Norman C Blizard

nblizard@ivytech.edu

Abstract

This program is a storefront application for BTSllc, designed to allow users to browse a catalog of BTS logo items and create orders. The application provides a graphical user interface (GUI) built with Tkinter, offering an intuitive way to view items, select quantities, and submit orders.,

SDEV 140 Project Documentation Rev0-page

Contents

[Overall Function of BTS\_storefront.py 2](#_Toc191228629)

[Key Components and Functionality 3](#_Toc191228630)

[1. Imports: 3](#_Toc191228631)

[2. CatalogItem` Class: 3](#_Toc191228632)

[3. Catalog Management: 3](#_Toc191228633)

[4. `displayCatalog()` Function: 3](#_Toc191228634)

[5. ‘createOrderForm()` Function: 4](#_Toc191228635)

[6. submitOrder(selectedItems, customerInfo)` Function: 4](#_Toc191228636)

[7. `main()` Function: 4](#_Toc191228637)

[8. Image Handling: 4](#_Toc191228638)

[How to Run the Program: 4](#_Toc191228639)

[Program Structure chart: 6](#_Toc191228640)

[Program Steps and Help File 7](#_Toc191228641)

[Introduction: 7](#_Toc191228642)

[System Requirements: 8](#_Toc191228643)

[Installation: 8](#_Toc191228644)

[Program Overview: 8](#_Toc191228645)

[Catalog Display: 8](#_Toc191228646)

[Order Submission: 8](#_Toc191228647)

[Usage Instructions: 9](#_Toc191228648)

[Running the Program: 9](#_Toc191228649)

[Main Window: 9](#_Toc191228650)

[Display Catalog: 10](#_Toc191228651)

[Create Order: 11](#_Toc191228652)

[11](#_Toc191228653)

[Entering Credit Card: 12](#_Toc191228654)

[Submitting Order: 13](#_Toc191228655)

[Error Message: 13](#_Toc191228656)

[Closing Windows: 14](#_Toc191228657)

[Help Documentation: 14](#_Toc191228658)

# Overall Function of BTS\_storefront.py

This Python program creates a simple GUI application using `tkinter` for a "BTS Logo Store." It displays a catalog of items with images and allows the user to create an order form, select items, and submit an order.

In summary, this program provides a basic but functional GUI-based e-commerce application for a "BTS Logo Store," demonstrating fundamental concepts of `tkinter`, image handling, and catalog management.

## Key Components and Functionality

### 1. Imports:

\* `tkinter`: For creating the graphical user interface (GUI).

\* `tkinter.ttk`: For themed widgets in tkinter (more modern look).

\* `tkinter.messagebox`: For displaying pop-up message boxes (e.g., for errors).

\* `tkinter.Scrollbar`: For adding scrollbar functionality to the order form

\* `PIL` (Pillow): For image handling (opening, resizing, and displaying images).

\* `os`: For interacting with the operating system, specifically to read the files in the folder.

### 2. CatalogItem` Class:

\* Defines a class to represent an item in the catalog. Each item has an `id`, `name`, `price`, and `imagePath`.

### 3. Catalog Management:

\* `catalog = {}`: A dictionary to store `CatalogItem` objects. The item ID is the key.

\* `addItemToCatalog(id, name, price, imagePath)`: Creates a `CatalogItem` object and adds it to the `catalog` dictionary.

\* `lookupItem(id)`: Retrieves a `CatalogItem` from the `catalog` based on its `id`. Returns `None` if the item is not found.

### 4. `displayCatalog()` Function:

\* Creates a new top-level window to display the catalog.

\* Iterates through the items in the `catalog`.

\* Displays the item's name and price.

\* Opens the image associated with the item, resizes it, and displays it as a label.

\* Includes error handling for `FileNotFoundError` and other exceptions that might occur during image loading.

\* The catalog is split into two columns for better layout.

### 5. ‘createOrderForm()` Function:

\* Creates a new top-level window for the order form.

\* Creates a scrollable order form in case there are a large number of items.

\* Creates entry fields for customer information (Name, address, phone, email, etc.)

\* Dynamically creates a section for each item in the catalog, including the item name, price, and a `Spinbox` to select the quantity.

\* Stores the selected quantities in the `selectedItems` dictionary.

\* Includes a "Submit Order" button.

### 6. submitOrder(selectedItems, customerInfo)` Function:

\* This function is called when the "Submit Order" button is clicked.

\* Calculates the total price of the order based on the selected quantities and item prices.

\* Includes a 7% sales tax calculation.

\* Extracts customer information.

\* Creates an invoice string with the order summary, customer details, subtotal, sales tax, and total price.

\* Displays the invoice in a new top-level window.

### 7. `main()` Function:

\* Creates the main application window.

\* Sets the window title.

\* Calculates the window size.

\* Dynamically displays images in the "logoimages" folder in a frame. The first image in the folder is displayed larger than the rest. Displays the images in a mosaic.

\* Adds items to the `catalog` using the `addItemToCatalog()` function.

\* Creates two buttons: "Display Catalog" and "Create Order," which call the respective functions when clicked.

\* Starts the `tkinter` main event loop (`root.mainloop()`), which makes the GUI interactive.

### 8. Image Handling:

\* The program uses the PIL (Pillow) library to open, resize, and display images.

\* It expects the images to be in a folder named "storeitems" relative to the script's location, as per the `imagePath` values when adding items to the catalog.

\* The program also expects a folder called "logoimages" relative to the script's location

## How to Run the Program:

You may use VS Code and set up a virtual environment if needed. Here are the instructions:

To activate a virtual environment in VS Code for Python, follow these steps [citation numbers are in brackets]:

1. Open the folder containing the project files in VS Code. Here is a screenshot of the required files

A screenshot of a computer program

AI-generated content may be incorrect.

2. Press Ctrl+Shift+P (Windows/Linux) or Cmd+Shift+P (macOS) to open the Command Palette[3][6].

3. Type "Python: Select Interpreter" and select it from the list[3][6].

4. VS Code will display a list of available Python interpreters. Look for the listed virtual environment, in the project folder[3][6].

5. Select the interpreter for this virtual environment. It will typically be named myvenv or have a path ending with "/bin/python"[3][5].

Once you've selected the interpreter, VS Code will automatically activate the virtual environment when you open a new terminal[5]. You can verify this by opening a new terminal and checking if the virtual environment name appears in the prompt.

If VS Code doesn't detect your virtual environment automatically, you can manually enter the path to the Python executable in your virtual environment when prompted[6].

## Alternative Method

If you prefer using the terminal directly:

1. Open a new terminal in VS Code.

2. Activate the virtual environment using the appropriate command:

- For macOS/Linux: `source .venv/bin/activate`

- For Windows: `.venv\Scripts\activate`[3][5]

After activation, you should see the virtual environment name in your terminal prompt, indicating it's now active for this Python project.

Citations:

[1] https://www.youtube.com/watch?v=hC5rfoIY8nU

[2] https://www.youtube.com/watch?v=yG9kmBQAtW4

[3] https://code.visualstudio.com/docs/python/environments

[4] https://code.visualstudio.com/docs/python/python-tutorial

[5] https://docs.posit.co/ide/server-pro/user/vs-code/guide/python-environments.html

[6] https://www.reddit.com/r/learnpython/comments/1dq5zor/how\_to\_use\_venv\_in\_vscode\_to\_run\_python\_scripts/

[7] https://stackoverflow.com/questions/54106071/how-can-i-set-up-a-virtual-environment-for-python-in-visual-studio-code

[8] <https://python-forum.io/thread-39414.html>

# Program Structure chart:

BTS Storefront Application Structure Chart

Main Program (bts\_storefront.py)

├───tkinter (module)

├───ttk (module)

├───scrolledtext (module)

├───messagebox (module)

├───Scrollbar (module)

├───re (module)

├───datetime (module)

├───os (module)

├───PIL (module)

│ └───Image, ImageTk (modules)

├───help\_document (data - dictionary)

├───display\_help()

│ ├───help\_window (Toplevel)

│ ├───ttk.Panedwindow

│ ├───button\_frame (Frame)

│ ├───text\_frame (Frame)

│ ├───text\_area (ScrolledText)

│ ├───show\_section(heading)

│ │ └───text\_area (ScrolledText)

│ └───ttk.Button (for each heading)

├───CatalogItem (class)

│ └─── \_\_init\_\_(self, id, name, price, imagePath)

├───catalog (data - dictionary)

├───addItemToCatalog(id, name, price, imagePath)

│ └───newItem (CatalogItem instance)

├───lookupItem(id)

│ └───catalog (data - dictionary)

├───displayCatalog()

│ ├───catalogWindow (Toplevel)

│ ├───left\_frame (Frame)

│ ├───right\_frame (Frame)

│ ├───catalog (data - dictionary)

│ ├───Image.open(item.imagePath)

│ ├───ImageTk.PhotoImage(image)

│ └───ttk.Label (for each item)

├───createOrderForm()

│ ├───orderForm (Toplevel)

│ ├───canvas (Canvas)

│ ├───scrollbar (Scrollbar)

│ ├───item\_frame (Frame)

│ ├───customerInfo (data - dictionary)

│ ├───ttk.Label and ttk.Entry (for customer info)

│ ├───selectedItems (data - dictionary)

│ ├───ttk.Spinbox (for item quantities)

│ └───submitOrder(selectedItems, customerInfo)

├───submitOrder(selectedItems, customerInfo)

│ ├───selectedItems (data - dictionary)

│ ├───customerInfo (data - dictionary)

│ ├───lookupItem(itemId)

│ ├───messagebox.showerror()

│ └───totalPrice (calculation)

# Program Steps and Help File

## Introduction:

This program is a storefront application for BTSllc, designed to allow users to browse a catalog of BTS logo items and create orders. The application provides a graphical user interface (GUI) built with Tkinter, offering an intuitive way to view items, select quantities, and submit orders.

## System Requirements:

Python 3.x

Tkinter (usually included with Python installations)

Pillow (PIL) library for image handling

## Installation:

1. Install Python: If you don't have Python installed, download it from the official Python website and follow the installation instructions.

2. Install Pillow: Open a terminal or command prompt and run the following command:

pip install pillow

## Program Overview:

The program consists of the following main components:

Catalog Display: Shows the available items with their names, prices, and images.

Order Form: Allows users to enter their information and select the quantity of each item they wish to purchase.

Order Submission: Processes the selected items and quantities, calculates the total price, including sales tax, and displays an order summary.

# Usage Instructions:

Follow these instructions to use the storefront program effectively.

## Running the Program:

Run the program from the command line:

`python bts\_storefront.py`

### Main Window:

The main window provides buttons to display the catalog and create an order.

A screenshot of a computer

AI-generated content may be incorrect.

Display Catalog:

Click the 'Display Catalog' button to view the available items. Close the window to return to the main menn.

A screenshot of a screen with a price list of items

AI-generated content may be incorrect.

Create Order:

Click the 'Create Order' button to open the order form and enter your details. Enter your name, address, city, state, zip code, phone number and email address in the spaces provided. Use the scroll bar at the right to scroll down the order entry page. To select an item, click on the spinner to the right of the item and select the quantity by clicking on the up or down arrows.

Go to each item and select the quantity desired. At the end of the order entry, click on ‘Submit Order to Mal’s E-commerce (note for this assignment, no real order is transmitted).A screenshot of a computer

AI-generated content may be incorrect.

## Entering Credit Card:

Click the 'Enter Credit Card' button to enter your details. Note that an error message will pop up if the credit card number is invalid. Once submitted, the credit card information can be cleared by pressing the clear button.

A screenshot of a computer

AI-generated content may be incorrect.

Submitting Order:

After filling out the order form, click 'Submit Order' to process your order.

When the order is submitted, a separate confirmation window will open wit the invoice including the name, address, etc and the order summary, including subtotal, Indiana Sales Tax, and Total Price. Note that this order is independent of the credit card processing, but in a real application would be bundled together.

A screenshot of a computer

AI-generated content may be incorrect.

### Error Message:

If you encounter an error message, carefully read the instructions to ensure that you enterred your information correctly

### Closing Windows:

Close individual windows as needed or exit the main window to close the program. Click on the X in the upper right corner to close any window or end the program from the main window.



A screenshot of a computer

AI-generated content may be incorrect.

### Help Documentation:

There is a help document in the program, click Open Help Document to access the help screen. Buttons are available for each topic.A screenshot of a computer

AI-generated content may be incorrect.

