

Project Report: AI-Based Crypto News Analysis Mobile App

Project Overview

This mobile app will help users understand how recent crypto news affects major coins like XRP, BTC, and others. For example, if Ripple applies for a banking license or a politician announces crypto policies, the app will show an AI-generated prediction about how that news might influence the market. The app will help users stay informed using smart, real-time AI analysis — without relying on outdated knowledge.

How the App Works

1. The app collects fresh crypto-related news from the internet
 2. The backend checks if the article is new and important
 3. A powerful AI model (DeepSeek Reasoner) reads and analyzes the news
 4. The result is stored in a database
 5. When users open the app, they instantly see news with predictions
 6. Users don't call the AI individually — all users see shared results
-

Technologies Used

Feature	Technology Used
Mobile App	Flutter
Backend Server	FastAPI
AI Analysis	DeepSeek Reasoner API
News Collection	Web Scraping or Paid API
Database	Supabase
Hosting	Render.com (Free Tier)

Feature	Technology Used
Context Retrieval	Vector DB (RAG with pgvector)

Why RAG Is Used in This Version

The AI model is only trained up to **June 2024**, but we are already in **July 2025**. So we can't depend on the AI's memory for current events.

That's why we're using **RAG (Retrieval-Augmented Generation)**. This means:

- We store fresh news articles as embeddings in a vector database
- When a new article comes, we find similar past articles
- We combine them into one prompt and send it to the AI
- The AI then gives a more accurate prediction using real-world, current data

This method keeps the analysis relevant and trustworthy.

DeepSeek API Cost Estimate (Real-Time)

Each news + analysis is expected to use around **1000 tokens**
DeepSeek pricing (standard hours):

- Input: **\$0.55 per 1M tokens**
- Output: **\$2.19 per 1M tokens**
- Estimated cost per article: ~\$0.00207

Monthly AI Cost Estimate

News Articles per Day Monthly Cost (30 Days)

10 ~\$0.62

15 ~\$0.93

This makes the AI cost **less than \$1 - \$2/month** for the initial version

Supabase (Database, Auth, and Storage)

Supabase will be used to store:

- User accounts
- News articles
- AI analysis
- Vector embeddings for RAG

Free Plan Includes:

- 1 GB total storage (includes file and database storage)
- 100 active users
- 500,000 API requests/month
- 2 million rows

Even if each article with embeddings takes 10 KB:

- 1000 articles = only 10 MB
- You'll stay **well below 1 GB**

Storage Cost (if upgraded to Pro Plan):

- First 100 GB included
- Over-usage: **\$0.021 per GB per month**
or **~\$0.00002919 per GB-Hour**

So even if you upgrade later, cost is extremely low — just **2 cents per GB/month** for over-usage.

Hosting (Render.com Free Tier)

We'll use Render to host the backend (FastAPI).

Free plan includes:

- 512 MB RAM
- 0.5 CPU
- ~100–200 user requests/day
- Server sleeps after 15 minutes of no activity

Enough for development and testing.

News Collection: Scraping or News API

We'll first try web scraping from trusted crypto sites. If any block scraping, we'll switch to a paid API.

Provider	Free Plan	Paid Plan
NewsData.io	200 articles/day	\$39/month
GNews API	Limited free access	\$49/month

Total Estimated Monthly Cost (Initial Version)

Item	Monthly Cost Estimate	Notes
DeepSeek API	\$0.62 – \$0.93	10–15 news items/day
Supabase	\$0	Under 1 GB (free plan sufficient)
Supabase (extra GB)	\$0.021 per GB/month	Only if Pro plan is needed later
Render Hosting	\$0	Free Tier
News API (optional)	\$0 – \$49	Only if scraping fails

Final monthly cost estimate:

- **Likely: ~\$1- \$2/month** (if scraping works)
- **Max: ~\$50/month** (if news API is needed)

Development Plan

Phase Task Description

- 1 Build the UI using Flutter
- 2 Set up FastAPI backend
- 3 Add scraping or connect to news API
- 4 Use vector DB to enable RAG

Phase Task Description

- 5 Connect DeepSeek and store predictions
 - 6 Show news + predictions in app UI
 - 7 Test and prepare for launch
-