

Assignment #3

Use everything you've learned so far to create a "travel log" app.

Guidelines

- Use [this repository](#) as a template for your own `hw3` repository in your GitHub account.
- The models (and associations) and database migrations have already been provided. Upon opening your workspace, run `rails db:setup` to create the database. Then, have a look at the model and migration files to see how the database is structured.
- Code the application per the requirements below, committing and syncing often.
- Submit your assignment via Canvas, using the address for your repository (i.e. <https://github.com/<YOUR GITHUB USERNAME>/hw3>) as the "Website URL".

User Stories

For this assignment, assume it's just for your personal use. We will learn how to adapt it for multiple users later.

- As me, I want to see a list of places, so I can remember all the places I've visited and find their journal entries.
- As me, I want to be able to add a new place (with its name), so I can see it in the the list of places I've visited.
- As me, I want to be able to click on the name of a place and see a list of its entries, so I can remember the things I did while I was there.
- As me, I want to log new entries for a place (including title, description, and date), so I can keep track of all the things I did in each place.

Wireframes

Travel Journal

Places I've Been

[Mexico City](#)

[Charleston](#)

[Beijing](#)

[Amsterdam](#)

[New Place](#)

Travel Journal

Mexico City

Ate tacos
2022-01-01



Lisa and Jim's Wedding
2022-01-01



Ate tacos again
2022-01-01



[New Post](#)

Rubric (25 points)

- Configure the `routes.rb` file to support the two resources needed
- Write controller code to support the wireframes and user stories
 - Backend logic for reading and creating data should be in the controllers as much as possible
- Create all the necessary views with the appropriate content to support the wireframes and user stories

Part 1 - Places (15 points)

- A view with the list of places (2), along with links to get to the details view for each individual place (2), and a link to the new place form (1)
- A view for each individual place showing its name (1) with a link back to the full list of places (1)
- A view with a form to create a new place (2)
- Places controller with index (1), show (1), new (1), and create (2) actions, each of which reads and/or creates data as needed
 - Controller actions that don't render a view should redirect to another route in the application (1)

Part 2 - Entries (10 points)

- In the individual place view, display a list of entries related to the place (2), and a link to the new entry form for the place (2)
- A form to create a new entry for a place (2) containing a hidden field for the entry's `place_id` foreign key (1)
- Entries controller with new (1) and create (2) actions, which display a form and create an entry for a place

Use of external resources

- You are welcome to use external resources for help with the assignment (including colleagues, AI, internet search, etc). However, the solution you submit must utilize the skills and strategies covered in class. Alternate solutions which do not demonstrate an understanding of the approaches used in class will receive significant deductions. Any concern should be raised with faculty prior to the due date.

Hints

- The "root route" (i.e. the landing page where the user will go when there's no path specified) can be configured in the routes file with `get("/", { :controller => "things", :action => "index" })`. This is not needed, but will simply make your app feel more complete and avoids showing the "Yay! You're on Rails" page.
- Use the `<input type="date">` instead of `type="text"` when dealing with a date field (e.g. an entry's `posted_on` column)