Doubts:

Deadline: 2 Days
Upload code on github, share the link.
Make the website live on vercel/heroku/ any platform.

In case you have any doubts, message me on internshala.
https://internshala.com/internship/detail/work-from-home-web-development-internship-at-truly-ias1751039621

Product Name: TrendWise

Overview

TrendWise is a full-stack SEO-optimized blog platform that:

- Fetches trending topics (Google, Twitter)
- Uses ChatGPT/ Gemini api to generate articles
- Displays them in a blog-style frontend
- Supports user authentication via Google
- Allows comments on articles

Tech Stack

Layer	Technology
Frontend	Next.js 14+ (App Router)
Styling	TailwindCSS
Authentication	NextAuth.js (Google login)

Backend	Node.js with Express/Fastify or API Routes
Crawler	Puppeteer / Cheerio / Playwright
Database	MongoDB / PostgreSQL
ORM	Mongoose
Hosting	Vercel (Frontend), Render/Heroku (Backend)

Functional Requirements

1. Backend Bot

- Fetch trending topics from Google Trends / Twitter.
- Search for related articles, images, and videos.
- Send this data to ChatGPT (OpenAI API) to generate SEO-optimized content.
- Store in DB:
    - `title`
    - `slug`
    - `meta`
    - `media` (images, tweets, videos)
    - `content`

---

2. Blog Frontend

- **Homepage**: List of all articles (thumbnail, title, excerpt).
- **Detail Page**:
    - Full article view
    - SEO metadata (title, description, OG tags)

    - Embedded media (images, videos, tweets)
- **Search Bar**: Search articles by keyword/title.

---

3. User Authentication

- Google Login via **NextAuth.js**
- Only authenticated users can:
    - Post comments on articles
    - View their own comment history

---

4. ChatGPT Integration

- Use OpenAI API to:
    - Generate article content with SEO structure
    - Include H1-H3 headings, meta descriptions
    - Embed relevant media (tweets, YouTube links, images)
- Ensure OG tags and meta tags are populated for each article.

---

5. Sitemap Generation

- Dynamic `sitemap.xml` with all article URLs
- Available at `/sitemap.xml`

---

6. Robots.txt

- Allow all crawling

- **Disallow** `/admin`
- Provide link to sitemap

---

7. Admin Section *(Optional but recommended)*

- View list of articles (CRUD optional)
- Trigger content bot manually for new trends
- Secure route behind auth middleware

Evaluation Criteria

Area	Criteria
Next.js Usage	App Router, SSR, ISR
API and Backend Logic	Effective integration with OpenAI and crawling logic
Article Generation Quality	Rich, SEO-optimized, media-rich articles
Auth & Comments	Secure Google Auth, functional commenting system
SEO Best Practices	Correct meta tags, OG tags, sitemap, robots.txt
UI/UX	Clean, responsive, user-friendly design
Deployment	Live deployment on Vercel (frontend) and Render/Heroku (backend)

Routes Overview

Route	Purpose
`/`	Homepage (List of Articles)
`/article/[slug]`	Article Detail Page

`/login`	Google Login
`/admin` *(optional)*	Admin Dashboard
`/api/article`	API to create/fetch articles
`/api/comment`	API to post/get comments
`/sitemap.xml`	Dynamic sitemap endpoint
`/robots.txt`	Robots.txt rules