

Introduction:

The Vending Machine system code is written on TypeScript. Here are The documentation:
One needs **node js** installed in the Computer to run this code.

After going to the directory **vending-machine**, run

npm install

This will install all the dependencies needed to run the code.

Then going to the directory **vending-machine/js**, run

node main.js

There will come a text

Main Menu (A = Config Mode, B = Sale Mode):

By input **A/B** and pressing **enter** the system will go to particular mode.

Returning To Main Menu:

For coming back to the **Main Menu** from any mode or any stage, Please input **C** on any input requesting field. As for example,

If input is **A**, it will go to the money config mode, if input is **B**, it will go to the Money store inventory mode and if the input is **C**, the system will go to the **Main Menu**.

Money Store Mode (A = Config Money Store, B = Money Store Inventory): C ↵

Main Menu (A = Config Mode, B = Sale Mode):

While coming back to the **Config Menu** from **Sale Menu**, an user will need a secret code to enter in **Config Mode**.

The secret code is **admin123**. For simplicity, the secret code is not configurable. It is hardcoded in the system.

Add Money:

For adding new money indexes in the system, an user must go to the **Add Money** mode via config mode. Here is given a sample input of 50 Taka note:

Type (Coin/Note): note ↵

Composition: PAPER ↵

Color: YELLOW ↵

Obverse: BB ↵

Reverse: TDM ↵

Value: 50 ↵

Count: 5 ↵

After adding successfully, there will be a Success message.
If any of the inputs are invalid, an error message will be shown by the system.

Update Money:

For updating the money index in the system, an user must go to the **Update Money** mode via config mode. Here is given a sample update of 50 Taka note:

Enter Code: PAPER-YELLOW-BB-TDM ↵

Enter Count: 12 ↵

After updating successfully, there will be a Success message.
If any of the inputs are invalid, an error message will be shown by the system.

Remove Money:

For removing money availability from the system, an user must go to the **Remove Money** mode via config mode. Here is given a sample to remove 50 Taka note:

Enter Code: PAPER-YELLOW-BB-TDM ↵

After removing successfully, there will be a Success message.
If any of the inputs are invalid, an error message will be shown by the system.

Add Product:

For adding a new product in the system, an user must go to the **Add Product** mode via config mode. Here is given a sample input of "Potato Chips":

Product Name: Potato Chips ↵

Product Code: 501 ↵

Product Price: 20 ↵

Product Count: 5 ↵

After adding successfully, there will be a Success message.
If any of the inputs are invalid, an error message will be shown by the system.

Update Product:

For updating the money index in the system, an user must go to the **Update Product** mode via config mode. Here is given a sample update of Potato Chips:

Product Code: 501 ↵

Product Count: 12 ↵

Product Price: 25 ↵

After updating successfully, there will be a Success message.

If any of the inputs are invalid, an error message will be shown by the system.

Remove Product:

For removing product availability from the system, an user must go to the **Remove Product** mode via config mode. Here is given a sample to remove of Potato Chips:

Enter Product Code: 501

After removing successfully, there will be a Success message.

If any of the inputs are invalid, an error message will be shown by the system.

Sale Mode:

A sample sale console operations example

Sale Menu (A= Purchase, B=Product Selection): A ↵

Enter Coins/Notes: STEEL-SILVER-JAMUNA-SHAPLA ↵

You have Entered Total Amount of: 5

Sale Menu (A= Purchase, B=Product Selection): A ↵

Enter Coins/Notes: PAPER-PINK-BB-BMM ↵

You have Entered Total Amount of: 15

Sale Menu (A= Purchase, B=Product Selection): B↵

Enter Code: 501 ↵ (consider 501 code is for Potato Chips and it's price is 10 tk.)

Please Collect your "Potato Chips".

Please collect 5 Tk as change.

Thanks for using smart vending!