🧠 designPro Lab #002: User Flow Wireframe Generator

Lab Type: Project
Estimated Time: 60–120 mins
Skill Level: Intermediate

```
# Let's begin by printing your name to personalize the notebook
your_name = "dhimani benjamin"
print(f"Welcome to the lab, {your_name}!")
```

🔍 STAR Method Lab Prompt

Situation:
Situation: Many users struggle with efficiently managing their daily tasks, leading to feelings of overwhelm, missed deadlines, and decreased productivity. The current market solutions are often too complex, lack personalization, or have poor accessibility for diverse user needs.

Task:
Design a user-friendly and accessible task management interface that helps individuals prioritize, track, and complete their daily tasks with reduced cognitive load. This involves simulating the initial task entry and breakdown process, focusing on inclusive design principles.

Action:
I will leverage the Gemini API to generate personalized, simplified task breakdowns and suggestions based on various user input styles (e.g., voice, text, image). Following this, I will create a low-fidelity wireframe (using a simple drawing tool or text-based mockups) to visualize the core user flow: Task Entry (diverse inputs) → AI Breakdown → Prioritization (visual, accessible indicators) → Tracking (simple progress visualization).

Expected Result:
A comprehensive user journey map, outlining key interaction points and accessibility considerations, and a conceptual wireframe demonstrating a clear, intuitive, and inclusive task management application's primary screens, with an emphasis on the AI-assisted task breakdown feature.

✍️ Your Assignment

Use this section to complete your deliverable:

```
(Example Format)

- **Feature Flow**: Sign up → Onboard → Use Tracker → View Results
- **Tools Used**: Gemini for steps, Figma for wireframes
- **Steps**: 1) Welcome, 2) Input data, 3) Graph view, 4) Daily tips
- **STAR Output**:
   - S: Users drop off before value is seen
   - T: Improve first-use flow clarity
   - A: Designed 4-screen wireframe
   - R: Clearer UX shown via mock
```