

Title of the Project

# NeighbourShare – A Simple Item Borrowing & Selling System (Python)

## Introduction

NeighbourShare is a small console-based Python application that helps people living in the same hostel, building, or neighbourhood borrow or buy items from each other.

The system allows users to add items, view all listed items, and search for any item by its name.

The idea is based on a real-life situation where people often need items like calculators, books, chargers, umbrellas, or sports equipment for a short time. Instead of buying these things, they can borrow them from nearby people.

## Problem Statement

Students and neighbours frequently face situations where:

They need an item only once or for a short time

They don't know who in the hostel/building owns what

Borrowing becomes difficult without proper communication

People waste money buying rarely-used items

NeighbourShare solves this problem by creating a simple program where users can list items they want to lend or sell, and others can search and contact them easily.

# Objectives

To build a simple console-based application using Python

To help users easily share or sell items inside a small community

To allow users to:

Add items

View all available items

Search items

To store item information using Python lists

To practice basic Python programming concepts

# Technologies Used

Programming Language: Python

Environment: Any Python interpreter or online compiler

Concepts Used:

Lists

Functions

Loops

Conditional statements

User input handling

## System Requirements

### Hardware

PC/Laptop with at least 2GB RAM

Any operating system (Windows / Linux / macOS)

### Software

Python 3.x

Any Python editor or online compiler

## System Design

### Modules

#### 1. Add Item

User enters item name, owner name, contact number, category (borrow/sell), and price (if selling).

#### 2. View All Items

Displays all stored items.

#### 3. Search Item

Searches for an item using its name.

#### 4. Exit

Ends the program.

## Data Storage

The program uses the following lists:

item\_name[]

owner\_name[]

contact\_number[]

category[]

price[]

No external database is used.

## Implementation

### Main Functionalities

Taking user input

Storing item details in lists

Showing all items

Searching using loops

Simple text menu with choices

### Code Structure

main() → menu and user choices

add\_item() → adds item data

view\_items() → displays all items

search\_item() → finds item by name

## Python Code (Used in the Project)

```
item_name = []
owner_name = []
contact_number = []
category = []
price = []

def add_item():
    print("\n--- Add New Item ---")
    name = input("Enter Item Name: ")
    owner = input("Enter Owner Name: ")
    contact = input("Enter Contact Number: ")
    cat = input("Enter Category (borrow/sell): ")

    if cat.lower() == "sell":
        item_price = float(input("Enter Price: "))
    else:
        item_price = 0

    item_name.append(name)
    owner_name.append(owner)
    contact_number.append(contact)
    category.append(cat)
    price.append(item_price)

    print("Item added successfully!")

def view_items():
    print("\n--- All Listed Items ---")

    if len(item_name) == 0:
        print("No items available.")
        return

    for i in range(len(item_name)):
        print("\n-----")
        print("Item Name :", item_name[i])
        print("Owner   :", owner_name[i])
```

```

print("Contact  :", contact_number[i])
print("Category  :", category[i])

if category[i].lower() == "sell":
    print("Price    :", price[i])
else:
    print("Price    : Not Applicable")

def search_item():
    print("\n--- Search Item ---")
    name = input("Enter item name to search: ")

    found = False

    for i in range(len(item_name)):
        if item_name[i].lower() == name.lower():
            print("\nItem Found:")
            print("-----")
            print("Item Name  :", item_name[i])
            print("Owner     :", owner_name[i])
            print("Contact   :", contact_number[i])
            print("Category  :", category[i])

            if category[i].lower() == "sell":
                print("Price    :", price[i])

            found = True

    if not found:
        print("Item not found!")

def main():
    while True:
        print("\n===== NeighbourShare App =====")
        print("1. Add Item")
        print("2. View All Items")
        print("3. Search Item")
        print("4. Exit")

        choice = input("Enter your choice: ")

        if choice == "1":
            add_item()
        elif choice == "2":
            view_items()
        elif choice == "3":
            search_item()
        elif choice == "4":

```

```
    print("Thank you for using the app!")
    break
else:
    print("Invalid choice! Please try again.")

main()
```

# Screenshots

X V Online Python Compil... codechef.com

```
1 item_name = []
2 owner_name = []
3 contact_number = []
4 category = []
5 price = []
6
7 def add_item():
8     print("\n--- Add New Item ---")
9     name = input("Enter Item Name: ")
10    owner = input("Enter Owner Name: "
11                  )
12    contact = input("Enter Contact
13                  Number: ")
14    cat = input("Enter Category
15                  (borrow/sell): ")
16
17    if cat.lower() == "sell":
18        item_price = float(input
19                            ("Enter Price: "))
20    else:
21        item_price = 0
```

Run

Visualize Code

Input

Enter Input here

2:08

61



## Online Python Compiler

codechef.com



### Python roadmap

Python3



```
18
19     item_name.append(name)
20     owner_name.append(owner)
21     contact_number.append(contact)
22     category.append(cat)
23     price.append(item_price)
24
25     print("Item added successfully!")
26
27
28 def view_items():
29     print("\n--- All Listed Items ---")
30
31     if len(item_name) == 0:
32         print("No items available.")
33         return
34
35     for i in range(len(item_name)):
36         print("\n"
37             "-----"
38         )
```

RunVisualize Code

Input

2:11

Online Python Compil... codechef.com

Courses Practice Compete Compiler

Learning Python? Check out our complete Python roadmap

Python3

```
        item_name[i])
38     print("Owner      :",
           owner_name[i])
39     print("Contact   :",
           contact_number[i])
40     print("Category  :",
           category[i])
41
42     if category[i].lower() ==
           "sell":
43         print("Price      :",
               price[i])
44     else:
45         print("Price      : Not
               Applicable")
46
47
48 def search_item():
49     print("\n--- Search Item ---")
50     name = input("Enter item name to
               search: ")
51
```

2:11

60



Online Python Compil...

codechef.com



Courses ▾

Practice

Compete

Compiler

Learning Python? Check out our complete  
Python roadmap



Python3



```
52     found = False
53
54     for i in range(len(item_name)):
55         if item_name[i].lower() ==
56             name.lower():
57                 print("\nItem Found:")
58                 print("-----")
59                 print("Item Name :",
60                     item_name[i])
61                 print("Owner      :",
62                     owner_name[i])
63                 print("Contact    :",
64                     contact_number[i])
65                 print("Category   :",
66                     category[i])
67
68                 if category[i].lower() ==
69                     "sell":
70                     print("Price      :",
71                         price[i])
72
73             else:
74                 print("-----")
75
76             print("Item Not Found")
77
78         else:
79             print("-----")
80
81             print("Item Not Found")
82
83             print("-----")
```

2:11

5G 60



Online Python Compil...

codechef.com



⋮

Learning Python? Check out our complete  
Python roadmap



Python3



```
66         found = True
67
68     if not found:
69         print("Item not found!")
70
71
72 def main():
73     while True:
74         print("\n===== NeighbourShare
75             App =====")
76         print("1. Add Item")
77         print("2. View All Items")
78         print("3. Search Item")
79         print("4. Exit")
80
81         choice = input("Enter your
82             choice: ")
83
84         if choice == "1":
85             add_item()
86         elif choice == "2":
87             view_items()
88         elif choice == "3":
```

Run

Visualize Code

2:11

5G 60



## Online Python Compil...

codechef.com



Learning Python? Check out our complete  
Python roadmap



Python3



```
77     print("3. Search Item")
78     print("4. Exit")
79
80     choice = input("Enter your
81         choice: ")
82
83     if choice == "1":
84         add_item()
85     elif choice == "2":
86         view_items()
87     elif choice == "3":
88         search_item()
89     elif choice == "4":
90         print("Thank you for using
91             the app!")
92         break
93     else:
94         print("Invalid choice!
95             Please try again.")
```

# Modules (Top-Down Design)

```
Main Program
└── Item Module
    ├── Add Item
    ├── Display Items
    └── Search Items
└── Exit Module
```

(Converted to match Python — excluded Java-only modules like File Handling, Borrow History, User Login, etc.)

# Algorithms

Algorithm: Add Item

1. Ask user for item details
2. Store entered data in lists
3. Display confirmation message

Algorithm: Search Item

1. Accept search keyword
2. Loop through item list

3. Display matching item(s)

## References

Python documentation

Classroom notes

Online Python problem-solving resources

## Conclusion

NeighbourShare demonstrates a simple item-sharing system built using Python. The project meets its goals:

Helps community members share items

Easy menu-driven console interface

Uses basic Python concepts

Solves a real-life problem for hostels and neighbourhoods

Possible improvements include:

Adding login system

Using dictionaries instead of multiple lists

Adding file storage

Creating a web or mobile interface