Kashif -- Project Technology Stack

Version: 1.1.3
Last Updated: February 2026
Platform: iOS, Android, Web (Admin Panel)
Architecture: Microservices

Table of Contents

---

1. Project Overview

Kashif is a road safety application that enables users to report potholes, view reports on an interactive map, earn gamification points, and redeem coupons. The system includes an AI-powered pothole detection service, a full microservice backend, a mobile application for iOS and Android, and a web-based admin panel.

---

2. Mobile Application

Core Framework

Technology	Version	Purpose
React Native	0.81.5	Cross-platform mobile framework
Expo SDK	54	Managed workflow, build tooling
Expo Router	6.0	File-based navigation

Technology	Version	Purpose
TypeScript	5.9	Type-safe development
React	19.1	UI component library

Key Libraries

Library	Purpose
react-native-maps (1.20.1)	Google Maps integration, markers, routes
react-native-google-places-autocomplete	Location search and autocomplete
expo-location (19.0)	GPS tracking, background location
expo-notifications (0.32)	Push notification handling
expo-task-manager (14.0)	Background task execution
expo-av (16.0)	Audio playback for alerts
expo-speech (14.0)	Text-to-speech warnings
expo-haptics (15.0)	Haptic feedback
expo-image-picker (17.0)	Camera and gallery access
expo-image-manipulator (14.0)	Image compression before upload
react-native-reanimated (4.1)	Fluid animations
react-native-gesture-handler (2.28)	Touch gesture handling
react-native-svg (15.15)	SVG rendering
react-native-qrcode-svg (6.3)	QR code generation for coupons
expo-blur (15.0)	Blur effects for modals
expo-linear-gradient (15.0)	Gradient backgrounds
expo-clipboard (8.0)	Copy to clipboard
expo-web-browser (15.0)	In-app browser for terms/privacy
@react-native-async-storage (2.2)	Local key-value storage
@react-native-community/netinfo (11.4)	Network connectivity detection
axios (1.13)	HTTP client for API communication

State Management

Pattern	Implementation
React Context API	AuthContext, LanguageContext, NotificationContext, OfflineContext, DataSyncContext

Pattern	Implementation
AsyncStorage	Persistent local storage for auth tokens, preferences, offline data

Application Screens

Screen	Description
Home (Map)	Interactive Google Maps with pothole markers, route navigation, heatmap overlay
Reports	User report listing with filtering
Coupons	Available coupons with QR codes
Profile	User stats, points, level progress
Settings	Language, notifications, account management
Alert Screen	Real-time pothole proximity warnings
Report Dialog	Photo capture, location selection, category picker
Onboarding Tutorial	First-launch walkthrough

Android Permissions

```
ACCESS_FINE_LOCATION
ACCESS_COARSE_LOCATION
ACCESS_BACKGROUND_LOCATION
ACCESS_NETWORK_STATE
INTERNET
FOREGROUND_SERVICE
FOREGROUND_SERVICE_LOCATION
POST_NOTIFICATIONS
```

iOS Permissions

```
NSLocationWhenInUseUsageDescription
NSLocationAlwaysAndWhenInUseUsageDescription
NSCameraUsageDescription
NSPhotoLibraryUsageDescription
UIBackgroundModes: location, fetch, remote-notification
```

3. Admin Panel

Core Framework

Technology	Version	Purpose

Technology	Version	Purpose
Next.js	15.5	React-based fullstack framework
React	18.3	UI component library
TypeScript	5.x	Type-safe development
Tailwind CSS	3.4	Utility-first CSS framework
PostCSS	8.4	CSS processing

Key Libraries

Library	Purpose
Leaflet (1.9)	Interactive map for report locations
react-leaflet (4.2)	React bindings for Leaflet
Recharts (2.13)	Analytics charts and visualizations
Lucide React (0.462)	Icon library
date-fns (4.1)	Date formatting and manipulation
axios (1.7)	HTTP client for API communication

Admin Panel Features

Module	Description
Dashboard	Summary statistics, charts, overview
Analytics	Detailed usage and report analytics
User Management	User listing, roles, account actions
Report Management	View, edit, approve, reject reports
Report Categories	Category CRUD operations
Coupon Management	Create, edit, expire coupons
Company Management	Partner company profiles
Map View	Geographic visualization of all reports
Notifications	Send push notifications to users
Audit Log	Administrative action tracking
Monitoring	Service health dashboard
Team Management	Admin team roles and permissions
QR Scanner	Coupon validation via scan

Deployment

Component	Detail
Docker Image	node:20-alpine (multi-stage build)
Internal Port	3001
Network	Connected to backend Docker network
Reverse Proxy	Nginx -> admin.kashifroad.com

---

4. Backend Microservices

All backend services are built with the same core stack:

Core Stack

Technology	Version	Purpose
Python	3.11	Programming language
FastAPI	0.109.0	Async REST API framework
Uvicorn	0.27.0	ASGI server
SQLAlchemy	2.0.25	ORM and database toolkit
Alembic	1.13.1	Database migrations
Pydantic	2.5.3	Data validation and serialization
pydantic-settings	2.1.0	Configuration management
Pika	1.3.2	RabbitMQ client
HTTPX	0.26-0.27	Async HTTP client for inter-service calls
psycopg2-binary	2.9.9	PostgreSQL adapter

Service Breakdown

Auth Service

Responsibility	Technology
User registration and login	FastAPI
JWT token generation/validation	python-jose (cryptography), HS256
Password hashing	passlib + bcrypt 4.0.1
Email verification	SMTP integration
Password reset flow	Token-based with email

Responsibility	Technology
File uploads (profile photos)	python-multipart

Reporting Service

Responsibility	Technology
Pothole report CRUD	FastAPI + SQLAlchemy
Geospatial queries	GeoAlchemy2 (0.14.3) + PostGIS
Image upload and storage	python-multipart, file system
Category management	Hierarchical category model

Gamification Service

Responsibility	Technology
Points system	Event-driven via RabbitMQ
Level progression	Rule-based calculation
Leaderboard	Database queries
Achievement tracking	SQLAlchemy models

Coupons Service

Responsibility	Technology
Coupon creation and management	FastAPI + SQLAlchemy
QR code validation	Unique code generation
Redemption tracking	Transaction-safe operations
Company partner management	Relational models

Notification Service

Responsibility	Technology
Push notifications	Firebase Admin SDK (6.8.0)
Email notifications	SMTP (TLS on port 587)
Event consumption	RabbitMQ consumer
Notification history	SQLAlchemy persistence

Pothole Detection Service

Responsibility	Technology
AI-based pothole detection	Roboflow API, YOLOv8
Image processing	OpenCV (headless), Pillow, NumPy
HEIC image support	pillow-heif (0.15)
GPS/EXIF extraction	exifread (3.0), piexif (1.1)
Scheduled processing	APScheduler (3.10)
Scientific computation	SciPy (1.11)
Memory limit	4 GB (Docker resource constraint)

Inter-Service Communication

Pattern	Implementation
Synchronous (REST)	HTTPX calls between services
Asynchronous (Events)	RabbitMQ message queues
Authentication	Internal API key for service-to-service calls

5. API Gateway

Technology	Version	Purpose
Nginx	Alpine (Docker)	Reverse proxy, load balancing

Gateway Configuration

Feature	Detail
Rate Limiting (General)	30 requests/second per IP
Rate Limiting (Auth)	5 requests/minute per IP
Rate Limiting (Password Reset)	3 requests/minute per IP
Rate Limiting (Upload)	10 requests/minute per IP
Max Upload Size	10 MB
CORS Policy	Whitelist-based origin map
Version Hiding	server_tokens off
Upstream Health Checks	Configured per service

Security Headers (Gateway Level)

```
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Referrer-Policy: strict-origin-when-cross-origin
```

6. Database Layer

Technology	Version	Purpose
PostgreSQL	15 (Alpine)	Relational database
PostGIS	Extension	Geospatial data support
GeoAlchemy2	0.14.3	Python ORM for geospatial queries

Database Instances

Each microservice has its own isolated database:

Database	Service	Container
kashif_auth	Auth Service	kashif-auth-db
kashif_reports	Reporting Service	kashif-reporting-db
kashif_gamification	Gamification Service	kashif-gamification-db
kashif_coupons	Coupons Service	kashif-coupons-db
kashif_notifications	Notification Service	kashif-notification-db

Data Persistence

All database data is stored in named Docker volumes to ensure persistence across container restarts and redeployments.

7. Message Broker

Technology	Version	Purpose
RabbitMQ	3 (Management)	Asynchronous message broker

Usage

Event Type	Producer	Consumer
User registration	Auth Service	Gamification, Notification
Report submission	Reporting Service	Gamification, Notification

Event Type	Producer	Consumer
Points awarded	Gamification Service	Notification
Coupon redeemed	Coupons Service	Notification

---

8. AI / Computer Vision

Technology	Purpose
Roboflow API	Cloud-based pothole detection model
YOLOv8	Object detection architecture
OpenCV (headless)	Image preprocessing and annotation
Pillow	Image format conversion
pillow-heif	HEIC/HEIF image support (iPhone photos)
NumPy	Array operations for image processing
SciPy	Scientific computations
exifread / piexif	GPS coordinate extraction from photos

Detection Pipeline

1. Image received (JPEG, PNG, or HEIC)
2. EXIF/GPS data extracted for geolocation
3. Image sent to Roboflow API for YOLOv8 inference
4. Results annotated on the image using OpenCV
5. Detected potholes auto-submitted as reports via Reporting Service

---

9. Push Notifications

Technology	Purpose
Firebase Cloud Messaging (FCM)	Push delivery to Android/iOS
firebase-admin SDK (6.8.0)	Server-side FCM integration
expo-notifications (0.32)	Client-side notification handling

Notification Flow

1. Backend event triggers notification via RabbitMQ
2. Notification Service consumes the event
3. FCM message dispatched to registered device tokens
4. Client receives and displays the notification

---

10. Email Service

Component	Detail
SMTP Server	mail.kashifroad.com (self-hosted)
SMTP Port	587 (STARTTLS)
From Address	noreply@kashifroad.com
Mail Server	Docker Mailserver (container)
Webmail Client	Roundcube (container)
Protocols	SMTP (25, 465, 587), IMAP (993)

Email Use Cases

- Email verification on registration
- Password reset tokens
- Administrative notifications

---

11. Containerization and Deployment

Technology	Purpose
Docker	Container runtime
Docker Compose	Multi-container orchestration

Container Inventory

Container	Image / Build	Purpose
kashif-gateway	nginx:alpine	API Gateway
kashif-auth	python:3.11-slim	Auth Service
kashif-reporting	python:3.11-slim	Reporting Service
kashif-gamification	python:3.11-slim	Gamification Service
kashif-coupons	python:3.11-slim	Coupons Service
kashif-notification	python:3.11-slim	Notification Service
kashif-pothole-detection	python:3.11-slim	AI Detection Service
kashif-auth-db	postgres:15-alpine	Auth Database
kashif-reporting-db	postgres:15-alpine	Reporting Database
kashif-gamification-db	postgres:15-alpine	Gamification Database
kashif-coupons-db	postgres:15-alpine	Coupons Database

Container	Image / Build	Purpose
kashif-notification-db	postgres:15-alpine	Notification Database
kashif-rabbitmq	rabbitmq:3-management	Message Broker
admin-admin-1	node:20-alpine	Admin Panel
mailserver	Docker Mailserver	Email Server
roundcube	Roundcube	Webmail Client

Docker Networking

- All Kashif services communicate over a dedicated bridge network (`kashif-network`)
- Admin panel connects to the backend network via Docker external network
- Database ports are not exposed to the host (internal only)
- Gateway binds to `127.0.0.1:8000` only (not public)

Health Checks

All services have Docker-level health checks configured:

- Interval: 30 seconds
- Timeout: 10 seconds
- Retries: 3
- Start period: 10-30 seconds

Log Management

All containers use the `json-file` logging driver with rotation:

- Max size: 10 MB per log file
- Max files: 5 per container

---

12. Server Infrastructure

Hardware and OS

Component	Detail
Provider	
OS	Ubuntu 22.04.5 LTS
Kernel	5.15.0-164-generic
RAM	8 GB
IP Address	87.106.51.243

Web Server (Host Level)

Technology	Version	Purpose
Nginx	1.18.0	Reverse proxy (host level)

Domains and SSL

Domain	Target	SSL
api.kashifroad.com	127.0.0.1:8000 (Gateway)	Let's Encrypt
admin.kashifroad.com	127.0.0.1:3001 (Admin)	Let's Encrypt

SSL/TLS Configuration

Setting	Value
Protocol	TLSv1.2 and TLSv1.3 only
Cipher Suite	TLS_AES_256_GCM_SHA384
Certificate Authority	Let's Encrypt
Certificate Validity	Auto-renewed (Certbot)
HSTS	max-age=31536000; includeSubDomains

---

13. Security Measures

13.1 SSH Hardening

Measure	Configuration
SSH Port	2299 (non-standard)
Authentication Method	Public key only
Password Authentication	Disabled
Max Authentication Tries	3
Login Grace Time	30 seconds
Root Login	Key only (without-password)
X11 Forwarding	Disabled
TCP Forwarding	Disabled
Agent Forwarding	Disabled
Client Alive Interval	Configured
Cloud-Init Override Protection	Config file made immutable (chattr +i)

13.2 Firewall

Layer	Technology	Configuration
Host Firewall	UFW	Default deny, whitelist rules
Cloud Firewall	IONOS	External port filtering
Docker Firewall	DOCKER-USER	iptables chain blocking ports
Intrusion Prevention	Fail2ban	4 active jails

Blocked Ports (External Access Denied)

Port	Service	Blocked By
4000	HopHop Backend	UFW + DOCKER-USER iptables
5432	PostgreSQL	UFW + DOCKER-USER iptables
5672	RabbitMQ	UFW + DOCKER-USER iptables
8080	HopHop Frontend	UFW + DOCKER-USER iptables

Allowed Ports (External Access)

Port	Service
80	HTTP (redirect)
443	HTTPS
2299	SSH
25	SMTP
465	SMTPS
587	SMTP (STARTTLS)
993	IMAPS

DOCKER-USER Rules Persistence

Docker port-blocking rules are persisted in `/etc/ufw/after.rules` to survive system reboots.

13.3 Nginx Security Headers (Host Level)

Header	Value
Server	nginx (version hidden)
Strict-Transport-Security	max-age=31536000; includeSubDomains

Header	Value
Content-Security-Policy	default-src 'self'; script-src 'self' 'unsafe-inline'; style-src 'self' 'unsafe-inline'; img-src 'self' data: blob:; font-src 'self' data:; connect-src 'self'
X-Content-Type-Options	nosniff
X-Frame-Options	DENY
X-XSS-Protection	1; mode=block
Referrer-Policy	strict-origin-when-cross-origin

13.4 API Security

Measure	Implementation
Authentication	JWT (HS256) with expiration
Token Expiration	30 minutes (access token)
Password Hashing	bcrypt (passlib)
Rate Limiting (Login)	5 requests/minute per IP
Rate Limiting (Password Reset)	3 requests/minute per IP
Rate Limiting (General API)	30 requests/second per IP
Rate Limiting (Upload)	10 requests/minute per IP
CORS	Whitelist-based origin validation
Input Validation	Pydantic schema enforcement
Email Validation	email-validator library
Internal Service Auth	Shared API key
File Upload Limit	10 MB (gateway), 50 MB (host nginx)

13.5 Database Security

Measure	Implementation
Network Isolation	Docker bridge network only
Port Exposure	No database ports exposed to host
Per-Service Databases	Each service has its own database and credentials

Measure	Implementation
Connection	Internal Docker DNS resolution

13.6 Container Security

Measure	Implementation
Minimal Base Images	Alpine and slim variants
Health Checks	All containers monitored
Log Rotation	max-size 10m, max-file 5
Resource Limits	4 GB memory cap on AI service
Restart Policy	unless-stopped
No Privileged Mode	All containers run unprivileged
Gateway Binding	127.0.0.1 only (not 0.0.0.0)

13.7 System Security

Measure	Implementation
Automatic Security Updates	unattended-upgrades enabled
Fail2ban	4 jails (SSH, nginx, repeat offend)
TLS 1.0 / 1.1	Disabled server-wide
Bash History Protection	Append-only (chattr +a)
SSH Config Protection	Immutable cloud-init config

13.8 Mobile Application Security

Measure	Implementation
Secure Token Storage	AsyncStorage with auth context
HTTPS Only	All API calls over TLS
APK Signing	PKCS12 release keystore (RSA 2048)
Minimal Permissions	Only required permissions declared
No Cleartext in Production	Enforced via Android config

---

14. Monitoring and Alerting

Server Monitoring Script

Feature	Detail

Feature	Detail
Script	/root/server-monitor.sh
Schedule	Cron-based execution
Checks	SSH key changes, Docker container health, service uptime, disk usage, failed login attempts
Alerting	Email notifications on anomalies

Backend Health Monitoring

Feature	Detail
Script	monitor.py
Schedule	Every 2 minutes (cron)
Checks	HTTP health endpoints per service
Auto-Recovery	Automatic container restart on failure

Docker Health Checks

All containers report health status to Docker runtime. Unhealthy containers trigger alerts via the monitoring system.

---

15. Build and Release Pipeline

Mobile Build (Android)

Method	Tool	Output
Cloud Build	EAS Build (Expo)	APK / AAB
Local Build	Gradle + Expo Prebuild	Signed APK

Local Build Configuration

Setting	Value
Java	OpenJDK 21 (Temurin)
Gradle	8.14.3
Android SDK	Build Tools 36.0.0
Compile SDK	36
Min SDK	24 (Android 7.0)
Target SDK	36
NDK	27.1.12297006

Setting	Value
Kotlin	2.1.20

APK Signing

Setting	Value
Keystore Format	PKCS12
Key Algorithm	RSA 2048-bit
Signature Algorithm	SHA384withRSA
Validity	10,000 days

Version Control

Tool	Detail
Git	Source control
GitHub	Remote repository hosting
GitHub CLI	Release creation and asset upload

---

16. Internationalization

Supported Languages

Language	Code	Direction
Arabic	ar	RTL
English	en	LTR
Kurdish	ku	RTL

Implementation

Component	Technology
Translation Files	JSON-based locale files (i18n)
RTL Support	Dynamic layout mirroring
Language Switching	Runtime language change via context
Text-to-Speech	expo-speech with locale selection
Admin Panel i18n	Multi-language content management

---

17. Third-Party Services

Service	Provider	Purpose
Google Maps SDK	Google	Map rendering, geocoding
Google Places API	Google	Location search autocomplete
Firebase Cloud Messaging	Google	Push notification delivery
Roboflow API	Roboflow	AI pothole detection model
Let's Encrypt	ISRG	Free SSL/TLS certificates
EAS Build	Expo	Cloud-based mobile builds
IONOS	IONOS	VPS hosting, cloud firewall

Architecture Diagram (Text)

```
                           Internet
                              |
                          [Firewall]
                              |
                     [Ubuntu 22.04 VPS]
                              |
                    [Nginx Reverse Proxy]
                     /                \
       api.kashifroad.com    admin.kashifroad.com
                |                       |
      [Docker: kashif-gateway]   [Docker: admin-admin-1]
                |                  (Next.js 15 / Node 20)
                |
      +---------+---------+---------+---------+---------+
      |         |         |         |         |         |
    [Auth]  [Report]  [Gamif]  [Coupons] [Notif]  [Pothole]
      |         |         |         |         |      (AI)
    [DB]      [DB]      [DB]      [DB]      [DB]
                (PostgreSQL 15 x5)
                              |
                        [RabbitMQ]
                     (Event Bus)
```

This document describes the complete technology stack of the Kashif project as of version 1.1.3 (February 2026).