

# รายละเอียดโปรแกรมจำลองการประมาณการลดความต้องการใช้ไฟฟ้า

โปรแกรมจำลองการประมาณการลดความต้องการใช้ไฟฟ้าถูกพัฒนาเพื่อใช้ในการสาธิตการประมาณการลดความต้องการใช้ไฟฟ้าเพื่อให้ผู้ร่วมสัมมนาได้เข้าใจวิธีการมากขึ้น โดยโปรแกรมหุ่นจำลองถูกพัฒนาขึ้นมาให้เป็นลักษณะเว็บไซต์ เมื่อสั่งทำงานโปรแกรมจะทำงานเป็น Web service ในเครื่องคอมพิวเตอร์ โดยจะปรากฏเป็นหน้าจอเทอร์มินอลขึ้นมาพร้อมทั้งบอก URL เว็บไซต์ การเข้าถึงการจำลองการประมาณการจะเข้าผ่านทางเบราว์เซอร์โดยใช้ URL ที่ปรากฏข้างต้น

```
C:\WINDOWS\system32\cmd.exe

C:\elec_sys>C:\Users\Laika\anaconda3\envs\elec_sys\python demo.py
Dash is running on http://127.0.0.1:8050/

* Serving Flask app 'demo'
* Debug mode: on
```

รูปที่ 1 แสดงหน้าจอ Terminal หลังจากให้โปรแกรมทำงานแล้ว

โปรแกรมถูกพัฒนาด้วยภาษาไพทอนซึ่งข้อดีคือการทำงานจะสามารถเปิดอ่านและเข้าใจได้โดยผู้ที่มีความรู้ด้านการเขียนภาษาไพทอน ด้านล่างจะเป็นการอธิบายการทำงานของโค้ดไพทอนในส่วนที่สำคัญ

```
style.css x demo.py x announcement.py x auctio
1 from dash import Dash, html, dcc
2 import dash
3
4 app = Dash(__name__, use_pages=True)
5
6 app.layout = html.Div([
7     dash.page_container
8 ])
9
10 if __name__ == '__main__':
11     app.run_server(debug=True)
```

รูปที่ 2 แสดงส่วนหลักของโปรแกรมที่เขียนด้วยภาษาไพทอน เป็นส่วนที่ทำให้ Web Service ทำงาน

```

17 # Create the layout of the first page
18 layout = html.Div([
19     html.Div(dcc.Link('Next', href='/auction', className='flow_link'),
20               style={'padding-left': '80%', 'padding-top': '10px'}),
21     # Display the logo in the center of the page
22     html.Img(src='data:image/png;base64,{}'.format(logo_b64), style={'display': 'block', 'margin': 'auto'},
23              width=200, height=200),
24     # Add some text below the logo
25     html.Div([html.H1('ประกาศจากการไฟฟ้า', style={'text-align': 'center', 'color': '#FF3333'}),
26               html.P('ขณะนี้การไฟฟ้าเริ่มการประมูลการใช้ไฟฟ้าประจำเดือนกันยายนแล้ว โดยการไฟฟ้า มีความต้องการลดไฟฟ้าลงอยู่ที่ {:.0f} กิโลวัตต์
27               style={'font-size': '20px'})),
28               style={'padding-left': '20%', 'padding-right': '20%', 'padding-top': '10px'})
29 ]

```

รูปที่ 3 แสดงส่วนของโปรแกรมในหน้าประกาศ ข้อความที่เขียนไว้ในบรรทัดที่ 26 จะถูกนำไปแสดงในข้อความประกาศ

```

23 html.Div([html.H1('การประมูลความต้องการไฟฟ้า'),
24           dcc.Dropdown(id='bidder-dropdown',
25                        options=[{'label': 'Factory {}'.format(i), 'value': i} for i in range(1, 11)],
26                        value='', style={'padding-right': '50%'}),
27           html.H2('นำเสนอการประมูล'),
28           html.Div([html.Label('ปริมาณที่ต้องการลด:'),
29                     html.Br(),
30                     html.Label('ราคาต่อหน่วยที่ต้องการเสนอ:'),
31                     html.Div([dcc.Input(type='number', id='reduced_load'),
32                               html.Label('kW'),
33                               dcc.Input(type='hidden', id='reduced_load2'),
34                               html.Br(),
35                               dcc.Input(type='number', id='reduced_price'),
36                               html.Label('บาท'),
37                               dcc.Input(type='hidden', id='reduced_price2')], style={'width': '40%', 'float': 'left'}),
38                     html.Div([html.P(id='summary', children='กรณจะระบุปริมาณที่ต้องการลดและราคาต่อหน่วยเพื่อเสนอประมูล', style={'width': '40%'}),
39                               html.Button('Submit', id='submit_button', style={'font-size': '20px'})], style={'padding-left': '10%', 'padding-right': '10%'}),
40           dcc.Graph(id='reference')

```

รูปที่ 4 แสดงส่วนของโปรแกรมในหน้าประมูล ในภาพจะเป็นการสร้างแบบฟอร์มสำหรับยื่นคำร้องการประมูล

```

86 def update_graph(bidder):
87     if type(bidder) == str or bidder is None:
88         data = [ '{:02d}::{:02d}'.format(h, m * 15) for h in range(24) for m in range(4)]
89         fig = go.Figure(go.Scatter(x=data,
90                                   y=[0 for _ in range(len(data))]))
91         fig.update_layout(xaxis_title='', yaxis_title='kW')
92         fig.update_layout(yaxis_range=[0, 5000])
93         fig.add_vline(x=config['Starting time'], line_dash="dash")
94         fig.add_vline(x=config['Ending time'], line_dash="dash")
95     else:
96         data = df_ref[df_ref['TIME'] == f'R{bidder}'].T.iloc[1:, 0]
97         data = data.reset_index()
98
99         v_max = data[data.columns[1]].max()
100
101         fig = px.line(data, x='index', y=data.columns[1])
102         fig.add_hline(y=v_max, line_dash="dash")
103         fig.add_vline(x=config['Starting time'], line_dash="dash")
104         fig.add_vline(x=config['Ending time'], line_dash="dash")
105         fig.update_layout(xaxis_title='', yaxis_title='kW')

```

รูปที่ 5 แสดงส่วนของโปรแกรมในหน้าประมูล แสดงการทำงานเมื่อมีการเลือกบริษัทยื่นประมูลและดึงข้อมูล Load Profile Reference มาแสดงเป็นกราฟ

```

1 from dash import html, dcc, callback
2 import dash
3 from dash.dependencies import Output, Input, State
4 import pickle as pkl
5 import glob
6 import pandas as pd
7 import plotly.express as px
8 import display_util as util
9
10 dash.register_page(__name__)
11
12 HEADERS = {'index': 'ลำดับที่', 'reduced_wattage': 'เสนอจะลด (kW)', 'price_per_watt': 'บาท/kW'}
13 ALIGNMENT = {'index': 'left', 'reduced_wattage': 'right', 'price_per_watt': 'right'}
14 FORMATS = {'reduced_wattage': 'num', 'price_per_watt': 'num'}
15

```

รูปที่ 6 แสดงส่วนของโปรแกรมในหน้าผลลัพธ์การประมูล แสดงการกำหนดชื่อของหัวตารางผลลัพธ์

```

38
39 def load_layout():
40     df = read_data()
41
42     df_temp = df.copy()
43     df_temp['cumsum'] = df_temp['reduced_wattage'].cumsum()
44     df_temp = pd.concat([df_temp, df_temp]).sort_values('index')
45     df_temp['cumsum'] = df_temp['cumsum'].shift(1).fillna(0)
46     df = util.format_table(df, FORMATS)
47
48     fig = px.line(df_temp, x='cumsum', y='price_per_watt')
49     fig.update_layout(xaxis_title='Accumulated Negative Power(kW)',
50                      yaxis_title='Price (THB/kW)')
51     fig.add_vline(x=config['Reduced demand'], line_dash="dash")
52
53     return html.Div([
54         html.Div(
55             [dcc.Link('Back', href='/auction', className='flow_link'),
56              dcc.Link('Next', href='/bidder-selection', className='flow_link', style={'padding-left': '80%'}),
57              style={'padding-top': '10px'}],
58             html.Div([html.H1(children='ผลการประมูลความต้องการไฟฟ้า'),
59                      html.Div(util.generate_table(df, HEADERS, ALIGNMENT), style={'float': 'left'}),
60                      dcc.Graph(id='cumsum', figure=fig, style={'float': 'right', 'width': '1100px'}),
61                      style={'padding-left': '10%', 'padding-right': '10%'}])
62

```

รูปที่ 7 แสดงส่วนของโปรแกรมในหน้าผลลัพธ์การประมูล แสดงการประมวลผลข้อมูลและวิธีการจัดเรียงหน้า  
จอ

```

1 from dash import html, dcc
2 import dash
3 import glob
4 import pickle as pkl
5 import pandas as pd
6 import numpy as np
7 import display_util as util
8
9 dash.register_page(__name__)
10
11 HEADERS = {'selected': 'รุ่นเสนอ', 'bidder': 'รายที่', 'reduced_wattage': 'Offered reduction (kW)', 'price_per_watt': 'Offered Price (Baht/kW)'}
12 ALIGNMENT = {'selected': 'center', 'bidder': 'center', 'reduced_wattage': 'right', 'price_per_watt': 'right'}
13 FORMATS = {'reduced_wattage': 'num', 'price_per_watt': 'num'}
14

```

รูปที่ 8 แสดงส่วนของโปรแกรมในหน้าผู้ชนะประมูล แสดงการกำหนดชื่อของหัวตารางผู้ชนะประมูล

```
20 def read_data():
21     files = glob.glob('./auctions/*.pkl')
22     data = []
23     for file in files:
24         with open(file, 'rb') as f:
25             row = pickle.load(f)
26             data.append(row)
27
28     df = pd.DataFrame(data, columns=['bidder', 'reduced_wattage', 'price_per_watt'])
29     df = df.sort_values(['price_per_watt', 'reduced_wattage'], ascending=[True, False]).reset_index(
30         drop=True).reset_index()
31     df['index'] = df['index'] + 1
32     df['cumsum'] = df['reduced_wattage'].cumsum()
33
34     df['selected'] = df['cumsum'] <= config['Reduced demand']
35
36     if df[df['cumsum'] == config['Reduced demand']].shape[0] == 0:
37         for i, row in df.iterrows():
38             if row['selected'] == False:
39                 break
40
41     remainder = df.loc[i, 'cumsum'] - config['Reduced demand']
42     remainder_row = row.copy()
43     remainder_row['index'] = remainder_row['index'] - 0.1
44     remainder_row['reduced_wattage'] = remainder_row['reduced_wattage'] - remainder
45     remainder_row['selected'] = True
46
47     row['reduced_wattage'] = remainder
48     row['cumsum'] = row['cumsum'] - remainder
49     row['selected'] = False
50
51     df = df[df['index'] != row['index']]
52     df.loc[df.shape[0] + 1] = remainder_row
53     df.loc[df.shape[0] + 1] = row
54     df = df.sort_values('index')
55     df.to_csv('./auctions/selected.csv', index=False)
56
57     return df
58
```

รูปที่ 9 แสดงส่วนของโปรแกรมในหน้าผู้ชนะประมูล แสดงขั้นตอนการประมวลผลเพื่อแสดงผลลัพธ์ผู้ชนะประมูล

```

32 def read_data():
33     data = pd.read_excel('./reference/reference_actual.xlsx', sheet_name=None)
34     df_ref = data['Load profiles_References']
35     df_ref = transform_table(df_ref)
36
37     df_act = data['Load profiles_Actual']
38     df_act = transform_table(df_act)
39     df_act['diff'] = df_ref['peak'] - df_act['peak']
40     df_act['diff'] = np.where(df_act['diff'] > 0, df_act['diff'], 0)
41     df_act['TIME'] = df_act['TIME'].str.replace('R', '').astype(int)
42
43     df = pd.read_csv('./auctions/selected.csv')
44     df = df.merge(df_act, left_on='bidder', right_on='TIME', how='left')
45     df['reduced_percentage'] = df['diff'] / df['reduced_wattage']
46     df['level'] = np.where(df['reduced_percentage'] < 0.5, 0,
47                             np.where(df['reduced_percentage'] < 0.75, 0.5,
48                                     np.where(df['reduced_percentage'] < 1, 0.75,
49                                             np.where(df['reduced_percentage'] <= 1.05, 1, 0))))
50     df['actual_amount'] = df['level'] * df['price_per_watt']
51     df['amount_paid'] = df['actual_amount'] * df['diff']
52     df['bidder'] = 'Factory ' + df['bidder'].astype(str)
53     df = df.fillna(0)
54     df = df[df['selected']]
55     df = df[['bidder', 'price_per_watt', 'reduced_wattage', 'diff', 'reduced_percentage', 'actual_amount',
56              'amount_paid', 'selected']]
57
58     df_sum = df.sum()
59
60     for col in df_sum.index:
61         if col not in ['diff', 'amount_paid']:
62             df_sum.loc[col] = np.nan
63     df_sum.loc['bidder'] = 'Total'
64     df.loc[df.shape[0]] = df_sum
65     df = util.format_table(df, FORMATS)
66
67     df = df.drop(columns='selected')

```

รูปที่ 10 แสดงส่วนของโปรแกรมในหน้าผลลัพธ์การลดการใช้ไฟฟ้า แสดงขั้นตอนการประมวลผลเพื่อสรุปผลคำตอบแทนการลดการใช้ไฟฟ้าที่คิดเป็นขั้นบันได

```

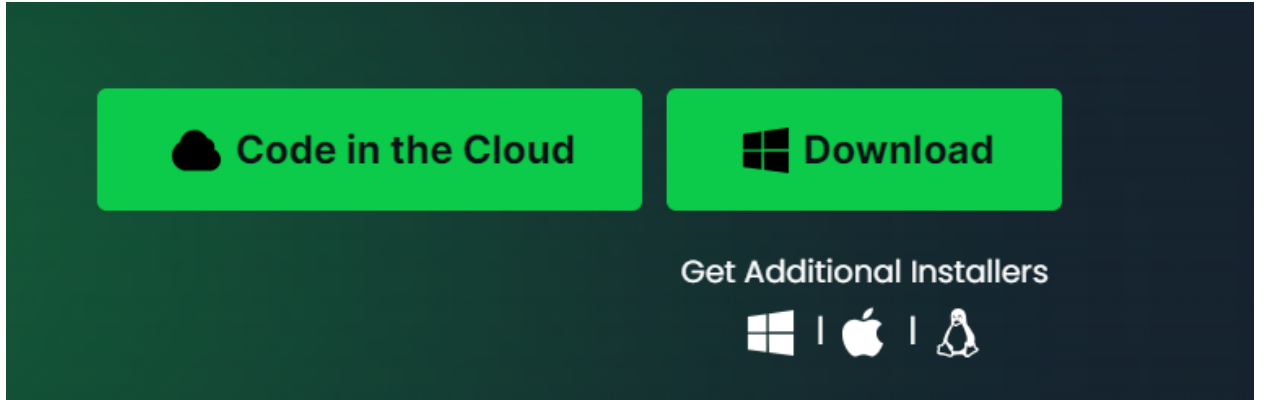
75 contents = [['<50% ของปริมาณ Reduced demand ที่เสนอ', 'ไม่ได้รับผลตอบแทน'],
76             ['50% ถึง <75% ของปริมาณ Reduced demand ที่เสนอ', '50% ของราคา Reduced demand ที่เสนอ'],
77             ['75% ถึง <100% ของปริมาณ Reduced demand ที่เสนอ', '75% ของราคา Reduced demand ที่เสนอ'],
78             ['100% ถึง 105% ของปริมาณ Reduced demand ที่เสนอ', 'เท่ากับราคา Reduced demand ที่เสนอ'],
79             ['>105% ของปริมาณ Reduced demand ที่เสนอ', 'ไม่ได้รับผลตอบแทน']]
80 contents = pd.DataFrame(contents, columns=['h1', 'h2'])
81 content_headers = {'h1': 'ผลการดำเนินงานตอบสนองด้านโหลดของผู้ใช้ไฟฟ้า', 'h2': 'อัตราผลตอบแทน'}
82
83 return html.Div([
84     html.Div([dcc.Link('Back', href='/bidder-selection', className='flow_link'),
85               dcc.Link('Home', href='/', className='flow_link', style={'padding-left': '80%'})],
86               style={'padding-top': '10px'}),
87     html.Div([html.H1('สรุปผลการลดการใช้ไฟฟ้าและคำตอบแทนจริง'),
88               util.generate_table(df, HEADERS, ALIGNMENT, summary=True),
89               html.Div(style={'padding-top': '50px'}),
90               util.generate_table(contents, content_headers, {'h1': 'left', 'h2': 'left'}, deactivate=True)],
91               style={'padding-left': '10%', 'padding-right': '10%'})

```

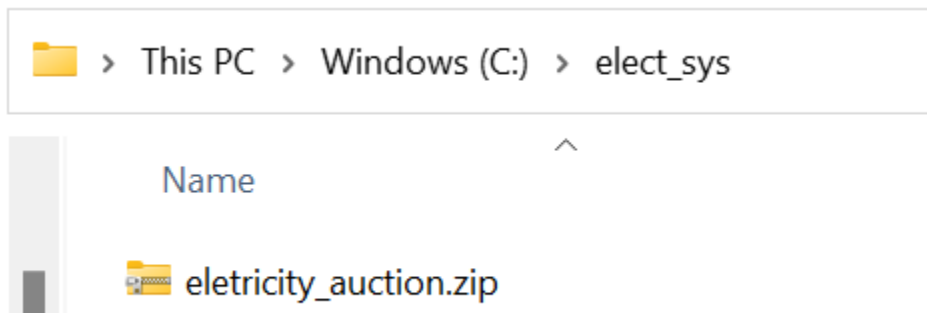
รูปที่ 11 แสดงส่วนของโปรแกรมในหน้าผลลัพธ์การลดการใช้ไฟฟ้า แสดงตารางอธิบายการคิดผลตอบแทน

## ขั้นตอนการติดตั้ง

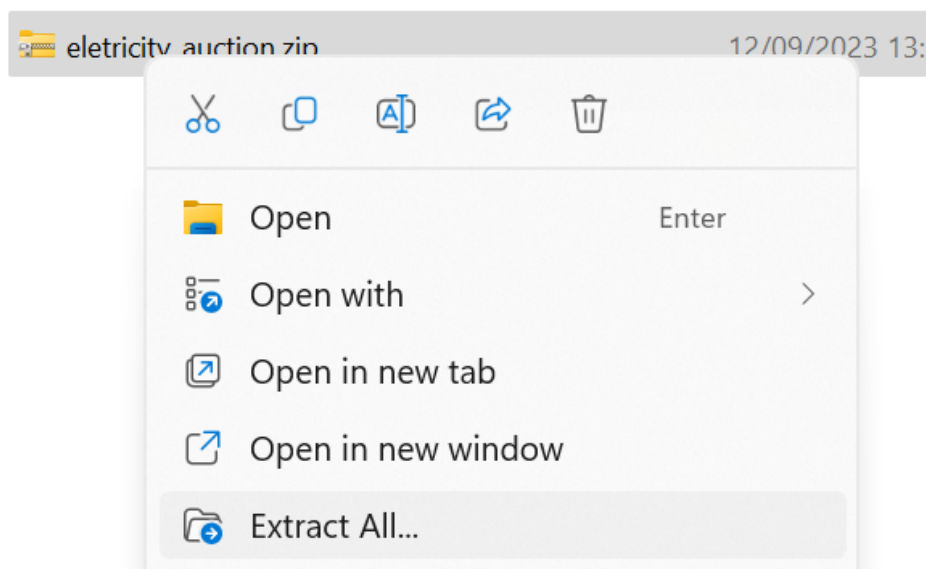
1. ติดตั้งโปรแกรม Anaconda จาก <https://www.anaconda.com/download> กดที่ปุ่ม Download เพื่อโหลดโปรแกรมและติดตั้ง



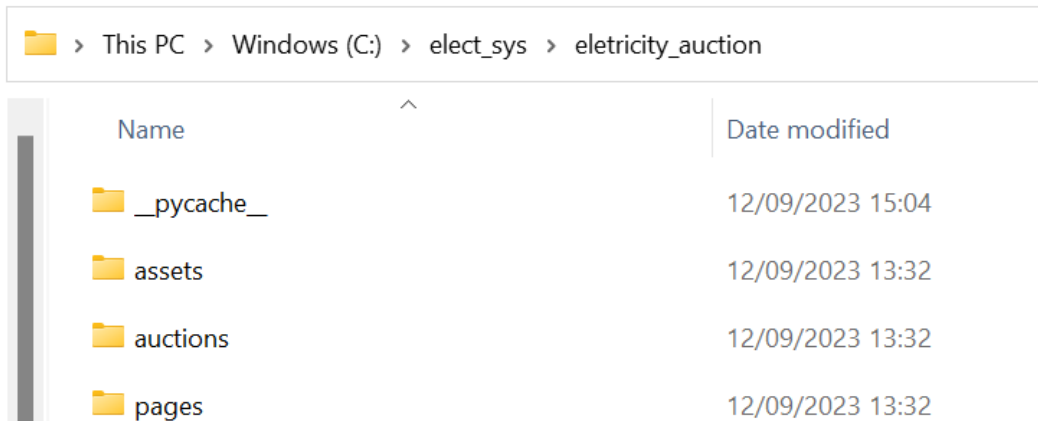
2. สร้างโฟลเดอร์ใหม่ขึ้นมาและนำไฟล์โปรแกรมมาวางไว้



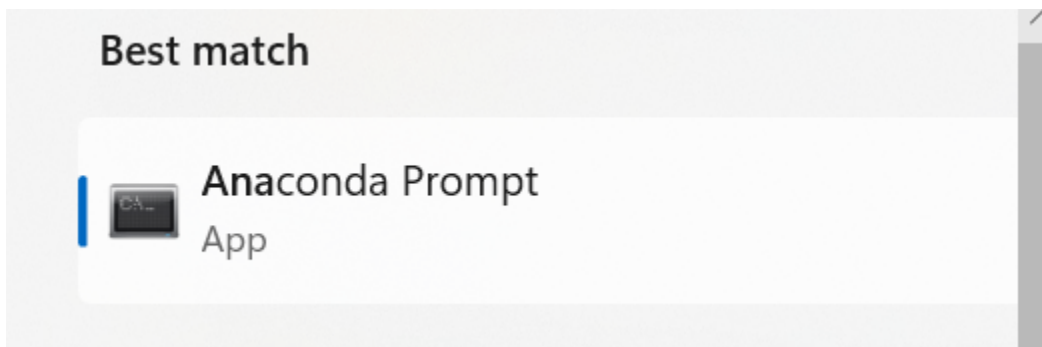
3. Extract ไฟล์โปรแกรม



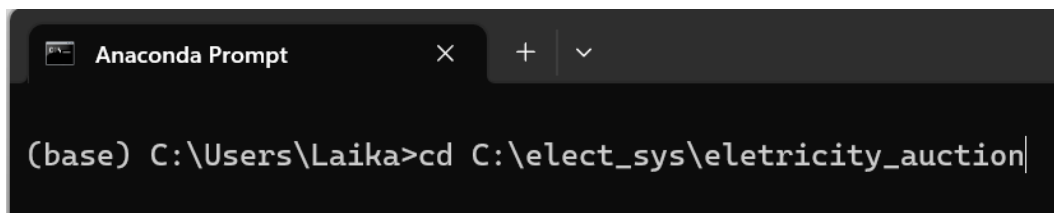
4. เข้าไปที่โฟลเดอร์ที่แตกไว้ และ Copy Path ที่ปรากฏบนโฟลเดอร์นั้น



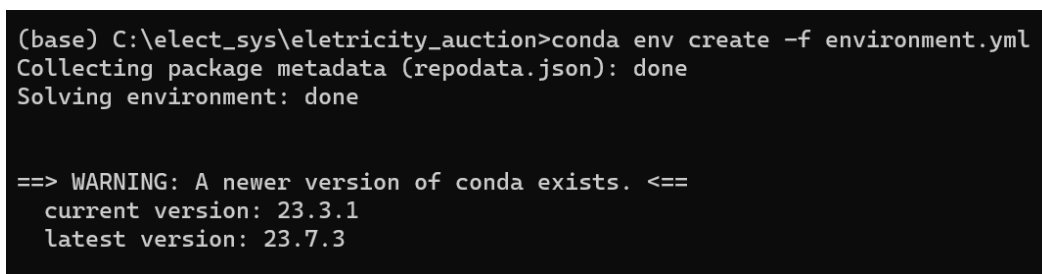
5. เปิด Anaconda Prompt ขึ้นมาทำงาน



6. พิมพ์คำสั่ง cd และวาง path ที่ copy ไว้แล้วกด enter เพื่อย้ายโฟลเดอร์ไปที่ path ดังกล่าว



7. หลังจากนั้นสั่ง conda env create -f environment.yml เพื่อติดตั้งโปรแกรมที่เกี่ยวข้อง ขั้นตอนนี้จะต้องการอินเทอร์เน็ตเพื่อโหลดไฟล์ หากถูกถามคำถามให้กด Y แล้ว Enter



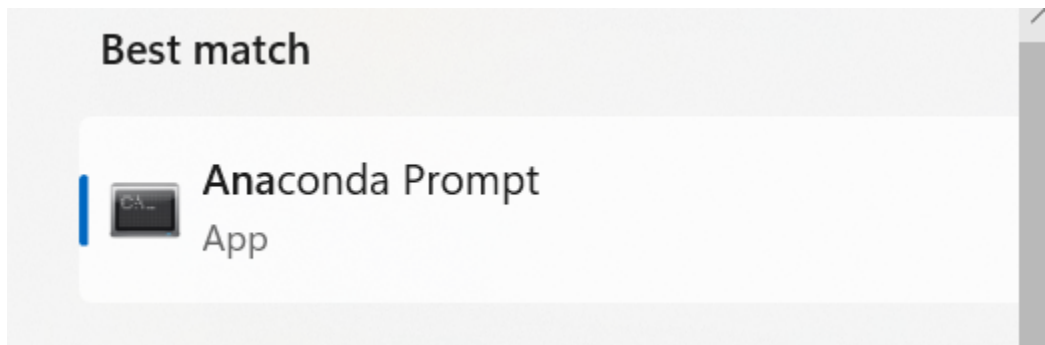
- เมื่อติดตั้งเสร็จแล้ว โปรแกรมจึงพร้อมที่จะใช้งาน

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate elec_sys
#
# To deactivate an active environment, use
#
#     $ conda deactivate

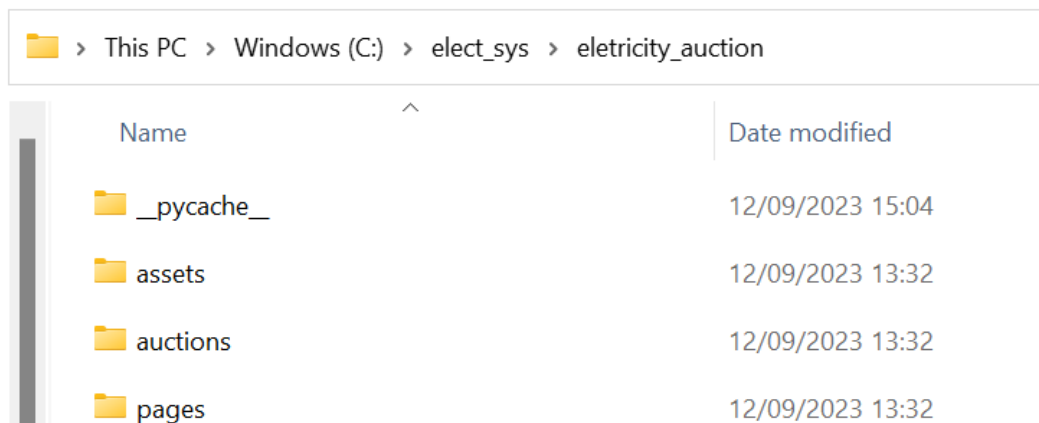
(base) C:\elect_sys\eletricity_auction>
```

## วิธีการรันโปรแกรม

- เปิด Anaconda Prompt ขึ้นมาทำงาน



- เข้าไปที่โฟลเดอร์ที่ติดตั้งไว้ และ Copy Path ที่ปรากฏบนโฟลเดอร์นั้น





3. พิมพ์คำสั่ง cd และวาง path ที่ copy ไว้แล้วกด enter เพื่อย้ายโฟลเดอร์ไปที่ path ดังกล่าว

```
Anaconda Prompt
(base) C:\Users\Laika>cd C:\elect_sys\eletricity_auction|
```

4. Activate environment ด้วยคำสั่ง conda activate elec\_sys

```
(base) C:\elect_sys\eletricity_auction>conda activate elec_sys
(elec_sys) C:\elect_sys\eletricity_auction>|
```

5. หลังจากนั้นสั่ง python demo.py เพื่อรันเว็บไซต์ขึ้นมา

```
(elec_sys) C:\elect_sys\eletricity_auction>python demo.py
Dash is running on http://127.0.0.1:8050/

* Serving Flask app 'demo'
* Debug mode: on
```

6. หลังจากนั้นให้เปิด Browser และพิมพ์ url ที่ปรากฏในขั้นตอนที่ 5 จึงจะสามารถใช้งานได้

