REQUEST FOR QUOTATION (RFQ 0003)

AI-Supported Software Development Support for a Disaster/Emergency Coordination System

Scope

İzmir Kâtip Çelebi University (İKÇÜ) hereby invites your offer for the Request for Quotation titled **"RFQ 0003 — AI-Supported Software Development Support for a Disaster/Emergency Coordination System"** under the Interreg NEXT MED **MEDAIGENCY** Project, with the details set out below.

Background Information

Disaster and emergency management — through its phases of prevention, mitigation, preparedness, response, and recovery — plays a decisive role in increasing the resilience of communities against predictable and unpredictable disasters. The Mediterranean region is under the increasing frequency and severity of natural disasters such as earthquakes, floods, and forest fires, as well as human-made crises such as conflicts, wars, and pandemics. These events cause mass injuries, loss of life, destruction, and displacement, placing a heavy burden even on the most advanced healthcare systems.

Advances in artificial intelligence (AI) offer opportunities to transform the management of health risks, access to services, equity, and system resilience in times of crisis. In this context, İKÇÜ is developing a general-purpose Disaster/Emergency Coordination System — usable by any institution or region — covering the process in which an assistance request from a citizen during a disaster/emergency is evaluated through AI-supported analysis and presented to an authority, the action plan approved by the authority is conveyed to field personnel, and status updates from the field are reported back to the authority.

Submission Deadline

Please submit your competitive offer before **submission deadline: 25.07.2026**, **23:59** (Türkiye time) via link.

Scope of Work

İKÇÜ is itself carrying out the design and development of **KAIROS (Katip AI Response & Operations System)** - an Autonomous Tactical GIS Assistant for Disaster Response. This consultancy does not outsource the project's core tasks; instead, it requests specialized technical co-development support that will complement and integrate with İKÇÜ's own internal workflows.

The selected Consultant(s) will work iteratively and in close collaboration with İKÇÜ's academic and technical team, ensuring that all outputs are:

- Aligned with İKÇÜ's processes and workflows
- Complementary to technologies already being developed in-house
- Sustainable beyond the pilot phase
- Compliant with ethics-by-design principles

This call is published as a **single lot**. The KAIROS system consists of four interdependent components, and applicants will submit an offer for the entire work (all four components):

Lot Scope and Components

A) Citizen Mobile Application

A.1 Functional Requirements

- Registration/login via phone number + SMS verification (OTP).
- Detailed request form with a free-text description.
- Automatic location sharing via GPS; manual location selection from the map when there is no signal.
- Request status tracking (submitted / under review / approved / team en route / completed) and updates via notifications.
- Using a lightweight model running fully offline on the device (requiring no internet) for a pre-assessment of whether a message is worth forwarding to the center / is urgent; filtering out unnecessary/meaningless message traffic without burdening the central system.
- Queuing the request on the device during network outages and sending it automatically once connectivity returns.
- Multi-language support.

A.2 Technical/Architectural Requirements and Required Team Expertise

- Platform: iOS and Android.
- A lightweight on-device natural language processing/model integration.
- Push notification infrastructure (FCM/APNs) integration.
- A local on-device database for the offline queue and last-location cache.
- REST/JSON API integration with the backend (compliant with the Component B — Backend API contract).
- Required team expertise: at least one mobile application developer (iOS+Android or cross-platform experience); a person experienced in on-device model execution (on-device ML/NLP).

A.3 Security / Performance / Delivery Criteria

- Communication over TLS 1.2+; making it harder to create fake requests through phone-number verification.
- An explicit consent screen and a KVKK (Turkish data-protection law) disclosure notice for sharing location data.
- Operation under low bandwidth.
- Deliverables: source code, release packages, API integration documentation, user guide, test reports (functional and offline scenario tests).

B) Backend and Database

B.1 Functional Requirements

- API for receiving the citizen's request.
- AI (LLM)-based request analysis: extracting the need type, category, and urgency level from free text as structured (JSON) output.
- Finding the most suitable resource for the request via a geospatial database query.
- Route calculation: with a routing engine that does not depend on public third-party services.
- Multi-stop task optimization: route/assignment suggestions enabling a field staff member to serve multiple nearby requests in a single run, in a suitable order.
- Authority approval/rejection API.
- Conveying the approved task to the field personnel application.
- Conveying status updates from field personnel to the authority panel in real time.
- User and role management (citizen, authority/administrator, and field personnel as separate roles with separate permissions).
- Database: designing and setting up a database schema suitable for geospatial queries (PostGIS or equivalent) that holds the location information of facilities (warehouse, hospital, assembly area, etc.), material/stock items, field personnel, citizen requests, tasks, decision/chat history, and audit-trail records.
- Designing the database schema to be extensible so that new data fields can be added in the future.

B.2 Technical/Architectural Requirements and Required Team Expertise

- Backend language/framework: a modern web framework supporting asynchronous (async) request handling; if an alternative is proposed, it will be justified.
- AI provider: a local/open language model running on the institution's own server; with a provider-agnostic architecture that does not require code changes if a transition to a cloud-based provider becomes necessary in the future.
- Routing engine: an open-source routing engine under the institution's control (self-hosted) running on the institution's own regional data.
- Database: PostgreSQL + PostGIS; connection pooling; regular backup and restore procedures.
- An asynchronous job queue for queuing AI calls and route calculations during peak demand.
- A real-time communication layer.
- API contract: a versioned REST API documented with OpenAPI/Swagger.
- Containerization (Docker) and configuration via environment variables; an architecture that can run both on-premises (on-prem) and on a server.

B.3 Security / Performance / Delivery Criteria

- JWT (or equivalent)-based authentication and role-based access control (RBAC).
- Rate limiting and input validation.
- KVKK-compliant data storage and access control.
- Pilot scale: smooth operation on a single, small-scale server for 10–15 concurrent users; the architecture being able to scale horizontally without code changes.

- Deliverables: source code, database schema/migration files, OpenAPI/Swagger documentation, Docker deployment files, an operations/maintenance guide (including backup, monitoring, and incident-response procedures).

C) Command/Administrator Web Panel

C.1 Functional Requirements

- Real-time listing of incoming requests.
- Natural language command interaction.
- Displaying requests, facilities, and suggested routes on the map.
- Reviewing the plan suggested by the AI (resource, route, task) and approving, rejecting, or manually editing it.
- Tracking the task status of field personnel in real time.
- Separate sessions for multiple authorities/operators.
- A summary dashboard (number of active requests, regional density, average response time, etc.).

C.2 Technical/Architectural Requirements and Required Team Expertise

- A modern frontend framework and a map library.
- A WebSocket client for real-time data streaming.
- A production-optimized build.
- A responsive design supporting desktop and tablet use.

C.3 Security / Performance / Delivery Criteria

- Role-based access control, session management, and timeout policies.
- Keeping an audit log for all approval/rejection/assignment actions.
- The ability to display a large number of concurrent requests/markers on the map without performance loss (with clustering if necessary).
- Deliverables: source code, production build/deployment scripts, user guide, training material, test reports.

D) Field Personnel Mobile Application

D.1 Functional Requirements

- Instant notification via push when a new task is assigned.
- Task detail: target location, task type, priority level, and the route shown on the map.
- Navigation: showing the route on the in-app map or redirecting to an external navigation app.
- Status update: standard status notifications such as “departed / arrived / completed / unable to reach.”
- Ability to accept or reject the assigned task.
- If multiple nearby tasks are assigned, displaying them as an ordered list.
- Offline operation: storing task data on the device during connection loss and synchronizing status updates once connectivity returns.

D.2 Technical/Architectural Requirements and Required Team Expertise

- Platform: iOS and Android.
- Offline-first architecture: a local on-device database and synchronization logic.
- Push notification infrastructure (FCM/APNs).
- Integration with the backend API and the real-time communication layer.

D.3 Security / Performance / Delivery Criteria

- Personnel login (independent user management) and device-based session management.
- Explicit permission management for sharing location data and KVKK compliance.
- Background location tracking not significantly affecting the device's battery life.
- Deliverables: source code, user guide, test reports (including offline scenario and notification-delivery tests).

Price Table to Be Completed

The applicant will complete the table below for the entire work (all four components). Prices will be given in EUR, VAT included.

Phase	Description	Time Frame	Price (EUR, VAT incl.)
Phase I	Requirements, architecture & API contract	Contract + 0–2 months	20%
Phase II	Core development (backend + components)	Months 2–4	40%
Phase III	Integration & end-to-end testing	Month 5	20%
Phase IV	Pilot-ready delivery, documentation & training	Month 6	20%

Note: The phase time frames are an indicative schedule; exact dates will be confirmed at the contract stage.

Technical Requirements (Candidate Profile)

- A bachelor's degree or higher in Computer Science, Software Engineering, Data Science, Artificial Intelligence, Information Systems, Biomedical Engineering, Medicine, or other related Health Sciences disciplines.
- Proven experience in software development, including full-stack development.
- Strong coding skills in relevant programming languages (e.g., Python, JavaScript, Java, C++, or equivalent).
- Experience designing and consuming API and microservice architectures.
- Proven experience in artificial intelligence and machine learning (including model development, training, and deployment).
- Experience working with Large Language Models (LLMs) and/or generative AI (strongly preferred).
- Familiarity with data engineering, analytics, and decision-support systems.

- Experience with cloud platforms (e.g., AWS, Azure, GCP) and scalable deployment environments.
- Capacity to maintain, update, and support software solutions beyond initial development.
- Ability to work in multidisciplinary, transnational teams.

Participation in Offers

All proposal documents shall be submitted in English and Turkish.

All offers shall be submitted via application link.

This call is published as a single lot; applicants will submit an offer for the entire work (all four components).

In this RFQ process, conducted under the framework of the MEDAIGENCY Project and in compliance with applicable international regulations, European Union funding rules, and relevant national legislation, participating firms may be required to submit supporting documentation to verify their legal, financial, and operational eligibility.

Initial applications will be collected via the designated online application link. At this stage, only the information requested in the application form must be completed and submitted.

Following the completion of the application stage and the preliminary evaluation, shortlisted applicants will be formally notified and provided with detailed instructions regarding the submission of the required official documents. Such documents may include, but are not limited to, company registration records, tax/VAT certificates (where applicable), financial identification documents, declarations of compliance, and any other documents required under national public-procurement legislation and EU-funded project procedures.

All requested documents will be collected and processed in accordance with the relevant data-protection regulations and procurement rules.

Evaluation Criteria

The following elements will be the primary considerations when evaluating the submitted offers and selecting the qualified supplier:

EVALUATION	Weight
Technical Evaluation*	60%
Financial Evaluation	40%

*TECHNICAL EVALUATION	Weight
Qualifications, expertise, and experience of the proposed personnel (education, certifications, participation in workshops/events)	50%
Working methodology and integration/collaboration plan with İKÇÜ's internal team	50%