PROJECT PROPOSAL

ON

Spiritual AI Chat Bot

Submitted By: - ROHIT KUMAR

ENROLMENT NO.: - 2450120000

Under Guidance of: - AKHILESH MISHRA

Submitted to the School of Computer and Information Sciences, IGNOU
In partial fulfilment of the requirements
For the award of the degree

MASTERS OF COMPUTER APPLICATIONS (MCA)

INDIRA GANDHI NATIONAL OPEN UNIVERSITY

SCHOOL OF COMPUTER AND INFORMATION SCIENCES
MAIDAN GARHI, NEW DELHI-110068

TABLE OF CONTENTS

1. PROJECT TITLE

Spiritual AI Chat Bot

2. INTRODUCTION OF THE PROJECT

The Spiritual Chatbot System is a web-based intelligent application designed to provide users with meaningful spiritual guidance and knowledge through conversational interaction. The system allows users to ask questions related to spirituality, books, teachings, and practices, and receive accurate, context-aware responses powered by an AI-driven chatbot engine. This project aims to make spiritual knowledge easily accessible in a modern, interactive, and user-friendly manner.

The platform supports both regular users and administrators, ensuring smooth interaction, secure access, and effective monitoring of system activity. Along with answering user queries, the system also tracks user behavior such as button clicks, chat history, and feedback, helping administrators analyze engagement and improve content quality.

- **User Authentication System**
  - Allows users to register and log in securely.
  - Maintains user roles to distinguish between users and administrators.
- **Spiritual Chatbot Interaction**
  - Enables users to ask text or voice-based spiritual questions.
  - Generates intelligent responses using a knowledge base sourced from spiritual books and content.
- **Chat History Management**
  - Stores complete chat conversations for future reference.
  - Allows users to view their past interactions with the chatbot.
- **Click Tracking & User Analytics**
  - Records user interactions with chatbot buttons such as Meet, Books, Follow, and Shop.
  - Helps in understanding user interests and engagement patterns.
- **Source Referencing System**
  - Maintains references to books or content sources used for chatbot responses.
  - Enhances transparency and reliability of information provided.
- **Admin Dashboard & Monitoring**
  - Enables admin to view registered users, chat histories, and click analytics.
  - Provides insights for system monitoring, auditing, and improvements

3. OBJECTIVE OF THE PROJECT

The primary objective of the Spiritual Chatbot System is to develop an intelligent, interactive platform that provides users with reliable spiritual guidance through conversational interaction. The project focuses on combining artificial intelligence, structured data management, and user analytics to create a meaningful and user-centric spiritual experience.

This system is designed to serve both end users seeking spiritual knowledge and administrators who monitor, analyze, and improve the overall performance and quality of the platform.

- **To Develop an Intelligent Spiritual Chatbot**
  - Provide accurate, context-aware answers to spiritual questions.
  - Support both text-based and voice-based user queries.

- **To Enable Secure User Management**
  - Allow users to register and log in securely.
  - Maintain role-based access control for users and administrators.

- **To Maintain Chat History and Session Data**
  - Store user conversations for future reference.
  - Track user sessions including login and logout activities.

- **To Track User Interaction and Engagement**
  - Record button clicks such as Meet, Books, Follow, and Shop.
  - Analyze user behavior to understand preferences and engagement trends.

- **To Provide Admin Monitoring and Control**
  - Enable administrators to view registered users and their activities.
  - Allow admins to access chat histories, click analytics, and audit logs.

- **To Ensure Data Integrity and Transparency**
  - Store references to books and sources used for chatbot responses.
  - Maintain system logs for monitoring and security purposes.

4. PROJECT CATEGORY

The Spiritual Chatbot System is a web-supported 3-tier architecture–based application that integrates Artificial Intelligence (AI), Machine Learning (ML), and Large Language Model (LLM) support along with modern software development practices. It combines web technologies and data analytics to deliver an interactive and intelligent user experience.

Primary Project Category

- Artificial Intelligence (AI) Based Application
  - Utilizes natural language processing (NLP) to understand user queries.
  - Generates intelligent and context-aware spiritual responses.

Secondary Project Categories

- Web Application Development
  - Developed as a web-based platform accessible through browsers.
  - Supports user and admin interfaces with secure authentication.
- Chatbot System
  - Implements conversational interaction between users and the system.
  - Designed for real-time question-answering and guidance.
- Data Management System
  - Stores user data, chat history, session details, and interaction logs.
  - Uses structured databases to maintain data integrity and consistency.
- Analytics & Monitoring System
  - Tracks user behavior such as button clicks and feedback.
  - Provides reports and insights through the admin dashboard.
- Technology-Oriented Classification
  - Front-End: Web-based user interface for chatbot interaction.
  - Back-End: Server-side logic for authentication, chatbot processing, and analytics.
  - Database: Centralized storage for users, chats, sessions, and sources.
  - AI/NLP Engine: Core module responsible for understanding and generating responses.

5. HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATIONS

Hardware Requirement

Server-Side and Client-Side Requirement: -

- **Intel ® I3processor 3.9GH**zor faster, **4 GB RAM.**
- **500 GB Hard Disk Drive.**

Software Requirement

Server-Side Requirement: -

- **Operating System:** Certified distributions of Windows 10, or Upper Version of Windows.
- **Back-End Tool:** Node JS, Express JS, FastAPI and MongoDB
- **Front–End Tool:** React JS, Next JS.
- **Language:** JavaScript, TypeScript and Python.

Client-Side Requirement: -

- **Operating System:** Window 10 or Upper Version of Windows or na
- **Web Browser:** Firefox, Chrome, Mozilla.

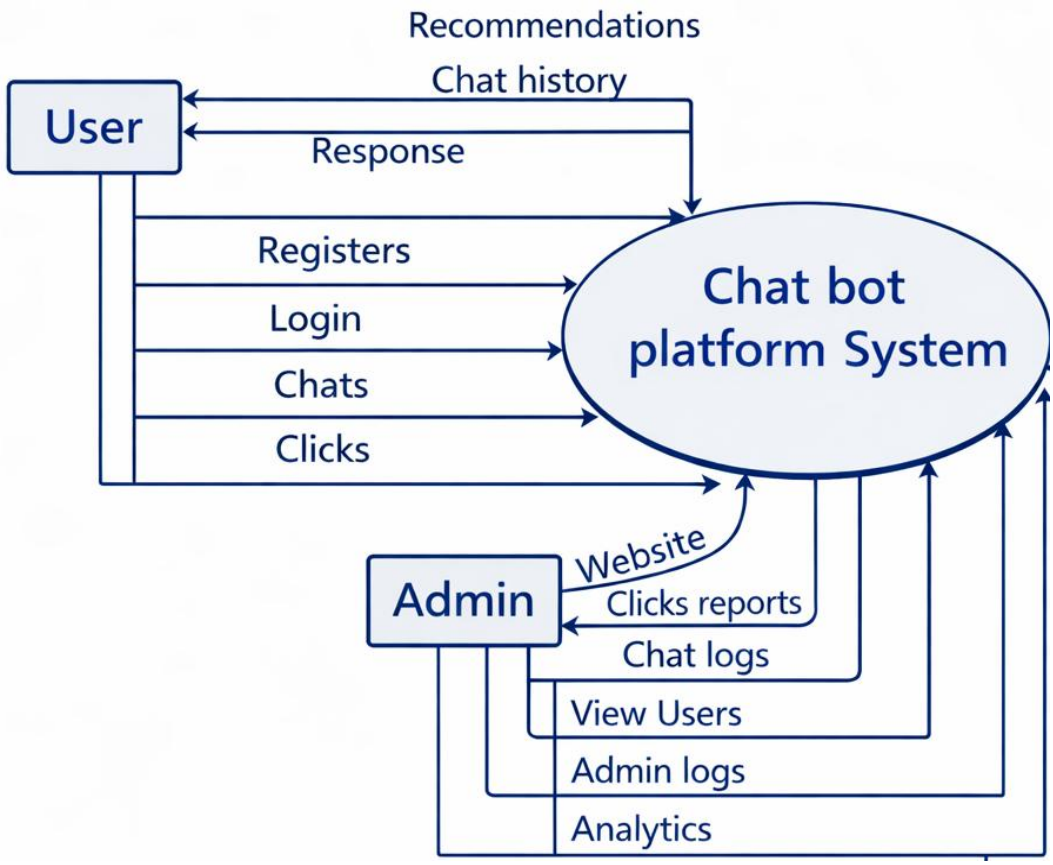
6. DRAWBACKS OF EXISTING SYSTEM

The traditional systems used for spiritual guidance and information dissemination rely heavily on manual processes and static content. These systems face several limitations in terms of accessibility, efficiency, and user engagement.

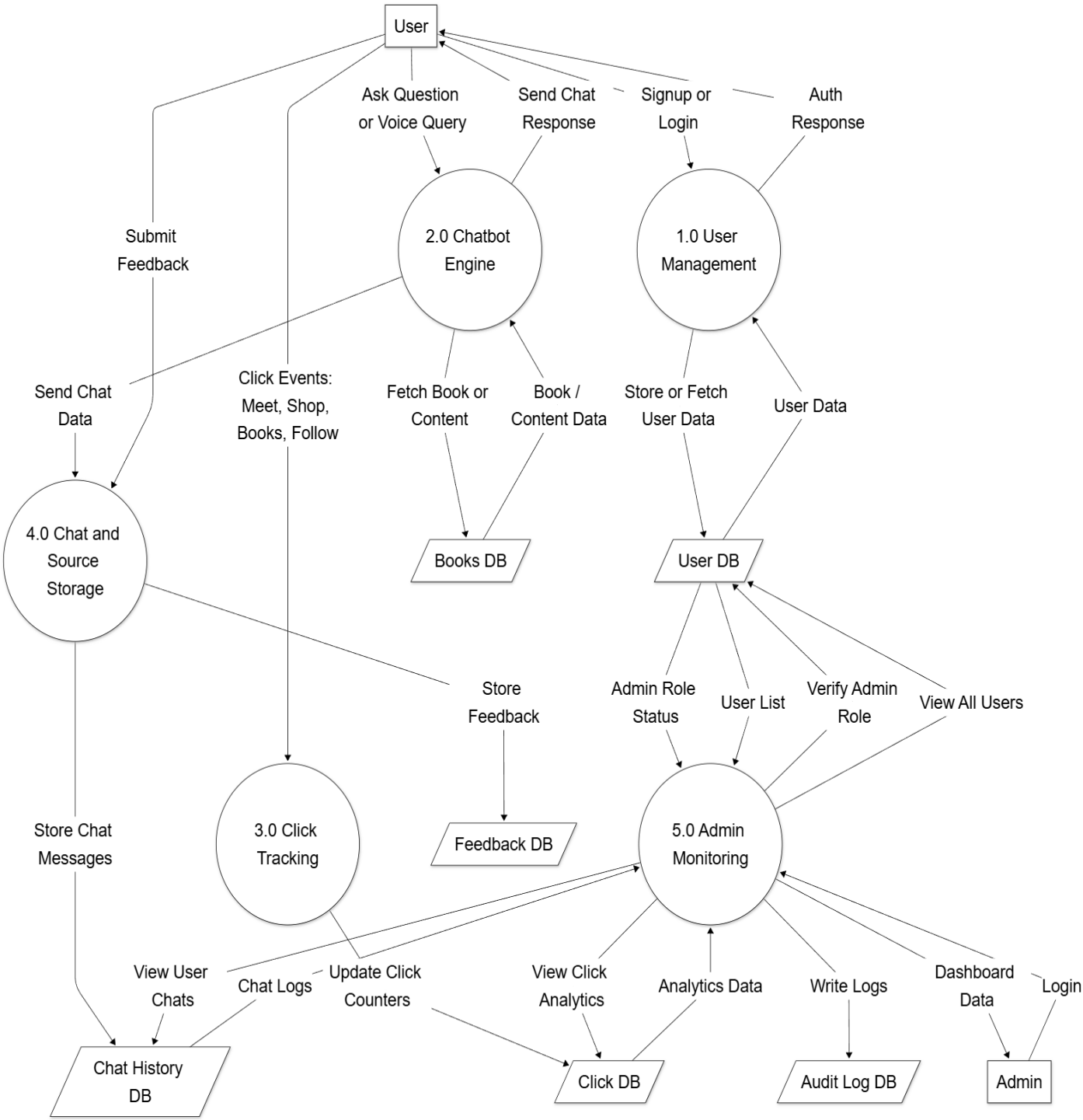
- **Limited Accessibility**
  - Spiritual guidance is often available only at specific locations or times.
  - Users cannot access instant support whenever they need it.
- **Lack of Personalization**
  - Existing systems do not adapt responses based on user behavior or interests.
  - All users receive the same information regardless of their preferences.
- **No Real-Time Interaction**
  - Traditional platforms lack interactive question-answer capabilities.
  - Users must rely on books or human experts, which may cause delays.
- **Manual Data Management**
  - User records, interactions, and feedback are maintained manually or not stored at all.
  - This leads to data inconsistency and difficulty in tracking user activity.
- **No Analytics or Monitoring**
  - Existing systems do not provide insights into user engagement.
  - Administrators cannot analyze which content is most useful or frequently accessed.
- **Scalability Issues**
  - Human-dependent systems cannot serve a large number of users simultaneously.
  - Increased demand leads to reduced service quality.

7. SOFTWARE DESIGNS

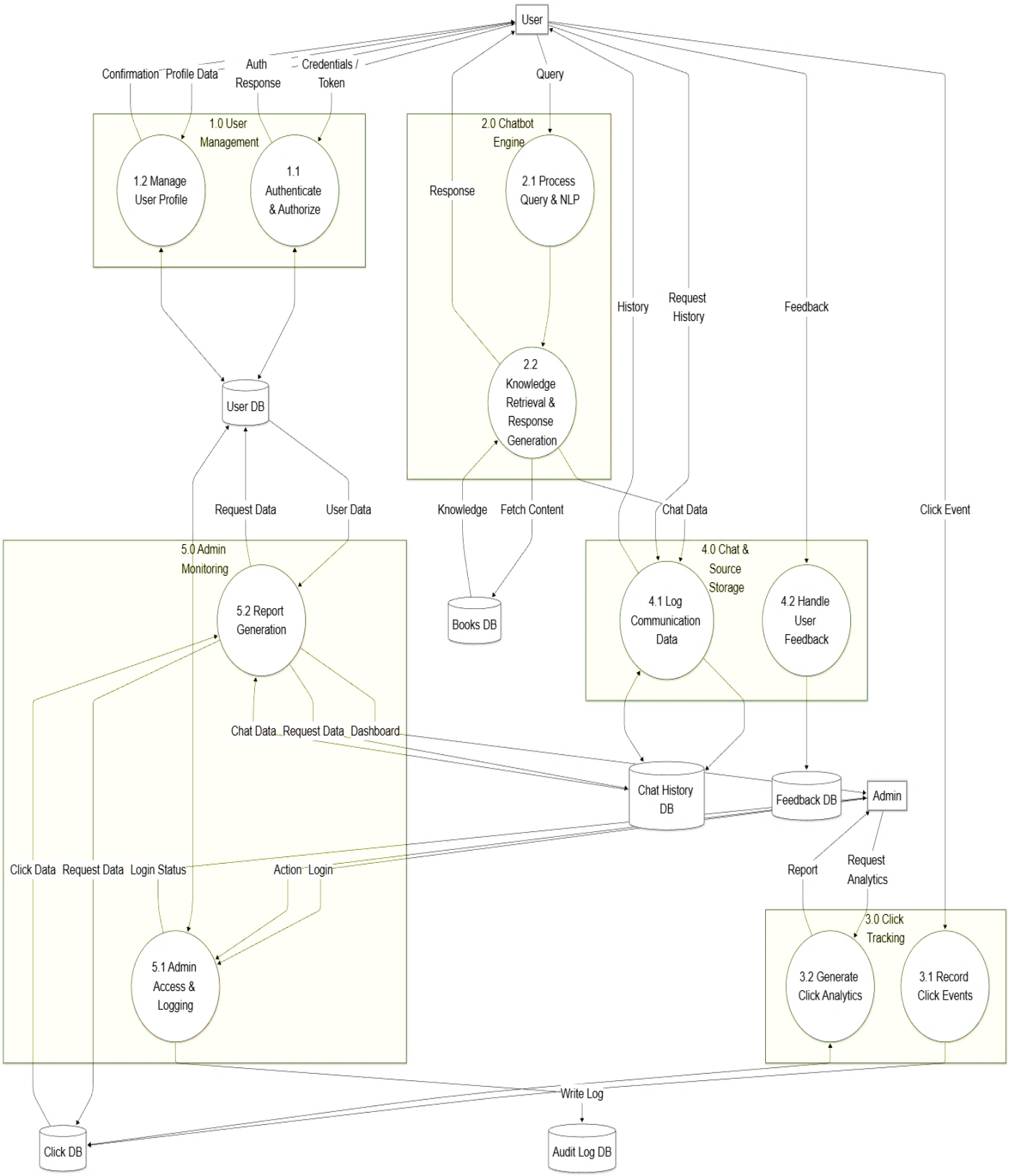
7.1 DATA FLOW DIAGRAM (DFD)



LEVEL-0 CONTEXT DIAGRAM

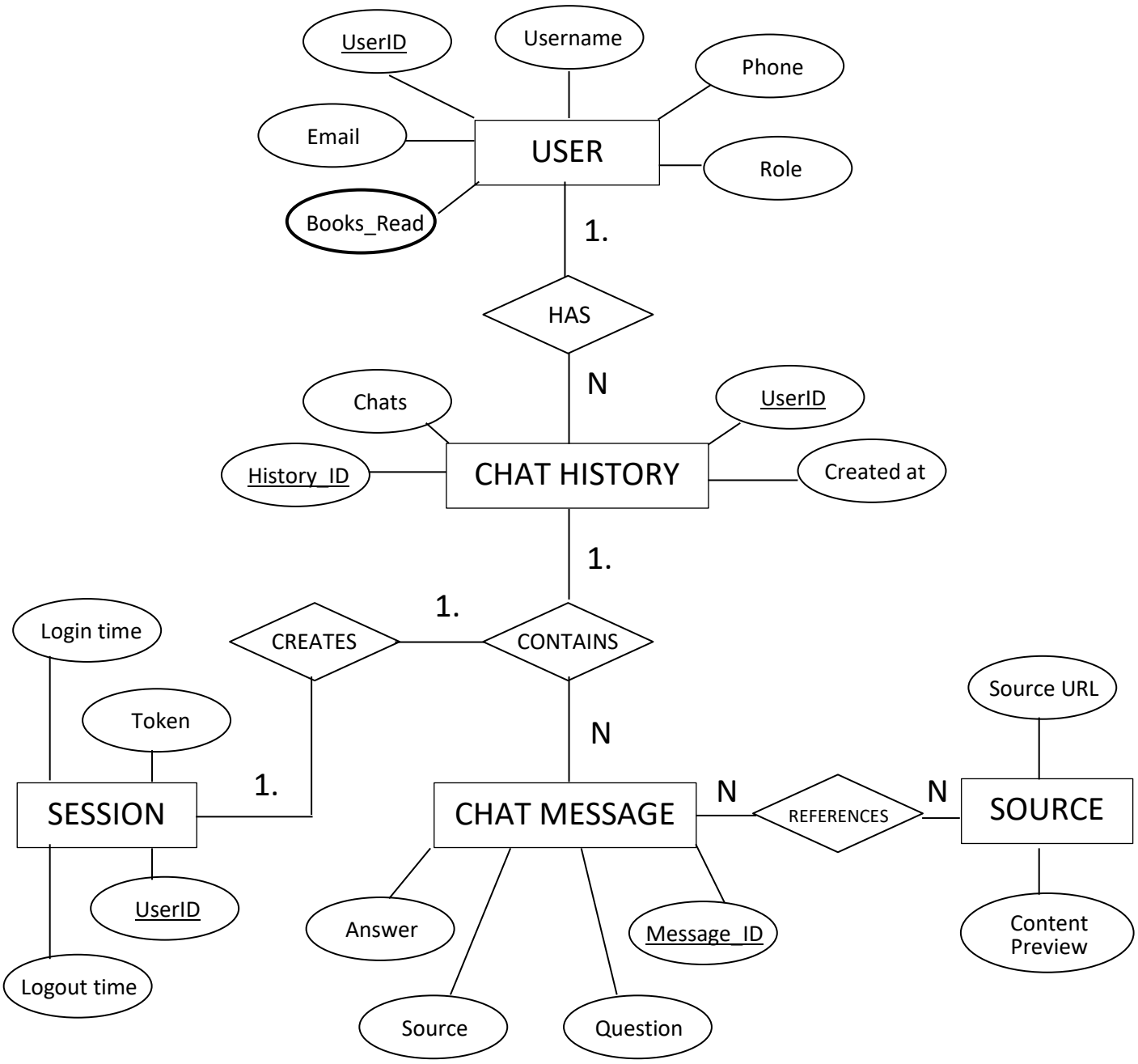


LEVEL-1 DATA FLOW DIAGRAM



LEVEL-2 DATA FLOW DIAGRAM

7.2 ER DIAGRAM



ENTITY RELATIONSHIP DIAGRAM

7.3 TABLE ARCHITECTURE

1.User

S No.	Column Name	Data Type	Constraints	Description
1.	User Id	Object ID	Primary key	It contains the id of User
2.	User Name	String	NOT NULL	It stores name of registered User
3.	Email	String	NOT NULL	It stores email of each User
4.	Password	String	NOT NULL	It Stroes password of each User
5.	Phone No.	Int	NOT NULL	It Stroes Phone No. of each User
6.	City	String	NOT NULL	It Stroes City of each User
7.	Country	String	NOT NULL	It Stroes Country of each User
8.	Gender	String	NOT NULL	It Stroes Gender of each User
9.	Date of Birth	Int	NOT NULL	It Stroes D.O.B of each User
10.	Role	String	NOT NULL	It Stroes Role of each User

2. **Login**

S No.	Column Name	Date Type	Constraints	Description
1.	User Id	Object Id	Primary key	It contains the id of User
2.	User email	String	NOT NULL	It contains the email of User

3. Chart Message

S No.	Column Name	Data Type	Constraints	Description
1.	Message ID	ObjectId	Primary Key	Unique Message Id of each User.
2.	Timestamp	Int	NOT NULL	Timings of messages.
3.	Question	String	NOT NULL	Question of user.
4.	Answer	String	NOT NULL	Answer for user.
5.	Source	Array	NOT NULL	Source from where data is found.

4. Chat History

S No.	Column Name	Data Type	Constraints	Description
1.	Unique _id	ObjectId	NOT NULL	Unique id of a database

2.	userId	ObjectId	NOT NULL	It contains the id a of User
3.	Chat message	Array	NOT NULL	Each user chat stored in his unique id
4.	CreatedAt	Int	NOT NULL	Inserted when user created a chat.

5. Chat Session

S No.	Column Name	Data Type	Constraints	Description
1.	User Id	Object Id	Primary Key	It contains the id of User.
2.	Token	String	NOT NULL	Each chat session has his unique token
3.	Login Time	Int	NOT NULL	Automatically creates when user login
4.	Logout Time	Int	NOT NULL	Automatically creates when user logout

6. User clicks

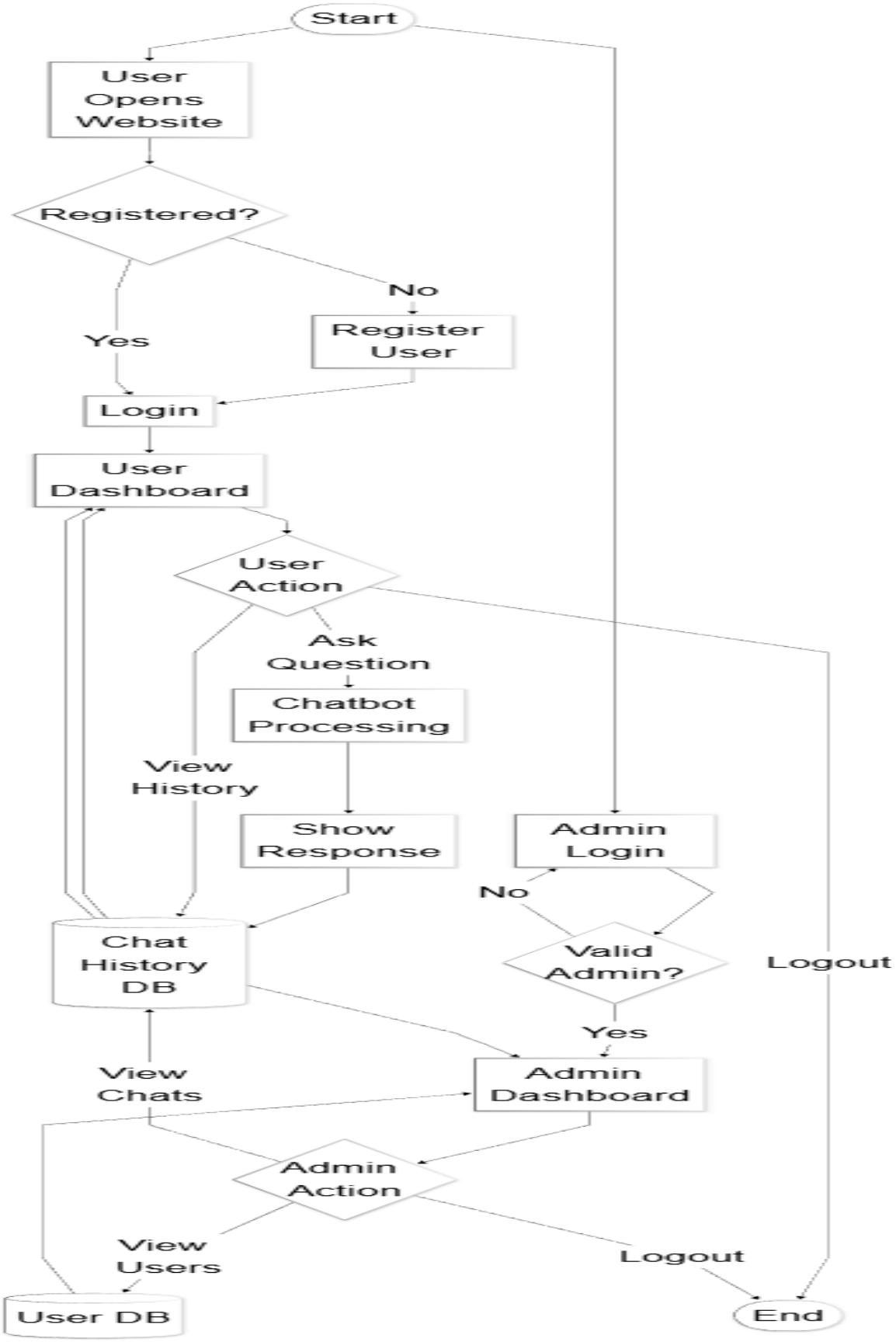
S No.	Column Name	Data Type	Constraints	Description

1.	User Id	Object Id	Primary Key	It contains the id of User.
2.	Shop	String	NOT NULL	Contains the shop component
3.	books	String	NOT NULL	Contains the books component
4.	Follow	String	NOT NULL	Contains the follow component
5.	Meet	String	NOT NULL	Contains the Meet component.
6.	Updated At	int	NOT NULL	Time when it updates

7. Souce

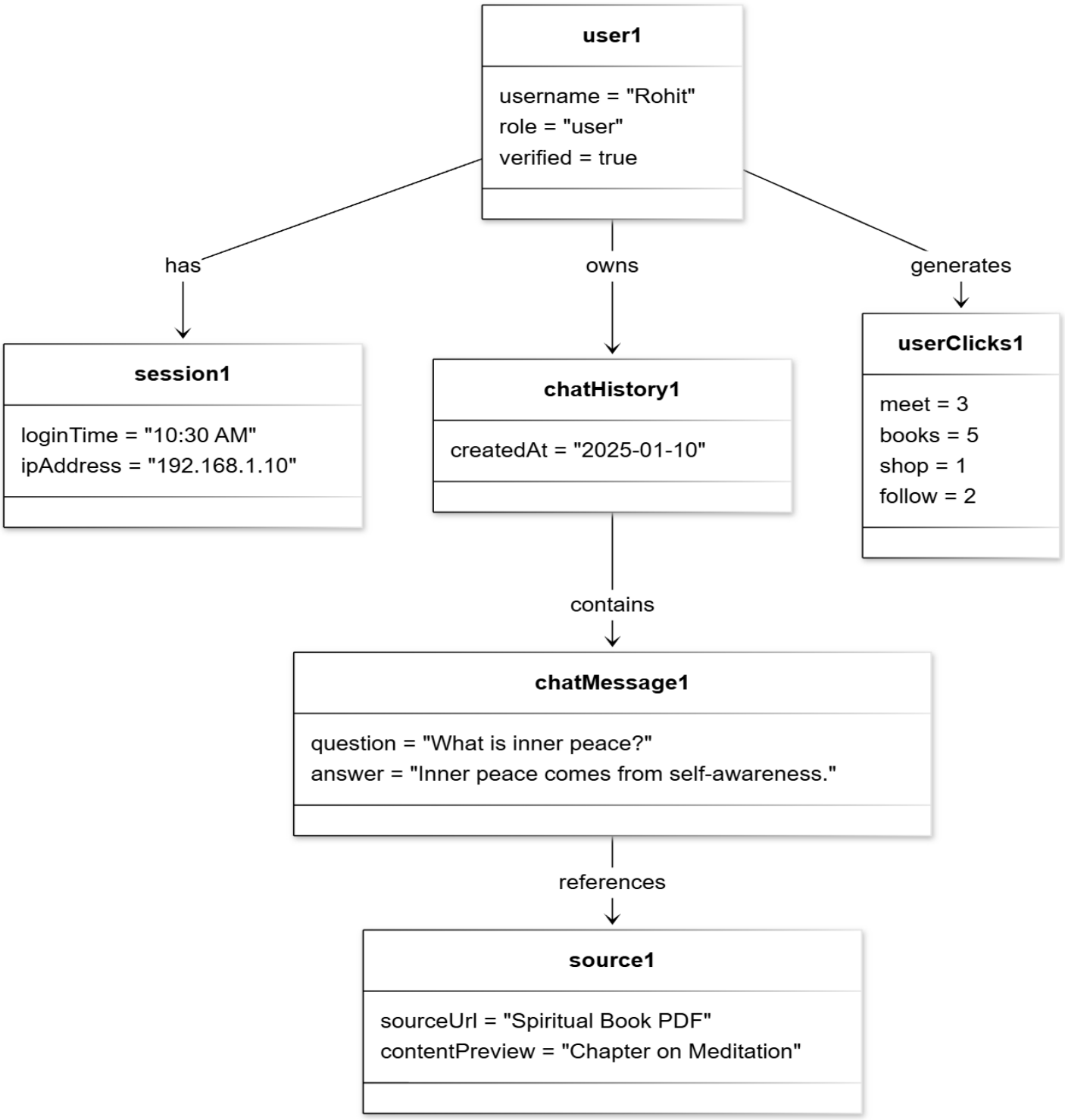
S No.	Column Name	Date Type	Constraints	Description
1.	Source URL	String	NOT NULL	Data source URL
2.	Content preview	String	NOT NULL	Where the content is previewing

7.4 PROCESS FLOW DIAGRAM / FLOWCHART

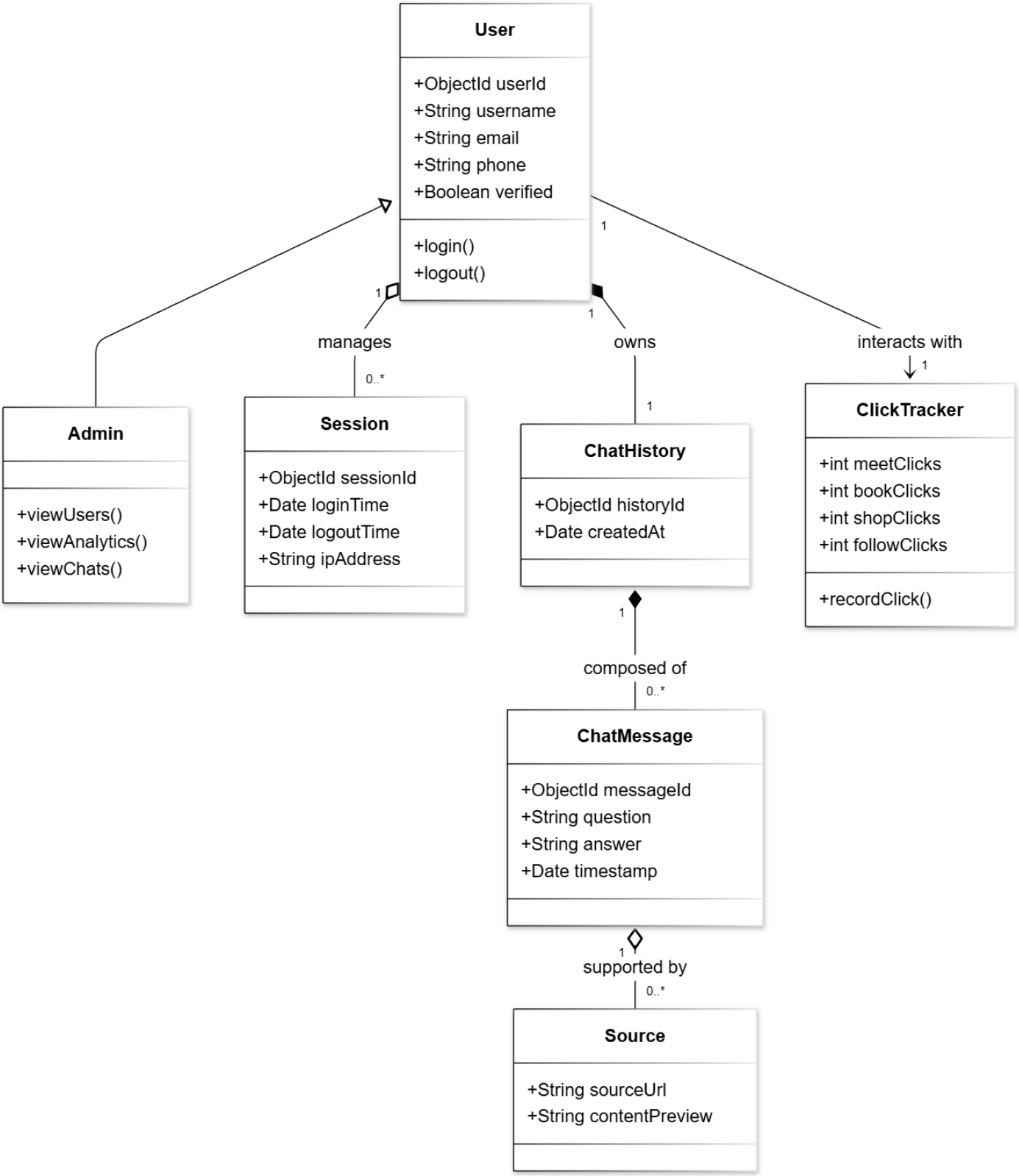


7.5 OBJECT DIAGRAM & CLASS DIAGRAM

OBJECT DIAGRAM



CLASS DIAGRAM



8. MODULES AND THEIR DESCRIPTIONS

The Spiritual Chatbot System is designed using a modular architecture to ensure scalability, maintainability, and clarity in functionality. Each module performs a specific task while interacting with other modules to provide a seamless and intelligent user experience. The system is broadly divided into five major modules, each responsible for handling a core functionality of the application.

Modules of the System

- User Management Module
- Session Management Module
- Chatbot & Chat History Module
- Source & Knowledge Management Module
- User Interaction & Analytics Module

1. User Management Module

Description

The User Management Module is responsible for handling all user-related operations within the system. It manages user registration, authentication, authorization, and role-based access control. This module acts as the foundation of the system, ensuring that only authorized users and administrators can access respective features.

Functions of User Management Module

- Handles user registration using details such as username, email, phone number, gender, and date of birth.
- Authenticates users during login by verifying credentials.
- Assigns roles (User or Admin) to determine access permissions.
- Stores user profile details securely in the database.
- Ensures data validation and uniqueness (email and phone).
- Manages verification status of users.

Importance of the Module

- Ensures system security by preventing unauthorized access.
- Enables personalized chatbot interactions by identifying users.
- Provides a base for tracking user activities and sessions.

- Supports admin functionalities by distinguishing admin users from normal users.

Entities Used

- User

2. Session Management Module

Description

The Session Management Module handles user session tracking throughout the system. It records login and logout activities, session duration, and user IP addresses. This module ensures that user interactions are monitored and managed efficiently.

Functions of Session Management Module

- Creates a session when a user logs into the system.
- Records login time and logout time.
- Tracks IP address for security and auditing purposes.
- Maintains multiple sessions for a single user if required.
- Helps detect unusual login patterns.

Importance of the Module

- Enhances system security through session monitoring.
- Helps administrators track user activity and usage patterns.
- Supports auditing and troubleshooting.
- Enables future enhancements such as session timeout or forced logout.

Entities Used

- Session

3. Chatbot & Chat History Module

Description

The Chatbot & Chat History Module is the core component of the Spiritual Chatbot System. It manages user interactions with the chatbot, processes queries, generates responses, and stores chat history for future reference.

Functions of Chatbot & Chat History Module

- Accepts user queries in text or voice format.
- Processes queries using natural language processing (NLP).
- Generates intelligent spiritual responses.
- Stores every question and answer in the chat history.
- Allows users to view previous conversations.
- Enables admins to monitor user chat activity.

Importance of the Module

- Provides real-time spiritual guidance to users.
- Ensures continuity by maintaining chat history.
- Improves user experience through conversational interaction.
- Enables content analysis and improvement through stored conversations.

Entities Used

- Chat History
- Chat Message

4. Source & Knowledge Management Module

Description

The Source & Knowledge Management Module manages the knowledge base used by the chatbot. It stores and retrieves information from spiritual books, PDFs, and other resources. Each chatbot response is linked to its source for transparency and credibility.

Functions of Source & Knowledge Management Module

- Maintains structured knowledge sources.
- Retrieves relevant content based on user queries.
- Links chatbot answers with their respective sources.
- Stores content previews and source URLs.
- Helps improve trust by showing information origins.

Importance of the Module

- Enhances response accuracy and relevance.
- Provides credibility to chatbot answers.
- Supports explainable AI by referencing sources.
- Allows easy content updates and expansion.

Entities Used

- Source

5. User Interaction & Analytics Module

Description

The User Interaction & Analytics Module tracks and analyzes user behavior within the system. It monitors button clicks such as Meet, Books, Shop, and Follow, helping administrators understand user engagement and system usage.

Functions of User Interaction & Analytics Module

- Records user click events.
- Maintains counters for different interaction types.
- Stores analytics data in the database.
- Provides insights to administrators via dashboard.
- Supports data-driven decision-making.

Importance of the Module

- Helps identify popular features.
- Improves system usability based on user behavior.
- Supports performance evaluation.
- Enables future personalization and recommendation systems.

Entities Used

- User Clicks

9. TESTING & VALIDATION

The **testing** process was carried out systematically using different testing techniques, including Unit Testing, Integration Testing, System Testing, Black-Box Testing, and White-Box Testing.

Validation ensures that the final system meets the intended objectives and user expectations.

1. Unit Testing

Description

Unit testing focuses on testing individual components or modules of the system independently. Each module was tested to ensure that it performs its intended function correctly in isolation.

Modules Tested

- User Management Module
- Session Management Module
- Chatbot Processing Module
- Chat History Storage Module
- Click Tracking Module
- Admin Monitoring Module

Unit Testing Activities

- Testing user registration and login functions
- Verifying password hashing and authentication logic
- Checking session creation and termination
- Testing chatbot response generation logic
- Validating click counter updates
- Testing data storage and retrieval from the database

Tools & Methods

- Manual test cases
- API testing tools
- Console logging for function validation

Outcome

- All individual modules performed correctly and passed unit testing, ensuring a strong foundation for higher-level testing.

2. Integration Testing

Description

Integration testing verifies the interaction between different modules of the system. Since the system consists of multiple interconnected modules, integration testing ensures seamless data flow and communication among them.

Integration Scenarios Tested

- User authentication with session creation
- Chatbot query processing with source retrieval
- Chat response storage in chat history
- Click tracking data generation and analytics
- Admin dashboard fetching data from multiple databases

Key Focus Areas

- Data consistency between modules
- Proper API communication
- Correct handling of dependent services

Outcome

- All modules integrated successfully, and data was transferred correctly across components without loss or inconsistency.

3. System Testing

Description

System testing evaluates the complete system as a whole. This phase verifies whether the fully integrated application meets the specified functional and non-functional requirements.

System Features Tested

- End-to-end user registration and login
- Chatbot interaction (text and voice queries)
- Chat history retrieval
- Feedback submission
- Click analytics tracking
- Admin dashboard operations

Non-Functional Testing

- Performance testing under multiple user requests

- Security testing for authentication and authorization
- Usability testing for user interface clarity

Outcome

- The system functioned correctly under real-world conditions and met all specified requirements.

4. Black-Box Testing

Description

Black-box testing focuses on validating the system's functionality without considering its internal code structure. The tester evaluates outputs based solely on inputs and expected results.

Black-Box Test Cases

- Valid and invalid login credentials
- User registration with duplicate email or phone
- Chatbot responses to various spiritual questions
- Click tracking accuracy
- Admin access control verification

Advantages

- Tests system from the user's perspective
- Ensures user requirements are met
- Helps identify missing or incorrect functionalities

Outcome

- The system produced correct outputs for all valid inputs and handled invalid inputs gracefully.

5. White-Box Testing

Description

White-box testing involves testing the internal logic, code structure, and execution paths of the system. This ensures that all logical paths and conditions are tested.

White-Box Testing Activities

- Code review and logic validation

- Condition and loop testing
- Branch coverage testing
- Error handling verification

Focus Areas

- Authentication logic
- Role-based access control
- Chat message storage logic
- Click counter update logic

Outcome

- All logical paths were tested successfully, and no critical code-level defects were found.

6. Validation

Description

Validation ensures that the system fulfills the intended objectives and meets user expectations. It answers the question: “Are we building the right system?”

Validation Methods Used

- Requirement-based validation
- User acceptance testing
- Output verification
- Admin feedback and review

Validation Criteria

- Accuracy of chatbot responses
- Relevance of spiritual content
- Proper tracking of user interactions
- Secure admin access and monitoring
- Ease of use and accessibility

Outcome

- The Spiritual Chatbot System was validated successfully against all functional and non-functional requirements. The system delivers accurate, interactive, and reliable spiritual guidance.

10. SECURITY MECHANISM IMPLEMENTATION

Security is a critical aspect of the Spiritual Chatbot System due to the involvement of sensitive user data, authentication credentials, session information, and administrative controls. The system implements multiple security mechanisms across different modules to ensure data confidentiality, integrity, and controlled access.

1. Authentication & Authorization Security

- User passwords are securely encrypted using hashing techniques before being stored in the database.
- During login, encrypted passwords are validated to prevent unauthorized access.

2. Session Security Mechanism

- Unique sessions are created for each authenticated user.
- Login and logout timestamps are recorded for monitoring.

3. Data Security & Database Protection

- Sensitive fields such as authentication tokens are excluded from default database queries.
- Input validation is performed to prevent injection attacks.

4. Chat & Content Security

- User queries are sanitized before processing.
- Chat histories are linked to authenticated users only.

5. Admin Panel Security

- Admin login requires valid credentials and role verification.
- All admin activities are logged for auditing purposes.

11. LIMITATIONS OF THE PROJECT

Although the Spiritual Chatbot System provides an intelligent and interactive platform for spiritual guidance, it has certain limitations that are inherent to its design, technology stack, and scope of implementation. These limitations identify areas where the system can be improved in future versions.

1. Dependency on Knowledge Sources

- Limited data sources may result in less comprehensive answers.
- New spiritual content must be manually added to improve coverage.

2. Limited Understanding of Complex Queries

- Deep philosophical or highly abstract spiritual questions.
- Ambiguous or poorly structured user inputs.

3. Language and Cultural Constraints

- The system currently supports a limited set of languages.
- Cultural or regional interpretations of spiritual concepts may not always be accurately represented.

4. Performance Dependency on Internet Connectivity

- The system is web-based and requires a stable internet connection.
- Slow network conditions may impact response time and user experience.

12. FUTURE SCOPE OF THE PROJECT

The Spiritual Chatbot System offers significant potential for enhancement and expansion. With advancements in Artificial Intelligence, Machine Learning, and web technologies, the system can be further improved to provide a richer and more personalized user experience.

The future scope of the Spiritual Chatbot System is extensive and promising due to rapid advancements in Artificial Intelligence, Machine Learning, and web technologies. In the future, the system can be enhanced by integrating more advanced Large Language Models (LLMs) to provide deeper, more personalized, and context-aware spiritual guidance. Multilingual support can be added to make the chatbot accessible to users from different regions and linguistic backgrounds. Voice-based interaction with speech-to-text and text-to-speech capabilities can further improve user engagement and inclusivity. The system can also be expanded to include personalized recommendations based on user behavior, emotions, and past interactions. Advanced analytics and dashboards can be developed for administrators to gain deeper insights into user engagement patterns. Additionally, the platform can be scaled to support mobile applications, wearable devices, and cloud-based deployments, making it more accessible and robust. With continuous learning and feedback mechanisms, the chatbot can evolve over time to deliver more accurate, empathetic, and meaningful spiritual support

13. BIBLIOGRAPHY OF THE PROJECT

1. Ian Goodfellow, Yoshua Bengio, and Aaron Courville,
Deep Learning, MIT Press, 2016.

2. Jurafsky, D., & Martin, J. H.,
Speech and Language Processing, Pearson Education, 3rd Edition, 2023.

3. Russell, S., & Norvig, P.,
Artificial Intelligence: A Modern Approach, Pearson Education, 4th Edition, 2021.

4. OpenAI Documentation,
Large Language Models and Natural Language Processing,
OpenAI Official Documentation.

5. MongoDB Inc.,
MongoDB Manual – Data Modeling and Security Practices,
MongoDB Official Documentation.

6. Node.js Foundation,
Node.js Documentation,
Node.js Official Website.

7. Express.js Documentation,
Express.js Web Framework Guide,
Express.js Official Website.

8. OWASP Foundation,
OWASP Top 10 – Web Application Security Risks,
OWASP Documentation.

9. Pressman, R. S.,
Software Engineering: A Practitioner's Approach, McGraw-Hill Education, 8th Edition, 2019.

10. Sommerville, I.,
Software Engineering, Pearson Education, 10th Edition, 2016.

11. W3C Consortium,
Web Application Architecture and Security Standards,
W3C Documentation.

12. Merkle, L.,
Designing Conversational AI Systems, O'Reilly Media, 2020.