**Chris Morrison   
Individual Report – Java Final Sprint SD12**

**Overview**

Throughout the development of this project I contributed extensively to the design and implementation of core system features and GitHub workflow. I was responsible for the main control flow in GymApp.java, the registration and login system, user-role routing, and the full set of features for both members and trainers. I also handled most of the GitHub activity, including branch creation, commit coordination, and initiating and reviewing many pull requests.

In addition to coding, I tested and validated functionality as features were completed, and helped troubleshoot bugs and research solutions. I regularly synchronized the database structure with code updates and helped ensure the success of the overall project.

**Branches & Feature Development Timeline:**

* chris-login-registration: This was one of the first branches created, where I implemented the login and registration logic. I used jBCrypt for password hashing and added role handling logic to route users based on their role input.
* chris-dashboard: After establishing basic login, I built out the main program loop inside GymApp.java, including designing the console UI, input validation, and the navigation structure that determines what happens when a user chooses to register, log in, or exit.
* chris-member-expense: As we moved deeper into role-specific features, this branch added the ability for members to view their total expenses. This required adding new methods in MembershipService, working with the database, and building a method for summing totals per user.
* chris-trainer-expense: Similar to member functionality, this allowed trainers to purchase memberships from their dashboard. It also required updates to trainer-specific menus.
* chris-trainer-class: This was a more involved branch focused on allowing trainers to add, view, and delete workout classes tied to their user ID. It required full integration between WorkoutClassService, database logic, and the user interfact.
* chris-trainer-update: This completed the CRUD set for trainers. It enabled updating the type or description of an existing class. It encompassed multi-step scanner input and confirmation.
* chris-sql-updates: Here, I added sample data and updated the scripts.sql file.
* feature/chris-auth-fix: During testing, I found that role inputs were case-sensitive and broke the login flow. I created this quick patch to normalize roles using .toLowerCase().
* chris-dashboard-update: This branch focused on UI - menu language cleanup, prompts, error messaging, and formatting improvements.
* chris-member-expense-view: One of the final branches I worked on, where I added new menu options for members to view expenses dynamically and refactored some of the connected logic to improve the scanner handling.

**Challenges Faced:**

* **Login routing bug**: User roles like "admin", "Admin", and "ADMIN" were not matching correctly. This broke login flows and required normalization.
* **Scanner logic issues**: We encountered the .nextInt() and .nextLine() bug, which caused unexpected behavior in menus after numeric input.
* **Service & database coordination**: Making sure our service layer methods (getWorkoutClassesByTrainerId for example) matched table columns/data types in PostgreSQL after some SQL exceptions early on.
* **GitHub sync issues**: Midway through the project, the repo got out of sync due to some merges and rebases not being done in the right order. Fixing it required carefully resolving merge conflicts and manually pulling/merging in changes. This took an unexpectedly huge amount of time!
* **Scanner reuse and user flow**: Because most functionality was nested inside menus and loops, we had to be careful about where and how we handled scanner.nextLine() and validation prompts.