■ Project Report – Habit Tracker

1. Introduction

The Habit Tracker is a lightweight, modern web application designed to help users build and maintain positive habits. It allows users to create habits, track daily completion, and visualize progress using charts. The app is installable as a Progressive Web App (PWA) and works offline, ensuring accessibility across devices.

2. Objectives

• To build a daily habit tracking system with persistent storage.
• To provide a user-friendly and professional interface.
• To enable visual insights into user progress via charts.
• To ensure the app is cross-platform and installable (PWA).

3. Features

1. Add Habits – Create new habits with name and description.
2. Mark Daily Completion – Track if a habit is done for today.
3. Progress Chart – Displays a 7-day overview of completions.
4. Delete Habits – Remove habits no longer needed.
5. Offline Support – Works without internet using IndexedDB.
6. PWA Installation – Can be installed on desktop/mobile.

4. System Requirements

Software:
• Node.js (v18 or above)
• npm (v9 or above)
• Modern browser (Chrome/Edge/Firefox)

Hardware:
• Minimum 4 GB RAM
• Processor: Dual-core or higher
• Storage: ~200 MB

5. Technology Stack

• Frontend: React 19 + Vite
• Styling: TailwindCSS v4
• Database: IndexedDB (via idb library)
• Charts: Chart.js + react-chartjs-2
• PWA: Vite PWA plugin

6. System Design

Architecture:
• UI Layer: React components (App, HabitCard, AddHabitModal, ProgressChart)
• Data Layer: IndexedDB for local persistence
• Service Worker: PWA support for offline usage

Data Flow:
1. User adds/marks a habit.
2. Data is stored in IndexedDB (db.js).
3. UI refreshes state from database.
4. ProgressChart visualizes stored data.

7. Implementation

• Frontend: Built using React + Vite for fast development.
• Database: IndexedDB ensures persistence without backend.
• Styling: TailwindCSS for modern, responsive UI.
• Charts: Chart.js provides interactive progress graphs.
• PWA: Allows installation and offline-first support.

8. Results

• Users can successfully add, mark, and delete habits.
• The progress chart updates dynamically with daily completions.
• The app is responsive, professional, and installable.

9. Future Enhancements

• ■ Reminders/Notifications for habits.
• ■ Calendar view of habit completions.
• ■ Cloud sync across devices with login.
• ■ Goal setting & streak tracking.

10. Conclusion

The Habit Tracker project demonstrates how modern web technologies can be combined to build a lightweight, offline-capable productivity app. It meets the objectives of tracking daily habits, visualizing progress, and providing a professional user experience.

11. References

• React Documentation – https://react.dev
• TailwindCSS – https://tailwindcss.com

- Chart.js – https://www.chartjs.org
- IndexedDB API – MDN Docs