

FINAL PROJECT REPORT

FOOD-EZ



Group Details

1. Akansh Mittal	-	1910110039
2. Aman Yadav	-	1910110052
3. Aniket Gupta	-	1910110066
4. Ayush Varma	-	1910110108

INDEX

1. Introduction
2. Features
3. Problems addressed
4. JWT Authentication and
DataBase ORM
- 5 Food Analysis and Statistics
6. Database Flowcharts and UML
diagram

Introduction

Motivated from the problem of standing in long queues in SNU mess to order food, we developed Food-Ez which is an online ordering and payment management system. Our product will assist in crowd management and payment services, by ordering food on mobile phones which will provide customers with a seamless dining experience while adhering to all social norms. Now, you no longer need to wait long to order items, simply add items in your cart, place order and sit and relax till your order is prepared and receive it.

Features

Our product makes the students login with their SNU Net IDs and then they will be taken to a food ordering page where all products and their prices and description will be displayed. The students can add dishes to their cart, place their order and pay with their DH card number or via paytm/UPI. Their order will be displayed in the “My Orders” section with the status of the order notifying if it’s ready. After the status changes to “Ready”, the student can go to collect their order.

Apart from the student login, we have an admin login route, which automatically determines an admin user and takes him to the dashboard. The dashboard displays various statistics like Total Users, Number of Order, Total Sales, Number of Pending Orders etc. It also displays charts that can be used to analyse consumption of food to minimize food wastage and can also be used to check the financials i.e. show the total expenses v/s total sales to analyze profits. Apart from the main dashboard, the admin has a bunch of other options at his disposal. These include “Users” that displays all the users in the database; “Products” which displays all the products/dishes that the mess sells; “Orders”

which displays all the orders placed by the students; the “Orders” section automatically filters to show the current pending orders. However, with just one click on the “Show All” button, all the orders can be displayed. The admin can also add new products by going to the “Add Product” section and entering product name, its price, product description, quantity etc. The quantity of the product can also be updated using the “Update Stock” option. This helps to manage the inventory and orders can be placed according to the quantity only, unlike the traditional system where the cashier places an order for an item which has already run out of stock at the food counter.

Problems Addressed

With our product, we have tried to tackle various problems that we have faced in the SNU Dining Hall itself. These problems include

1. **Waiting in long queues for hours** - With a digital ordering platform, many students can simultaneously order food and therefore, the students need not wait in queues to order food.
2. **Hectic ordering Process** - With digital payments, the ordering process is quick, easy and efficient.
3. **Lack of Order Management** - The unavailability of physical receipts and an organized table of orders will make it easier for the mess management to deliver correct orders to the correct people without creating chaos.
4. **Ordering already out-of-stock dishes** - The use of an inventory system also avoids the placing of orders of items that run out of stock, which is a common problem that we face in the SNU DH.
5. **Lack of Crowd Management** - The use of digital platforms reduces physical interaction in the mess which is necessary in current covid times and since we will not be using physical receipts for ordering, it will save a lot of paper.

6. **Excessive Food Wastage** - The display of statistics of food consumption helps the mess management to prepare the adequate amount of food after taking proper and reliable data which helps to minimize food wastage.

JWT Authentication

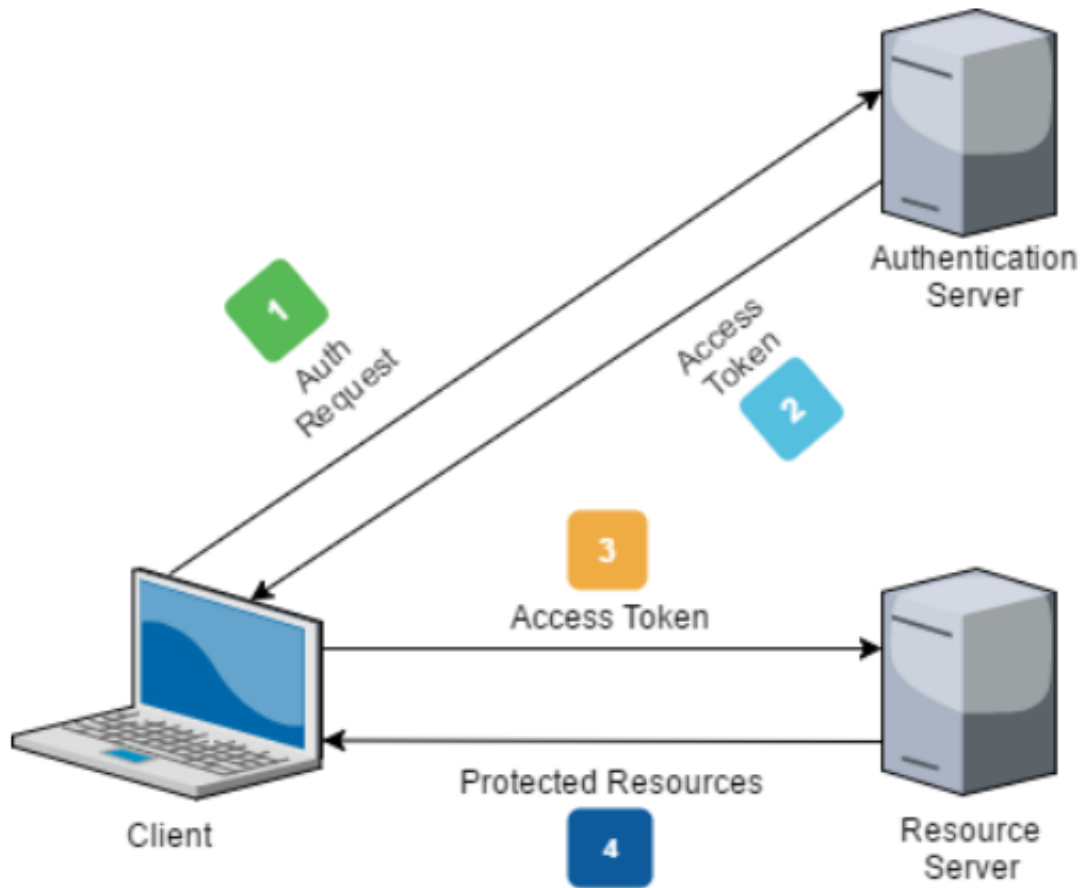
```
ALGORITHM = "HS256"

def create_access_token(data: dict,
                        expires_delta: Optional[timedelta] = None):
    to_encode = data.copy()
    if expires_delta:
        expire = datetime.utcnow() + expires_delta
    else:
        expire = datetime.utcnow() + timedelta(minutes=15)
    to_encode.update({"exp": expire})
    encoded_jwt = jwt.encode(to_encode,
                             SECRET_KEY,
                             algorithm=ALGORITHM)

    return encoded_jwt

def verify_token(token : str, credentials_exception):
    try:
        payload = jwt.decode(token, SECRET_KEY, algorithms=[ALGORITHM])
        username: str = payload.get("sub")
        if username is None:
            raise credentials_exception
        token_data = TokenData(username=username)
        return token_data
    except JWTError:
        raise credentials_exception
```

JWT Authentication Flow



DataBase Models for SQLAlchemy ORM

```
# database models

class User(Base):
    __tablename__ = "users"
    id = Column(Integer, primary_key=True)
    name = Column(String)
    username = Column(String)
    password = Column(String)
    is_superuser = Column(Boolean, default=False)
    orders = relationship('Orders', back_populates='user_name')

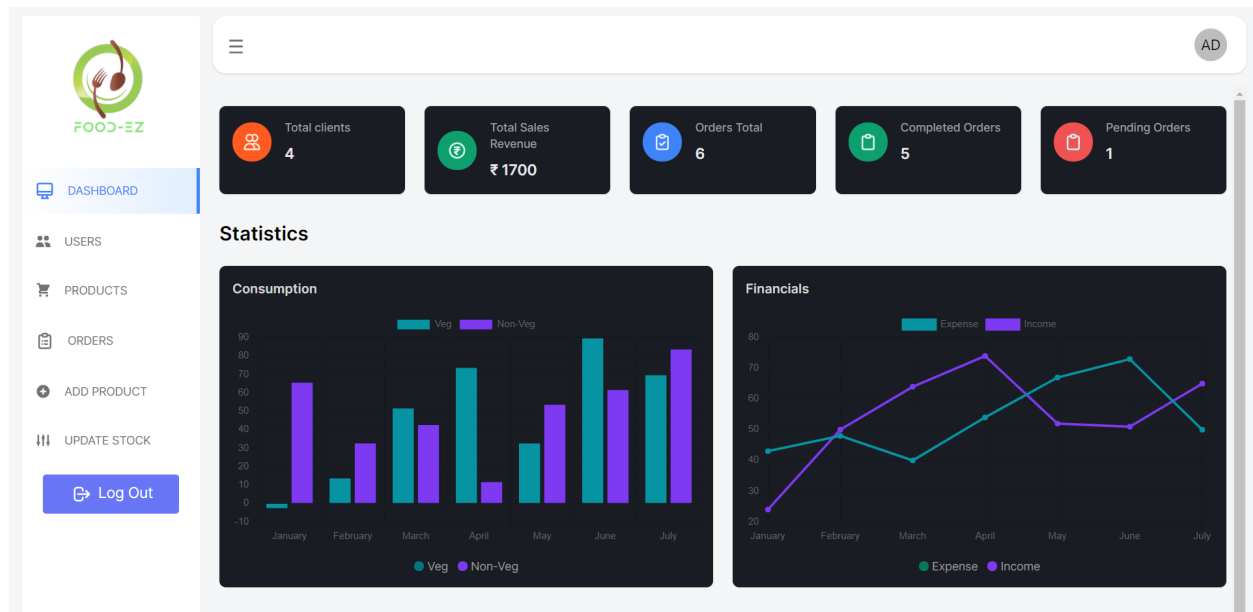
class Product(Base):
    __tablename__ = "products"
    id = Column(Integer, primary_key=True)
    name = Column(String)
    description = Column(String)
    price = Column(Float)
    qty = Column(Integer)
    is_veg = Column(Boolean)
    orders = relationship('Orders', back_populates='product_name')

class Orders(Base):
    __tablename__ = "orders"
    id = Column(Integer, primary_key=True)
    product_id = Column(Integer, ForeignKey('products.id'))
    user_id = Column(Integer, ForeignKey('users.id'))
    qty = Column(Integer)
    status = Column(String, default='pending')
    created_at = Column(DateTime(timezone=True), server_default=func.now())
    user_name = relationship('User', back_populates='orders')
    product_name = relationship('Product', back_populates='orders')
```

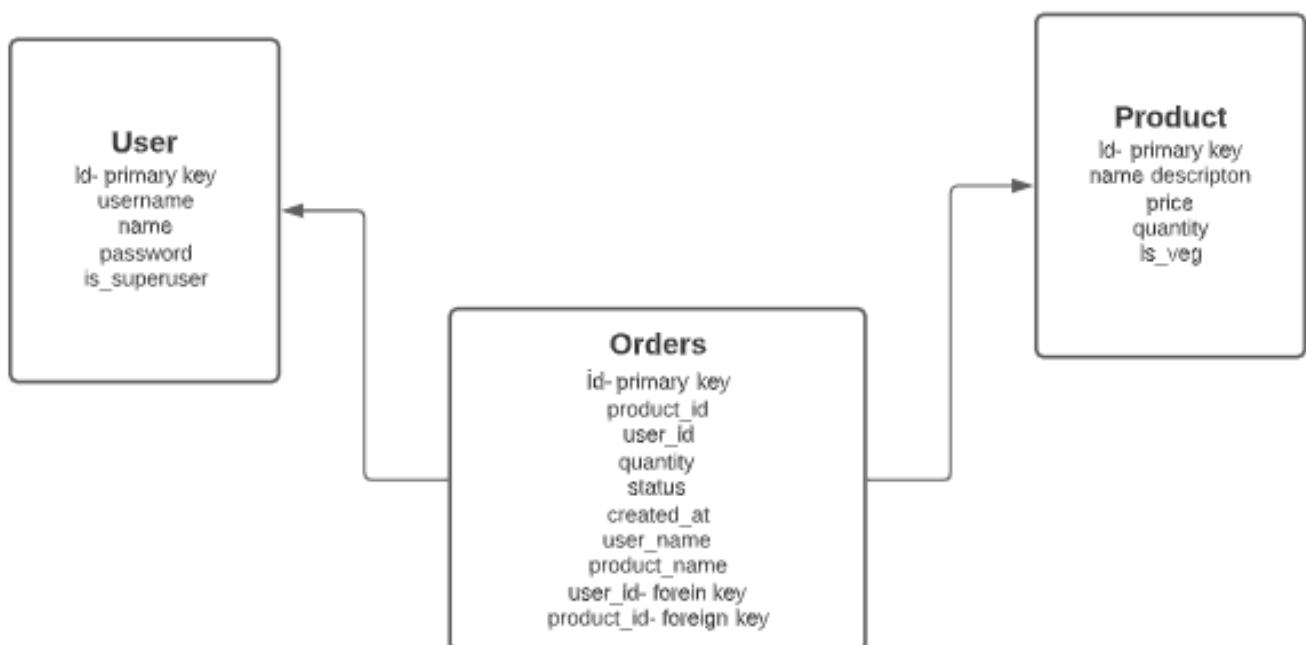
Food Analysis and Statistics

By keeping record of daily food consumption of both Veg and Non-Veg, we determine the overall wastage of food in mess for every month. First bar graph represents Veg and Non Veg consumption every month looking at which, the mess committee can then prepare food in adequate quantity on the basis of these statistics so that the food is just enough to feed people and the wastage can be minimized.

In the second graph, we analyze the overall expenses of a mess with its revenue. For every month, admin can, with the help of the graph, identify how much the cost of the food was and how much revenue was earned with the sales and thus, will realize the profit margins for the month. In short, the statistics displayed will aid the mess management deliver better services and earn more profits.



Schema of the Database



UML diagram for database

UML Diagram for the database

