

DEXTER

PERSONAL MENTAL HEALTH ASSISTANT

By Nabhan, Aldrin, Anand and Midhun

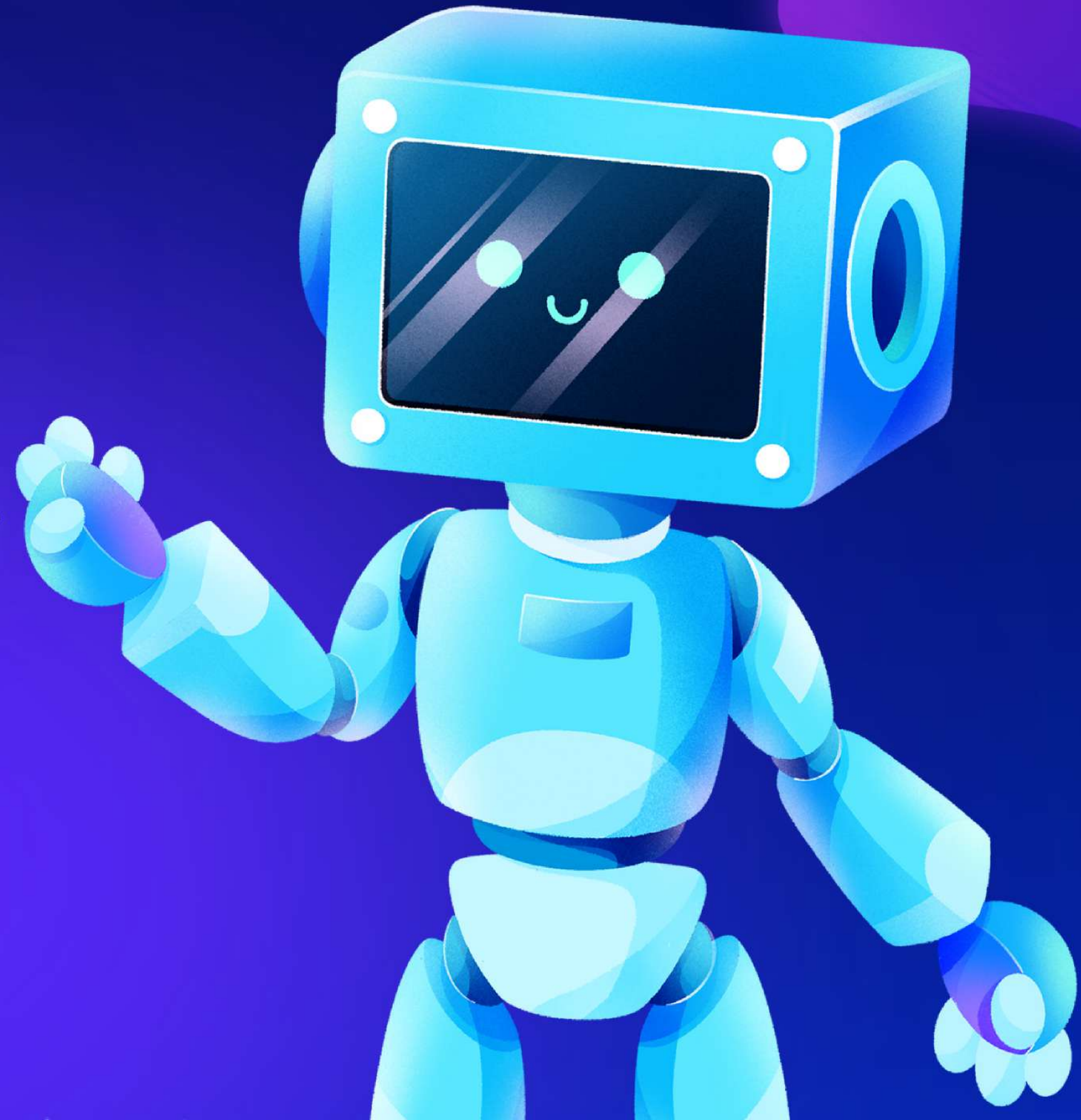




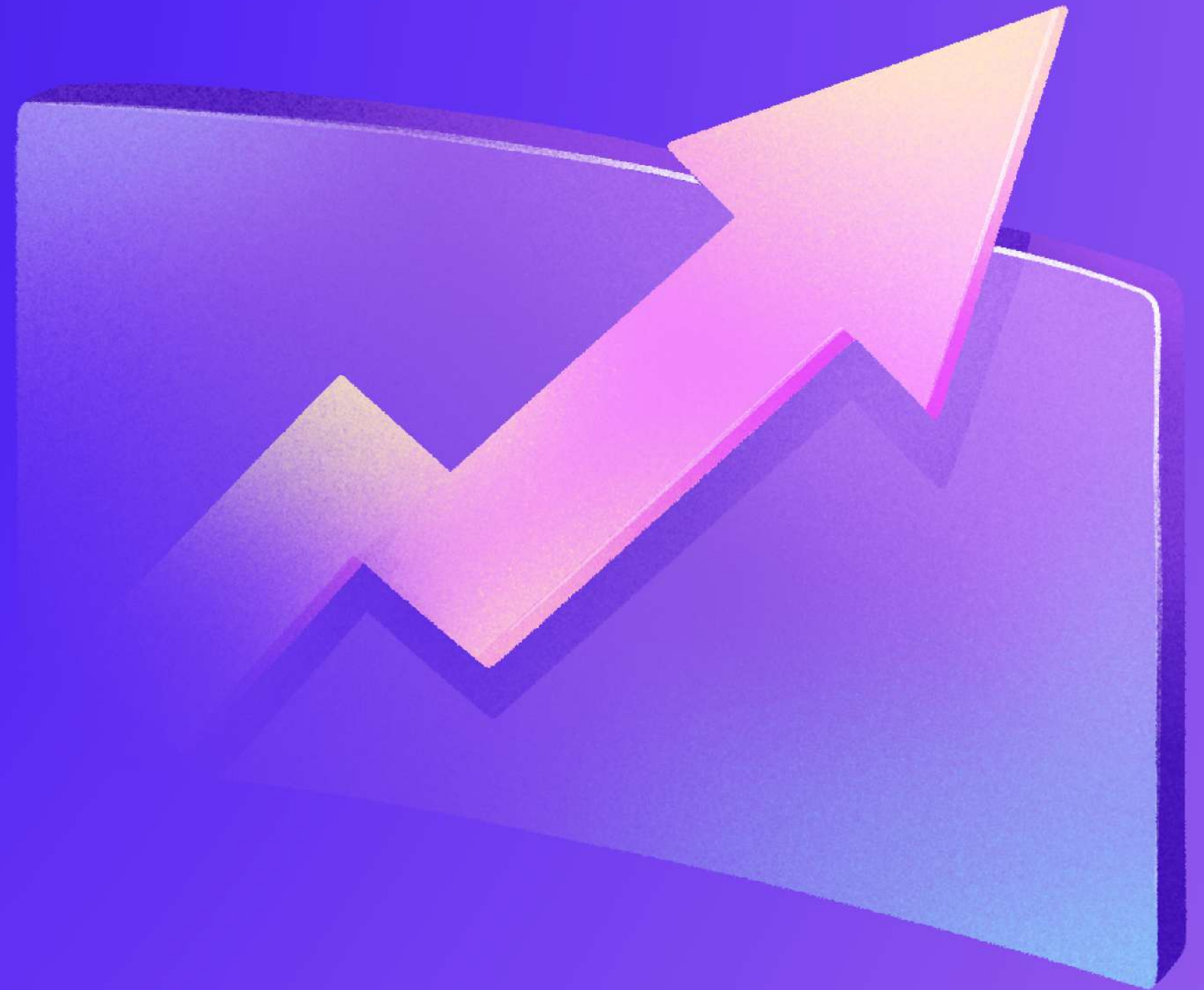
TABLE OF CONTENTS

• Problem Statement	01
• Unique Idea	02
• Features Offered	03
• Process Flow	04
• Project Architecture	05
• Technologies Used	06
• Team Contribution	07
• conclusion	08



PROBLEM STATEMENT

Although access to professional help and mental health resources is crucial for general well-being, many people still face barriers in this regard. Seeking help might be hampered by a lack of mental health specialists, and lengthy wait times. By using natural language understanding (NLU) to create an interactive chatbot, users can get help instantly, lessen the strain on mental health services, and use a private platform to ask for help.



UNIQUE IDEA

Personalized Interactions

Dexter uses advanced natural language understanding (NLU) to tailor responses based on user queries, providing a more personalized and empathetic interaction.

Instant response

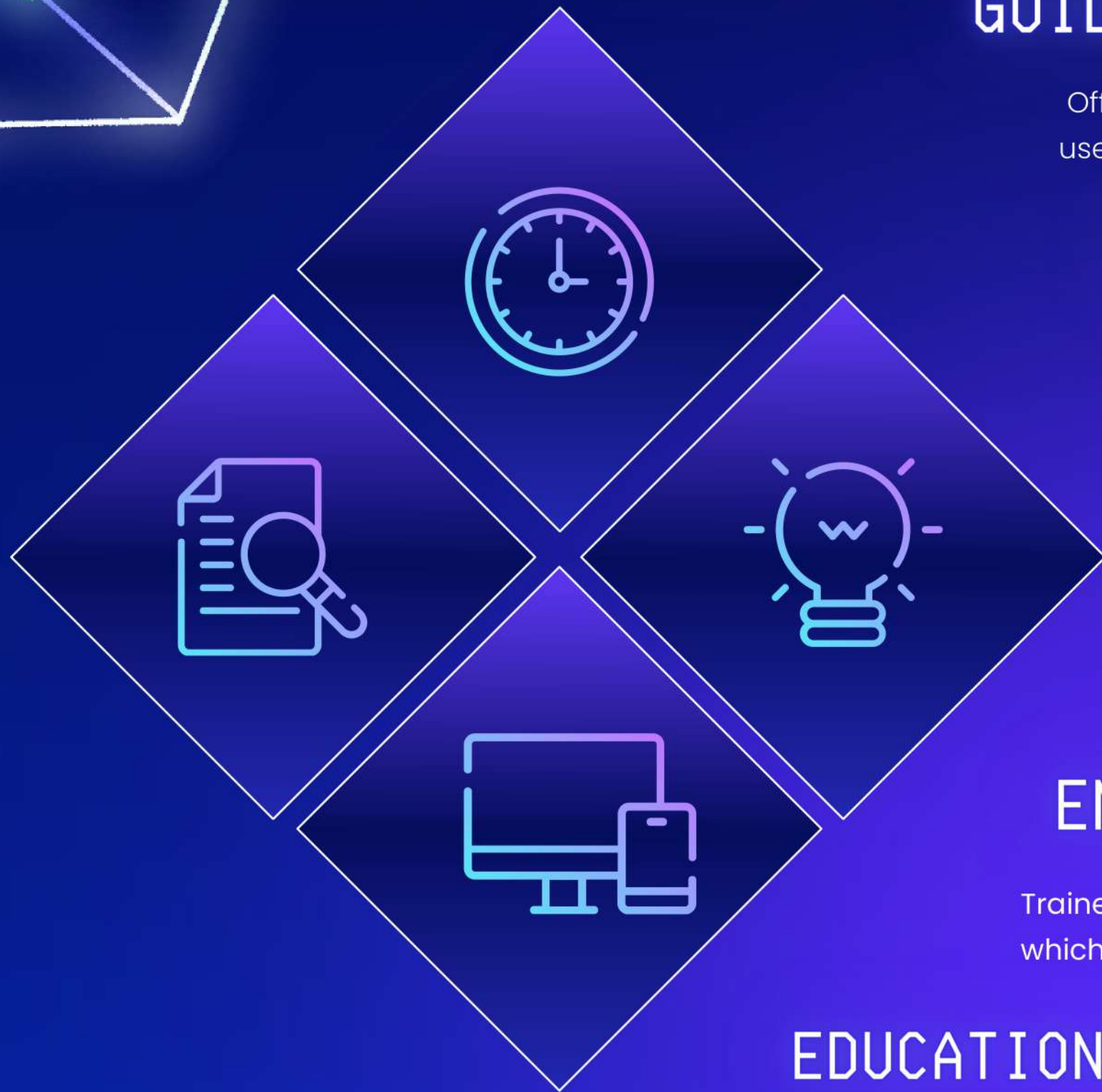
Leveraging the powerful Microsoft/dialo gpt model and optimized deployment on Intel Developer Cloud ensures quick and efficient responses to user inquiries, providing real-time support.

Easy Deployment

created with a combination of custom HTML and CSS for front-end design, Hugging Face models, and Intel Developer Cloud deployment, guaranteeing a scalable and reliable solution.



FEATURES OFFERED



GUIDANCE AND SUPPORT

Offers practical advice and emotional support to help users manage their mental health effectively.

INCREASED EFFICIENCY

Using intel developer cloud and microsoft dialo gpt provides output at a faster rate

ANONYMOUS INTERACTION

Users can communicate with dexter 24/7 while protecting their identity and info about mental health

ENHANCED ACCURACY

Trained and fine-tuned to provide users with the output which has high accuracy and relatable to questions

EDUCATIONAL CONTENT

Provides information and resources on various mental health topics, helping users learn more about mental health conditions and coping strategies.



PROCESS FLOW

01

DATA PREPARATION/FINE-TUNING

We start by gathering and structuring data relevant to mental health awareness. This data is crucial for fine-tuning the BART model using transfer learning techniques. Fine-tuning enhances the model's ability to understand and respond effectively to mental health-related queries.

- Dataset: We've created a dataset named output.json containing questions and answers related to mental health.
- Data Loading: We load the dataset from output.json, where we extract questions and answers.
- Tokenizer: We use the AutoTokenizer from Hugging Face's transformers library to tokenize text inputs.
- Model: The language model used for fine-tuning is BartForConditionalGeneration from Hugging Face, is called micosoft dialo/gpt
- Fine-tuning Parameters: Fine-tuning is performed using Trainer from transformers.Trainer with parameters specified in TrainingArguments.

PROCESS FLOW



02

Model Deployment

After fine-tuning, the model is deployed to a production environment where it can interact with users.

- The fine-tuned model (BartForConditionalGeneration) and tokenizer (AutoTokenizer) are saved to ./fine_tuned_model for deployment.
- We use these files in our deployment setup to ensure the chatbot can load and utilize the trained model efficiently.

PROCESS FLOW

03

Frontend Development:

We create a user-friendly interface for users to interact with the chatbot.

- **HTML/CSS Interface:** We design a frontend using HTML and CSS. Key elements include:
- **Styling:** The interface uses Bona Nova SC font and incorporates colors like #6d839b and #007bff.
- **Layout:** Elements are organized into a centered container (container) with separate sections for chat output (chat-output) and user input (input-container).
- **Functionality:** Buttons (button) are styled for interaction (Send button), and there's an About button (type2) toggling a message (about-message).
- **Interaction:** JavaScript functions (sendMessage(), toggleAboutMessage()) handle user input and toggle visibility of the About message.

PROCESS FLOW

04

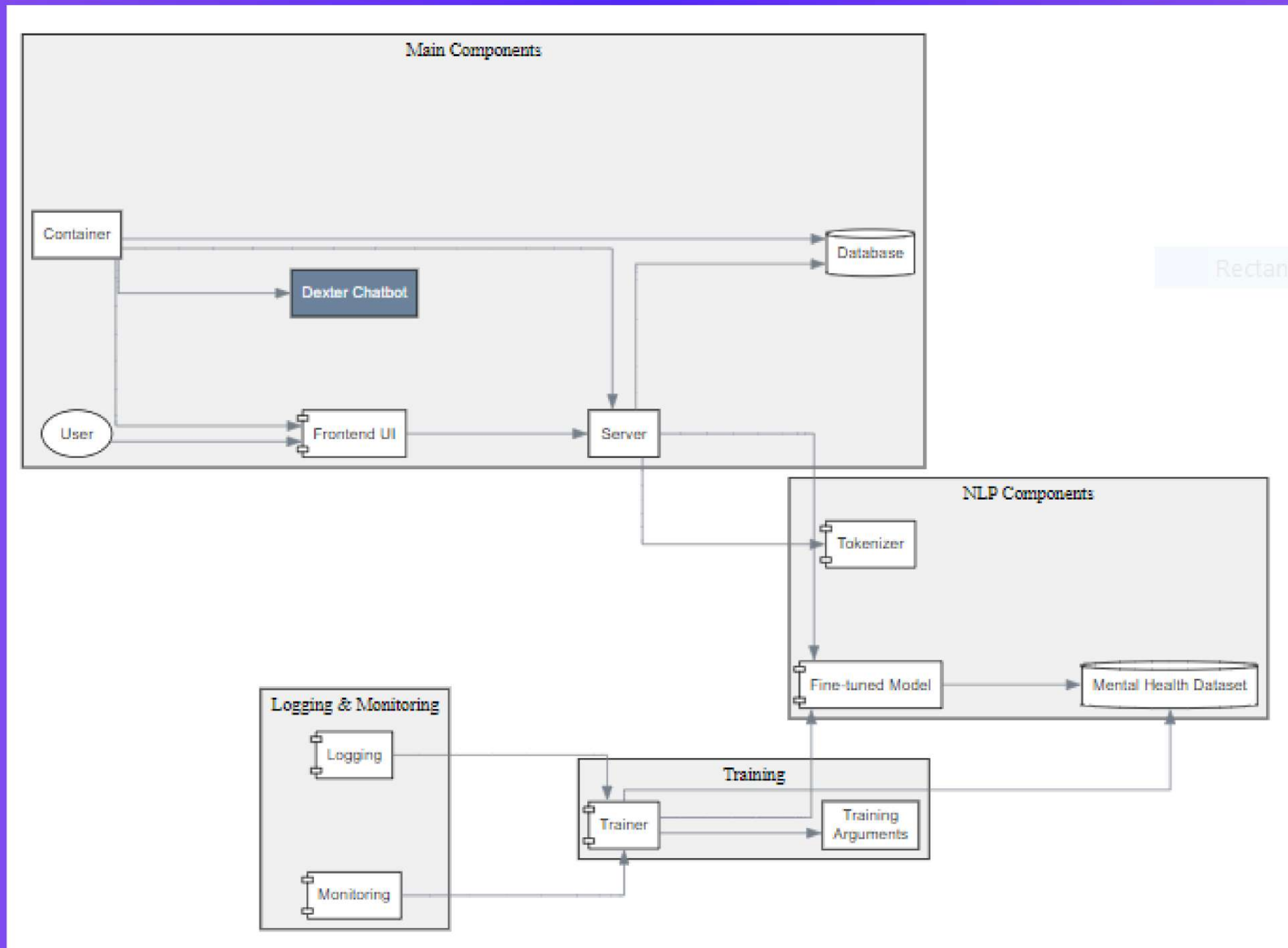
Real-time Interaction

We enable seamless communication between users and the chatbot.

- User Input Handling: The `sendMessage()` JavaScript function captures user input from input and appends it to chat-output.
- Bot Response: After user input is sent (onclick), simulated bot responses are generated using functions like `getBotResponse(message)`.
- Messaging: Messages are styled differently (user-message, bot-message) based on the speaker (User, Dexter).
- Exit Command: The chat ends gracefully upon receiving bye from the user, setting `chatActive` to false.

PROJECT ARCHITECTURE





created using graphviz

TECHNOLOGIES USED

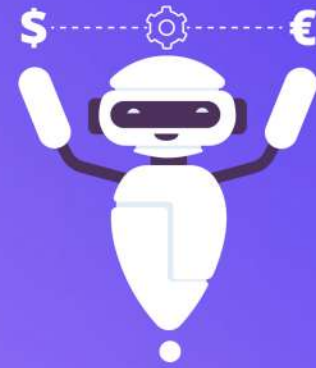
- **Transformers Library:** Used for natural language processing tasks like tokenization, model training, and inference using transformer-based models.
- **Hugging Face Transformers:** Open-source library providing pre-trained models and tools for building conversational AI systems and natural language understanding (NLU) applications.
- **Sklearn** (Scikit-learn): Python library for machine learning built on NumPy, SciPy, and matplotlib, providing simple tools for data analysis and modeling.
- **HTML and CSS:** Standard languages for creating and styling web pages, used to design and structure the frontend interface of the chatbot.
- **Python:** High-level programming language used for backend server scripting, model fine-tuning, and general-purpose tasks in the project.
- **JSON:** Lightweight data interchange format used for storing and exchanging structured data between the server and frontend components of the chatbot.
- **Graphviz:** Open-source graph visualization software used to create diagrams and flowcharts, aiding in visual representation of the project's architecture and workflow.

CONTRIBUTIONS



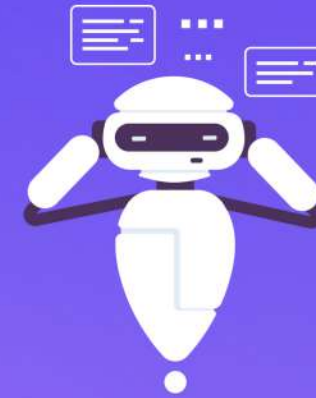
ALDRIN

Found the most suitable Model (microsoft dialogpt) for our Chatbot named DEXTER and also played a vital role in creating the dataset, chatbot code, training and configuring it



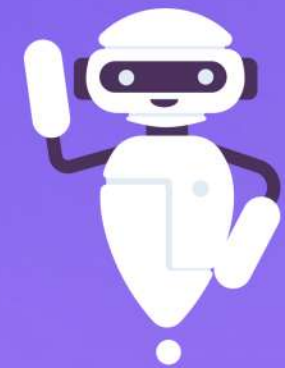
NABHAN

Found the Topic for chatbot, created frontend interface and report (html,css,js) helped in creating dataset, optimizing and finetuning model for our chatbot



ANAND

Optimizing, finetuning and adjusting the model ,adjusting train loss and also helped in building front end interface and dataset creation for the chatbot



MIDHUN

created the dataset, pre processing, cleaning and conversion to json file. Also helped in creating custom chatbot code

CONCLUSION

- We successfully developed a chatbot named Dexter, aimed at providing mental health awareness and support using NLP techniques with the Transformers library and Hugging Face models, we fine-tuned our model to understand and respond to user queries effectively. We used HTML and CSS for our frontend design to create an easy-to-use interface, and Python was used for backend tasks like data processing and server management. We developed a user-friendly platform that improves access to mental health resources and assistance by integrating these technologies.
-

THANK YOU!

