A

Industrial-Oriented Mini Project

On

**MUMMYMATE – YOUR AI CHATBOT**

(Submitted in partial fulfilment of the requirements for the award of Degree)

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

By

K. Naga Shreeya (227R1A0525)

M. Varun Sai Varma (227R1A0536)

Under the Guidance of

**Ms. S. Sanjana**

(Assistant Professor)



## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**CMR TECHNICAL CAMPUS**

**UGC AUTONOMOUS**

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

**May, 2025.**

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**CERTIFICATE**

This is to certify that the project entitled “**MummyMate – Your AI Chatbot**” being submitted by **K. Naga Shreeya (227R1A0525), M. Varun Sai Varma (227R1A0536)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

#### **Ms. S. Sanjana Dr. Nuthanakanti Bhaskar**

#### **Assistant Professor HoD**

**INTERNAL GUIDE**

**Dr. A. Raji Reddy Signature of External Examiner**

**DIRECTOR**

**Submitted for viva voice Examination held on**

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Ms. S. Sanjana,** Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark.

We take this opportunity to extend our heartfelt appreciation to the Project Review Committee (PRC) Coordinators—**Y Varalaxmi, B Sekhar, D Nageswar Rao, and SVSV Prasad Sanaboina**—for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

Our sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of our project.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

We also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, we sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

**K. Naga Shreeya (227R1A0525)**

**M. Varun Sai Varma (227R1A0536)**

**INSTITUTE VISION:**

To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

**INSTITUTE MISSION:**

1. To create state of art facilities for effective Teaching- Learning Process.
2. Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.
3. Infuse Professional, Ethical and Societal values among Learning Community.

**DEPARTMENT VISION:**

To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

**DEPARTMENT MISSION:**

1. To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.
2. To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student’s practical knowledge.
3. To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

MummyMate is an advanced AI-based Pregbot system developed to provide intelligent support to women and their families throughout the pregnancy journey. This innovative system is designed to understand and respond to a wide variety of user needs by integrating cutting-edge technologies such as Machine Learning (ML) and Natural Language Processing (NLP). The primary objective of this project is to evaluate existing Pregbot technologies and platforms while creating a robust and user-friendly solution that serves as an exemplary model for future developments in this domain. Pregbots, or pregnancy chatbots, are specialized automated conversational systems designed to assist pregnant individuals by offering accurate, timely, and relevant information. These bots serve various functions, such as answering common questions about pregnancy, offering health tips, and even providing emotional support through constant availability. With the rise of AI and conversational interfaces, Pregbots have emerged as vital digital companions, especially for users seeking quick information or guidance without scheduling in-person consultations. MummyMate enhances the traditional Pregbot model by incorporating deep learning techniques that significantly improve the accuracy and contextual understanding of user queries. Unlike earlier NLP-based bots that often relied on keyword matching and simple pattern recognition, MummyMate uses Python-based deep learning neural networks trained on a rich dataset of frequently asked questions and medically accurate answers. This allows the system to predict and deliver precise responses even for complex and previously unseen questions.

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **FIGURE NAME** | **PAGE NO** |
| Figure 3.1 | System Architecture of MummyMate – Your AI Chatbot | 11 |
| Figure 3.3.1 | Use Case Diagram of MummyMate – Your AI Chatbot | 14 |
| Figure 3.3.2 | Class Diagram of MummyMate – Your AI Chatbot | 15 |
| Figure 3.3.3 | Sequence Diagram of MummyMate – Your AI Chatbot | 16 |
| Figure 3.3.4 | Collaboration Diagram of MummyMate – Your AI Chatbot | 16 |
| Figure 5.1 | Index page of MummyMate – Your AI Chatbot | 31 |
| Figure 5.2 | Registration page of MummyMate – Your AI Chatbot | 32 |

|  |  |  |
| --- | --- | --- |
| Figure 5.3 | Login page of MummyMate – Your AI Chatbot | 33 |
| Figure 5.4 | Health tracking page of MummyMate – Your AI Chatbot | 33 |
| Figure 5.5 | Chatbot page of MummyMate – Your AI Chatbot | 34 |
| Figure 5.6 | Family planning page of MummyMate – Your AI Chatbot | 35 |
| Figure 5.7 | Our features page of MummyMate – Your AI Chatbot | 35 |
| Figure 5.8 | Pregnancy support & care page of MummyMate – Your AI Chatbot | 36 |

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TABLE NAME** | **PAGE NO** |
| Table 6.2.1 | Uploading Data To Database | 41 |
| Table 6.2.2 | Updated database Table | 42 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ABSTRACT** | |  |  | i |
| **LIST OF FIGURES** | | |  | ii |
| **LIST OF TABLES** | | | | v |
| **1.** | **INTRODUCTION** | | | 1 |
| **2.** | **LITERATURE SURVEY** | | | 4 |
|  | 2.1 | EXISTING SYSTEM | | 6 |
|  | 2.2 | PROBLEM STATEMENT | | 7 |
|  | 2.3 | HARDWARE & SOFTWARE REQUIREMENTS | | 9 |
|  |  | 2.3.1 | HARDWARE REQUIREMENTS | 9 |
|  |  | 2.3.2 | SOFTWARE REQUIREMENTS | 9 |
| **3.** | **SYSTEM ARCHITECTURE & DESIGN** | | | 11 |
|  | 3.1 | SYSTEM ARCHITECTURE | | 11 |
|  | 3.2 | SYSTEM STUDY | | 11 |
|  | 3.3 | SYSTEM DESIGN | | 13 |
|  | 3.4 | SOFTWARE ENVIRONMENT | | 17 |
| **4.** | **IMPLEMENTATION** | | | 24 |
|  | 4.1 | ALGORITHMS USED | | 24 |
|  | 4.2 | SAMPLE CODE | | 26 |
| **5.** | **RESULTS & DISCUSSION** | | | 31 |
| **6.** | **VALIDATION** | | | 38 |
|  | 6.1 | INTRODUCTION | | 38 |
|  | 6.2 | TEST CASES | | 38 |
|  |  | 6.2.1 | UPLOADING DATA TO DATABSE | 41 |
|  |  | 6.2.2 | UPDATED DATABASE TABLE | 42 |
| **7.** | **CONCLUSION & FUTURE ASPECTS** | | | 44 |
| **8.** | **BIBLIOGRAPHY** | | | 46 |
|  | 8.1 | REFERENCES | | 46 |
|  | 8.2 | GITHUB LINK | | 46 |

1. **INTRODUCTION**

**1. INTRODUCTION**

### 

Pregnancy is a transformative phase in the lives of women and their families, filled with excitement, anticipation, and challenges. During this period, women often require support and reliable information to navigate the physical and emotional changes that occur. However, accessing personalized advice and real-time support can be difficult due to various barriers, including limited access to healthcare professionals, the overwhelming volume of information, and the uniqueness of each pregnancy experience. The need for a system that can provide immediate, personalized, and empathetic care has never been more apparent.

The rise of digital health technologies, particularly those powered by artificial intelligence (AI), has opened new possibilities for addressing these gaps in pregnancy support. Machine learning (ML) and natural language processing (NLP) are at the forefront of this transformation, offering the ability to create intelligent systems capable of understanding, learning, and responding to users' needs. These technologies not only provide accurate information but also allow for interaction in a natural, conversational manner, making healthcare more accessible and user-friendly.

MummyMate is an innovative system that leverages ML and NLP to offer personalized care and guidance for women and families throughout the pregnancy journey. By integrating these technologies, MummyMate can engage in meaningful conversations with users, providing real-time advice, reminders, and emotional support. This system addresses a wide range of pregnancy-related topics, from tracking fetal development and health tips to answering questions about prenatal care. By using ML, MummyMate continuously learns from user interactions, refining its advice based on individual experiences and concerns.

One of the key advantages of MummyMate is its ability to adapt to the specific needs of each user. Every pregnancy is unique, and generalized information may not always be applicable to every situation. Through ML, MummyMate can analyze the data provided by users—such as their medical history, lifestyle choices, and preferences—and tailor its responses accordingly. This level of personalization ensures that women receive advice that is relevant to their specific circumstances, helping them make informed decisions about their

health and well-being.

In addition to personalized care, MummyMate also offers emotional support, a critical component often overlooked in traditional healthcare systems. Pregnancy can be an emotional rollercoaster, and women may experience anxiety, stress, and uncertainty. Through NLP, MummyMate is designed to recognize emotional cues and respond empathetically, offering reassurance and understanding. This human-like interaction fosters a sense of connection and comfort, making MummyMate a valuable companion for expectant mothers as they navigate the complexities of pregnancy.

1. **LITERATURE SURVEY**

* + 1. **LITERATURE SURVEY**

Here is a literature survey of 10 relevant works, including their titles, authors, and descriptions, that pertain to the use of machine learning (ML) and natural language processing (NLP) in healthcare and pregnancy support systems:

1. Title: "*Artificial Intelligence in Healthcare: Transforming the Practice of Medicine*"

• Authors: Eric Topol

• Description: This work provides a broad overview of how AI technologies, including ML and NLP, are revolutionizing healthcare. It discusses various applications, such as patient management systems, personalized medicine, and AI-driven diagnostics, relevant to pregnancy care systems like MummyMate.

1. Title: "*Machine Learning in Healthcare: A Review of Algorithms and Applications*"

• Authors: Mihaela van der Schaar, Ziad Obermeyer

• Description: The paper discusses various machine learning algorithms and their applications in healthcare, highlighting potential areas for improving predictive analysis and patient engagement in pregnancy care systems.

1. Title: "*NLP in Healthcare and Biomedical Text Mining: Applications, Tools, and Challenges*"

• Authors: Jian Tang, Zhiyong Lu

• Description: This paper explores the role of NLP in extracting and interpreting healthcare data, including medical records, patient interactions, and literature. The techniques discussed could support pregnancy-related chatbot systems in understanding and responding to user queries.

1. Title: "*A Survey on Chatbot Implementation in Healthcare*"

• Authors: Sameera A. Abdul-Kader, John Woods

• Description: A comprehensive review of chatbot technologies and their application in healthcare. It discusses how NLP-based chatbots can improve patient support systems, including pregnancy-related use cases.

1. Title: "*Personal Health Monitoring Using Machine Learning and Wearable Sensors*"

• Authors: Min Chen, Yixue Hao

• Description: This study highlights the use of ML and wearable technologies in personal health monitoring. It provides insights into how pregnancy tracking systems can be developed using similar techniques to track vital signs and provide real-time feedback.

1. Title: "*AI and Pregnancy: Opportunities and Risks*"

• Authors: Valerie Floris, Barbara Holzinger

• Description: The paper specifically explores the application of AI in supporting pregnancy, identifying key opportunities for improving maternal care, but also addresses ethical concerns and risks associated with reliance on AI systems for pregnancy-related decision-making.

1. Title: "*A Framework for Predicting Pregnancy Complications Using Machine Learning*"

• Authors: Maryam Hosseini, Ali Alavi

• Description: This research introduces an ML-based framework for predicting pregnancy complications using patient data. The system can be integrated with conversational agents like MummyMate to provide personalized risk assessments during pregnancy.

1. Title: "*Intelligent Personal Assistants for Healthcare: A Survey of Existing Systems*"

• Authors: Henry C. Wheadon, Felicity Hall

• Description: The paper surveys various intelligent personal assistants and their effectiveness in healthcare settings. It discusses the architecture and capabilities of systems that could inspire the design of pregnancy support tools.

1. Title: "*Conversational Agents in Healthcare: The Design of Empathetic and Ethical Chatbots*"

• Authors: Sandra Carrillo, Tomás Encinas

• Description: This work focuses on the development of empathetic and ethical chatbots using NLP, a key feature for systems like MummyMate that need to provide emotional support during pregnancy.

1. Title: "*Predictive Analytics in Healthcare: Implications for Personalized Medicine*"

• Authors: Jennifer Liu, Sunil Kumar

• Description: The paper examines how predictive analytics using ML can improve personalized healthcare. It discusses the implications for pregnancy care, especially in predicting individual risks and outcomes, thus making such analytics critical for systems like MummyMate.

**2.1 EXISTING SYSTEM:**

Existing systems for supporting women during pregnancy typically rely on a combination of traditional healthcare services, mobile apps, and online resources. Many women access information through prenatal visits with healthcare professionals, supplemented by educational websites, pregnancy forums, and general health apps. Some mobile apps offer basic pregnancy tracking features, such as fetal development updates, appointment reminders, and health tips. However, these solutions often lack personalization and real-time interaction. Most apps deliver generalized content based on static information rather than tailoring advice to the unique circumstances of each individual pregnancy. Additionally, while some apps include AI-based chatbots, they are often limited in their conversational abilities and struggle to provide emotional support or in-depth responses to complex questions. As a result, many women find themselves seeking additional guidance, feeling overwhelmed by the volume of information available, or frustrated by the lack of personalized care. This highlights the need for more advanced, intelligent systems that can offer customized, empathetic, and real-time support, such as MummyMate.

**DISADVANTAGES:**

The existing systems for supporting women during pregnancy, while helpful, come with several notable disadvantages. First, many apps and online resources provide generalized information, which may not account for the unique circumstances of each woman’s pregnancy.

This lack of personalization can lead to irrelevant advice, making it difficult for users to apply the information to their specific situations. Second, most systems offer limited interactivity. Although some apps include basic chatbots, these often struggle to maintain meaningful, in-depth conversations or address complex concerns, resulting in a superficial user experience.

Another disadvantage is the lack of real-time support. Traditional healthcare services, such as prenatal visits, may not always be available when immediate questions or concerns arise, leaving women without timely guidance. Existing digital systems also tend to rely on static content rather than dynamic, adaptive recommendations. Moreover, these systems generally fall short in providing emotional support. Pregnancy can be an emotionally charged period, and most platforms do not incorporate empathy-driven responses, which are crucial for managing anxiety and stress during this time. Lastly, there is a lack of integration between different tools—apps for tracking health metrics, providing advice, or offering support typically function in isolation, leading to fragmented care rather than a seamless, comprehensive experience. These limitations underscore the need for more advanced, personalized, and responsive systems.

**2.2 PROPOSED SYSTEM:**

The proposed system, MummyMate, is designed to address the limitations of existing pregnancy support platforms by leveraging machine learning (ML) and natural language processing (NLP) to provide personalized, real-time, and empathetic care. Unlike traditional apps that offer generalized advice, MummyMate tailors its recommendations to each user's unique circumstances, including medical history, lifestyle, and preferences, ensuring that the information is relevant and actionable. The system engages users in meaningful, conversational interactions, allowing them to ask questions and receive detailed responses instantly. Through continuous learning powered by ML, MummyMate adapts over time, refining its advice based on user feedback and evolving needs. Additionally, the system provides emotional support by recognizing and responding to emotional cues through NLP, offering reassurance and empathy during stressful moments. By integrating various functionalities, such as pregnancy tracking, health tips, appointment reminders, and emotional support, into a single platform,

MummyMate offers a comprehensive, seamless solution to guide women and their families through the entire pregnancy journey.

**ADVANTAGES OF THE PROPOSED SYSTEM:**

The proposed system, MummyMate, offers several significant advantages over existing pregnancy support platforms. First, its personalization is a key benefit, as MummyMate tailors advice and recommendations based on each user’s medical history, lifestyle, and specific pregnancy needs, providing more relevant and actionable guidance than generalized resources. The system’s use of real-time interaction ensures that women can receive immediate answers to their questions, addressing concerns quickly without waiting for healthcare appointments or relying on static content.

Another major advantage is MummyMate’s emotional support feature, which leverages natural language processing (NLP) to recognize emotional cues and respond empathetically. This allows users to feel understood and supported during stressful or anxious moments, offering comfort that most existing systems lack. Additionally, through machine learning (ML), MummyMate continuously improves its responses by learning from user interactions, making its advice more refined and accurate over time.

MummyMate’s integrated approach is another benefit, as it combines features like pregnancy tracking, health tips, appointment reminders, and emotional support into one cohesive platform, eliminating the need for multiple apps. This seamless integration creates a more comprehensive and efficient experience for users, helping them manage all aspects of pregnancy in one place. Ultimately, MummyMate’s combination of personalization, real-time support, empathy, and comprehensive care sets it apart from traditional pregnancy support systems.

* 1. **HARDWARE & SOFTWARE REQUIREMENTS:**
     1. **HARDWARE REQUIREMENTS:**

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements,

* Processor : Intel Core i3
* Hard disk : 20GB
* RAM : 4GB

### SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

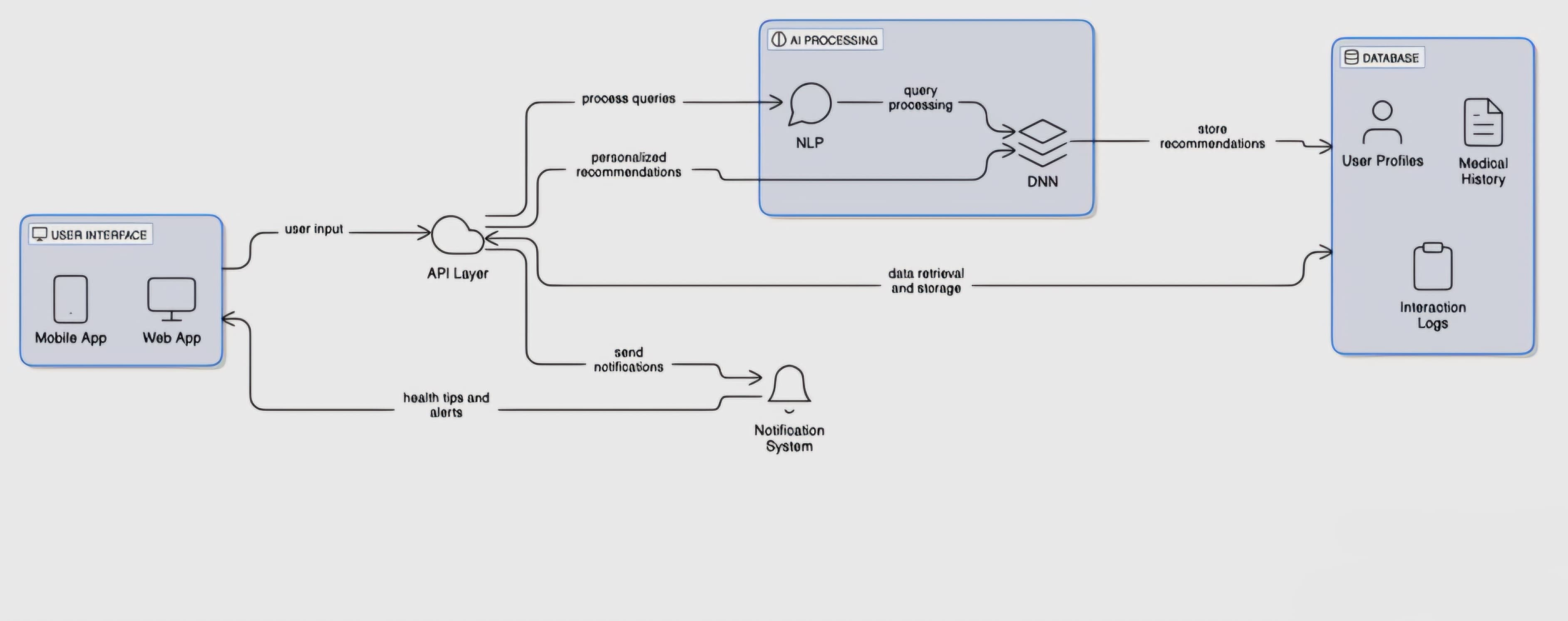
* Operating system : Windows 10
* Language : Python
* Back-End : Django-ORM
* Designing : HTML, CSS, JavaScript
* Database : MYSQL

1. **SYSTEM ARCHITECTURE & DESIGN**

**3. SYSTEM ARCHITECTURE & DESIGN**

**3.1 SYSTEM ARCHITECTURE:**

Project architecture refers to the structural framework and design of a project, encompassing its components, interactions, and overall organization. It provides a clear blueprint for development, ensuring efficiency, scalability, and alignment with project goals. Effective architecture guides the project's lifecycle, from planning to execution, enhancing collaboration and reducing complexity.



**Figure 3.1:** System Architecture Of MUMMYMATE – Your AI Chatbot

**3.2 SYSTEM STUDY:**

**FEASIBILITY STUDY:**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

• ECONOMICAL FEASIBILITY

• TECHNICAL FEASIBILITY

• SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

**TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**3.3 SYSTEM DESIGN**

**UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software project

**GOALS:**

**The Primary goals in the design of the UML are as follows:**

1. **Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.**
2. **Provide extendibility and specialization mechanisms to extend the core concepts.**
3. **Be independent of particular programming languages and development process.**
4. **Provide a formal basis for understanding the modeling language.**
5. **Encourage the growth of OO tools market.**
6. **Support higher level development concepts such as collaborations, frameworks, patterns and components.**
7. **Integrate best practices.**

**USECASE DESCRIPTION :**

**A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.**



**Figure 3.3.1:** Use Case Diagram Of MUMMYMATE – Your AI Chatbot

**CLASS DIAGRAM:**

**In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.**



**Figure 3.3.2:** Class Diagram Of MUMMYMATE – Your AI Chatbot

**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Figure 3.3.3:** Sequence Diagram Of MUMMYMATE – Your AI Chatbot

**COLLABORATION DIAGRAM:**

Collaboration diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**Figure 3.3.4:** Collaboration Diagram Of MUMMYMATE – Your AI Chatbot

**Modules:**

**USER- Upon signing up, you’ll create a profile that helps MummyMate understand your unique journey. By inputting details such as your due date and health concerns, MummyMate can deliver customized advice and information.**

**CHAT WITH CHATBOT-**

* + **Friendly Conversations: Ask me anything about pregnancy, and I’ll provide tailored answers just for you.**
  + **Resources at Your Fingertips: I can share articles, videos, and local resources to help you feel informed and prepared.**
  + **Emotional Support: I’ll check in with you regularly to see how you’re feeling and offer encouragement.**
  + **Health Monitoring: Share any symptoms you’re experiencing, and I’ll guide you with appropriate advice and reminders.**

**REGISTER-**

* + **Personalized Experience: By providing details like your due date and health concerns, MummyMate can offer customized advice, tips, and resources.**
  + **Track Your Journey: Keep track of important milestones and receive reminders for prenatal appointments and health check-ins.**
  + **Emotional Support: Let us understand your unique situation, so we can check in on your well-being and provide the right support.**

**During the registration process, you’ll be asked to provide:**

* + **Basic information (name, email, etc.)**
  + **Your due date and any relevant health information**
  + **Preferences for receiving tips and updates**

**3.4 SOFTWARE ENVIRONMENT**

**What is Python :-**

**Below are some facts about Python.**

**Python is currently the most widely used multi-purpose, high-level programming language.**

**Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.**

**Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.**

**Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.**

**The biggest strength of Python is huge collection of standard library which can be used for the following –**

* + **Machine Learning**
  + **GUI Applications (like Kivy, Tkinter, PyQt etc. )**
  + **Web frameworks like Django (used by YouTube, Instagram, Dropbox)**
  + **Image processing (like Opencv, Pillow)**
  + **Web scraping (like Scrapy, BeautifulSoup, Selenium)**
  + **Test frameworks**
  + **Multimedia**

**Advantages of Python :-**

**Let’s see how Python dominates over other languages.**

**1. Extensive Libraries**

**Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.**

**2. Extensible**

**As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.**

**3. Embeddable**

**Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.**

**4. Improved Productivity**

**The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.**

**5. IOT Opportunities**

**Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.**

**6. Simple and Easy**

**When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.**

**7. Readable**

**Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.**

**8. Object-Oriented**

**This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.**

**9. Free and Open-Source**

**Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.**

**10. Portable**

**When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.**

**11. Interpreted**

**Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.**

**Any doubts till now in the advantages of Python? Mention in the comment section.**

**Advantages of Python Over Other Languages**

**1. Less Coding**

**Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.**

**2. Affordable**

**Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.**

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

**3. Python is for Everyone**

**Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and machine learning, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.**

**Disadvantages of Python**

**So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.**

**1. Speed Limitations**

**We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.**

**2. Weak in Mobile Computing and Browsers**

**While it serves as an excellent server-side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbonnelle. The reason it is not so famous despite the existence of Brython is that it isn’t that secure.**

**3. Design Restrictions**

**As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-typing. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.**

**4. Underdeveloped Database Access Layers**

**Compared to more widely used technologies like JDBC (Java DataBase Connectivity) and ODBC (Open DataBase Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.**

**5. Simple**

**No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.**

**This was all about the Advantages and Disadvantages of Python Programming Language.**

1. **IMPLEMENTATION**

**4. IMPLEMENTATION**

The implementation of MummyMate involves the development of an intelligent chatbot system capable of understanding and responding to user queries related to pregnancy and child care. Built using Python and the Django web framework, MummyMate integrates deep learning and Natural Language Processing (NLP) techniques to enable real-time, human-like interactions. The system is trained on a structured dataset consisting of frequently asked questions and their respective answers, allowing the model to learn from patterns and provide accurate responses to new inputs. The NLTK library is used to process and clean the text data, ensuring that the input to the deep learning model is meaningful and relevant.

**4.1 Algorithms Used in MummyMate**

The core intelligence of MummyMate is powered by several key machine learning and NLP algorithms, including:

1. Tokenization and Lemmatization (NLTK)
2. Bag of Words (BoW)
3. TF-IDF (Term Frequency-Inverse Document Frequency)
4. Deep Neural Networks (DNN)

**1. Tokenization and Lemmatization**

Tokenization is the process of breaking down text into smaller units such as words or phrases. Lemmatization then converts these tokens into their base or dictionary form. For example, the words “running” and “ran” are both converted to “run.” This process reduces redundancy and ensures that the model understands the core meaning of each word. These steps are performed using the NLTK library, which helps in preparing the data for further analysis and training.

**2. Bag of Words (BoW)**

The Bag of Words model converts text into numerical features by counting the frequency of each word in the dataset. It creates a matrix of words where each word is represented as a vector. Though simple, this method is effective for basic text classification tasks and gives the model a way to handle and interpret raw text data.

**3. TF-IDF (Term Frequency-Inverse Document Frequency)**

TF-IDF is a more refined method than BoW, used to evaluate how important a word is to a particular document relative to the entire corpus. It reduces the weight of commonly used words and increases the weight of rare but important ones. This helps MummyMate better understand the relevance of terms in a user's query and improves its ability to find accurate answers.

**4. Deep Neural Networks (DNN)**

The heart of MummyMate’s intelligence lies in its deep neural networks. These consist of multiple layers of artificial neurons that can learn complex patterns in the input data. The neural network is trained on a labeled dataset of question-answer pairs, enabling it to identify relationships between different forms of queries and their appropriate responses. Over time, the model improves its predictions as it is exposed to more data.

**4.2 Sample code**

import nltk

from nltk.stem.lancaster import LancasterStemmer

stemmer = LancasterStemmer()

import numpy

import tflearn

import tensorflow

import random

import json

import pickle

with open("dataset/question.json") as file:

data = json.load(file)

try:

with open("data.pickle", "rb") as f:

words, labels, training, output = pickle.load(f)

except:

words = []

labels = []

docs\_x = []

docs\_y = []

for intent in data["intents"]:

for pattern in intent["patterns"]:

wrds = nltk.word\_tokenize(pattern)

words.extend(wrds)

docs\_x.append(wrds)

docs\_y.append(intent["tag"])

if intent["tag"] not in labels:

labels.append(intent["tag"])

words = [stemmer.stem(w.lower()) for w in words if w != "?"]

words = sorted(list(set(words)))

labels = sorted(labels)

training = []

output = []

out\_empty = [0 for \_ in range(len(labels))]

for x, doc in enumerate(docs\_x):

bag = []

wrds = [stemmer.stem(w.lower()) for w in doc]

for w in words:

if w in wrds:

bag.append(1)

else:

bag.append(0)

output\_row = out\_empty[:]

output\_row[labels.index(docs\_y[x])] = 1

training.append(bag)

output.append(output\_row)

training = numpy.array(training)

output = numpy.array(output)

with open("data.pickle", "wb") as f:

pickle.dump((words, labels, training, output), f)

tensorflow.reset\_default\_graph()

print(str(len(training[0]))+" "+str(len(output[0])))

net = tflearn.input\_data(shape=[None, len(training[0])])

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, 8)

net = tflearn.fully\_connected(net, len(output[0]), activation="softmax")

net = tflearn.regression(net)

model = tflearn.DNN(net)

#try:

# model.load("model/model.tflearn")

#except:

#model.fit(training, output, n\_epoch=1000, batch\_size=8, show\_metric=True)

model.load("model/model.tflearn")

def checkResponse(response,inputs,patterns):

arr = response.lower().split(" ")

status = 0

temp = inputs.lower().split(" ")

for i in range(len(temp)):

if temp[i] in arr:

status = 1

break

if status == 0:

for i in range(len(patterns)):

arr = patterns[i].lower().split(" ")

for j in range(len(temp)):

if temp[j] in arr:

status = 1

j = len(temp)

i = len(patterns)

break

return status

def bag\_of\_words(s, words):

bag = [0 for \_ in range(len(words))]

s\_words = nltk.word\_tokenize(s)

s\_words = [stemmer.stem(word.lower()) for word in s\_words]

for se in s\_words:

for i, w in enumerate(words):

if w == se:

bag[i] = 1

return numpy.array(bag)

print("Start talking with the bot (type quit to stop)!")

while True:

inp = input("You: ")

if inp.lower() == "quit":

break

results = model.predict([bag\_of\_words(inp, words)])

results\_index = numpy.argmax(results)

tag = labels[results\_index]

for tg in data["intents"]:

if tg['tag'] == tag:

responses = tg['responses']

patterns = tg['patterns']

value = random.choice(responses)

status = checkResponse(value,inp,patterns)

if status == 1:

print(value)

else:

print("Sorry! I am not trained to answer above question")

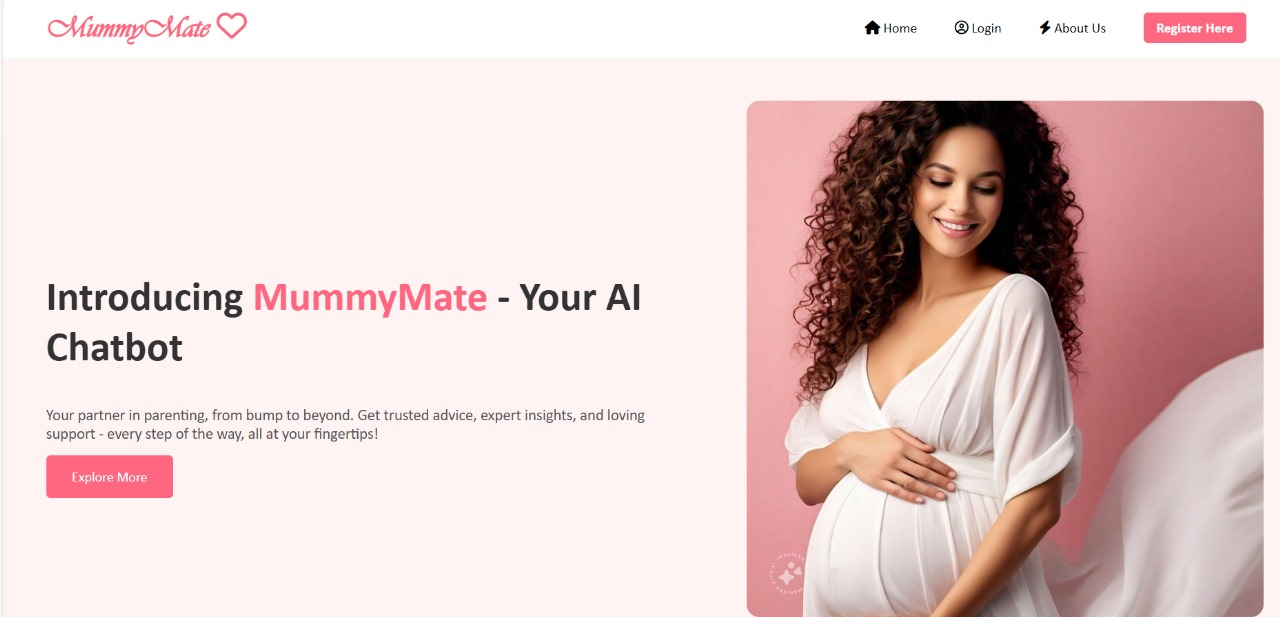
1. **TESTIG AND DEBUGGING/RESULTS**

**5. TESTING AND DEBUGGING/RESULTS**

The following screenshots showcase the results of our project, highlighting its key features and functionalities. These visual representations offer a clear overview of how our system performs under various conditions, effectively demonstrating its usability, accuracy, and intuitive user interface. The screenshots serve as a visual aid to support the technical and operational accomplishments of our project, reinforcing the practical impact of the implemented AI-based MummyMate system.

**5.1 Index page**

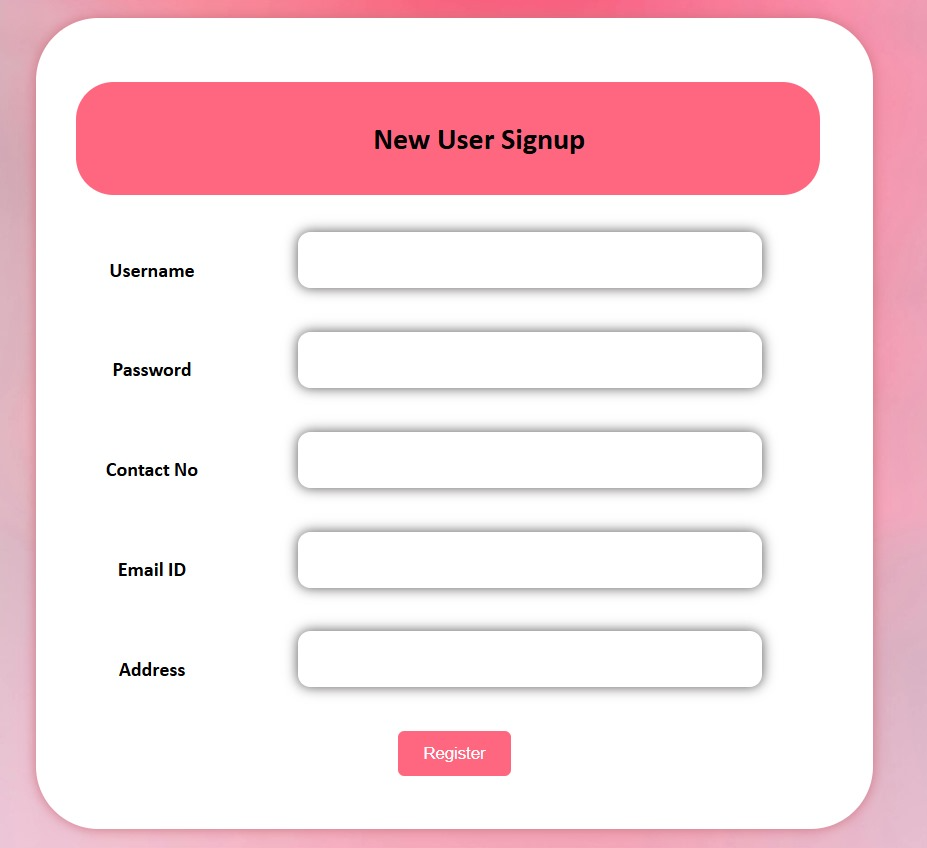
The index page serves as the main entry point of our website, featuring a user-friendly layout with navigation buttons that direct users to various essential sections. These include the Home page for general information, the Login page for existing users to access their accounts, the About Us page to learn more about the purpose and team behind the project, and the Registration page where new users can create an account. This structured navigation ensures easy access to all key areas of the website, enhancing the overall use experience.

****

**Figure 5.1:** Index Page Of MUMMYMATE – Your AI Chatbot

**5.2 Registration page**

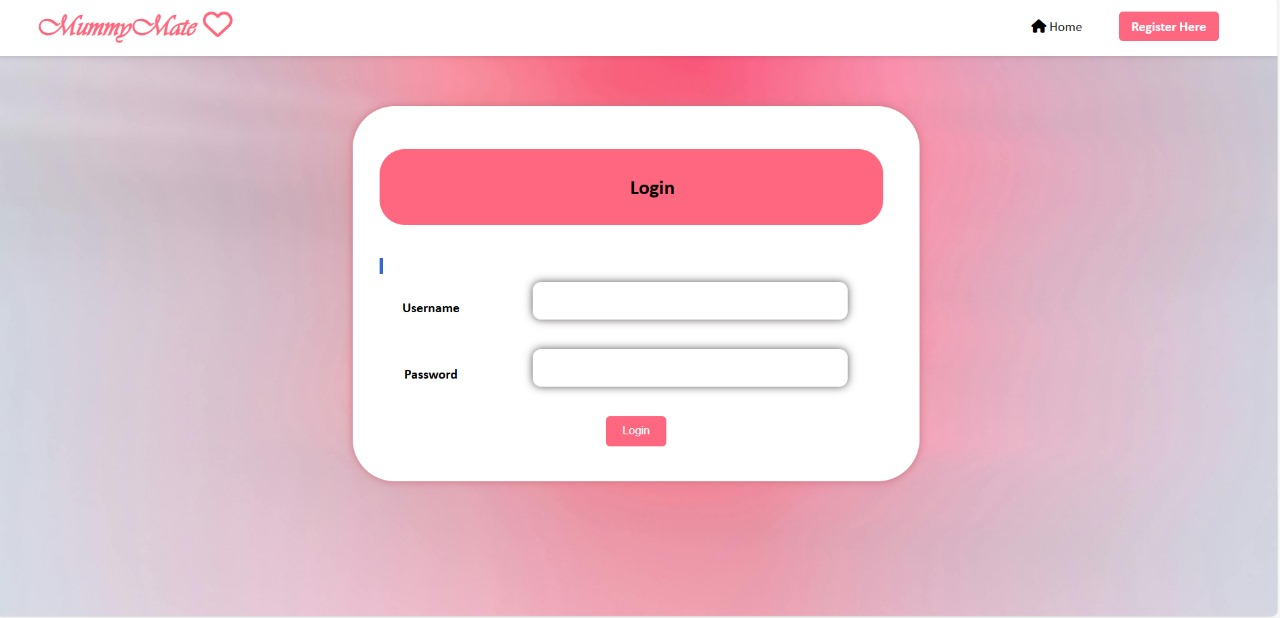
The registration page is designed to collect essential user information required to create a new account on the platform. Users are prompted to fill in details such as their username, password, contact number, email ID, and address. This information is securely stored and used to personalize the user experience. The intuitive layout ensures a smooth registration process, allowing users to quickly sign up and gain access to the services provided by our system.

****

**Figure 5.2:** Registration page of MUMMYMATE – Your AI Chatbot

**5.3 Login page**

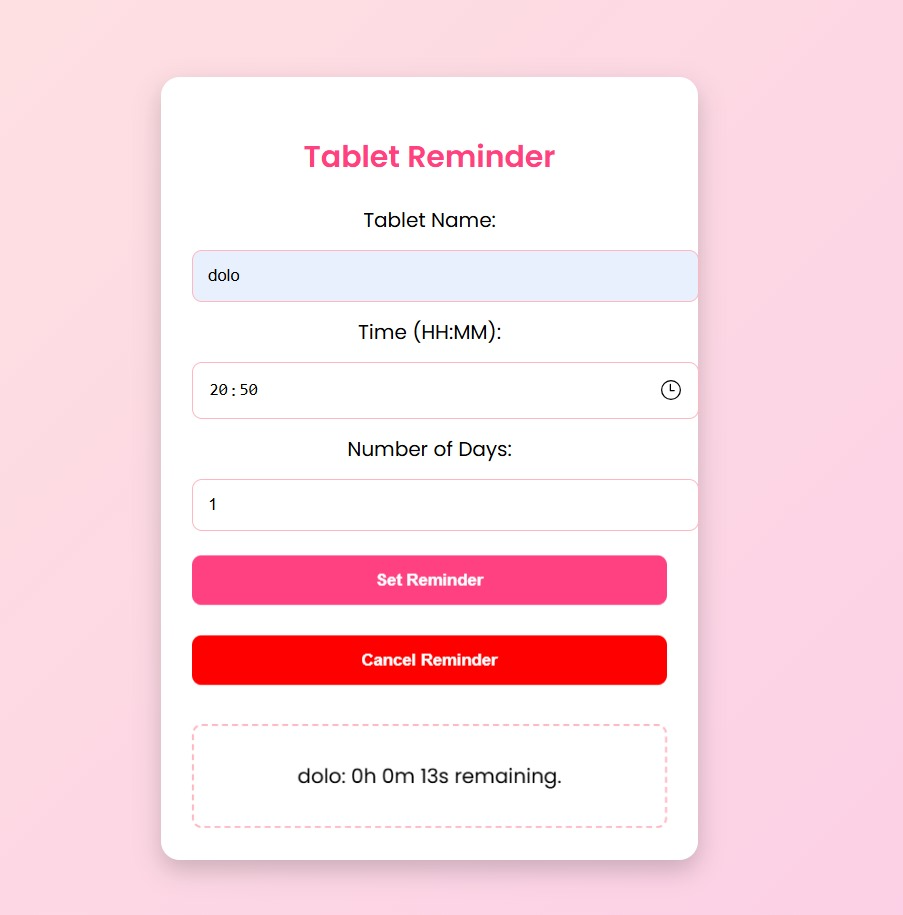
The login page allows registered users to securely access their accounts by entering their username and password. After completing the registration process, users can return to this page to log in and utilize the features and services offered by the system. The login mechanism ensures authorized access, maintaining the privacy and security of user data.

****

**Figure 5.3:** Login Page of MUMMYMATE – Your AI Chatbot

**5.4 Health tracking page**

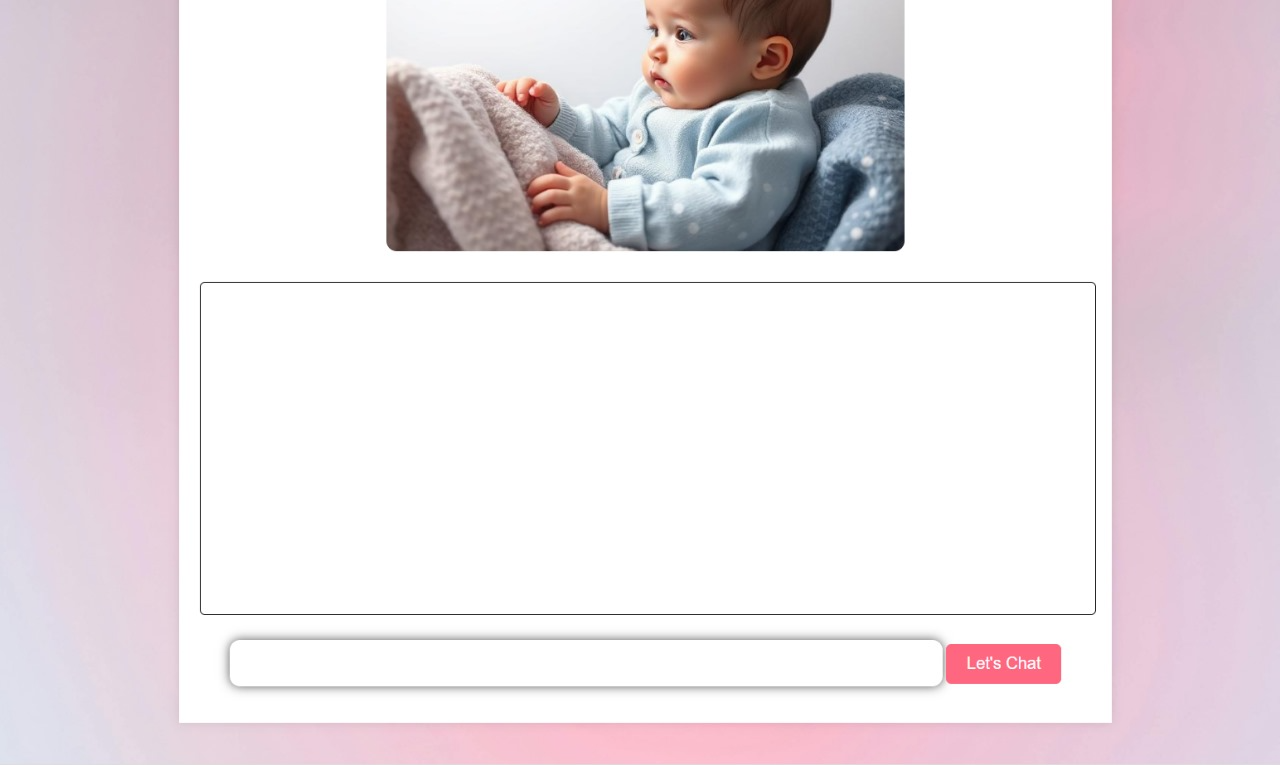
The Health tracking page is designed to help users manage their medication schedules effectively. On this page, users can enter details such as the tablet name, the specific time to take it (in hh:mm format), and the number of days the reminder should be active. Once submitted, the system sets a reminder and triggers an alarm at the scheduled time to ensure timely medication.

****

**Figure 5.4:** Health tracking page of MUMMYMATE – Your AI Chatbot

**5.5 Chatbot page**

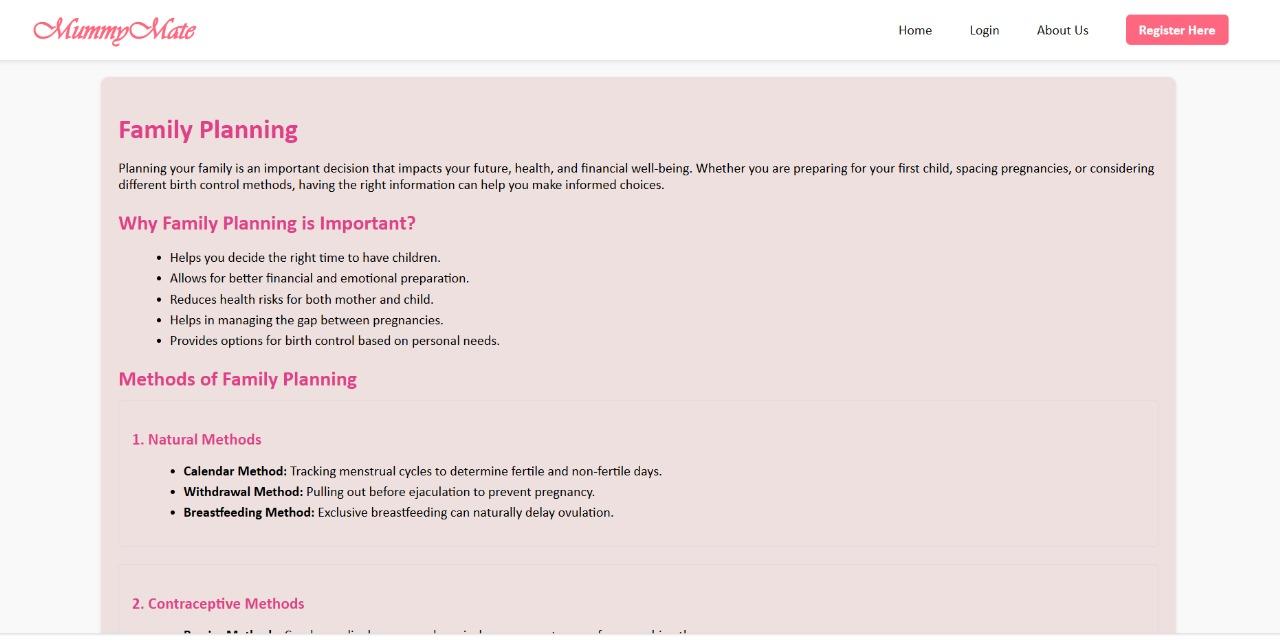
The Chatbot page serves as the interactive hub of our system, where users can ask questions related to pregnancy and receive accurate, AI-generated responses. Powered by advanced machine learning and NLP algorithms, the chatbot understands user queries and provides relevant answers in real time. This feature is designed to offer quick support and guidance on common pregnancy concerns, allowing users to access reliable information in a conversational and user-friendly manner.

****

**Figure 5.5:** Chatbot Page of MUMMYMATE – Your AI Chatbot

**5.6 Family planning page**

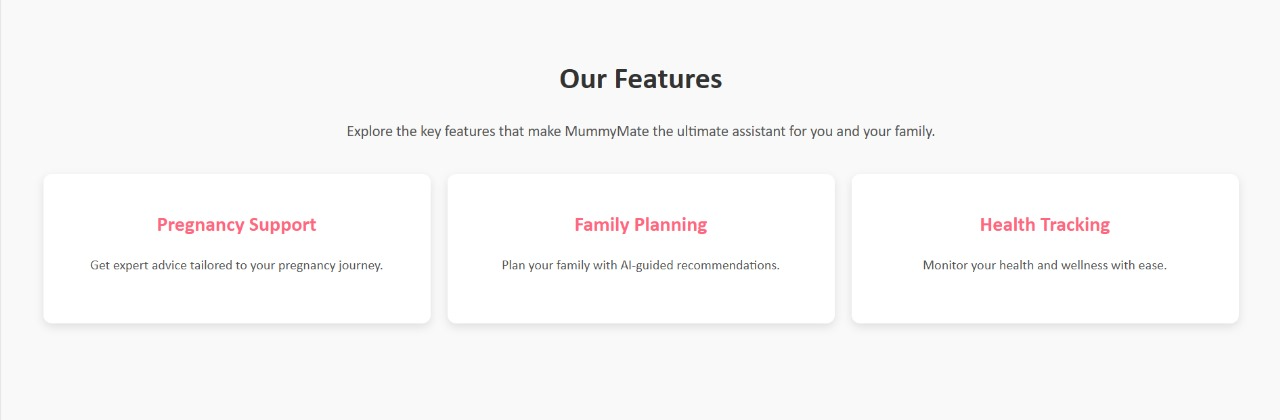
The Family Planning page provides comprehensive information on family planning methods and contraceptive options. It is designed to educate users on various ways to plan their families responsibly and make informed decisions about reproductive health. The content covers different contraceptive methods, their uses, benefits, and considerations, offering valuable guidance to individuals and couples. This page aims to promote awareness and support users in making choices that align with their personal and family goals.

****

**Figure 5.6:** Family Planning Page of MUMMYMATE – Your AI Chatbot

**5.7 Our Features page**

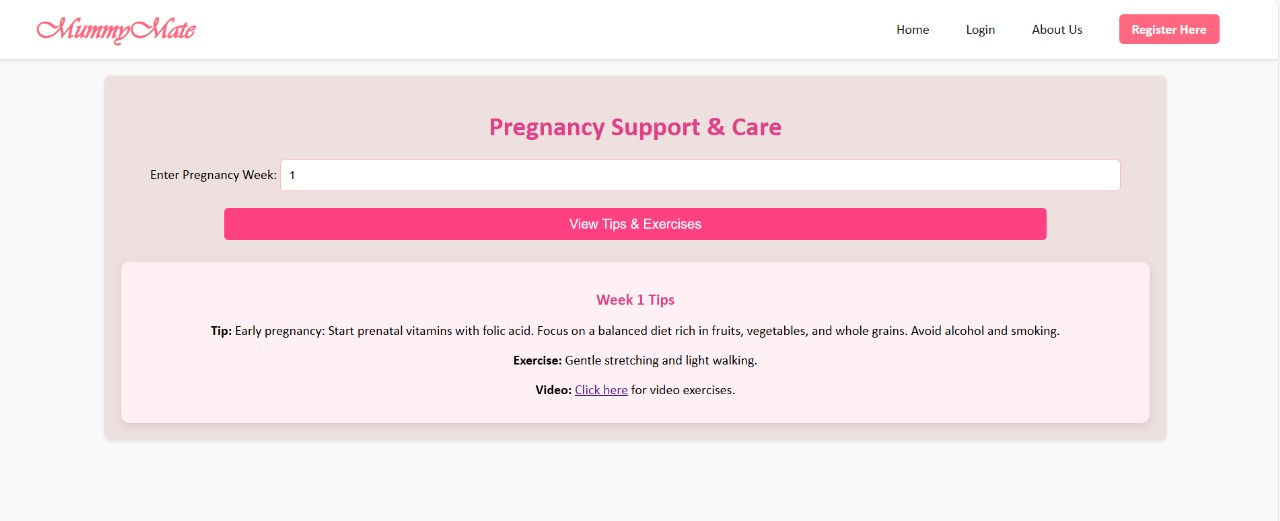
The Our Features page offers a clear overview of the core functionalities of our system, presented through interactive icons for easy navigation. These icons represent key services such as Pregnancy Support, Family Planning, and Health tracking. By clicking on each icon, users are seamlessly redirected to the respective pages, allowing them to access detailed information and tools tailored to their needs. This page serves as a central hub, making it convenient for users to explore and utilize the main features of our platform.

****

**Figure 5.7:** OurFeatures Page of MUMMYMATE – Your AI Chatbot

**5.8 Pregnancy support & care page**

The Pregnancy Support and Care page is designed to provide personalized guidance based on the current week of pregnancy entered by the user. After submitting the pregnancy week, the system generates tailored tips on nutrition, safe exercises, and overall wellness. It also provides video links to recommended exercises to ensure users follow correct and safe practices. This feature aims to support pregnant women throughout their journey by offering expert-backed advice and resources for a healthy and informed pregnancy.

****

**Figure 5.8:** Pregnancy support & care page of MUMMYMATE – Your AI Chatbot

1. **VALIDATION**

**6.VALIDATION**

**6.1 INTRODUCTION**

Validation plays a crucial role in ensuring that MummyMate – Your AI Chatbot functions accurately, securely, and efficiently in a real-world environment. As the chatbot deals with sensitive and personalized data, including pregnancy-related health inputs, symptoms, user preferences, and frequently asked questions, it is essential to validate both the functional behavior of the chatbot and the backend processes, especially database operations.

This phase of the project aims to verify that all components of the system work together seamlessly—from the user interface that collects input, to the logic that processes it, and finally, to the database where the data is stored and retrieved. It ensures that the system not only performs expected tasks but also handles edge cases such as incorrect inputs, unexpected behavior, or simultaneous user access.

For MummyMate, validation involves:

* Confirming that user data is properly captured and uploaded to the database.
* Verifying that previously stored data can be successfully updated.
* Ensuring the chatbot responds dynamically based on the updated information.
* Testing the integrity and security of data storage.

The objective of this phase is to detect any flaws or inconsistencies early, reduce the chances of failure during live use, and enhance the reliability of the system. Effective validation assures that the chatbot will deliver accurate information, tailored advice, and a smooth user experience, particularly in a domain as sensitive as pregnancy and maternal care.

**6.2 TEST CASES**

Test cases form a fundamental part of the validation process in the MummyMate – Your AI Chatbot project. They are designed to ensure that all functionalities of the chatbot—from data handling to navigation and interaction—are working as expected. The test cases are

based on different testing techniques including unit testing, integration testing, functional testing, system testing, black box testing, white box testing, and user acceptance testing.

Each test case is crafted to verify a specific feature or function of the system, validate inputs and outputs, and ensure seamless integration between software components. Below is an overview of how these test types contributed to the development of meaningful and effective test cases.

* **Unit Test Cases** were designed to validate individual modules such as data input validation, response generation, and user registration. These tests ensured that specific components behaved correctly with clearly defined inputs and expected outputs.
* **Integration Test Cases** focused on verifying the interaction between components, such as the chatbot interface and the database. These test cases confirmed that data submitted by users was correctly transmitted, processed, and stored, and that updates reflected accurately across components.
* **Functional Test Cases** were used to systematically check if the features mentioned in business and technical requirements were working correctly. This included validating that the chatbot accepts valid inputs, rejects invalid data, performs its designated functions, and produces appropriate outputs.
* **System Test Cases** ensured that the fully integrated system delivered predictable and accurate results. These test cases covered complete user scenarios, including inputting details, receiving personalized replies, and navigating between pages or features.
* **White Box Test Cases** were applied where knowledge of the internal code structure was required—for instance, testing logical branches for different symptom categories or pregnancy stages.
* **Black Box Test Cases** validated the system without considering the internal code. These test cases involved entering different types of inputs into the chatbot and checking if the correct responses and actions were triggered, such as data uploads or error messages.
* **User Acceptance Test Cases** were created to ensure that the final version of the chatbot met end-user expectations. These included testing if the chatbot responded naturally, offered helpful suggestions, and performed all expected tasks in a user-friendly manner.

The main objectives of these test cases were:

* To ensure that all field entries accept valid data and reject invalid input.
* To verify that no duplicate entries are stored in the database.
* To confirm that all links and navigation options function correctly and take the user to the right pages.
* To test that the chatbot’s response time is fast and not delayed.

All the test cases mentioned were executed, and the results were successful. No critical defects were encountered, and the system performed reliably under all tested scenarios.

**6.2.1 UPLOADING DATA TO DATABASE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input Data** | **Expected Output** | **Testing Type** | **Status** |
| TC\_Upload\_01 | Upload valid user data to database | Name: “Sarah”, Week: “8”, Symptoms: “Nausea” | Data is saved correctly in the appropriate table | Unit Testing, Integration Testing | Passed |
| TC\_Upload\_02 | Submit form with missing required fields | Missing name or week | Error message displayed; data not saved | Functional Testing, Black Box Testing | Passed |
| TC\_Upload\_03 | Upload duplicate entry | Submit same user data twice | System prevents duplication; one record saved | Functional Testing, Integration Testing | Passed |
| TC\_Upload\_04 | Submit data with invalid format | Enter numbers in name field | Validation error displayed; data not saved | Unit Testing, Functional Testing | Passed |
| TC\_Upload\_05 | Large data input test | Long symptom description input | Data saved without truncation or error | System Testing, Black Box Testing | Passed |

**6.2.2 UPDATED DATABASE TABLE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Description** | **Input Data** | **Expected Output** | **Testing Type** | **Status** |
| TC\_Update\_01 | Update pregnancy stage for returning user | Change from "Week 8" to "Week 9" | Existing record updates to "Week 9" without duplication | Integration Testing, Functional Testing | Passed |
| TC\_Update\_02 | Add new symptoms to existing record | Add "Back pain" to previous symptom list | New symptom saved and reflected in chatbot replies | Functional Testing, System Testing | Passed |
| TC\_Update\_03 | Modify personal detail (e.g., name correction) | Change "Srah" to "Sarah" | Name field updates correctly in DB | White Box Testing, Integration Testing | Passed |
| TC\_Update\_04 | Simultaneously update multiple fields | Change week, add symptom, update contact info | All fields updated in the same record without error | System Testing, Functional Testing | Passed |
| TC\_Update\_05 | Attempt invalid update | Enter letters in numeric field | Error message shown, DB remains unchanged | Functional Testing | Passed |

1. **CONCLUSION**

**7.CONCLUSION**

In conclusion, MamaBot marks a significant leap in digital pregnancy support by integrating machine learning (ML) and natural language processing (NLP) to deliver real-time, personalized, and empathetic care for women and families. It addresses major limitations of traditional systems—such as the lack of tailored guidance, emotional support, and cohesive services—by offering a smart, conversational interface that mimics human interaction. This allows users to access accurate, context-aware responses, fostering both trust and convenience throughout the pregnancy journey. Its continuous learning capability enables the system to refine responses over time, enhancing engagement and reliability.

Beyond information delivery, MamaBot contributes to holistic pregnancy care by offering timely reminders, emotional reassurance, and decision-making support. It bridges the gap between clinical consultations and day-to-day concerns, helping reduce anxiety and improve health outcomes. Its scalability and automation also make it accessible and cost-effective, especially in areas with limited healthcare resources. Ultimately, MamaBot empowers users with dependable, compassionate, and user-centric assistance—setting a new benchmark in AI-driven healthcare support.

1. **BIBLEOGRAPHY**

**8. BIBLIOGRAPHY**

**8.1 REFERENCES**

1. Kumar, A., & Sinha, R. (2022). AI in Healthcare: Current Challenges and Future Directions*. International Journal of AI and Healthcare*.
2. Zhang, Y., & Chen, H. (2022). Machine Learning Applications in Personalized Medicine. *Bioinformatics*.
3. Singh, A., & Sharma, V. (2022). Chatbot Systems in Healthcare: Current Trends and Future Prospects*. Health Information Science and Systems*.
4. Srinivasan, S., & Kumar, P. (2022). Innovations in AI for Pregnancy and Postpartum Care. *AI in Healthcare Journal*.
5. Floris, V., & Holzinger, B. (2021). AI and Pregnancy: Opportunities and Risks. *Health Informatics Journal.*
6. Carrillo, S., & Encinas, T. (2021). Conversational Agents in Healthcare: The Design of Empathetic and Ethical Chatbots. *ACM Transactions on Computing for Healthcare*.
7. Van der Schaar, M., & Obermeyer, Z. (2020). Machine Learning in Healthcare: A Review of Algorithms and Applications*. Annual Review of Biomedical Data Science*.
8. Chen, M., & Hao, Y. (2020). Personal Health Monitoring Using Machine Learning and Wearable Sensors*. IEEE Transactions on Biomedical Engineering.*
9. Topol, E. (2019). Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again. *Basic Books*.
10. Tang, J., & Lu, Z. (2019). NLP in Healthcare and Biomedical Text Mining: Applications, Tools, and Challenges. *Journal of Biomedical Informatics*.
11. Liu, J., & Kumar, S. (2019). Predictive Analytics in Healthcare: Implications for Personalized Medicine*. Journal of Healthcare Analytics*.
12. Wheadon, H. C., & Hall, F. (2018). Intelligent Personal Assistants for Healthcare: A Survey of Existing Systems. *Journal of Healthcare Engineering*.

**8.2 GITHUB LINK**