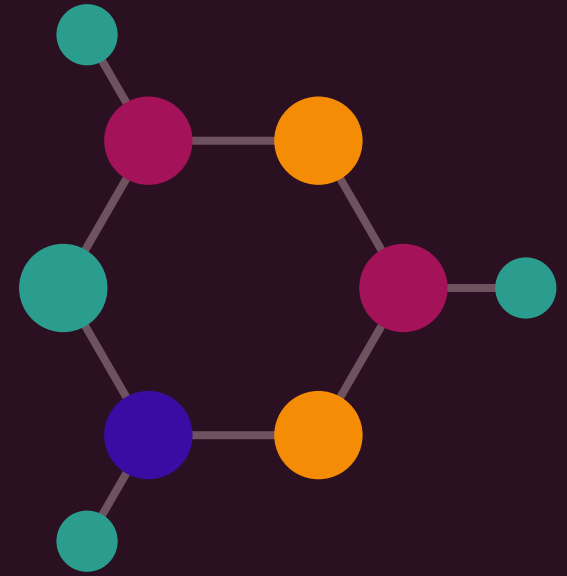


B.TECH · FINAL YEAR

Final Year Project — Complete Defence Deck

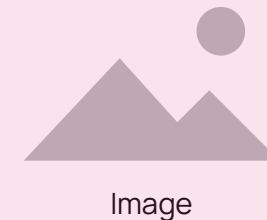


All project chapters in one editable deck

Presenter Name · 7 · 2026

Course map

- 01 Voice-Controlled Home Automation
- 02 Air Quality Monitoring Network
- 03 Solar Power Monitoring Dashboard
- 04 Expense Tracker with Budget Insights
- 05 Earthquake-Resistant Low-Cost Housing
- 06 Fake News Detection using NLP



01

VOICE-CONTROLLED HOME AUTOMATION



Switch lights and fans with your voice

Project 1 · 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

vs

Proposed system

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

System architecture

01 **Input** Data captured from users or sensors

02 **Process** Core logic validates and analyses

03 **Store** Secure database and backups

04 **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

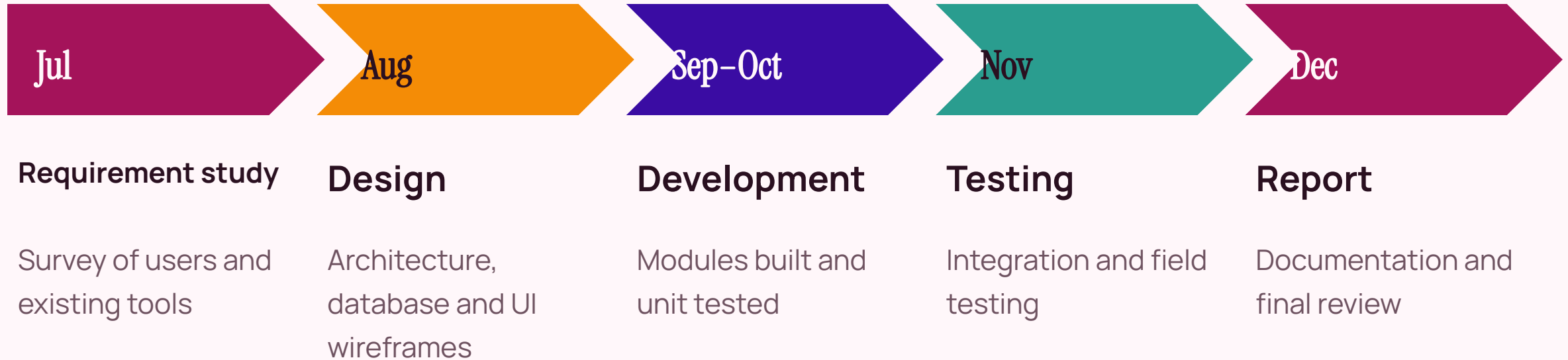
Automation flows

04

Mobile app

Manual control

Project timeline



Implementation snapshots

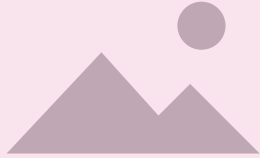


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

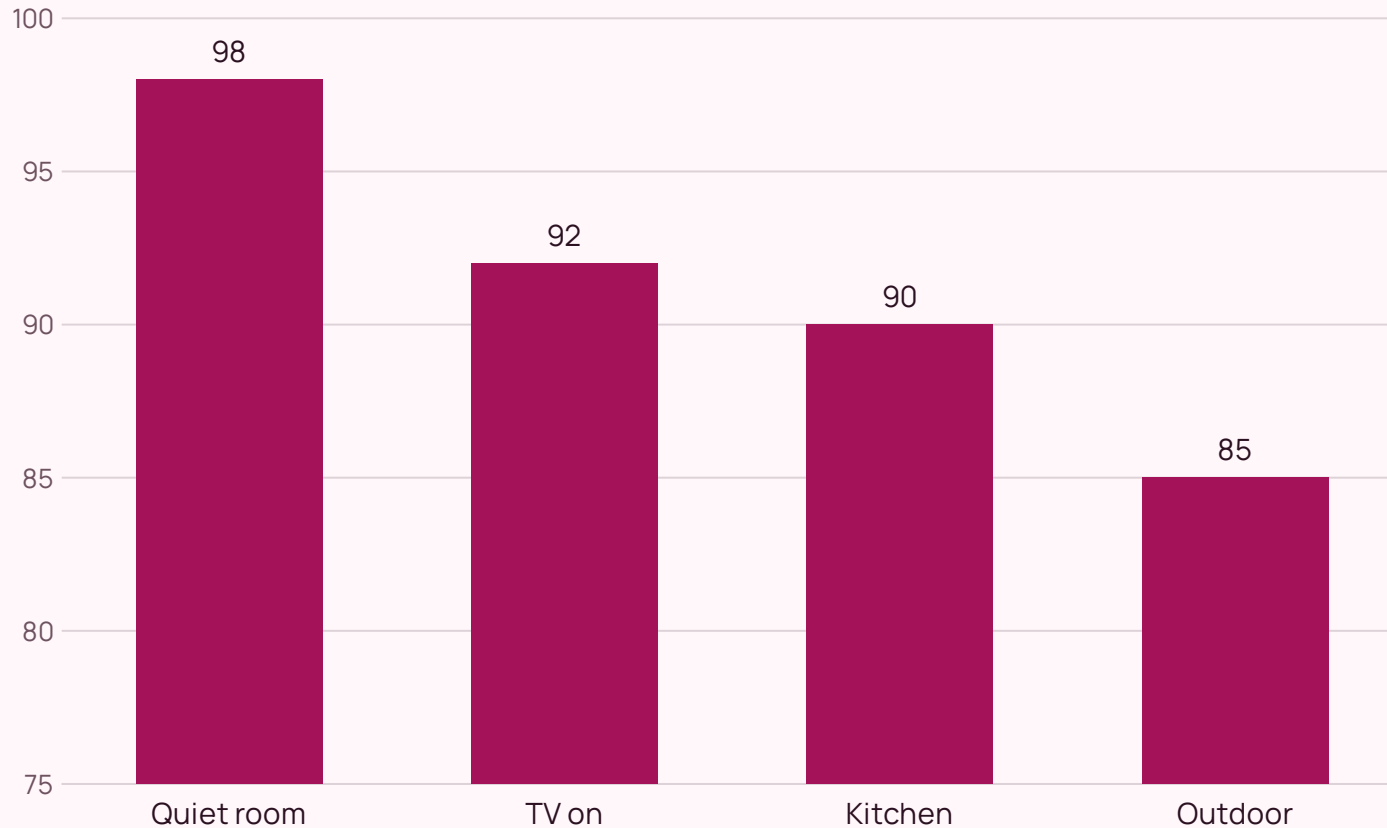


Image 3

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).

02

AIR QUALITY MONITORING NETWORK



Low-cost sensors mapping pollution across a campus

Project 2 · Air Quality Monitoring Network

Introduction

- Air pollution varies sharply over short distances
- Official monitoring stations are sparse and expensive
- Low-cost sensor nodes can fill the gaps
- We deployed a small network and a live map

CONTEXT

PM2.5 particles are small enough to enter the bloodstream.

Problem statement

01

Sparse data

One station may cover a whole district.

02

High cost

Reference monitors cost lakhs of rupees each.

03

Low awareness

People rarely see local, live readings.

Objectives

- Build nodes measuring PM2.5, PM10, CO₂ and humidity
- Calibrate against a reference instrument
- Publish a live campus air-quality map
- Send alerts when levels cross safe limits

Literature survey

Study	Approach	Limitation
Kim et al.	Low-cost PM sensors	Drift without calibration
City AQ networks	Dense sensor grids	Proprietary platforms
Patel & Joshi	LoRa-based nodes	Small pilot only

Representative summaries – replace with your own citations.

Existing vs proposed system

Existing system

- Few official stations
- Hourly, delayed data
- No local alerts

vs

Proposed system

- Dense low-cost node network
- Minute-level live map
- Threshold alerts

System architecture

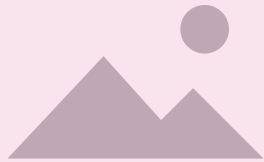
01 Measure PMS5003 and CO₂ sensors in each node

02 Transmit LoRa sends readings to a gateway

03 Calibrate Correction model on the server

04 Visualise Live map and alerts

Block diagram



Block diagram / flowchart

Every module talks through one simple, testable interface.

Solar-powered nodes send readings over LoRa to a gateway, which forwards them to a time-series database. A calibration model corrects drift before the map is updated.

Technology stack

01

ESP32 + LoRa

Node controller and
radio

02

PMS5003

Particulate matter
sensor

03

InfluxDB

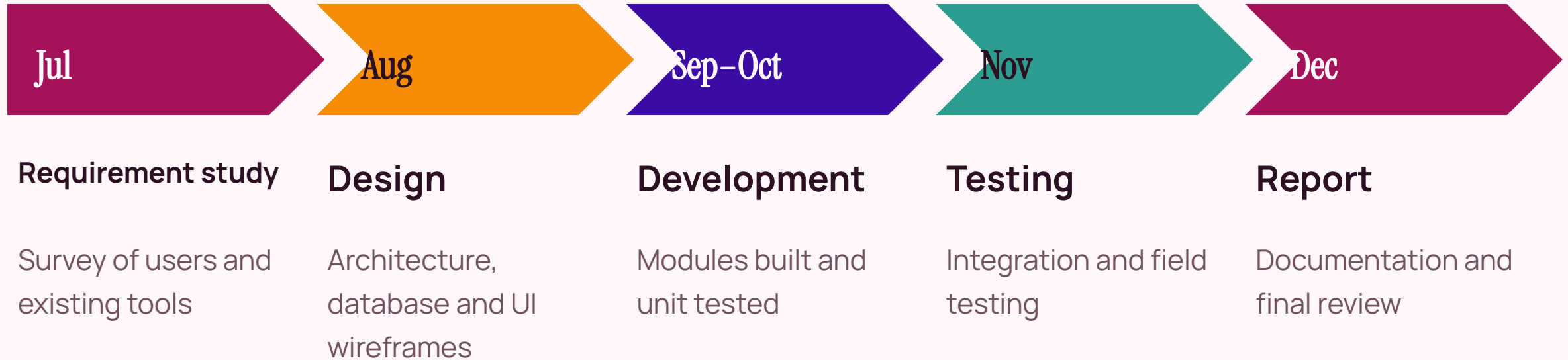
Time-series storage

04

Leaflet

Interactive map

Project timeline



Implementation snapshots

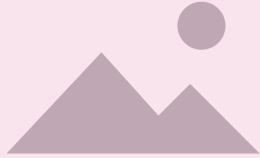


Image 1

Sensor node

Weatherproof enclosure with solar panel



Image 2

Gateway

Rooftop LoRa receiver

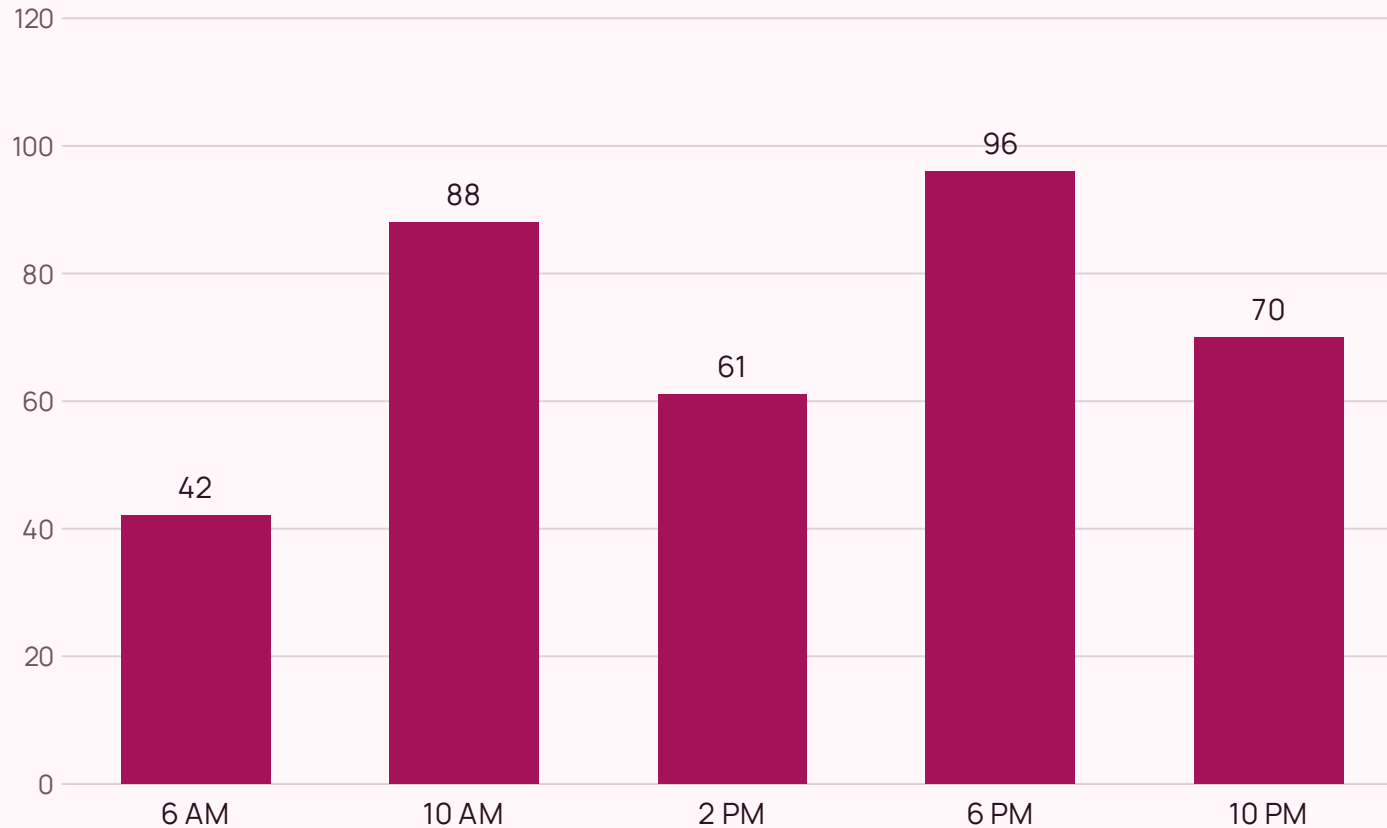


Image 3

Live map

Colour-coded AQI by location

Results



Sample data for illustration – replace with your own figures.

0.91

correlation with the reference monitor

Readings peak with morning and evening traffic, matching the reference station trend.

Key outcomes

12

nodes deployed

0.91

correlation after calibration

1 min

update interval

Figures from our test setup (sample data).

Conclusion

- Low-cost nodes give useful, local air-quality data
- Calibration is essential for accuracy
- A live map raises awareness quickly

Future scope

01

City scale

Expand to nearby wards

02

Forecasting

Predict next-day AQI

03

Health alerts

Notify sensitive groups

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

SOLAR POWER MONITORING DASHBOARD



Live generation tracking for rooftop solar

Project 3 · 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

vs

Proposed system

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

System architecture

01

Input

Data captured from users or sensors

02

Process

Core logic validates and analyses

03

Store

Secure database and backups

04

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 solar power monitoring dashboard simple to maintain and extend.



Block diagram / flowchart

Technology stack

01

INA219 sensors

Voltage and current

02

ESP32

Data logger

03

InfluxDB

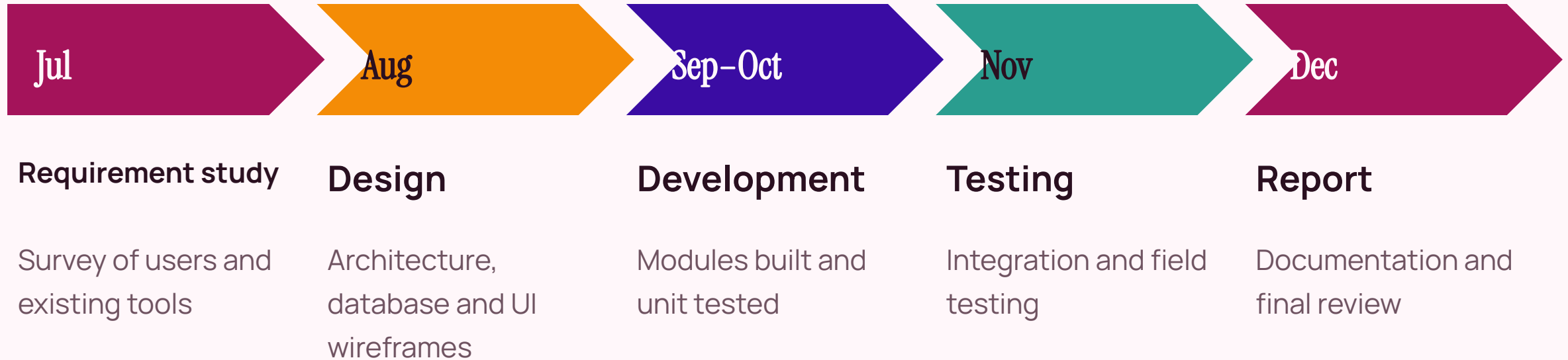
Time-series data

04

Grafana

Dashboard

Project timeline



Implementation snapshots

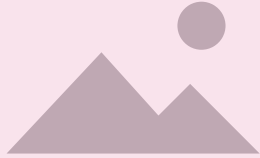


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

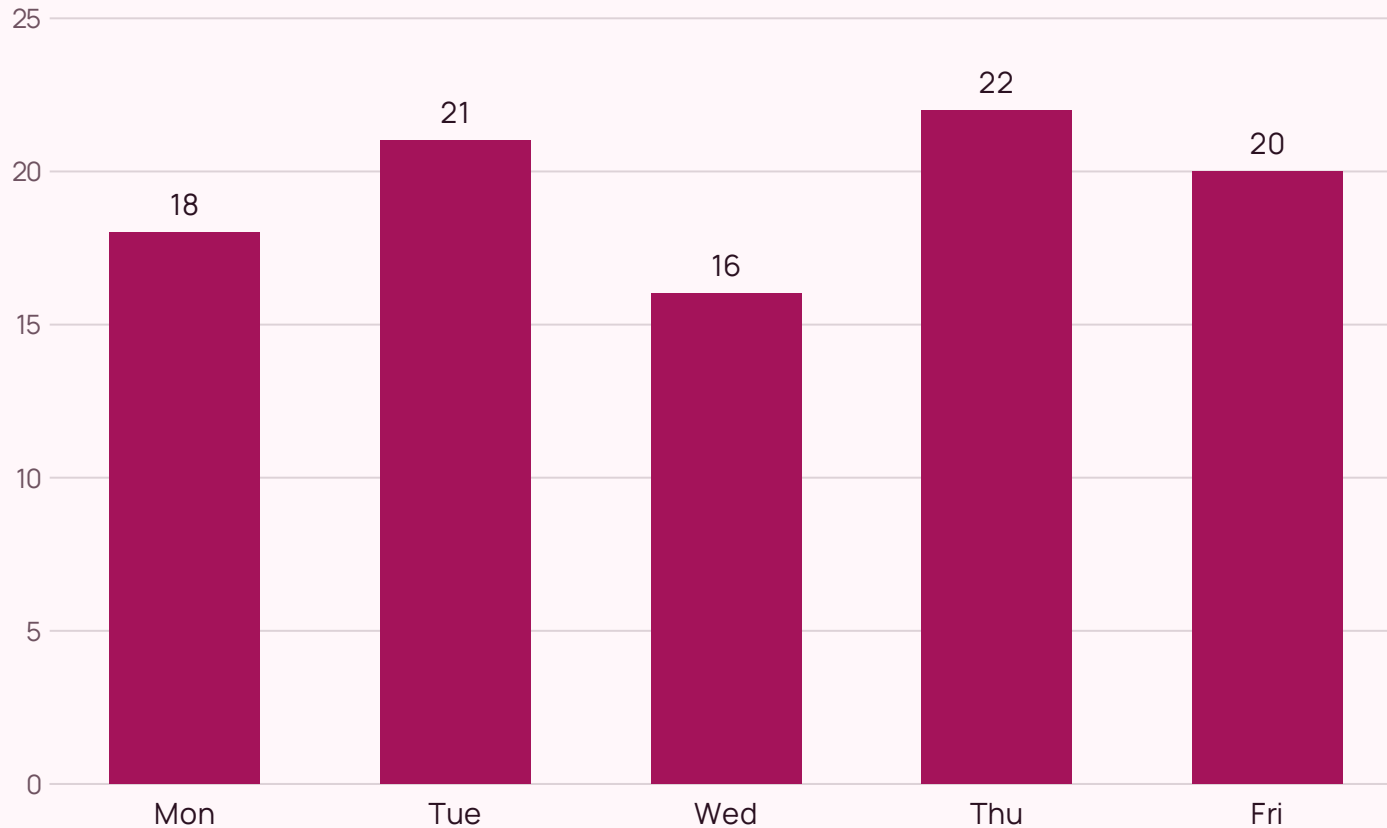


Image 3

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).

04

EXPENSE TRACKER WITH BUDGET INSIGHTS



Personal finance app for students

Project 4 · Expense Tracker with Budget Insights

Introduction

- Tracking personal expenses matters to college students every day
- Current methods are manual, slow and hard to scale
- Affordable technology now makes automation practical
- This project builds expense tracker with budget insights as a working prototype

CONTEXT

Most college students we surveyed still rely on paper or spreadsheets for tracking personal expenses.

Problem statement

01

Manual work

Tracking personal expenses takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

College students lack data to make decisions.

Objectives

- Automate the core steps of tracking personal expenses
- Give college students 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

vs

Proposed system

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

System architecture

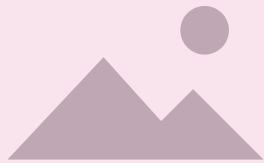
01 Input Data captured from users or sensors

02 Process Core logic validates and analyses

03 Store Secure database and backups

04 Output Dashboard, alerts and reports

Block diagram



Block diagram / flowchart

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 expense tracker with budget insights simple to maintain and extend.

Technology stack

01

Kotlin

Android app

02

Room DB

Offline storage

03

Firebase

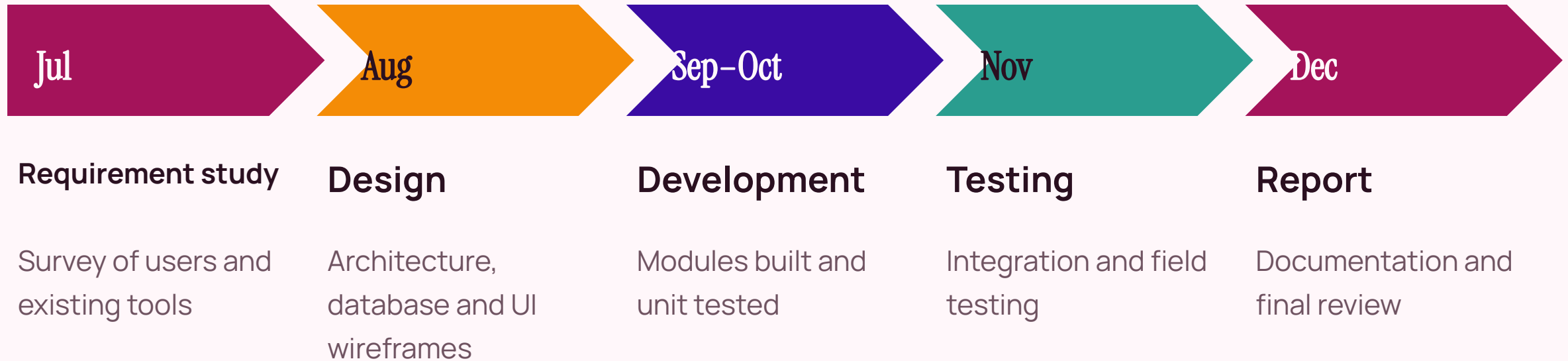
Sync and backup

04

MPAndroidChart

Spending charts

Project timeline



Implementation snapshots

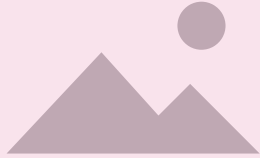


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

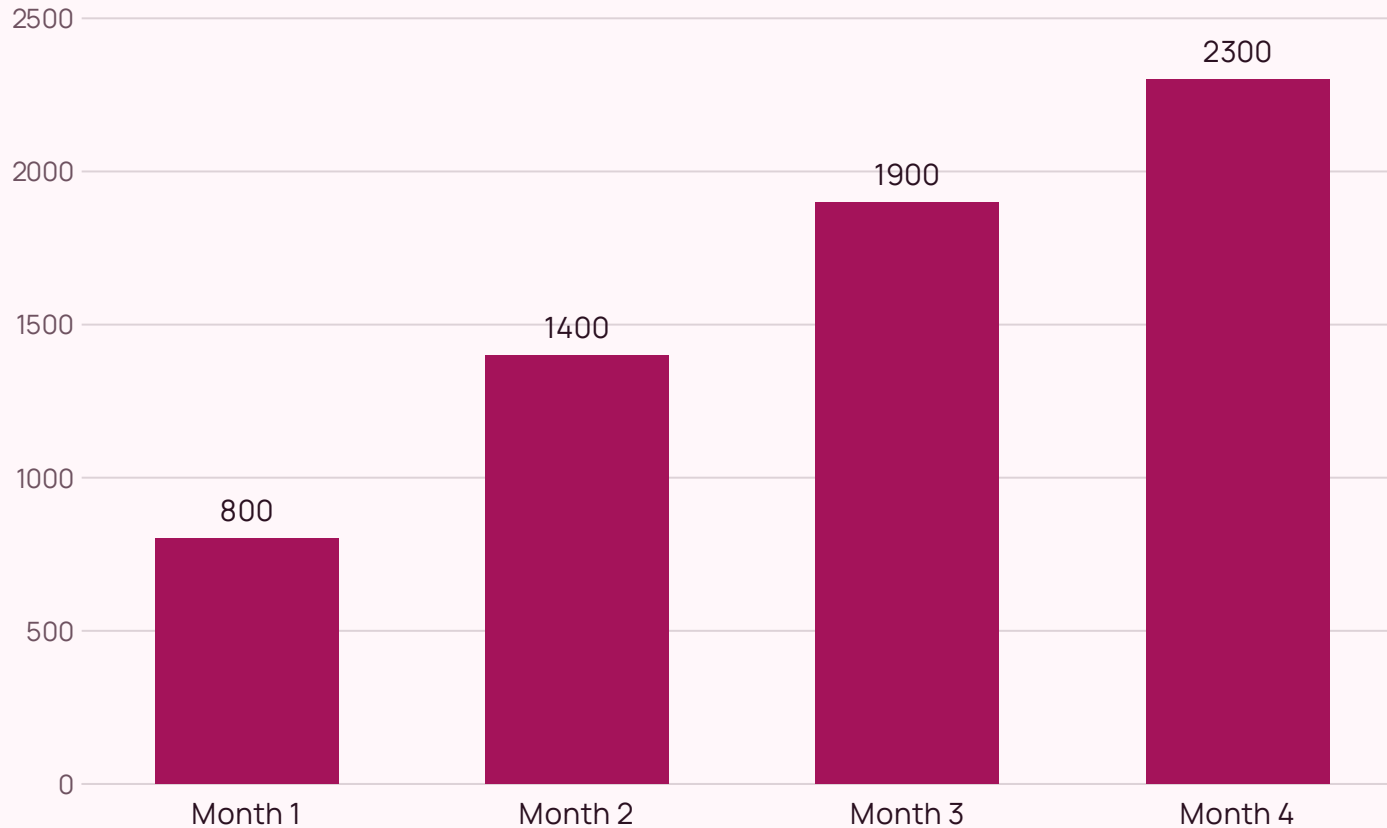


Image 3

Reports

Exportable summaries

Results



2.9×

more monthly savings

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

Sample data for illustration – replace with your own figures.

Key outcomes

2.9×

more monthly savings

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- Expense Tracker with Budget Insights meets its main objectives
- Automation saves time and reduces errors for college students
- 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).

05

EARTHQUAKE-RESISTANT LOW- COST HOUSING



Design study for rural homes

Project 5 · Earthquake-Resistant Low-Cost Housing

Introduction

- Safe rural house design matters to rural families and masons every day
- Current methods are manual, slow and hard to scale
- Affordable technology now makes automation practical
- This project builds earthquake-resistant low-cost housing as a working prototype

CONTEXT

Most rural families and masons we surveyed still rely on paper or spreadsheets for safe rural house design.

Problem statement

01

Manual work

Safe rural house design takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Rural families and masons lack data to make decisions.

Objectives

- Automate the core steps of safe rural house design
- Give rural families and masons 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

vs

Proposed system

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

System architecture

01 **Input** Data captured from users or sensors

02 **Process** Core logic validates and analyses

03 **Store** Secure database and backups

04 **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 earthquake-resistant low-cost housing simple to maintain and extend.



Block diagram / flowchart

Technology stack

01

STAAD.Pro

Structural analysis

02

AutoCAD

Drawings

03

Shake table

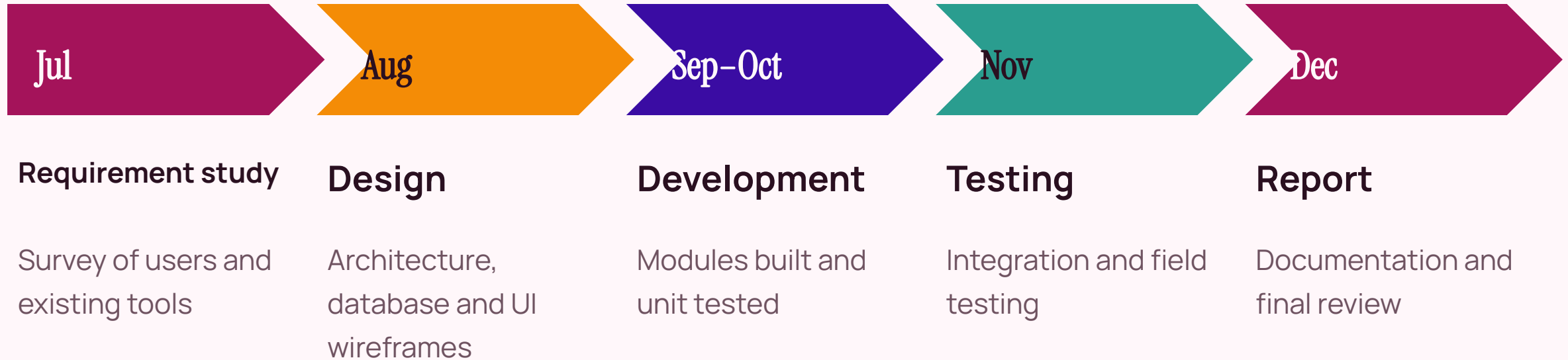
Model testing

04

Local materials

Brick and bamboo

Project timeline



Implementation snapshots

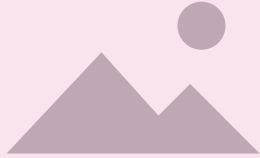


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

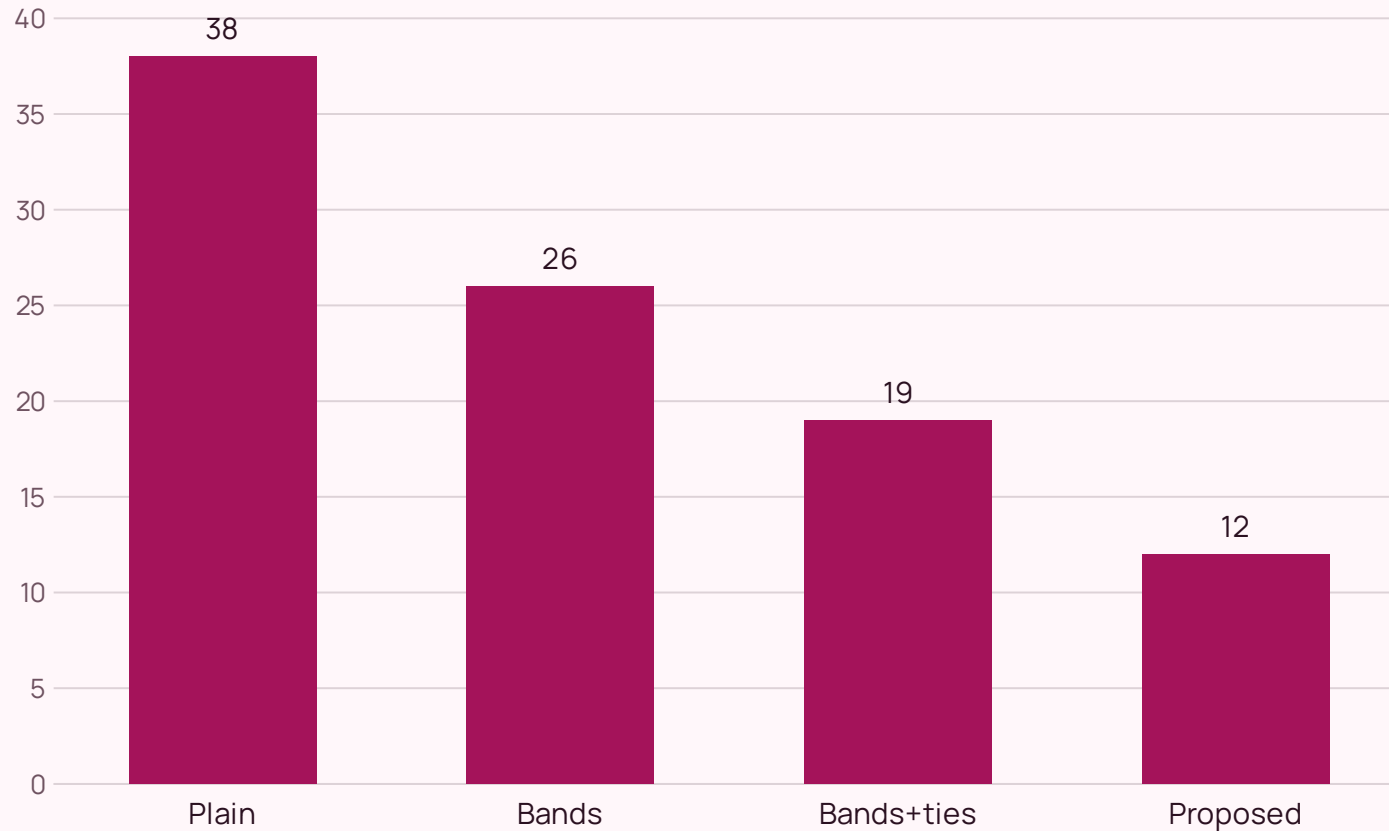


Image 3

Reports

Exportable summaries

Results



Sample data for illustration – replace with your own figures.

68%

less drift than a plain wall

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

Key outcomes

68%

less drift than a plain wall

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- Earthquake-Resistant Low-Cost Housing meets its main objectives
- Automation saves time and reduces errors for rural families and masons
- 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).

06

FAKE NEWS DETECTION USING NLP



Flagging misleading headlines automatically

Project 6 · Fake News Detection using NLP

Introduction

- Checking news credibility matters to readers and fact-checkers every day
- Current methods are manual, slow and hard to scale
- Affordable technology now makes automation practical
- This project builds fake news detection using nlp as a working prototype

CONTEXT

Most readers and fact-checkers we surveyed still rely on paper or spreadsheets for checking news credibility.

Problem statement

01

Manual work

Checking news credibility takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Readers and fact-checkers lack data to make decisions.

Objectives

- Automate the core steps of checking news credibility
- Give readers and fact-checkers 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

vs

Proposed system

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

System architecture

01

Input

Data captured from users or sensors

02

Process

Core logic validates and analyses

03

Store

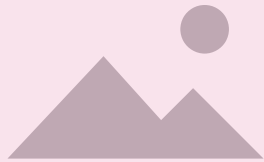
Secure database and backups

04

Output

Dashboard, alerts and reports

Block diagram



Block diagram / flowchart

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 fake news detection using nlp simple to maintain and extend.

Technology stack

01

Python

Pipeline

02

scikit-learn

Classical models

03

BERT

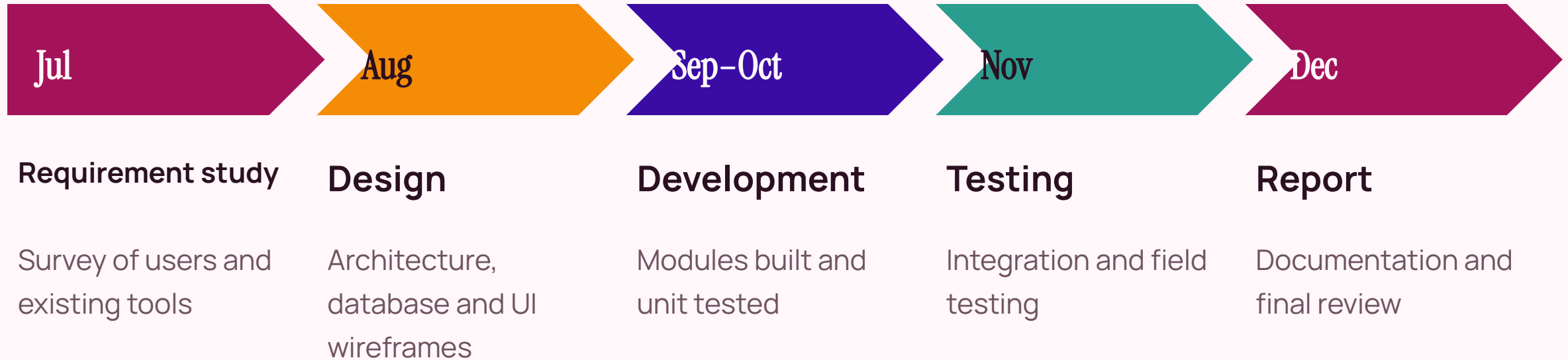
Transformer model

04

Streamlit

Demo app

Project timeline



Implementation snapshots

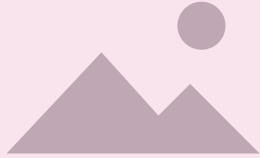


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

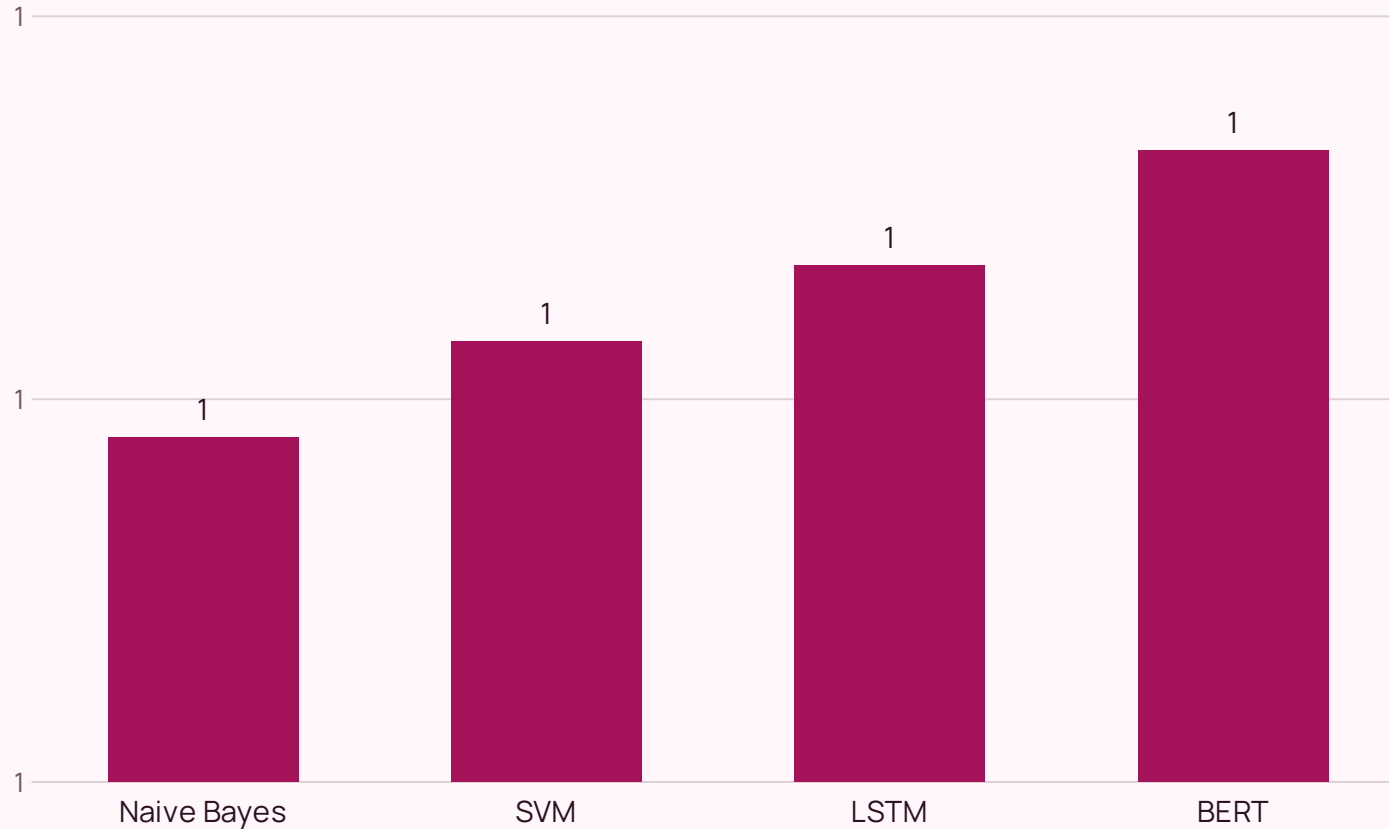


Image 3

Reports

Exportable summaries

Results



0.93

F1 score with BERT

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

Sample data for illustration – replace with your own figures.

Key outcomes

0.93

F1 score with BERT

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- Fake News Detection using NLP meets its main objectives
- Automation saves time and reduces errors for readers and fact-checkers
- 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).

07

SMART PARKING MANAGEMENT SYSTEM



Find a free parking slot in seconds

Project 7 · 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

vs

Proposed system

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

System architecture

01 **Input** Data captured from users or sensors

02 **Process** Core logic validates and analyses

03 **Store** Secure database and backups

04 **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

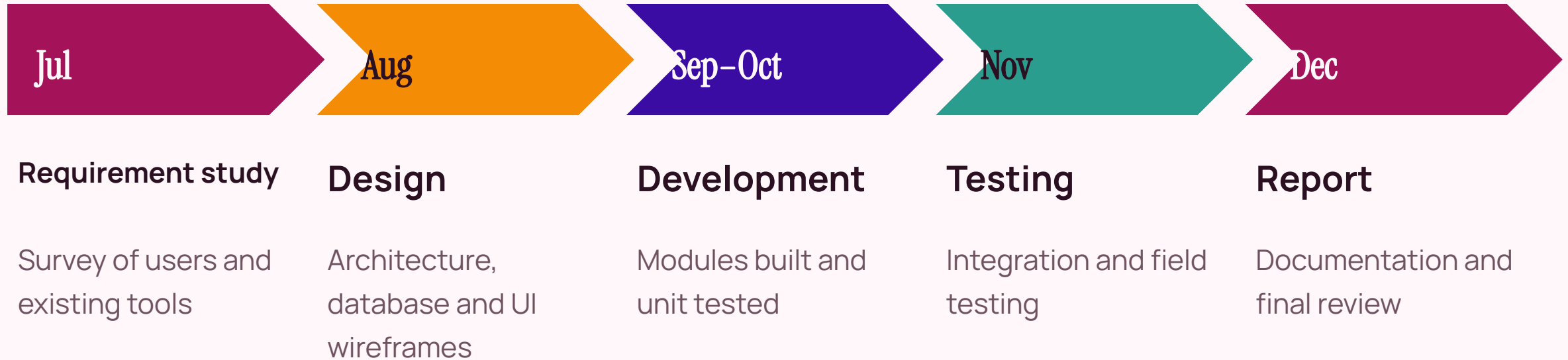
Live updates

04

Flutter

Driver app

Project timeline



Implementation snapshots

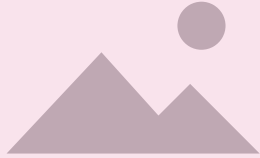


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

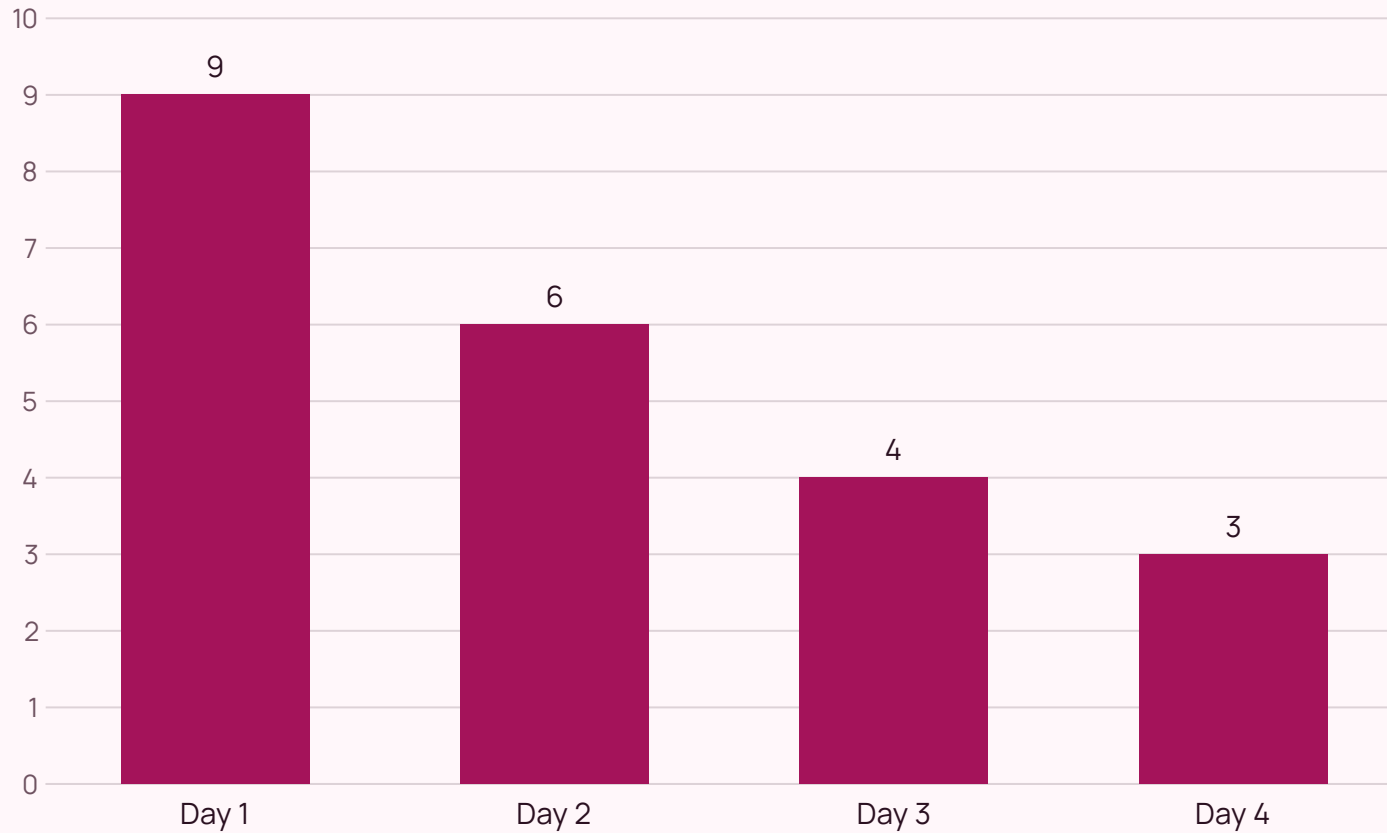


Image 3

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).

Thank you

Questions and discussion.

Presenter Name

hello@yourdomain.in