"Task-Flow: A Comprehensive Task Management System"

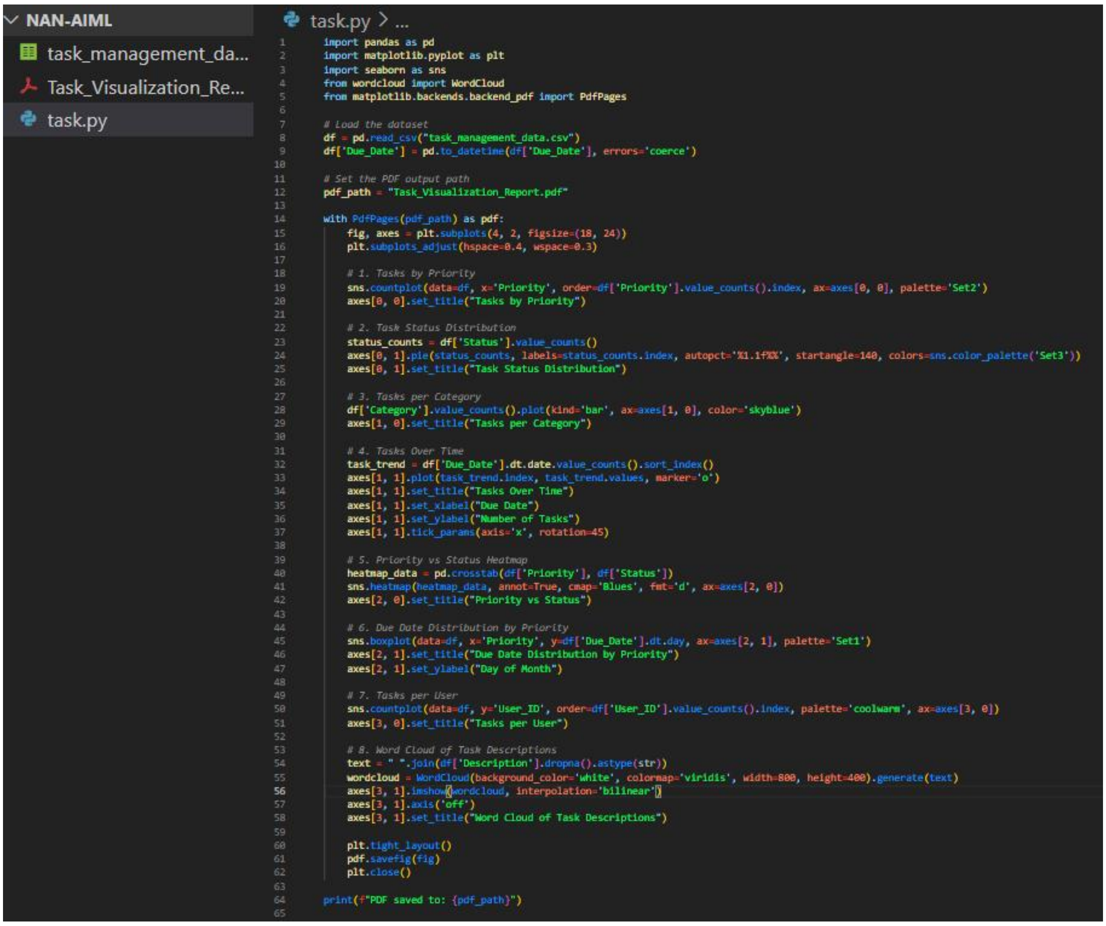
Data Visualization

Names: **Sec: 1**

Nandini Nagar (2320030397)

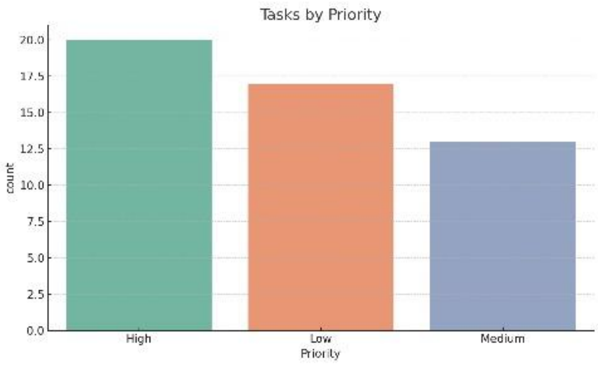
R. Akhilesh (2320030369)
Lahari (2320030306)

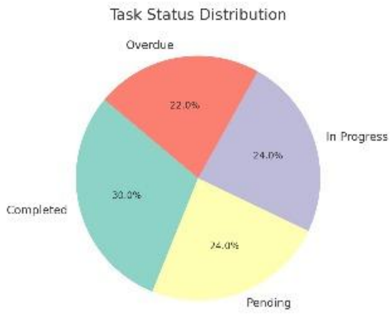
Code:

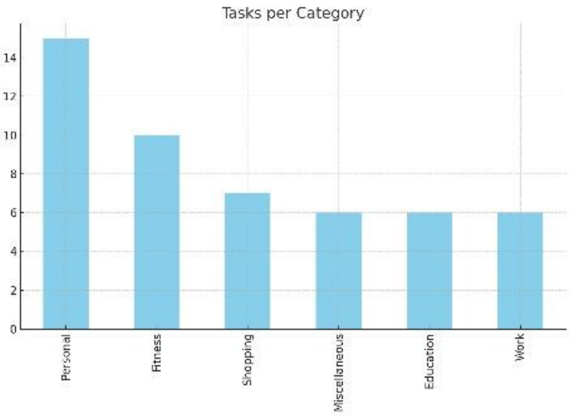


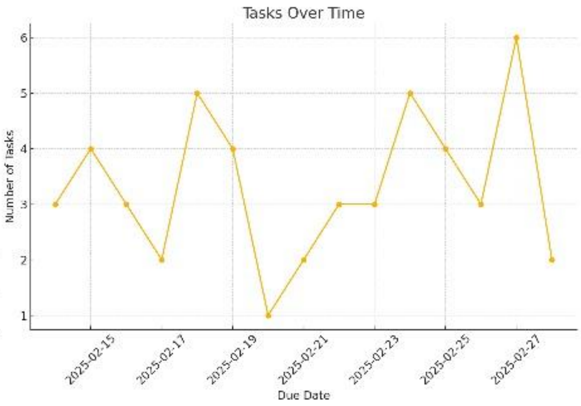
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
from matplotlib.backends.backend_pdf import PdfPages

# Load the dataset
df = pd.read_csv("task_management_data.csv")
df['Due_Date'] = pd.to_datetime(df['Due_Date'], errors='coerce')

# Set the PDF output path
pdf_path = "Task_Visualization_Report.pdf"

with PdfPages(pdf_path) as pdf:
    fig, axes = plt.subplots(4, 2, figsize=(18, 24))
    plt.subplots_adjust(hspace=0.4, wspace=0.3)

    # 1. Tasks by Priority
    sns.countplot(data=df, x='Priority', order=df['Priority'].value_counts().index, ax=axes[0, 0], palette='Set2')
    axes[0, 0].set_title("Tasks by Priority")

    # 2. Task Status Distribution
    status_counts = df['Status'].value_counts()
    axes[0, 1].pie(status_counts, labels=status_counts.index, autopct='%1.1f%%', startangle=140, colors=sns.color_palette('Set3'))
    axes[0, 1].set_title("Task Status Distribution")

    # 3. Tasks per Category
    df['Category'].value_counts().plot(kind='bar', ax=axes[1, 0], color='skyblue')
    axes[1, 0].set_title("Tasks per Category")

    # 4. Tasks Over Time
    task_trend = df['Due_Date'].dt.date.value_counts().sort_index()
    axes[1, 1].plot(task_trend.index, task_trend.values, marker='o')
    axes[1, 1].set_title("Tasks Over Time")
    axes[1, 1].set_xlabel("Due Date")
    axes[1, 1].set_ylabel("Number of Tasks")
    axes[1, 1].tick_params(axis='x', rotation=45)

    # 5. Priority vs Status Heatmap
    heatmap_data = pd.crosstab(df['Priority'], df['Status'])
    sns.heatmap(heatmap_data, annot=True, cmap='Blues', fmt='d', ax=axes[2, 0])
    axes[2, 0].set_title("Priority vs Status")

    # 6. Due Date Distribution by Priority
    sns.boxplot(data=df, x='Priority', y=df['Due_Date'].dt.day, ax=axes[2, 1], palette='Set1')
    axes[2, 1].set_title("Due Date Distribution by Priority")
    axes[2, 1].set_ylabel("Day of Month")

    # 7. Tasks per User
    sns.countplot(data=df, y='User_ID', order=df['User_ID'].value_counts().index, palette='coolwarm', ax=axes[3, 0])
    axes[3, 0].set_title("Tasks per User")

    # 8. Word Cloud of Task Descriptions
    text = " ".join(df['Description'].dropna().astype(str))
    wordcloud = WordCloud(background_color='white', colormap='viridis', width=800, height=400).generate(text)
    axes[3, 1].imshow(wordcloud, interpolation='bilinear')
    axes[3, 1].axis('off')
    axes[3, 1].set_title("Word Cloud of Task Descriptions")

    plt.tight_layout()
    pdf.savefig(fig)
    plt.close()

print(f"PDF saved to: {pdf_path}")
```

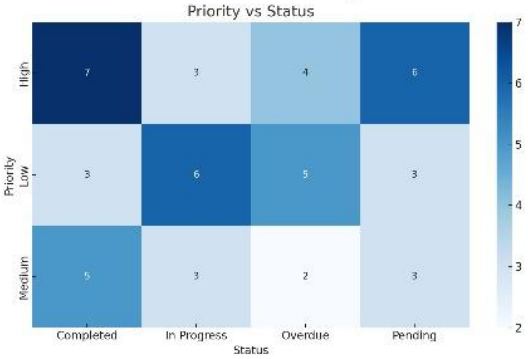
Data Visualization:

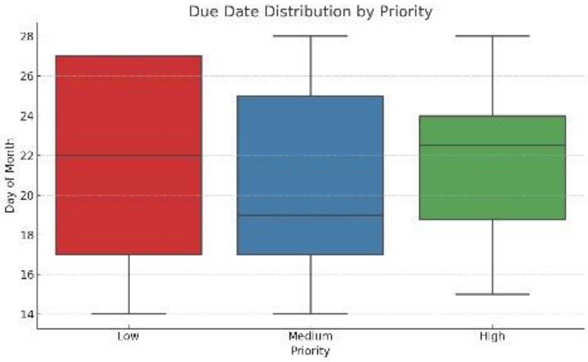

Tasks by Priority
count
High
Low
Medium
Priority

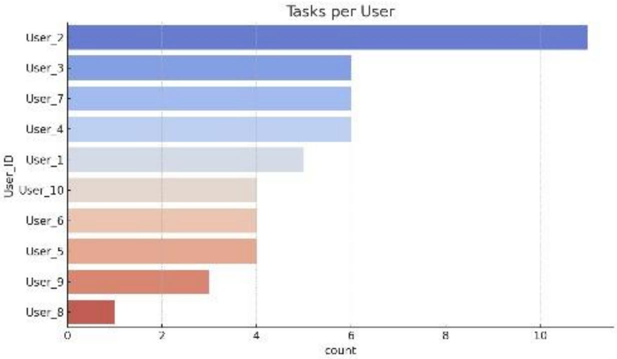

Task Status Distribution
Overdue
In Progress
Completed
Pending

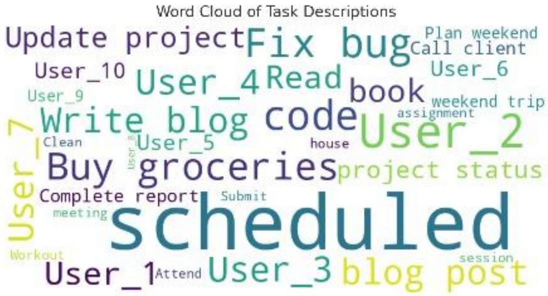

Tasks per Category
Personal
Fitness
Shopping
Miscellaneous
Education
Work


Tasks Over Time
Number of Tasks
Due Date


Priority vs Status
Priority
High
Low
Medium
Completed
In Progress
Overdue
Pending
Status


Due Date Distribution by Priority
Day of Month
Low
Medium
High
Priority


Tasks per User
User_ID
User_2
User_3
User_7
User_4
User_1
User_10
User_6
User_5
User_9
User_8
count


Word Cloud of Task Descriptions
Update project
Fix bug
Plan weekend
Call client
User_10
User_4
Read
User_6
User_9
book
weekend trip
User_7
Write blog
code
assignment
User_2
Clean
User_5
house
Buy groceries
project status
Complete report
Submit
meeting
scheduled
Workout
session
User_1
Attend
User_3
blog post