Edit Code

Jump to bottom

Update laptop_price_calculator_dev_10.1.E.py #13

Open ISAACLINDROOS wants to merge 1 commit into `main` from `development`

Conversation 0 Commits 1 Checks 0 Files changed 1

Changes from all commits File filter Conversations Jump to

53 .../Laptop Price Calculator 10.1.E (Development)/build/laptop_price_calculator_dev_10.1.E.py

```
49      return ASSETS_PATH / Path(path)
50
51
52  - # Pandas Dataframe Configuration: (Set Columns)
53    columns = ['brand', 'model', 'processor_brand',
      'processor_name', 'processor_gnrtn', 'ram_gb', 'ram_type',
      'ssd', 'hdd', 'os', 'os_bit', 'graphic_card_gb',
54              'weight', 'display-size', 'warranty',
      'touchscreen', 'msoffice', 'latest_price', 'old_price',
      'discount', 'star_rating', 'ratings', 'reviews']
55    df = pd.read_csv(ASSETS_PATH /
      "laptops_dataset_clean_refined.csv",
56                     usecols=['os', 'ssd', 'ram_gb',
      'display_size', 'touchscreen', 'brand'])
57
58  - # Pandas Dataframe refinement: (Removing all Rows with NaN
      Values within the Main Dataframe & Updateing the Index axis
      Accordingly)
59    df = df.dropna()
60    print("Main Dataframe:\n", df)
61    df = df.dropna(axis=0)
62    df = df.dropna().reset_index(drop=True)
63    print(df)
64
65
66  - # Function: "Display settings (Pandas dataframe)"
```

```
49      return ASSETS_PATH / Path(path)
50
51
52  + # Pandas Main Dataframe Configuration: (Set Columns)
53    columns = ['brand', 'model', 'processor_brand',
      'processor_name', 'processor_gnrtn', 'ram_gb', 'ram_type',
      'ssd', 'hdd', 'os', 'os_bit', 'graphic_card_gb',
54              'weight', 'display-size', 'warranty',
      'touchscreen', 'msoffice', 'latest_price', 'old_price',
      'discount', 'star_rating', 'ratings', 'reviews']
55    df = pd.read_csv(ASSETS_PATH /
      "laptops_dataset_clean_refined.csv",
56                     usecols=['os', 'ssd', 'ram_gb',
      'display_size', 'touchscreen', 'brand'])
57
58  +
59  + # Pandas Main Dataframe refinement: (Removing all Rows with
      NaN Values within the Main Dataframe & Updateing the Index
      axis Accordingly)
60    df = df.dropna()
61    print("Main Dataframe:\n", df)
62    df = df.dropna(axis=0)
63    df = df.dropna().reset_index(drop=True)
64    print(df)
65
66
67  + # Pandas Results Dataframe Configuration: (Set Columns)
68  + columns = ['brand', 'model', 'processor_brand',
      'processor_name', 'processor_gnrtn', 'ram_gb', 'ram_type',
      'ssd', 'hdd', 'os', 'os_bit', 'graphic_card_gb',
69  +           'weight', 'display-size', 'warranty',
      'touchscreen', 'msoffice', 'latest_price', 'old_price',
      'discount', 'star_rating', 'ratings', 'reviews']
70  + df_results = pd.read_csv(ASSETS_PATH /
      "laptops_dataset_clean_refined.csv",
71  +                 usecols=['os', 'ssd', 'ram_gb',
      'display_size', 'touchscreen', 'brand'])
72  +
73  + # Pandas Results Dataframe refinement: (Removing all Rows
      with NaN Values within the Main Dataframe & Updateing the
      Index axis Accordingly)
74  + df_results = df.dropna()
75  + df_results = df.dropna(axis=0)
76  + df_results = df.dropna().reset_index(drop=True)
```

```
                                                                                +
                                                                                +
                                                                                + # Function: "Display settings (ALL Pandas dataframes)"
def set_pandas_display_options() -> None:                                       def set_pandas_display_options() -> None:
    display = pd.options.display                                                    display = pd.options.display
    display.max_columns = 1000                                                      display.max_columns = 1000
    display.max_rows = 1000                                                         display.max_rows = 1000
    display.max_colwidth = 199                                                      display.max_colwidth = 199
    display.width = 1000                                                            display.width = 1000
-                                                                               
set_pandas_display_options()                                                    set_pandas_display_options()
                                                                                +
                                                                                +
# Individual dataframe Configuration:                                           # Individual dataframe Configuration:
os_windows_df = df.os == "Windows"                                              os_windows_df = df.os == "Windows"
os_mac_df = df.os == "Mac"                                                      os_mac_df = df.os == "Mac"
def user_select_display_size_13_3():                                            def user_select_display_size_13_3():
-     df.update(display_size_13_3_df, overwrite=True)                           +     df.update(display_size_13_3_df, overwrite=False)
      print(df)                                                                       print(df)
def user_select_display_size_14():                                              def user_select_display_size_14():
-     df.update(display_size_14_df, overwrite=True)                             +     df.update(display_size_14_df, overwrite=False)
      print(df)                                                                       print(df)
def user_select_display_size_15_6():                                            def user_select_display_size_15_6():
-     df.update(display_size_15_6_df, overwrite=True)                           +     df.update(display_size_15_6_df, overwrite=False)
      print(df)                                                                       print(df)
def user_select_display_size_16():                                              def user_select_display_size_16():
-     df.update(display_size_16_df, overwrite=True)                             +     df.update(display_size_16_df, overwrite=False)
      print(df)                                                                       print(df)
      print(df)                                                                       print(df)
- def results_dataframe():
-     # Pandas Dataframe Configuration: (Set Columns)
-     columns = ['brand', 'model', 'processor_brand',
  'processor_name', 'processor_gnrtn', 'ram_gb', 'ram_type',
  'ssd', 'hdd', 'os', 'os_bit', 'graphic_card_gb',
-                'weight', 'display-size', 'warranty',
  'touchscreen', 'msoffice', 'latest_price', 'old_price',
  'discount', 'star_rating', 'ratings', 'reviews']
-     df = pd.read_csv(ASSETS_PATH /
  "laptops_dataset_clean_refined.csv",
-                      usecols=['os', 'ssd', 'ram_gb',
  'display_size', 'touchscreen', 'brand'])
-     df = df.dropna()
-     print("Main Dataframe:\n", df)
-     df = df.dropna(axis=0)
-     df = df.dropna().reset_index(drop=True)
-     print(df)
-
-
# Datalabel readout Configuration:                                              # Datalabel readout Configuration:
- datalabel = results_dataframe                                                 + datalabel = df_results
```

```
# Create a Results GUI class to connect the Results GUI and the Main Window's GUI:

ResultsWindow = ResultsWindowGUI()

-

# Function "Refresh Window":
def refresh():
    window.update()

-

# Function "Quit Window + Confirmation":
def confirm():
    answer = askyesno(title='Exit Confirmation',
                      message='Are you sure that you want to quit?', icon='warning')
    if answer:
        closewindow()

-

# Function "Generation + Confirmation":
def confirmgen():
    answer = askyesno(title='Confirm Price Generation',
                      message='Are you sure that you want to Generate results?')
    if answer:

        ResultsWindow.showresultswindow()
```

```
# Create a Results GUI class to connect the Results GUI and the Main Window's GUI:

ResultsWindow = ResultsWindowGUI()

# Function "Refresh Window":
def refresh():
    window.update()

+

# Function "Quit Window + Confirmation":
def confirm():
    answer = askyesno(title='Exit Confirmation',
                      message='Are you sure that you want to quit?', icon='warning')
    if answer:
        closewindow()

+

# Function "Generation + Confirmation":
def confirmgen():
    answer = askyesno(title='Confirm Price Generation',
                      message='Are you sure that you want to Generate results?')
    if answer:
+       df['count'] = df[['os', 'ssd', 'ram_gb', 'touchscreen', 'brand']].sum(axis=1)
+       rslt_df = df[df['count'] > 3]
+       print(rslt_df)
        ResultsWindow.showresultswindow()
```