

Week 4 Project (Sinatra): Lodge Reservations System

This will be a website for a lodge (bed & breakfast type thing). There will be several static pages for general info about the lodge. An on-line calendar will show availability for prospective visitors. A form will collect reservation requests. Once approved (by lodge management), the reservation will be added to the database.

"Should" cases

The following things should be done through a web application.

1. Create reservation records
2. Edit reservation records
3. Fetch and read reservation records
4. Delete reservation records
5. Assign group activities to a reservation (many to one)
6. Assign guests to a reservation (many to one)
7. Assign guests to individual activities (many to many)

"Should not" cases

We should not be able to:

- Have more than one "reservation" on a single date.
- ??? (I'm sure I'll figure some out as I go.)

Extensions

If everything is functional, I would like to:

- Figure out a way for the managers to "log-in" to the reservation management system, so it is hidden from the public.
- Integrate the forecast.io API to display the current weather at the lodge.
- Create a photo gallery (like a carousel?) to display pictures. (Static page of photos would be an easy first step if I can't get a carousel working.)
- Be able to create / print invoices for guests (merging reservation information into a template).
- Use a pop-up calendar to select reservation dates in the form. (Like they use on airline websites...)

Testing

You should write tests.

Gem

I'm hoping to find one that has something to do with calendar reservations. Otherwise, I'll probably use one for the forecast API. ?? Would love suggestions...

FILES

DATABASE

lodge.db

TABLE: reservations

TABLE: activities

TABLE: group_activities

TABLE: guests

RUBY

menu.rb

And TBD...

ATTRIBUTES

Reservation

PK @id

M @last_name

M @start_date

M @end_date

M @no_children

M @no_adults

FK @group_activity_id

FK @location_id

GroupActivity

PK @id

M @name

IndividualActivity

PK @id

M @name

M @price

M @limit

Guest

PK @id

M @first_name

M @last_name

M @age

PK = Primary Key

M = Mandatory

FK = Foreign Key

CLASSES / METHODS

Reservation

.new

#insert

.search*

#save* (update)

.all*

.delete*

Group/Individual Activity

.new

#insert

.all*

delete*

Guest

.new

#insert

.all*

.delete*

MODEL RELATIONSHIP

