**FOOD HUNT**

**Technologies Used-  
1) Java  
2) JavaScript**

**Used: -**

1. **JDBC**
2. **Servlets**

**Going to follow MV (Model, View, Controller) architecture – This will lead to increased maintainability (we won’t be putting all the code in the servlet class)**

**View- JSP or HTML files**

**Controller – Servlet classes that intercepts request and response**

**Model – Data Access Object (DAO) class that talks to the database.**

**Modules Made**

**Each module comprises of**

1. Data Layer – Database file establishing Data base connections
2. Business layer – Classes which has different functionalities
3. Servlets

**1)Deals and Offers –** a)Restaurant Idb) Premium customer or notc) Deal Idd) Deal Active or Not

**2) Reservation –** a) Reservation Id  
 b) Booking Time/Booking Id /Booking Date / Booked Seats  
 c) Restaurant Id  
 d) Feedback  
 e) Cancelled  
 f) Username  
 g) Establishing Database Connection ( import java.sql.Connection)

**3) Restaurant Management**

**Business Layer** a) Add Restaurant  
 b) Modify Restaurant  
 c) Delete Restaurant

**Data Layer** a) Restaurant Id  
 b) Restaurant Name  
 c) Restaurant Address  
 d) Overall Rating  
 e) Table Arrangement  
 f) Homedelivery  
 g) Pictures  
 h) Get Latitude  
 i) Get Longitude

1. **Searching the Restaurants**

**Business Layer**

1. Nearby Restaurants – Based on the similarity score
2. **User Management**

**Data Layer**

a) Username

b) Password

c) Is Veg

d) Premium

e) Email Address

f) Address

g) Active user or not

**The biggest learning of this project :-**

**Fetching Location of Restaurants**

Using Google Places API’s for finding nearest restaurants in my Java code. Now client libraries are even provided to ease it. You don’t need API key for your application in this for authentication.

**Problems Faced**

1. Fetching records from database was taking time.

**Solutions**1) Not using nested queries

2) Not using inner join in sql queries and using LEFT and RIGHT JOIN (By default inner join) (Inner Join is slow as it has to find common in both the tables). Multiple INNER JOIN is too slow for SQL server.

3)Used Indexing to increase the speed of database operations.

1. Session Handling was not working properly.

**Solutions** 1)Servlet Specification has HTTPSession API for session tracking.  
 2) Server sets up Session Id to track each user  
 3) Session Id exchange by 2 ways either by Cookies or URL rewriting

**Cookies**

1. **In servlets cookie is an API**
2. **It is done by request.getSession() API call**

**URL rewriting**

1. **Fallback option for Session Management and comes into picture only when cookies are disabled.**
2. **URL rewritten with session id at the end of the URL.**
3. **Developers need to encode all the URL’s ---tedious but important in case cookies are disabled.**

**Data Structure Used**

1. ArrayList
2. List
3. Used Colections compareTo method