PHP & DATABASES 2

Task 4

Dante Camenzuli

Difficulties

I hadn't encountered too many difficulties whilst putting the code together, but I did run into a few issues that I could not solve as neatly as I would have liked.

Most problems I encountered involved missing variables or the wrong variable name in the model or controller. This particular framework does make things generally frustrating since one would need to consult two other pages besides the html page, they're working on to make sure everything is working in conjunction to one another.

I did have one issue that I could not solve the CodeIgniter way and that was the implementation of a drop-down list populated with information gathered from the database. I followed the information from the official website as well as found other tutorials on YouTube and blogs, but to no avail. I had to use the typical PHP route to create the drop-down lists necessary whilst still using the model and controller.

The strangest problem I've encountered was the style sheet. No matter what put in it, nothing would surface on the view page. The link was correct, but I might as well have done without it completely. I found myself relying on Bootstrap's styling system even though it not always worked the way I intended.

Website Functions

CXP_Controller

index()

Builds the index page of the website.

No parameters required.

bookseats()

Builds the booking page of the website.

No parameters required

create()

Builds the page to add movie information to the database.

No parameters required.

change_cinemas()

Amends the movies that assigned to each cinema.

No parameters required.

delete()

Deletes the selected movie from database.

Parameters:

> $slug: Movie title slug is used for identification purposes.

edit()

Builds the page to edit movie information

Parameters:

> $slug: Movie title slug is used for identification purposes.
>
> $submit: Boolean necessary to identify if the form has been submitted.

movielist()

Builds the movie list page of the website, accessible to the admin.

No parameters required.

ticket()

Builds the receipt page after booking has been completed.

Parameters:

$ticket_id: ID of the booking required to retrieve all necessary information from the database.

view()

Builds the page to view the information of a selected movie.

Parameters:

$slug: Movie title slug is used for identification purposes.

_do_booking()

Executes guest booking.

No parameters required.

_do_create()

Executes movie creation.

No parameters required.

_do_edit()

Executes amending of movie information.

Parameters:

$movie: Array containing all information submitted from form.

_build_dir()

Creates upload directory if none available.

Parameters:

$dir: contains directory path.

_upload_image()

Uploads a copy of the image to the upload folder.

Parameters:

$name: ID of the movie is used as the name of the image file.

_get_image_path()

Retrieves the image of the selected movie.

Parameters:

$id: ID used to retrieve a single image.

$to_array: used in case more than one image file matches the $id.

Returns:

File directory/ies.

Movie_model

assign_cinema()

Assigns a movie to a cinema in the database.

Parameters:

$movie_id: The movie ID

$cinema_id: The cinema ID

Returns:

Success or failure on update.

create_booking()

Creates appropriate rows in tbl_bookings and tbl_booked_seats for a logged in guest user.

Parameters:

$user_id: Stores the user's ID

$movie_id: Stores the movie ID

$showtime: Stores the selected showtime ID

$seats: Stores array of theatre seats selected

Returns:

Ticket ID if successful.

create_movie()

Creates a row in tbl_movies with information of a single movie, as well as appropriate rows in tbl_movie_categories to store the genres associated with that movie.

Parameters:

$title: Stores movie title.

$release_date: Stores release date

$rating: Stores rating ID

$actors: Stores string of actors

$writers: Stories string of writers

$directors: Stores string of directors

$price: Stores ticket price

$categories: Stores array of movie genres.

Returns:

Movie ID if successful.

delete_movie()

Deletes a movie from database.

Parameters:

$slug: Stores the movie title's slug.

Returns:

Success or failure on deletion.

get_booked_seats()

Retrieves the booked seats of a purchased movie ticket.

Parameters:

$ticket_id: Stores the ticket ID.

Returns:

Arrays consisting of the seat row/s and numbers.

get_category()

Retrieves a single genre from the database:

Parameters:

$id: Stores the genre ID.

Returns:

Array of selected genre.

get_categories_array()

Retrieves the list of genres from the database.

No parameters required.

Returns:

Array of genres.

get_cinemas()

Retrieves the list of theatres from the database.

No parameters required.

Returns:

Array of cinemas.

get_cinema_movies()

Retrieves the list of theatres and the movies assigned to them.

No parameters required.

Returns:

Array of cinemas, and movie titles and their IDs.

get_columns()

Retrieves the list of seat numbers in a theatre.

No parameters required.

Returns:

Array of seat numbers.

get_movie()

Retrieves the information of a single movie.

Parameters:

$slug: Stores the movie title's slug for identification.

Returns:

Array of movie information.

get_movies()

Retrieves the list of movies available and the cinema they are assigned to, if any.

No parameters required.

Returns:

Arrays containing the movie's ID, title, and slug, as well as the cinema ID and name of the theatre.

get_movie_category()

Retrieves the list of genres of the movie selected.

Parameters:

$movie_id: Stores the movie's ID.

Returns:

Array of the genre's ID associated with the movie selected.

get_ratings()

Retrieves the list of ratings from the database

No parameters required.

Returns:

Array of rating information.

get_rows()

Retrieves the list of seat rows in a theatre.

No parameters required.

Returns:

Array of seat rows.

.get_showtimes()

Retrieves the list of showtimes of a movie.

No parameters required.

Returns:

Array of showtimes.

get_ticket()

Retrieves the information of a purchased movie ticket.

Parameters:

$ticket_id: Stores the ticket ID

Returns:

Array consisting of the movie title, cinema, and showtime.

replace_categories()

Updates the genres assigned to a movie.

Parameters:

$id: Stores the movie's ID

$categories: Stores any genres selected.

Returns:

Success or failure on update.

update_movie()

Updates the information of a selected movie in the database.

Parameters:

$id: Stores movie’s ID.

$title: Stores movie title

$release_date: Stores release date.

$rating: Stores rating ID.

$actors: Stores string of actors.

$writers: Stores string of writers.

$directors: Stores string of directors.

Returns:

Success of failure on update.