**Caferteria Problem**

* **Problem:**
* **Users**:

1. User.
2. Admin.

* **Functionality of**

1. **Admin**:

* Log-In
* Add Food & Food details.
* Update & Delete food.
* Change food availability status.
* CRUD operations on orders.

1. **User**:

* View Food & Food details.
* Registration of new user.
* Log-in of existing user.
* CRUD operation on CART.
* OTP validation.
* Place the order & check order.
* Profile management.
* **Modules(Apps)**:

1. **Admin**
2. **User**
3. **Food**
4. **Pages**

**TABLES**

1. **Cusine(categories)**:

Id (Primary Key)

Category varchar(100)

1. **Food**:

Id (Primary Key)

Name of Food varchar(100)

Price number(10,2)

Available Boolean

Cusine (Foreign Key to Cusine Table)

1. **Order**:

Id (Primary Key)

User-Id (Foreign Key to User Table)

Order-details varchar(1000){order will be stored in form of an array}

Total-price number(10,2)

Time of order date & time

Status number(1)[-1 processing,0 ready,1 delete]

**Work-Flow**:

1. Basic Page(Index, About Us, Contact Us).
2. Admin Log-In.
3. CRUD operation on Food.
4. User Authentication.
5. Cart.
6. OTP.
7. Order & Order Status

**3-Tier Architecture**

(Client, Server, Database)

HTTPs

Client

Server

Request Queries

DB

Response Data

**Client**:

1. Understands Front-End Language.
2. HTML, CSS, JS is used.

**Server**:

1. Takes request from **Client** & sends response as mentioned.
2. If any data is required then sends queries to database and fetches data and gives it to **Client**.
3. Uses Back-end tools like Django, Node.js, php.

**Database**:

1. Commonly use MySQL, Oracle, Mongo(no-sql) databases.

**Library**: Some built-in functions are present and we have to use

them as mentioned.

**Framework**: Built-in functions can be slightly modified and used

after importing them

Django is a battery included Framework,

i.e. things like validation & session management are previously available.

& can be used with some modification.

Browser

DATABASE

TEMPLATES

URL

ROOT URL

APP URL

VIEW

MODEL(ORM)

Django Server uses MVT architechture

(Model View Template)