jira\_sprint\_plugin/

│

├── main.py # Main logic

├── jira\_utils.py # Jira API logic

├── reports.py # Report generation (Say/Do, Velocity, etc.)

├── charts.py # Plotting

├── config.yaml # PI, Sprint info etc.

├── requirements.txt

└── manifest.json # For Chrome extension (optional)

config.yaml

pi: "PI 2"

sprints:

- name: "Sprint 1"

start\_date: "2025-06-01"

end\_date: "2025-06-10"

- name: "Sprint 2"

start\_date: "2025-06-11"

end\_date: "2025-06-20"

- name: "Sprint 3"

start\_date: "2025-06-21"

end\_date: "2025-06-30"

- name: "Sprint 4"

start\_date: "2025-07-01"

end\_date: "2025-07-10"

- name: "Sprint 5"

start\_date: "2025-07-11"

end\_date: "2025-07-20"

requirements.txt

requests

pandas

pyyaml

matplotlib

jira\_utils.py

import requests

def get\_jira\_issues(jira\_url, token, board\_id, sprint\_id):

headers = {"Authorization": f"Bearer {token}", "Content-Type": "application/json"}

url = f"{jira\_url}/rest/agile/1.0/board/{board\_id}/sprint/{sprint\_id}/issue"

response = requests.get(url, headers=headers)

return response.json()["issues"]

reports.py

import pandas as pd

def process\_issues(issues):

data = []

for issue in issues:

fields = issue["fields"]

data.append({

"key": issue["key"],

"summary": fields["summary"],

"status": fields["status"]["name"],

"story\_points": fields.get("customfield\_10016", 0),

})

return pd.DataFrame(data)

def sprint\_summary(df):

committed = df["story\_points"].sum()

delivered = df[df["status"] == "Done"]["story\_points"].sum()

say\_do = round(delivered / committed, 2) if committed else 0

return committed, delivered, say\_do

charts.py

import matplotlib.pyplot as plt

def plot\_velocity\_chart(sprint\_data):

sprints = [s['sprint'] for s in sprint\_data]

delivered = [s['delivered'] for s in sprint\_data]

plt.bar(sprints, delivered, color='green')

plt.title("Velocity Chart")

plt.ylabel("Story Points Delivered")

plt.xlabel("Sprints")

plt.savefig("velocity\_chart.png")

def plot\_commit\_vs\_delivered(sprint\_data):

sprints = [s['sprint'] for s in sprint\_data]

committed = [s['committed'] for s in sprint\_data]

delivered = [s['delivered'] for s in sprint\_data]

x = range(len(sprints))

plt.bar(x, committed, width=0.4, label='Committed', align='center')

plt.bar(x, delivered, width=0.4, label='Delivered', align='edge')

plt.xticks(x, sprints)

plt.legend()

plt.title("Committed vs Delivered")

plt.savefig("committed\_vs\_delivered.png")

main.py

import yaml

from jira\_utils import get\_jira\_issues

from reports import process\_issues, sprint\_summary

from charts import plot\_velocity\_chart, plot\_commit\_vs\_delivered

JIRA\_URL = "https://yourdomain.atlassian.net"

TOKEN = "your\_api\_token"

BOARD\_ID = "123" # your Jira board ID

with open("config.yaml") as f:

config = yaml.safe\_load(f)

pi\_data = []

for idx, sprint in enumerate(config["sprints"], start=1):

print(f"Fetching {sprint['name']}...")

sprint\_id = idx # assuming sprint\_id matches index; replace with actual if needed

issues = get\_jira\_issues(JIRA\_URL, TOKEN, BOARD\_ID, sprint\_id)

df = process\_issues(issues)

committed, delivered, say\_do = sprint\_summary(df)

print(f"{sprint['name']} - Committed: {committed}, Delivered: {delivered}, Say/Do: {say\_do}")

pi\_data.append({

"sprint": sprint["name"],

"committed": committed,

"delivered": delivered,

"say\_do": say\_do

})

# Generate charts

plot\_velocity\_chart(pi\_data)

plot\_commit\_vs\_delivered(pi\_data)