



SHAKTI 2.0 - Setup Guide (Part 3)

📁 Complete CSV Dataset File

16. src/data/fullTrainedDataset.js

This file contains the complete 2000+ patterns from your uploaded CSV:

```
javascript
```

```
// Full trained dataset from shakti_2.0_dataset.csv
export const fullTrainedDataset = {
  greeting: {
    inputs: [
      'hello', 'hi', 'hey', 'good morning', 'good evening', 'hey there',
      'greetings', 'hi there', 'hello bot', 'hi bot', 'hey bot',
      'hello mam', 'hi mam', 'hey mam', 'hello sir', 'hi sir', 'hey sir',
      'hello buddy', 'hi buddy', 'hey buddy', 'hello friend', 'hi friend',
      'hey friend', 'hello please', 'hi please', 'hey please',
      'hello today', 'hi today', 'hey today', 'hello now', 'hi now',
      'good morning sir', 'good morning mam', 'good morning bot',
      'good morning friend', 'good morning buddy', 'good morning today',
      'good evening sir', 'good evening mam', 'good evening bot',
      'good evening friend', 'good evening buddy', 'good evening today',
      'greetings sir', 'greetings mam', 'greetings bot', 'greetings friend',
      'hey there sir', 'hey there mam', 'hey there bot', 'hey there friend',
      'hi there sir', 'hi there mam', 'hi there bot', 'hi there friend',
      'yo hello', 'yo hi', 'yo hey', 'yo greetings',
      'listen hello', 'listen hi', 'listen hey',
      'well hello', 'well hi', 'well hey',
      'ok hello', 'ok hi', 'ok hey',
      'so hello', 'so hi', 'so hey',
      'actually hello', 'actually hi', 'actually hey',
      'umm hello', 'umm hi', 'umm hey',
      'please hello', 'please hi', 'please hey'
    ],
    response: 'Hello! How can I help you today?'
  },
  farewell: {
    inputs: [
      'bye', 'goodbye', 'see you', 'cya', 'good night', 'i have to go',
      'talk to you later', 'bye now', 'goodbye now', 'see you now',
      'bye bot', 'goodbye bot', 'see you bot', 'cya bot',
      'bye sir', 'goodbye sir', 'see you sir', 'cya sir',
      'bye mam', 'goodbye mam', 'see you mam', 'cya mam',
      'bye friend', 'goodbye friend', 'see you friend', 'cya friend',
      'bye buddy', 'goodbye buddy', 'see you buddy', 'cya buddy',
      'bye today', 'goodbye today', 'see you today', 'cya today',
      'bye please', 'goodbye please', 'see you please', 'cya please',
      'good night sir', 'good night mam', 'good night bot',
      'good night friend', 'good night buddy', 'good night today',
      'i have to go sir', 'i have to go mam', 'i have to go bot',
    ]
  }
}
```

'i have to go friend', 'i have to go buddy', 'i have to go now',
'talk to you later sir', 'talk to you later mam', 'talk to you later bot',
'yo bye', 'yo goodbye', 'yo see you', 'yo cya',
'listen bye', 'listen goodbye', 'listen see you',
'well bye', 'well goodbye', 'well see you',
'ok bye', 'ok goodbye', 'ok see you',
'so bye', 'so goodbye', 'so see you',
'actually bye', 'actually goodbye', 'actually see you',
'umm bye', 'umm goodbye', 'umm see you'

],

response: 'Goodbye! Have a wonderful day.'

},

identity: {

inputs: [

'who are you', 'what is your name', 'are you a bot', 'what are you',
'introduce yourself', 'who are you bot', 'who are you sir',
'who are you mam', 'who are you friend', 'who are you buddy',
'who are you today', 'who are you now', 'who are you actually',
'what is your name bot', 'what is your name sir', 'what is your name mam',
'what is your name friend', 'what is your name today', 'what is your name now',
'are you a bot sir', 'are you a bot mam', 'are you a bot friend',
'are you a bot buddy', 'are you a bot today', 'are you a bot now',
'what are you bot', 'what are you sir', 'what are you mam',
'what are you friend', 'what are you buddy', 'what are you today',
'introduce yourself bot', 'introduce yourself sir', 'introduce yourself mam',
'introduce yourself friend', 'introduce yourself buddy', 'introduce yourself now',
'yo who are you', 'yo what is your name', 'yo are you a bot',
'listen who are you', 'listen what is your name',
'well who are you', 'well what is your name',
'ok who are you', 'ok what is your name',
'so who are you', 'so what is your name',
'actually who are you', 'actually what is your name',
'umm who are you', 'umm what is your name',
'please who are you', 'please what is your name'

],

response: 'I am SHAKTI 2.0, a human interaction dataset model.'

},

wellbeing: {

inputs: [

'how are you', 'how is it going', 'how are things', 'how do you feel',
'whats up', 'what is up', "what's up",
'how are you bot', 'how are you sir', 'how are you mam'

'how are you friend', 'how are you buddy', 'how are you today',
'how are you now', 'how are you actually', 'how are you please',
'how is it going bot', 'how is it going sir', 'how is it going mam',
'how is it going friend', 'how is it going today', 'how is it going now',
'how are things bot', 'how are things sir', 'how are things mam',
'how are things friend', 'how are things today', 'how are things actually',
'how do you feel bot', 'how do you feel sir', 'how do you feel mam',
'how do you feel friend', 'how do you feel buddy', 'how do you feel now',
'whats up bot', 'whats up sir', 'whats up mam', 'whats up friend',
'whats up buddy', 'whats up today', 'whats up now', 'whats up actually',
'yo how are you', 'yo how is it going', 'yo whats up',
'listen how are you', 'listen how is it going',
'well how are you', 'well how is it going',
'ok how are you', 'ok how is it going', 'ok whats up',
'so how are you', 'so how is it going', 'so whats up',
'actually how are you', 'actually how is it going',
'umm how are you', 'umm how is it going', 'umm whats up'

],

response: "I'm functioning perfectly. How can I help you?"

},

capabilities: {

inputs: [

'what can you do', 'help me', 'what are your features', 'how do you work',
'what can you do bot', 'what can you do sir', 'what can you do mam',
'what can you do friend', 'what can you do buddy', 'what can you do please',
'what can you do today', 'what can you do now', 'what can you do actually',
'help me bot', 'help me sir', 'help me mam', 'help me friend',
'help me buddy', 'help me please', 'help me now', 'help me actually',
'what are your features bot', 'what are your features sir',
'what are your features mam', 'what are your features friend',
'what are your features today', 'what are your features now',
'how do you work bot', 'how do you work sir', 'how do you work mam',
'how do you work friend', 'how do you work buddy', 'how do you work today',
'yo what can you do', 'yo help me', 'yo how do you work',
'listen what can you do', 'listen help me',
'well what can you do', 'well help me',
'ok what can you do', 'ok help me',
'so what can you do', 'so help me',
'actually what can you do', 'actually help me',
'umm what can you do', 'umm help me',
'please what can you do', 'please help me'

],

response: 'I can answer questions and chat with you.'

},

gratitude: {

inputs: [

 'thanks', 'thank you', 'thx', 'thanks a lot', 'appreciate it',
 'thanks bot', 'thanks sir', 'thanks mam', 'thanks friend',
 'thanks buddy', 'thanks today', 'thanks now', 'thanks actually',
 'thank you bot', 'thank you sir', 'thank you mam', 'thank you friend',
 'thank you buddy', 'thank you today', 'thank you now', 'thank you actually',
 'thx bot', 'thx sir', 'thx mam', 'thx friend', 'thx buddy',
 'thx today', 'thx now', 'thx please',
 'thanks a lot bot', 'thanks a lot sir', 'thanks a lot mam',
 'thanks a lot friend', 'thanks a lot today', 'thanks a lot now',
 'appreciate it bot', 'appreciate it sir', 'appreciate it mam',
 'appreciate it friend', 'appreciate it buddy', 'appreciate it please',
 'yo thanks', 'yo thank you', 'yo thx', 'yo appreciate it',
 'listen thanks', 'listen thank you',
 'well thanks', 'well thank you',
 'ok thanks', 'ok thank you', 'ok thx',
 'so thanks', 'so thank you',
 'actually thanks', 'actually thank you',
 'umm thanks', 'umm thank you', 'umm thx',
 'please thanks', 'please thank you'

],

response: 'You are very welcome!'

},

compliment: {

inputs: [

 'you are smart', 'good job', 'well done', 'you are amazing', 'nice work',
 'you are smart bot', 'you are smart sir', 'you are smart mam',
 'you are smart friend', 'you are smart buddy', 'you are smart today',
 'good job bot', 'good job sir', 'good job mam', 'good job friend',
 'good job buddy', 'good job today', 'good job now', 'good job actually',
 'well done bot', 'well done sir', 'well done mam', 'well done friend',
 'well done buddy', 'well done today', 'well done now', 'well done please',
 'you are amazing bot', 'you are amazing sir', 'you are amazing mam',
 'you are amazing friend', 'you are amazing buddy', 'you are amazing today',
 'nice work bot', 'nice work sir', 'nice work mam', 'nice work friend',
 'nice work buddy', 'nice work today', 'nice work now', 'nice work actually',
 'yo good job', 'yo well done', 'yo nice work', 'yo you are amazing',
 'listen good job', 'listen well done',
 'well good job', 'well well done',
 'ok good job', 'ok well done', 'ok nice work',

'so good job', 'so well done',
'actually good job', 'actually well done', 'actually you are amazing',
'umm good job', 'umm well done'

],

response: 'Thank you! I appreciate the positive feedback.'

},

insult: {

inputs: [

'you are stupid', 'you are dumb', 'bad bot', 'useless', 'you suck',
'you are stupid bot', 'you are stupid sir', 'you are stupid mam',
'you are stupid friend', 'you are stupid buddy', 'you are stupid today',
'you are dumb bot', 'you are dumb sir', 'you are dumb mam',
'you are dumb friend', 'you are dumb buddy', 'you are dumb today',
'bad bot sir', 'bad bot mam', 'bad bot friend', 'bad bot buddy',
'bad bot today', 'bad bot now', 'bad bot please', 'bad bot actually',
'useless bot', 'useless sir', 'useless mam', 'useless friend',
'useless buddy', 'useless today', 'useless now', 'useless actually',
'you suck bot', 'you suck sir', 'you suck mam', 'you suck friend',
'you suck buddy', 'you suck today', 'you suck now', 'you suck actually',
'yo you are stupid', 'yo you are dumb', 'yo bad bot', 'yo useless',
'listen you are stupid', 'listen bad bot', 'listen useless',
'well you are stupid', 'well bad bot',
'ok you are stupid', 'ok bad bot', 'ok useless',
'so you are stupid', 'so bad bot',
'actually you are stupid', 'actually bad bot',
'umm you are stupid', 'umm bad bot', 'umm useless'

],

response: 'I apologize if I disappointed you. I am still learning.'

},

love: {

inputs: [

'i love you', 'i like you', 'will you marry me', 'you are cute',
'i love you bot', 'i love you sir', 'i love you mam',
'i love you friend', 'i love you buddy', 'i love you today',
'i love you now', 'i love you please', 'i love you actually',
'i like you bot', 'i like you sir', 'i like you mam',
'i like you friend', 'i like you buddy', 'i like you today',
'i like you now', 'i like you please', 'i like you actually',
'will you marry me bot', 'will you marry me sir', 'will you marry me mam',
'will you marry me friend', 'will you marry me buddy', 'will you marry me today',
'you are cute bot', 'you are cute sir', 'you are cute mam',
'you are cute friend', 'you are cute buddy', 'you are cute today'

```
'you are cute now', 'you are cute please', 'you are cute actually',
'yo i love you', 'yo i like you', 'yo you are cute',
'listen i love you', 'listen i like you',
'well i love you', 'well i like you',
'ok i love you', 'ok i like you', 'ok you are cute',
'so i love you', 'so i like you',
'actually i love you', 'actually i like you',
'umm i love you', 'umm i like you',
'please i love you', 'please i like you', 'please will you marry me'
],
response: 'That is very kind of you to say!'
},
```

```
confusion: {
```

```
inputs: [
```

```
'i dont understand', 'what do you mean', 'can you explain', 'i am confused',
'i dont understand bot', 'i dont understand sir', 'i dont understand mam',
'i dont understand friend', 'i dont understand buddy', 'i dont understand today',
'i dont understand now', 'i dont understand actually', 'i dont understand please',
'what do you mean bot', 'what do you mean sir', 'what do you mean mam',
'what do you mean friend', 'what do you mean buddy', 'what do you mean today',
'what do you mean now', 'what do you mean please', 'what do you mean actually',
'can you explain bot', 'can you explain sir', 'can you explain mam',
'can you explain friend', 'can you explain buddy', 'can you explain today',
'can you explain now', 'can you explain please', 'can you explain actually',
'i am confused bot', 'i am confused sir', 'i am confused mam',
'i am confused friend', 'i am confused buddy', 'i am confused today',
'i am confused now', 'i am confused actually', 'i am confused please',
'yo i dont understand', 'yo what do you mean', 'yo can you explain',
'listen i dont understand', 'listen what do you mean',
'well i dont understand', 'well what do you mean',
'ok i dont understand', 'ok what do you mean', 'ok can you explain',
'so i dont understand', 'so what do you mean',
'actually i dont understand', 'actually what do you mean',
'umm i dont understand', 'umm what do you mean'
```

```
],
```

```
response: 'I apologize. Let me try to be clearer.'
```

```
}
```

```
};
```

```
// Get total count of patterns
```

```
export const getPatternCount = () => {
  let count = 0;
  Object.values(fullTrainedDataset).forEach(intent => {
```

```
    count += intent.inputs.length;
  });
  return count;
};

// Get all intent categories
export const getIntentCategories = () => {
  return Object.keys(fullTrainedDataset);
};
```

🔧 Advanced Configuration Files

17. src/config/config.js

javascript

```
// Application Configuration
export const APP_CONFIG = {
    appName: 'SHAKTI 2.0',
    version: '2.0.0',
    description: 'Advanced AI Chatbot with State Space Fusion',

    // State Space Configuration
    stateSpace: {
        maxContextHistory: 10,
        maxSemanticMemory: 20,
        maxAttentionWeights: 10
    },

    // NLP Configuration
    nlp: {
        matchThreshold: 0.7,
        fillerWords: [
            'actually', 'well', 'umm', 'yo', 'hey', 'listen',
            'ok', 'so', 'please', 'now', 'today', 'friend',
            'buddy', 'sir', 'mam', 'bot'
        ]
    },

    // API Configuration
    api: {
        defaultProvider: 'gemini',
        timeout: 30000,
        maxRetries: 3
    }

    // UI Configuration
    ui: {
        defaultTheme: 'dark',
        animationDuration: 300,
        maxMessageLength: 5000
    }
};

// Theme Colors
export const THEMES = {
    dark: {
        name: 'Dark',
        bgGradient: 'from-slate-900 via-purple-900 to-slate-900',
        fgColor: 'white',
        accentColor: '#4DB6AC'
    }
};
```

```
cardBg: 'bg-slate-800',
textPrimary: 'text-white',
textSecondary: 'text-slate-400',
inputBg: 'bg-slate-700 border-slate-600',
userMsgBg: 'bg-purple-600',
botMsgBg: 'bg-slate-800 border-slate-700',
btnBg: 'bg-purple-600 hover:bg-purple-700',
accentColor: 'text-purple-400',
focusRing: 'focus:ring-purple-500'

},
light: {
  name: 'Light',
  bgGradient: 'from-sky-400 via-blue-400 to-sky-500',
  cardBg: 'bg-white',
  textPrimary: 'text-gray-800',
  textSecondary: 'text-gray-600',
  inputBg: 'bg-blue-50 border-blue-200',
  userMsgBg: 'bg-blue-600',
  botMsgBg: 'bg-white border-blue-100',
  btnBg: 'bg-blue-600 hover:bg-blue-700',
  accentColor: 'text-blue-600',
  focusRing: 'focus:ring-blue-500'
}
};
```

18. src/hooks/useLocalStorage.js

```
javascript
```

```
import { useState, useEffect } from 'react';

export const useLocalStorage = (key, initialValue) => {
  const [storedValue, setStoredValue] = useState(() => {
    try {
      const item = window.localStorage.getItem(key);
      return item ? JSON.parse(item) : initialValue;
    } catch (error) {
      console.error('Error reading localStorage:', error);
      return initialValue;
    }
  });
  const setValue = (value) => {
    try {
      const valueToStore = value instanceof Function ? value(storedValue) : value;
      setStoredValue(valueToStore);
      window.localStorage.setItem(key, JSON.stringify(valueToStore));
    } catch (error) {
      console.error('Error setting localStorage:', error);
    }
  };
  return [storedValue, setValue];
};
```

19. src/hooks/useChat.js

```
javascript
```

```
import { useState, useCallback } from 'react';
import { findTrainedResponse } from '../services/nlpService';
import { callAPI } from '../services/apiService';
import { updateStateSpace } from '../utils/stateSpace';
import { fullTrainedDataset } from '../data/fullTrainedDataset';

export const useChat = (apiProvider, apiKey, apiEndpoint) => {
  const [messages, setMessages] = useState([
    { role: 'assistant', content: "Hello! I'm SHAKTI 2.0, your advanced AI assistant. How can I help you today?" }
  ]);
  const [loading, setLoading] = useState(false);
  const [stateSpace, setStateSpace] = useState({
    contextHistory: [],
    semanticMemory: [],
    attentionWeights: []
  });
  const [error, setError] = useState(null);

  const buildContextPrompt = useCallback((userMessage) => {
    const recentContext = stateSpace.contextHistory.slice(-5)
      .map(ctx => `User: ${ctx.user}\nAssistant: ${ctx.assistant}`)
      .join('\n\n');

    const semanticContext = stateSpace.semanticMemory.slice(-10)
      .map(m => m.entity)
      .join('; ');
  });

  return 'You are SHAKTI 2.0, a human interaction dataset model.\n\nRecent conversation context:\n' +
    recentContext + '\n\nKey semantic entities in memory: ' +
    (semanticContext || 'none yet') + '\n\nCurrent user message: ' +
    userMessage + '\n\nProvide a helpful, context-aware response.';
}, [stateSpace]);

const sendMessage = useCallback(async (userMessage) => {
  if (!userMessage.trim() || loading) return;

  setMessages(prev => [...prev, { role: 'user', content: userMessage }]);
  setLoading(true);
  setError(null);

  try {
    // Check trained dataset first
    const trainedMatch = findTrainedResponse(userMessage, fullTrainedDataset);
  
```

```
if (trainedMatch) {
    await new Promise(resolve => setTimeout(resolve, 500));
    setMessages(prev => [...prev, { role: 'assistant', content: trainedMatch.response }]);
    setStateSpace(prev => updateStateSpace(prev, userMessage, trainedMatch.response));
} else {
    // Fall back to API
    const contextPrompt = buildContextPrompt(userMessage);
    const assistantMessage = await callAPI(apiProvider, apiKey, apiEndpoint, contextPrompt);

    setMessages(prev => [...prev, { role: 'assistant', content: assistantMessage }]);
    setStateSpace(prev => updateStateSpace(prev, userMessage, assistantMessage));
}

} catch (err) {
    setError(err.message);
    setMessages(prev => [...prev, {
        role: 'assistant',
        content: 'Error: ' + err.message + '. Please check your API configuration.'
    }]);
} finally {
    setLoading(false);
}
}, [loading, apiProvider, apiKey, apiEndpoint, buildContextPrompt]);

const clearChat = useCallback(() => {
    setMessages([
        { role: 'assistant', content: "Hello! I'm SHAKTI 2.0, your advanced AI assistant. How can I help you today?" }
    ]);
    setStateSpace({
        contextHistory: [],
        semanticMemory: [],
        attentionWeights: []
    });
    setError(null);
}, []);

return {
    messages,
    loading,
    stateSpace,
    error,
    sendMessage,
    clearChat
}
```

```
};  
};
```

20. src/components/MessageBubble.jsx

```
jsx
```

```
import React from 'react';

const MessageBubble = ({ message, darkMode }) => {
  const isUser = message.role === 'user';

  const userMsgBg = darkMode ? 'bg-purple-600' : 'bg-blue-600';
  const botMsgBg = darkMode ? 'bg-slate-800 border-slate-700' : 'bg-white border-blue-100';
  const textPrimary = darkMode ? 'text-white' : 'text-gray-800';

  return (
    <div className={'flex ' + (isUser ? 'justify-end' : 'justify-start')}>
      <div
        className={
          'max-w-[80%] px-4 py-3 rounded-2xl ' +
          (isUser
            ? userMsgBg + ' text-white'
            : botMsgBg + '' + textPrimary + '' + (darkMode ? '' : 'shadow-sm'))
        }
      >
        <p className="whitespace-pre-wrap">{message.content}</p>
      </div>
    </div>
  );
};

export default MessageBubble;
```

21. src/components>LoadingIndicator.jsx

```
jsx
```

```
import React from 'react';

const LoadingIndicator = ({ darkMode }) => {
  const botMsgBg = darkMode ? 'bg-slate-800 border-slate-700' : 'bg-white border-blue-100';
  const dotColor = darkMode ? 'bg-purple-400' : 'bg-blue-600';

  return (
    <div className="flex justify-start">
      <div className={botMsgBg + ' px-4 py-3 rounded-2xl ' + (darkMode ? '' : 'shadow-sm')}>
        <div className="flex gap-2">
          <div
            className={'w-2 h-2 rounded-full animate-bounce ' + dotColor}
            style={{ animationDelay: '0ms' }}
          />
          <div
            className={'w-2 h-2 rounded-full animate-bounce ' + dotColor}
            style={{ animationDelay: '150ms' }}
          />
          <div
            className={'w-2 h-2 rounded-full animate-bounce ' + dotColor}
            style={{ animationDelay: '300ms' }}
          />
        </div>
      </div>
    </div>
  );
};

export default LoadingIndicator;
```

22. src/components/Header.jsx

jsx

```
import React from 'react';
import { Zap, Sun, Moon, Settings, Trash2 } from 'lucide-react';

const Header = ({  
    darkMode,  
    setDarkMode,  
    showSettings,  
    setShowSettings,  
    clearChat,  
    providerName  
}) => {  
    const cardBg = darkMode ? 'bg-slate-800' : 'bg-white';  
    const textPrimary = darkMode ? 'text-white' : 'text-gray-800';  
    const textSecondary = darkMode ? 'text-slate-400' : 'text-gray-600';  
    const accentColor = darkMode ? 'text-purple-400' : 'text-blue-600';  
    const btnStyle = darkMode  
        ? 'bg-slate-700 hover:bg-slate-600 text-purple-400'  
        : 'bg-blue-100 hover:bg-blue-200 text-blue-700';  
  
    return (  
        <div className={cardBg + '/90 backdrop-blur-sm px-6 py-4' + (darkMode ? 'border-b border-purple-500/30' : 'border-b border-gray-200/30')}>  
            <div className="flex items-center justify-between">  
                <div className="flex items-center gap-3">  
                    <Zap className={'w-8 h-8 ' + accentColor} />  
                    <div>  
                        <h1 className={'text-xl font-bold ' + textPrimary}>SHAKTI 2.0</h1>  
                        <p className={'text-xs ' + textSecondary}>  
                            {providerName} • Trained Model  
                        </p>  
                    </div>  
                </div>  
                <div className="flex gap-2">  
                    <button  
                        onClick={() => setDarkMode(!darkMode)}  
                        className={'px-4 py-2 rounded-lg flex items-center gap-2 transition-colors ' + btnStyle}>  
                        >  
                        {darkMode ? <Sun className="w-4 h-4" /> : <Moon className="w-4 h-4" />}  
                    </button>  
                    <button  
                        onClick={() => setShowSettings(!showSettings)}  
                        className={'px-4 py-2 rounded-lg flex items-center gap-2 transition-colors ' + btnStyle}>  
                        >  
                        <Settings className="w-4 h-4" />  
                </div>  
            </div>  
        </div>  
    );  
};
```

```
</button>
<button
  onClick={clearChat}
  className={'px-4 py-2 rounded-lg flex items-center gap-2 transition-colors ' + btnStyle}
>
  <Trash2 className=""
```