جامعة بنها الأهلية
BNU
BENHA NATIONAL UNIVERSITY
www.bnu.edu.eg

كلية علوم الحاسب
FCS-BNU
Faculty of Computer Science

GRADUATION PROJECT PROPOSAL

PROJECT NAME	AI Customer Support Service for Egyptian Arabic (Kalamna)	
PROJECT Supervisor	Dr. Ahmed Yousry & Eng. Aya Ayman	
Team Members	1. Ahmed Ehab Ahmed Elattar	2. Zeyad Wael Helmay
	3. Ahmed Yasser Ahmed Al Geddawy	4. Eyad Hesham Ibrahaim
	5. Bassant Hossam Salah Eldeen	6. Shahda Mohamed Abd Alkareem
	7. Toni Ihab Youssef	8.
	9.	10.
	11.	12.
Team Leader	Bassant Hossam Salah Eldeen	
PHONE	01020877700	
EMAIL	Bassanthosamm1@outlook.com	

SUMMARY	Kalamna is a smart AI-powered customer support platform built for Egyptian companies. It automates customer interactions **across WhatsApp, Messenger, websites, and mobile apps** while understanding the Egyptian Arabic dialect in **both text and voice**. Using advanced AI for emotion detection, speech understanding, and business-specific customization, Kalamna enables any business to deliver fast, personalized, and always-available customer service in a natural Egyptian tone.
PROBLEM STATEMENT	Many Egyptian businesses struggle to provide consistent and high-quality customer support **due to limited staff, time, and resources. Small and medium enterprises such as restaurants, clothing stores**, and service providers often face repetitive customer questions, missed messages, and slow response times, especially outside working hours. Most existing chatbot solutions are either expensive, lack support for the Egyptian Arabic dialect, or feel robotic and impersonal. As a result, customers experience frustration and businesses lose engagement, loyalty, and potential sales opportunities.
EXPECTED CHALLENGES	While Kalamna uses existing AI models such as **NileChat**, **Gemini**, **Qwen**, several challenges are expected : 1- Managing **pricing and API availability** is critical to keep the platform affordable for Egyptian businesses. 2- Achieving accurate **emotion detection** in the Egyptian dialect and fine-tuning model prompts for local communication styles will also be challenging. 3- Designing a **customizable system** that adapts easily to different business types, while maintaining **fast, reliable, and secure performance**, will require careful backend design and orchestration.
PROPOSED SOLUTION	Kalamna offers a flexible AI-powered customer support system that connects businesses with their customers through **WhatsApp, Messenger, websites, and mobile apps**. It uses advanced models such as **NileChat**, **Gemini**, **Qwen** to understand Egyptian Arabic in both text and voice, detect emotions, and generate natural, human-like responses. Through a secure **FastAPI backend** and **React dashboard**, each business can customize the chatbot's tone, answers, and operating hours, as well as manage FAQs and monitor analytics. The system's **modular design** ensures affordability, scalability, and easy integration, allowing any company to have always-available customer service.

COMPETITORS/MAIN WORK DONE IN THE LITERATURE	Several companies in Egypt and the MENA region already offer chatbot or communication solutions, but most focus on automation rather than real conversational intelligence. **1. WideBot (Egypt)** – A no-code chatbot builder that supports Arabic dialects. It helps businesses automate FAQs but lacks advanced features like **emotion detection**, **voice support**, and deep customization. **2. Cequens (MENA)** – A communication platform that provides **WhatsApp, SMS, and voice APIs** for enterprises. While strong in infrastructure, it has limited **AI understanding** and no dialect or emotion adaptation. **3. DXwand (Egypt)** – A conversational AI company supporting multiple Arabic dialects. It offers intelligent chat and voice systems but is **expensive** and mainly targets **large enterprises**, not small or medium businesses. But **Kalamna** focuses on **Egyptian Arabic**, **emotion-aware AI**, and **affordable customization** for **SMEs**. It combines **multimodal communication** (text, voice, and emotion) with flexible integration , something existing competitors currently lack.
DESIRED OUTCOME	The goal of Kalamna is to build a smart, emotion-aware customer support system that enables Egyptian businesses to communicate naturally and efficiently with their customers. We aim to deliver an **affordable, scalable, and customizable platform** that supports **Egyptian Arabic**, starting with **text-based interaction** through a **web or app widget**, then expanding to **voice support** and **Meta integrations** such as **WhatsApp** and **Messenger**. The system will adapt to different business needs and ensure smooth integration with existing platforms. By combining **AI automation** with **human-like interaction**, Kalamna will enhance customer satisfaction, reduce support workload, and strengthen the digital presence of local businesses across various industries.
WHAT DO YOU NEED TO LEARN IF ANY	To successfully develop Kalamna, our team needs to gain deeper knowledge across multiple areas of full-stack AI system development. On the **frontend**, we will focus on learning **React.js** to build the **interactive chatbot widget** and the **business dashboard** for managing knowledge bases, analytics, and customization settings. On the **backend**, we aim to master **FastAPI** as the middleware that connects the AI layer, database, and frontend. We will also learn to implement **Redis** for caching, **PostgreSQL** for structured data storage, and **pgvector** for vector-based semantic search to support **RAG (Retrieval-Augmented Generation)**. Also We plan to gain hands-on experience in **AI orchestration** using **LangChain** , integrating models like **NileChat**, **Gemini**, **Qwen** , as well as connecting **Meta services** such as **WhatsApp** and **Messenger** through their APIs. Finally, we will learn how to **deploy and manage the system** on cloud servers using **VPS**, **Coolify**, **Docker**, and **GitHub CI/CD**, ensuring secure, scalable, and automated deployment workflows.
EXPECTED TOOLS/TECHNOLOGIES NEEDED	Kalamna will be built using a full-stack architecture combining modern web tools and advanced AI orchestration. The **frontend** will use **React.js** and **Tailwind CSS** to build the chatbot widget and business dashboard with a responsive interface. The **backend**, powered by **FastAPI (Python)**, will manage authentication, chat sessions, and integrations, supported by **Redis** for caching and real-time performance. The **AI layer** will orchestrate multiple models like **NileChat model** , **Gemini or Qwen APIs** , using **LangChain** for intent detection, emotion recognition, and RAG-based retrieval with **pgvector** in **PostgreSQL**. **Whisper** and **TTS APIs** will later enable voice support. Integrations with **Meta services** (WhatsApp, Messenger, Instagram) will connect businesses directly with customers. Deployment will use **Docker**, **Coolify**, and a **VPS** with **GitHub CI/CD**, while **JWT**, **TLS encryption** ensure security and reliability.

PROJECT SCHEDULE/TIME PLAN	The project will be completed over two academic terms. **Term 1** focuses on developing and delivering the core MVP, while **Term 2** extends the system with advanced features, integrations, and performance improvements. • **Term 1 – MVP Development :** 1- Conduct full **system analysis and prototype design**, including requirements gathering, wireframes, and validation. 2- Develop the **backend** using FastAPI, PostgreSQL, and Redis, and build the **chatbot widget** along with a sample **admin dashboard** using React. 3- Integrate the initial **AI model** (NileChat or Gemini) with basic **emotion detection** and **RAG-based retrieval**. 4- Implement **authentication**, chat session handling, and deploy a working **text-based MVP** by the end of the term. • **Term 2 – Advanced Development and Integration :** 1- Complete any remaining MVP tasks and begin **Phase 2** of development. 2- Add **voice support** (Whisper + TTS), **advanced emotion detection**, and **Meta integrations** (WhatsApp, Messenger, Instagram). 3- Expand the **admin dashboard** to manage FAQs, tone customization, and analytics, and enhance the **chatbot widget** for better reliability and customization. 4- Optimize scalability, deploy the **enhanced version**, and deliver the **final demo** with full AI orchestration and business integrations.
RESOURCE REQUIREMENTS IF ANY	To successfully develop and deploy Kalamna, we will require access to several online services and cloud resources beyond the standard development tools. We will need **API access** to models such as **NileChat**, **Gemini**, or **Qwen** for natural language understanding, emotion detection, and text generation. Some of these APIs may require paid access or usage credits for large-scale testing and fine-tuning. For deployment, we will require **cloud hosting** through a **VPS server** or managed platforms such as **Coolify** or **Railway**, which may include minimal subscription or hosting fees. The **PostgreSQL database** and **Redis cache** will also be deployed on cloud infrastructure to ensure reliability and scalability. These resources will support the system's AI processing, storage, and deployment needs throughout the development and testing phases.