CyberSuccess
Project Milestone 4

Revised List of Features:

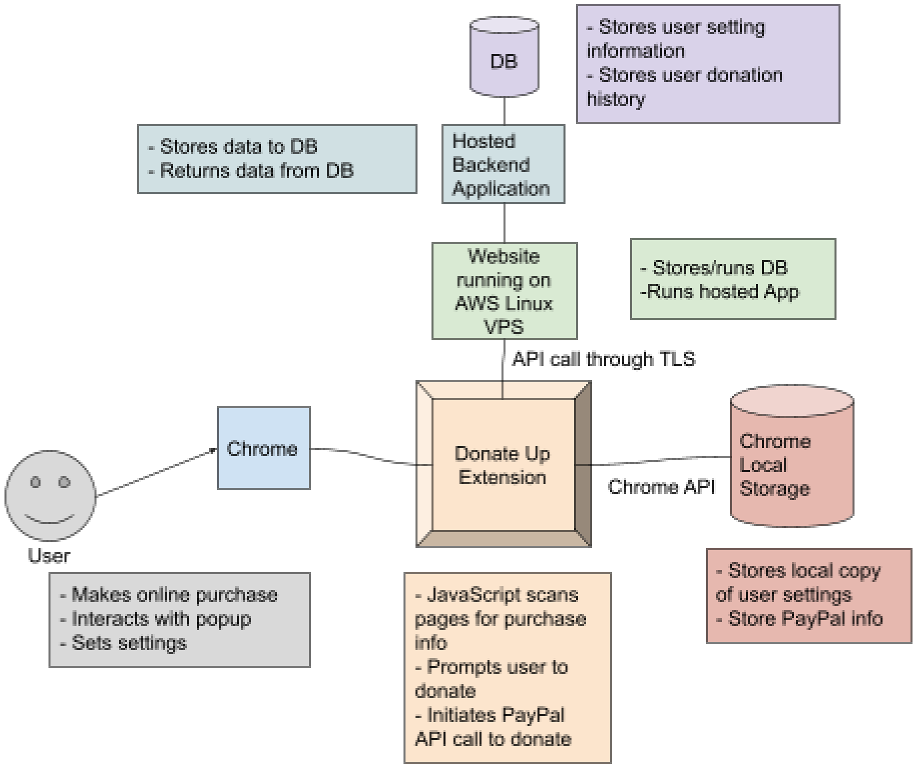
- Home HTML page -> http://donateup.life/
- Start of chrome extension page with HTML and js.
- Sign up/Login page ..Login/password checking files
- Quick Sign Up Link
- Pay Pal link donation… Enter any value to donate!

Back End:

- Auto-detect cost of purchase for PayPal link
- Bought Webserver

Architecture Diagram:

- This deliverable is a picture or diagram that shows each architectural component of your application.



Front End design:

- **This deliverable is a series of diagrams that show the basic design of your application's front end.**

- Home HTML page -> http://donateup.life/

Web Service Design:

- Nginx webserver redirecting to python flask web app
- POST requests to flask application as API for chrome extension

Database design:

- PostgreSQL database used for storing user authentication information as well as user donation history.
- Scheme for tables:

User Table

user_id (Key)	email	password	name
int(16)	text	char(32) [md5]	text

Charity Table

*charity_id (Key)	charity_name	charity_link	charity_description
int(16)	text	text	text

Donation History Table

donation_id (Key)	user_id	charity_id	amount	Datetime
int(16)	int(16)	int(16)	float(32)	timestamp

Eliza Mimi To	Project milestone document. Starting HTML page for the log-in route after signing in
Jesse Andringa	Research on chrome extension developer and publishing. Project milestone.
Timothy Ashour	Worked with Flask and researched integrating with postgresql, Project milestone document

Nathan Jager	Project milestone document
Jeremy Schlagel	Clean up the html code. Make separate .css file. Make separate .js file.
Benjamin Weiler	Worked with Flask and researched implementation with python, HTML, and Java. Project milestone document

CHALLENGES:

- **Identify at least 3 challenges and/or risks to your project at this point.**
    - Implementing python with flask - automatic price detection
    - Have sessions continue going in the background to help with ease for only one login.
    - Have passwords be secure
- **What is your backup/risk mitigation plan in case you are unable to resolve the challenge(s)?**
    - Only use automatic detection for one website
    - Have a login right before the donation
    - Have the passwords be in plain text format (since this isn’t going into production yet) and/or make the database only accessible to certain users