

Chatbot Lab Report -

Overall Design

The Chatbot project is designed to facilitate user interaction through a simple web interface. The architecture follows the MVC (Model-View-Controller) design pattern dividing structure into repository, service and servlet modules. The application uses JSP for the front-end, Hibernate for data management, and MySQL as the database.

JSP Components

The JSP components include the main interface for user interaction, which allows users to submit queries and view the chatbot's responses. The key JSP file, 'chatbot.jsp', contains a form for input and a section to display chat history. JSTL tags are utilized for iterating over conversations stored in the database.

Chatbot Logic

The chatbot logic is implemented in the ChatServlet. It processes user queries and generates appropriate responses. The HibernateUtil class serves as a utility for managing Hibernate's SessionFactory, which is essential for interacting with the database in a Hibernate application. The Conversation class is an entity class represents a record of a conversation between the user and the chatbot. The ConversationDAO class is a Data Access Object (DAO) in your Hibernate-based application that manages the persistence and retrieval of Conversation entities in the database. The ChatServlet class is a Java servlet that acts as the controller for your chatbot application. It handles HTTP requests and responses, processes user input, and manages interactions with the underlying services and data.

Database Schema

The database schema consists of a single table named 'conversations'. The schema is as follows:

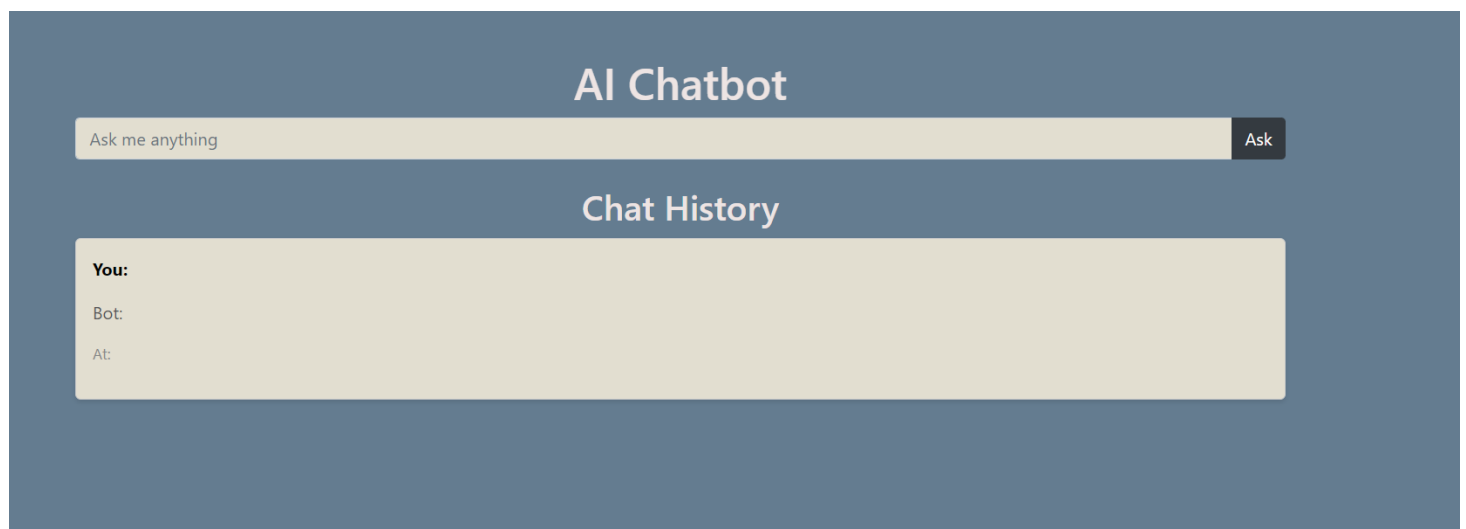
- id (INT, AUTO_INCREMENT, PRIMARY KEY): Unique identifier for each conversation.
- user_query (VARCHAR(255)): Stores the user's query or question.
- chatbot_response (VARCHAR(255)): Stores the chatbot's response to the user's query.
- timestamp (TIMESTAMP, DEFAULT CURRENT_TIMESTAMP): Records the date and time of the conversation.

Functionality

The application allows users to submit queries through a web interface. Upon submission, the query is processed by the servlet, which simulates a response and saves the conversation in the database. Users can view their chat history, which is retrieved from the database and displayed on the interface.

Chatbot Output –

Initial page –



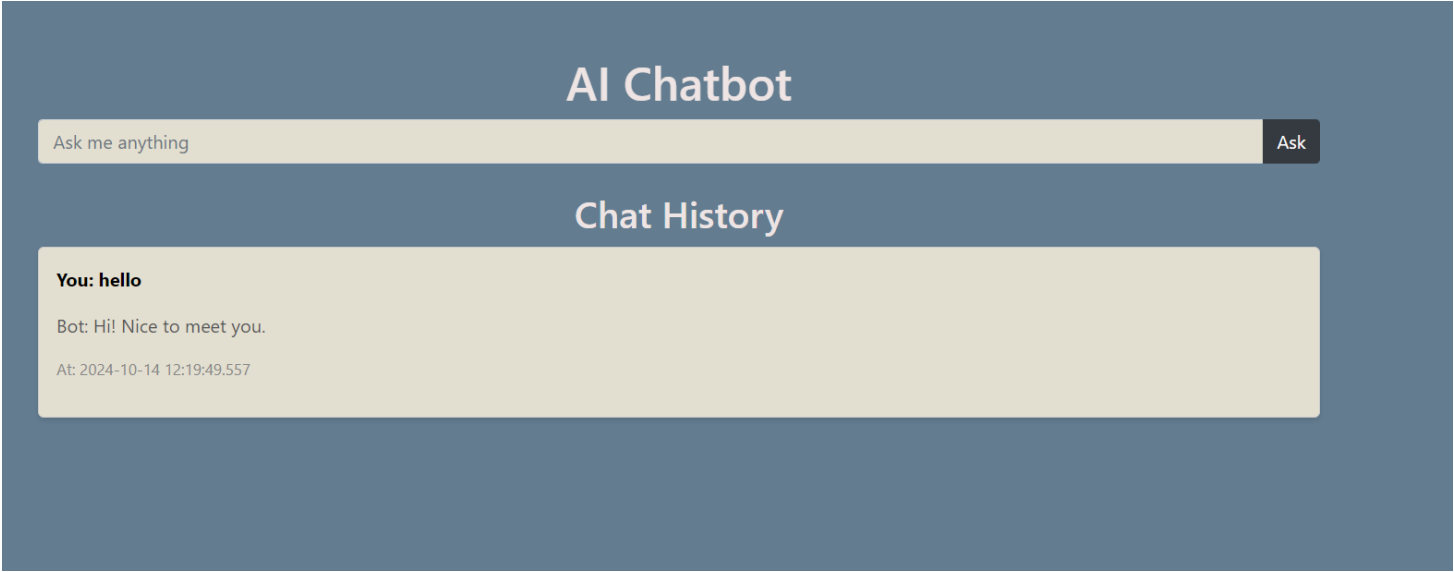
AI Chatbot

Ask me anything Ask

Chat History

You:
Bot:
At:

Response –



Response stored in database -

Result Grid

Filter Rows:

Edit:

Export/Import

	id	user_query	chatbot_response	timestamp
▶	1	hello	Hi! Nice to meet you.	2024-10-14 12:19:50
✱	NULL	NULL	NULL	NULL