Ruby on Rails Code Project

As part of our consideration for new hires to our Ruby on Rails dev teams, we ask that you complete the following project to the best of your ability. Please try to complete the project within two days.

Our goal for this project is to see:

- How you structure your code (following Rails conventions where they're useful, deviating *in a comprehensible, maintainable way* where they're not)
- How you write tests & what you choose to cover with tests
- Demonstrate end-to-end competency (given behavioral spec, how far can you take the project?)
- How you communicate your goals/decisions/reasoning, and ensure what you build is maintainable by someone *else* who might have to expand on this six months from now

Project Specs - Make a barebones messaging board

1. Users can register and then must be authenticated before logging in (we use Devise). Users should have an email (used for logging in), and a name at a minimum. You can add things like a description or "about me" if you'd like.
2. Users can create posts (title, body, author (user_id), etc)
3. Users can comment on other users' posts (Comments have post_id, body, author (user_id))
4. There is a posts index page that lists all posts (for entire site). This is the 'homepage'. This should display a list of posts with the title and the author's name
5. Order posts with most recent posts at the top of the page
6. There is a post show page, that shows a single post. Underneath the post are all the comments, ordered by date created (oldest at the top)
7. When commenting on a post, you have freedom as far as the layout goes, you can just have a little box that opens, and the user types in their comment. You can also use a modal, etc, whatever you want. It makes more sense to have the form for adding a comment be close to where the comment will end up.

Project Requirements

- Upload finished project to GitHub
- Use latest version of Rails 5 or greater
- Use Postgres for the database
- Use Ruby 2.6 or newer (we use 2.6.9 on our main project)
- Use Rspec to add tests that cover critical behavior
    - What to test & how to test is up to you
    - If you write feature specs, please use Capybara to simulate the user interaction
- If not specified, feel free to use any open-source gem(s) to get the job done

Bonus features (not required, but nice to have)

- Style using Bootstrap, Tailwinds, or a CSS library of your choice
- Add a modal for creating a new post

- Add a modal for logging in
- Paginate posts#index and/or posts#show (comments)
- Deploy project to Heroku (make an account for free)

Don’t hesitate to contact us if you have any questions about the assignment, or need guidance:
andy@brivity.com , anthony@brivity.com