

[Improvements]

Improve the appearance of the sidebar and chat widget section to match the original Figma design, but make the font size and icons more proportional.

[Additional instructions]

Frontend Design Guide for AI Chatbot MVP

1. Purpose of This Document

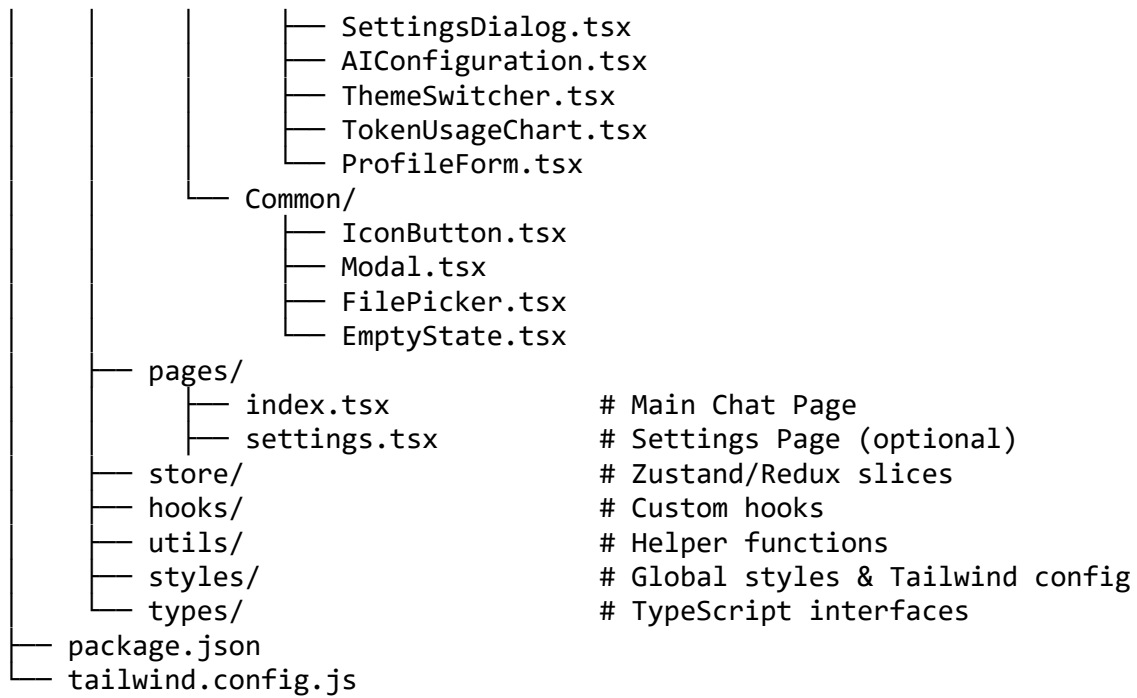
These additional instructions include setting up the working directory, explaining feature designs, wireframe diagrams, and instructions for mockups. The backend section has been intentionally left out for backend developers.

2. Technologies

- **Framework:** React (Next.js)
 - **State Management:** Zustand / Redux Toolkit
 - **Styling:** TailwindCSS + shadcn/ui
 - **Charts:** Recharts (for token usage)
 - **TypeScript:** Strong typing for maintainability
-

3. Directory Structure (Frontend)

```
frontend/
├── public/                # Static assets (logo, icons, images)
├── src/
│   ├── components/       # Reusable components grouped by domain
│   │   ├── Sidebar/
│   │   │   ├── Sidebar.tsx
│   │   │   ├── ConversationList.tsx
│   │   │   └── ConversationActionsMenu.tsx
│   │   ├── Chat/
│   │   │   ├── ChatTopBar.tsx
│   │   │   ├── ChatArea.tsx
│   │   │   ├── UserBubble.tsx
│   │   │   ├── AIBubble.tsx
│   │   │   └── PromptInput.tsx
│   └── Settings/
```



4. Feature Design Details

4.1 Sidebar

- **Hamburger Icon:** toggles show/hide sidebar.
- **Search Icon:** opens search bar for conversation history.
- **New Conversation Icon:** adds a new conversation entry.
- **Conversation List Widget:**
 - Displays all past conversations.
 - Three-dot menu with actions: **Pin, Rename, Delete.**
- **Settings Icon:** opens a floating popup with 4 tabs:
 - **AI Configuration:** set max tokens, choose AI personality (default, robot, cynic, expert).
 - **Theme Switcher:** toggle light/dark themes (inverse color scheme).
 - **Token Usage:** daily usage chart (Recharts) + date range filter.
 - **Profile:** manage email, avatar, and role/occupation.

4.2 Chat Widget

- **Top Bar:** logo (left), profile (right).
- **Conversation Section:**
 - **User messages:** light gray background in light mode, includes copy & edit.
 - **AI messages:** same background as chat area, includes **download (.docx)**, copy, regenerate.
- **Input Prompt:**

- Text field with placeholder: “Drop Your Question”.
- **Attach (+)** icon: supports image/document uploads.
- **Send** icon (paper plane): enabled after input or file attachment.

4.3 Responsiveness & Accessibility

- Mobile-first design.
- Sidebar collapsible on smaller screens.
- Accessible contrast ratios.
- Keyboard navigable (focus states, ARIA labels).

5. State & Data Shape


```
export interface Conversation {
  id: string;
  title: string;
  pinned: boolean;
  createdAt: string;
  updatedAt: string;
}

export interface Message {
  id: string;
  conversationId: string;
  role: 'user' | 'assistant';
  content: string;
  createdAt: string;
  attachments?: Array<{ id: string; name: string; type: string; url: string }>;
}

export interface TokenUsageData {
  day: string;
  inputTokens: number;
  outputTokens: number;
}
```

6. Wireframe Diagram (Frontend Components)

+-----+ <div> <div>Top Bar: [Logo]</div> <div>[Profile]</div> </div> +-----+	
<div> <div> <div> <div>[≡] Hamburger</div> <div>[🔍] Search</div> <div>[+] New Conv.</div> <div>Conversations List</div> <div>- Conv 1 [:]</div> </div> <div> <div>Conversation Area</div> <div> <div>User Bubble (copy, edit)</div> <div>AI Bubble (copy, download, regenerate)</div> </div> <div>Input Section:</div> </div> </div> </div>	

- Conv 2 [:]	[Drop Your Question...] [+] 
[⚙️] Settings	

Settings Popup (floating modal):

Tabs: AI Config Theme Token Usage Profile Content changes per tab

7. Mockup Example Instruction

To simulate backend responses during frontend development:

1. Create mock API calls in `src/pages/api/` (Next.js API routes) or a `mocks/` folder.
2. Example mock in `api/chat.ts`:

```
export async function sendMessageMock(question: string) {
  return {
    messageId: 'mock-1',
    answer: 'This is a mock AI answer.',
    sql: 'SELECT * FROM sales LIMIT 5;',
    tokens: { prompt: 123, completion: 45 },
  };
}
```

3. **Instruction for Backend Developer:** Once backend endpoints are ready, remove all `sendMessageMock` and replace with real API calls.

8. Logical Flow (Frontend Perspective)

User → Sidebar → Select/New Conversation

- Chat Page → Type Message / Attach File → Send
- Loading State → Display AI Response in Chat Area
- User can Copy / Edit / Regenerate messages
- Settings Popup → adjust AI Config, Theme, Token Usage, Profile

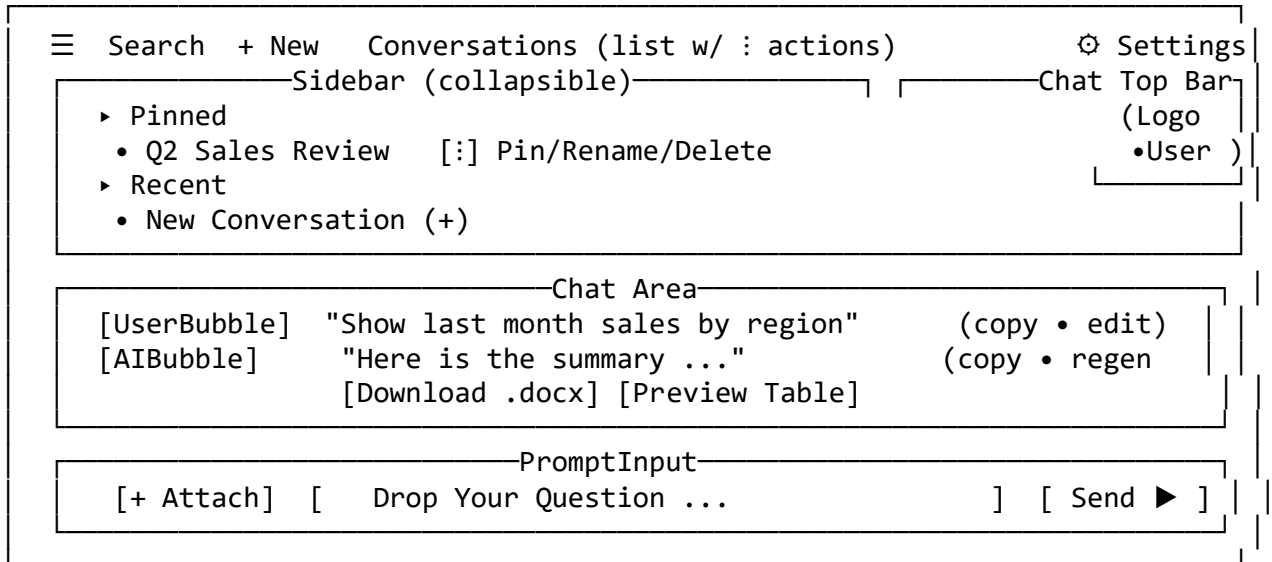
9. Backend Section

⚠️ Leave this section empty. The backend developer will implement all backend logic, APIs, and services.

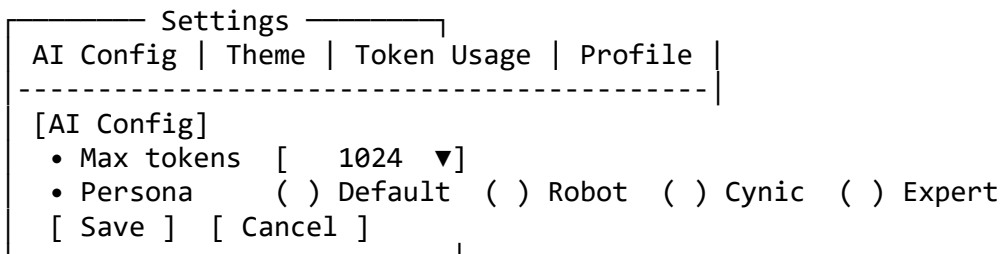
9. UI Component Wireframes (Textual)

These wireframes are **structural guides**—exact visuals will follow Tailwind/shadcn tokens.

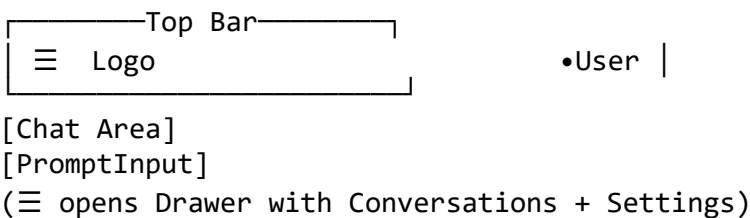
9.1 Main Layout (Desktop)



9.2 Settings Dialog (Tabs)



9.3 Mobile Layout (Collapsed Sidebar)



10. Component Responsibilities & Props (Detailed)

10.1 Sidebar

- Sidebar: controls open/close state; hosts search, new conversation, list.
- ConversationList props:

- items: Conversation[]
 - activeId?: string
 - onSelect(id), onPin(id), onRename(id, title), onDelete(id)
- ConversationActionsMenu props: onPin, onRename, onDelete

10.2 Chat

- ChatTopBar props: { logoUrl, userAvatarUrl }
- ChatArea props: { messages: Message[], onRegenerate(messageId) }
- UserBubble props: { message, onCopy(), onEdit(messageId, text) }
- AIBubble props: { message, onCopy(), onRegenerate(), onDownloadDocx(messageId) }
- PromptInput props: { onSend(text, files?), disabled?, isLoading? }
- FilePicker props: { accept: string[], maxSizeMB: number, onFiles(files) }

10.3 Settings

- SettingsDialog state: active tab; controlled via parent (sidebar settings icon)
 - AIConfiguration props: { value: {maxTokens:number, persona:string}, onChange(v) }
 - ThemeSwitcher props: { theme: 'light'|'dark', onToggle() }
 - TokenUsageChart props: { data: TokenUsageData[], from?:string, to?:string, onRangeChange(f,t) }
 - ProfileForm props: { value: {email, avatarUrl, role}, onChange(v) }
-