Below are step-by-step instructions on how to modify the provided Python code and use it to generate Gantt (Waterfall) charts. This version also clarifies the need to input actual data into the code and avoid any randomization in the task data.

Step-by-Step Instructions:

1. Install Required Libraries

You need to install matplotlib and numpy libraries to run the code. Follow the installation steps based on your operating system.

Linux:

sudo apt-get update

sudo apt-get install python3-pip

pip3 install matplotlib numpy

macOS:

brew install python

pip3 install matplotlib numpy

Windows:

Install Python from python.org and then open Command Prompt:

pip install matplotlib numpy

2. Input Task Data

- **Replace the Placeholder Data**: Input actual data for each task in the Gantt chart rather than relying on randomization or placeholders.
- **Start and End Dates**: You must manually define the start and end dates for each task according to your project's timeline.

Example Tasks:

```
tasks = [
    {"name": "Week 1: Orientation and Shadowing", "start": datetime(2024, 12, 25), "end": datetime(2025, 1, 1)},
    {"name": "Weeks 2-4: Integration and Initial Contributions", "start": datetime(2025, 1, 2), "end": datetime(2025, 1, 22)},
    {"name": "Days 30-60: Independent Project Management", "start": datetime(2025, 1, 23), "end": datetime(2025, 2, 23)},
    {"name": "Days 60-90: Full Integration and Leadership", "start": datetime(2025, 2, 24), "end": datetime(2025, 3, 26)}
]
```

- **Modify Dates**: Adjust the start and end dates based on your specific onboarding plan and deadlines.
- **Example Date Format**: Use the datetime(year, month, day) format in Python.

3. Generate the Gantt Chart

Once the task data is defined, run the Python code to generate the Gantt chart.

4. Modify the Data as Needed

Change Dates or Tasks: To modify the chart, simply update the tasks array with the correct task names and start/end dates.

Add/Remove Tasks: You can add or remove tasks by adding new dictionaries or removing existing ones from the tasks list.

Step 5: Example Gantt Chart (Waterfall) Images

To help visualize the output of the script, I've attached example images of the Gantt charts that are generated by the code. These charts visually represent tasks over time, with each bar

indicating the duration of a task, and the labels providing a clear description of the tasks themselves.

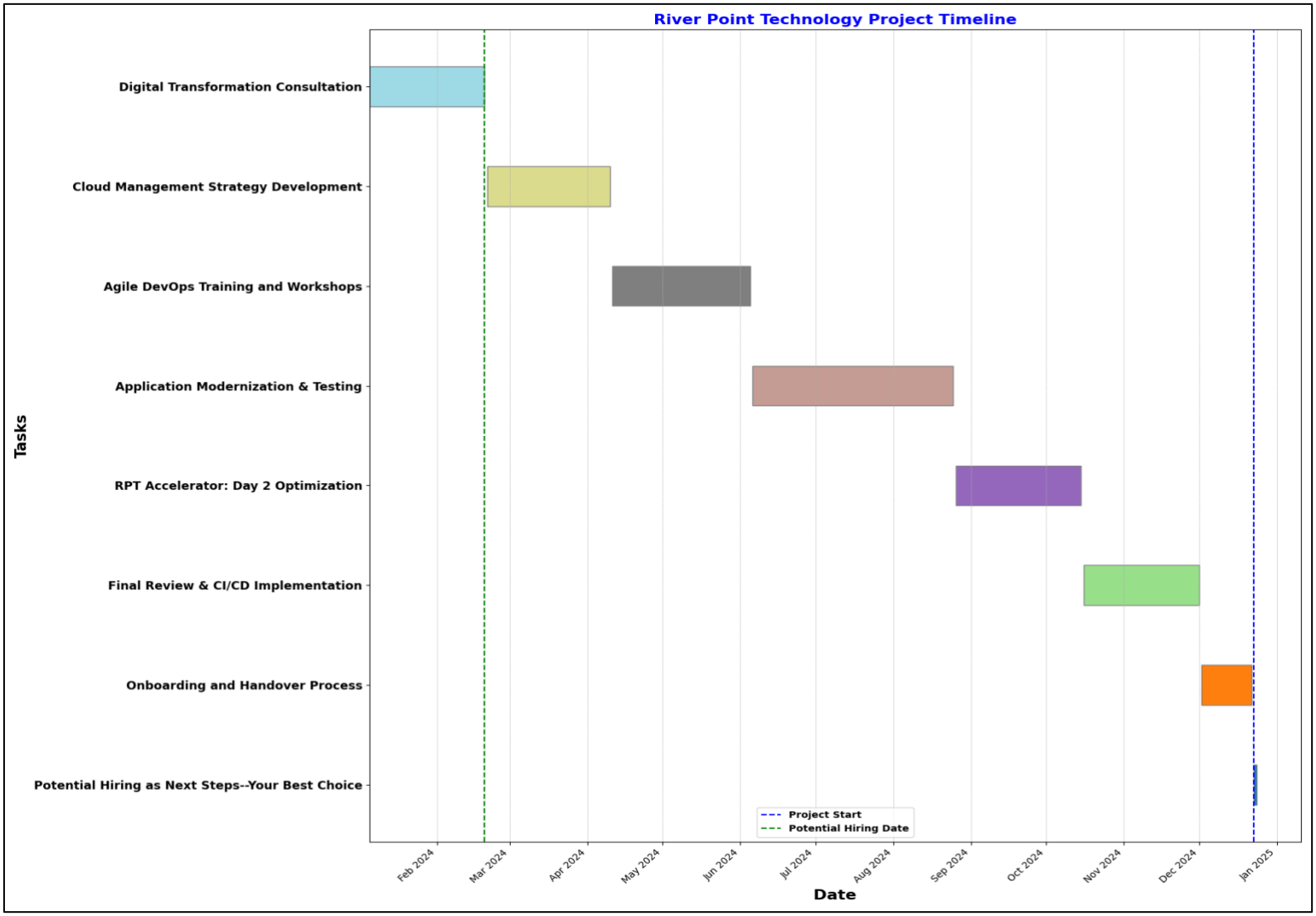
- **Example 1:** A project timeline using River Point Technology's strategic initiatives. The chart reflects phases like Cloud Management Strategy Development, Agile DevOps Training, and the RPT Accelerator for ongoing optimization.
- **Example 2:** An onboarding plan for new hires, showing week-by-week breakdowns for orientation, integration, project management, and leadership phases.

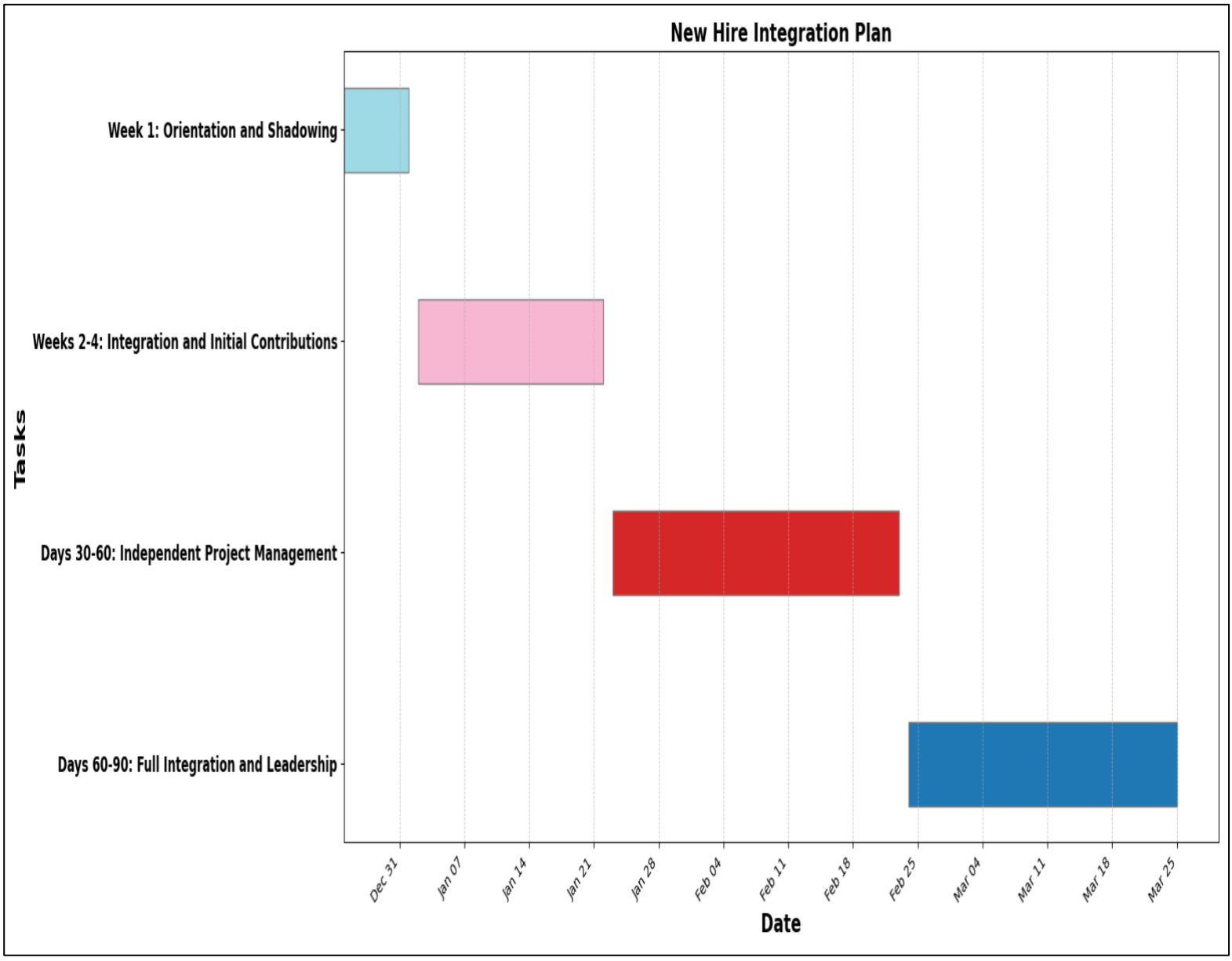
These examples show how project timelines can be broken down into actionable visual steps, providing an at-a-glance overview for project managers and stakeholders.

If you'd like to explore more, consider running the provided Python script on your own data to customize these charts to your specific projects.

Let me know if you'd like to explore adding a GUI for easier use or further customization options.

EXAMPLES


River Point Technology Project Timeline
Digital Transformation Consultation
Cloud Management Strategy Development
Agile DevOps Training and Workshops
Application Modernization & Testing
RPT Accelerator: Day 2 Optimization
Final Review & CI/CD Implementation
Onboarding and Handover Process
Potential Hiring as Next Steps--Your Best Choice
Tasks
Project Start
Potential Hiring Date
Feb 2024
Mar 2024
Apr 2024
May 2024
Jun 2024
Jul 2024
Aug 2024
Sep 2024
Oct 2024
Nov 2024
Dec 2024
Jan 2025
Date


New Hire Integration Plan
Week 1: Orientation and Shadowing
Weeks 2-4: Integration and Initial Contributions
Days 30-60: Independent Project Management
Days 60-90: Full Integration and Leadership
Tasks
Dec 31
Jan 07
Jan 14
Jan 21
Jan 28
Feb 04
Feb 11
Feb 18
Feb 25
Mar 04
Mar 11
Mar 18
Mar 25
Date