XIHawks Mid Level Developer Task: Smart Link Website

Position: Mid Level Full Stack

What to Build:

You need to make a simple website with two main parts. One part is for admins to add and manage website links. The other part is for users to see and click on these links. We will use AI to write short descriptions for the links.

What You Will Use:

- **Front End (what users see):** Next.js, Redux Toolkit and Styled Components (for custom looks).
- **Back End:** Node.js, Express, PostgreSQL.
- **Smart AI:** Google Gemini (to help write descriptions).

---

1. About This Project

You will build a website like a phone book for useful websites. It will have an "admin page" where you can add, change, or remove website links. There will also be a public page where anyone can look at these links.

2. What You Need To Do

2.1. Back End

- **Tools:**
    - Node.js and Express.js (for making the server work).
    - PostgreSQL (a database to store all information).
    - JWT (for keeping users safe when they log in).
    - Google Gemini (our AI helper for descriptions).
- **Data You Need To Store:**
    - **Users:**
        - **id**: unique number for each user.
        - **username**: user's chosen name (must be different for everyone).
        - **email**: user's email (must be different, good email form).
        - **password**: user's secret code (must be saved safely).
        - **role**: what kind of user they are ('admin' or 'user'). user is the normal type.
        - **createdAt**, **updatedAt**: when the user was made/changed.
    - **Site Links (Website Details):**
        - **id**: unique number for each link.
        - **siteUrl**: the website address (like google.com). Must be a real web address.
        - **title**: the name of the website (like "Google Search").

        - **coverImage**: a picture for the link (optional, give its web address).
        - **description**: a short text about the link (the AI will write this). Must be there after AI makes it.
        - **category**: what kind of website it is (like 'Technology', 'Design', 'News'). Must be chosen.
        - **createdAt**, **updatedAt**: when the link was made/changed.
- **API Points (Ways Your Website Talks):**
    - **Login/Sign Up:**
        - POST /api/auth/signup: To create a new user.
        - POST /api/auth/login: To let a user sign in.
    - **Managing Site Links (Websites):**
        - POST /api/sites: Add a new link (only admins can do this).
        - GET /api/sites: Get a list of all links.
        - GET /api/sites/:id: Get details for one link.
        - PUT /api/sites/:id: Change details of a link (only admins).
        - DELETE /api/sites/:id: Remove a link (only admins).
    - **AI Help:**
        - POST /api/ai/generate-description: Send a website title and category to this. It will ask Gemini (the AI) to give back a short description (like 2-3 sentences).
- **User Safety (Login and Roles):**
    - Use JWT to keep people safe when they use protected pages.
    - Admins can do everything: add, change, remove links. They can also use the AI helper.
    - Normal users can only see the links. They cannot add, change, or remove anything.
- **Fixing Mistakes:** Make sure your server can handle problems well (like wrong passwords, missing info, or database errors).

2.2. Front End (What Users See and Use)

- **Tools:**
    - Next.js (for making the web pages).
    - Redux Toolkit
    - Styled Components (for making your pages look exactly how you want).
- **Web Pages You Need:**
    - **/login Page:** A page to sign in.
    - **/signup Page:** A page to create a new account.
    - **/admin/dashboard Page (Only for Admins):**
        - This is one page for everything an admin does.
        - **Form to Add/Change Links:** Spaces to type in Website Address, Title, Picture Link, and choose a Category (like a list of choices: Technology, Design, etc.).

        - A button next to the Title and Category fields that says "Ask AI for Description". When you click it, the AI will write a description for you in the right box.
        - **Table of Links:** Shows all the website links in a list. Each row has the Title, Category, Description, and buttons to "Change" or "Remove" that link.
        - **Search Box:** A place to type words to find links by their Title or Website Address.
        - **Category Filter:** A list of choices to only show links from a certain category.
    - **/ (Home Page for Everyone):**
        - This page is for all users to see.
        - It shows all the public website links as nice-looking cards.
        - Each card will show the Title, Description, Category, and the Cover Image.
        - If a user clicks on any part of a card, it should open the actual website in a new tab.
        - You can add simple filter options here too (like buttons to show only "Technology" links).
- **Look and Feel:**
    - Use Styled Components for styling.
    - Make sure the website looks correct on big screens (computers), medium screens (tablets), and small screens (phones).
    - Show clear messages when something is loading or if there's an error.

2.3. Ready for Real World (Deployment)

- Write clear steps on how to get your front end and back end ready to work on the internet.
- Tell us how to set up important secret codes (like database passwords, AI keys) without putting them in your code directly.

3. What To Give Us

Send us a zipped folder (like a .zip file) with these parts:

1. **Back End Folder (backend/):**
    - All your Node.js and Express.js code.
    - The package.json file (lists all tools you used).
    - A README.md file (how to start your backend on a local computer, and how to set up secret codes).
2. **Front End Folder (frontend/):**
    - All your Next.js code.
    - The package.json file (lists all tools you used).

    - A README.md file (how to start your frontend on a local computer, and how it works to the backend).

4. How We Will Judge Your Work

We will look at these points:

- **Does It Work? (40% of marks):**
    - Do all the website talking points (APIs) work fine?
    - Can you sign in and sign up?
    - Can admins add, change, remove links?
    - Does the AI make descriptions correctly?
    - Do search and filters work on the admin page?
    - Does the user page show links and open them when clicked?
- **Back End Quality (20% of marks):**
    - Is your code clean and easy to read?
    - Did you use Express.js and PostgreSQL well?
    - Does it handle errors well and keep things safe (like passwords)?
    - Is the AI connected correctly?
- **Front End Quality (20% of marks):**
    - Did you use Next.js, and Styled Components well?
    - Is the page layout clear?
    - Does it look good on all screen sizes (phones, tablets, computers)?
    - Does it get information from the backend smoothly?
- **AI Part (10% of marks):**
    - Did you use Gemini the right way to make descriptions from titles and categories?
    - Does the website show the AI's answer well?
- **Code Order & Guides (10% of marks):**
    - Are your project files well-organized?
    - Are your README.md files clear and helpful?
    - Is your database setup file correct?