

Design & Implementation of a Simple ChatBot: Moyna

by

Md. Yasin Molla

ID: CSE2001019017

Sujan Ahmmad

ID: CSE2001019149

Panna Akter

ID: CSE2001019223

Most. Surmita Akter

ID: CSE2001019269

Rayhanul Islam

ID: CSE1701010009

Supervised by

Salma Tabashum

Submitted in partial fulfillment of the requirements for the degree of Bachelor of
Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

September 2023

Design & Implementation of Moyna

by

Md. Yasin Molla

ID: CSE2001019017

Sujan Ahmmad

ID: CSE2001019149

Panna Akter

ID: CSE2001019223

Most. Surmita Akter

ID: CSE2001019269

Rayhanul Islam

ID: CSE1701010009

Supervised by

Salma Tabashum

Submitted in partial fulfillment of the requirements for the degree of Bachelor of
Science in Computer Science and Engineering



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)**

September 2023

APPROVAL

The project titled “**Moyna**” submitted by **Md. Yasin Molla** (CSE2001019017), **Sujan Ahmmad** (CSE2001019149), **Panna Akter** (CSE2001019223) **Most. Surmita Akter** (CSE2001019269, **Rayhanul Islam** (CSE1701010009) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Salma Tabashum

Lecturer

Department of Computer Science and Engineering
Sonargaon University (SU)

Supervisor

(Examiner Name and Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name and Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name and Signature)

Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Salma Tabashum, Lecturer**, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this project has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Salma Tabashum)
Supervisor

Signature

Md. Yasin Molla
ID: CSE2001019017

Sujon Ahmmad
ID: CSE2001019149

Panna Akter
ID: CSE2001019223

Most. Surmita Akter
ID: CSE2001019269

Rayhanul Islam
ID: CSE2001019269

ABSTRACT

Moyna is a web-based chatbot project built using PHP programming language, designed to provide a user-friendly interface for simulating conversations with end-users. This chatbot application enables users to ask questions and make queries, receiving automated responses generated by the chatbot.

Moyna offers a versatile and user-friendly platform for efficient and interactive communication between users and the chatbot, with administrators having full control over its responses and system settings. This project represents an innovative approach to enhancing user engagement and support in online environments through the integration of chatbot technology.

The heart of the system lies in its robust admin panel, where administrators can effortlessly manage and fine-tune the chatbot's responses. The admin panel offers an array of features related to chatbot information, including customization options for the welcome message, handling queries with no results, and even the ability to manage avatars.

A key feature of this system is its response management. Administrators can store responses in the system, associating them with relevant questions to ensure prompt and accurate responses to user queries. Additionally, the system records query that couldn't be answered, allowing administrators to assess their relevance to the site and create responses for these unanswered questions when appropriate.

Technologically, the project employs a dynamic blend of front-end technologies, such as HTML5, CSS, JavaScript, jQuery, Ajax, and Bootstrap, using the AdminLTE template to create an engaging user experience. The back-end is powered by PHP (version 8.0.2) and utilizes a MySQL database with MySQLi Object-Oriented Programming (OOP) for seamless data management.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We are auspicious that we had the kind association as well as supervision of **Salma Tabashum**, Lecturer, Department of Computer Science and Engineering (CSE), Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude our honorable departmental head **Bulbul Ahamed**, (Associate Professor & Head, Department of Computer Science and Engineering) for his kind concern and precious suggestions and special gratitude to **Brig. Gen. (Retd) Prof Habibur Rahman Kamal, ndc, psc**, Dean, Faculty of Science and Engineering.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

AJAX	Asynchronous JavaScript and XML
CPU	Central Processing Unit
DBMS	Database Management System
FTP	File Transfer Protocol
CSS	Cascading Style Sheet
HTML	Hypertext markup language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
NPL	Natural Language Processing
OOP	Object Oriented Program
PHP	Hypertext Pre-Processor
RAM	Random Access Memory
ROM	Read Only Memory
SQL	Structured Query Language
UI	User Interface

TABLE OF CONTENTS

Title	Page No.
APPROVAL	ii
DECLARATION	iii
ABSTRACT	iv
ACKNOWLEDGEMENT	v
LIST OF ABBREVIATION	vi
CHAPTER 1	1-2
INTRODUCTION TO SIMPLE CHATBOT SYSTEM	
1.1 Background and context.....	1
1.2 Problem Statement	1
1.3 Objectives of the Project	1
1.4 Scope and Limitations	1
1.5 Significance of the Project.	2
1.6 Organization of the report	2
CHAPTER 2	3-4
Literature Review	
2.1 Overview of Chatbot Technology	3
2.2 Previous Work and Research in Chatbots	3
2.3 Related Technologies and Tools	3
2.4 Key Concepts and Terminologies	4
CHAPTER 3	5-6
System Architecture	
3.1 High Level System Architecture	5
3.2 Front-End Technologies.....	5
3.3 Back-End Technologies	5
3.4 Communication	5
3.5 Workflow	6
CHAPTER 4	7-8
Requirements Analysis	
4.1 User Requirements	7

4.2	System Requirements	7
4.3	Use Cases and Scenarios	7
4.4	Functional and Non-Functional Requirements	8
CHAPTER 5		9-14
Design and Implementation		
5.1	User Interface Design	9
5.2	Database Schame Design	9
5.3	Front-End Implementation	10
5.4	Back-End Implementation	10
5.5	Integration of Admin Panel	11
5.6	Security Measures	12
CHAPTER 6		15
Testing and Validation		
6.1	Testing Strategies and Methodologies	15
6.2	Test Cases and Test Data	15
6.3	Bug Tracking and Resolution	15
CHAPTER 7		16-17
System Evaluation		
7.1	Performance Evaluation	16
7.2	User Feedback and Usability Testing	16
7.3	Scalability and Maintencance Considerations	16
7.4	Comparison with Initial Objectives	17
CHAPTER 8		18
Appendices		
8.1	Code Snippets.....	18
CHAPTER 9		40-41
Conclusion		
9.1	Summary of Achivements	40
9.2	Contributions of the Field.....	40
9.3	Future Work.....	41
REFERENCES		42

LIST OF FIGURES

<u>Figure No.</u>	<u>Title</u>	<u>Page No.</u>
Fig-3.1	High-Level Architecture	17
Fig-5.1	User page	19
Fig-5.1	User Profile	19
Fig-5.1	Database Schema Design	20
Fig-5.5	Admin login page	21
Fig-5.5	Admin Dashboard	21
Fig-5.6	System Setting page	22
Fig-5.6	Response list page	22
Fig-5.6	Create Response page	23
Fig-5.6	Unanswered list page	23
Fig-5.6	Some question and answer	23-24
Fig-8.1	Code Snippets	28-31

CHAPTER 1

INTRODUCTION TO CHATBOT SYSTEM

1.1 Background and Context

The rapid advancement of chatbot technology in recent years has opened up new possibilities for enhancing user interactions on websites and applications. Chatbots can provide instant responses to user queries, streamline customer support, and improve user engagement. A chatbot is a computer program that can understand human language and converse with a user via a website or a messaging app. Chatbots can handle various tasks online — from answering simple questions and scheduling calls to gathering customer feedback. Brands use bots to automate their business processes, speed up customer service, and lower support costs

1.2 Problem Statement

Many websites and applications lack efficient means of user interaction. Users often struggle to find the information they need, leading to frustration and potential loss of engagement. The project “Moyna” aims to address this issue by developing a Simple Chatbot System that can provide immediate responses to user queries.

1.3 Objectives of the Project

The primary objective of this project is to create a user-friendly chatbot application capable of simulating human-like conversations. This involves developing both the front-end user interface and the back-end logic to enable meaningful interactions. Chatbots can make this engagement more interactive. Buyers rarely talk to the people within businesses, so chatbots open a communication channel where customers can engage without the stress of interacting with another person.

1.4 Scope and Limitations

The project scope encompasses the development of the chatbot System, including the admin panel for response management. However, it does not cover advanced AI natural language processing (NLP) capabilities. The project is limited to responding to predefined queries. Customer perceptions - A chatbot personality can make your bot

more engaging, likable, and relatable. This will encourage customers to interact with it and help to overcome any skepticism or negative attitudes that consumers have towards chatbot technology.

Absence of human touch - A well-created personality will help your chatbot feel less robotic and deliver a more humanized experience.

Lack of empathy - The addition of a personality can be used to make your bot appear more emotive and empathetic.

1.5 Significance of the Project

Moyna is significant because it offers a cost-effective solution for enhancing user interactions on websites. It can improve user engagement, answer frequently asked questions, and serve as a valuable tool for businesses and organizations.

1.6 Organization of the Report

The implementation of translational medicine is associated with considerable costs of equipment, staff competence, and doctor-patient (DP) and clinic-patient (CP) communication. The application of DP and CP systems evolved from e-mail letters to website assistance chat and smartphone apps in the context of the m-health paradigm. The report is organized into several chapters, each addressing specific aspects of the project, including system architecture, requirements analysis, design and implementation, testing and validation, system evaluation, project management, and conclusions.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview of Chatbot Technology

Chatbots are software applications designed to simulate human conversation. They have gained popularity across various industries, including customer support, e-commerce, and healthcare. The use of chatbots evolved rapidly in numerous fields in recent years, including Marketing, Supporting Systems, Education, Health Care, Cultural Heritage, and Entertainment. In this paper, we first present a historical overview of the evolution of the international community's interest in chatbots. Next, we discuss the motivations that drive the use of chatbots, and we clarify chatbots' usefulness in a variety of areas. Moreover, we highlight the impact of social stereotypes on chatbots design.

2.2 Previous Work and Research

Prior research and existing chatbot solutions have contributed to the development of this project. Studying these resources helped in understanding best practices and design considerations. Whereas a search engine provides only search results (a list of related literature), a chatbot can be considered a more advanced research assistant in that it provides its own answers to users' questions based on the related literature that it has learned.

2.3 Related Technologies

Latent Semantic Analysis (LSA) may be used together with AIML for the development of chatbots. It is used to discover likenesses between words as vector representation.

Chat script, being the successor of the AIML language, is an expert system, which consists of an open-source scripting language and the engine that runs it. It is comprised of rules which are associated with topics, finding the best item that matches the user query string and executing a rule in that topic.

Rive Script is a plain text, line-based scripting language for the development of chatbots and other conversational entities. It is open-source with available interfaces for Go, Java, JavaScript, Perl, and Python.

Natural Language Processing (NLP), an area of artificial intelligence, explores the manipulation of natural language text or speech by computers.

Natural Language Understanding (NLU) is at the core of any NLP task. It is a technique to implement natural user interfaces such as a chatbot. NLU aims to extract context and meanings from natural language user inputs, which may be unstructured and respond appropriately according to user intention. Technologies such as HTML5, CSS, JavaScript, jQuery, Ajax, Bootstrap, PHP, and MySQL were leveraged in the development of the chatbot system.

2.4 Key Concepts

Key concepts in chatbot technology include natural language processing (NLP), intent recognition, response generation, and user interaction models. Minimal human interference in the use of devices is the goal of our world of technology. Chatbots can reach out to a broad audience on messaging apps and be more effective than humans are. At the same time, they may develop into a capable information-gathering tool. They provide significant savings in the operation of customer service departments. With further development of AI and machine learning, somebody may not be capable of understanding whether he talks to a chatbot or a real-life agent.

CHAPTER 3

SYSTEM ARCHITECTURE

3.1 High-Level Architecture

The system architecture is designed and developed with PHP, JavaScript, HTML, CSS, Bootstrap, and MySQL involves the organization and interaction of various components to create a functional and visually appealing web application. This architecture provides an overview of how these technologies work together to deliver a seamless user experience.

3.2 Front-End Technologies

- **HTML (HyperText Markup Language):** HTML is used to structure the content of the web pages. It defines the layout and elements on the user interface.
- **CSS (Cascading Style Sheets):** CSS is responsible for styling the HTML elements. It controls the presentation, including colors, fonts, layout, and responsiveness.
- **JavaScript:** JavaScript adds interactivity to the frontend. It handles user interactions, such as form validation, dynamic content updates, and client-side data processing.
- **Bootstrap:** Bootstrap is a CSS framework that provides pre-designed UI components and responsive layouts. It simplifies the frontend development process by offering a set of reusable styles and components.

3.3 Back-End Technologies

HTML5, CSS, JavaScript, and Bootstrap facilitate a responsive and visually appealing user interface. PHP handles server-side logic and interacts with the MySQL database.

- **PHP (Hypertext Preprocessor):** PHP is a server-side scripting language used for processing requests, interacting with the database, and generating dynamic content. It serves as the core of the backend logic.
- **MySQL:** MySQL is a relational database management system (RDBMS) that stores and manages data. It is used to create, read, update, and delete records, ensuring data persistence and retrieval.

3.4 Communication

- **HTTP (Hypertext Transfer Protocol):** HTTP is the protocol that facilitates communication between the frontend and backend. It defines how requests and responses are formatted and transmitted.
- **AJAX (Asynchronous JavaScript and XML):** AJAX is a technique used to send and receive data between the frontend and backend asynchronously. It enables dynamic updates without requiring a full page refresh.

3.5 Workflow

- **User Interaction:**
Users interact with the web application through the frontend, which includes HTML forms, buttons, and other UI elements.
- **Frontend Processing:**
JavaScript handles client-side validations and interactions. It may make AJAX requests to the backend for data retrieval or submission.
- **Backend Processing:**
PHP processes incoming requests, including data validation, business logic, and database interactions. It may utilize MySQL to fetch or store data.
- **Database Interaction:**
MySQL manages the storage and retrieval of data. It ensures data consistency and persistence.
- **Response Generation:**
PHP generates dynamic HTML content based on the processed data and sends it back to the frontend.
- **Presentation:**
CSS styles the HTML content, ensuring a visually appealing and responsive user interface.
- **User Feedback:**
The updated content is displayed to the user, providing feedback based on their interactions.

CHAPTER 4

REQUIREMENTS ANALYSIS

4.1 User Requirements

User Requirements refer to the needs, expectations, and constraints placed on the Chatbot System by the end-users or customers who will interact with it. These requirements are typically gathered through interviews, surveys, and observations. Key considerations in this section include:

- **User Profiles:** Define the various types of users who will interact with the chatbot. This may include customers, support staff, administrators, or other stakeholders.
- **User Goals:** Document what users aim to achieve by using the chatbot. For instance, users may seek information, request assistance, or perform specific tasks.
- **User Expectations:** Understand the user's expectations regarding response times, accuracy, and user-friendliness.
- **User Feedback Mechanisms:** Specify how users can provide feedback or report issues with the chatbot's performance.

4.2 System Requirements

System Requirements focus on the technical and infrastructure aspects necessary for the Chatbot System's successful operation. This section addresses the following key points:

- **Hardware Requirements:** Detail the hardware components needed to host and run the chatbot, including servers, memory, and storage capacity.
- **Software Requirements:** List the software components required, such as the programming languages, frameworks, and libraries.
- **Integration Requirements:** Identify any existing systems or external services that the chatbot must integrate with, such as databases, APIs, or third-party platforms.
- **Scalability and Performance:** Specify the system's scalability needs and performance expectations to handle different levels of user interaction.

4.3 Use Cases

Use Cases describe the various scenarios or interactions between users and the Chatbot System. Each use case represents a specific user goal or task that the chatbot must support. Key elements of this section include:

- **Use Case Diagram:** Provide a visual representation of the chatbot's interactions with different users and external systems.
- **Use Case Descriptions:** Detail the steps involved in each use case, including user inputs, system responses, and any error-handling procedures.
- **Actor Descriptions:** Define the roles or actors involved in each use case, such as end-users, administrators, or external systems.
- **Alternate and Exception Flows:** Describe alternative paths and error scenarios that may occur during a use case's execution.

4.4 Functional and Non-Functional Requirements

Functional Requirements outline the specific features and functionalities that the Chatbot System must deliver. They define what the system should do to meet user needs. Examples of functional requirements include:

- **Natural Language Processing:** Specify how the chatbot should understand and respond to user input.
- **Data Storage:** Detail how user interactions and data should be stored and managed.

Non-Functional Requirements focus on the qualities or characteristics that the Chatbot System must possess. These requirements address attributes like performance, security, and usability. Examples of non-functional requirements include:

- **Response Time:** Specify the maximum acceptable response time for the chatbot.
- **Security:** Define security measures to protect user data and prevent unauthorized access.
- **Usability:** Describe the user interface design principles that should be followed to ensure an intuitive user experience.

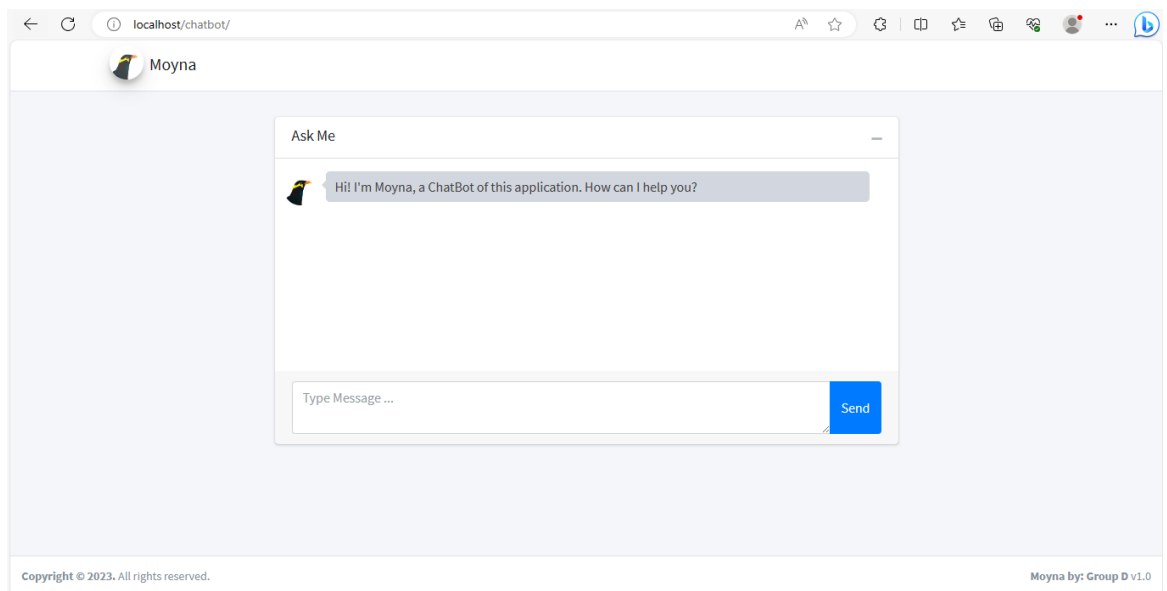
CHAPTER 5

DESIGN AND IMPLEMENTATION

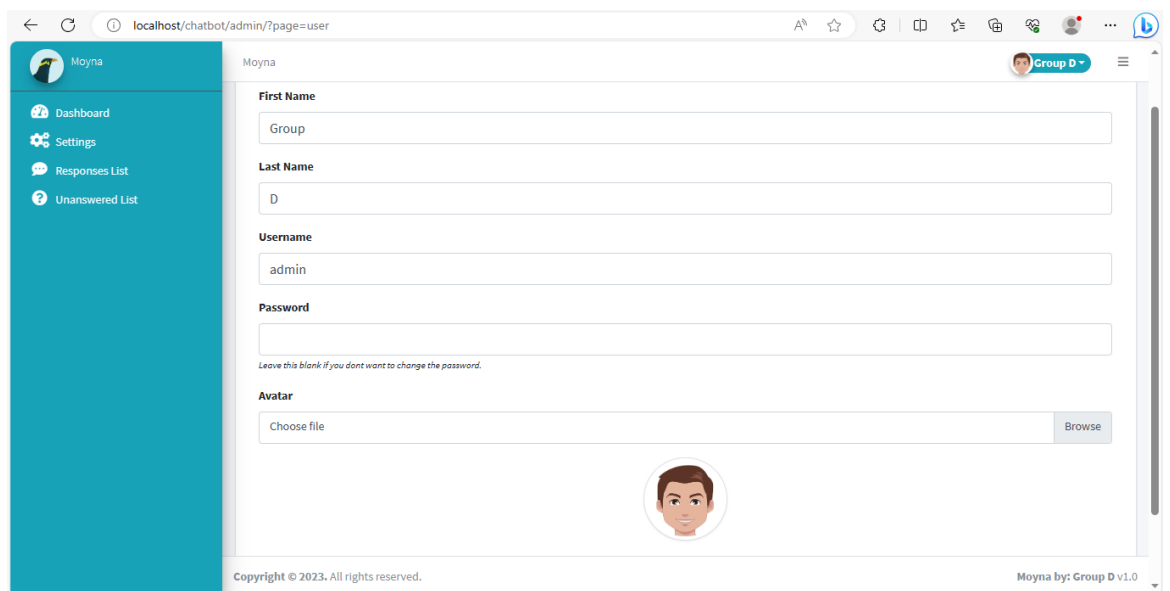
5.1 User Interface Design

The user interface is designed to be user-friendly, with a chat-like conversation flow. It incorporates HTML5, CSS, JavaScript, and Bootstrap for responsiveness and visual appeal.

User Interface (UI) :



User profile:



5.2 Database Schema Design

The database schema includes tables for user data, responses, questions, and system settings. Relationships between tables ensure data consistency.

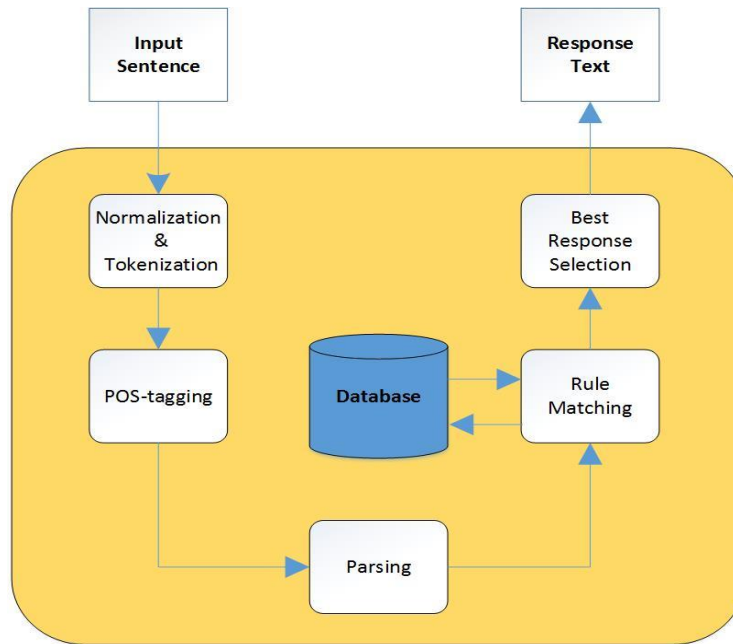


Fig: 1 (Database Schema)

5.3 Front-End Implementation

Front end has the utmost significant part in online framework promotion. Generally, Front end is the introduction layer the took a shot. User can easily cooperate to our framework by the framework design. The part of a website that the user interacts with directly is termed the front end. It is also referred to as the ‘client side of the application. It includes everything that users experience directly: text colors and styles, images, graphs and tables, buttons, colors, and a navigation menu. HTML, CSS, and JavaScript are the languages used for Front End development. Responsiveness and performance are the two main objectives of the Front End. The developer must ensure that the site is responsive i.e. it appears correctly on devices of all sizes no part of the website should behave abnormally irrespective of the size of the screen.

HTML, CSS, and JavaScript are used to create an interactive user interface. The chatbot interface dynamically displays responses and allows users to input queries.

5.4 Back-End Implementation

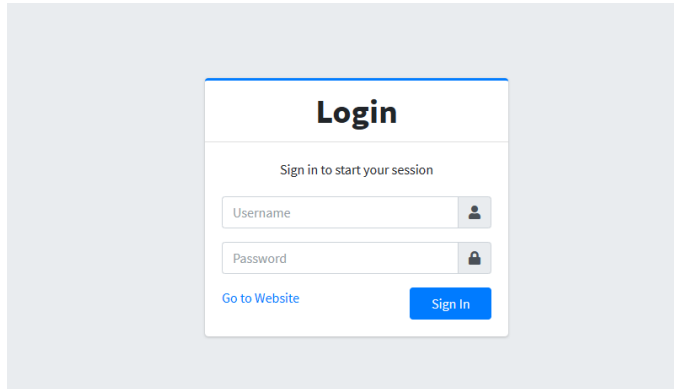
Back end have been one kind of machine which is conduct a site and the customer have been acquainted it whatever apparently partner if it also likewise with client-

side upliftment, yet it is consistently coming up brief outside of hasitation volatility The backend is the server side of the website. It stores and arranges data, and also makes sure everything on the client side of the website works fine. It is part of the website that you cannot see and interact with. It is the portion of software that does not come in direct contact with the users. The parts and characteristics developed by backend designers are indirectly accessed by users through a front-end application. Activities, like writing APIs, creating libraries, and working with system components without user interfaces or even systems of scientific programming, are also included in the backend. PHP (version 8.0.2) processes user queries, retrieves responses from the database, and handles admin panel functionalities. MySQL is used for secure data storage and retrieval.

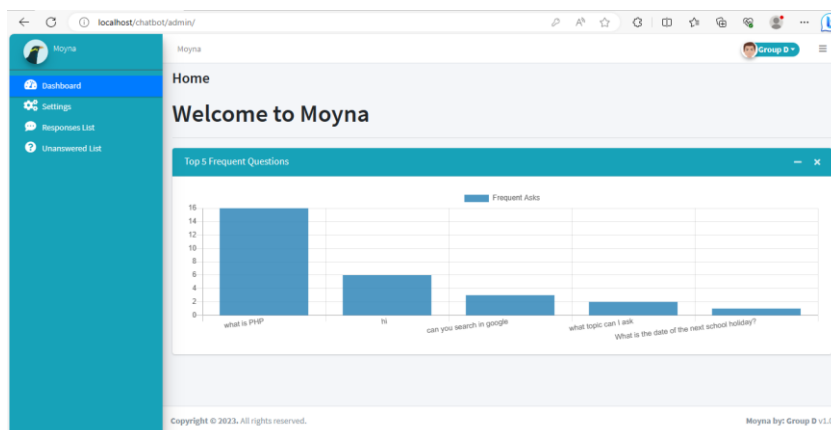
5.5 Integration of Admin Panel

The admin panel is seamlessly integrated into the system, providing administrators with the ability to manage responses, system settings, and user queries.

Admin login page:



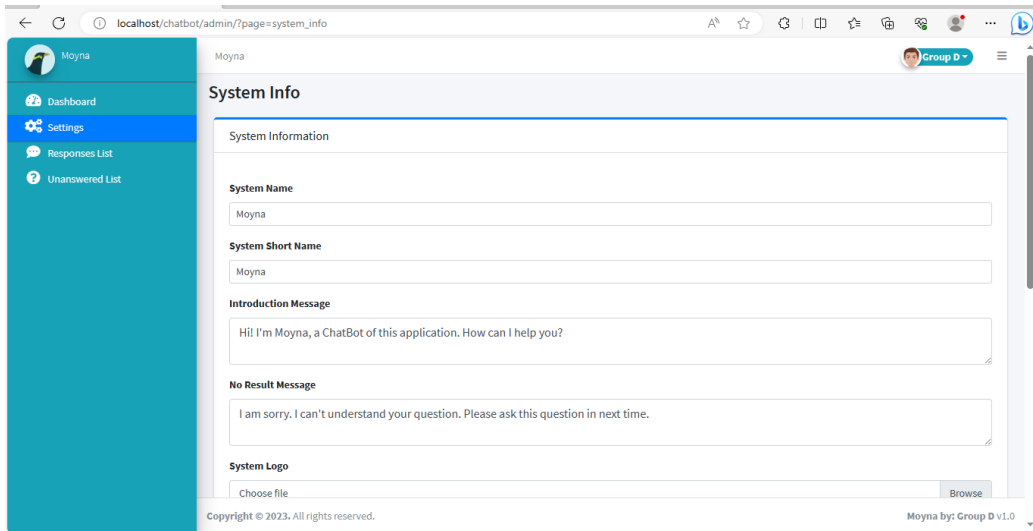
Admin Dashboard:



5.6 Security Measures

Security measures include input validation, protection against SQL injection, and user authentication for admin access. User data is encrypted and stored securely.

System Setting page:

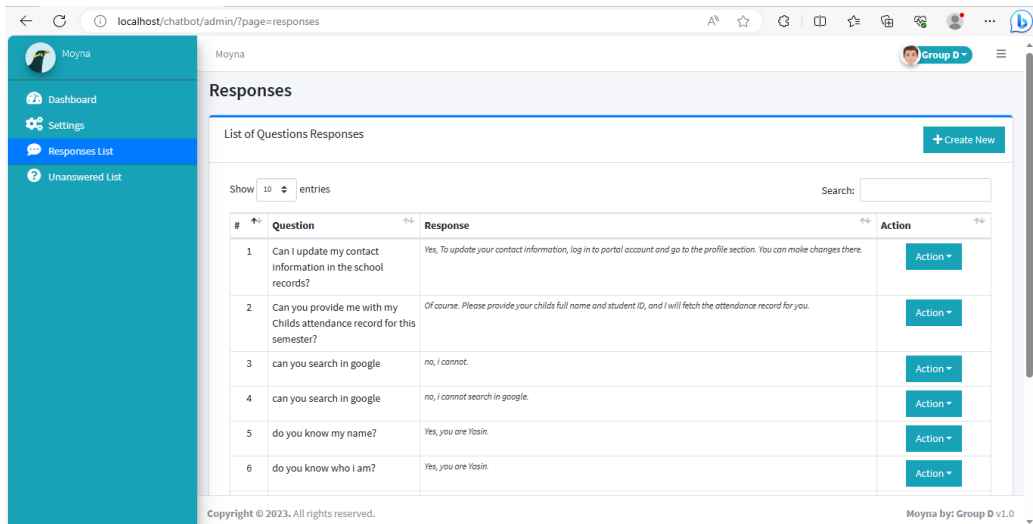


The screenshot shows the 'System Info' page in the Moyna admin interface. The page is titled 'System Info' and contains several form fields for system configuration. The left sidebar shows the navigation menu with 'Settings' selected. The main content area includes:

- System Information** section with the following fields:
 - System Name:** Moyna
 - System Short Name:** Moyna
 - Introduction Message:** Hi! I'm Moyna, a ChatBot of this application. How can I help you?
 - No Result Message:** I am sorry, I can't understand your question. Please ask this question in next time.
 - System Logo:** Choose file (with a 'Browse' button)

Copyright © 2023. All rights reserved. Moyna by: Group D v1.0

Response list page:



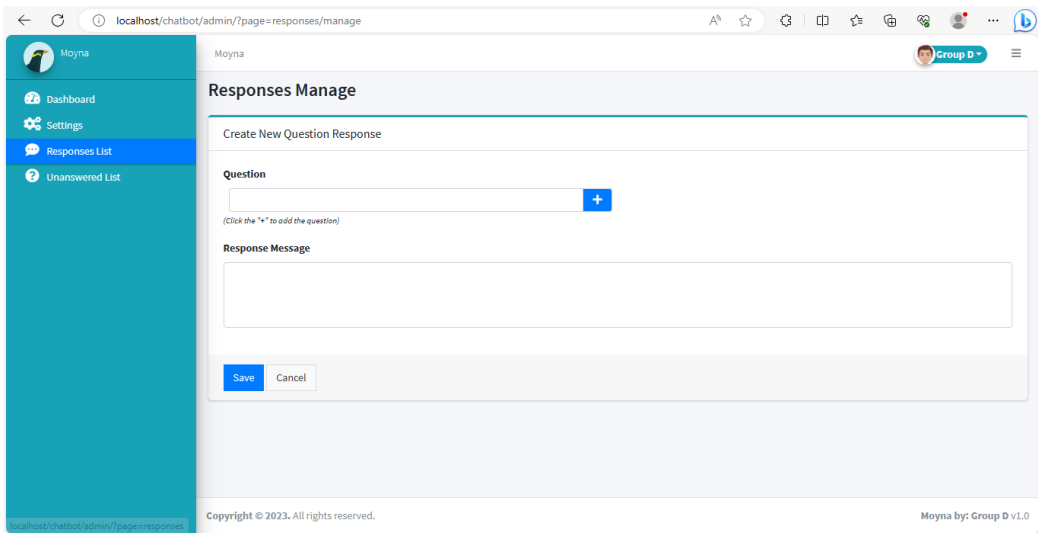
The screenshot shows the 'Responses' page in the Moyna admin interface. The page is titled 'Responses' and displays a list of questions and their corresponding responses. The left sidebar shows the navigation menu with 'Responses List' selected. The main content area includes:

- List of Questions Responses** section with a '+ Create New' button.
- A search bar and a 'Show 10 entries' dropdown.
- A table with the following columns: #, Question, Response, and Action.

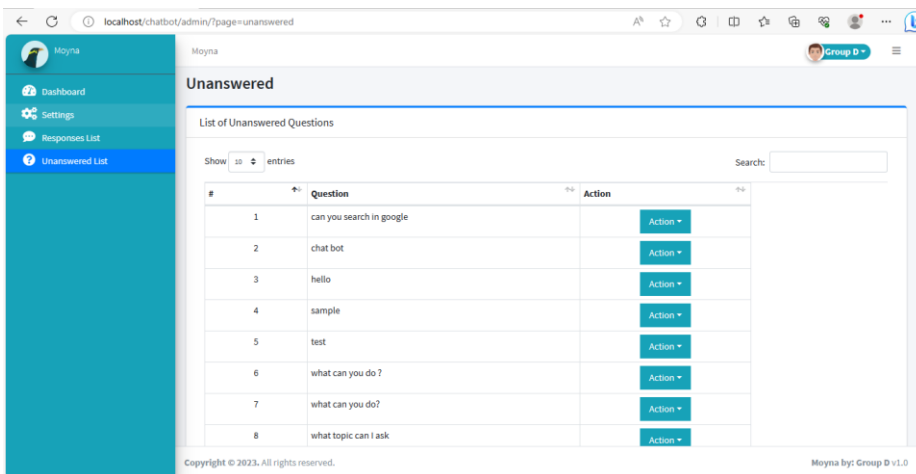
#	Question	Response	Action
1	Can I update my contact information in the school records?	Yes, To update your contact information, log in to portal account and go to the profile section. You can make changes there.	Action
2	Can you provide me with my Childs attendance record for this semester?	Of course. Please provide your childs full name and student ID, and I will fetch the attendance record for you.	Action
3	can you search in google	no, I cannot.	Action
4	can you search in google	no, I cannot search in google.	Action
5	do you know my name?	Yes, you are Yasin.	Action
6	do you know who I am?	Yes, you are Yasin.	Action

Copyright © 2023. All rights reserved. Moyna by: Group D v1.0

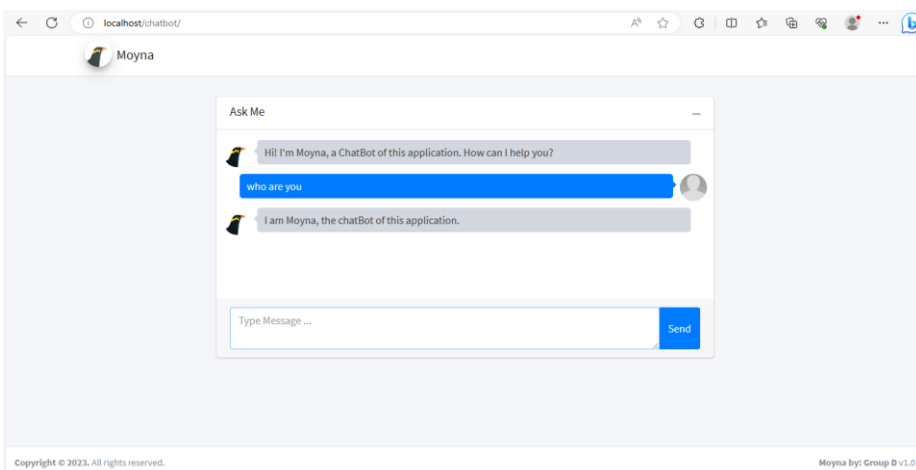
Create Response page:



Unanswered list page:



Some question and answer:



Ask Me

I am Moyna, the chatBot of this application.

what topic can I ask

You can ask me about something related to this website.

What is the date of the upcoming parent-teacher conference?

The next parent-teacher conference is scheduled for December 15th, 2023. You can sign up for slots through our online scheduling system.

Type Message ... Send

Ask Me

what is PHP

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

What are the office hours of the school administration?

The school administration office is open from 8:00 AM to 4:30 PM, Saturday through Thursday.

Type Message ... Send

Ask Me

sample

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue, lectus non tincidunt viverra, lacus erat venenatis mauris, sed hendrerit libero diam ac tellus. Integer imperdiet massa lacus, sed porta ligula efficitur at. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae;

can you search in google

no, i cannot.

Type Message ... Send

CHAPTER 6

TESTING AND VALIDATION

6.1 Testing Strategies

Testing strategies encompass unit testing, integration testing, and user acceptance testing. It ensures the application functions as intended. Conversational interfaces let you connect with customers at scale and 24/7. However, when they fail to deliver a great customer experience, they also do it at scale. Therefore, testing should be the key aspect of any chatbot development process. It helps evaluate your bot's performance in terms of accuracy, speed, and usefulness. It can also give hints on improving your bot's language and personality or which features to add or remove to streamline the flow.

6.2 Test Cases and Results

Test cases include scenarios where users interact with the chatbot, and expected outcomes are defined. Testing results indicate the success of these scenarios.

- Chatbot should get load with the webpage
- It should pop up or show its presence by blink
- A little knock or sound should happen on display
- Verify it is following the language support if programmed
- Verify its showing greeting if programmed as per the Time zone

6.3 Bug Tracking

A bug tracking system is used to identify and prioritize issues. Bugs are resolved systematically to enhance the application's stability and functionality. A major component of a bug tracking system is a database that records facts about known bugs," says Wikipedia. "Facts may include the time a bug was reported, its severity, the erroneous program behavior and details on how to reproduce the bug; as well as the identity of the person who reported it and any programmers who may be fixing it

CHAPTER 7

SYSTEM EVALUATION

7.1 Performance Evaluation

Performance metrics such as response time, system uptime, and server resource utilization are evaluated. The system is designed to handle a specific number of concurrent users. Whether your customer gets in touch with you over the phone, email or via your website, you want to give them the best possible service. Chatbots are a great tool to help you give customers a positive online experience. However, to make sure they're working effectively, you need to measure and track their performance. This article will show the best way to do so.

7.2 User Feedback

User feedback is collected to assess the chatbot's usability and effectiveness. Feedback surveys and user interviews provide valuable insights. Customer feedback will help online business owners understand how to develop the business around the target audience. Online business owners use different ways to collect customer feedback. At times, conventional feedback forms do not help e-commerce site owners get their feedback properly from online shoppers.

7.3 Scalability and Maintenance

The system's scalability is assessed, and maintenance requirements are outlined. Scalability considerations include accommodating a growing user base and handling increased traffic. A chatbot may need maintenance for a variety of reasons, including –

- **Improved accuracy** – Chatbots are prone to giving incorrect and irrelevant responses if not maintained regularly with the latest queries and content. So, they need to be trained with the latest database and conversational strands to offer relevant responses to queries.
- **Better engagement** – No matter how advanced a chatbot is, after a while, it will cease to deliver accurate responses due to a lack of maintenance. And this can hugely affect its ability to engage with users.

- **Efficiency and promptness** – An updated chatbot will obviously show more efficiency toward user queries. Most of the time, such tools also deliver quick answers and therefore ensure improved satisfaction levels among customers.
- **A seamless user experience** – A chatbot can provide a seamless experience to users only when it's optimized and kept free of minor issues. The more you take care of the maintenance, the more enjoyable experience you can deliver to users.
- **Cost savings** – Chatbot maintenance can not only improve its performance but can also make your support agents more productive as well

7.4 Comparison with Objectives

The project's outcomes are compared to the initial objectives to determine the extent to which goals were achieved. Any deviations or improvements are noted.

CHAPTER 8

APPENDICES

8.1 Code Snippets

Snip from index page:

```
<!DOCTYPE html>
<html lang="en" class="" style="height: auto;">
<?php require_once('config.php'); ?>
<?php require_once('inc/header.php') ?>
<body class="hold-transition layout-top-nav" >
  <div class="wrapper">
    <?php require_once('inc/topBarNav.php') ?>

    <?php $page = isset($_GET['page']) ? $_GET['page'] : 'portal'; ?>
    <!-- Content Wrapper. Contains page content -->
    <div class="content-wrapper" style="min-height: 567.854px;">
      <!-- Content Header (Page header) -->
      <div class="content-header">

        <!-- /.container-fluid -->
      </div>
      <!-- /.content-header -->
      <!-- Main content -->
      <section class="content">
        <div class="container">
          <?php
            if(!file_exists($page.".php") && !is_dir($page)){
              include '404.html';
            }else{
              if(is_dir($page))
                include $page.'/index.php';
              else
                include $page.'.php';
            }
          ?>
        </div>
      </section>
      <!-- /.content -->
      <div class="modal fade" id="confirm_modal" role='dialog'>
      <div class="modal-dialog modal-md modal-dialog-centered"
role="document">
      <div class="modal-content">
```

```

        <div class="modal-header">
          <h5 class="modal-title">Confirmation</h5>
        </div>
        <div class="modal-body">
          <div id="delete_content"></div>
        </div>
        <div class="modal-footer">
          <button type="button" class="btn btn-primary" id='confirm'
onclick="">Continue</button>
          <button type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
        </div>
      </div>
    </div>
    <div class="modal fade" id="uni_modal" role='dialog'>
      <div class="modal-dialog modal-md modal-dialog-centered"
role="document">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title"></h5>
          </div>
          <div class="modal-body">
          </div>
          <div class="modal-footer">
            <button type="button" class="btn btn-primary" id='submit'
onclick="$('#uni_modal form').submit()">Save</button>
            <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
          </div>
        </div>
      </div>
    </div>
    <div class="modal fade" id="uni_modal_right" role='dialog'>
      <div class="modal-dialog modal-full-height modal-md" role="document">
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title"></h5>
            <button type="button" class="close" data-dismiss="modal" aria-
label="Close">
              <span class="fa fa-arrow-right"></span>
            </button>
          </div>
          <div class="modal-body">
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

<div class="modal fade" id="viewer_modal" role='dialog'>
  <div class="modal-dialog modal-md" role="document">
    <div class="modal-content">
      <button type="button" class="btn-close" data-
dismiss="modal"><span class="fa fa-times"></span></button>
      <img src="" alt="">
    </div>
  </div>
</div>
</div>
<!-- /.content-wrapper -->
<?php require_once('inc/footer.php') ?>
</body>
</html>

```

Snip from config file:

```

<?php
ob_start();
ini_set('date.timezone','Asia/Dhaka');
date_default_timezone_set('Asia/Dhaka');
session_start();

require_once('initialize.php');
require_once('classes/DBConnection.php');
require_once('classes/SystemSettings.php');
$db = new DBConnection;
$conn = $db->conn;

function redirect($url='') {
    if(!empty($url))
        echo '<script>location.href="'.base_url . $url.'"</script>';
}

function validate_image($file){
    if(!empty($file)){
        // exit;
        if(is_file(base_app.$file)){
            return base_url.$file;
        }else{
            return base_url.'dist/img/no-image-available.png';
        }
    }else{
        return base_url.'dist/img/no-image-available.png';
    }
}

function isMobileDevice(){

```

```

$aMobileUA = array(
    '/iphone/i' => 'iPhone',
    '/ipod/i' => 'iPod',
    '/ipad/i' => 'iPad',
    '/android/i' => 'Android',
    '/blackberry/i' => 'BlackBerry',
    '/webos/i' => 'Mobile'
);

//Return true if Mobile User Agent is detected
foreach($aMobileUA as $sMobileKey => $sMobileOS){
    if(preg_match($sMobileKey, $_SERVER['HTTP_USER_AGENT'])){
        return true;
    }
}
//Otherwise return false..
return false;
}
ob_end_flush();
?>

```

Snip from header code:

```

<?php
    // require_once('sess_auth.php');
?>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title><?php echo $_settings->info('title') != false ? $_settings-
>info('title').' | ' : '' ?><?php echo $_settings->info('name') ?></title>
    <link rel="icon" href="<?php echo validate_image($_settings-
>info('logo')) ?>" />
    <!-- Google Font: Source Sans Pro -->
    <!-- <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:300,400,400i,
700&amp;display=fallback"> -->
    <!-- Font Awesome -->
    <link rel="stylesheet" href="<?php echo base_url ?>plugins/fontawesome-
free/css/all.min.css">
    <!-- Ionicons -->
    <!-- <link rel="stylesheet"
href="https://code.ionicframework.com/ionicons/2.0.1/css/ionicons.min.css">
-->
    <!-- Tempusdominus Bootstrap 4 -->

```

```

<link rel="stylesheet" href="<?php echo base_url
?>plugins/tempusdominus-bootstrap-4/css/tempusdominus-bootstrap-4.min.css">
  <!-- DataTables -->
  <link rel="stylesheet" href="<?php echo base_url ?>plugins/datatables-
bs4/css/dataTables.bootstrap4.min.css">
  <link rel="stylesheet" href="<?php echo base_url ?>plugins/datatables-
responsive/css/responsive.bootstrap4.min.css">
  <link rel="stylesheet" href="<?php echo base_url ?>plugins/datatables-
buttons/css/buttons.bootstrap4.min.css">
  <!-- Select2 -->
  <link rel="stylesheet" href="<?php echo base_url
?>plugins/select2/css/select2.min.css">
  <link rel="stylesheet" href="<?php echo base_url ?>plugins/select2-
bootstrap4-theme/select2-bootstrap4.min.css">
  <!-- iCheck -->
  <link rel="stylesheet" href="<?php echo base_url ?>plugins/ichack-
bootstrap/ichack-bootstrap.min.css">
  <!-- JQVMap -->
  <link rel="stylesheet" href="<?php echo base_url
?>plugins/jqvmap/jqvmap.min.css">
  <!-- Theme style -->
  <link rel="stylesheet" href="<?php echo base_url
?>dist/css/adminlte.css">
  <link rel="stylesheet" href="<?php echo base_url
?>dist/css/custom.css">
  <!-- overlayScrollbars -->
  <link rel="stylesheet" href="<?php echo base_url
?>plugins/overlayScrollbars/css/OverlayScrollbars.min.css">
  <!-- Daterange picker -->
  <link rel="stylesheet" href="<?php echo base_url
?>plugins/daterangepicker/daterangepicker.css">
  <!-- summernote -->
  <link rel="stylesheet" href="<?php echo base_url
?>plugins/summernote/summernote-bs4.min.css">
  <!-- SweetAlert2 -->
  <link rel="stylesheet" href="<?php echo base_url ?>plugins/sweetalert2-
theme-bootstrap-4/bootstrap-4.min.css">
  <style type="text/css">/* Chart.js */
    @keyframes chartjs-render-
animation{from{opacity:.99}to{opacity:1}}.chartjs-render-
monitor{animation:chartjs-render-animation 1ms}.chartjs-size-
monitor,.chartjs-size-monitor-expand,.chartjs-size-monitor-
shrink{position:absolute;direction:ltr;left:0;top:0;right:0;bottom:0;overfl
ow:hidden;pointer-events:none;visibility:hidden;z-index:-1}.chartjs-size-
monitor-
expand>div{position:absolute;width:1000000px;height:1000000px;left:0;top:0}
.chartjs-size-monitor-
shrink>div{position:absolute;width:200%;height:200%;left:0;top:0}

```



```

</style>

<!-- jQuery -->
<script src="<?php echo base_url
?>plugins/jquery/jquery.min.js"></script>
<!-- jQuery UI 1.11.4 -->
<script src="<?php echo base_url ?>plugins/jquery-ui/jquery-
ui.min.js"></script>
<!-- SweetAlert2 -->
<script src="<?php echo base_url
?>plugins/sweetalert2/sweetalert2.min.js"></script>
<!-- Toastr -->
<script src="<?php echo base_url
?>plugins/toastr/toastr.min.js"></script>
<script>
    var _base_url_ = '<?php echo base_url ?>';
</script>
<script src="<?php echo base_url ?>dist/js/script.js"></script>

</head>

```

Snip from footer code:

```

<script>
$(document).ready(function(){
    window.viewer_modal = function($src = ''){
        start_loader()
        var t = $src.split('.')
        t = t[1]
        if(t == 'mp4'){
            var view = $("<video src='"+$src+"' controls autoplay></video>")
        }else{
            var view = $("<img src='"+$src+"' />")
        }
        $('#viewer_modal .modal-content video,#viewer_modal .modal-content
img').remove()
        $('#viewer_modal .modal-content').append(view)
        $('#viewer_modal').modal({
            show:true,
            backdrop:'static',
            keyboard:false,
            focus:true
        })
        end_loader()
    }
    window.uni_modal = function($title = ' , $url='', $size=""){

```

```

start_loader()
$.ajax({
  url:$url,
  error:err=>{
    console.log()
    alert("An error occured")
  },
  success:function(resp){
    if(resp){
      $('#uni_modal .modal-title').html($title)
      $('#uni_modal .modal-body').html(resp)
      if($size != ''){
        $('#uni_modal .modal-
dialog').addClass($size+' modal-dialog-centered')
      }else{
        $('#uni_modal .modal-
dialog').removeAttr("class").addClass("modal-dialog modal-md modal-dialog-
centered")
      }
      $('#uni_modal').modal({
        show:true,
        backdrop:'static',
        keyboard:false,
        focus:true
      })
      end_loader()
    }
  }
})
}

window._conf = function($msg='', $func='', $params = []){
  $('#confirm_modal
#confirm').attr('onclick', $func+"("+ $params.join(',')+")")
  $('#confirm_modal .modal-body').html($msg)
  $('#confirm_modal').modal('show')
}
})
</script>
<footer class="main-footer text-sm">
  <strong>Copyright © <?php echo date('Y') ?>.
  <!-- <a href=""></a> -->
</strong>
  All rights reserved.
  <div class="float-right d-none d-sm-inline-block">
    <b><?php echo $_settings->info('short_name') ?> by: Group D</b>
v1.0
  </div>
</footer>

```

```

</div>
<!-- ./wrapper -->

<script>
    $.widget.bridge('uibutton', $.ui.button)
</script>
<!-- Bootstrap 4 -->
<script src="<?php echo base_url
?>plugins/bootstrap/js/bootstrap.bundle.min.js"></script>
<!-- ChartJS -->
<script src="<?php echo base_url
?>plugins/chart.js/Chart.min.js"></script>
<!-- Sparkline -->
<script src="<?php echo base_url
?>plugins/sparklines/sparkline.js"></script>
<!-- Select2 -->
<script src="<?php echo base_url
?>plugins/select2/js/select2.full.min.js"></script>
<!-- JQVMap -->
<script src="<?php echo base_url
?>plugins/jqvmmap/jquery.vmap.min.js"></script>
<script src="<?php echo base_url
?>plugins/jqvmmap/maps/jquery.vmap.usa.js"></script>
<!-- jQuery Knob Chart -->
<script src="<?php echo base_url ?>plugins/jquery-
knob/jquery.knob.min.js"></script>
<!-- daterangepicker -->
<script src="<?php echo base_url
?>plugins/moment/moment.min.js"></script>
<script src="<?php echo base_url
?>plugins/daterangepicker/daterangepicker.js"></script>
<!-- Tempusdominus Bootstrap 4 -->
<script src="<?php echo base_url ?>plugins/tempusdominus-bootstrap-
4/js/tempusdominus-bootstrap-4.min.js"></script>
<!-- Summernote -->
<script src="<?php echo base_url ?>plugins/summernote/summernote-
bs4.min.js"></script>
<script src="<?php echo base_url
?>plugins/datatables/jquery.dataTables.min.js"></script>
<script src="<?php echo base_url ?>plugins/datatables-
bs4/js/dataTables.bootstrap4.min.js"></script>
<script src="<?php echo base_url ?>plugins/datatables-
responsive/js/dataTables.responsive.min.js"></script>
<script src="<?php echo base_url ?>plugins/datatables-
responsive/js/responsive.bootstrap4.min.js"></script>
<!-- overlayScrollbars -->
<!-- <script src="<?php echo base_url ?>
<script src="<?php echo base_url ?>dist/js/adminlte.js"></script>

```

```

<div class="daterangepicker ltr show-ranges opensright">
  <div class="ranges">
    <ul>
      <li data-range-key="Today">Today</li>
      <li data-range-key="Yesterday">Yesterday</li>
      <li data-range-key="Last 7 Days">Last 7 Days</li>
      <li data-range-key="Last 30 Days">Last 30 Days</li>
      <li data-range-key="This Month">This Month</li>
      <li data-range-key="Last Month">Last Month</li>
      <li data-range-key="Custom Range">Custom Range</li>
    </ul>
  </div>
  <div class="drp-calendar left">
    <div class="calendar-table"></div>
    <div class="calendar-time" style="display: none;"></div>
  </div>
  <div class="drp-calendar right">
    <div class="calendar-table"></div>
    <div class="calendar-time" style="display: none;"></div>
  </div>
  <div class="drp-buttons"><span class="drp-selected"></span><button
class="cancelBtn btn btn-sm btn-default"
type="button">Cancel</button><button class="applyBtn btn btn-sm btn-
primary" disabled="disabled" type="button">Apply</button> </div>
</div>
  <div class="jqvmap-label" style="display: none; left: 1093.83px; top:
394.361px;">Idaho</div>

```

Snip from User profile page:

```

<?php
$user = $conn->query("SELECT * FROM users where id ='".$_settings-
>userdata('id')."'");
foreach($user->fetch_array() as $k =>$v){
    $meta[$k] = $v;
}
?>
<?php if($_settings->chk_flashdata('success')): ?>
<script>
    alert_toast("<?php echo $_settings->flashdata('success') ?>", 'success')
</script>
<?php endif;?>
<div class="card card-outline card-primary">
  <div class="card-body">
    <div class="container-fluid">
      <div id="msg"></div>
      <form action="" id="manage-user">

```

```

        <input type="hidden" name="id" value="<?php echo
$_settings->userdata('id') ?>">
        <div class="form-group">
            <label for="name">First Name</label>
            <input type="text" name="firstname" id="firstname"
class="form-control" value="<?php echo isset($meta['firstname']) ?
$meta['firstname']: '' ?>" required>
        </div>
        <div class="form-group">
            <label for="name">Last Name</label>
            <input type="text" name="lastname" id="lastname"
class="form-control" value="<?php echo isset($meta['lastname']) ?
$meta['lastname']: '' ?>" required>
        </div>
        <div class="form-group">
            <label for="username">Username</label>
            <input type="text" name="username" id="username"
class="form-control" value="<?php echo isset($meta['username']) ?
$meta['username']: '' ?>" required autocomplete="off">
        </div>
        <div class="form-group">
            <label for="password">Password</label>
            <input type="password" name="password" id="password"
class="form-control" value="" autocomplete="off">
            <small><i>Leave this blank if you dont want to change
the password.</i></small>
        </div>
        <div class="form-group">
            <label for="" class="control-label">Avatar</label>
            <div class="custom-file">
                <input type="file" class="custom-file-input rounded-
circle" id="customFile" name="img" onchange="displayImg(this,$(this))">
                <label class="custom-file-label"
for="customFile">Choose file</label>
            </div>
        </div>
        <div class="form-group d-flex justify-content-center">
            
        </div>
    </form>
</div>
<div class="card-footer">
    <div class="col-md-12">
        <div class="row">

```

```

        <button class="btn btn-sm btn-primary" form="manage-
user">Update</button>
    </div>
</div>
</div>
</div>
</div>
<style>
    img#cimg{
        height: 15vh;
        width: 15vh;
        object-fit: cover;
        border-radius: 100% 100%;
    }
</style>
<script>
    function displayImg(input,_this) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#cimg').attr('src', e.target.result);
            }

            reader.readAsDataURL(input.files[0]);
        }
    }

    $('#manage-user').submit(function(e){
        e.preventDefault();
        start_loader()
        $.ajax({
            url: _base_url_+'classes/Users.php?f=save',
            data: new FormData($(this)[0]),
            cache: false,
            contentType: false,
            processData: false,
            method: 'POST',
            type: 'POST',
            success:function(resp){
                if(resp ==1){
                    location.reload()
                }else{
                    $('#msg').html('<div class="alert alert-
danger">Username already exist</div>')
                    end_loader()
                }
            }
        })
    })
</script>

```

Snip from response page:

```
<?php if($_settings->chk_flashdata('success')): ?>
<script>
    alert_toast("<?php echo $_settings->flashdata('success') ?>", 'success')
</script>
<?php endif;?>
<div class="card card-outline card-primary">
    <div class="card-header">
        <h3 class="card-title">List of Questions Responses</h3>
        <div class="card-tools">
            <a href="?page=responses/manage" class="btn btn-flat btn-info"><span class="fas fa-plus"></span> Create New</a>
        </div>
    </div>
    <div class="card-body">
        <div class="container-fluid">
            <table class="table table-bordered table-striped">
                <colgroup>
                    <col width="5%">
                    <col width="20%">
                    <col width="60%">
                    <col width="15%">
                </colgroup>
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Question</th>
                        <th>Response</th>
                        <th>Action</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    $i = 1;
                    $qry = $conn->query("SELECT q.id,
r.response_message, q.question FROM `questions` q inner join `responses` r
on q.response_id = r.id order by q.question asc ");
                    while($row = $qry->fetch_assoc()):
                    ?>
                    <tr>
                        <td class="text-center"><?php echo $i++;
?></td>
                        <td><?php echo $row['question'] ?></td>
                        <td><span class="truncate"><?php echo
$row['response_message'] ?></span></td>
                        <td align="center">
```

```

                <button type="button" class="btn btn-flat
btn-info dropdown-toggle dropdown-icon" data-toggle="dropdown">
                    Action
                <span class="sr-only">Toggle
Dropdown</span>
            </button>
            <div class="dropdown-menu" role="menu">
                <a class="dropdown-item"
href="?page=responses/manage&id=<?php echo $row['id'] ?>"><span class="fa
fa-edit text-primary"></span> Edit</a>
                <div class="dropdown-divider"></div>
                <a class="dropdown-item delete_data"
href="javascript:void(0)" data-id="<?php echo $row['id'] ?>"><span
class="fa fa-trash text-danger"></span> Delete</a>
            </div>
        </td>
    </tr>
</tbody>
</table>
</div>
</div>
<script>
    $(document).ready(function(){
        $('.delete_data').click(function(){
            _conf("Are you sure to delete this
data?", "delete_question", [$(this).attr('data-id')])
        })
        $('.table').dataTable();
    })
    function delete_question($id){
        start_loader();
        $.ajax({
            url:_base_url_+"classes/Master.php?f=delete_response",
            method:"POST",
            data:{id: $id},
            error:err=>{
                console.log(err)
                alert_toast("An error occured.", 'error');
                end_loader();
            },
            success:function(resp){
                if(resp == 1){
                    location.reload();
                }else{
                    alert_toast("An error occured.", 'error');
                    end_loader();
                }
            }
        })
    }
}

```



```

        }
    }
})
}
</script>

```

Snip from response manage page:

```

<style>
    .q-item {
        border-radius: 50px;
    }
</style>
<?php
if(isset($_GET['id']) && !empty($_GET['id'])){
    $qry = $conn->query("SELECT q.id, r.response_message, q.question FROM
`questions` q inner join `responses` r on q.response_id = r.id where q.id =
{$_GET['id']}");
    foreach($qry->fetch_array() as $k => $v){
        if(!is_numeric($k)){
            $$k = $v;
        }
    }
}
?>
<div class="card card-outline card-info">
    <div class="card-header">
        <h3 class="card-title"><?php echo isset($_GET['id']) ? "Update " :
"Create New " ?>Question Response</h3>
    </div>
    <div class="card-body">
        <form action="" id="response-form">
            <input type="hidden" name="id" value="<?php echo
isset($_GET['id']) ? $_GET['id'] : '' ?>">
            <?php if(!isset($id)): ?>
            <div class="form-group">
                <label for="question_field" class="control-
label">Question</label>
                <div class="input-group col-lg-6">
                    <input type="text" id="question" class="form-control
form-control-sm" data-original-title="" title="">

                    <div class="input-group-append">
                        <a href="javascript:void(0)" class="input-group-
text bg-primary" id="add_question"><i class="fas fa-plus"></i></a>
                    </div>
                </div>
            </div>

```

```

        <small><i>(Click the "<b>+</b>" to add the
question)</i></small>
    </div>
    <?php else: ?>
        <div class="form-group">
            <label for="question_field" class="control-
label">Question</label>
            <input type="text" id="question_field"
name="question[]" value="<?php echo isset($question) ? $question : '' ?>"
class="form-control form-control-sm" required>
            </div>
        <?php endif; ?>
        <div class="form-group">
            <div class="container-fluid" id="question-holder">
                <?php

                if(isset($_GET['q']) && !empty($_GET['q'])) {
                    ?>
                    <span class="badge badge-primary q-item m-2 p1-2">
                        <span style='font-size:12px'><?php echo $_GET['q']
?><input type='hidden' name='question[]' value='<?php echo $_GET['q']
?>'></span>
                        <span class='p-2'><a href='javascript:void(0)'
onclick="$(this).closest('.q-item').remove();" class='text-white'><i
class='fa fa-times'></i></a></span></span>
                    </span>
                    <script>
                        $(document).ready(function(){
                            $('#response_message').focus();
                        })
                    </script>
                <?php
                }

                ?>

            </div>
        </div>
        <div class="form-group">
            <label for="response_message" class="control-
label">Response Message</label>
            <textarea name="response_message" id="response_message"
cols="30" rows="3" class="form-control" style="resize: none" required><?php
echo isset($response_message) ? $response_message : ''; ?></textarea>
            </div>
        </form>
    </div>
    <div class="card-footer">

```

```

        <button class="btn btn-flat btn-primary" form="response-
form">Save</button>
        <a class="btn btn-flat btn-default"
href="?page=responses">Cancel</a>
    </div>
</div>
<script>
    $(document).ready(function(){
        function add_question(){
            var question = $('#question').val()
            if(question == '' || question == null)
                return false;
            var span = $('<span class="badge badge-primary q-item m-2 pl-
2"></span>')
                span.append("<span style='font-size:12px'>"+question+"<input
type='hidden' name='question[]' value='"+question+"'></span>")
                span.append("<span class='p-2'><a href='javascript:void(0)'
onclick=\"$(this).closest('.q-item').remove();\" class='text-white'><i
class='fa fa-times'></i></a></span></span>")
                span.appendTo('#question-holder');
                $('#question').val('')
                $('#question').removeClass("border-danger")
                $('.err-msg').remove();
                $('#question').focus();
            }
        $('#add_question').click(function(){
            add_question()
        })
        $('#question').keypress(function(e){
            if(e.which === 13){
                add_question()
                return false;
            }
        })
        $('#response-form').submit(function(e){
            e.preventDefault();
            $('.err-msg').remove();
            if($('#input[name="question[]"]').length <= 0){
                $('#question').addClass("border-danger")
                $('#question-holder').after("<span class='alert alert-
danger err-msg'><i class='fa fa-exclamation-triangle'></i> You must add
atleast one (1) possible question for the response.</span>")
                $('#question').focus();
                return false;
            }
            start_loader();
            $.ajax({
                url:_base_url_+"classes/Master.php?f=save_response",

```

```

        method:'POST',
        data:$(this).serialize(),
        error:err=>{
            console.log(err)
            alert_toast("An error occured",'error');
            end_loader();
        },
        success:function(resp){
            if(resp == 1){
                location.href = "./?page=responses";
            }else{
                alert_toast("An error occured",'error');
                end_loader();
            }
        }
    })
}
})
}
</script>

```

Snip from system info page:

```

<?php if($_settings->chk_flashdata('success')): ?>
<script>
    alert_toast("<?php echo $_settings->flashdata('success') ?>",'success')
</script>
<?php endif;?>

<style>
    img#cimg,img#cimg2,img#cimg3{
        height: 15vh;
        width: 15vh;
        object-fit: cover;
        border-radius: 100% 100%;
    }
</style>
<div class="col-lg-12">
    <div class="card card-outline card-primary">
        <div class="card-header">
            <h5 class="card-title">System Information</h5>
        </div>
        <div class="card-body">
            <form action="" id="system-frm">
                <div id="msg" class="form-group"></div>
                <div class="form-group">
                    <label for="name" class="control-label">System
Name</label>

```

```

        <input type="text" class="form-control form-control-sm"
name="name" id="name" value="<?php echo $_settings->info('name') ?>">
    </div>
    <div class="form-group">
        <label for="short_name" class="control-label">System
Short Name</label>
        <input type="text" class="form-control form-control-sm"
name="short_name" id="short_name" value="<?php echo $_settings-
>info('short_name') ?>">
    </div>
    <div class="form-group">
        <label for="" class="control-label">Introduction
Message</label>
        <textarea name="intro" id="" cols="30" rows="2"
class="form-control"><?php echo $_settings->info('intro') ?></textarea>
    </div>
    <div class="form-group">
        <label for="" class="control-label">No Result
Message</label>
        <textarea name="no_result" id="" cols="30" rows="2"
class="form-control"><?php echo $_settings->info('no_result')
?></textarea>
    </div>
    <div class="form-group">
        <label for="" class="control-label">System Logo</label>
        <div class="custom-file">
            <input type="file" class="custom-file-input rounded-
circle" id="customFile" name="img" onchange="displayImg(this,$(this))">
            <label class="custom-file-label" for="customFile">Choose
file</label>
        </div>
    </div>
    <div class="form-group d-flex justify-content-center">
        
    </div>
    <div class="form-group">
        <label for="" class="control-label">Bot Avatar</label>
        <div class="custom-file">
            <input type="file" class="custom-file-input rounded-
circle" id="customFile" name="bot_avatar"
onchange="displayImg2(this,$(this))">
            <label class="custom-file-label" for="customFile">Choose
file</label>
        </div>
    </div>
    <div class="form-group d-flex justify-content-center">

```

```

        
    </div>
    <div class="form-group">
        <label for="" class="control-label">User Avatar</label>
        <div class="custom-file">
            <input type="file" class="custom-file-input rounded-
circle" id="customFile" name="user_avatar"
onchange="displayImg3(this,$(this))">
            <label class="custom-file-label" for="customFile">Choose
file</label>
        </div>
    </div>
    <div class="form-group d-flex justify-content-center">
        
    </div>

</form>
</div>
<div class="card-footer">
    <div class="col-md-12">
        <div class="row">
            <button class="btn btn-sm btn-primary" form="system-
frm">Update</button>
        </div>
    </div>
</div>
</div>
</div>
<script>
    function displayImg(input,_this) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#cimg').attr('src', e.target.result);
                _this.siblings('.custom-file-
label').html(input.files[0].name)
            }

            reader.readAsDataURL(input.files[0]);
        }
    }
    function displayImg2(input,_this) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();

```

```

        reader.onload = function (e) {
            _this.siblings('.custom-file-
label').html(input.files[0].name)
            $('#cimg2').attr('src', e.target.result);
        }

        reader.readAsDataURL(input.files[0]);
    }
}
function displayImg3(input,_this) {
    if (input.files && input.files[0]) {
        var reader = new FileReader();
        reader.onload = function (e) {
            _this.siblings('.custom-file-
label').html(input.files[0].name)
            $('#cimg3').attr('src', e.target.result);
        }

        reader.readAsDataURL(input.files[0]);
    }
}
$(document).ready(function(){
    $('.summernote').summernote({
        height: 200,
        toolbar: [
            [ 'style', [ 'style' ] ],
            [ 'font', [ 'bold', 'italic', 'underline',
'strikethrough', 'superscript', 'subscript', 'clear' ] ],
            [ 'fontname', [ 'fontname' ] ],
            [ 'fontsize', [ 'fontsize' ] ],
            [ 'color', [ 'color' ] ],
            [ 'para', [ 'ol', 'ul', 'paragraph', 'height' ] ],
            [ 'table', [ 'table' ] ],
            [ 'view', [ 'undo', 'redo', 'fullscreen', 'codeview',
'help' ] ]
        ]
    })
})
</script>

```

Snip from unanswered list:

```

<?php if($_settings->chk_flashdata('success')): ?>
<script>
    alert_toast("<?php echo $_settings->flashdata('success') ?>", 'success')
</script>
<?php endif;?>
<div class="card card-outline card-primary">

```

```

<div class="card-header">
  <h3 class="card-title">List of Unanswered Questions</h3>
</div>
<div class="card-body">
  <div class="container-fluid">
    <table class="table table-bordered table-striped">
      <colgroup>
        <col width="15%">
        <col width="40%">
        <col width="25%">
        <col width="20%">
      </colgroup>
      <thead>
        <tr>
          <th>#</th>
          <th>Question</th>
          <!-- <th>Total Who Asks</th> -->
          <th>Action</th>
        </tr>
      </thead>
      <tbody>
        <?php
          $i = 1;
          $qry = $conn->query("SELECT * FROM
`unanswered` order by question asc ");
          while($row = $qry->fetch_assoc()):
            ?>
              <tr>
                <td class="text-center"><?php echo $i++;
?></td>
                <td><?php echo $row['question'] ?></td>
                <td align="center">
                  <button type="button" class="btn btn-flat
btn-info dropdown-toggle dropdown-icon" data-toggle="dropdown">
                    Action
                    <span class="sr-only">Toggle
Dropdown</span>
                  </button>
                  <div class="dropdown-menu" role="menu">
                    <a class="dropdown-item"
href="?page=responses/manage&q=<?php echo $row['question'] ?>"><span
class="fa fa-edit text-primary"></span> Create Response</a>
                    <div class="dropdown-divider"></div>
                    <a class="dropdown-item delete_data"
href="javascript:void(0)" data-id="<?php echo $row['id'] ?>"><span
class="fa fa-trash text-danger"></span> Delete</a>
                  </div>
                </td>
              </tr>
            ?>
          </while>
        </tbody>
      </table>
    </div>
  </div>

```



```

        </tr>
        <?php endwhile; ?>
    </tbody>
</table>
</div>
</div>
</div>
<script>
    $(document).ready(function(){
        $('.delete_data').click(function(){
            _conf("Are you sure to delete this
data?", "delete_question", [$(this).attr('data-id')])
        })
        $('.table').dataTable();
    })
    function delete_question($id){
        start_loader();
        $.ajax({
            url:_base_url_+"classes/Master.php?f=delete_unanswer",
            method:"POST",
            data:{id: $id},
            error:err=>{
                console.log(err)
                alert_toast("An error occured.", 'error');
                end_loader();
            },
            success:function(resp){
                if(resp == 1){
                    location.reload();
                }else{
                    alert_toast("An error occured.", 'error');
                    end_loader();
                }
            }
        })
    }
</script>

```

CHAPTER 9

CONCLUSION

9.1 Summary of Achievements

The chapter summarizes the project's achievements, emphasizing the successful development of the Simple ChatBot Application. Digitization is transforming society into a “mobile-first” population. As messaging applications grow in popularity, chatbots are increasingly playing an important role in this mobility-driven transformation. Intelligent conversational chatbots are often interfaces for mobile applications and are changing the way businesses and customers interact.

9.2 Contributions to the Field

- **Innovative Technology Integration:**
Describe how your Chatbot System integrates cutting-edge technologies or techniques that were not commonly used in similar systems before. For example, incorporating advanced natural language processing (NLP) algorithms or leveraging emerging AI models.
- **Unique Problem-Solving:**
Highlight how your project addresses a unique or previously unsolved problem within the chatbot domain. Explain the significance of solving this problem and its potential impact on the industry.
- **Enhanced User Experience:**
Discuss how your chatbot system improves the user experience compared to existing solutions. This could include features such as better user interfaces, more accurate responses, or improved conversation flow.
- **Scalability and Adaptability:**
If your system introduces novel approaches to scalability or adaptability, detail how these innovations benefit chatbot deployment in various scenarios. For instance, explain how your chatbot can easily adapt to different industries or languages.
- **Efficiency and Cost Reduction:**
If your project introduces efficiency gains or cost reductions, provide data or examples to illustrate these improvements. This could involve reducing the time and resources required to train and deploy chatbots.
- **Open-Source Contributions:**

If applicable, discuss any open-source components or libraries developed as part of your project. Open-sourcing components can contribute to the broader chatbot development community.

- **User-Centric Design:**
Explain how your system incorporates user feedback and user-centered design principles to enhance usability. Discuss how this approach can be a model for creating user-friendly chatbot systems.
- **Cross-Domain Applicability:**
If your Chatbot System is designed to work across different industries or domains, highlight how its adaptability contributes to broader applicability and usefulness.
- **Performance Benchmarks:**
If your system achieves remarkable performance in terms of response times, accuracy, or resource efficiency, provide comparative benchmarks with other chatbot solutions to showcase its superiority.
- **Research Contribution:**
If your project involved novel research, such as the development of new algorithms or models, describe how your findings contribute to the academic and research aspects of chatbot technology.
- **Documentation and Knowledge Sharing:**
Emphasize the value of thorough documentation and knowledge sharing within your project. If you have created tutorials, best practices, or guidelines for chatbot development, explain how they can benefit the community.
- **Collaboration and Partnerships:**
If your project involved collaborations with industry partners, academic institutions, or other organizations, highlight the synergy and knowledge exchange that occurred as a result.

9.3 Future Work

In the near future, when AI is combined with the development of 5G technology, businesses, employees, and consumers are likely to enjoy enhanced chatbot features such as faster recommendations and predictions, and easy access to high-definition video conferencing from within a conversation. These and other possibilities are in the investigative stages and will evolve quickly as internet connectivity, AI, NLP, and ML advance. Eventually, every person can have a fully functional personal assistant right in their pocket, making our world a more efficient and connected place to live and work.

REFERENCES

Link:

1. <https://www.techtarget.com/searchcustomerexperience/definition/chatbot#:~:text=A%20chatbot%20is%20a%20software,assistants%20to%20handle%20simple%20tasks.>
2. <https://www.w3schools.com/html/>
3. <https://www.w3schools.com/css/default.asp>
4. <https://www.w3schools.com/js/default.asp>
5. https://www.w3schools.com/bootstrap/bootstrap_ver.asp
6. <https://www.w3schools.com/php/default.asp>
7. <https://www.w3schools.com/sql/default.asp>
8. <https://www.w3schools.com/mysql/default.asp>
9. <https://datatables.net/>
10. <https://adminlte.io/>
11. <https://stackoverflow.com/>