Mid-assignment Submission
A0180340U Rayner Lim Ri Han

Source on Github: https://github.com/rlrh1996/rails-react-todolist
Live Application on Heroku: https://rails-react-todolist.herokuapp.com/

Use Cases

Task 1 - Managing Tasks

- Create
  - Create a new task with task description
- Read
  - View list of all tasks with task descriptions
- Update
  - Toggle whether a task is completed or not
  - Edit a task's description
- Delete
  - Delete a task

Task 2 - Categorizing Tasks

- Create
  - Create a new task with task description
- Read
  - View list of all tasks with task descriptions and tags
  - Filter tasks by active, completed, and tags
  - Search filtered tasks by task description
- Update
  - Toggle whether a task is completed or not
  - Edit a task's description
  - Add or remove a task's tags
- Delete
  - Delete a task

Execution Plan

1. Learn the three R's
   a. Ruby
   b. Rails magic
   c. Relational databases
2. Design the backend application structure
   a. RESTful API using JSON
      i. GET - read - index in Rails

ii. POST - create - create in Rails
iii. PUT - update - update in Rails
iv. DELETE - delete - destroy in Rails

b. Database schema
i. Completed - boolean
ii. Description - text
iii. Tags - acts-as-taggable-on gem magic

3. Design the frontend application user interface (UI)
   a. Single Page Application (SPA) using React for great user experience
   b. Two-column layout similar to Wunderlist
      i. Small left sidebar for searching and filtering
      ii. Large main area at right to add new task and view all tasks as rows
   c. Styling using Bootstrap 4 and Font Awesome for ease and efficiency
4. Create Rails application and set up version control
   a. rails new todolist --webpack=react --database=postgresql
   b. Generate Rails model, setup acts-as-taggable-on gem, seed & migrate DB
   c. Setup Git and Github
5. Implement RESTful API using JSON
   a. Create Rails controller
   b. Test with Postman
6. Implement barebones Task 1 - Managing Tasks
   a. Implement minimal working UI
   b. Do informal testing and fix bugs
   c. Overhaul UI to proposed design in preparation for next step
7. Implement Task 2 - Categorizing Tasks part by part
   a. Search by task description, filter by active or completed
   b. View tags, filter by tags
   c. Edit tags
   d. Do informal testing and fix bugs
   e. Make minor UI improvements to enhance user experience
8. Deploy to Heroku
   a. Automatically deploy from Github repo
   b. Do informal testing and fix bugs

Challenges

1. How to do proper testing?! What is code coverage? How to write test cases?
2. Learning about relational databases (beyond what is needed for this assignment)
3. Learning about Rails magic and just accepting that it works
4. Learning language features unique to Ruby like symbols and blocks