IMPORT LIB

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

IMPORTING DATA

```
df = pd.read_excel(r"C:\Users\anush\Downloads\DA\PROJECT\Dataset\AI dts.xlsx")
```

```
AI_Project_Master = pd.read_excel(r"C:\Users\anush\Downloads\DA\PROJECT\Dataset\AI dts.xlsx", sheet_name=0)
AI_Usage_Telemetry = pd.read_excel(r"C:\Users\anush\Downloads\DA\PROJECT\Dataset\AI dts.xlsx", sheet_name=1)
AI_Skills_Training = pd.read_excel(r"C:\Users\anush\Downloads\DA\PROJECT\Dataset\AI dts.xlsx", sheet_name=2)
AI_Revenue_impact = pd.read_excel(r"C:\Users\anush\Downloads\DA\PROJECT\Dataset\AI dts.xlsx", sheet_name=3)
dept_performance= pd.read_excel(r"C:\Users\anush\Downloads\DA\PROJECT\Dataset\AI dts.xlsx", sheet_name=4)
```

exploring data

```
AI_Project_Master.head()
```

	project_id	project_name	department	start_date	end_date	project_status	project_cost
0	P0001	Customer Churn Predictor	Marketing	2020-08-16	2020-10-10	Completed	504885.87
1	P0002	Sales Forecasting Engine	Operations	2023-01-21	2023-05-20	Active	82929.71
2	P0003	AI Inventory Optimizer	R&D	2022-10-31	2024-08-07	Active	413698.55
3	P0004	Fraud Risk Classifier	Marketing	2022-07-08	2024-04-01	Completed	1741214.64
4	P0005	Document Automation AI	Product	2021-11-27	2022-10-07	Proposed	1915281.88

```
AI_Project_Master.tail()
```

	project_id	project_name	department	start_date	end_date	project_status	project_cost
195	P0196	Dynamic Ride Allocation Engine	Sales	2020-12-23	2021-05-14	Active	71044.70
196	P0197	Predictive Inventory Demand Clustering	Sales	2021-05-02	2023-01-24	Proposed	365007.48
197	P0198	Smart Label Detection Vision Model	Supply Chain	2022-05-12	2023-12-01	Completed	145844.92
198	P0199	Customer Repeat Purchase Predictor	Legal	2022-11-17	2024-08-15	Proposed	1017600.61
199	P0200	Email Intent Classification System	Sales	2022-12-03	2023-11-06	On Hold	1928771.84

```
AI_Project_Master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   project_id      200 non-null    object
 1   project_name    200 non-null    object
 2   department      200 non-null    object
 3   start_date      200 non-null    datetime64[ns]
 4   end_date        172 non-null    datetime64[ns]
 5   project_status  200 non-null    object
 6   project_cost    200 non-null    float64
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 11.1+ KB
```

```
AI_Project_Master.describe()
```

	start_date	end_date	project_cost
count	200	172	2.000000e+02
mean	2021-09-30 18:00:00	2023-01-29 08:13:57.209302272	9.804229e+05
min	2020-01-03 00:00:00	2020-08-16 00:00:00	3.423786e+04
25%	2020-12-08 00:00:00	2022-03-10 18:00:00	5.010976e+05
50%	2021-10-24 00:00:00	2023-01-28 00:00:00	9.383854e+05
75%	2022-07-25 06:00:00	2023-12-13 06:00:00	1.494349e+06
max	2023-04-13 00:00:00	2025-06-13 00:00:00	1.998760e+06
std	NaN	NaN	5.656746e+05

```
AI_Project_Master.shape
```

```
(200, 7)
```

```
AI_Project_Master.columns
```

```
Index(['project_id', 'project_name', 'department', 'start_date', 'end_date',
       'project_status', 'project_cost'],
      dtype='object')
```

EXPLORING DATA AI_Usage_Telemetry

```
AI_Usage_Telemetry.head()
```

	usage_id	project_id	feature_name	users_active	usage_count	usage_date	platform	dashb
0	U000001	P0036	User Authenticatior	246	1476	2024-04-14	Mobile	dashb
1	U000002	P0101	Role-Based Access Control	330	4620	2022-03-11	Mobile	dashb
2	U000003	P0064	Real-Time Notifications	36	288	2025-05-26	Internal	dashb
3	U000004	P0036	Data Encryption Module	63	1197	2025-04-18	Mobile	dashb
4	U000005	P0053	Predictive Analytics Engine	72	648	2024-04-06	Internal	dashb

```
AI_Usage_Telemetry.tail()
```

	usage_id	project_id	feature_name	users_active	usage_count	usage_date	platform	dashb
6495	U006496	P0051	Context-Aware Recommendations	184	368	2020-04-25	API	dashb
6496	U006497	P0155	Data Encryption Module	410	1640	2022-06-04	Internal	dashb
6497	U006498	P0058	Data Ingestion Pipeline	253	3795	2023-10-26	Internal	dashb
6498	U006499	P0036	Real-Time Event Stream	251	1004	2022-05-10	Internal	dashb
6499	U006500	P0145	Data Ingestion Pipeline	300	1200	2022-03-08	Mobile	dashb

```
AI_Usage_Telemetry.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6500 entries, 0 to 6499
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   usage_id      6500 non-null   object
 1   project_id    6500 non-null   object
 2   feature_name  6500 non-null   object
 3   users_active  6500 non-null   int64
 4   usage_count   6500 non-null   int64
 5   usage_date    6500 non-null   datetime64[ns]
 6   platform      6500 non-null   object
 7   dashb         6500 non-null   object
dtypes: datetime64[ns](1), int64(2), object(5)
memory usage: 406.4+ KB
```

```
AI_Usage_Telemetry.describe()
```

	users_active	usage_count	usage_date
count	6500.000000	6500.000000	6500
mean	249.259385	2648.053538	2022-09-18 20:40:10.338461440
min	0.000000	0.000000	2020-01-01 00:00:00
25%	124.000000	736.000000	2021-05-09 00:00:00
50%	248.500000	1975.000000	2022-08-30 12:00:00
75%	376.000000	4030.500000	2024-02-04 00:00:00
max	500.000000	9880.000000	2025-06-23 00:00:00
std	144.851221	2290.989111	NaN

```
AI_Usage_Telemetry.shape
```

```
(6500, 8)
```

```
AI_Usage_Telemetry.columns
```

```
Index(['usage_id', 'project_id', 'feature_name', 'users_active', 'usage_count',
       'usage_date', 'platform', 'dashb'],
      dtype='object')
```

EXPLORING DATA

```
AI_Usage_Telemetry.isnull().sum()
```

```
usage_id        0
project_id      0
feature_name    0
users_active    0
usage_count     0
usage_date      0
platform        0
dashb           0
dtype: int64
```

```
AI_Usage_Telemetry.duplicated().sum()
```

```
np.int64(0)
```

```
AI_Usage_Telemetry = AI_Usage_Telemetry.drop(columns=['dashb'])
```

```
AI_Usage_Telemetry.head()
```

	usage_id	project_id	feature_name	users_active	usage_count	usage_date	platform
0	U000001	P0036	User Authenticatior	246	1476	2024-04-14	Mobile
1	U000002	P0101	Role-Based Access Control	330	4620	2022-03-11	Mobile
2	U000003	P0064	Real-Time Notifications	36	288	2025-05-26	Internal
3	U000004	P0036	Data Encryption Module	63	1197	2025-04-18	Mobile
4	U000005	P0053	Predictive Analytics Engine	72	648	2024-04-06	Internal

data expolaration and handling ->AI_Skills_Training

```
AI_Skills_Training.head()
```

	training_id	employee_id	department	course_name	training_hours	completion_status	completion_date
0	T000001	E02787	HR	Generative AI Fundamentals	40.7	In Progress	NaT
1	T000002	E02813	Supply Chain	AI Product Management	41.8	Completed	2020-03-14
2	T000003	E00406	HR	Image Classification with CNNs	9.8	Completed	2022-09-02
3	T000004	E02172	Sales	AI for Business Strategy	75.4	Completed	2020-04-19
4	T000005	E01959	Sales	AI-Driven Decision Making	63.3	Completed	2024-03-24

```
AI_Skills_Training.tail()
```

	training_id	employee_id	department	course_name	training_hours	completion_status	completion_date
1195	T001196	E00312	R&D	Text Mining and Information Retrieval	22.4	Completed	2025-06-17
1196	T001197	E02578	Marketing	Text Mining and Information Retrieval	31.7	Completed	2025-06-20
1197	T001198	E02262	HR	Natural Language Processing Basics	6.0	Completed	2020-07-28
			Supply				

```
AI_Skills_Training.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1200 entries, 0 to 1199
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   training_id        1200 non-null   object
 1   employee_id        1200 non-null   object
 2   department         1200 non-null   object
 3   course_name        1200 non-null   object
 4   training_hours     1200 non-null   float64
 5   completion_status  1200 non-null   object
 6   completion_date    834 non-null    datetime64[ns]
dtypes: datetime64[ns](1), float64(1), object(5)
memory usage: 65.8+ KB
```

```
AI_Skills_Training.describe()
```

	training_hours	completion_date
count	1200.000000	834
mean	60.135583	2022-10-02 05:09:03.884892160
min	1.100000	2020-01-04 00:00:00
25%	32.200000	2021-06-01 00:00:00
50%	61.000000	2022-10-10 00:00:00
75%	89.525000	2024-02-19 12:00:00
max	119.700000	2025-06-21 00:00:00
std	34.237248	NaN

```
AI_Skills_Training.columns
```

```
Index(['training_id', 'employee_id', 'department', 'course_name',
       'training_hours', 'completion_status', 'completion_date'],
      dtype='object')
```

```
AI_Skills_Training.shape
```

```
(1200, 7)
```

```
AI_Skills_Training.isnull().sum()
```

```
training_id            0
employee_id            0
department             0
course_name            0
training_hours         0
completion_status      0
completion_date      366
dtype: int64
```

```
AI_Skills_Training.duplicated().sum()
```

```
np.int64(0)
```

DATA EXPO AND HANDLING -->AI_Revenue_impact

```
AI_Revenue_impact.head()
```

	impact_id	project_id	impact_date	revenue_generated	cost_savings	impact_type	region
0	I00001	P0103	2022-05-23	0.00	136455.12	Savings	APAC
1	I00002	P0021	2021-10-21	0.00	41252.40	Savings	AMER
2	I00003	P0098	2023-05-08	113428.85	14876.77	Revenue	APAC
3	I00004	P0168	2023-08-26	0.00	129823.85	Savings	LATAM
4	I00005	P0116	2024-12-23	84732.90	62398.45	Revenue	APAC

```
AI_Revenue_impact.tail()
```

	impact_id	project_id	impact_date	revenue_generated	cost_savings	impact_type	region
2064	I00075	P0032	2024-06-25	222371.63	0.00	Revenue	LATAM
2065	I00076	P0087	2025-06-17	173017.12	119593.63	Revenue	AMER
2066	I00077	P0022	2024-01-29	107651.79	0.00	Revenue	EMEA
2067	I00078	P0140	2021-08-24	20017.23	45974.38	Savings	LATAM
2068	I00079	P0043	2023-11-13	111782.41	0.00	Revenue	AMER

```
AI_Revenue_impact.describe()
```

	impact_date	revenue_generated	cost_savings
count	2069	2069.000000	2069.000000
mean	2022-10-12 16:45:00.434992896	74641.028468	51554.504553
min	2020-01-03 00:00:00	0.000000	0.000000
25%	2021-06-03 00:00:00	0.000000	0.000000
50%	2022-10-18 00:00:00	39543.300000	40526.230000
75%	2024-02-08 00:00:00	147966.160000	95947.020000
max	2025-06-22 00:00:00	249056.920000	149996.490000
std	NaN	83064.633220	49905.189717

```
AI_Revenue_impact.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2069 entries, 0 to 2068
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   impact_id          2069 non-null   object
 1   project_id         2069 non-null   object
 2   impact_date        2069 non-null   datetime64[ns]
 3   revenue_generated  2069 non-null   float64
 4   cost_savings       2069 non-null   float64
 5   impact_type        2069 non-null   object
 6   region             2069 non-null   object
dtypes: datetime64[ns](1), float64(2), object(4)
memory usage: 113.3+ KB
```

```
AI_Revenue_impact.isnull().sum()
```

```
impact_id            0
project_id           0
impact_date          0
revenue_generated    0
cost_savings         0
impact_type          0
region               0
dtype: int64
```

```
AI_Revenue_impact.duplicated().sum()
```

```
np.int64(69)
```

```
AI_Revenue_impact.drop_duplicates(inplace=True)
```

DATA EXPO AND MANIPULATION-->dept_performance

```
dept_performance.head()
```

	department_id	department_name	total_revenue	total_cost	employees_count	ai_projects_active	ai_adoption_level	err0r
0	D0001	Marketing	9694841.60	5559957.44	314	19	Medium	no
1	D0002	Sales	39278865.09	19796917.10	148	7	Low	no
2	D0003	Finance	12631616.88	6924176.59	590	15	Low	no
3	D0004	Operations	48207122.01	28452996.52	649	16	Low	no
4	D0005	Marketing	47436721.68	33516623.15	1163	10	High	no

```
dept_performance.tail()
```

	department_id	department_name	total_revenue	total_cost	employees_count	ai_projects_active	ai_adoption_level	err0r
95	D0096	Product	42066344.00	28757887.42	232	7	High	no
96	D0097	Product	47944385.14	41944024.05	285	23	Medium	no
97	D0098	Sales	11012125.05	9757193.33	579	27	High	no
98	D0099	Sales	858880.65	396988.98	139	9	Medium	no
99	D0100	Customer	9327193.03	5177506.54	957	20	Medium	no

```
dept_performance.describe()
```

	total_revenue	total_cost	employees_count	ai_projects_active
count	1.000000e+02	1.000000e+02	100.000000	100.000000
mean	2.539355e+07	1.672342e+07	599.900000	14.030000
std	1.504612e+07	1.061176e+07	344.408921	9.110384
min	1.485118e+05	8.862436e+04	9.000000	0.000000
25%	1.095816e+07	7.130213e+06	286.500000	6.000000
50%	2.562027e+07	1.575096e+07	609.500000	15.000000
75%	3.908693e+07	2.444413e+07	895.500000	21.000000
max	4.971740e+07	4.194402e+07	1172.000000	30.000000

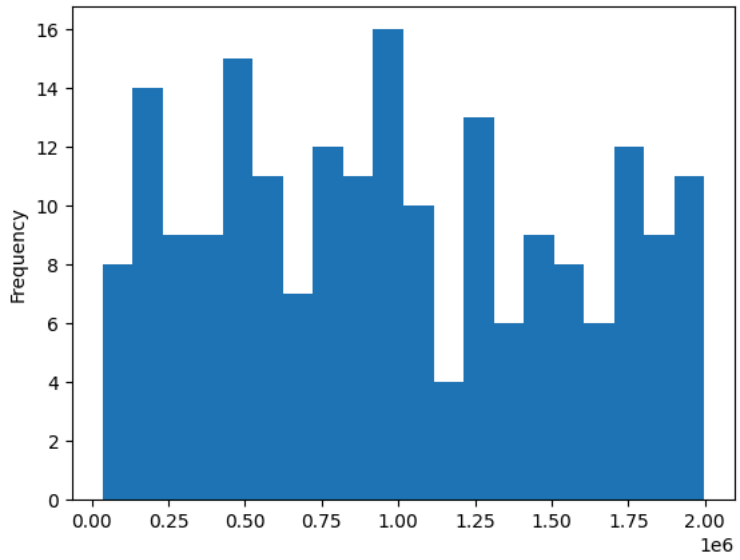
```
dept_performance.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 8 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   department_id       100 non-null    object
 1   department_name     100 non-null    object
 2   total_revenue       100 non-null    float64
 3   total_cost          100 non-null    float64
 4   employees_count     100 non-null    int64
 5   ai_projects_active  100 non-null    int64
 6   ai_adoption_level   100 non-null    object
 7   err0r               100 non-null    object
dtypes: float64(2), int64(2), object(4)
memory usage: 6.4+ KB
```

```
dept_performance = dept_performance.drop(columns='err0r')
```

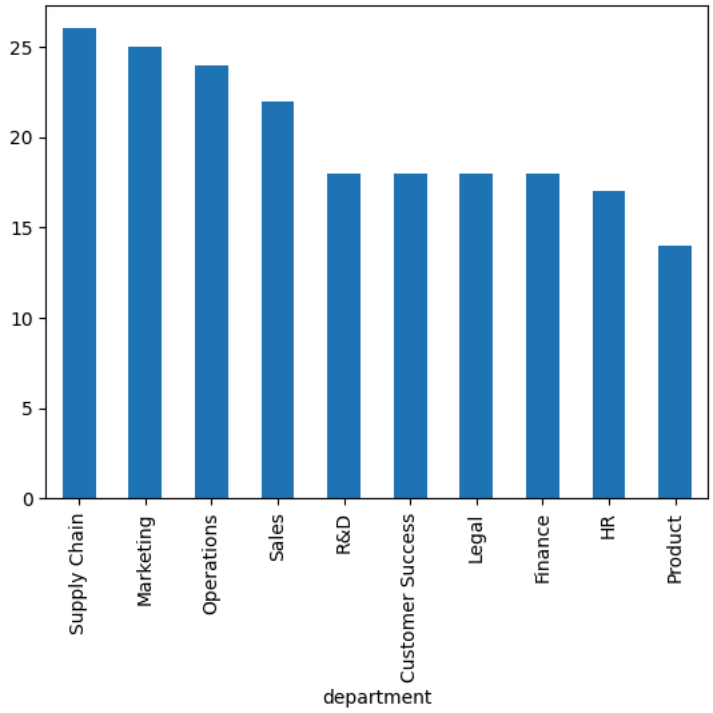
```
import matplotlib.pyplot as plt
import seaborn as sns
## What is the distribution of project costs?
AI_Project_Master["project_cost"].plot(kind="hist", bins=20)
```

```
<Axes: ylabel='Frequency'>
```



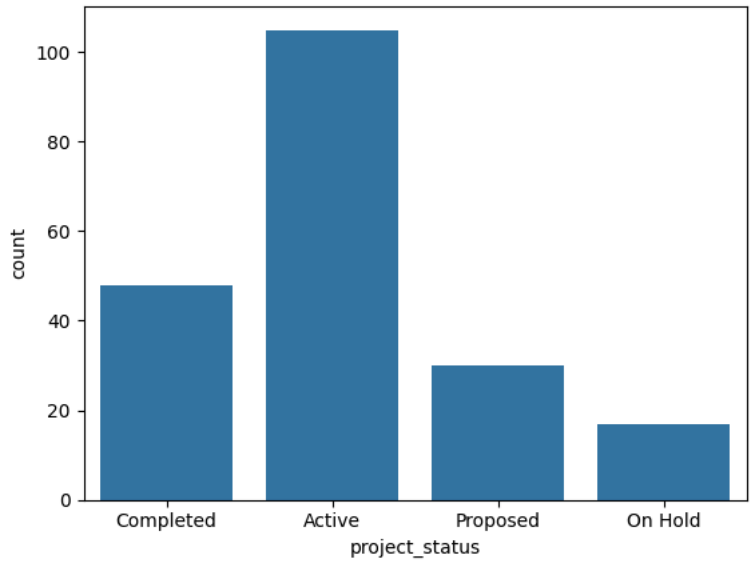

```
# Which departments run the most projects?

AI_Project_Master["department"].value_counts().plot(kind="bar")
```

```
<Axes: xlabel='department'>
```



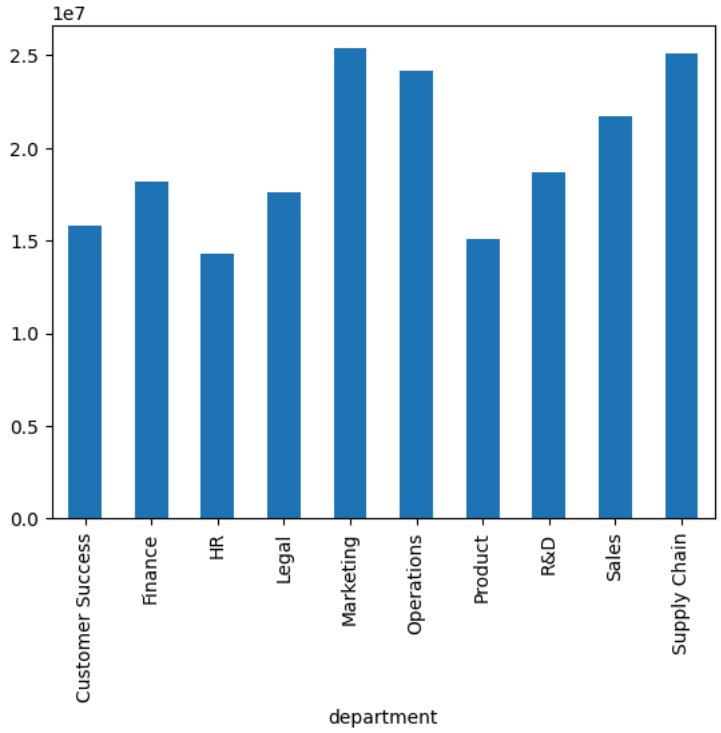

```
# How many projects are in each project status (Active, Completed, Proposed)?
sns.countplot(data=AI_Project_Master, x="project_status")
```

```
<Axes: xlabel='project_status', ylabel='count'>
```



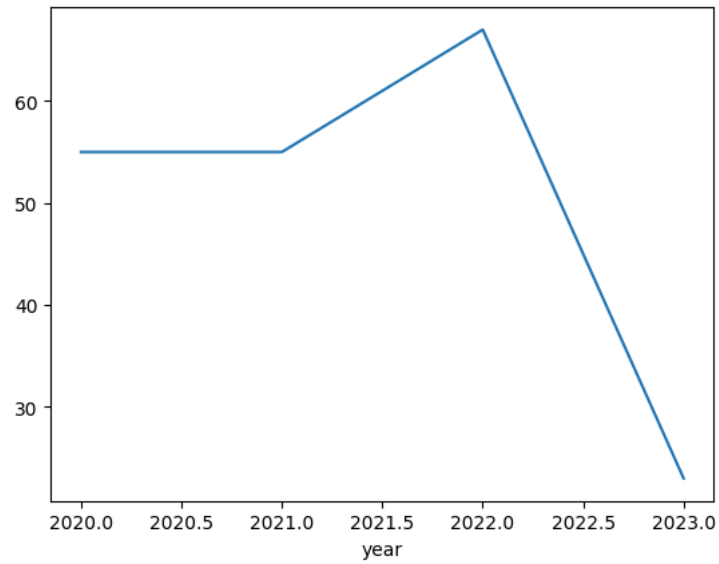

```
# Which department has the highest total project cost?

dept_cost = AI_Project_Master.groupby("department")["project_cost"].sum()
dept_cost.plot(kind="bar")
```

```
<Axes: xlabel='department'>
```



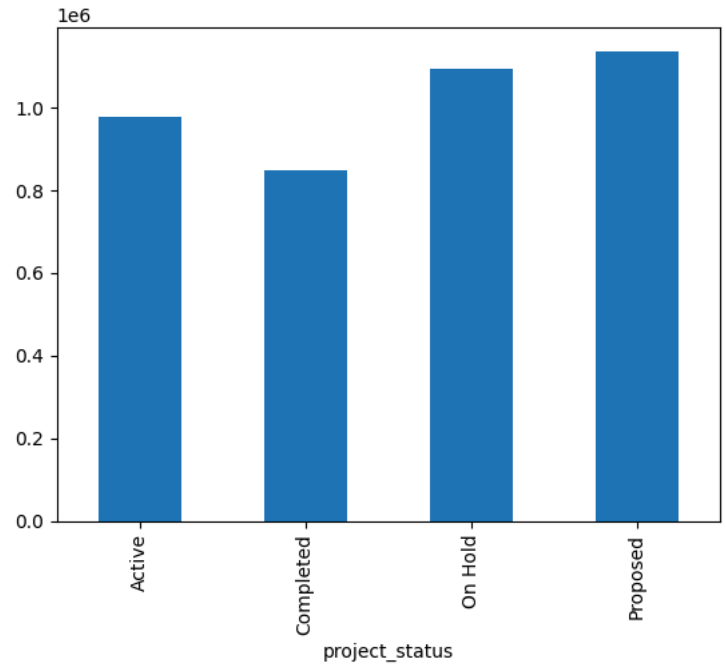

```
# What is the trend of project start dates over time? (Are projects increasing per year?)
AI_Project_Master["year"] = AI_Project_Master["start_date"].dt.year
AI_Project_Master["year"].value_counts().sort_index().plot(kind="line")
```

```
<Axes: xlabel='year'>
```

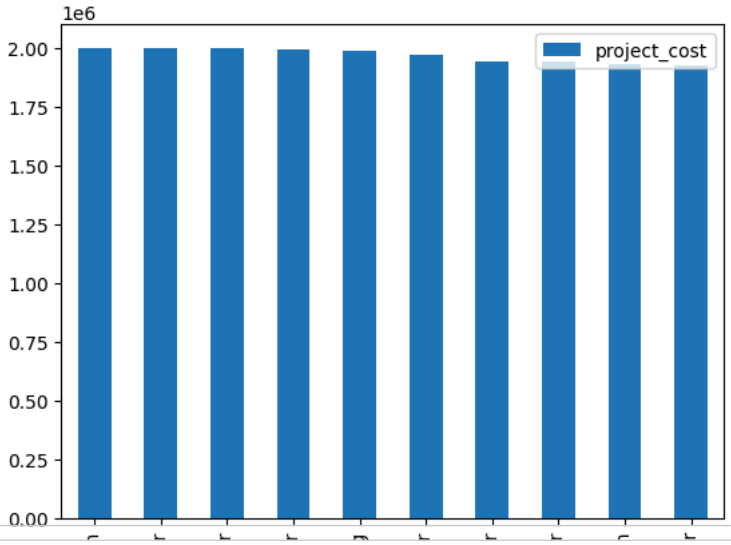



```
## What is the average project cost by project status?

avg_cost = AI_Project_Master.groupby("project_status")["project_cost"].mean()
avg_cost.plot(kind="bar")
```

```
<Axes: xlabel='project_status'>
```




```
# Which projects had the highest cost?

top10 = AI_Project_Master.nlargest(10, "project_cost")
top10.plot(x="project_name", y="project_cost", kind="bar")
```




```
# Relationship between project duration and project cost
AI_Project_Master["duration_days"] = (AI_Project_Master["end_date"] - AI_Project_Master["start_date"]).dt.days
AI_Project_Master.plot(kind="scatter", x="duration_days", y="project_cost")
```

```
<Axes: xlabel='duration_days', ylabel='project_cost'>
```