A

Synopsis on

**Chatbot Based Registration System(MUSEUM)**

*in partial fulfillment of the requirement for the degree*

of

Bachelor of Technology In

COMPUTER SCIENCE AND ENGINEERING

Submitted by

**Mridul Kumar Pandey(2301330100126)**

**Navneet Kumar (2301330100132)**

**Muskan Sengar (2301330100129)**

**Prabhakar Lal (2301330100142)**

Under the supervision of **Mrs. Chitvan Agarwal** (Assistant Professor)



# NOIDA INSTITUTE OF ENGINEERING AND TECHNOLOGY GREATER NOIDA

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Topics** | **Page No.** |
| 1. | Introduction | 3-4 |
| 2. | Existing Systems | 5-7 |
| 3. | Problem Statement | 8 |
| 4. | Proposed Methodology | 9-15 |
| 5. | Feasibility Study | 16-17 |
| 6. | Facilities required for proposed work | 18-19 |
| 7. | Conclusion and References | 20-22 |

Supervisor Sign:

**Introduction**

In the digital age, traditional systems of registration and customer engagement are rapidly evolving to embrace automation and interactivity. Museums, as institutions of cultural preservation and education, are no exception to this trend. The growing need for user-friendly, efficient, and interactive visitor management solutions has led to the adoption of AI-powered technologies such as chatbots. This project, titled **"Chatbot-Based Registration System for a Museum,"** aims to develop an intelligent and interactive chatbot interface that facilitates seamless visitor registration and enhances the overall visitor experience.

A **chatbot** is an AI-based software application designed to simulate human conversation. It can communicate with users through text or voice and respond based on predefined rules or AI models. In this project, a chatbot is developed to handle the **registration process** for museum visitors, replacing traditional paper forms or complex web-based registration systems. By integrating the chatbot with a backend database, the system ensures that all visitor information is securely stored and easily retrievable for administrative purposes.

The chatbot-based system is designed to be **interactive**, **responsive**, and **user-friendly**, ensuring that users of all age groups can register for their museum visits with ease. The chatbot asks visitors for essential information such as name, contact number, age, preferred visiting date and time slot, and whether they are visiting individually or in a group. Once the details are collected, the system confirms the registration and provides relevant instructions or QR codes (optional) for verification at the museum entrance.

**Technology Used**

The chatbot-based registration system employs the following technologies:

* **Programming Language:** Python
* **Chatbot Framework:** Dialogflow or Python-based NLP libraries (e.g., ChatterBot, Rasa, or custom logic with NLTK)
* **Frontend Interface:** Web-based (HTML/CSS/JavaScript) or CLI (Command Line Interface) for demo purposes
* **Backend:** Flask (for web integration) or Node.js (optional)
* **Database:** MongoDB / SQLite / Firebase for storing visitor details
* **Natural Language Processing (NLP):** Used to understand and interpret user inputs in natural language
* **Cloud Services (optional):** For deployment (e.g., Heroku, Render, or Google Cloud)

**Field of the Project**

This project falls under the specialized field of **Artificial Intelligence** and more specifically under **Conversational AI** and **Human-Computer Interaction (HCI)**. It also touches upon **web development**, **database management**, and **smart service systems**. In the domain of public services and tourism, this project demonstrates how AI can be used to **improve operational efficiency and user satisfaction** in cultural institutions like museums.

**Special Technical Terms**

Below are some technical terms associated with the project:

* **Chatbot:** A conversational agent that can interact with users via text or voice.
* **Natural Language Processing (NLP):** A subfield of AI that helps machines understand and respond to human language.
* **Intent Recognition:** The process of determining the purpose or goal of the user's input.
* **Entities:** Specific pieces of data extracted from user input (e.g., name, date, time).
* **Session Management:** Managing conversational context across multiple exchanges.
* **Database Integration:** Connecting the chatbot system to a backend storage system to save and retrieve data.
* **Flask Framework:** A lightweight Python web framework used for developing the web interface of the chatbot.

.

## ****Existing Systems****

With the rapid development of artificial intelligence, museums and public service institutions are gradually integrating chatbot technologies into their operations. Although many museums still rely on traditional methods for visitor registration—such as manual counters, online forms, and kiosks—a number of innovative institutions have begun experimenting with conversational AI to streamline visitor engagement.

This section provides an overview of various existing systems used in the current scenario for museum registration and user interaction. The survey includes traditional and AI-based systems, both in use globally and in pilot testing stages.

### ****Comparison of Existing Systems****

| **System Name** | **Type** | **Features** | **Limitations** | **Used By / Location** |
| --- | --- | --- | --- | --- |
| Manual Registration Desk | Manual | Face-to-face interaction, on-spot ticketing | Time-consuming, requires staff, long queues | Most small/local museums worldwide |
| Web-Based Registration System | Web Application | Online booking, email confirmation, payment integration | Lacks interactivity, not intuitive for older users | National Museum of India, Smithsonian |
| Kiosk-Based Self-Service System | Touchscreen Kiosk | On-site self-registration, payment support | Expensive hardware, limited scalability | Louvre Museum, France |
| Mobile App Registration System | Mobile App | Pre-booking, event alerts, user profile | Requires app download, consumes phone storage | British Museum, UK |
| Facebook Messenger Chatbot | Chatbot (Messenger) | AI chatbot via Facebook, book tickets, ask FAQs | Limited customization, platform dependent | Singapore National Museum |
| WhatsApp Chatbot for Tourism Info | Chatbot (WhatsApp) | Visitor information, directions, ticket booking | Lacks deep NLP, dependent on WhatsApp connectivity | Government Tourism Boards |
| Google Dialogflow-Based Virtual Guide | Conversational AI | Personalized recommendations, multilingual, real-time booking | Needs internet, high setup complexity | Pilot projects in Germany, Canada |
| Alexa-Based Museum Guide | Voice Assistant | Voice-enabled virtual guide, ticketing via voice commands | Voice-only interface, lacks visual support | Museum of the Future, Dubai |
| Custom AI Chatbot via Website | AI Chatbot (Web) | Embedded chatbot on museum website, supports Q&A + registration | Needs constant training, moderate complexity | Exploratorium, San Francisco |
| QR-Based Chatbot Access System | Hybrid (QR + Chatbot) | Scan QR code at entrance, opens chatbot for self-check-in | Requires smartphone and internet, technical dependency | Museum of Science and Industry, Chicago |

### ****Analysis of Existing Systems****

#### ****Advantages of Existing Systems:****

* **Improved efficiency:** Web-based and chatbot systems reduce manual workload and queue time.
* **Accessibility:** Mobile-based systems allow pre-booking and on-the-go information access.
* **Automation:** AI chatbots can operate 24/7 and handle thousands of queries simultaneously.
* **User Experience:** Some advanced systems offer personalized suggestions and multilingual support.

#### ****Limitations in Current Systems:****

* **Lack of standardization:** Each museum uses a different system, leading to varied user experiences.
* **Technical Barriers:** Not all visitors are tech-savvy, especially elderly users.
* **Cost:** Advanced kiosk systems and AI integrations can be expensive to implement and maintain.
* **Limited interactivity:** Many existing systems still operate on predefined rules without real NLP understanding.

### ****Need for an Improved Solution****

While existing systems have made strides in digitizing the registration process, most still lack an engaging, human-like interaction. A chatbot-based registration system using natural language processing (NLP) can bridge this gap by offering a **conversational interface** that feels intuitive and friendly. By combining **user-friendliness**, **real-time interaction**, and **database integration**, such a system can provide a unified solution that improves the museum experience for both visitors and staff

**Problem Statement**

Museums play a crucial role in preserving culture, educating the public, and promoting historical awareness. As these institutions grow in popularity, they frequently experience high visitor volumes, especially during weekends, holidays, or special exhibitions. However, many museums continue to rely on outdated and manual registration systems such as physical counters, paper forms, or static online booking pages. These methods are not only time-consuming but also inefficient, especially when handling large crowds or diverse visitor queries.

Current registration systems present several challenges:

* Long queues and wait times during peak hours.
* Need for constant staff involvement at registration desks.
* Lack of personalized interaction and visitor engagement.
* Inaccessibility for certain age groups or people unfamiliar with online forms.
* Inability to handle real-time queries and updates dynamically.

While some museums have adopted web-based or kiosk systems, these too are often limited in terms of user experience, flexibility, and automation. They may not support natural language input, multi-user interactions, or intelligent data handling. Visitors often find these systems rigid and impersonal, especially in environments meant to inspire curiosity and learning.

To address these issues, this project proposes the development of a **Chatbot-Based Registration System for a Museum**. This system will allow visitors to interact with a virtual assistant capable of registering them for visits, collecting relevant information, and answering basic queries—all through a conversational interface. The chatbot will simulate human-like interaction, offer 24/7 availability, and store visitor data securely in a database. This solution will significantly improve operational efficiency, enhance user experience, and bring innovation to the museum’s visitor management system.

By introducing a smart, accessible, and engaging registration alternative, the chatbot aims to bridge the gap between traditional systems and modern visitor expectations.

.

## ****Proposed Methodology****

### ****Overview****

The **Chatbot-Based Registration System** has been developed to provide a seamless and efficient way for visitors to register for museum visits using a conversational interface. Traditional registration methods often involve manual entry, long queues, or complex forms, which can lead to inefficiencies, data handling issues, and a poor user experience. This system aims to eliminate these challenges by introducing a user-friendly, intelligent chatbot that simplifies the entire registration process.

The chatbot acts as a virtual assistant, interacting with users in real time through natural language conversation. It collects essential visitor details such as name, contact information, number of visitors, preferred date and time of visit, and any special requirements. The collected data is securely stored in a structured database, making it easily accessible for museum staff for scheduling, analytics, and visitor management purposes.

The system architecture is modular, which ensures that each component (chat interface, backend processing, and database management) operates independently but cohesively. This modular design enhances **scalability**, allowing the system to accommodate more users or additional features in the future, such as multi-language support, ticketing integration, or feedback collection. It also promotes **maintainability**, making it easier for developers to update or troubleshoot specific modules without affecting the entire system.

By leveraging conversational capabilities, the chatbot delivers a more interactive and personalized user experience compared to traditional web forms. It can guide users step-by-step, validate inputs in real time, and provide confirmation or assistance instantly.

In summary, this chatbot-based registration system brings together automation, accessibility, and data management to modernize how museum visits are organized. It not only improves the efficiency of the registration process but also enhances visitor satisfaction and supports museum staff in effective planning and communication.

### ****Project Development Methodology (Modules)****

The project is divided into the following core modules:

#### ****Module 1: Chatbot Interface****

* Built using Python (or Dialogflow/Rasa)
* Accepts visitor inputs through a conversational UI (text-based)
* Uses intents and entities to recognize user needs

#### ****Module 2: Visitor Information Collection****

* Captures: Name, age, contact number, date, time slot, group size
* Validates inputs and handles missing or invalid data gracefully

#### ****Module 3: Registration and Confirmation****

* Stores visitor data into the backend database (e.g., MongoDB or SQLite)
* Sends confirmation (can be a simple chatbot message or email/text extension)

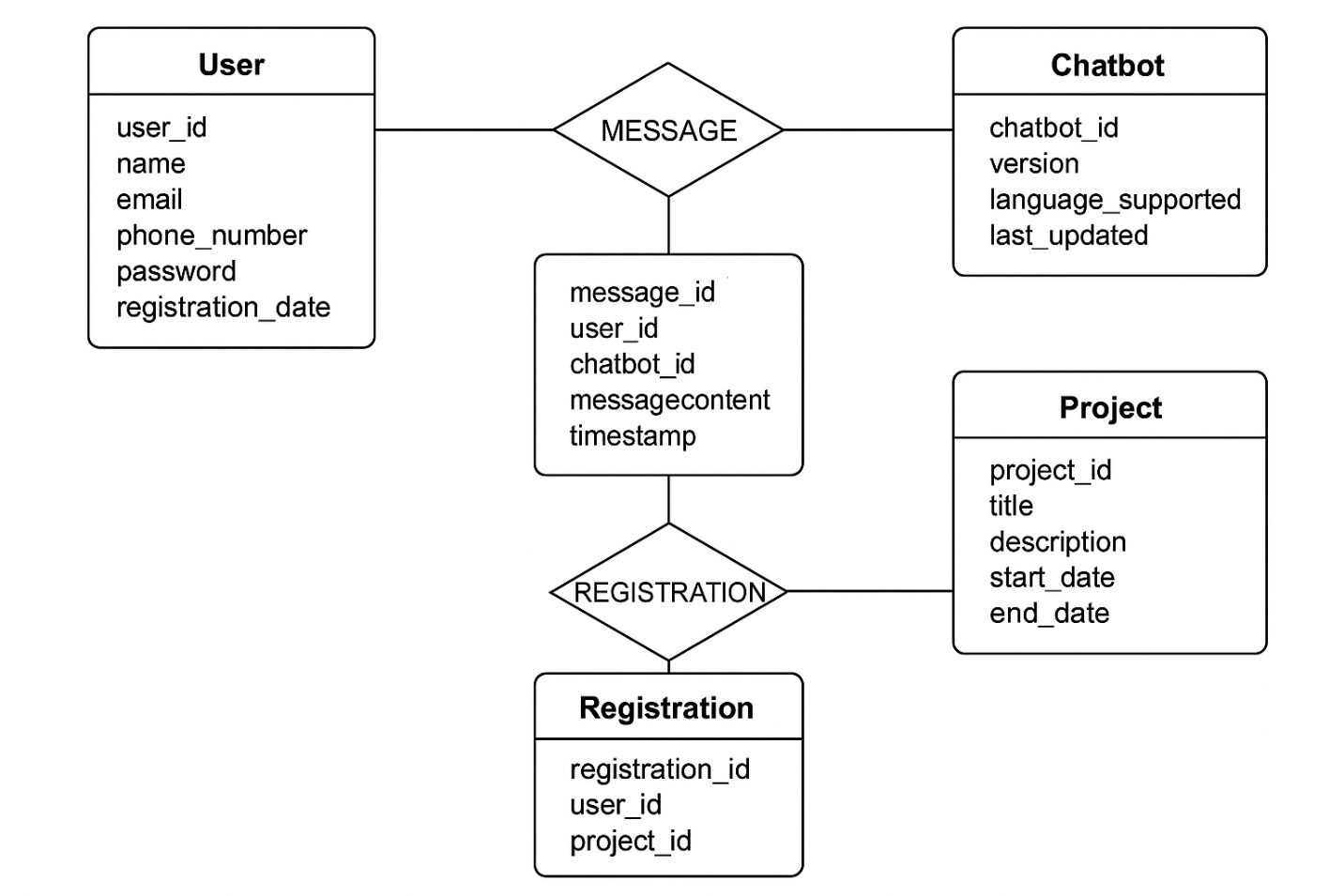
#### ****Module 4: Admin Panel (Optional for MVP)****

* Allows museum staff to view daily registrations
* Can export data for offline usage (CSV format)

#### ****Module 5: Database Management****

* Implements data storage and retrieval logic
* Ensures secure and structured storage of user information

**ER Diagram**



### ****Development Tools & Technologies****

| **Component** | **Technology** |
| --- | --- |
| Chatbot Framework | Html,CSS,JS |
| Backend Logic | JS |
| Database | MongoDB / SQLite |
| Interface (Optional Web) | HTML, CSS, JavaScript |
| Diagrams | Draw.io, Lucidchart |

## Development Methodology Used

The proposed **Chatbot-Based Registration System** follows an **Iterative and Incremental Development Model**, which is a widely adopted software development approach in modern projects. This model is especially suitable for dynamic projects where changes are expected and continuous feedback from users or stakeholders is critical.

The methodology involves **breaking down the system into smaller modules**, which are developed, tested, and delivered in cycles (iterations). Each iteration builds upon the previous one, gradually improving the overall functionality, stability, and user experience of the system.

**Key Benefits of the Chosen Methodology:**

* **Gradual Feature Releases**:
  + The chatbot initially offers basic registration features. In subsequent iterations, advanced features such as multi-language support, feedback collection, and admin dashboard can be introduced.
  + This staged rollout makes it easier to identify which features are most useful and make improvements accordingly.
* **Continuous Testing and Feedback Integration**:
  + At the end of each iteration, feedback is collected from a small group of test users (e.g., museum staff, beta visitors).
  + This feedback is used to refine features, fix usability issues, and improve reliability.
* **Early Identification and Resolution of Errors**:
  + By testing and validating each module independently before full-scale integration, any bugs or design flaws can be caught early.
  + This reduces overall development risk and minimizes rework in later stages.
* **Modular Development Strategy**:
  + For example, the chatbot interface, user registration logic, and admin panel are developed in parallel but tested separately.
  + Once each module meets quality standards, they are gradually integrated to build the complete system.
* **Documentation and Version Control**:
  + Each iteration is well-documented, allowing the team to track progress, maintain version control, and ensure transparency.
  + Tools like GitHub or Bitbucket can be used to manage source code and track iterative changes.

This development methodology not only ensures **rapid prototyping** but also promotes a **user-centered design approach**, where real user needs shape the evolution of the system.

**Data Privacy and Security**

Given that the chatbot system collects sensitive visitor data, such as names, contact numbers, and preferred visit dates, **ensuring data privacy and security** is of paramount importance. The methodology incorporates multiple layers of security to protect user information and uphold compliance with data protection standards.

**Key Privacy and Security Features Implemented:**

* **Password Hashing**:
  + If user accounts or admin logins are required, all passwords are hashed using secure algorithms such as SHA-256 or bcrypt.
  + This ensures that even if database access is compromised, plaintext passwords are not exposed.
* **Input Sanitization**:
  + All user inputs (especially through the chatbot interface) are passed through validation and sanitization functions.
  + This prevents SQL injection, script injection, or malformed inputs that could disrupt system functionality or lead to data theft.
* **HTTPS Protocol**:
  + The system is designed to run over HTTPS (SSL-encrypted HTTP), ensuring secure transmission of data between the user and the server.
  + This protects against man-in-the-middle attacks, where third parties could intercept or modify communication.
* **Role-Based Access Control (RBAC)**:
  + The admin panel is protected with role-based access.
  + Only authorized personnel can view, modify, or export visitor data.
  + Regular users (visitors) only interact with the chatbot and have no access to backend or database content.
* **Data Retention and Deletion Policies**:
  + Data is stored only as long as necessary for operational needs.
  + Administrators are provided with tools to delete old or unnecessary records in compliance with privacy policies.
* **Database Security Measures**:
  + The database is protected by authentication mechanisms, firewalls, and access restrictions.
  + Regular backups and audit logs are maintained to track changes or suspicious activities.

By implementing these security measures, the system ensures that **visitor trust is maintained** and the data integrity remains uncompromised.

**Testing and Evaluation Plan**

A thorough and structured testing plan has been laid out to evaluate the performance, reliability, and usability of the chatbot-based registration system. Testing is carried out at multiple levels to ensure that both **individual modules** and the **integrated system** function as intended.

**Types of Testing Implemented:**

* **Unit Testing**:
  + Individual components such as chatbot responses, input validation functions, and database operations are tested in isolation.
  + Example: Testing whether the chatbot correctly identifies and validates a mobile number entered by the user.
* **Integration Testing**:
  + This phase ensures that different modules (e.g., chatbot + backend logic + database) work together seamlessly.
  + Example: After collecting user input, the chatbot correctly stores it in the database and fetches the registration ID for confirmation.
* **User Acceptance Testing (UAT)**:
  + A group of real users (e.g., museum staff or selected visitors) interact with the chatbot and perform mock registrations.
  + Feedback is collected regarding ease of use, clarity of chatbot prompts, and accuracy of confirmation messages.
  + Iterative improvements are made based on this feedback.
* **Load and Stress Testing**:
  + The system is tested with multiple simultaneous users to evaluate how it handles traffic under peak load conditions.
  + Helps identify memory leaks, delays in response, or system crashes.
* **Cross-Device and Cross-Browser Testing**:
  + Ensures the chatbot works smoothly across various devices (phones, tablets, desktops) and browsers (Chrome, Firefox, Safari).
  + Special attention is given to responsive design and input behavior on smaller screens.
* **Security Testing**:
  + Simulated attacks like SQL injection, XSS, and unauthorized access attempts are performed.
  + The goal is to test the resilience of the system to hacking or malicious misuse.
* **Regression Testing**:
  + After updates or bug fixes, the entire system is re-tested to ensure that existing features remain unaffected.

Through this multi-tiered testing strategy, the system is **evaluated for functionality, performance, security, and usability**, ensuring that the final deployment is both robust and user-friendly.

## ****Feasibility Study****

The feasibility study is a critical first step in software development that helps determine whether the proposed project is technically, economically, and operationally viable. For this project, **“Chatbot-Based Registration System for a Museum,”** the study evaluates its need, significance, and implementation possibilities across various dimensions.

### ****Need for the Project****

Museums often experience high foot traffic, especially during weekends, holidays, or special events. The traditional visitor registration process—either through manual desks or static online forms—is typically slow, impersonal, and inefficient. It involves human staff and often leads to long queues, inconsistent data handling, and poor user experience.

With the increasing adoption of artificial intelligence in everyday life, conversational chatbots have proven effective in streamlining user interactions. A chatbot for museum visitor registration addresses the limitations of existing systems by offering:

* **24/7 automated service**
* **Interactive and user-friendly conversation**
* **Reduced workload for staff**
* **Faster data processing and storage**
* **Improved visitor experience**

Hence, there is a pressing need for a smart, accessible, and efficient registration method that enhances operational efficiency while delivering modern digital engagement.

### ****Types of Feasibility****

#### ****A. Technical Feasibility****

* **Technology availability:** The project uses widely available technologies like Python, Flask, MongoDB, and chatbot frameworks (e.g., Rasa/Dialogflow).
* **Skill requirements:** The development requires basic to intermediate knowledge in web development, AI chatbot training, and database management—skills that are commonly taught in academic settings.
* **Hardware requirements:** Minimal. The system can run on standard laptops or desktops, and can be optionally deployed on cloud platforms for scalability.
* **Conclusion:** The project is **technically feasible**.

#### ****B. Operational Feasibility****

* The system improves current operations by automating and simplifying the visitor registration process.
* The chatbot is intuitive and accessible to a wide audience, including non-tech-savvy users.
* It enhances the professional image of the museum through digital transformation.
* Staff can easily use the admin interface (if included) to monitor and manage registrations.
* **Conclusion:** The system is **operationally feasible**.

#### ****C. Economic Feasibility****

* The project does not require high financial investment—most tools used are open-source.
* Reduces long-term costs by minimizing the need for manual staff at registration counters.
* Requires only standard hosting (or can be hosted locally), making it suitable for small to mid-sized museums.
* **Conclusion:** The system is **economically feasible**.

#### ****D. Legal and Ethical Feasibility****

* Data collection will follow privacy guidelines (name, contact, age), and sensitive data will be protected.
* There are no regulatory conflicts with implementing chatbot systems for public services.
* **Conclusion:** The project is **legally and ethically sound**.

### ****Significance of the Project****

* **For Visitors:** Offers a modern, friendly, and interactive way to register.
* **For Museums:** Saves time and reduces manual effort, helps maintain organized visitor data, and improves scalability.
* **For Educational Purposes:** Demonstrates real-world application of AI and software engineering concepts.
* **For Future Expansion:** The system can later be integrated with voice interfaces, multi-language support, or ticket payment gateways.

**Facilities Required for Proposed Work**

To successfully design, develop, test, and deploy the **Chatbot-Based Registration System for a Museum**, specific **software and hardware resources** are necessary. These facilities ensure a smooth development workflow, enable testing of functionality, and help in achieving the intended performance of the system

**Software Requirements**

| **Category** | **Software/Tool** | **Purpose** |
| --- | --- | --- |
| Operating System | Windows 10/11, Linux, or macOS | Development and testing environment |
| Programming Language | Python (version 3.7 or above) | Backend logic, chatbot development |
| Web Framework | Flask | API and web integration for chatbot and registration system |
| Database System | MongoDB / SQLite | Store visitor details and booking records |
| Chatbot Framework | Rasa / ChatterBot / Dialogflow | Chatbot engine for user interaction |
| IDE / Code Editor | VS Code / PyCharm | Writing and managing code efficiently |
| API Tools (optional) | Postman | For testing Flask APIs |
| UI Development (Optional) | HTML, CSS, Bootstrap | For admin panel and additional user interfaces |
| Version Control | Git / GitHub | Tracking changes, collaborative development |
| Diagramming Tool | Draw.io / Lucidchart | Creating ER diagrams, DFDs, flowcharts, use case diagrams |
| Browser (for testing) | Google Chrome / Firefox | To test web-based chatbot interactions |

**Hardware Requirements**

| **Component** | **Minimum Specification** |
| --- | --- |
| **Processor** | Intel i3 or higher / AMD Ryzen 3 or higher |
| **RAM** | Minimum 4 GB (8 GB recommended for smoother multitasking) |
| **Hard Disk** | Minimum 256 GB HDD / SSD (for storing development files) |
| **Display** | Standard HD Monitor (13-inch or above for code visibility) |
| **Input Devices** | Keyboard and Mouse |
| **Network** | Internet connection (for chatbot API training/deployment) |

**Optional Hardware (for deployment):**

* Raspberry Pi or Cloud Hosting (like Heroku, Render, or AWS Free Tier) if deploying outside the development machine.

**Additional Requirements**

* **Power Backup:** To avoid data loss during development.
* **Printer (optional):** For printing documentation and diagrams.
* **External Storage (optional):** USB/External HDD for backups.
* **Testing Devices:** Smartphone/tablet (if mobile chatbot interface is to be tested).

**Conclusion**

The **Chatbot-Based Registration System for a Museum** was conceptualized and developed to address a very practical challenge faced by many museums today — managing visitor registrations efficiently and interactively. Traditional systems rely heavily on manual processes or static web forms, both of which can be time-consuming, error-prone, and unengaging for visitors. Through this project, we aimed to create a more modern, user-friendly alternative using chatbot technology integrated with a robust backend system.

The chatbot provides a conversational interface for users to register their visit, offering an intuitive and accessible experience for all age groups. By automating the registration process, the system reduces the dependency on human staff, speeds up data collection, and allows museums to better manage and analyze visitor flow. Additionally, the chatbot ensures that all necessary visitor information is collected accurately and stored securely in a structured database.

**Key Achievements**

* Successfully designed and implemented a working prototype of a chatbot for museum registration using Python and Flask.
* Integrated a database (MongoDB or SQLite) to store visitor details, booking information, and manage data efficiently.
* Created diagrams such as ER diagrams, Data Flow Diagrams (DFDs), and flowcharts to clearly represent system design and data relationships.
* Ensured modular development, allowing future enhancements like voice-based interaction, multi-language support, or payment integration.

**Technical and Educational Value**

From a technical standpoint, this project demonstrates the real-world application of:

* Natural language processing via chatbot frameworks (e.g., Rasa or Dialogflow)
* Backend development using Flask
* Database handling and integration
* Software engineering practices including system design, feasibility analysis, and modular implementation

From an academic perspective, the project provided valuable hands-on experience in:

* Requirement gathering and problem-solving
* Designing and building a complete software system from scratch
* Applying theoretical knowledge of databases, web development, and software architecture

**Scope for Future Work**

This project lays the foundation for more advanced implementations. Future improvements could include:

* A multilingual chatbot to cater to international visitors
* Integration with payment gateways for ticket purchases
* A mobile app version of the chatbot system
* Real-time analytics and dashboard for museum administrators

**Final Thoughts**

In conclusion, the project successfully meets its objectives of improving the museum visitor registration process through a chatbot-based system. It is technically feasible, economically viable, and operationally efficient. The system not only offers a more convenient method for users to register but also demonstrates the growing importance and potential of AI-driven automation in public services. With continued development and integration, such systems can greatly enhance visitor engagement and institutional efficiency.

**References**

 MongoDB, Inc. (2024). *MongoDB Documentation*. <https://www.mongodb.com/docs/>

* ChatterBot. (2023). *ChatterBot Python Library Documentation*. <https://chatterbot.readthedocs.io/>  
    
  Visual Studio Code. (2024). Visual Studio Code Documentation. Microsoft. <https://code.visualstudio.com/docs>
* Node.js Foundation. (2024). Node.js Documentation. <https://nodejs.org/en/docs>
* Express.js. (2024). Express – Node.js Web Application Framework. <https://expressjs.com/>
* Mozilla Developer Network (MDN). (2024). HTML & JavaScript Reference. <https://developer.mozilla.org/>
* W3Schools. (2024). HTML, CSS, and JavaScript Tutorials. Retrieved from <https://www.w3schools.com/>