App.jsx

*import* React**,** { useState**,** useEffect } *from* "react"**;**

*import* io *from* "socket.io-client"**;**

*import* axios *from* "axios"**;**

*import* { MoreVertical**,** Paperclip } *from* "lucide-react"**;**

*import* "bootstrap/dist/css/bootstrap.min.css"**;**

*import* "./App.css"**;**

*// ✅ Connect to backend using Socket.io*

const socket **=** io**.**connect("http://localhost:5000")**;**

*// Dummy logged-in user (Replace this with actual authentication logic)*

const loggedInUser **=** { id**:** 999**,** name**:** "Me" }**;**

*// ✅ Define contacts*

const contacts **=** [

  { id**:** 1**,** name**:** "Alice"**,** role**:** "Frontend Developer" }**,**

  { id**:** 2**,** name**:** "Bob"**,** role**:** "Backend Developer" }**,**

  { id**:** 3**,** name**:** "Charlie"**,** role**:** "UI/UX Designer" }**,**

  { id**:** 4**,** name**:** "David"**,** role**:** "Project Manager" }**,**

]**;**

*export* **default** function ChatPage() {

  const [messages**,** setMessages] **=** useState({})**;**

  const [message**,** setMessage] **=** useState("")**;**

  const [file**,** setFile] **=** useState(null)**;**

  const [activeContact**,** setActiveContact] **=** useState(contacts[0])**;**

  const [status**,** setStatus] **=** useState({})**;** *// ✅ Stores message status (sent/failed)*

*// ✅ Register user on mount*

  useEffect(() **=>** {

    socket**.**emit("register"**,** loggedInUser**.***id*)**;**

    console**.**log("✅ Registered with server as:"**,** loggedInUser**.***id*)**;**

*// ✅ Listen for incoming messages*

    socket**.**on("message"**,** (**msg**) **=>** {

      console**.**log("📩 New message received:"**,** msg)**;**

      setMessages((**prev**) **=>** ({

**...**prev**,**

        [msg**.***senderId* **===** loggedInUser**.***id* **?** msg**.***receiverId* **:** msg**.***senderId*]**:** [

**...**(prev[msg**.***senderId* **===** loggedInUser**.***id* **?** msg**.***receiverId* **:** msg**.***senderId*] **||** [])**,**

          msg**,**

        ]**,**

      }))**;**

*// ✅ Mark message as sent*

      setStatus((**prev**) **=>** ({

**...**prev**,**

        [msg**.***tempId*]**:** "sent"**,**

      }))**;**

    })**;**

**return** () **=>** {

      socket**.**off("message")**;**

    }**;**

  }**,** [])**;**

*// ✅ Send message function*

  const sendMessage **=** *async* () **=>** {

**if** (**!**message**.**trim() **&&** **!**file) **return;**

    let fileUrl **=** null**;**

**if** (file) {

      const formData **=** **new** FormData()**;**

      formData**.**append("file"**,** file)**;**

**try** {

        const response **=** **await** axios**.**post("http://localhost:5000/upload"**,** formData**,** {

          headers**:** { "Content-Type"**:** "multipart/form-data" }**,**

        })**;**

        fileUrl **=** response**.***data***.***fileUrl***;**

      } **catch** (error) {

        console**.**error("Error uploading file:"**,** error)**;**

        setStatus((**prev**) **=>** ({ **...**prev**,** tempId**:** "failed" }))**;**

**return;**

      }

    }

    const tempId **=** Date**.**now()**;** *// ✅ Temporary ID to track message status*

    const msgData **=** {

      tempId**,** *// ✅ Track message*

      senderId**:** loggedInUser**.***id***,**

      receiverId**:** activeContact**.***id***,**

      text**:** message**.**trim() **||** null**,**

      fileUrl**,**

    }**;**

**try** {

      socket**.**emit("message"**,** msgData)**;**

      console**.**log("📤 Message sent:"**,** msgData)**;**

      setMessages((**prev**) **=>** ({

**...**prev**,**

        [activeContact**.***id*]**:** [**...**(prev[activeContact**.***id*] **||** [])**,** msgData]**,**

      }))**;**

*// ✅ Show "Message Sent ✅"*

      setStatus((**prev**) **=>** ({ **...**prev**,** [tempId]**:** "sent" }))**;**

    } **catch** (error) {

      console**.**error("❌ Message failed:"**,** error)**;**

*// ✅ Show "Message Not Sent ❌"*

      setStatus((**prev**) **=>** ({ **...**prev**,** [tempId]**:** "failed" }))**;**

    }

    setMessage("")**;**

    setFile(null)**;**

  }**;**

**return** (

    <div *className***=**"main-container">

      <div *className***=**"header">

        <h3>Jobify</h3>

      </div>

      <div *className***=**"chat-container d-flex">

        <div *className***=**"contacts-list">

          <h5>Contacts</h5>

          <ul *className***=**"list-unstyled">

            {contacts**.**map((**contact**) **=>** (

              <li *key***=**{contact**.***id*}>

                <button

*className***=**{`btn w-100 text-start mb-2 py-2 px-3 ${

                    activeContact**.***id* **===** contact**.***id* **?** "btn-primary" **:** "btn-outline-light"

                  }`}

*onClick***=**{() **=>** setActiveContact(contact)}

                >

                  <strong>{contact**.***name*}</strong>

                  <div *className***=**"text-muted" *style***=**{{ fontSize**:** "0.85rem" }}>

                    {contact**.***role*}

                  </div>

                </button>

              </li>

            ))}

          </ul>

        </div>

        <div *className***=**"chat-box">

          <div *className***=**"chat-header d-flex justify-content-between align-items-center p-2">

            <div>

              <h5 *className***=**"mb-0">{activeContact**.***name*}</h5>

              <small *className***=**"text-muted">{activeContact**.***role*}</small>

            </div>

            <MoreVertical *className***=**"menu-icon" />

          </div>

          <div *className***=**"chat-body">

            {(messages[activeContact**.***id*] **||** [])**.**map((**msg,** **index**) **=>** (

              <div *key***=**{index} *className***=**{`message ${msg**.***senderId* **===** loggedInUser**.***id* **?** "sent" **:** "received"}`}>

                {msg**.***text* **&&** <p>{msg**.***text*}</p>}

                {msg**.***fileUrl* **&&** (

                  <a *href***=**{`http://localhost:5000${msg**.***fileUrl*}`} *target***=**"\_blank" *rel***=**"noopener noreferrer">

                    📄 View File

                  </a>

                )}

                {*/\* ✅ Message Sent / Failed Status \*/*}

                {status[msg**.***tempId*] **===** "sent" **&&** <small *className***=**"message-status">✅ Sent</small>}

                {status[msg**.***tempId*] **===** "failed" **&&** <small *className***=**"message-status error">❌ Not Sent</small>}

              </div>

            ))}

          </div>

          <div *className***=**"chat-footer">

            <label *className***=**"attach-btn">

              <Paperclip *size***=**{20} />

              <input *type***=**"file" *onChange***=**{(**e**) **=>** setFile(e**.***target***.***files*[0])} *accept***=**".pdf,.docx" />

            </label>

            <input *type***=**"text" *value***=**{message} *onChange***=**{(**e**) **=>** setMessage(e**.***target***.***value*)} *placeholder***=**"Type a message..." />

            <button *onClick***=**{sendMessage} *className***=**"send-btn">Send</button>

          </div>

        </div>

      </div>

    </div>

  )**;**

}

App.css

*/\* ✅ Global Styling \*/*

body {

  background-color**:** black**;**

  color**:** white**;**

  margin**:** 0**;**

  font-family**:** Arial**,** sans-serif**;**

}

*/\* ✅ Header Styling \*/*

*.header* {

  background-color**:** black**;**

  color**:** white**;**

  padding**:** 15px 20px**;**

  font-size**:** 20px**;**

  font-weight**:** bold**;**

  text-align**:** center**;**

  border-bottom**:** 1px solid **#444;**

}

*/\* ✅ Chat Layout \*/*

*.chat-container* {

  width**:** 100%**;**

  height**:** calc(100vh - 50px)**;**

  display**:** flex**;**

}

*/\* ✅ Contacts Sidebar \*/*

*.contacts-list* {

  width**:** 25%**;**

  background**:** **#222;**

  padding**:** 15px**;**

  border-right**:** 1px solid **#444;**

  display**:** flex**;**

  flex-direction**:** column**;**

}

*.contacts-list* h5 {

  color**:** white**;**

  margin-bottom**:** 15px**;**

}

*.contacts-list* ul {

  list-style**:** none**;**

  padding**:** 0**;**

  flex-grow**:** 1**;**

  overflow-y**:** auto**;**

}

*.contacts-list* li {

  padding**:** 8px**;**

  cursor**:** pointer**;**

  border-radius**:** 8px**;**

  transition**:** 0.3s**;**

  color**:** white**;**

}

*.contacts-list* li*:hover***,**

*.contacts-list* li*.active* {

  background**:** **#007bff;**

}

*.contacts-list* button {

  border**:** none**;**

  width**:** 100%**;**

  text-align**:** left**;**

  padding**:** 12px**;**

  border-radius**:** 8px**;**

  transition**:** 0.3s**;**

  background**:** none**;**

  color**:** white**;**

}

*.contacts-list* button*:hover***,**

*.contacts-list* *.active* {

  background**:** rgba(255**,** 255**,** 255**,** 0.1)**;**

}

*.contacts-list* button div {

  font-size**:** 0.85rem**;**

  color**:** lightgray**;**

}

*.contacts-list* hr {

  border**:** none**;**

  height**:** 1px**;**

  background**:** **#444;**

  margin**:** 8px 0**;**

}

*/\* ✅ Chat Box \*/*

*.chat-box* {

  width**:** 75%**;**

  display**:** flex**;**

  flex-direction**:** column**;**

}

*/\* ✅ Chat Header \*/*

*.chat-header* {

  background**:** **#007bff;**

  padding**:** 15px**;**

  display**:** flex**;**

  justify-content**:** space-between**;**

  align-items**:** center**;**

  color**:** white**;**

}

*.chat-header* h5 {

  margin**:** 0**;**

  font-weight**:** bold**;**

}

*.chat-header* small {

  display**:** block**;**

  font-size**:** 14px**;**

  color**:** **#ddd;**

}

*.menu-icon* {

  cursor**:** pointer**;**

}

*/\* ✅ Chat Messages \*/*

*.chat-body* {

  flex-grow**:** 1**;**

  background**:** **#333;**

  padding**:** 15px**;**

  overflow-y**:** auto**;**

  display**:** flex**;**

  flex-direction**:** column**;**

}

*/\* ✅ Message Styling \*/*

*.message* {

  max-width**:** 60%**;**

  padding**:** 12px**;**

  margin-bottom**:** 10px**;**

  border-radius**:** 10px**;**

  font-size**:** 14px**;**

  display**:** flex**;**

  flex-direction**:** column**;**

}

*.sent* {

  background**:** **#007bff;**

  color**:** white**;**

  align-self**:** flex-end**;**

  text-align**:** right**;**

}

*.received* {

  background**:** **#444;**

  color**:** white**;**

  align-self**:** flex-start**;**

  text-align**:** left**;**

}

*/\* ✅ File Link Styling \*/*

*.file-link* {

  display**:** flex**;**

  align-items**:** center**;**

  color**:** **#ddd;**

  font-size**:** 14px**;**

  text-decoration**:** none**;**

  margin-top**:** 5px**;**

}

*.file-link:hover* {

  color**:** white**;**

}

*/\* ✅ Chat Footer \*/*

*.chat-footer* {

  display**:** flex**;**

  align-items**:** center**;**

  padding**:** 10px**;**

  background**:** **#111;**

}

*.chat-footer* input {

  flex-grow**:** 1**;**

  padding**:** 10px**;**

  border**:** none**;**

  border-radius**:** 5px**;**

  background**:** **#ddd;**

  color**:** black**;**

}

*.send-btn* {

  background**:** **#007bff;**

  color**:** white**;**

  border**:** none**;**

  padding**:** 12px 20px**;**

  margin-left**:** 10px**;**

  cursor**:** pointer**;**

  border-radius**:** 5px**;**

  transition**:** 0.3s ease-in-out**;**

}

*.send-btn:hover* {

  background**:** **#0056b3;**

}

*.send-btn:active* {

  transform**:** scale(0.98)**;**

}

*/\* ✅ Attachment Button \*/*

*.attach-btn* {

  background**:** transparent**;**

  border**:** none**;**

  color**:** white**;**

  margin-right**:** 10px**;**

  cursor**:** pointer**;**

  position**:** relative**;**

}

*/\* ✅ Hidden File Input \*/*

*.attach-btn* input {

  position**:** absolute**;**

  left**:** 0**;**

  top**:** 0**;**

  opacity**:** 0**;**

  width**:** 100%**;**

  height**:** 100%**;**

  cursor**:** pointer**;**

}

*/\* ✅ File Preview \*/*

*.file-preview* {

  color**:** lightgray**;**

  margin-left**:** 10px**;**

  font-size**:** 14px**;**

}

*/\* ✅ Message Status Styling \*/*

*.message-status* {

  font-size**:** 12px**;**

  margin-top**:** 5px**;**

  color**:** lightgreen**;**

}

*.message-status.error* {

  color**:** red**;**

}

Message.jsx

*import* React *from* "react"**;**

*import* { Star } *from* "lucide-react"**;**

*export* **default** function Message({ **msg,** **isStarred,** **toggleStar,** **markAsRead** }) {

**return** (

    <div

*className***=**{`message ${msg**.***sender* **===** "me" **?** "text-end" **:** "text-start"} mb-2`}

*onClick***=**{markAsRead}

    >

      <div

*className***=**{`p-2 rounded ${msg**.***sender* **===** "me" **?** "bg-primary text-white" **:** "bg-light"}`}

*style***=**{{ display**:** "inline-block"**,** maxWidth**:** "75%" }}

      >

        {msg**.***text*}

      </div>

      <button

*className***=**"btn btn-sm ms-2"

*onClick***=**{(**e**) **=>** {

          e**.**stopPropagation()**;**

          toggleStar()**;**

        }}

      >

        <Star *size***=**{16} *fill***=**{isStarred **?** "yellow" **:** "none"} />

      </button>

    </div>

  )**;**

}

Server.js

express **=** require("express")**;**

const http **=** require("http")**;**

const { Server } **=** require("socket.io")**;**

const cors **=** require("cors")**;**

const path **=** require("path")**;**

const multer **=** require("multer")**;**

const fs **=** require("fs")**;**

const app **=** express()**;**

const server **=** http**.**createServer(app)**;**

const io **=** **new** Server(server**,** {

  cors**:** {

    origin**:** "http://localhost:3000"**,**

    methods**:** ["GET"**,** "POST"]**,**

  }**,**

})**;**

*// Middleware*

app**.**use(cors())**;**

app**.**use(express**.**json())**;**

app**.**use(express**.**urlencoded({ extended**:** true }))**;**

*// ✅ Ensure uploads directory exists*

const uploadDir **=** path**.**join(\_\_dirname**,** "uploads")**;**

**if** (**!**fs**.**existsSync(uploadDir)) {

  fs**.**mkdirSync(uploadDir)**;**

}

*// ✅ Serve uploaded files*

app**.**use("/uploads"**,** express**.**static(uploadDir))**;**

*// ✅ File Upload Setup*

const storage **=** multer**.**diskStorage({

  destination**:** (**req,** **file,** **cb**) **=>** {

    cb(null**,** uploadDir)**;**

  }**,**

  filename**:** (**req,** **file,** **cb**) **=>** {

    cb(null**,** Date**.**now() **+** "-" **+** file**.***originalname*)**;**

  }**,**

})**;**

const upload **=** multer({ storage })**;**

*// ✅ API Route for File Upload*

app**.**post("/upload"**,** upload**.**single("file")**,** (**req,** **res**) **=>** {

**if** (**!**req**.***file*) {

**return** res**.**status(400)**.**json({ message**:** "No file uploaded" })**;**

  }

  res**.**json({ fileUrl**:** `/uploads/${req**.***file***.***filename*}` })**;**

})**;**

*// Store connected users (userId -> socketId)*

const users **=** {}**;**

io**.**on("connection"**,** (**socket**) **=>** {

  console**.**log(`🔌 User connected: ${socket**.***id*}`)**;**

*// ✅ Register users*

  socket**.**on("register"**,** (**userId**) **=>** {

    users[userId] **=** socket**.***id***;**

    console**.**log(`✅ User ${userId} registered as ${socket**.***id*}`)**;**

  })**;**

*// ✅ Handle incoming messages*

  socket**.**on("message"**,** (**msg**) **=>** {

    console**.**log("📩 Message received:"**,** msg)**;**

    const { senderId**,** receiverId**,** text**,** fileUrl } **=** msg**;**

    const messageData **=** { senderId**,** receiverId**,** text**,** fileUrl }**;**

**if** (users[receiverId]) {

      io**.**to(users[receiverId])**.**emit("message"**,** messageData)**;**

    }

    io**.**to(socket**.***id*)**.**emit("message"**,** messageData)**;**

  })**;**

*// 🔥 Handle user disconnect*

  socket**.**on("disconnect"**,** () **=>** {

    console**.**log(`❌ User disconnected: ${socket**.***id*}`)**;**

**for** (const userId **in** users) {

**if** (users[userId] **===** socket**.***id*) {

**delete** users[userId]**;**

        console**.**log(`❌ Removed user ${userId}`)**;**

      }

    }

  })**;**

})**;**

*// Start server*

server**.**listen(5000**,** () **=>** {

  console**.**log("🚀 Server running on http://localhost:5000")**;**

})**;**