GAVIN MOY

Unit Testing

Email: gmoy@ucsc.edu
University of California, Santa Cruz

CMPS115 - Software Engineering Project I

Contents

ArtInfo.js

ArtInfo.js requires an image to be in the Redux Store.

ArtInfo.js has been tested in an un-automated environment several ways. The container holds several interactable fields.

- Three text fields for art piece info. (Artist Name, Art Title, Art Info)
- Two Buttons (Back, Upload)

The three text fields are able to take in a string, and then the **Upload** button will POST them to the database. The image will then be uploaded to the bucket. The page will then redirect to the **Artist Dashboard**.

For the following un-automated tests, input will be referenced in the order presented in the itemized section above. (Image, Artist Name, Art Title, Art Description)

Output was checked manually in the database.

Test 1:

If the User were to arrive the page without having an image in the Store, after pressing upload,an alert should pop up preventing the upload and reroute to dashboard.

Input: null,Test Name ,Test Title ,Test Description

Expected Output:"File Not Chosen", No Changes in the database.

Actual Output: "File Not Chosen", No Changes in the database.

Test 1 was successful.

Test 2:

If the User were to attempt to submit without filling all the fields, the user should be prompted to fill it in. Paramater *null* represents a null or empty string.

Input: test.png,*null* ,Test Title ,Test Description

Expected Output: Please Fill Out This Field.

Actual Output: Please Fill Out This Field.
Test 2 was successful.

Test 3:

This test tests If Everything was input correctly.

Input: test.png, test Artist, test Title, test Description

Expected Output: route to **Artist Dashboard**, image.png in bucket, test Artist, test Title, test Description in Database

Actual Output: route to **Artist Dashboard**, image.png in bucket, test Artist, test Title, test Description in Database
Test 3 was successful.

UPLOADPAGE.JS

UploadPage.js prompts a user to select an image. Upon clicking the **Choose File** the user can select and image from their device storage. Once an image is selected, the user should be able to hit a **Next** button which routes them to **Art Info**. The user should not be able to hit **Next** until an image has been selected

UploadPage.js has been tested in an un-automated environment. The container holds two interactable fields.

- The Choose File Button
- The Next Button

Output was checked using the Redux Dev Tools Extension chrome extension to check the state of the store

Test 1:

Attempting to hit the next button without selecting an image.

Input: no image is selected, press next

Expected Output: the user will not be able to click next

Actual Output: was not able to click next because it was grayed out

Test 1: successful

Test2:

Selecting an image and hitting next

Input:press **Choose file** button, Select text.jpeg, hit **Next**

Expected Output: Hitting **Choose file** will open a window to select an image. After selecting an image, the **Next** button should no longer be grayed out. Pressing the **Next** button should route to the **ArtInfo** page. The Redux store should be updated to hold an image in imageReducer.image.

Actual Output: Hitting **Choose file** will opened a window to select an image. After selecting an image, the **Next** button was no longer grayed out. Pressing the **Next** button routed to the **ArtInfo** page. According to Redux Dev Tools imageReducer.image was updated from null to text.jpeg.

Test 2: successful