SW Engineering CSC648/848

WorkWaves

Section 01 , Team 03

Team: Team 03

Member Name	
Brenda Beltran	**Team Lead**
Brandon Flores	**Scrum Master**
Sidney Thomas	**Git Master**
Jaime Guardado	**Support**
Banting Lin	**Front-End Lead**
Abdarrahman Ayyaz	**Back-End Lead**

Milestone 4

November 23 2023

History table (revisions)
(Note: you will update this document based on instructors' feedback so this is important)

Revision Dates
M1 Revised: 10/01/23

1) QA testing

I. Unit Test

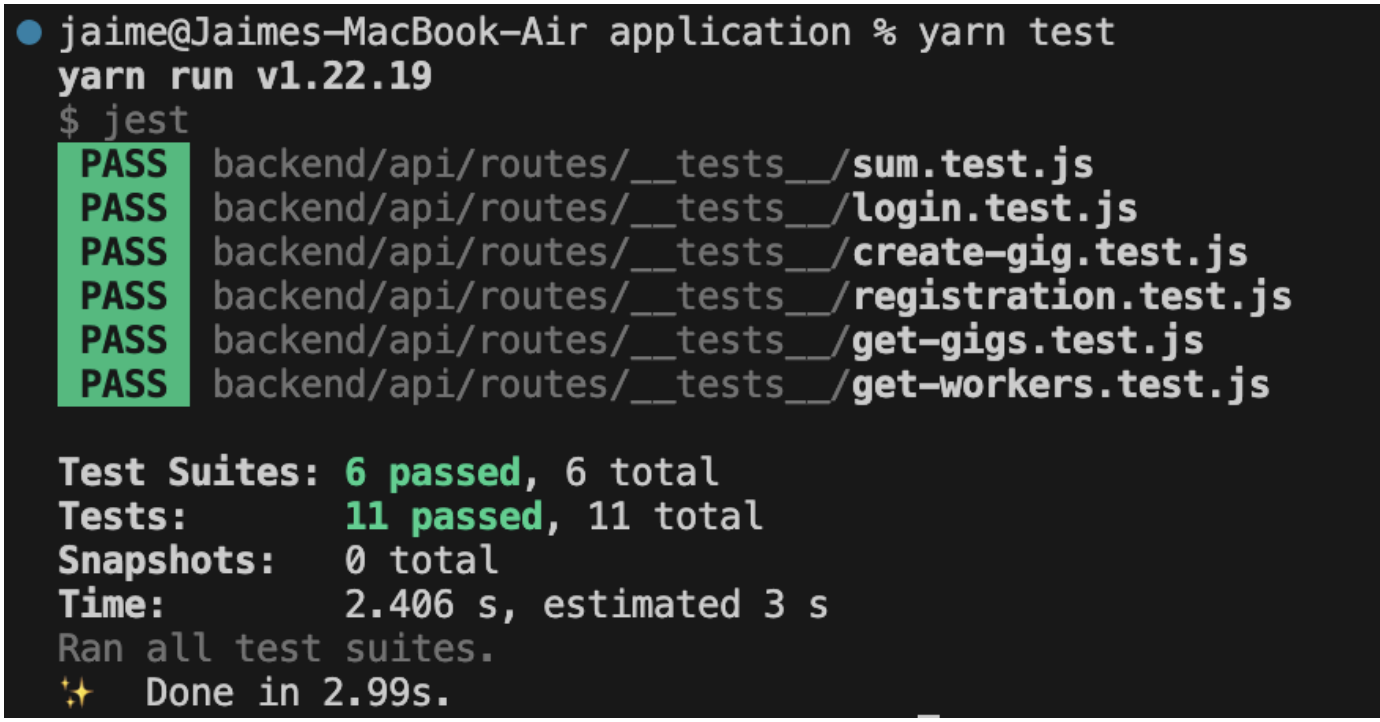
Select 5 P1 features to be tested.

In our QA testing milestone we will be testing the five following features

- **Registration** - will allow the user to make a new account using their created email and password to be able to login and then add themselves to the database.
- **Login** - will allow the user to sign into their personal account using their created credentials and have access to different account settings and personal requests.
- **Creating gigs -** will allow a logged in user to create a new gig card detailing the work they need to be done which will be posted publicly for other users to see and add to their list of gigs.
- **Getting gigs -** will get the information regarding the title and location of the job card that was created for workers to view. It will retrieve the full JSON object for each gig.
- **Getting workers -** will get the information about the worker trying to apply for the gig. It will retrieve the full JSON object for each worker trying to apply.

    - The following image demonstrates the unit test cases we have created for our five features. We have added a sum.test mock test to make sure that the testing implementation was working properly to use for testing our five features.

    - In the same image, we have ran our five unit test cases with our features using the Jest framework as well as Supertest which mocks HTTP requests (GET, PUT, POST, PATCH, DELETE) and as you can see they are all able to be tested successfully without fails or errors.



- After running our unit test cases, we can analyze that our functional statements were the following. Because each api endpoint has statements (conditional etc), all our tests are the functionality of each endpoint (100% functional coverage) and each test has statements in it, rather than in the unit test itself so it could be considered ***100% statement coverage.***

1. **login.test**: takes in mock user credentials to simulate a login via supertest, there's two cases: with and without credentials. Should return 404 and 200 ***100% FUNCTIONAL***
2. **create-gig.test:** takes in mock gig data to simulate a POST via supertest for making a new gig. Two cases. ***100% FUNCTIONAL***
3. **registration.test:** simulates registering with fake credentials, two cases ***100% FUNCTIONAL***
4. **get-workers.test:** simulates a GET request via supertest. two case. ***100% FUNCTIONAL***
5. **get-gigs.test:** simulates a GET request via supertest. two case here too ***100% FUNCTIONAL***

Integration Test Cases

Test Case ID: 001
Test Case Description: Sign up Form

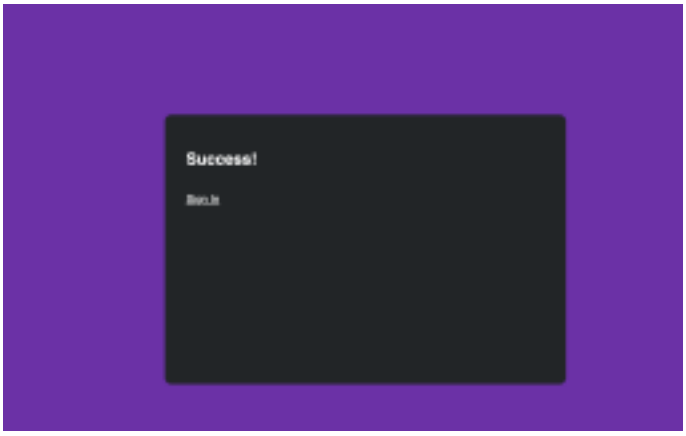
Dates Tested: 11/27/2023

Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working. Then going to the sign up page so that the feature can be tested.
2. Test Data:
  - username:sample1
  - email:sample1@email.com
  - password:Sample1#
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
- The way to execute this test is basically input the username, email and password and then click on the sign up button.

Expected Behavior:
- [Describe what you expect the web application to do]
- The application should successfully create a new user and save the information for sign in.

Actual Behavior:
- [Describe what the web application actually does]
- This was the actual behavior seen.

Visual Documentation: [Attach screenshots, videos, or gifs if applicable]



Test Result:
- PASS

Test Case ID: 002
Test Case Description: Login Form

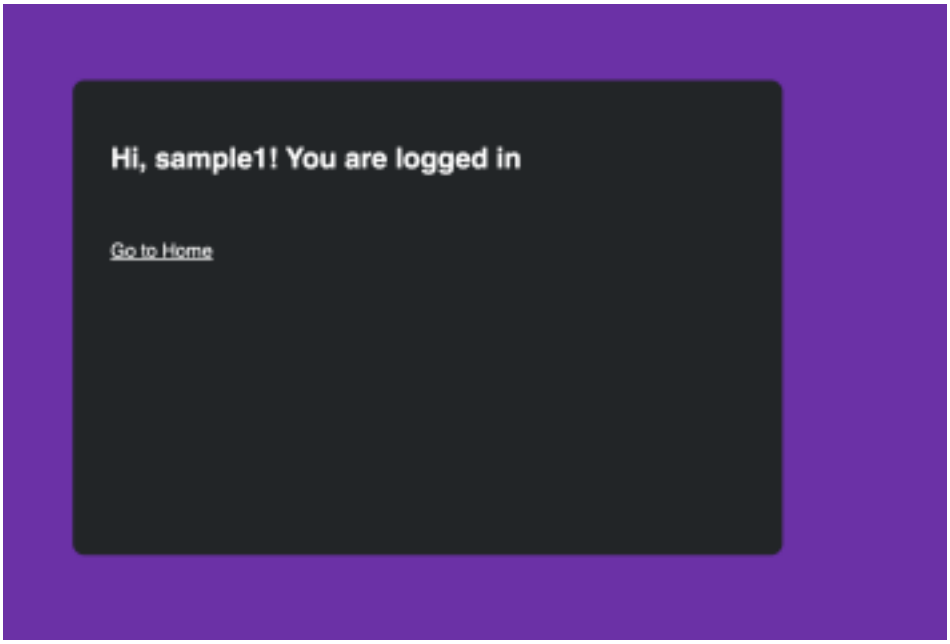
Dates Tested: 11/27/2023

Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working. Then going to the login page so that the feature can be tested.
2. Test Data:
  - username:sample1
  - password:Sample1#
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
- type in username and password and then click on the sign in button.

Expected Behavior:
- [Describe what you expect the web application to do]
- The application should let the user successfully sign in with the credentials.

Actual Behavior:
- [Describe what the web application actually does]
- The user signs in successfully.

Visual Documentation: [Attach screenshots, videos, or gifs if applicable]



Test Result:
- PASS

Test Case ID: 003
Test Case Description: Search Bar(Worker)

Dates Tested: 11/27/2023

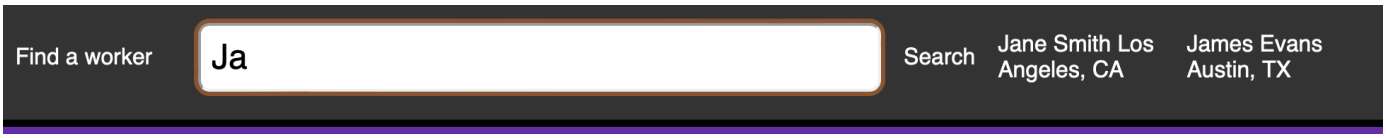
Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the home page. 2. Test Data:
  - N/A
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
  - After going to the home page, use the search bar to search up a specific worker like “James or Jane”.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to find the specific result that I am looking for.

Actual Behavior:
- [Describe what the web application actually does]
- It was able to find the exact search I was looking for, which was computer technician.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable]



Test Result:
- PASS

Assignee (if applicable):
- [Specify the person responsible for debugging if the test fails]
- Some things are being worked on but no bugs currently.

Debugging Status (if applicable):
- [Open/Closed]
- [If debugging is done, provide details on how the bug was fixed, including GitHub URL if applicable]

Test Case ID: 004
Test Case Description: Search Bar(Employer)

Dates Tested: 11/27/2023

Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the home page. 2. Test Data:
  - N/A
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
  - After going to the home page, use the search bar to search up a specific post

like “Computer”.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to find the specific result that I am looking for.

Actual Behavior:
- [Describe what the web application actually does]
- It was able to find the exact search I was looking for, which was computer technician.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable]

Test Result:
- PASS

Assignee (if applicable):
- [Specify the person responsible for debugging if the test fails]
- Some things are being worked on but no bugs currently.

Debugging Status (if applicable):
- [Open/Closed]
- [If debugging is done, provide details on how the bug was fixed, including GitHub URL if applicable]

Test Case ID: 005
Test Case Description: Category Data Sidebar Load

Dates Tested: 11/27/2023
Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the home page. 2. Test Data:

- N/A

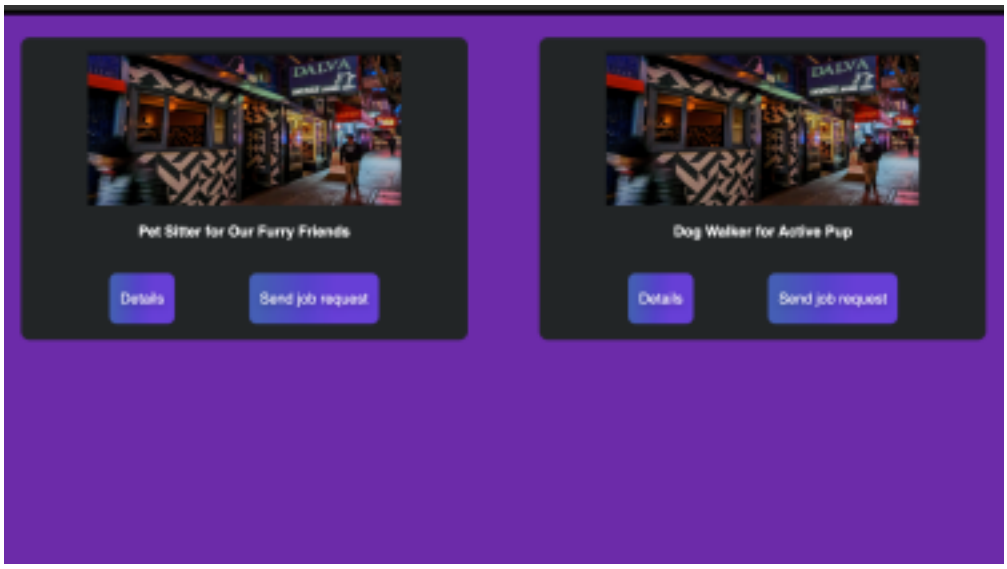
3. Steps to Reproduce:
   - [Provide a step-by-step guide on how to execute the test]
   - After going to the home page, use the category sidebar to go to specific categories.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to find the specific result that I am looking for depending on the category.

Actual Behavior:
- [Describe what the web application actually does]
- It filtered out the category for me which was pets in the case of my test.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable]



Test Result:
- PASS

Test Case ID: 006
Test Case Description: Clicking gig card for more details

Dates Tested: 11/27/2023

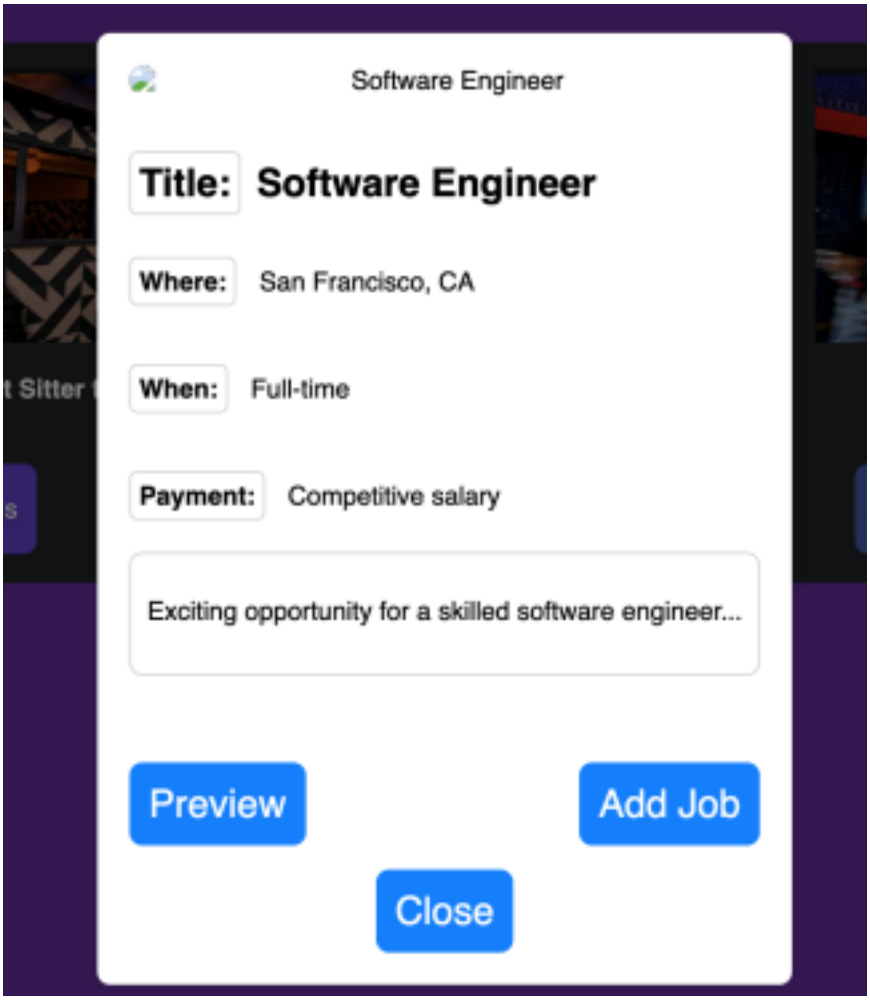
Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the home page. 2. Test Data:
  - N/A
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
  - After going to the home page, click on the button that says details to see details for the post.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to give me a detailed view about the post.

Actual Behavior:
- [Describe what the web application actually does]
- It displayed a nice card detail view to give me information on the post.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable]



Test Result:
- PASS

Assignee (if applicable):
- [Specify the person responsible for debugging if the test fails]
- Some things are being worked on but no bugs currently.

Debugging Status (if applicable):
- [Open/Closed]
- [If debugging is done, provide details on how the bug was fixed, including GitHub URL if applicable]

Test Case ID: 007
Test Case Description: Closing detail card for details

Dates Tested: 11/27/2023

Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the home page. 2. Test Data:
  - N/A
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
  - After seeing the detailed view click on the close button to close the detailed view.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to close the detailed view.

Actual Behavior:
- [Describe what the web application actually does]
- It closed the detailed view successfully and I was able to go back to the home page.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable] N/A

Test Result:
- PASS

Assignee (if applicable):
- [Specify the person responsible for debugging if the test fails]
- Some things are being worked on but no bugs currently.

Debugging Status (if applicable):
- [Open/Closed]
- [If debugging is done, provide details on how the bug was fixed, including GitHub URL if applicable]

Test Case ID: 008
Test Case Description: Nav Bar

Dates Tested: 11/28/2023

Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the login page. 2. Test Data:
  - N/A
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]
  - If on a page like sign up, click on the logo to go to the home page and the nav bar should be on all pages.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to navigate to home page when needed.

Actual Behavior:
- [Describe what the web application actually does]
- I was able to go to the home page successfully from the sign up page.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable] N/A

Test Result:
- PASS

Test Case ID: 009
Test Case Description: Send Job Request

Dates Tested: 11/28/2023
Test Scenario:
1. Prerequisites:
  - Need to ensure that the site url is working and set to show the home page. 2. Test Data:
  - N/A
3. Steps to Reproduce:
  - [Provide a step-by-step guide on how to execute the test]

- After seeing the post click on “send job request”.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to give me an option to send the request and then successfully do it.

Actual Behavior:
- [Describe what the web application actually does]
- It did not function properly.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable] N/A

Test Result:
- FAIL

Assignee (if applicable):
- [Specify the person responsible for debugging if the test fails] - Sidney - Git master
- Banting - Front-end Lead

Debugging Status (if applicable):
- OPEN

Test Case ID: 010
Test Case Description: Switching from Worker to Employer

Dates Tested: 11/28/2023
Test Scenario:
1. Prerequisites:
- Need to ensure that the site url is working and set to show the home page. 2. Test Data:
- N/A
3. Steps to Reproduce:
- [Provide a step-by-step guide on how to execute the test]
- Press the button that says find worker and the feed should show worker posts

and vice versa.

Expected Behavior:
- [Describe what you expect the web application to do]
- I expect it to show the worker posts when it is set to the worker and the employer posts when it is set to the employer.

Actual Behavior:
- [Describe what the web application actually does]
- It did not function and still needs to be worked on.

References:
- [Include links to any related bug reports or documentation]

Visual Documentation: [Attach screenshots, videos, or gifs if applicable] N/A

Test Result:
- FAIL

Assignee (if applicable):
- [Specify the person responsible for debugging if the test fails]
- Sidney - Git master
- Banting - Front-end Lead

Debugging Status (if applicable):
- [Open/Closed]
OPEN
- [If debugging is done, provide details on how the bug was fixed, including GitHub URL if applicable]
---

2) Coding practices:

1. Coding style:
    - The coding style framework we chose to use was ESLint. This tool helps us to stay consistent with proper coding style which we enforce to our teammates for maintaining good code quality. It helps us identify any coding errors and any unalignment which helps us refactor anything specifically off in our code . In our application/.eslintrc.json path file, we created a set of rules guidelines under the “rules” section where we enforced our specific coding style which includes proper indentation, double quoting, and naming.

    - Source files
      ESLINT
      TESTS