

Anything Explainer - Hackathon Documentation

Team: HTG ThinkTank | Hackathon: lablab.ai x Trae AI IDE

Introduction

Anything Explainer is a full-stack web app built in Node.js + Bootstrap that allows users to instantly get AI-powered explanations for any concept, molecule, formula, or technical topic. Users select one of three levels:

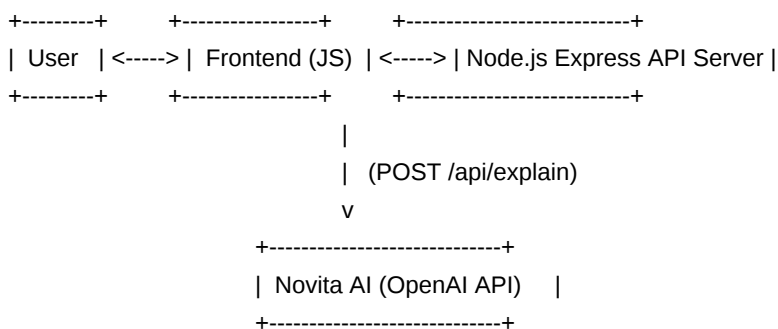
- Simple (for beginners/children)
- Step-by-step (for students)
- Technical (for advanced/professional users)

Every answer can be saved to a local knowledge gallery. The backend calls Novita AI's OpenAI-compatible endpoint and adapts its response to the chosen level.

Architecture Overview

User interacts with the Bootstrap frontend, selects the explanation level, and submits a prompt. The frontend calls a Node.js/Express backend API (/api/explain), which selects a custom prompt and sends a request to Novita AI. The backend responds with the AI-generated explanation. The frontend shows the answer and can save it in a local knowledge gallery.

[Architecture Diagram]



Features

- Multi-level Explanations: Simple, Step-by-step, Technical
- Modern Bootstrap UI: Clean, responsive, mobile-ready
- Knowledge Gallery: Save and view all explanations
- Novita AI Integration: Secure, OpenAI-compatible backend call

Component Breakdown

Anything Explainer - Hackathon Documentation

Team: HTG ThinkTank | Hackathon: lablab.ai x Trae AI IDE

Frontend: index.html, main.js, style.css

Backend: server.js (serves static frontend, handles /api/explain, uses .env for Novita API key)

Novita AI Integration: Uses OpenAI-compatible endpoint with custom prompt based on user's selected level.

Data Flow Diagram

```
graph LR
    User -- "fills prompt+level" --> Frontend
    Frontend -- "POST /api/explain" --> Backend
    Backend -- "Calls" --> NovitaAI
    NovitaAI -- "Response" --> Backend
    Backend -- "Explanation" --> Frontend
    Frontend -- "Save" --> LocalStorage
```

Project Structure

```
/trae_hack2-main/
|-- .env.example
|-- index.html
|-- main.js
|-- server.js
|-- style.css
|-- README.md
|-- package.json
|-- screenshots/
```

Key Code Snippets

server.js (/api/explain route):

```
app.post('/api/explain', async (req, res) => {
  const { prompt, level } = req.body;
  const systemPrompt = SYSTEM_PROMPTS[level] || SYSTEM_PROMPTS.medium;
  const payload = {
    model: MODEL,
    messages: [
      { role: "system", content: systemPrompt },
      { role: "user", content: prompt }
    ],
    response_format: { type: "text" },
    stream: false
  };
  const response = await fetch(NOVITA_API_URL, { ... });
  const data = await response.json();
  res.json({ explanation: data.choices[0]?.message?.content || "No explanation received." });
});
```

main.js (API call):

```
async function fetchExplanation(prompt, level) {
  const response = await fetch('/api/explain', {
```

Anything Explainer - Hackathon Documentation

Team: HTG ThinkTank | Hackathon: lablab.ai x Trae AI IDE

```
method: 'POST',
headers: { 'Content-Type': 'application/json' },
body: JSON.stringify({ prompt, level })
});
return (await response.json()).explanation;
}
```

Setup / Run Instructions

1. Install Node.js (v16+ recommended)
2. Clone/download project, run: npm install
3. Copy .env.example to .env, add your Novita API key
4. Run the app: node server.js
5. Visit <http://localhost:3000> in your browser

Screenshots

Add screenshots from the screenshots/ folder here, e.g. homepage, explanation, gallery view.

Conclusion / Future Enhancements

- Add support for images/diagrams in explanations
- User accounts & cloud-based knowledge gallery
- Integrate with other LLMs or APIs (e.g., Zilliz for semantic search)

This project showcases Novita AI as the backend intelligence, adapting its output for beginner, student, or advanced needs-making instant, tailored learning available for all.