Desi Mart Application Development Report

Student: Sairaj Naik Guguloth
Note: Deployment url: https://desi-mart.herokuapp.com/.

Introduction

DesiMart is ecommerce website which sells Indian grocery which allows customers to purchase a wide variety of food items from the comfort of their homes. It is a one stop destination to find everything Indian. This application has a very user-friendly interface that makes it easy for customers, add items to their shopping cart, and complete their purchase with a few clicks. It provides a convenient and accessible way for customers to purchase high-quality products which are essential to Indian cooking and dietary habits.

Design

This application was designed using the Model, View and Template (MVT) architectural pattern. This architecture helped us develop the application into logical components, such as the model which contains the business logic, the view which contains the application logic and the templates that contain the presentation logic. Python Django has been used to develop this web application. Django makes web development easy by including support for tasks such as creating models for database, routes for URLs, an interface for administration etc. Even though Django is not as flexible as other lightweight frameworks such as Flask, it enables us to build a website with adequate inbuilt libraries and settings.

Development and Implementation

This application was developed in the codio environment. The data from the dataset was copied to a file named data.csv, after filtering the number of records. The models.py file contains the database models, which uses the Object Relational Mapper (ORM), where the database is created from python classes. The fields of the database are created from the members of the class . The parse_csv.py file parses this csv file and adds data to the Year and Temperature models.

views.py file contains the methods that act as controllers between the templates and the model files. It fetches the values from the models according to the request and sends the fetched values to the corresponding template, reducing the dependency between the models and templates. The HTML templates are generated by the application of the Django Template Language , that includes a distinct syntax for inserting the dynamic content. Various templates for different pages of the application extend from the base template named base.html. Tests were written using Python's unittest library, to test the views methods and the database models. The software was deployed in the early stage of its development, which aligns with the concept of Continuous Delivery, where the software is delivered regularly after the testing process. This improves the productivity and the quality of the application .

Features

- Login, Register and continue as guest options for a user. Logged users can view orders
- Product filter feature of various categories, Search feature to browse product
- Product view page with details of the product, Similar product recommendations .
- Admin features like dashboard viewing, access to all user orders and reports.
- Add to cart and check out feature for a customer to place an order