




School of Science

COSC2675 Rapid Application Development

Assignment

	Assessment Type: group assignment Submit online via Bitbucket or Github Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
	Due date: end of Week 11 ; Deadlines will not be advanced but they may be extended. Please check Canvas → Assignments → Assignment for the most up to date information. As this is a major assignment, a university standard late penalty of 10% per each working day applies for up to 5 working days late, unless special consideration has been granted.
	Weighting: 40 marks/100 (note: that includes demo marks associated with this assignment)

1. Overview

The objective of this assignment is to train on high level web development skill including task analysis, fast development, testing and team collaboration. The different stages of this assignment are designed to gradually introduce different scales of development.

Develop this assignment in an iterative fashion with good time management practice. You should get started now.

If there are questions, you must ask via online forums or similar communication channels. You should phrase your question in a general manner. The teaching team is not supposed to debug for you. Do not post your source code. Small code snippets are fine.

2. Assessment Criteria

This assessment will determine your ability to:

1. Independently analyse the task requirement and formulate a development plan.
2. Systematically develop a complete web application.
3. Follow Ruby and Rails coding convention as described in the course materials.
4. Follow agile style of development.
5. Test application regularly and develop the application based on testing.
6. Document code when necessary
7. Use external gems or libraries
8. Control versions and stages of the development
9. Collaborate effectively with a small team.
10. Meet deadlines.

3. Learning Outcomes

This assessment is relevant to the following Learning Outcomes:

1. Programming capability using Ruby
2. Development capability of a full-stack application based on Ruby on Rails framework.
3. Knowledge and experience of industry level software engineering practices e.g. TDD, Agile.
4. Understanding and experience of team development with tight time constraints.

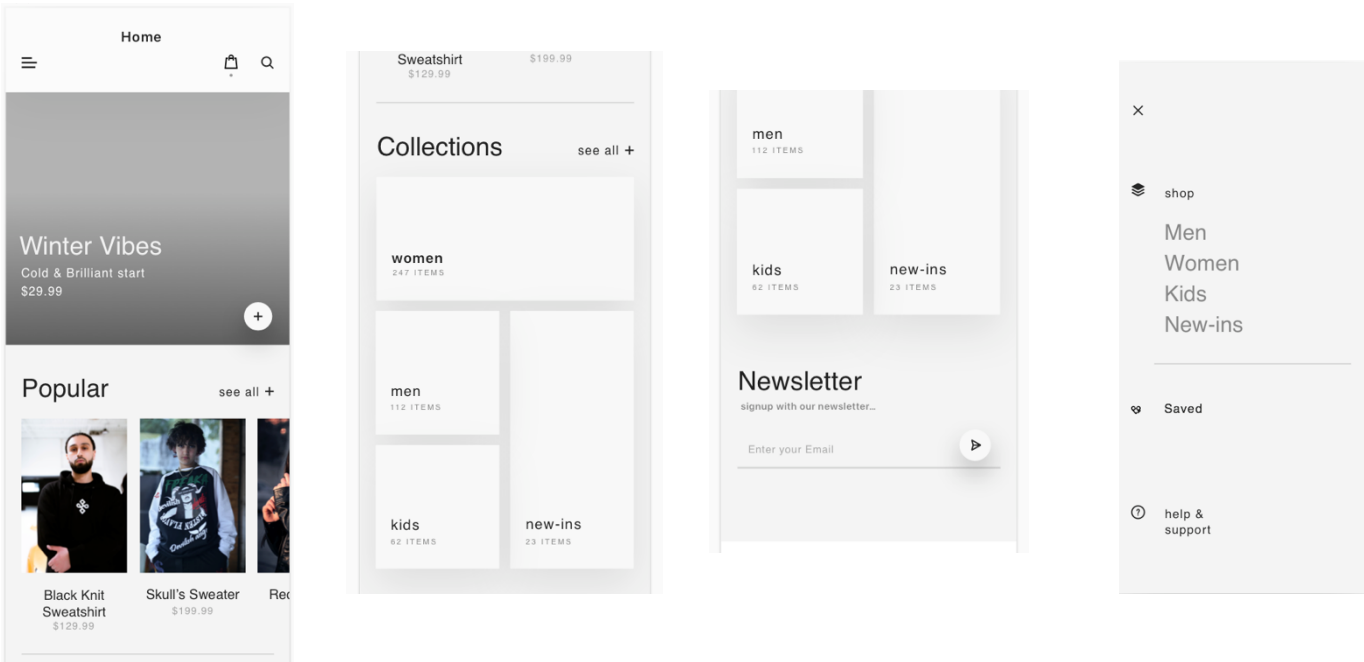
4. Assessment details

Please ensure that you have read sections 1-3 of this document before going further. Your code must meet the following code and documentation requirements.

Suppose you and your partner founded a startup which is an online fashion store. You need to build a web application for the store. The web app should be mobile browser friendly so perspective customers can shop at ease, mainly on a phone. Because of the limited budget and limited time, your team decided to build a MVP using Ruby on Rails.

You can simulate mobile interface using device toolbar on your computer browser.

===== PASS LEVEL =====



A new visitor can land on the homepage of your store as shown above. The top right has clickable icons for shopping bag and search functionalities (required in higher levels). The top left is a menu button which leads to a drop down menu/side bar (see above). Help & support is simply a static page.

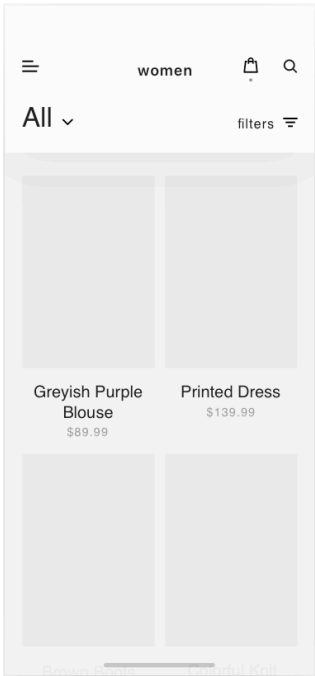
Top banner: The front page will randomly show an item from the shop collection, for example “Winter Vibes” in the previous page. When this item is clicked or the “+” button is clicked, the item is added to a “Save List”, then another randomly selected item will replace it. The front page will also show “Popular” items which are items that are frequently saved or purchased by customers. For PI

level, you can randomly assign a popularity score to these items. You can assume the width of the page never changes. An item here can be clicked to join the “Saved List”.

Collection: Visitors can see the full collections which are categorized into four groups: women, men, kids and new arrivals. Visitors can see all collections by clicking the top right button in the collection section, or see individual category by clicking that category on the frontpage or in the menu. The total number of items in each category is shown. Each item takes the same space to show a photo, the name and price. “New arrivals” are items stocked within three months. An item can be listed under multiple categories.

Saved List: When an item is clicked, it is saved in the “Saved List”. If it is clicked again, then it is deselected. You may add a tag or icon on the item to show that. “Saved List” is stored locally. Visitors can see their list when revisit the site (from the same device).

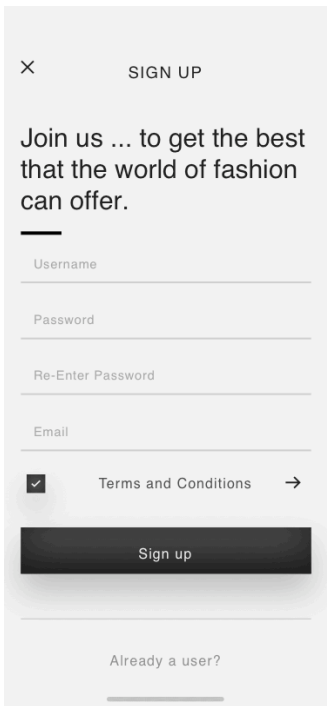
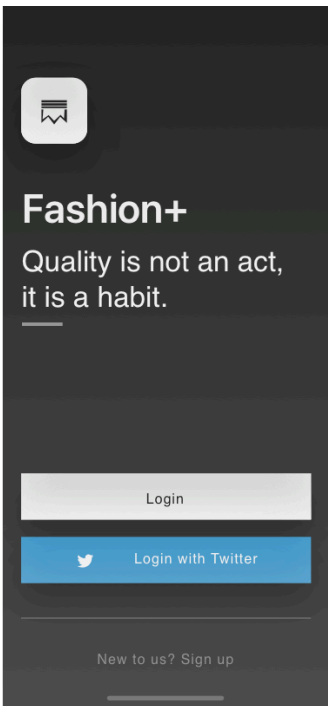
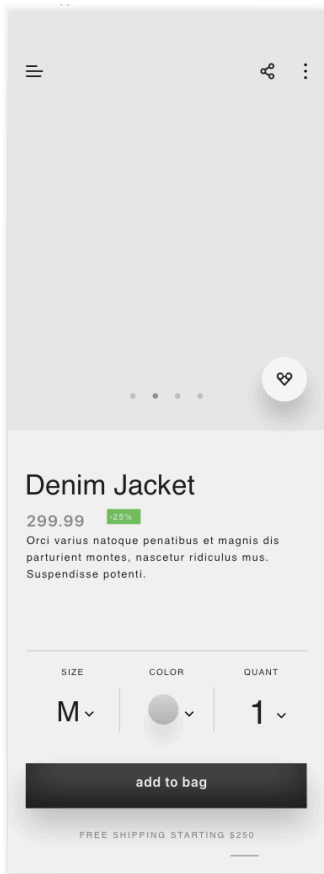
Newsletter: a visitor can subscribe the newsletter. The entered email must be valid so input like x@y@z, test.com etc should be rejected. A text-only test email should be sent to the enter email, e.g. your own RMIT student email.



===== Credit LEVEL =====

NOTE: You should only attempt this level when you have completed PASS level.

Product detail: At Credit level, a visitor can click the item in the saved list. The details of the product will appear. Each product may have multiple images which can rotate on the top. Size, color and quantity can be selected (without effecting the unit price) before adding to the shopping bag. When the “add to bag” button is clicked or the bag icon at the top right corner of the homepage is clicked, you app will prompt the user to login or to register.



Login: After login, the home page will appear the same as that in Pass level. Twitter users can login with their Twitter account. There is a “[Forgot password?](#)” option on the login page to help registered users (for HD+ levels).

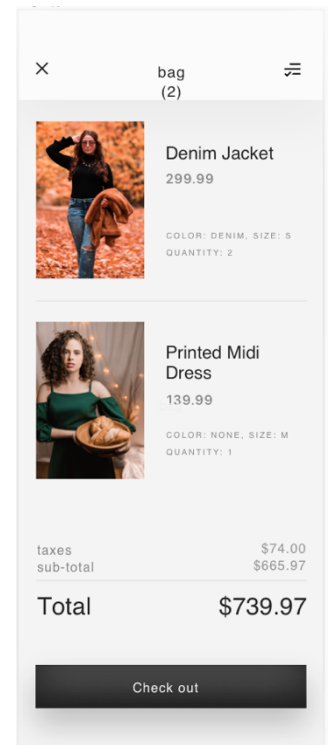
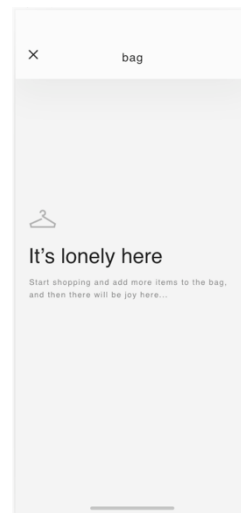
Signup: The login page is connected to Signup page via link “[New to us? Sign up](#)”. The user must provide name, two identical password and a an email address to sign up. If the visitor is already registered, then he/she can click the “[Already a user?](#)” link for the login page (see below).

Input validation: When a new user signs up, you need to validate the input.

- The email address must be valid as <>@<>.<>
- The initial pass word should be in between 8-20 characters, excluding special characters.
- The confirm password must be identical to the initial password.

Checkout: logged-in user can view the shopping bag. The total number of items is displayed on the top. When hit the checkout button, the bag will be emptied. You can assume the transaction is always successful. Your app can either stay at this empty page or redirect to the homepage.

Your app should support user “**RAD**”, “**rad2021rmit@gmail.com**” with password “**Rails2021**”.



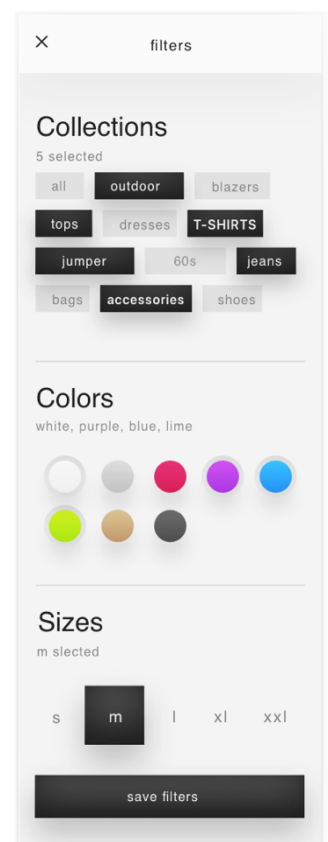
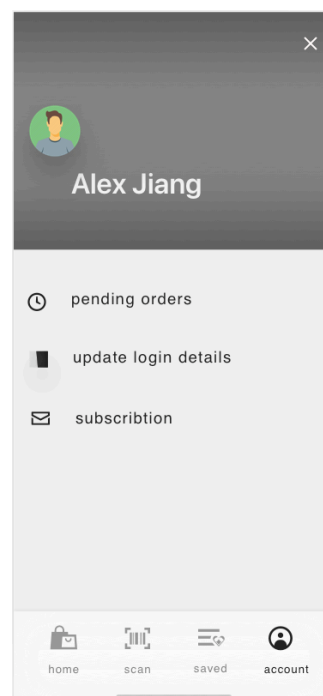
===== Distinction LEVEL =====

NOTE: You should only attempt this level when you have completed CREDIT level.

Footer: At distinction level, your app always display a footer showing four options: home, scan, saved and account. Scan is a functionality for 90+ level. “Saved” shows the saved list (with the footer). Account is the user profile page (see right).

Profile: this page shows the Gravatar profile (see info re Gravatar in tutorials or search online). “[Pending orders](#)” shows the shopping bag. “[Update login details](#)” allows user to update their password and email address. Same validation rules apply here as in sign-up. “[Subscription](#)” simply turns on/off newsletters. User can logout from this page.

Persistence: a logged-in user may leave the app with items still in the bag. When he/she logs in again, these items should remain in the bag. A user could try to add an item in the bag when not logged



in. The app will prompt the user to login (see CR level). That item should be in the bag after login. Items saved in the “Saved List” before login should remain as well.

Filter: A visitor/user can set a filter while viewing the collection or a category. The filter allows user to select several pre-defined tags, color and size (see example on Page 4). Note multiple tags can be selected simultaneously except the “all” button, as it ticks every tag or deselects all of them when pressed again. Multiple colors can be selected or deselected. No need to change image of the item according to the color selection. Size is mutually exclusive, only one can be selected.

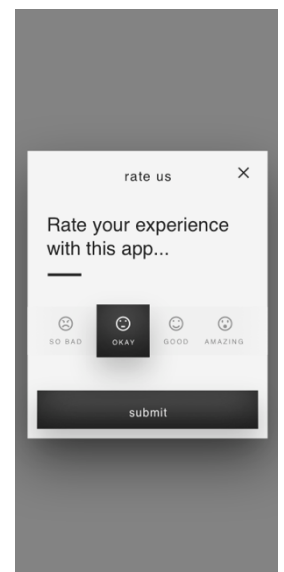
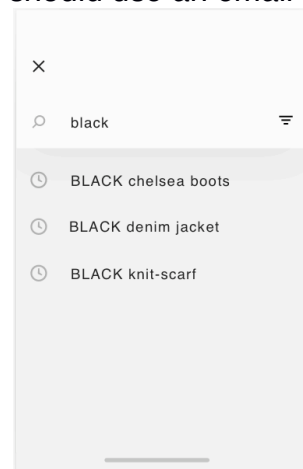
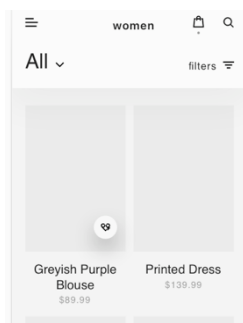
Popularity: The popularity score at this level and above is real. It is the sum of the number of times the item appeared in the saved list and the number of times it appeared in the shopping bag.

===== 80+ LEVEL =====

NOTE: You should only attempt this level when you have completed **DISTINCTION** level.

Forget password : the login page at this level can help user to retrieve the password simply by sending a temporary login link to the user’s email. User can bypass login when the user clicks the link in the email. You should use an email address that you have access e.g. your RMIT student email address.

Search: Search function works for both unregistered users and logged-in members. Items that contain the search key word will be listed. The returned items can be clicked to show the item details.



DIY login: The login, signup functionalities explained in CR level must be implemented from scratch without using any related third-party gems.

Favourite: At this level, to select an item, user can click the bottom right corner of the item’s image to add the item to the saved list. A favourite tag/icon will always appear whenever the item is displayed in the app. The tag/icon is clickable allowing selection and deselection. Clicking elsewhere on the image will show item details (as shown on Page 3).

Rating: after the first checkout of a newly registered user, a rating page will pop out. It only appears once for each registered user.

===== 90+ LEVEL =====

NOTE: You should only attempt this level when you have completed **80+ level**.

Incremental search: the search function mentioned in the HD level is incremental at this level. As user types the search keyword, matches are presented and updated in real time. It allows user to find an item without completing the text entry.

Deletion: In the shopping bag page, user can swipe left an item to remove it from the bag.

Multiple browsers: A user may login from multiple devices or browsers. The saved list and the shopping bag are synchronized among these devices, e.g. when an item is added, it appears on all logged in devices. When one device completes the checkout, purchased items will disappear from other devices.

Camera: This is to simulate modern online store apps which allow user to take a photo of a garment. So the app can use advanced AI techniques to find similar products in the collection. Your task here is simply engaging with the device camera to take a photo and uploading the photo to the server. You may test this on computer browsers with HTML5 features. Not required to work on iOS.

Admin portal: A simple portal, of which the address is `<your app's url>/admin`. It allows admin to view the summary of saved items, purchased items, overall rating and newsletter subscriptions. The admin credential is “**admin**” with password “**Rails2021**”.

5. Referencing guidelines

If you have used sources of information other than the contents directly under Canvas→Modules, you must give acknowledge the sources and give references using IEEE referencing style. Where: Add a code comment near the work to be referenced and include the reference in the IEEE style.

How: To generate a valid IEEE style reference, please use the [citethisforme tool](#) if unfamiliar with this style. Add the detailed reference before any relevant code (within code comments).

6. Academic integrity and plagiarism (standard warning)

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and/or ideas of others you have quoted (i.e. directly copied), summarised, paraphrased, discussed or mentioned in your assessment through the appropriate referencing methods,
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from Internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source
- Copyright material from the internet or databases
- Collusion between students

For further information on our policies and procedures, please refer to the [University website](#).

7. Assessment declaration

When you submit work electronically, you agree to the [assessment declaration](#).

8. Submission

8.1 The submission can be from **an individual** or **a team no more than three**. For a three-member team, **15% mark reduction** will apply. For example, a 59-mark submission from a three-member team will result in 50 for each member.

8.2 Your application is expected to be production ready. Proper **test and data validation** should be embedded in your application **regardless of the level that you are attempting**.

8.3 There is no need for your application to look and behave exactly like the above example. You can make necessary adjustment to improve the design. However the aforementioned functionalities are the minimum requirement and should be well supported.

8.4 You are required to deploy the final version onto **Heroku**.

8.5 In the **README.md** file of your application, you must follow the follow format

Line 1 Name(s) & student number(s) of the development team

Line 2 PI or CR or DI or 80+ or 90+ # the HIGHEST level you have attempted, meaning all levels below have been successfully completed.

Line 4 onwards The time sheet for each team member (see Point 8.6 below)

Heroku deployment URL and the last Heroku deployment log

8.6 Your assignment should hosted on a **PRIVATE** bitbucket or Github repository named as [RAD2021_{studentNumber\(s\)_in_ascending_order_separated_by_underscore}](#). You should update your repo as frequent as possible. This is a assessment point.

8.7 You are required to keep track of the hours that you spent on the development including testing, discussion, reading.

For one professional developer working alone, the expected completion time is about 8 ~ 10 full days.

For time tracking purpose, we recommend [toggl.com](#), a free online time tracker. You can sign in with your RMIT student account (under "SIGN UP WITH GOOGLE"). This log needs to be included in the **README.md** file (See Point 8.5 above).

8.8 Below is a list of gems that you can use. You may use **devise** for login at a level lower than HD. No other gems should be used. Keep an eye on the announcement.

```
gem 'rails',
gem 'bootstrap-sass',
gem 'puma',
gem 'sass-rails',
gem 'uglifier',
gem 'coffee-rails',
gem 'jquery-rails',
gem 'turbolinks',
gem 'jbuilder',
gem 'carrierwave',
gem "mini_magick"
gem "font-awesome-rails"
gem 'bcrypt',
```

8.9 You can use media storage services like Amazon S3 to store images as Heroku may not keep media files for long.

8.10 More details about submission and demo will be provided close to the due date.

Submission failed to run on Cloud 9 and Heroku would not be marked.

Rubric

Assessment Task	Marks
Pass Level	50 marks
Credit Level	60 marks
Distinction Level	70 marks
80+ Level	80 marks
90+ Level	91 marks
At least one commit before Week 8; At least one commit between Week 8-9; At least one commit between Week 10-11;	3 marks
Test cases	3 marks
Final demo in Week 12	3 marks
1-member submission 2-member submission 3-member submission	final = mark * 100% final = mark * 100% final = mark * 85%
Submission from a group larger than 3 receives zero mark	