



Campus Craves

CS253 Course Project

Supervised Learners

Our Team

Course Instructor: Dr. Indranil Saha

Mentor TA: Mr. Ashish Singh

Group Members:

- Aayush Gautam 220020
- Abhishek Kumar 210040
- Anand Chutani 220130
- Anany Dev Choudhary 220135
- Chatla Sowmya Sri 200293
- Lokesh Mehra 220591
- Pranshu Mani Tripathi 220800
- Siddharth Pathak 211034
- Sidharth A S 221056
- Sparsh Gupta 221084
- Ujjwal Kumar 211123

Table of contents

01 **Introduction**

Utility
Audience

02 **Major Requirements**

Seller's Requirements
Buyer's Requirements

03 **Implementation**

Architecture
Frontend
Backend etc.

04 **UI Demo**

05 **Future Plans**

06 **Lessons Learnt**



01

Introduction

The product, Campus Craves primarily aims to digitize all the canteens throughout IITK Campus, making ordering during peak hours hassle-free.



Intended Audience

IITK Canteen Owners and Campus community

Buyer: Those who'll buy product/services i.e. the campus residents.

Seller: Those who sell products/services i.e. the Canteen Owners



02

Major

Requirements



Major Requirements

Seller

- Create a seller account / store.
- Add/delete/modify categories in catalog.
- Add/delete/modify products/services in categories.
- Update the inventory if an item in/out of stock.
- View the list of orders placed/in-process
- Order Status: Confirmed/Delivered/Cancelled

Major Requirements

Buyer

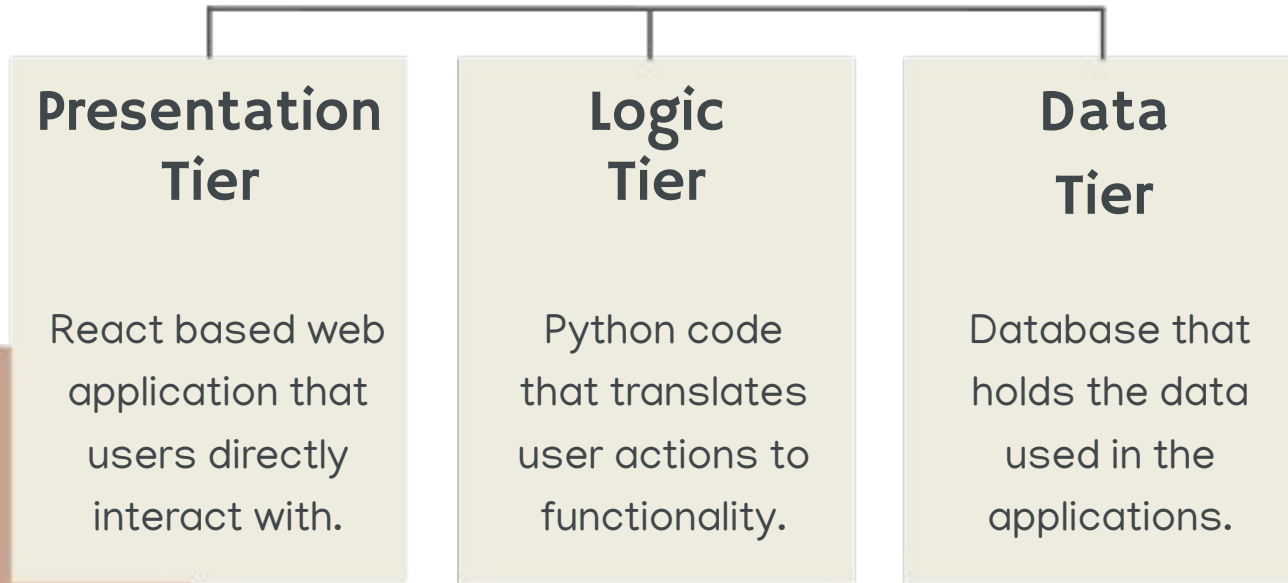
- Sign in/sign up to the account via email.
- Check status of previous orders
- Add/delete/modify personal details.
- Update the cart and manage the quantities.
- View the list of orders
- Order Status: Confirmed/Delivered/Cancelled



03

Implementation

Three Tier Architecture



Frontend

React JS

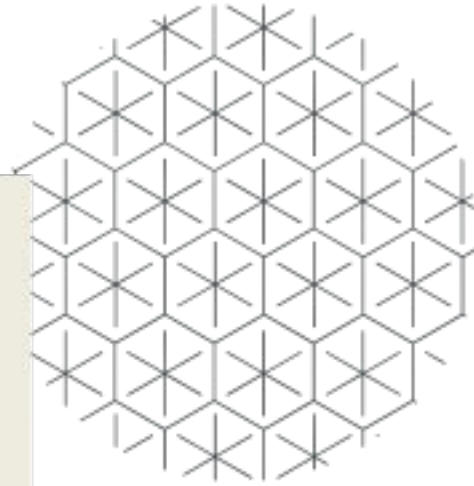
Javascript framework that makes the website dynamic.

Tailwind CSS

A utility-first CSS framework designed to simplify and expedite web development by providing a set of predefined classes

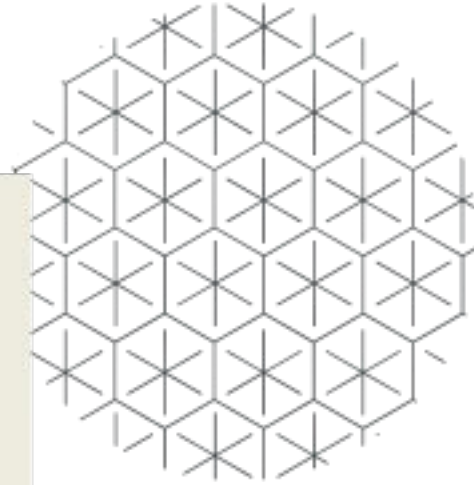
Backend

Backend is implemented in Python. We used Django, an open-sourced python based framework used to create web services on top of python.



Database

We have used SQLite3, a lightweight, file-based relational database bundled with Django. Django ORM stands for **Django Object-Relational Mapping**, it simplifies development and requires minimal setup.



OTP Service

SendGrid

We use SendGrid's SMTP servers to send our OTPs to users via email.



Authentication and Authorization

Authentication: stored the email and password in our database. The password uses **cryptographic** techniques of **hashing and salting** to prevent the data being exposed in case of data leaks. **Authorization:** we have made use of **JWT (Json Web Token)**



Development & Version Control Environment



Git

Used as our version control system

GitHub

Used to manage our repositories and collaborators



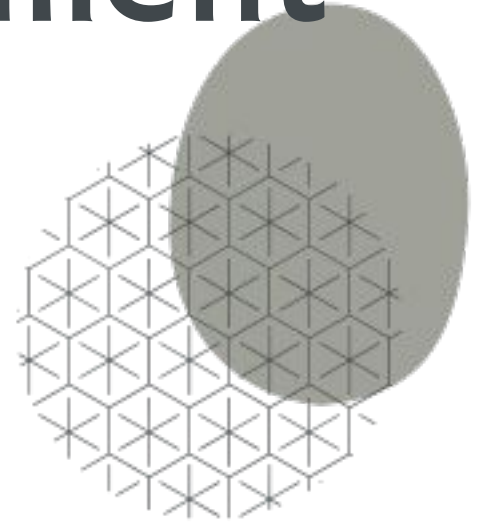
04

UI Demo

Let's check out our website

05

Future Development Plans



Future Development Plans

- Adding an online real payment option.
- Deploying a corresponding mobile app.
- Customer Reviews and Ratings.
- Possibility of using other languages in the future.



06

Lessons Learnt

Our Leanings

- Full-stack development using React, Django, and SQLite
- Use version control (Git/GitHub) for collaboration
- Time management across multiple project components
- Communication strategies that worked well
- How you resolved conflicting ideas or approaches

Acknowledgement

We would like to thank the Instructor of the course, **Dr. Indranil Saha**, for teaching us Software Development and Operations concepts.

We would also like to thank the TA in-charge, **Mr. Ashish Singh**, for guiding us throughout the process of making this software.



Thank You

