Final Project Report

Project Title: Fitness Workout Tracker

1. Formalized Proposal

Maintaining a consistent workout routine can be challenging. People often forget which exercises they completed, which muscles they worked, or how to structure their workouts efficiently. A digital solution can help users organize their exercises, track performance, and plan workouts to meet their fitness goals.

The **Fitness Workout Tracker** is designed to address this problem. It allows users to store exercises, create daily routines, and track progress. Exercises can be sorted by difficulty, duration, or muscle group. Users can build routines, follow them in order, and save their progress for later use.

To manage data efficiently, the project implements:

- **Binary Search Tree (BST):** Stores exercises alphabetically or by difficulty for fast retrieval.
- **Queue:** Maintains the order of exercises in a daily routine.
- **Insertion Sort:** Reorders exercises by duration or difficulty for display.

The GUI supports these functionalities through:

- A sidebar for workout categories (e.g., cardio, strength).
- Main display for exercises and routine details.
- Buttons for adding exercises, starting routines, sorting, saving, and loading.
- Input fields for exercise name, sets, reps, duration, muscle group, and difficulty.

Proposed Timeline:

- **Sprint 1:** Develop core classes (Exercise, Workout, WorkoutManager) and unit tests.
- **Sprint 2:** Design GUI layout and populate with sample exercises.
- **Sprint 3:** Integrate BST, queue, and insertion sort with GUI.
- **Sprint 4:** Test, validate functionality, polish GUI, and prepare demo.

2. Time and Change Logs

Week 1:

- **Accomplished:** Created initial GUI layout, sidebar, input fields, and buttons.
- **Blockers:** None major; data structures not yet implemented.
- **Next Steps:** Implement core classes and integrate BST and queue.

Week 2:

- **Accomplished:** Set up main classes and started connecting them to GUI. Users can begin adding and viewing workouts.
- **Blockers:** GUI updates weren't fully reflecting changes in workouts.
- **Next Steps:** Complete sorting, filtering, and smooth GUI interactions.

Week 3:

- **Accomplished:** Sorting and filtering fully integrated. GUI updates properly when exercises are added.
- **Blockers:** Minor sorting display issues, resolved.
- **Next Steps:** Polish GUI, clean up code, and test thoroughly.

Week 4:

- Finalized GUI, created presentation slides, and recorded presentation.

3. Lessons Learned

- **Scope & Changes:** The project scope remained consistent, focusing on integrating GUI with efficient data structures. The main change was improving the GUI responsiveness to reflect the underlying data structures in real time.
- **Blockers & Solutions:** GUI updates initially didn't reflect routine changes. This was solved by connecting GUI elements directly to the data structures (BST and queue).
- **Peer Review Adjustments:**
    - Made the routine builder more visible.
    - Considered showing exercise names in a dedicated column.
    - Acknowledged adding motivational messages for routines as a potential enhancement.

4. Code

The project source code is fully commented and follows style guidelines.

GitHub Repository: https://github.com/codewithrasa/FitnessWorkoutTracker.git

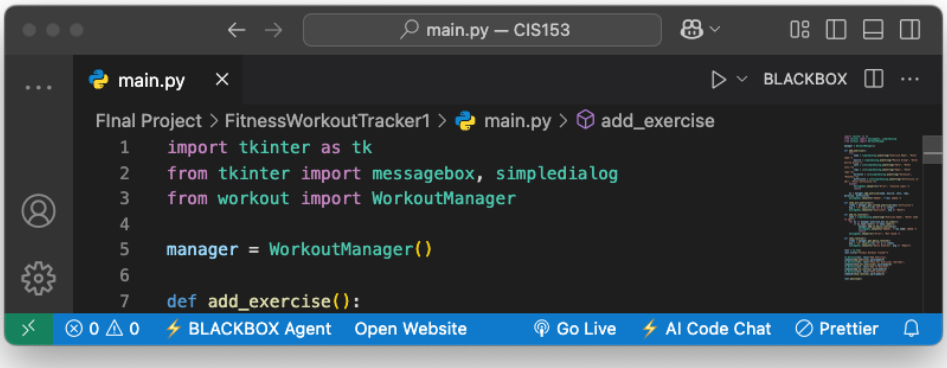
Key Files:

- exercise.py – Defines the Exercise class along with its attributes and validation.
- workout.py – Manages adding, editing, and deleting exercises as part of a
- workout_manager.py – Implements the Binary Search Tree, Queue, and sorting logic to organize exercises.

All code includes inline comments explaining logic, data structures, and algorithms

5. User's Manual

1. **Launching the App:** Run main.py in Python. Ensure all dependencies are installed.



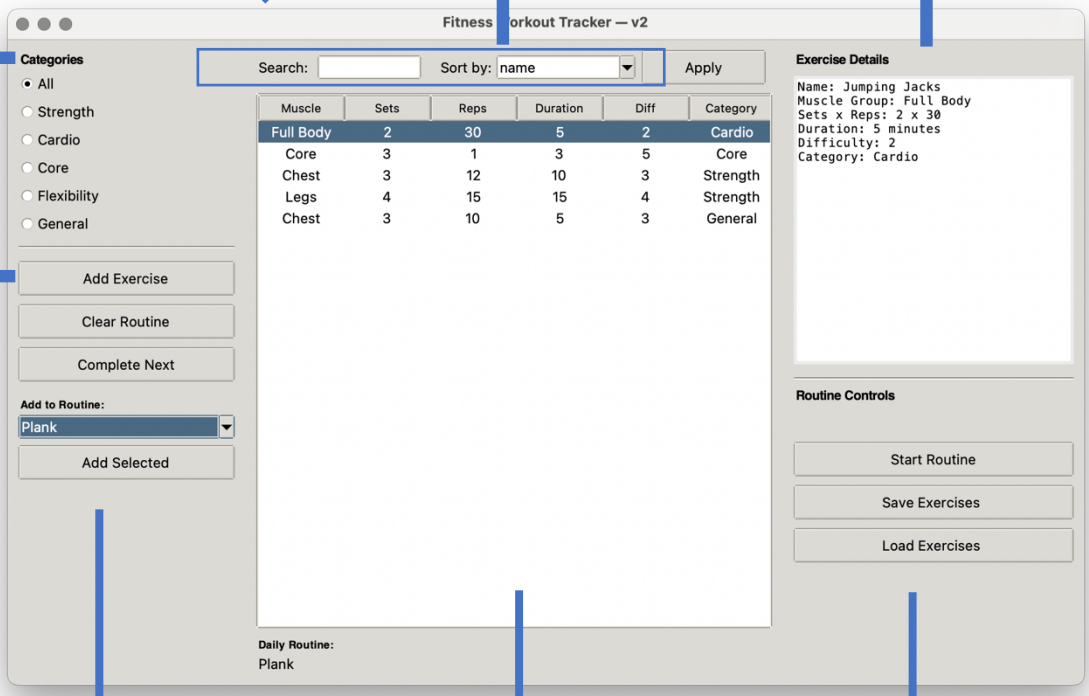
Thia is the main GUI

Search exercise and sort by Name, Duration and Difficulty.

You can see exercise details here after you add the exercise.

Filter by Categories

Add exercise Button

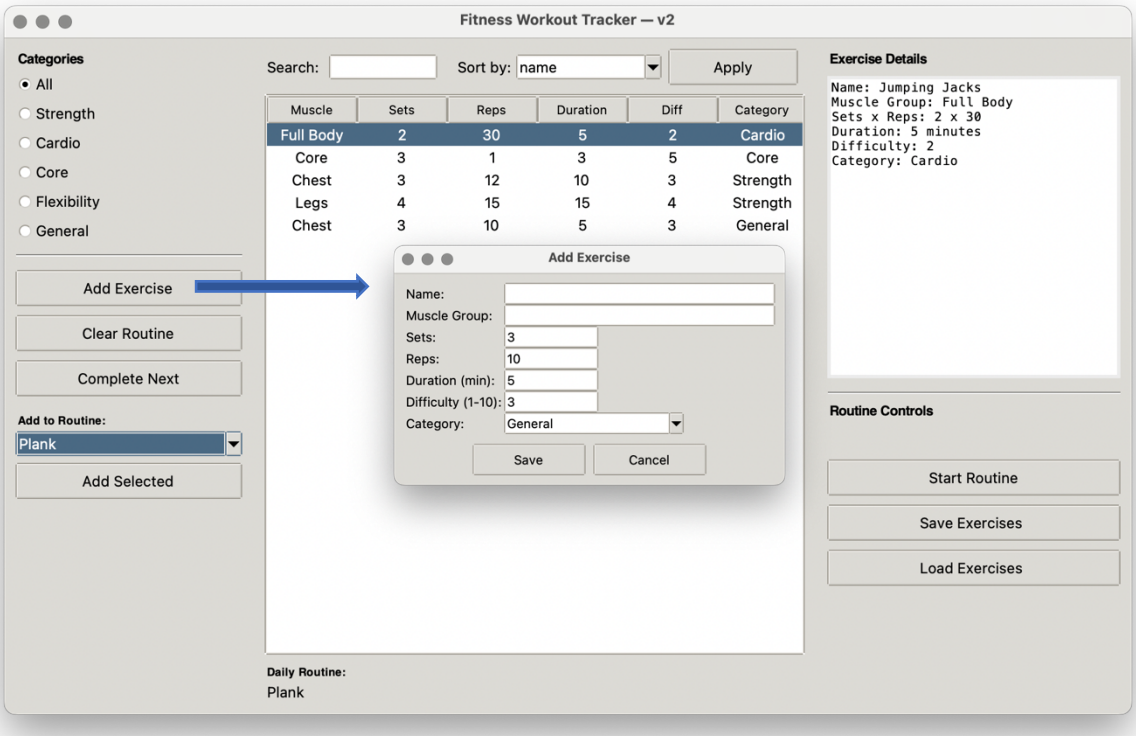


Select the existing exercise and added to daily routine

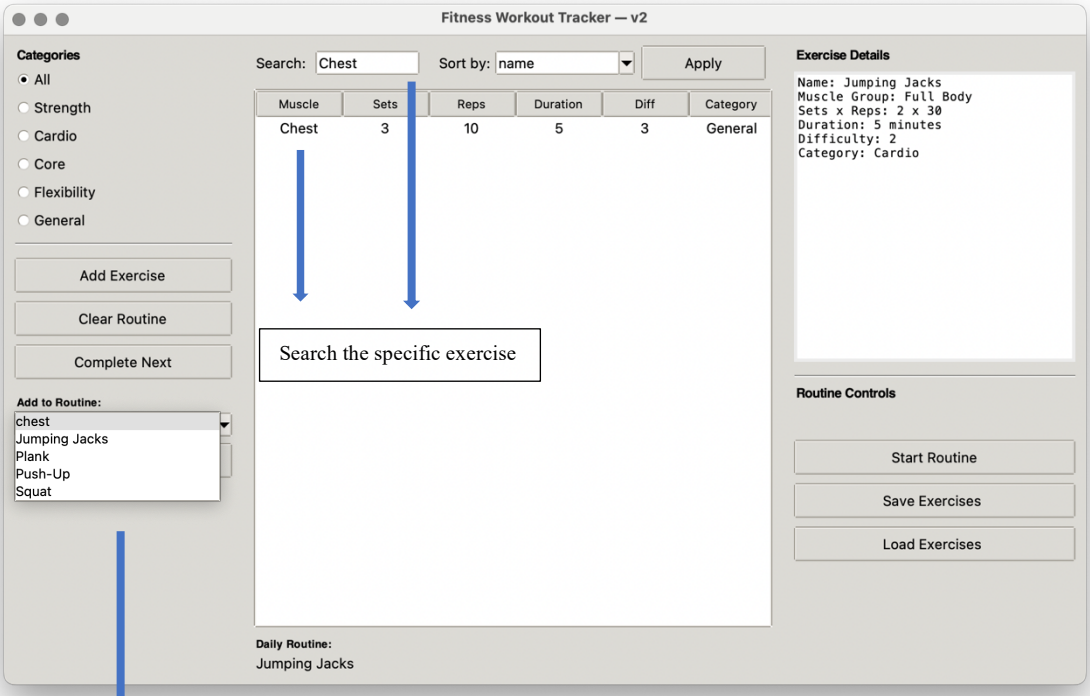
The section that we can see what exercise we have

Daily routine buttons

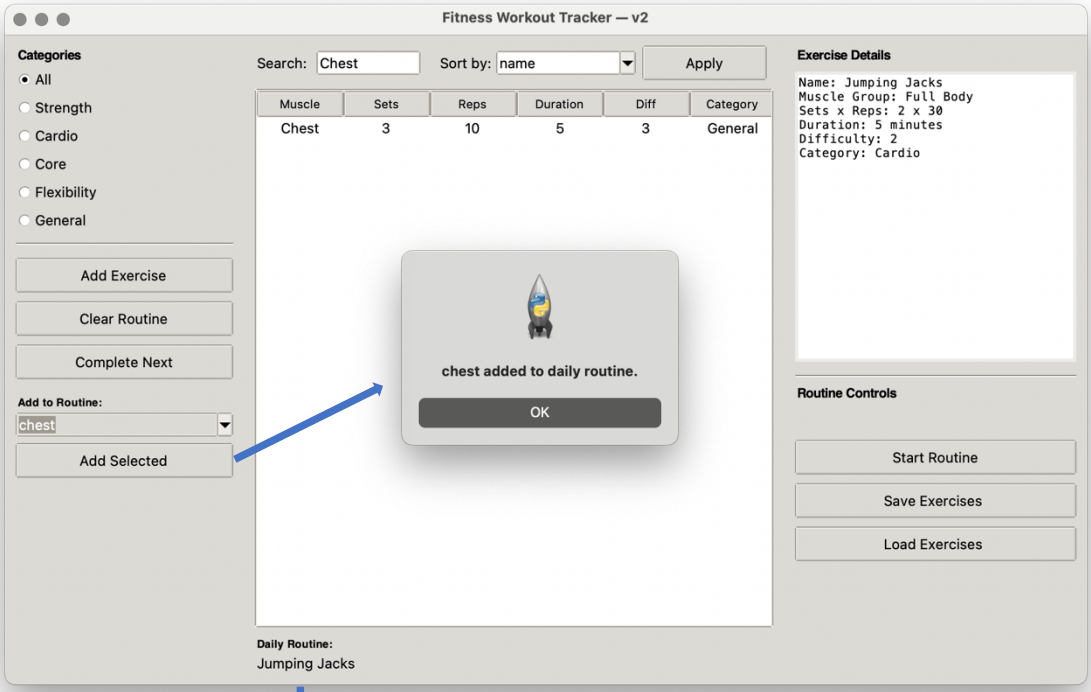
2. **Adding Exercises:** click on Add Exercise and then enter exercise details and click "Save" The BST stores them efficiently.



3. **Viewing Exercises:** Use the Drop down and filters to browse exercises.

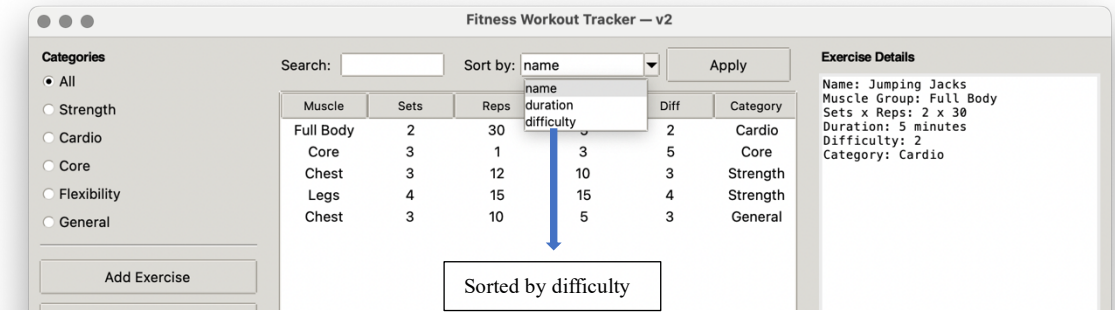


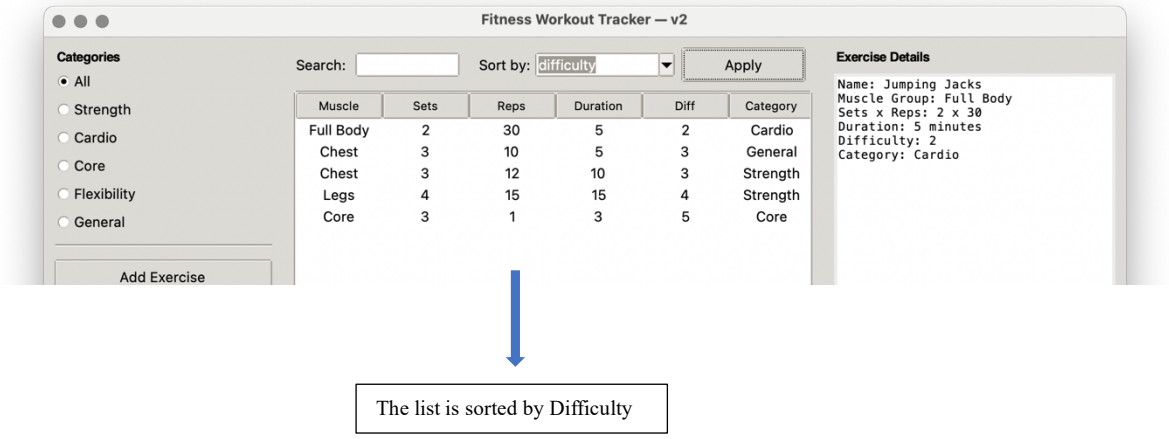
Select the existing exercise and added to daily routine



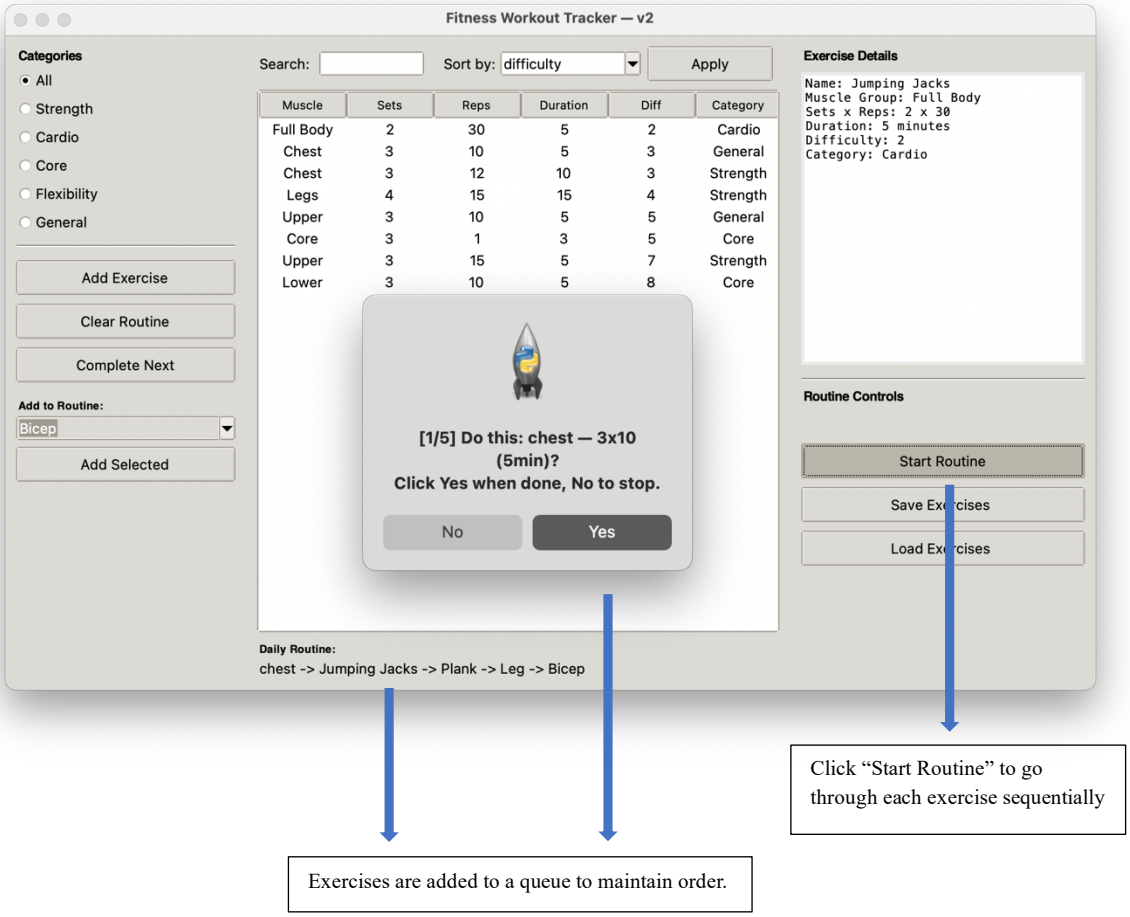
Showing what you have in our daily routine

4. **Sorting Exercises:** Click “Sort by Duration” or “Sort by Difficulty” to reorder the list.

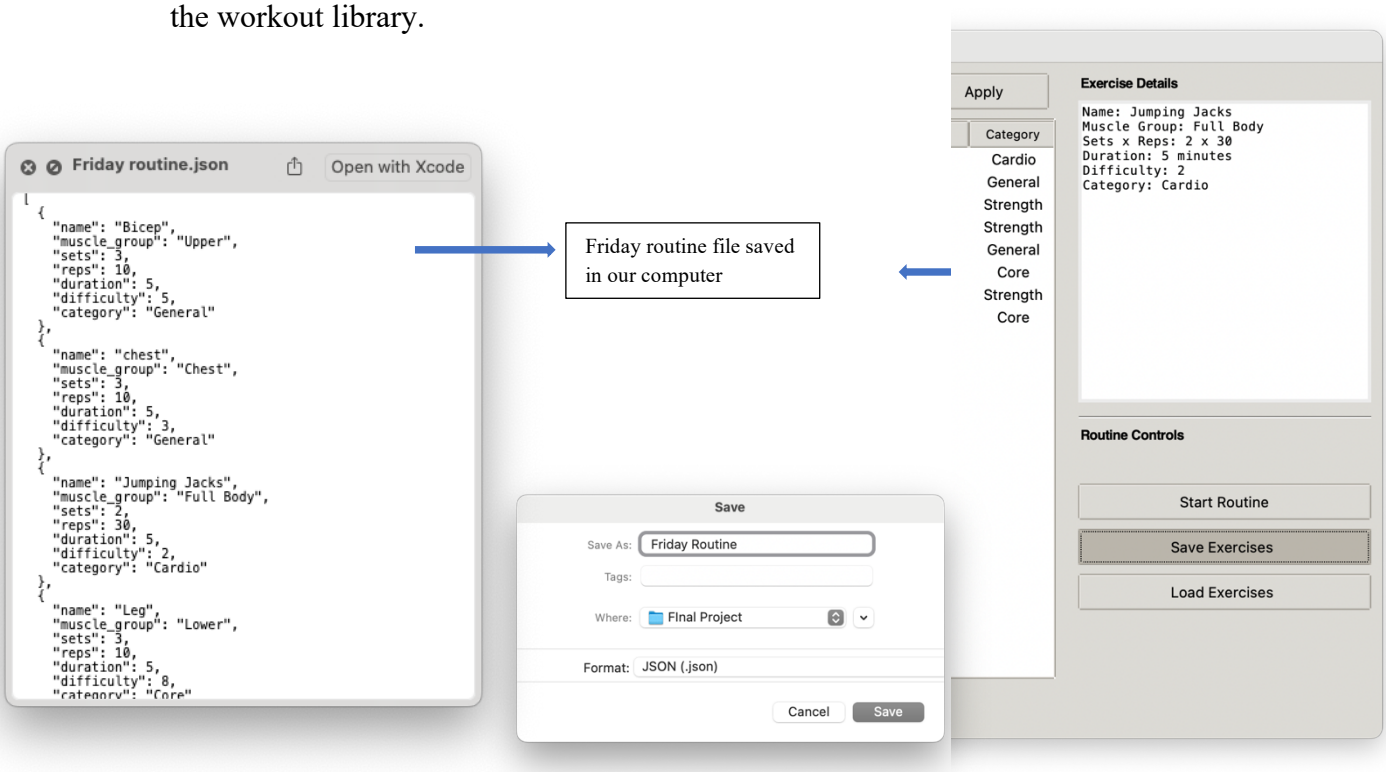




5. **Starting a Routine:** Click “Start Routine” to go through each exercise sequentially. Exercises are added to a queue to maintain order.



6. **Saving/Loading:** Use the “Save Exercises” and “Load Exercises” buttons to persist the workout library.



6. Conclusion/Summary

MERUSE Principles Applied:
The Fitness Workout Tracker follows MERUSE principles to make the code solid and easy to use. The main parts of the program like exercises, workouts, and the manager—are separate from the GUI, so everything is organized and reusable. Searching for exercises is fast with a Binary Search Tree, and routines follow the right order thanks to the queue. The code is clear and well-commented, the interface is easy to navigate, and the program handles mistakes without crashing. Overall, the design is clean and works efficiently.

Project Summary:
This project built a complete Fitness Workout Tracker where users can add exercises, create daily routines, and keep track of their workouts. The GUI shows exercises, lets users sort and filter them, and guides users through their routines step by step. Behind the scenes, the BST, queue, and sorting make sure everything runs smoothly and in the right order.

Future Versions:
Future improvements could add motivational messages while doing exercises, track progress with charts, and save workouts to the cloud so users can access them from any device. Other ideas include smarter workout suggestions, more filters, and connecting to fitness trackers.