Now the last step for creating a Restaurant Recommendation System is to write a function that will recommend restaurants:

```
def recommend(name, cosine_similarities = cosine_similarities):

    # Create a list to put top restaurants
    recommend_restaurant = []

    # Find the index of the hotel entered
    idx = indices[indices == name].index[0]

    # Find the restaurants with a similar cosine-sim value and order them from bigges number
    score_series = pd.Series(cosine_similarities[idx]).sort_values(ascending=False)

    # Extract top 30 restaurant indexes with a similar cosine-sim value
    top30_indexes = list(score_series.iloc[0:31].index)

    # Names of the top 30 restaurants
    for each in top30_indexes:
        recommend_restaurant.append(list(df_percent.index)[each])

    # Creating the new data set to show similar restaurants
    df_new = pd.DataFrame(columns=['cuisines', 'Mean Rating', 'cost'])

    # Create the top 30 similar restaurants with some of their columns
    for each in recommend_restaurant:
        df_new = pd.concat([df_new, df_percent[['cuisines', 'Mean Rating', 'cost']][df_percent.index == each].sample()])

    # Drop the same named restaurants and sort only the top 10 by the highest rating
    df_new = df_new.drop_duplicates(subset=['cuisines','Mean Rating', 'cost'], keep=False)
    df_new = df_new.sort_values(by='Mean Rating', ascending=False).head(10)

    print('TOP %s RESTAURANTS LIKE %s WITH SIMILAR REVIEWS: ' % (str(len(df_new)), name))

    return df_new
recommend('Pai Vihar')
```

Name	Mean Rating	cost
Chianti	4.59	1.5
Bologna	4.48	1.0
Lavonne	4.35	800.0
Foxtrot - House Of Subculture	4.35	1.0
The Lantern Restaurant & Bar - The Ritz-Carlton...	4.28	3.5
Forage	4.03	1.5
Foodhall	3.80	1.0
1992 Chats - Space	3.45	200.0