🎤 Presentation Script — To-Do CLI App

👋 Introduction

"Hello, my name is **Austin Carlson**, and today I'll be presenting my Python project — a **Command-Line To-Do List Application**.

I built this project as part of my Software Engineering Bootcamp to demonstrate my understanding of Python fundamentals such as **lists, functions, error handling, and file I/O**.

This app lets a user manage tasks directly from the terminal, with tasks stored in a **Python list** during runtime and also saved to a **text file** so they persist between sessions. If the text file doesn't already exist, the program **creates it automatically** the first time you run the app."

📖 Overview of Assignment Objectives

"My project meets the following assignment requirements:

- **User Interface**: A menu-driven CLI that welcomes the user.
- **Core Features**: Add, View, Delete, and Quit.
- **User Interaction**: Uses `input()` to capture user choices and validate them.
- **Error Handling**: Catches invalid inputs, missing tasks, and incorrect menu selections using `try/except/else/finally`.
- **Organization**: Functions with clear names and docstrings.
- **Persistence**: Tasks are saved to a file called `tasks.txt` so they're not lost when the program exits. If `tasks.txt` isn't found, it's created automatically."

💻 Walkthrough Demo

1. Launch the app

"When I run the program, it prints a **welcome banner** and automatically shows my tasks.

- If I already have tasks saved, they're displayed sorted by importance: **High** → **Medium** → **Low**.
- If the list is empty, it shows a friendly message:
  'Your list is empty — add your first task from the menu!'
- And importantly, if the `tasks.txt` file doesn't exist yet, the program will **create it for me**, ensuring I always have a storage file ready."

---

2. Add a Task

"Now let's add a task.

- I select option **1 (Add a task)**.
- The app prompts me to enter a task description.
  I'll type: `Finish Bootcamp Assignment`.
- Next, it asks for importance: High, Medium, or Low.
  I'll choose **H for High**.

The program responds:
`✔ Added (HIGH): Finish Bootcamp Assignment`

And if I quit and restart the app, the task is still there because it's saved to `tasks.txt`."

---

3. View Tasks

"I'll choose option **2 (View tasks)**.
The app displays tasks sorted by importance with colored labels:

- 🔴 High in Red
- 🟡 Medium in Yellow
- 🟢 Low in Green"

---

4. Delete a Task

"Now let's delete a task.

- I choose option **3 (Delete a task)**.
- The app shows the task list with numbers.
- I'll enter the number of the task to delete, and it confirms with a message like:
  `🗑️ Deleted (HIGH): Finish Bootcamp Assignment`.

It then shows the updated list. If no tasks remain, it says:
`Your list is now empty.`"

⚠️ Error Handling Tests

Invalid Menu Choice

"I'll test by typing `7` at the main menu.
The program says:
`✖ Invalid choice. Please select 1, 2, 3, or 4.`
and stays in the menu loop."

Empty Task Add

"If I try to add a task but just press Enter without typing anything, the program says:
`✖ Task cannot be empty. Try again or C to cancel.`
It re-prompts until I type something valid or cancel."

Invalid Importance

"If I type `J` for importance, the program responds:
`✖ Invalid choice. Please enter H, M, L, or C.`
It re-prompts until I choose correctly."

Delete Invalid Task

"If I try to delete a task number that doesn't exist, the program says:
`✖ That task number doesn't exist. Try again or C to cancel.`
It won't crash — it just loops until I give a valid number or cancel."

✅ Conclusion

"In summary, this To-Do CLI app demonstrates:

- Menu-driven **user interface**
- **Core task management features** (add, view, delete)
- **Error handling** for invalid inputs
- **Persistent storage** in `tasks.txt` — and if the file doesn't exist, it's created automatically
- Clean, organized **Python code** with functions and comments

This project helped me reinforce Python fundamentals while building a real, interactive tool.

Thank you for watching my presentation!"