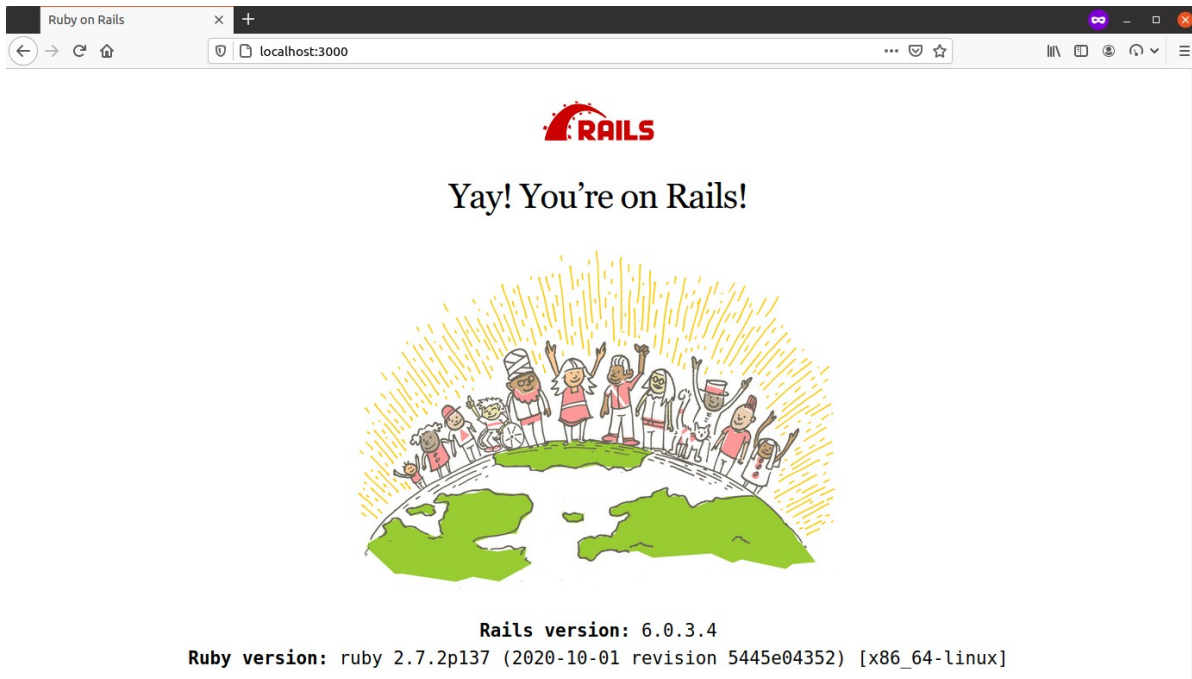


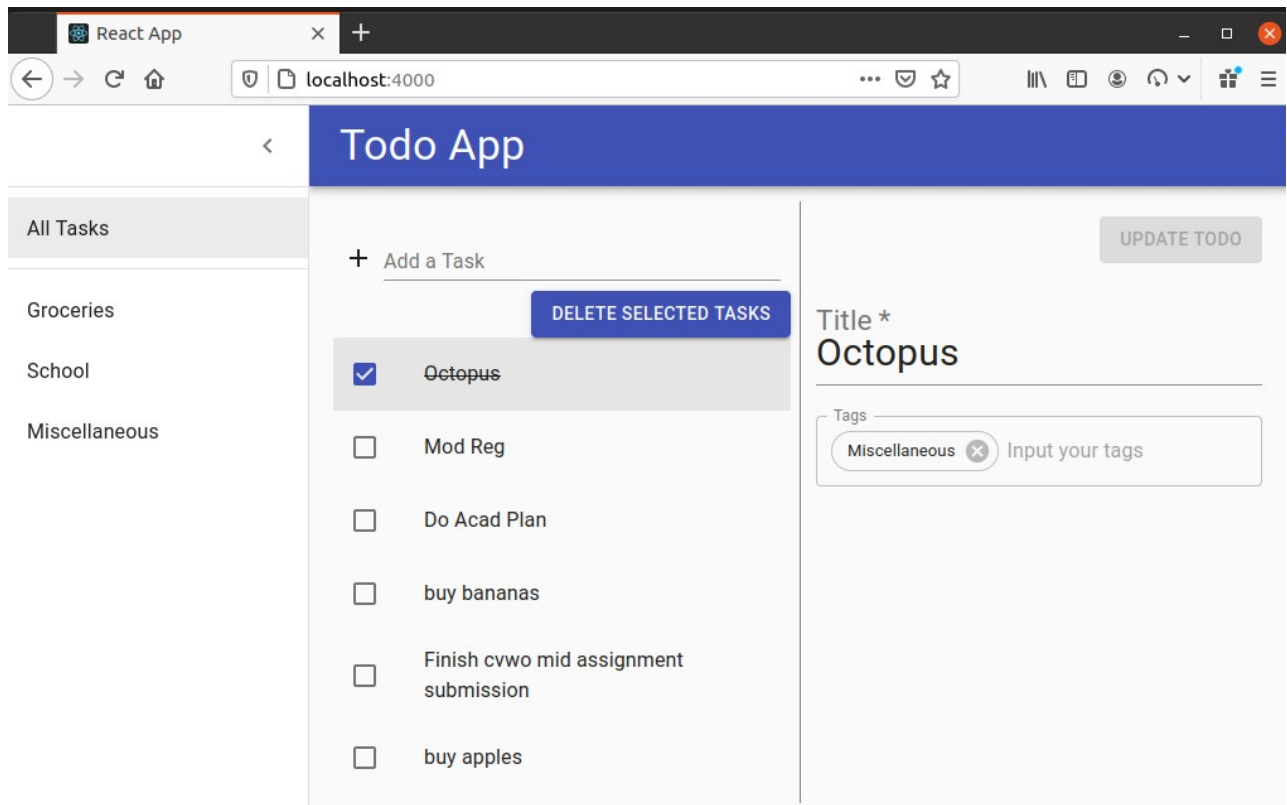
Name: Elizabeth Chow Ting San A0192318A

CVWO Assignment Mid-Assignment Submission

Default Rails startup Page



The Todo App is built with RESTful API architecture with Ruby on Rails as the backend and React Typescript as the frontend. Material UI was used for styling the page.



Task 1 – Managing Tasks

Basic CRUD Operations – Create, Read, Update and Delete Todos

CreateTodo

User enters task name (maxLength = 50) in the input and press 'Enter' to save the task in the postgres database. React will send a POST Request to Rails API to create the entry.

GetTodo

Upon starting the app, all the tasks that were created (and not deleted) will be rendered on the page. React sends a GET Request to Rails API and renders the JSON object received.

UpdateTodo

When the checkbox of the item is checked/unchecked, the item will be striked-through/no strike-through respectively. React sends a PUT Request to Rails API to update that the task is done.

EditTodo

User is able to change the title and tags of the task and when the 'Update Todo' Button is clicked, the new title of the task will be saved. Reacts sends a PUT Requests to Rails to update the name and tags of the task.

DeleteTodo

User can checked on a single/multiple tasks and clicked on 'Delete selected Task' Button to delete the task(s) when it is completed. React sends a DELETE Request to Rails API to delete the tasks.

Task 2 – Categorising Tasks

The screenshot shows the 'Todo App' interface. On the left is a sidebar with a back arrow and four categories: 'All Tasks', 'Groceries', 'School' (which is highlighted), and 'Miscellaneous'. The main content area is divided into two sections. The left section contains a list of tasks: 'Finish cvwo mid assignment submission' (with an unchecked checkbox), 'Do Acad Plan' (with an unchecked checkbox), and 'Mod-Reg' (with a checked checkbox). Above this list is a blue button labeled 'DELETE SELECTED TASKS'. The right section is a form for editing a task. It has a title field with the text 'Finish cvwo mid assignment submission' and a label 'Title *'. Below the title is a 'Tags' section with a search bar containing 'School' and a button 'Input your tags'. A blue button labeled 'UPDATE TODO' is in the top right corner of the form area.

The tags database is created with Ruby Gem ActsAsTaggableOn.

getTag

User will be able to see all the tags created at the side panel upon starting the app. React sends a GET Request to Rails API to retrieve all the tags that contains tasks related to the tags.

getSelectedTag

User will be able to see all the tasks related to the selected tag when the tag name is clicked. React sends a GET Request to Rails API to retrieve all the tasks that contain that particular tag.

Possible Improvements for Next Submission

1. Improve the CreateTodo function

The task can be automatically tagged with the selected tag when the user adds a task in selected tag page.

2. Add a due date option for the task

The user can set the deadline of the task and possibly set a reminder for it.

3. Add a search function

Add a search bar so that users can search for a particular task.

4. Abstraction

Abstraction can be done for API calls using React Hooks.

5. Hosting

The app will be hosted on Heroku.

6. Tags

User can modify a tag or delete the tag completely from the database. Currently, it cannot be done.

7. Styling