193Tees

User guide

Preface

Team:

Rose-Marie Eter - 4th year Computer Science & Engineering
Jade MacDonnell - 4th year Computer Science
Phuong Nguyen - 4th year Computer Science & Engineering
Christopher Ta - 4th year Computer Science

In collaboration with:

Travis Heppe - Software Engineer @ Google

Audience:

This document is intended for future developers and users of the application.

Overview of Product

193Tees is a platform for local student artists to showcase their creativity and gain recognition as well as to make extra side cash. This website offers a mobile responsive shopping interface for shoppers in addition to an administrative user interface for business owners. The site's system architecture is based off the Javascript MEAN web stack (MongoDB - NoSQL database, Express.js - route handling framework, Angular.js - front end views and applications, Node.js - Javascript runtime environment). Key features of our website include: secure credit card transactions, gradual rollouts of new versions, an administrative dashboard interface, probe alerts and pager rotations to ensure site reliability, and data analytics to track the effectiveness of our advertisements. Each feature was developed using existing components and API's, in addition to a suite of Google tools (AdWords, Analytics, AppEngine).

Installation Instructions

As a user, there are no special installation instructions to access the platform. Simply visit https://www.193tees.com/.

For instructions on further development of this project, follow the steps below:

1. You will be making 3 files which are to be hidden from the public repository: .env, keys.json, and your equivalent ECS193-8e7b5d077bda.json. Refer to the example files on github labeled ".env.example", "keys.json.example", "ECS193-8e7b5d077bda.json.example". You will need to sign up for credentials on each of the following platforms to get the information necessary to make your

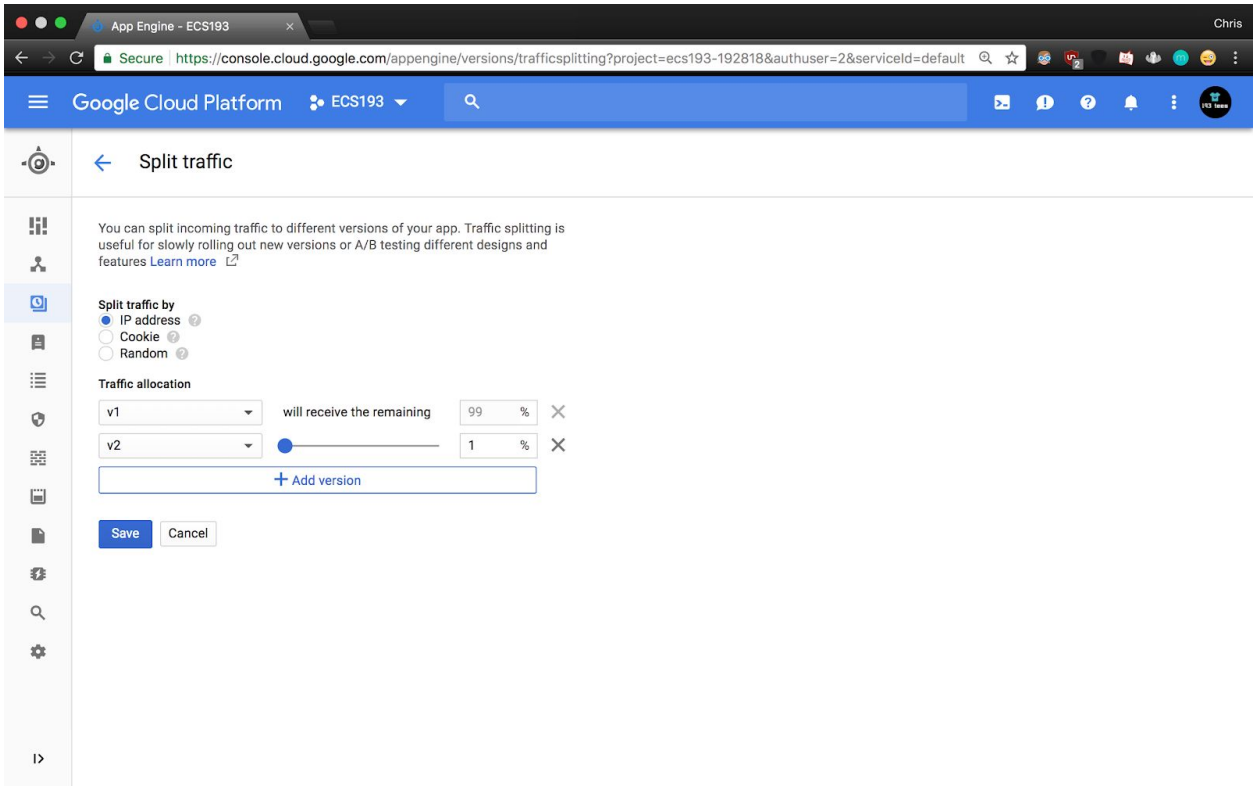
key files: https://developers.facebook.com/ , https://developer.paypal.com/ , https://mlab.com/ , https://analytics.google.com/ .

2. Clone the 193prober repository on your local machine for development, and on an AWS EC2 instance. This is the SRE prober hosted on AWS to check repeated uptime. The repository should be private as the source code contains sensitive information, including API keys and phone numbers of your team members.
    a. Use SSH to connect to the AWS instance and clone the 193prober repository
    b. Download and install the R dependencies twilio and httr in your permanent environment. Additionally, download and install the utility “screen”. This is useful for running the prober in the background.
    c. Run the prober by typing “Rscript prober.R”
    d. You can put the running script in the background by entering ctrl-d
    e. You can get the running script screen back by typing “screen -r”
3. Clone the repository from https://github.com/CTaCDE/bitcommerce
4. Change into the directory and install the node dependencies by typing “npm install”
5. Install the google command line tools (this is for deployment and releasing versions through flags): https://cloud.google.com/sdk/
6. Install the Jasmine testing framework with “npm install -g jasmine”
7. At this point, the repository is now setup and ready for local development. Change the file “bin/www” on line 3 to state “var localdev = 1”. This creates the localhost environment with a secure connection to develop on API’s that require a secure environment (Facebook).
8. To deploy new versions to google cloud, first reset changes to “bin/www”, and ensure that your working repository is clean and pushed to origin/master on github. To deploy a new live version and update the existing site, type the following command: “gcloud app deploy -v v1”.
9. To deploy a separate version to google cloud, checkout the github branch that contains the version2 code. Be sure to rebase this branch with master to include the latest changes. Type the following command to deploy: “gcloud app deploy -v v2 --no-promote”.
10. Once the versions have been deployed, head to https://cloud.google.com/ and set desired traffic splitting settings in Navigation Menu -> App Engine -> Versions -> Split Traffic

A/B Testing and Traffic Splitting

Splitting traffic on google cloud platform allows the project members to control new versions through flags over binaries. This allows for fast rollback and gradual traffic migration to ensure site reliability. The project has been setup to allow for quickly splitting traffic between multiple versions. Remember when deploying a second version without making it live, deploy it with the following command on the second version branch: $ gcloud app deploy -v v2 --no-promote

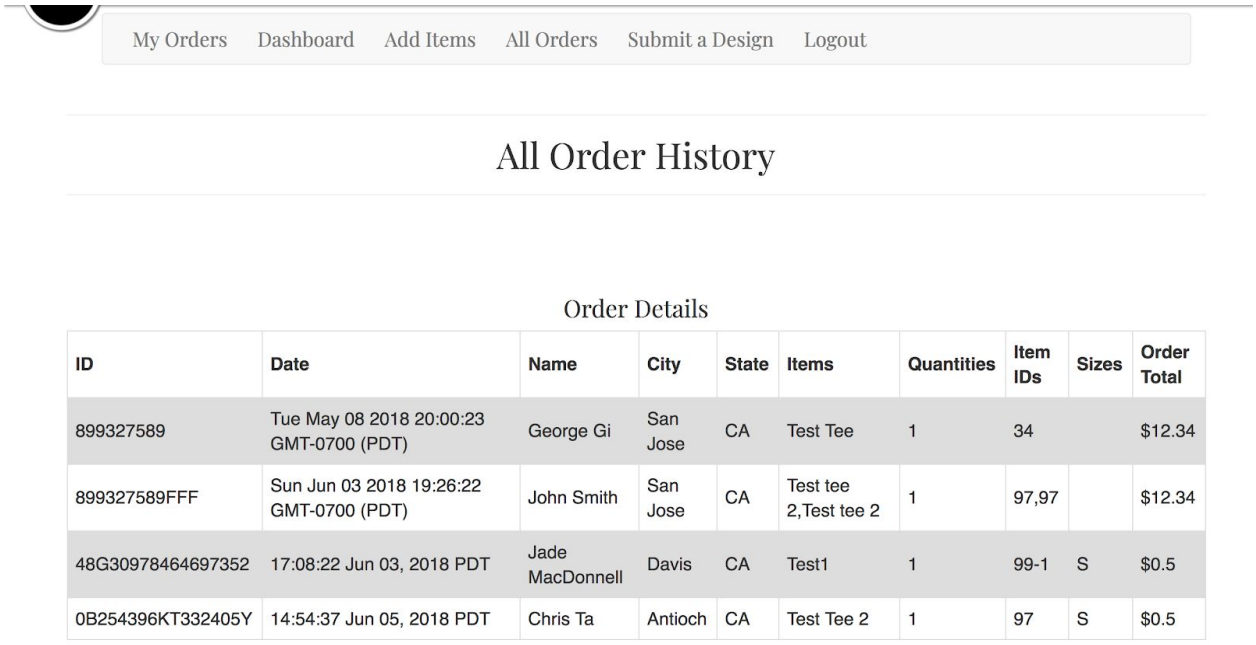
To split traffic, go to the App Engine Console -> Sidebar menu -> App Engine -> Versions -> Split traffic to see the following page:



It is strongly suggested to split traffic by IP address, this way the user will stay on one version throughout their browsing experience. At this time, there is no mechanism to differentiate analytics and sales between the two versions. The current analytics tag will aggregate both versions and combine them into a cumulative result.

Tracking Site Orders

For Admins, there is an “All Order” page which shows a table of all orders that were ever made. The table contains: Order ID, Date, Name of customer, City, State, Item name, Quantity sold, Item ID, Size, and Order Total. This table is generated automatically through the Paypal IPN and logs the orders to the database.



ID	Date	Name	City	State	Items	Quantities	Item IDs	Sizes	Order Total
899327589	Tue May 08 2018 20:00:23 GMT-0700 (PDT)	George Gi	San Jose	CA	Test Tee	1	34		$12.34
899327589FFF	Sun Jun 03 2018 19:26:22 GMT-0700 (PDT)	John Smith	San Jose	CA	Test tee 2,Test tee 2	1	97,97		$12.34
48G30978464697352	17:08:22 Jun 03, 2018 PDT	Jade MacDonnell	Davis	CA	Test1	1	99-1	S	$0.5
0B254396KT332405Y	14:54:37 Jun 05, 2018 PDT	Chris Ta	Antioch	CA	Test Tee 2	1	97	S	$0.5

Functionality

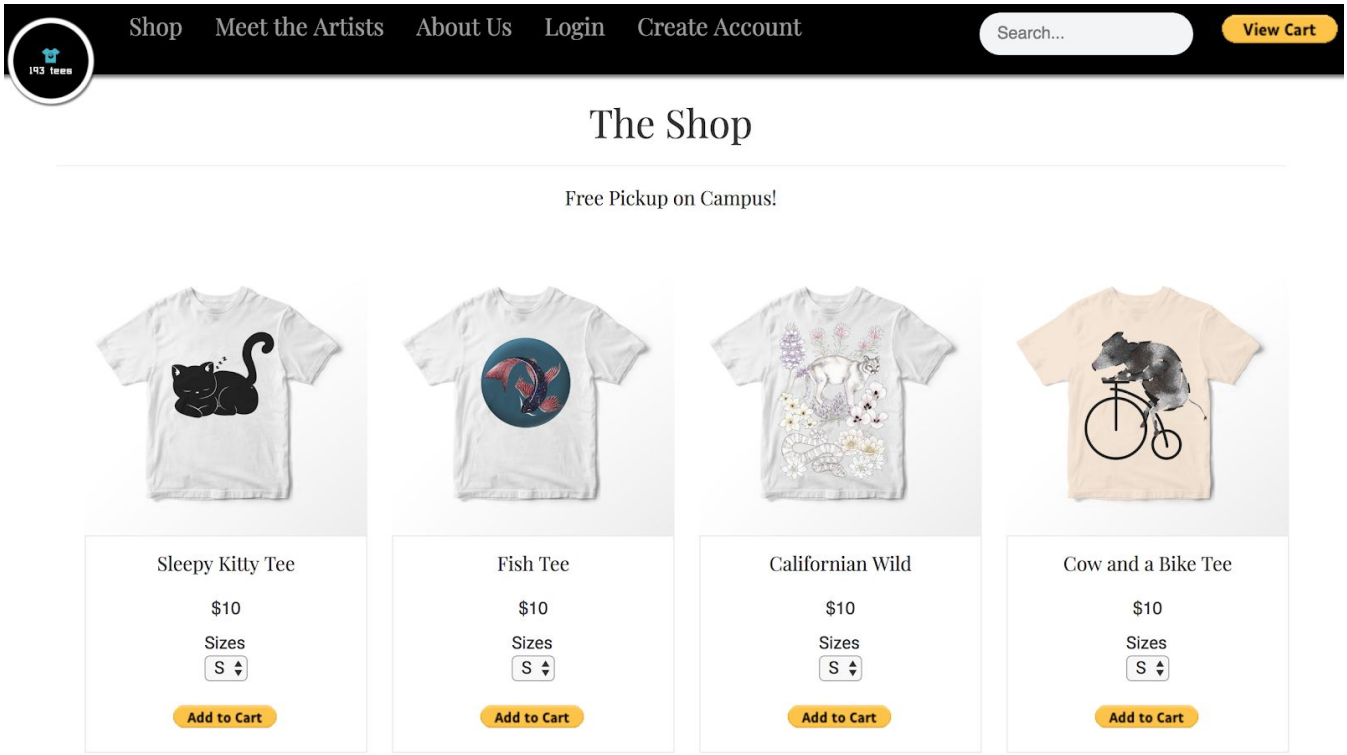
Home Page:

Shows all merchandise sorted by popularity. Each T-shirt has a unique name with the price and sizes listed right under the image. For more details about the shirt, the user can click on the shirt and that will lead to the T-shirt detail page. If the user wants the shirt, they can immediately add the desired item to cart with “Add to Cart” button, which will open a new tab to show what is currently in the cart.

Navigation Bar:

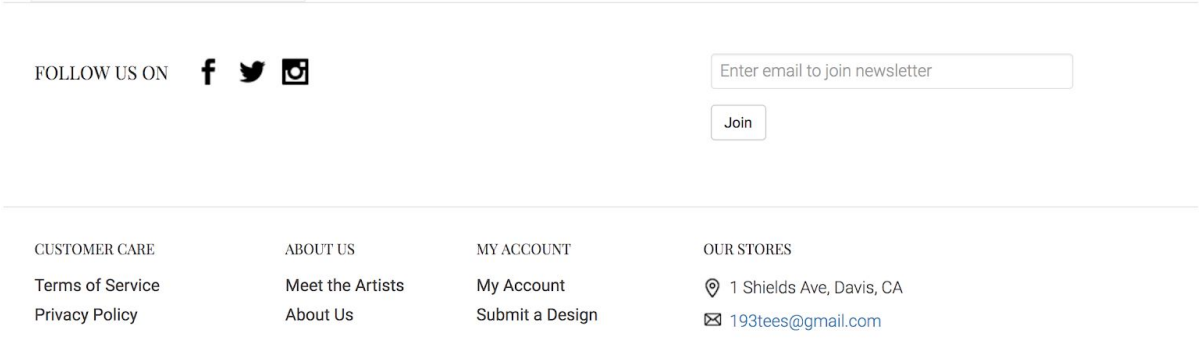
As for the navigation bar at the top of each page, this gives the user options to navigate through the website to see the artists who designed some of the shirts, know about us, login if the user already has an account, or create a new account if the user doesn’t have an account. On the far right of the navigation bar, a search box is provided for the user to easily search for a shirt if they type in a keyword in the name of a shirt. The “View Cart” button takes the user to their shopping cart. As for the circular logo icon

and the “Shop” button on the far left of the navigation bar, both will take the user to home if they’re not already there.



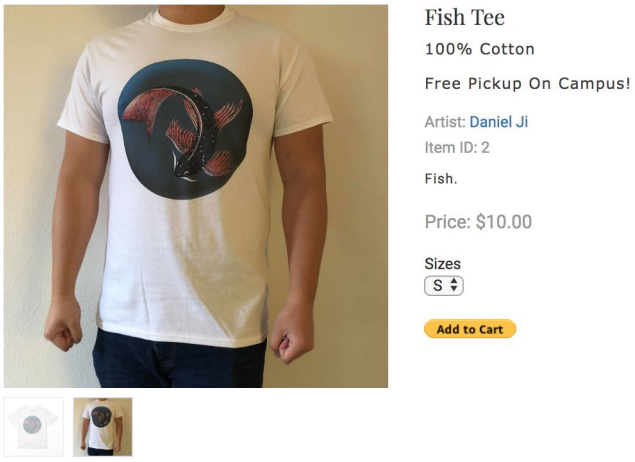
Footer:

The footer is the same for all pages on our website. The footer allows users to click on icons that would lead to our social medias: Facebook, Twitter, and Instagram. Moreover, the user can enter their email to join our newsletter. At the very bottom of the footer, there are sections and subsections with links to different pages of our website.



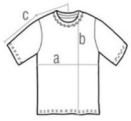
T-shirt Detail page:

Shows more info about a shirt and includes a size chart and model for reference. A user can hover the cursor over the image and a zoomed in image version will pop out to show more magnifying detail.



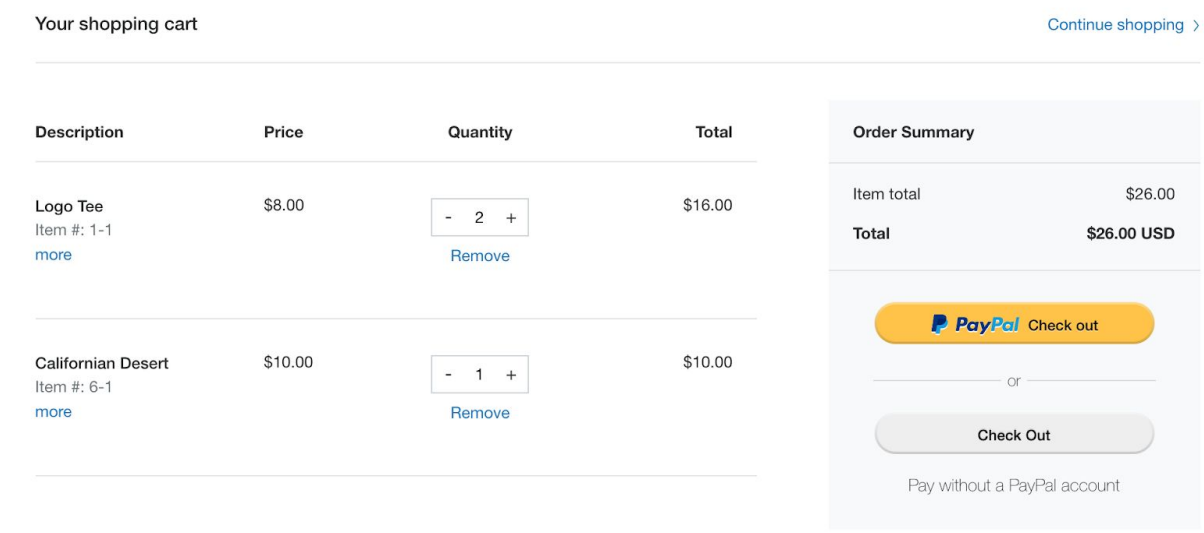
Size Chart

	Width (a)		Length (b)		CB Sleeve (c)	
	inches	cm	inches	cm	inches	cm
S	18.5	47	26.0	66	15.9	40.5
M	19.7	50	27.2	69	16.7	42.5
L	20.9	53	28.3	72	17.5	44.5
XL	22.0	56	29.1	74	18.3	46.5



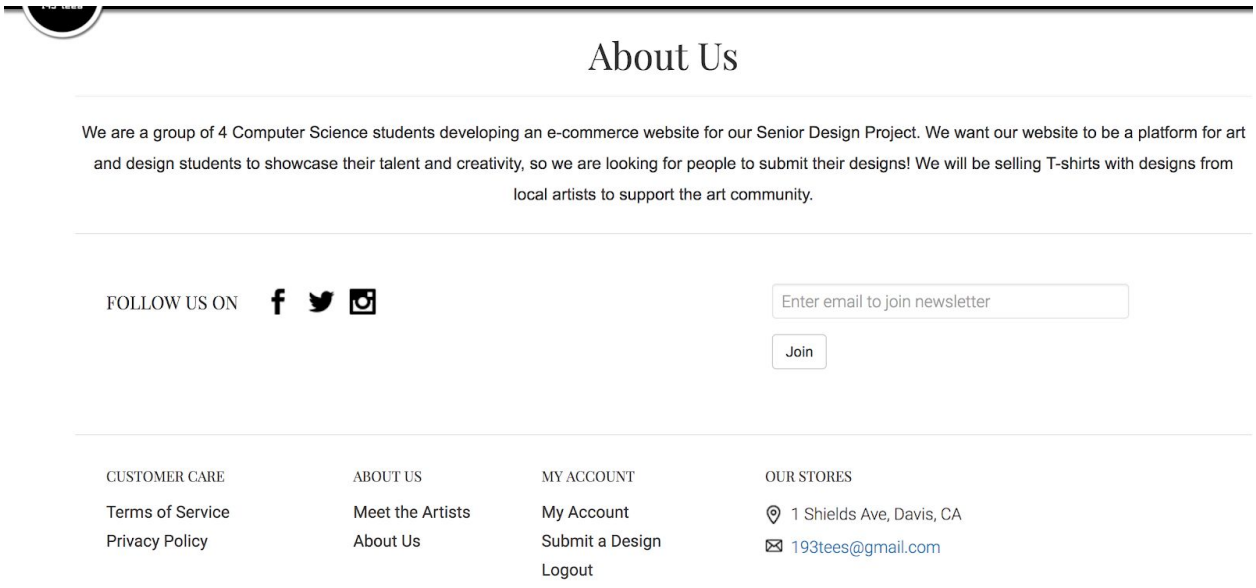
Shopping Cart:

Displays the items in user's cart, ability to add and remove items or change quantity, and a order summary with options to check out with or without Paypal.



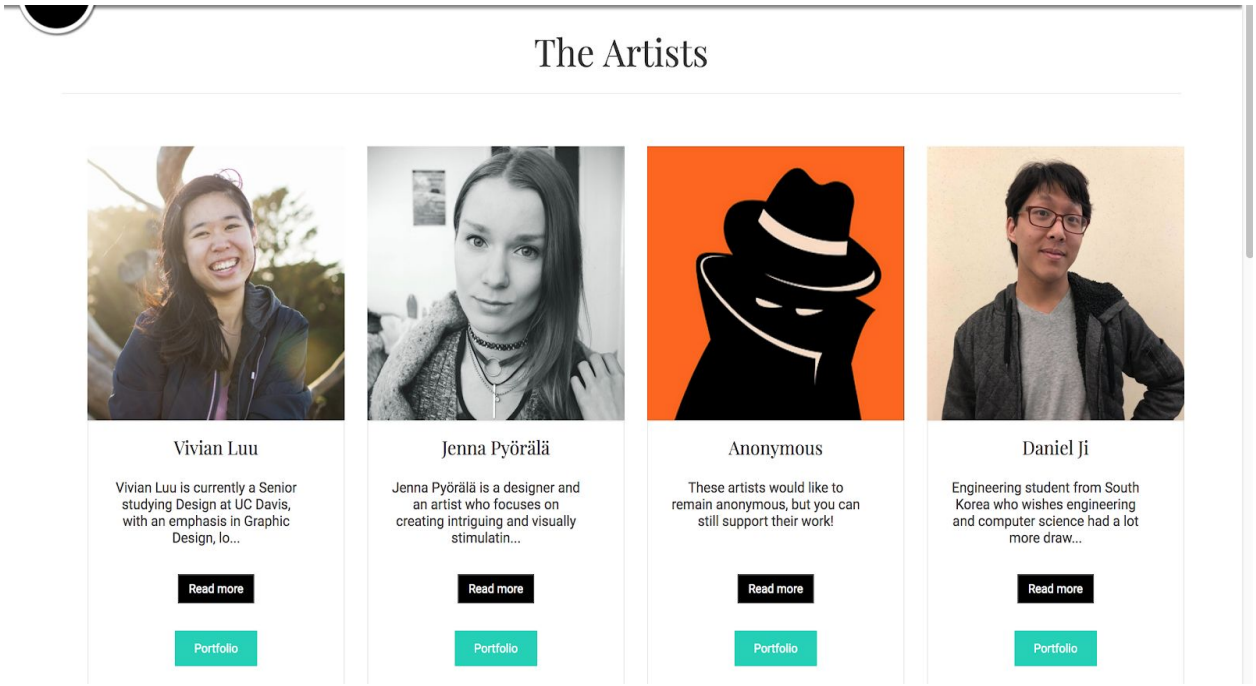
About Us:

Description of the original team and project website.



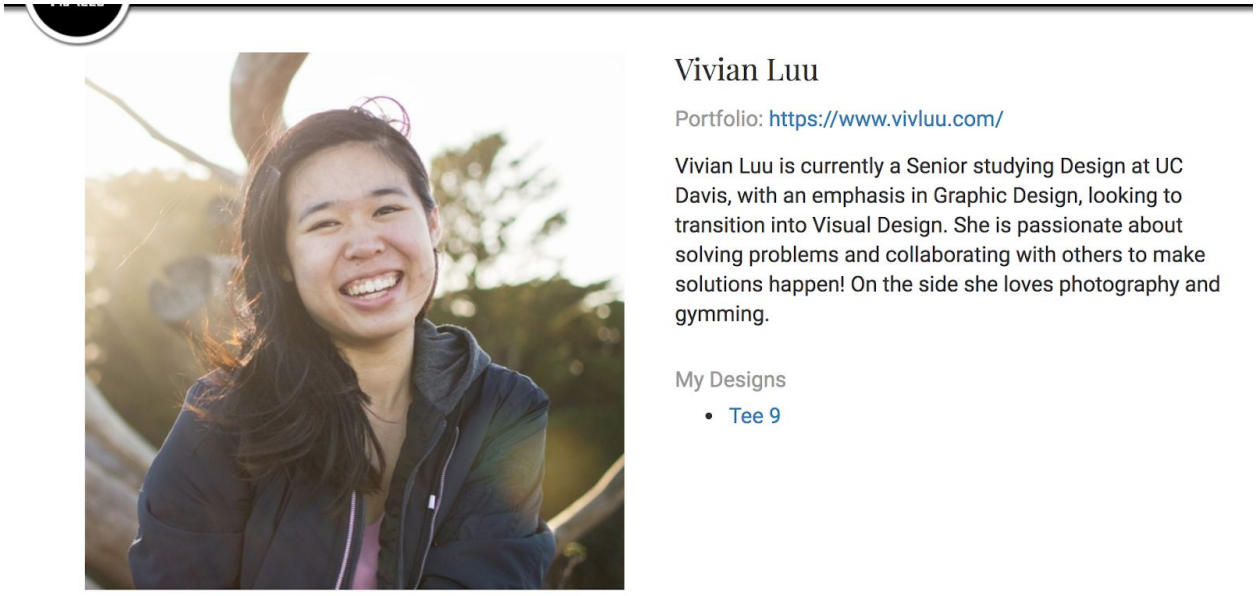
Meet the Artist page:

Displays all artists, a few words from their bio with a button to read more, and a button that links to the artist's portfolio. Some artists may want to remain anonymous which should be respected, as shown below.



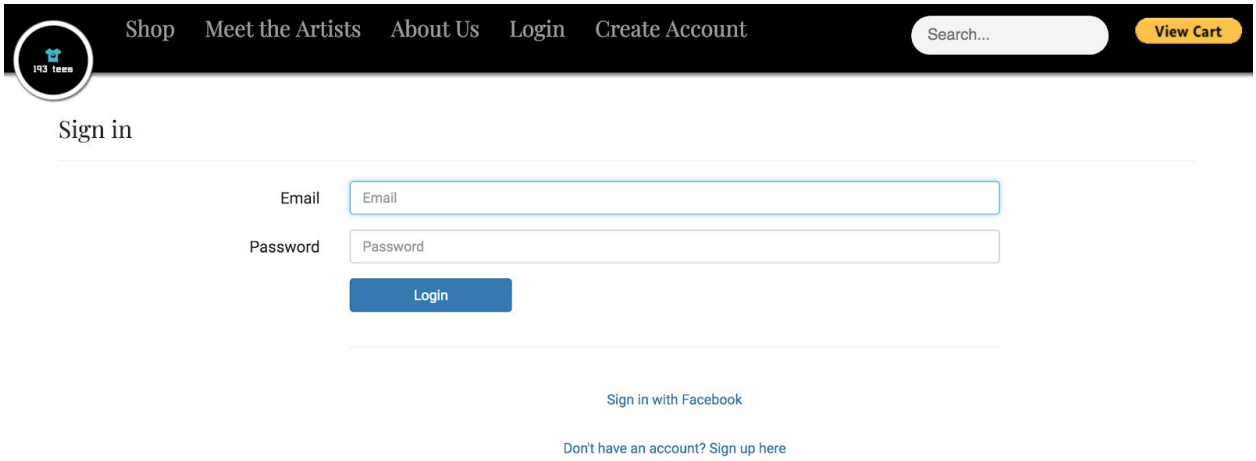
Artist Detail Page:

Displays a bio of a selected artist, a link to their portfolio, and lists which shirts they designed



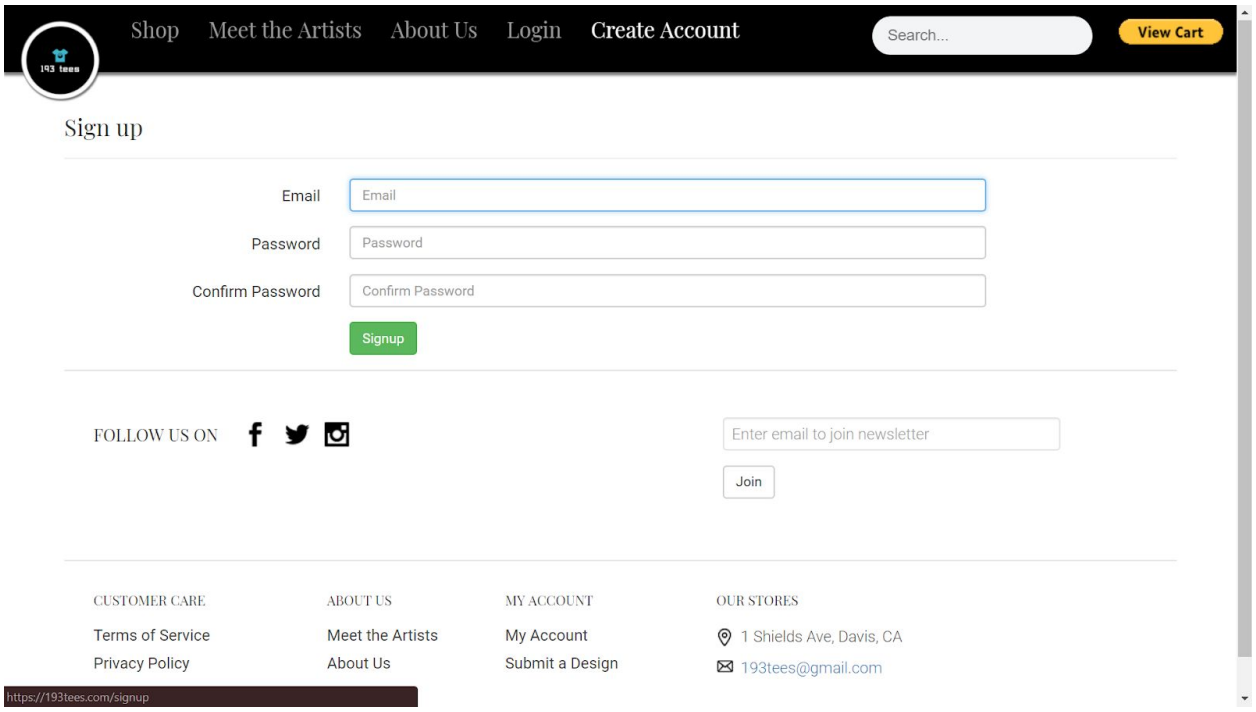
Login Page:

A user can sign into their 193Tees account manually or through Facebook. If a user doesn't have an account yet, there is an option to sign up and create a brand new account.



Sign Up Page:

A user can create a 193Tees account by entering their email, creating a password, and clicking sign up.

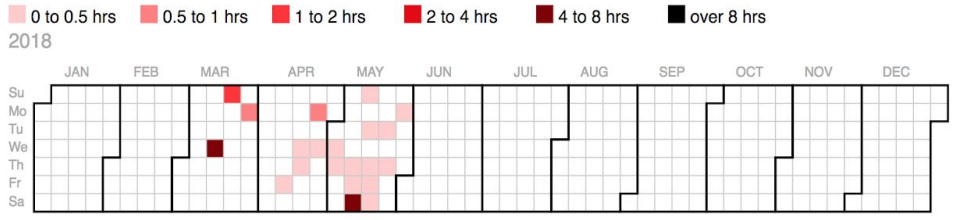


Prober Dashboard:

Keeps track of when the website was down (only available to Admin). The prober is hosted on an AWS EC2 instance. The prober sends an SMS notification to the on-call rotation team through the Twilio API when the site is down.

Prober Dashboard

Website Outages



Downtime Start (UTC)	Downtime End (UTC)	Duration (sec)	Error Code
2018-05-28 19:10:58	2018-05-28 19:31:00	1202.0597	502
2018-05-24 23:14:46	2018-05-24 23:19:46	300.3248	502
2018-05-22 03:57:21	2018-05-22 04:02:22	300.2986	502

Submit Design:

Allow user to submit their artwork/design. Clear specifications are provided regarding design requirements and users are able to upload a file from their computer to our designated Google Cloud Storage bucket.

Submit a Design

File Input

Choose File No file chosen

Submit

We are a group of 4 Computer Science students developing a platform for art and design students to showcase their talent and creativity, so we are looking for people to submit their designs! We are selling T-shirts with designs from local artists to support the art community. For each shirt sold ($10), the designer/artist will get $1 (about 25% of the earnings after tshirt fees, paypal fees, etc.). We also can promote the artist/designer on our website by sharing an "About Me" and linking a portfolio. If you are interested, you should submit a design!

Tell us where on the T-shirt you would like your design to be printed. Files should be set up to sizes not to exceed (to approximate size in inches divide pixels at each dimension by 300):

a) Full front or back, 14 x 16 @ 300dpi (4200 x 4800 pixels).

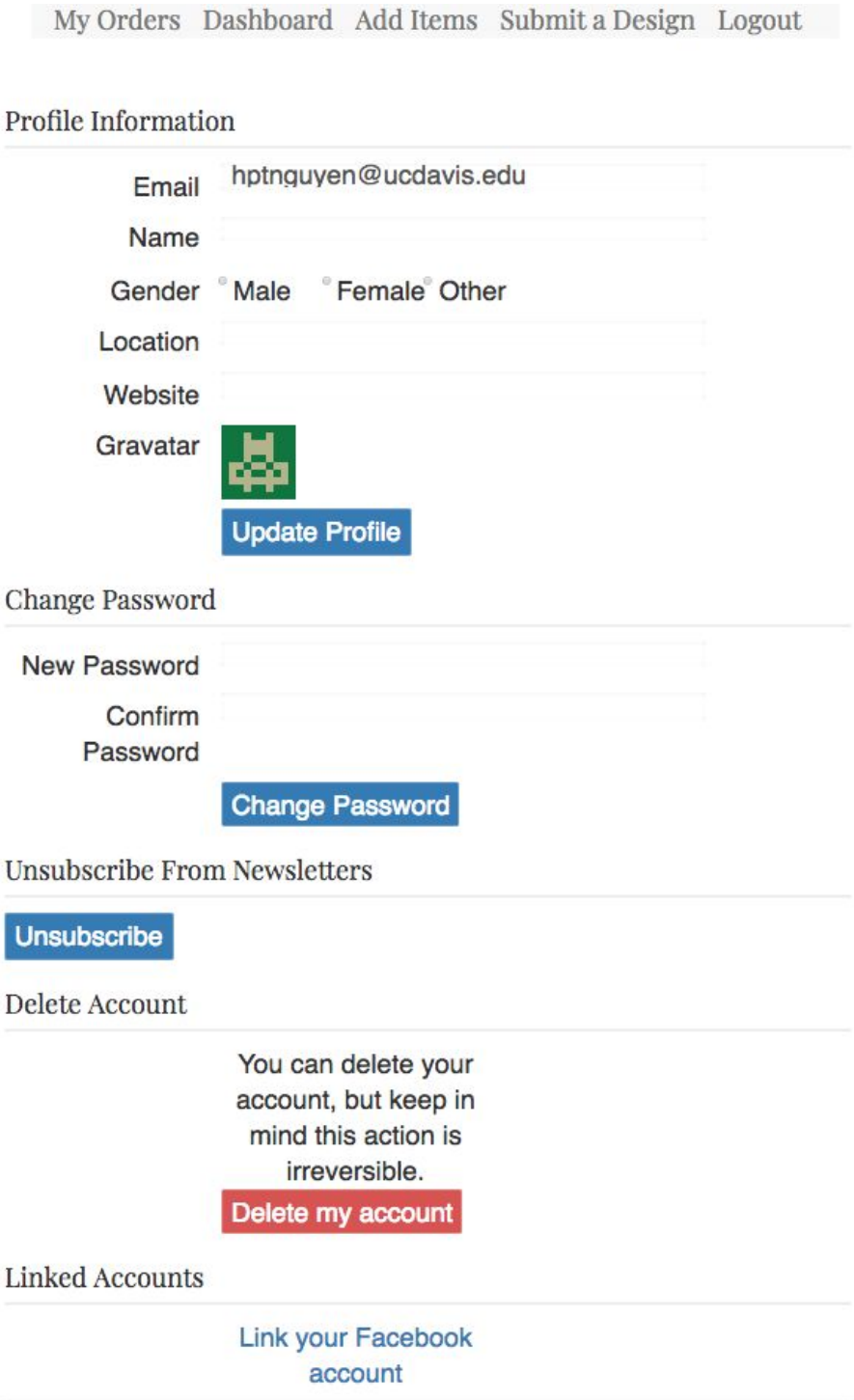
b) Left or right chest, 4 x 4 though 1 x 4 is recommended (300 x 1200)

The design can be anything of your choice (so long as no copyrights are infringed and the design is not offensive).

Email us if you have any questions!

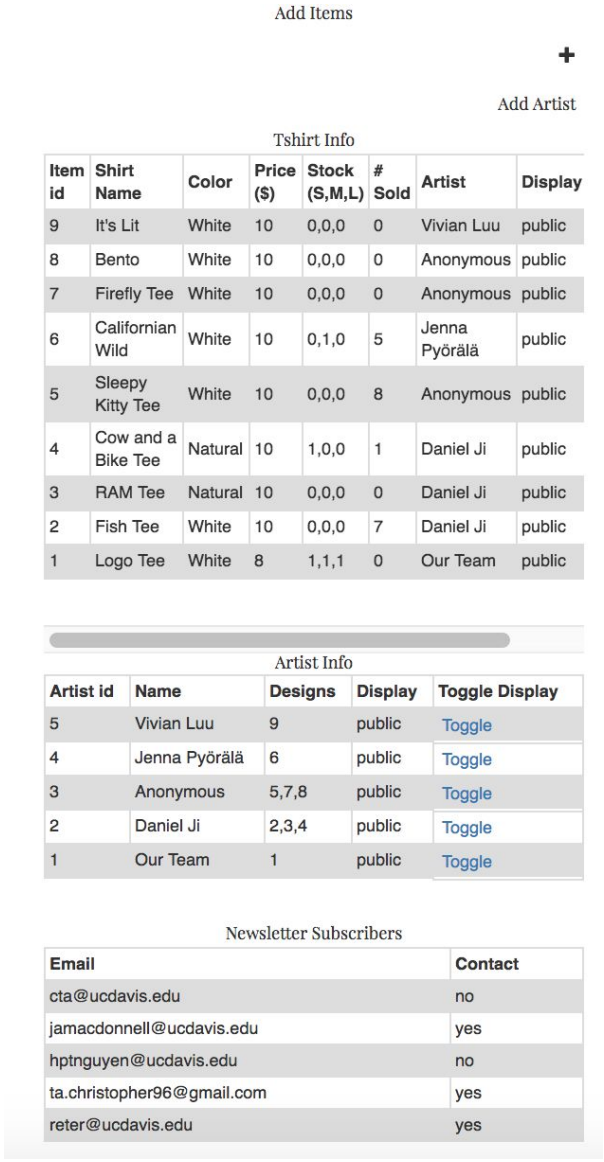
My Account:

Shows user's account and options to update the account. The "Add Items" and "Dashboard" options on top only show up if the account is an Admin account


My Orders Dashboard Add Items Submit a Design Logout
Profile Information
Email hptnguyen@ucdavis.edu
Name
Gender Male Female Other
Location
Website
Gravatar
Update Profile
Change Password
New Password
Confirm Password
Change Password
Unsubscribe From Newsletters
Unsubscribe
Delete Account
You can delete your account, but keep in mind this action is irreversible.
Delete my account
Linked Accounts
Link your Facebook account

Add Items:

This page allows Admin to add items and artists to database. The “Toggle” option allows Admin to change the display setting to either “Public” or “Hidden” for each item and artist. This page also show all current newsletter subscribers.



Add Items

+

Add Artist

Tshirt Info

Item id	Shirt Name	Color	Price ($)	Stock (S,M,L)	# Sold	Artist	Display
9	It's Lit	White	10	0,0,0	0	Vivian Luu	public
8	Bento	White	10	0,0,0	0	Anonymous	public
7	Firefly Tee	White	10	0,0,0	0	Anonymous	public
6	Californian Wild	White	10	0,1,0	5	Jenna Pyörälä	public
5	Sleepy Kitty Tee	White	10	0,0,0	8	Anonymous	public
4	Cow and a Bike Tee	Natural	10	1,0,0	1	Daniel Ji	public
3	RAM Tee	Natural	10	0,0,0	0	Daniel Ji	public
2	Fish Tee	White	10	0,0,0	7	Daniel Ji	public
1	Logo Tee	White	8	1,1,1	0	Our Team	public

Artist Info

Artist id	Name	Designs	Display	Toggle Display
5	Vivian Luu	9	public	Toggle
4	Jenna Pyörälä	6	public	Toggle
3	Anonymous	5,7,8	public	Toggle
2	Daniel Ji	2,3,4	public	Toggle
1	Our Team	1	public	Toggle

Newsletter Subscribers

Email	Contact
cta@ucdavis.edu	no
jamacdonnell@ucdavis.edu	yes
hptnguyen@ucdavis.edu	no
ta.christopher96@gmail.com	yes
reter@ucdavis.edu	yes

Troubleshooting

PROBLEM:

I am getting an “Insecure Login Blocked: You can't get an access token or log in to this app from an insecure page” error when attempting to login with Facebook.

SOLUTION:

Reload the website such that you are on https://www.193tees.com/

PROBLEM:

The server encountered an error. Please try again later.

SOLUTION:

1. The website is down due to maintenance/downtime, check again in 15 minutes.
2. For developers, this usually means that the server crashed due to an internal bug, or that the account has run out of GCP credit.
    a. For the case of the internal bug, go to the traffic splitting settings and serve traffic to an older stable version
    b. For the case of running out of GCP credit, you will need to disable billing on the current account in Navigation Menu -> Billing -> Disable billing account.
    c. Enter a new GCP code in http://bit.ly/gcp-redeem and connect this account to your existing project

---

PROBLEM:

I am getting the following error from connect to AWS: “Warning: Identity file awspw.pem not accessible: No such file or directory. ubuntu@ec2-18-219-80-177.us-east-2.compute.amazonaws.com: Permission denied (publickey).”

SOLUTION:

1. You are likely not in the 193prober directory. The awspw.pem file is located in that local repository.
2. It is recommended to make an alias similar to the following: “alias aws='cd ~/Documents/ECS_193_Content/193prober && ssh -i "awspw.pem" ubuntu@ec2-18-219-80-177.us-east-2.compute.amazonaws.com'”

PROBLEM:

I am trying to deploy a live version of the website, but GCP exits with “Permission Denied”.

SOLUTION:

1. Although the error does not indicate it, this usually means that your account has run out of GCP credit. Follow the steps above to change billing accounts.

PROBLEM:

I try running the server and I get an error that looks like: “npm: X module not found”.

SOLUTION:

1. Run “npm install” to download node module dependencies.

PROBLEM:

I try running the server and I get an error: “TypeError: OAuth2Strategy requires a clientID option”

SOLUTION:

1. This is because either the .env or keys.json file is missing from your local directory. Request these files from a team member.

PROBLEM:

I can’t connect to localhost, it says the server is not found.

SOLUTION:

1. Ensure that you’ve changed bin/www localdev variable to 1. This makes an https secure connection with localhost on your device.

FAQ

Q: Where does the name 193tees come from?
A: 193tees is inspired by the name of the Computer Science & Engineering senior design class, ECS 193

Q: How does your website help artists?
A: This website provides a public platform for artists to showcase their art by putting their designs on t-shirts. There is a page dedicated to individual artist's profile, bio, and links to their portfolio and social media. Moreover, the artists are given a cut of our profit for every shirt sold with their design.

Q: I am an artist and I would like to submit a design. Where do I go?
A: At the footer of all the pages on our website, there is a section called "My Account" which contains a subsection call "Submit Design". You can click on that and it will link to a page prompting you to create an account in order to submit a design.

Q: How do I purchase an item?
A: Our home page has a listing of all our shirts, users can click on the "Add to Cart" button and a new tab will open to show the cart. User can then make purchase "Paypal Check out" or use regular "Check out" to buy item.

Q: Where is the free pickup option?
A: Once user finished add items to cart and select check out, user will get redirect to a check out page to enter in their address and payment information. "Select Payment Method" during checkout page will produce a drop down menu with the options of $0 for pick and $5 for shipping.

Contact Info

Rose-Marie Eter
Email: reter@ucdavis.edu
Jade MacDonnell
Email: jamacdonnell@ucdavis.edu
Phuong Nguyen
Email: hptnguyen@ucdavis.edu
Christopher Ta
Email: cta@ucdavis.edu

Glossary

See the glossary below in the appendix.

Appendix

Requirements Document

Appendix: Requirements Document

E-Commerce Website
193Tees

Team Members: Rose-Marie Eter, Jade MacDonnell, Phuong Nguyen, Christopher Ta

Index Table

Revision History

6/2/18 - User Guide/Delivery version
5/28/18 - Beta version
4/6/18 - Post alpha version
2/10/18 - Initial version

Introduction

We have developed an ecommerce website, 193Tees, for selling T-Shirts. Our online store aims to become a platform through which students can submit their own T-shirt designs and ultimately express their creativity. Our website is an appealing and efficient medium that allows students to share their artistic side and potentially gain recognition from the public, exhibiting their aesthetic art and designs promoting a cause. We initially sold a fixed number of prototyped designs to ensure that our website is working properly.

We are using Google App Engine to host our server as our client is from Google and is willing to provide us some guidance with the process. We are using MongoDB to create the

database to keep track of our merchandise (i.e. design, quantity, etc.). As for the server side, we are working with Express and NodeJS. To make our front end user-friendly and simple, we are using AngularJS.

The website also employs site reliability engineering techniques (SRE). Specifically from this domain, we have developed mechanisms to monitor uptime gradual rollouts of new versions. Uptime monitoring is implemented with a software module known as a prober, which periodically pings the website to ensure availability. A method for gradual rollouts of new versions is being investigated for implementation.

Glossary Terms

MEAN Stack - A full Javascript technology stack consisting of MongoDB, Express, Angular, and Node.js. The purpose of using this stack is using the Javascript language from the front to back end.

MongoDB - NoSQL database for managing and maintaining records pertaining to the platform.

Express.js - A Node.js module for developing server-side code.

AngularJS - A front end Javascript framework developed by Google to build responsive single view web applications. High learning curve.

Node.js - Javascript runtime environment to run Javascript apps in real time.

GitHub - Third party version control and project repository hosting.

Cloud System - An outsourcing of computer resources (servers, database storage). Google Cloud Platform (GCP) and Amazon Web Services (AWS) are the two primary cloud hosting services.

Graphical User Interface (GUI) - A type of user interface that allows users to interact with electronic devices through graphical icons and visual indicators such as secondary notation, instead of text-based user interfaces, typed command labels or text navigation.

Slack - Communication platform for internal use between team members. Has Google Drive and Github integrations to optimize workflow.

Technology Survey

After comparing the pros and cons of 2 potential technology stacks (LAMP and MEAN) we decided to build our website using MEAN services. The Javascript technology stack can be developed, scaled, and prototyped quickly using most existing technologies. With a flexible MongoDB database, we are able to store and query new types of user data seamlessly as we develop our user stories. Although Google Cloud is more costly than AWS, we would be receiving sponsorship and additional technical support by choosing to use Google cloud. However, due to AWS's free quota limit per month, we still employ this technology to host our 24/7 dashboard to notify and track website uptime. Lastly, to follow a proper software engineering workflow, all project progress is tracked with Github as a version control system.

LAMP:
Apache

+ It can be installed easily
+ The changes made are recorded immediately, even without restarting the server
- Requires a strict updating policy that needs to be done regularly without fail
- Recognizing and disabling unwanted services and modules. Leaving them on could cause serious threats
- Apache is a process based server

MySQL db

+ Multi-row transactions such as an accounting system, would be better suited for a relational database
- Risk of SQL injection attack
- Slow with unstructured data that grows and shrinks

PHP

+ PHP is a highly portable application development solution.
- Performance of Zend PHP is less than half of Node.js
- PHP is not a great fit for the implementation of MVC (Model-View-Controller) approach that prescribes clear Separation of Concerns (SOC) between data, behavior and representation.

MEAN:
MongoDB

+ No-relational database structure -- table entries may contain differing attributes
+ Returns JSON objects -- easy to parse
+ MongoDB Cloud Manager and MongoDB Ops Manager provide continuous backup with point in time recovery, and users can enable alerts in Cloud Manager to detect if their deployment is internet exposed
- Mongo does not support multi-document transactions
- Only document level atomic operations provided

- Distributed transactions are hard to scale

ExpressJS
- + Very high level, little code is needed to create running server and deploy the application

AngularJS
- + Robust front end framework for single page applications
- - Steep learning curve

NodeJS
- + Easy to use dependency and module management
- + Uses JavaScript, which is easy to learn

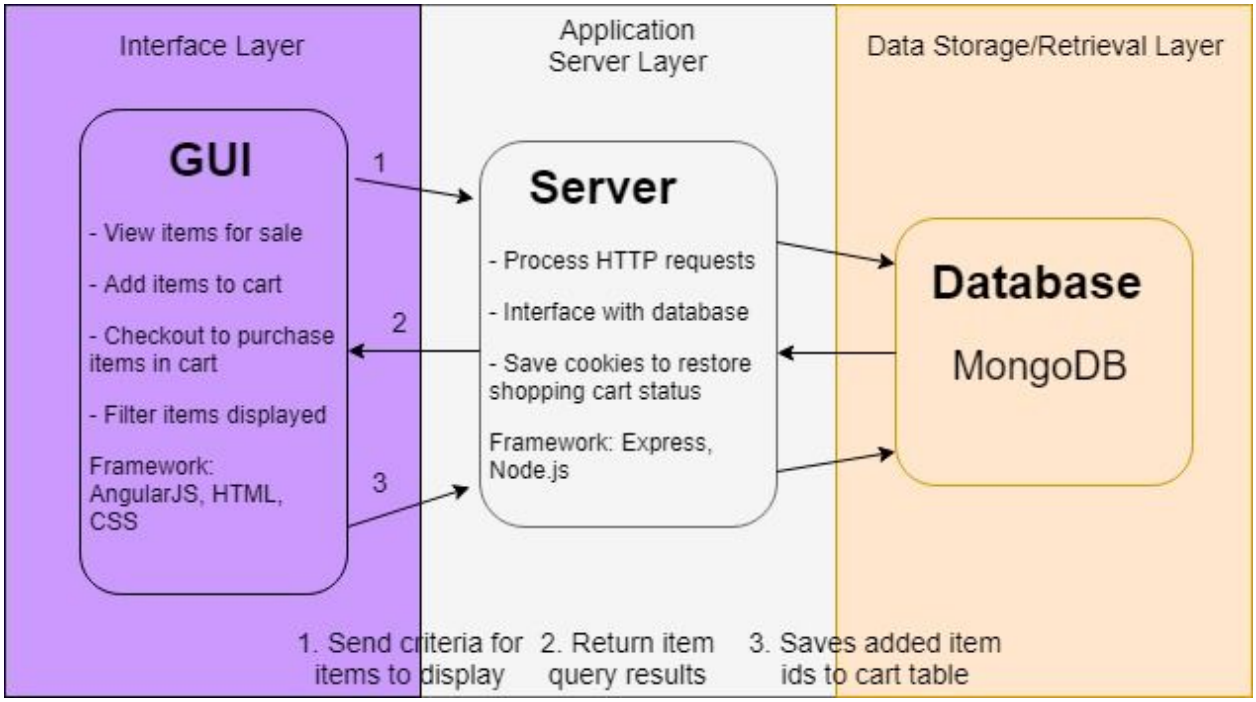
Google Cloud Platform
- + Offers a large suite of features which may be helpful for our project (Analytics, AdWords, Tag Manager, version deployment history)
- - Expensive: ~$100.00 a month to run with little traffic

Amazon Web Services
- + Easy to setup and develop remote components quickly
- + Free quota available for each month -- low costs to run external applications hosted here

Github
- + Popular and widely used version control system
- - Steep learning curve
- - Mutable history

System Architecture Overview

Software Architecture -



Interface Layer

- This is the part of the application with which the user interacts. It graphically displays the information stored in our database (items for sale, artist information) and provides features that allow the user to online shop. These features include the ability to filter items displayed, add items for sale to a shopping cart, and to checkout and purchase items in cart.

Application Server Layer

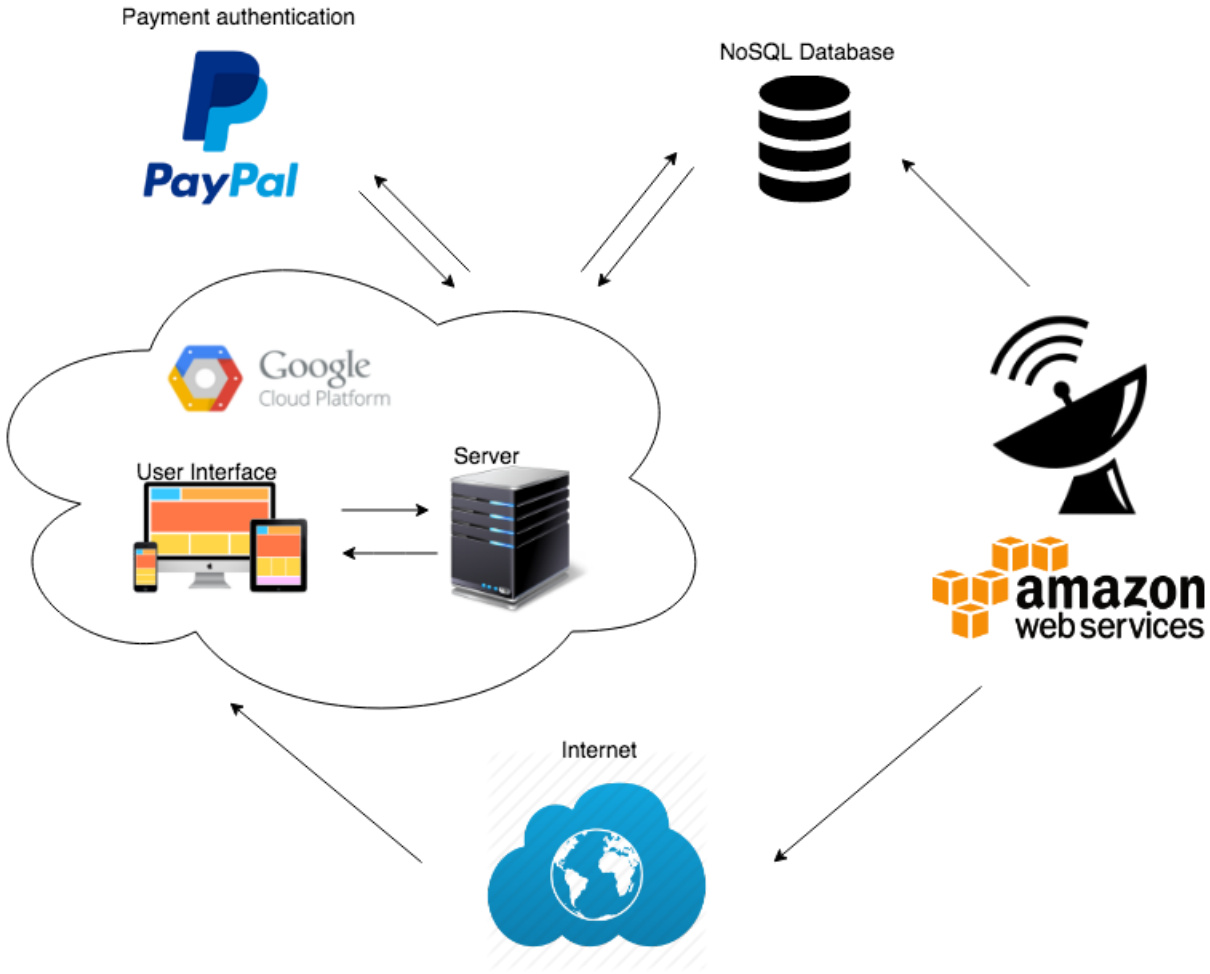
- This layer of the application is a running web server that is responsible for processing HTTP requests and serving users the correct webpage based on their url route. To build the requested web page, the server also connects to the database and retrieves the data necessary for that web page. Furthermore, the server updates the database when the user performs certain actions through the interface, such as updating the item inventory.

Data Storage/Retrieval layer

- This layer stores the data for the website. Data such as item information, artist information, website downtime. This is implemented through an mLab database, where

individual tables are created for each of these collections. Queries are done through NodeJS functions.

System Diagram



Use Cases/User Stories

Home page

- As a user I can view a gallery of items so that I can add some to my shopping cart Acceptance Test: Ensure the items get stored per unique user's shopping cart
- As a user I can create a unique login ID and password. Acceptance Test: Create user login account and store user's data in our database, attempt to login with the account
- As a user I can log into the website using an external 3rd party account (Facebook, Google, etc). Acceptance test: Allow the user to login through 3rd party API and store their unique information
- As a user I can search for a product by name/category. Acceptance Test: Create a search bar to efficiently query user input and display specified items

- As a user I can click to view details for each product. Acceptance Test: Create an enlarged details page tailored to each item being sold
- As a user I can sign up for a mailing list which will send me promotional emails Acceptance test: Sign up for the mailing list and periodically receive these emails from an official domain

Prober Dashboard page

- As a user, if I am an admin, I am able to view the logs and downtime of the website according to the prober. Acceptance test: Login as an admin account and be able to view the dashboard
- As a user, if I am an admin, I am able to change the call rotation and schedule of who gets contacted when the website goes down. Acceptance test: Change the call rotation and purposefully crash the website. Check that the contacted person is who was specified

Add items page

- As a user, if I am an admin, I am able to add a new item/artist to the database through a gui. Acceptance test: Add an item and see if it shows up in the live store page
- As a user, if I am an admin, I am able to temporarily disable items in the store by toggling them. Acceptance test: Toggle a button and see if it becomes hidden in the live store page
- As a user, if I am an admin, I am able to edit items in the database through the GUI. Acceptance test: Edit an item and see if the changes propagated to the live store page

View Cart page

- As a user, I can modify the quantity of items in my shopping cart. Acceptance test: Attempt to change the quantity of the shopping cart, check database for matching quantity
- As a user, I can go return to the item I was previously on by clicking “continue shopping”. Acceptance test: Click on the continue shopping button and ensure the user gets back to the same page
- As a user, I can close my browser and have my shopping cart contents saved. Acceptance Test: Close the browser and ensure the user’s cart contents are the same
- As a user, I am able to login and view my order history. Acceptance test: Purchase an item and ensure the item is in my order history

Checkout page

- As a user, I can checkout items in my cart through Paypal. Acceptance Test: Get the user’s payment and shipping information from the merchant Paypal upon transaction completion
- As a user, I can add and remove items before I checkout. Acceptance Test: Implement the add and remove button in the checkout page that can add and remove items

- As a user, I can assume that once I checkout items from my cart, an automated email will be sent immediately to the product supplier for my order to be processed. Acceptance Test: Set up an automated email system that sends specific order/item data to the product supplier triggered upon every order that is checked out

Submissions Page

- As a user, I can submit any art file of the appropriate format through this clean and efficient interface. Acceptance Test: Create a web page that presents a secure file upload feature as well as some formatting guidelines for users to be aware of

Meet Artist Page:

- As a user, I can see all the artists featured by 193Tees. This page also give me a button to read more about the bio and a button to view their portfolio. Acceptance Test: Create a web page featuring pictures and biography previews of any artists who have contributed their t-shirt designs for sale

Artists Detail Page:

- As a user, I can read more about the artists and see the list of T-shirts they designed. Acceptance test: Click on an a particular artist and be able to click on any link to their T-shirt

About Us:

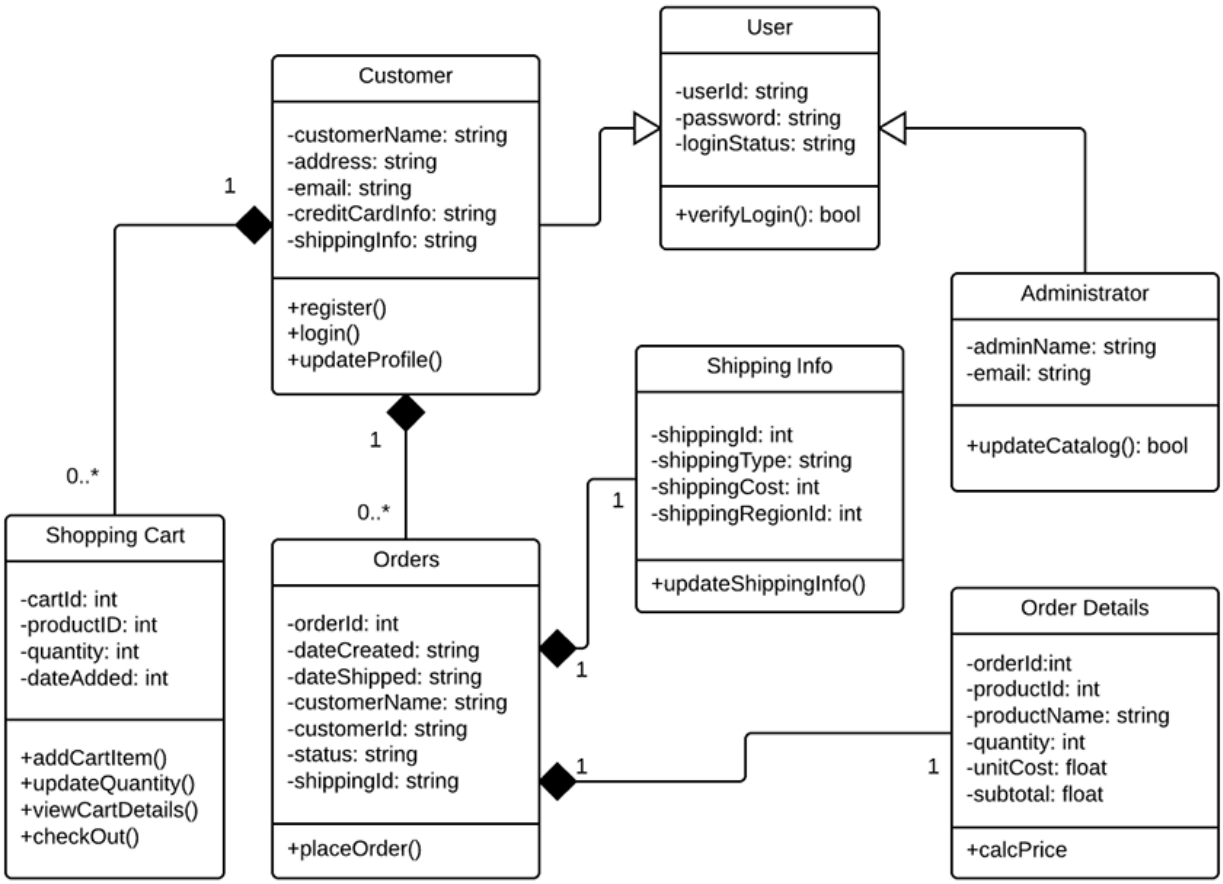
- As a user, I can read about 193Tees and what their website is about. Acceptance test: Able to view team's bio

Prototyping Code

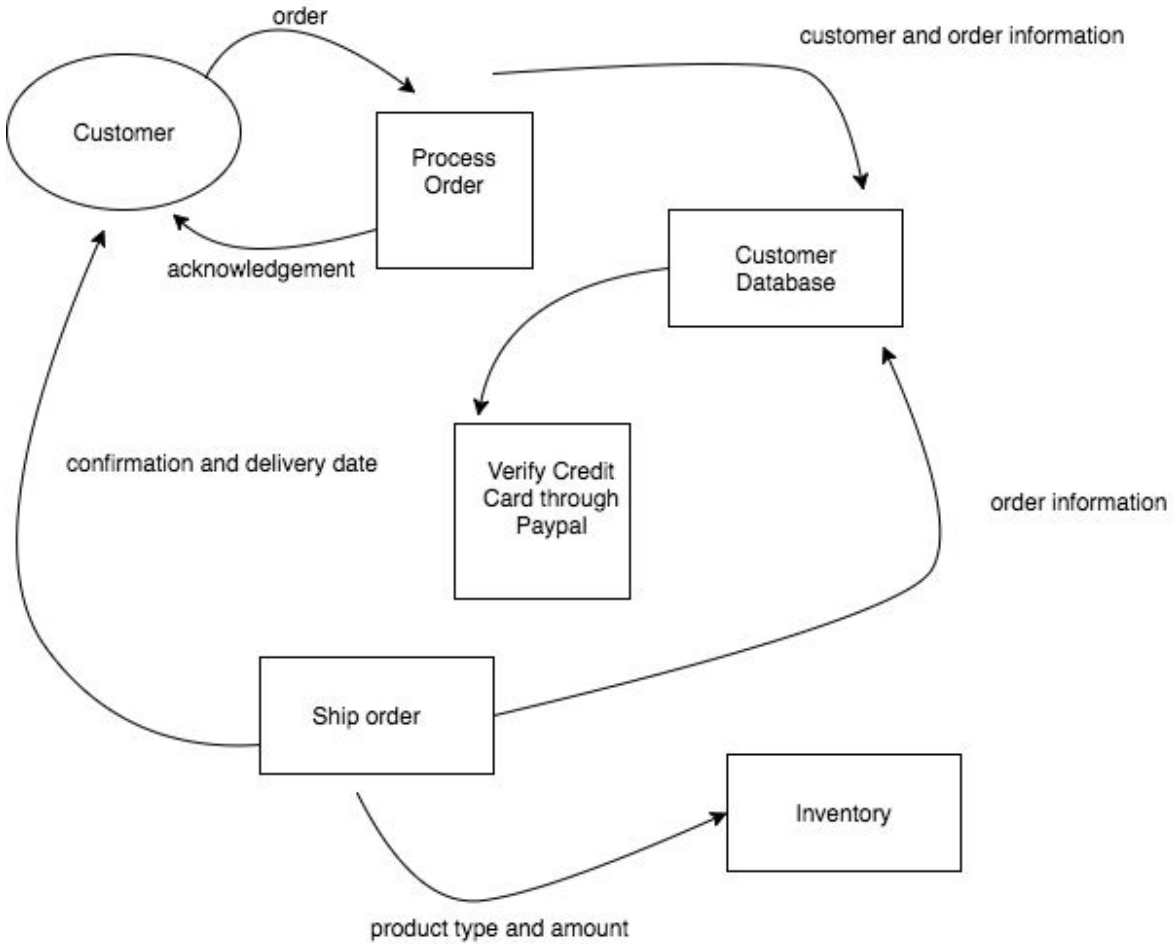
GitHub Repo - https://github.com/CTaCDE/bitcommerce

System Models

UML - https://d2slcw3kip6qmk.cloudfront.net/marketing/pages/chart/class-diagram-for-online-shopping-system-UML/UML_class_diagram_shopping-800x621.PNG



Data Flow Diagram



Appendix/Technologies Employed

Software:

MongoDB (mlab stores all our data)

ExpressJS (Node framework for routes/controllers)

AngularJS (Front-end framework for single view applications)

NodeJS (Javascript runtime environment and package management)

Google Cloud Platform (App Engine, Google Analytics)

Amazon Web Services (EC2 runs the prober 24/7 to notify us when the website is down)

Hardware:

T-shirts (Printed and produced by UC Davis Repro Graphics)