KitRate backend API Testing

Manual Functionality Testing:

Page loading:

- Expected: Home page should load in the terminal with no errors
- Testing: type "python3 manage.py runserver” into the terminal
- Result: No errors reported with debug set to True. Page loads as expected.
- Fix: N/A

- Expected: With page deployed, home page should load with no errors
- Testing: open page from Heroku app
- Result: CSS not working
- Fix: Install ‘whitenoise’ and add to settings.py. Now CSS from the framework is working as expected.

Products:

- Expected: /products url displays a list of products from the database with associated data
- Testing: input /products URL and observe returned data
- Result: List of products displayed as expected
- Fix: N/A

- Expected: /products/1 url displays product with ID=1 from the database with associated data
- Testing: input /products/1 URL and observe returned data
- Result: Product id=1 displays as expected
- Fix: N/A

- Expected: With a logged in admin user, /admin/products url displays a list of products from the database
- Testing: input admin/products URL and observe returned data
- Result: Log in successful, URL displays products as expected
- Fix: N/A

- Expected: With a logged in admin user, create a new product with an image over 2MB should receive an error
- Testing: Create a new product, submit a large image and observe error
- Result: Error presented when uploading a large image
- Fix: N/A

- Expected: With a logged in admin user, successfully create a new product
- Testing: Create a new product; fill in one form field at a time, submitting between each and look for an error. Form should only submit when all data is present.
- Result: Field errors displayed as expected. Product created as expected.
- Fix: N/A

- Expected: With a logged in admin user, make edits to an existing product from the /products/{id} URL.
- Testing: Edit a product and confirm update has been successful
- Result: Existing product updated as expected
- Fix: N/A

- Expected: With a logged in admin user, delete an existing product from the /products/{id} URL.
- Testing: Delete a product and confirm deletion has been successful
- Result: Existing product deleted as expected
- Fix: N/A

Favourites:

- Expected: /favourites url displays a list of favourites from the database with associated data
- Testing: input /favourites URL and observe returned data
- Result: List of favourites displayed as expected
- Fix: N/A

- Expected: /favourites/1 url displays favourites with ID=1 from the database with associated data
- Testing: input /favourites/1 URL and observe returned data
- Result: Favourite id=1 displays as expected
- Fix: N/A

- Expected: With a logged in admin user, /admin/favourites url displays a list of favourites from the database
- Testing: input admin/favourites URL and observe returned data
- Result: Log in successful, URL displays favourites as expected
- Fix: N/A

- Expected: With a logged in admin user, successfully create a new favourite
- Testing: Create a new favourite; fill in one form field at a time, submitting between each and look for an error. Form should only submit when all data is present.
- Result: Field errors displayed as expected. Favourite created as expected.
- Fix: N/A

- Expected: With a logged in admin user, make edits to an existing favourite from the /favourites/{id} URL.
- Testing: Edit a favourite and confirm update has been successful
- Result: Existing favourite updated as expected
- Fix: N/A

- Expected: With a logged in admin user, delete an existing favourite from the /favourites/{id} URL.
- Testing: Delete a favourite and confirm deletion has been successful
- Result: Existing favourite deleted as expected
- Fix: N/A

Reviews:

- Expected: /reviews/1 url displays review with ID=1 from the database with associated data
- Testing: input /reviews/1 URL and observe returned data
- Result: Review id=1 displays as expected
- Fix: N/A

- Expected: With a logged in admin user, /admin/reviews url displays a list of reviews from the database
- Testing: input admin/reviews URL and observe returned data
- Result: Log in successful, URL displays reviews as expected
- Fix: N/A

- Expected: With a logged in admin user, successfully create a new review
- Testing: Create a new review; fill in one form field at a time, submitting between each and look for an error. Form should only submit when all data is present.
- Result: Field errors displayed as expected. Review created as expected.
- Fix: N/A

- Expected: With a logged in admin user, make edits to an existing review from the /reviews/{id} URL.
- Testing:Edit a review and confirm update has been successful
- Result: Existing review updated as expected

- Fix: N/A

- Expected: With a logged in admin user, delete an existing review from the /reviews/{id} URL.
- Testing: Delete a review and confirm deletion has been successful
- Result: Existing review deleted as expected
- Fix: N/A

Categories:

- Expected: /categories url displays a list of categories from the database with associated data
- Testing: input /categories URL and observe returned data
- Result: List of categories displayed as expected
- Fix: N/A

- Expected: /categories/1 url displays category with ID=1 from the database with associated data
- Testing: input /categories/1 URL and observe returned data
- Result: Category id=1 displays as expected
- Fix: N/A

- Expected: With a logged in admin user, /admin/categories url displays a list of categories from the database
- Testing: input admin/categories URL and observe returned data
- Result: Log in successful, URL displays products as expected
- Fix: N/A

- Expected: With a logged in admin user, successfully create a new category
- Testing: Create a new category; fill in one form field at a time, submitting between each and look for an error. Form should only submit when all data is present.
- Result: Field errors displayed as expected. Category created as expected.
- Fix: N/A

- Expected: With a logged in admin user, make edits to an existing category from the / categories/{id} URL.
- Testing: Edit a category and confirm update has been successful
- Result: Existing category updated as expected
- Fix: N/A

- Expected: With a logged in admin user, delete an existing category from the /categories/{id} URL.
- Testing: Delete a category and confirm deletion has been successful
- Result: Existing category deleted as expected
- Fix: N/A

Subscriptions:

- Expected: /subscriptions url displays a list of subscriptions from the database with associated data
- Testing: input /subscriptions URL and observe returned data
- Result: List of subscriptions displayed as expected
- Fix: N/A

- Expected: /subscriptions/1 url displays subscription with ID=1 from the database with associated data
- Testing: input /subscriptions/1 URL and observe returned data
- Result: Subscription id=1 displays as expected
- Fix: N/A

- Expected: With a logged in admin user, /admin/subscriptions url displays a list of subscriptions from the database
- Testing: input admin/subscriptions URL and observe returned data
- Result: Log in successful, URL displays subscriptions as expected
- Fix: N/A

- Expected: With a logged in admin user, successfully create a new subscription
- Testing: Create a new subscription; fill in one form field at a time, submitting between each and look for an error. Form should only submit when all data is present.
- Result: Field errors displayed as expected. Subscription created as expected.
- Fix: N/A

- Expected: With a logged in admin user, make edits to an existing subscription from the / subscriptions/{id} URL.
- Testing: Edit a subscription and confirm update has been successful
- Result: Existing subscription updated as expected
- Fix: N/A

- Expected: With a logged in admin user, delete an existing subscription from the / subscriptions/{id} URL.
- Testing: Delete a subscription and confirm deletion has been successful
- Result: Existing subscription deleted as expected
- Fix: N/A