Assignment

Choose one of the following project outlines to complete your graded assignment. All of the projects have very similar underlying structures, and similar to the My Bucket List app, so pick the project theme that you enjoy the thought of the most. Take your time to read through the full brief and plan out how to approach completing each task. Some tasks are related to each other and a successful project is more likely with good planning up front.

Project options

Read through and choose one of the following three project options.

General notes

A User indicates someone who has identified themselves within the application by registering and logging in.

A Visitor indicates anyone visiting the website that has not identified themselves by registering and logging in as well as Users.

Option 2: Book reviews

Create an application that allows people to review their favourite books. Include features that allow:

1. About page

Visitors to open a simple static HTML page, about.html, that includes the title of the project option, a brief description of what the website is for and who built it. This page does not need to include any of the bootstrap styling indicated below.

1. Login

Visitors to register and log in as a User with their :email using a basic sign up and log in form. Use a model named User for this task. Users to create Reviews that includes information about the book: a :book_title, a :body, an :image_url for the book review, and an :author (i.e. the author of the book being reviewed). Use a model called Review for this task. The Review needs to be associated with the User that created it.

3. Index

Visitors to view the six most-recently created Reviews on the homepage / root path.

4. Search

Visitors to search for Reviews with a search term that will match on either :title or :author from a form on the homepage. The form should submit in a way that the URL for any particular search can be copied and sent to others.

5. Show Review

Visitors to view all of the details for a Review plus all Comments related to that Review on one view. The :title, :body, and :author should be displayed as text and the :image_url used to display the image at that URL. This view will be linked from the individual Reviews displayed on the search results view.

6. Edit review

Users to edit all details of a Review by clicking on a link from the “Show review” view.

7. Save reviews

Users to save new or updated Reviews only if a :title is provided i.e. :title is required, and that an :author is no longer than 100 characters in length, although a :author is optional. Any validation errors should be displayed helpfully for User.

8 Add comments

Users to add a comment on the “show review” view when they are signed in. Use a model named Comment for this task. The Comment needs to be associated with the Review that it was created from and the User that created it.

9. Bookmark reviews

Users to add any Review created by any User to their “bookmarked Reviews list” from the show Review view. It should not be possible to add a Review more than once. Their bookmarked Review list should be available on a separate view, listing all of the Reviews that they've saved. A link to the form to create a new Review needs to be included on this view.

10. Site navigation

Visitors to have a link to the homepage, the about page, and the login/register view in the header of every view. Users to have a link to the homepage, the about page, and their bookmarked reviews list in the header of every view.

Project requirements

The outlines above don't go into as much detail as previous course projects: this is to assess your ability to interpret the project brief and create the appropriate models, controllers, views, and routes based on the guidelines below. In particular, the project submission must demonstrate all aspects of developing with the Ruby on Rails framework taught in this course, including:

- Use of views, partials, and layouts.
- Use of the Rails asset pipeline for CSS rules and images (see CSS, JavaScript and app design below for guidance on what CSS rules you should use)
- Use of the Rails router, including resources.
- Use of controllers, controller actions and the application controller.
- Use of models, associations, and validations.
- Use of automated system and model tests.
- Use of Active Record and database interactions.
- Use of migrations and managing database table structures. It's also necessary to demonstrate the following principles of software development:
- Good source control management with git.
- Good code structure and reuse.
- Good naming and organisation of automated tests.
- Good code formatting throughout a project codebase.
- CSS, JavaScript and app design

The project does not need to demonstrate CSS techniques, document structure with HTML, or interactive front-ends with JavaScript beyond anything indicated in the project briefs above. The visual layout of views must be clear in order for assessors to understand the functionality but beyond that there are no grades associated with style, HTML structure or use of CSS.

The Bootstrap CSS library, version 4, has been included in the references section of this unit. Bootstrap is a comprehensive library of components that make ist easier to build prototype applications without the need to work on lots of custom CSS. Use the Bootstrap library to give the application for your project a solid layout and structure. You'll need to use the documentation at Twitter Bootstrap in order to know what CSS classes can be used for what visual elements.

Bootstrap only provides components rather than a finalised and comprehensive style guide so stick to simple usage, and focus on making a working app that demonstrates your Rails knowledge! The Card component is useful for displaying details of any of the models needed for the project outlines above. The simple version of the navbar component is useful for quickly creating a header and navbar for your application. Note that the Bootstrap is a large CSS and UI component library, and contains a lot of things that will not be applicable to this project, particularly any component that uses JavaScript.

Submission and grading This project forms 100% of your grade for the course. It will be graded based on interpretation of the project outline chosen, and correct and appropriate use of the different aspects of Ruby and Ruby on Rails covered in this course.

Once you have completed your project, you should push your code to a github repository and submit the GitHub URL for it.

If you have reached this point in the Web Application Development program and not yet booked a one-on-one, you will be asked to do so. This is because we are required to have a minimum amount of face-to-face communication with each learner as they progress through the program. If you have any personal circumstances that make a video one-on-one via the Internet difficult, please let us know.