



# Project Document

## Habit Tracker App with Streaks and Analysis

---

### ◆ Problem Statement:

In today's fast-paced digital lifestyle, many people struggle to build and maintain healthy habits. Existing habit-tracking apps often lack personalized insights, motivation boosters, or a clean user interface that encourages consistent usage. Without proper analytics and rewarding feedback mechanisms, users tend to abandon their goals early. This project aims to develop a modern mobile habit tracker that focuses on streak-based motivation, visual habit analysis, and ease of use to support users in forming sustainable habits.

---

### ◆ Background:

Habit formation is a psychological process that requires consistency and motivation. Apps like "Habitica" and "Streaks" offer solutions but often overwhelm users with features or under-deliver on insightful feedback.

By using mobile development frameworks like **Flutter** (for cross-platform deployment), we can create an elegant and efficient app. Integrating local storage (e.g., **Hive** or **SQLite**) or cloud-based tracking via **Firebase**, combined with visual reports and motivational features, will provide users a smarter way to manage their daily habits.

This project focuses on creating a lightweight, visually appealing, and functional application with minimal distractions and powerful tracking tools.

---

### ◆ Aim and Objectives to Cover:

#### Aim:

To build a cross-platform mobile application that allows users to track daily habits, view progress through analytics, and maintain long-term streaks with motivational features.

#### Objectives:

- Research UI/UX design best practices for mobile habit apps.
- Design a mobile-friendly and minimalistic interface using **Flutter**.
- Implement habit creation, editing, and deletion functionality.

- Store user data locally and/or with **Firestore**.
  - Build a streak-tracking mechanism with reward prompts.
  - Add a dashboard for visual analytics (charts, streak history).
  - Provide reminder notifications using **Flutter Local Notifications**.
  - Enable theme switching (light/dark modes).
  - Ensure responsiveness and performance optimization.
  - Conduct user testing to improve usability.
- 

## ♦ Methodology / Approach:

### Week 1 – Research & Planning:

- Study competitor habit apps.
- Create user stories and wireframes.  
Define technical tools (Flutter, Firestore/Hive, Charting library).

### Week 2 – UI/UX Design & Setup:

- Design splash screen, habit creation page, dashboard.
- Setup Firestore or Hive for local/cloud storage.  
Implement theme and navigation structure.

### Week 3 – Core Feature Development:

- Add habit creation, editing, and deletion functionality.
- Add daily check-in and streak maintenance logic.

### Week 4 – Analytics and Visualization:

- Create visual dashboard using charts (e.g., **fl\_chart**).
- Implement habit success rates, missed days, and longest streaks.

### Week 5 – Notification and Motivation Logic:

- Add daily notification reminders.
- Include motivational prompts and weekly summaries.

### Week 6 – Testing & Finalization:

- Bug fixing, testing on Android/iOS devices.  
Optimize app for performance.  
Final app packaging for submission.
- 

#### ♦ **Legal / Ethical / Social Issues:**

- User data (habits) will be stored securely.
  - If Firebase is used, authentication and Firestore security rules will be implemented.
  - No sensitive or personal health data will be collected.
  - The app will respect user privacy, and all permissions will be optional.
- 

#### ♦ **Project & Risk Management:**

##### **Technical Risks:**

- Firebase limitations or quota issues.
- App crashes due to platform differences (Android/iOS).  
Complexity of maintaining streak logic.

##### **Mitigation:**

- Use local storage fallback in case of connectivity issues.
- Test app on multiple devices/emulators.  
Use clear condition-based logic to manage streak flow.

##### **Timeline Management:**

- Follow **Agile sprints (weekly goals)**.
- Conduct mid-project review to freeze scope.  
Final deliverable will be reviewed before submission.

##### **User Testing Risks:**

- Limited real-time testers.  
Feature feedback delay.

### Mitigation:

- Conduct internal testing with team members.
  - Use online form/survey to gather feedback.
- 

### Submission Guidelines:

- **Duration:** 6 Weeks
  - **Final Deliverable:** ZIP file containing:
    - Complete project source code
    - APK file (or iOS TestFlight link)
    - Screenshots & video demo (optional)
    - Short write-up (README.md)
  - **Submission Method:** Upload the ZIP to the classroom folder or drive link shared with your domain lead.
- 

### Final Task:

Once submitted, kindly **create a LinkedIn post** with:

- A summary of your project
- Screenshots of the app
- Tag **Rayonix Solutions**
- Mention your role and what you learned