

T4A2-B - Full Stack Application Presentation

By Georgie Liefman

WhereTo?

WhereTo? is an application that allows for its users to share their travel stories and reviews of places around the world they have visited with an online community. Not only does WhereTo? provide a sense of community for users, it also provides real consumer evaluations of accommodations, dining experiences, and tourism activities so that travelers may feel confident about their vacation plans before they commit.

Target Audience

Given the nature of the application, *WhereTo?*'s target audience is broadly all travel lovers. *WhereTo?*'s target demographic is travel aficionados who, rather than using traditional travel agencies, turn to the internet for assistance with travel services and an eagerness to hear about other traveler's adventures.

Functionalities and Features

- The application allows users to post reviews/stories about their previous travel experiences.
- User posts consist of a title, description, destination, price and category of the subject of their review (for example restaurant, hotel or activity).
- Authors of posts are able to delete their posts if they wish too.
- Users are able to sign up to the site and user authentication is used where appropriate.
- Users can comment and delete comments on a post.
- Users are able to heart and unheart reviews.

User Stories

- As an avid traveler I want to thoroughly review and rate my previous experiences, so I can help inform other travelers.
- As an avid traveler I want to post about my previous travel stories, so I am able to come back and reflect on my journeys.
- As a user of *WhereTo?* that posts reviews, I want to be able to alter my old posts, so I can delete posts in case my opinions change overtime.
- As a holiday planner I want to read other users' travel reviews, so that I can plan my next holiday informed by real travelers' reviews.
- As a member of the travel community I want to be able to read, comment on and like the reviews from other users, so I can engage with other travelers and their posts.
- As a member of the travel community I want posts and reviews to be categorised, so I am able to quickly find what I am looking for and not have to read through whole posts I am not interested in.
- As a new user I want to be able to create an account, so I can have access to features only available to *WhereTo?* users.

Important Libraries and Technologies

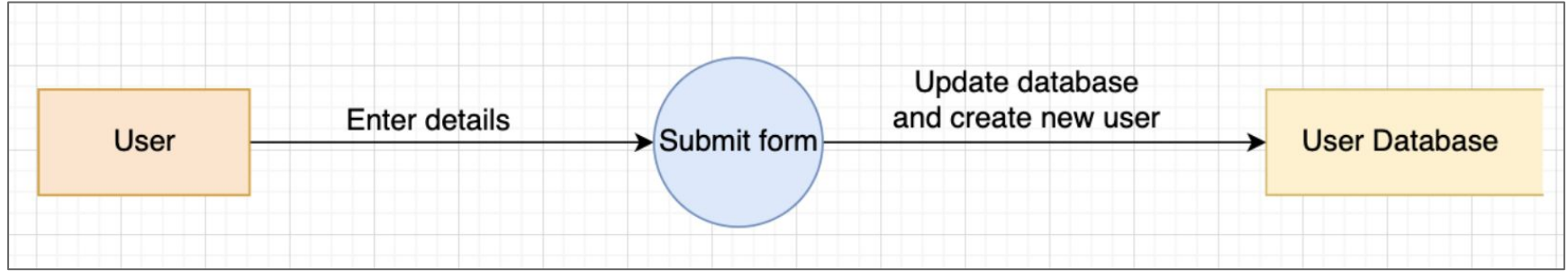
- Flask
- SQLite
- Bootstrap
- EC2
- Load Balancer
- API Gateway
- Docker
- Git/Github (webhooks)

List of Pages on Site

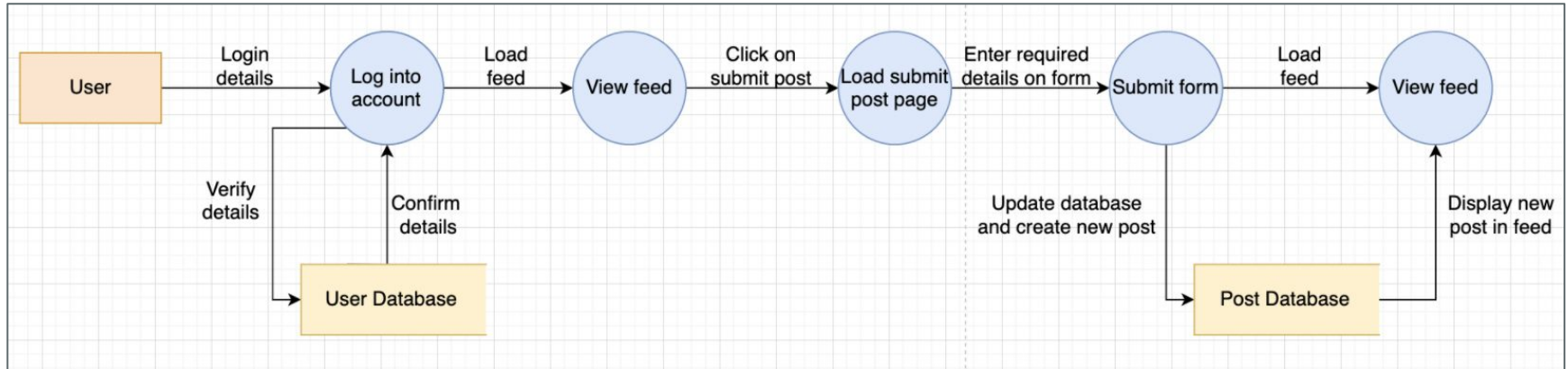
- Login Page: Allows users to login to site.
- Sign Up Page: Allows users to signup to the site.
- Homepage: Displays all previous reviews created on site.
- Individual Users' Pages: Displays all posts created by the individual user that is registered to the page.
- Create a Review Page: Is the page users use to create reviews.

Significant Data Flow Diagrams

Sign Up a New User

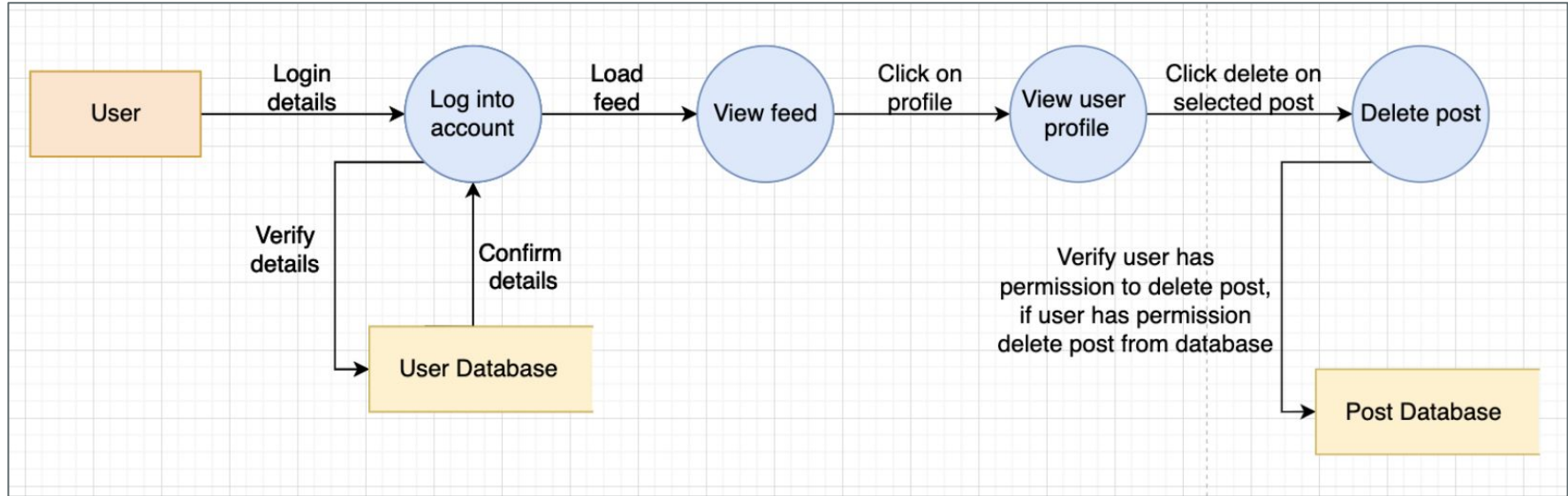


Create a Post



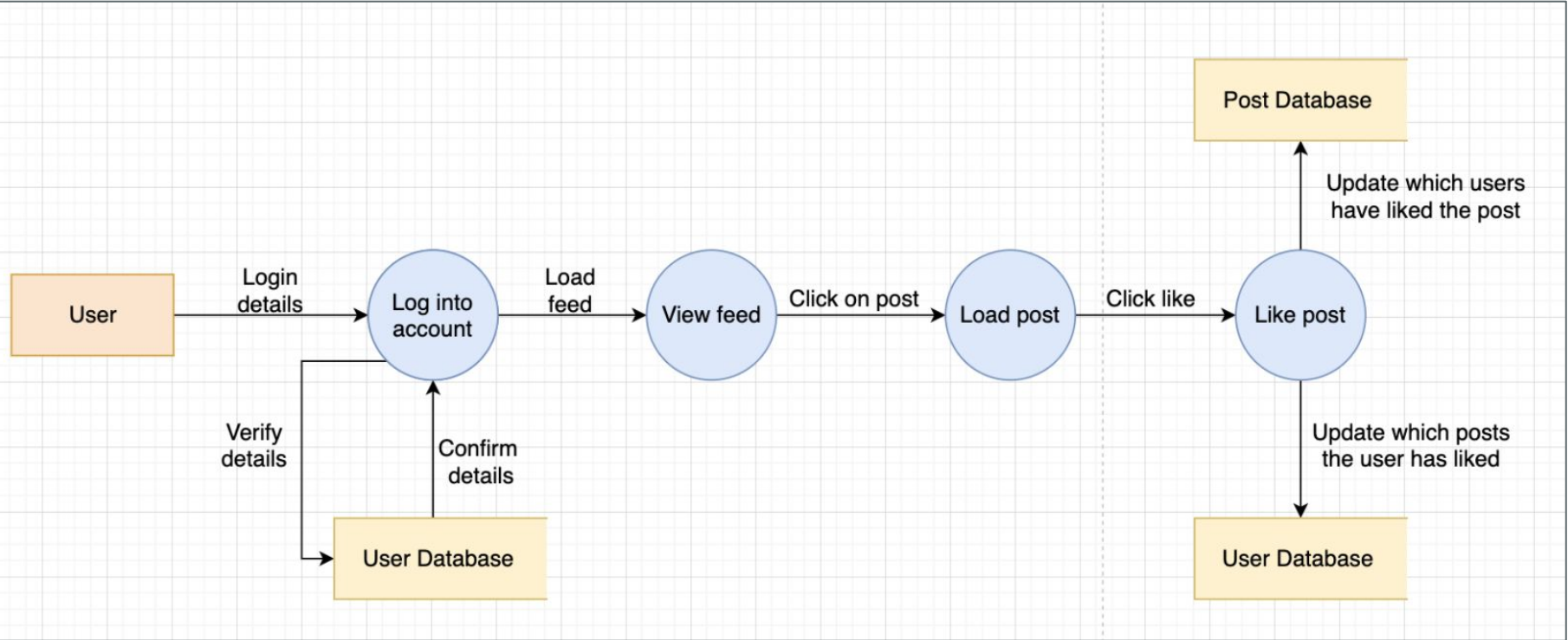
Significant Data Flow Diagrams Continued

Deleting a Post

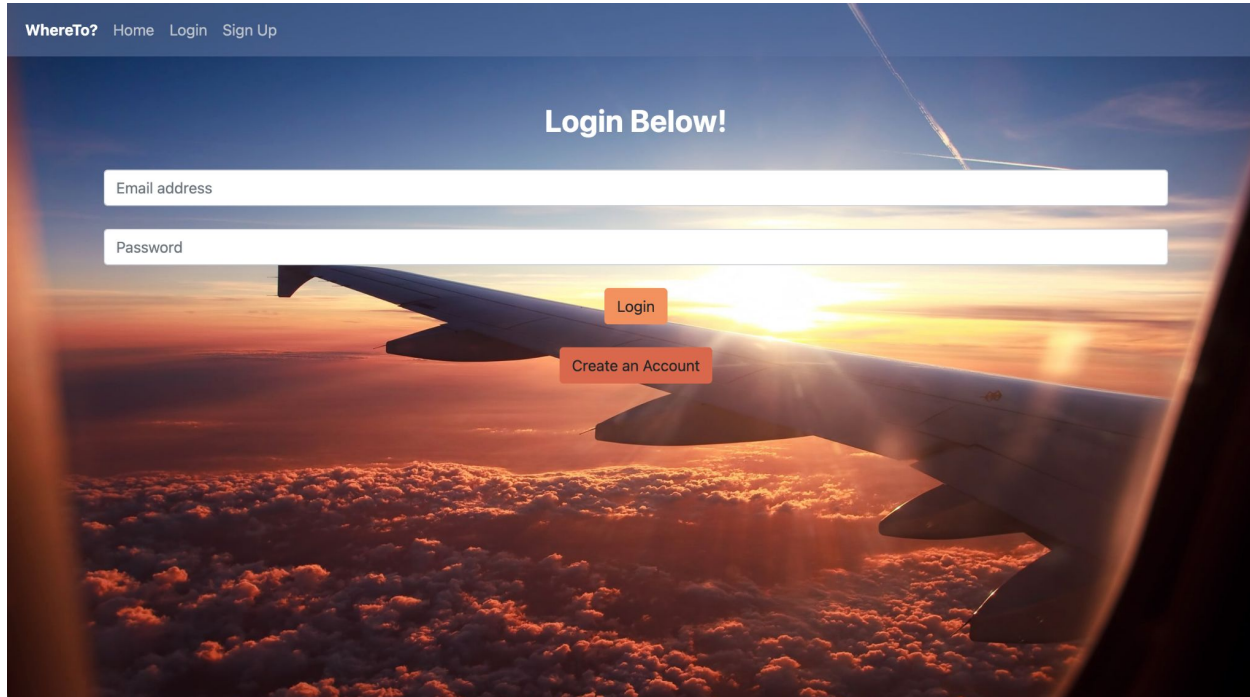


Significant Data Flow Diagrams Continued

Hearting a Post



Screenshots of Login Page



WhereTo? Home Login Sign Up

Login Below!

Email address

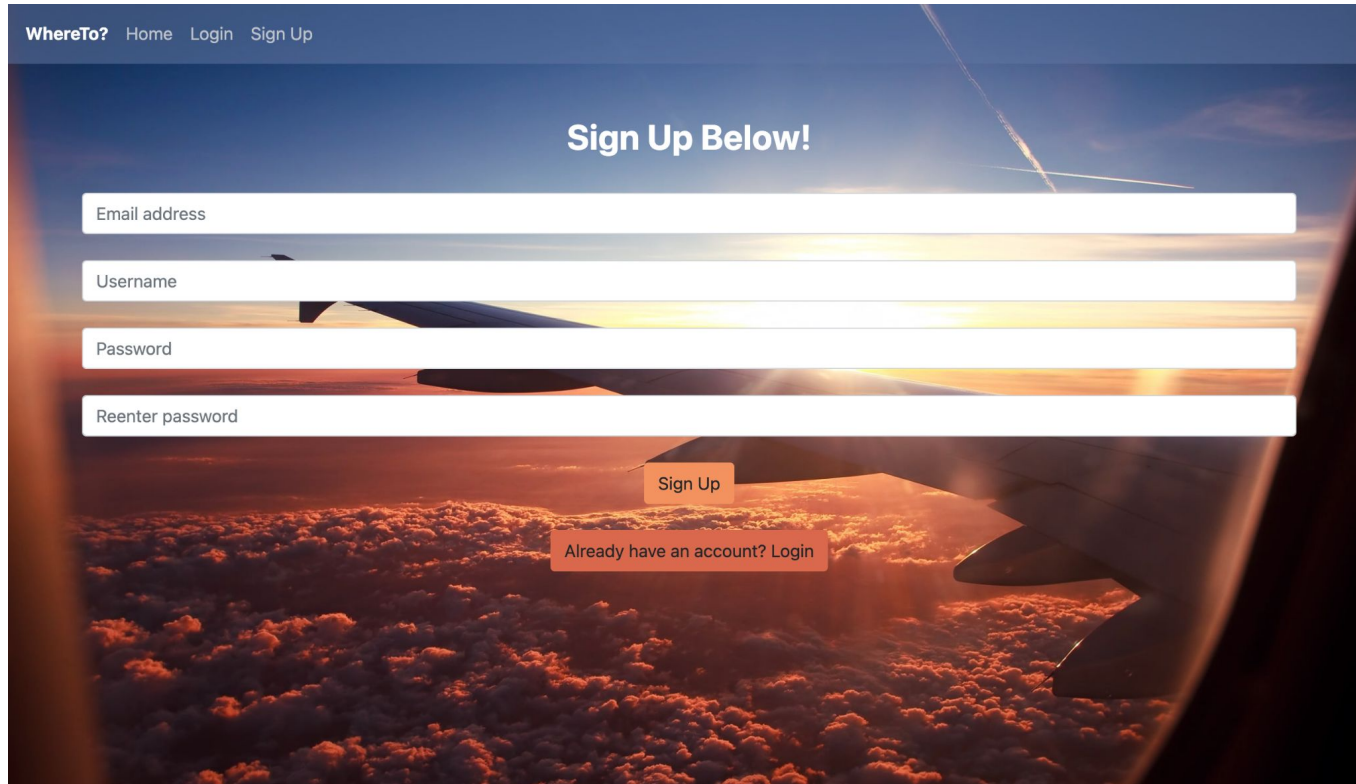
Password

Login

Create an Account

The background of the login page features a vibrant sunset or sunrise scene with a bright sun low on the horizon, casting a warm glow over a sea of clouds. The wing of an airplane is visible in the foreground, extending from the right side towards the center, adding a sense of travel and adventure to the design.

Screenshot of Sign Up Page



The screenshot shows a web page for signing up. The background is a vibrant sunset or sunrise scene with a large airplane wing visible on the right side. The sky is a mix of orange, yellow, and blue. The page has a dark blue header with the text 'WhereTo?' and navigation links 'Home', 'Login', and 'Sign Up'. The main heading 'Sign Up Below!' is centered in white. Below it are four white input fields for 'Email address', 'Username', 'Password', and 'Reenter password'. At the bottom, there are two orange buttons: 'Sign Up' and 'Already have an account? Login'.

WhereTo? [Home](#) [Login](#) [Sign Up](#)

Sign Up Below!

Email address

Username

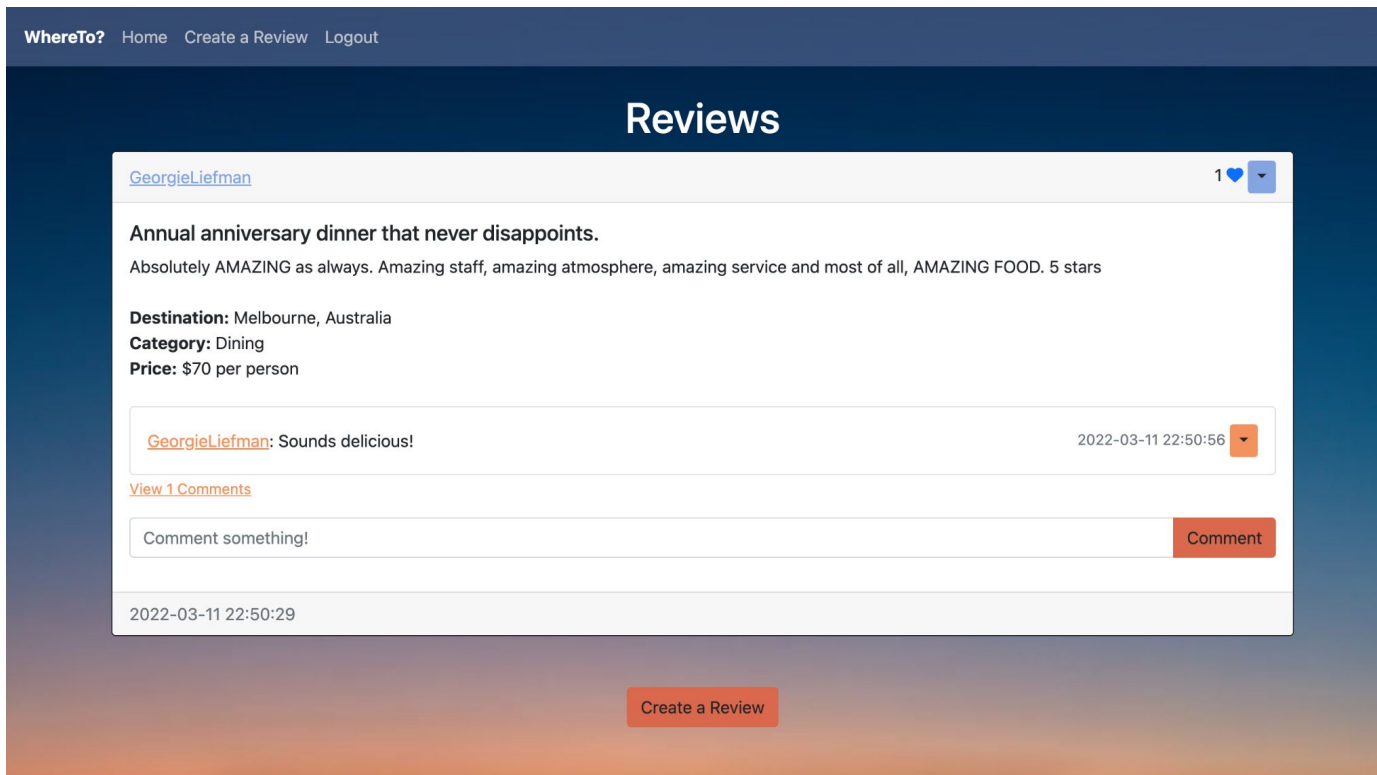
Password

Reenter password

[Sign Up](#)

[Already have an account? Login](#)

Screenshots of Homepage



Screenshots of Create Review Page

WhereTo? [Home](#) [Create a Review](#) [Logout](#)

Create a Review

Title of your review

Your review

Location of experience

Type of experience

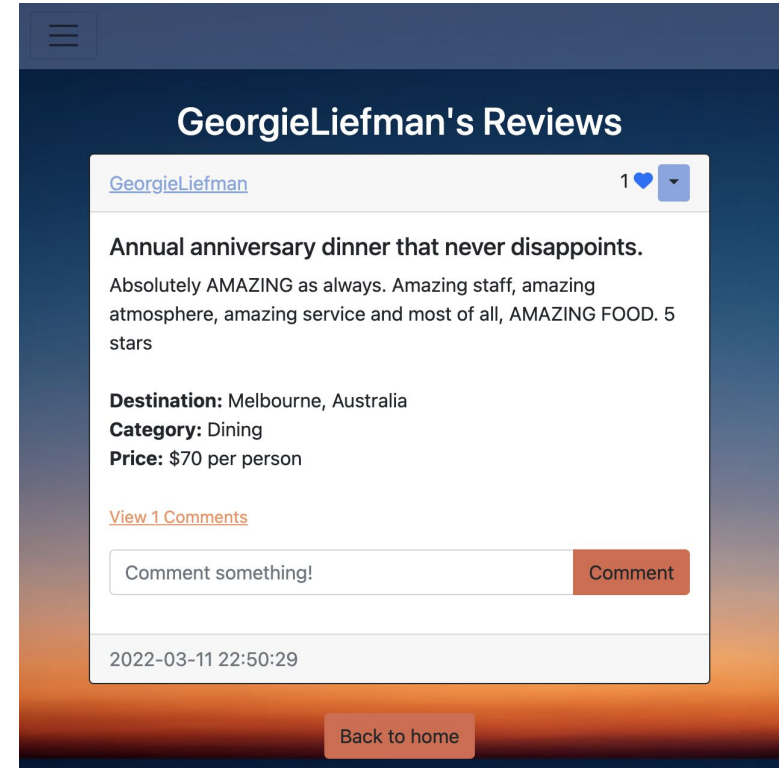
Price of experience

[Submit Review](#)

[Back to home](#)

Screenshot of User's Profile (Mobile view)

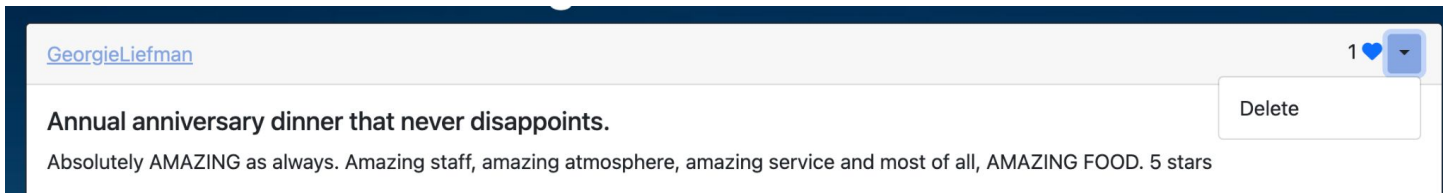
- The website has been made fully responsive using Bootstrap.
- On the mobile view of the site the navbar becomes a hamburger menu to improve the overall navigation experience.



Notable Features of a Review

- Users are able to comment and heart reviews.
- To view comments which are on a posting, users must click the view comments tab to extend the posting (the tab is only accessible if there is any comments).
- Only the original posters of reviews and comments are able to delete their postings.
- Only the original poster of the review has the ability to see the delete button for the review, this is pictured below.

View of review when original poster



View of review when not original poster



Notable Features of a Review Continued

- When a user hearts a review the heart counter on the post is increased by 1. The opposite is true when a user unhearts a post.
- When a user hearts a post the heart turns from being non-filled in to being solid. Again the opposite is true when a user unhearts a post.
- Both of these events are pictured below.

A review prior to a user hearting it.



A review after a user hearting it.



Flashing Messages in Program

- Flashing messages are used in the program to inform users of events and errors.
- When invalid user input occurs in the application, users are flashed an error message to inform them what the error is and what they need to do to continue using the application as intended.
- When a user does something of note successfully such as creating a post, a message confirming this event has been successful flashes.
- Screenshots of both types of these messages are included below.

Error Message

A review cannot be created without a title. ✕

Successful Message

Logged in! ✕

Testing During Development

Unit testing and a manual testing suite was used to test all features of the site during development. A screenshot of the testing suite used is attached below.

Test	Feature being tested	User Input	Expected Outcome	Actual Outcome	Status	Comments
#1	Signing up a user.	Fill out the form correctly as expected.	A new user should be created and added to the database. User should also be displayed the homepage.	The application performed as expected.	Success	
#2	Signing up a user.	Enter the repeated password incorrectly.	A message should pop up telling the user the passwords do not match and a new user should not be created. The user should remain on the signup page.	The application performed as expected.	Success	
#3	Signing up a user.	Enter a email already in use.	A message should pop up telling the user the email is already in use and a new user should not be created. The user should remain on the signup page.	The application performed as expected.	Success	
#4	Signing up a user.	Enter a username already in use.	A message should pop up telling the user the username is already in use and a new user should not be created. The user should remain on the signup page.	The application performed as expected.	Success	
#5	Signing up a user.	Enter an invalid email.	A message should pop up telling the user the email is not valid and a new user should not be created. The user should remain on the signup page.	The application performed as expected.	Success	
#6	Logging in a user.	Fill out the form with correct registered credentials.	The user should be logged in and redirected to the homepage.	The application performed as expected.	Success	
#7	Logging in a user.	Enter an unregistered email when attempting to login.	A message should pop up telling the user that the email is not registered to an account and they will remain on the login page.	The application performed as expected.	Success	
#8	Logging in a user.	Enter an incorrect password.	A message should pop up saying the password is incorrect and the user should remain on the login page.	The application performed as expected.	Success	
#9	Creating a post.	Fill out the form to create a post correctly as expected.	A message saying the review has been successfully created should pop up and the user should be redirected to home where their review is now being displayed.	The application performed as expected.	Success	
#10	Creating a post.	Submit an empty review.	An error message should pop up and no review should be created. The user should remain on the review page.	The application performed as expected.	Success	
#11	Creating a post.	Not enter a destination for the review.	An error message should pop up saying the destination cannot be empty and no review should be created. The user should remain on the review page.	A review was created when it should not have been.	Failed	Fixed conditional statements relating to creating a post and it is now functioning as expected.
#12	Creating a post.	Not enter a category for the review.	An error message should pop up saying the category cannot be empty and no review should be created. The user should remain on the review page.	The application performed as expected.	Success	
#13	Deleting a post.	The OP of the review deletes the post.	The user should be able to delete the post and once it is deleted the user should be redirected to the homepage.	Can delete initial review but unable to delete other reviews also created by user.	Failed	Removed line of code causing error and made it so only original posters have ability to even view/click delete review button.
#14	Deleting a post.	Someone other the poster attempts to delete the review.	The user should not have the option or anyway to delete the review.	The application performed as expected.	Success	

Testing During Development Continued

#15	Creating a comment.	Post a comment on a review.	The comment should be posted successfully and the user should be redirected to the homepage.	The application performed as expected.	Success
#16	Creating a comment.	Post a comment on a review which has already been commented on.	The comment should be posted successfully under any comments already posted and the user should be redirected to the homepage.	The application performed as expected.	Success
#17	Creating a comment.	Submit an empty comment.	An error message should pop up and no comment should be created. The user should remain on the current page.	The application performed as expected.	Success
#18	Deleting a comment.	The OP of the comment deletes the comment.	The user should be able to delete the comment and once it is deleted the user should be redirected to the homepage.	The application performed as expected.	Success
#19	Deleting a comment.	Someone other the poster attempts to delete the commenry.	The user should not have the option or anyway to delete the comment.		
#20	Deleting a comment.	The OP of the comment deletes the comment on a post with multiple comments.	The comment should be deleted successfully and no other comments on the post should be impacted and the user should be redirected to the homepage.	The application performed as expected.	Success
#21	Liking a review.	A user likes a comment.	The user should be able to like the post, this will register, change the like icon and the like count should be +1 whatever it was before. The user should be redirected to the homepage.	The application performed as expected.	Success
#22	Unliking a review.	A user unlikes a comment they have previously liked.	The user should be able to unlike the post, this will register, change the like icon and the like count should be -1 whatever it was before. The user should be redirected to the homepage.	The application performed as expected.	Success
#23	Going to a user's profile.	Click on a user's name on a post they have created.	The user should be redirected to the poster's profile and all their previous posts should be displayed.	The application performed as expected.	Success
#24	Logging out a user.	Click logout on the navbar.	The user should be logged out and redirected to the login page.	The application performed as expected.	Success
#25	Using the navbar.	Click home on the navbar.	The user should be redirected to the homepage.	The application performed as expected.	Success
#26	Using the navbar.	Click login on the navbar.	The user should be redirected to the login page.	The application performed as expected.	Success
#27	Using the navbar.	Click signup on the navbar.	The user should be redirected to the signup page.	The application performed as expected.	Success

Only two of the tests failed during development and the features concerned have since been fixed.

Final Steps Left to Complete

- The application has been fully tested during development and the development process is finished.
- I still need to deploy the application to AWS and complete testing during production.
- After production testing is completed I need to complete the ReadMe and then I will be ready to submit.