## ABSTRACT

The Personal Virtual Assistant AI Chatbots and AI Text-to-Image Generator is an advanced system designed to provide users with intelligent assistance through voice and text-based automation. The project features three primary components: a Virtual Chatbot, a Personal Chatbot, and an AI Image Generator, each offering unique functionalities. The Virtual Chatbot listens to user voice commands and responds either by retrieving predefined answers or conducting web searches when no saved command matches. The Manage Commands Page enables users to store commands with corresponding URLs or responses for quick execution. The Personal Chatbot Page is divided into two sections: one that answers user queries and recognizes images using AI-driven models, and another that fetches real-time news, weather updates, and relevant images from online sources. Additionally, the AI Image Generator transforms text prompts into AI-generated images using advanced text-to-image conversion techniques. The system maintains strict security through a Login, Signup, Forgot Password, Reset Password, and Logout mechanism, ensuring single-user access per device. Features like voice input for hands-free interaction, security verification via reset tokens, and an eye icon for password visibility enhance user experience. The integration of speech recognition, AI automation, and real-time data processing makes this project a highly interactive and efficient virtual assistant system that simplifies user tasks, automates responses, and generates images based on textual descriptions.

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **CHAPTER NAME** | **PAGE.NO** |
| **LIST OF FIGURES** |  | **i** |
| **LIST OF FLOWCHARTS** |  | **ii** |
| **LIST OF TABLES** |  | **iii** |
| **LIST OF ABBREVIATIONS** |  | **iv** |
| **CHAPTER – 1** | **INTRODUCTION** | **1** |
| 1.1 | Introduction | 1 |
| 1.2 | Project Objectives | 1 |
| **CHAPTER – 2** | **SYSTEM ANALYSIS** | **2** |
| 2.1 | Existing System | 2 |
|  | 2.1.1 Disadvantages | 2 |
| 2.2 | Proposed System | 2 |
|  | 2.2.1 Advantages | 3 |
| **CHAPTER – 3** | **SYSTEM REQUIREMENTS AND SPECIFICATIONS** | **4** |
| 3.1 | Hardware Requirements | 4 |
| 3.2 | Software Requirements | 4 |
| 3.3 | Software Description | 4 |
|  | 3.3.1 HTML (Hyper Text Markup Language) | 4 |
|  | 3.3.2 CSS (Cascading Style Sheets) | 4 |
|  | 3.3.3 JavaScript | 4 |
|  | 3.3.4 Why these Technologies were used in this project | 4 |
| 3.4 | Feasibility Study | 5 |
|  | 3.4.1 Technical Feasibility | 5 |
|  | 3.4.2 Operational Feasibility | 5 |
|  | 3.4.3 Economical Feasibility | 5 |
| **CHAPTER – 4** | **SYSTEM DESIGN** | **6** |
| 4.1 | System Architecture | 6 |

|  |  |  |
| --- | --- | --- |
|  | 4.1.1 Explaination of above Flow Chart | 6 |
|  | 4.1.1.1 User Authentication | 6 |
|  | 4.1.1.2 Virtual Chatbot System | 6 |
|  | 4.1.1.3 Command Management System | 6 |
|  | 4.1.1.4 AI Image Generation and Information Retrieval | 6 |
|  | 4.1.1.5 Logout Process | 6 |
| 4.2 | Modules Description | 7 |
|  | 4.2.1 Modules | 7 |
|  | 4.2.2 Voice Activated Chatbot Module | 7 |
|  | 4.2.3 Personal Text and Image-Based Chatbot Module | 7 |
|  | 4.2.4 AI Image Generator Module | 7 |
|  | 4.2.5 Reset Password Module | 7 |
|  | 4.2.6 Navigation and Accessibility Module | 7 |
| **CHAPTER – 5** | **WORKING PROCESS** | **8** |
| 5.1 | Definition | 8 |
| 5.2 | Signup Process | 8 |
| 5.3 | Login Process | 8 |
| 5.4 | Logout Process | 9 |
| 5.5 | Speak Buttons | 9 |
| 5.6 | Microphone Buttons | 9 |
| 5.7 | Toggle-Eye Icons | 10 |
| 5.8 | Three Models of Working Process | 10 |
|  | 5.8.1 Model 1 Voice Activated Chatbot | 10 |
|  | 5.8.2 Model 2 Personal Chatbot (Both Voice and Chat – Based) | 11 |
|  | 5.8.3 Model 3 Text-To-AI-Image Generator | 12 |
| 5.9 | Overview of Working Process | 13 |
| **CHAPTER – 6** | **SYSTEM CODING AND IMPLEMENTATION** | **14** |
| 6.1 | Source Code | 14 |
|  | 6.1.1 index.html | 14 |
|  | 6.1.2 styles.css | 14 |

|  |  |  |
| --- | --- | --- |
|  | 6.1.3 script.js | 14 |
| 6.2 | System Testing | 14 |
|  | 6.2.1 Testing Methodologies | 14 |
|  | 6.2.1.1 Unit Testing | 14 |
|  | 6.2.1.2 Integration Testing | 14 |
|  | 6.2.1.3 User Acceptance Testing | 14 |
|  | 6.2.2 Testing Strategy | 14 |
|  | 6.2.3 Types of Tests | 14 |
| **CHAPTER 7** | **EXECUTION STEPS** | **16** |
| 7.1 | Installation of Softwares | 16 |
|  | 7.1.1 Installation of Visual Studio Code | 16 |
|  | 7.1.1.1 Configuring the Environment for Development | 16 |
|  | 7.1.1.2 How to get Visual Studio Code (the Code Editor) | 16 |
|  | 7.1.1.3 How to Configure the Code Editor | 17 |
| 7.2 | Running a Web Browser to manage the project | 17 |
|  | 7.2.1 API Registration and Integration Procedure | 17 |
|  | 7.2.1.1 Procedure for Getting API Keys | 17 |
|  | 7.2.1.2 JavaScript API Integration | 17 |
| 7.3 | Debugging and Testing | 18 |
|  | 7.3.1 Voice Input Testing | 18 |
|  | 7.3.2 Testing Chatbot Responses | 18 |
|  | 7.3.3 Testing AI Image Generation | 18 |
|  | 7.3.4 Navigation Testing | 18 |
|  | 7.3.5 Using Developer Tools for Debugging | 18 |
| **CHAPTER – 8** | **RESULTS** | **19** |
| 8.1 | Login, Signup and Forgot Password Page | 19 |
|  | 8.1.1 Opening Screen of Project | 19 |
|  | 8.1.2 Forgot Password | 19 |
| 8.2 | Model-1 Voice Activated Chatbot | 20 |
| 8.3 | Model-2 Personal Chatbot | 21 |

|  |  |  |
| --- | --- | --- |
| 8.4 | Model-3 Text-To-AI-Image Generator | 21 |
| **CHAPTER – 9** | **CONCLUSION** | **22** |
| **CHAPTER – 10** | **FUTURE ENHANCEMENTS** | **23** |
| **CHAPTER – 11** | **BIBLIOGRAPHY** | **24** |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **FIGURE NAME** | **FIG.NO** | **PAGE.NO** |
| 1. | Data Flow Diagram of Signup Process | Fig 5.2.1 | 8 |
| 2. | Data Flow Diagram of Login Process | Fig 5.3.1 | 9 |
| 3. | Data Flow Diagram of Forgot Password Process | Fig 5.3.2 | 9 |
| 4. | Data Flow Diagram of Reset - PasswordProcess | Fig 5.3.3 | 9 |
| 5. | Data Flow Diagram of Logout Process | Fig 5.4.1 | 9 |
| 6. | Opening Screen of Voice Activated Chatbots | Fig 5.8.1 | 10 |
| 7. | Manage Commands page | Fig 5.8.2 | 11 |
| 8. | Example For Addition of Commands in Manage Commands Page | Fig 5.8.3 | 11 |
| 9. | Example of Personal Chatbot Page | Fig 5.8.4 | 12 |
| 10. | Current Info and Image Generation Page | Fig 5.8.5 | 12 |
| 11. | Text -to -AI Image Generator Page | Fig 5.8.6 | 13 |
| 12. | Login Page | Fig 8.1.1.1 | 19 |
| 13. | Signup Page | Fig 8.1.1.2 | 19 |
| 14. | Forgot Password Page | Fig 8.1.2.1 | 19 |
| 15. | Reset Token | Fig 8.1.2.2 | 20 |
| 16. | Password Reset Page | Fig 8.1.2.3 | 20 |
| 17. | Voice Activated Chatbot Page | Fig 8.2.1 | 20 |
| 18. | When Command is giving through your voice | Fig 8.2.2 | 20 |
| 19. | Manage Commands Page | Fig 8.2.3 | 20 |
| 20. | Adding Commands | Fig 8.2.4 | 20 |
| 21. | Example of Personal Chatbot Page | Fig 8.3.1 | 21 |
| 22. | Example of Image Recognition | Fig 8.3.2 | 21 |
| 23. | Example of Weather Retrieving | Fig 8.3.3 | 21 |
| 24. | Example of Image Generation | Fig 8.3.4 | 21 |
| 25. | Text-To-AI-Image-Generator Page | Fig 8.4.1 | 21 |
| 26. | Example of AI Image Generation | Fig 8.4.2 | 21 |

**i**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **FLOWCHART NAME** | **FLOWCHART.NO** | **PAGE.NO** |
| 1. | Based on the overview of System Design | Flowchart 4.1 | 6 |
| 2. | Based on the overview of Module Design | Flowchart 4.2 | 7 |

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **TABLE NAME** | **TABLE.NO** | **PAGE.NO** |
| 1. | Added Commands in the Database | Table 5.8.1 | 11 |

|  |  |
| --- | --- |
| AI | Artificial Intelligence |
| API | Application Programming Interface |
| CSS | Cascading Style Sheets |
| GUI | Graphical User Interface |
| HTML | Hyper Text Markup Language |
| ID | Identifier |
| JS | JavaScript |
| JSON | JavaScript Object Notation |
| MDN | Mozilla Developer Network |
| NLP | Natural Language Processing |
| OTP | One Time Password |
| SDK | Software Development Kit |
| UAT | User Acceptance Testing |
| URL | Uniform Resource Locator |
| VS CODE | Visual Studio Code |
| W3C | World Wide Web Consortium |

**iv**

# CHAPTER – 1 INTRODUCTION

## INTRODUCTION

### INTRODUCTION

In the modern digital era, artificial intelligence (AI) has revolutionized the way humans interact with technology. Virtual assistants and AI-powered chatbots have become essential tools in simplifying tasks, automating responses, and enhancing user experiences. The Personal Virtual Assistant AI Chatbot and AI Text-to-Image Generator is a sophisticated system designed to provide users with a seamless and intelligent assistance platform that combines voice recognition, natural language processing (NLP), and AI-based image generation.

This project comprises three main components: Virtual Chatbot, Personal Chatbot, and AI Text-to-Image Generator. The Virtual Chatbot Page allows users to give voice commands, which the system either processes based on saved responses or searches on the web if no predefined command is available. The Manage Commands Page enables users to store, edit, and delete commands along with corresponding URLs or answers for quick retrieval. The Personal Chatbot Page is divided into two sections: one that handles general user queries and image recognition, and another that provides real-time information such as news, weather, and relevant images sourced from online platforms. The AI Image Generator translates user text prompts into AI-generated images, leveraging advanced text-to-image transformation technologies.

The system ensures secure access by implementing a Signup, Login, Forgot Password, Reset Password, and Logout mechanism with a single-user limitation per device. Security features like reset tokens, security questions, and password visibility toggles are integrated to safeguard user data. Additionally, voice input features enhance accessibility, allowing users to interact with the chatbot hands-free. The project incorporates AI-based APIs for natural language understanding, real-time data retrieval, and image generation, making it a comprehensive virtual assistant solution.

By integrating AI-powered automation with voice and text interaction, this project offers a highly interactive and efficient platform. It caters to diverse user needs, from handling queries and automating responses to generating AI-driven images, making it a versatile and valuable tool for personal and professional use.

### PROJECT OBJECTIVES

The objective of this project is to develop an AI-powered virtual assistant that can process voice and text-based queries, retrieve real-time information, automate responses, and generate AI-based images from user-provided text prompts. It aims to enhance user convenience through intelligent automation, real-time data fetching, and interactive chatbot functionalities.

# CHAPTER – 2 SYSTEM ANALYSIS

## SYSTEM ANALYSIS

#### Existing System

The existing systems for virtual assistants and AI-driven chatbots often rely on predefined responses, limited integration capabilities, and require manual user interactions. Many traditional AI chatbots can only handle simple text-based queries and lack the ability to process voice inputs, recognize images, or provide real-time updates. Additionally, most existing systems do not allow users to customize commands or attach URLs and answers, making them less flexible for personalized user experiences.

In conventional AI chatbot applications, users typically interact through text input only, limiting accessibility for those who prefer voice commands. Furthermore, the inability to store and manage customized commands means that users must repeatedly input the same information instead of automating frequent actions. Existing chatbots also struggle with retrieving real-time data, such as weather updates, news articles, or generating AI-based images. They often depend on static databases rather than integrating with external sources to fetch the latest information.

Another major drawback is the absence of a centralized system that combines multiple AI functions such as conversational AI, image recognition, and text-to-image generation. Current systems often require users to switch between multiple applications to perform different AI-related tasks, which reduces efficiency and user convenience.

###### Disadvantages of the Existing System

* Limited to text-based interactions, making it less user-friendly.
* Lack of personalized command management and automation features.
* Users cannot attach URLs or answers to specific commands.
* Does not have different user- friendly features for sp.
* Requires switching between multiple applications for different AI functionalities.

#### Proposed System

The proposed system introduces a Personal Virtual Assistant AI Chatbot integrated with AI-based image generation capabilities. This system allows users to interact with a virtual assistant using both predefined and custom commands. Unlike existing chatbots, users can add, edit, and manage their commands, associating them with specific responses or URLs for direct navigation to relevant platforms. The chatbot also includes an image upload feature, enabling AI-driven image recognition to provide relevant information about the uploaded content.

A key enhancement in the proposed system is its ability to fetch real-time information, including news updates, weather reports, and general knowledge from reliable sources. This eliminates the need for users to search externally, improving efficiency and convenience. Additionally, the chatbot incorporates a secure login system with authentication protocols to enhance user data privacy and security.

The AI-powered text-to-image generation feature allows users to create AI-based images based on their textual descriptions. This capability is seamlessly integrated with the chatbot, ensuring a smooth user experience without switching between different applications. The entire system is designed to be user-friendly, with voice input options and an intuitive interface that simplifies interactions.

###### Advantages of the Proposed System

* Supports both text and voice-based interactions for better accessibility.
* Allows users to store and manage personalized commands with attached URLs or responses.
* Retrieves real-time news, weather updates, and image-based content.
* Integrates an AI-powered text-to-image generator for creative content generation.
* Includes image recognition capabilities for identifying and processing user-uploaded images.
* Secure login, signup, and password recovery mechanisms to protect user data.

# CHAPTER – 3

**SYSTEM REQUIREMENTS & SPECIFICATIONS**

## SYSTEM REQUIREMENTS AND SPECIFICATIONS

### HARDWARE REQUIREMENTS

* **Processor:** Intel Core i3 or higher / AMD equivalent
* **RAM:** Minimum 4GB (8GB recommended for smooth development)
* **Storage:** At least 20GB of free space for project files and dependencies
* **Display:** Minimum resolution of 1366×768 pixels
* **Internet Connection:** Required for API integration and testing
* **Input Devices:** Keyboard, Mouse, and Microphone (for voice input features)

### SOFTWARE REQUIREMENTS

* **Operating System:** Windows, macOS, or Linux
* **Development Tools:** Visual Studio Code, Sublime Text, or any code editor
* **Programming Languages:** HTML, CSS, JavaScript
* **APIs**: Certain AI-related, Google-related APIs for chatbot functionalities, and image generation APIs
* **Browser:** Google Chrome, Mozilla Firefox, or any modern web browser

### SOFTWARE DESCRIPTION

HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), and JavaScript are the core technologies used to develop and design web applications. Each of these technologies plays a crucial role in the structure, presentation, and functionality of a website or web-based project.

* + 1. **HTML (Hyper Text Markup Language):** HTML is the standard markup language used to create and structure web pages. It provides the basic framework of a webpage by defining elements such as headings, paragraphs, links, buttons, forms, and other content. HTML is essential for ensuring that the webpage is readable and properly structured.
		2. **CSS (Cascading Style Sheets):** CSS is used to control the visual appearance and layout of web pages. It allows developers to apply styles, such as colors, fonts, spacing, and positioning, to HTML elements. CSS enhances the aesthetic appeal of the project and ensures that the user interface is visually appealing and responsive across different devices.
		3. **JavaScript:** JavaScript is a programming language that enables interactivity and dynamic functionality within a webpage. It allows users to interact with the application through actions like clicking buttons, entering text, and navigating between different pages. JavaScript plays a significant role in event handling, animations, and real-time data updates.
		4. **Why These Technologies Were Used in the Project:** This project was developed using HTML, CSS, and JavaScript because these technologies provide an efficient and flexible way to build a fully functional web-based AI-powered chatbot and image generator. HTML ensures a well-structured layout, CSS enhances the visual appeal and responsiveness, while JavaScript facilitates interactive elements such as voice recognition, command processing, and API integration. The combination of these three technologies ensures a seamless and engaging user experience, making the system both functional and visually appealing.

### FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. Feasibility study of this proposed system is to be carried out to make for efficiency.

This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements is essential. Three key considerations involved in the feasibility analysis are

1. Technical Feasibility
2. Operational Feasibility
3. Economic Feasibility
	* 1. **Technical Feasibility:** The project is technically feasible as it utilizes widely supported and well- documented technologies - HTML, CSS, and JavaScript - which are the core technologies for web development. These technologies ensure the system runs efficiently on modern web browsers without requiring additional software installations. The project also integrates various AI-related APIs for chatbot functionalities, information retrieval, and AI image generation, which enhances its capabilities. Since the APIs used are well-established and scalable, there are minimal technical risks. Furthermore, web-based deployment ensures compatibility across different operating systems and devices.
		2. **Operational Feasibility:** The system is designed for easy usability with intuitive navigation, making it accessible to users with minimal technical knowledge. Features like voice input, eye icons for password visibility, and a chatbot interface enhance the user experience. The project structure ensures smooth operation by organizing functionalities into well-defined sections such as Personal Chatbot Page, Virtual Chatbot Page, and Text-to-AI Image Generator Page. Since the application is limited to a single user per device, security and personalized usage are maintained effectively. The integration of automated responses and AI-powered content generation further ensures efficiency in handling user queries.
		3. **Economic Feasibility:** The project is cost-effective as it is built using open-source technologies (HTML, CSS, JavaScript), eliminating the need for expensive development tools. Hosting the application on a cloud-based platform or a local server reduces infrastructure costs. APIs used in the project may have free-tier options, minimizing operational expenses. The return on investment (ROI) is high due to the potential usability of the chatbot for various applications, such as personal assistance, AI-based search automation, and image generation, making the system valuable without significant recurring costs.

The feasibility study confirms that the project is viable from technical, operational, and economic perspectives. The technology stack ensures smooth development and scalability, while the operational structure guarantees user- friendly interactions. Additionally, cost-effective implementation ensures long-term sustainability. With proper testing and deployment, the system can effectively serve its intended purpose with minimal risks.

# CHAPTER – 4 SYSTEM DESIGN

## SYSTEM DESIGN

### SYSTEM ARCHITECTURE

****

###### Explanation Of Above Flow chart

* + - 1. User Authentication Flow:
				* The system starts with a Login Page where users can enter their credentials.
				* If no user is registered, they are directed to the Sign-Up Page.
				* If users forget their password, they can navigate to the Forgot Password Page and reset it through the Reset Password Page.
			2. Virtual Chatbot System:
				* After logging in, users can access the Virtual Chatbot Page, which serves as the main interface for AI interactions.
				* From here, users can either proceed to the Personal Chatbot Page or Manage Commands Page for additional functionalities.
			3. Command Management System:
				* Users can visit the Manage Commands Page if they want to add or view custom chatbot commands.
				* Commands may include predefined responses or actions like opening URLs.
			4. AI Image Generation and Information Retrieval:
				* If users require current information or images, they are directed to the Current Info and Images Generation Page.
				* Users can also explore the Text-to-AI Image Generator Page for AI-generated images.
			5. Logout Process:
				* Once the user is done, they can navigate to the Logout Page, which ensures a secure exit from the system.
				* The Logout page verifies credentials before logging the user out.

### MODULES DESCRIPTION

##### MODULES

1. User Authentication Module
2. Voice-Activated Chatbot Module
3. Personal Text and Image-Based Chatbot Module
4. AI Image Generator Module
5. Reset Password Module
6. Navigation and Accessibility Module



###### Explanation of above modules

* + - 1. **User Authentication Module:** Manages user sign-up, login, and logout functionalities. Ensures only one account per device can be registered. Provides a forgot password option for account recovery. Validates credentials and prevents unauthorized access. Displays alerts for incorrect login attempts or non-existent users. Enhances security by verifying reset tokens before password changes.
			2. **Voice-Activated Chatbot Module:** Enables users to interact using voice commands. Supports a "Start Listening" button for speech recognition. Matches voice input with saved commands in the system. If no match is found, it performs a Google search automatically. Allows users to manage and customize commands for efficiency. Integrates text-to-speech for chatbot responses.
			3. **Personal Text and Image-Based Chatbot Module:** Provides a personal AI chatbot for user queries. Uses Gemini API for text-based AI responses. Supports image analysis by allowing users to upload pictures. Retrieves relevant information from trusted sources. Provides limited but contextual responses based on user input. Works alongside the Current Info and Image Generation Module.
			4. **AI Image Generator Module:** Generates images from text prompts using Hugging Face API. Allows users to input a description and receive AI-generated images. Supports different styles and customization options. Provides an easy-to-use interface with voice input support. Ensures smooth navigation to other chatbot features. Allows users to copy, download, or use the generated images.
			5. **Reset Password Module:** Helps users recover access by resetting their passwords. Requires email and security question verification. Generates a reset token only if the correct details are provided. Users must enter a new password and confirm it before resetting. Redirects to the Login Page after a successful reset. Ensures security by preventing unauthorized password changes.
			6. **Navigation and Accessibility Module:** Provides a structured flow between different system pages. Supports voice commands and text input for accessibility. Includes "Back" and "Go To" options for seamless transitions. Implements eye icons for password visibility control. Ensures users can navigate between chatbot, image generation, and reset modules. Maintains user-friendly alerts and instructions for smooth interaction.

# CHAPTER – 5 WORKING PROCESS

## WORKING PROCESS

### DEFINITION:

The working process refers to a systematic sequence of steps followed to achieve a specific goal or complete a task efficiently. It involves planning, execution, monitoring, and refinement to ensure smooth operations and desired outcomes. A well-defined working process enhances productivity, minimizes errors, and streamlines workflow management.

The working process of this project involves user authentication, voice and text-based interactions, and AI- powered responses to generate text and images. It seamlessly integrates multiple modules to ensure smooth navigation, accurate outputs, and enhanced user experience.

In this project working process is divided into 3 models with secure login and logout process:

**Model 1**: Voice Activated Chatbot.

**Model 2:** Personal Chatbot (Both Voice and Chat-Based).

**Model 3:** AI Text-To-Image Generator.

Let’s start discussing secure signup, login and logout process before dealing with the above three models.

#### Signup Process:

Users can be asked across various fields such as username, email, password, confirm password, security – question and security – answer. Already registered users can login via login link which has been displayed on the signup page.



#### Login Process:

Users begin at the login page, registered users can log in using valid credentials, gaining access to the core functionalities. New users can sign up via signup link which has been displayed in the login page and then log in to proceed. If the password has been forgotten by the user, the login page also has the reset password link option if user clicks it then he will be directed to the forgot password page. In this page the user wants to enter the registered email which the user has given in signup process, if registered email entered by the user is correct and then the security – question automatically will be displayed which has given by the user in signup process and then ask the user to enter security – answer. If the user enter the security – answer correctly which has given by the user in the signup process, the ok option along with reset code will be sent which can be copied or noted by the user, and when the user click ok option, he will be directed to the reset password page and paste the copied token in the reset password field of the reset password page so that he can reset new password.

 



#### Logout Process:

Users can securely log out by clicking the logout button on the AI image generator page. The logout process ensures user data and session security. If the user data can be matched with login details, then he can be logged out . Otherwise, he cannot be logged out of the application.



For all the login, signup, logout processes we have speak buttons, microphone buttons, toggle-eye icons (for password fields and security-answer fields).

#### Speak Buttons:

When the user clicks speak button, it will asks the required fields by voice one by one sequentially. User must ensure that speak button asks required fields sequentially even if you skip to say the voice input for any field or breaks the process in middle, so user must speaks all fields with only voice input without skipping any field or breaking the process in middle when user click speak button.

#### Microphone Buttons:

When the user clicks microphone button user can speaks whichever field user wants. No sequential process will be there for microphone button. So user can have the facility to speak for only certain fields.

#### Toggle-Eye Icons:

We have the toggle-eye icons for password fields and security-answer fields. When the user clicks on toggle-eye icon he can view what he enters the input in the password fields and security-answer fields. When the user clicks again on the toggle-eye icon, it can hides the input entered by the user.

With the secure signup, login, and logout processes explained, we can now proceed to explore the core models of the project. Each model plays a vital role in enhancing functionality and improving user experience. Let's dive into the detailed discussion of these models.

#### Three Models of Working Process:

##### MODEL 1: VOICE ACTIVATED CHATBOT:

Real-time information retrieval and interactive chatbot features are also features of Model 2, the Personal Chatbot and Information Generator. The Personal Chatbot Page and the Current Info and Image Generation Page are the two main portions of this concept. They each have different functions while sharing some characteristics.

The Personal Chatbot Page has a chatbot UI that responds to different informational prompts and manages user inquiries via the Gemini API. Users can upload images using the Image Button because it also supports image recognition. When an image is sent, the chatbot recognizes it and gives pertinent details about it.

Directly beneath the Start Listening button is the Manage Commands button, which provides customization and command over the chatbot's responses. The Manage Commands Page is accessed by clicking on it, allowing the user to add, edit, or remove commands. Every command can be associated with an answer, a URL, or both. To keep all related commands in one location, multiple commands for the same URL or response are kept in a single row and separated by commas. The user may manage several entries more easily with this format.

The system prioritizes the URL when a command has both a URL and an answer assigned to it. This guarantees that the system will prioritize navigating to the designated URL above speaking the response if the command is detected.

To enable voice interaction with the modified settings, the user can go back to the main page and click the Start Listening button once they have configured their chosen commands in the Manage Commands Page. According to their preferences, visitors can effortlessly transition between personalized website navigation and responses thanks to this flow.

Although voice contact is emphasized a lot in this approach, microphone functionality is also important. The microphone features of the Model 1 are incorporated into the Start Listening button itself, supporting the voice- first design of this model in contrast to the specialized microphone buttons seen in other project sections (such as Login, Signup, etc.).

For smooth navigation, this model also features two buttons in the upper right corner. Users may quickly access the project's many features by clicking on one button, which takes them to the AI Image Generator Page, and another, which takes them to the Personal Chatbot Page (Chat-Based). Users may easily choose between models because to its user-friendly design, which improves the user experience overall.

 

|  |  |  |
| --- | --- | --- |
| **Commands** | **URL** | **Answer** |
| Open YouTube, Edit, Delete, SaveYouTube Open Edit, Delete, Save | https[://www.youtube.com/](http://www.youtube.com/) Edit, Delete, Save | Edit, Delete, Save |

###### Table 5.8.1 Added database commands

Note: When you provide a command with both an answer and a URL, the URL is given precedence over the answer.

##### MODEL 2: PERSONAL CHATBOT (BOTH VOICE AND CHAT - BASED):

Real-time information retrieval and interactive chatbot features are also features of Model 2, the Personal Chatbot and Information Generator. The Personal Chatbot Page and the Current Info and Image Generation Page are the two main portions of this concept. They each have different functions while sharing some characteristics.

The Personal Chatbot Page has a chatbot UI that responds to different informational prompts and manages user inquiries via the Gemini API. Users can upload images using the Image Button because it also supports image recognition. When an image is sent, the chatbot recognizes it and gives pertinent details about it. Users can input text or use voice commands to interact with the chatbot. The page also includes playback controls like Play, Pause, Speed, and Skip for convenient interaction with spoken responses. Additional submission options like Submit to Voice and Text and Submit provide flexible methods for processing input.

For broader information retrieval, the Current Info and Image Generation Page offers enhanced capabilities. This section retrieves latest information, news, and weather updates using APIs such as the Google API, GNews API, and Weather API. While this page does not support image uploads, it can generate images based on user queries if the prompt contains keywords like "images," "photos," or "pictures". Firstly it generates up-to six images one by one in column wise with equal size. If user images queries includes the word “more”, then six more images will generate. It is to be noted that the user cannot be able to generate approximately six images for every time, because sometimes user can be able to generate below six images, it is completely based on user query strengths and API responses according to fetching images from different magazines related to user queries. When images are generated, users can interact with them through options like Download, Share, and Preview to ensure easy access and distribution. Download button enables user to download the respective image, while share button enables user to share the image via WhatsApp, Email etc. and preview button enables user to view the image in large size.

Both sections provide consistent user controls such as voice buttons, playback controls, and submission options, ensuring a smooth and interactive user experience. The integration of multiple APIs ensures that this model provides both tailored responses and comprehensive real-time information, catering to a variety of user needs.

This comprehensive model ensures seamless functionality by combining tailored chatbot responses, image recognition, and dynamic information retrieval in a unified platform. The flexible design, featuring interactive elements and multiple submission options, enhances accessibility, and empowers users to engage with the system efficiently.

Additionally, each page in this model has two buttons in the top right corner for seamless navigation. One button directs users to the Virtual Chatbot Page (Voice-Activated), and the other directs them to the AI Image Generator Page, ensuring quick access to the project's various features. This intuitive design allows users to switch between models effortlessly, enhancing the overall user experience.



If the user clicks, the Current Information and Image Generation Option will appear.:



##### MODEL 3: TEXT – TO – AI IAMGE GENERATOR:

Using a sophisticated AI API, Model 3, the AI word-to-Image Generator, is intended to convert user word prompts into imaginative visuals. This paradigm is accessible and easy to use since it enables users to create visual content using basic text input.

Users can start by clicking the Generate button after entering their preferred prompt in the designated input field. After processing the word prompt, the AI creates an image in line with its findings. Essential controls on the page include the Reset button, which removes the current image from the display field to make room for new input, and the Download button, which saves the created image. This efficient method guarantees seamless user engagement and rapid content production.

The Hugging Face API is used by the model to effectively produce these AI-based pictures. Users can create original images that are suited to their concepts by utilizing creative prompts. A Reset button makes it possible for users to remove the current image without completely reloading the page, which improves user ease.

Additionally, this model has two buttons in the upper right corner to improve navigation: Go to Virtual Chatbot Page (Voice-Activated) and Go to Personal Chatbot Page. By enabling smooth model change, these buttons guarantee that users may quickly explore various features.

A Logout button is also included in this model, which takes users to the Logout Page. To verify their identity, users must enter their username and password on the logout page. After successfully logging in, they are prompted with the question, "Are you really want to logout?" The user will return to the Text-to-AI Image Generator Page if they click Cancel, and they will log out if they click OK. Controlled application access and appropriate account management are guaranteed by this extra security.



### OVERVIEW OF WORKING PROCESS:

The "Personal Virtual Assistant AI Chatbots and AI Text-to-Image Generator" project successfully combines sophisticated speech and text-based interaction models with safe authentication capabilities. The project's signup, login, and logout procedures guarantee user privacy and data protection. Important information like username, email, password, and security question-answer combinations are gathered on the signup page in order to retrieve the account. While a reset token mechanism can be used to restore forgotten passwords, the login page validates user credentials. By verifying the user's identification prior to terminating the session, the logout procedure improves security. Accessibility and usability are enhanced throughout these security-focused sites by extra features like the Speak button, microphone support, and toggle-eye icons.

By using voice commands, Model 1, the Voice-Activated Chatbot, improves user interaction. By matching saved entries in the "Manage Commands" tab, the chatbot interprets spoken commands from users via the "Start Listening" button. The system prioritizes opening the linked webpage if a command matches a URL; if not, it provides a predetermined response. The system searches Google Chrome if there isn't a match. The Manage Commands area allows users to add, edit, or remove commands, guaranteeing chatbot personalization. A smooth user experience is produced by the dedicated navigation buttons that make switching between this model, the Personal Chatbot, and the AI Image Generator easier.

The Personal Chatbot, Model 2, combines powerful information retrieval with text and voice-based interaction. While the Current Info and Image Generation Page uses the Google, GNews, and Weather APIs for real-time information, news, and weather updates, the Personal Chatbot Page uses the Gemini API to respond to user inquiries. Up to six photos can be produced by users in an organized, column-wise style; further images can be produced as needed. While download, share, and preview buttons enable effective content management, submission and playback options guarantee seamless communication.

Using the Hugging Face API, Model 3, the AI word-to-Image Generator, enables users to turn word prompts into imaginative images. Users can easily download content, create images, and reset inputs. The Virtual Chatbot and Personal Chatbot pages can be switched between with ease thanks to navigation buttons. The protection of user accounts is guaranteed by a secure logout procedure. All things considered, the project effectively combines interactive chatbot features, safe authentication, and AI-driven content production, providing a user-friendly system for information retrieval and imaginative image production.

# CHAPTER – 6 SYSTEM CODING & IMPLEMENTATION

1. **SYSTEM CODING AND IMPLEMENTATION**
	1. **Source Code:** If you want complete source code you can refer the below link https://github.com/srinivas-191/Personal-virtual-assistant-ai-chatbots-and-text-to-ai-image-generator
		1. **index.html**
		2. **styles.css**
		3. **script.js**
	2. **SYSTEM TESTING**

System testing ensures that the entire project functions as expected by validating all features, interactions, and workflows. This project undergoes various testing phases to verify chatbot responses, voice inputs, command management, AI image generation, and secure user authentication. The testing process helps in identifying errors, optimizing system performance, and ensuring smooth user experience.

##### TESTING METHODOLOGIES

* + - 1. **Unit Testing:** Unit testing is performed on individual components to verify their functionality. In this project, unit tests were conducted on features such as user authentication, voice input recognition, chatbot command execution, and AI-generated image retrieval to ensure each module operates independently without errors.
			2. **Integration Testing:** Integration testing ensures that different modules work together seamlessly. The chatbot’s voice input, command management, and API-based information retrieval were tested together to confirm data consistency and smooth workflow between components.
1. **Top-Down Integration:** Testing starts from the chatbot interface and moves downward, verifying interactions between APIs, command storage, and database functionalities.
2. **Bottom-Up Integration:** Lower-level components like voice input processing, text-to-speech responses, and API calls were tested first before integrating with the main chatbot interface.
	* + 1. **User Acceptance Testing (UAT):** The system was tested by real users to ensure ease of use and accurate responses. Users interacted with the chatbot using both text and voice inputs to validate real-world usability, ensuring seamless communication, correct responses, and efficient AI-generated images.

##### TESTING STRATEGY

The project follows a structured testing strategy to ensure system reliability. Various testing phases such as system testing, unit testing, and user acceptance testing were conducted to verify the accuracy and performance of the chatbot and AI image generator. The goal was to detect defects early and improve system efficiency.

* **System Testing:** Performed on the entire application to ensure seamless functionality of voice input, chatbot commands, and AI image generation.
* **Unit Testing:** Focused on individual components like login, command execution, and chatbot interactions to identify and resolve specific errors.

##### TYPES OF TESTS

* **Unit Testing:** Verified the correct functioning of individual components such as text-to-speech processing and chatbot response generation.
* **Integration Testing:** Ensured smooth data exchange between chatbot, voice input, and AI image generator modules.
* **Functional Testing:** Checked whether chatbot responses, AI-generated images, and command executions work as expected.

**Valid Input:** Verified correct responses for predefined commands. **Invalid Input:** Tested how the system handles incorrect user inputs. **Functions:** Ensured chatbot functions work properly without failures. **Output:** Verified AI-generated images match user prompts.

**Systems/Procedures:** Ensured each process, from voice input to image generation, follows correct procedures.

* **System Test:** Verified end-to-end functionality, ensuring chatbot, AI image generation, and authentication work together efficiently.
* **White Box Testing:** Focused on internal code structures, verifying logical flow and API integration.
* **Black Box Testing:** Validated system behavior without analyzing internal code, focusing on chatbot response accuracy and UI functionality.

# CHAPTER – 7 EXECUTION STEPS

## EXECUTION STEPS

### INSTALLATION OF SOFTWARES

##### INSTALLATION OF VISUAL STUDIO CODE

This outlines the step-by-step process to set up, run, and test the "Personal Virtual Assistant AI Chatbots and AI Text-to-Image Generator" project. Since HTML, CSS, and JavaScript were used in the development of this project, along with API integrations, the execution processes center on configuring the coding environment, integrating APIs, and launching the application.

* + - 1. Configuring the Environment for Development:

The following equipment and software are needed to develop the project:

* + - * + Code Editor: Set up a text editor like Notepad++, Sublime Text, or Visual Studio Code (VS Code).
				+ Web browser: For testing and debugging, use Google Chrome.
				+ Live Server (Optional): To view updates in real time when using Visual Studio Code, install the "Live Server" plugin.
			1. How to get Visual Studio Code (the Code Editor):

**Step 1.** Visit the page where VS Code is downloaded:

* + - * + Launch a web browser and go to https://code.visualstudio.com/download, the official VS Code download website.

**Step 2.** Selecting Your Operating System in Step Two:

* + - * + Your operating system (Windows, macOS, or Linux) will be automatically identified by the download page, which will then provide the relevant installer.
				+ You can manually choose the download button for your operating system if you're using a different one.

**Step 3.** Download the installer in step three:

* + - * + Select the operating system-specific blue download button. As a result, the download begins automatically.

**Step 4.** To Run the Installer:

* + - * + To start the installation process, find the installer file (such as VSCodeUserSetup-x64-1.xx.xx.exe for Windows) and double-click on it after the download is finished.

**Step 5.** Comply with the installation guidelines:

* + - * + License Agreement: Click "Next" to accept the license agreement.
				+ Installation Location: Click "Next" after selecting the preferred installation location (the default is typically acceptable).
				+ Start Menu Folder: Click "Next" after selecting the name of the VS Code Start Menu folder (the default is typically acceptable).
				+ Extra Tasks: Click "Next" after choosing to add Visual Studio Code to the Windows File Explorer context menu (right-click menu) or make a desktop icon.
				+ Install: Press "Install" to begin the process.
				+ Finish: Click "Finish" to close the installer and start Visual Studio Code after the installation is finished.
			1. How to Configure the Code Editor:
1. Launch Visual Studio Code and make a new project folder.
2. Make the following files inside the folder:
	1. The main HTML file, index.html
	2. style.css (the styling CSS file)
	3. script.js, a JavaScript logic file
3. Write a simple HTML structure by opening index.html in Visual Studio Code.
4. Attach the JavaScript and CSS files to index.html.
5. Use the Live Server extension or right-click index.html and choose "Open with Chrome" to launch the project in a browser.

### RUNNING A WEB BROWSER TO MANAGE THE PROJECT:

To launch the project after the development setup is finished, take these actions::

1. Launch VS Code and open the project folder.
2. Launch index.html in a web browser (it is best to use Google Chrome).
3. Verify that every button, input field, and user interface element is shown accurately.
4. Examine how well the pages (Login, Signup, Chatbot, AI Image Generator, etc.) navigate between one another.
5. Right-click index.html and choose "Open with Live Server" to launch it if you're using Live Server.

###### API Registration and Integration Procedure:

The following procedures describe how to acquire and integrate the various APIs used by the project for chatbot responses, AI image creation, and real-time data retrieval:

* + - 1. Procedure for Getting API Keys:
1. Google API (Gemini & Custom Search):
	* Go to the Google Cloud Console website.
	* Start a new project and activate the Gemini and Custom Search APIs.
	* For integration, create an API key and make a note of it.
2. Hugging Face API (AI Image Generation):
	* Complete the Hugging Face registration form.
	* Create an API token by going to the API area.
	* To utilize the API token later in the JavaScript code, copy it..
3. Weather API (Real-Time Weather Updates):
	* Go to OpenWeatherMap to access the Weather API (Real-Time Weather Updates).
	* Register to receive a free API key for accessing weather data.
4. GNews API (Retrieval of News):
	* Sign up at GNews API.
	* To retrieve real-time news, create an account and generate an API key.
		+ 1. JavaScript API Integration:
				- Use the get() method to make requests and manage answers effectively.
				- Put error handling in place for situations in which an API request is unsuccessful or yields insufficient information.
				- Launch the project and use the browser console (F12 > Console Tab) to confirm API answers

### .7.3 DEBUGGING AND TESTING:

Following API integration, the project's performance and functionality are examined:

###### Voice Input Testing:

* + - * Press the microphone buttons to see if the voice input is translated to text.
			* Examine voice commands in both the personal and virtual chatbots.

###### Testing Chatbot Responses:

* + - * Type text inquiries to see if the chatbot responds appropriately.
			* Verify whether unknown search terms go to Google Search.

###### Testing AI Image Generation:

* + - * Use the AI Image Generator to enter several prompts and see whether any images are produced.
			* To clear the image field, test the Reset button.

###### Navigation Testing:

* + - * To navigate between the pages for the AI Image Generator, Personal Chatbot, and Virtual Chatbot, click on the navigation buttons.

###### Using Developer Tools for Debugging:

* + - * Press F12 in Chrome to launch the Chrome Developer Tools.
			* Look for errors in the console and address them.

# CHAPTER – 8 RESULTS

## RESULTS

### LOGIN, SIGNUP AND FORGOT PASSWORD PAGE

##### OPENING SCREEN OF PROJECT

The opening screen of project contains login page. If you’re already existing user, enter your credentials and click on login button, else click on signup option to create an account for yourself.



**Fig 8.1.1.1 Login Page**

****

**Fig 8.1.1.2 Signup Page**

* + 1. **Forgot password**

If you Want to reset your password or when you forgot your password, you must click on the reset option, beside forgot password prompt and can reset your old password by entering your registered email and you’ve to answer your security question linked with your email.



###### Fig 8.1.2.1 Forgot Password Page

If you answer correctly to your security question, a token will be generated to use it in your reset password screen as shown below.



###### Fig 8.1.2.2 Reset Token Fig 8.1.2.3 Password Reset Page

In the above shown figure, you can enter your reset token and new password and then confirm your password to change the password for your registered account.

### MODEL-1 VOICE ACTIVATED CHATBOT

Once you’re logged in successfully, you’ll be automatically redirected to Voice activated chatbot as shown below. If you want to redirect to personal chatbot or text-to-AI image generator page, you can choose from the options as shown in the same page.



Fig 8.2.1 Voice Activated Chatbot Page Fig 8.2.2 When Command is giving through your Voice

If you’re logging in for the first time, you can add various commands by tapping on the manage commands button in your voice activated chatbot page. If not, your command will be shown the Google results.



###### Fig 8.2.3 Manage Commands Page Fig 8.2.4 Adding Commands

As said earlier, you can add commands on your own and can give the link directly to which the given command has to be navigated as shown in fig 8.2.1.3.

### MODEL-2 PERSONAL CHATBOT

You can navigate to personal chatbot by clicking on the “Go to personal chatbot page” option in voice activated chatbot page or from text-to-AI image generator page. In the below pictures, user given different prompts and chatbot given answers to the user prompts upto it’s ability.



**Fig 8.3.1 Example of Personal Chatbot Page Fig 8.3.2 Example Image Recognition**

****

**Fig 8.3.3 Example of current weather retrieving Fig 8.3.4 Example of Image Generation**

### MODEL-3 TEXT-TO-AI-IMAGE GENERATOR

You can navigate to text-to-AI-image-generator page by clicking on the “Go to text to ai image generator page” option in voice activated chatbot page or from personal chatbot page. In the picture Fig 8.4.2, user given the prompt “superman riding” in the search bar. The AI Image generated according to the user prompt.



**Fig 8.4.1 Text-To-AI-Image-Generator Page Fig 8.4.2 Example of AI Image Generation**

# CHAPTER – 9 CONCLUSION

## CONCLUSION

Voice-activated chatbot interactions, AI-generated responses, and image generation are all skillfully combined into one system by the Personal Virtual Assistant AI Chatbots and AI Text-to-Image Generator project. It supports real-time information retrieval, AI-driven image generation, and smooth voice and text input connection between users and the chatbot. The use of the Hugging Face, Google, Gemini, and Weather/GNews APIs improves the chatbot's capacity to deliver precise and dynamic responses. The system guarantees effective navigation and usability through command management, API integration, and structured user authentication. While the AI-driven image generator enables users to visualize creative ideas through AI-generated artwork, the chatbot functions as an interactive and entertaining platform suitable for customer service, virtual assistance, and educational resources. Thorough testing ensured seamless interaction and effective output generation by validating the system's functionality, dependability, and user experience. Offline functionality, enhanced chatbot intelligence, support for mobile applications, and multilingual voice recognition are possible future improvements. All things considered, this project shows how AI may be used practically in virtual assistants and creative AI tools, making it a significant advancement in AI-driven automation and human-computer interaction.

# CHAPTER – 10 FUTURE ENHANCEMENTS

## FUTURE ENHANCEMENTS

A number of improvements might be made to the AI Chatbots and AI Text-to-Image Generator project for Personal Virtual Assistants to increase its usefulness and user experience. The chatbot's offline capabilities, which enables it to offer simple answers without an online connection, is one significant enhancement. Improving speech recognition and natural language processing (NLP) will increase chatbot accuracy and user engagement. A larger audience will be able to use the system with linguistic support. It is possible to enhance the AI picture generator to offer additional customization choices and higher-resolution photos. Performance can be improved by using caching techniques and optimizing API response times. In order to increase usability across platforms, future enhancements might possibly incorporate support for mobile applications. The system will be significantly improved by these additions, becoming more adaptable, effective, and user-friendly.

# CHAPTER – 11 BIBLIOGRAPHY

## BIBLIOGRAPHY

1. Brown, T., et al. (2020). "Language Models are Few-Shot Learners." Advances in Neural Information Processing Systems (NeurIPS). Available at: https://arxiv.org/abs/2005.14165.
2. Ramesh, A., et al. (2021). "Zero-Shot Text-to-Image Generation." OpenAI Research. Available at: https://openai.com/dall-e.
3. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." Devlin, J., et al., 2019. Activities of NAACL-HLT. https://arxiv.org/abs/1810.04805 is the URL for information.
4. In 2014, Goodfellow and colleagues published "Generative Adversarial Networks." arXiv preprint. (https://arxiv.org/abs/1406.2661) Now accessible.
5. A. Vaswani and colleagues (2017). "Attention is All You Need." Neural information processing system advancements. https://arxiv.org/abs/1706.03762 is the resource.
6. A. Radford and associates (2021). "Learning Transferable Visual Models From Natural Language Supervision." Research on OpenAI. https://arxiv.org/abs/2103.00020 is the URL for this article.
7. B. Zhou and associates (2017). "Places: A 10 Million Image Database for Scene Recognition." IEEE Transactions on Machine Intelligence and Pattern Analysis.
8. APIs Documentation:

**Custom Search Engine ID and Google API**– sed to retrieve pertinent data and real-time search results. [https://developers.google.com/custom-search is the documentation.](https://developers.google.com/custom-search)

**Gemini API** – Text-based interactions and chatbot responses driven by AI are made possible by the Gemini API. https://ai.google.dev/gemini-api is the Documentation.

**GNews API** – The most recent news stories from various sources are retrieved by the GNews API. https://gnews.io/docs is the Documentation.

**OpenWeather API** – Real-time weather updates for many places are provided by the OpenWeather API. https://openweathermap.org/api is the Documentation.

**Hugging Face API** – The Hugging Face API uses user word prompts to create AI-based graphics. https://huggingface.co/docs is the Documentation.

1. Mozilla Developer Network (MDN) is the JavaScript reference. https://developer.mozilla.org/ is the URL.
2. W3C Standards for CSS and HTML Guidelines. https[://www.w3.org/](http://www.w3.org/) is the URL.