

1. The Problem We Want You to Solve

Tech professionals and enthusiasts struggle to stay updated with the rapidly evolving technology landscape. They need a centralized platform that not only aggregates trending tech news but also enables intelligent discussion and analysis of these stories.

2. Your Challenge

Build a modern, interactive tech news aggregation platform that pulls trending stories and enables AI-powered discussions around them.

3. Core Features Required

News Aggregation & Display:

- Pull trending technology stories from sources like TechCrunch, The Verge, Hacker News, etc.
- Clean, responsive news feed with article previews
- Category-based filtering (AI/ML, Startups, Cybersecurity, Mobile, Web3, etc.)
- Search functionality across articles
- Bookmark/save articles for later

AI-Powered News Chat:

- Interactive chat interface where users can discuss any news article
- AI assistant that can:
 - Summarize articles when asked
 - Answer questions about specific news stories
 - Provide context and background on tech topics
 - Compare related stories or developments
- Chat should intelligently pull in relevant article content when discussing specific stories

Smart Features:

- Trending topics dashboard
- Related articles suggestions
- Article sentiment analysis (positive/negative/neutral)
- Time-based filtering (today, this week, this month)

4. Design Requirements

- **Primary Tool:** Use Lovable for rapid UI/UX design and prototyping
- **Design Focus:** Clean, modern news platform aesthetic (think Feedly meets Discord)
- **UX Priorities:**

- Fast article browsing and filtering
 - Seamless transition between reading and chatting
 - Mobile-responsive design
 - Dark/light mode toggle
- **Chat UX:** Make the AI chat feel natural and contextually aware of the articles being discussed

5. Technical Requirements

- **Development:** Use Cursor for AI-assisted development (required)
- **AI Integration:** OpenAI or Anthropic API for chat functionality
- **News Sources:** Web scraping or RSS feeds from tech news sites
- **Framework:** Any modern web framework (React, Vue, Next.js, etc.)
- **Real-time:** Consider real-time updates for news feed
- **Deployment:** Live deployment (Vercel, Netlify, or Lovable hosting)

6. Technical Challenges to Solve

- **News Data:** Implement reliable news fetching (RSS, APIs, or ethical scraping)
- **Content Processing:** Extract and clean article content for AI processing
- **Chat Context:** Make AI aware of which articles are being discussed
- **Performance:** Handle large amounts of news data efficiently
- **Rate Limiting:** Manage API calls for both news fetching and AI chat

7. Deliverables

a. **Live web app URL** for testing b. **A follow up interview** covering:

- Your approach to news data aggregation
- How you made the AI chat contextually aware of articles
- Your UX decisions for news browsing vs. chat interaction
- How you used Cursor to accelerate development
- Future features you'd implement with more time
- Technical challenges and solutions
- Performance optimizations implemented

8. Evaluation Criteria

- **User Experience:** Intuitive news browsing and natural chat interaction
- **Technical Implementation:** Clean code, proper error handling, performance
- **AI Integration:** How well the chat understands and discusses articles
- **Design Quality:** Visual appeal and responsiveness
- **Feature Completeness:** Core functionality working smoothly

9. Timeline: 7-10 days

10. Bonus Points For:

- Implementing real-time news updates
- Advanced filtering and search capabilities
- Analytics dashboard for trending topics
- Smart UX/UI Decisions

Example User Flow:

1. User opens app → sees trending tech stories organized by category
2. User clicks on an AI startup funding article
3. User opens chat and asks: "What does this mean for the competitive landscape?"
4. AI responds with context about the article, related funding trends, and competitor analysis
5. User can seamlessly switch between reading more articles and continuing the discussion