

Group 2

CyberSuccess

Project Milestone 5

Application Title: DonateUp

Test Plans:

1. Test Websites/Product URLs DonateUp accepts.

Donate Up's website and URL-page scan functionality is dependent upon an immense amount of URL and item ID HTML information, as the python code uses URLs to request pages HTML then searches for an item with an ID. Through testing and identifying failures the code can only improve as more product HTML IDs are added to identify item prices. The data collected during all tests should account for failure types: URL or ID. This information will be used to update the accepted URL and IDs on the price scrape python file. Test cases should be enacted in the python app.py file and have the URL variable be manipulated by posts within the test cases. Ideally, all test cases should account for the majority of Ebay, Amazon, Nike, and Etsy item ID's in which the python API requests. Failed test cases should be specific and output the exact problem from invalid URLs, request failures, and unable to identify ID(NoneType errors). During integration testing our group will see if the URLs pass on the actual website, once passed our group will begin to use URLs of choice. During User Acceptance testing users will use URLs of their choice from the above website, and take notes on the functionality, failure types, and useability.

2. Chrome extension:

- a. For the chrome extension there will be several tests. The first tests will be to make sure that the chrome extension recognizes and pops up at all of the decided cart pages that we want it to work with: nike, ebay, adidas.
- b. Another test case for the extension will be that when the donation button is hit, that it will post the correct user and donation data to the db.

3. User login/signup

User signup will be a submission of: email, account name, password, and password confirmation.

Test cases will check that email is valid, the email does not already exist in the database, the password meets password requirements, the confirmation password matches the password field.

To test these, data will be submitted to positively and negatively confirm that the above functions.

User login will be a submission of: email and password.

Test cases will check that the email is valid, that it exists in the database, that the password matches the password in the database for the account.

































Similar to signup, positive and negative tests will be created for these.

Eliza Mimi To	Made the About Us and My profile page. Working on figuring out Heroku by seeing which files need to be through that. Latest commit of the pages I created: https://github.com/CSCI-3308-CU-Boulder/3308SP21_021_2/tree/main/Code/AboutUsMyProfile
Jesse Andringa	Working on an extension to pop up when on a cart page. Also logs all URLs the person has open. Test cases for an extension. Writing the extension form to post with flask.
Timothy Ashour	Created database and started testing database connections with the app. Got test database reading to work. Have not gotten the database to work in production.





















	https://github.com/CSCI-3308-CU-Boulder/3308SP21_021_2/commit/9783567c6741ea6dad15327f6862c38a72e1e744
Nathan Jager	<p>Working on figuring out Heroku for replacing flask for better integration and web deployment since the group has decided that transferring page info would be better if not working with flask and instead having a set webserver to run on.</p> <p>https://github.com/CSCI-3308-CU-Boulder/3308SP21_021_2/tree/main/Code/heroku</p> <p>Waiting on cu-csci-2400-fall-2020 to give heroku permissions</p>
Jeremy Schlagel	<p>Finished cleaning code, added an animated modal, added date to footer, attached all pages.</p> <p>https://github.com/CSCI-3308-CU-Boulder/3308SP21_021_2/tree/main/Code/donate_up_website_together/Donate%20up</p>
Benjamin Weiler	<p>Implemented flask to host the website and created a URL input function. Test Websites/Product URLs DonateUp test.</p> <p>https://github.com/CSCI-3308-CU-Boulder/3308SP21_021_2/tree/main/Code/donate_website</p>

Project management board screenshots









































Research

		Subite...	Person	Status	Date	
	Making Chrome extensions			 +2	Working on it	Mar 8
	API			 +2	Done	Mar 8
	VPS				Done	Mar 8
	DMN				Working on it	Mar 8
	TLS				Done	Mar 8
	Bank Accessibility				Working on it	Mar 8
	Design Integration			 +3	Working on it	Mar 13
	UI and Backend integrations			 +2		Mar 13
+ Add						

Design

		Subite...	Person	Status	Date	
	Website UI			 +2	Working on it	Mar 8
	Theme				Done	Feb 21
	Account table			 +2		Mar 8
	charity table			 +2		Mar 8
	history table			 +2	Working on it	Mar 8
+ Add						

Development

		Subite...	Person	Status	Date	
	Website Framework			 +4	Done	Mar 30
	Website autodetect purchase			 +2	Working on it	Mar 30
	Basic Website Functionality				Done	Mar 15
	Bank communication				Done	Mar 18
	Autodetect purchase API				Working on it	Mar 20
	Functionality Website			 +2	Done	Mar 31
	Chrome Extension Framework			 +3	Working on it	Apr 5
	Functionality-Chrome				Working on it	Apr 8
	Integration				Working on it	Apr 10
	Testing			 +2	Working on it	Apr 8
+ Add						