

Project Design Phase-I

Solution Architecture

Date	31.10. 2023
Team ID	NM2023TMID032
Project Name	Project –food tracking system
Maximum Marks	4 Marks

Solution Architecture:

Core Features of Customer App

Searching menu: Allow your users to search for different restaurants, cafes by location, and cuisines. Using the search filter, users can easily find their favorite eating places, list menu, offers, etc.

Order placement: The user can place an order of selected dishes and food. They just need to cross-verify their preferred dish, delivery time, and proceed check-out.

Tracking Delivery Partners: With real-time tracking features, it becomes easy for users to track delivery drivers and know their real-time location information. Users can check the time taken by the food delivery executive to deliver their parcel.

Payment gateway integration: You provide the users with multiple payment options like credit/debit cards, different wallets like Google Pay, Paytm, Phonepe, UPI, etc

Core Features of Delivery Partner

Delivery Partner's profile: Through this feature, a driver can keep his profile update. It contains his full name, address, email, contact number, photo, and other personal information.

Notification for orders: Through push notifications, drivers can get constant updates & alerts for new orders. It will help in the accurate delivery service of your restaurant.

Map for the delivery route: Integrate Google Map or other providers and allow drivers to choose the shortest and fastest routes to reach the location.

Core Features of Food Partners/ Restaurants

Restaurant Profile/Menu: Through this feature, a restaurant owner can add their restaurant details, menu and its availability, price, preparation times, etc

Notification for orders: Through push notifications, Restaurants can get constant updates

& alerts for new orders. It will help in the accurate delivery service of your restaurant.

Notifications for Pickup Partners: They will get alerts about delivery partners, their location when they will pick up, etc.

Payment Details: Information about the payment received from the food delivery system for their orders

Core Features for Food Ordering System Admin

Restaurant management: Being on the admin panel, one can directly manage all the restaurants by adding, updating, and removing any eating joint from the list. He can also check active restaurant status.

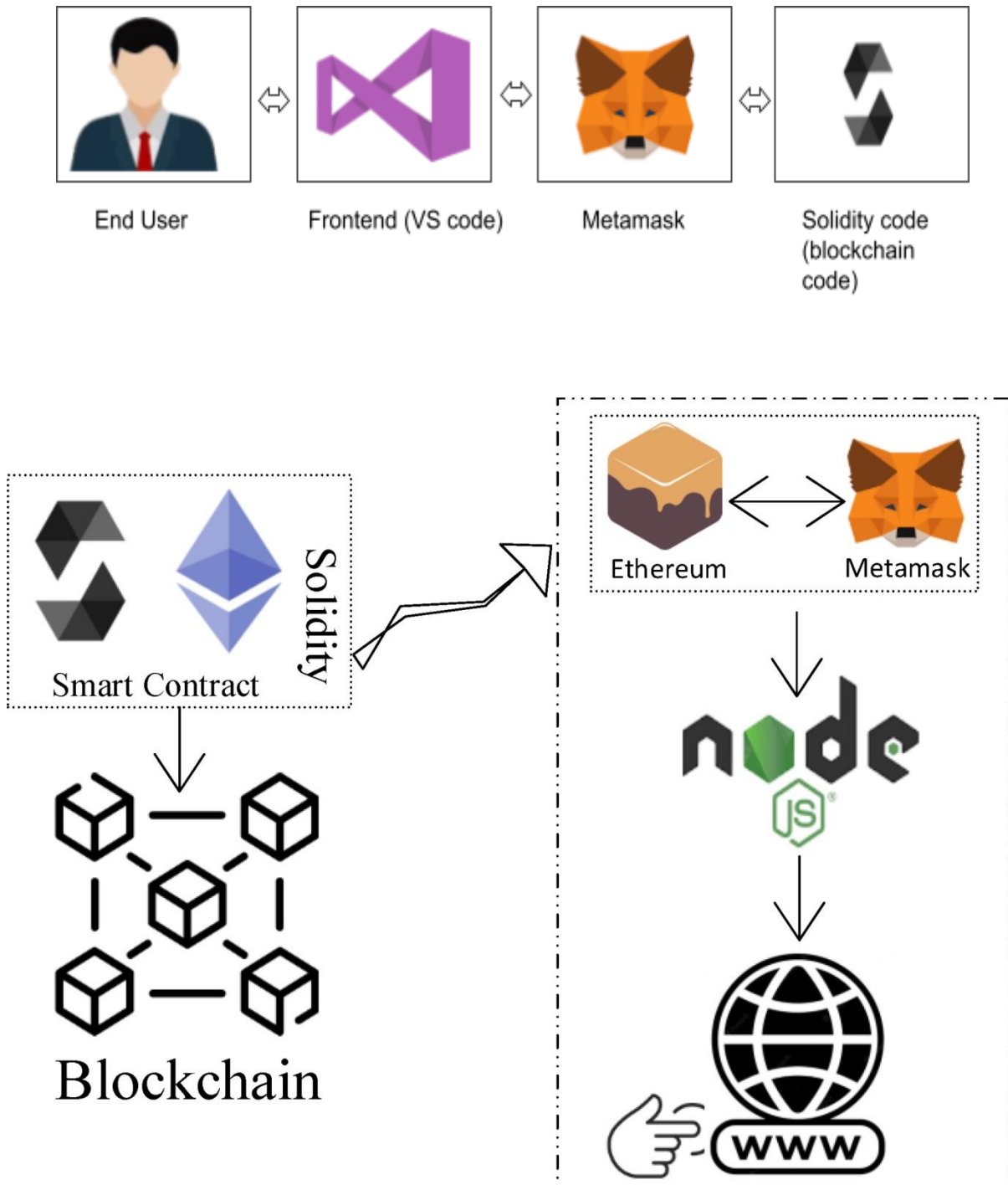
Analytics & report generation: Using analysis and report features, you can get real-time insights of reports and other accounting information which helps you to identify the growth and opportunities to expand reach.

Monitoring every action: Monitor all the food orders, delivery partners, deliveries, reviews & ratings of delivery partners, canceled orders, and other important data related to the driver's performance.

Application Flow

We are considering here microservices-based architecture. Different services are listed in the architecture diagram

Example - Solution Architecture Diagram:



Steps to complete the project

The screenshot displays the Remix Ethereum IDE interface. The top-left sidebar contains the 'DEPLOY & RUN TRANSACTIONS' panel, which is set to the 'Custom (80001) network'. The 'ACCOUNT' section shows the address '0xb2a...16402 (0.19681276 ETH)'. The 'GAS LIMIT' is set to '3000000'. The 'VALUE' is set to '0 Wei'. The 'CONTRACT' section shows 'FoodTracking - food tracking system.sol' with the 'evm version: paris' and a 'Deploy' button. Below the 'Deploy' button is a 'Publish to IPFS' checkbox. The 'Transactions recorded' section shows one transaction. The 'Deployed Contracts' section shows the 'FOODTRACKING AT 0xF2D...239E' contract.

The main editor displays the Solidity code for the 'FoodTracking' contract. The code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3
4 contract FoodTracking {
5     address public owner;
6
7     enum FoodStatus {
8         Unverified,
9         Verified,
10        Consumed
11    }
12
13    struct FoodItem {
14        string itemId;
15        string productName;
16        string origin;
17        uint256 sentTimestamp;
18        FoodStatus status;
19    }
20
21    mapping(string => FoodItem) public foodItems;
22
23    event FoodItemSent(
24        string indexed itemId,
25        string productName,
```

The bottom-right panel shows the transaction details for the deployment. It indicates that the contract was deployed at block 41822917, transaction index 6, from the account '0xb2a...16402' to the contract 'FoodTracking.(constructor)'. The value was 0 wei, and the data was 0x608...20833. The logs show the deployment hash '0x113...b02b0'.

Chro x Meta x (2) V x Dow x Chai x 6_Pr x Chai x Addi x Mun x Poly x Remi x nme x +

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null

Home food tracking system.sol README.txt

SOLIDITY COMPILER

COMPILER +

0.8.18+commit.87f61d96

☐ Include nightly builds

Solidity compiler

☐ Compile

☐ Hide warnings

Advanced Configurations

Compile food tracking system...

Compile and Run script

CONTRACT

FoodTracking (food tracking system.sol)

Publish on Ipfs

Publish on Swarm

Compilation Details

ABI Bytecode

103

0

☐ listen on all transactions

Search with transaction hash or address

✓ [block:41822917 txIndex:6] from: 0xb2a...16402 to: FoodTracking.(constructor) value: 0 wei data: 0x608...20033 logs: 0

hash: 0x113...b02b0

status

true Transaction mined and execution succeed

transaction hash

0xb48013af98abfffb30838b371e49b73aaf8eb578959fe1c8ec67bf3a696b3188

block hash

0x113c7881eb257d6663e7836d566449d2489e8888a2be1dc4749fdd0f68b02b0

block number

41822917

contract address

0xf2d0845ABA67f6f436ff0789A422bA0180d23986

from

0xb2a646Ca263982A1FCAEDc53985cad85d5f16402

to

FoodTracking.(constructor)

gas

1274886 gas

transaction cost

1274886 gas

input

0x608...20033

decoded input

{}

decoded output

-

Debug

28°C Mostly cloudy

Search

16:09 30-10-2023