In [58]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [59]:
```
import pandas as pd

# Load the dataset
file_path = r'C:\Users\ruben\Form20 Generated (2).xlsx'
df = pd.read_excel("Form20 Generated (2).csv")

# Display the first few rows of the dataframe
print("Original DataFrame:")
df.head(7)
```

```
Original DataFrame:
```

Out[59]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	U
0	Serial No. Of Polling Station	NaN	No of Valid Votes Cast in favour of	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	Manish Sisodia	Ravinder Singh Negi	Rakesh	Laxman Rawat	Pratap Chandra	Rakesh Suri	Vinay Kumar Singh	Shatrughan Kumar Singh	Sanjeev Bhati	
2	1	1	241	75	8	8	0	0	0	0	0	
3	2	2	298	157	0	2	0	0	0	0	3	
4	3	3	320	158	8	8	1	1	0	0	0	
5	4	4	224	247	2	13	0	0	0	0	0	
6	5	5	147	226	1	11	0	0	0	0	0	

In [60]:
```
df.tail(7)
```

Out[60]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnam
219	197	197	322	442	2	21	0	0	0	0	
220	Total EVM\nVotes	NaN	69974	66703	665	2767	95	60	43	46	1
221	Total Postal Ballot\nVotes	NaN	189	253	11	35	0	0	0	0	
222	Total Votes\nPolled	NaN	70163	66956	676	2802	95	60	43	46	1
223	Place:- PATPARGANJ	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
224	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N
225	Page 8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	N

In [61]:
```
# Drop duplicate values
df.drop_duplicates(inplace=True)

# Display the dataframe after dropping duplicates
print("\nDataFrame after Dropping Duplicates:")
print(df.head())

# Save the updated dataframe to a new Excel file
df.to_excel('updated_dataset.xlsx', index=False)
```

```
DataFrame after Dropping Duplicates:
                    Unnamed: 0 Unnamed: 1  \
0  Serial No. Of Polling Station        NaN
1                            NaN        NaN
2                              1          1
3                              2          2
4                              3          3

                           Unnamed: 2           Unnamed: 3 Unnamed: 4  \
0  No of Valid Votes Cast in favour of                  NaN        NaN
1                       Manish Sisodia  Ravinder Singh Negi     Rakesh
2                                  241                   75          8
3                                  298                  157          0
4                                  320                  158          8

     Unnamed: 5      Unnamed: 6   Unnamed: 7         Unnamed: 8  \
0           NaN             NaN          NaN                NaN
1  Laxman Rawat  Pratap Chandra  Rakesh Suri  Vinay Kumar Singh
2             8               0            0                  0
3             2               0            0                  0
4             8               1            1                  0

               Unnamed: 9    Unnamed: 10     Unnamed: 11   Unnamed: 12  \
0                     NaN            NaN             NaN           NaN
1  Shatrughan Kumar Singh  Sanjeev Bhati  Surender Gupta  Gopal Prasad
2                       0              0               0             0
3                       0              3               1             0
4                       0              0               0             1

   Unnamed: 13    Unnamed: 14           Unnamed: 15 Unnamed: 16 Unnamed: 17
0          NaN            NaN  Total of Valid Votes        NOTA       Total
1  Manoj Kumar  Surjeet Singh                   NaN         NaN         NaN
2            1              0                   333           1         334
3            0              0                   461           0         461
4            1              1                   499           3         502
```

In [62]:
```
df.head()
```

Out[62]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7	Unnamed: 8	Unnamed: 9	Unnamed: 10	U
0	Serial No. Of Polling Station	NaN	No of Valid Votes Cast in favour of	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	NaN	NaN	Manish Sisodia	Ravinder Singh Negi	Rakesh	Laxman Rawat	Pratap Chandra	Rakesh Suri	Vinay Kumar Singh	Shatrughan Kumar Singh	Sanjeev Bhati	
2	1	1	241	75	8	8	0	0	0	0	0	
3	2	2	298	157	0	2	0	0	0	0	3	
4	3	3	320	158	8	8	1	1	0	0	0	

In [63]:
```
print("Original DataFrame:")
print(df.head())

# Rename
df.columns = [f'feature{i+1}' if 'Unnamed' in col else col for i, col in enumerate(df.columns)]

# Display the with renamed columns
print("\nDataFrame with Renamed Columns:")
print(df.head())

# Save the
df.to_excel('updated_dataset.xlsx', index=False)
```

```
Original DataFrame:
                     Unnamed: 0 Unnamed: 1  \
0  Serial No. Of Polling Station        NaN
1                            NaN        NaN
2                              1          1
3                              2          2
4                              3          3

                           Unnamed: 2           Unnamed: 3 Unnamed: 4  \
0  No of Valid Votes Cast in favour of                  NaN        NaN
1                        Manish Sisodia  Ravinder Singh Negi     Rakesh
2                                   241                   75          8
3                                   298                  157          0
4                                   320                  158          8

     Unnamed: 5      Unnamed: 6   Unnamed: 7         Unnamed: 8  \
0           NaN             NaN          NaN                NaN
1  Laxman Rawat  Pratap Chandra  Rakesh Suri  Vinay Kumar Singh
2             8               0            0                  0
3             2               0            0                  0
4             8               1            1                  0

               Unnamed: 9    Unnamed: 10     Unnamed: 11   Unnamed: 12  \
0                     NaN            NaN             NaN           NaN
1  Shatrughan Kumar Singh  Sanjeev Bhati  Surender Gupta  Gopal Prasad
2                       0              0               0             0
3                       0              3               1             0
4                       0              0               0             1

   Unnamed: 13    Unnamed: 14           Unnamed: 15 Unnamed: 16 Unnamed: 17
0          NaN            NaN  Total of Valid Votes        NOTA       Total
1  Manoj Kumar  Surjeet Singh                   NaN         NaN         NaN
2            1              0                   333           1         334
3            0              0                   461           0         461
4            1              1                   499           3         502

DataFrame with Renamed Columns:
                       feature1 feature2  \
0  Serial No. Of Polling Station      NaN
1                            NaN      NaN
2                              1        1
3                              2        2
4                              3        3

                            feature3             feature4 feature5  \
0  No of Valid Votes Cast in favour of                  NaN      NaN
1                        Manish Sisodia  Ravinder Singh Negi   Rakesh
2                                   241                   75        8
3                                   298                  157        0
4                                   320                  158        8

       feature6        feature7     feature8           feature9  \
0           NaN             NaN          NaN                NaN
1  Laxman Rawat  Pratap Chandra  Rakesh Suri  Vinay Kumar Singh
2             8               0            0                  0
3             2               0            0                  0
4             8               1            1                  0

                feature10      feature11       feature12     feature13  \
0                     NaN            NaN             NaN           NaN
1  Shatrughan Kumar Singh  Sanjeev Bhati  Surender Gupta  Gopal Prasad
2                       0              0               0             0
3                       0              3               1             0
4                       0              0               0             1

     feature14      feature15             feature16 feature17 feature18
0          NaN            NaN  Total of Valid Votes      NOTA     Total
1  Manoj Kumar  Surjeet Singh                   NaN       NaN       NaN
2            1              0                   333         1       334
3            0              0                   461         0       461
4            1              1                   499         3       502
```

In [64]:
```
df.head()
```

Out[64]:

	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature13
0	Serial No. Of Polling Station	NaN	No of Valid Votes Cast in favour of	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	Manish Sisodia	Ravinder Singh Negi	Rakesh	Laxman Rawat	Pratap Chandra	Rakesh Suri	Vinay Kumar Singh	Shatrughan Kumar Singh	Sanjeev Bhati	Surender Gupta	Gopa Prasad
2	1	1	241	75	8	8	0	0	0	0	0	0	0
3	2	2	298	157	0	2	0	0	0	0	3	1	0
4	3	3	320	158	8	8	1	1	0	0	0	0	1

In [65]:
```
df.drop(columns=['feature2'],inplace=True)
print("\nDataframe after droping 'feature2' columns:")
```

```
Dataframe after droping 'feature2' columns:
```

In [66]:
```
df.head(7)
```

Out[66]:

	feature1	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature13	feature1
0	Serial No. Of Polling Station	No of Valid Votes Cast in favour of	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
1	NaN	Manish Sisodia	Ravinder Singh Negi	Rakesh	Laxman Rawat	Pratap Chandra	Rakesh Suri	Vinay Kumar Singh	Shatrughan Kumar Singh	Sanjeev Bhati	Surender Gupta	Gopal Prasad	Man Kum
2	1	241	75	8	8	0	0	0	0	0	0	0	
3	2	298	157	0	2	0	0	0	0	3	1	0	
4	3	320	158	8	8	1	1	0	0	0	0	1	
5	4	224	247	2	13	0	0	0	0	0	0	0	
6	5	147	226	1	11	0	0	0	0	0	1	0	

In [67]:
```
# Drop
df.drop(index=0, inplace=True)
# Display
print("\nDataFrame after Dropping the Row:")
print(df.head())
df.to_excel('updated_dataset.xlsx', index=False)
```

```
DataFrame after Dropping the Row:
  feature1        feature3             feature4 feature5      feature6  \
1      NaN  Manish Sisodia  Ravinder Singh Negi   Rakesh  Laxman Rawat
2        1             241                   75        8             8
3        2             298                  157        0             2
4        3             320                  158        8             8
5        4             224                  247        2            13

         feature7     feature8          feature9               feature10  \
1  Pratap Chandra  Rakesh Suri  Vinay Kumar Singh  Shatrughan Kumar Singh
2               0            0                  0                       0
3               0            0                  0                       0
4               1            1                  0                       0
5               0            0                  0                       0

       feature11       feature12     feature13    feature14      feature15  \
1  Sanjeev Bhati  Surender Gupta  Gopal Prasad  Manoj Kumar  Surjeet Singh
2              0              0             0            1              0
3              3              1             0            0              0
4              0              0             1            1              1
5              0              0             0            0              0

  feature16 feature17 feature18
1       NaN       NaN       NaN
2       333         1       334
3       461         0       461
4       499         3       502
5       486         9       495
```

In [68]:
```
df.head(7)
```

Out[68]:

	feature1	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature13	feature1
1	NaN	Manish Sisodia	Ravinder Singh Negi	Rakesh	Laxman Rawat	Pratap Chandra	Rakesh Suri	Vinay Kumar Singh	Shatrughan Kumar Singh	Sanjeev Bhati	Surender Gupta	Gopal Prasad	Man Kum
2	1	241	75	8	8	0	0	0	0	0	0	0	
3	2	298	157	0	2	0	0	0	0	3	1	0	
4	3	320	158	8	8	1	1	0	0	0	0	1	
5	4	224	247	2	13	0	0	0	0	0	0	0	
6	5	147	226	1	11	0	0	0	0	0	1	0	
7	6	174	258	1	14	0	1	0	0	0	1	0	

In [69]:
```
# Rename NaN values in the 'feature1' column adn featureN columns where the null values in datasets
df['feature1'].fillna('Serial No. Of Polling Station', inplace=True)
df['feature16'].fillna('Total of Valid Votes', inplace=True)
df['feature17'].fillna('NOTA', inplace=True)
df['feature18'].fillna('Total', inplace=True)


# Display renaming NaN values
print("\nDataFrame after Renaming NaN Values in 'feature1' Column:")
print(df.head())
df.to_excel('updated_dataset.xlsx', index=False)
```

```
DataFrame after Renaming NaN Values in 'feature1' Column:
                      feature1        feature3             feature4  \
1  Serial No. Of Polling Station  Manish Sisodia  Ravinder Singh Negi
2                              1             241                   75
3                              2             298                  157
4                              3             320                  158
5                              4             224                  247

  feature5      feature6        feature7     feature8           feature9  \
1   Rakesh  Laxman Rawat  Pratap Chandra  Rakesh Suri  Vinay Kumar Singh
2        8             8               0            0                  0
3        0             2               0            0                  0
4        8             8               1            1                  0
5        2            13               0            0                  0

                feature10      feature11       feature12     feature13  \
1  Shatrughan Kumar Singh  Sanjeev Bhati  Surender Gupta  Gopal Prasad
2                       0              0               0             0
3                       0              3               1             0
4                       0              0               0             1
5                       0              0               0             0

     feature14      feature15             feature16 feature17 feature18
1  Manoj Kumar  Surjeet Singh  Total of Valid Votes      NOTA     Total
2            1              0                   333         1       334
3            0              0                   461         0       461
4            1              1                   499         3       502
5            0              0                   486         9       495
```

```
C:\Users\ruben\AppData\Local\Temp\ipykernel_23692\1265646483.py:2: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on w
hich we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'
or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['feature1'].fillna('Serial No. Of Polling Station', inplace=True)
C:\Users\ruben\AppData\Local\Temp\ipykernel_23692\1265646483.py:3: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on w
hich we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'
or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['feature16'].fillna('Total of Valid Votes', inplace=True)
C:\Users\ruben\AppData\Local\Temp\ipykernel_23692\1265646483.py:4: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on w
hich we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'
or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['feature17'].fillna('NOTA', inplace=True)
C:\Users\ruben\AppData\Local\Temp\ipykernel_23692\1265646483.py:5: FutureWarning: A value is trying to be set on
a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on w
hich we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)'
or df[col] = df[col].method(value) instead, to perform the operation inplace on the original object.


  df['feature18'].fillna('Total', inplace=True)
```

In [81]:
```
df.head(9)
```

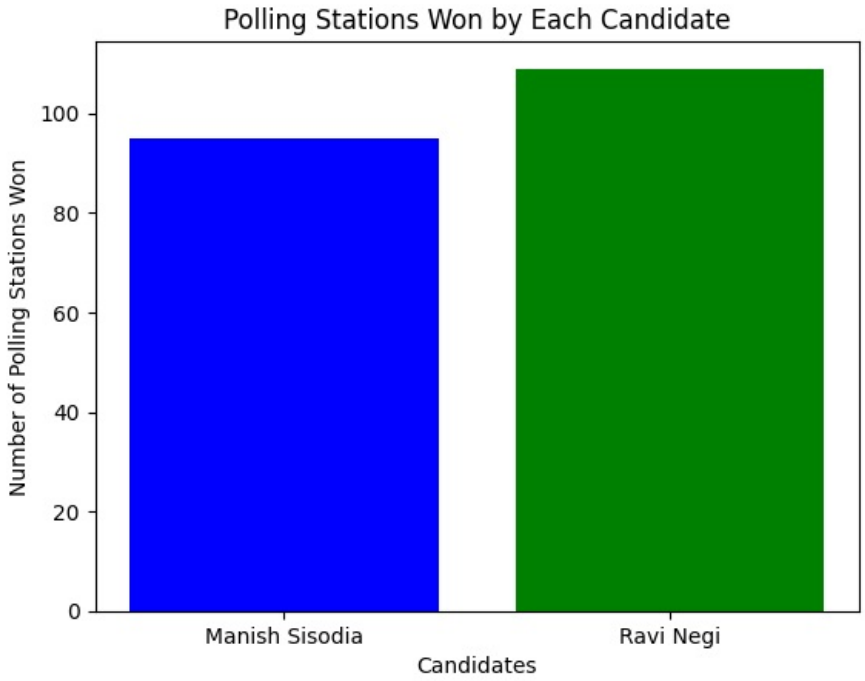
Out[81]:

	feature1	feature2	feature3	feature4	feature5	feature6	feature7	feature8	feature9	feature10	feature11	feature12	feature1
1	Serial No. Of Polling Station	NaN	NaN	Rakesh	Laxman Rawat	Pratap Chandra	Rakesh Suri	Vinay Kumar Singh	Shatrughan Kumar Singh	Sanjeev Bhati	Surender Gupta	Gopal Prasad	Mani Kuma
2	1	241.0	75.0	8	8	0	0	0	0	0	0	0	
3	2	298.0	157.0	0	2	0	0	0	0	3	1	0	
4	3	320.0	158.0	8	8	1	1	0	0	0	0	1	
5	4	224.0	247.0	2	13	0	0	0	0	0	0	0	
6	5	147.0	226.0	1	11	0	0	0	0	0	1	0	
7	6	174.0	258.0	1	14	0	1	0	0	0	1	0	
8	7	246.0	284.0	1	10	0	0	0	0	0	0	0	
9	8	239.0	356.0	0	16	0	0	0	0	0	1	0	

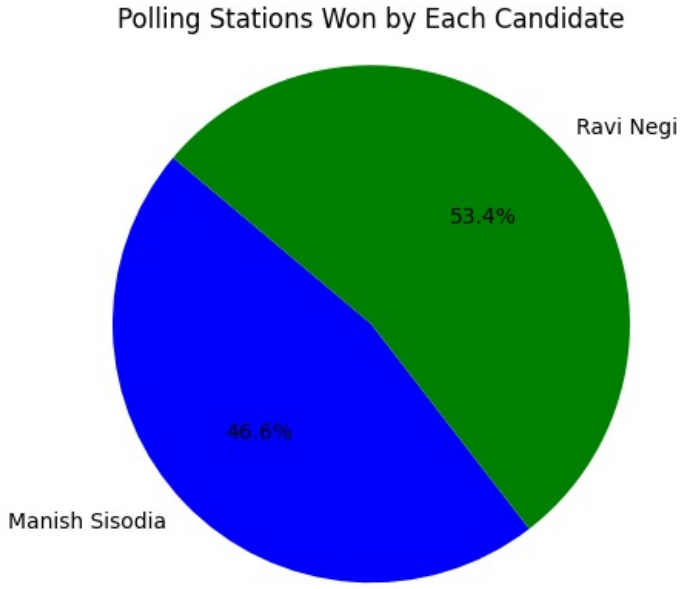
In [79]:
```
print("Original DataFrame:")
print(df.head())
#To convert to numeric to feature2
df['feature2'] = pd.to_numeric(df['feature2'], errors='coerce')
df['feature3'] = pd.to_numeric(df['feature3'], errors='coerce')

# Calculate the winner in datsets
df['Winner'] = df.apply(lambda row: 'Manish Sisodia' if row['feature2'] > row['feature3'] else 'Ravi Negi', axi

# Task1
manish_wins = df[df['Winner'] == 'Manish Sisodia'].shape[0]
```

```python
ravi_wins = df[df['Winner'] == 'Ravi Negi'].shape[0]

print(f"Polling stations won by Manish Sisodia: {manish_wins}")
print(f"Polling stations won by Ravi Negi: {ravi_wins}")

# Task2
manish_lost_by_50 = df[(df['Winner'] == 'Ravi Negi') & ((df['feature3'] - df['feature2']) > 50)].shape[0]

print(f"Booths where Manish Sisodia lost by more than 50 votes: {manish_lost_by_50}")

# Task3
ravi_won_by_100 = df[(df['Winner'] == 'Ravi Negi') & ((df['feature3'] - df['feature2']) >= 100)].shape[0]
print(f"Booths where Ravi Negi won by 100 or more votes: {ravi_won_by_100}")

# Create a histogram to represent the results of Task 1
candidates = ['Manish Sisodia', 'Ravi Negi']
wins = [manish_wins, ravi_wins]
#To create bar chat
plt.bar(candidates, wins, color=['blue', 'green'])
plt.xlabel('Candidates')
plt.ylabel('Number of Polling Stations Won')
plt.title('Polling Stations Won by Each Candidate')
plt.show()
```

```
Original DataFrame:
                       feature1        feature2             feature3  \
1  Serial No. Of Polling Station  Manish Sisodia  Ravinder Singh Negi
2                               1             241                   75
3                               2             298                  157
4                               3             320                  158
5                               4             224                  247

  feature4      feature5        feature6     feature7           feature8  \
1   Rakesh  Laxman Rawat  Pratap Chandra  Rakesh Suri  Vinay Kumar Singh
2        8             8               0            0                  0
3        0             2               0            0                  0
4        8             8               1            1                  0
5        2            13               0            0                  0

                feature9      feature10       feature11     feature12  \
1  Shatrughan Kumar Singh  Sanjeev Bhati  Surender Gupta  Gopal Prasad
2                       0              0               0             0
3                       0              3               1             0
4                       0              0               0             1
5                       0              0               0             0

     feature13      feature14             feature15 feature16 feature17  \
1  Manoj Kumar  Surjeet Singh  Total of Valid Votes      NOTA     Total
2            1              0                   333         1       334
3            0              0                   461         0       461
4            1              1                   499         3       502
5            0              0                   486         9       495

          Winner
1      Ravi Negi
2  Manish Sisodia
3  Manish Sisodia
4  Manish Sisodia
5      Ravi Negi
Polling stations won by Manish Sisodia: 95
Polling stations won by Ravi Negi: 109
Booths where Manish Sisodia lost by more than 50 votes: 81
Booths where Ravi Negi won by 100 or more votes: 48
```

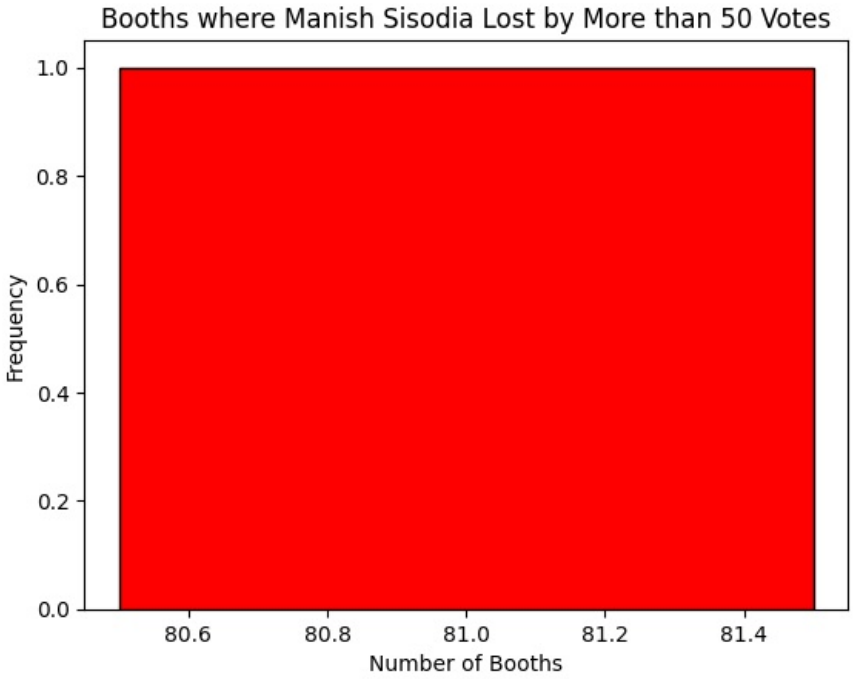


In [83]:
```
plt.pie(wins, labels=candidates, colors=['blue', 'green'], autopct='%1.1f%%', startangle=140)
plt.title('Polling Stations Won by Each Candidate')
plt.axis('equal')
plt.show()
```

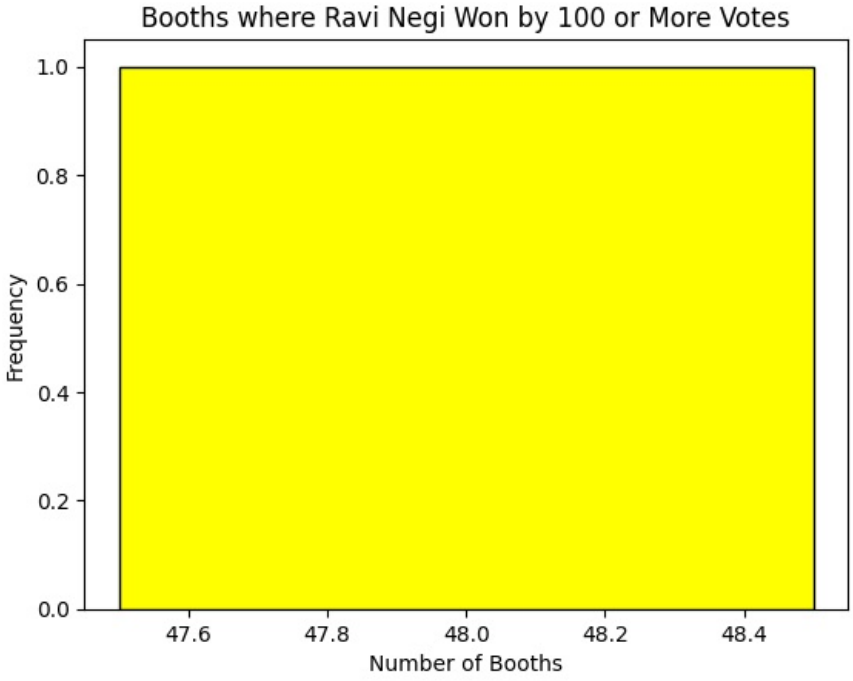


In [84]:
```
plt.hist([manish_lost_by_50], bins=1, color='red', edgecolor='black')
plt.xlabel('Number of Booths')
plt.ylabel('Frequency')
plt.title('Booths where Manish Sisodia Lost by More than 50 Votes')
```

```
plt.show()
```



In [86]:
```
plt.hist([ravi_won_by_100], bins=1, color='yellow', edgecolor='black')
plt.xlabel('Number of Booths')
plt.ylabel('Frequency')
plt.title('Booths where Ravi Negi Won by 100 or More Votes')
plt.show()
```



In [ ]:
```

```