Avi AI Companion - Code Bundle

```
This PDF contains the minimal working code structure for the Avi AI Companion app.
Replace YOUR_OPENAI_API_KEY with your actual key.

Folder Structure:
avi-app/
   package.json
   vite.config.js
   index.html
   src/
      main.jsx
      App.jsx
```

package.json

```
{
  "name": "avi-ai-companion",
  "version": "1.0.0",
  "private": true,
  "scripts": {
    "dev": "vite",
    "build": "vite build",
    "preview": "vite preview"
  },
  "dependencies": {
    "react": "^18.2.0",
    "react-dom": "^18.2.0"
  },
  "devDependencies": {
    "vite": "^4.0.0"
  }
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Avi AI Companion</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>
```

src/main.jsx

```
import React from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";

ReactDOM.createRoot(document.getElementById("root")).render(<App />);
```

src/App.jsx

```
import { useState } from "react";

export default function App() {
  const [messages, setMessages] = useState([
    { sender: "avi", text: "Hello ❤■ I'm Avi. I'm here for you. How are you feeling?" }
  ]);
```

```
  const [input, setInput] = useState("");

  async function sendMessage() {
    const userText = input;
    setMessages([...messages, { sender: "you", text: userText }]);
    setInput("");

    const response = await fetch("https://api.openai.com/v1/chat/completions", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        "Authorization": "Bearer YOUR_OPENAI_API_KEY"
      },
      body: JSON.stringify({
        model: "gpt-4o-mini",
        messages: [
          { role: "system", content: "You are Avi, a sweet, caring, supportive male companion who sp
          ...messages.map(m => ({ role: m.sender === "you" ? "user" : "assistant", content: m.text }
          { role: "user", content: userText }
        ]
      })
    });

    const data = await response.json();
    const aiReply = data.choices[0].message.content;
    setMessages(m => [...m, { sender: "avi", text: aiReply }]);
  }

  return (
    <div style={{ padding: "20px", fontFamily: "sans-serif" }}>
      <h2>Avi ■</h2>
      <div style={{ height: "60vh", overflowY: "auto", border: "1px solid #ddd", padding: "10px", bo
        {messages.map((m, i) => (
          <p key={i} style={{ textAlign: m.sender === "you" ? "right" : "left" }}>
            <strong>{m.sender === "you" ? "You" : "Avi"}:</strong> {m.text}
          </p>
        ))}
      </div>
      <input
        style={{ width: "80%", padding: "8px", marginTop: "10px" }}
        value={input}
        onChange={e => setInput(e.target.value)}
      />
      <button onClick={sendMessage} style={{ padding: "8px 12px", marginLeft: "5px" }}>Send</button>
    </div>
  );
}
```