Demo Script

1. Introduction

Speaker:

“Hey everyone, welcome to our demo of Fitquest! We’ve designed this as a Duolingo-style fitness platform that turns workouts into a fun, interactive game. Today, we’ll walk through it.

1.1 Functional Requirements

1. Authentication
   - Users create accounts and log in with two-factor verification via email.
2. Workout
   - Choose from various routines, track calories, and see progress over time.
3. Gamification
   - Earn points, level up, and see a confetti animation if you hit a new level.
4. Social & Notifications
   - Get reminders to work out and share achievements on the leaderboard.
5. Integration
   - All workout and user data syncs with Firebase; the app fetches workout details via a REST API.

1.2 Non-Functional Requirements

1. Usability
   - Clean, friendly interface that’s easy for anyone to pick up.
2. Performance
   - Quick loading times and no big lags when switching features.
3. Reliability
   - Stable session tracking and consistent data in Firebase.
4. Scalability
   - Able to handle a growing user base and more workouts.
5. Security & Privacy
   - Encrypted credentials, secure logins, and proper handling of user data.

1.3 Use Case Overview

- Creating an account and logging in securely.
- Selecting or tracking workouts (beginner/intermediate/advanced).
- Receiving points for completed workouts, which reflect on a leaderboard.
- Viewing stats on the dashboard for calories and time spent.

That’s the big picture of what we’re offering.

---

2. Demo Part

2.1 Create Account & Login

Speaker:

“First up is account creation. Users enter an email (preferably Gmail) and password. We send a verification email to confirm identity. Once verified, they’re free to log in. If something’s off (like an invalid email or password), the system shows clear error messages. Everything about the user, including the verification status, goes to Firebase.”

2.2 Dashboard & Calories/Time Visualization

Speaker:

“After logging in, you’re taken to the dashboard, which shows:

- The calories you’ve burned.
- The time you’ve spent working out.
- Graphs to compare your activity over time.

These stats auto-refresh from Firebase whenever you complete a new workout.”

2.3 Workouts, Points, and Leveling Up

Speaker:

“On to workouts. We calculate calories burned via a REST API, factoring in user level (beginner, intermediate, advanced) to determine reps. You earn points when you finish a workout, which might lead to a level-up and a quick burst of confetti. Points and levels get updated in Firebase, so your progress is always up to date.”

2.4 Leaderboard & Firebase Updates

Speaker:

“Finally, we have a leaderboard to keep things competitive. If you pass someone in points, you’ll see it right away because the data updates in real time. This makes the whole experience more motivating—nobody wants to stay at the bottom!”

---

3. Testing Part

3.1 Create Account – Sequence Diagram

Speaker:

“Let’s peek at how we handle registrations behind the scenes:

1. User inputs details, taps ‘Create Account.’
2. RegistrationInterface checks all fields; if invalid, user sees an error message.
3. RegistrationController ensures the email isn’t used, and if it’s new, sends a verification email.
4. Once verified, we store everything in Firebase, then move the user to ‘What’s Your Goal?’.”

Black Box Testing

- “We tested things like empty/incorrect fields, weak passwords, and duplicate emails. Each invalid scenario triggered the correct error, while valid data went through fine.”

White Box Testing

- “We did basis path testing to make sure every possible code branch got run, like handling good input (straight to success) vs. repeated email (error).”

3.2 Login – Sequence Diagram

Speaker:

“Logins are pretty straightforward:

1. The user enters credentials.
2. If everything checks out (via AuthenticationController), they go to the dashboard.
3. If not, they get an error like ‘Wrong password’ or ‘No user found.’”

Black Box Testing

- “We tried valid credentials, invalid email/password, and empty inputs. Each test behaved as expected—successful sign-in or an error.”

White Box Testing

- “Same deal with basis path: one path for correct input, another for invalid input, etc. So we know our login logic works every time.”

3.3 Workout – Sequence Diagram

Speaker:

“For workouts:

1. User taps on a category (Abs, Chest, etc.).
2. The AppInterface requests routine info from ExerciseManager.
3. Data is fetched and sent back, listing the exercises, durations, and focus areas.
4. The user can start a workout, and those details update in Firebase.”

Black Box Testing

- “We made sure the user sees correct info and can start/pause/stop the workout with no weird bugs—like missing exercises or a timer that doesn’t stop.”

White Box Testing

- “Basis path testing again—one path tests normal use, another might handle skipping an exercise, and so on.”

3.4 History – Sequence Diagram

Speaker:

“To check past workouts:

1. User taps ‘History.’
2. The AppInterface calls HistoryManager for the records.
3. Data comes back from Firebase, showing each workout with date/time.

Black Box Testing

- “We checked if completed workouts show up chronologically with no missing details. If you have none, it should say so instead of showing an empty list with no explanation.”

White Box Testing

- “One path for user who has previous workouts, another for a user with zero workouts—both tested thoroughly.”

3.5 Dashboard Testing

Black Box Testing

1. Completing a Workout and Checking the Dashboard
    - The dashboard correctly displayed the newly completed workout with accurate calorie and time data.
2. Multiple Workouts in One Day
    - The system aggregated two workouts into a total of 92 kcal and 16.2 minutes without error.
3. Day With No Workouts
    - The dashboard accurately showed 0 calories and 0 minutes, and the graph updated properly to reflect no activity.
4. Calorie Calculation Accuracy
    - A 16-minute workout for an 80kg user displayed 90 kcal, which aligned with the expected range, confirming correct calculations.

White Box Testing

- Basis Path Testing
    - Path 1: Completing a single workout, checking calculations for a single dataset.
    - Path 2: Multiple workouts, confirming proper summation of calories/time.
    - Path 3: No workouts, ensuring the system handles empty data gracefully.

By traversing all these execution paths in the dashboard's flowchart, we covered every logical branch—from checking whether workout data exists to deciding how to display it. This let us verify that all conditions and calculations were correct internally.

Gamification (Leaderboard) Testing

Black Box Testing

1. Completing a Workout and Checking Leaderboard
    - After finishing a workout, the system immediately updated the user's points on the leaderboard.
2. Top Position Indicators
    - Gold, silver, and bronze medals displayed correctly for 1st, 2nd, and 3rd places.
3. Sorting Accuracy
    - The leaderboard accurately ranked users based on point totals (e.g., 11145 for 1st place, 5000 for 2nd, 39 for 3rd), confirming correct ordering.

White Box Testing

- Basis Path Testing & Sequence Analysis
    - The leaderboard’s cyclomatic complexity (4) showed we had multiple branching points for awarding medals, handling ties, and sorting ranks. We tested each path to confirm medals and ranks appeared correctly.
    - Data persisted reliably across sessions, ensuring point totals remained accurate even after app restarts.

That’s the general tour of our FitQuest, how it works, and how we tested it. We kept everything synced through Firebase and used thorough testing to catch bugs early. Enjoy checking it out!