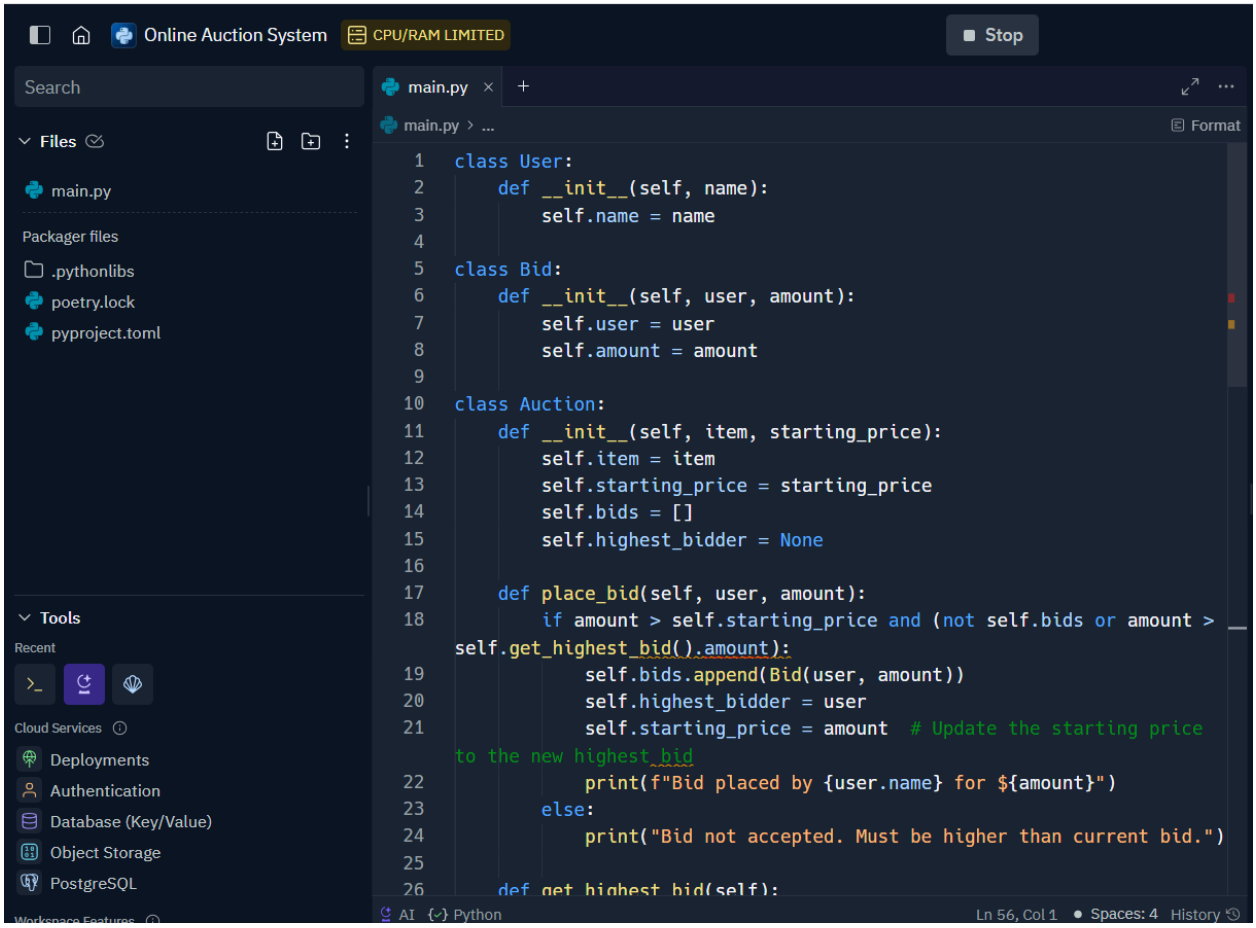
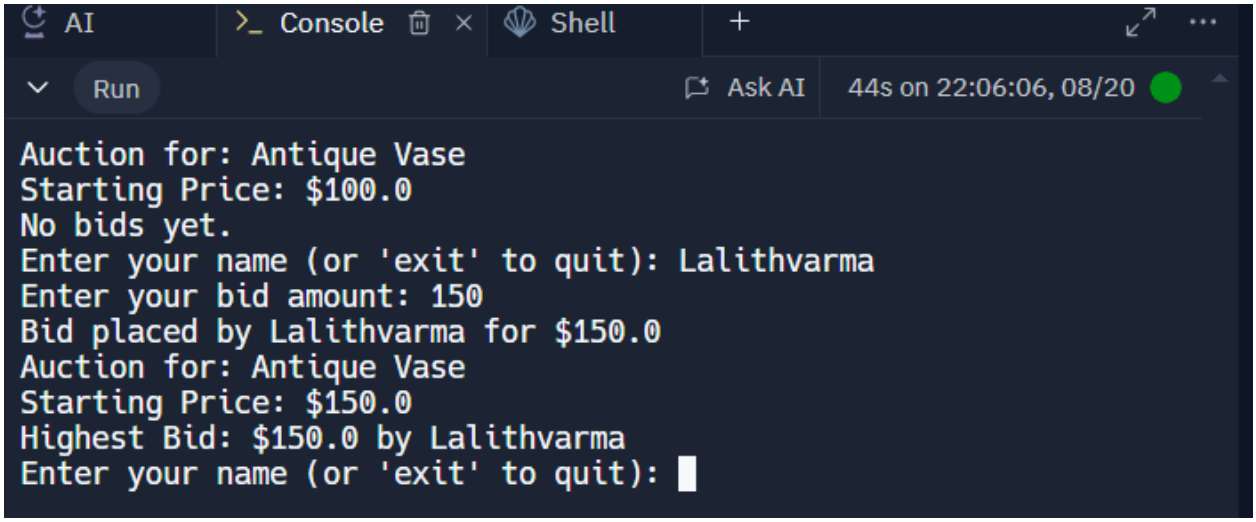
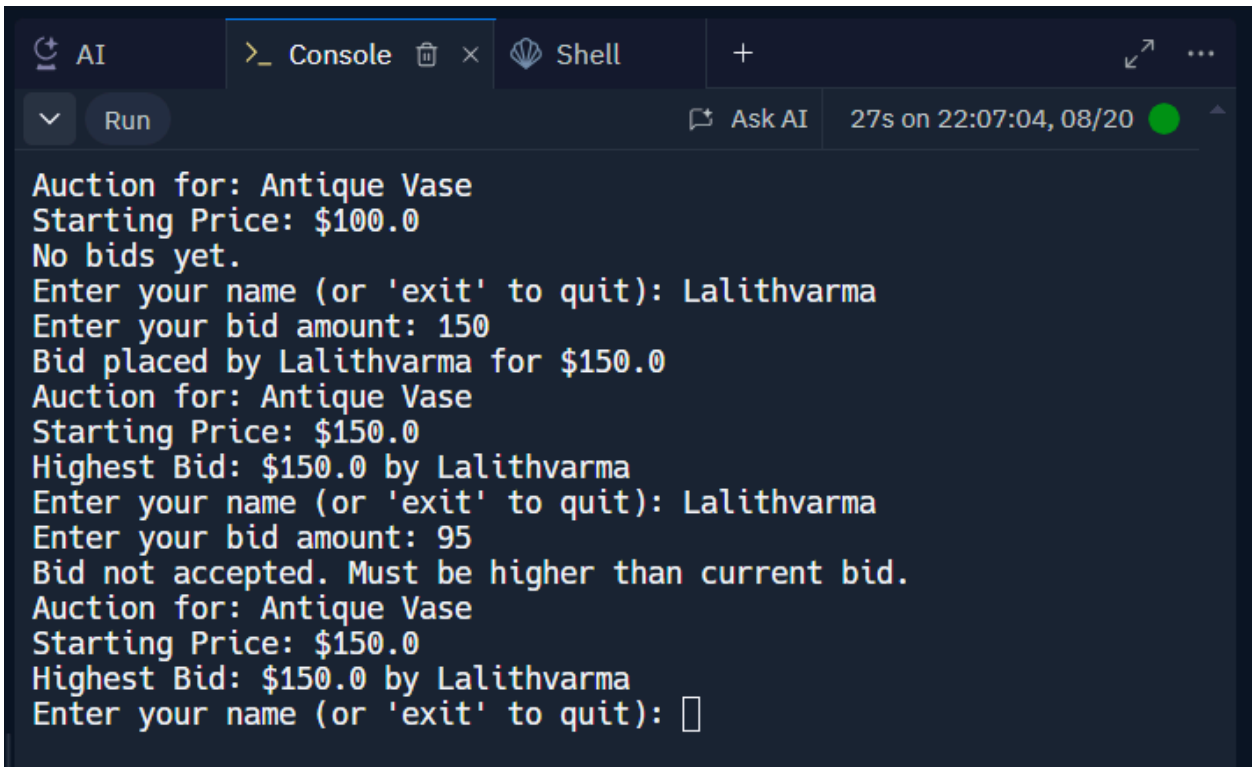
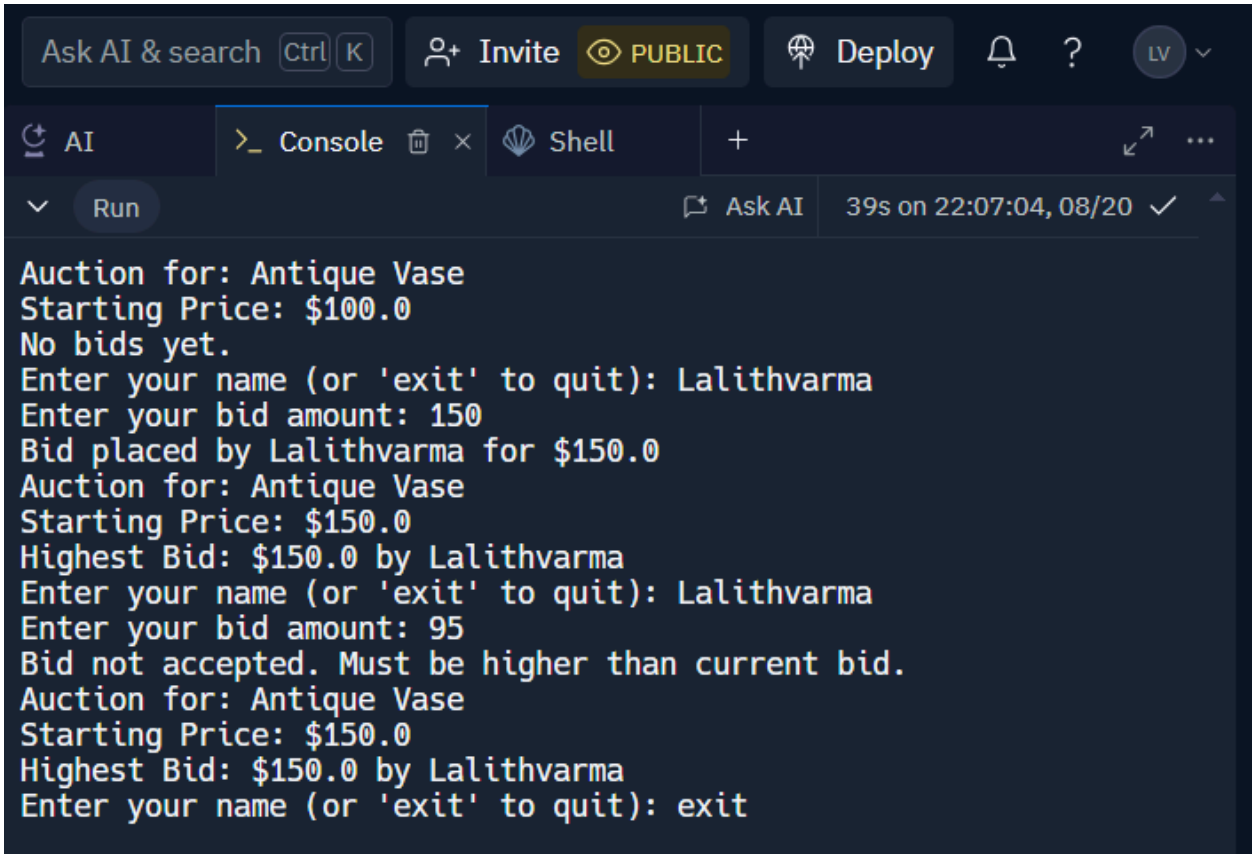
Online Auction System

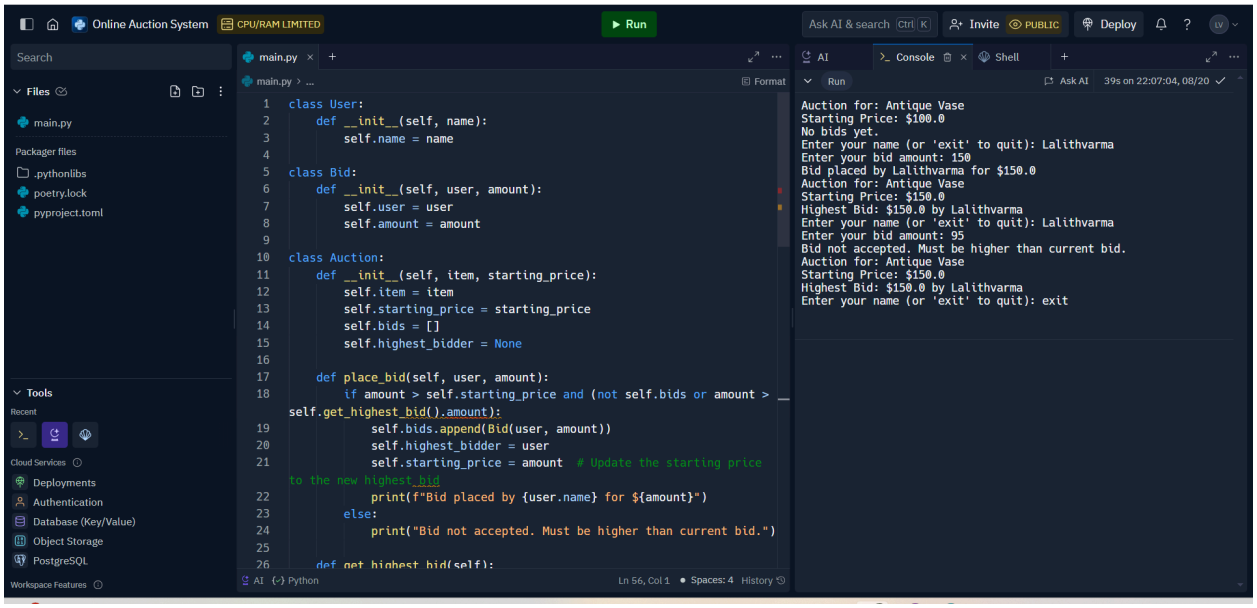
Output:-




AI
Console
Shell
Run
Ask AI
44s on 22:06:06, 08/20
Auction for: Antique Vase
Starting Price: $100.0
No bids yet.
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 150
Bid placed by Lalithvarma for $150.0
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit):


AI
Console
Shell
Run
Ask AI
27s on 22:07:04, 08/20
Auction for: Antique Vase
Starting Price: $100.0
No bids yet.
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 150
Bid placed by Lalithvarma for $150.0
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 95
Bid not accepted. Must be higher than current bid.
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit):


Ask AI & search Ctrl K
Invite PUBLIC
Deploy
LV
AI
Console
Shell
Run
Ask AI
39s on 22:07:04, 08/20
Auction for: Antique Vase
Starting Price: $100.0
No bids yet.
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 150
Bid placed by Lalithvarma for $150.0
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 95
Bid not accepted. Must be higher than current bid.
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit): exit


Online Auction System
CPU/RAM LIMITED
Run
Search
Files
main.py
Packager files
.pythonlibs
poetry.lock
pyproject.toml
Tools
Recent
Cloud Services
Deployments
Authentication
Database (Key/Value)
Object Storage
PostgreSQL
Workspace Features
class User:
def __init__(self, name):
self.name = name
class Bid:
def __init__(self, user, amount):
self.user = user
self.amount = amount
class Auction:
def __init__(self, item, starting_price):
self.item = item
self.starting_price = starting_price
self.bids = []
self.highest_bidder = None
def place_bid(self, user, amount):
if amount > self.starting_price and (not self.bids or amount > self.get_highest_bid().amount):
self.bids.append(Bid(user, amount))
self.highest_bidder = user
self.starting_price = amount # Update the starting price to the new highest bid
print(f"Bid placed by {user.name} for ${amount}")
else:
print("Bid not accepted. Must be higher than current bid.")
def get_highest_bid(self):
Auction for: Antique Vase
Starting Price: $100.0
No bids yet.
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 150
Bid placed by Lalithvarma for $150.0
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit): Lalithvarma
Enter your bid amount: 95
Bid not accepted. Must be higher than current bid.
Auction for: Antique Vase
Starting Price: $150.0
Highest Bid: $150.0 by Lalithvarma
Enter your name (or 'exit' to quit): exit