Photo by Pexels

Low Level Design for Product Chat Assistant

Enhancing Online Shopping with AI

Table of Contents

01 Introduction to Product Chat Assistant

02 Scope of the Project

03 Architecture Overview

04 Data Preparation Using OpenAI & Gemini

05 Data Preprocessing Techniques

06 Pushing Dataset to Hugging Face

07 Loading Llama2 for Fine-Tuning

08 Pushing the Trained Model Adapter File

09 Creating a Streamlit App

10 Deployment and Conclusion

1

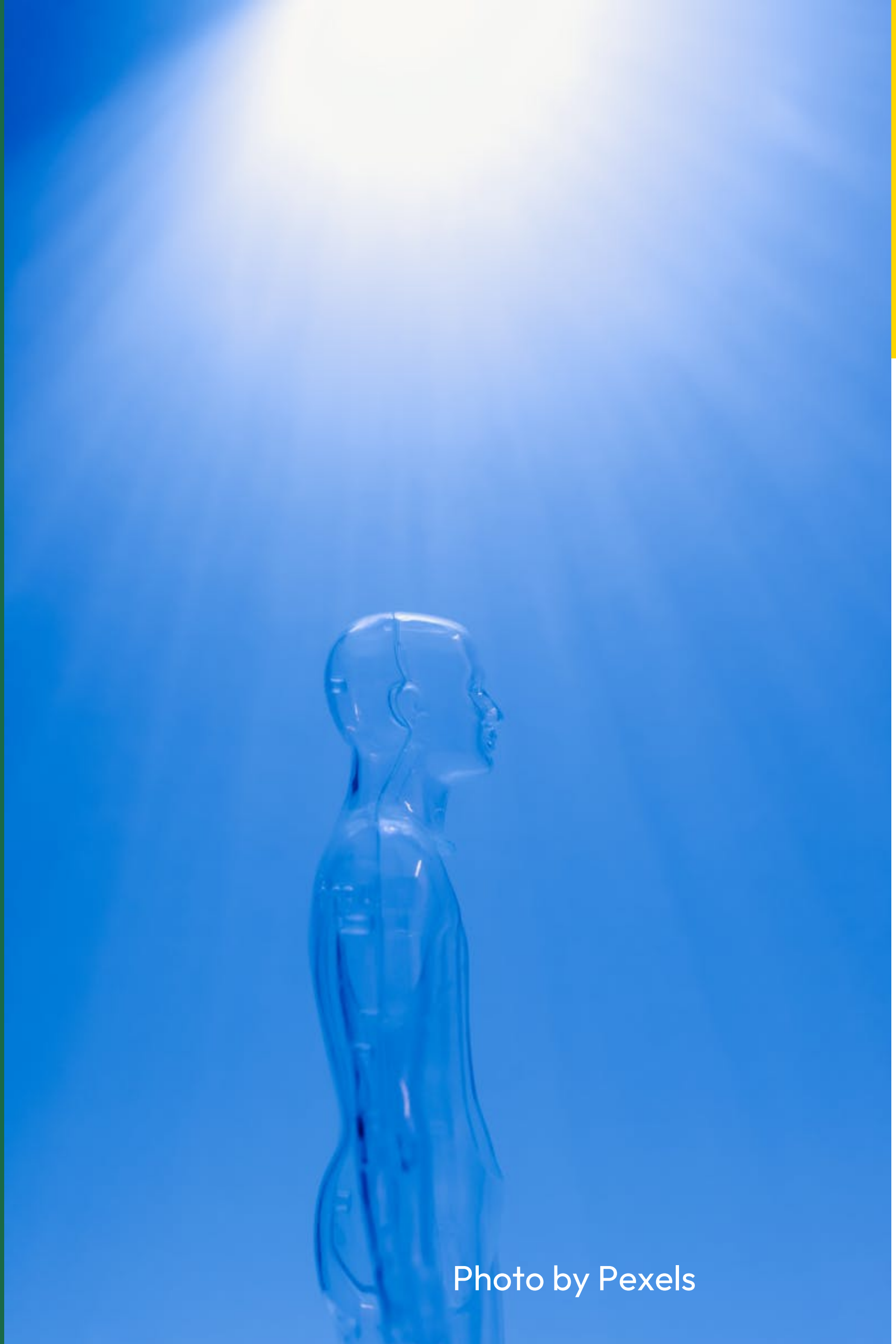

Photo by Pexels

Introduction to Product Chat Assistant

AI-driven App

- The Product Chat Assistant is an AI-driven application designed to enhance the online shopping experience.
- It interacts with users naturally, providing detailed product information for informed decision-making.
- The assistant answers queries and negotiates prices to improve customer satisfaction and drive sales.
- Key features include product information retrieval, interactive conversations, price negotiation, and personalized recommendations.

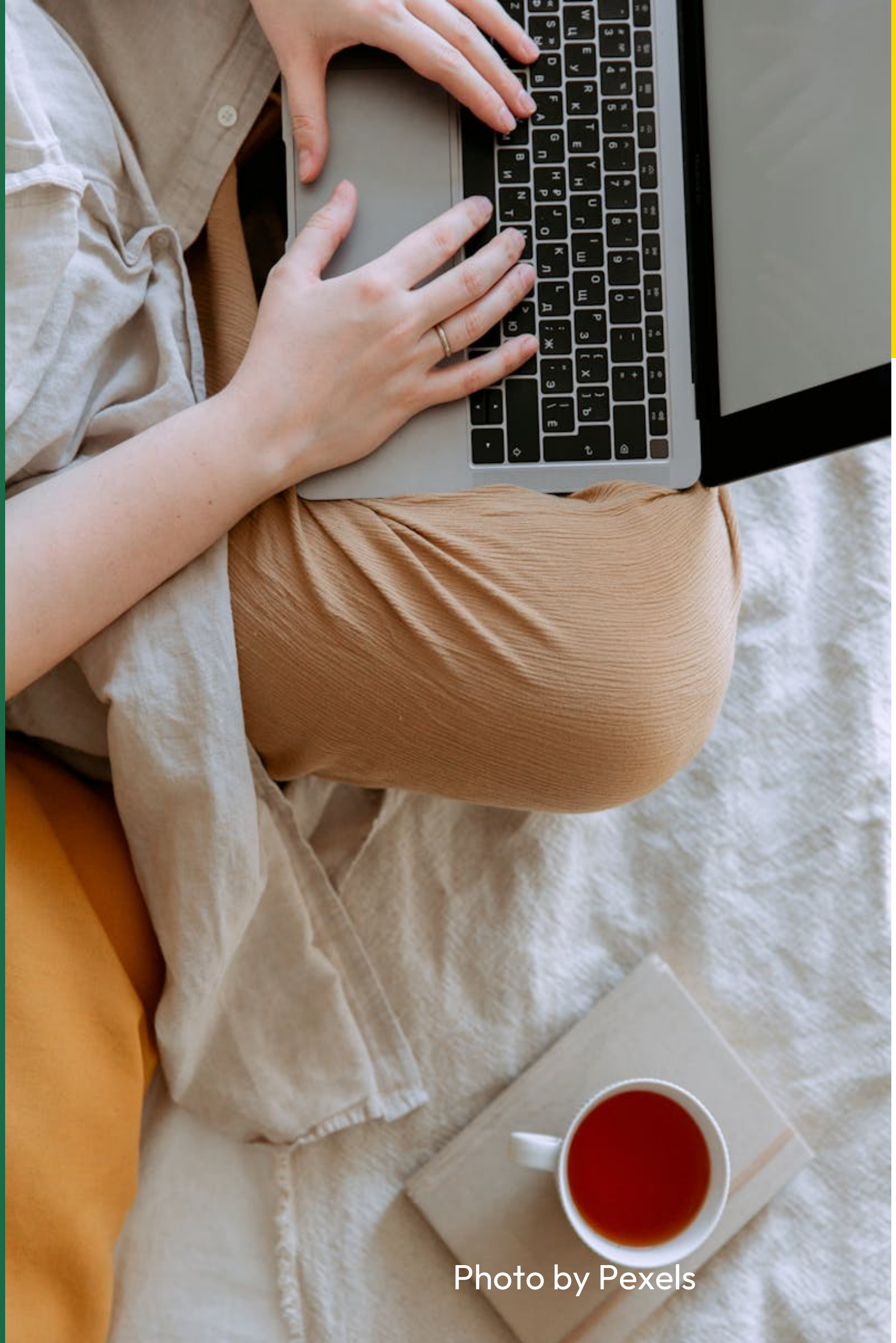

Photo by Pexels

2

Scope of the Project

AI Assistant Goal

- The primary aim is to develop an AI assistant for personalized product support.
- The assistant will integrate with multiple channels, including websites, mobile apps, and social media.
- It focuses on delivering a consistent user experience across all platforms.
- The scope includes data preparation, AI model training, and deployment.

Photo by Pexels

3

Architecture Overview

System Design

- The architecture includes data preparation using OpenAI & Gemini, and data preprocessing steps.
- It also covers pushing datasets to Hugging Face for further processing and storage.
- Loading Llama2 for fine-tuning the AI model is a critical part of the architecture.
- Finally, pushing the trained model's adapter file on the hub ensures easy accessibility.

4

Data Preparation Using OpenAI & Gemini

Photo by Pexels

Data Preparation

- Data preparation involves using OpenAI & Gemini tools for initial dataset creation.
- The tools help in gathering relevant data, ensuring quality and consistency.
- Proper data preparation is crucial for effective AI model training.
- This step sets the foundation for the subsequent processes in the project.

o by Pexels

5

Data Preprocessing Techniques

Clean & Process

- Data preprocessing involves cleaning and organizing the data to make it suitable for AI model training.
- It includes removing inconsistencies, handling missing values, and normalizing the data.
- This step improves the quality and reliability of the input data.
- Effective preprocessing enhances the performance of the AI model.

6

Pushing Dataset to Hugging Face

Photo by Pexels

Data Storage

- After preprocessing, the dataset is pushed to Hugging Face for storage and further processing.
- Hugging Face provides tools and infrastructure for managing large datasets efficiently.
- It allows easy access and sharing of data with other components of the system.
- This step ensures that the dataset is readily available for model training.

7

Loading Llama2 for Fine-Tuning

Photo by Pexels

Model Training

- Llama2 is loaded for fine-tuning the AI model to improve its performance.
- Fine-tuning involves adjusting the model parameters based on the prepared dataset.
- This step is crucial for achieving high accuracy and relevance in responses.
- Properly trained models enhance the user experience significantly.

Photo by Pexels

8

Pushing the Trained Model Adapter File

Model Deployment

- The trained model's adapter file is pushed to the hub for easy accessibility and deployment.
- This file contains the configurations and parameters necessary for the AI model to function.
- Deploying the adapter file ensures that the model can be used effectively in real-world scenarios.
- It simplifies the integration process with different platforms.

Photo by Pexels

Creating a Streamlit App

User Interface

- A Streamlit app is created to provide a user-friendly interface for interacting with the AI assistant.
- The app allows users to input queries, receive responses, and navigate through product information easily.
- It enhances the accessibility and usability of the AI assistant.
- Streamlit simplifies the development process with its easy-to-use framework.

Photo by Pexels

10

Deployment and Conclusion

Final Steps

- The deployment involves integrating the Streamlit app with various platforms like websites and mobile apps.
- It also includes testing and optimizing the system for performance and reliability.
- The conclusion summarizes the project outcomes and the benefits of the AI assistant.
- Future enhancements and potential improvements are also discussed.