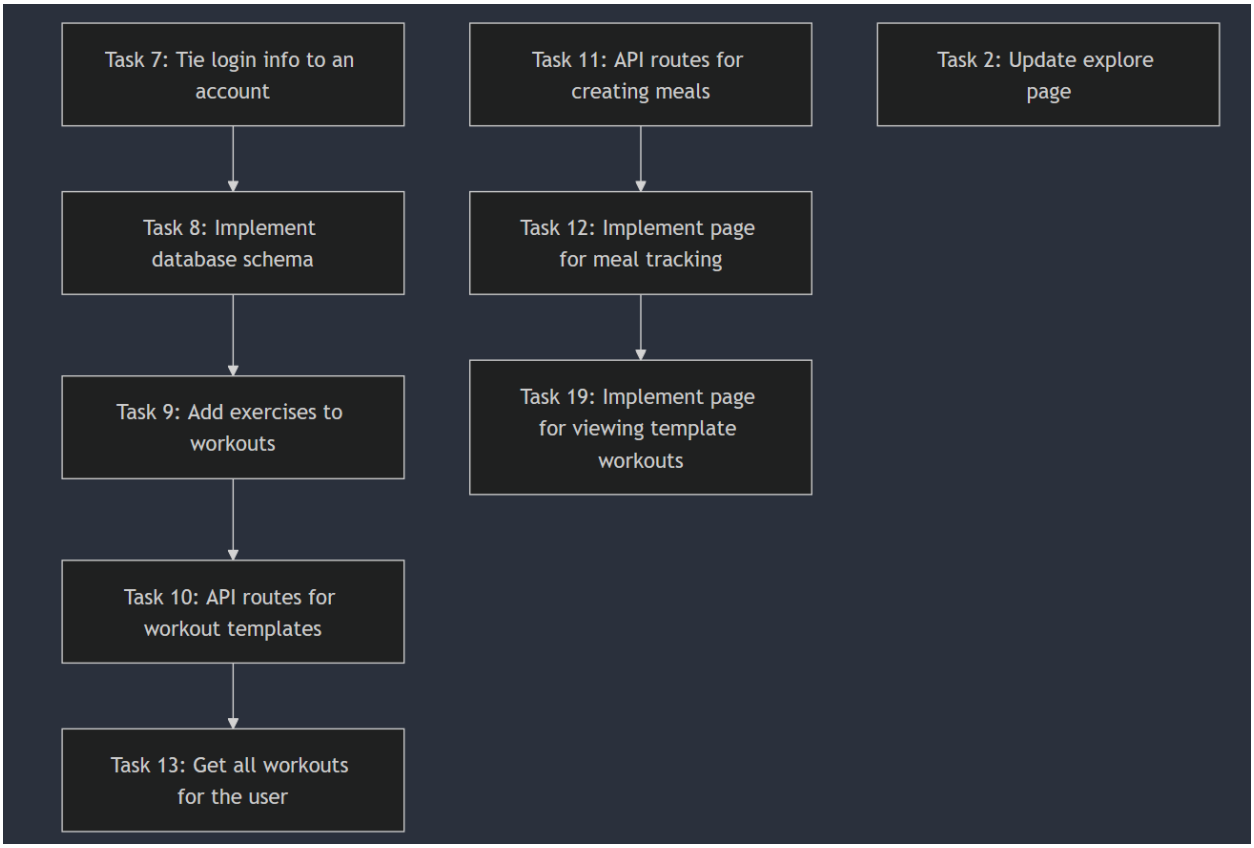
Sprint 2 Schedule

List of Tasks with Dependencies:

User Story	Task Description	Dependencies
(19)	Attach API to pull list of foods to add to a meal	None
	A new API route endpoint for fetching exercises	None
	A module with a service+controller	Attach API route
	Swagger documentation	A new API route endpoint
(7)	GET endpoint returning profile details including health record for a user	None
	Pull account details from DB and view it in profile page	GET endpoint
(7)	Update account details to DB with edit profile button	None
(2)(19)(20)	API route to update a Workout (includes exercises associated with workout)	None
	API routes for creating meals	API route to update workout
	API routes to get all workouts for user	API route to update workout
(2)	Implement page for performing activity (includes sets, reps, weight for each exercise)	None
	Testing	None
(7)	Tie login info to an account	None
(7)	Create initial account details to save to DB during registration	None
(12)	Implement page for meal tracking	None
(19)	Implement page for viewing template workouts	None
	Track protein consumption	Implement page for meal tracking
(2)(19)	Update explore page to record default values for each exercise	None

User Story	Task Description	Dependencies
(2) (19) (20)	Backend - Add exercises to workouts (includes default values for sets, reps, weight)	None
(2)(19)(20)	Backend API routes to create workout template, create workout from template, get workout by ID	None
(19)	Attach API to pull list of foods to add to a meal	None

Network Diagram:



Critical Path:

The critical path in this sprint includes tasks related to backend API development, schema implementation, and meal tracking. The tasks identified as critical include:

- **Task 7:** Tie login info to an account
- **Task 8:** Implement database schema in prisma.schema file
- **Task 9:** Add exercises to workouts
- **Task 10:** API routes to create workout templates, create workout from template, and get workout by ID

- **Task 11:** API routes for creating meals
- **Task 12:** Implement page for meal tracking
- **Task 19:** Implement page for viewing template workouts
- **Task 13:** Get all workouts for the user
- **Task 2:** Update explore page to record the default values for each exercise

These tasks are interconnected, and their timely completion is essential for the sprint's overall success. Any delay in these tasks could directly impact the completion of subsequent tasks and the overall sprint duration.

Sprint Management:

To keep this sprint on schedule, we focused on maintaining alignment with the critical path. The core tasks, such as implementing the database schema (Task 8) and adding exercises to workouts (Task 9), were prioritized. The dependencies between tasks were constantly reviewed during daily standups to ensure there were no blockers.

- **Daily Standups** helped us catch any potential delays early and resolve issues quickly.
- **Task Prioritization** was adjusted to ensure that tasks on the critical path were completed without delays.
- **Collaborative Effort:** Tasks with dependencies, such as implementing the meal tracking page (Task 12) and the API routes for creating meals (Task 11), were completed by multiple team members to prevent bottlenecks.

What Went Right:

- **Task Completion:** Most tasks were completed as planned, with the backend API (Task 10) and database schema (Task 8) being implemented on time. The team was able to coordinate well, especially with tasks dependent on the backend schema, like adding exercises to workouts (Task 9).
- **Team Coordination:** Effective collaboration between team members, especially for tasks with dependencies, ensured minimal delays.

What Went Wrong:

- **Delays in Task 19:** There were slight delays in Task 19 (Implement page for viewing template workouts). This was due to dependencies on Task 12 (meal tracking page), which took longer to finalize due to layout changes. The team was able to mitigate this by allocating additional resources to non-blocking tasks like backend work and API route implementation.

Lessons Learned:

- **Early Planning:** Ensuring that critical path tasks are identified early and their dependencies carefully tracked helps avoid delays in the sprint.
- **Resource Allocation:** By recognizing dependencies early, we were able to distribute resources effectively to mitigate any potential bottlenecks. Moving forward, prioritizing tasks with the most dependencies will help prevent delays like those experienced with Task 19.

Comparison of Planned vs. Actual Velocity:

We had planned to complete 10 tasks by the end of the sprint, with the expectation of completing 5 tasks per week. In reality, we completed 9 tasks, with Task 19 being delayed but still in progress.

The planned velocity was closely matched by our actual velocity, indicating that our sprint was relatively on track despite the minor delay in Task 19.

Conclusion:

In Sprint 2, we were able to successfully meet the majority of our goals, with the critical path tasks being completed on time. We identified areas for improvement, especially in task dependency management, and will carry these lessons into the next sprint to improve our efficiency.