

ARCHITECTURE & WORKING OF THE PROJECT

- The architecture followed in this project was Event-Driven with RESTful CRUD operations on MVC (Model - View - Controller) framework.
- OpenAI API was integrated with Socket.io for real time communication and the data was stored in MongoDB
- Events were created for the bilateral communication between the user and the OpenAI API.
- Specific events triggered prompt generation to keep the conversation between the user and the bot alive.
- A user comes on the application, the connection is established and he generates prompts by giving data to the OpenAI API and the response gets back in real time. The conversation that user had with the bot is getting stored on MongoDB and can be easily retrieved by the getHistory end point. (shown below).
- 2 CRUD endpoints were created for sending the data to the api to retrieve information about the stuff that the user asks. Second was created to retrieve user chat history.

Postman Requests

1) getChatHistory

- Endpoint: baseUrl/api/chat/history
- Method: GET
- Response:

```
{
  "success": true,
  "messages": [
    {
      "_id": "656256a75dfed9d9ae3eb007",
      "content": "The father of the atomic bomb is J. Robert Oppenheimer. He was the scientific director of the Manhattan Project, which developed the first atomic bombs during World War II.",
      "senderType": "chatbot",
      "timestamp": "2023-11-25T20:18:47.640Z",
      "__v": 0
    },
    {
      "_id": "656256a75dfed9d9ae3eb006",
```

```

    "content": "who is the father of atomic bomb",
    "senderType": "user",
    "timestamp": "2023-11-25T20:18:47.639Z",
    "__v": 0
  },
  {
    "_id": "6562290f536db7342ec40e47",
    "content": "hello",
    "senderType": "user",
    "timestamp": "2023-11-25T17:04:15.549Z",
    "__v": 0
  },
  {
    "_id": "6562290f536db7342ec40e48",
    "content": "Hi there! How can I assist you today?",
    "senderType": "chatbot",
    "timestamp": "2023-11-25T17:04:15.549Z",
    "__v": 0
  }
]
}

```

2) sendChat

- Endpoint: baseUrl/api/chat/send
- Method: POST
- Headers: Content-Type : application/json
- Body:

```

{
  "message": "who is the father of atomic bomb"
}

```

- Response:

```

{
  "success": true,
  "userMessage": {
    "content": "who is the father of atomic bomb",
    "senderType": "user",
    "_id": "656256a75dfed9d9ae3eb006",
    "timestamp": "2023-11-25T20:18:47.639Z",
    "__v": 0
  },
  "chatbotMessage": {

```

```
"content": "The father of the atomic bomb is J. Robert Oppenheimer. He was the scientific  
director of the Manhattan Project, which developed the first atomic bombs during World War  
II.",  
  "senderType": "chatbot",  
  "_id": "656256a75dfed9d9ae3eb007",  
  "timestamp": "2023-11-25T20:18:47.640Z",  
  "__v": 0  
}  
}
```

Setup of the application:

/chatbot

```
| |-- /controllers  
| | |-- chatController.js  
| |-- /models  
| | |-- Message.js  
| |-- /routes  
| | |-- chatRoutes.js  
| |-- server.js  
| |-- /public  
| | |-- index.html  
| | |-- styles.css  
| | |-- client.js  
|-- /node_modules  
|-- .env  
|-- .gitignore  
|-- package.json  
|-- README.md
```