HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

SCHOOL OF INFORMATION AND TECHNOLOGY

---------------o0o---------------

FINAL PROJECT REPORT

Course: Graduation research 1

Đề tài : Website Application - Blog

Guidance lecturer:	**Thầy Ban Hà Bằng**
Student:	**Lê Ngọc Kiên – 20184280**

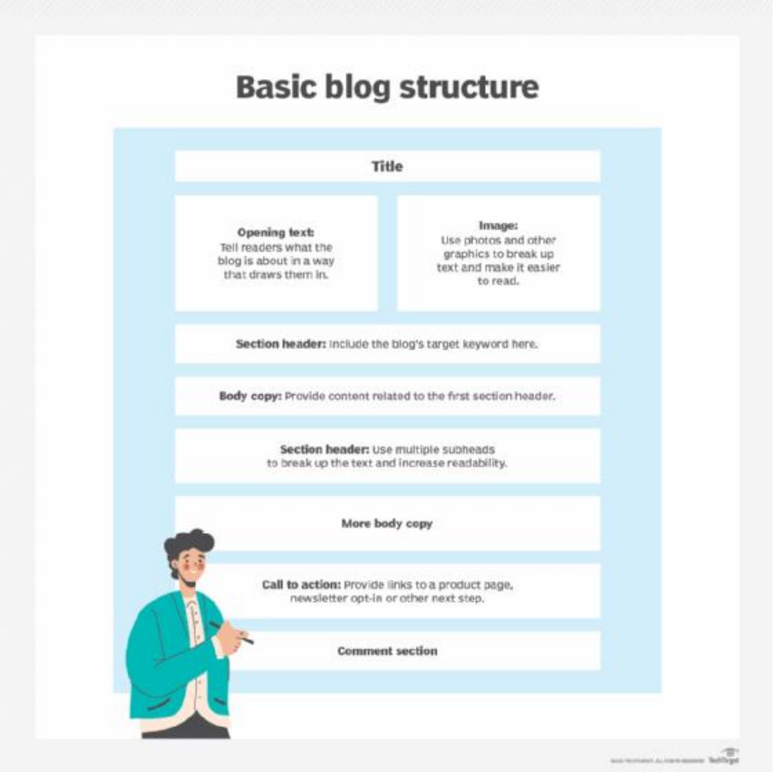
TP. HÀ NỘI, THÁNG *2* NĂM *2023*

Table of content

CHAPTER 1. PROJECT DESCRIPTION

A blog is a website or page that is a part of a larger website. Typically, it features articles written in a conversational style with accompanying pictures or videos. Blogging is a fun and flexible way for self-expression and social connection, so it is no wonder blogs have become very popular. A blog is frequently updated for personal commentary or business content. Blogs are often interactive and include sections at the bottom of individual blog posts where readers can view information. Most are written in a conversational style to reflect the voice and personal views of the blogger. Some businesses use blogs to connect with target audiences and sell products.

Blogs were originally called weblogs, which were websites that consisted of a series of entries arranged in reverse chronological order, so the newest posts appeared at the top. Here is the simple description of a blog:


Basic blog structure
Title
Opening text: Tell readers what the blog is about in a way that draws them in.
Image: Use photos and other graphics to break up text and make it easier to read.
Section header: Include the blog's target keyword here.
Body copy: Provide content related to the first section header.
Section header: Use multiple subheads to break up the text and increase readability.
More body copy
Call to action: Provide links to a product page, newsletter opt-in or other next step.
Comment section

1. MAIN MODULES

I have built a blog that is an information sharing website with the following features:

- Login, Register, update personnel info.
- Allow users to view articles/posts. User can view detail of the post.
- Allow users to view articles/posts by separates categories.
- Allows users to create articles/posts, attach information such as photos, topics, ...
- Allows users to update/delete their articles/posts.

2. TECHNOLOGY

For this project, the technologies I used are as follows:

- Back-end:
  - NodeJS is the main framework backend.
  - ExpressJS supports HTTP methods and middleware creating a powerful and easy-to-use API for connection to Front-end.
  - JsonWebToken(JWT) is a means of requesting information transfer between two parties Client - Server, the information in the JWT string is formatted in JSON.
  - MySQL is the database management system

- Front-end:
  - ReactJS is the main framework frontend.
  - Sass is used to handle the global.
  - Basic primary language is javascript

- Other: Github as source code management

CHAPTER 2: PROJECT SPECIFIC DESCRIPTION

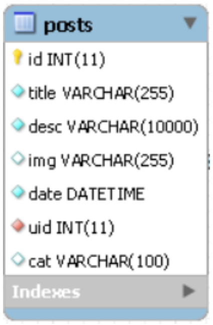
1. SYSTEM DESIGN

The system will have only **1 agent** – **user/guest**

1.1. Usecases

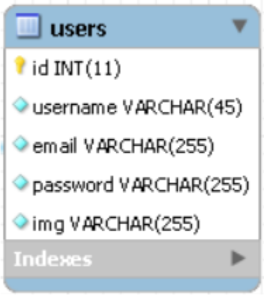
- **Users manage Posts**
    - View post's details.
    - Write new post.
        - Fill the form: title, description
        - Upload a picture (optional)
        - Choose category (optional)
    - Update post (User's post).
        - Update title, description, picture
    - Delete post (User's post).
- **User interact with posts**
    - View posts in categories
    - Comment in post (not yet)
- **Guests Register an account**
    - Fill info: email, username, password, img(updating), ...
- **Guests view posts**
    - View posts in general (not specific)
    - View posts in category (not specific)

1.2. Database



posts

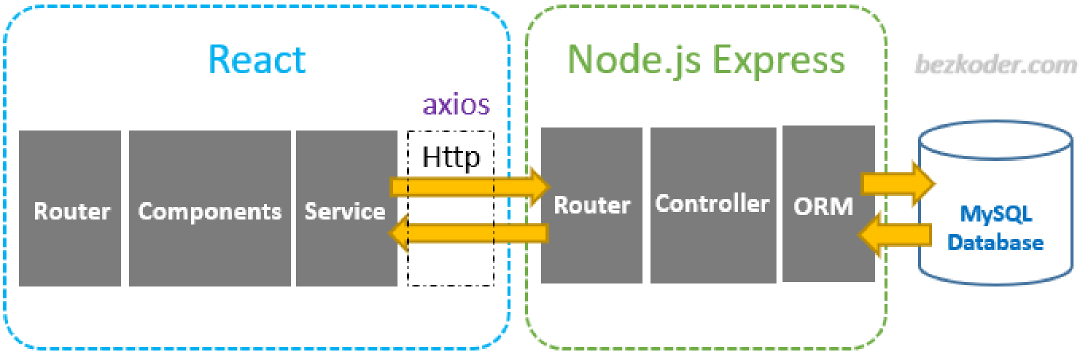
id INT(11)
title VARCHAR(255)
desc VARCHAR(10000)
img VARCHAR(255)
date DATETIME
uid INT(11)
cat VARCHAR(100)
Indexes



users

id INT(11)
username VARCHAR(45)
email VARCHAR(255)
password VARCHAR(255)
img VARCHAR(255)
Indexes

1.3. System diagram



1.4. Back-end detail

NodeJS as the back-end framework is an open-source and cross-platform JavaScript runtime environment that is used to run web applications outside of the client's browser. The platform is considered a perfect solution for data-intensive applications thanks to its asynchronous event-driven model.

Currently the project has the following modules:

- **Authentication**: tasks that require authentication. This module provides protection (block access if the user is not authenticated).
- **Post CRUD**: control API of posts.
- **Upload post**: add a new post with different implication.
- **User CRUD**: control API of users.

The backend module can be a pure module that provides functionality (by providing providers) or a module that has a controller and can receive and respond to requests, or both. Within a module, there are There may be files:

- **express**: supports HTTP methods and middleware creating API for connection to Front-end.
- **bcryptjs**: a form of hash function commonly used to encrypt passwords for the purpose of increasing security.
- **cookie_parser**: a middleware in Expressjs used to parse cookies
- **jsonwebtoken**: means of requesting information transfer between two parties Client - Server, the information in the JWT string is formatted in JSON.
- **multer**: a middleware for Express and Nodejs that makes it easy to handle multipart/form-data when users upload files.

1.5. Front-end detail

Frontend source code is divided into separate folders:

- **Public/upload**: Contains media data resources used in the application.
- **Components**: Components are used to create application screens. Each component has action files, reducers used to create global state and actions to communicate with the API. A component can be imported by another component.
- **Hooks**: custom hooks used in the application.
- **Router**: Navigate between screens of the app, change the app URL and keep the interface in sync with the URL.
- **Page**: App main screens UI
- **mixin.scss**: for responsive design base on devices.

2. PRODUCT

2.1. Front-end display

2.1.1. Settings & deployment

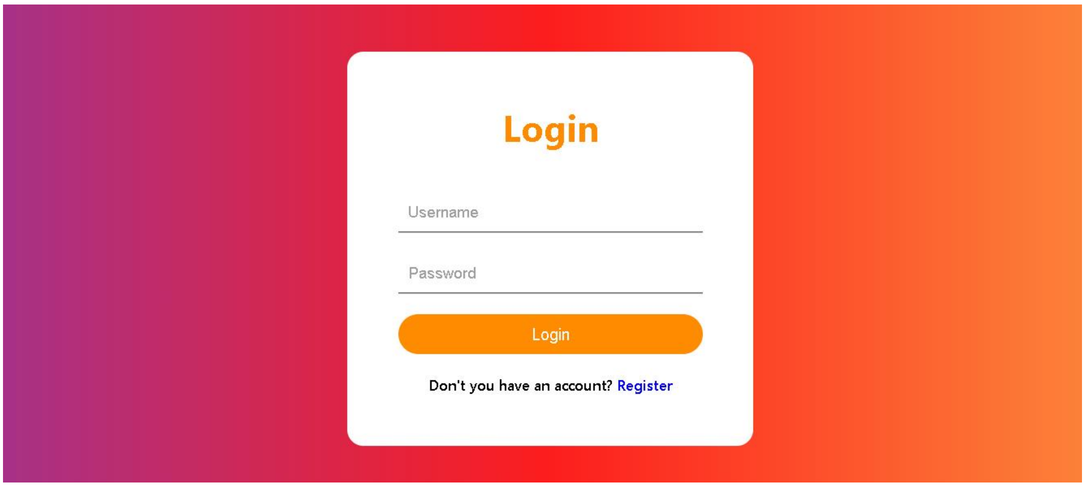
Front-end can be run locally by following these steps:

- B1: Clone repository: https://github.com/lengockien2000/GR1.git :`cd client`
- B2: Setting dependency: `npm install`

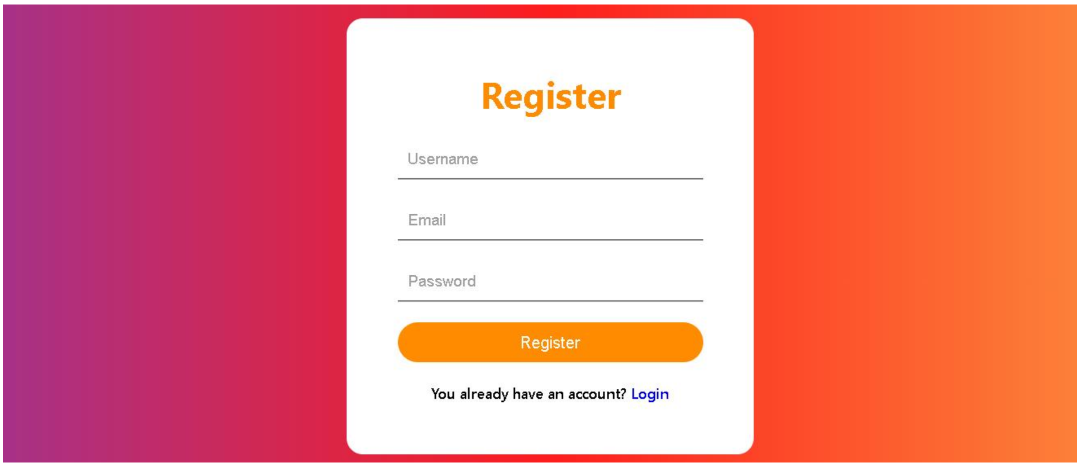
  Run server development: `npm start`
- B3: Go to browser at: http://localhost:3000/

2.1.2. Interfaces

- Login/ Register:

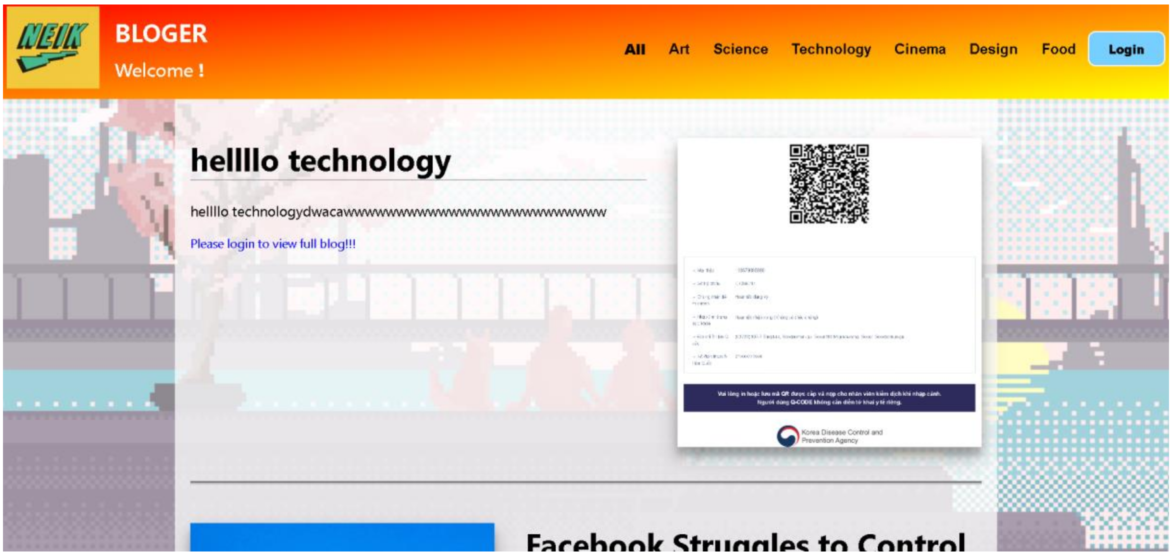


Don't you have an account? Register



You already have an account? Login

- Other Interfaces:

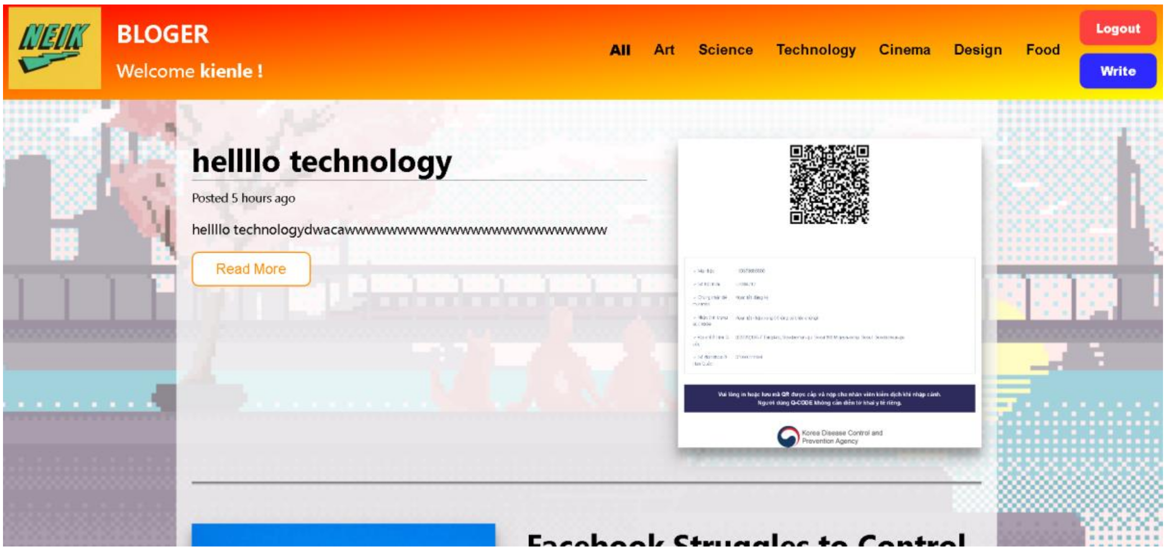


hellllo technology

hellllo technologydwacawwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww

Please login to view full blog!!!

Facebook Struggles to Control

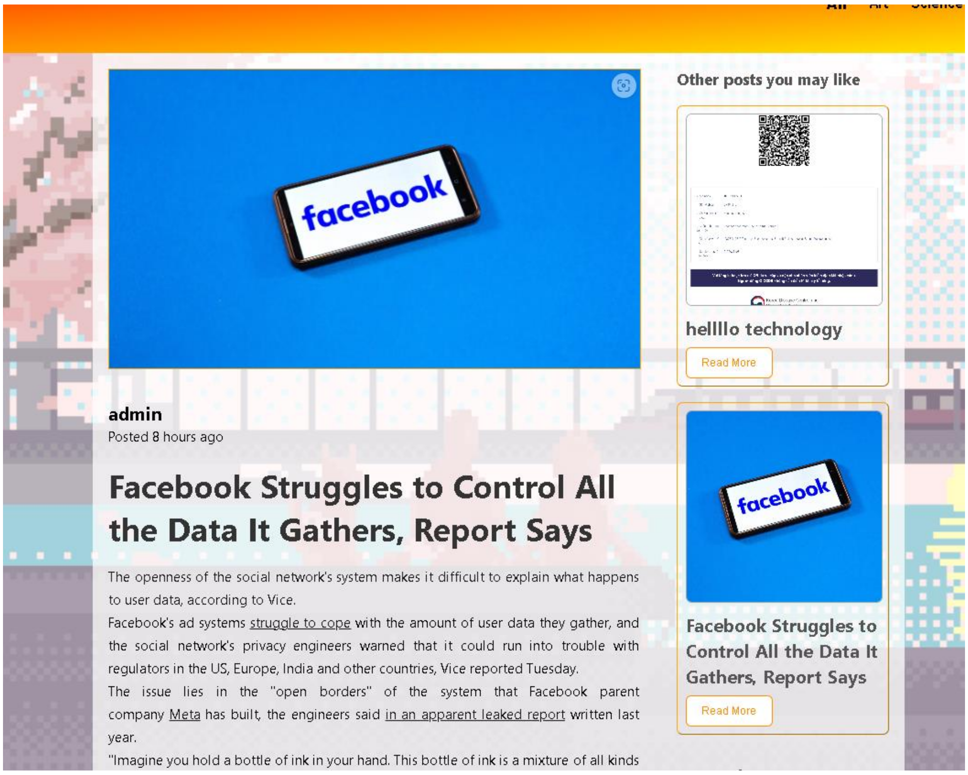


hellllo technology

Posted 5 hours ago

hellllo technologydwacawwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww

Facebook Struggles to Control



Other posts you may like

hellllo technology
Read More
facebook
Facebook Struggles to Control All the Data It Gathers, Report Says
Read More

admin

Posted 8 hours ago

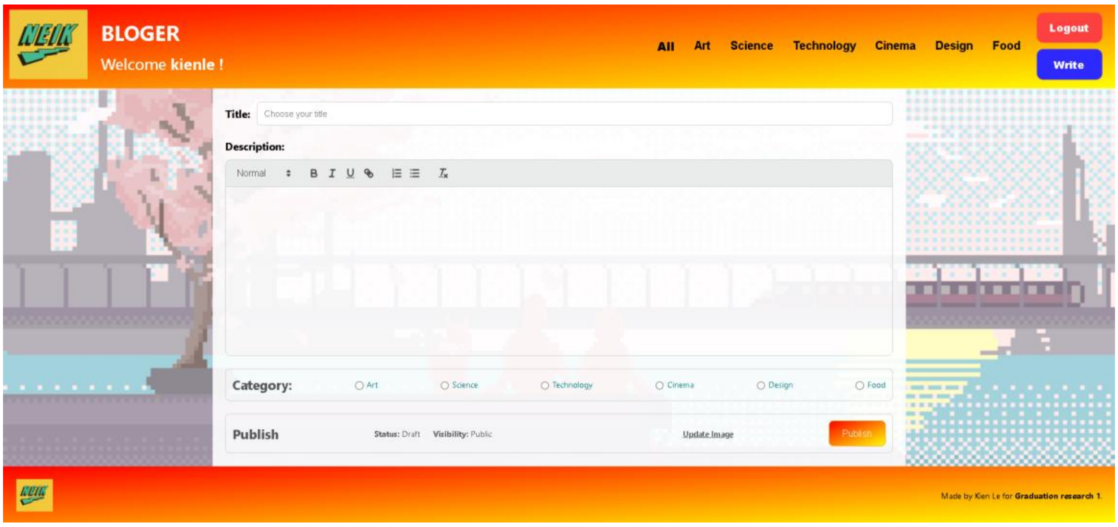
Facebook Struggles to Control All the Data It Gathers, Report Says

The openness of the social network's system makes it difficult to explain what happens to user data, according to Vice.

Facebook's ad systems struggle to cope with the amount of user data they gather, and the social network's privacy engineers warned that it could run into trouble with regulators in the US, Europe, India and other countries, Vice reported Tuesday.

The issue lies in the "open borders" of the system that Facebook parent company Meta has built, the engineers said in an apparent leaked report written last year.

"Imagine you hold a bottle of ink in your hand. This bottle of ink is a mixture of all kinds



Title: Choose your title

Description:

Category: ○ Art ○ Science ○ Technology ○ Cinema ○ Design ○ Food

Publish **Status:** Draft **Visibility:** Public Update Image Publish

Made by Kien Le for **Graduation research 1**.

2.2. API Endpoints, Back-end

Setting & deploy

Back-end can be run locally by following these steps:

- Update database as described (for local using MySQL Workbench)
- B1: clone repository: https://github.com/lengockien2000/GR1.git: `cd api.`
- B2: manage dependencies: `npm install`

  Run back-end: `npm start`

3. Conclusion

Through a large exercise, from designing and developing a complete product, I have learned a lot of experience and skills from basic to advanced in the field of web application development in general, and Technical Restful APIs,... in particular. Besides, I also improved soft skills such as time management skills, document reading skills, etc. In addition, I also learned more about software development support tools such as source code management tools. , code tweaking utility, ...

I would like to thank Mr. Ban Ha Bang for guiding and checking my knowledge in the process of approaching and implementing this topic.