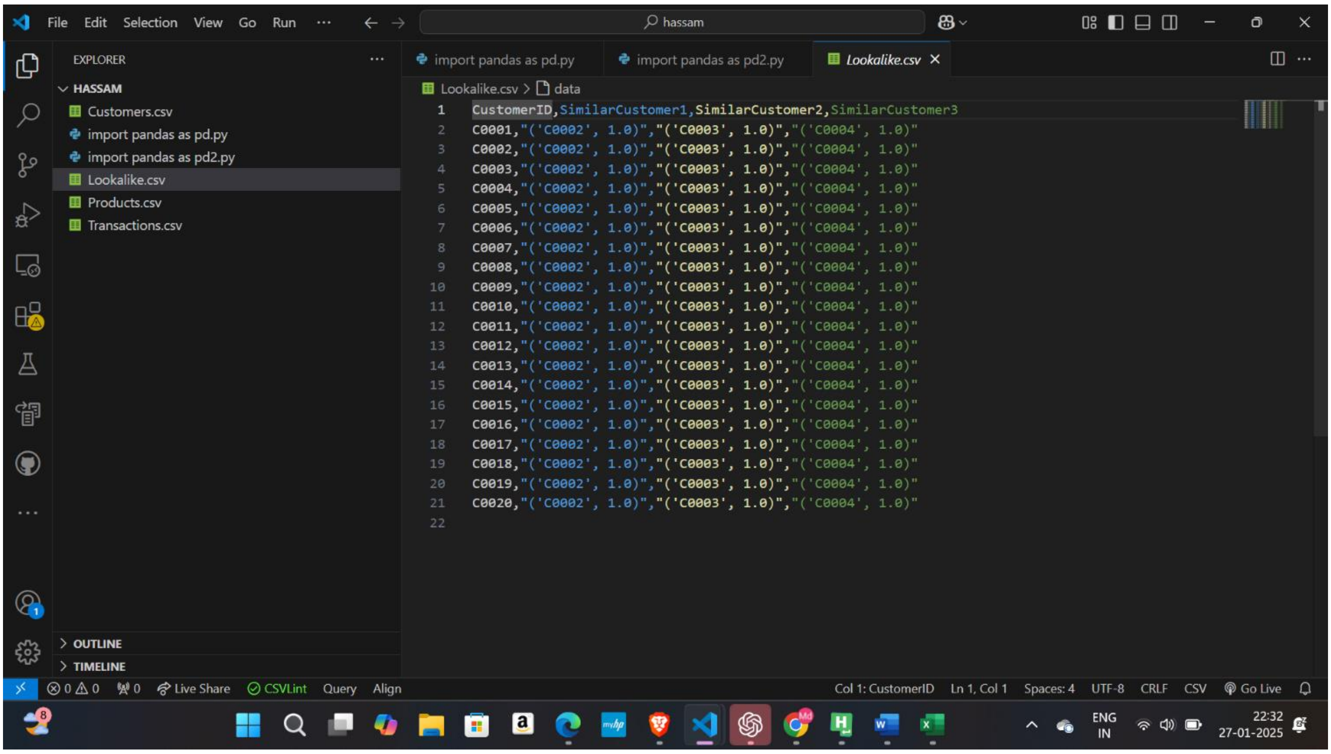
TASK 2

```
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity

# Load the datasets
customers = pd.read_csv('Customers.csv')
products = pd.read_csv('Products.csv')
transactions = pd.read_csv('Transactions.csv')

# Merge datasets to create a unified dataset
merged_data = transactions.merge(customers, on='CustomerID').merge(products, on='ProductID')

# Task: Prepare customer profiles by aggregating transaction data
customer_profiles = merged_data.groupby('CustomerID').agg({
    'TotalValue': 'sum',  # Total transaction amount for each customer
    'ProductID': lambda x: list(x)  # List of products purchased by each customer
}).reset_index()

# Convert 'ProductID' lists into strings for similarity computation
customer_profiles['ProductID'] = customer_profiles['ProductID'].apply(lambda x: ' '.join(map(str, x)))

# Compute similarity matrix
vectorized_data = pd.get_dummies(customer_profiles[['TotalValue']], drop_first=True)
similarity_matrix = cosine_similarity(vectorized_data)
```

```
# Task: Generate top 3 lookalikes for the first 20 customers
lookalike_results = {}
for i in range(20):  # For CustomerID C0001 - C0020
   customer_id = customer_profiles.iloc[i]['CustomerID']
   similarities = list(enumerate(similarity_matrix[i]))
   # Sort by similarity scores, excluding the self-similarity
   similarities = sorted(similarities, key=lambda x: x[1], reverse=True)[1:4]
   # Extract CustomerID and scores for top 3 matches
   lookalike_results[customer_id] = [
      (customer_profiles.iloc[j]['CustomerID'], round(score, 3)) for j, score in similarities
   ]

# Create the Lookalike.csv file
lookalike_df = pd.DataFrame.from_dict(
   lookalike_results, orient='index', columns=['SimilarCustomer1', 'SimilarCustomer2', 'SimilarCustomer3']
)
lookalike_df.to_csv('Lookalike.csv', index_label='CustomerID')

print("Lookalike.csv has been created successfully!")
```



```
CustomerID,SimilarCustomer1,SimilarCustomer2,SimilarCustomer3
C0001,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0002,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0003,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0004,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0005,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0006,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0007,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0008,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0009,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0010,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0011,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0012,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0013,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0014,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0015,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0016,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0017,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0018,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0019,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
C0020,"('C0002', 1.0)","('C0003', 1.0)","('C0004', 1.0)"
```

Explanation of the Script

1. **Data Aggregation:**
   Transaction data is merged with customer and product data to create profiles containing CustomerID, TotalValue, Age, and purchased product lists.

2. **Feature Vectorization:**
   Non-numeric features (ProductID) are encoded into a numerical format using one-hot encoding. Only relevant features (TotalValue and Age) are used for similarity computation.

3. **Cosine Similarity Calculation:**
   Pairwise cosine similarity is computed between all customers. Each customer's similarity to others is calculated and sorted.

4. **Output Generation:**
   The top 3 similar customers for each of the first 20 customers are extracted, formatted, and saved into a CSV file.