

## Project Design Phase-I Solution Architecture

<b>Date</b>	<b>26 October 2023</b>
<b>Team ID</b>	<b>NM2023TMID05931</b>
<b>Project Name</b>	<b>Food Tracking System</b>
<b>Maximum Marks</b>	<b>4 Marks</b>

### Solution Architecture:

Designing a solution architecture for a food tracking system involves several components to ensure efficiency, accuracy, and user-friendliness. Here's a high-level overview of the architecture:

- The best tech solution to solve existing business problems is UI [user interface] , Backend service , Data processing.
- Structure of project : User Interface , Backend service , Notification system.
- Characteristic : User-Friendly ,Accuracy , Customizability , Data Encryption , Authentication.
- Behavior : Real-time Updates , Intelligent Insights , Notifications and Engagement.
- Features : 1. User Profile Management:
  - User registration and login
  - Personal profile setup (age, weight, height, dietary preferences)
  - Profile editing and customization

## 2. Food Logging:

- Manual entry of food items (name, quantity, meal times)
- Barcode scanning for quick data entry
- Calorie and nutrient tracking
- Meal categorization (breakfast, lunch, dinner, snacks)
- Water intake tracking

## 3. Notifications and Reminders:

- Reminders for meal logging and hydration
- Achievement notifications
- Goal completion notifications

## • Development phases : 1. Planning:

- Requirement gathering and analysis
- Feasibility study
- Project planning and scope definition

## 2. Design:

- UI/UX design
- Database schema design
- System architecture design

## 3. Development:

- Frontend development
- Backend development
- Integration with external services

## 4. Testing:

- Unit testing
- Integration testing
- User acceptance testing

#### 5. Deployment:

- Deployment of the system on servers or cloud platforms
- App deployment on app stores

#### • Solution requirements :1. Scalability:

- Ability to handle a large number of users and data points
- Scalable server infrastructure for future growth

#### 2. Security:

- Data encryption during transmission and storage
- Secure user authentication and authorization mechanisms

#### 3. Reliability:

- High availability and uptime
- Reliable data backup and recovery mechanisms

#### 4. Performance:

- Fast response times for user interactions
- Efficient data processing algorithms

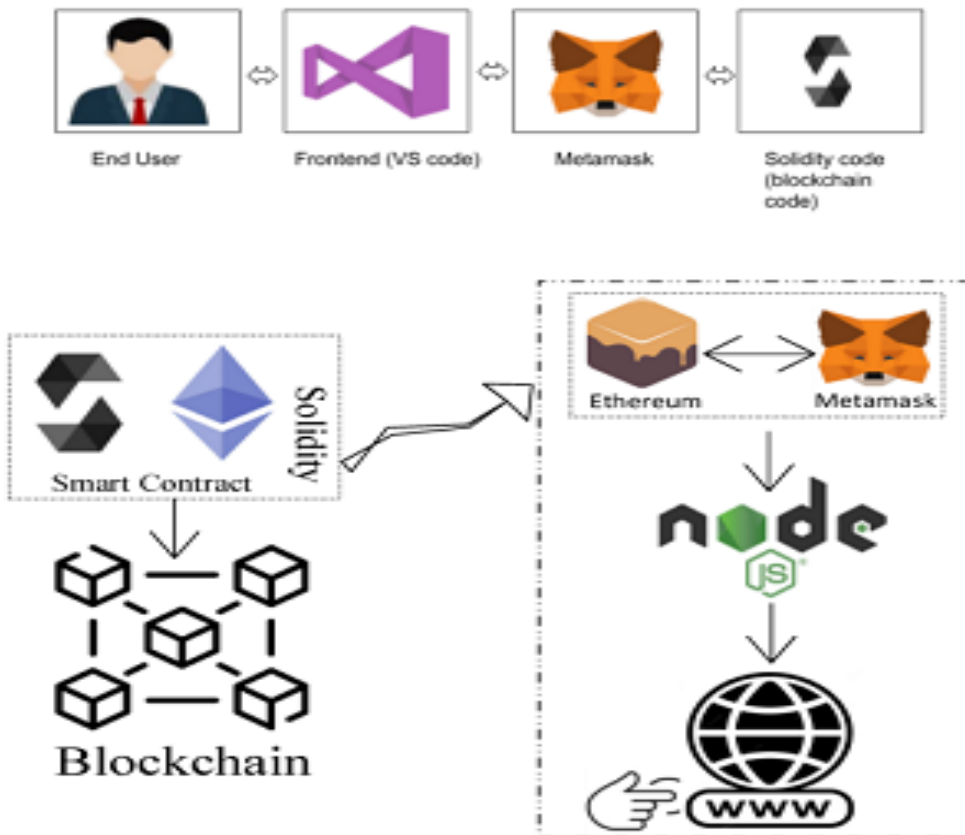
#### 5. Compliance:

- Compliance with data protection laws
- Adherence to health and nutrition data regulations

#### 6. Usability:

- Intuitive and user-friendly interfaces
- Accessibility features for users with disabilities
- Functional specifications :
  1. User Registration and Authentication:
    - Users can create accounts with unique usernames and passwords.
    - Secure authentication mechanisms, including two-factor authentication if needed.
  2. Food Logging:
    - Users can manually input food items, specifying quantity and meal times.
    - Barcode scanning functionality for quick food entry.
  3. Nutritional Database:
    - Comprehensive database of food items with accurate nutritional information.
    - Regular updates to the database to include new food items and their nutritional values.
  4. Goal Setting and Tracking:
    - Users can set personalized health goals (weight loss, calorie intake, etc.).
    - Progress tracking with visual representations (graphs, charts) for goals.

## Solution Architecture Diagram:



### Steps to complete the project

#### Step 1:-

1. Open the Zip file and download the zip file. Extract all zip files

#### Step 2 :

1. Open vs code in the left top select open folder. Select extracted file and open .
2. Select the projectname.sol file and copy the code.
3. Open the remix ide platform and create a new file by giving the name of projectname.sol and paste the code which you copied from vs code.
4. Click on solidity compiler and click compile the projectname.sol
5. Deploy the smart contract by clicking on the deploy and run transaction.
6. select injected provider - MetaMask. In environment

7. Click on deploy. Automatically MetaMask will open and give confirmation. You will get a pop up click on ok.

8. In the Deployed contract you can see one address copy the address.

9. Open vs code and search for the connector.js. In contract.js you can paste the address at the bottom of the code. In export const address.

10. Save the code.

### **Step 3:** open file explorer

1. Open the extracted file and click on the folder.

2. Open src, and search for utiles.

3 . You can see the frontend files. Select all the things at the top in the search bar by clicking alt+ A. Search for cmd

4. Open cmd enter commands

```
npm install
```

```
npm bootstrap
```

```
npm start
```

5. It will install all the packages and after completing it will open {LOCALHOST IP ADDRESS} copy the address and open it to chrome so you can see the frontend of your project.