

# SOLUTECH MID LEVEL MOBILE ENGINEER: FLUTTER CODE CHALLENGE

## Gamified Habit Tracker App

### Objective

Develop a Gamified Habit Tracker App in Flutter that works on both mobile (Android/iOS) and web. The project should demonstrate proficiency in Flutter development, responsive design, CI/CD pipelines, and modern development best practices that as a Mid Level Engineer you should be able to demonstrate.

Here are a list of UI guides (You create one that is more appealing, otherwise use these as guides):

- <https://dribbble.com/shots/18932568-Morin-Habit-Tracker-App>
- <https://dribbble.com/shots/18500320-Habit-Tracker-App-Design>
- <https://dribbble.com/shots/20235106-Fitlab-Fitness-Workout-Tracker-App>
- <https://dribbble.com/shots/21367995-Habit-Tracker-Mobile-IOS-App>

**Note:** We prioritize *good coding practices and UX* in this challenge. To stand out, showcase your skills in creating reusable patterns, implementing high-quality designs, great UX, writing clear and comprehensive documentation, applying test-driven development (TDD), and demonstrating a strong understanding of cross platform mobile development. You are encouraged to add any additional relevant features that enhance the app's functionality or user experience.

For this Challenge you can use a backend as a Service of your choosing (e.g. Firebase, Supabase, Appwrite) or better yet create your own backend to demonstrate how you would consume APIs in a production application.

Deadline: 28th February 2025

## Core Features

### 1. Authentication

- Implement secure sign-up, login, and logout using email and password.
- Add social login functionality with at least one provider (e.g., Google or Facebook) following best security practices.

### 2. Habit Management

- Create Habits: Allow users to create habits with details such as name, description (optional), frequency (e.g., daily, weekly), and start date.
- Edit Habits: Enable users to update existing habits.
- Delete Habits: Provide an option to remove habits.
- Track and Visualize Progress:
  - Show completion status in a calendar view.
  - Highlight streaks (e.g., "You've completed this habit for X consecutive days").
  - Display habit stats such as percentage completed and longest streak.

### 3. Gamification

- Stats to show how the user is fairing
- Habit Levels and XP: Award XP for completing habits and level up as progress is made.
- Rewards and Badges: Grant badges for reaching milestones like streaks, completing all habits in a day, or other achievements.
- Daily Challenges and Streak Rewards: Introduce optional challenges for bonus XP and unlock streak-based rewards.

### 4. Offline-First Functionality

- Use local storage (e.g., SQLite or Hive) to ensure functionality even without the internet.
- Ability to know which records have not yet been synchronized with the server.
- Sync data with the server automatically when the device is back online.
- I should be able to resume from another device after syncing is complete

### 5. Notifications and Reminders

- Provide reminders for pending habits using notifications.
- Allow users to set custom reminder times for individual habits.

### 6. Theming and Responsiveness

- Implement a responsive design suitable for mobile and web.
- Add a toggle for light and dark mode, ensuring the app is visually appealing in both themes.

## Technical Requirements & Evaluation Criteria

1. Code Quality
  - Clean, modular, and maintainable code following clean architecture principles.
  - Clear separation of concerns through layers and good design patterns i.e. it should be designed in such a way that switching a database or backend should not require considerable refactoring
2. UI/UX Design
  - Intuitive and responsive design with light/dark modes.
  - Consistent and visually appealing design for both mobile and web platforms.
3. Functionality
  - Fully functional app with core features like habit creation, tracking, and gamification.
  - Offline-first capabilities with smooth data syncing.
4. Security
  - Secure authentication and proper handling of sensitive user data.
5. CI/CD Implementation
  - A robust pipeline that automates testing, building, and releasing both APK and web app simultaneously.
  - Consistent versioning for releases across platforms.
  - Fully hosted web app with auto-deployment from CI/CD pipelines.
6. Testing Coverage
  - Comprehensive test coverage, including unit tests for logic and integration tests for workflows.
7. Gamification Features
  - Engaging XP, badges, rewards, and challenges that enhance user motivation.
8. Minimal Dependencies
  - Reliance on native Flutter capabilities and essential libraries only, ensuring maintainability and performance.

#### 9. Bonus Points

- Advanced features like analytics, habit sharing, or custom gamification (e.g., user-created badges).

### Submission Requirements

#### 1. Codebase

- Host the source code on GitHub or an equivalent repository.

#### 2. Mobile APK & Web App Deployment

- Set up a CI/CD pipeline to build and release an APK and the web app build
- Deployment for the web app using a platform like Firebase Hosting or GitHub Pages should be automated.
- Include the live link in the README (Automated)

#### 3. CI/CD Pipeline

- Implement a CI/CD pipeline that:
  - Runs automated tests on every push or pull request.
  - Builds APKs for mobile.
  - Builds and deploys the web app.

#### 4. Tests

- Write unit tests for core logic and integration tests for workflows.
- Ensure these tests run as part of the CI/CD pipeline.

#### 5. README

- Provide setup instructions for local development and Testing
- Document the CI/CD pipeline process.