Chapter: 01 - Introduction

1.1 Introduction

Canim is a blogging-based Q&A-type web project. That manages user posting blogs by the way here, the user can be a signed-in “user” or “admin”. Each user can track likes, comments & watches for each post. A signed user can answer each question.

1.2 Objectives

As it consumes the capability of management for each various type of post and category so it becomes a minimal process like the following:

- It makes the blogging system easy
- It makes more sense for Q&A architecture
- Let people understand all kinds of consequences
- To be more resourceful so provide user’s useful resources
- Helps users to assimilate new materials

1.2 Technologies

The whole project is based on multi-paradigm latest technologies & which are broken down into parts:

- Frontend: React.JS, Tailwind CSS, React Router DOM, React Icons, Redux & React Hook From
- Backend: Node.JS, Express.JS, Dotenv, Cors Policy, Mongoose, Bcrypt.JS & Nodemailer
- Database: MongoDB & Cloudinary
- Deploy: Vercel (Frontend), Render (Backend) & Cloudinary (Images)

1.3 Features

As there are no age restrictions, therefore it is open to all types of users. Apart from that each port can relate to use potentially and pay them for reputation which the creator gains from likes, comments, and watches along with answering tracker. The following represents the top features:

- Secure authentication with sign-in, sign-up & forgot-password
- As an author and signed-in user can CRUD or Create, Read, Update or Delete a post & tag
- Signed-in user can only interact with his making posts & tags
- Admin can also CRUD or Create, Read, Update or Delete a post & tag
- Admin can interact self with all users’ posts & tags
- Admin can also feature any post at any time
- Each tag able to display retained posts
- Each post is able to track likes, comments & watches of signed-in users
- Each post contains a copy-link & read-time capacity
- Secure payment method and approach membership
- Secure profile upgradation with form validation
- Income-generated revenue from each post
- Personal information security
- Private routing for users and admin
- A user-friendly UI/UX

These are a bunch of top features which is properly interactive and responsive based on devices.

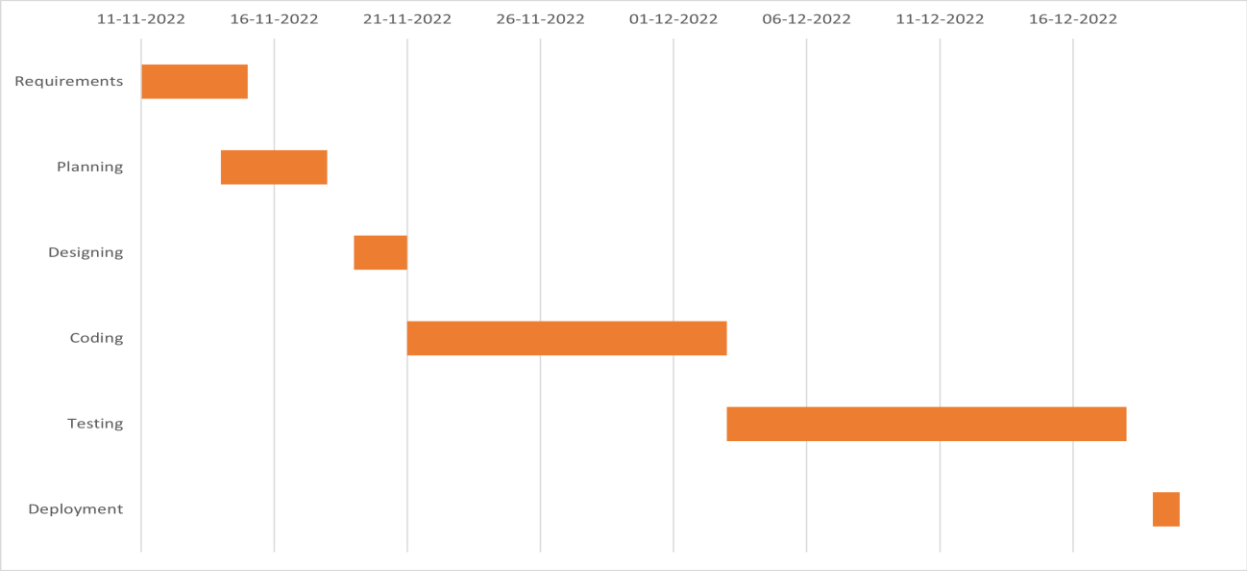
Chapter 02 – Proposed Methods

2.1 System Design Methodology

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of system theory to product development. There is some overlap with the disciplines of system analysis, systems architecture, and systems engineering.

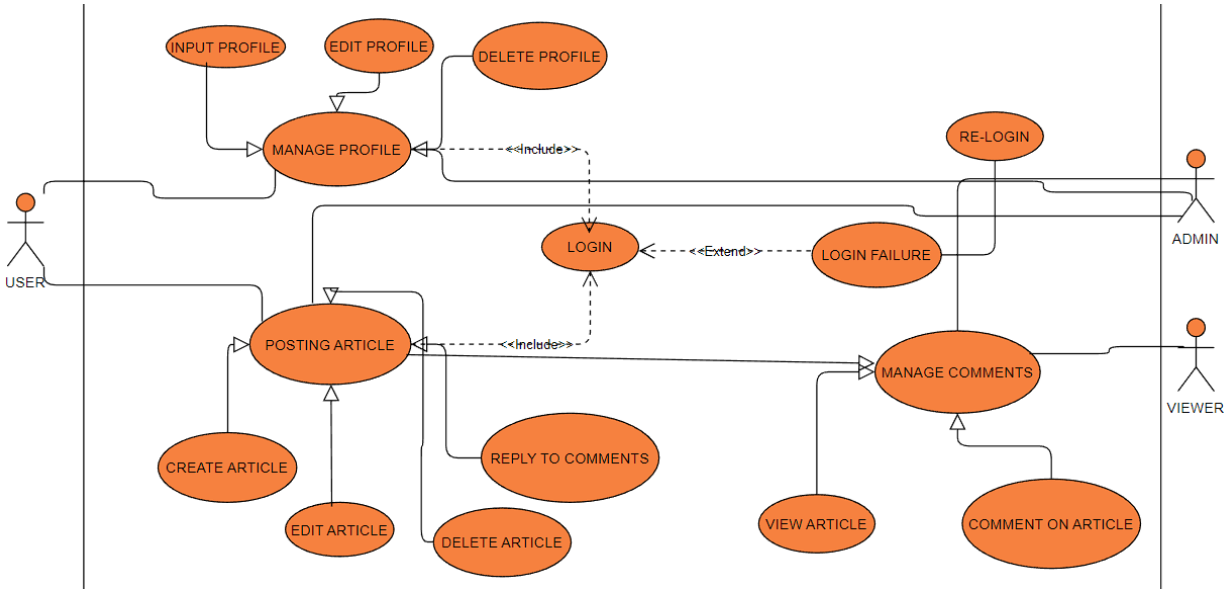
2.2 Gantt Chart

A Gantt Chart is a horizontal bar chart used in project management to visually represent a project plan over time and in a mannered sequence. Follow illustrate our project Gantt Chart:



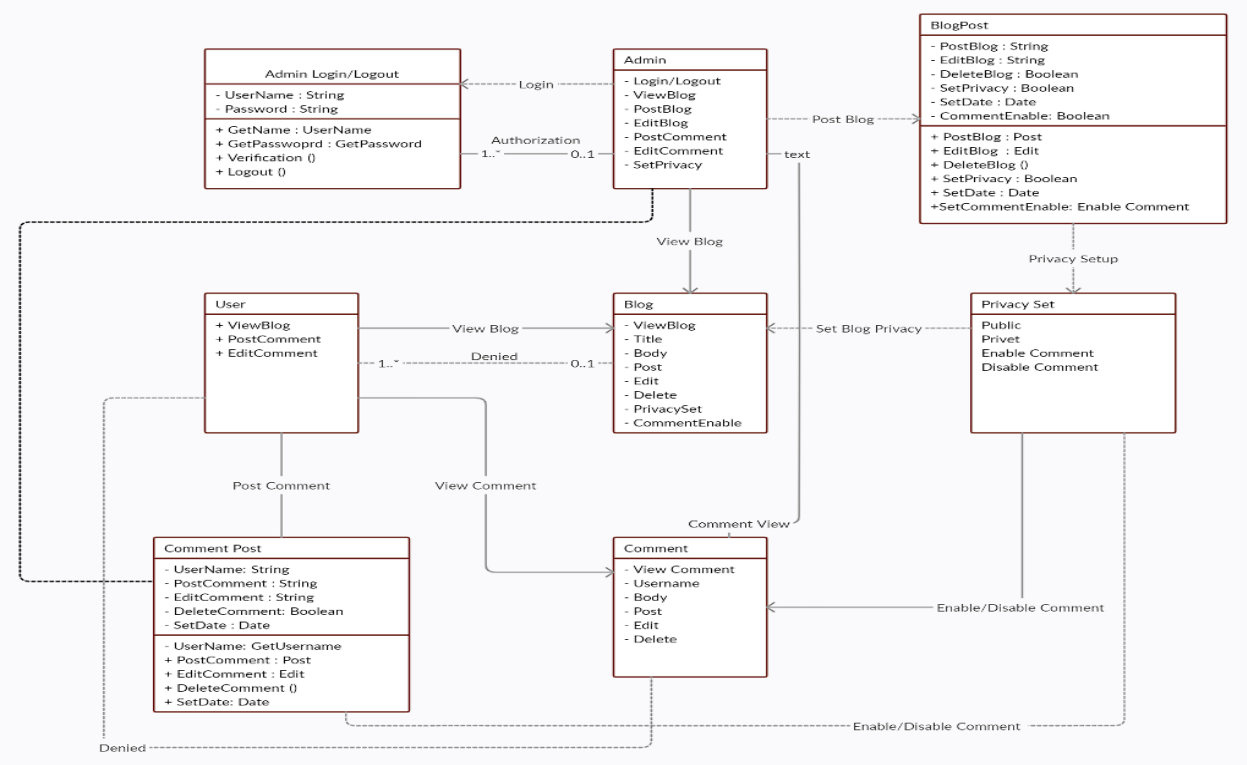
2.3 Use Case Diagram

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.



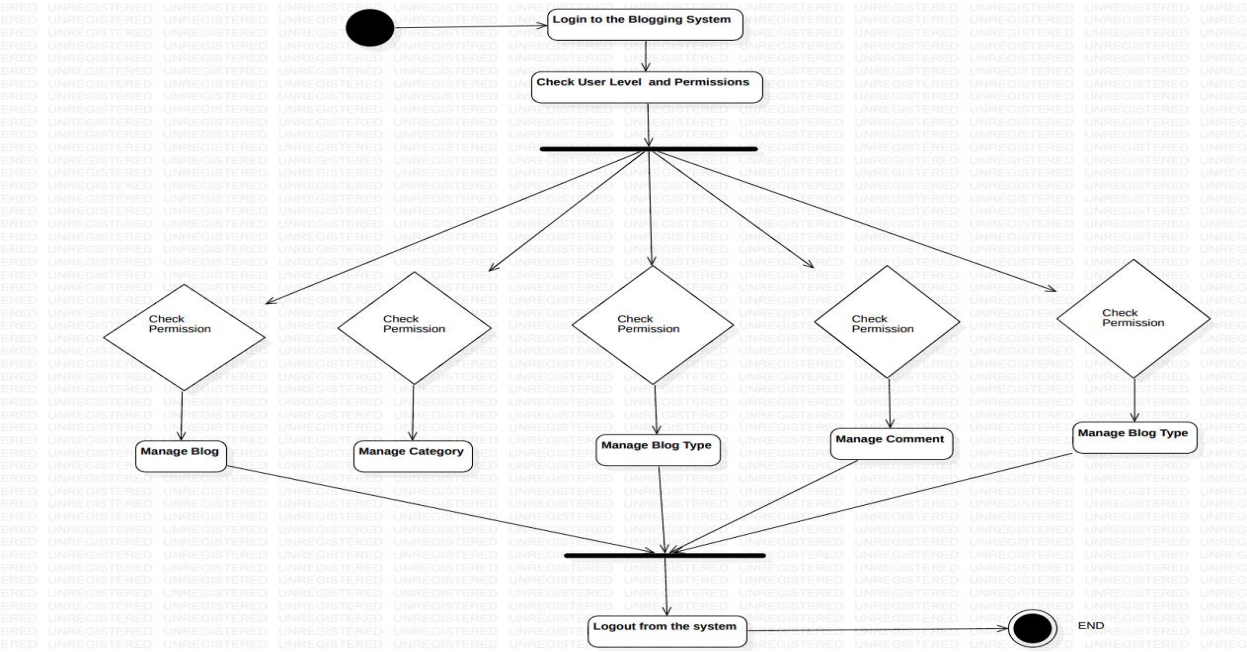
2.4 ER Diagram

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. ER diagrams illustrate the logical structure of databases.



2.5 Activity Diagram

The activity diagram is another important diagram in UML to describe the dynamic aspects of the system. An activity diagram is basically a flowchart to represent the flow from one activity to another activity.



Chapter 03 – System Implementation and Testing

3.1.1 Planning

Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.

3.1.2 Requirement Analysis

In our proposed system every post will have a JWT containing information about the creator and viewer. The proposed system allows users or witnesses to report incidents or complaints to the security department at any time (24/7) by using the internet only. Key stakeholders and users meet to identify business requirements that are quantifiable, relevant, and detailed.

3.1.3 Designing

In this Phase after successfully, analyzing the requirements and planning the next and most important part is design. Web design encompasses many different skills and disciplines in the production and maintenance of websites. The different areas of web design include web graphic design; interface design; authoring, including standardized code and proprietary software; user experience design; and search engine optimization. Often many individuals will work in teams covering different aspects of the design process, although some designers will cover them all. The term "web design" is normally used to describe the design process relating to the front-end (client side) design of a website including writing markup. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third-party modules (if any). The internal design of all the modules of the proposed Fig: 3.6 architecture should be clearly defined with the minutest of the details in the Design Document Specification.

3.1.4 Developing, Coding & Implementation

In this Phase Partially or completely designed according to the product requirements document, the next development phase can start, or both design and development work continuing depend on each other. The goal of this phase is to create an application that meets the needs identified in the previous steps, and as the needs generally evolve and new ideas of implementation may arise during the implementation phase, it is generally advised to use a methodology that allows flexibility and proactivity.

3.1.5 Testing

Test the code against the requirements to make sure the product is actually solving customer needs. This phase includes unit testing, integration testing, system testing, and acceptance testing. During that phase, Test-Driven Development (TDD) is one approach that aims to make sure that tests are dressed to test and simulate real behavior. Testing helps actors involved in the web application development process understand the code’s purposes and makes the handout process easier. Good code coverage has proven to minimize bugs and is proof of the good design for easy post-production support and maintenance, and easy system evolution.

3.1.6 Deployment

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Sometimes product deployment happens in stages as per the business strategy of that organization. The product may first be released in a limited segment and tested in the real business environment (UAT- User acceptance testing). Then based on the feedback, the product may be released as it is or with suggested enhancement in the targeted market segment. After the product is released in the market is maintained.

3.2 SDLC Model

SDLC stands for Software Development Life Cycle. It is a process used by software development organizations to plan, design, develop, test, and deploy high-quality software products. Different SDLC models exist, but they all generally include the following phases: requirement gathering and analysis, design, implementation or coding, testing, and maintenance. Each phase has specific goals and deliverables that must be met before moving on to the next phase. The choice of which SDLC model to use will depend on the specific needs of the organization and the project. Examples of SDLC models include Waterfall, Agile, and Scrum. Software development is a cumbersome activity requiring proper identification of requirements, their implementation, and software deployment. However, the activities do not end there. After the distribution of the software, proper maintenance has to be provided in a timely manner. SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software.

3.3 Agile Model

Agile is a software development methodology that prioritizes flexibility and collaboration. It is an iterative and incremental approach to software development, where requirements and solutions evolve through the collaborative effort of self-organizing and cross-functional teams. Agile methodologies are based on the Agile Manifesto, a set of values and principles for Agile software development that prioritizes individuals and interactions, working software, and customer collaboration over processes and tools.

Some of the key principles of Agile software development include:

- Breaking down a project into smaller, manageable chunks of work called iterations or sprints.
- Prioritizing customer needs and delivering working software as early and frequently as possible.
- Encouraging face-to-face communication among team members.
- Allowing for changes and adjustments to be made throughout the project.
- Continuously testing and integrating code.

Some of the popular Agile methodologies include Scrum, Kanban, and Lean software development. It’s an aggressive modeling approach to quite induce software development.

3.4 Waterfall Model

The Waterfall model is a linear sequential approach to software development, where progress is seen as flowing steadily downwards (like a waterfall) through the phases of a project, which include requirements gathering and analysis, design, implementation, testing, deployment, and maintenance.

The main characteristics of the Waterfall model are:

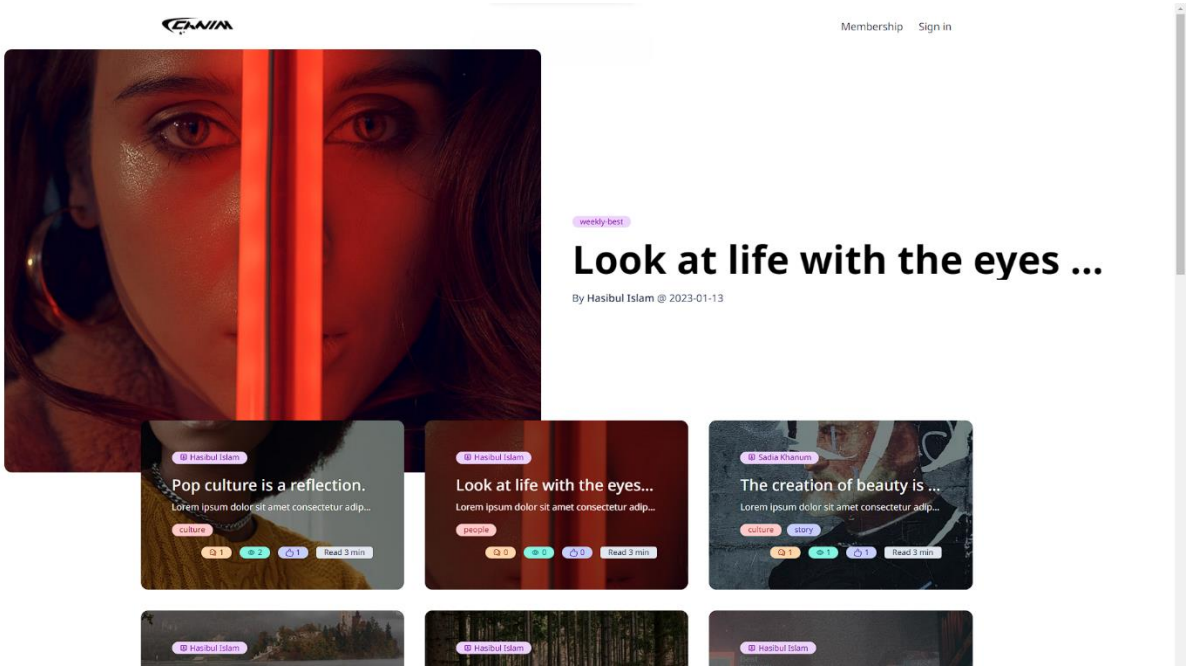
- The phases of the model are executed in a linear sequence, meaning that one phase must be completed before the next phase begins.
- The requirements are gathered and analyzed at the beginning of the project, and then the design, implementation, and testing phases follow in a linear sequence.
- Once a phase is completed, it is considered to be "frozen" and changes to that phase are minimized.
- The testing phase is done only after the implementation phase is completed.
- The waterfall model assumes that the client knows exactly what they want and that the requirements will not change during the development of the software.

The Waterfall model is best suited for projects where the requirements are well understood and unlikely to change, and the final product can be well-defined at the start of the project. It is not suitable for projects where the requirements are not well understood or are likely to change.

Chapter 05 – Result

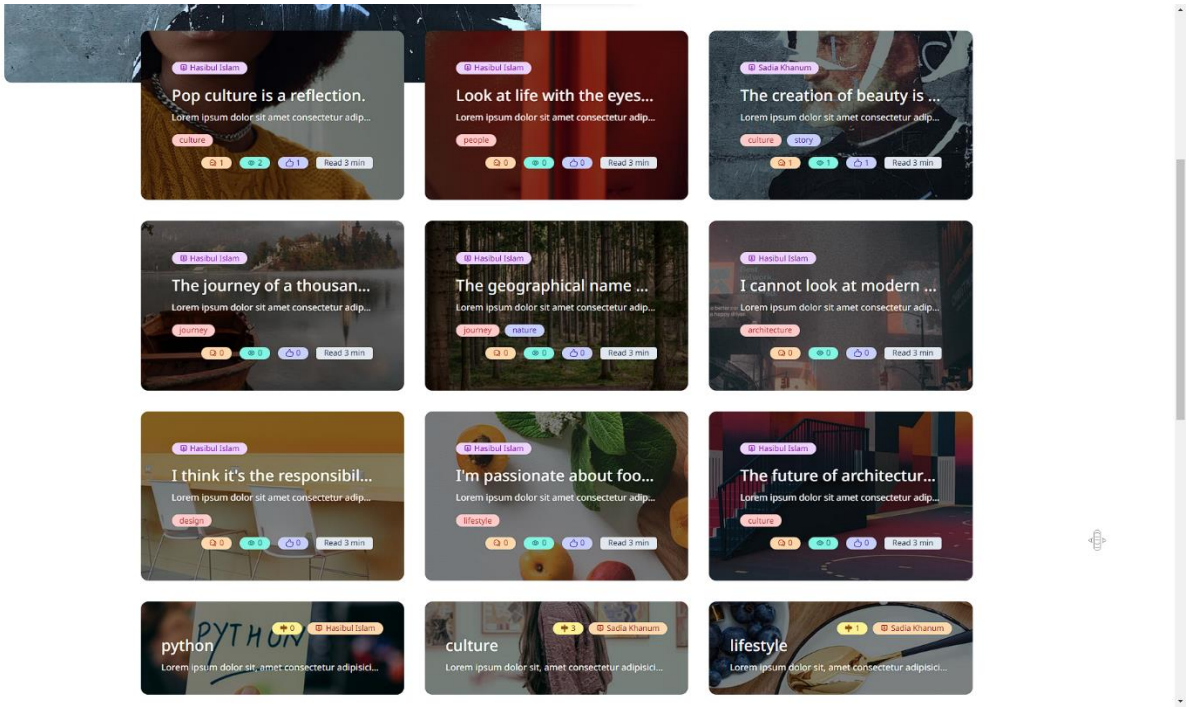
5.1 Hero Section

Here display those posts approved by the admin as “Featured” “Weekly Best” or “Preemptive” that are visible to all the user and non-user using this website.



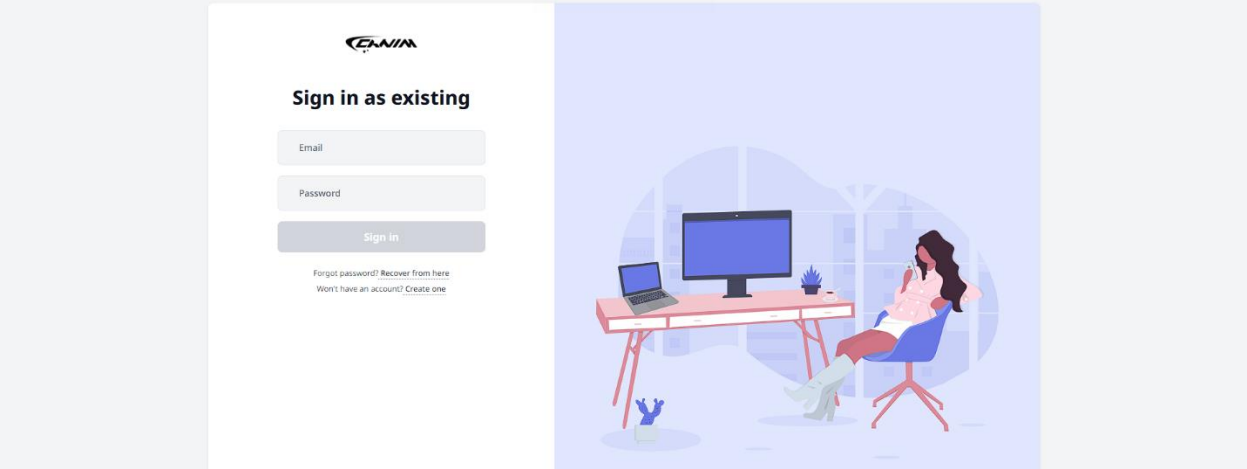
5.2 Banner Section

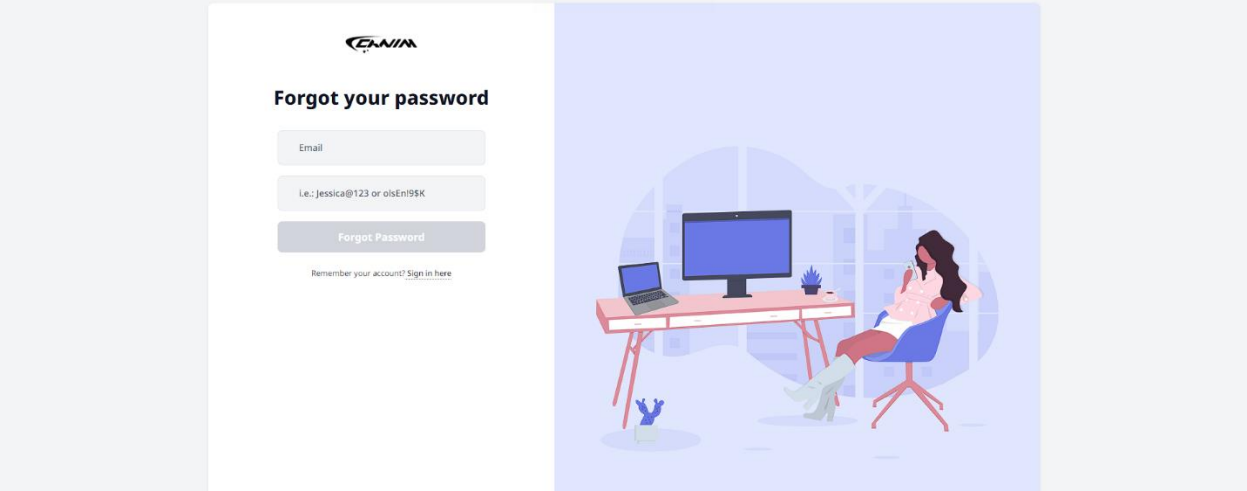
Here display all posts and tags simultaneously to the user with carded description that means watches, likes, and comments for post card and count of post for tag cards.

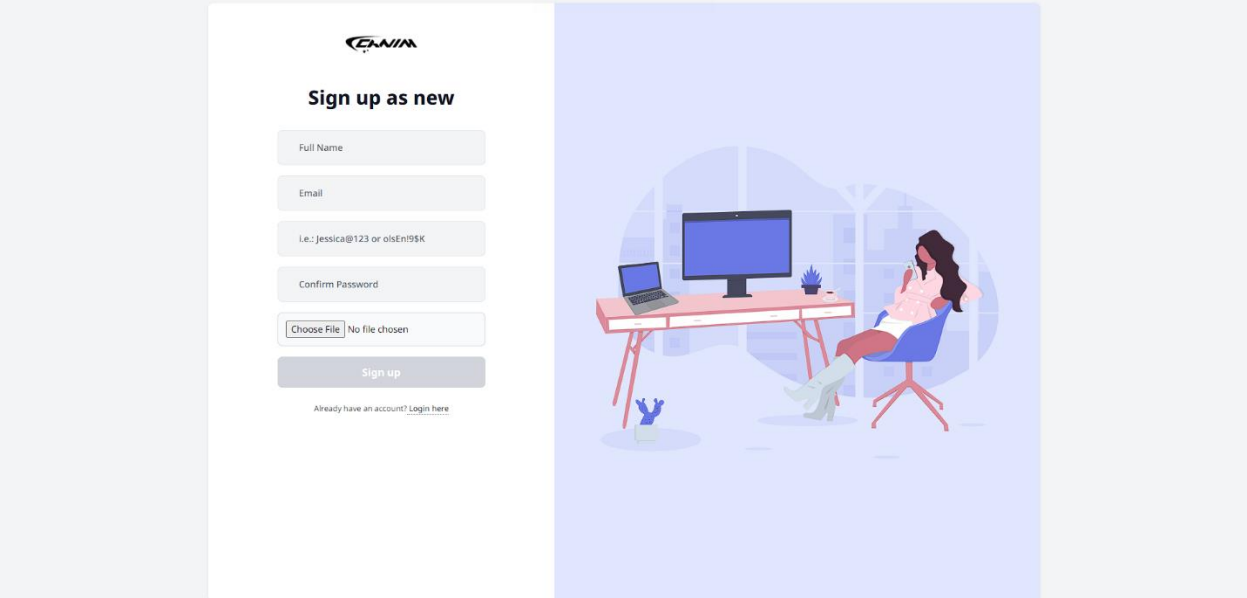


5.3 Authentication Pages

An end-user can choose what to do such as: “Sign-in” or “Sign-up” or “Forgot-password” be authorized.

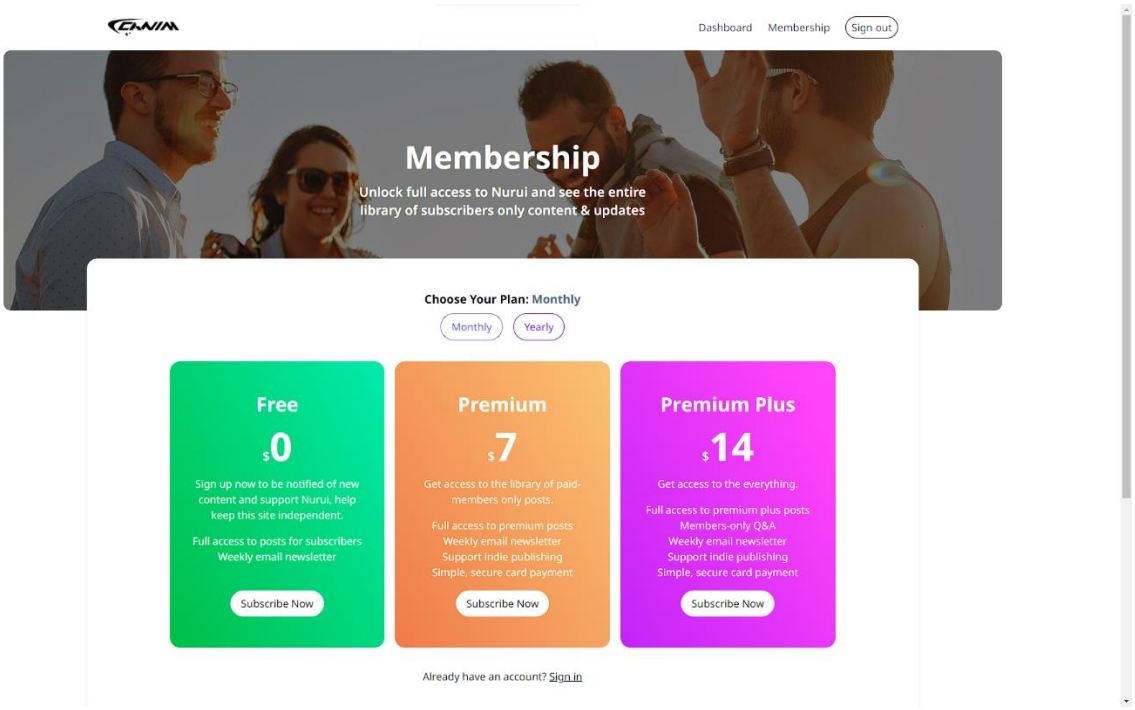






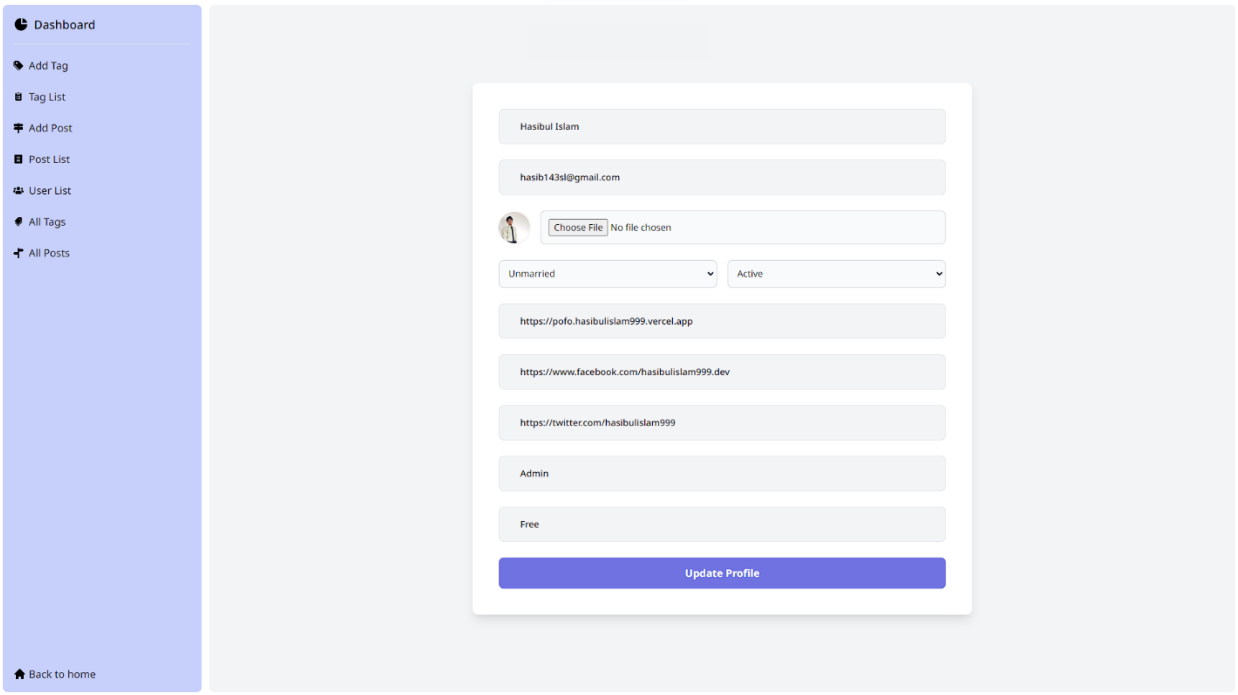
5.4 Membership Page

To read premium blogs a user must have a “Premium” or “Premium Plus” membership to step up. “Premium” for a monthly subscription and “Premium Plus” for a yearly subscription.



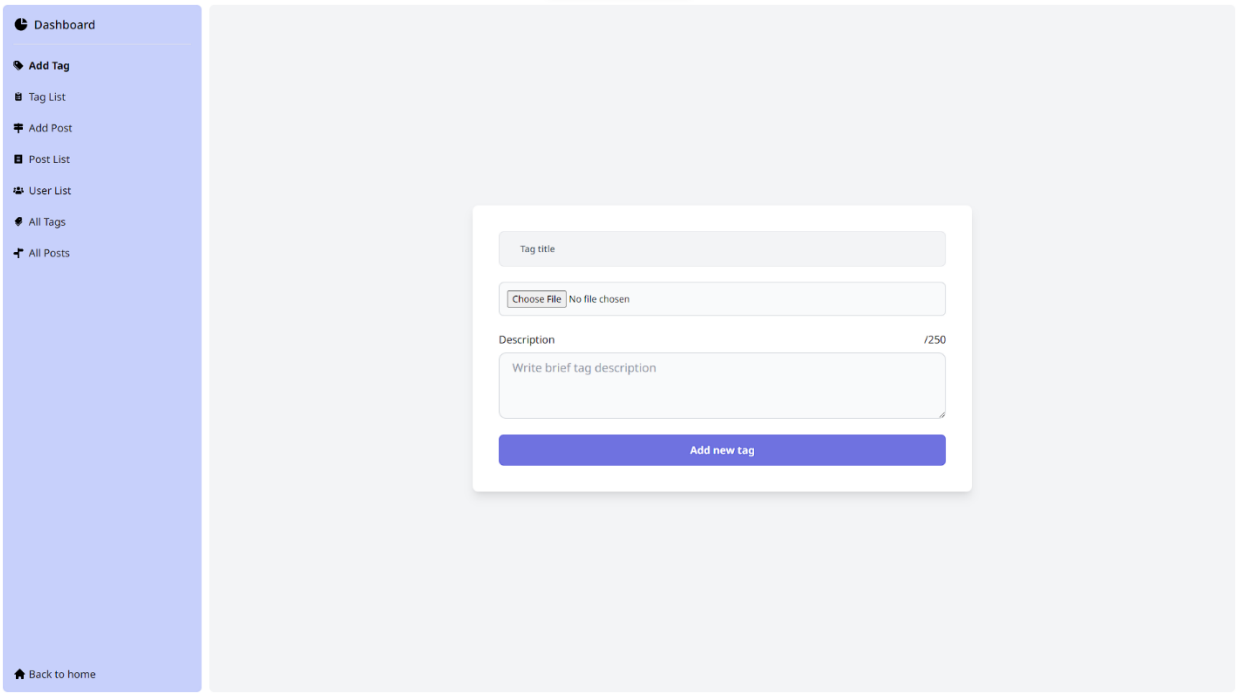
5.5 Dashboard – Update Profile

Dashboard or in other words, it is known as “Panel” for user or admin. Here, the index page is “Update Profile” the user must require demand to update their profile at the basis of time to time.



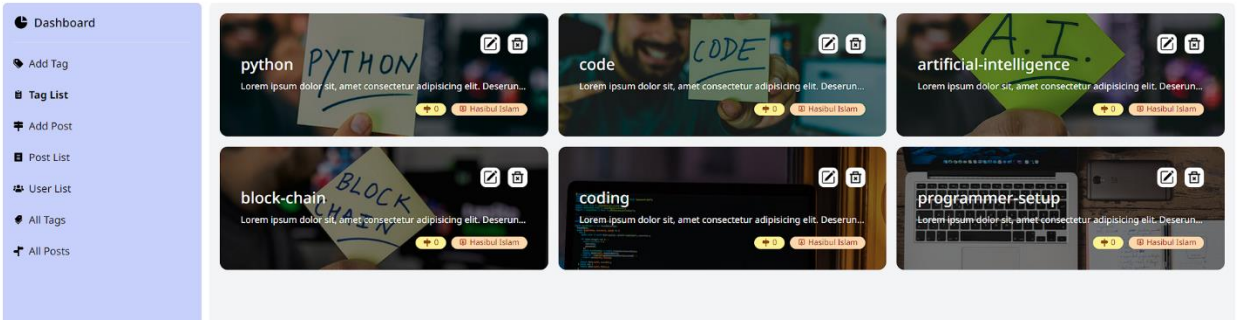
5.5.1 Dashboard – Add Tag

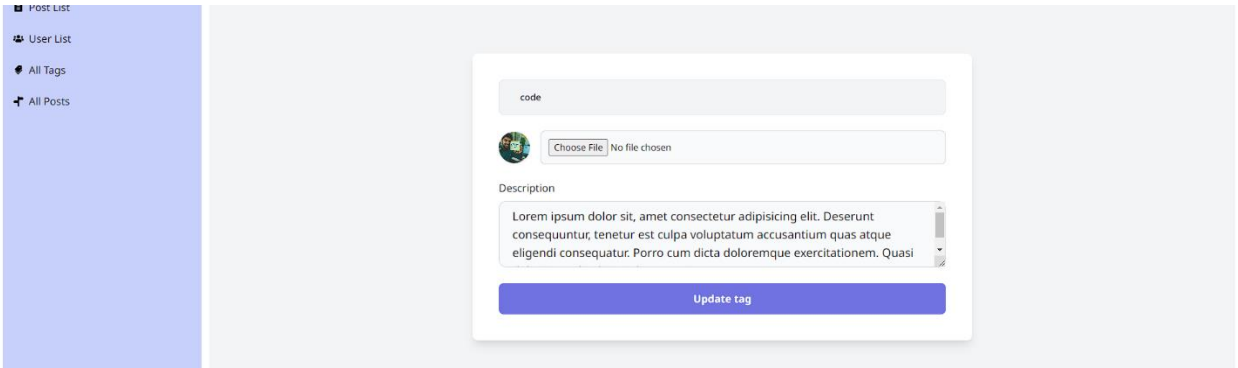
From this route, a “user” or an “admin” can add a literal tag.



5.5.2 Dashboard – Tag List

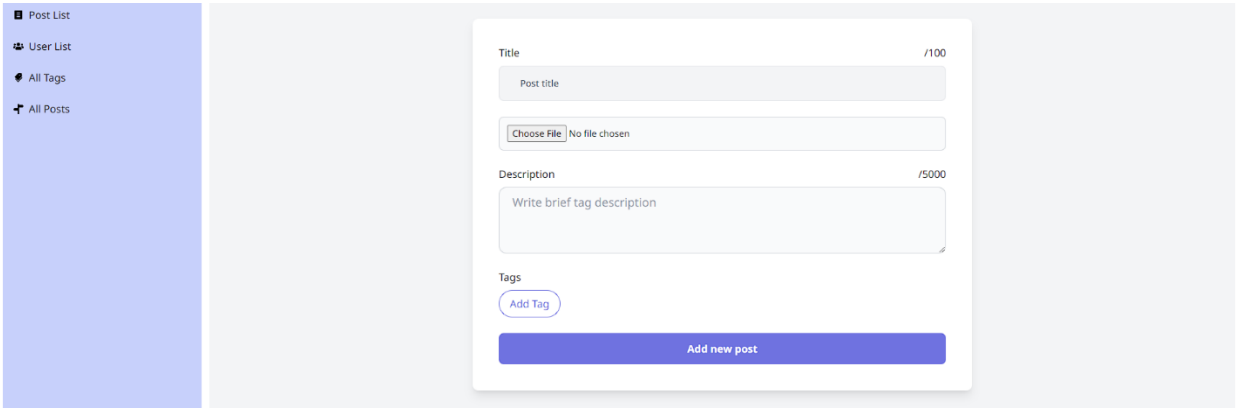
In this route, the user whether it is a “subscribed user” or an “admin” can see only his/her created tags. Where that user is able to “remove” his/her required unwanted tag/tags. Throughout their requirement, they can update their tags. By the way “update” can be done by “admin” to as like as the subscribe “user”.





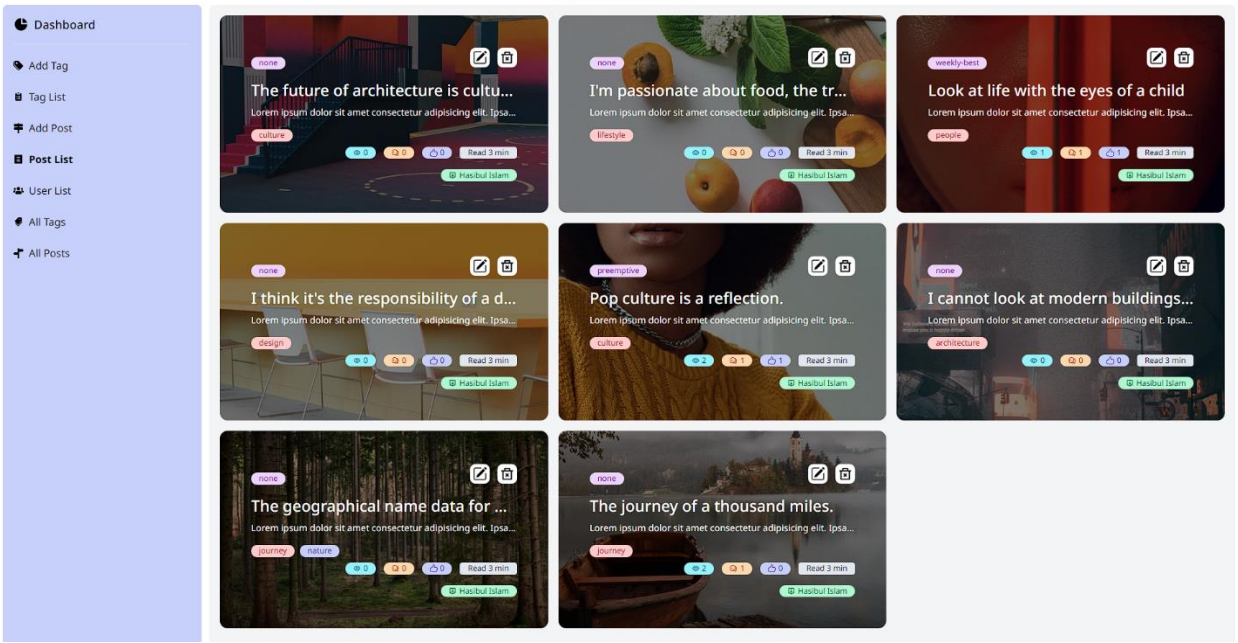
5.5.3 Dashboard – Add Post

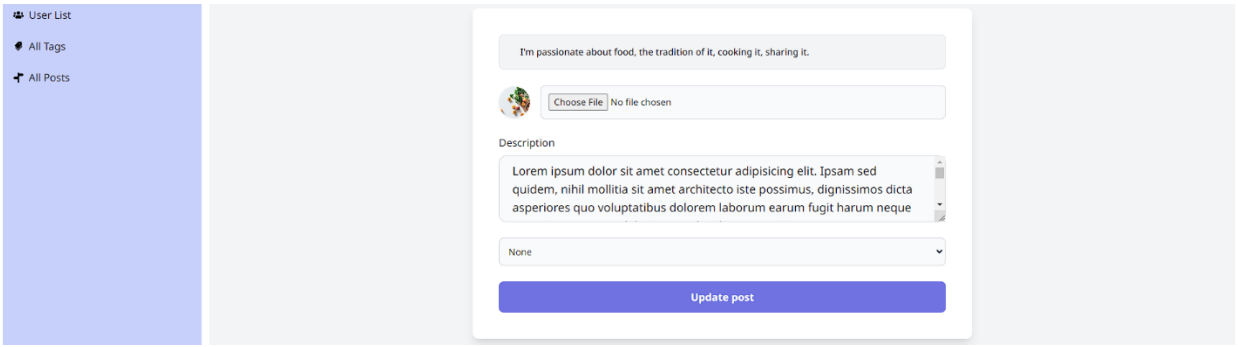
From this route, a “user” or an “admin” can add a literal post.



5.5.4 Dashboard – Post List

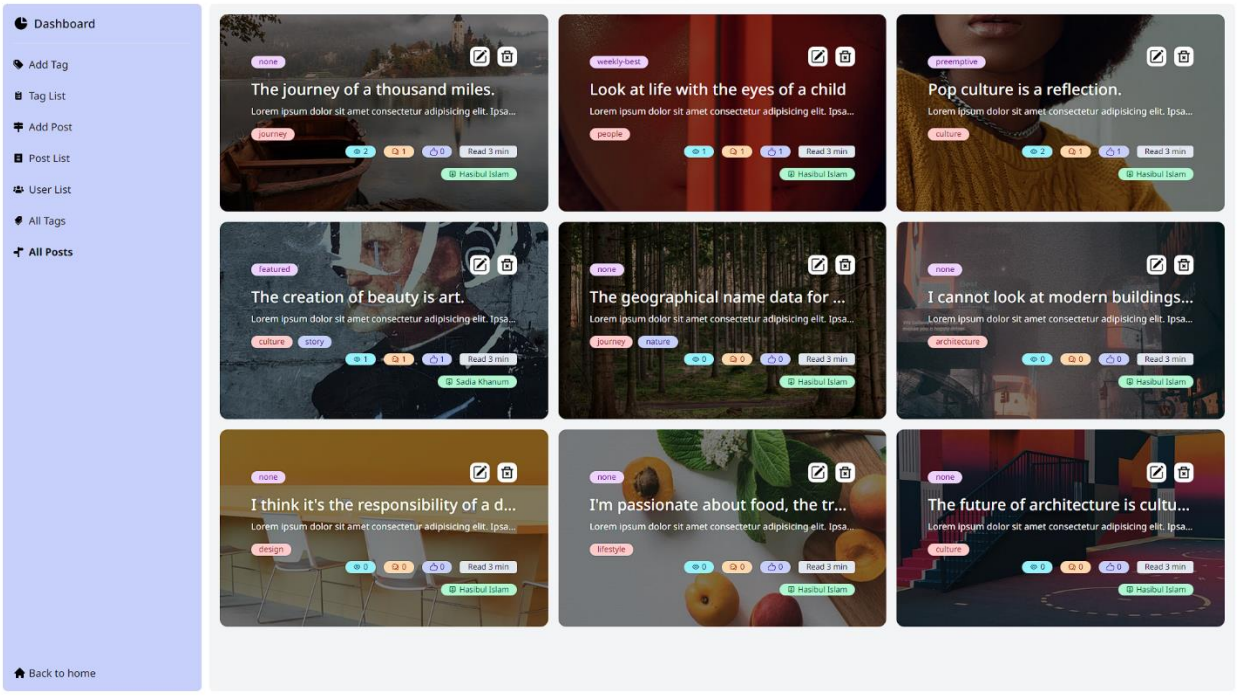
In this route, the user whether it is a “subscribed user” or an “admin” can see only his/her created tags. Where that user is able to “remove” his/her required unwanted post/posts. Throughout their requirement, they can update their tags. By the way “update” can be done by “admin” to as like as the subscribe “user”.





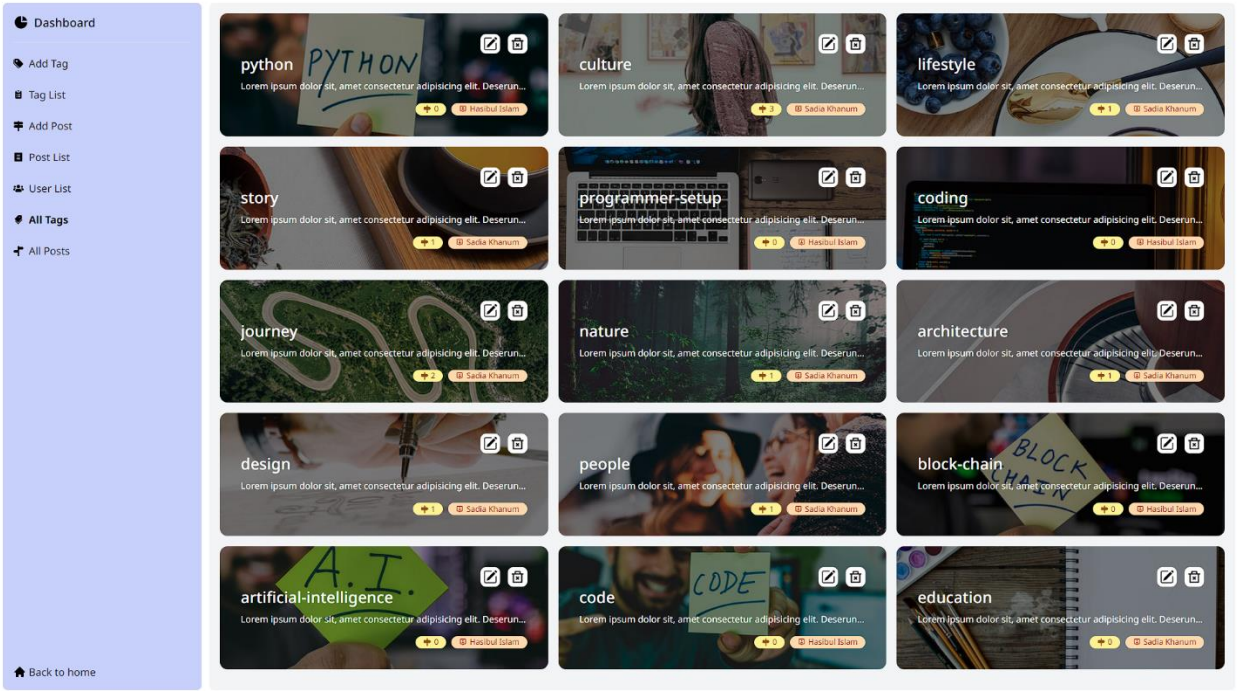
5.5.5 Dashboard – All Posts

Here can be displayed all the posts from all the users have been created. Their routes also behave as they behave from the user's account means the privilege of CRUD operation.



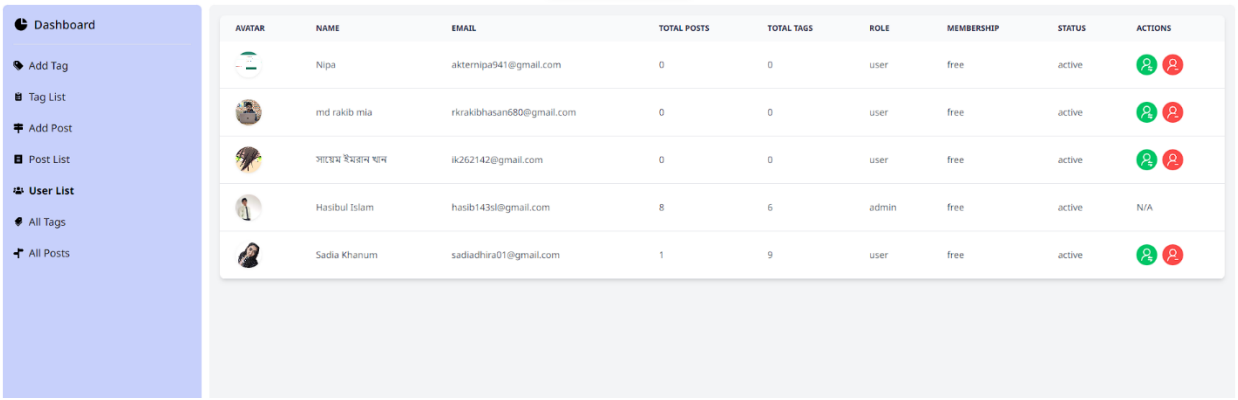
5.5.6 Dashboard – All Tags

Here can be displayed all the tags from all the users have been created. Their routes also behave as they behave from the user's account means the privilege of CRUD operation.



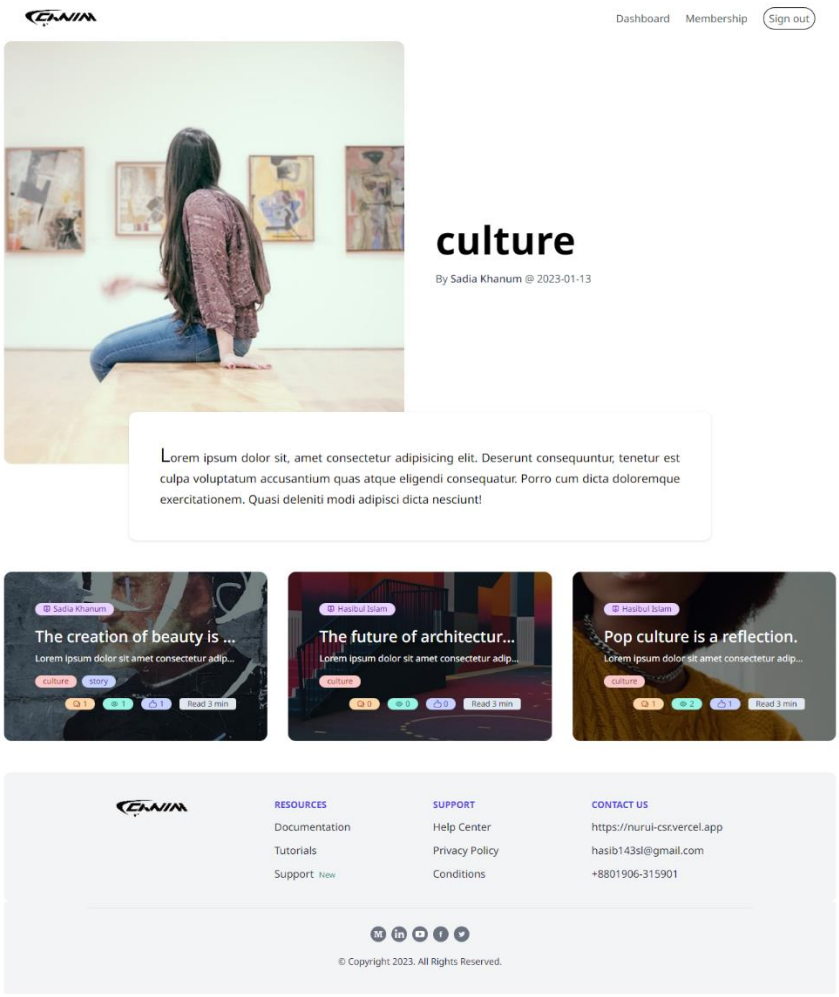
5.5.7 Dashboard – User's List

From this "admin" can control all the users like "removing" a user or make "admin" a user.



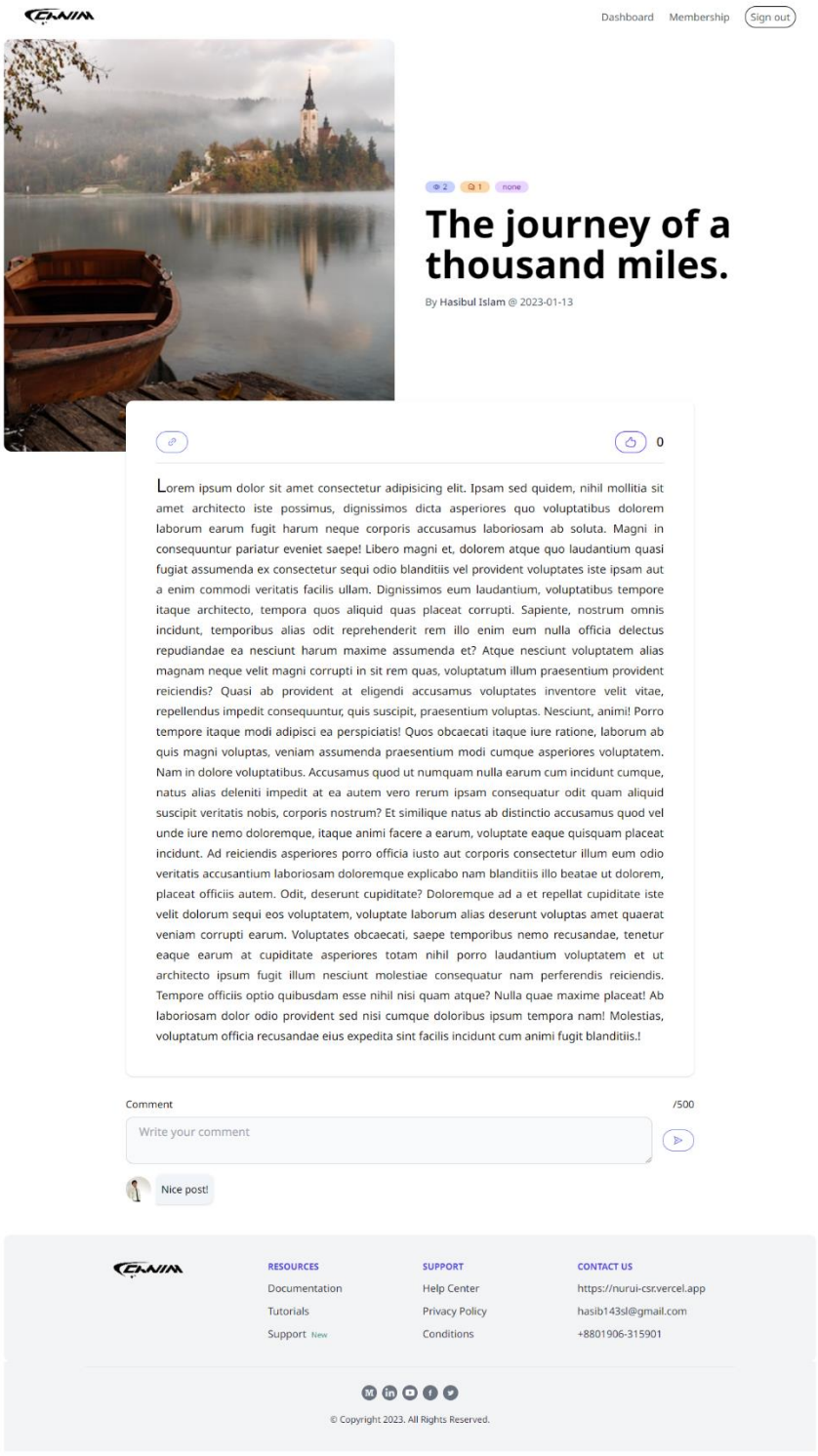
5.5.8 Tag Details Page

From this route, a user is able to see that particular tag and associate posts. Although, users can access associated posts for free but won't be able to CRUD them as a bunch of descriptions. They are able to know about the tag even to beyond to depth about the tag as well as too.



5.5.9 Tag Details Page

In this route, users can see the about the post but only signed-in users can comment here. By the way, likes and watches are also trackable for the signed user. But any kind of user can copy the link of the post so that user allow to share with other about the post.



5.6 Discussion

Our System is fully internet based. So, it requires an active internet connection. This system requires a Smartphone, Tablet, Laptop, or Desktop. To reduce falsity, we will use different types of authentication systems like JWT & Bcrypt.JS. For the payment system, we use the Stripe payment method and security.

Chapter 06 – Conclusion

6.1 Benefits

- **Subscription Based**: The user can choose a "Monthly" or a "Yearly" based subscription based on the subscription package name.
- **Full Privacy**: All of the information about users, posts, tags, payment, etc. are quite private even though the user login password is not accessed by the "admin" so, a privacy guarantee here.
- **User Friendly**: All the features are easily interactable with users based on their utility.
- **Source Code**: As this project is free to all so the source code is available on GitHub.

6.2 Conclusion

We can understand the necessity for an online blogging and Q&A system where every single person gets their required questions answered in a few. Today's modern world is technology based. So, to make advance or to cope with these make us advance the world by gaining proper knowledge.

Reference

1. **Information Security**: https://jwt.io/
2. **Password Security**: https://dcode.io/
3. **Refetching Data**: https://redux.js.org/
4. **Error Solution**: https://stackoverflow.com/

The End

Table of Contents