```
import pandas as pd

dataset = pd.read_csv('/content/Ads_CTR_Optimisation.csv')
display(dataset.head())
```

	Ad 0	Ad 1	Ad 2	Ad 3	Ad 4	Ad 5	Ad 6	Ad 7	Ad 8	Ad 9
0	1	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	0	0	0	0

```
import numpy as np

n_users = len(dataset)
n_ads = len(dataset.columns)

numbers_of_selections = np.zeros(n_ads, dtype=int)
sums_of_rewards = np.zeros(n_ads, dtype=int)

print(f"Number of users (rows): {n_users}")
print(f"Number of ads (columns): {n_ads}")
print(f"Initial numbers_of_selections: {numbers_of_selections}")
print(f"Initial sums_of_rewards: {sums_of_rewards}")
```

```
Number of users (rows): 10000
Number of ads (columns): 10
Initial numbers_of_selections: [0 0 0 0 0 0 0 0 0 0]
Initial sums_of_rewards: [0 0 0 0 0 0 0 0 0 0]
```

```
import random

epsilon = 0.1
total_reward = 0

for user in range(n_users):
    ad_index = -1
    if random.random() < epsilon:
        # Exploration
        ad_index = random.randrange(n_ads)
    else:
        # Exploitation
        max_average_reward = -1
        for i in range(n_ads):
            if numbers_of_selections[i] > 0:
                average_reward = sums_of_rewards[i] / numbers_of_selections[i]
            else:
                average_reward = -1 # Handle division by zero
            if average_reward > max_average_reward:
                max_average_reward = average_reward
                ad_index = i

    # Update
    reward = dataset.values[user, ad_index]
    numbers_of_selections[ad_index] += 1
    sums_of_rewards[ad_index] += reward
    total_reward += reward

print(f"Total reward: {total_reward}")
print(f"Numbers of selections: {numbers_of_selections}")
print(f"Sums of rewards: {sums_of_rewards}")
```

```
Total reward: 2554
Numbers of selections: [ 101   95   82   80 9101  102  114  113  113   99]
Sums of rewards: [  23   12    3    5 2458    0   17   21   11    4]
```

```
average_rewards = np.divide(sums_of_rewards, numbers_of_selections, out=np.zeros_like(sums_of_rewards, dtype=float), where=numbers_of_se
best_ad_index = np.argmax(average_rewards)
best_ad_average_reward = average_rewards[best_ad_index]

print(f"The best ad is Ad {best_ad_index} with an average reward of {best_ad_average_reward:.4f}")
```

```
The best ad is Ad 4 with an average reward of 0.2701
```

```
import matplotlib.pyplot as plt
```

```
plt.figure(figsize=(10, 6))
plt.bar(range(n_ads), numbers_of_selections)
plt.title('Number of Times Each Ad Was Selected')
plt.xlabel('Ad Index')
plt.ylabel('Number of Selections')
plt.xticks(range(n_ads))
plt.show()
```