

B.TECH • FINAL YEAR

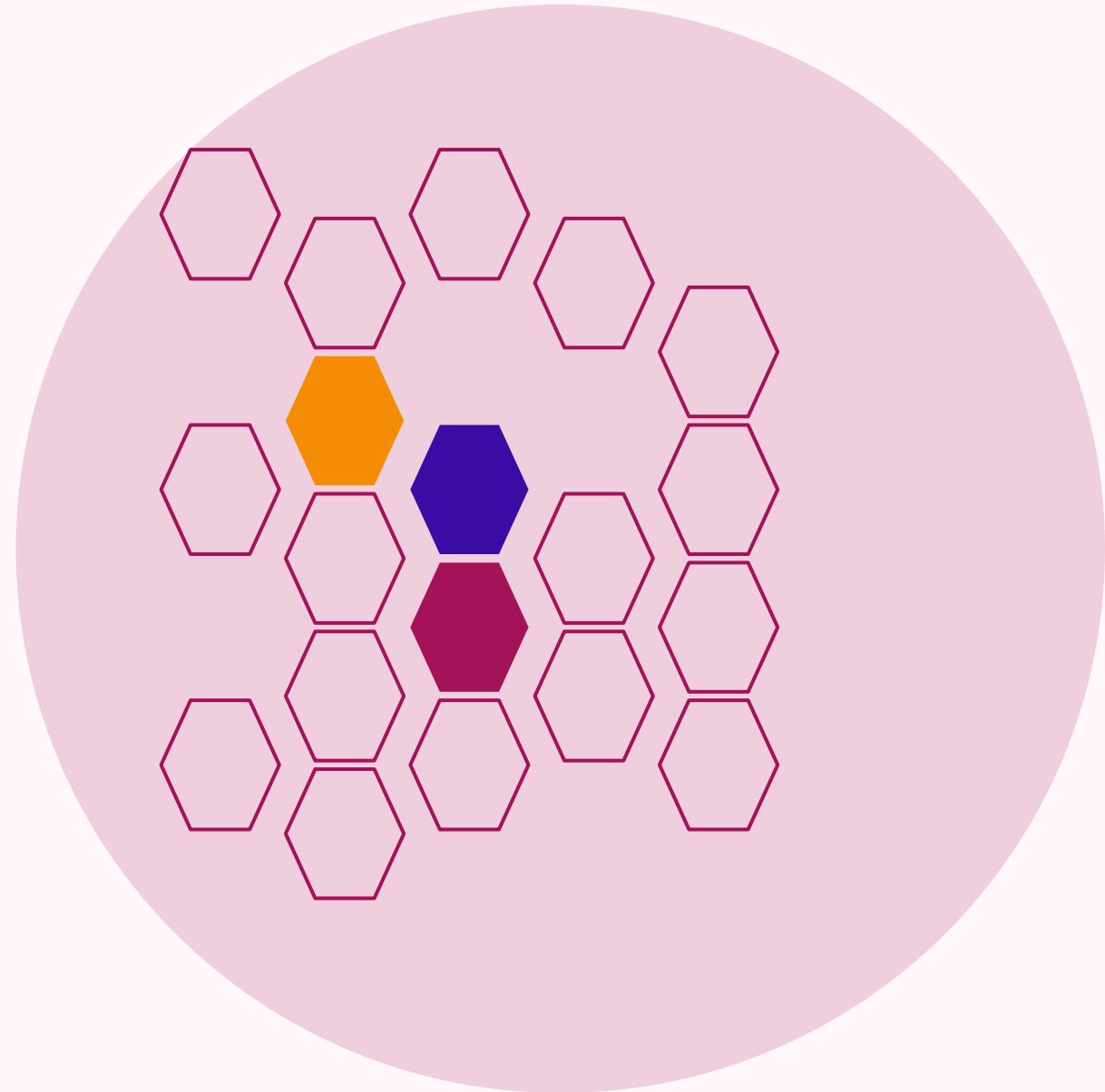
Engineering Project Showcase

All project chapters in one editable deck

PRESENTED BY
Presenter Name

MODULES
6

EDITION
2026



Course map

01

**Online Library
Management System**

02

**Solar Power Monitoring
Dashboard**

03

**Smart Irrigation System
using IoT**

04

Campus Placement Portal

05

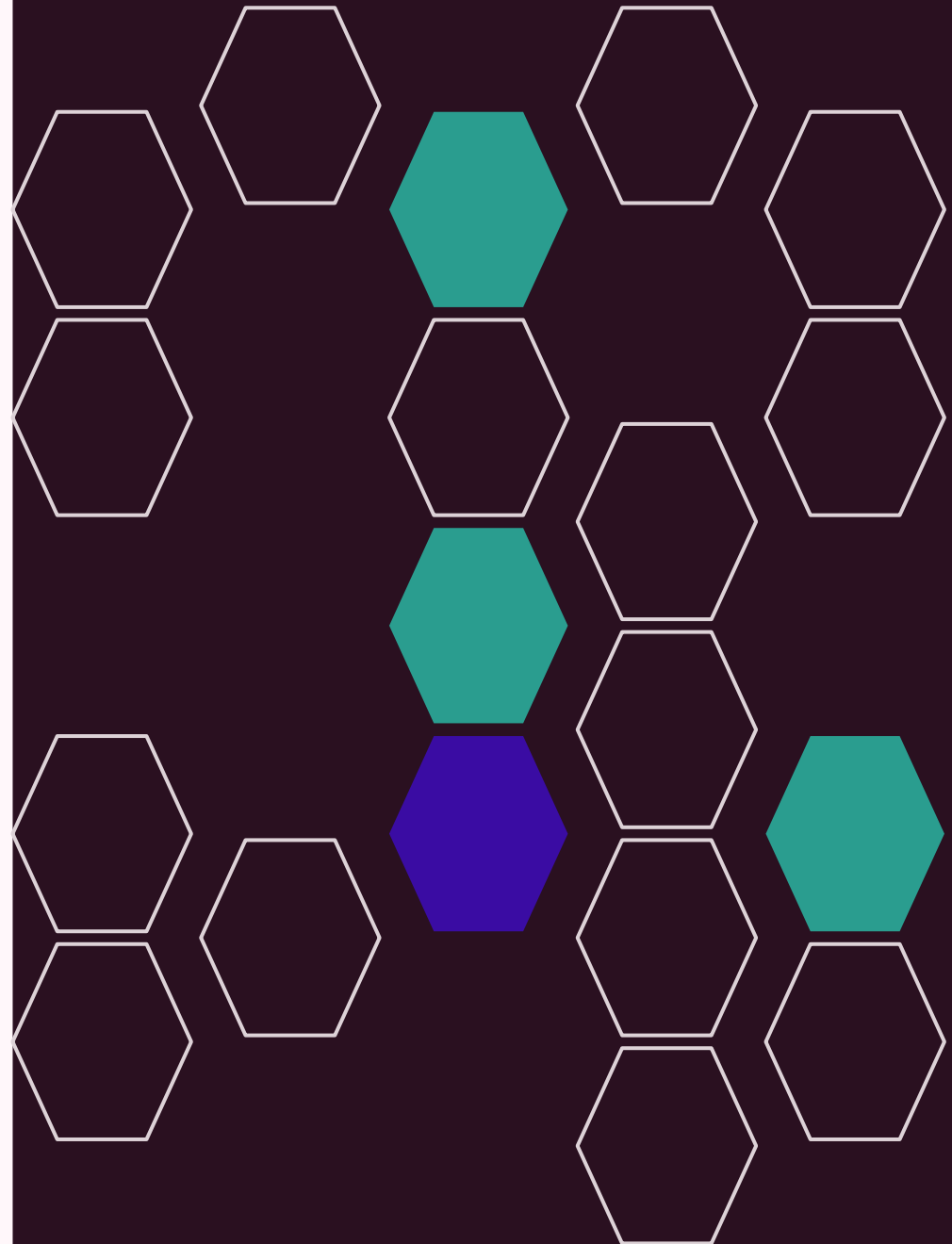
**Face Recognition
Attendance System**

06

**Air Quality Monitoring
Network**

01

ONLINE LIBRARY MANAGEMENT SYSTEM



KEY IDEA

Search, issue and return books from one web portal

Project 1 · Online Library Management System

Introduction

- College libraries manage thousands of titles and members
- Manual registers make search and tracking slow
- A web portal can automate issue, return and fines
- We designed a role-based system for students and librarians

CONTEXT

Many college libraries still track issues in paper ledgers.

Problem statement

01

Slow search

No quick way to check if a book is available.

02

Lost records

Paper ledgers get damaged or misplaced.

03

Late returns

Fines are calculated by hand and often missed.

Objectives

- Catalogue books with search by title, author and subject
- Automate issue, return and fine calculation
- Send due-date reminders by email
- Give librarians reports on stock and usage



Image

Literature survey

Study	Approach	Limitation
Koha (open source)	Full ILS suite	Complex for small colleges
Excel registers	Spreadsheet tracking	No access control
Barcode systems	Scanner-based issue	No student portal

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Paper ledgers and manual search
- ✓ Fines calculated by hand
- ✓ No reminders

Proposed system

- Searchable online catalogue
- Automatic fines and reminders
- Role-based dashboards

System architecture



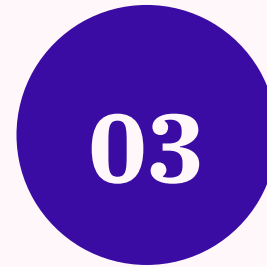
Catalogue

Books added with ISBN
and copies



Search

Students search and
reserve



Circulate

Issue and return with due
dates



Report

Stock, fines and usage
reports

Block diagram

Every module talks through one simple, testable interface.

A three-tier web application: a responsive front end, a REST API for books, members and loans, and a relational database with scheduled jobs for reminders.



Block diagram / flowchart

Technology stack

01

React

Responsive web front
end

02

Node.js

REST API and
business logic

03

MySQL

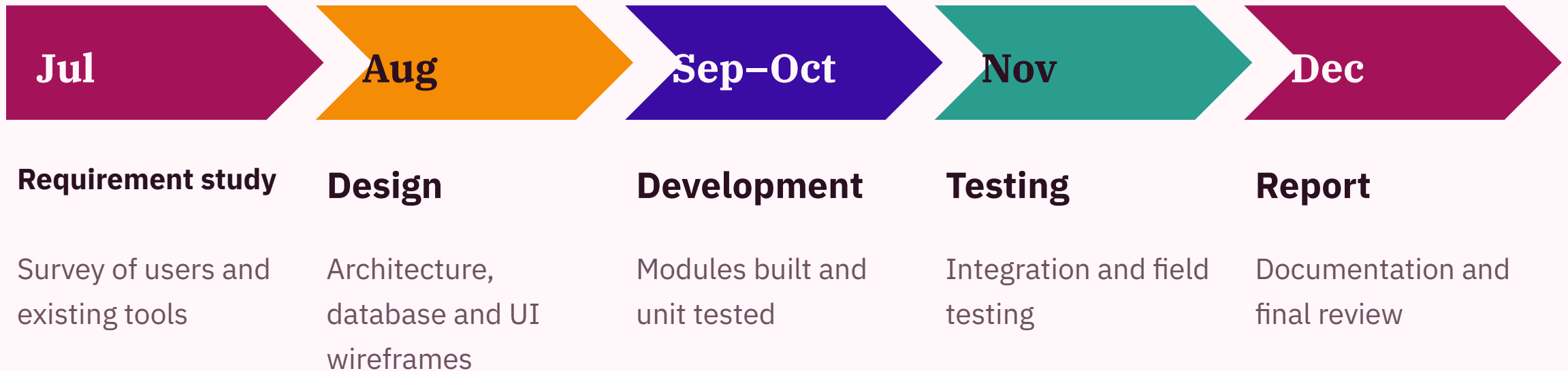
Books, members and
loans

04

Nodemailer

Due-date reminder
emails

Project timeline



Implementation snapshots



Image 1

Catalogue search

Filters by subject and availability



Image 2

Issue desk

Librarian issues a book in two clicks

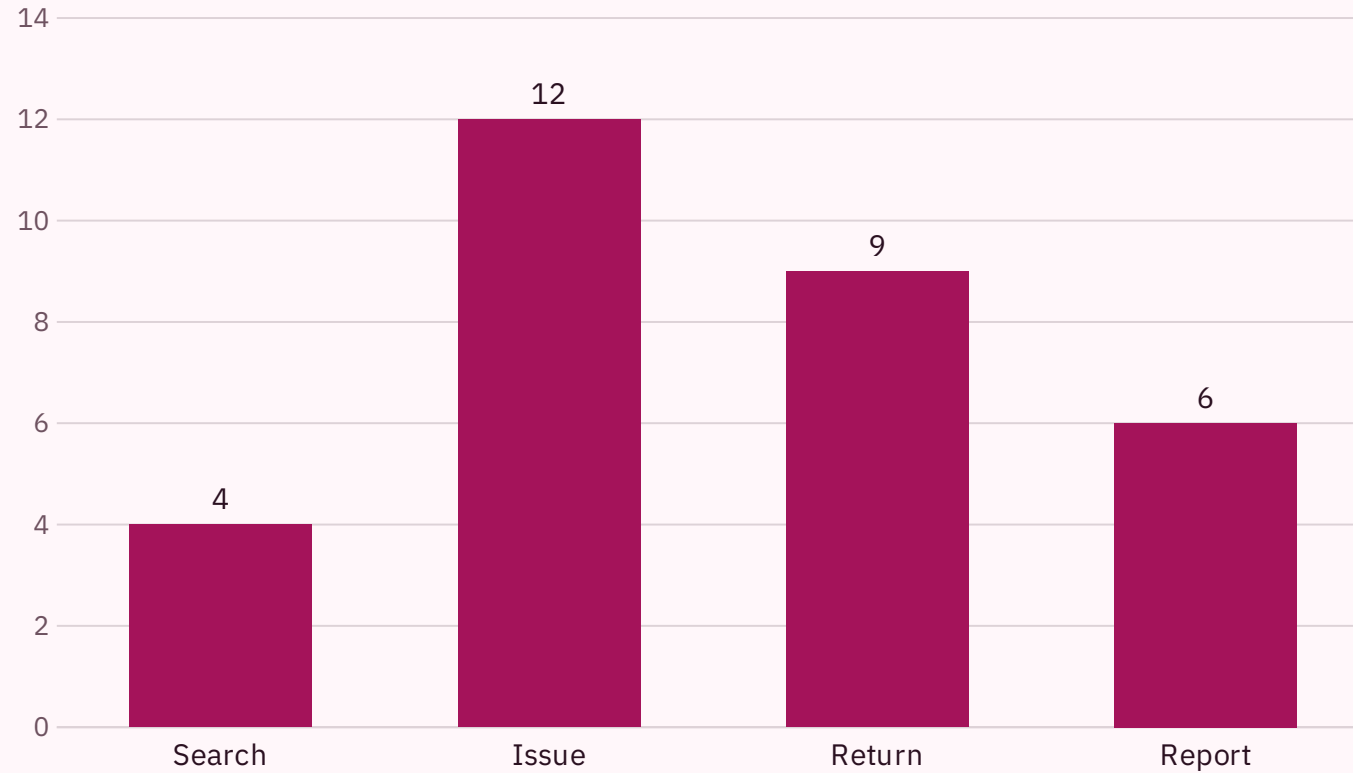


Image 3

Student account

Loans, due dates and fines

Results



Sample data for illustration — replace with your own figures.

5x

faster issue and return

Common library tasks now take seconds instead of minutes at the counter.

Key outcomes

5x

faster counter tasks

100%

loans tracked digitally

Daily

automatic reminders

Figures from our test setup (sample data).

Conclusion

- The portal replaces slow paper processes
- Automatic fines and reminders reduce late returns
- Reports help plan purchases



Image

Future scope

01

Mobile app

Scan a book to issue it

02

E-books

Digital lending section

03

Recommendations

Suggest books by course

References

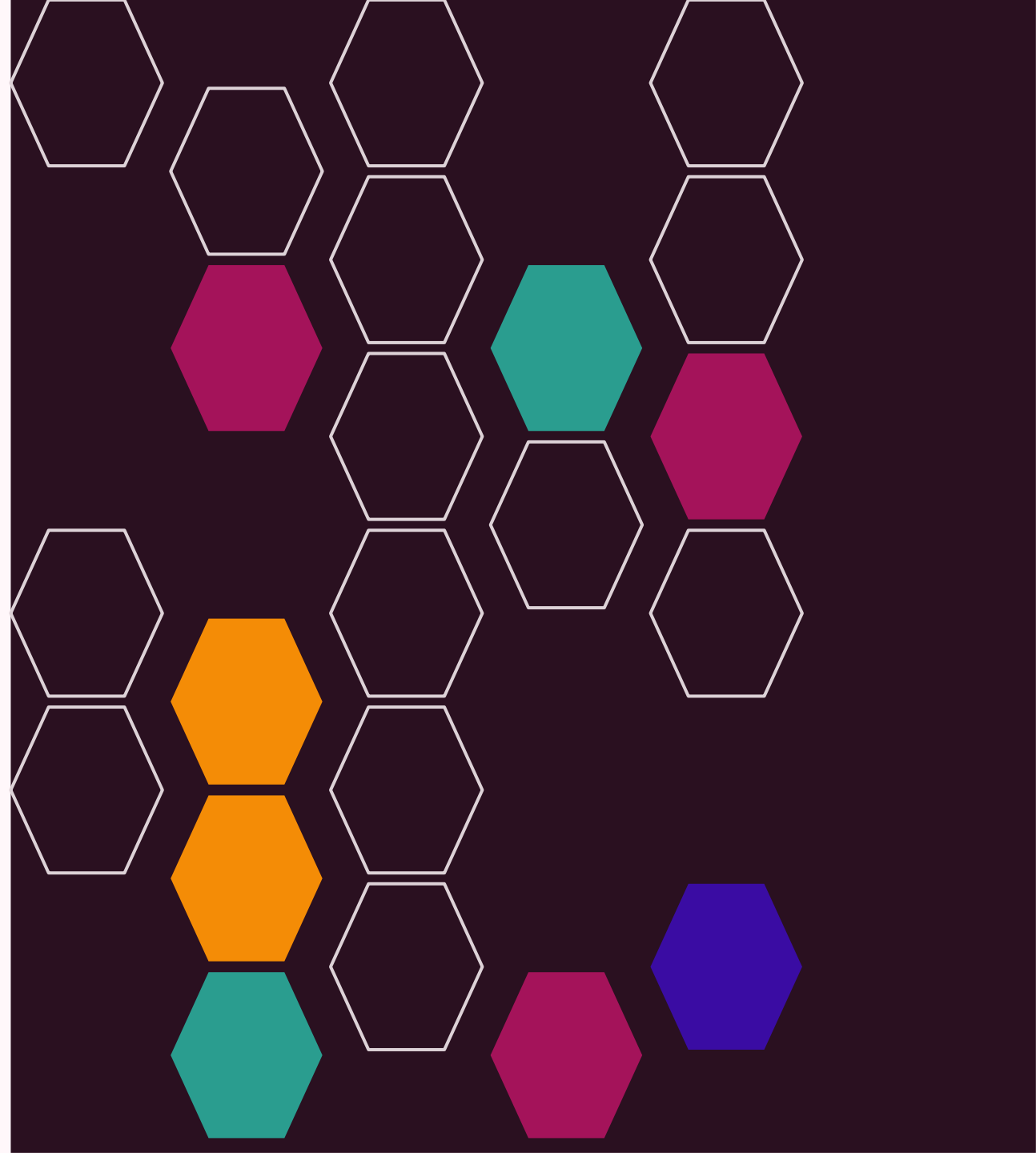
- [1] A. Author, “Title of the paper,” Journal Name, vol. 12, pp. 1–10, 2023.
- [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- [3] D. Author, “Conference paper title,” in Proc. Conference, 2024.
- [4] Official documentation of the tools used (accessed 2025).



Image

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



Image

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



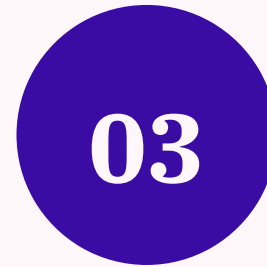
Input

Data captured from users
or sensors



Process

Core logic validates and
analyses



Store

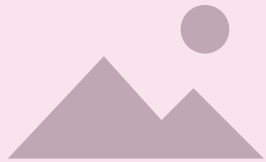
Secure database and
backups



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 solar power monitoring dashboard simple to maintain and extend.

Technology stack

01

INA219 sensors

Voltage and current

02

ESP32

Data logger

03

InfluxDB

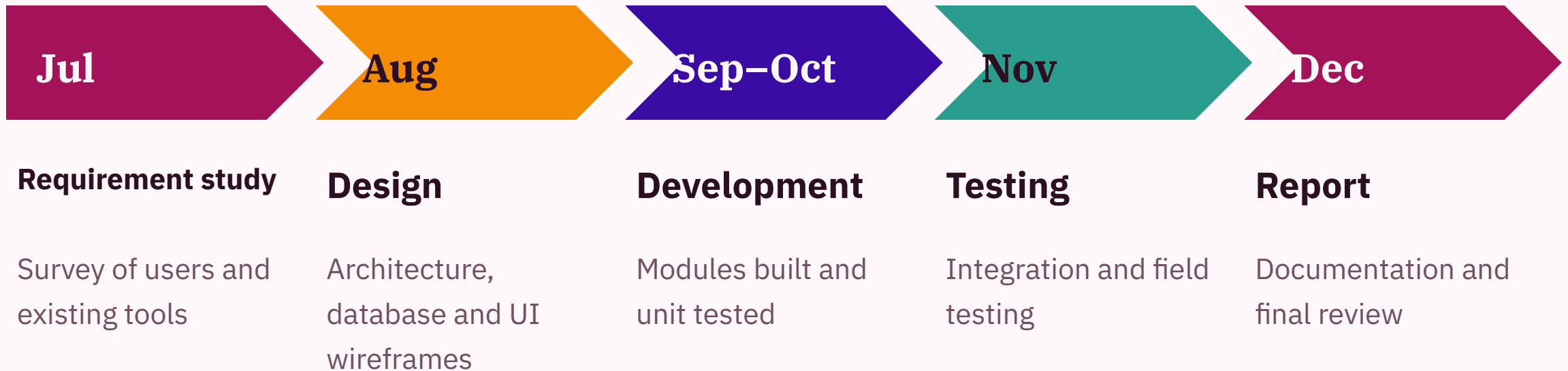
Time-series data

04

Grafana

Dashboard

Project timeline



Implementation snapshots

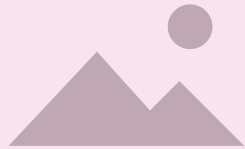


Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

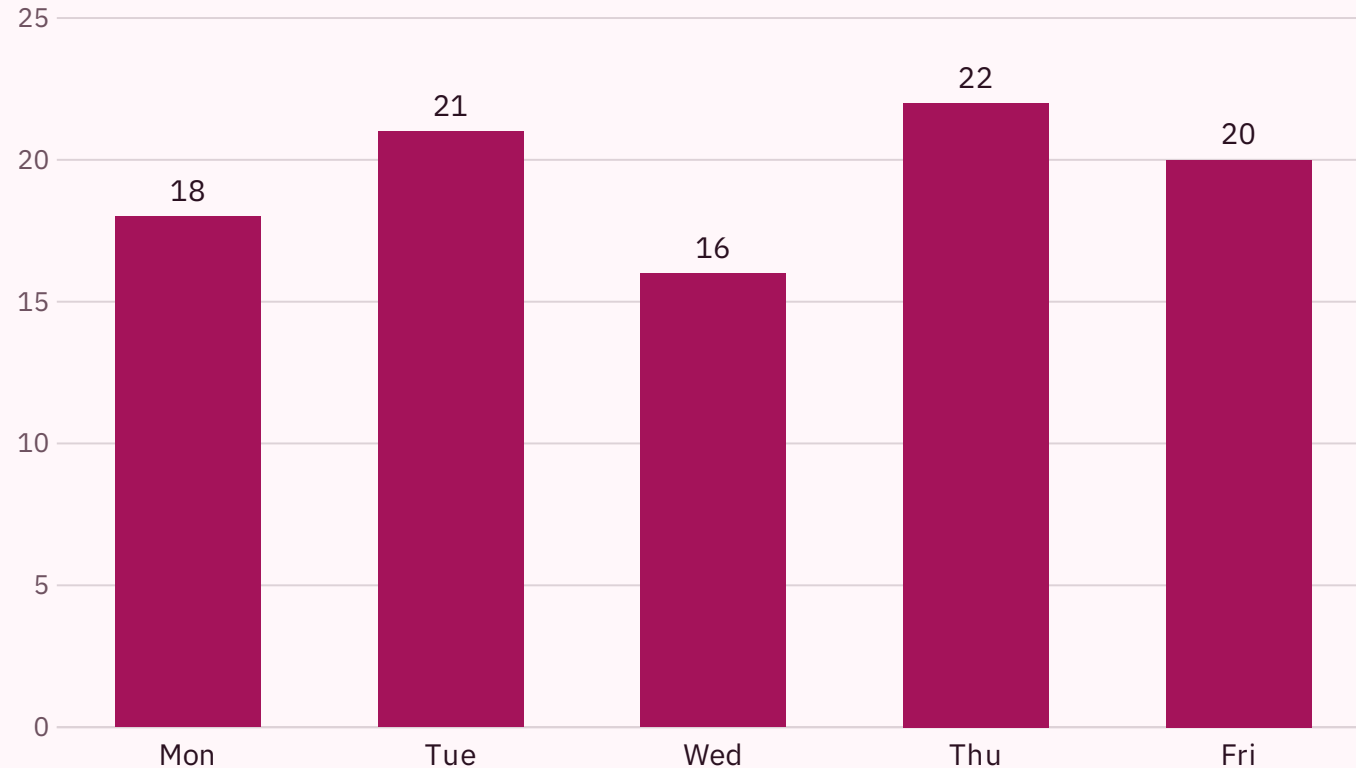


Image 3

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



Image

Future scope

01

Mobile app

Native apps for Android and iOS

02

AI insights

Predictions from historical data

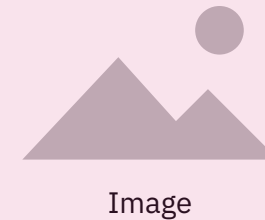
03

Scale-up

Deploy across more sites

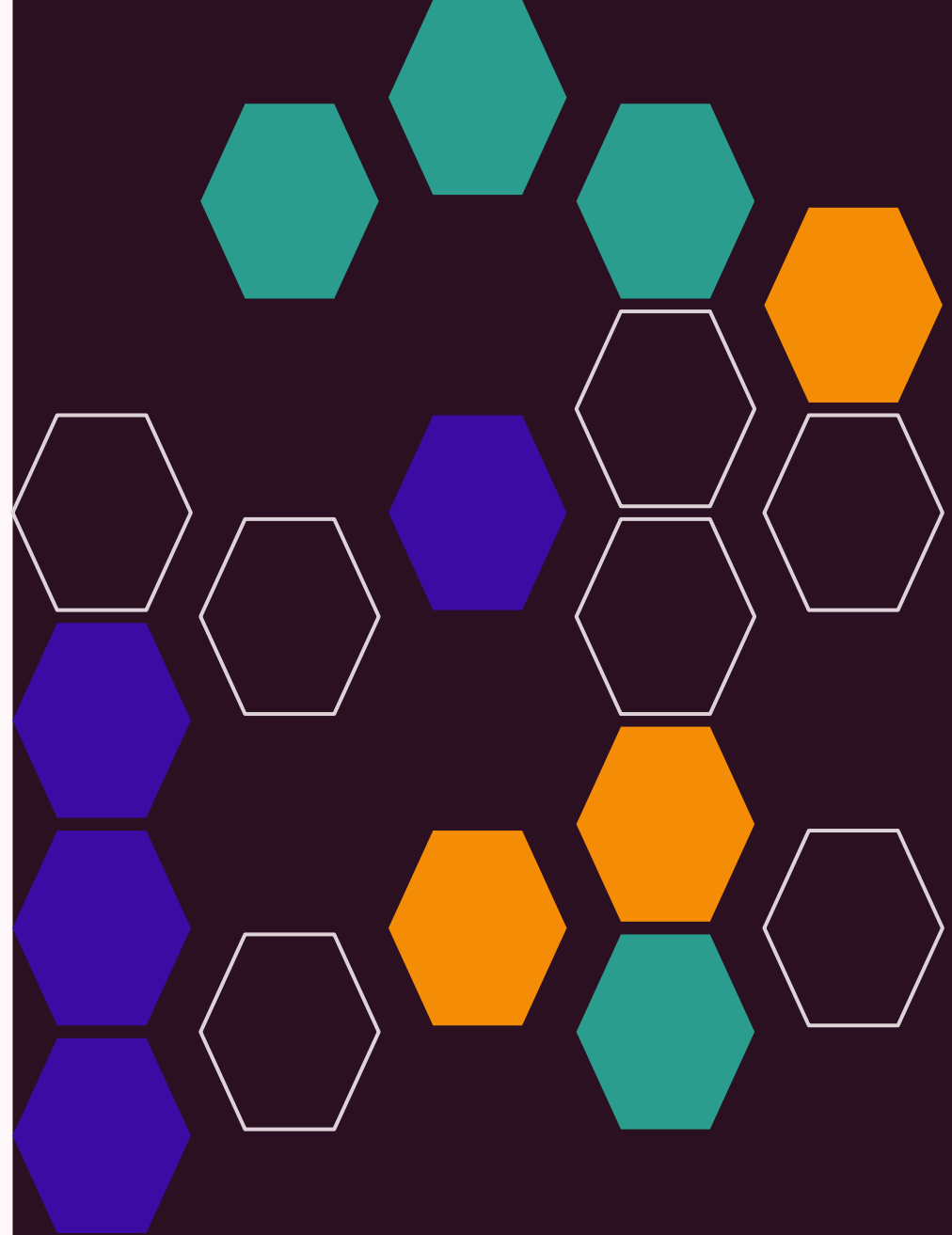
References

- [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

SMART IRRIGATION SYSTEM USING IOT



KEY IDEA

Soil-moisture driven watering for small farms

Project 3 · 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



Image

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



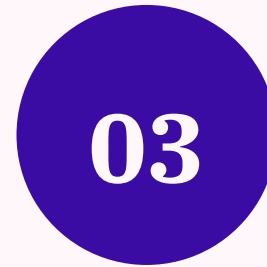
Sense

Moisture, DHT11 and rain sensors read field data



Process

ESP32 compares readings with crop thresholds



Act

Relay switches the pump on or off



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.



Block diagram / flowchart

Technology stack

01

ESP32

Wi-Fi microcontroller
running the control
loop

02

Sensors

Capacitive soil
moisture, DHT11, rain
sensor

03

Firebase

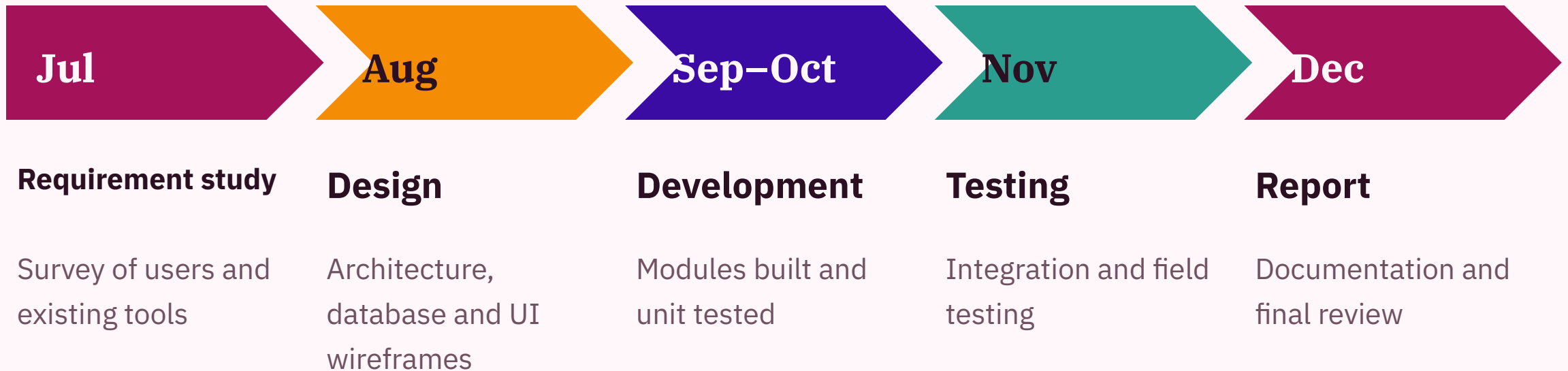
Real-time database
for readings

04

Flutter

Cross-platform
mobile dashboard

Project timeline



Implementation snapshots

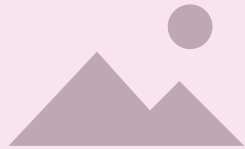


Image 1

Hardware prototype

Controller, relay and sensors on a test bed



Image 2

Field setup

Probe placement in a 10 m² test plot

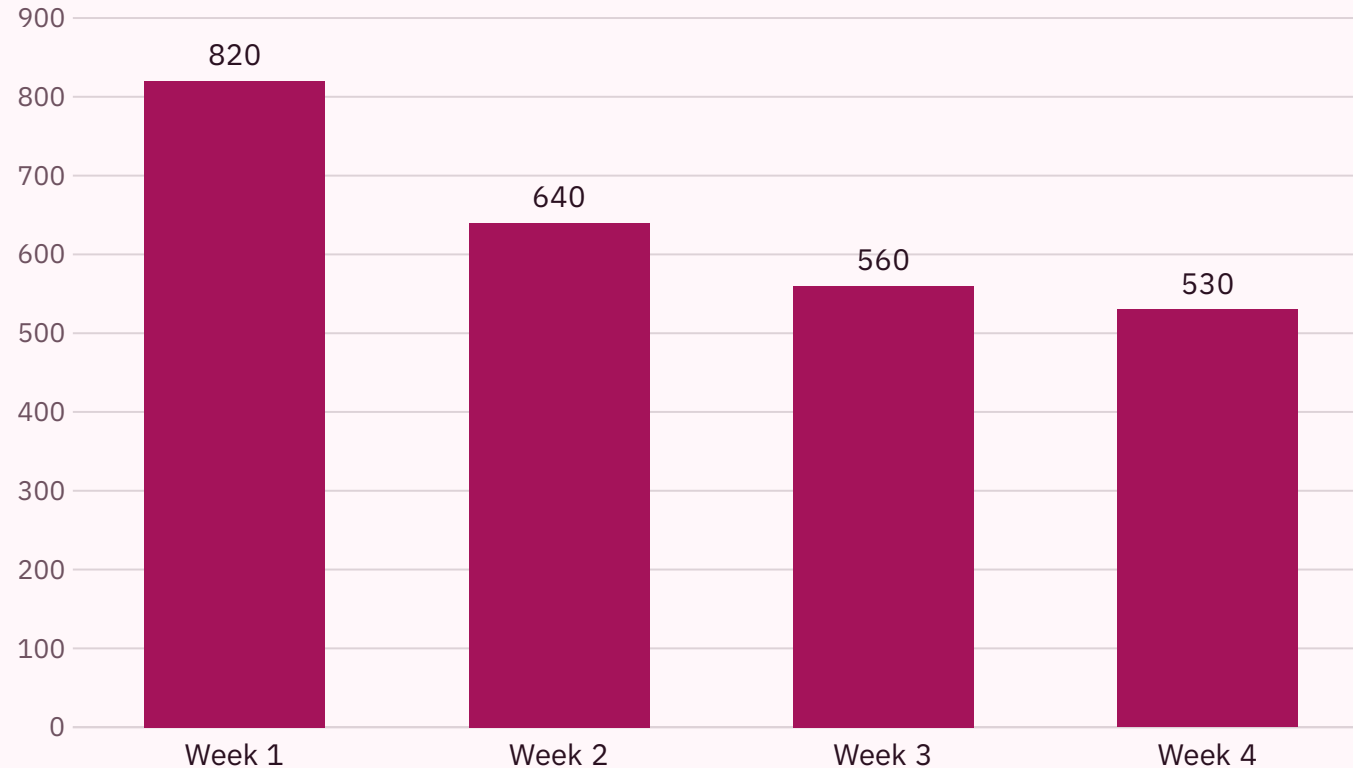


Image 3

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



Image

Future scope

01

Weather forecast

Skip watering when rain is expected

02

Solar power

Run the unit fully off-grid

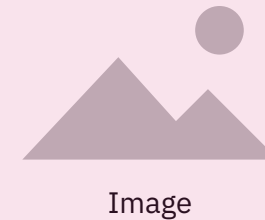
03

Multi-zone

Control several valves per farm

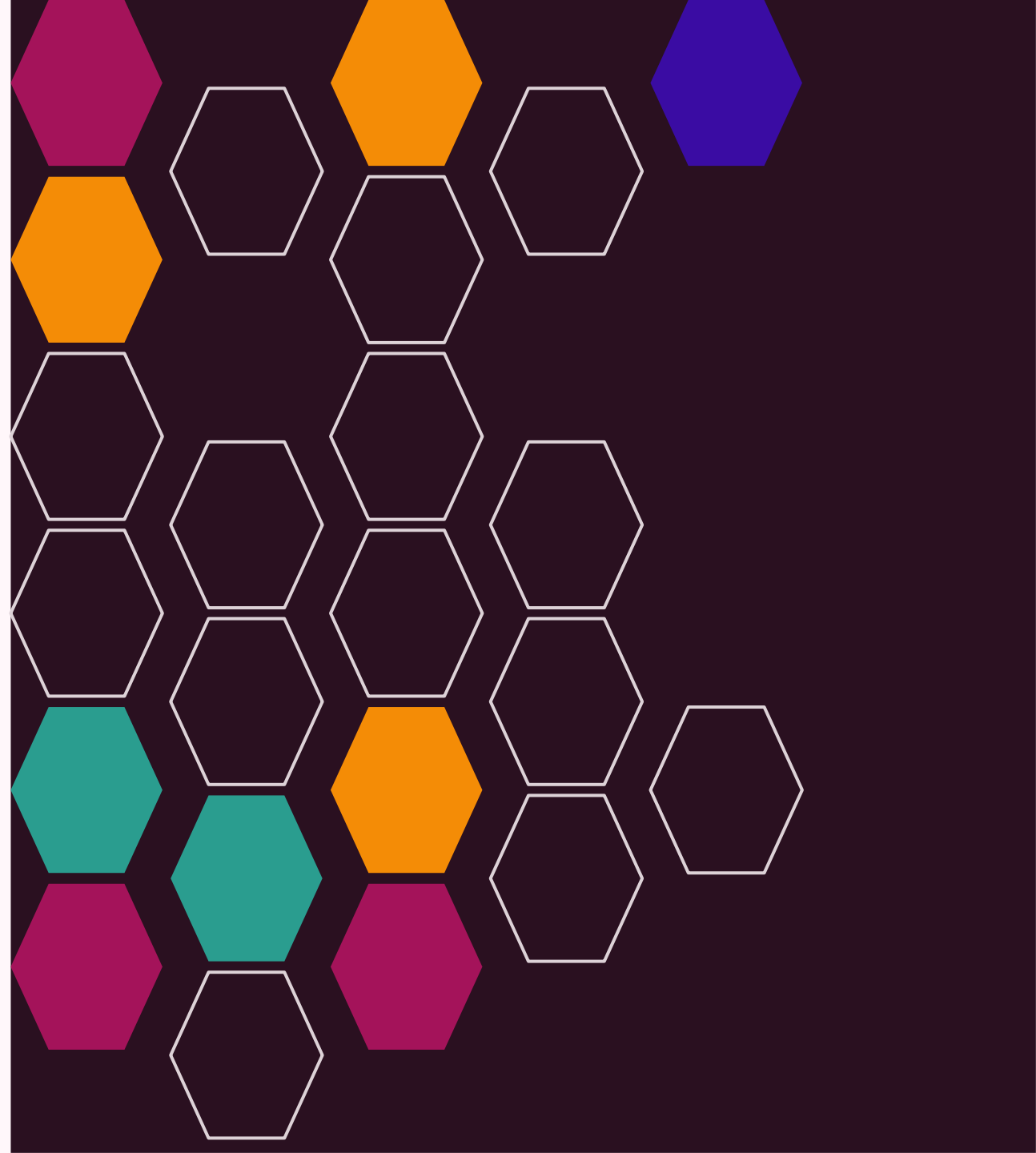
References

- [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).



04

CAMPUS PLACEMENT PORTAL



KEY IDEA

One place for drives, applications and results

Project 4 · Campus Placement Portal

Introduction

- Placement drive management matters to students and the placement cell every day
- Current methods are manual, slow and hard to scale
- Affordable technology now makes automation practical
- This project builds campus placement portal as a working prototype

CONTEXT

Most students and the placement cell we surveyed still rely on paper or spreadsheets for placement drive management.

Problem statement

01

Manual work

Placement drive management takes hours of repetitive effort.

02

Errors

Mistakes creep in and are hard to trace.

03

No insight

Students and the placement cell lack data to make decisions.

Objectives

- Automate the core steps of placement drive management
- Give students and the placement cell a simple, mobile-friendly interface
- Store data securely and generate reports
- Evaluate the system against the existing process



Image

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



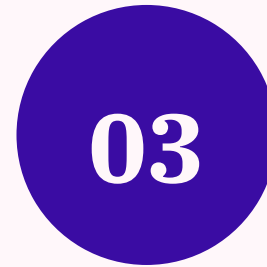
Input

Data captured from users
or sensors



Process

Core logic validates and
analyses



Store

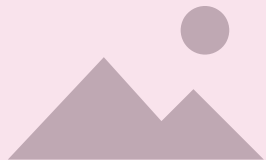
Secure database and
backups



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 campus placement portal simple to maintain and extend.

Technology stack

01

Next.js

Student and admin
portal

02

Node.js

APIs

03

MongoDB

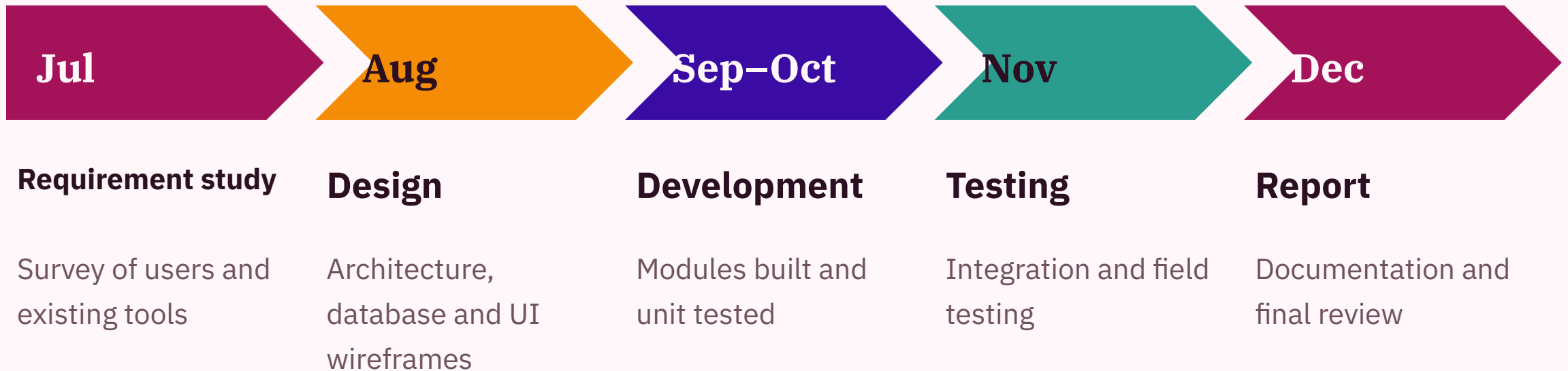
Profiles and drives

04

Mail service

Notifications

Project timeline



Implementation snapshots



Image 1

Main dashboard

Overview of live data



Image 2

Data entry

Simple forms with validation

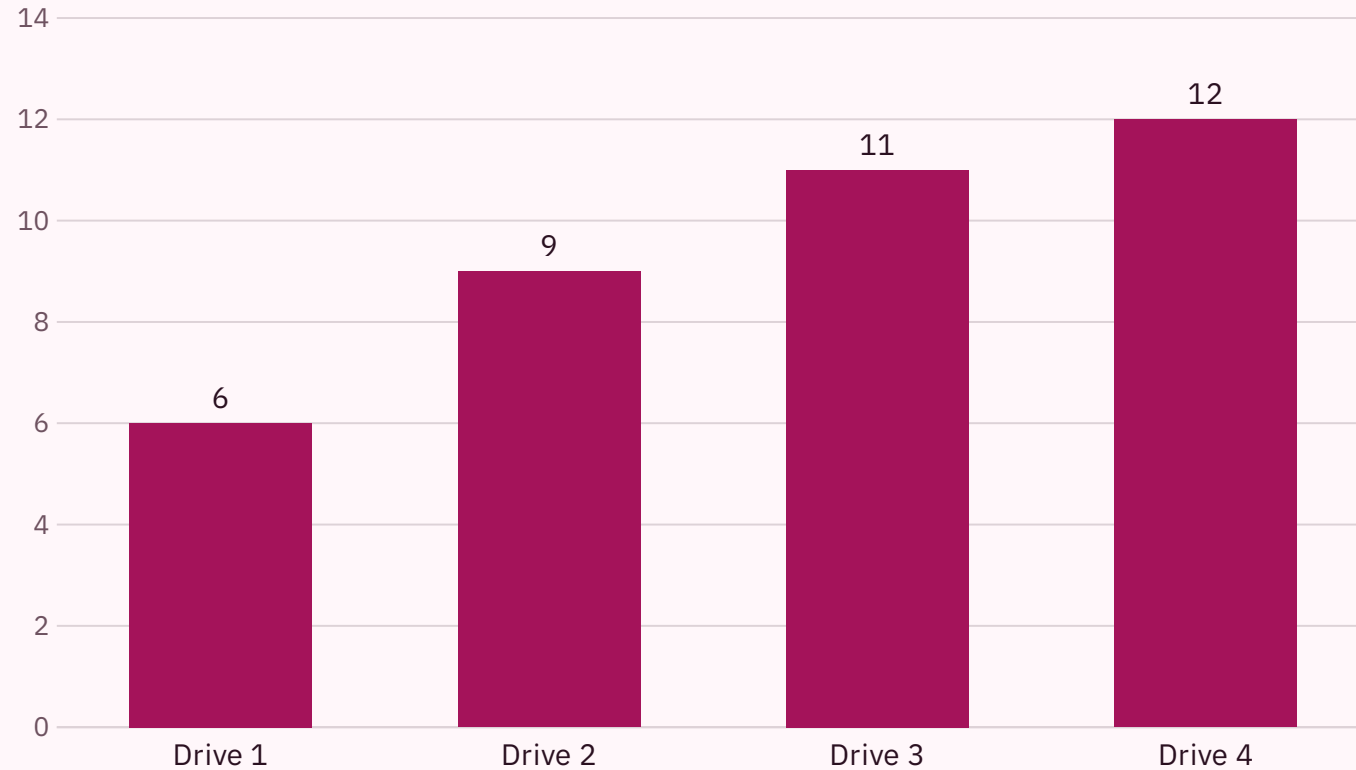


Image 3

Reports

Exportable summaries

Results



Sample data for illustration — replace with your own figures.

70%

less manual coordination

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

Key outcomes

70%

less manual coordination

100%

records digitised

< 1 s

average response

Figures from our test setup (sample data).

Conclusion

- Campus Placement Portal meets its main objectives
- Automation saves time and reduces errors for students and the placement cell
- The design is simple to extend and maintain



Image

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

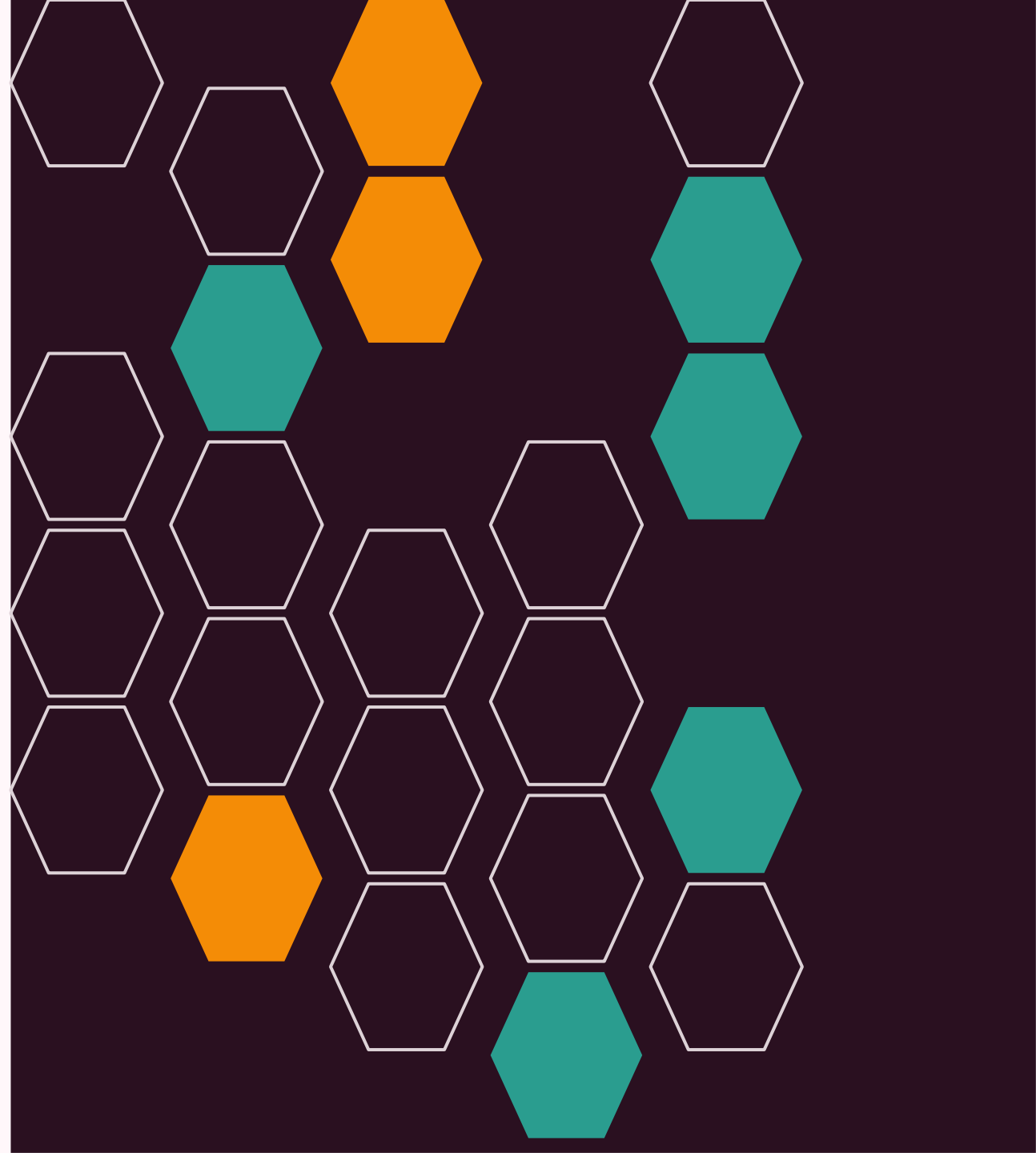
- [1] A. Author, “Title of the paper,” Journal Name, vol. 12, pp. 1–10, 2023.
- [2] B. Author and C. Author, Book Title, 2nd ed. Publisher, 2022.
- [3] D. Author, “Conference paper title,” in Proc. Conference, 2024.
- [4] Official documentation of the tools used (accessed 2025).



Image

05

FACE RECOGNITION ATTENDANCE SYSTEM



KEY IDEA

Contactless classroom attendance with deep learning

Project 5 · Face Recognition Attendance System

Introduction

- Attendance is recorded manually in most classrooms
- Roll calls waste teaching time and allow proxies
- Face recognition can mark attendance in seconds
- We built a camera-based system with a teacher dashboard

CONTEXT

Manual registers are easy to proxy and hard to analyse.

Problem statement

01

Lost time

Roll calls take minutes of every lecture.

02

Proxy attendance

Signatures and verbal responses are easy to fake.

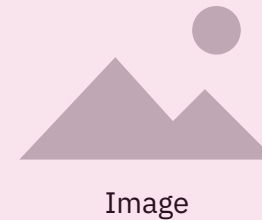
03

No analytics

Paper registers make trends hard to see.

Objectives

- Detect and recognise faces from a classroom camera
- Mark attendance automatically with a timestamp
- Generate subject-wise reports for teachers
- Respect privacy: store embeddings, not raw photos



Literature survey

Study	Approach	Limitation
Viola–Jones	Haar cascade detection	Weak in poor lighting
FaceNet	Deep embeddings	Needs GPU for training
RFID systems	Card tapping	Cards can be shared

Representative summaries — replace with your own citations.

Existing vs proposed system

Existing system

- ✓ Manual roll call or registers
- ✓ Proxy attendance possible
- ✓ Reports compiled by hand

Proposed system

- Automatic camera-based marking
- Proxy-resistant face matching
- Instant reports and exports

System architecture



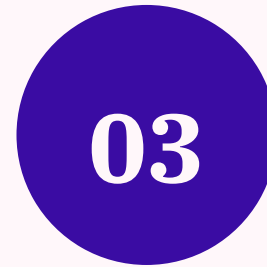
Capture

Classroom camera sends
frames



Detect

MTCNN finds faces in each
frame



Recognise

Embeddings matched
against enrolled students



Record

Attendance saved with
time and subject

Block diagram

Every module talks through one simple, testable interface.

A camera module streams frames to a local server. Faces are detected, converted to embeddings and matched to the enrolled class. Results go to a database that powers the teacher dashboard.



Block diagram / flowchart

Technology stack

01

Python

Core processing
pipeline

02

**OpenCV +
MTCNN**

Face detection

03

FaceNet

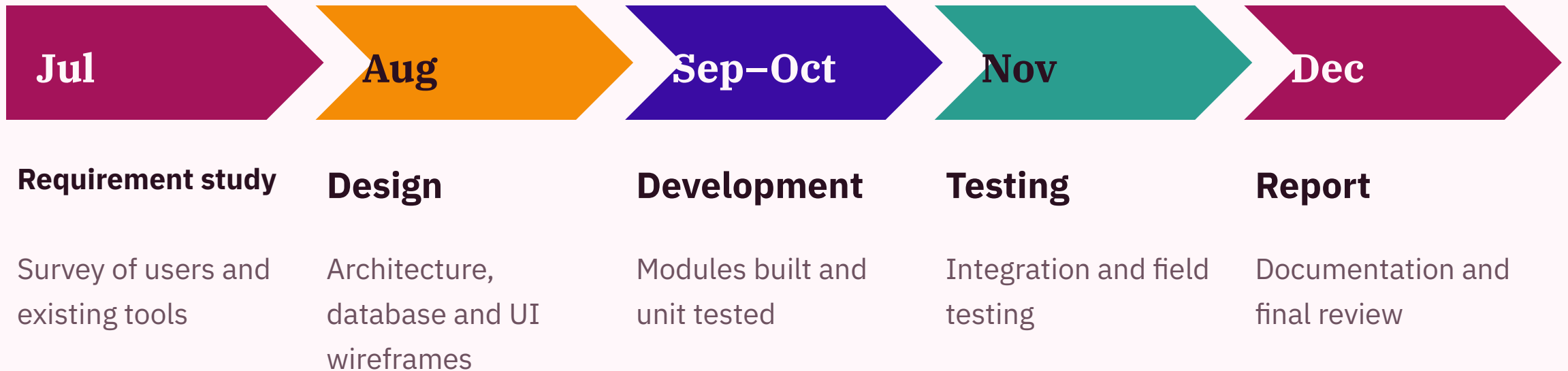
Face embeddings and
matching

04

Django

Teacher dashboard
and reports

Project timeline



Implementation snapshots



Image 1

Enrolment screen

Students registered with 5 sample images



Image 2

Live recognition

Bounding boxes with names in real time

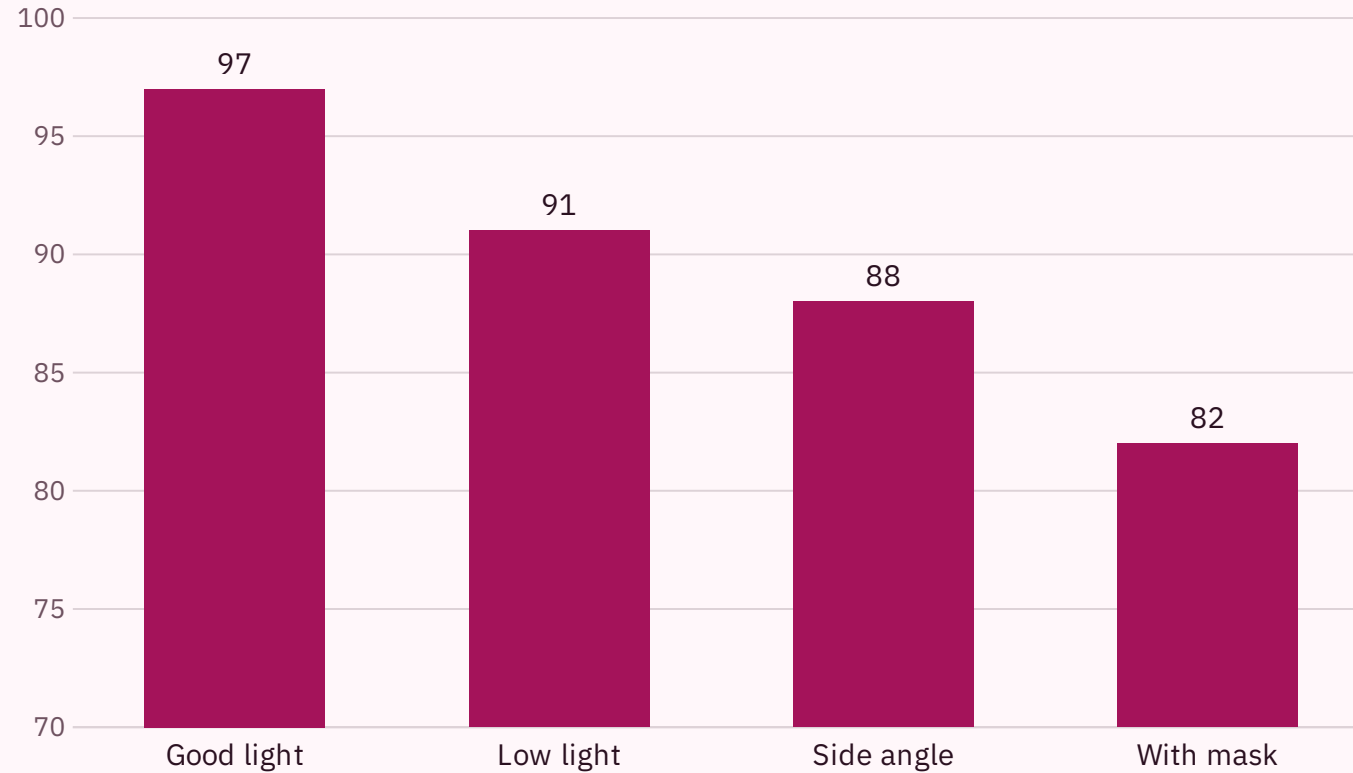


Image 3

Reports

Subject-wise attendance export

Results



Sample data for illustration — replace with your own figures.

97%

accuracy in normal lighting

Accuracy stays high in normal classrooms and drops gracefully in harder conditions.

Key outcomes

97%

recognition accuracy

15 s

to mark a full class

0

raw photos stored

Figures from our test setup (sample data).

Conclusion

- Face recognition makes attendance fast and proxy-resistant
- Embeddings keep the system privacy-friendly
- Teachers get reports without manual work



Image

Future scope

01

Mobile app

Students check their own attendance

02

Edge device

Run fully on a Raspberry Pi

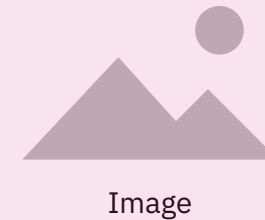
03

Liveness check

Stop photo-based spoofing

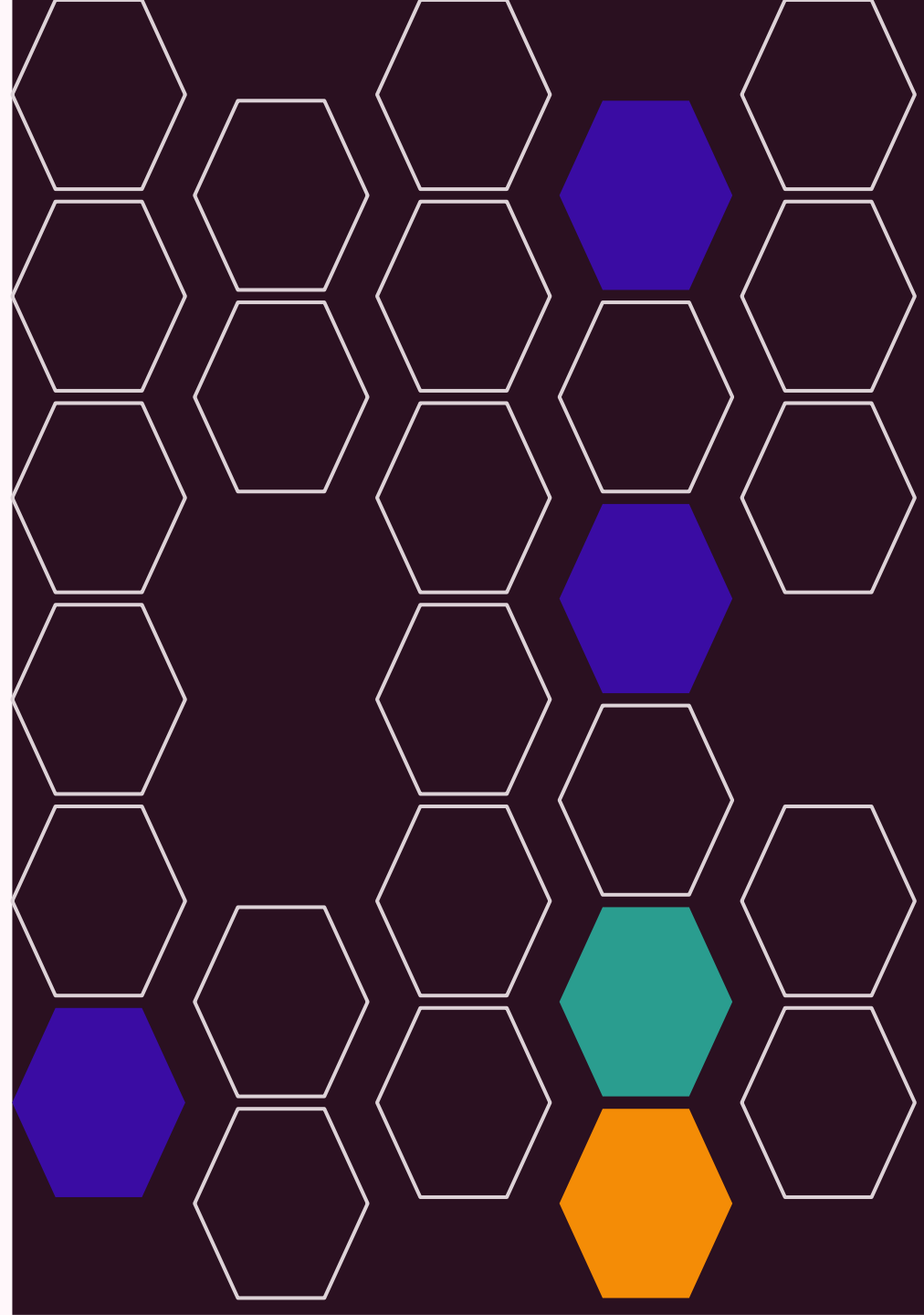
References

- [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

AIR QUALITY MONITORING NETWORK



KEY IDEA

Low-cost sensors mapping pollution across a campus

Project 6 · 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



Image

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

Proposed system

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

System architecture



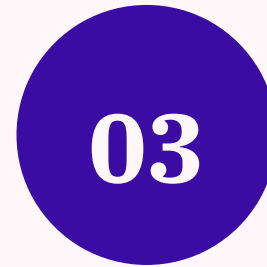
Measure

PMS5003 and CO₂ sensors
in each node



Transmit

LoRa sends readings to a
gateway



Calibrate

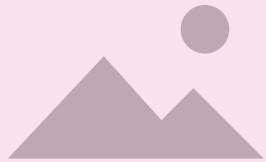
Correction model on the
server



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

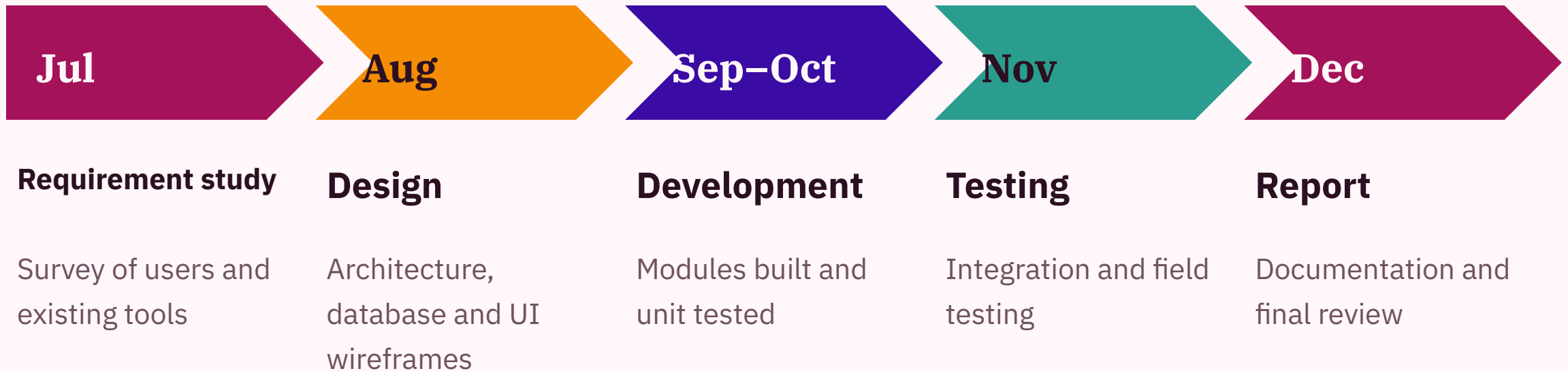
Time-series storage

04

Leaflet

Interactive map

Project timeline



Implementation snapshots



Image 1

Sensor node

Weatherproof enclosure with solar panel



Image 2

Gateway

Rooftop LoRa receiver

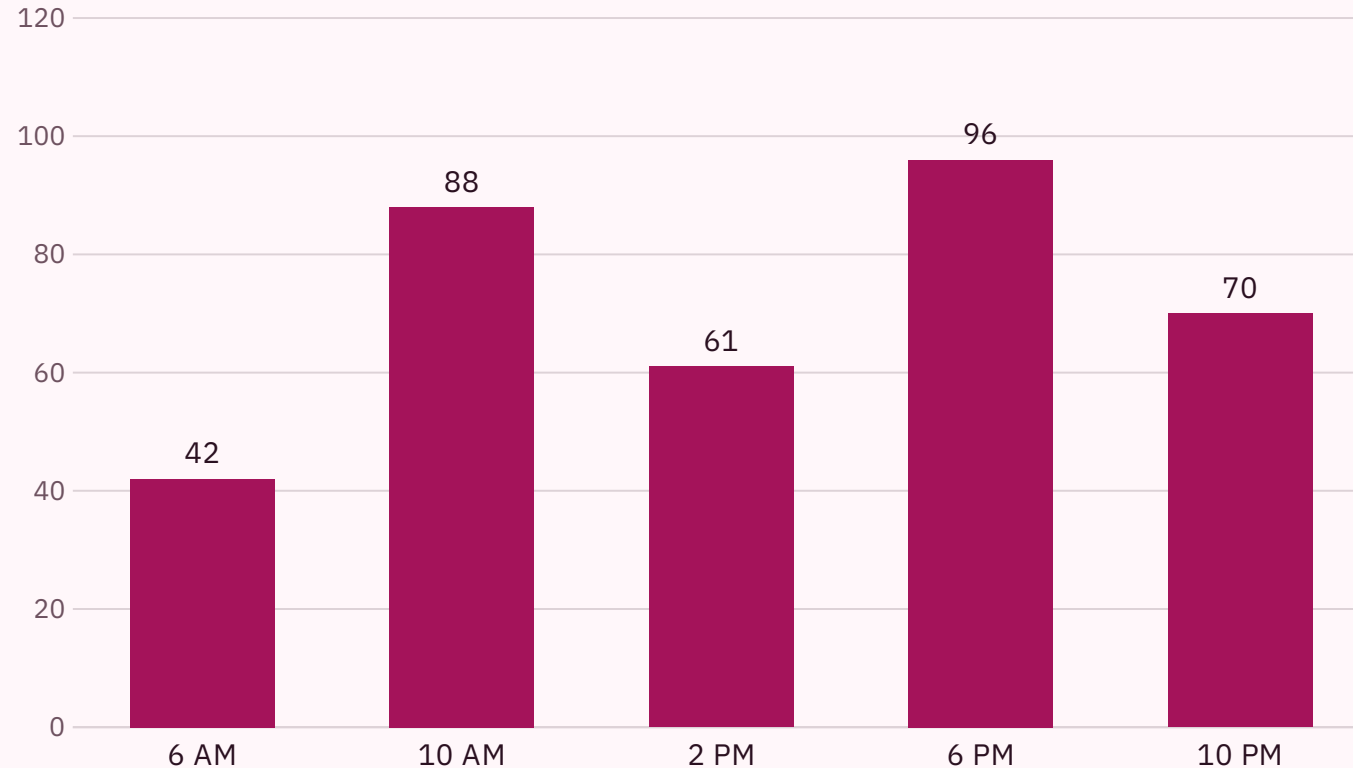


Image 3

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



Image

Future scope

01

City scale

Expand to nearby wards

02

Forecasting

Predict next-day AQI

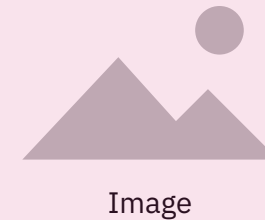
03

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).



Thank you

Questions and discussion.

Presenter Name

hello@yourdomain.in