



Project Report: Bolt (Chat/File Sharing Application)

1. Project Introduction

Bolt is a modern, real-time chat and file sharing application built with React (using TypeScript), Vite, and Tailwind CSS for the frontend, and Node.js/Express for the backend. It provides features like user authentication, real-time messaging, file uploads, user search, and chat room management.

2. 🛠 Tools & Technologies Used

- **Frontend:** React + TypeScript
 - **Build Tool:** Vite
 - **Styling:** Tailwind CSS
 - **State Management:** React Context API
 - **Backend:** Node.js (likely with Express.js)
 - **File Handling:** File Upload system
 - **Code Quality:** ESLint, PostCSS
-

3. Folder & File Structure Explained

Root Files: - `index.html` - Main HTML entry point - `package.json` - Lists project dependencies and scripts - `vite.config.ts` - Configuration for Vite build tool - `tailwind.config.js` - Tailwind CSS config file - `tsconfig.json` - TypeScript configuration - `.gitignore` - Files ignored by Git - `postcss.config.js` - Tailwind + PostCSS integration - `eslint.config.js` - JavaScript/TypeScript linting rules

/src Folder: (Main React app code) - `main.tsx` - Entry point for rendering the React App - `App.tsx` - Main application component with routing and layout - `index.css` - Global styles (integrates Tailwind CSS) - `vite-env.d.ts` - TypeScript environment definitions

/components Folder: (UI and feature components) - `AuthForm.tsx` - Handles login/register forms - `FileUpload.tsx` - UI for uploading files - `MessageList.tsx` - Displays list of messages in chat - `UserProfile.tsx` - Shows user profile information - `UserSearch.tsx` - Allows searching for other users - `LoadingSpinner.tsx` - Reusable loading spinner - `RoomBrowser.tsx` - Displays and manages available chat rooms

/contexts Folder: (likely contains React Contexts for state like auth, chat, etc.)

/server Folder: (backend code - needs further exploration) - Presumably contains API routes, database connection, auth logic, etc.

/.bolt Folder: - Possibly used for deployment configuration or special tooling

4. Key Code Explanations

App.tsx:

Manages the routing and layout of the application. Likely uses React Router and context providers.

AuthForm.tsx:

Contains logic for both Login and Signup forms. Uses local state for input and triggers API requests.

FileUpload.tsx:

Handles file input and triggers file upload to server. Likely uses `FormData` and `fetch`/`axios`.

MessageList.tsx:

Maps and displays messages from a state (could be via WebSocket or polling).

RoomBrowser.tsx:

Displays a list of chat rooms and allows users to switch or join rooms.

5. User Flow / Features

1. **User visits site**
 2. **Logs in or registers**
 3. **Joins a chat room**
 4. **Searches other users**
 5. **Sends and receives messages**
 6. **Uploads and shares files**
-

6. Conclusion & Future Scope

Bolt is a solid, modular chat app framework built with modern web technologies. It supports extensibility and scalability.

Future Enhancements Could Include: - Real-time WebSocket integration - Push notifications - Profile avatars - Dark mode - Admin/moderation tools - Database integration for persistent chat history

Prepared by: ChatGPT – Project Report Generator **Date:** June 11, 2025