# Family Calendar App - Problem Statement & Planning

## Business Problem

Modern families often face challenges staying organized amidst busy schedules. With appointments, meals, chores, events, and household responsibilities constantly overlapping, families frequently rely on a fragmented system of physical calendars, digital tools, and multiple apps. This disjointed setup can lead to:

* Missed events or double-booking
* Meal planning confusion and unnecessary takeout expenses
* Lack of clarity on chore assignments
* Inefficient weekly planning due to the absence of a unified feedback process.

While several family planning apps exist, they fall short in two critical areas:

1. Integrated Weekly Check-In - None offer a true “Family Meeting” module that gathers calendar, goals, tasks and meals into one digestible view for collaborative review.
2. Customizable Navigation - Most apps enforce a rigid interface with no option to hide or disable unused features, reducing accessibility for families with varying needs.

## Proposed Solution

The Family Calendar App is a customizable, all-in-one planning tool designed to help families stay organized and aligned. Inspired by real needs within my own family, this app prioritizes usability, affordability, and modular functionality.

Unlike existing solutions like Skylight or Cozi, the Family Calendar App introduces a unique “Family Meeting” feature. This is a centralized dashboard for reviewing weekly events, goals, meal plans, and tasks in one shared space. This encourages meaningful weekly check-ins and goal setting, which are often overlooked in competing apps.

Additionally, the app allows families to toggle components on and off, empowering them to create a personalized planning experience. Whether they want to focus solely on meals and chores, or use every feature, the app adapts to their lifestyle, not the other way around.

As a solo developer, I am building this app with accessibility in mind, minimizing hardware requirements and offering a low-cost subscription model that doesn’t sacrifice functionality.

## Deliverables

* Problem Statement Document
* Functional and Non-Functional Requirements
* Wireframes
* ERD
* Class Diagrams
* Flutter App Source Code
* User Testing Documentation

## Timeline

* Week 1: Problem statement, requirements, wireframes, and diagrams
* Week 2-3: Initial Flutter build
* Week 4: Firebase integration
* Week 5-8: User testing
* Week 9: Bug fixes based on testing feedback
* Week 10-11:
  + - * + Develop static website with app guidelines and navigation to app
        + Create demo videos for PWA installation and usability
        + Deploy to web (free tier and subscriptions)
* Week 12+:
  + - * + Document real user feedback
        + Refactor/update app based on feedback
        + Deploy to mobile app stores (Android and iOS)
        + Update tutorials for mobile installation

## Deployment Plan

* Source code will be hosted and maintained on GitHub
* Flutter Web Build deployed via Firebase Hosting (subscription supported)
* Demo video to showcase usability across devices
* The mobile experience will be enhanced with platform-specific builds (Android/iOS)
* Additional deployment options may include the Microsoft Store and Apple App Store for Mac
* A long-term goal is to transition from Firebase to a custom backend for greater control over data and access management.

## Validation and Verification

* Manual feature-by-feature testing throughout development.
* Responsive testing across multiple screen sizes (TV/tablet/mobile)
* Feedback from 2-3 real users during the user testing phase