



Trivanta Edge

Parking, Perfected

sales@trivantaedge.com  
info@trivantaedge.com  
+91 9373015503  
Kalyan, Maharashtra

## Automated Parking System Implementation Guide

Manual Button Control → Mobile App Control → Complete Journey

**Document Overview:** This guide details existing manual parking systems (Puzzle Parking, Tower Parking, Stacker, Hydraulic Systems) and a mobile application for control. It provides a complete step-by-step implementation plan.

### Table of Contents

- System Overview & Architecture
- Components Required (Hardware & Software)
- System Design & Flow Diagrams
- Phase-wise Implementation
- Technical Specifications
- Cost Breakdown
- Safety & Compliance
- Testing & Deployment

## 1 System Overview & Architecture

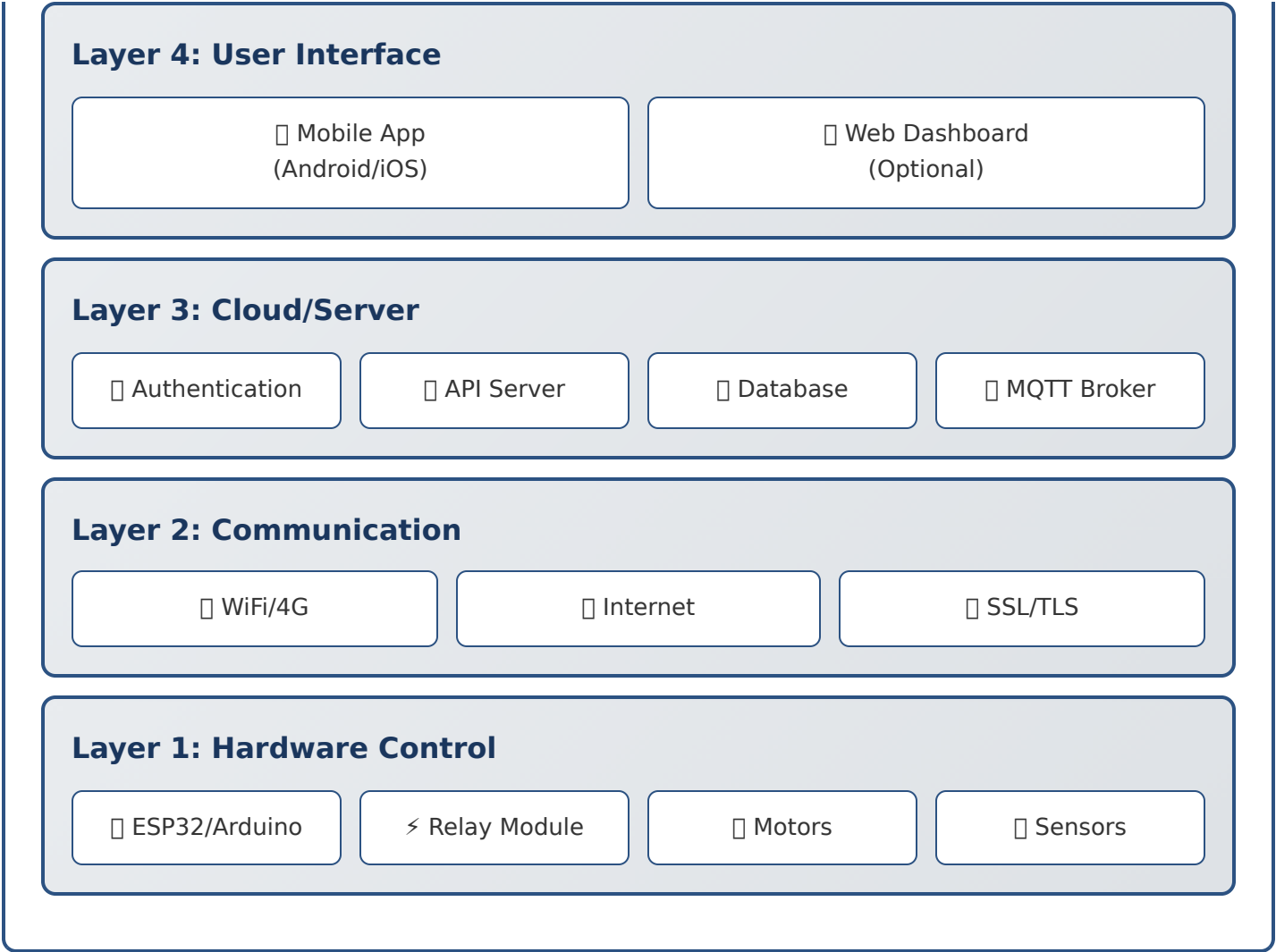
### 1.1 Current System (Manual)

#### Existing Components

- Control Panel:** Manual push buttons (Up, Down, Left, Right, Stop)
- Motors:** Hydraulic motors/Electric motors
- Contactors/Relays:** Motor on/off switching
- Sensors:** Limit switches, proximity sensors, position encoders
- Power Supply:** 3-phase AC power for motors
- Safety Systems:** Emergency stop, overload protection

### 1.2 Target System (Automated)

#### System Architecture (4 Layers)



1.3 Communication Flow





## 2 Components Required

2.1 Hardware Components				
Component	Purpose	Specification	Quantity	Est. Cost (₹)
ESP32 Dev Board	Main controller with WiFi	ESP32-WROOM-32, Dual-core, WiFi+BT	1	500-800
16-Channel Relay Module	Control motors/contactors	5V/12V trigger, 10A per channel	1-2	800-1500
Power Supply	Power ESP32 & relays	12V 5A SMPS	1	300-500
Optocouplers	Electrical isolation	PC817, 4-channel module	2	150-300
Level Shifter	3.3V ↔ 5V conversion	Bi-directional, 4-channel	1	100-200
Enclosure Box	House electronics	IP65 rated, 200x150x75mm	1	400-800
Indicator LEDs	Status indication	5mm, Red/Green/Yellow	10	50-100
Emergency Stop Button	Hardware safety	NC type, 22mm mushroom	1	200-400
Wiring & Connectors	Connections	2.5mm² wire, terminal blocks	-	500-1000
TOTAL HARDWARE COST				₹3,000-5,600

## 2.2 Software Components

Component	Technology	Purpose	Cost
Microcontroller Code	Arduino IDE (C++)	ESP32 programming	Free
Backend Server	Node.js + Express	API & business logic	Free (open-source)
Database	MongoDB Atlas / MySQL	Store user data, logs	Free tier available
MQTT Broker	Mosquitto / AWS IoT	Real-time messaging	Free/₹500-2000/month
Cloud Hosting	AWS / DigitalOcean	Deploy backend	₹500-2000/month
Mobile App	Flutter / React Native	User interface	Free (open-source)
SSL Certificate	Let's Encrypt	Secure communication	Free

## 3📖 Technical Terms - Detailed Explanation

### 📦 ESP32 Microcontroller

**Kya hai?** Ek chhota computer chip (📱📱 mobile ka processor) jo WiFi/Bluetooth built-in hai.

**Kaam:** Internet se commands receive karta hai aur motors ko control karta hai. Example: Jaise TV ka remote signal receive karta hai waise hi ESP32 mobile app se signal receive karta hai.

**Kyun use karein?** Arduino se better kyunki WiFi built-in hai, zyada powerful hai, aur affordable bhi.

### ⚡ Relay Module

**Kya hai?** Electrical switch jo automatically on/off hota hai (jaise ghar ka switch but automatic).

**Kaam:** ESP32 low voltage (3.3V) pe kaam karta hai but motor high voltage (230V AC) pe chalta hai. Relay beech mein bridge ka kaam karta hai - ESP32 signal deta hai → Relay on/off hota hai → Motor chalta/rukta hai.

**Analogy:** Jaise light switch ko manually press karte ho, waise hi ESP32 relay ko electronically press karta hai.

### 📡 MQTT Protocol

**Kya hai?** Internet messaging system (📱📱 WhatsApp but machines 📡 📡).

**Kaam:** Mobile app aur ESP32 ke beech real-time messages bhejne ke liye. Very lightweight aur fast.

**Example:** App message bhejta hai "motor\_up" → MQTT broker ESP32 ko forward karta hai → ESP32 motor up kar deta hai. Reverse bhi same - ESP32 status bhejta hai → App mein display hota hai.

### ☁️ Cloud Server

**Kya hai?** Internet pe ek computer jo 24/7 chalta rehta hai (📱📱 Facebook ka server).

**Kaam:** User authentication (login check), data storage, aur app-ESP32 ke beech middleman.

**Benefit:** Kahin se bhi control kar sakte - ghar se, office se, dusre city se. Bina cloud ke sirf WiFi range mein kaam karega.

📦 **API (Application Programming Interface)**

**Kya hai?** Software ka menu card - jo operations available hain unki list.

**Example APIs:**

- POST /login → User login kare
- POST /moveUp → Platform upar move kare
- GET /status → Current position check kare

**Analogy:** Jaise restaurant menu mein "Paneer Tikka" likha hai waise API mein "moveUp" function likha hai.

📦 **Optocoupler**

**Kya hai?** Electrical safety device - do circuits ko physically separate rakhta hai lekin signal pass karta hai.

**Kyun zaroori?** Agar motor circuit mein koi problem ho (short circuit, high voltage) to ESP32 safe rahega. Light signal se communicate karta hai (electrical connection nahi).

**Analogy:** Jaise glass wall se aap dusre side dekh sakte ho but touch nahi kar sakte, waise hi optocoupler signal pass karta hai but electricity nahi.

4📦 **Phase-wise Implementation Plan**

**Phase 1: Planning & Design (Week 1-2)**

📦 **Goal:** System ki complete understanding aur design finalize karna.

Step 1.1: Current System Documentation

1. **Circuit Diagram banao:** Existing control panel ka wiring diagram draw karo
  - Konsa button kis motor se connected hai
  - Kahan relay/contactor hai
  - Power supply connections
  - Sensor wiring
2. **Motor List:** Sabhi motors ki list
  - Motor 1: Vertical movement (Up/Down)
  - Motor 2: Horizontal movement (Left/Right)
  - Motor 3: Platform rotation (agar applicable)
3. **Sensor Mapping:**
  - Upper limit switch → Pin X
  - Lower limit switch → Pin Y
  - Position encoder → Pin Z

Step 1.2: Requirements Analysis

Requirement	Description	Priority
User Authentication	Sirf authorized users hi access karein	High
Real-time Status	Live position tracking	High
Emergency Stop	Turant motor stop (hardware + software)	Critical
Manual Override	Purane buttons bhi kaam karein	High
Multi-user Support	Multiple operators	Medium
Activity Logs	Kisne kab kya kiya - history	Medium
Offline Mode	Internet na ho to local WiFi pe kaam kare	Low

Step 1.3: App Features Design

Screens required:

1. **Login Screen:** Username/Password entry
2. **Dashboard:** Quick status overview
  - Current platform position
  - System status (Running/Idle/Error)
  - Quick action buttons
3. **Control Screen:** Main control panel
  - Directional buttons (↑↓←→)
  - Emergency stop (red button)
  - Slot selection (for puzzle parking)
  - Manual/Auto mode toggle
4. **History Screen:** Activity logs
5. **Settings Screen:** Configuration options

## Phase 2: Hardware Setup (Week 3-4)

⚠ **Safety First:** Iss phase mein electrical work hai. Agar experience nahi hai to electrician ko involve karo.

### Step 2.1: Component Purchase & Testing

1. Online order karo (Amazon/Robu.in):
  - ESP32 DevKit V1
  - 16-channel relay module
  - Power supply (12V 5A)
  - Connecting wires, breadboard
2. Individual testing:
  - ESP32 ko computer se connect kar Arduino IDE test karo
  - Relay module test karo - manually on/off check karo

### Step 2.2: Prototype Circuit Assembly

Connections (Simplified): ESP32 → Relay Module ————— GPIO 25 → Relay 1 (Motor Up) GPIO 26 → Relay 2 (Motor Down) GPIO 27 → Relay 3 (Motor Left) GPIO 14 → Relay 4 (Motor Right) GND → GND 5V → VCC Relay Module → Motor Contactor ————— Relay COM → Contactor coil (+) Relay NO → Power supply Contactor coil (-) → GND Sensors → ESP32 ————— Upper Limit Switch → GPIO 32 Lower Limit Switch → GPIO 33 Left Limit → GPIO 34 Right Limit → GPIO 35 Sensor GND → ESP32 GND

📌 **Pro Tip:** Pehle breadboard pe sab kuch test karo. Motors ki jagah LED lagao testing ke liye. LED on/off ho rahi hai matlab relay kaam kar raha hai.

### Step 2.3: Basic ESP32 Programming

```
// Basic Relay Control Test Code // Arduino IDE mein paste karo #define RELAY_UP 25
#define RELAY_DOWN 26 void setup() { pinMode(RELAY_UP, OUTPUT); pinMode(RELAY_DOWN,
OUTPUT); digitalWrite(RELAY_UP, LOW); // Relay OFF digitalWrite(RELAY_DOWN, LOW);
Serial.begin(115200); Serial.println("System Ready!"); } void loop() { // Test: Up
relay ON for 2 seconds Serial.println("Moving UP"); digitalWrite(RELAY_UP, HIGH);
delay(2000); digitalWrite(RELAY_UP, LOW); delay(3000); // Wait // Test: Down relay ON
for 2 seconds Serial.println("Moving DOWN"); digitalWrite(RELAY_DOWN, HIGH);
delay(2000); digitalWrite(RELAY_DOWN, LOW); delay(3000); // Wait }
```

### Step 2.4: WiFi Connection Setup

```
// WiFi Connection Test #include <WiFi.h> const char* ssid = "YourWiFiName"; const
char* password = "YourPassword"; void setup() { Serial.begin(115200); WiFi.begin(ssid,
password); Serial.print("Connecting to WiFi"); while (WiFi.status() != WL_CONNECTED) {
delay(500); Serial.print("."); } Serial.println("\nConnected!"); Serial.print("IP
Address: "); Serial.println(WiFi.localIP()); } void loop() { // WiFi connected hai ya
nahi check if(WiFi.status() == WL_CONNECTED) { Serial.println("WiFi OK"); } else {
Serial.println("WiFi Lost! Reconnecting..."); WiFi.begin(ssid, password); }
delay(5000); }
```

## Phase 3: Backend Development (Week 5-6)

### Step 3.1: Server Setup

#### ☐ Option 1: Cloud Server (Recommended)

**Services:** AWS EC2 / DigitalOcean Droplet / Heroku

**Cost:** ₹500-1500/month

**Benefits:** 24/7 uptime, anywhere access, scalable

#### ☐ Option 2: Local Server

**Setup:** Old laptop/Raspberry Pi as server

**Cost:** One-time (₹3000-5000 for RPi)

**Limitation:** Internet connectivity zaroori, power backup needed

### Step 3.2: Database Schema Design

Table Name	Fields	Purpose
users	id, username, password_hash, role, created_at	User authentication
devices	id, device_name, mac_address, status, last_seen	ESP32 device tracking
parking_slots	id, slot_number, status, vehicle_info, timestamp	Slot occupancy
activity_logs	id, user_id, action, timestamp, status	History tracking
system_status	id, position_x, position_y, motor_status, error_code	Real-time system state

### Step 3.3: API Endpoints Design

Endpoint	Method	Purpose	Request	Response
/api/auth/login	POST	User login	{username, password}	{token, user_info}
/api/control/move	POST	Send movement command	{direction: "up"}	{status: "success"}
/api/status/current	GET	Get system status	-	{position, motor_status}
/api/history/logs	GET	Fetch activity logs	?from=date&to=date	[{action, timestamp}]
/api/emergency/stop	POST	Emergency stop all	-	{status: "stopped"}

### Step 3.4: MQTT Setup

#### ☐ MQTT Communication Topics

##### Subscribe Topics (ESP32 listens):

- parking/device123/command/move → Movement commands
- parking/device123/command/stop → Stop command
- parking/device123/config/update → Configuration updates



**Publish Topics (ESP32 sends):**

- parking/device123/status/position → Current position
- parking/device123/status/motor → Motor states
- parking/device123/alert/error → Error notifications
- parking/device123/sensor/limit → Sensor readings

**Phase 4: Mobile App Development (Week 7-9)**

**Step 4.1: Technology Selection**

Technology	Pros	Cons	Learning Curve
Flutter	Single codebase, fast, beautiful UI	Dart language seekhni padegi	Medium (2-3 weeks)
React Native	JavaScript use, large community	Performance issues (sometimes)	Easy (if JS jaante ho)
Native Android	Best performance, full control	iOS alag se banana padega	Hard (3-4 weeks)

📌 **Recommendation:** Flutter use karo - ek code se Android + iOS dono mil jayenge aur UI bahut accha banta hai.

**Step 4.2: App Screen Wireframes**

📱 **Screen 1: Login**



**Trivanta Edge Parking**

📄 Username

📄 Password

LOGIN

📱 **Screen 2: Dashboard**

**System Status**

🟢 Online

**Current Position**

Level 2, Slot 5

**Motor**

**Last Action**



Step 4.3: Key Features Implementation

- **JWT Authentication:** Secure token-based login
- **WebSocket/MQTT Integration:** Real-time status updates
- **Haptic Feedback:** Button press pe phone vibrate
- **Push Notifications:** Errors/alerts ke liye
- **Offline Detection:** Internet na ho to warning
- **Activity History:** Last 50 actions display

Phase 5: Integration & Testing (Week 10-11)

Step 5.1: Complete ESP32 Code


```
// Complete ESP32 Code with MQTT #include <WiFi.h> #include <PubSubClient.h> // WiFi
credentials const char* ssid = "YourWiFi"; const char* password = "YourPassword"; //
MQTT Broker const char* mqtt_server = "your-broker.com"; const int mqtt_port = 1883;
const char* mqtt_user = "device123"; const char* mqtt_pass = "devicePass"; // Pin
definitions #define RELAY_UP 25 #define RELAY_DOWN 26 #define RELAY_LEFT 27 #define
RELAY_RIGHT 14 #define LIMIT_UP 32 #define LIMIT_DOWN 33 WiFiClient espClient;
PubSubClient client(espClient); void setup() { Serial.begin(115200); // Pin setup
pinMode(RELAY_UP, OUTPUT); pinMode(RELAY_DOWN, OUTPUT); pinMode(RELAY_LEFT, OUTPUT);
pinMode(RELAY_RIGHT, OUTPUT); pinMode(LIMIT_UP, INPUT_PULLUP); pinMode(LIMIT_DOWN,
INPUT_PULLUP); // All relays OFF initially stopAllMotors(); // Connect WiFi
connectWiFi(); // Connect MQTT client.setServer(mqtt_server, mqtt_port);
client.setCallback(mqttCallback); } void loop() { if (!client.connected()) {
```

```
reconnectMQTT(); } client.loop(); // Check limit switches checkSafety(); // Send status every 2 seconds static unsigned long lastUpdate = 0; if (millis() - lastUpdate > 2000) { publishStatus(); lastUpdate = millis(); } } void mqttCallback(char* topic, byte* payload, unsigned int length) { String message = ""; for (int i = 0; i < length; i++) { message += (char)payload[i]; } Serial.println("Command: " + message); if (message == "UP") moveUp(); else if (message == "DOWN") moveDown(); else if (message == "LEFT") moveLeft(); else if (message == "RIGHT") moveRight(); else if (message == "STOP") stopAllMotors(); } void moveUp() { if (digitalRead(LIMIT_UP) == HIGH) { // Not at limit stopAllMotors(); digitalWrite(RELAY_UP, HIGH); Serial.println("Moving UP"); } else { Serial.println("Upper limit reached!"); } } void stopAllMotors() { digitalWrite(RELAY_UP, LOW); digitalWrite(RELAY_DOWN, LOW); digitalWrite(RELAY_LEFT, LOW); digitalWrite(RELAY_RIGHT, LOW); } void checkSafety() { // If limit reached, stop motor if (digitalRead(LIMIT_UP) == LOW) { digitalWrite(RELAY_UP, LOW); client.publish("parking/device123/alert/error", "Upper limit"); } } void publishStatus() { String status = "{\"motor\": \"idle\", \"position\": \"level2\"}"; client.publish("parking/device123/status/position", status.c_str()); }
```

Step 5.2: Testing Checklist

Test Case	Procedure	Expected Result	Status
WiFi Connection	ESP32 power on karo	3 seconds mein WiFi connect	☐ / ☐
App Login	Correct credentials enter	Dashboard open ho	☐ / ☐
UP Command	App se UP button press	Motor up chale	☐ / ☐
Limit Switch	Upper limit tak jao	Motor automatically stop	☐ / ☐
Emergency Stop	Motion mein STOP press	Immediately stop	☐ / ☐
Manual Override	Physical button press	Motor chale (app se bhi)	☐ / ☐
Internet Failure	WiFi router off karo	App mein "Offline" dikhe	☐ / ☐
Multiple Users	2 phones se simultaneously	Dono control kar sakein	☐ / ☐
Status Update	Position change karo	App mein real-time update	☐ / ☐
History Logs	10 actions karo	History screen mein dikhe	☐ / ☐

5🔒 Safety & Compliance

 **CRITICAL: Safety Measures (Zaroori)**

## 5.1 Hardware Safety

1. **Emergency Stop Circuit:**
  - Hardware-level emergency stop (NC type)
  - Directly power supply cut kare
  - Software stop pe depend nahi
2. **Limit Switches:**
  - Over-travel prevent karne ke liye
  - Mechanical + software dono checks
3. **Overload Protection:**
  - MCB/MCCB use karo
  - Proper fuse ratings
4. **Electrical Isolation:**
  - Optocouplers between control & power
  - Proper earthing

## 5.2 Software Safety

Safety Feature	Implementation	Purpose
Command Validation	Server pe har command check	Invalid commands block
Timeout Protection	5 sec no response = stop	Hanging prevent
Interlock Logic	UP+DOWN same time nahi	Conflicting commands prevent
Watchdog Timer	ESP32 hang ho to auto-reset	System reliability
Authentication	JWT tokens, session management	Unauthorized access prevent

## 5.3 Compliance Requirements

- ☐ **Indian Standards:**
  - IS 14665:** Mechanical parking systems safety requirements
  - IS 15743:** Automated parking system guidelines
  - IE Rules 1956:** Electrical installation standards
- Zaroori Documents:**
  - Safety certificate from authorized agency
  - Electrical clearance
  - Fire safety NOC (if applicable)
  - Local municipal approvals

## 6 Complete Cost Breakdown

Category	Item	Cost (₹)
Hardware	ESP32 + Relay + Power Supply	3,000 - 5,000
	Optocouplers, connectors, wiring	1,500 - 2,500
	Enclosure box (IP65)	500 - 1,000
	Testing equipment (multimeter, etc)	1,000 - 2,000
	Hardware Subtotal	6,000 - 10,500
Cloud/Server	Cloud hosting (1 year)	6,000 - 18,000
	Domain name (.com)	800 - 1,500
	SSL certificate (Let's Encrypt)	0 (Free)
	Server Subtotal (Yearly)	6,800 - 19,500
Development	If self-developed (learning time)	0 (your time)
	If outsourced (developer hiring)	50,000 - 2,00,000
	Development Subtotal	0 - 2,00,000
Miscellaneous	Testing & debugging	2,000 - 5,000
	Documentation	500 - 1,000
	Safety certifications (optional)	10,000 - 30,000
TOTAL (DIY - Self Development)		₹25,000 - 50,000
TOTAL (Outsourced Development)		₹75,000 - 2,50,000

## 7 Project Timeline

### 12-Week Implementation Plan

#### Week 1-2: Planning & Design

- System documentation
- Requirements analysis
- Component procurement

#### Week 3-4: Hardware Setup

- ESP32 programming basics
- Relay testing
- WiFi connectivity setup

#### Week 5-6: Backend Development

- Server setup
- API development

- Database design
- MQTT broker configuration

**Week 7-9: Mobile App Development**

- UI design & implementation
- API integration
- Real-time features
- Testing on device

**Week 10-11: Integration & Testing**

- Hardware-software integration
- Complete system testing
- Bug fixes
- Safety validation

**Week 12: Deployment & Training**

- Final installation
- User training
- Documentation handover
- Go-live support

## 8▯ Learning Resources & Tutorials

### 8.1 Hardware & ESP32

Topic	Resource	Duration	Link/Source
ESP32 Basics	Random Nerd Tutorials	3-5 days	YouTube/Website
Relay Control	Arduino Project Hub	1-2 days	arduino.cc
WiFi & MQTT	ESP32 MQTT Tutorial	2-3 days	YouTube channels
Sensor Integration	Circuit Digest	2-3 days	circuitdigest.com

### 8.2 Backend Development

Topic	Resource	Duration	Link/Source
Node.js Basics	FreeCodeCamp Tutorial	1 week	YouTube
Express.js API	Traversy Media	3-4 days	YouTube
MongoDB	Net Ninja	3-4 days	YouTube
JWT Authentication	Web Dev Simplified	1-2 days	YouTube
MQTT Broker Setup	Steve's IoT Guide	2-3 days	mqtt-steve.net

8.3 Mobile App Development

Topic	Resource	Duration	Link/Source
Flutter Basics	Flutter Official Course	1-2 weeks	flutter.dev
Dart Language	Dart Official Tutorial	3-5 days	dart.dev
REST API Integration	Flutter HTTP Package	2-3 days	pub.dev
MQTT in Flutter	mqtt_client package	2-3 days	pub.dev
State Management	Provider/Riverpod	3-5 days	YouTube tutorials

9Common Problems & Solutions

9.1 Hardware Issues

Problem	Possible Cause	Solution
ESP32 not connecting to WiFi	Wrong credentials / Signal weak	<ul style="list-style-type: none"><li>• Check SSID/password</li><li>• Move router closer</li><li>• Use WiFi extender</li></ul>
Relay not clicking	Insufficient power / Wrong wiring	<ul style="list-style-type: none"><li>• Check 5V supply</li><li>• Verify relay trigger voltage</li><li>• Test with LED first</li></ul>
Motor running in wrong direction	Relay wiring swapped	<ul style="list-style-type: none"><li>• Swap COM/NO connections</li><li>• Or swap in code logic</li></ul>
ESP32 keeps resetting	Power supply insufficient	<ul style="list-style-type: none"><li>• Use 2A+ power supply</li><li>• Add capacitor (100μF)</li><li>• Check for short circuits</li></ul>
Sensors not reading	Wrong pin / No pull-up resistor	<ul style="list-style-type: none"><li>• Enable INPUT_PULLUP</li><li>• Check sensor voltage (3.3V)</li><li>• Test with multimeter</li></ul>

## 9.2 Software Issues

Problem	Possible Cause	Solution
App not connecting to server	Wrong API endpoint / Firewall	<ul style="list-style-type: none"><li>• Check server IP/URL</li><li>• Verify port is open (443/8883)</li><li>• Check SSL certificate</li></ul>
MQTT messages not received	Topic mismatch / Broker down	<ul style="list-style-type: none"><li>• Check topic spelling (case-sensitive)</li><li>• Test broker with MQTT client</li><li>• Verify credentials</li></ul>
Commands delayed	Network latency / Server overload	<ul style="list-style-type: none"><li>• Optimize code</li><li>• Increase server resources</li><li>• Use QoS 1 for MQTT</li></ul>
App crashes on startup	Dependencies missing / Version conflict	<ul style="list-style-type: none"><li>• Check pubspec.yaml</li><li>• Run flutter clean</li><li>• Update packages</li></ul>
Database connection failed	Wrong credentials / IP whitelist	<ul style="list-style-type: none"><li>• Verify DB connection string</li><li>• Add server IP to whitelist</li><li>• Check firewall rules</li></ul>

## 📁 Advanced Features (Future Enhancements)

### 📁 1. AI-Powered Parking Optimization

**Feature:** Automatically suggest optimal parking positions based on vehicle size, frequency of use

**Implementation:** Machine learning model training on historical data

**Benefit:** Faster retrieval times, better space utilization

### 📁 2. Camera Integration

**Feature:** Number plate recognition, vehicle damage detection

**Hardware:** ESP32-CAM module (₹500-800)

**Benefit:** Automated vehicle tracking, security enhancement

### 📁 3. Voice Control

**Feature:** "Alexa, move car to position 5"

**Integration:** Alexa Skills Kit / Google Actions

**Benefit:** Hands-free operation

### 📁 4. Payment Integration

**Feature:** Automated billing for commercial parking

**APIs:** Razorpay / PayTM integration



**Benefit:** Cashless operations

5. Analytics Dashboard

**Feature:** Usage statistics, peak hours, maintenance predictions

**Tools:** Grafana / PowerBI dashboards

**Benefit:** Data-driven decision making

6. Predictive Maintenance

**Feature:** Alert before component failure (based on vibration sensors, usage cycles)

**Sensors:** Accelerometer, current sensors

**Benefit:** Reduced downtime, cost savings

11 Business Model & Monetization

11.1 Revenue Opportunities

Model	Target Customer	Pricing	Revenue Potential
Hardware Sale	Existing parking system owners	₹25,000 - 50,000 per unit	High margin
Complete System	New installations	₹3-10 lakhs (with mechanical)	Very high
SaaS Subscription	All automated systems	₹500-2000/month per system	Recurring revenue
AMC (Maintenance)	Installed base	₹12,000-36,000/year	Stable income
White Label	Other parking companies	Licensing fee	Scalable

11.2 Market Analysis

Target Market:

- Residential:** High-rise apartments, gated communities
- Commercial:** Malls, office buildings, hotels
- Institutional:** Hospitals, universities, airports
- Industrial:** Manufacturing units, warehouses

Market Size (India):

- Automated parking market: ₹500-800 crores (growing 15% annually)
- Major cities: Mumbai, Delhi, Bangalore, Pune, Chennai
- Growing due to: Space constraints, luxury housing demand

### 11.3 Competitive Advantages

▢ **Your USPs (Unique Selling Points):**

- 1. **Retrofit Solution:** Upgrade existing systems (competitors sell complete new systems)
- 2. **Cost-Effective:** ₹25K vs ₹2-5 lakhs for competitors' automation
- 3. **Local Support:** Based in Maharashtra, quick response
- 4. **Customization:** Can adapt to any existing parking type
- 5. **Cloud + Local:** Works even without internet (hybrid mode)

## 1▢2▢ Legal & Documentation

### 12.1 Required Licenses & Registrations

Document	Issuing Authority	Cost	Validity
GST Registration	GST Department	Free	Annual filing
MSME/Udyam	MSME Ministry	Free	Lifetime
Product Liability Insurance	Insurance company	₹10-30K/year	1 year
CE Certification (if exporting)	Testing lab	₹50K-1L	As per standards
ISO Certification (optional)	Certifying body	₹30K-80K	3 years

### 12.2 Terms of Service & Warranty

▢ **Sample Warranty Terms**

- **Hardware:** 1 year replacement warranty on electronics
- **Software:** Lifetime free updates for bugs, paid for features
- **Cloud Services:** 99.5% uptime guarantee
- **Support:** 24x7 email, phone during business hours
- **Exclusions:** Physical damage, misuse, unauthorized modifications

## 1▢3▢ Marketing & Sales Strategy

### 13.1 Launch Plan

**Phase 1: Pilot Project (Month 1-2)**

- Install in 2-3 friendly customers (discount/free)
- Gather feedback, refine product
- Create case studies & video testimonials

Phase 2: Local Marketing (Month 3-4)

- Website launch with demos
- Google My Business listing
- Social media (LinkedIn, Instagram)
- Local newspaper ads

Phase 3: Partnership (Month 5-6)

- Tie-ups with builders/architects
- Dealer network in major cities
- Collaborations with parking system manufacturers

Phase 4: Scale-up (Month 7+)

- Expand to other states
- Hire sales team
- Attend industry exhibitions

13.2 Marketing Channels

Channel	Cost	Expected ROI	Timeline
Google Ads (Search)	₹10-20K/month	2-3 leads/day	Immediate
LinkedIn Ads (B2B)	₹15-25K/month	5-10 quality leads/month	1-2 weeks
Trade Shows/Exhibitions	₹50K-1L per event	20-50 leads per event	Event-based
Email Marketing	₹2-5K/month	2-5% conversion	Ongoing
Referral Program	₹5-10K per referral	High (trusted source)	After first 10 customers

14 Quick Start Checklist

Week 1 Action Items

1. ☐ Order ESP32 DevKit + Relay Module from Amazon/Robu.in
2. ☐ Install Arduino IDE on laptop
3. ☐ Test blink program on ESP32
4. ☐ Document your current parking system wiring
5. ☐ Take photos of control panel from all angles
6. ☐ List all motors and their functions
7. ☐ Create a simple circuit diagram

## 📅 Week 2 Action Items

1. ☐ Complete "ESP32 WiFi" tutorial
2. ☐ Test relay control with LED
3. ☐ Set up free MongoDB Atlas account
4. ☐ Learn basic Node.js (1 hour daily)
5. ☐ Design app wireframes on paper
6. ☐ Choose: Flutter vs React Native
7. ☐ Install development tools

## ⚠️ Safety Pre-Checks (MUST DO)

1. ☐ Disconnect main power before any wiring work
2. ☐ Use multimeter to verify voltage levels
3. ☐ Install emergency stop button (hardware)
4. ☐ Test each component individually first
5. ☐ Keep fire extinguisher nearby during testing
6. ☐ Have electrician review connections
7. ☐ Insure equipment against damage

## 🆘 Need Help? Contact Trivanta Edge

### 📞 Support Options

<div><input type="checkbox"/> <b>Email Support</b></div> <div><a href="mailto:info@trivantaedge.com">info@trivantaedge.com</a></div> <div>Response within 24 hours</div>	<div><input type="checkbox"/> <b>Phone Support</b></div> <div><a href="tel:+919373015503">+91 9373015503</a></div> <div>Mon-Sat, 10 AM - 6 PM</div>
<div><input type="checkbox"/> <b>Consultation</b></div> <div>Schedule a call</div> <div>Free 30-min consultation</div>	<div><input type="checkbox"/> <b>Installation Service</b></div> <div>On-site setup available</div> <div>Kalyan &amp; surrounding areas</div>

## 🚀 Ready to Start Your Automation Journey!

### Remember:

- Start small - test with one motor first
- Safety is paramount - never skip safety features
- Document everything - it helps in troubleshooting
- Join online communities - ESP32 forums, IoT groups
- Don't hesitate to ask for help - we're here to support!

*"The best time to start was yesterday. The next best time is NOW!"* 📅

**Document Information**

Version: 1.0 | Last Updated: October 2025  
Prepared by: Trivanta Edge Technical Team  
For: Automated Parking System Implementation  
Classification: Internal Use / Client Reference