PROGRAM 1-DYNAMIC CHATBOT

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Chat Module</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="chat-container">
    <div class="chat-box" id="chat-box"></div>
    <input type="text" id="user-input" placeholder="Type your message...">
    <button onclick="sendMessage()">Send</button>
  </div>
<script src="script.js"></script>
</body>
</html>
```

script.js

```
const chatBox = document.getElementById('chat-box');
const userInput = document.getElementById('user-input');

// Predefined question-answer pairs
const predefinedQA = {
```

```
  "How are you?": "I'm good, thank you!",
  "What is your name?": "My name is ChatBot.",
  "What can you do?": "I can answer predefined questions.",
  "Goodbye": "Goodbye! Have a nice day!",
  "How was your day?": "It’s nearing the end of the day, your phone dings and a “How was your day?” text stares back at you. Maybe it’s from your crush, best friend, or even your parent. They just want to check in, but how do you respond? Whether you’ve had a good or bad day, we’ve got you covered! Keep reading to learn how you can respond on any type of day to just about anyone."
};

function sendMessage() {
  const userMessage = userInput.value.trim();

  // Display user message in the chat box
  displayMessage(userMessage);

  // If the user's message matches a predefined question, get the answer
  const answer = predefinedQA[userMessage];
  if (answer) {
    displayMessage(answer);
  } else {
    displayMessage("I'm sorry, I don't understand.");
  }

  // Clear user input
  userInput.value = '';
}
```

```
function displayMessage(message) {
  const messageDiv = document.createElement('div');
  messageDiv.textContent = message;
  chatBox.appendChild(messageDiv);

  // Scroll to bottom
  chatBox.scrollTop = chatBox.scrollHeight;
}
```

Styles.css

```
.chat-container {
  max-width: 400px;
  margin: 20px auto;
  padding: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

.chat-box {
  height: 300px;
  overflow-y: scroll;
  border: 5px dashed #ccc;
  border-radius: 5px;
  padding: 10px;
  margin-bottom: 10px;
```

```
}
input[type="text"] {
  width: calc(100% - 60px);
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 5px;
}

button {
  padding: 8px 15px;
  background-color: #007bff;
  color: #fff;
  border: none;
  border-radius: 5px;
  cursor: pointer;
}
```

PROGRAM 2- VOTING APPLICATION

App.js

```
import React, { useState } from 'react';
import './App.css';
function App() {
const [votes, setVotes] = useState({ option1: 0, option2: 0, option3: 0, option4: 0, option5:0 });
const handleVote = (option) => {
setVotes((prevVotes) => ({
```

```
 ...prevVotes,
 [option]: prevVotes[option] + 1,
 }));
};
return (
<div className="App">
   <h1>Voting Application</h1>
   <div className="options">
      <table border="2">
         <tr>
            <th>S.No</th>
            <th>Party Name</th>
            <th>Party Symbol</th>
            <th>Button</th>
            <th>Count</th>
         </tr>
         <tr>
            <td>1</td>
            <td>BJP</td>
            <td><img
src="https://th.bing.com/th/id/OIP.stuO9HtrREb2xPI9Tlu0LgHaHr?rs=1&pid=ImgDetMain"
alt="abc"></img></td>
            <td><button onClick={()=>handleVote('option1')}>button</button></td>
            <td><span> {votes.option1}</span></td>
         </tr>
         <tr>
```

```
            <td>2</td>

            <td>Congress</td>

            <td><img
src="https://th.bing.com/th/id/OIP.9KAiukGd7AdZcXag2a02ugAAAA?rs=1&pid=ImgDetMain"
alt=""></img></td>

            <td><button onClick={()=>handleVote('option2')}>button</button></td>

            <td><span> {votes.option2}</span></td>

        </tr>

        <tr>

            <td>3</td>

            <td>JDS</td>

            <td><img src="https://th.bing.com/th/id/OIP.R-
UMqTwZZ4fcDi63nGJwqgHaFP?rs=1&pid=ImgDetMain" alt=""></img></td>

            <td><button onClick={()=>handleVote('option3')}>button</button></td>

            <td><span> {votes.option3}</span></td>

        </tr>

        <tr>

            <td>4</td>

            <td>BSP</td>

            <td><img src="https://allpngfree.com/thumbnail/bahujan-samaj-party-logo-png-images--
thumbnail-1673701310.jpg" alt=""></img></td>

            <td><button onClick={()=>handleVote('option4')}>button</button></td>

            <td><span> {votes.option4}</span></td>

        </tr>

        <tr>

            <td>5</td>

            <td>CPI</td>
```

```
            <td><img src="https://m.media-amazon.com/images/I/61VJoR4iywL.SL1500.jpg"
alt=""></img></td>

            <td><button onClick={()=>handleVote('option5')}>button</button></td>

            <td><span> {votes.option5}</span></td>

          </tr>

       </table>

    </div>

</div>

);

}

export default App;
```

app.css

```
.App {
  text-align: center;
}


.App-logo {
  height: 40vmin;
  pointer-events: none;
}


@media (prefers-reduced-motion: no-preference) {
  .App-logo {
    animation: App-logo-spin infinite 20s linear;
  }
}
```

```
.App-header {
  background-color: #282c34;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  font-size: calc(10px + 2vmin);
  color: white;
}
.App-link {
  color: #61dafb;
}
h1{
  color: blue;
  font-weight: bold;
}
img{
  width: 65px;
}
table{
  margin-left: auto;
  margin-right: auto;
  width: 500px;
  height: 500px;
```

```
  border-color: black;
  border-radius: 5px;
}
*{
  background-color: antiquewhite;
}
td,th{
  font-weight: bold;
  font-size: larger;
}
button{
  font-weight: bold;
  font-size: medium;
}
@keyframes App-logo-spin {
  from {
    transform: rotate(0deg);
  }
  to {
    transform: rotate(360deg);
  }
}
```

Index.js(No changes)

```
import React from 'react';
import ReactDOM from 'react-dom/client';
```

```
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
 <React.StrictMode>
  <App />
 </React.StrictMode>
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Program 3- Blog Application

First.js

```
import React from "react";
import "./first.css";
import Posts from "./second";
const Post = ({ post: { title, body, imgUrl, author }, index }) =>
{
 return (
 <div className="post-container">
 <h1 className="heading">{title}</h1>
 <img className="image" src={imgUrl} alt="post" />
```

```
<p>{body}</p>
<div className="info">
<h4>Written by: {author}</h4>
</div>
</div>
);
};
export default Post;
```

Second.js

```
import React from "react";
import "./first.css";
import Post from "./first.js";
const Posts = () => { const blogPosts = [
{
title: "JAVASCRIPT",
body: `JavaScript is the world most popular lightweight, interpreted compiled programming
language. It is also known as scripting language for web pages. It is well-known for
the development of web pages, many non-browser environments also use it. JavaScript can be
used for Client-side developments as well as Server-side developments`,
author: "Nishant Singh ", imgUrl:
"https://media.geeksforgeeks.org/img-practice/banner/diving-into-excel-thumbnail.png",
},
{
title: "Data Structure ",
```

body: `There are many real-life examples of

a stack. Consider an example of plates stacked over one another in the canteen. The plate which is at

the top is the first one to be removed, i.e. the plate which has been placed at the bottommost

position remains in the stack for the longest period of time. So, it

can be simply seen to follow LIFO(Last In First Out)/FILO(First In Last Out) order.`, author: "Suresh

Kr",

imgUrl:

"https://media.geeksforgeeks.org/img-practice/banner/coa-gate-2022-thumbnail.png",

},

{

title: "Algorithm",

body: `The word Algorithm means “a process or set of rules to be followed in calculations

or other problem-solving operations”. Therefore Algorithm refers to a set of rules/instructions

that step-by-step define how a work is to be executed upon in order to get the expected results. `,

author: "Monu Kr", imgUrl:

"https://media.geeksforgeeks.org/img-practice/banner/google-test-series-thumbnail.png",

},

{

title: "Computer Network",

body: `An interconnection of multiple devices, also known as hosts, that are connected using

multiple paths for the purpose of sending/ receiving data media. Computer networks can also

include multiple devices/mediums which help in the communication between two different devices;

these are known as Network devices and include things such as routers, switches, hubs, and bridges.

`,

author: "Sonu Kr", imgUrl:

"https://media.geeksforgeeks.org/img-practice/banner/cp-maths-java-thumbnail.png",

},

];

return (

{blogPosts.map((post, index) => (

<Post index={index} post={post} />

))}

);

};

export default Posts;

first.css

```
body {
 background-color: #0e9d57;
 }
 .posts-container { display: flex;
 justify-content: center; align-items: center;
 }
```

```
.post-container { background: #e2e8d5;
display: flex;
flex-direction: column; padding: 3%;
margin: 0 2%;
height: 40%;
}
.heading { height: 126px;
text-align: center; display: flex;
align-items: center;
}
.image {
width: 100%; height: 210px;
}
```

Index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import Posts from './second';
import reportWebVitals from './reportWebVitals';
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <Posts />
```

```
  </React.StrictMode>
);
// If you want to start measuring performance in your app, pass a function
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA-vitals
reportWebVitals();
```

Program 4-Mongodb

```
const { MongoClient, ObjectID, ObjectId } = require(&#39;mongodb&#39;);

// Connection URL
const url = &#39;mongodb://localhost:27017&#39;;

// Database Name
const dbName = &#39;College&#39;;

// Create a new MongoClient
const client = new MongoClient(url, { useNewUrlParser: true,
useUnifiedTopology: true });

// Connect to the MongoDB server
async function connectDB() {
   try {
      await client.connect();
```

```
        console.log(&#39;Connected to the database&#39;);
    } catch (error) {
        console.error(&#39;Error connecting to the database:&#39;, error);
    }
}
   // Insert operation (Create)
async function insertStudent(student) {
    const db = client.db(dbName);
    try {const result = await db.collection(&#39;students&#39;).insertOne(student);
    console.log(Student with id ${result.insertedId} inserted successfully);
    } catch (err) {
      console.error(&#39;Error inserting student:&#39;, err);
    }
}


    // Update operation
    async function updateStudent() {
        const db = client.db(dbName);
        try {
        const result = await db.collection(&#39;students&#39;).updateOne(
            {_id:new
ObjectId(&#39;66667a24ace74f68b2cdcdf6&#39;)},{$set:{Dept:&quot;CSE&quot;}});
            console.log(Student data updated successfully);
        } catch (err) {
```

```
        console.error(&#39;Error updating student:&#39;, err);
      }
   }

// Find all students
async function findAllStudents() {
      const db = client.db(dbName);
      try {
      const students = await db.collection(&#39;students&#39;).find({}).toArray();
      console.log(&#39;All students:&#39;, students);
      } catch (err) {
        console.error(&#39;Error finding students:&#39;, err);}
   }

// Delete operation
async function deleteStudent(Id) {
   const db = client.db(dbName);
   try {
   const result = await db.collection(&#39;students&#39;).deleteOne({ _id:new
ObjectId(Id) });
   console.log(Student with id ${Id} deleted successfully);
   } catch (err) {
     console.error(&#39;Error deleting student:&#39;, err);
   }
```

```
}
  // Perform operations (uncomment as needed for demonstration)
  connectDB()
  .then(async () =&gt; {

  // Insert a student
  const exampleStudent = { name: &#39;Monisha&#39;, age: 18, cgpa:6.38,
Dept:&quot;CSE&quot;};
  await insertStudent(exampleStudent);

  // Find all students
  await findAllStudents();

  // Update a student
  await updateStudent();

  // Delete a student
  const studentIdToDelete = &#39;666bea5ca3a164ff0e37ba34&#39;; // Replace with an
existing student id
  await deleteStudent(studentIdToDelete);

  // Close the connection
  client.close();
});
```

Program 5 –Jquery

Pg.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Password Strength Checker</title>
<style>
 .password-strength {
 margin-top: 10px;
 padding: 5px;
 width: 250px;
 font-size: 14px;
 }
 .strength-meter {
 height: 20px;
 margin-top: 15px;
 }
 .very-weak { background-color: #ff4d4d; }
 .weak { background-color: #ffa500; }
 .medium { background-color: #ffd700; }
 .strong { background-color: #7fff00; }
 .very-strong { background-color: #00ff00; }
```

```
</style>
</head>
<body>
<div>
 <label for="password">Enter your password:</label>
 <input type="password" id="password" name="password" />
</div>
<div class="password-strength">
 <div class="strength-meter"></div>
</div>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script src="strengthchecker.js"></script>
</body>
</html>
```

Strengthchecker.js

```
$(document).ready(function() {
 $('#password').on('input', function() {
 var password = $(this).val();
 var meter = $('.strength-meter');
 var strength = 0;
 var progressClass = '';

 // Check length
```

```
if (password.length > 8) {
strength += 1;
}

// Check if password contains uppercase letter
if (password.match(/[A-Z]/)) {
strength += 1;
}

// Check if password contains lowercase letter
if (password.match(/[a-z]/)) {
strength += 1;
}

// Check if password contains number
if (password.match(/\d/)) {
strength += 1;
}

// Check if password contains special character
if (password.match(/[!@#$%^&*(),.?":{}|<>]/)) {
strength += 1;
}
```

```
// Determine progress bar class
switch(strength) {
case 0: progressClass = 'very-weak'; break;
case 1: progressClass = 'weak'; break;
case 2: progressClass = 'medium'; break;
case 3: progressClass = 'strong'; break;
case 4: progressClass = 'very-strong'; break;
default: progressClass = 'very-weak';
}
// Update progress bar
meter.removeClass().addClass('strength-meter').addClass(progressClass);
});
});
```