

B.TECH · FINAL YEAR

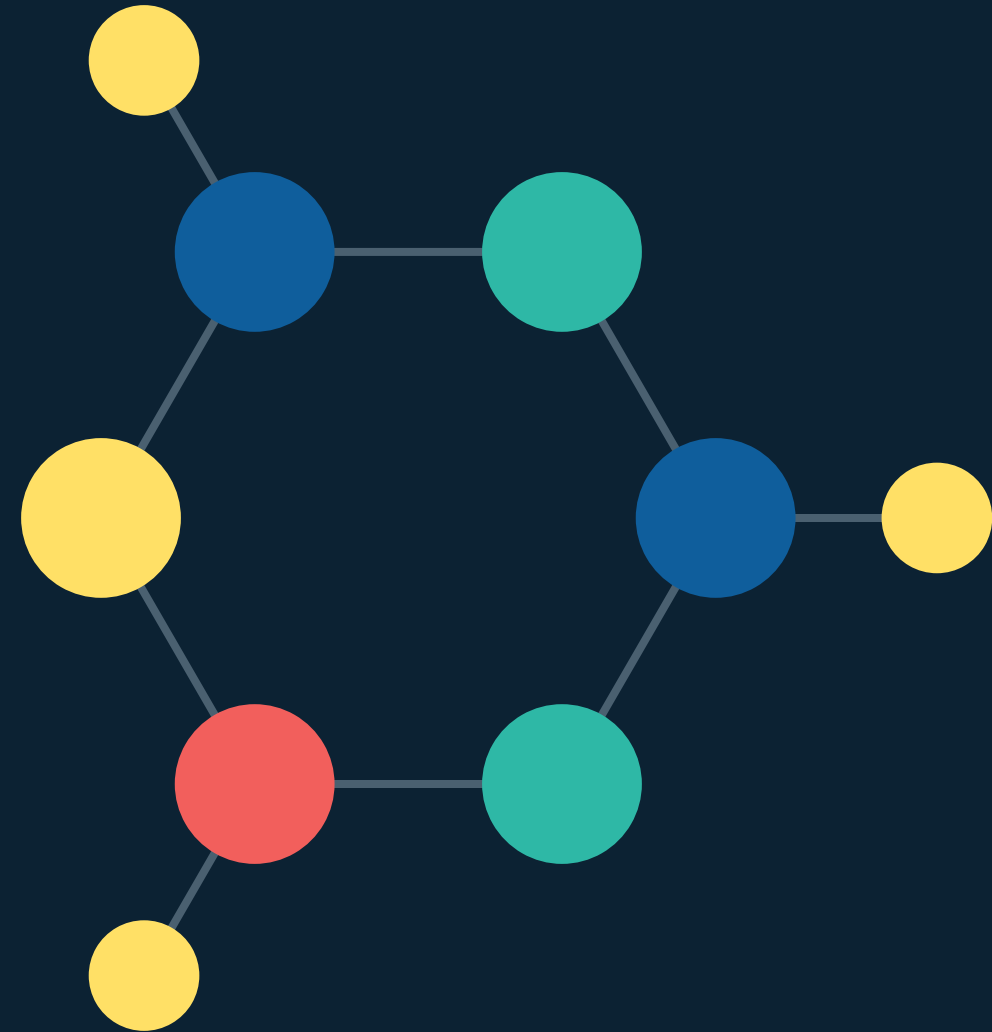
Capstone Project Compendium

All project chapters in one editable deck

PRESENTED BY
Presenter Name

MODULES
5

EDITION
2026



Course map

01

**Smart Parking
Management System**

02

**Solar Power Monitoring
Dashboard**

03

**Voice-Controlled Home
Automation**

04

**Smart Irrigation System
using IoT**

05

**Hospital Appointment
Booking System**

01

**SMART PARKING
MANAGEMENT SYSTEM**

KEY IDEA

Find a free parking slot in seconds

Project 1 · Smart Parking Management System

Introduction

- ▶ Parking slot management matters to drivers and mall operators every day
- ▶ Current methods are manual, slow and hard to scale
- ▶ Affordable technology now makes automation practical
- ▶ This project builds smart parking management system as a working prototype

CONTEXT

Most drivers and mall operators we surveyed still rely on paper or spreadsheets for parking slot management.

Problem statement

01

Manual work

Parking slot management takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Drivers and mall operators lack data to make decisions.

Objectives

- ▶ Automate the core steps of parking slot management
- ▶ Give drivers and mall operators a simple, mobile-friendly interface
- ▶ Store data securely and generate reports
- ▶ Evaluate the system against the existing process

Literature survey

Study	Approach	Limitation
Study A	Rule-based automation	Not adaptive
Study B	Mobile data collection	No analytics
Commercial tools	Full-featured suites	Costly and complex

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Manual records
- ✓ Slow turnaround
- ✓ No reports

Proposed system

- Automated workflow
- Real-time dashboard
- One-click reports

System architecture

1

Input

Data captured from users or sensors

2

Process

Core logic validates and analyses

3

Store

Secure database and backups

4

Output

Dashboard, alerts and reports

Block diagram

Every module talks through one simple, testable interface.

The system is built in layers: an input layer, a processing service, a database and a dashboard. Each layer can be tested independently, which keeps smart parking management system simple to maintain and extend.



Block diagram / flowchart

Technology stack

01

**Ultrasonic
sensors**

Slot occupancy

02

Raspberry Pi

Edge controller

03

MQTT

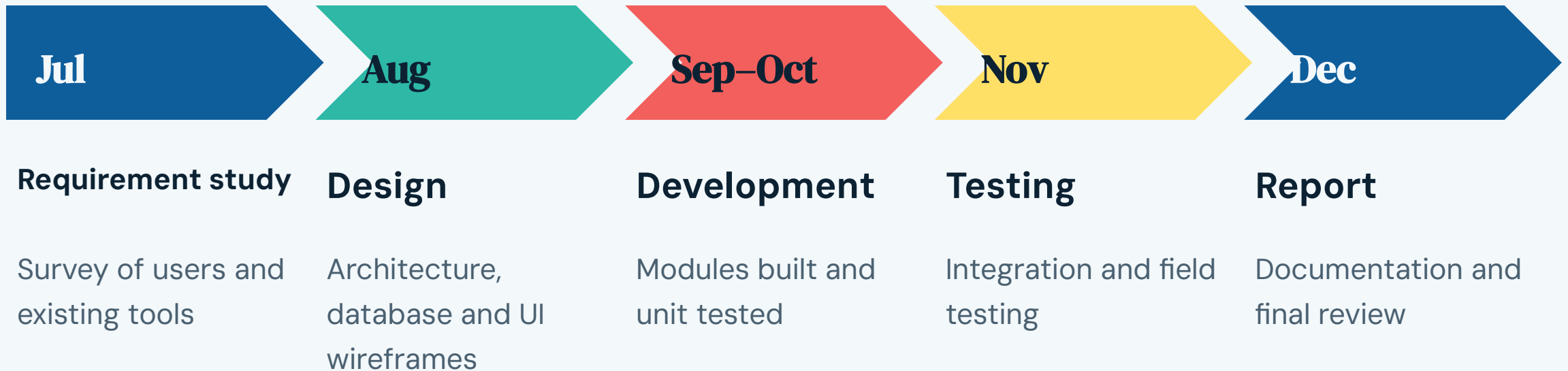
Live updates

04

Flutter

Driver app

Project timeline



Implementation snapshots



Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

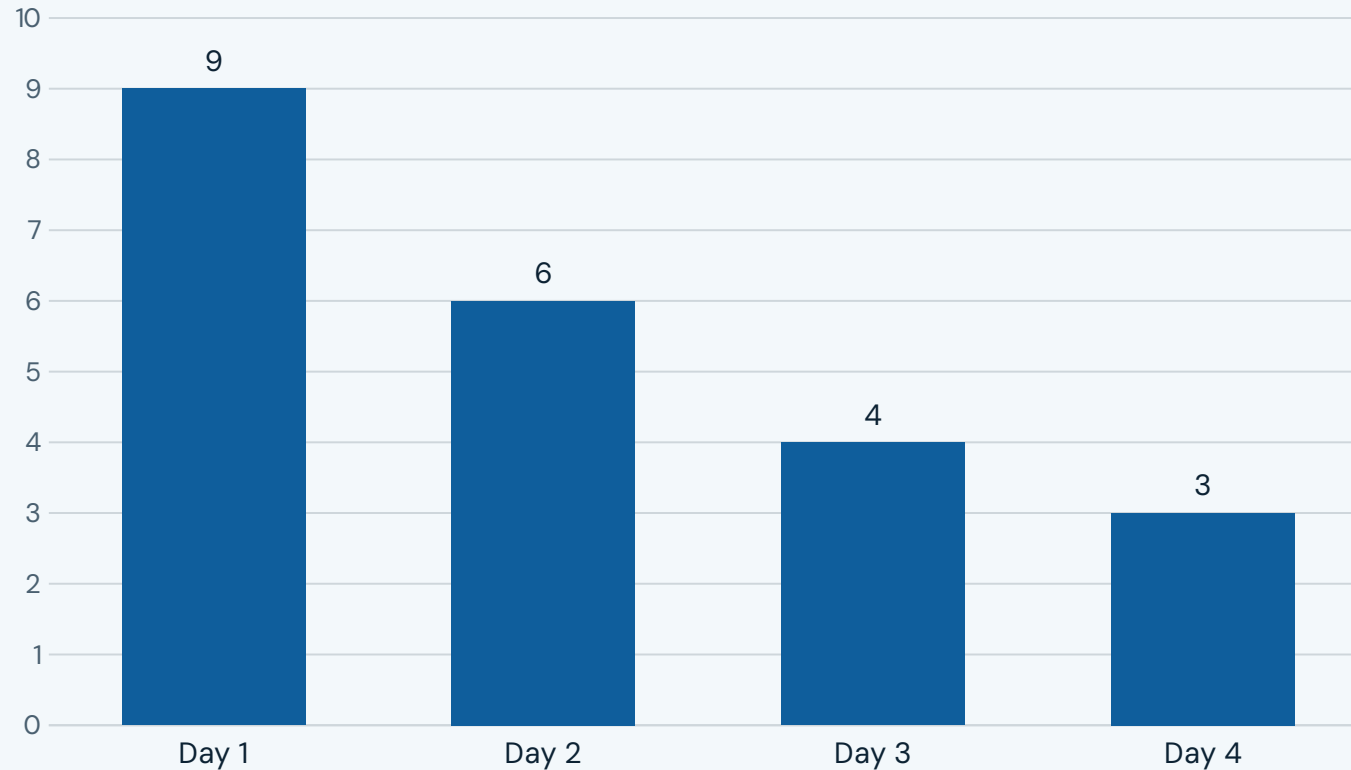


Image 3

Reports

Exportable summaries

Results



Sample data for illustration — replace with your own figures.

3×

faster slot finding

The prototype improved steadily over the evaluation period (sample data).

Key outcomes

3×

faster slot finding

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- ▶ Smart Parking Management System meets its main objectives
- ▶ Automation saves time and reduces errors for drivers and mall operators
- ▶ The design is simple to extend and maintain

Future scope

01

Mobile app

Native apps for Android and iOS

02

AI insights

Predictions from historical data

03

Scale-up

Deploy across more sites

References

- ▶ [1] A. Author, "Title of the paper," Journal Name, vol. 12, pp. 1–10, 2023.
- ▶ [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- ▶ [3] D. Author, "Conference paper title," in Proc. Conference, 2024.
- ▶ [4] Official documentation of the tools used (accessed 2025).

02

SOLAR POWER MONITORING DASHBOARD

KEY IDEA

Live generation tracking for rooftop solar

Project 2 · Solar Power Monitoring Dashboard

Introduction

- ▶ Solar plant monitoring matters to homeowners and installers every day
- ▶ Current methods are manual, slow and hard to scale
- ▶ Affordable technology now makes automation practical
- ▶ This project builds solar power monitoring dashboard as a working prototype

CONTEXT

Most homeowners and installers we surveyed still rely on paper or spreadsheets for solar plant monitoring.

Problem statement

01

Manual work

Solar plant monitoring takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Homeowners and installers lack data to make decisions.

Objectives

- ▶ Automate the core steps of solar plant monitoring
- ▶ Give homeowners and installers a simple, mobile-friendly interface
- ▶ Store data securely and generate reports
- ▶ Evaluate the system against the existing process

Literature survey

Study	Approach	Limitation
Study A	Rule-based automation	Not adaptive
Study B	Mobile data collection	No analytics
Commercial tools	Full-featured suites	Costly and complex

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Manual records
- ✓ Slow turnaround
- ✓ No reports

Proposed system

- Automated workflow
- Real-time dashboard
- One-click reports

System architecture

1

Input

Data captured from users or sensors

2

Process

Core logic validates and analyses

3

Store

Secure database and backups

4

Output

Dashboard, alerts and reports

Block diagram

Every module talks through one simple, testable interface.



Block diagram / flowchart

The system is built in layers: an input layer, a processing service, a database and a dashboard. Each layer can be tested independently, which keeps solar power monitoring dashboard simple to maintain and extend.

Technology stack

01

INA219 sensors

Voltage and current

02

ESP32

Data logger

03

InfluxDB

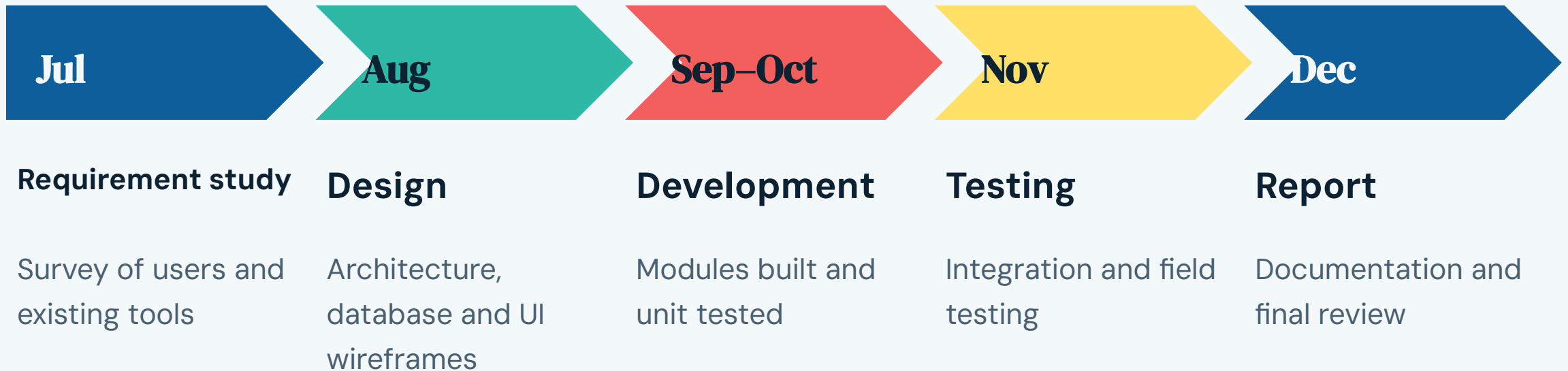
Time-series data

04

Grafana

Dashboard

Project timeline



Implementation snapshots



Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

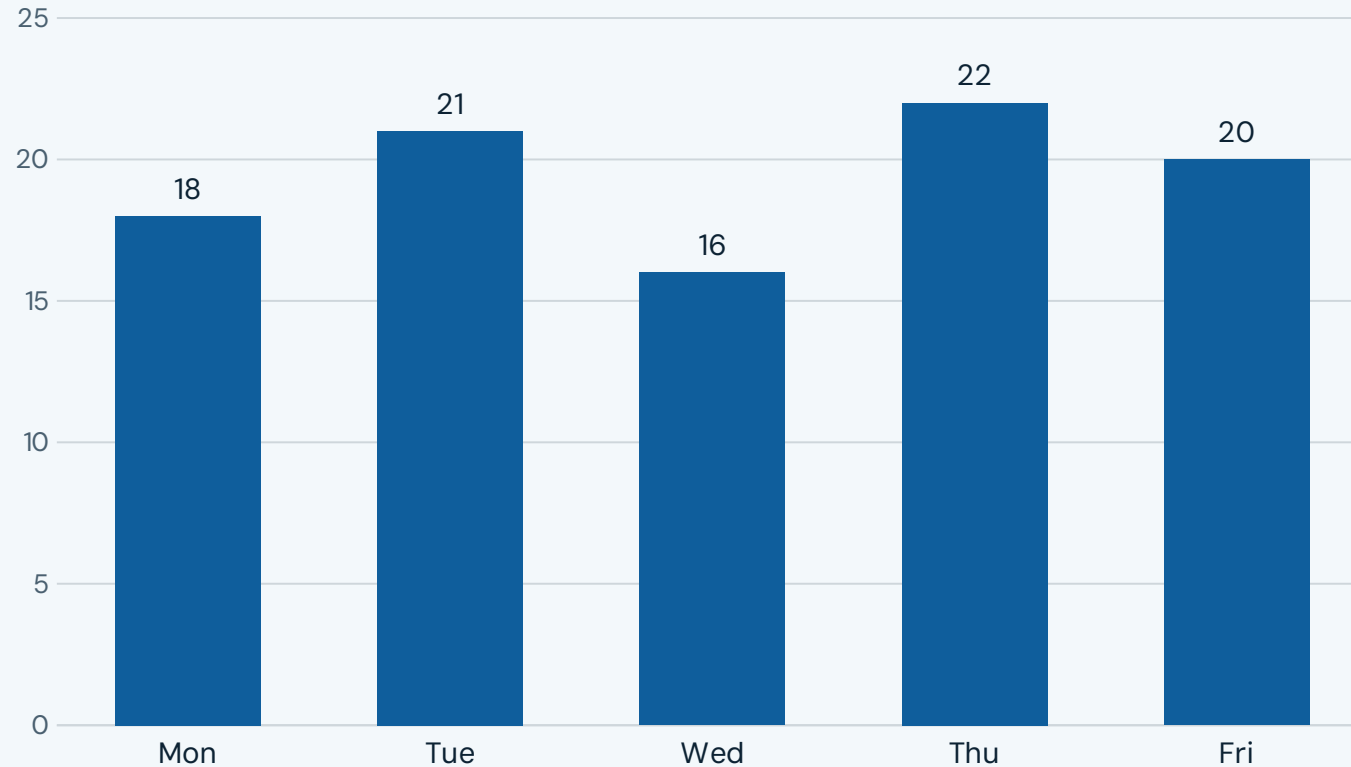


Image 3

Reports

Exportable summaries

Results



Sample data for illustration — replace with your own figures.

12%

more yield after fault alerts

The prototype improved steadily over the evaluation period (sample data).

Key outcomes

12%

more yield after fault
alerts

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- ▶ Solar Power Monitoring Dashboard meets its main objectives
- ▶ Automation saves time and reduces errors for homeowners and installers
- ▶ The design is simple to extend and maintain

Future scope

01

Mobile app

Native apps for Android and iOS

02

AI insights

Predictions from historical data

03

Scale-up

Deploy across more sites

References

- ▶ [1] A. Author, "Title of the paper," Journal Name, vol. 12, pp. 1–10, 2023.
- ▶ [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- ▶ [3] D. Author, "Conference paper title," in Proc. Conference, 2024.
- ▶ [4] Official documentation of the tools used (accessed 2025).

03

VOICE-CONTROLLED HOME AUTOMATION

KEY IDEA

Switch lights and fans with your voice

Project 3 · Voice-Controlled Home Automation

Introduction

- ▶ Controlling home appliances matters to families and elderly users every day
- ▶ Current methods are manual, slow and hard to scale
- ▶ Affordable technology now makes automation practical
- ▶ This project builds voice-controlled home automation as a working prototype

CONTEXT

Most families and elderly users we surveyed still rely on paper or spreadsheets for controlling home appliances.

Problem statement

01

Manual work

Controlling home appliances takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Families and elderly users lack data to make decisions.

Objectives

- ▶ Automate the core steps of controlling home appliances
- ▶ Give families and elderly users a simple, mobile-friendly interface
- ▶ Store data securely and generate reports
- ▶ Evaluate the system against the existing process

Literature survey

Study	Approach	Limitation
Study A	Rule-based automation	Not adaptive
Study B	Mobile data collection	No analytics
Commercial tools	Full-featured suites	Costly and complex

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Manual records
- ✓ Slow turnaround
- ✓ No reports

Proposed system

- Automated workflow
- Real-time dashboard
- One-click reports

System architecture

1

Input

Data captured from users or sensors

2

Process

Core logic validates and analyses

3

Store

Secure database and backups

4

Output

Dashboard, alerts and reports

Block diagram

Every module talks through one simple, testable interface.

The system is built in layers: an input layer, a processing service, a database and a dashboard. Each layer can be tested independently, which keeps voice-controlled home automation simple to maintain and extend.



Block diagram / flowchart

Technology stack

01

ESP8266

Wi-Fi relay board

02

**Voice assistant
API**

Speech commands

03

Node-RED

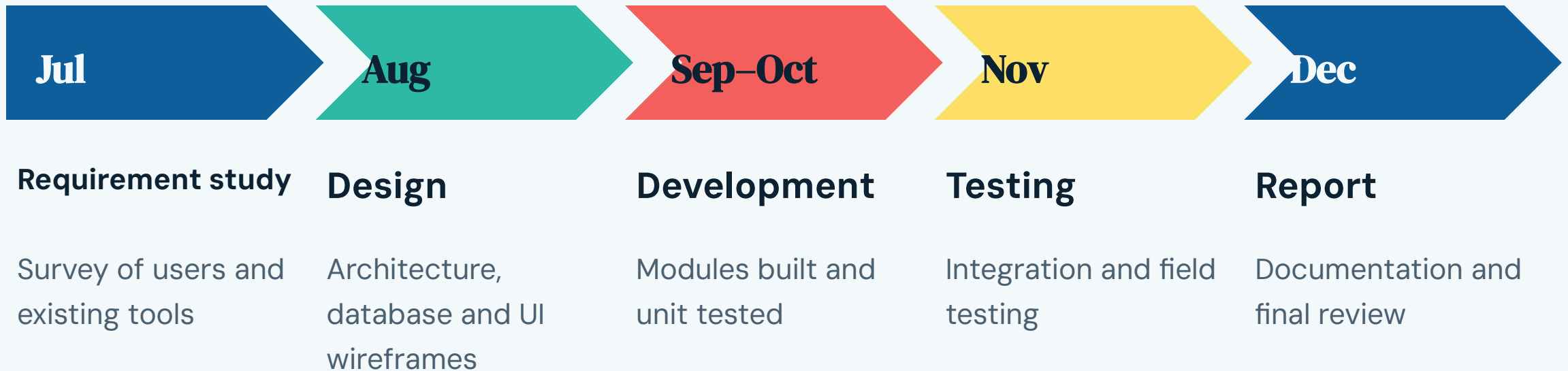
Automation flows

04

Mobile app

Manual control

Project timeline



Implementation snapshots



Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

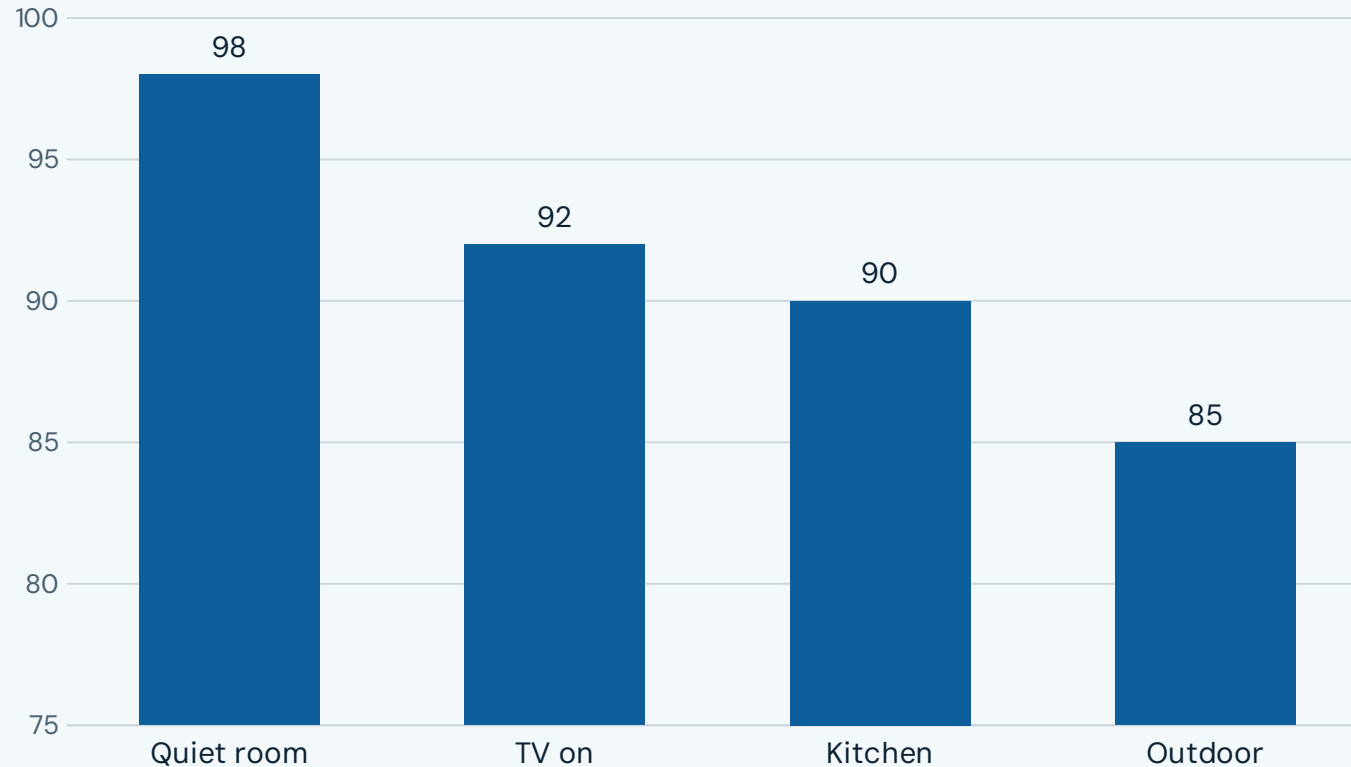


Image 3

Reports

Exportable summaries

Results



Sample data for illustration — replace with your own figures.

95%

commands recognised

The prototype improved steadily over the evaluation period (sample data).

Key outcomes

95%

commands recognised

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- ▶ Voice-Controlled Home Automation meets its main objectives
- ▶ Automation saves time and reduces errors for families and elderly users
- ▶ The design is simple to extend and maintain

Future scope

01

Mobile app

Native apps for Android and iOS

02

AI insights

Predictions from historical data

03

Scale-up

Deploy across more sites

References

- ▶ [1] A. Author, "Title of the paper," Journal Name, vol. 12, pp. 1–10, 2023.
- ▶ [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- ▶ [3] D. Author, "Conference paper title," in Proc. Conference, 2024.
- ▶ [4] Official documentation of the tools used (accessed 2025).

04

SMART IRRIGATION SYSTEM USING IOT

KEY IDEA

Soil-moisture driven watering for small farms

Project 4 · Smart Irrigation System using IoT

Introduction

- ▶ Agriculture depends heavily on timely, measured irrigation
- ▶ Manual watering wastes water, power and labour
- ▶ Low-cost sensors and microcontrollers make automation affordable
- ▶ This project builds a sensor-driven pump controller with a phone dashboard

CONTEXT

Flood irrigation is still the most common method on small Indian farms.

Problem statement

01

Over-watering

Fields are watered on a fixed schedule, not on actual soil need.

02

No visibility

Farmers cannot see soil conditions without visiting the field.

03

High cost

Commercial drip automation is too expensive for small plots.

Objectives

- ▶ Measure soil moisture, temperature and humidity in real time
- ▶ Switch the pump automatically using threshold logic
- ▶ Show live readings and alerts on a mobile dashboard
- ▶ Keep the total hardware cost under ₹2,500 per unit

Literature survey

Study	Approach	Limitation
Kumar et al.	GSM-based pump control	No sensor feedback
Rao & Singh	Timer-based drip system	Ignores weather and soil
Fernandes	Cloud dashboard prototype	Needs constant Wi-Fi

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Fixed timers or manual switching
- ✓ No data on soil condition
- ✓ Water and power wasted

Proposed system

- Sensor-based automatic switching
- Live dashboard with alerts
- Lower water and power use

System architecture

- 1 Sense**

Moisture, DHT11 and rain sensors read field data
- 2 Process**

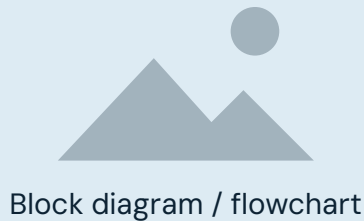
ESP32 compares readings with crop thresholds
- 3 Act**

Relay switches the pump on or off
- 4 Notify**

Readings and alerts sync to the app

Block diagram

Every module talks through one simple, testable interface.



Sensors feed an ESP32 controller. A relay module drives the pump, and readings are pushed every minute to a lightweight cloud database that the mobile dashboard reads.

Technology stack

01

ESP32

Wi-Fi
microcontroller
running the control
loop

02

Sensors

Capacitive soil
moisture, DHT11, rain
sensor

03

Firebase

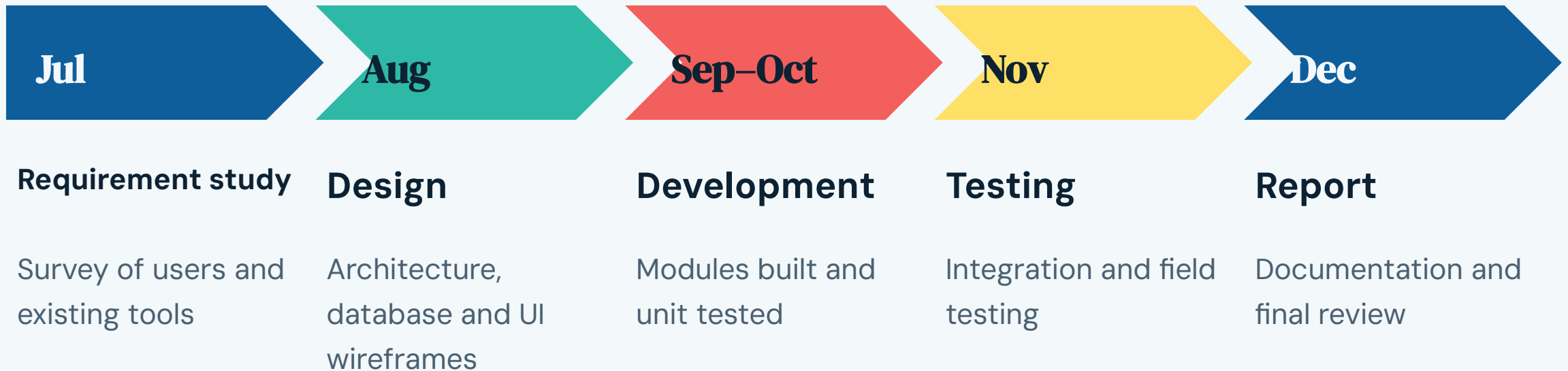
Real-time database
for readings

04

Flutter

Cross-platform
mobile dashboard

Project timeline



Implementation snapshots



Image 1

Hardware prototype

Controller, relay and sensors on a test bed



Image 2

Field setup

Probe placement in a 10 m² test plot

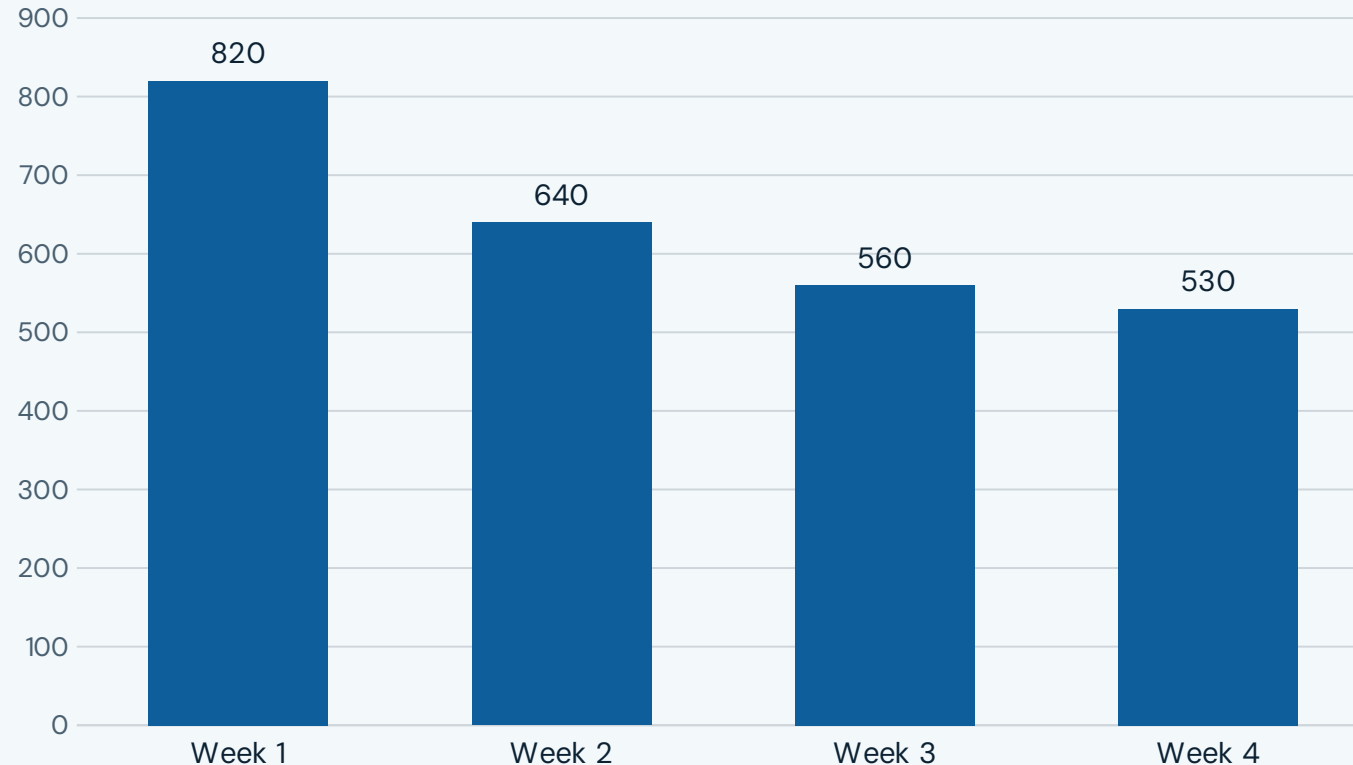


Image 3

Mobile dashboard

Live readings, history and pump status

Results



Sample data for illustration — replace with your own figures.

35%

less water than the manual baseline

Automatic switching cut water use steadily as thresholds were tuned for the crop.

Key outcomes

35%

less water used in the test plot

₹2,300

hardware cost per unit

< 2 s

pump response time

Figures from our test setup (sample data).

Conclusion

- ▶ A low-cost controller can automate irrigation reliably
- ▶ Sensor feedback avoids over-watering and saves power
- ▶ The dashboard gives farmers visibility without field visits

Future scope

01

Weather forecast

Skip watering when rain is expected

02

Solar power

Run the unit fully off-grid

03

Multi-zone

Control several valves per farm

References

- ▶ [1] A. Author, "Title of the paper," Journal Name, vol. 12, pp. 1–10, 2023.
- ▶ [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- ▶ [3] D. Author, "Conference paper title," in Proc. Conference, 2024.
- ▶ [4] Official documentation of the tools used (accessed 2025).

05

HOSPITAL APPOINTMENT BOOKING SYSTEM

KEY IDEA

Online OPD booking with token tracking

Project 5 · Hospital Appointment Booking System

Introduction

- ▶ Outpatient appointment booking matters to patients and hospital staff every day
- ▶ Current methods are manual, slow and hard to scale
- ▶ Affordable technology now makes automation practical
- ▶ This project builds hospital appointment booking system as a working prototype

CONTEXT

Most patients and hospital staff we surveyed still rely on paper or spreadsheets for outpatient appointment booking.

Problem statement

01

Manual work

Outpatient appointment booking takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Patients and hospital staff lack data to make decisions.

Objectives

- ▶ Automate the core steps of outpatient appointment booking
- ▶ Give patients and hospital staff a simple, mobile-friendly interface
- ▶ Store data securely and generate reports
- ▶ Evaluate the system against the existing process

Literature survey

Study	Approach	Limitation
Study A	Rule-based automation	Not adaptive
Study B	Mobile data collection	No analytics
Commercial tools	Full-featured suites	Costly and complex

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Manual records
- ✓ Slow turnaround
- ✓ No reports

Proposed system

- Automated workflow
- Real-time dashboard
- One-click reports

System architecture

1

Input

Data captured from users or sensors

2

Process

Core logic validates and analyses

3

Store

Secure database and backups

4

Output

Dashboard, alerts and reports

Block diagram

Every module talks through one simple, testable interface.

The system is built in layers: an input layer, a processing service, a database and a dashboard. Each layer can be tested independently, which keeps hospital appointment booking system simple to maintain and extend.



Block diagram / flowchart

Technology stack

01

React

Patient and staff
web app

02

Spring Boot

Booking APIs

03

PostgreSQL

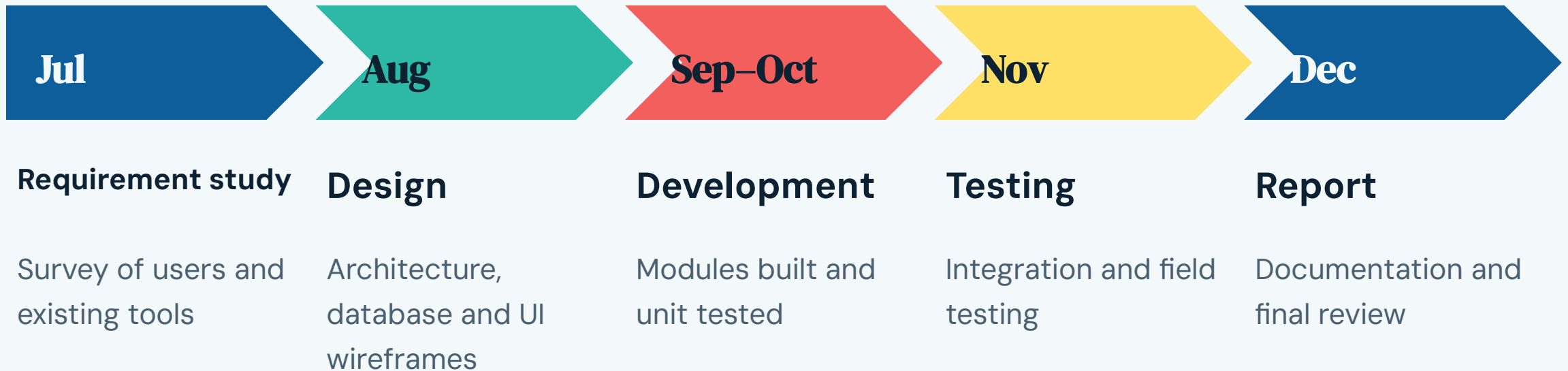
Appointments and
doctors

04

Twilio-style SMS

Token alerts (any
SMS gateway)

Project timeline



Implementation snapshots



Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

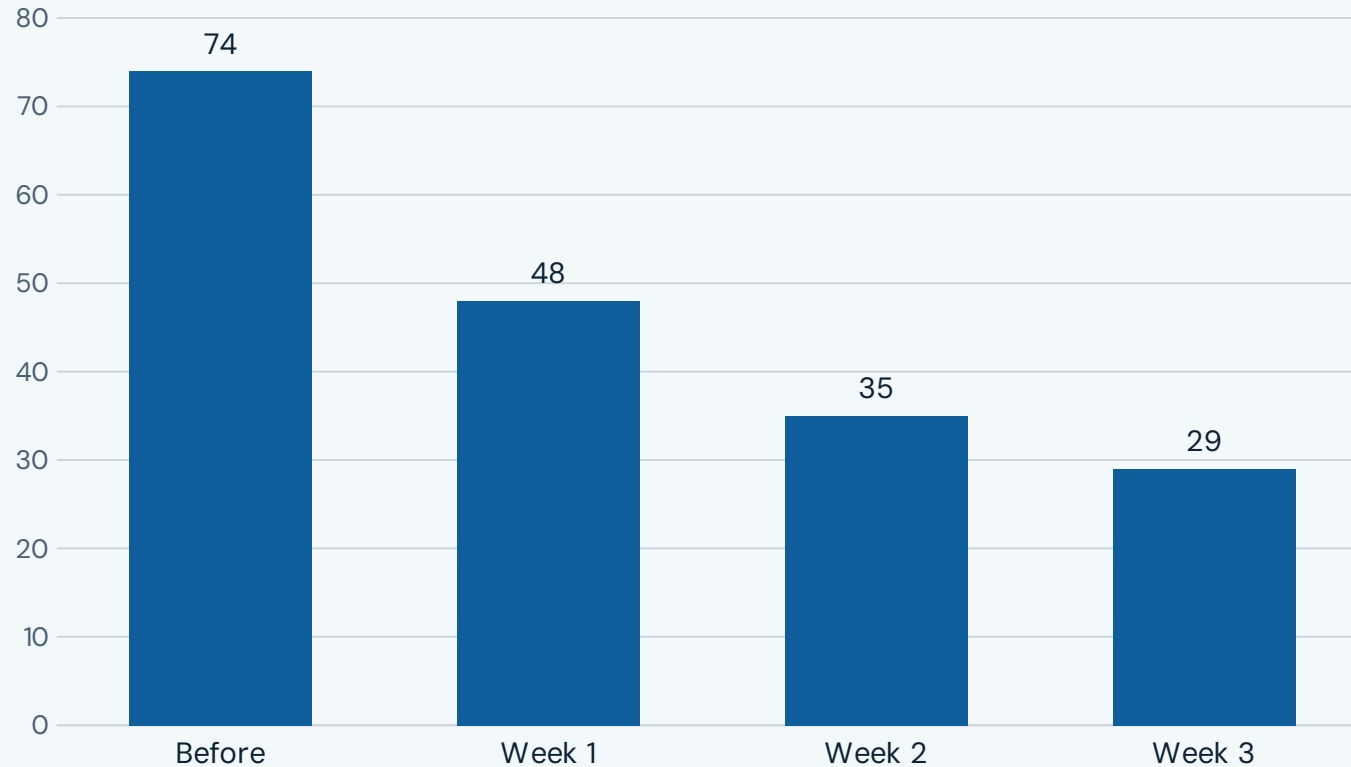


Image 3

Reports

Exportable summaries

Results



Sample data for illustration — replace with your own figures.

60%

shorter waiting time

The prototype improved steadily over the evaluation period (sample data).

Key outcomes

60%

shorter waiting time

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- ▶ Hospital Appointment Booking System meets its main objectives
- ▶ Automation saves time and reduces errors for patients and hospital staff
- ▶ The design is simple to extend and maintain

Future scope

01

Mobile app

Native apps for Android and iOS

02

AI insights

Predictions from historical data

03

Scale-up

Deploy across more sites

References

- ▶ [1] A. Author, "Title of the paper," Journal Name, vol. 12, pp. 1–10, 2023.
- ▶ [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- ▶ [3] D. Author, "Conference paper title," in Proc. Conference, 2024.
- ▶ [4] Official documentation of the tools used (accessed 2025).

Thank you

Questions and discussion.

Presenter Name

hello@yourdomain.in

