For a red belt:
1. CRUD operations all working.
   a. Able to create new restaurants
   b. All restaurants are displayed upon visiting the website
   c. Able to leave reviews on each restaurant
   d. User may choose a restaurant and see all its reviews
   e. Able to edit each restaurant
      i. Edit form is pre-populated with the restaurant's existing data
   f. Able to delete each restaurant
2. Routing
   a. Root route displays all restaurants
   b. Route '/new' displays the form to create a new restaurant
      i. Successful creation and cancel redirect to the root route
   c. Route '/reviews/:id' displays all the reviews of a particular restaurant
   d. Route '/edit/:id' displays the pre-populated form to edit a restaurant
      i. Successful edit and cancel redirect to the root route
   e. Route '/write/:id' displays the form to review a particular restaurant
      i. Successful submission and cancel redirect to the restaurant's list of reviews
3. Backend validations
   a. All restaurants must have names of at least 3 characters
   b. All restaurants must have cuisines of at least 3 characters
   c. Error messages inform the client if the restaurant's name or cuisine is not meeting requirements
      i. Both on create and edit
   d. All reviews must have the user's name of at least 3 characters
   e. All reviews must have a description of at least 3 characters
   f. All reviews must have a number of stars between 1 and 5
   g. Error messages inform the client if the review is not meeting requirements

For a black belt:
1. Deployed, and not on Heroku
2. At least two of the following four features:
   a. Each restaurant's delete button is only enabled if the root component is loaded within 30 seconds of the restaurant's creation. Note: This does NOT mean you should be able to watch the button change from enabled to disabled. An enabled button would switch to disabled only if the component is reloaded after the 30 seconds have passed.
   b. Do not allow duplicate restaurant names in the database. Display an error message if the user attempts to create a restaurant with a name that already exists in the database.
   c. Nest the edit component so it appears alongside the list of restaurants without requiring routing. Upon successful submission or clicking cancel, have the edit component disappear from view.
   d. Sort the reviews by their stars in descending order so the highest rated restaurants are at the top.

Also:
1. Create a short video walking through the features of your app. Show your validations working. There is no need to speak or show your code.
2. Remove your github repo after deploying.

Don't worry about:
1. CSS
2. What the prompts are asking for. Just ask if you are unsure about what the goals are!