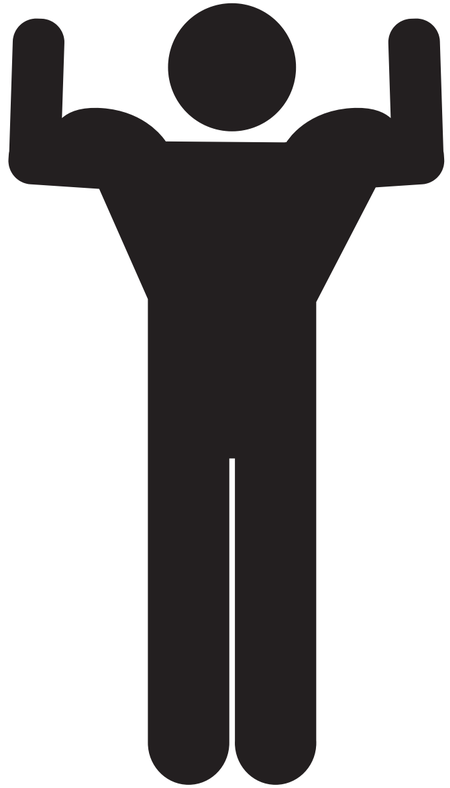
CS50 FINAL PROJECT

GYMLOG

- Federico Totaro Bessi
- Prato, Tuscany, Italy
- GitHub: FedericoTotaro
- EdX: federicototaro

- 22/12/2025



GYMLOG

Workout tracking is often:

- Unstructured
- Hard to review
- Repetitive to set up

GYMLOG

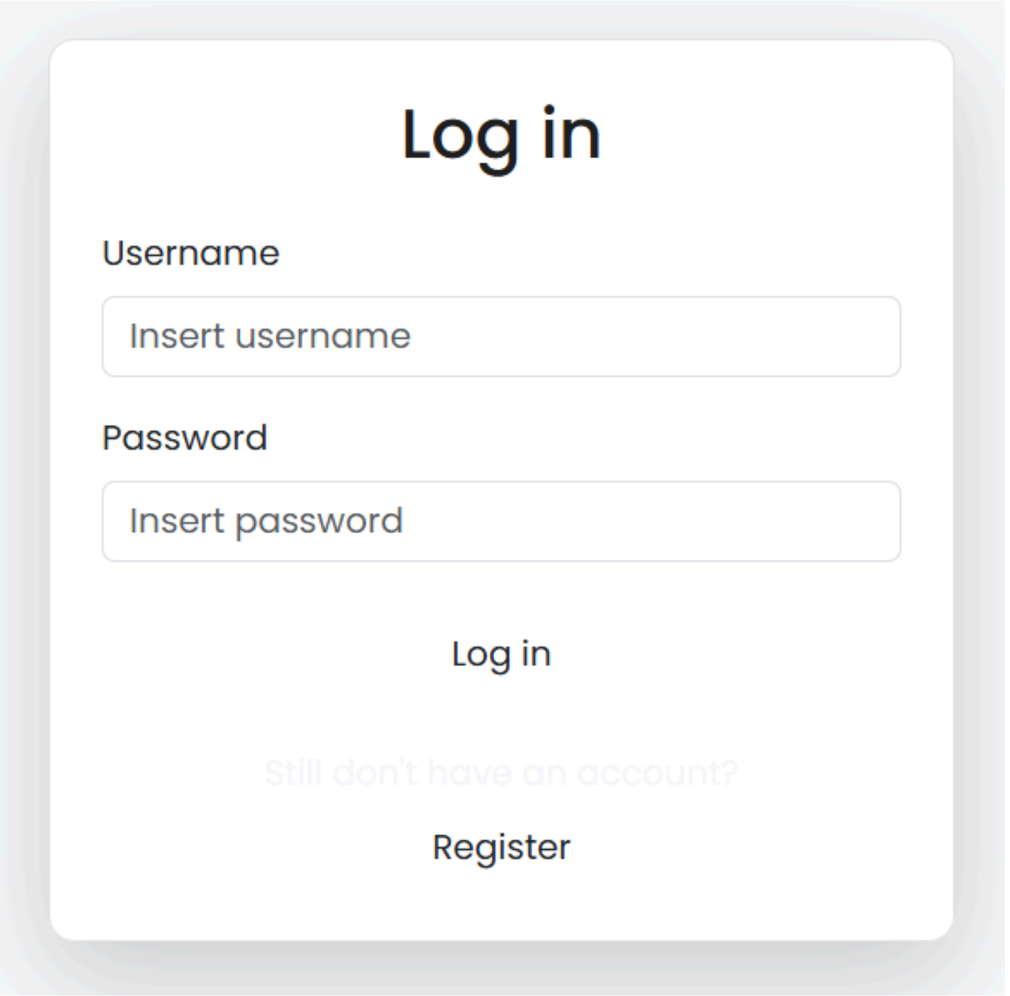
What GymLog does:

- Create training programs
- Define exercises and sets
- Train using saved programs
- Automatically store history

GYMLOG

How GymLog works:

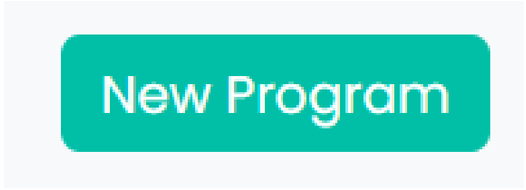
- Register / Login
- Create a program
- Add exercises
- Train
- Review history

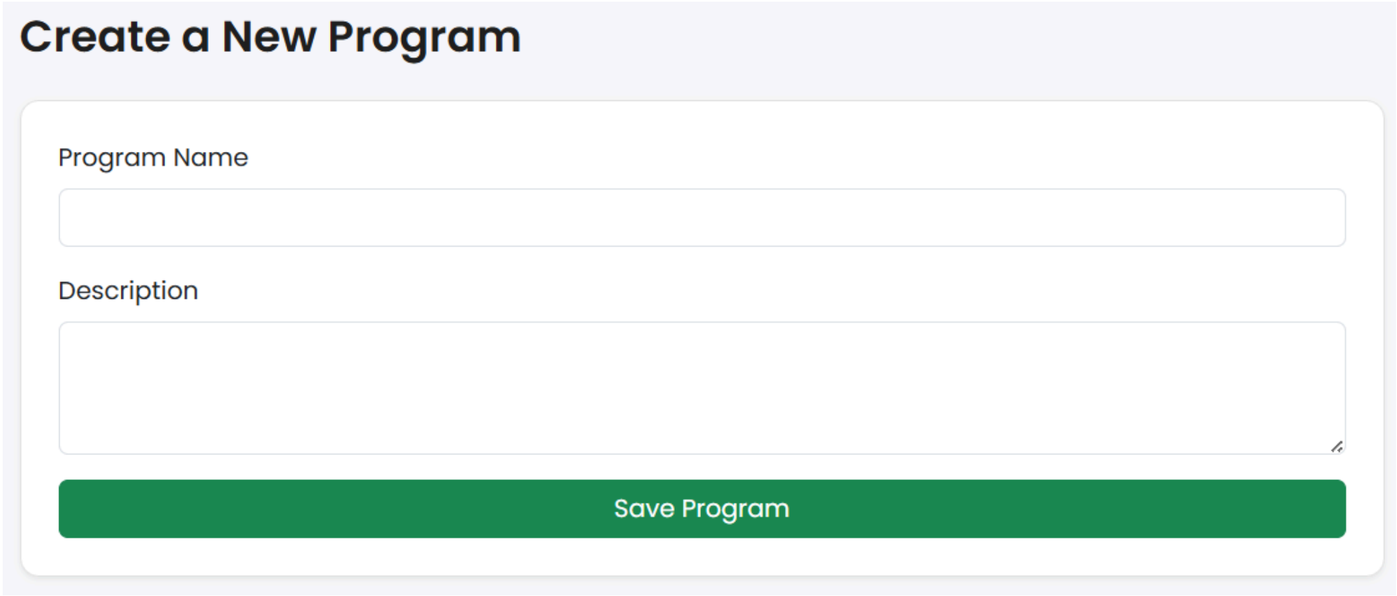


GYMLOG

How GymLog works:

- Register / Login
- Create a program
- Add exercises
- Train
- Review history

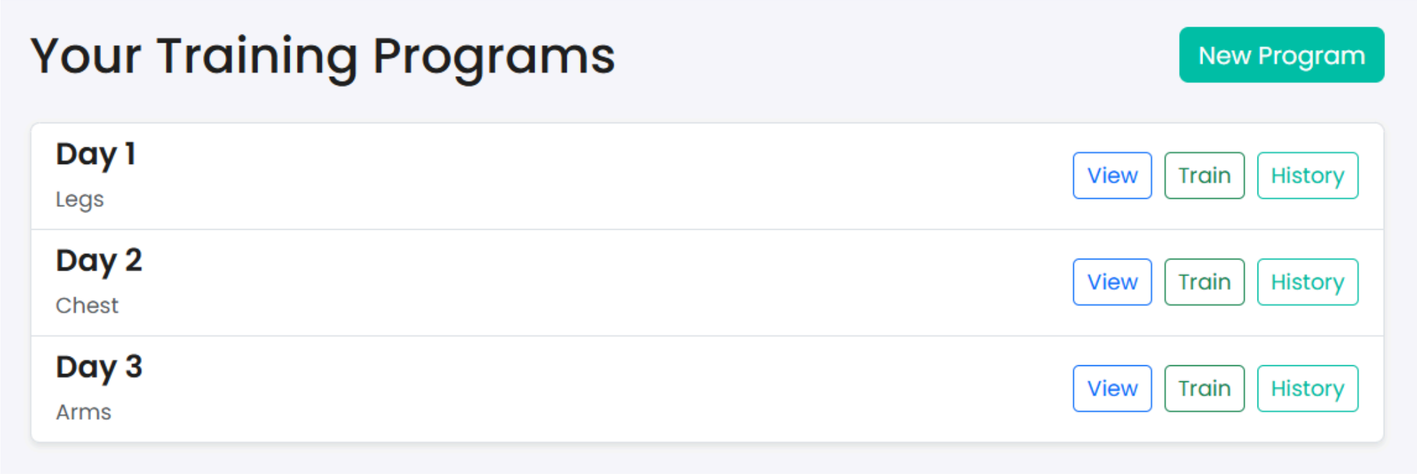




GYMLOG

How GymLog works:

- Register / Login
- Create a program
- Add exercises
- Train
- Review history



GYMLOG

How GymLog works:

- Register / Login
- Create a program
- Add exercises
- Train
- Review history

Squats

18/12/2025

Set	Reps	Weight (kg)
1	10	60.0
2	10	65.0
3	10	70.0

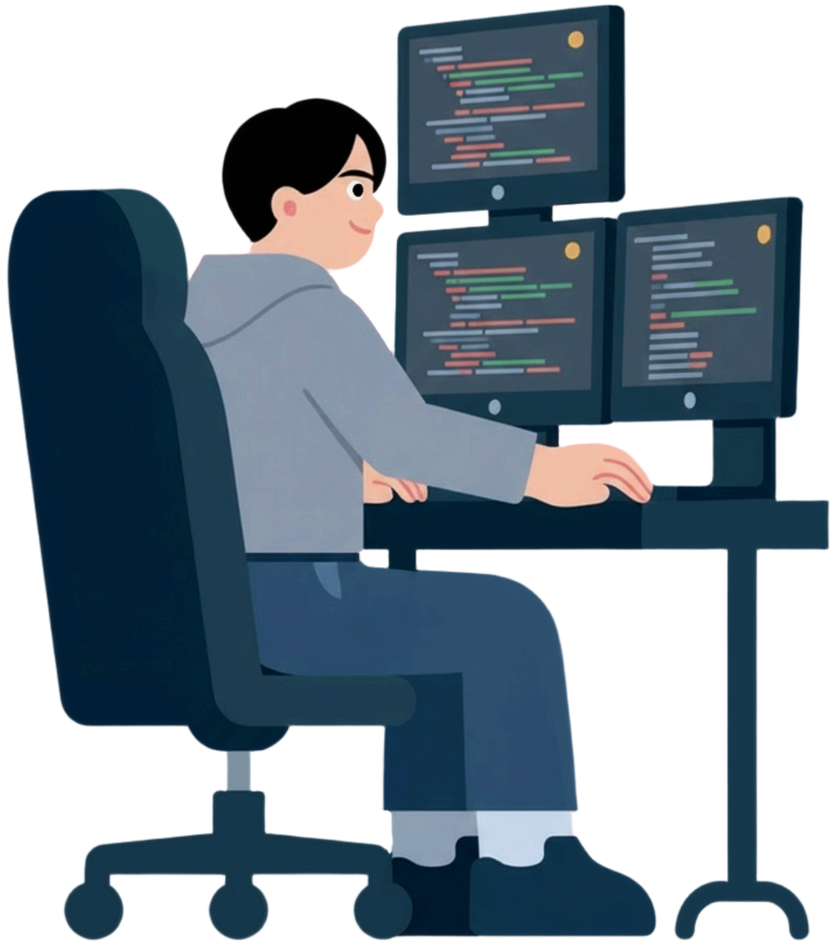
22/12/2025

Set	Reps	Weight (kg)
1	8	80.0
2	7	85.0
3	5	85.0

GYMLOG

Tools used:

- Python + Flask
- SQLite
- HTML + Jinja
- Bootstrap
- CS50 SQL library



GYMLOG

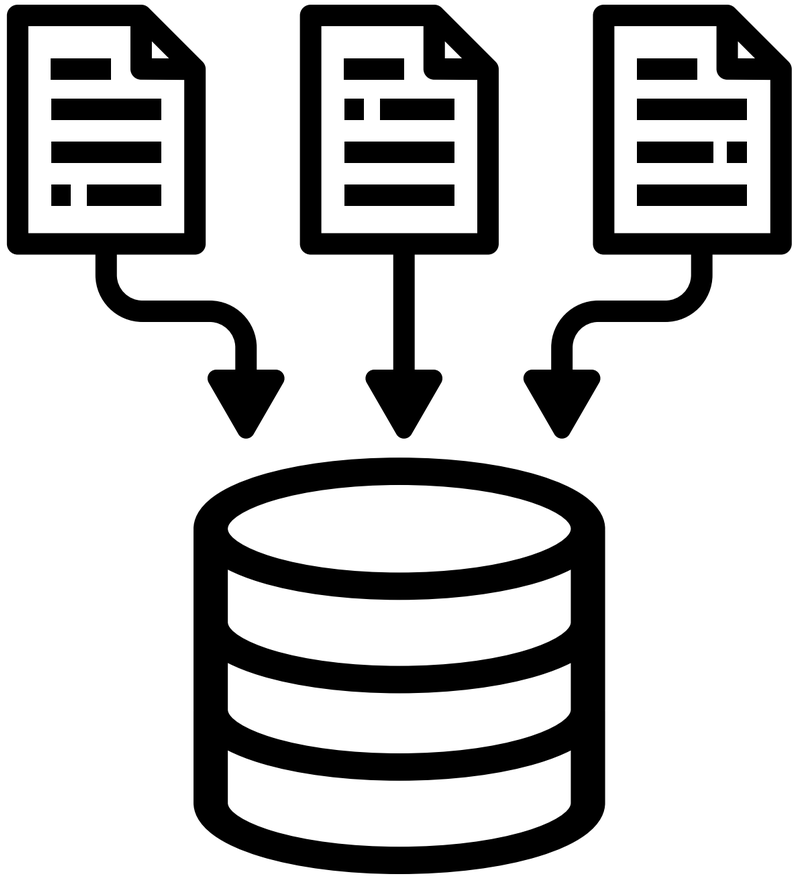
Data & Design Choices:

- Programs ≠ Workouts
- Sets logged individually
- User-specific data

GYMLOG

Challenges & Learning:

- Database relationships
- Flask routing
- Debugging state issues



GYMLOG

CONCLUSION

- Practical
- Extensible
- CS50 concepts applied

THANK YOU!

Federico Totaro Bessi
CS50