



Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

BSc Software Systems Development (Honours)

Dynamic Web Development Assignment

Report

Xinyu Miao (20069571)

Lecturer: David Drohan

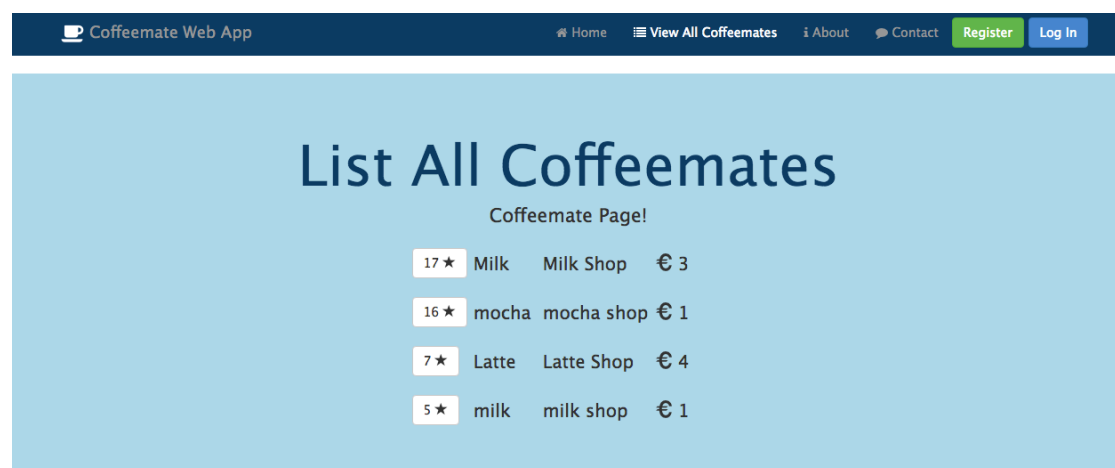
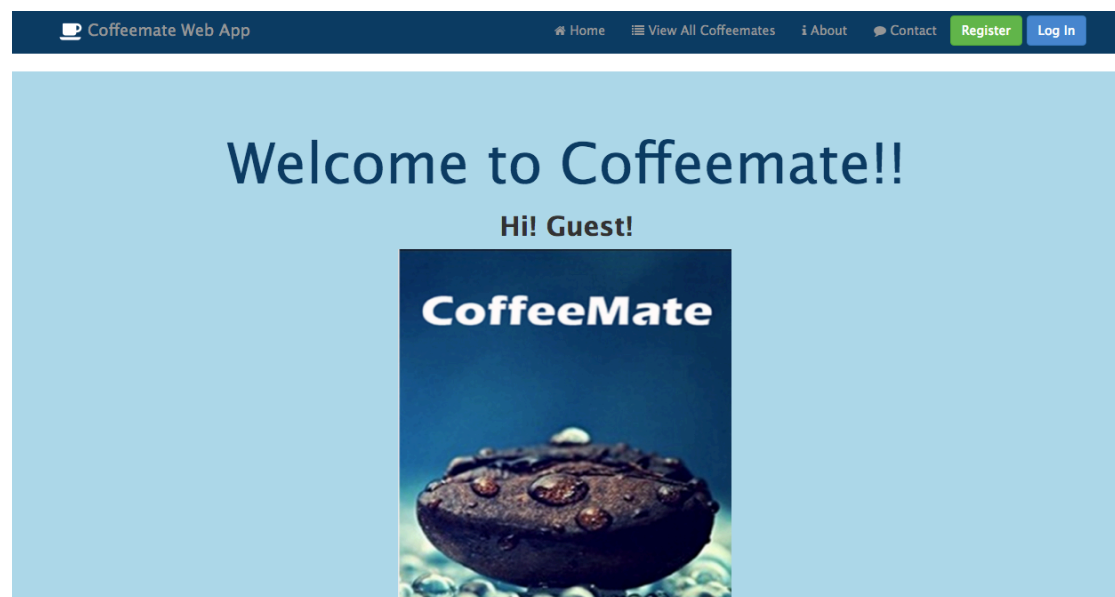
Date: 26/04/2016

Overview

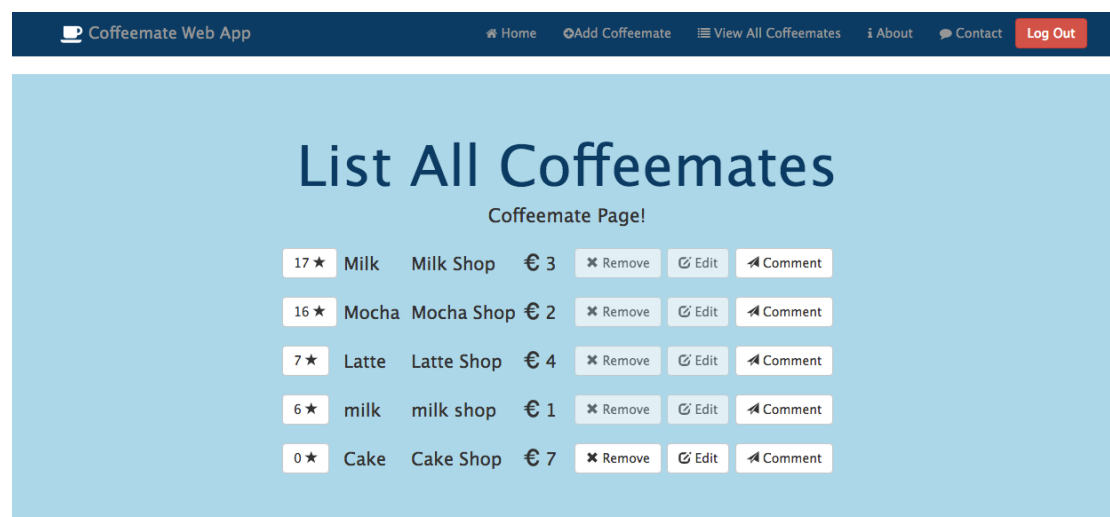
This assignment is about Coffeemate Web Application, which is developed by me using Mongo DB, Express, Angular JS and Node.js. The add, edit, delete and list functionalities were implemented for both client and server. In addition, I added the user model in the web application and the register and login need the validation. I chose bootstrap framework to design the user interface. All the users can make comments on the Coffeemate and leave reviews about the application.

Features and Functionalities

(1) If the users don't login, they won't see the add Coffeemate button and the edit, remove and comment buttons in the list view page.



(2) When the users login, they can see the account information “Hi! (Username)!” on the home page. In addition, the register and login button change to logout button. In the list view page, the users can only edit or remove the coffee mates that belong to them.

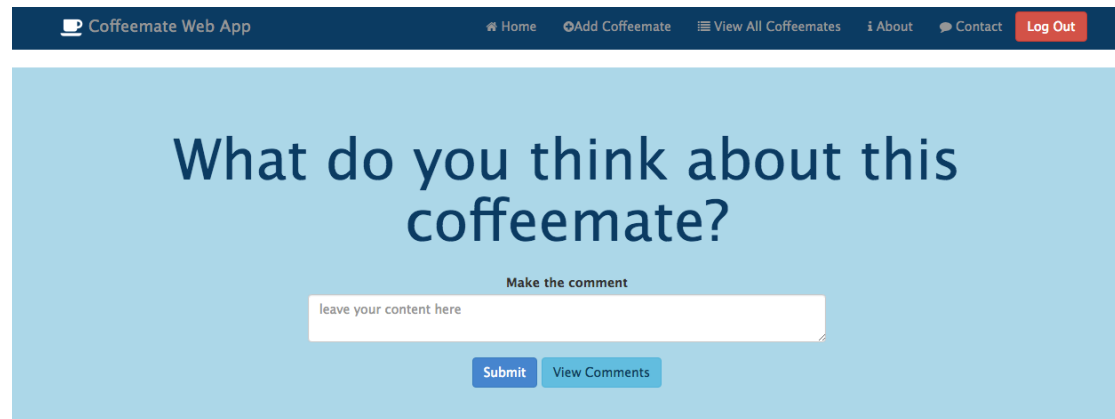


The users can vote by clicking the star-rating button in each list. In addition, the list of coffee mates is ordered by the rating of each coffee mate.

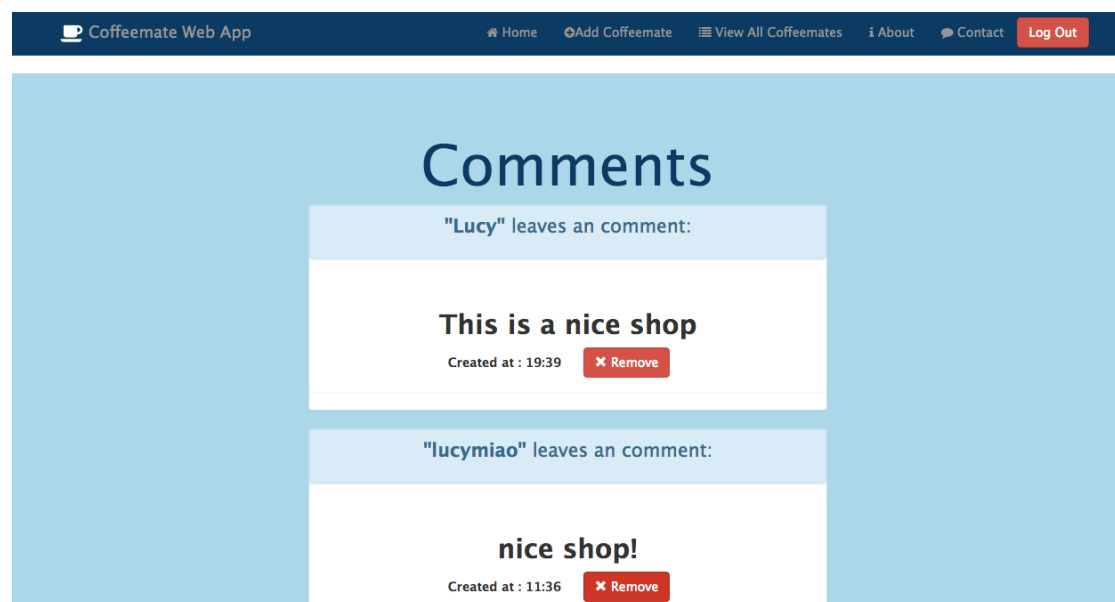
I set the user permission here. In another world, if the coffee mate does not belong to you, the remove and edit buttons are disabled. If the coffee mate belongs to you, you can click on the remove button to delete the coffee mate or click on the edit button to update the coffee mate.

Extra Features

- (1) The users can click the comment button and leave a comment for each coffee mate.



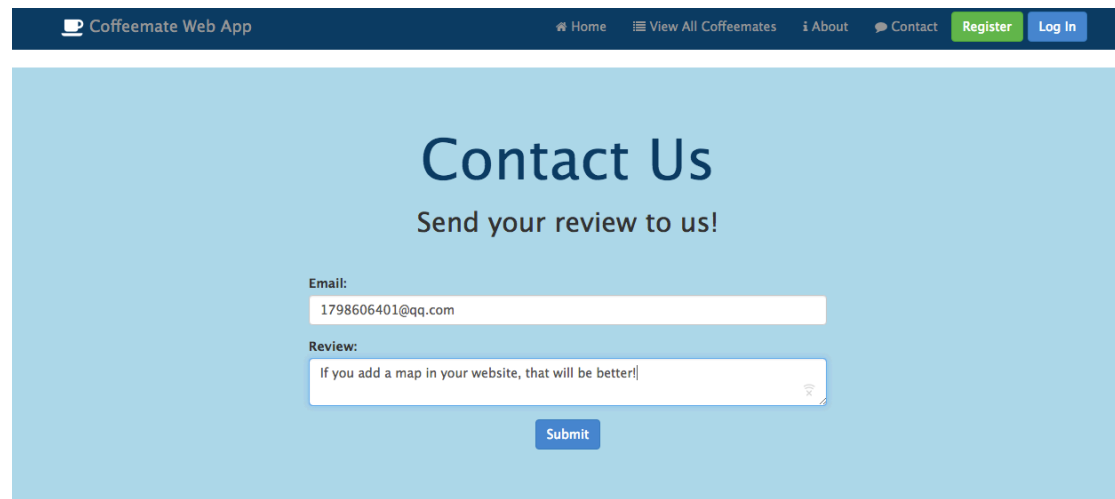
The screenshot shows the 'Make the comment' form in the Coffeemate Web App. The form has a dark blue header with the app name and navigation links. The main content area is light blue and contains the text 'What do you think about this coffeemate?'. Below this is a text input field with the placeholder 'leave your content here'. At the bottom of the form are two buttons: 'Submit' and 'View Comments'.



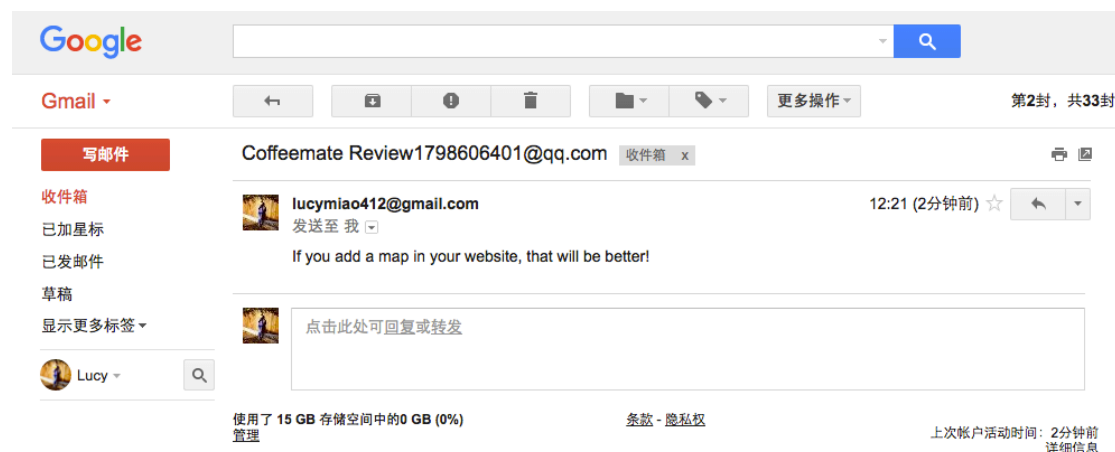
The screenshot shows the 'Comments' page in the Coffeemate Web App. The page has a dark blue header with the app name and navigation links. The main content area is light blue and contains the title 'Comments'. Below the title are two comment cards. The first card is for a comment by 'Lucy' that says 'This is a nice shop', created at 19:39, with a 'Remove' button. The second card is for a comment by 'lucymiao' that says 'nice shop!', created at 11:36, with a 'Remove' button.

If the users click on the view comments button, they can view the comments of each coffee mate. In addition, in the view comments page, the users can only remove the comments that belong to them. If the comments don't belong to the users, the remove buttons are disabled.

(2) In the contact page, all the users can leave their reviews about the application, and the administrator will receive the reviews by email. I chose Nodemailer to implement this feature. Nodemailer is an e-mail sending module for Node.js.



The screenshot shows the 'Contact Us' page of the 'Coffeemate Web App'. The page has a light blue background. At the top, there is a dark blue navigation bar with links: Home, View All Coffeemates, About, Contact, Register, and Log In. The main heading is 'Contact Us' in a large, dark blue font, followed by the subheading 'Send your review to us!'. Below this, there is a form with two input fields. The first field is labeled 'Email:' and contains the text '1798606401@qq.com'. The second field is labeled 'Review:' and contains the text 'If you add a map in your website, that will be better!'. A blue 'Submit' button is located below the review field.



Reference

Nodemailer <https://nodemailer.com/>