

OFFLINE AI CHATBOT USING AWS & DEVOPS

TEAM MEMBERS

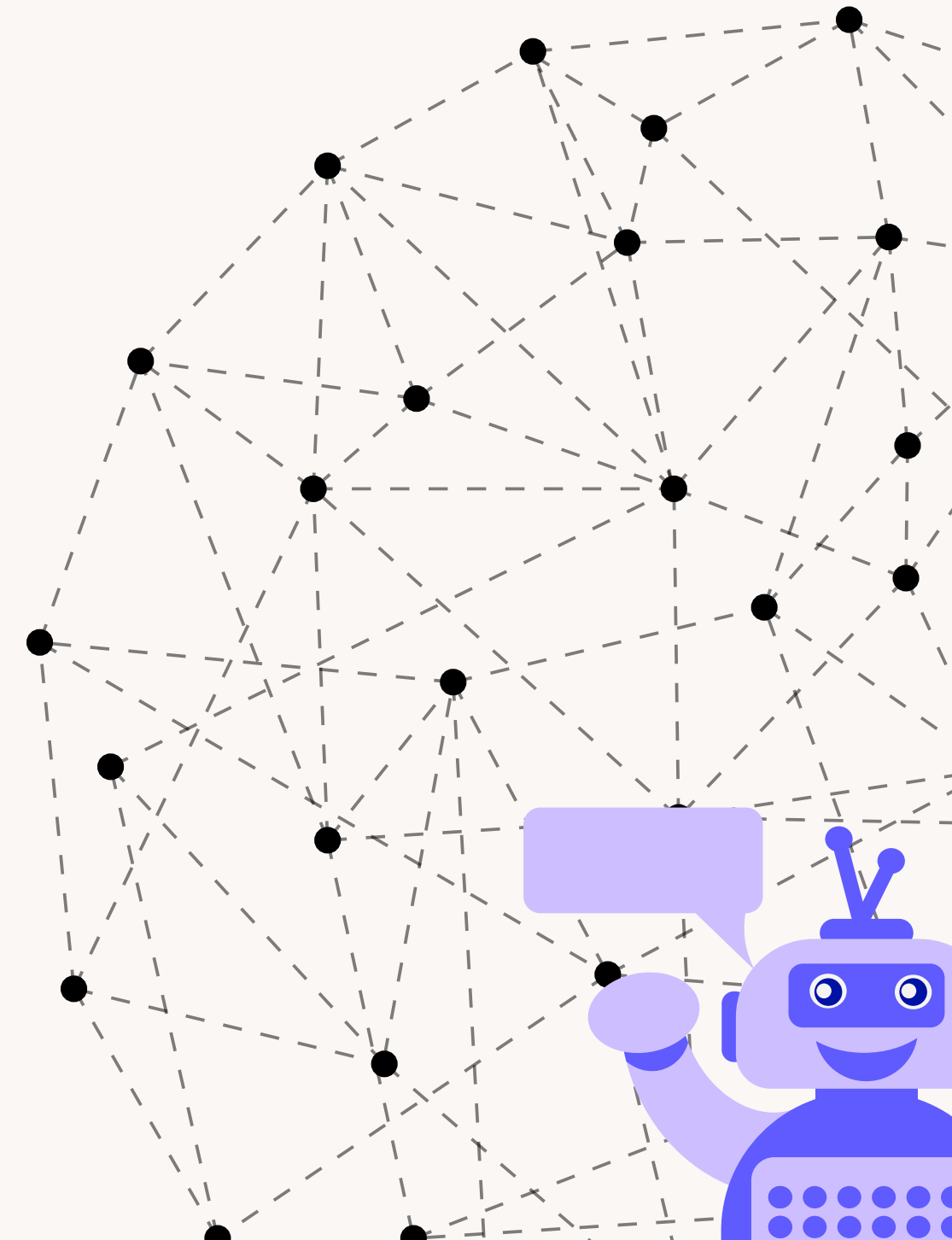
Botsa Gowthami (23mh1a05f1)

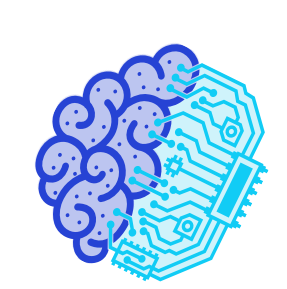
Konthala Yashwanth Reddy (23mh1a05m5)

Nuthalapathi Harthika (23mh1a0545)

Dulipalla Radhika (23mh1a0587)

K Shravani (23mh1a42e8)





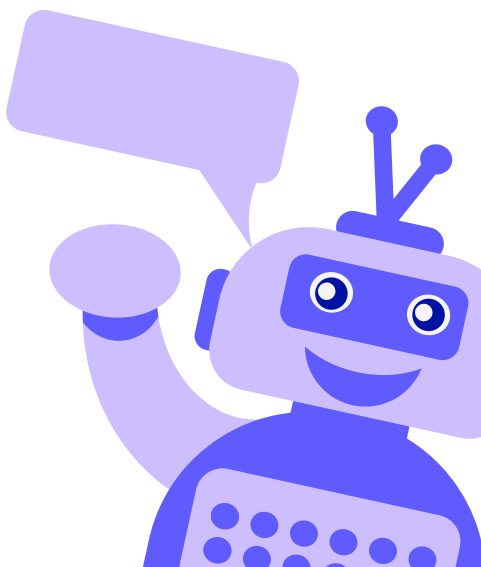
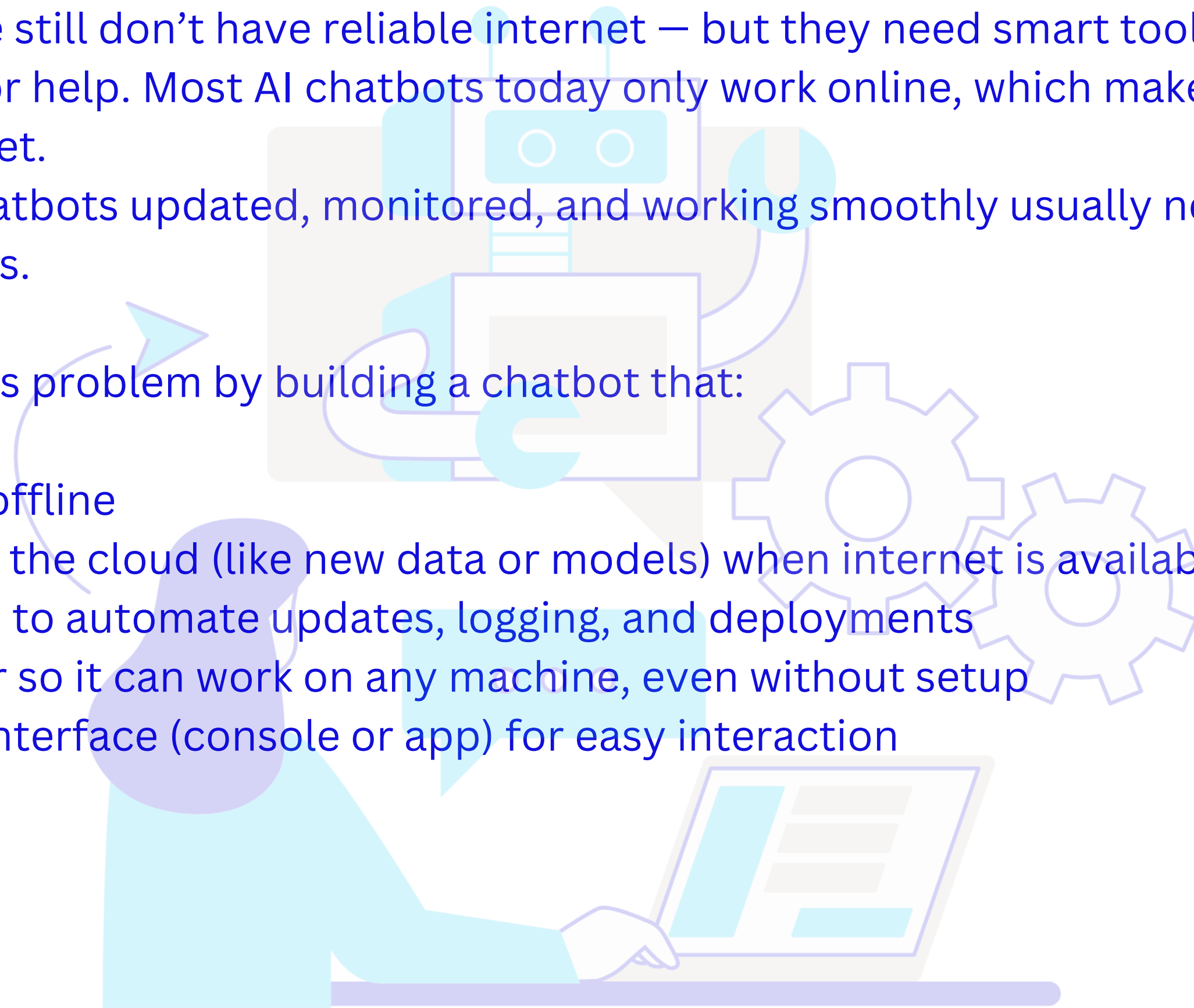
PROBLEM STATEMENT :

In many places, people still don't have reliable internet — but they need smart tools like chatbots for learning, support, or help. Most AI chatbots today only work online, which makes them useless when there's no internet.

Also, keeping these chatbots updated, monitored, and working smoothly usually needs manual work or complex setups.

We wanted to solve this problem by building a chatbot that:

- Works completely offline
- Gets updates from the cloud (like new data or models) when internet is available
- Uses DevOps tools to automate updates, logging, and deployments
- Runs inside Docker so it can work on any machine, even without setup
- Has a simple chat interface (console or app) for easy interaction



SOLUTION OVERVIEW

We built an offline AI chatbot that runs without internet using a lightweight NLP model, and auto-updates itself from AWS S3 using Docker + CI/CD + DevOps tools. It's smart, portable, and cloud-connected.

Offline AI chatbot (Core Brain)

- Uses a lightweight NLP model (e.g., DistilBERT or GPT-2) to understand and respond to user queries.
- Packaged into a Docker container so it can run on any system without setup.
- Designed to work completely offline once installed.

Cloud Integration (AWS)

- S3 stores the latest version of the AI model or chatbot data.
- EC2 is used for cloud testing and fallback hosting.
- IAM controls access securely.
- When the system detects an internet connection, it checks S3 for:
 1. New model updates
 2. Downloads and replaces the local model automatically

DevOps Automation

- AWS CodePipeline connects GitHub → S3 → EC2 for auto-deploying new versions.
- Docker is used to containerize the whole app for offline use.
- CloudWatch monitors logs, updates, and sync actions for debugging and transparency.
- Version checker ensures local chatbot syncs only when updates are available.



Chat Interface (UI)

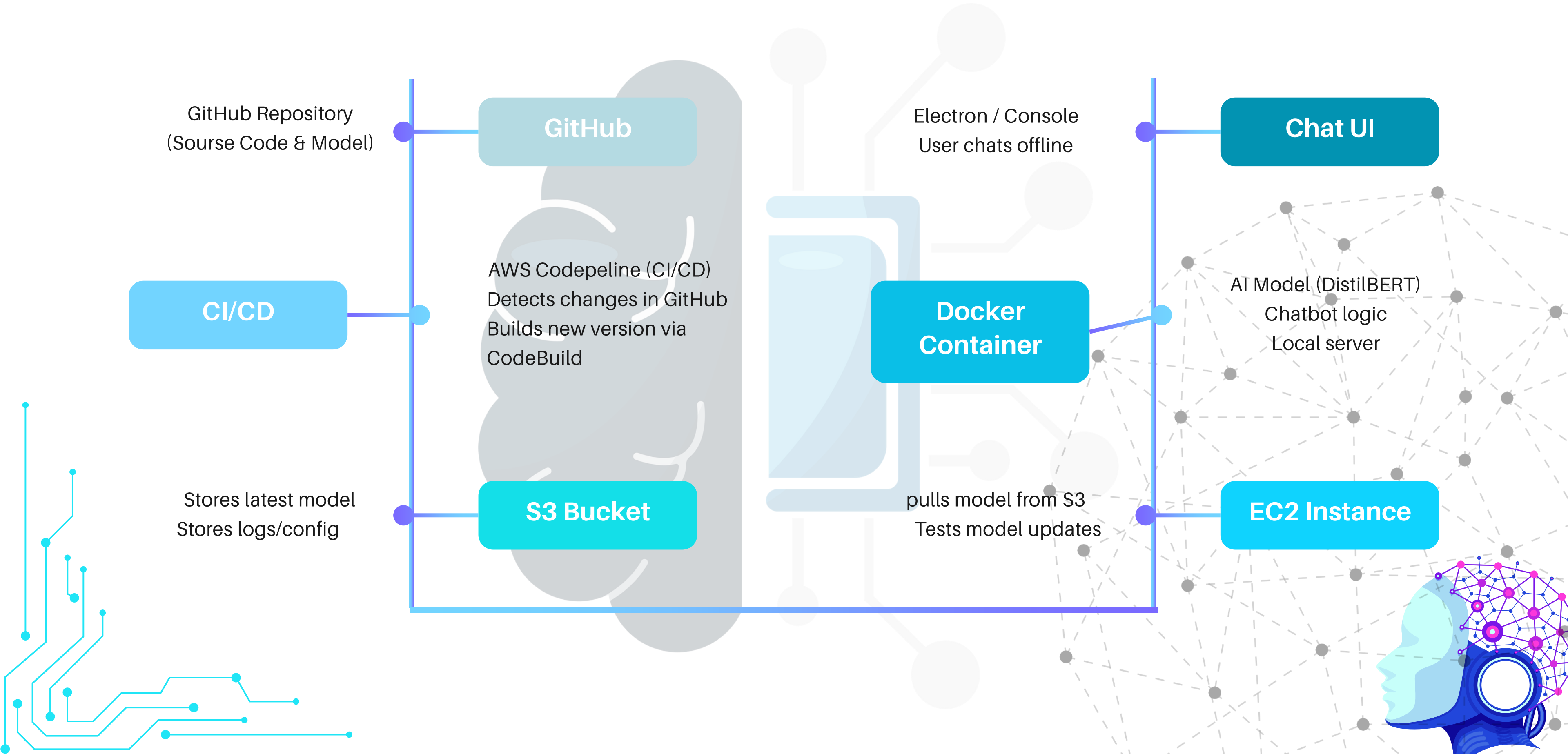
- A simple chat interface is built using:
- Electron.js or React.js for local desktop UI
- Or fallback to a console UI if short on time
- UI connects directly to the local chatbot backend for fast, real-time responses.

System Integration



- A simple chat interface is built using:
- Electron.js or React.js for local desktop UI
- Or fallback to a console UI if short on time
- UI connects directly to the local chatbot backend for fast, real-time responses.

System Architecture



ADVANTAGES



Lightweight AI Model : Uses DistilBERT or GPT-2 - runs locally with low resource usage.

Works Without Internet : Fully functional offline chatbot.

Cloud-Managed Updates : Automatically syncs model/data from AWS S3 when online.

DISADVANTAGES



Initial Setup is Complex: Requires knowledge of AWS , Docker , DevOps- learning curve

Limited When Offline Can't update or log to CloudWatch without internet.

Testing Requires Internet at Times For cloud parts (CI/CD, S3 pull), you still need to test with connectivity

Key Features

- Offline Functionality – Works without internet after setup
- Lightweight NLP Model – Uses DistilBERT/GPT-2 for fast local responses
- Cloud Sync – Downloads latest model/data from AWS S3 when online
- CI/CD Automation – GitHub → CodePipeline → S3 for smooth updates
- Dockerized App – Portable and easy to run on any machine
- Chat UI – Clean interface built with Electron or console UI
- Logging with CloudWatch – Monitors errors, sync status, and performance
- Secure Cloud Access – IAM roles to protect AWS resources
- Auto Update Script – Checks for model updates automatically
- Modular Architecture – Easily upgradable (model, UI, sync logic)



Real-World Use Cases

Use Case

Description

Rural Education

Students can ask questions even without internet access.

Healthcare Support

Provide offline medical info/chatbot in remote clinics.

Disaster Response

Deployed in areas with no network during emergencies.

Field Work

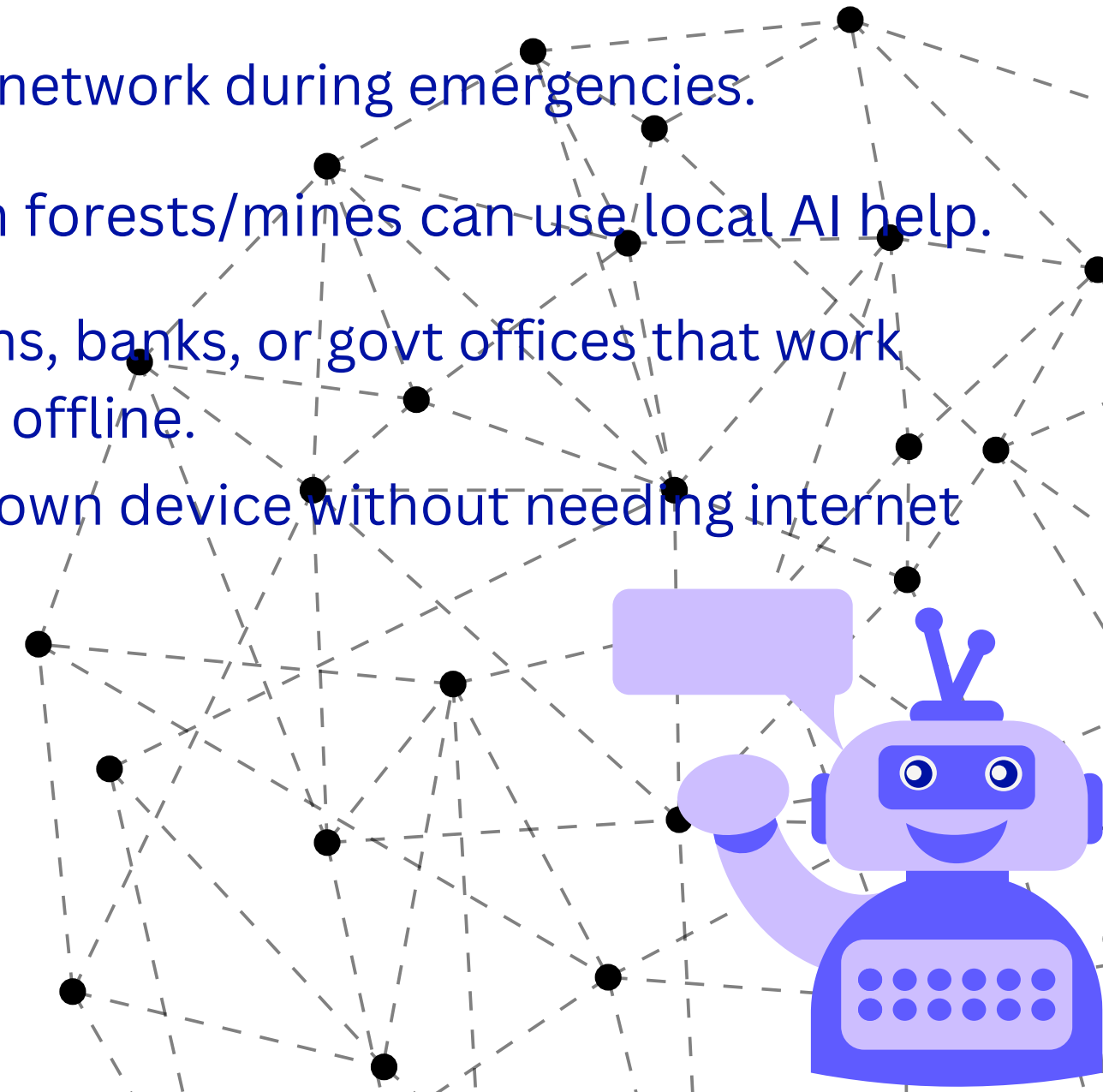
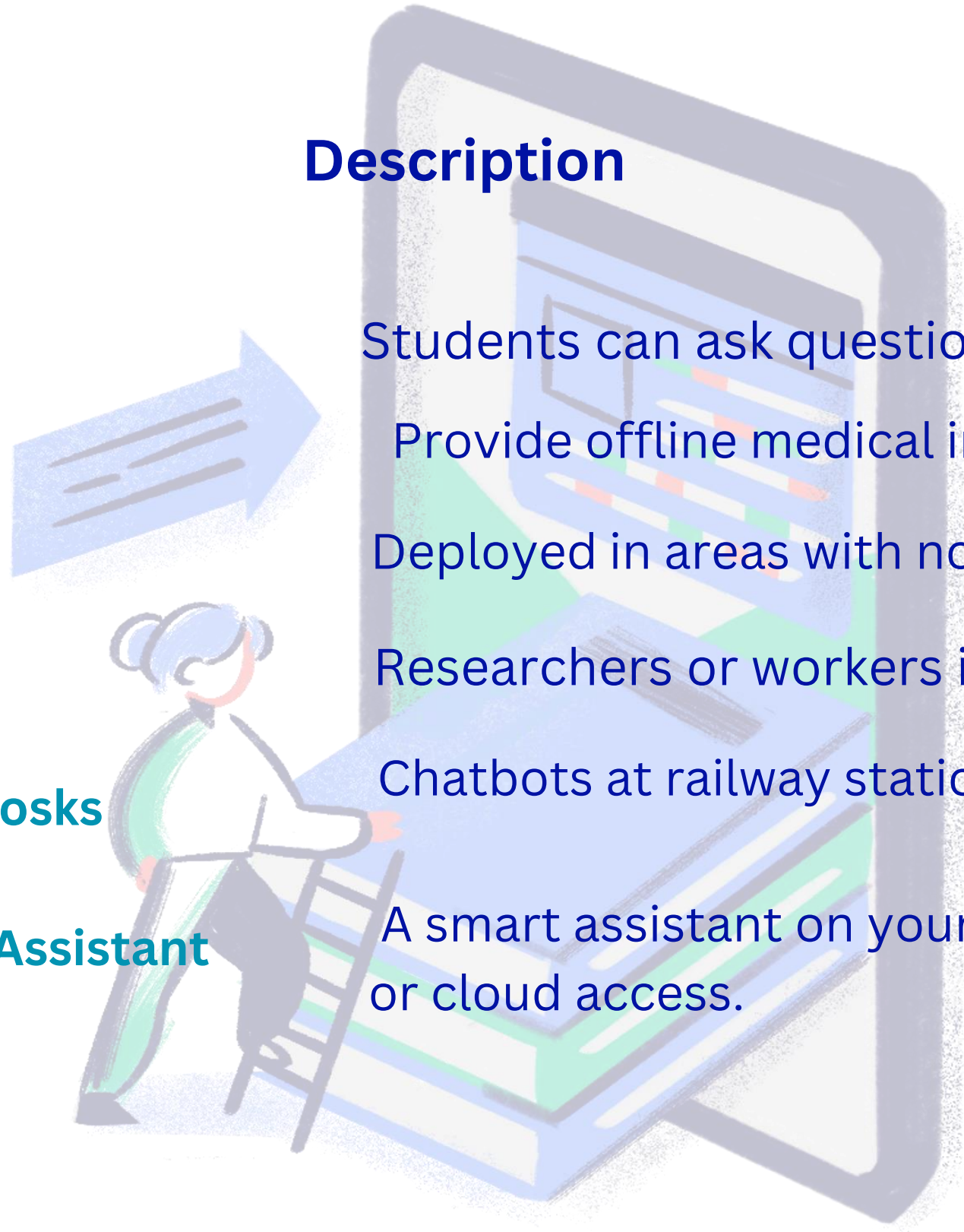
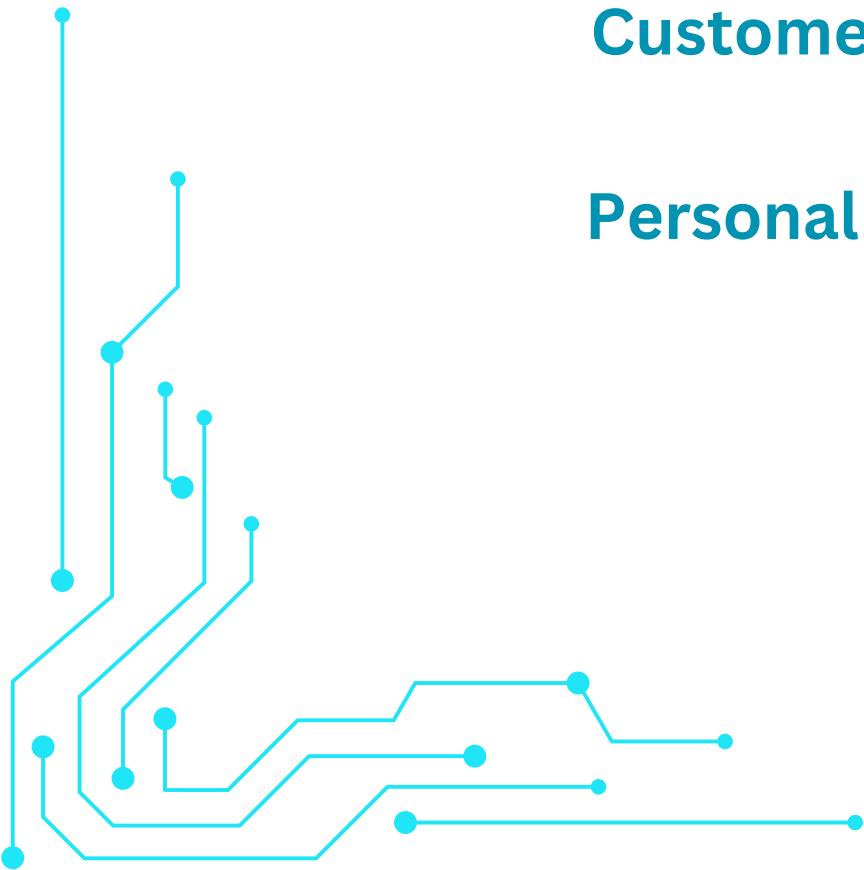
Researchers or workers in forests/mines can use local AI help.

Customer Support Kiosks

Chatbots at railway stations, banks, or govt offices that work offline.

Personal Knowledge Assistant

A smart assistant on your own device without needing internet or cloud access.



CONCLU -SION



Built a lightweight AI chatbot that works fully offline

Uses AWS S3 to sync updated models when internet is available

Packaged using Docker for easy offline deployment

- Managed using DevOps tools like CodePipeline and CloudWatch
- Includes a simple chat interface (console or Electron)
- Solves real-world need for AI support in low/no connectivity areas
- Gave us hands-on experience with AI, AWS, Docker, and CI/CD

THANK
YOU

