4a. Alpha-Beta Pruning

In [1]:

```
def alphabeta(depth, nodeIndex, maximizingPlayer, values, alpha, beta, path):
    if depth == 3:
        return values[nodeIndex], path + [nodeIndex]

    if maximizingPlayer:
        best = float('-inf')
        best_path = []
        for i in range(2):
            val, new_path = alphabeta(depth + 1, nodeIndex * 2 + i, False, values, alpha, beta, pat
            if val > best:
                best = val
                best_path = new_path
            alpha = max(alpha, best)
            if beta <= alpha:
                break
        return best, best_path
    else:
        best = float('inf')
        best_path = []
        for i in range(2):
            val, new_path = alphabeta(depth + 1, nodeIndex * 2 + i, True, values, alpha, beta, path
            if val < best:
                best = val
                best_path = new_path
            beta = min(beta, best)
            if beta <= alpha:
                break
        return best, best_path
```

In [2]:

```
# Example tree with depth 3 and 8 terminal nodes
values = [3, 5, 2, 9, 12, 5, 23, 23]

# Start the Alpha-Beta Pruning algorithm
optimal_value, optimal_path = alphabeta(0, 0, True, values, float('-inf'), float('inf'), [])
print("The optimal value is:", optimal_value)
print("The path taken is:", optimal_path)
```

```
The optimal value is: 12
The path taken is: [0, 1, 2, 4]
```

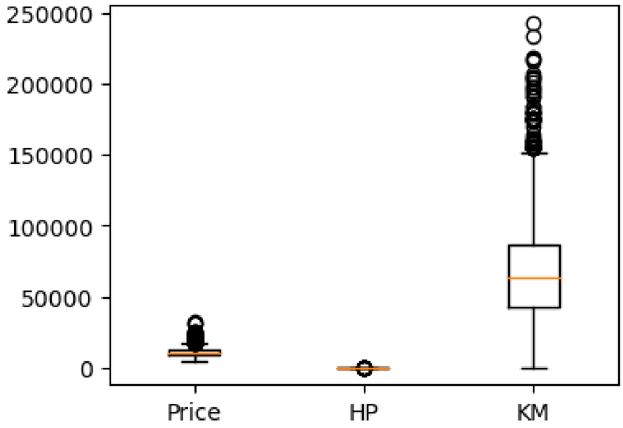
4b. Box Plot

In [3]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

In [4]:

```
data=pd.read_csv('./corolla.csv')
plt.figure(figsize = ( 4 , 3 ))
plt.boxplot([data["Price"],data["HP"],data["KM"]])
plt.xticks([1,2,3],["Price","HP","KM"])
plt.show()
```


Price
HP
KM