y 0.4pt

LLM-Driven First Aid Assistant for Disaster Zones

GROUP 7

Team Members:

- Amal Ajay A - KTE22EC013
- Joshwin Binu - KTE22EC038
- Judin Joe Mathew - KTE22EC039
- Justin Varghese - KTE22EC040

Project Guide:

Asst. Prof. Shinoj K Sukumaran

Introduction: The Challenge

- **Kerala's Vulnerability:** Prone to floods, landslides, and other natural disasters.
- **Critical Infrastructure Failure:** Disasters often knock out power, internet, and mobile networks.
- **The Consequence:** Isolated communities are cut off from vital, life-saving information precisely when they need it most.

y 0.4pt

Our Solution: The Offline Crisis Assistant

Why *Offline*?

Cloud-Based Assistants (e.g., Siri, Alexa)

- ✗ **Require Internet**
- ✗ **Fail when networks are down**
- ✗ **Not tailored for disaster medicine**

Our Crisis Assistant

- ✓ **Fully Offline Operation**
- ✓ **Rugged and Portable**
- ✓ **Specialized for First Aid & Survival**

A rugged, portable device powered by a specialized offline LLM to provide immediate, expert-level emergency guidance without any connectivity.

y 0.4pt

Objectives

- To design and develop a rugged, portable emergency assistant device.
- To optimize and run a fine-tuned offline LLM on embedded hardware (Raspberry Pi).
- To support intuitive voice and text input for context-aware guidance.
- To provide reliable information on first aid, survival, and emergency protocols.
- To function entirely without internet dependency.

y 0.4pt

Abstract

In disaster-stricken areas where internet connectivity is often unavailable, access to timely and reliable emergency information is critical. This project presents the design and development of a rugged, portable device powered by a fine-tuned Large Language Model (LLM) that operates entirely offline. The system provides context-aware assistance on first aid, CPR, wound care, and survival techniques via text or speech input. Trained on verified content from the WHO and Red Cross, the LLM ensures reliable responses. Built on a Raspberry Pi with a microphone, speaker, and battery pack, this project demonstrates how embedded AI can strengthen disaster preparedness and response without relying on cloud services.

Literature Review

	Paper & Key Points	Relevance to Our Project
	Zheng et al. (2024) *A Review on Edge Large Language Models* Techniques for running LLMs on low-power devices.	Provides the foundational methodology for model optimization (quantization, pruning) crucial for our Raspberry Pi deployment.
	Basit et al. (2024) *MedAide: On-Premise Medical Assistance* Offline medical chatbot using LoRA fine-tuning.	Directly validates our core concept of an offline, reliable medical assistant, informing our fine-tuning strategy.
	Goecks & Waytowich (2023) *DisasterResponseGPT* Generates disaster response plans from scenarios.	Informs the design of our system's logic for structured emergency guidance and planning in chaotic environments.

y 0.4pt

Phase 1 Work Plan (July 8 – September 30)

Stage	Dates	Key Tasks
1. Ideation	Jul 8 – 21	Brainstorming, topic finalization, preliminary literature survey on edge AI and disaster tech.
2. Scoping	Jul 22 – Aug 4	Finalize objectives, draft hypothesis, initial block-level design.
3. Planning	Aug 5 – 18	Complete system architecture, assign team roles (HW/SW), list requirements.
4. Feasibility	Aug 19 – Sep 1	Preliminary tests: model size vs. hardware capacity, component testing.
5. Refinement	Sep 2 – 22	Iterate on design based on test results, compile initial analysis.
6. Reporting	Sep 23 – 30	Prepare and review Phase 1 final report and presentation.

Output Obtained

```
jane  ~  01:51  ) cd
jane  ~  01:51  ) cd Documents/SRE-RAG
jane  ../SRE-RAG  main ?  01:51  ) cd knowledgebase
jane  ../SRE-RAG/knowledgebase  main ?  v3.13.3  01:52  ) python3 rag\_assisatant\_.py
⚠ Rich not installed. Using basic formatting.
🏥 OFFLINE CRISIS ASSISTANT - DEMO VERSION
Simulating LLM-powered emergency guidance system
No internet connection required!

Choose mode:
1. Demo Mode (for presentation)
2. Interactive Mode (for testing)
3. Show System Status

Select mode (1, 2, or 3): 1
```

```
🏥 OFFLINE CRISIS ASSISTANT - SYSTEM STATUS
============================================================
✅ System Mode: OFFLINE OPERATIONAL
✅ Knowledge Base: LOADED
✅ Emergency Procedures: 6 PROCEDURES READY
✅ Response Time: <3 SECONDS
✅ Network Dependency: NONE (FULLY OFFLINE)
============================================================

📋 AVAILABLE EMERGENCY PROCEDURES:
------------------------------------------------------------
• Severe Bleeding Control - HIGH PRIORITY
• Cardiopulmonary Resuscitation (CPR) - CRITICAL PRIORITY
• Thermal Burns Treatment - HIGH PRIORITY
• Airway Obstruction (Choking) - CRITICAL PRIORITY
• Bone Fractures and Sprains - MODERATE PRIORITY
• Medical Shock Treatment - CRITICAL PRIORITY
============================================================

🎯 DEMO: Processing Sample Emergency Queries
```

```
📋 STEP-BY-STEP PROCEDURE:
  1. SCENE SAFETY - Ensure area is safe for you and victim
  2. UNIVERSAL PRECAUTIONS - Wear gloves or use barrier if available
  3. EXPOSE THE WOUND - Remove/cut clothing to see injury clearly
  4. DIRECT PRESSURE - Apply firm pressure with clean cloth/gauze
  5. MAINTAIN PRESSURE - Do not lift to check if bleeding stopped
  6. ELEVATE IF POSSIBLE - Raise injured area above heart level
  7. PRESSURE POINTS - If bleeding continues, apply pressure to arterial points
  8. SECURE BANDAGE - Apply pressure bandage, check circulation below wound
  9. TREAT FOR SHOCK - Keep victim warm, lying down, legs elevated
  10. MONITOR VITALS - Check breathing, pulse, consciousness level

⚠️ CRITICAL WARNINGS:
  ⚠️ NEVER remove embedded objects (knives, glass, etc.)
  ⚠️ Do NOT use tourniquets unless trained (life/limb situations only)
  ⚠️ If blood soaks through bandage, add more layers - don't remove
  ⚠️ Watch for signs of shock: pale, cold, weak pulse, confusion

🏥 SEEK IMMEDIATE MEDICAL HELP IF:
  • Bleeding won't stop after 10 minutes of direct pressure
```

```
================================================================================
🚨 EMERGENCY RESPONSE: SEVERE BLEEDING CONTROL
================================================================================
⚠️ URGENCY LEVEL: HIGH
📍 QUERY: Someone is bleeding heavily from their arm
⏰ TIME: 2025-09-30 01:56:26
🎯 CONFIDENCE: 6.7%
📚 SOURCE: WHO Emergency Care Guidelines 2023

🔧 REQUIRED SUPPLIES:
  • Clean cloth or sterile gauze pads
  • Pressure bandages or elastic wrap
  • Medical gloves (if available)
  • Scissors to cut clothing
  • Blanket for shock treatment

📋 STEP-BY-STEP PROCEDURE:
  1. SCENE SAFETY - Ensure area is safe for you and victim
  2. UNIVERSAL PRECAUTIONS - Wear gloves or use barrier if available
  3. EXPOSE THE WOUND - Remove/cut clothing to see injury clearly
  4. DIRECT PRESSURE - Apply firm pressure with clean cloth/gauze
  5. MAINTAIN PRESSURE - Do not lift to check if bleeding stopped
  6. ELEVATE IF POSSIBLE - Raise injured area above heart level
  7. PRESSURE POINTS - If bleeding continues, apply pressure to arterial points
  8. SECURE BANDAGE - Apply pressure bandage, check circulation below wound
  9. TREAT FOR SHOCK - Keep victim warm, lying down, legs elevated
  10. MONITOR VITALS - Check breathing, pulse, consciousness level
```

y 0.4pt

```
sh: ro: no such file or directory: ./bin/llama
❯ ro -p "/home/pi/bin/" -f server -m Q2_K.gguf -c 512 -t 3 ./bin/llama-cli -m ../models/Phi-3-mini-4k-instruct-q4.gguf -p "What is the capital of France?" --n-predict 32
build: 4447 (5f5a1ad2) with gcc 12beta (rev2.2-17ubuntu12) 14.2.0 for x86_64-linux-gnu
main: llama backend init
main: load the model and apply lora adapter, if any
llama_model_loader: loaded meta data with 24 key-value pairs and 195 tensors from ../models/Phi-3-mini-4k-instruct-q4.gguf (version GGUF V3 (latest))
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv   0:                       general.architecture str              = phi3
llama_model_loader: - kv   1:                               general.name str              = Phi3
llama_model_loader: - kv   2:                        phi3.context_length u32              = 4096
llama_model_loader: - kv   3:                      phi3.embedding_length u32              = 3072
llama_model_loader: - kv   4:                   phi3.feed_forward_length u32              = 8192
llama_model_loader: - kv   5:                           phi3.block_count u32              = 32
llama_model_loader: - kv   6:                      phi3.attention.head_count u32              = 32
llama_model_loader: - kv   7:                   phi3.attention.head_count_kv u32              = 32
llama_model_loader: - kv   8:     phi3.attention.layer_norm_rms_epsilon f32              = 0.000010
llama_model_loader: - kv   9:                      phi3.rope.dimension_count u32              = 96
llama_model_loader: - kv  10:                          general.file_type u32              = 15
llama_model_loader: - kv  11:                       tokenizer.ggml.model str              = llama
llama_model_loader: - kv  12:                         tokenizer.ggml.pre str              = default
llama_model_loader: - kv  13:                      tokenizer.ggml.tokens arr[str,32064]   = ["<unk>", "<s>", "</s>", "<0x00>", "<...
llama_model_loader: - kv  14:                      tokenizer.ggml.scores arr[f32,32064]   = [0.000000, 0.000000, 0.000000, 0.000...
llama_model_loader: - kv  15:                  tokenizer.ggml.token_type arr[i32,32064]   = [3, 3, 3, 6, 6, 6, 6, 6, 6, 6, 6, 6, ...
llama_model_loader: - kv  16:                      tokenizer.ggml.bos_token_id u32              = 1
llama_model_loader: - kv  17:                      tokenizer.ggml.eos_token_id u32              = 32000
llama_model_loader: - kv  18:                tokenizer.ggml.unknown_token_id u32              = 0
llama_model_loader: - kv  19:                tokenizer.ggml.padding_token_id u32              = 32000
llama_model_loader: - kv  20:               tokenizer.ggml.add_bos_token bool             = true
llama_model_loader: - kv  21:               tokenizer.ggml.add_eos_token bool             = false
llama_model_loader: - kv  22:                    tokenizer.chat_template str              = {{ bos_token }}{% for message in mess...
llama_model_loader: - kv  23:               general.quantization_version u32              = 2
llama_model_loader: - type  f32:   65 tensors
llama_model_loader: - type q4_K:  81 tensors
llama_model_loader: - type q6_K:   32 tensors
llama_model_loader: - type q8_0:   17 tensors
print_info: file format = GGUF V3 (latest)
print_info: file type   = Q4_K - Medium
print_info: file size   = 2.23 GiB (5.01 BPW)
load: special_eos_id = 32007 (<|end|>) - was not control-type; this is probably a bug in the model. its type will be overridden
load: special token: 32000 '<|endoftext|>' was not control-type; this is probably a bug in the model. its type will be overridden
load: printing all EOG tokens:
load:  - 32000 ('<|endoftext|>')
load:  - 32007 ('<|end|>')
load: special tokens cache size = 67
load: token to piece cache size = 0.1499 MB
print_info: arch             = phi3
print_info: vocab_only       = 0
print_info: n_ctx_train      = 4096
print_info: n_embd           = 3072
```

```
llama_context: freq_base     = 10000.0
llama_context: freq_scale    = 1
llama_context:        CPU  output buffer size =     0.12 MiB
llama_kv_cache: size =  384.00 MiB ( 512 cells,  32 layers,  1/1 seqs), K (f16):  192.00 MiB, V (f16):  192.00 MiB
llama_context: flash_attention was auto, set to enabled
llama_context:        CPU compute buffer size =   72.01 MiB
llama_context: graph nodes  = 1123
llama_context: graph splits = 1
common_init_from_params: added <|endoftext|> logit bias = -inf
common_init_from_params: added <|end|> logit bias = -inf
common_init_from_params: setting dry_penalty_last_n to ctx_size = 4096
common_init_from_params: warning: no BOS token with an empty rope - please wait ... (--no-warmup to disable)
main: llama threadpool init, n_threads = 4
main: chat template example:
<|system|>
You are a helpful assistant<|end|>
<|user|>
Hello<|end|>
<|assistant|>
Hi there<|end|>
<|user|>
How are you?<|end|>
<|assistant|>

system_info: n_threads = 4 (n_threads_batch = 4) / 10 | CPU : SSE3 = 1 | SSSE3 = 1 | AVX = 1 | AVX_VNNI = 1 | AVX2 = 1 | F16C = 1 | FMA = 1 | BMI2 = 1 | LLAMAFILE = 1 | OPENMP = 1 | REPACK = 1
 |

main: interactive mode on.
sampler seed: 4264471993
sampler params:
	repeat_last_n = 64, repeat_penalty = 1.000, frequency_penalty = 0.000, presence_penalty = 0.000
	dry_multiplier = 0.000, dry_base = 1.750, dry_allowed_length = 2, dry_penalty_last_n = 4096
	top_k = 40, top_p = 0.950, min_p = 0.050, xtc_probability = 0.000, xtc_threshold = 0.100, typical_p = 1.000, top_n_sigma = -1.000, temp = 0.800
	mirostat = 0, mirostat_lr = 0.100, mirostat_ent = 5.000
sampler chain: logits -> logit-bias -> penalties -> dry -> top-n-sigma -> top-k -> typical -> top-p -> min-p -> xtc -> temp-ext -> dist
generate: n_ctx = 4096, n_batch = 2048, n_predict = 32, n_keep = 1

== Running in interactive mode. ==
 - Press Ctrl+C to interrupt at any time.
 - Press Return to return control to the AI.
 - To return control without starting a new line, end your input with '/'.
 - If you want to submit another line, end your input with '\'.
 - Not using system message. To change it, set a different value via -sys PROMPT

<|user|> What is the capital of France?<|end|><|assistant|> The capital of France is Paris. It is not only the country's largest city but also a global center for art, fashion, gastronomy, and
```

y 0.4pt

References

- World Health Organization, *First Aid Guidelines*, 2023. https://www.who.int
- International Federation of Red Cross, *First Aid Manual*, 2022. https://www.ifrc.org
- Vicuna Team, “Vicuna: An Open LLM Dialogue Assistant,” 2023. https://lmsys.org
- ACM TIST / ArXiv, *LLMs for Emergency Response*, 2023. https://arxiv.org/abs/2306.08956
- Basit et al., *MedAide*, 2024. https://arxiv.org/abs/2403.00830
- Goecks & Waytowich, *DisasterResponseGPT*, 2023. https://arxiv.org/abs/2306.17271

y 0.4pt

Thank You

Questions?

group-7@example.com github.com/group-7