SUMMARY DOCUMENT

Surya Lakshmi Narayanan lakshmis@purdue.edu

Introduction to Cara: A Simple and User-Friendly E-commerce Platform:

Cara is an e-commerce platform aimed at providing users with a secure and user-friendly online shopping experience. The current work for the final project aims at showcasing the abilities of a complete online shopping application. The backend of the website is set up using Django where 3 models have been created, one for the Products, one for the User Profiles and the other for the reviews being provided by the users. The front end of the project renders the images, descriptions, and titles of the item amongst other features while also managing the users from the SQLite database.

The intended audience of the website is any user who would like to have access to a user-friendly shopping experience.

Users will be able to navigate seamlessly between the webpages and also utilize the functionalities that have been built in with utmost ease.

This phase of the project builds on the previous milestone 2, making changes and enhancements as needed. Any new webpages seen in this phase should be considered as enhancements to include functionalities which were developed to provide the users a better experience with the website.

Application Setup:

- Download the folder from GitHub.

Virtual environment setup :

- While developing the application, virtual environment wrapper was used. Virtualenv and Virtualenv Wrapper do not directly provide functionality for managing environment variables through .env files. Instead, they focus on creating isolated Python environments for projects.
- Hence, developers are required to create a virtual environment of their choice, either in the similar way provided below or differently.

- Commands used to setup Virtual ENV while development which can be used to install the application :
  - pip install virtualenv virtualenvwrapper
  - mkvirtualen <environment_name>
  - workon <environment_name>
- The next step is to setup the system and for this : ○ pip install -r requirements.txt
- Once the requirements are installed, the Database is required to be setup, the command to do this is
  - ○ python manage.py makemigrations
  - ○ python manage.py migrate
- Now that we have installed the dependencies and have the environment setup. We will be running the server using the command:
  - ○ python manage.py runserver

User Account:

A default user account has been created with the following credentials :

Username: CNIT
Password: Project123

Apart from this, there is also a comprehensive registration page which is present for the users to create new user profiles.

URLS:

On your browser go to : http://127.0.0.1:8000

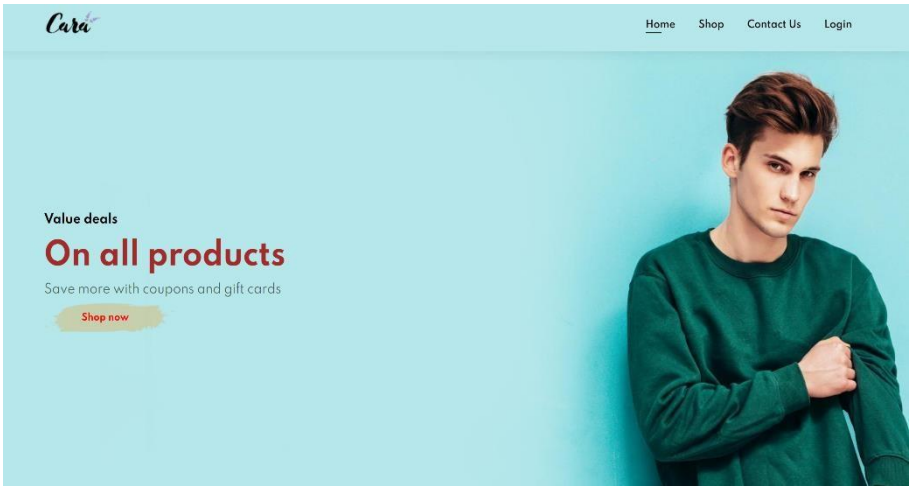
You will be able to access Home, Shop, Contact Us and the Login tabs when you are not logged in.

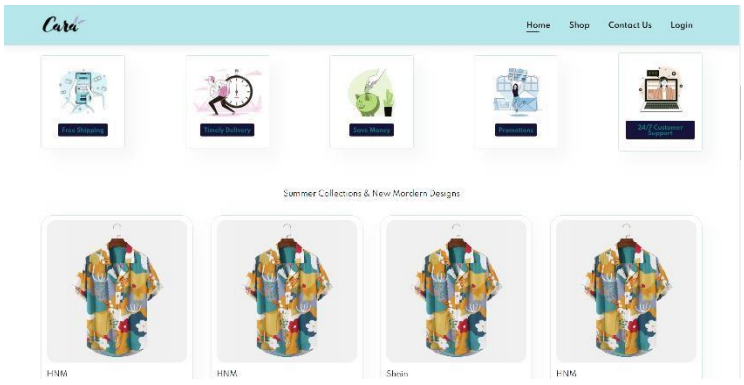
Django views:

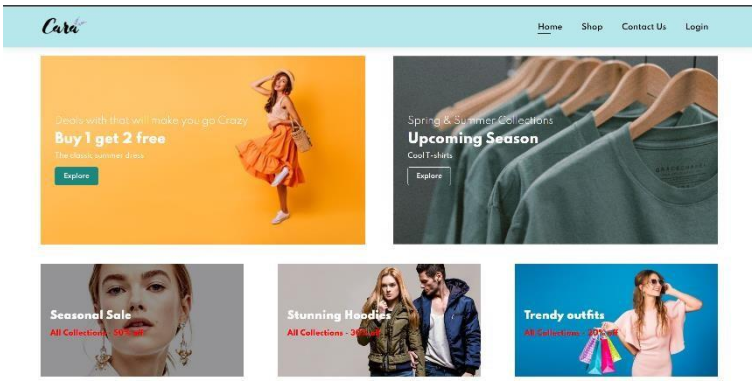
The home, list, detail, add and edit functions have been implemented as needed using Django.

Home View:

- Once a user accesses the URL, the index.html page is rendered which attracts users to the store's attractive offers, since this is an ecommerce website, the users no matter logged in or not are provide with the offers and the fashion choices available for them.

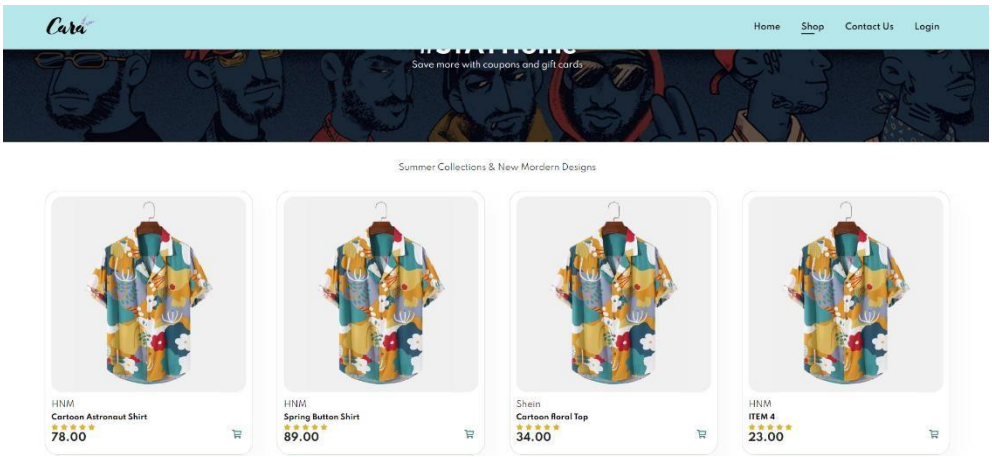






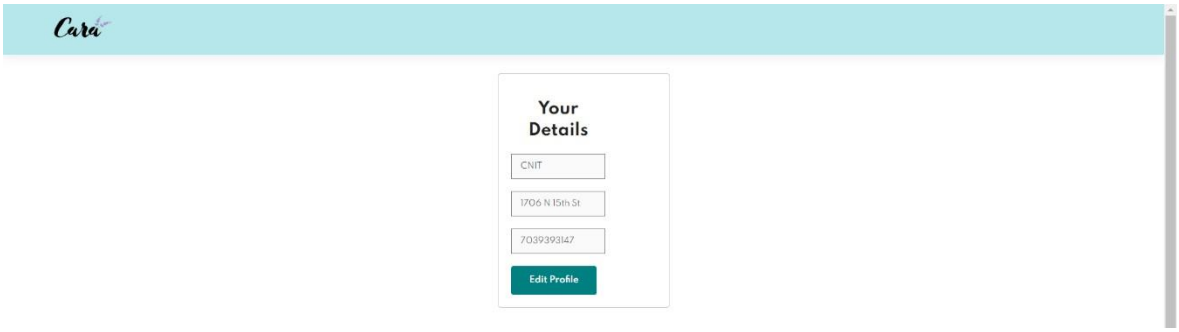
List View:

- The users have access to the Shop page via the NAVBAR where they are able to view various different products whose Name, Description and price have all been dynamically populated from the SQLite3 Database using API calls.



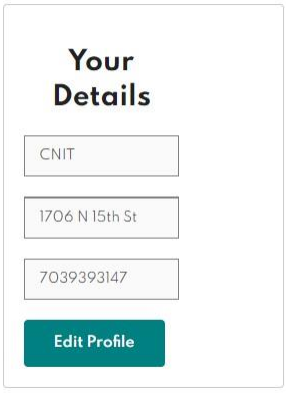
Detail View:

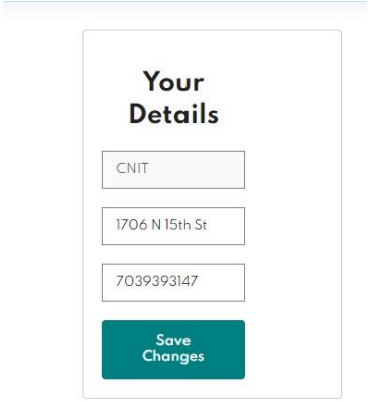
The Website consists of both a profile page where information about the user is displayed:



EDIT View:

For a new incoming user, the address and the phone number are both displayed with placeholder text allowing the user to edit the profile and save it. Once saved the new details are rendered.





Add View:

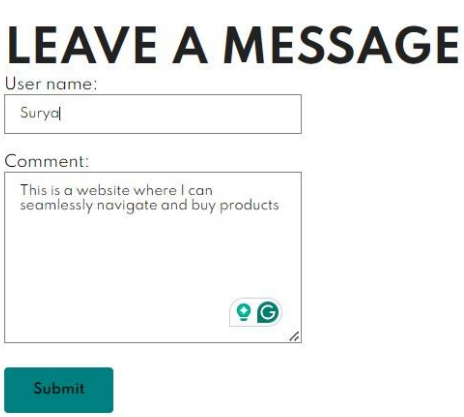
Multiple functionalities of the website provide the user with the ability to add details to the database. As shown in the previous view, the user not only is able to see the details of their profile but also is able to add new information in the case of a new user.

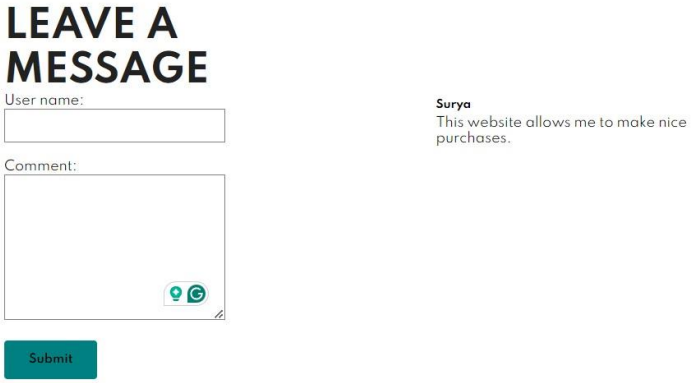
Similarly, once logged into the system, the user is able to add feedback or reviews to the website which are stored in the database. This feature is present on the Contact Us TAB of the website.

The other add view which is available is the Signup Sheet for registration of user, however this is unauthenticated.

Dynamic, Asynchronous Interactions:

The JavaScript function manages the submission and display of user feedback using AJAX requests. When a user submits feedback via the form, an asynchronous request is sent to the server, where the data is processed and stored. Upon successful submission, a JSON response is received, updating the webpage's content dynamically to reflect the new feedback. This seamless process enhances user interaction by providing real-time feedback without reloading the page.





Integrating external data from a web service API:

I have used the google maps API to show the location of the head office, this is done in order to allow customers to have a better understanding of how the store can be contacted in case of any issue with their orders.

GET IN TOUCH

Visit one of our agency locations or contact us today

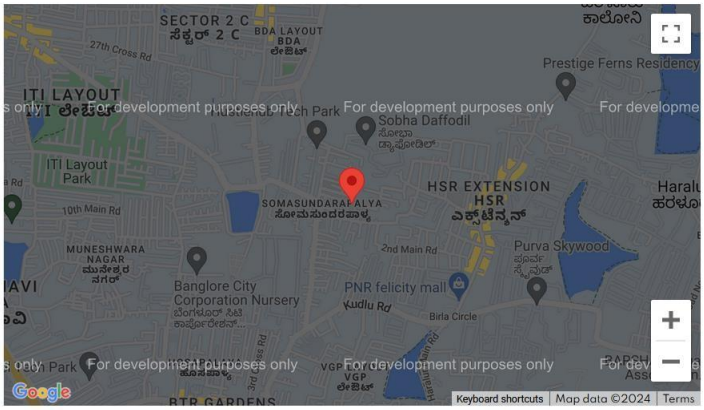
Head Office

Royal East, HSR Second Sector, Bengaluru

lakshmis@purdue.edu

+91 7538860269

Monday to Saturday 9:00 AM to 16:00 PM



External Libraries used for the project :

1. Django Rest Framework
2. DB-SQLite
3. Google Maps API