

Daily Report Generator

Team Project Documentation (Flutter Windows Desktop)

Project Duration: 1 Day Prototype

Team Size: 4 Members

Platform: Windows Desktop (Flutter)

Storage: JSON Files

Future Scope: Android, Web, Cloud Sync

1. Project Overview

Problem

Interns and employees usually store work logs in:

- WhatsApp
- Notepad
- Word documents
- Excel sheets

At the end of the internship they struggle to create:

- Weekly reports
 - Monthly reports
 - Final internship report
-

Solution

Internship Tracker Pro allows users to:

- Add daily work logs
 - Track hours worked
 - Track tasks completed
 - Save learnings
 - View progress dashboard
 - Generate PDF reports instantly
-

2. Features

Feature 1 - Daily Work Entry

User can add:

- Date
 - Project Name
 - Hours Worked
 - Tasks Completed
 - Learnings
 - Notes
-

Feature 2 - Dashboard

Display:

- Total Days Worked
 - Total Hours Worked
 - Total Tasks Completed
 - Total Projects
-

Feature 3 - History

Show all saved entries.

Example:

24 June
23 June
22 June
21 June

Click entry to view details.

Feature 4 - PDF Generator

Generate:

Weekly Report

Heading

User name Details,
Company name
Clg name (optional)

Week 1

Day 1

Task A

Task B

Day 2

Task C

Task D

Monthly Report

June Report

Total Hours: 80

Tasks Completed: 25

Feature 5 (Bonus)

AI Report Formatter

Input:

fixed login bug
created dashboard

Output:

Resolved authentication issues
and implemented dashboard functionality.

3. Technology Stack

Frontend

Flutter

Storage

JSON File

Example:

```
[
  {
    "date": "2026-06-24",
    "project": "Internship Tracker",
    "hours": 6,
    "tasks": [
      "Created Login UI",
      "Implemented Dashboard"
    ],
    "learning": "State Management",
    "notes": "Completed first prototype"
  }
]
```

PDF

Flutter PDF Package

pdf:

printing:

4. Folder Structure

lib/

└─ main.dart

└─ models/

```
|   └─ work_log.dart

|   └─ screens/
|       ├── dashboard_screen.dart
|       ├── add_log_screen.dart
|       ├── history_screen.dart
|       └─ report_screen.dart

|   └─ services/
|       ├── json_service.dart
|       └─ pdf_service.dart

|   └─ widgets/
|       ├── dashboard_card.dart
|       └─ worklog_tile.dart

└─ utils/
```

5. Team Distribution

Member 1

UI Developer

Screens:

- Dashboard
- Sidebar Navigation
- Theme

Deliverables:

dashboard_screen.dart
navigation
cards

Member 2

Daily Log System

Screens:

Add Entry Screen

Edit Entry Screen

Work:

- Form Validation
- Save Button
- Edit Function

Files:

add_log_screen.dart

work_log.dart

Member 3

JSON Storage

Responsibilities:

- Create JSON Service
- Read Data
- Write Data
- Delete Data
- Update Data

Files:

json_service.dart

Functions:

saveLog()

loadLogs()

deleteLog()

updateLog()

Member 4

Reports + PDF

Responsibilities:

- Weekly Reports
- Monthly Reports
- Export PDF

Files:

pdf_service.dart
report_screen.dart

Functions:

generateWeeklyReport()
generateMonthlyReport()

6. Screen Flow

Dashboard

|
|
+----- Add Entry
|
+----- History
|
+----- Reports

7. Dashboard Design

Top Cards:

Total Days

Total Hours

Tasks Completed

Projects

Recent Entries:

24 June

Worked 6 Hours

23 June

Worked 5 Hours

8. Add Entry Form

Fields

Date

Project Name

Hours Worked

Tasks Completed

Learnings

Notes

Button

Save Entry

9. WorkLog Model

```
class WorkLog {  
    String date;  
    String project;  
    int hours;  
    List<String> tasks;  
    String learning;  
    String notes;  
}
```

10. JSON Service Logic

Save

Read JSON

Add New Entry

Write JSON

Load

Open JSON

Convert to Objects

Display

Delete

Remove Entry

Rewrite JSON

11. PDF Structure

Cover Page

Internship Tracker Report

Name

Company

Date

Content

Day 1

Task A

Task B

Learning

Summary

Total Days

Total Hours

Total Tasks

(Optional Build and Webpage to Navigate and to use applications and give downloading access).

Final Presentation Pitch

Internship Tracker Pro is a desktop productivity application built using Flutter. It helps interns and trainees maintain daily work logs, track progress, store learning records, and automatically generate professional PDF reports. The application uses local JSON-based storage and can be extended to Android, Web, and Cloud platforms in the future.

References:-

(Some existing product out there use best feature and design reference layout all accordingly).

Design Design:

Colors:-

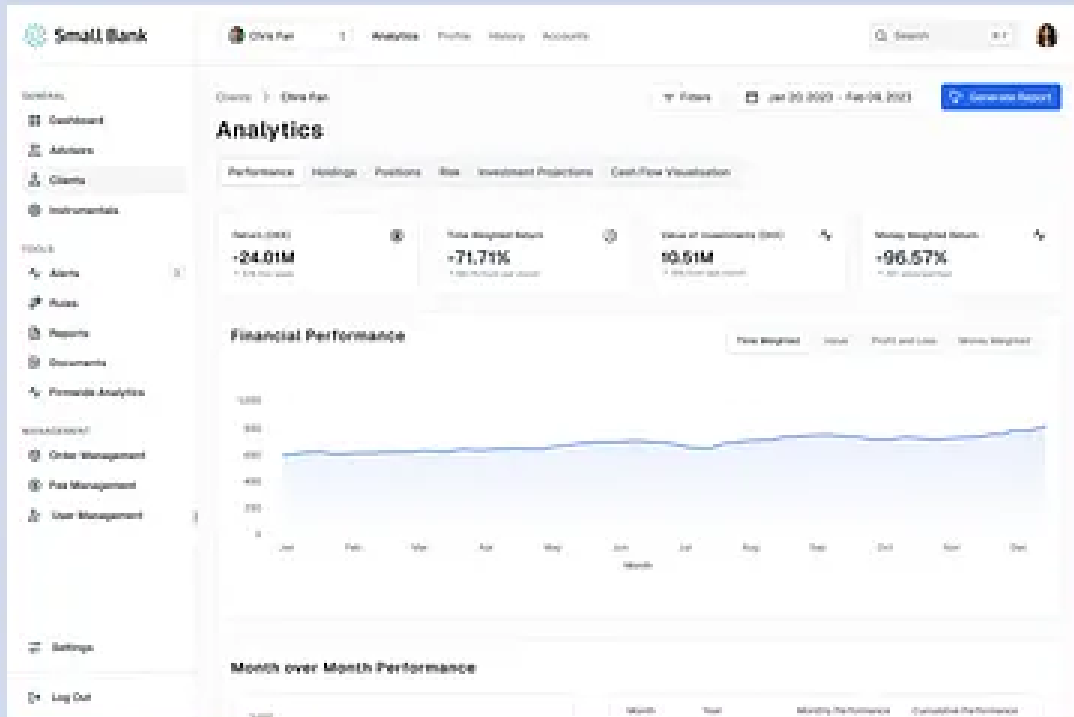
Primary color : White

Secondary Color : blue

Font size,

Main heading : 32pix / (Optional 42Pix, use this only for main or most big Heading for normal headline stick with 32Pix).

Sub data : 22pix



The screenshot displays the 'axil' 'Create A Report' form. The form includes a sidebar with navigation links: Dashboard, Assets, Reports, UTILITIES, Servers, Users, and Feedback. The main content area is titled 'Create A Report' and shows a date range of '06/01/18 12:00:00 A.M. CT - 06/07/18 12:00:00 A.M. CT'. The form includes fields for:

- Start Date: 06/01/18 12:00:00 A.M. CT
- End Date: 06/07/18 12:00:00 A.M. CT
- Poll Rate: Daily
- Recurring Report: Does not repeat
- Report Title: Add A Title

Below the form fields is a 'Run Query' button. To the right of the form is a table for adding rows:

Asset	Sensor Value	Mode
abc-123, def-456, ghi-789, abc-123, def-456, ghi-789	Mode	

The table has a 'Run Query' button and a 'New Row' button. The 'Run Query' button is highlighted in blue.

