Terrier Rails Assessment

Overview

The purpose of this assessment is to demonstrate competence in the concepts, process, and tooling required to develop web applications with Ruby on Rails. The end result will be a basic application that stores technicians and work orders in a database, displays them on a scheduling grid, and allows the user to select slots in the schedule.

Deliverables

The result of the assignment should consist of:

1) A public git repo (github or otherwise) containing the complete source code of Ruby on Rails application that meets the requirements below.
2) Sufficient information to run the application on a new machine (complete Gemfile, Ruby version, rake tasks to run, etc.) - can be included in Readme.
3) A brief description of the approach taken to designing the application, problems faced, and possible future improvements.
4) (BONUS) The URL of the application hosted on a platform like Heroku or Render.

Background

In pest control, scheduled work is done by technicians at customer locations. An instance of work done by a specific technician at a specific location and date/time is called a work order (i.e. work orders are a join table between technicians and locations). Work order durations are stored in minutes.

For this assignment, a pared down data model with these three tables will be used to demonstrate the basic data storage and user interface needs of a work order scheduling system.

Requirements

The application should fulfill the following requirements:

1) Have database tables and model classes that map to the columns in the attached CSV files.
2) Provide a documented mechanism to load the CSV files into the database (e.g. rake task). This mechanism should be *idempotent*.
3) Render a single page containing a full-page scheduling “grid” for a complete day (all work orders occur on the same day), with:

    1) One column per technician, with the technician's name at the top.
    2) A column on the left containing labels for the time of day (one per hour).
    3) Time in the y-direction, with the earliest time at the top (below the technician's name).
    4) The time scale for each column should be the same.
    5) A block in the correct column/y-position for each work order. The block should contain the location name and city, and work order start time and price (**HINT: these blocks will *not* fit in a standard table layout**).
4) When the user clicks on an open space in the grid, a popup should appear alerting them of how much time is available between the previous and next work orders.

Technology

1) The application must be written in Ruby on Rails.
2) Any SQL database can be used to store the relevant records.
3) A rake task or tasks must be provided to import the data from the provided CSV files (rake tasks should be idempotent).
4) The view can be rendered with any combination of server-side or client-side technologies, but the interaction described in requirement 4 *must* be implemented in the client (i.e. it must trigger a client-side alert or popup, not load a separate page).