National University of Singapore
School of Computing
Computing for Voluntary Welfare Organisations (CVWO) AY2018/19

Assignment: **Riding on Rails**

Issue date: 8th December 2018
Due date: **31th December 2018 (mid-assignment submission)**
27th January 2019 (final submission)

1 Overview

In this assignment, you will learn about the programming skills required for CVWO's summer project and go through a website development cycle. The focus of this assignment is to promote proper code structure, coding techniques and familiarity with various development tools. You are expected to build up a good web development foundation through this assignment, and prepare yourself for one development cycle during your stint over the summer.

Let's begin...

Reminder:	Please read the entire assignment before starting.

Aside from web development foundations, this assignment mainly aims to familiarize you with Ruby on Rails, an open source Ruby framework with strong MVC patterns. As CVWO uses Rails as its main framework, it is important to have a good understanding of this framework in order to work effectively on the projects.

Tip:	If you do not know what MVC is, it is strongly recommended that you do your own reading before attempting this assignment. During the course of CVWO and beyond, **independent learning is the key** to staying ahead.

2 Assignment

2.1 Setup

Navigate to the folder in which you want your Rails app to be in and run this command in the terminal: `rails new application-name`

You can replace application-name with any name you want.

Installation Guide:	Section 3 contains useful information to aid you in this assignment. Included is a tutorial on how to install Rails.

For this assignment, you are allowed to use *any* gem (Ruby plugin) to make your life easier.

2.2 Task

2.2.1 Task 1 - Managing Tasks

Implement a todo manager to keep track of all the tasks that you have to do. As a start, your todo manager should be able to perform basic CRUD operations. CRUD is an acronym for Create, Read, Update and Delete. They are the basic functionality that your application should have.

For styling, Rails does not have its own default themes. This means that you have more flexibility in either designing your todo manager from scratch using CSS, or you could choose from a variety of front-end frameworks to make your life easier.

2.2.2 Task 2 - Categorizing Tasks

Implement a tagging system to organise your tasks so that they are easy to search for. Since we are all busy people, it is very likely that your todo manager would fill up fast. Hence, it is important to be able to organize your tasks and search through them.

2.2.3 Optional Tasks

You are free to implement other features to improve the overall user experience, e.g. AJAX. Extra credit might be awarded, depending on the completeness of your implementation.

If you are unsure of what else to add, consider the objective of this application (apart from pedagogy!) Think of various use cases that users might want to use your software for. What else might such a user want?

2.3 Submission

For the mid-assignment submission, please provide the following:

1. A `README` file containing your name and matriculation number.
2. A screenshot showing that you have successfully installed Rails. Just the default startup screen will do.
3. A short write-up on the basic use cases and your execution plan. You can also write down your suggestions and tell us the problems you are currently facing. This write-up must be no longer than 3 A4 sides, with a 12pt. Georgia or equivalent font face. Please submit in pdf format.

For the final submission, please provide the following:

1. A `README` file containing your name and matriculation number.
2. Source code: maintain the default file structure of the Rails application as much as possible.
3. A short write-up on what you feel about your accomplishments in this assignment as well as a short user manual. This write-up must be no longer than 3 A4 sides, with a 12pt. Georgia or equivalent font-face. Please submit in pdf format.
4. Proof of working application: It is highly desirable to have a working copy accessible from the Internet. Alternatively, submit a database Dump with your source code.

2.3.1 Git

It might be your first time developing any piece of code which is non-trivial to write and maintain. As such, you might like to store copies of your work over time. This is called versioning. An additional function software engineering teams require is the ability to share code as it is being written. One such software used by software engineers (and CVWO) to solve both problems is Git. Git is however far more able than this and more information can be found on the Wikipedia article.

Important:	It will save you time and effort to use a versioning software since you will most likely run into bugs as you change (supposedly unrelated) pieces of code in your application. As you will be using git on your first day of CVWO, versioning your assignment using git is **compulsory**.

Create a Git repository (either on GitHub or GitLab) and email weineng.a@gmail.com the link to your repository. We will clone and grade your code from there.

You must push your deliverable to the repository by the given deadlines. Work-in-progress code is OK for the mid-assignment submission.

2.4 Grading Scheme

We grade your assignments based on how much effort you have put into the assignment. The key point of this assignment is for you to demonstrate what you have learned in this process. The grading is flexible. This grade will be considered during the application process for CVWO.

3 What you need to know

> *"Babies are always more trouble than you thought – and more wonderful."*
>
> —Charles Osgood

The following sections are provided to aid you in your assignment. They include modern programming languages and technologies that are widely used in the industry.

3.1 Ruby on Rails

> *"Ruby on Rails® is an open-source web framework that's optimized for programmer happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration."*
>
> —From the homepage of Rails

3.1.1 Introduction

Ruby on Rails is a free and open source web framework. As the name suggests, this framework is written in Ruby. Notice that at this point, we have not explicitly asked you to learn the language. This is because the language is relatively easy to pick up and you should be able to get a sufficient grip on the language by yourself.

Rails provides a reusable boilerplate code which reduces the complexity of the code. There are also many auxiliary libraries that we can use to speed up the development process. Furthermore, it is also relatively easy to deploy as it is supported by many hosting service providers. Hence, we have chosen to utilize Rails for its suitability in carrying our projects further.

Rails usually ship with a default web server and database adapter, Puma and SQLite, which should be sufficient for this assignment. We do not impose any restrictions on the choice of either for the purpose of this assignment.

Tip:	Although not needed in order to complete the assignment as you could learn by trial and error, learning Ruby before attempting the assignment would make your life much easier. Unlike other frameworks such as PHP, Ruby is much easier to pick up although there are some gotchas that you may or may not run into during the assignment. There are many good online resources available.

3.1.2 Installation

Step 1: Install the latest version of Ruby by following instructions from these links: `https://www.ruby-lang.org/en/downloads/`. If you are using Windows, you may use the Window Subsystem for Linux (WSL) to run windows. Alternatively, download your favourite virtual machine running Linux.

Step 2: Once Ruby is installed, test it out by running interactive Ruby with the command `irb`. If you are unable to do so, please check your Ruby install.

Step 3: After you have ensured the validity of your Ruby install, set the path `/ruby_root/bin` to your environment variables.

Step 4: Next, proceed to install Rails by running the command `gem install rails`.

Step 5: Navigate to the folder that you want to store your new Rails application in.

Step 6: Run the command: `rails new application-name`. Please replace application-name with the actual name of your application. If this works without giving you any problems, then you would have been done with the setup of your first Rails application.

Now Rails is up and running on your local machine!

Although having a more in-depth lesson on Rails would be appropriate here, staying true to one of Rails' major philosophy, DRY, we shall leave this task to a more established resource from the community, the Rails Guide `http://guides.rubyonrails.org/`. The following chapters are compulsory reading and they would help greatly in the assignment given later.

- **Chapter 1**: Getting Started with Rails
  `http://guides.rubyonrails.org/getting_started.html`
- **Chapter 2**: Active Record Basics
  `http://guides.rubyonrails.org/active_record_basics.html`
- **Chapter 3**: Active Record Migrations
  `http://guides.rubyonrails.org/active_record_migrations.html`
- **Chapter 4**: Active Record Validations
  `http://guides.rubyonrails.org/active_record_validations.html`
- **Chapter 5**: Active Record Callbacks
  `http://guides.rubyonrails.org/active_record_callbacks.html`
- **Chapter 6**: Active Record Associations
  `http://guides.rubyonrails.org/association_basics.html`
- **Chapter 7**: Active Record Query Interface
  `http://guides.rubyonrails.org/active_record_querying.html`
- **Chapter 8**: Layouts and Rendering in Rails
  `http://guides.rubyonrails.org/layouts_and_rendering.html`

- **Chapter 9**: Form Helpers
  http://guides.rubyonrails.org/form_helpers.html
- **Chapter 10**: Action Controller Overview
  http://guides.rubyonrails.org/action_controller_overview.html
- **Chapter 11**: Rails Routing from the Outside In
  http://guides.rubyonrails.org/routing.html

3.2 Relational Database

> **Suggestion:** Although you will not likely be writing any SQL for this assignment, it is important to have at least a basic understanding of databases.

3.2.1 An overview of relational databases

Relational databases are databases that store data in tables, each containing a number of columns and rows. It is called ”relational” because tables are linked to other tables through foreign keys. For example, the Singapore Government's address database may use the NRIC/FIN of individuals to link addresses in the `addresses` table to names in the `names` table. In this case, the column containing NRIC/FIN in the `addresses` table is called the foreign key as it references the primary key in the `names` table.

In this assignment, we will be going through fundamental database concepts. Usually, an application will use one database, with several applications sharing the same database server. For instance, if you and a friend each had a blog, each blog needs one database but the same server could have two databases defined, one for each blog.

At the highest level, a database contains a schema. A schema is a blueprint of your tables, containing their structures and relationships.

Tables can contain one or more columns, each defining an attribute that requires stor- age. (For example, a students table containing students' personal data might contain five columns: `matric_no`, `name`, `address`, `phone`, `date_of_birth`. In MySQL, columns have types: in this case, name is of type text (`varchar(255)`) and `date_of_birth` is of type `datetime`).

Keywords to look up: SQL, primary key, foreign key, Entity-Relationship Diagram. Try to get the overview before diving into details.

Actual information is stored as rows in a table. Each row must be uniquely identifiable; if two rows cannot be distinguished from each other, then there is no way to change the content or delete one of the rows. Therefore, it is common practice to dedicate one column that is guaranteed to be unique for each row. We call such a column a primary key. In this instance, the `matric_no` is probably a good choice to be the primary key as it uniquely identifies a row, since no two students can share

a matric number. You will be prevented from inserting a new row when another row with the same primary key already exists in the table.

Relations indicate relationships between two tables. For example, the `home_faculties` table may contain two columns: `matric_no` and `faculty`, indicating a mapping from a student to her faculty. We can link this to the students' table by using `matric_no`. As discussed earlier, `matric_no` in the `home_faculties` table is called the foreign key. The foreign key must map into the primary key column set of another table. Note also that students may become members of two faculties when doing double degrees.

What happens if you need to relate two tables together to retrieve an attribute? For instance, you might have a student in the double-degree programme and you need to retrieve his CAP for each of his faculties.

Other important concepts include*indexed columns* (where searching within the column is fast, at the expense of increased time required for modification), *unique keys* (which enforces uniqueness among values for non-primary key columns), and relation cascading (where deleting a row from a table can automatically update/delete all related entries in tables which reference this key).

This is called an Entity-Relationship Diagram

After this section, you should be ready to produce a schema for your application. Consider how efficient your schema will be: How complex will it be to access the most commonly accessed data? Think about the number of queries and the number of tables accessed to complete a single user query. Your schema should be graphical, with the table names, column names/types, primary keys, and relationships clearly indicated.

Aspiration:	Draw your application's database schema

3.3 HTML and CSS

Below is a list of HTML tags you will be using frequently.

```
<a>,<body>,<br>,<button>,<div>,<em>,<form>
<head>,<h1> - <h6>,<hr>,<html>,<img>
<input>,<label>,<legend>,<li>,<option>,<p>,<script>
<strong>,<style>,<table>,<tbody>,<td>,<textarea>,<tfoot>,<th>
<thead>,<title>,<tr>,<ul>
```

Please go through the tutorial from HTML Dog. You can experiment with various HTML tags on JSBin. We will not formally introduce CSS as it is better to learn from practice. Later sections will cover CSS briefly as the need arises.

Suggestion: Modern browsers all come with a debugging inspector for HTML, CSS and JavaScript. You should decide on which browser you will be primarily debugging and testing with. Note however that there are compatibility differences between browsers and you **must** test your application using different browsers. For CVWO, we usually recommend VWOs to use Firefox or Chrome; however, certain staff members still prefer to use Internet Explorer, in which case we try to ensure that they are using Internet Explorer 10 or above.

3.4 JavaScript

If you did not take CS1101S: Programming Methodology, Eloquent JavaScript is a good starting point. You are not expected to master JavaScript, however, you must demonstrate basic understanding. If you already know basic JavaScript syntax, you might like to start from Chapter 11 - Web programming: A crash course.

Don't just read the book chapter by chapter. Practice is the key! Alternatively, you can refer to existing JavaScript examples on the web, and use the book to clarify any doubts.

3.5 Tools

Various tools can be used for web application development. In all cases, you should always find one that suits your working style. As such, there is no one answer to "which is the best IDE to use". Nonetheless, strictly speaking, any text editor will suffice. However, the following software is used by some of our CVWO members:

- Visual Studio Code
- RubyMine: Note that if you are a student, you are eligible for a free copy of the paid version of the app. See here.
- Sublime Text
- Vim

3.6 Best Practices

> *"If you give a hacker a new toy, the first thing he'll do is take it apart to figure out how it works."*
>
> —Jamie Zawinski

3.6.1 Abstraction and Modularisation

People who are new to web development usually end up with an unreadable mess. As such, as a project grows in size and scope, you may find it increasingly difficult to maintain and add new features. This is undesirable, especially for large projects.

The key insight here is to abstract common patterns away and reuse them where necessary.

For instance, you may decide to use the same headers and footers for every page in your app. These blocks can be abstracted into separate files and included where required instead of copying and pasting them everywhere. A logical structuring of your files helps increase its maintainability and encourages future code reuse. You may want to take a look at some websites and consider how you could incorporate abstraction.

3.6.2 Documenting Your Work

Programming is all about communicating computational processes. This means that your code should be clear and easily understood. Remember that code is read many more times than it is written!

Help others comprehend your code by including clear comments. However, having good comments does not excuse writing bad code. A consistently high quality of maintainable code is expected of you so that subsequent batches of students are able to work on your project. Some tips:

- Give sensible names to variables and functions so that their contents and goals are easily understood.
- Use standard algorithms where available. Mark modified algorithms as such.
- Use the standard library of the language you are using. JavaScript and Rails come with a large standard library that accomplishes many things. If the functionality you require is not in the standard library, find a well-maintained third-party library that accomplishes what you need. Write your own implementation only as a last resort.
- Don't over clutter your code with unnecessary comments. That means you don't have to comment every single line. Comments are excessive when it's blatantly obvious what the code does. A comment on top of each function describing what it does (often) makes your code more comprehensible than multiple in-line comments.

3.6.3 Additional Readings

- 10 HTML Tag Crimes You Should Not Commit
- DOM Manipulation Library: jQuery

4 Optionals

> *"For a long time it puzzled me how something so expensive, so leading edge, could be so useless, and then it occurred to me that a computer is a stupid machine with the ability to do incredibly smart things, while computer programmers are smart people with the ability to do incredibly stupid things. They are, in short, a perfect match."*
>
> —Bill Bryson

4.1 Cron

Cron is a time-based job scheduler in Unix-like computer operating systems and enables users to schedule jobs (commands or shell scripts) to run automatically at a certain time or date. It is commonly used to automate system maintenance or administration, though its general purpose nature means that it can be used for other purposes, such as connecting to the Internet and downloading email.

Set a Cron Job under Linux. This example is to create a cron job to automatically backup the database.

Inside terminal:

```
> sudo crontab -e

// START THE TOOL TO SET THE CRON JOB

// EVERY DAY RUN THE BACKUP SCRIPT

0 0 * * * bash backup.sh
```

Inside `backup.sh`

```
cd /root/autobackups

mysqldump -u root -DatabasePassword DatabaseName > FileName
```

You are encouraged to find novel uses of Cron for your application. Backing up your database daily is merely one of the endless possibilities that Cron allows you to implement.

4.2 Hosting

During the development phase, you will likely be running and hosting your application on your local machine, therefore only you will be able to access it. Deployment means making the application available for use to the public. To do so, you need a server to build and run your application and to accept client requests. In addition, a domain name is necessary so that people can access your application through an URL.

For this assignment, you are encouraged to host your final product on **Heroku**. Heroku is a platform as a service (**PaaS**) which provides an easy-to-use platform

to deploy a web application. After a proper setup, you can deploy your application simply by `git push heroku master`. This tutorial teaches you how to set up a Rails project on Heroku.

You need to register a Heroku account first, which requires your credit card information. For this assignment, the free tier should be more than sufficient.

Alternatively, you can use other cloud services such as AWS Elastic Beanstalk if you wish.

The end:	Good luck and have fun!