**Goals**
- Demonstrate the process of identifying restaurant customers via face recognition tool to promote products and services designed specifically for them.
- Fully automate the self-service kiosk with personalized menu items.
- Create a dynamic menu in the form of a scalable web application.

**Motivations and Objectives**

- **Motivations**
  - Excess time and energy that becomes wasted in ordering efforts especially during peak hours
  - Lack of modern-day facial recognition technology on kiosks to self-automate the ordering process.
  - Lack of past transactions & order recommendations on kiosks

- **Objectives**
  - Create an efficient way to choose your favorite order or an order that you’d prefer based on your purchase history (which allows minimal time wasted in recreating/picking an order as well as waiting in line)
  - Utilize NodeJS, AngularJS, mongoDB, Microsoft Face API

**Challenges**
- Precisely and accurately utilize the Microsoft Face API to distinguish faces as well as connecting mongoDB to our server.
- Gathering large dataset with transaction data to train our model for better precision.
- Learning how to train a model with a large data in order to further improve our recognition and recommender system.

**Acknowledgement**
We would like to thank Professor Striki for guidance and encouragement.

**Methodology**

- **Face Recognition**
  - A front-facing camera captures the face image of the customer and the backend function sends a POST request to Microsoft Face API to identify the user.
  - Face Detect API will verify the image taken has a human face in it and sends a unique Face ID with a metadata of the user’s facial features.
  - Face Identify API compares the face across all the face images stored in our database and returns the closest matching user with a confidence threshold > 0.92.

- **Recommender system**
  - We used js-recommender library to calculate similarity scores (Euclidean distances) based on purchase frequency.
  - Similar users to the customers will tend to have a purchase history similar to that of customer
  - Recommend the previous purchases from past similar users to the customer since they have similar purchase patterns

**Results**

- **Results**
  - Our output shows the purchase history of the user once their face becomes recognized.
  - The recommended items that the user would most likely consider buying would be presented as well

**References**