

Assignment Document: HabitForge – Frontend-only Login + Habit Tracker

Welcome Intern!

Your assignment is to build **HabitForge** – a beautiful, responsive **habit tracker** app that supports **multiple users**, **login with mobile + OTP-style password**, and allows each user to **track their daily habits over a week**.

 This app is **purely frontend** and does **not** require any real authentication service or backend API.

Objective

Create a **multi-user habit tracker** that allows login using **mobile number + 4-digit OTP-like password**, and gives each user their **own dashboard** where they can:

- Create new habits
- Mark them as done per day
- View habits stored only for their account

All data must be stored in **localStorage**, and the app should work even after a page refresh.

Features To Build

1. Login / Sign Up Page (Frontend-only)

- User enters **mobile number (10 digits)** and **4-digit password (OTP-style)**.
- If mobile number already exists:
 -  Login only if password matches
 -  Show error if password is wrong
- If new number:
 - Register and store `{ phone, password }` in localStorage

 After login, user is redirected to their **Habit Tracker dashboard**.

2. Habit Tracker Page (User-specific)

Each logged-in user can:

-  Add a new habit with:
 - Name (e.g., "Exercise")
 - Category (e.g., Fitness, Study)
 - Color badge (any HEX color or selected color button)
-  See 7-day toggle grid (Mon–Sun) for each habit
 - Click on a day to toggle "Done" or "Not Done"

 Habits must be saved in localStorage per user, so users don't see each other's habits.

Tech Stack (Required)

- **React** (Functional components only)
- **Context API** (for managing auth and habit state)
- **LocalStorage** (no backend or database)
- **Tailwind CSS** (or CSS Modules if you prefer)
- **Do not** use Firebase, Supabase, or Auth0



Bonus Features (Optional but impressive)

- Show **streak count** next to each habit (number of completed days)
- Dark mode toggle
- Logout button
- Clear My Habits button

Submission Checklist

- GitHub repo with meaningful commit history
- Clear `README.md` with instructions
- Live demo link (Vercel/Netlify)
- 1–2 min Screen Recorded video explaining architecture and Answering

Area	Sample Questions
Auth	How do you check if a user is already registered?
State	How did you update a single habit's Monday toggle?
Structure	Why did you split habit and auth contexts?
Edge Cases	What happens if localStorage is cleared?
Security	What are the limitations of this kind of login system?



Important Notes

- This is **not a real login system** — passwords are not encrypted.
- Your code should be clean, readable, and modular.
- Don't overcomplicate things. Focus on **correctness** and **clarity**.