

***Front-end Framework***

**Mock Project**

|  |  |
| --- | --- |
| Document Code | 25e-BM/HR/HDCV/FSOFT |
| Version | 1.1 |
| Effective Date | 7/1/2023 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Record oF changes | | | | | |
| No | Effective Date | Change Description | Reason | Reviewer | Approver |
| 1 | 30/Jan/2023 | Create a new assignment | Create new |  |  |
| 2 | 07/Feb/2023 | Update Fsoft Template | Update |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |
| --- | --- |
|  | **CODE: FEFW\_Angular Project**  **TYPE: Project**  **LOC: N/A**  **DURATION: N/A** |

# Order Food App

Problem Description:

While most "todo" demos provide an excellent cursory glance at a framework's capabilities, they typically don't convey the knowledge & perspective required to actually build *real* applications with it.

Project Overview

"Food Order" is a application allow user can order food. It uses API for all requests, including authentication.

General functionality:

* Authenticate users via JWT (login/signup pages + logout button)
* CRUD Users (sign in, register, )
* CRUD Order ( create order, show infor order )
* CRUD Food ( get list food, search food name, search food tag, get list tags)

Frontend Specs

Routing Guidelines

* Home page (URL: / )
  + List of tags
  + List of foods
* Sign in/Sign up pages (URL: /login, /register )
  + Uses JWT (store the token in localStorage)
* Order page to view info order (URL: /order)
* Checkout page to order food
* Cart page (URL: /cart-page)
  + Show list cart
  + Allow delete food
  + Allow Order Food
* Profile page (URL: /profile)
  + Show basic user info

Styles

Import library Primeng

Backend Specs

API URL:  https://intern-greenwich-be.onrender.com/api

Endpoints:

### Authentication:

1. POST /api/users/login

Example request body:

1. {
2. "email": "jake@jake.jake",
3. "password": "jakejake"
4. }

No authentication required, returns a User

Required fields: email, password

### Registration:

1. POST /api/users/register

Example request body:

{

"name": "TUi la huy",

"email": "bjwww@gmail.com",

"password": "123456",

"confirmPassword": "123456",

"address": "Hẻm 39/28 tổ 20 khu phước hải thị trấn long thành tỉnh đồng nai"

}

No authentication required, returns a User

Required fields: email, name, password, confirmPassword, address

### Get Profile

1. GET /api/user/:id

Authentication required, returns a User that's the current user

### Update Profile

1. PUT /api/user/:id

Example request body:

{

"name": "TUi la huy",

"email": "bjwww@gmail.com",

"password": "123456",

"confirmPassword": "123456",

"address": "Hẻm 39/28 tổ 20 khu phước hải thị trấn long thành tỉnh đồng nai"

}

Authentication required, returns the User

Accepted fields: email, username, password

### List Foods

1. GET /api/foods

Return list food

Query Parameters:

Filter by tag:

1. GET /api/foods/tag/{tag\_name}

Filter by name food:

1. GET /api/foods/search/{food\_name}

### Get Info Order

1. GET /api/track/:id

Param id: id user

Return infor order

### Create Order

1. POST /api/orders/create

Example request body:

{

"items": [

{

"food": {

"\_id": "642bc5f916a230a308ec05dd",

"id": "642bc5f916a230a308ec05dd",

"name": "Pizza Pepperoni",

"cookTime": "10-20",

"price": 10,

"favorite": false,

"origins": [

"italy"

],

"stars": 4.5,

"imageUrl": "assets/food-1.jpg",

"tags": [

"FastFood",

"Pizza",

"Lunch"

]

},

"quantity": 1,

"price": 10

},

{

"food": {

"\_id": "642bc77a16a230a308ec05de",

"id": "642bc77a16a230a308ec05de",

"name": "Meatball",

"cookTime": "10-20",

"price": 20,

"favorite": true,

"origins": [

"persia",

"middle east",

"china"

],

"stars": 4.7,

"imageUrl": "assets/food-2.jpg",

"tags": [

"SlowFood",

"Lunch"

]

},

"quantity": 1,

"price": 20

},

{

"food": {

"\_id": "642bcbb716a230a308ec05df",

"id": "642bcbb716a230a308ec05df",

"name": "Hamburger",

"price": 5,

"cookTime": "10-15",

"favorite": false,

"origins": [

"germany",

"us"

],

"stars": 3.5,

"imageUrl": "assets/food-3.jpg",

"tags": [

"FastFood",

"Hamburger"

]

},

"quantity": 1,

"price": 5

}

],

"totalPrice": 35,

"addressLatLng": {

"lat": 10.8376584,

"lng": 106.8329438

},

"name": "TUi la huy",

"address": "dqeq"

}

Authentication required, will return an Order

JSON Objects returned by API:

### **Other status codes:**

401 for Unauthorized requests, when a request requires authentication but it isn't provided

403 for Forbidden requests, when a request may be valid but the user doesn't have permissions to perform the action

404 for Not found requests, when a resource can't be found to fulfill the request.