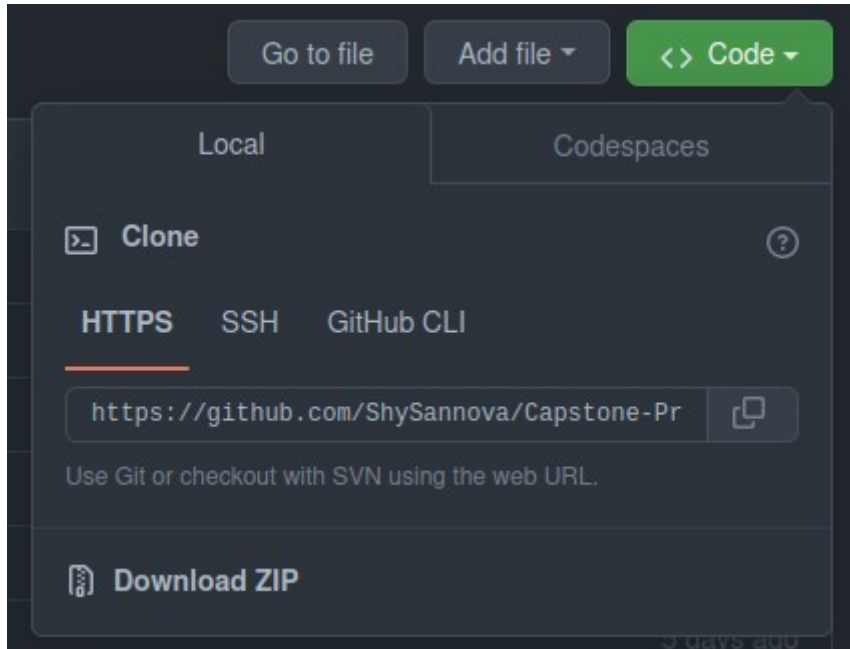


To Get Started

Follow The Steps :

1. Download the **CODE** or **CLONE** it from the repo



2. After Extraction Or Cloning, open the respective folder in vscode editor and **Run** this code in the terminal for :

-- BACKEND (restaurant-api)

1. npm init

After getting the node modules ready, **add a file** named as **.env**

Open and **Add** the following **Variables**

STRIPE_KEY = Your **Stripe Private Key**

CLI_DOMAIN = Your **Frontend URL/home/orders**

URI = Your **Mongodb URI**

Once **above steps** are completed **Run** this code

1. npm start

```
server is up and running on port 8000
Database Connection Successful
```

(Once you **see the above line** in your terminal , then you are ready to move to **Frontend**)

-- FRONTEND (Capstone-Project)

1. yarn install

After getting the node modules ready, **add a file** named as **.env**

VITE_BASE_URL = Your Backend URL

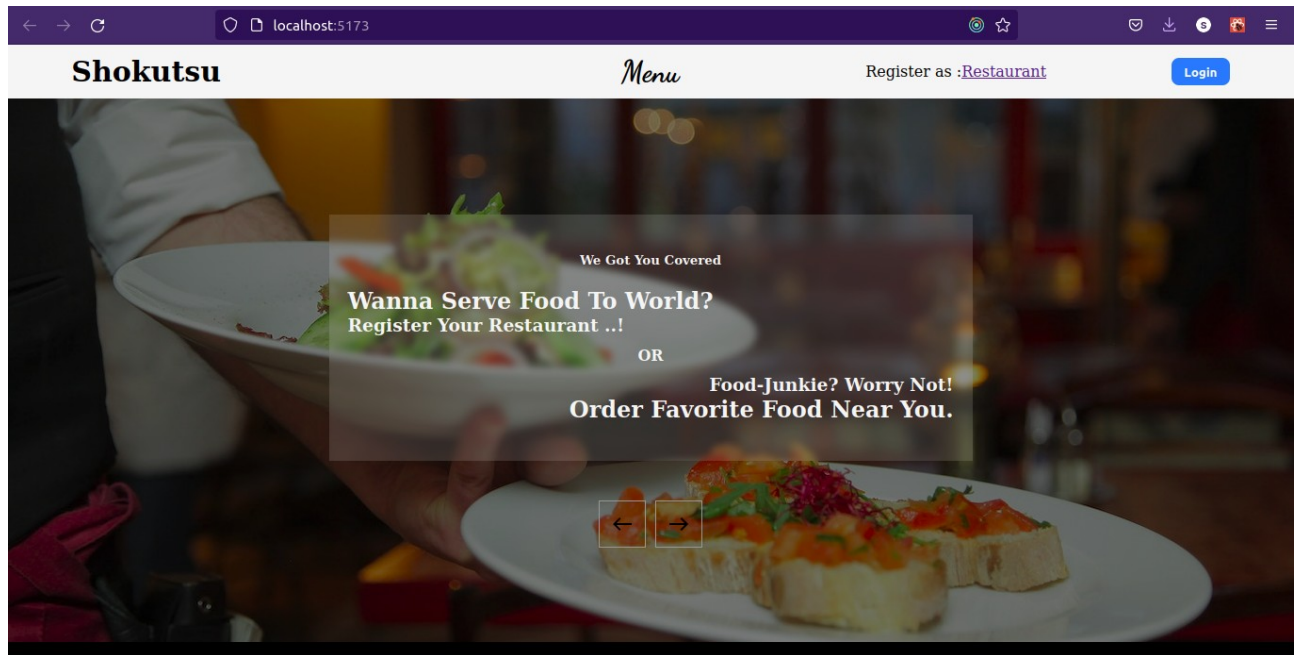
Once **above steps** are completed **Run** this code

1. yarn dev

```
→ Local: http://localhost:5173/  
→ Network: use --host to expose
```

(Once you see **the above line** in your terminal, you can use the **URL** to see your project live)

OPEN BROWSER AND USE THE LOCAL URL



ENJOY YOUR FOOD ORDERING SITE

P.S

You can use demo@gmail.com for **User Login**
Password : Demo123

For **Restaurant Login** please register with **Dummy Info**

ENDPOINTS LIST FOR FURTHER ADJUSTMENTS

User / Customer Endpoints

Request	endpoints	Functionality	Headers	Body	Params
get	/customer/all	Provides all users	Not Required	Not Required	Not Required
	/customer/cart/:id	Provides all Cart items of user	Not Required	Not Required	Required
	/customer/orders	Provides all Orders	Not Required	Not Required	Not Required
	/customer/orders/:id	Provides all Orders of user	Not Required	Not Required	Required
	/customer/:id	Finds user which matches the given id	Not Required	Not Required	Required
	/customer/finduserbyemail	Finds user which matches the given email	Not Required	Not Required	Not Required
post	/customer/signin	Lets you Login as user	Required	Required	Not Required
	/customer/signup	Lets user register	Required	Required	Not Required
	/customer/cart	Create Cart	Required	Required	Not Required
	/customer/orders	Create Orders	Required	Required	Not Required
	/customer/resetpassword/:id	Resets Password of user	Required	Required	Required
patch	/customer/:id	Update user info	Required	Required	Required
delete	/customer/cart/deleteAll/:id	deletes all cart items of user	Not Required	Not Required	Required

Restaurant Endpoints

Request	endpoints	Functionality	Headers	Body	Params
get	/restaurant/all	Provides all restaurants	Not Required	Not Required	Not Required
	/restaurant/orders/:id	Provides all Orders of restaurant	Not Required	Not Required	Required
	/restaurant/:id	Finds restaurant which matches the given id	Not Required	Not Required	Required
post	/restaurant/signup	Lets you register as restaurant	Required	Required	Not Required
	/restaurant/signin	Lets you Login as restaurant	Required	Required	Not Required
	/restaurant/email	Finds restaurant which matches the given email	Required	Required	Not Required
patch	/restaurant/order/:id	Updates order made to restaurant	Required	Required	Required
	/restaurant/:id	Update restaurant info	Required	Required	Required
	/restaurant/resetpassword/:id	Resets Password of restaurant	Required	Required	Required

Product Endpoints

Request	endpoints	Functionality	Headers	Body	Params
get	/product/all	Provides all Dishes/Product	Not Required	Not Required	Not Required
	/product/own/:id	Provides all Dishes/Product of Restaurant	Not Required	Not Required	Required
	/product/:id	Finds Dishes/Product which matches the given id	Not Required	Not Required	Required
post	/product/add	Adds Dish/Product	Required	Required	Not Required
patch	/product/:id	Updates Dish/Product	Required	Required	Required
delete	/product/:id	Deletes Product	Not Required	Not Required	Required

Stripe Checkout Endpoints

Request	endpoints	Functionality	Headers	Body	Params
post	/create-checkout-session	Lets you make Payment	Required	Required	Not Required

P.S : To know what **body data** is used, please refer to **schema.pdf**