# INTELLIGENT ASSISTANT

*A Report*

*Submitted in partial fulfilment of the Requirements for the completion of*

*THEME BASED PROJECT*

## BACHELOR OF ENGINEERING

IN

## INFORMATION TECHNOLOGY

By

## M. Sanath 1602-21-737-171

## K. Shashank Goud 1602-21-737-174

## C. Shreya Sree 1602-21-737-177

## Under the guidance of

## Mr. K. Srinivasa Chakravarthy

## Assistant Professor



**Department of Information Technology**

**Vasavi College of Engineering (Autonomous)**

**ACCREDITED BY NAAC WITH 'A++' GRADE.**

**(Affiliated to Osmania University and Approved by AICTE)**

**Ibrahim Bagh, Hyderabad-31**

**2024**

**Vasavi College of Engineering (Autonomous)**

***ACCREDITED BY NAAC WITH 'A++' GRADE***

**(Affiliated to Osmania University and Approved by AICTE)**

**Ibrahim Bagh, Hyderabad-31**

**Department of Information Technology**



### DECLARATION BY CANDIDATES

We, **M. Sanath ,K. Shashank Goud and C. Shreya Sree,** bearing hall ticket number, **1602-21-737-171 ,1602-21-737-174 and 1602-21-737-177,** hereby declare that the project report entitled  **”Intelligent Assistant”** under the guidance of **Mr. K. Srinivasa Chakravarthy, Assistant Professor**, Department of 5Information Technology, Vasavi College of Engineering, Hyderabad, is submitted in partial fulfillment of the requirement for the completion of Theme-based project , VI semester, Bachelor of Engineering in Information Technology.

This is a record of bonafide work carried out by us and the results embodied in this project report have not been submitted to any other institutes.

## M. Sanath, 1602-21-737-171

## K. Shashank Goud, 1602-21-737-174

## C. Shreya Sree, 1602-21-737-177

**Vasavi College of Engineering (Autonomous)**

***ACCREDITED BY NAAC WITH 'A++' GRADE***

**(Affiliated to Osmania University and Approved by AICTE)**

**Ibrahim Bagh, Hyderabad-31**

**Department of Information Technology**

## A logo of a flower Description automatically generated

### BONAFIDE CERTIFICATE

This is to certify that the project entitled “**Intelligent Assistant**” being submitted by **M. Sanath, K. Shashank Goud and C. Shreya Sree** bearing **1602-21-737-171 ,1602-21-737-174 and 1602-21-737-177,** in partial fulfillment of the requirements for the completion of Theme-based project of Bachelor of Engineering in Information Technology is a record of bonafide work carried out by them under my guidance.

|  |  |  |
| --- | --- | --- |
| **Mr. K. Srinivasa Chakravarty** | **External Examiner** | **Dr. K. Ram Mohan Rao** |
| **Assistant Professor** |  | **Professor, HOD IT** |
|  |  |  |

## ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of the project would not have been possible without the kind support and help of many individuals. We would like to extend our sincere thanks to all of them.

It is with immense pleasure that we would like to take the opportunity to express our humble gratitude to **Mr. K. Srinivasa Chakravarthy, Assistant Professor, Information Technology** under whom we executed this project. His constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks.

We are very much thankful to **Dr. K. Ram Mohan Rao, Professor and HOD, Information Technology,** for his kind support and for providing necessary facilities to carry out the work.

We wish to convey our special thanks to Dr. S. V. Ramana**, Principal** of **Vasavi College of Engineering** for giving the required information in doing my project work. Not to forget, we thank all other faculty and non-teaching staff, and my friends who had directly or indirectly helped and supported me in completing my project in time.

We also express our sincere thanks to the Management for providing excellent facilities. Finally, we wish to convey our gratitude to our family who fostered all the facilities that we need.

**ABSTRACT**

Develop a personal intelligent assistant application capable of understanding natural language commands, accessing and processing information from various sources, and performing automated tasks to improve user efficiency and provide a personalized experience.

**Existing System:**

The existing systems for intelligent assistants, such as Siri, Google Assistant, and Alexa, are built on artificial intelligence (AI) and machine learning (ML) technologies. These systems use natural language processing (NLP) algorithms to understand and respond to user commands and queries. They can perform various tasks like answering questions, providing information.

**Proposed Solutions:**

In addition to the existing features mentioned above we're also adding extra feature by incorporating image input, the assistant will directly assist students in efficiently annotating objects within the image. This functionality differentiates our assistant from current offerings like Google Assistant, Siri, and Alexa, which primarily focus on web searches and task completion.

**Conclusion:**

This project has successfully developed a novel intelligent assistant that bridges the gap between image understanding and student learning. By leveraging the Gemini API, custom search functionalities, and image processing techniques, the assistant empowers students to efficiently annotate objects within images based on prompts. This innovative approach sets it apart from existing virtual assistants like Google Assistant, Siri, and Alexa, which primarily focus on web searches and task completion.

### TABLE OF CONTENTS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. INTRODUCTION | | | | 1 | | | |
| 1.1 Overview | | | | 1 | | |
| 1.2 Problem Statement | | | | 1 | | |
| 1.3 Motivation of theme & title | | | | 1 | | |
| 2. LITERATURE SURVEY | | | | 2 | | | |
| 3. EXISTING SYSTEM | | | | 4 | | | |
| 4. PROPOSED SOLUTION | | | | 4 | | | |
| 4.1. System Design | | | | 4 | | |
| 4.1.1 Architecture Diagram | | | | 4 | | |
| 4.1.2 Use-Case Diagram | | | | 5 | | |
| 4.1.2.1 Use-case descriptions | | | | 5 | | |
| 4.2 Functional Modules | | | | 6 | | |
| 4.2.1 Pseudocode & Screenshorts | | | | 6 | | |
| 5. EXPERIMENTAL SETUP & IMPLEMENTATION | | | | 9 | | | |
| 5.1 System Specifications | | | | 9 | | |
| 5.1.1 Hardware Requirements | | | | 9 | | |
| 5.1.2 Software Requirements | | | | 9 | | |
| 5.2 Datasets | | | | 9 | | |
| 5.3 Methodology/Algorithm | | | | 9 | | |
| 6. RESULTS | | | | 11 | | | |
| 7. CONCLUSION & FUTURE SCOPE | | | | 15 | | | |
| 8. REFERENCES | | | | 15 | | | |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No** |
| 4.1.1 | Architecture Diagram | 4 |
| 4.1.2 | Use-Case Diagram | 5 |
| 4.2.a | Index.html Code | 7 |
| 4.2.b | Object Annotation Code | 8 |
| 6.1 | Intelligent Assistant | 11 |
| 6.2 | Code Assistant | 11 |
| 6.3 | Object Annotation Web Page | 12 |
| 6.4 | Object Annotation Result | 12 |

**1. INTRODUCTION**

**1.1 OVERVIEW**

The project involves developing an advanced intelligent assistant that integrates three core features : image annotation, voice commands, and real-time query support. This assistant is designed to improve productivity and streamline workflows by offering a seamless, user-friendly interface that combines multiple functionalities.

Key Features:

Image Annotation:

Automatic Analysis: Users can upload images, which the assistant automatically analyzes using sophisticated image processing algorithms.

Accurate Tagging: The system annotates images with relevant tags and descriptions, eliminating the need for manual input and ensuring high accuracy.

Voice Commands:

Natural Language Processing: The assistant supports voice commands, allowing users to interact with it using natural language.

Hands-Free Operation: This feature is particularly useful for multitasking or in environments where manual input is impractical, enhancing convenience and efficiency.

Real-Time Query Support:

Comprehensive Knowledge Base: The assistant can respond to project-related questions in real-time by accessing a constantly updated knowledge base.

Instant Responses: Users receive immediate and accurate information, facilitating quick decision-making and effective problem-solving.

**1.2 PROBLEM STATEMENT**

Develop a intelligent assistant application capable of understanding natural language commands, accessing and processing information from various sources, and performing automated tasks to improve user efficiency and provide a personalized experience.

**1.3 MOTIVATION OF THEME & TITLE**

This intelligent Assistant is very helpful for the people who are very busy in their daily chores and have less time to complete their project work within time so in this case intelligence assistant is very helpful. This reduces the work time on a particular task. By using the assistant we can minimize work time. This also helps the students to access web easily and search the items they want.

**2. LITERATURE SURVEY**

**1. Introduction to Voice-Enabled Virtual Assistants**

Voice-enabled virtual assistants have become ubiquitous in modern technology, offering users a hands-free and intuitive way to interact with devices and services. These assistants, such as Amazon Alexa, Google Assistant, and Apple Siri, leverage natural language processing (NLP) and artificial intelligence (AI) to understand and respond to user queries and commands.

**2. Traditional Voice Recognition Methods**

**Keyword Spotting**: Early voice recognition systems relied on predefined keywords or phrases to trigger actions. While simple, they lacked flexibility and struggled with variations in speech patterns and accents.

**Hidden Markov Models (HMMs)**: HMMs were widely used for speech recognition, modeling sequential data and capturing transitions between phonemes. However, they required extensive training data and were computationally expensive.

**3. Machine Learning Approaches**

**Supervised Learning**: Techniques like Gaussian Mixture Models (GMMs) and Deep Neural Networks (DNNs) have been employed for voice recognition tasks. These models require labeled training data and often struggle with variations in speech and background noise.

**Unsupervised Learning**: Clustering algorithms like k-means and Gaussian Mixture Models (GMMs) have been explored for speaker diarization and voice activity detection. However, their performance can be limited in real-world environments.

**4. Deep Learning for Voice Recognition**

**Recurrent Neural Networks (RNNs)**: RNNs, particularly Long Short-Term Memory (LSTM) networks, have shown promise in modeling sequential data like speech. They can capture temporal dependencies and handle variable-length input sequences effectively.

**Convolutional Neural Networks (CNNs)**: CNNs have been applied to speech recognition tasks, particularly in acoustic modeling. They can learn hierarchical representations of audio features, making them suitable for tasks like phoneme classification and speech-to-text conversion.

**5. End-to-End Voice Recognition Systems**

**End-to-End Models**: Recent advancements in deep learning have led to the development of end-to-end voice recognition systems, where a single neural network can directly map input audio to text output. These models eliminate the need for handcrafted features and intermediate processing steps, leading to more streamlined architectures.

**Attention Mechanisms**: Attention mechanisms have been integrated into voice recognition systems to focus on relevant parts of the input sequence, improving performance on long audio recordings and reducing computational overhead.

**6. Applications in Voice-Enabled Assistants**

**Natural Language Understanding (NLU)**: Voice-enabled assistants use NLU techniques to parse user queries and extract relevant information. This allows them to perform tasks like setting reminders, answering questions, and controlling smart home devices.

**Personalization and Context Awareness**: Voice assistants can personalize responses based on user preferences and historical interactions. They also leverage context from previous queries to provide more accurate and relevant responses over time.

**7. Challenges and Future Directions**

**Robustness to Variability**: Voice recognition systems must be robust to variations in speech patterns, accents, background noise, and environmental conditions. Continued research into data augmentation, robust training techniques, and multi-modal fusion will help improve performance.

**Privacy and Security**: As voice-enabled assistants become more prevalent, ensuring user privacy and data security is paramount. Techniques like federated learning and on-device processing can mitigate privacy concerns by minimizing data sharing and exposure.

**Multimodal Integration**: Integrating voice recognition with other modalities like vision and touch can enhance user experiences and enable new applications. Future research should explore multimodal fusion techniques and interaction paradigms for seamless integration.

By leveraging deep learning and advanced AI techniques, voice-enabled virtual assistants have the potential to revolutionize human-computer interaction and enhance productivity in various domains. Future research should focus on addressing challenges related to robustness, privacy, and multimodal integration to unlock the full potential of these systems.

This literature survey provides an overview of voice recognition methods, including traditional approaches, machine learning techniques, and recent advancements in deep learning. It also discusses applications in voice-enabled assistants and highlights challenges and future directions for research in the field.

**3. EXISTING SYSTEM**

The existing systems for intelligent assistants, such as Siri, Google Assistant, and Alexa, are built on artificial intelligence (AI) and machine learning (ML) technologies. These systems use natural language processing (NLP) algorithms to understand and respond to user commands and queries. They can perform various tasks like answering questions, providing information.

**4. PROPOSED SOLUTION**

In addition to the existing features mentioned above we're also adding extra feature by incorporating image input, the assistant will directly assist students in efficiently annotating objects within the image. This functionality differentiates our assistant from current offerings like Google Assistant, Siri, and Alexa, which primarily focus on web searches and task completion.

**4.1 SYSTEM DESIGN**

**4.1.1 ARCHITECTURE DIAGRAM**

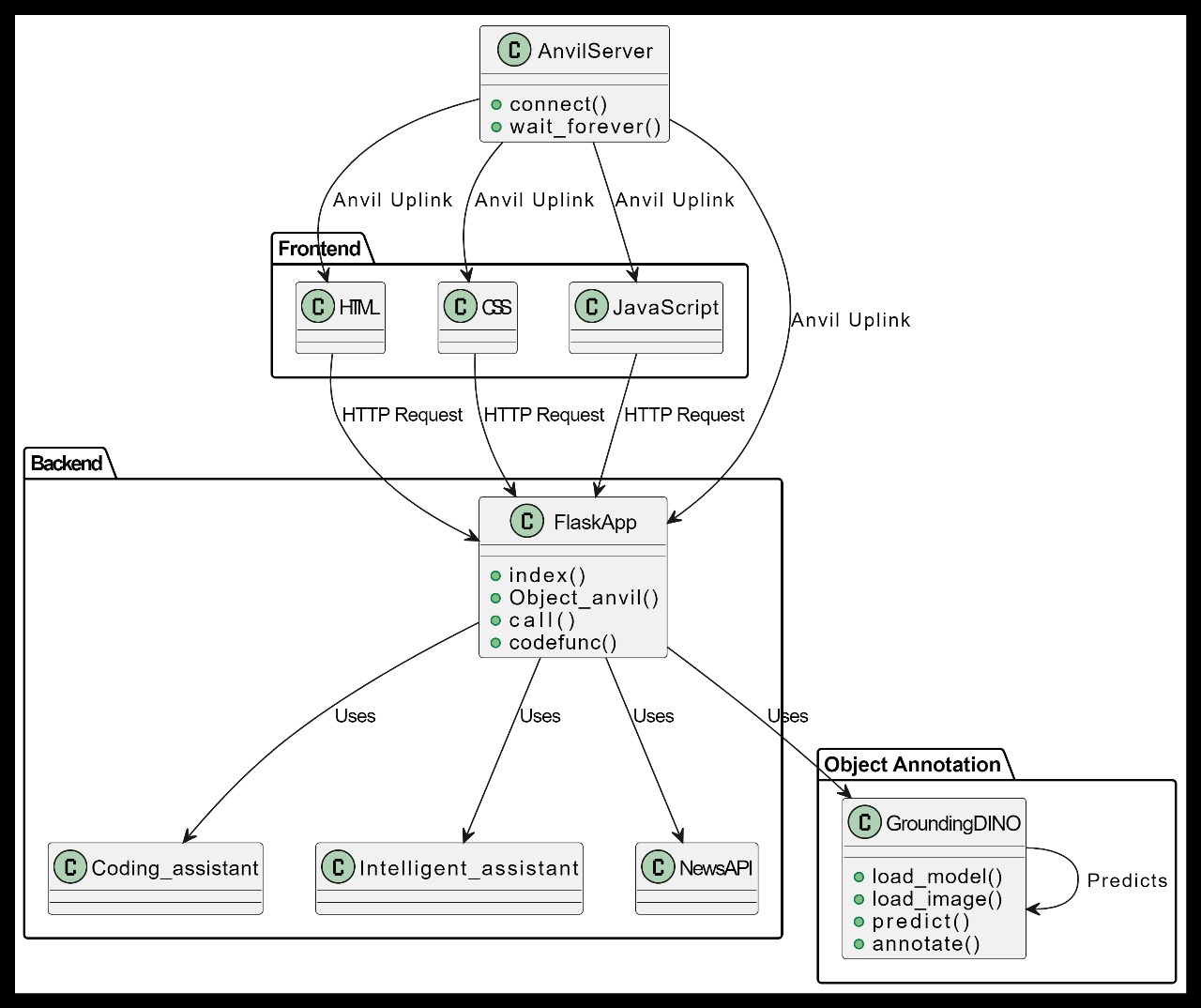


Fig-4.1.1: Architecture Diagram

**4.1.2 USE CASE DIAGRAM**

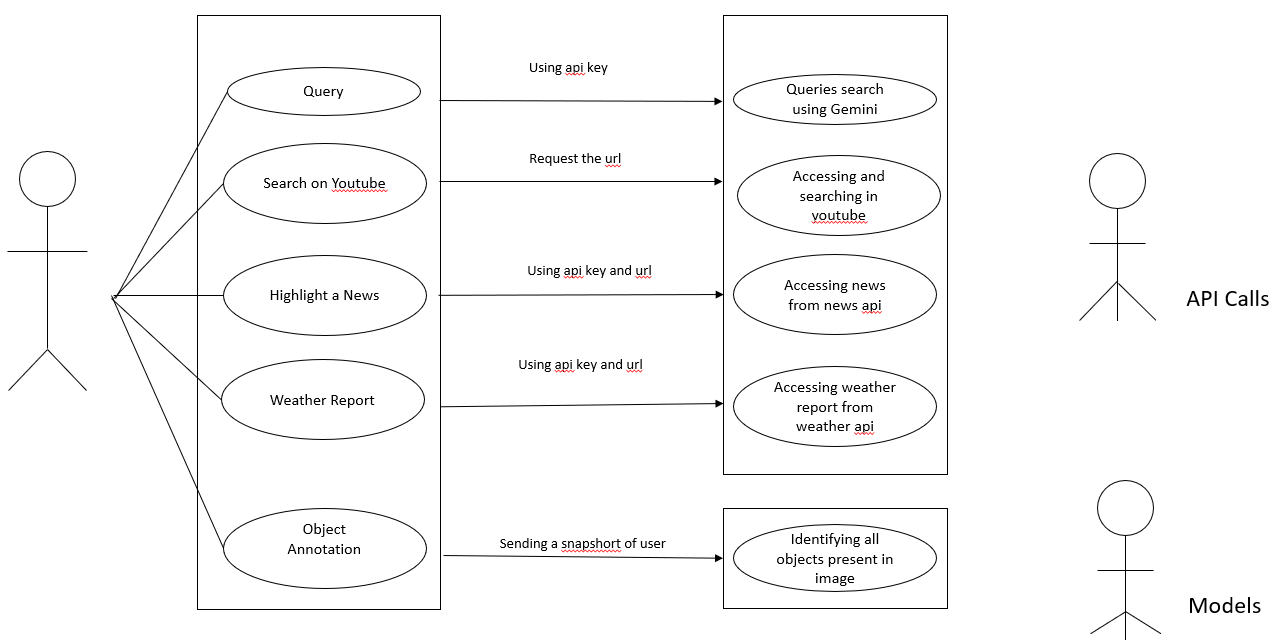
****

Fig-4.1.2:Use-Case Diagram

**4.1.2.1 USE-CASE DESCRIPTION**

**Use Case:** Object Identification

**Primary Actor**: User

**Preconditions**: The user has taken a snapshot of an image.

**Postconditions**: The system has identified all objects present in the image.

**Description**: The user sends a snapshot to the system, which then uses object identification algorithms and an API key to identify all objects present in the image. The results are displayed to the user.

**Use Case**: Weather Report

**Primary Actor**: User

**Preconditions**: The user has requested a weather report.

**Postconditions**: The system has provided the current weather report for a specified location.

**Description**: The user requests a weather report by providing a location using an API key. The system then retrieves and displays the current weather report for that location.

**Use Case**: Highlight a News

**Primary Actor**: User

**Preconditions**: The user has requested news on a specific topic.

**Postconditions**: The system has displayed news articles related to the specified topic.

**Description**: The user requests news on a specific topic by providing a keyword using an API key. The system then retrieves and displays news articles related to that keyword.

**Use Case**: Search on Youtube

**Primary Actor**: User

**Preconditions**: The user has requested to search for a specific video or channel on Youtube.

**Postconditions**: The system has displayed the search results for the specified query.

**Description**: The user requests to search for a specific video or channel on Youtube by providing a query using an API key. The system then retrieves and displays the search results for that query.

These are just a few examples of use case descriptions based on the given use case diagram. The actual descriptions may vary depending on the specific requirements and functionality of the system.

**4.2 FUNCTIONAL MODULES**

**4.2.1 SCREENSHOTS & PSEUDOCODE**

**Index.html :**

This HTML code sets up a webpage for a voice-activated assistant named Jarvis. Key components include:

**Head Section** :

Meta tags, page title, and links to Bootstrap, icons, and custom stylesheets.

**Body Section:**

**A main container with two sections** :

"Oval" Section: Contains a canvas, prompt text, input field, and buttons for sending text, voice input, opening chat, and settings.

An off-canvas sidebar for chat history.

**Scripts**:

External libraries (jQuery, Bootstrap JS).

Custom scripts for additional functionalities.

An inline script for handling voice input button clicks with an AJAX request.

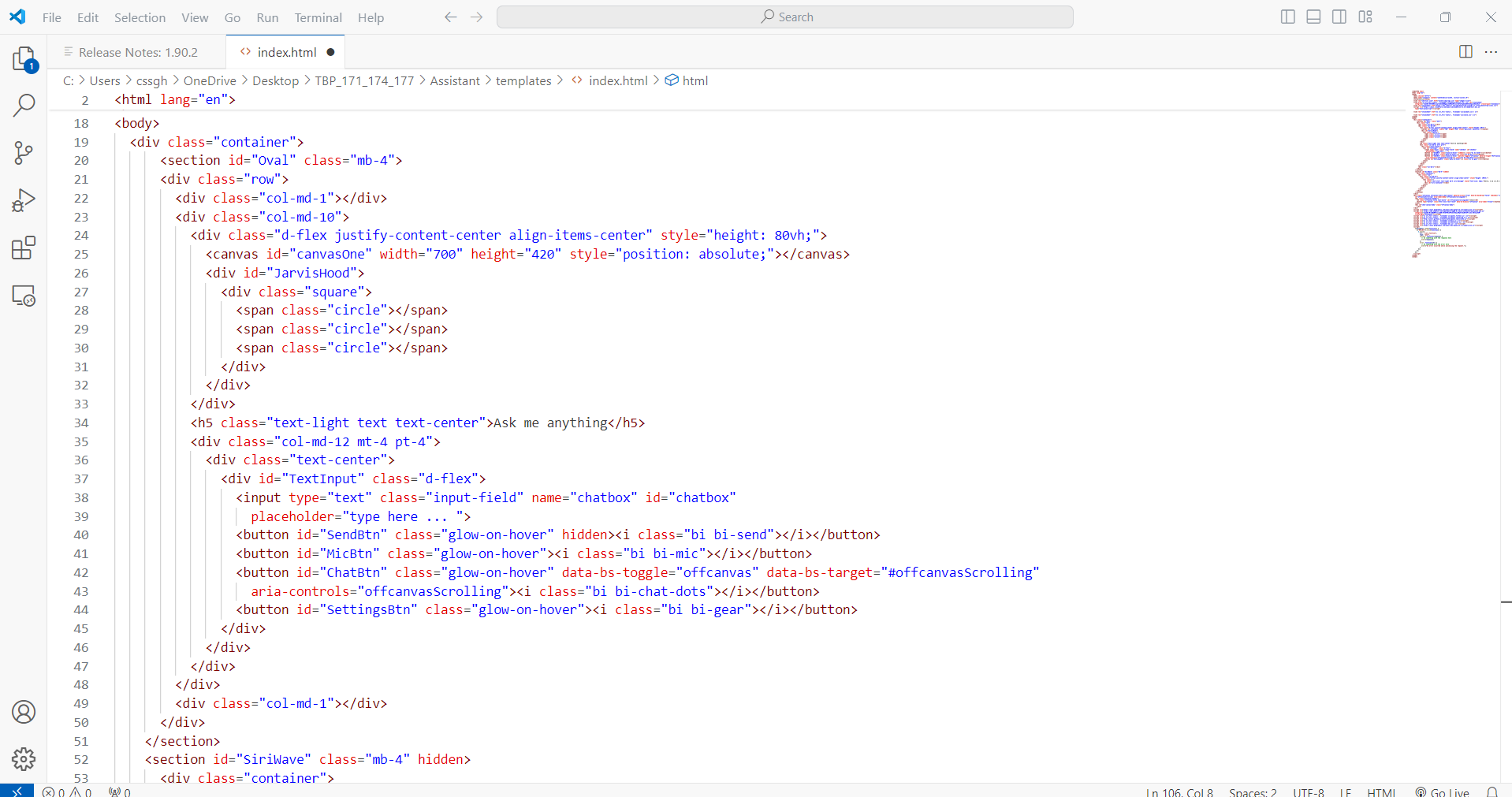
The page is styled with Bootstrap and custom CSS, aiming to provide a responsive, interactive assistant interface.****

Fig-4.2.a Index.html Code

**App.py:**

This Flask application sets up a web server with several routes and functionalities, including video streaming from a webcam and handling text messages.

**Imports:**

Flask modules for web server functionality.

OpenCV (cv2) for video capture.

Custom functions callgem and gemin.

**Camera Setup:**

Initializes webcam capture.

Video Streaming:

generate\_frames(): Reads frames from the webcam, encodes them as JPEG, and yields them in a streaming format.

**Route /video**: Streams video to the client.

**Routes:**

Renders the main HTML page (index.html).

video: Provides the video stream.

call: Calls the callgem function and returns its response.

send\_message: Receives a POST request with a message, processes it using codefunc, and returns the response.

**Backend Function:**

codefunc(message): Processes the message using gemin and returns the result.

**Server Run:**

Starts the Flask server in debug mode.

This app provides a live video stream, handles text-based requests, and processes them with custom backend logic.

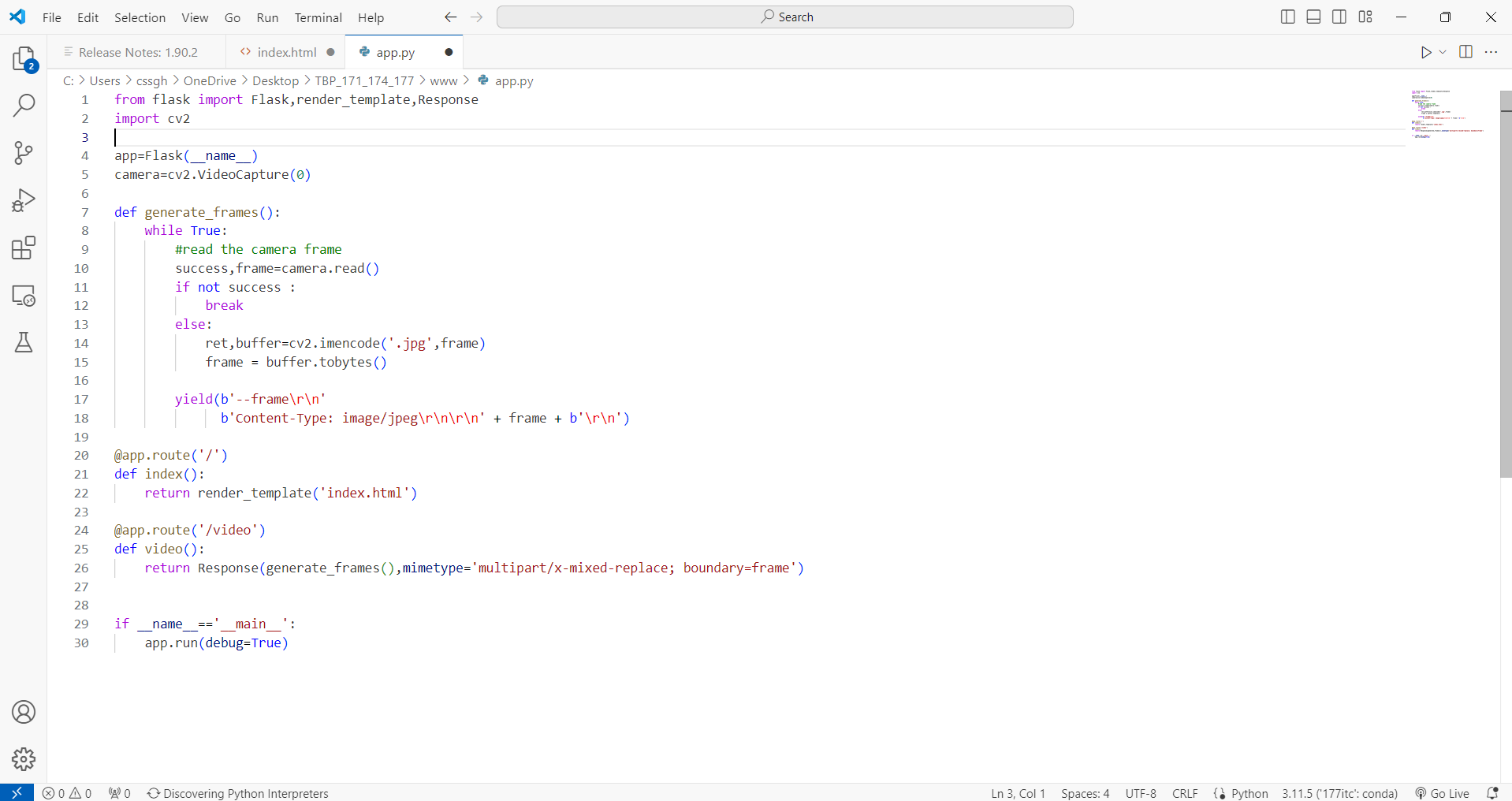
****

Fig-4.2.b Object Annotation Code

**5. EXPERIMENTAL SETUP & IMPLEMENTATAION**

**5.1 SYSTEM SPECIFICATIONS**

**5.1.1 HARDWARE REQUIREMENTS**

* Computer with a decent processor (preferably with a GPU if you are dealing with image processing and deep learning models).
* Microphone for capturing voice commands.
* Speakers or headphones for audio output.

**5.1.2 SOFTWARE REQUIREMENTS**

* Operating System: Windows, macOS, or Linux.
* Python: Version 3.7 or above.
* IDE/Code Editor: VSCode, Jupyter Notebook, etc.
* Web Browser: Latest version of Chrome, Firefox, etc.
* **Python Packages and Libraries:**
* Flask: For creating the web server.
* SpeechRecognition: For converting speech to text.
* Pyttsx3: For converting text to speech.
* Requests: For interacting with external APIs.
* Google Generative AI Library: For using the Gemini model for generating responses.
* GroundingDINO: For object detection and annotation.
* Anvil-Uplink: For connecting with Anvil server.
* Supervision: For processing and annotating images.
* **API Keys:**
* Google API Key: For accessing Google services, including the Gemini model.
* News API Key: For fetching news headlines.

**5.2 DATASET**

* COCO

**5.3 METHODOLOGY/ALGORITHM**

1. **Load the Model**: Load the pre-trained GroundingDINO model along with its configuration.

2. **Load the Image**: Load the image for object annotation.

3. **Preprocess the Image**: Preprocess the image to prepare it for input into the model. This may involve resizing, normalization, or other preprocessing steps specified by the model requirements.

4. **Predict Objects**: Use the loaded model to predict objects within the image. This step involves passing the preprocessed image through the model and obtaining predictions for objects present in the image.

5. **Annotate Objects**: For each predicted object, annotate the image with bounding box coordinates and a caption describing the object. This step involves interpreting the model predictions and generating annotations that can be overlaid onto the image.

6. **Return Annotated Image**: Return the annotated image with bounding boxes and captions indicating the objects detected by the model.

This algorithm outlines the high-level steps involved in object annotation using the GroundingDINO model. Actual implementation details may vary depending on the specific requirements of your project and the capabilities of the GroundingDINO model.

**6. RESULTS**

**Intelligent Assistant:**

* This is the main interface of our project in which there are three buttons which are used to redirect to different features of our project.
* The assistant feature is represented by voice icon or button.
* The other button indicates chatbot which is used for coding problem.
* The other button is used to directly go to the interface of our object annotator.

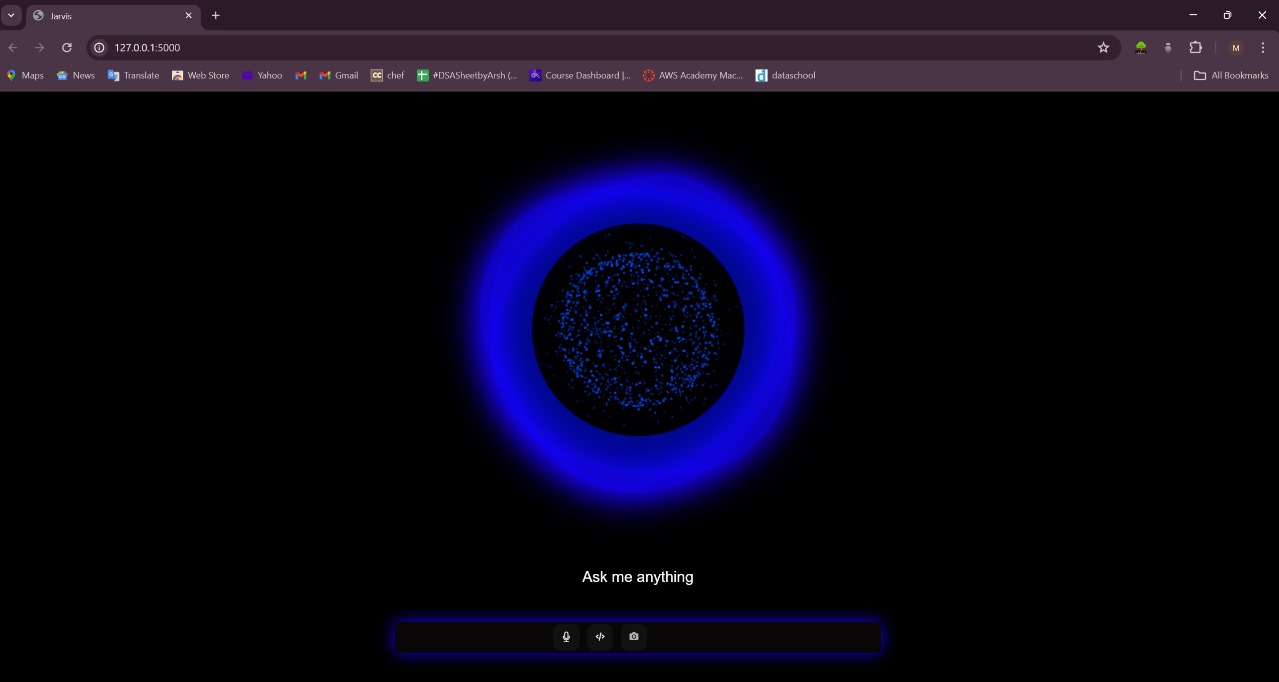
****

Fig-6.1 : Intelligent Assistant

**Code Assistant:**

* This is the interface when we open the chatbot.
* We can give any coding related doubts.
* The doubts given by the user are resolved and the output is given.

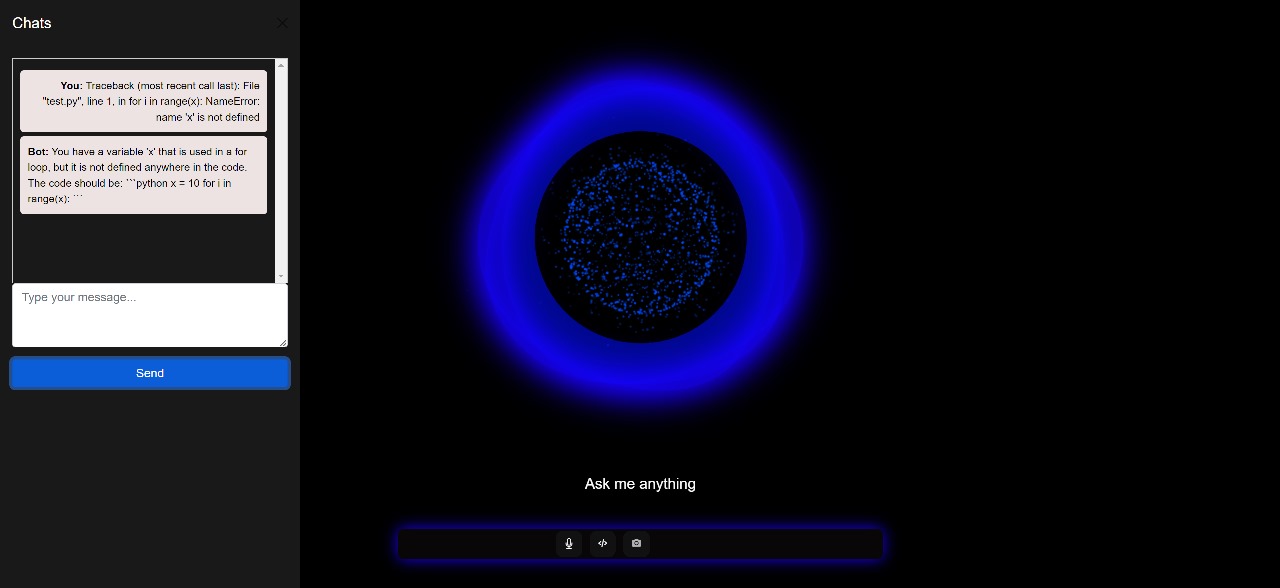


Fig-6.2 : Code Assistant

**Object Annotation Web Page:**

* This is the interface for object annotator.
* This is used annotate the object by uploading images links and their extensions.
* We also give the name of the object to be annotated.

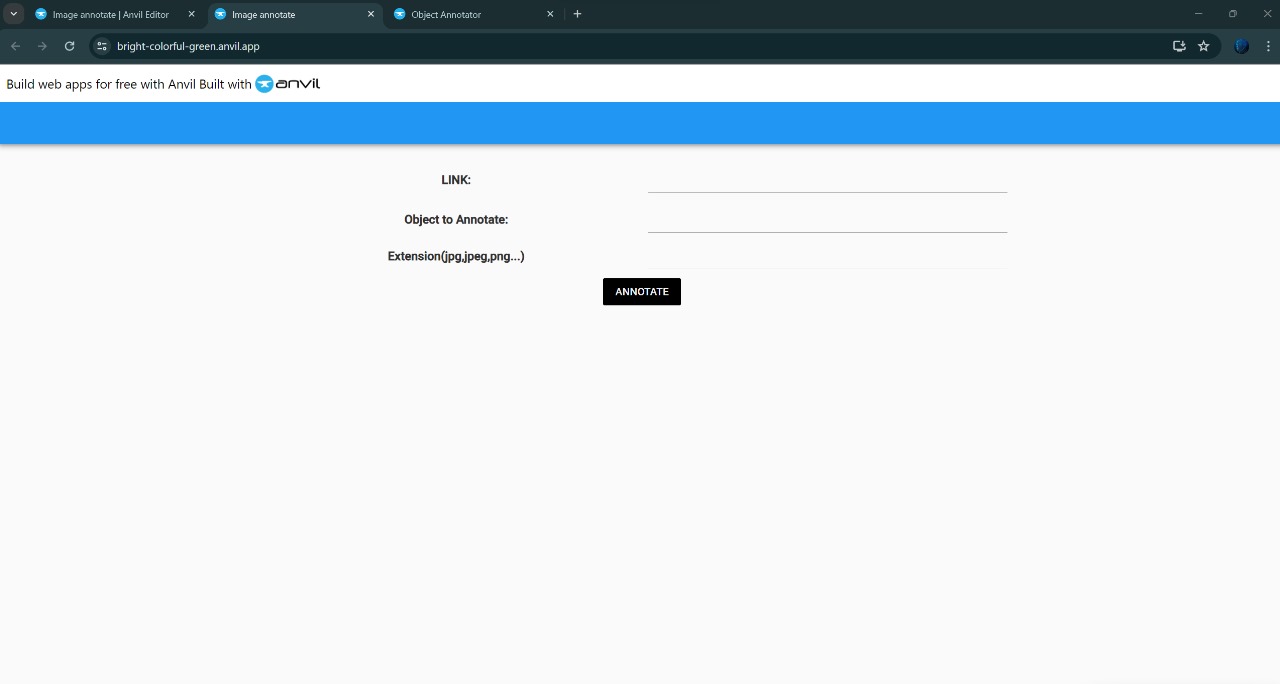
****

Fig-6.3 : Object Annotation Web Page

**Object Annotation Result:**

* This is the interface for object annotator result.
* The results of the annotated image are shown.
* We have uploaded the below picture.
* We also gave the name of the object that has to be annotated.
* In the below picture we are annotating the surfer board.
* In this result we can also locate the accuracy of the annotated object.

****

Fig-6.4 : Object annotation Result

**7. CONCLUSION & FUTURE SCOPE**

This project has successfully developed a novel intelligent assistant that bridges the gap between image understanding and student learning. By leveraging the Gemini API, custom search functionalities, and image processing techniques, the assistant empowers students to efficiently annotate objects within images based on prompts. This innovative approach sets it apart from existing virtual assistants like Google Assistant, Siri, and Alexa, which primarily focus on web searches and task completion.

**Advanced Image Recognition**: Move beyond basic object annotation. Integrate object property recognition (size, color, material) and scene understanding (actions, relationships between objects).

**Multimodal Learning**: Allow users to combine image input with voice prompts or text descriptions for richer annotations.

**Subject-Specific Annotations**: Tailor the annotation experience to different subjects (science, history, literature) by offering pre-defined annotation labels or integrating with educational databases.

**8. REFERENCES**

1. **Machine Learning and Object Detection:**

* Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
* Ren, S., He, K., Girshick, R., & Sun, J. (2015). "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In Advances in neural information processing systems (pp. 91-99).

[https://github.com/IDEA-Research/GroundingDINO](https://github.com/IDEA-Research/GroundingDINO](#_8._REFERENCES)

2. **Anvil Server:**

[https://anvil.works/docs](https://anvil.works/docs](https://anvil.works/docs%5d(https://anvil.works/docs)

3.**Google Generative AI:**

<https://developers.google.com/generative-ai>

4.**Object Annotation Tools:**

[https://pypi.org/project/supervision/](#_8._REFERENCES)

5.**General References on AI and ML:**

* Russell, S. J., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach. Pearson.
* Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.