FROSTY DELIGHTS

A Ice Cream Website Using Django and HTML,CSS

TEAM
INTRODUCTION

- Poornachandra D H
- Nikhil Gowda M G
- Vikas B
- Pramod

INTRODUCTION TO DJANGO

WHAT IS DJANGO?

Django is a Python framework that makes it easier to create web sites using Python.
Django takes care of the difficult stuff so that you can concentrate on building your web applications.
Django emphasizes reusability of components, and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete)

INTRODUCTION TO
DJANGO

HOW DOES DJANGO WORK?

Django follows the MVT design pattern (Model View Template).

- **Model - The data you want to present, usually data from a database.**
- **View - A request handler that returns the relevant template and content - based on the request from the user.**
- **Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.**

INTRODUCTION TO
DJANGO

MODEL

- The model provides data from the database.
- In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.
- The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it.
- Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.
- The models are usually located in a file called models.py.

INTRODUCTION TO
DJANGO

VIEW

- A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result.
- The views are usually located in a file called views.py.

TEMPLATE

- A template is a file where you describe how the result should be represented.
- Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results

USES OF DJANGO

- **Web applications**
  Django is a platform for creating web applications quickly and efficiently. It's used to develop common functions like authentication, database retrieval .

- **Customizable apps**
  Django is used to develop highly customizable apps, such as social media websites.

- **Document management systems**
  Django can be used to create document management systems, CRM systems, and real estate property evaluation systems.

- **Data analysis**
  Django can be used to create platforms for data analysis and complex calculations.

OUR

PROJECT OVERVIEW

Purpose of the Website:

Provide users with an interactive platform to explore and purchase ice cream products, showcase a variety of options.

Target Audience:

Families and individuals looking for premium ice cream products

USE OF TECH STACKS

Django Framework
Utilize Django's robust framework for building scalable web applications. Benefit from Django's built- in security features to protect user data. Leverage Django's modular architecture for efficient development and maintenance.

HTML and CSS for Frontend Development
Create visually appealing and responsive web pages using HTML. Style the website with CSs to ensure a consistent and professional look. Implement modern web design practices to enhance user experience.

INSTALLATION AND CONFIGURATION

Installing Python And Django

● Using pip to install Django Verifying the installation Installing Django in a virtual environment

Setting up the Django Environment

● Creating a virtual environment
Activating the virtual environment
Installing necessary packages

CREATING THE DJANGO APP

Starting a New Django App

Using a command to create a new app settings.py Registering the new app in Understanding the app directory structure

Structuring the App

Organizing models, views, and templates Creating necessary directories for Defining a settings in static and media files Best practices for maintaining an app structure

Configuring App Settings

Defining app- specifica settings in settings.py Setting up URLs for the new app Configuring app- level middleware

DATABASE SETUP

Configuring Database Settings

Modifying the settings.py file to connect to the database Setting up database user and database-specific permissions Testing the database connection

Migrating Database Models

Creating initial models in the app Running the initial migrations Understanding the migration workflow

DYNAMIC CONTENT MANAGEMENT

Admin Panel Customization

Adding and modifying admin filters
Configuring admin site for better data representation

Managing Ice Cream Listings

Creating models for ice cream products ,Displaying listings on the frontend

Updating Content Dynamically

Implementing content update forms for real- time content updates Ensuring data validation during updates

THANK YOU