🚀 **Frontend Development Guide for SportsChat+**

This guide ensures all developers follow a structured approach to setting up and working with the frontend for SportsChat+.

**1️. Project Structure**

📂 sportschat-frontend/

├── public/ # Static files (favicon, index.html, manifest.json)

├── src/ # Source code

│ ├── components/ # Reusable UI components

│ │ ├── Button.js

│ │ ├── Navbar.js

│ │ ├── ChatMessage.js

│ │ ├── InputField.js

│ ├── pages/ # Page Components

│ │ ├── HomePage.js

│ │ ├── LoginPage.js

│ │ ├── ChatPage.js

│ ├── services/ # API calls to backend

│ │ ├── api.js

│ ├── App.js # Main component

│ ├── index.js # React entry point

│ ├── App.css # Global styles

├── package.json

└── README.md

**2️. Setting Up React Router**

📌 **Install React Router:**

npm install react-router-dom

📌 **Update App.js to handle navigation between pages:**

import { BrowserRouter as Router, Routes, Route } from "react-router-dom";

import HomePage from "./pages/HomePage";

import LoginPage from "./pages/LoginPage";

import ChatPage from "./pages/ChatPage";

function App() {

return (

<Router>

<Routes>

<Route path="/" element={<HomePage />} />

<Route path="/login" element={<LoginPage />} />

<Route path="/chat" element={<ChatPage />} />

</Routes>

</Router>

);

}

export default App;

**3️. Reusable UI Components**

📌 **Create a components/ folder for reusable UI elements.**

**Example Components:**

📂 sportschat-frontend/src/components/Navbar.js

import { Link } from "react-router-dom";

function Navbar() {

return (

<nav>

<Link to="/">Home</Link>

<Link to="/login">Login</Link>

<Link to="/chat">Chat</Link>

</nav>

);

}

export default Navbar;

📂 sportschat-frontend/src/components/Button.js

function Button({ text, onClick }) {

return <button onClick={onClick}>{text}</button>;

}

export default Button;

**4️. API Calls for Backend Integration**

📌 **API calls are centralized in services/api.js**

📂 sportschat-frontend/src/services/api.js

import axios from "axios";

const API = axios.create({

baseURL: "http://localhost:5000",

withCredentials: true,

});

export const login = async (username, password) => {

return API.post("/login", { username, password });

};

export const getCurrentUser = async () => {

return API.get("/me");

};

export const logout = async () => {

return API.post("/logout");

};

export default API;

**5️. Styling Options**

You can style your application using: ✅ CSS Modules (.module.css) ✅ Styled Components (npm install styled-components) ✅ Bootstrap or Tailwind CSS

📌 **Install Bootstrap:**

npm install bootstrap

📌 **Import Bootstrap in index.js:**

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

**6️. Final Steps**

✔ Ensure each module (Login, Chat, Dashboard, etc.) follows this structure.  
✔ Ensure React Router is used for navigation.  
✔ Use services (api.js) to interact with the backend.  
✔ Organize the UI with reusable components.  
✔ Implement styling with Bootstrap, Tailwind, or CSS Modules.

**🚀 Next Steps**

Each developer should:

1. **Set up their environment** using the steps above.
2. **Follow the component and page structure.**
3. **Use the API service file** to interact with the backend.
4. **Ensure clean UI** using reusable components.
5. **Submit a pull request** after finishing their assigned module.