

# Habit Tracker Progressive Web App - Project Report

## 1. Introduction

In today's productivity-driven world, maintaining consistent habits is vital for personal growth. This project presents a **Habit Tracker Progressive Web App (PWA)** built using modern web technologies, allowing users to track their daily habits, maintain streaks, view analytics, and receive reminders. The app is installable, works offline, and is designed to be responsive and user-friendly.

---

## 2. Abstract

The goal of this project is to create a **cross-platform, offline-first web application** that enables users to monitor and maintain daily habits. It uses **IndexedDB** to store data locally and **Workbox** to ensure offline capabilities via service workers. Built with **React** and styled using **Tailwind CSS**, the app also features progress visualization using charts. With daily check-ins and streak tracking, users are encouraged to remain consistent in their habits.

---

## 3. Tools Used

- **React.js**: For building the frontend UI.
  - **Tailwind CSS**: For modern and responsive utility-first styling.
  - **IndexedDB + idb**: For local storage and offline data persistence.
  - **Workbox (Vite PWA Plugin)**: For PWA features like service workers and caching.
  - **Chart.js or Recharts**: For visualizing streak progress and analytics.
- 

## 4. Steps Involved in Building the Project

- **Project Initialization**
  - Setup the React project using **Vite**.
  - Installed dependencies: **tailwindcss**, **idb**, **vite-plugin-pwa**, **chart.js**.
- **Tailwind CSS Configuration**
  - Created **tailwind.config.ts** file.

- Included necessary paths for purging unused styles.
    - Built responsive UI using Tailwind classes.
  - **IndexedDB Integration**
    - Used idb package to simplify interactions with IndexedDB.
    - Created a store to save habit records (name, check-ins, streaks, etc.).
  - **Habit Cards and Streak Logic**
    - Each habit card includes a title, streak counter, and check-in button.
    - Compared last check-in date to determine if streak continues or resets.
  - **Service Worker and PWA Setup**
    - Used vite-plugin-pwa to create a manifest file.
    - Registered a service worker using Workbox.
    - Enabled installability and offline functionality.
  - **Analytics and Charts**
    - Used Chart.js or Recharts to visualize streak trends and activity graphs.
  - **Mock Reminders Feature**
    - Implemented basic JavaScript alert-based reminders.
    - Simulated future notification logic via service worker fallback (optional).
- 

## 5. Conclusion

This project showcases how a reliable and installable **Habit Tracker PWA** can be developed using modern frontend and offline-first tools. By combining React, IndexedDB, and service workers, users benefit from a smooth experience that works without internet access. Future improvements may include cloud sync, advanced reminders, user authentication, and cross-device sync.