

San Francisco State University

SW Engineering CSC648/848 Spring 2020
“Gator Crawl”

Section 02 Team 01

Francis Cruz fcruz2@mail.sfsu.edu

George Freedland

Zain Khan

Jugal Bhatt

Jeffery Wan

Huy Phan

Product summary :

Name of the product :

Gator Crawl

Explicit itemized list of ALL major committed functions :

- Users can login and register
- Users can post/sell items
- Users can search through products
- Users can filter through products
- Users can view a product
- Users can close a post
- Users can buy items
- Users can build profiles
- Admin can approve a posting
- Admin can delete a posting
- Admin can reject a posting
- Admins can approve a sale

What is unique in your product

Something that is unique about our product is that an admin has to confirm if a posting is valid enough for it to be on the website and has to approve of any sale. These are two functions we add to ensure security and safety of the users. The idea of this is to prevent any scams and thefts that may occur.

URL to your product

<http://ec2-3-20-240-35.us-east-2.compute.amazonaws.com/login>

Milestone Documents - M1 to M4

Milestone 1 Feedback

TEAM 1

CEO/CTO Feedback

Thank you for taking all our requirements and vision at heart. It seems that there is an overall understanding of our vision and we ap

Instructor Feedback

Overall Objective of Milestone 1 [Need improvement, On-track, Above Expectations]: **On track with some improvements needed**

ID	Item	Criteria	Feedback
01	Expected size of this document	About 7-10 pages	Good. Please number your pages.
02	Title Page	Format and Structure	Missing History table (revisions) Francis Cruz has 3 roles, team lead and front-back end lead? And then at the end of the document we have all the roles that seems different than the cover page.
03	Executive Summary	~1 page. Why we should fund this project?	Good
04	Personas and main Use Cases	About 1/3 of a page per persona 4-5 <u>main</u> use cases. Descriptive title and number to each use case	Good. Just add a very descriptive title and number for each of your use cases.
05	List of main data items and entities	Name, meaning, usage, etc.	Good. Just the data structure should be all type in, instead of copying and pasting an image (source of the image should be identified for reference).
06	Initial list of functional requirements	High level functions you plan to develop	Good
07	List of non-functional requirements	Performance, expected load, security requirements, storage, availability, fault tolerance	Good. Most of them are exactly what we have in the project document, few interpreted and adjust accordingly. Good.
08	Competitive analysis	3-4 competitive products	Good. Just add the sources of your data for the competition.
09	High-level system architecture and technologies used	Itemized list of all main SW components	Missing the HL system architecture
10	Team and roles	List student names	Good
11	Checklist	Answers to the items	Good

Milestone 2 Feedback

Specific Feedback:

ID	Item	Criteria	Feedback
01	Functional Requirements - prioritized	Expand or repeat functional requirements from Milestone 1 into Milestone 2 w/ reference numbers (1- <i>must have</i> ; 2 – <i>desired</i> ; 3 – <i>opportunistic</i>)	On Track. The reference numbers could be more descriptive, instead of a.i.1. it would be better to use only numbers: 1.1.1. Great prioritization thought.
02	UI Mockups and Storyboards (high level only)	All major use cases from M1. Format is very flexible, recommend hand drawings	On Track. Great UI mockups: Landing, Search, Product, Shopping, User dashboard
03	High level Architecture, Database Organization	Make sure the titles and var. names are in easy to understand plain English and consistent with data definitions. <i>Media storage</i> , APIs (if any), algorithm and SW tools	On Track. Good job organizing the DB and describing each section.
04	High Level UML Diagrams	<i>HL Class diagrams, and Component and deployment</i>	On Track. Good.
05	Key risks for your project at this time	Skills, schedule, technical, teamwork, legal/content	On Track. Skills ✓ Schedule: Jira/Slack ✓ Swagger.io ✓ Open Source ✓
06	Project management	No more than half a page how you managed and plan to M2 and <u>future</u> tasks. <u>Must</u> start using Trello or similar tools for task management	On Track. Jira/Slack ✓

Milestone 3 Feedback

Specific M3 Feedback:

ID	Item	Criteria	Feedback
01	UI and functionality feedback (P1 functions only)	<ul style="list-style-type: none"> - Test main use cases - Check functionality and record issues/observe bugs - Check UI and usability - Check if UI is responsive to change of browser size - Performance in page/image rendering, search - Verify enough WWW pages are implemented and connected 	On Track. Identified bugs should be fixed and all functionality finalized. Simple UI is good. If I remember well the entry point is the login page. Please consider having an open HOME page with general information for the users. Make sure your application works with most of the main browsers.
02	Functionality Check	<ul style="list-style-type: none"> • Home page • Search (including search field validation) • Search results • Filtering • Search Details and maps (if applicable) • Messaging/contact seller/user (if applicable) • Data Upload dashboards (user, admin) • UI responsiveness (resize the browser) • Performance (e.g. display of results list) 	On Track. Most of the functions were ready. Finalize those pending to fully work, front-end integrated with back-end.
03	Brief review of code, github, database etc.	<ul style="list-style-type: none"> • All key DB tables completed (users, items, messages, categories etc.) • Search fully working • Home page and search results integrated with back end • Search field input validation to allow only up to 40 alphanumeric characters (letters and numbers) • Search arguments persistent • Be ready to show examples of code so coding style and code comments can be checked. 	Need Improvement. Make sure you work on the integration. Add comments to your code documentation.

04	Project status	<ol style="list-style-type: none"> 1. <i>Teamwork:</i> 2. <i>Risks</i> 3. <i>Coding practices</i> 4. <i>Usage of proper SE code management practices</i> 5. <i>How did you address site security and safe coding practices</i> 6. <i>Digital content</i> 7. <i>Other</i> 	On Track. Overall good teamwork so far. Keep it up. Make sure to address site security.
05	List of P1 features committed for delivery agreed	<i>Team already reviewed and committed for final P1 list?</i>	On Track. I counted 14 items in your P1 list. Make sure all of them are fully implemented.
06	Overall Instructor Review	<ul style="list-style-type: none"> • Git/Github organization • Git/Github usage • Code documented • MVC/OO patterns followed up • Frameworks • Database organization • Blobs being used? • Adherence to best practices of security • Efficiency • Other 	On Track. Good Documentation and Organization is part of the SE Objectives. Make sure your github reflects that.

Milestone 4 Frozen Version

1) Product summary (e.g. how would you market and sell your product – about ½ page)

Name of the product

Gator Crawl

Explicit itemized list of ALL major committed functions

- Users can login and register
- Users can post, search, view, close, and buy items
- Users can bid on items if product permits
- Users can build profiles
- Users can message each other
- Admin can delete, reject, and accept items for sale

URL to your product accessible to instructors, on deployment server

Link may be subject to change. AWS changes IP of server when stopped and started again.

Current URL for Milestone 3: <http://ec2-3-133-87-174.us-east-2.compute.amazonaws.com>

2) Usability test plan - Search /Filter Functionality

Test Objective

We are going to discuss the usability test plan for our search and filter functionality. We are focussing it on this test plan because search is of the main features of many online shopping stores. The search and filtering functionality is important. Since it helps the users find their products within seconds of looking. If we don't have a good search or filter function implemented in our website; it will be hard for the user to find the item they are looking for. Which would lead to a bad user experience and interface design. Having a bad user experience and interaction will cause the users to be upset with our website because it won't be as efficient as we like. We want to ensure if the user types in “computer” in the search bar; that they are only presented with all products that have the keyword computer in their name or description. Or if they filter by “electronic”, then all electronic results will be displayed.

Test Background and Setup

System Setup for Search: To set up for our system to run the Search system is pretty simple. First we need to ensure that our search functions are up and running on our website. The second thing we need to set up is to ensure our text forum is allowing for user input. The third step is to ensure the system takes

what was captured from the text forum and search our database.. The system will take what the user entered as an input and try to match it with the best products. The final step up for search, the system will retrieve all items matching that text forum input and display the results to the user. After finding the best match products; it will display those products to the user.

System Setup for Filter: To set up for our system to run the Filter system is similar to the Search. First we need to ensure that our filter functions are up and running on our website. The second thing we need to set up is to ensure our drop down menu is allowing for user clicks.. The third step for the setup, is to ensure the system takes what was captured and filters our database for items matching that description.. The final step up for the filter is, when the system will retrieve all items matching that filter input and display the results to the user.

Starting point : Our starting point is to ensure that our website is taking in user inputs from the search bar and the drop-down menu.

Intended Users : This function is going to be intended for all users who have access to the site.

Both users and admins will be allowed access to this functionality.

URL : <http://localhost:3000/searchresults>

Usability Task Description

The first step to test the Search and filter functionality is to go to our website and log into your account.

. For ***Search*** -. The first step is to click on the search tab on the top ribbon. The next step is to enter an item you are looking for in the search bar. Then hit enter to finalize that search. (e.g If you want to look for shoes; then type in shoes in the search bar and press enter). This will show all matches with the keyword of shoes.

For ***Filter*** - The first step is to click on search on the top ribbon. The next step is to click on the drop down menu. Then select one of the following options and press enter.(e.g If you are looking for clothes on our website, then select “Apparel” from the drop down menu and hit enter). This will show all results matching apparel.

We would measure effectiveness based on the search results generated by the search and filter functions. If you can find what you want by your search. Then it had strong effectiveness. Otherwise it has poor effectiveness.

We would measure efficiency based on time spent looking for a product. If it took a shorter time to find the product you are looking for by searching or filtering than it has high efficiency. However it took a shorter time by looking through the products one by one then it has low efficiency.

Lickert Subjective Test :

1. The search function was easy to use? (Strongly Disagree to Strongly Agree)
2. The filter function was easy to use? (Strongly Disagree to Strongly Agree)
3. You found what you were looking for with the search results? (Strongly Disagree to Strongly Agree)

3) QA test plan - max 2 pages

For the same function you chose for the usability test, write a QA test plan (check class slides), with sections as follows:

- Test objectives:
 - Product Search and filter functionality
- HW and SW setup (including URL):
 - Mac using using Google Chrome and Safari
- Feature to be tested
 - Product Search and filter functionality
- QA Test plan:

Test #	Test Title	Test Description	Test Input	Expected Output	Test Results
1	Search All	Test find all functionality	none	Get 4 products	PASS
2	Search with Category Filter	Test search functionality with just Category Filter	Category dropdown menu with the category "clothing" selected.	Get 1 product with the name "Nike Air Max"	PASS
3	Search with Category Filter and Query	Test search functionality with Category Filter and Query	Category dropdown menu with category "Electronics" and Query "TV"	Get 1 product with the name "Old TV"	PASS
4	Search All (safari)	Test find all functionality	none	Get 4 products	PASS

5	Search with Category Filter (safari)	Test search functionality with just Category Filter	Category dropdown menu with the category "clothing" selected.	Get 1 product with the name "Nike Air Max"	PASS
6	Search with Category Filter and Query (safari)	Test search functionality with Category Filter and Query	Category dropdown menu with category "Electronics" and Query "TV"	Get 1 product with the name "Old TV"	PASS

4) Code Review:

Note: peer review must include checking for basic header and in-line comments

- Our sole coding language for front end and back end is JavaScript. For the backend we use NodeJS and for the frontend we use React with Redux. For development we use the Atom IDE on which we install the linter-eslint package along with eslint-config-airbnb for rules. This essentially gives every developer the same errors not just for syntactical mistakes but for general indentation and structure. All the rules are in the .eslint files.

Search and filter

ZK

Zain Khan <zainkhanx93@gmail.com>
To George William Freedland

Reply Reply All Forward ...

Tue 4/21/2020 7:55 PM

```

import React, { useState, useEffect } from 'react';
import axios from 'axios';

import MainNavBar from '../Nav/MainNavBar';

const SearchResults = () => {
  const [query, setQuery] = useState("");
  const [filter, setFilter] = useState("");
  const [categories, setCategories] = useState([]);
  const [products, setProducts] = useState([]);

  const handleSearch = () => {
    if (query) {
      if (filter) {
        // code for searching with Search and Filter function ( look up prod by name and category)
        axios.get(`/api/products/${query}/${filter}`).then((res) => {
          setProducts(res.data);
        });
      } else {
        // code for searching with Search function (Can search for anything by name)
        axios.get(`/api/products/${query}`).then((res) => {
          setProducts(res.data);
        });
      }
    } else if (filter) {
      // code for searching with Filter function ( Eletronics, Apperel, etc)
      axios.get(`/api/products/all/${filter}`).then((res) => {
        setProducts(res.data);
      });
    }
  };
};

```

Search and filter



Zain Khan <zainkhanx93@gmail.com>
To George William Freedland

Reply Reply All Forward ...

Tue 4/21/2020 7:55 PM

```
import React, { useState, useEffect } from 'react';
import axios from 'axios';
```

```
import MainNavBar from '../Nav/MainNavBar';
```

```
const SearchResults = () => {
  const [query, setQuery] = useState("");
  const [filter, setFilter] = useState("");
  const [categories, setCategories] = useState([]);
  const [products, setProducts] = useState([]);
```

Nav folder should be renamed to Navigation, let's use full variable names when possible so it's easy to manage/go through the application

```
const handleSearch = () => {
  if (query) {
    if (filter) {
      // code for searching with Search and Filter function ( look up prod by name and category)
      axios.get(`/api/products/${query}/${filter}`).then((res) => {
        setProducts(res.data);
      });
    } else {
      // code for searching with Search function (Can search for anything by name)
      axios.get(`/api/products/${query}`).then((res) => {
        setProducts(res.data);
      });
    }
  } else if (filter) {
    // code for searching with Filter function ( Eletronics, Apperel, etc)
    axios.get(`/api/products/all/${filter}`).then((res) => {
      setProducts(res.data);
    });
  }
};
```

Comment which component are we sending the products response to

spelling

Every axios request should also have a .catch(err => {}) added after the .get() so in case there is a bad request we can display it neatly on the front instead of giving an error to the browser.

spelling, list all categories

.catch(err => {});

5) Self-check on best practices for security – ½ page

- List major assets you are protecting
 - Passwords
- Say how you are protecting each asset (1-2 lines of text per each)
 - We are hashing the user passwords in the backend before saving to the database
- Confirm that you encrypt PW in the DB
 - We are encrypting PW in the database
- Confirm Input data validation (list what is being validated and what code you used) – we request you validate search bar input for up to 40 alphanumeric characters;
 - Inputs are validated in the frontend

6) Self-check: Adherence to original Non-functional specs – performed by team leads

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class, some may be chosen by the student team but all tools and servers have to be approved by class CTO).

DONE

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers

DONE

3. Selected application functions must render well on mobile devices

DONE

4. Data shall be stored in the team's chosen database technology on the team's deployment server.

DONE

5. No more than 50 concurrent users shall be accessing the application at any time

6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.

ON TRACK

7. The language used shall be English.

DONE

8. Application shall be very easy to use and intuitive.

9. Google analytics shall be added

ON TRACK

10. No email clients shall be allowed

DONE

11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

DONE

12. Site security: basic best practices shall be applied (as covered in the class)

ON TRACK

13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development

ON TRACK

14. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only"* at the top of the WWW page. (Important so not to confuse this with a real application).

DONE


End of Frozen Milestone 4

Product Screenshots

Login

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.

Gator Crawl



Login


[Forgot password?](#)

[Create new account](#)

Registration

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.

Gator Crawl







Register

[Already have an account? Click here to log in.](#)

Home Page

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.

GATOR CRAWL



Hi the!

Search Gator Crawl


Post

Search Results


Filters

Categories:
All


Admin Panel




Nike Air Max
Price: \$100
Category: Clothing
Posted on: 2020-05-10




iClicker
Price: \$100
Category: Electronics
Posted on: 2020-05-10




Vintage SFSU Shirt
Price: \$100
Category: Electronics
Posted on: 2020-05-10



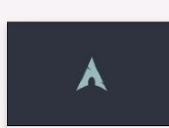
wallpaper
Price: \$2
Category: Home & Garden
Posted on: 2020-05-11




idk
Price: \$100
Category: Electronics
Posted on: 2020-05-11



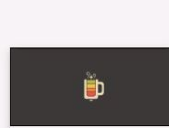
asdfasdf
Price: \$100
Category: Clothing




1234
Price: \$1234
Category: Clothing



123
Price: \$123
Category: Electronics



123
Price: \$123
Category: Electronics







Nike Air Max
Price: \$100
Category: Clothing

Create A Posting

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.

GATOR CRAWL



Hi the!

Post

Search Results

Filters

Categories:
All

Admin Panel

Enter Your Item Info

Photo:

Choose File | No file chosen

Product Name:

Product Name

You must enter a name for your product

Description:

Description

You must enter a description


Price:

\$ 100

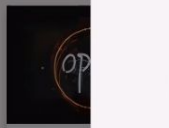
Category:

Collectables & Art


Create Post



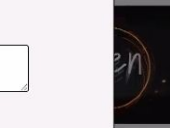
Nike Air Max
Price: \$100
Category: Clothing
Posted on: 2020-05-10




iClicker
Price: \$100
Category: Electronics
Posted on: 2020-05-10




Vintage SFSU Shirt
Price: \$100
Category: Electronics
Posted on: 2020-05-10



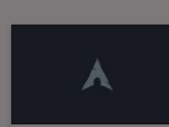
wallpaper
Price: \$2
Category: Home & Garden
Posted on: 2020-05-11




idk
Price: \$100
Category: Electronics
Posted on: 2020-05-11



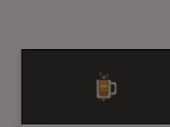
asdfasdf
Price: \$100
Category: Clothing




1234
Price: \$1234
Category: Clothing



123
Price: \$123
Category: Electronics




123
Price: \$123
Category: Electronics




Nike Air Max
Price: \$100
Category: Clothing

Viewing a Product

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.

GATOR CRAWL 

Back To Home




Nike Air Max

Description: First Product
Category: Clothing
Price: \$100
Sold By:lcruz2@mail.sfsu.edu
Date Posted: 2020-05-10


[Bookmark](#) [Message Seller](#)



Messages

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.

GATOR CRAWL 


No new messages

Chat App 

 Search 

Chats Users

P Public

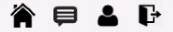
admin@mail.sfsu.edu 

Type something interesting

Send

Shopping Cart

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.



My Shopping Cart

Profile

My Cart

My Products

About Us



Nike Air Max

First Product

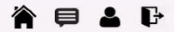
\$ 100

Message

Delete

Your Products

SFSU Software Engineering Project CSC 648-848, Spring 2020. For Demonstration Only.



[Back To Home](#)



Product Name

Description: Description

Category: Collectables & Art

Price: \$100

Sold By: admin@mail.sfsu.edu

Date Posted: 2020-05-12

This is your product

Categories

Result Grid

ID	Name	CreatedAt	UpdatedAt	ProductID
1	Clothing	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
2	Electronics	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
3	Collectables & Art	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
4	Home & Garden	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
5	Sporting Goods	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
6	Toys & Hobbies	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
7	Other	2020-05-10 23:39:39	2020-05-10 23:39:39	NULL
8	Clothing	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
9	Electronics	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
10	Collectables & Art	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
11	Home & Garden	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
12	Sporting Goods	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
13	Toys & Hobbies	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
14	Other	2020-05-11 23:03:25	2020-05-11 23:03:25	NULL
15	Clothing	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
16	Electronics	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
17	Collectables & Art	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
18	Home & Garden	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
19	Sporting Goods	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
20	Toys & Hobbies	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
21	Other	2020-05-11 23:05:16	2020-05-11 23:05:16	NULL
NULL	NULL	NULL	NULL	NULL

Categories 1 Apply Revert

Action Output

	Time	Action	Response	Duration / Fetch Time
✓ 13	20:27:25	SELECT * FROM testdb.Categories LIMIT 0, 1000	21 row(s) returned	0.0040 sec / 0.000...

User

[illegible]

Products

100%

1:1

Result Grid

Filter Rows:

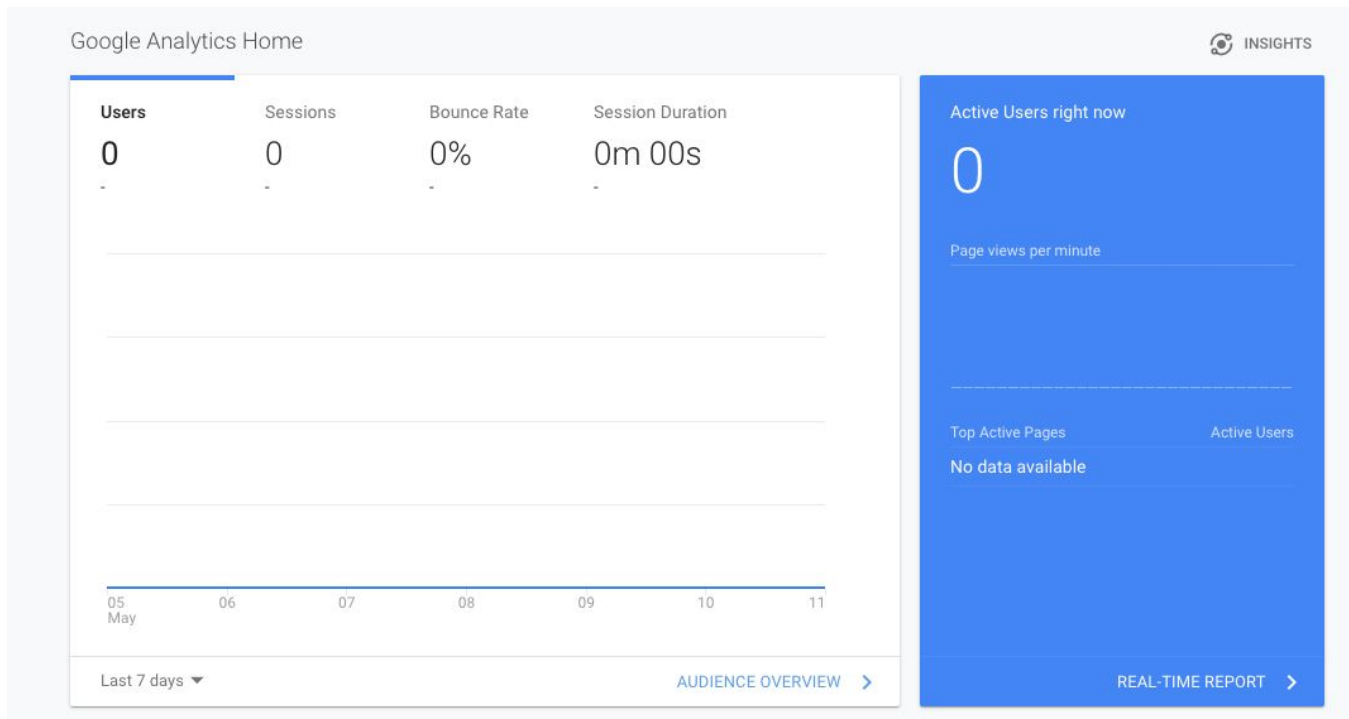
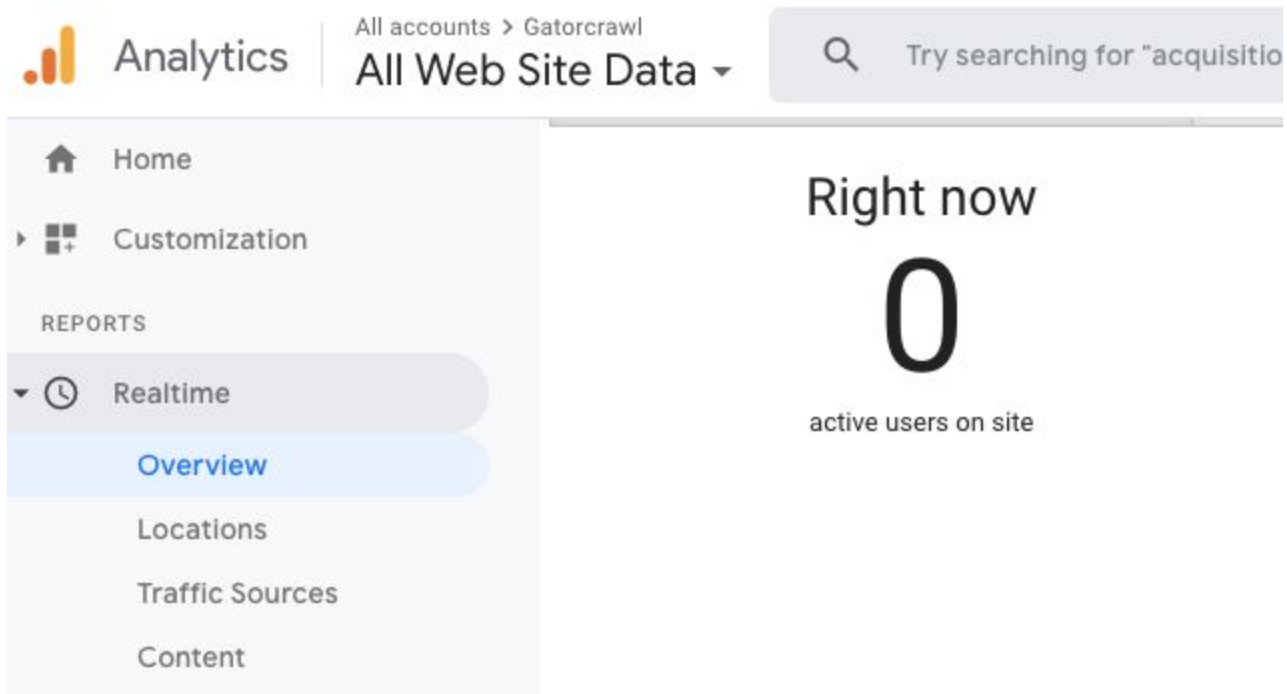
Q Search

Edit:

Export/Import:

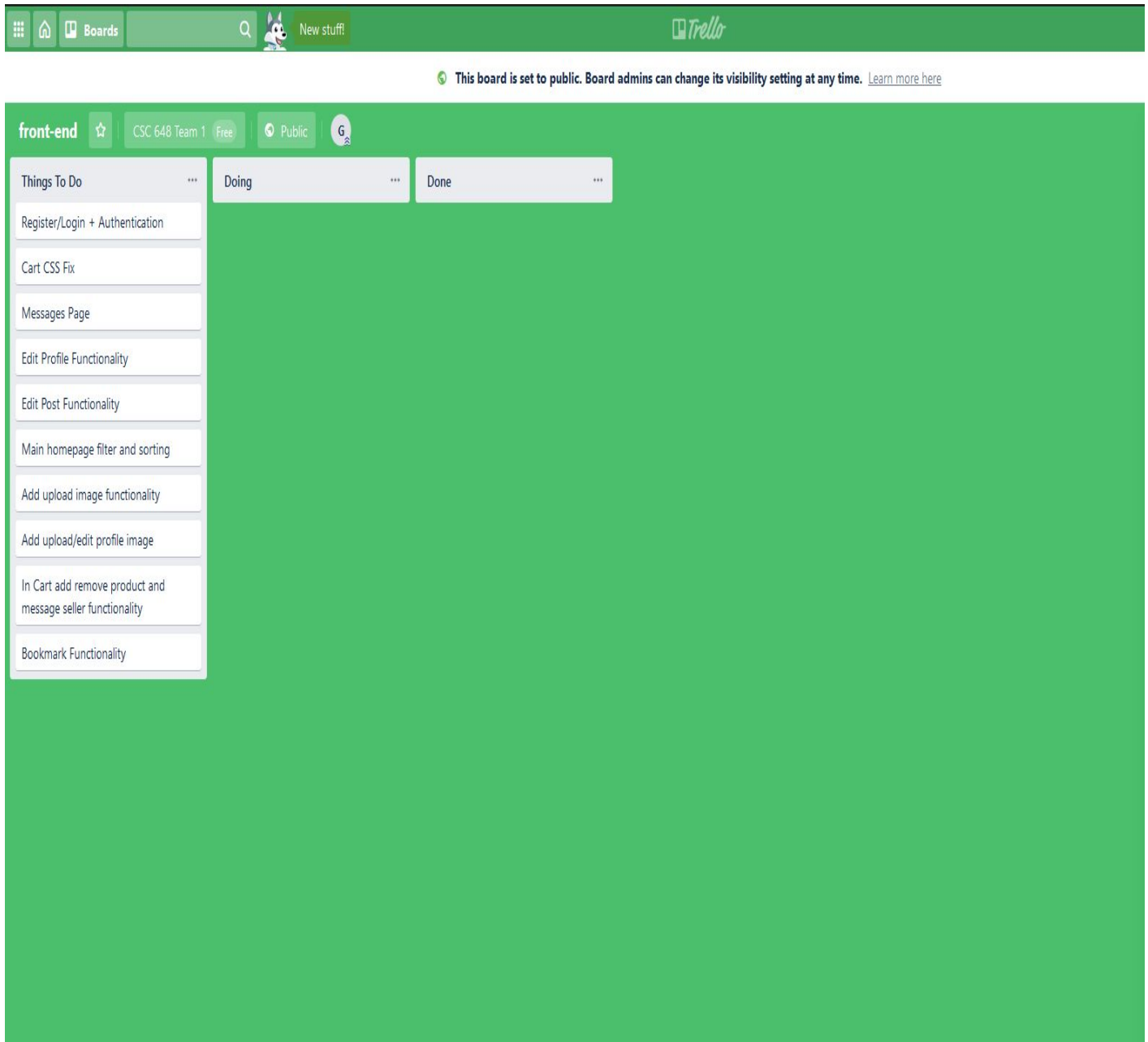
	id	buyerid	sellerid	productid	price	approved	createdAt	updatedAt	
▶	2	1	2	4	20	1	2020-05-10 23:39:39	2020-05-11 23:02:23	
	3	1	2	2	100	1	2020-05-11 23:03:25	2020-05-11 23:03:36	
	4	1	2	4	20	1	2020-05-11 23:03:25	2020-05-11 23:04:13	
	5	1	2	2	100	1	2020-05-11 23:05:16	2020-05-11 23:05:27	
	6	1	2	4	20	1	2020-05-11 23:05:16	2020-05-12 01:50:04	
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

Google analytics plot for your WWW site (1 page)



Screenshots of the task management system (like Trello)

Front-end



Team member contributions

Jugal Contribution

Contributions to the overall project

Jugal Nikul Bhatt <jbhatt@mail.sfsu.edu>

To: Zain Khan <zainkhan93@gmail.com>, Huy Tuong Phan <tphan7@mail.sfsu.edu>

Cc: George William Freedland <gfreedland@mail.sfsu.edu>, Francis Niko Cruz <fcruz2@mail.sfsu.edu>, Jeffrey Sze Min Wan <jwan@mail.sfsu.edu>

a)

Contributed to backend documentation on swagger.io

Implemented google analytics for website

Implemented google analytics for custom events such as sign up.

worked on final documentation

b)

I have had one commit

From: Zain Khan <zainkhan93@gmail.com>

Sent: 11 May 2020 19:55

To: Huy Tuong Phan <tphan7@mail.sfsu.edu>

Cc: George William Freedland <gfreedland@mail.sfsu.edu>; Francis Niko Cruz <fcruz2@mail.sfsu.edu>; Jeffrey Sze Min Wan <jwan@mail.sfsu.edu>; Jugal Nikul Bhatt <jbhatt@mail.sfsu.edu>

Subject: Re: Contributions to the overall project

Everything looks good to me. Good Job everyone.

On Mon, May 11, 2020 at 7:52 PM Huy Tuong Phan <tphan7@mail.sfsu.edu> wrote:

Great Job George

George Contribution

From: George William Freedland <gfreedland@mail.sfsu.edu>

Sent: Monday, May 11, 2020 3:56 PM

To: Francis Niko Cruz <fcruz2@mail.sfsu.edu>; Zain Khan <zainkhan93@gmail.com>

Cc: Huy Tuong Phan <tphan7@mail.sfsu.edu>; Jeffrey Sze Min Wan <jwan@mail.sfsu.edu>; Jugal Nikul Bhatt <jbhatt@mail.sfsu.edu>

Subject: RE: Contributions to the overall project

a)

George Freedland – Front End Team Lead - Contributions

- Set up front end boilerplate project.
- Set up React + Redux Frameworks.
- Worked together on every milestone documentation with team.
- Assigned tasks to team with Trello + Slack.
- Reviewed and corrected pull requests and branches from peers.
- Sent documentation links to teammates.
- Made tutorial videos for teammates on how to code in React + Redux <https://www.youtube.com/channel/UC0Ysufab5BhO3bjxAuwpRkg>
- Set up routes for application.
- Connected all backend routes to the frontend.
- Added several backend routes when necessary.
- Implemented the design and functionality for Navigation, Modals, Forms, Home page, Profile Page, Individual Product Page, and My Products page.
- Added an admin panel for approving or denying products
- Added image upload functionality
- Added Login/Register/Logout functionality and design with Authorization, Validation, and Cookies.

b)

I had 18 branches with 35 commits.

From: Francis Niko Cruz <fcruz2@mail.sfsu.edu>

Niko Contribution

From: Francis Niko Cruz <fcruz2@mail.sfsu.edu>

Sent: Sunday, May 10, 2020 3:42 PM

To: Zain Khan <zainkhanx93@gmail.com>

Cc: Huy Tuong Phan <tphan7@mail.sfsu.edu>; George William Freedland <gfreedland@mail.sfsu.edu>; Jeffrey Sze Min Wan <jwan@mail.sfsu.edu>; Jugal Nikul Bhatt <jbhatt@mail.sfsu.edu>

Subject: Re: Contributions to the overall project

a)

Github Master:

- Set up system for pull request reviews.
- Merged all working code into master.
- Reverted/fixed any broken code that happened to go live on Master branch.

Back End Team Lead:

- Set up initial boilerplate for Back End.
- Created Models and controllers for Users, Products, Categories, and Sales
- Created CRUD routes for Users, Products, Categories, and Sales
- Created routes for image uploading to amazon s3 bucket.
- Created seeder files for use in development cycle.
- Contributed to backend documentation on swagger.io
- Reviewed all pull requests from back end developers.
- Provided assistance and guidance to those that needed help learning the backend stack.

Team Lead:

- Set up communication channels (slack), task management boards (trello), and project wiki (notion)
- Set up AWS EC2 server for the project
- Installed necessary packages for AWS server and maintained server whenever Gator Crawl needed to go live.
- Set up AWS s3 bucket for image storage
- Created initial documents and delegated tasks for each milestone.
- Maintained communication to developers and team leads on when deadlines for tasks are due while also being flexible.

b)

I had 33 commits, consisting mostly of merges from dev branches into master.

Zain Contribution

On May 10, 2020 at 2:26 PM, <Zain Khan> wrote:

Sorry, there is a correction to my initial email.

A)

- Designs mock ups
- Created a list of functionality list
- Worked on Cart.js
- Worked on Cart.css
- Fixed the scalability issues on the Cart files
- Organized meetings outside of classroom
- Milestone contributions
- Communications
- Worked on the final documentation

b)

- I have had 13 commits to the master branch mainly based of the front end Cart design and viewing issues

From,
Zain Khan

Huy Contribution

On Sun, May 10, 2020, 1:47 PM Huy Tuong Phan <phan7@mail.sfsu.edu> wrote:

A/ I modified User Model, Controller for these functions from backend:

- + Register users
- + Login function
- + Authentication
- + Chat Application

I do help frontend with Chat Application

B/ I have none commit on GitHub, all of my commits go under Niko account because he is the backend team lead. My branch have to work properly to be able to push to master so Niko have to tested himself then he did the pushed. Therefore, all my commits went under his contribution but I have a list of my commits.

Commits on Apr 6, 2020

add register function

HuyPhan committed on Apr 6

Commits on Apr 7, 2020

need to fix passport.js local strategy to auth user

HuyPhan committed on Apr 7

Commits on Apr 20, 2020

fix login function

HuyPhan committed 20 days ago

Commits on May 5, 2020

update old stuff

HuyPhan committed 5 days ago

Commits on May 7, 2020

add message.scss

HuyPhan committed 3 days ago

fix backend server.js ERR CONNECTION

HuyPhan committed 3 days ago

move chat heading

HuyPhan committed 3 days ago

Update for message function

HuyPhan committed 3 days ago

Commits on May 7, 2020

fix merged problem

HuyPhan committed 3 days ago

Commits on May 8, 2020

Updated

HuyPhan committed 2 days ago

Update Chat function

HuyPhan committed 2 days ago

Original Email

From: Zain Khan <zainkhan93@gmail.com>

Sent: Saturday, May 9, 2020 9:54 PM

To: Francis Niko Cruz <fcruz2@mail.sfsu.edu>; George William Freedland <gfreedland@mail.sfsu.edu>; Huy Tuong Phan <tphan7@mail.sfsu.edu>; Jeffrey Sze Min Wan <jwan@mail.sfsu.edu>; Jugal Nikul Bhatt <jbhatt@mail.sfsu.edu>

Subject: Contributions to the overall project

Please sure to "Reply to all"

This is a collection of e-mails that each team member sends to ALL of their respective team members (so that all team members can see all the mail on this) outlining:

a) His/her contributions to team project and teamwork (technical and any other) in no more than half a page – point format is OK.

b) Number of submissions he/she made to github team dev. branch

Length of overall e-mail is ¼ page max. This is kind of "peer review" – this makes sure you are comfortable with what you have done since all other team members will see it. Team leads and front-end leads: mention your role too and explain what you have done (be brief)

NOTE: You must submit copies of raw (original) e-mails sent from each team member to all other team members showing e-mail address lines (sender and recipients)

a)

- Designs mock ups
- Created a list of functionality list
- Worked on Cart.js
- Worked on Cart.css
- Fixed the scalability issues on the Cart files
- Worked on the final documentation

b)

- I have had 13 commits to the master branch mainly based of the front end Cart design and viewing issues

9) Post analysis – lessons learned (one-page max)

a) Main Challenges

The one main challenge the majority of our team members faced is the lack of knowledge regarding the frameworks and languages chosen for the project. A lot of us were inexperienced with Javascript and the frameworks of React and Node leading us to have to learn the technologies on the fly while creating Gator Crawl. This in turn led to slow production of the product.

Another main challenge we had was time management. With the new challenge of learning a new language and framework, we waited until the very last few weeks to start learning what we do not know. We did not use the first couple of weeks where we did not code to start learning and as a result our product did not get completely finished. Those that did know how to use React and Node were ultimately overwhelmed with the amount of work they had to do because a lot of the delegated tasks could not be finished in time.

b) What we can do better to address the challenges.

We have greatly improved our knowledge of React and Node and will be prepared for any future assignments regarding these technologies.

It would also be very beneficial for us all if we get started earlier on every coding project from now on.