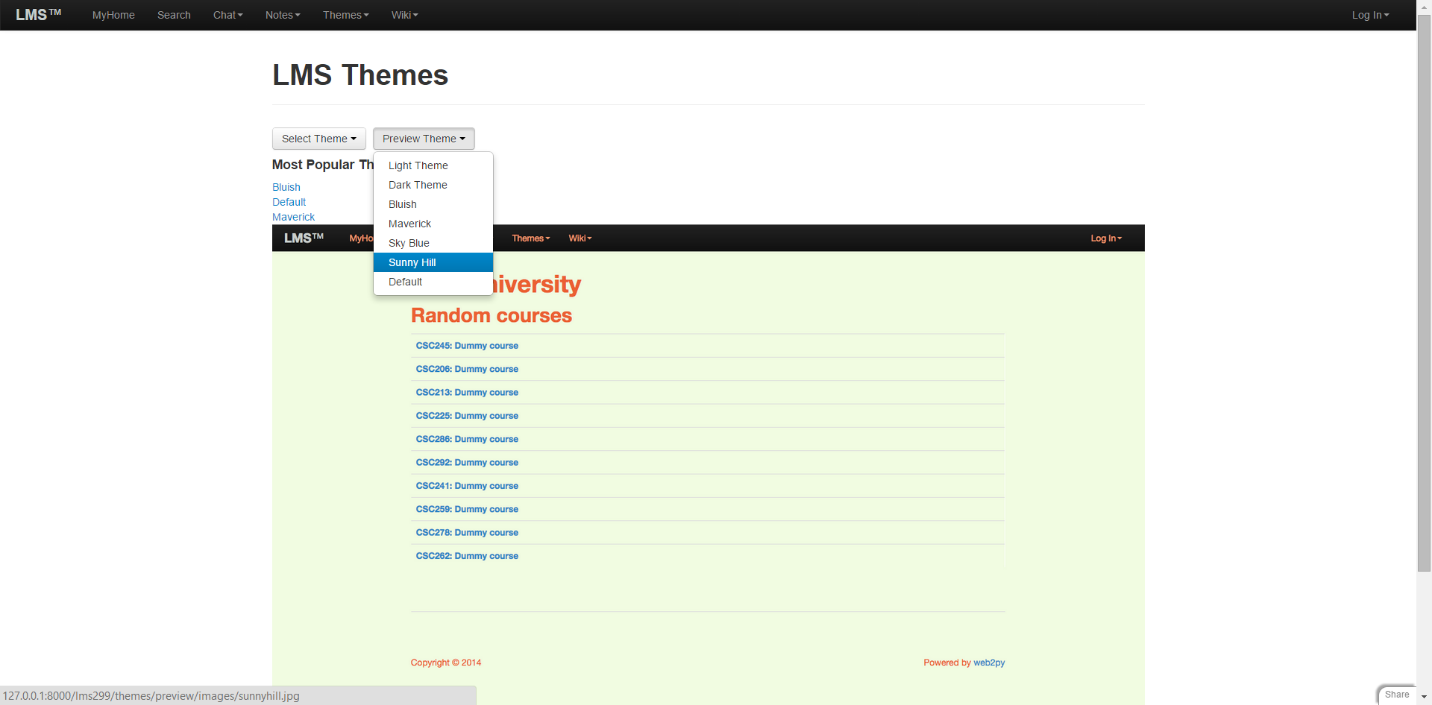
View (index): Renders two dropdown menus of links to each theme. One list sets the theme selected (*theme\_picked*) while the other previews the theme (*preview*) without changing the current theme. A list of links to pick each of the 3 most popular themes (*theme\_picked*).



*preview:*

Controller: Expects a subfolder and filename for the requested css theme. Sets the preview session theme to the requested theme. If one or both parameters are missing, the preview theme is set the the default theme. Returns an empty dictionary.

View (index): Renders a single static image corresponding to the preview\_theme within the preview\_theme div of the index view.



*course\_themes*:

Controller: *Requires the user to be an authorized administrator.* Returns a dictionary of themes associated by course. (To do: expand to allow administrators to set a default theme for each course)

View: Renders the themes by course within the standard layout

