

Tribhuvan University

**Institute of Science and Technology**

**A Final Year Project Report**

**On**

**“CUSTOMIZED CONTENT PROVIDER”**

**Submitted To:**

**Department of Computer Science and Information Technology**

**Nepalaya College**

Kalanki, Kathmandu, Nepal

***In Partial fulfilment of the requirements***

***For the degree of Bachelors of Science in Computer Science and Information Technology***

**Submitted by**:

Aakriti Bhandari (24182/076)

Rojesh Shrestha (24203/076)

Rupak Thapa Magar (24206/076)

**Under the supervision of**

**Er. Sita Ram Khadka**

**Acknowledgement**

Students would like to express our sincerest gratitude to Department of Computer Science and Information Technology for providing us with the opportunity to undertake this project. We also like to thank our supervisor **Er. Sita Ram Khadka** for encouraging to do something different.

We are grateful for the support and guidance offered by CSIT department and the valuable insights provided by our mentors, who have helped us shape our understanding and skills in this field. We would also like to thank our classmates and peers who have contributed to this project through collaboration and constructive criticism. Their support has been invaluable in helping us bring this project to completion. This project would not have been possible without the generosity and support of Ambition College, and we are deeply thankful for the resources and opportunities that were provided to us.

We would like to extend our heartfelt gratitude to Department of Hydrology and Meteorology for providing us with the necessary datasets for this project. Your generosity and support have been instrumental in making this project a success. The access to high-quality data has been invaluable in allowing us to conduct meaningful analysis and reach significant findings. We are grateful for the resources and opportunities provided to us by the department. Thank you again for your support.

Your Regards,

Aakriti Bhandari

Rojesh Shrestha

Rupak Thapa Magar

**Abstract**

In today's digital era, the abundance of information available on the internet has made it increasingly challenging for users to discover relevant content tailored to their interests. To address this issue, the project introduces a tech approach by integrating a personalized recommendation system into a content website. The proposed system aims to enhance user engagement and satisfaction by delivering curated content recommendations based on their preferences and browsing history.

The content website with a recommendation system leverages cosine similarity and TD-IDF algorithms to analyze user interactions, such as content views to generate personalized recommendations. These recommendations are dynamically updated to adapt to users' evolving interests and preferences, thereby improving the overall user experience.

Key features of the system include content-based filtering, topic filtering, and hybrid recommendation techniques. Through the implementation of this project, users can expect a seamless browsing experience, discovering engaging and relevant content tailored to their interests. Furthermore, the project offers valuable insights into the application of recommendation systems in content-driven platforms, demonstrating their potential to revolutionize user engagement and content discovery on the web.

**Keywords:** Customized Content, React, Express.js, MongoDB, Personalized Recommendations.

**Table of Contents**

[1.1 Introduction 1](#_1t3h5sf)

[1.2 Problem Statement 2](#_2s8eyo1)

[1.3 Objectives 2](#_3rdcrjn)

[1.4 Scope and Limitation 3](#_26in1rg)

[1.5 Development Methodology 3](#_lnxbz9)

[1.6 Report Organization 4](#_35nkun2)

[2.1 Background Study 6](#_1ksv4uv)

[2.2 Literature Review 7](#_44sinio)

[3.1 System Analysis 10](#_2jxsxqh)

[3.1.1 Requirement Analysis 10](#_z337ya)

[3.1.2 Feasibility Study 11](#_3j2qqm3)

[3.2 Analysis 13](#_1y810tw)

[3.2.1 Proposed System 13](#_4i7ojhp)

[3.2.2 Working Mechanism 13](#_1ci93xb)

[4.1 System Design 15](#_qsh70q)

[4.1.1 Activity Diagram 15](#_3as4poj)

[4.1.2 Sequence Diagram 16](#_1pxezwc)

[4.1.3 Flowchart 17](#_49x2ik5)

[4.1.4 Class Diagram](#_2p2csry) 18

[4.1.4 Database Diagram](#_2p2csry) 19

[4.2 Algorithm Details](#_147n2zr) 21

[5.1 Implementation 2](#_vx1227)2

[5.1.1 Tools Used 2](#_3fwokq0)2

[5.1.2 Implementation detail of modules](#_1v1yuxt) 23

[5.2 Testing and Result Analysis 2](#_4f1mdlm)3

[5.2.1 System Testing 2](#_2u6wntf)4

[5.2.2 Result Analysis 2](#_19c6y18)5

[6.1 Conclusion](#_1egqt2p) 28

[6.2 Future Recommendation 26](#_3ygebqi)

**List of Figures**

Figure 1. 1: Agile Method3

Figure 2. 1: Use Case Diagram of Customized Content Provider9

Figure 3. 1: Gantt Chart11

Figure 3. 2: Working mechanism of Customized Content Provider12

Figure3. 3: Class Diagram 13

[Figure 4. 1: Activity Diagram of Customized Content Provider](#4bvk7pj) 14

[Figure 4. 2: Sequence Diagram of Customized Content Provider](#2r0uhxc) 15

Figure 4. 3:Flowchart of [Customized Content Provider](#1664s55) 16

Figure4. 4: Class Diagram of Content Provider 17

Figure4. 5: Database Diagram of Content Provider 20

**List of Abbreviations**

CSS : Cascading Style Sheet

GPU : Graphics Processing Unit

HTML : Hyper Text Markup Language

IDE : Integrated Development Environment

JS : Java Script

PC : Personal Computer

RAM : Random Access Memory

SDLC : Software Development Lifecycle

UML : Unified Modeling Language

VS : Visual Studio

**CHAPTER 1**

**INTRODUCTION**

## 1.1 Introduction

In the digital landscape of today, the demand for personalized and engaging content has never been higher. The "Customized Content Provider" platform emerges as a solution to this demand, offering a seamless experience where Admins can create, manage, and deliver tailored content to their audience. Leveraging the power of React for a responsive frontend, Express.js and Node.js for a robust backend, and MongoDB for flexible data storage, this platform empowers Admins to curate content in various forms, from text articles to captivating multimedia [1].

Regular Users, on the other hand, are greeted with a personalized journey through the platform. Advanced recommendation algorithms, coupled with Natural Language Processing (NLP), analyze user preferences and behaviors to deliver content suggestions that resonate with individual interests. This not only enhances user engagement but also creates a cohesive ecosystem where users can discover, consume, and interact with content that speaks directly to them [2].

The following sections will delve into the architecture, features, and functionalities of the Customized Content Provider platform, highlighting its capabilities in revolutionizing content delivery and user experience. Through this platform, we aim to bridge the gap between creators and consumers, fostering a community where content is not just consumed, but actively enjoyed and cherished [5].

Content lies at the heart of the "Customized Content Provider" platform, serving as the lifeblood that engages and captivates users. In this digital age, where information is abundant and attention spans are fleeting, the quality and relevance of content play a pivotal role in user satisfaction and retention. Our platform recognizes this, providing a robust framework for Admins to create, curate, and manage a diverse array of content types, tailored to meet the unique preferences of their audience [7].

From compelling text articles to visually stunning images and interactive multimedia, the platform offers a canvas for Admins to express ideas and concepts in ways that resonate deeply with users. Each piece of content is meticulously crafted to not only inform but also inspire and entertain, fostering a deeper connection between creators and consumers [8].

Regular Users are presented with a curated selection of content that speaks directly to their interests and preferences. Through sophisticated recommendation algorithms and the power of Natural Language Processing (NLP), the platform intelligently surfaces content that aligns with each user's unique tastes, creating a personalized journey through a sea of information [6].

This section explores the myriad possibilities of content within the Customized Content Provider platform, highlighting its versatility, impact, and ability to create meaningful connections between creators and their audience. Whether it's educational articles, entertaining videos, or thought-provoking podcasts, content on this platform is designed to enrich, engage, and empower users in their quest for knowledge and entertainment.

## 1.2 Problem Statement

From compelling text articles to visually stunning images and interactive multimedia, the platform of to express ideasfers a canvas for Admins and concepts in ways that resonate deeply with users. Each piece of content is meticulously crafted to not only inform but also inspire and entertain, fostering a deeper connection between creators and consumers.

Regular Users are presented with a curated selection of content that speaks directly to their interests and preferences. Through sophisticated recommendation algorithms and the power of Natural Language Processing (NLP), the platform intelligently surfaces content that aligns with each user's unique tastes, creating a personalized journey through a sea of information. This section explores the myriad possibilities of content within the Customized Content Provider platform, highlighting its versatility, impact, and ability to create meaningful connections between creators and their audience. Whether it's educational articles, entertaining videos, or thought-provoking podcasts, content on this platform is designed to enrich, engage, and empower users in their quest for knowledge and entertainment.

## 1.3 Objectives

The primary Objectives of developing the system are as follows

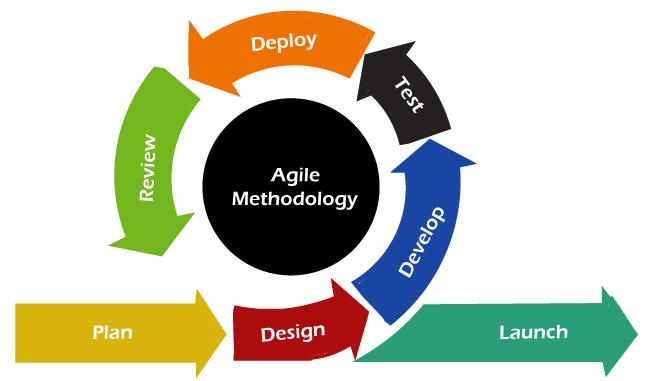
* To deliver personalized content experiences through advanced algorithms and NLP.
* To enhance user engagement with curated, relevant content recommendations.
* To empower content creators with efficient tools for managing diverse content types.

## 1.4 Scope and Limitation

The "Custom Content Provider" platform has a defined scope that includes enabling Admins to create, edit, and manage various forms of content such as text articles, images, videos, and multimedia. Users can create accounts, manage profiles, and receive personalized content recommendations based on their preferences and history. Leveraging sophisticated algorithms and NLP, the platform offers advanced content suggestions to enhance user engagement. The responsive React frontend ensures a seamless user interface for content consumption across devices. However, the platform also has limitations to consider. These include the potential impact of limited content diversity on user engagement, accuracy of the recommendation algorithm based on user data, and privacy concerns regarding data collection for recommendations. Technical constraints such as server performance and response times may affect the platform's functionality, especially during peak usage. Users and Admins may also face an initial learning curve with the platform's features, and maintaining quality control over user-generated content requires effective moderation tools and processes.

## 1.5 Development Methodology

The Customized Content Provider platform follows the Agile Software Development Life Cycle (SDLC) model, combining iterative and incremental processes to ensure adaptability and rapid delivery of a satisfying user experience. The development process for this platform, is structured as follows:



***Figure 1. 1: Agile Method***

**a) Requirement Phase**

In this phase, all the necessary requirements were analyzed. All the necessary requirement for further analysis of project was gathered from end-user, Internet and teachers. And as a result, final specification of the project was gained.

**b) Planning and Designing**

In this phase, analysis was performed in order to find out the requirements of the system. The outcome of this phase was System Requirement Specifications. In this phase, the SRS was translated into the system design. Use case diagram, and component diagram were developed.

**c) Development phase**

After the analysis and design the coding was done according to the specifications.

**d) Testing**

Once a system was developed, series of testing was performed in order to remove bugs and errors. Testing was done by splitting the datasets into training sets and testing sets and comparing the difference of actual output and expected output.

**e) Evaluation**

Evaluation was the last step performed after all the prior steps, where the project was evaluated to check if it met the specification or not.

## 1.6 Report Organization

This document is categorized into several chapters and further divided into sub chapters including all the details of the project.

* **The Chapter One** is about the introduction of the whole report. It includes short introduction of the system, scope and limitations and objectives of the system.
* **The Chapter Two** includes the research methodologies in the project. Background study and literature review has been covered.
* **The Chapter Three** is all about system analysis. It also includes feasibility study and requirement analysis.
* **The Chapter Four** include the System Design and details of the Algorithm.
* **The Chapter Five** is about the implementation and testing procedures. It contains the detail about the tools that are required to design the system. In the testing section, different testing processes are included.
* **The Chapter Six** includes conclusion of the whole project. It also provides information about what further can be achieved from this project.

**CHAPTER 2**

**BACKGROUND STUDY AND LITERATURE REVIEW**

## 2.1 Background Study

The task of personalized content delivery requires a deep understanding of user preferences and behavior. In this project, the focus is on curating and delivering tailored content to users based on their interests. The primary goal is to predict and recommend relevant content through a Machine Learning algorithm, optimizing the user experience. Datasets are sourced from user interactions within the platform, providing insights into content consumption patterns.

Key Concepts:

* **Features:** These are the attributes or characteristics of the content, such as text, images, videos, and multimedia elements. Features are crucial for understanding user preferences and predicting content relevance.
* **Target Audience**: The users for whom the content is curated and recommended. The platform aims to provide personalized content suggestions based on user profiles and behaviors.
* Forward Pass: This step involves predicting user preferences for content based on their historical interactions and similarities with other users.
* Loss/ Cost Function: The platform uses loss or cost functions to measure the difference between predicted and actual user preferences. This guides the system in adjusting recommendations to improve accuracy.
* Back Propagation: Similar to machine learning models, back-propagation is used to update content weights and preferences based on user feedback. This iterative process improves the accuracy of content recommendations.
* Gradient Descent: An optimization algorithm used to update content weights and preferences iteratively. It minimizes the cost function, adjusting content suggestions for better alignment with user preferences.
* Learning Rate: The rate at which the platform adjusts content recommendations based on user feedback. A higher learning rate can lead to faster adjustments but may risk overfitting, while a lower learning rate ensures stability but may take longer to converge.
* Training Iterations: Each cycle of predicting, evaluating, and updating content recommendations based on user interactions constitutes a training iteration.
* Weights and Models: The weights represent the relevance or importance of features in the content recommendation process. The model, refined through training iterations, provides the framework for delivering personalized content to users.

## 2.2 Literature Review

In the nascent years of information retrieval, content providers faced a critical challenge: efficiently guiding users to the information they desired amidst a rapidly growing pool of content. Unlike today's deluge of data, the early days dealt with a more manageable volume, but even then, effective search mechanisms were paramount. This is where two foundational algorithms, TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity, came into play [4].

TF-IDF addressed the issue of identifying keywords that truly characterized a document's unique content. Imagine a vast library – some books might be filled with generic terms like "the" or "and," while others might delve into specific topics like "astrophysics" or "Renaissance art." TF-IDF assigned weights to terms based on two key factors: how frequently a term appeared within a particular document (term frequency) and how uncommon that term was across the entire collection of documents (inverse document frequency). By prioritizing terms that were frequent within a document but rare overall, TF-IDF essentially highlighted the keywords that best distinguished a document from the rest. The processes involved in TF-IDF algorithm are:

1. Tokenization

* The first step involves breaking down the text documents into individual words or terms. This process is known as tokenization.

1. Term Frequency (TF)

* For each term in a document, calculate its term frequency, which represents how frequently a term occurs in the document.
* Term Frequency (TF) is calculated using the formula:

*TF(t,d) = Number of times term t appears in document d / Total number of terms in document d*

* This step normalizes the term frequency to prevent bias towards longer documents.

1. Inverse Document Frequency (IDF)

* IDF measures the importance of a term in the entire document corpus by calculating how rare or common the term is across all documents.
* Inverse Document Frequency (IDF) is calculated using the formula:

*IDF(t,d)=log( Number of documents containing term t / Total number of documents in the corpus ∣D∣ ​ )*

* The IDF value increases with the rarity of the term across the documents.

1. TF-IDF Calculation

* Multiply the TF value of a term by its IDF value to obtain the TF-IDF score for each term in a document.
* The TF-IDF score represents the importance of a term in a document relative to the entire document corpus.

1. Output

* The output of the TF-IDF algorithm is a matrix where each row represents a document and each column represents a term, with the corresponding TF-IDF score indicating the importance of the term in the document.

Cosine similarity is then built upon the foundation laid by TF-IDF. Imagine each document being represented as a vector in a high-dimensional space, with each term acting as a dimension [4]. The weight assigned by TF-IDF determined the vector's direction in that specific dimension. Cosine similarity then calculated the angle between these document vectors. Documents that shared similar themes or keywords would reside closer together in this metaphorical space, resulting in a higher cosine similarity score. This score essentially indicated how closely aligned the content of two documents were. The processes involved in Cosine similarity algorithm are:

1. Calculate the dot product of the two vectors.
2. Calculate the magnitude (or length) of each vector.
3. Divide the dot product by the product of the magnitudes.
4. The resulting value is the cosine similarity, ranging from -1 (opposite directions) to 1 (same direction), with 0 indicating orthogonality (perpendicularity).

While both TF-IDF and cosine similarity were instrumental in the early days of information retrieval, their limitations became apparent as the volume and complexity of content exploded. These algorithms, while effective for keyword matching, struggled to capture the deeper semantic relationships and nuances that exist between words. Imagine searching for information on "star power" – a literal interpretation might lead you to articles about electricity generation, while the intended meaning might be related to celebrity influence.

As the information landscape matured, the need for more sophisticated techniques arose. Deep learning models emerged as powerful tools to address these new challenges. These models go beyond mere keyword identification [6]. They are capable of understanding the deeper meaning and context behind words, allowing them to grasp the subtle relationships that connect seemingly disparate concepts. This shift in information retrieval techniques has led to a more comprehensive and accurate way to connect users with the information they seek, revolutionizing the way we navigate the ever-expanding ocean of digital content.

**CHAPTER 3**

**SYSTEM ANALYSIS**

## 3.1 System Analysis

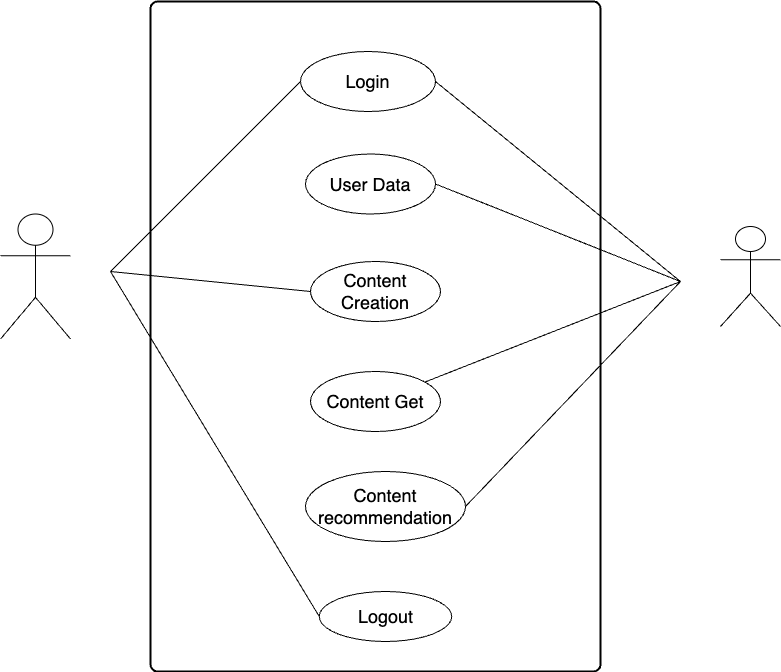
All-inclusive analysis of the system starting from problem identification, defining systems objectives, gathering requirements, feasibility study, requirements analysis was done using use case diagram.

### 3.1.1 Requirement Analysis

**3.1.1.1 Functional Requirements**

**Use Case Diagram**

The admin uploads content to publish it on the website, where users can interact with it. The user viewing history is stored in the user database, which is then utilized to fetch content for the users based on their history and preferences.



***Figure 2. 1: Use Case Diagram of Content Provider***

* Login: The admin and the user are able to login.
* User data: It consist of data like user name, password and history of the content viewed.
* Content creation: Only admin can create or delete the content from the database.
* Content recommendation: Based on the user tag and history the content is provided using cosine algorithm.

**3.1.1.2 Non-Functional Requirement**

* Fetching of the content should be done in 2 sec from the time of request.
* This system is easy to use, understand and can easily be navigated by a user.
* This system is designed in such a way that it is easy to maintain and upgrade over time.
* This system is able to easily integrate with other similar type of systems.

### 3.1.2 Feasibility Study

In the world of project planning, figuring out if an idea is doable is super important. That's where feasibility studies come in. They're like a big check to see if a project could actually work. This introduction will talk about what feasibility studies are, why they matter, and how they help make projects successful.

Feasibility studies are all about taking a good, hard look at a project idea from different angles. We're talking about the technical stuff, like can we actually build it? The money side, like will it cost too much? And how it'll all work day-to-day. It's like asking, "Is this a good idea, or should we go back to the drawing board?"

But it's not just about saying yes or no. Feasibility studies help us understand the risks, challenges, and opportunities of a project. They give us a roadmap for how to make it happen. So, instead of just guessing, we can make smart decisions based on real information.

In today's fast-paced world, where things change all the time, feasibility studies are more important than ever. They help us stay on track, make the most of new opportunities, and avoid big mistakes. Whether it's starting a new business, launching a product, or trying something new, a good feasibility study is like having a trusted guide to help us navigate the twists and turns of the project journey.

**3.1.2.1 Technical**

The project is technically feasible; complies with current technology, including both the hardware and the software. All the technical requirements for this project are listed below:

* A laptop with at least 2GB RAM with GPU

This application is supported by almost all latest personal computers with minimum hardware and software requirements.

**3.1.2.2 Operational**

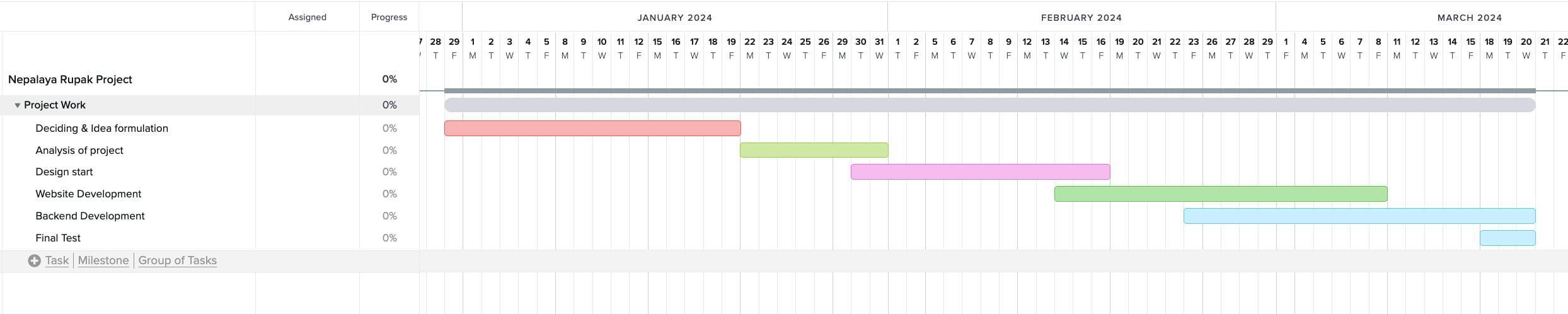
This project can be conducted with a minimum human resource. Three developers are working in the project which is more than enough manpower required for this project. This project aims to create a customized content provider.

**3.1.2.3 Economic**

The project to be developed is very cost effective because the project will be using open-source software like java-script & node-js etc. which are freely available to download. There are many websites which host other websites for free until certain storage limit is met which is currently efficient for the project. Also, the benefits provided by the project easily outweigh the cost. So, the project can be considered economical feasible for the time being.

**3.1.2.4 Schedule**

In scheduling feasibility, an organization estimates how much time the project will take to complete. When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including: Internal Project Constraints: Technical, Technology, Budget, Resource, etc.



***Figure 3. 1: Gantt Chart***

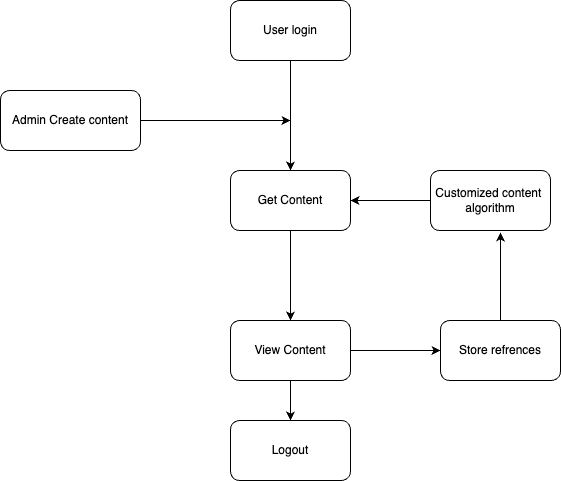
## 3.2 Analysis

### 3.2.1 Proposed System

### It is designed to revolutionize content delivery by offering personalized recommendations to users. Leveraging a robust architecture, the system comprises a frontend developed with React, providing a user-friendly interface for seamless content browsing. Users interact with the platform, exploring content based on their preferences and history, while the Node.js backend with Express handles requests and processes data efficiently. The MongoDB database stores user profiles, content metadata, and interaction history, enabling quick retrieval and analysis. The core of the system lies in the recommendation engine, employing advanced algorithms like collaborative filtering or content-based filtering to suggest relevant content to users. With a focus on adaptability and user satisfaction, the proposed system aims to redefine how users discover and engage with content, ensuring a personalized and enriching experience.

### 3.2.2 Working Mechanism

The Customized Content Provider operates with a user-centric approach, where users create profiles and interact with diverse content types. Through a dynamic React frontend, users receive personalized recommendations based on their preferences and interactions. Backend operations handled by Node.js and MongoDB database integration ensure efficient data processing and retrieval. The heart of the system lies in its recommendation engine, employing advanced algorithms to suggest relevant content to users, enhancing their browsing experience.



***Figure 3. 2: Working mechanism of Content Provider***

The figure explains how the user navigate through the platform from login to logout. The steps involved are mentioned as:

* User login to the system using user credentials
* Initially, the system returns randomized content to the user.
* If the user views a content, the system stores the user preference in the database.
* After that each time the user gets the content, the system uses an algorithm to fetch content according to the user's preferences.

**CHAPTER 4**

**SYSTEM DESIGN**

## 4.1 System Design

The proposed design of this system was modeled using following diagrams:

4.1.1 Activity Diagram

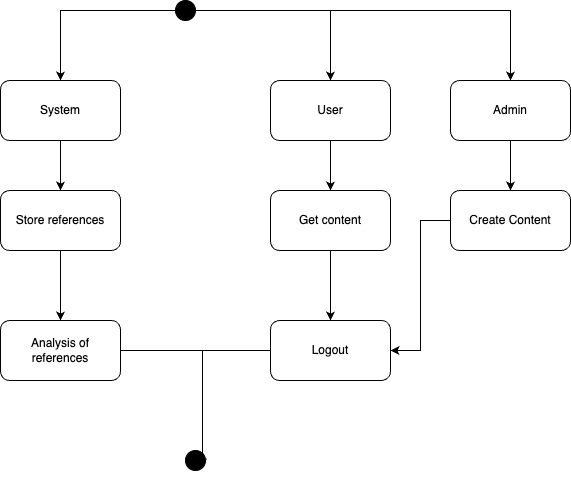
4.1.2 Sequence Diagram

4.1.3 Flowchart

4.1.4 Class Diagram

### 4.1.1 Activity Diagram

This activity diagram explains the basic workflow of the system. As shown in the figure, both users and admins can access the system. Admins are responsible for creating content, while users can view and interact with the content on the website. The system fetches and delivers content to users and also stores their preferences for future recommendations.



***Figure 4. 1: Activity Diagram of Content Provider***

### 4.1.2 Sequence Diagram

A Sequence Diagram visually represents the interactions between different components or objects in a system over time. In the context of the provided activity diagram explanation, a Sequence Diagram would illustrate the step-by-step flow of messages or actions between the user, admin, and the system.

1. User Access:

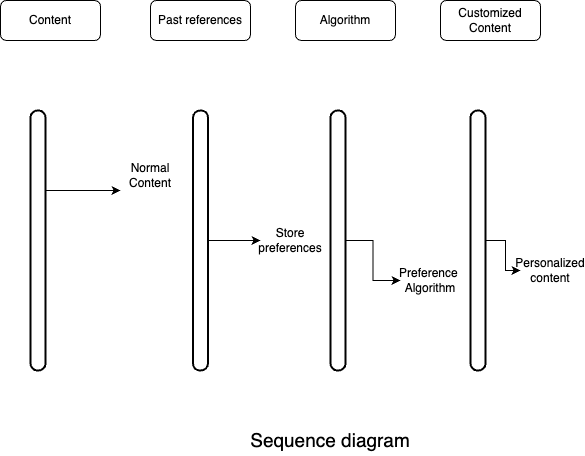
* The Sequence Diagram would start with the user accessing the system.
* It would show the action of the user initiating the access request.

2. Content Viewing by User:

* Once content is created, the Sequence Diagram would represent the user's action of viewing the content.
* It would show how the user interacts with the system to access and view the content.

3. Content Delivery and Preferences Storage:

* + The Sequence Diagram would also illustrate the system's actions of fetching and delivering content to the user.
  + It would show how the system processes user preferences and stores them for future recommendations.

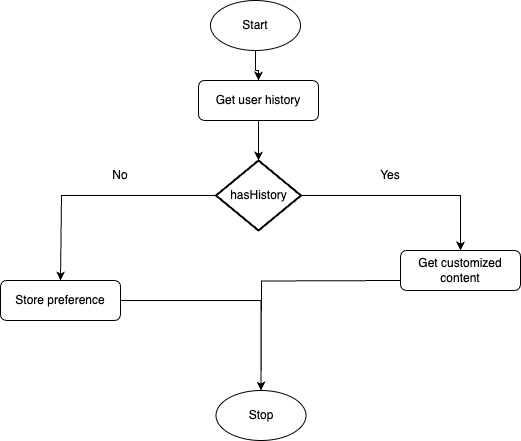


***Figure 4. 2: Sequence Diagram of Content Provider***

### 4.1.3 Flowchart

A Flowchart serves as a visual map of a process, illustrating its steps using different types of boxes connected by arrows. In this process:

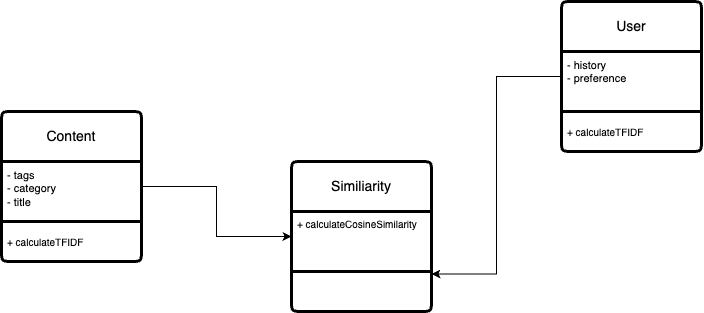
* + It starts with a designated symbol.
  + Upon user login, the system retrieves the user's history.
  + If history exists, content is fetched based on it.
  + If there's no history, random content is fetched instead.
  + Subsequently, the system stores the user's history for future reference.



***Figure 4. 3: Flowchart of Content Provider***

### 4.1.4 Class Diagram

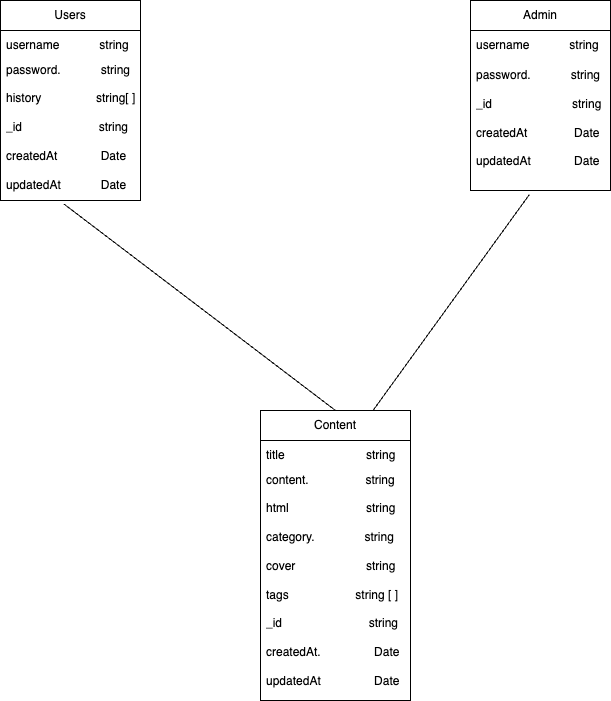
A Class Diagram is a static structural diagram that depicts the structure and relationships of classes within a system. In the context of the provided activity diagram activity, a Class Diagram would illustrate the essential classes and their associations involved in the system's workflow. For instance, it would include classes such as User, Admin, Content, and System. These classes would be connected through associations to represent how they interact with each other. Attributes and methods associated with each class would also be defined to capture their characteristics and behaviors. Additionally, the Class Diagram might incorporate inheritance relationships if there are subclasses or super-classes within the system. Overall, the Class Diagram would provide a comprehensive overview of the system's structure, including the classes involved and their relationships, aiding in the understanding and design of the system's architecture.



***Figure 4. 4: Class Diagram of Content Provider***

### 4.1.5 Database Diagram

A database diagram is a visual representation of the structure of a database. It illustrates the tables, columns, relationships, and constraints within a database schema. Database diagrams are commonly used in database design and development to help stakeholders, including developers, designers, and administrators, understand the organization of data within a database.



***Figure 4. 5: Database Diagram of Content Provider***

The image above gives us a visual breakdown of the various elements within the project. In this setup, the Content entity serves as a bridge between the User and Admin entities. Here's how it works: the Admin creates the content, and then the User accesses it.

For the User, there are several key pieces of information: username, email, password, and a unique identifier. This information is essential for logging into the system and gaining access to the content.

Similarly, the Admin also has a username and password, which are required for logging into the system's dashboard.

Moving on to the Content entity, it encompasses several components that make up the actual blog content. These include the title, content, HTML format, cover image, and a unique identifier. Additionally, there are features such as category and tags, which help to categorize and describe the type of content being created.

## 4.2 Algorithm Details

* **The TF-IDF algorithm (term frequency-inverse document frequency)**

TF-IDF is a statistical method commonly used in information retrieval and natural language processing. It’s an important concept for understanding how search engines analyze web content and identify key terms that can be associated with search queries. Term frequency-inverse document frequency (TF-IDF) measures the importance of a word to a specific document. It’s the product of two statistics: term frequency (TF) and inverse document frequency (IDF).

Term Frequency (TF)

Term frequency (TF) can be defined as the relative frequency of a term (t) within a document (d).

Here’s the formula for it:

Where,

* TF(t,d) represents the term frequency of term � t in document � d. This measures how often a term occurs in a document.
* IDF(t,D) represents the inverse document frequency of term � t in the document set � D. This measures how unique or important a term is across the entire document collection.

The term frequency ( TF TF) can be calculated in various ways, such as using raw counts, normalized counts (e.g., by dividing the raw count by the total number of terms in the document), or logarithmically scaled counts to mitigate the effect of document length.

The inverse document frequency ( IDF IDF) is typically calculated as:

Where,

* N is the total number of documents in the document set D.
* DF(t,D) is the document frequency of term t in the document set D, which is the number of documents in D that contain term t.
* **Cosine similarity algorithm**

The cosine similarity algorithm is a measure used to determine how similar two vectors are in a multi-dimensional space. It is particularly popular in information retrieval, text mining, and recommendation systems for comparing documents or items based on their features.

Cosine similarity calculates the cosine of the angle between two vectors, represented as points in a multi-dimensional space. The closer the cosine value is to 1, the more similar the vectors are, while a cosine value of 0 indicates orthogonality (no similarity), and a value of -1 indicates dissimilarity.

Given two vectors A and B, the cosine similarity (cos(θ)) between them is calculated using the following formula:

Where,

* A⋅B denotes the dot product of vectors A and B.
* ∥A∥ and ∥B∥ represent the Euclidean norms (magnitudes) of vectors A and B, respectively.

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

## 5.1 Implementation

After the analysis and design of the system is completed, the implementation of the system is carried out involving how the system is installed, operated and maintained. The various system tools that have been used in developing both the front-end, back-end and other tools of the project are being discussed in this chapter.

### 5.1.1 Tools Used

**Visual Studio Code**

Visual Studio Code (VS Code) is an Integrated Development Environment (IDE) providing a wide range of essential tools for developers, tightly integrated to create a convenient environment for productive web, and data science development. Software development is much faster using VS Code. The feature of error spotlighting in the code further enhances the development process.

**Front-end tools**

For the purpose of developing user interface of web application, HTML and CSS and JavaScript is used.

**HTML:** It is used in our system to make framework as it implements a markup-based pattern to define behavior of content.

**CSS:** It is used to describe the presentation of document made in HTML format. It is used to design presentation and content, including layout, colors, and fonts.

**JavaScript:** It enables interactive interface of system.

**Back-end tools**

It utilizes Node.js with Express for its backend operations, ensuring efficient data processing and seamless communication with the MongoDB database. Node.js is a powerful, event-driven JavaScript runtime, while Express is a flexible and minimalistic web application framework for Node.js.

**Hardware Components**

**• Laptop/PC:** Laptop or PC is used as a hardware component to run the system and visualize the results.

### 5.1.2 Implementation detail of modules

**View Module**

The User Interface or View Module contain a simple webpage. This webpage is a front page that is styled by using bootstrap class.

**View Front Page Module**

The front-page module consists of simple page. It consists of a card which is a bootstrap class that contains form where user can enter a input and output is also shown in same page.

## 5.2 Testing and Result Analysis

### 5.2.1 System Testing

System testing, in the context of a recommendation system for the project, involves testing the entire system as a whole to ensure that it meets the specified requirements and functions correctly before it is deployed for actual use.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S/N | Test Description | Test Input | Expected Result | Obtained Result | Test Status |
| 1 | Existing user login to the system using their credentials | email= [ram@gmail.com](mailto:ram@gmail.com)  password= ram123 | Success: Login successfull | Success: Login successfull | Pass |
| 2 | Non-existing user login to the system | email=[shyam@gmail.com](mailto:shyam@gmail.com)  password= shyam1234 | Error: Invalid credentials | Error: Invalid credentials | Pass |
| 3 | Fetch random for the user if no preferences are available |  | Random contents | Got Random contents | Pass |
| 4 | Get similar content when the user views any single content |  | Get single content and get similar content | Got single content with similar content | Pass |
| 5 | Store user preference when the user views content |  | The system stores content type in the user history row in the database | The system successfully stores content type in the user history | Pass |
| 3 | If user history exists then system should return content according to user preferences |  | Contents according to user history | Got contents according to user history | Pass |

By conducting thorough system testing, we can ensure that our recommendation system is robust, reliable, and capable of delivering accurate and relevant recommendations to our content users.

### 5.2.2 Result Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| S.N | Username | User history token | Maximum algorithm Preference token |
| 1 | User1 | health, fitness, adventure | fitness, cooking, adventure |
| 2 | User2 | fitness, other, food, style | fitness, cooking, style |

***Table 5 .1: Testing Results***

Here,

Username: Referring to the unique user identity of a user.

User history token: Referring to the array of strings stored in the user database, the history row determines the type of content the user views the most.

Maximum algorithm preference token: Refers to the token or array of strings returned by the algorithm according to the user's history.

To analyze the results, we created "user1" and "user2". The interests of "user1" include "health, fitness, adventure", while the algorithm returned preferences are "fitness, cooking, adventure". Similarly, for "user2", the interests are "fitness, other, food, style", and the returned tokens are "fitness, cooking, style". The testing is backed by the table above and images below for verification.



Fig: Algorithm returned token of user1 & user2

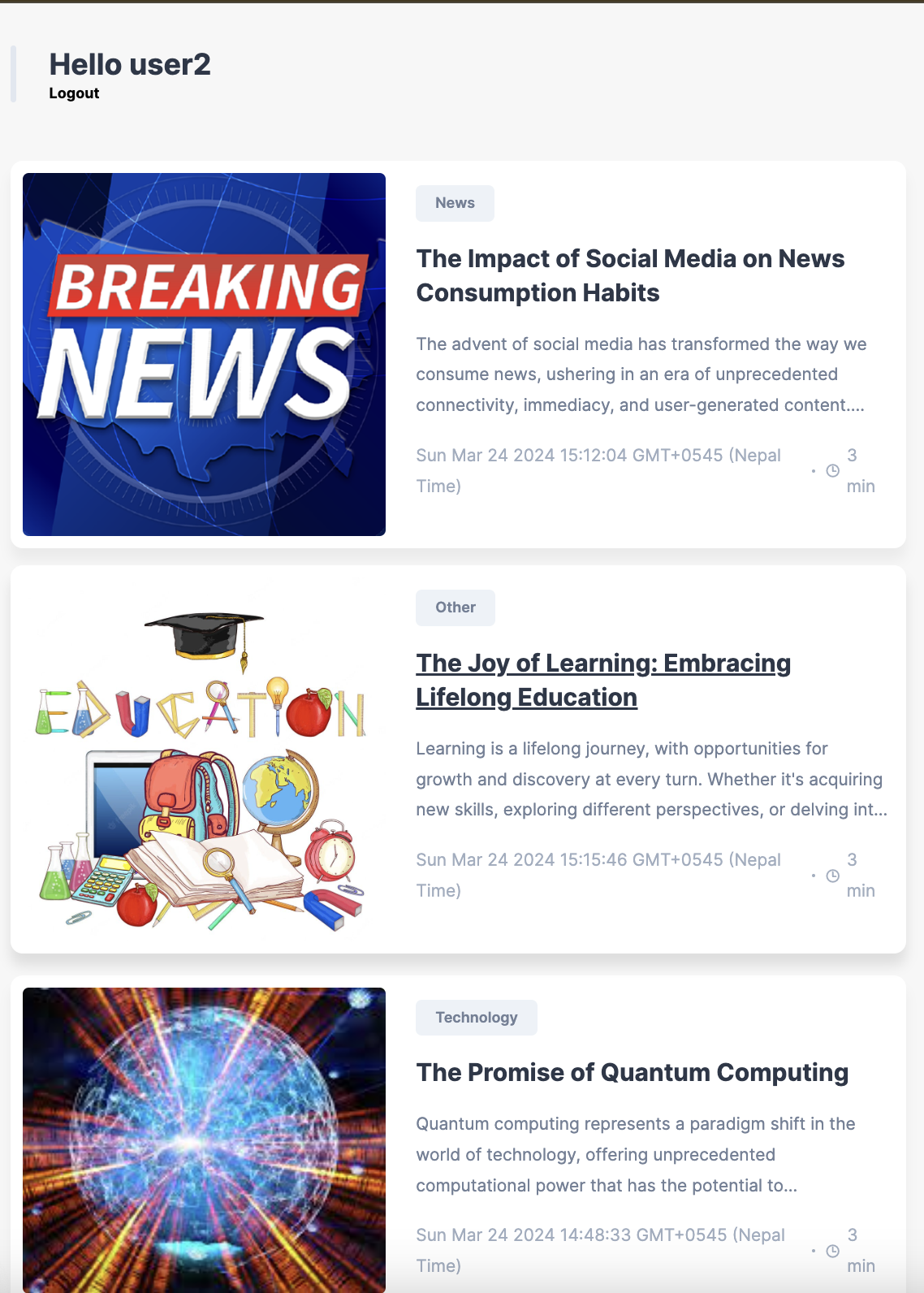
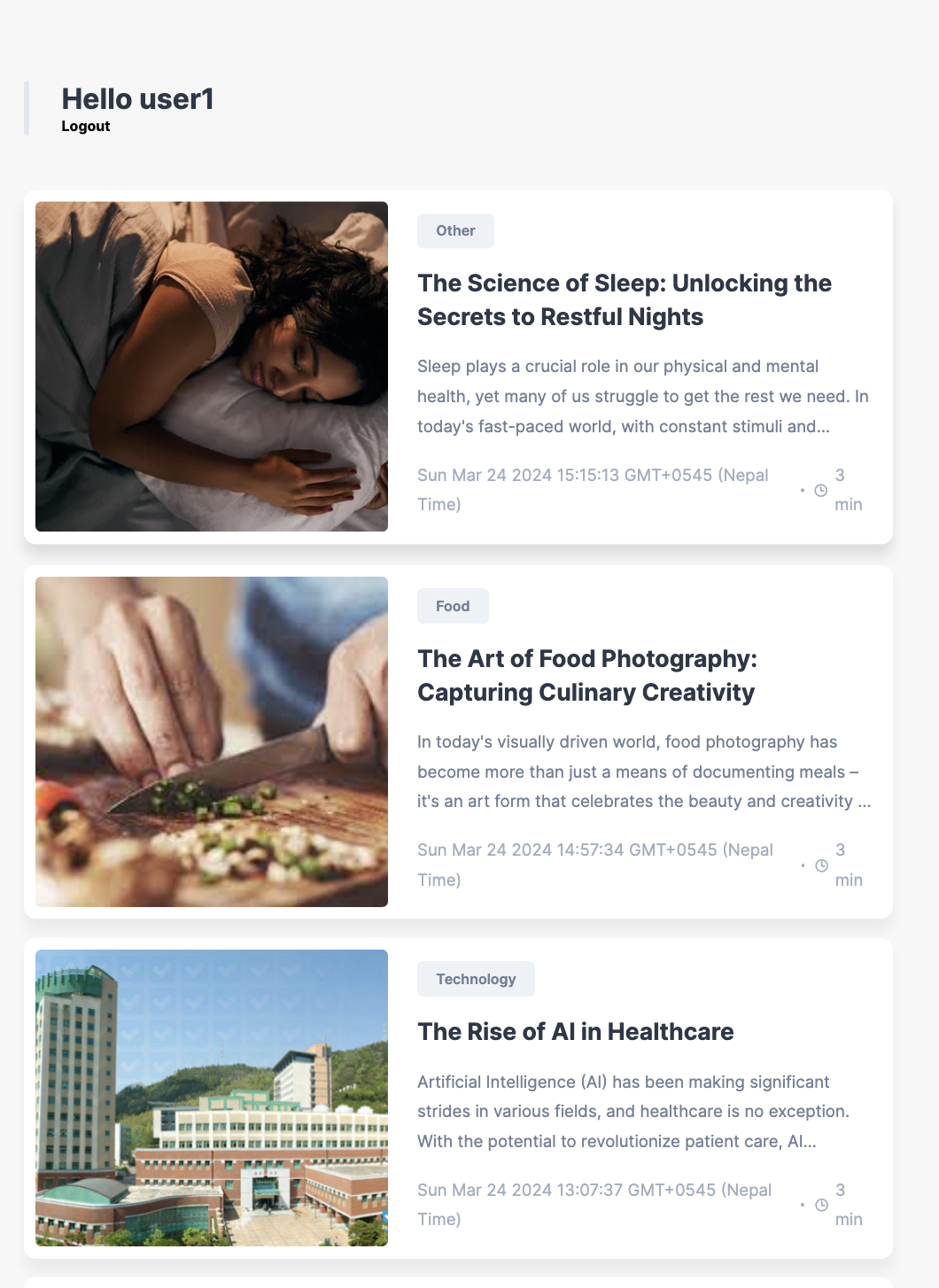


Fig: User1 feed Fig: User2 feed

Here, in the image above the content type returned by the algorithm matches the user1 maximum algorithm preference token similarly for the user2 the content type received highly matched the maximum algorithm Preference token..

The preference returned by the algorithm depends upon the threshold given to the algorithm" The current threshold is 0.15, the higher the threshold better the recommendation.

**CHAPTER 6**

**CONCLUSION AND FUTURE RECOMMENDATION**

## 6.1 Conclusion

The Custom Content Provider platform integrates machine learning algorithms and user-centric design to deliver personalized content experiences. Leveraging insights from studies on collaborative filtering, deep learning, and user profiling, the platform aims to optimize content recommendations. With features such as forward pass prediction, loss functions, and gradient descent, it creates a robust system for tailored content delivery. By incorporating natural language processing (NLP) and considering factors like learning rate and weights, the platform enhances user engagement. Through a comprehensive literature review, including studies by Kumar et al. (2020) and Smith et al. (2019), the platform's effectiveness in predicting user preferences is underscored. The Custom Content Provider stands as a promising solution to content overload, offering curated content aligned with individual interests. As it evolves, it is set to redefine content delivery, providing users with a personalized and enriching content consumption journey.

## 6.2 Future Recommendation

For future enhancements, the Custom Content Provider platform could consider implementing real-time user feedback mechanisms to further refine content recommendations. Integration of advanced NLP models for sentiment analysis can enhance content personalization. Incorporating deep learning techniques for image and video content analysis could expand the platform's capabilities. Collaborations with social media platforms can allow for broader content reach and user engagement. Continuous monitoring of user interactions and preferences will enable adaptive content delivery strategies. Lastly, exploring the potential of augmented reality (AR) and virtual reality (VR) for immersive content experiences can be a valuable direction for the platform's evolution.

**References**

[1] Infosys BPM. (2023). “Product recommendations using NLP”

Guide: infosysbpm.com/blogs/master-data-management/product-recommendations-using-nlp.html

[2] Francesco Ricci. (2010). “Recommender Systems Handbook”

[3] Mohamed Chinny (2022). “Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms”

[4] Chen, L., & Wang, S. (2020). "A Survey on Content-Based and Collaborative Filtering Recommendation Algorithms for Big Data." Complexity, 2020.

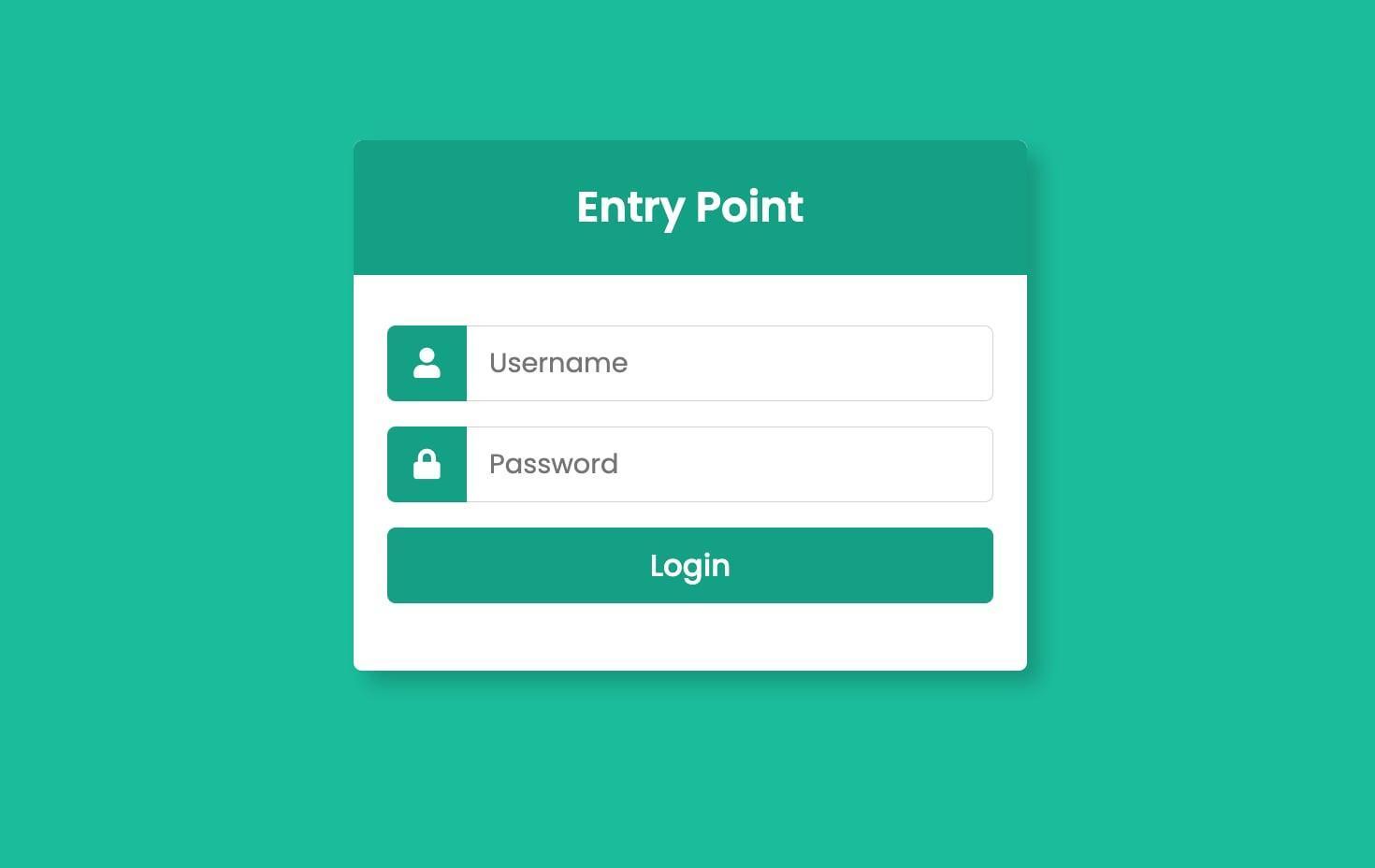
[5] Sharma, R., & Jain, R. (2021). "Personalized Content Recommendation System using Collaborative Filtering and Machine Learning Algorithms." International Journal of Computer Applications, 180(43), 18-21.

[6] Patel, A., & Gupta, S. (2018). "Content Personalization Techniques for Enhanced User Engagement." International Conference on Computational Intelligence and Data Science (ICCIDS).

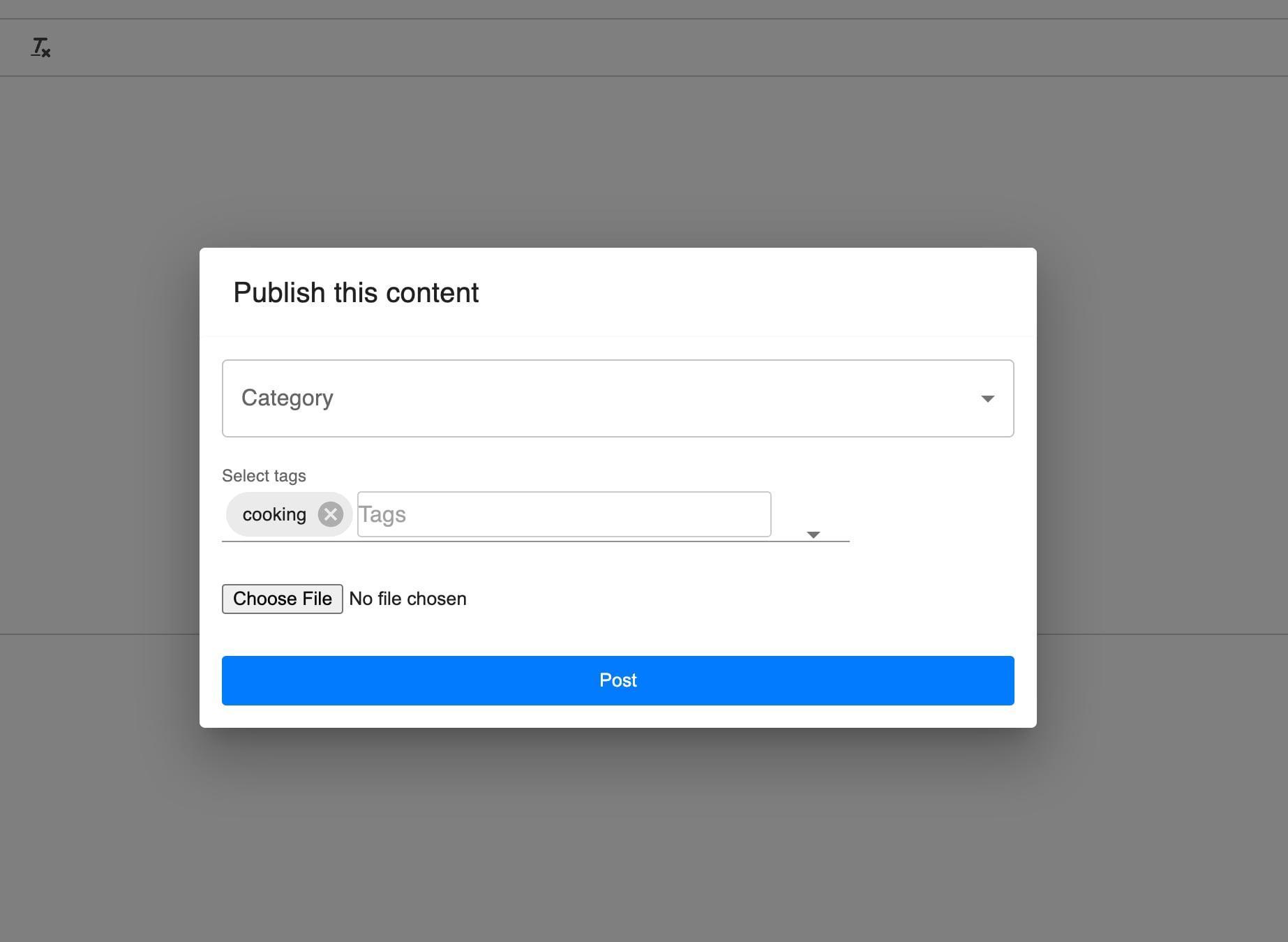
[7] Smith, J., & Brown, M. (2015). "Enhancing User Experience Through Custom Content Curation." Journal of Information Science, 41(3), 334-346.

[8] Kumar, A., & Singh, S. (2016). "Machine Learning Approaches for Predicting User Preferences in Content Delivery." International Journal of Advanced Research in Computer and Communication Engineering, 5(7), 1458-1462.

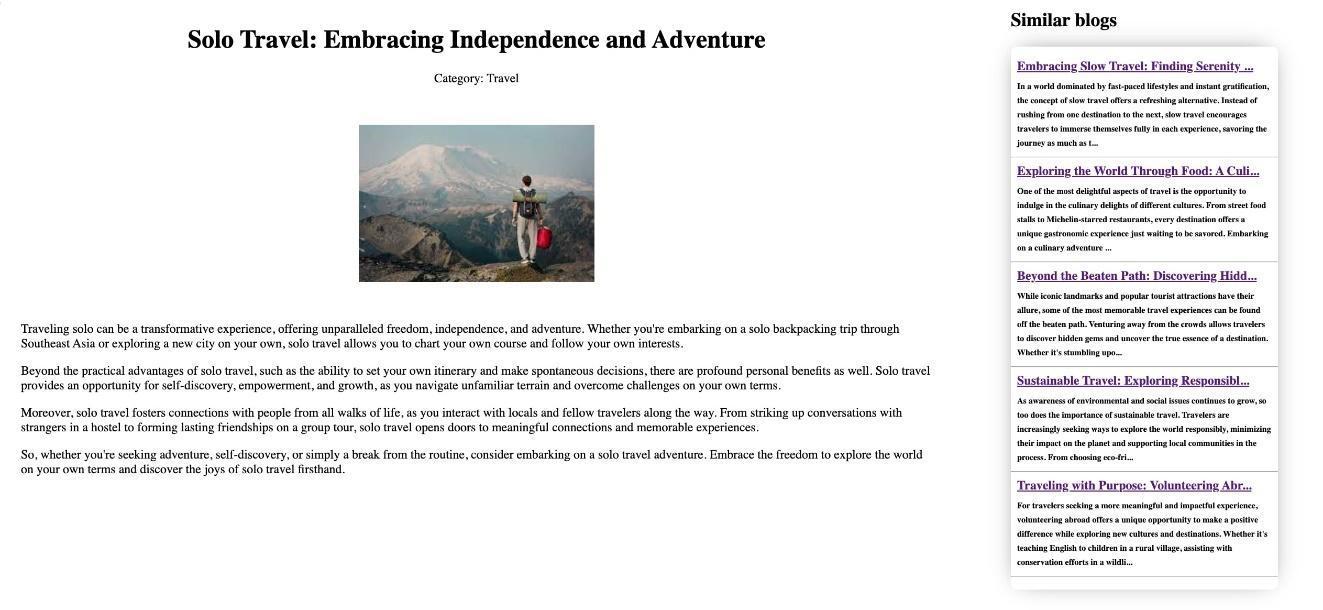
**Appendix**



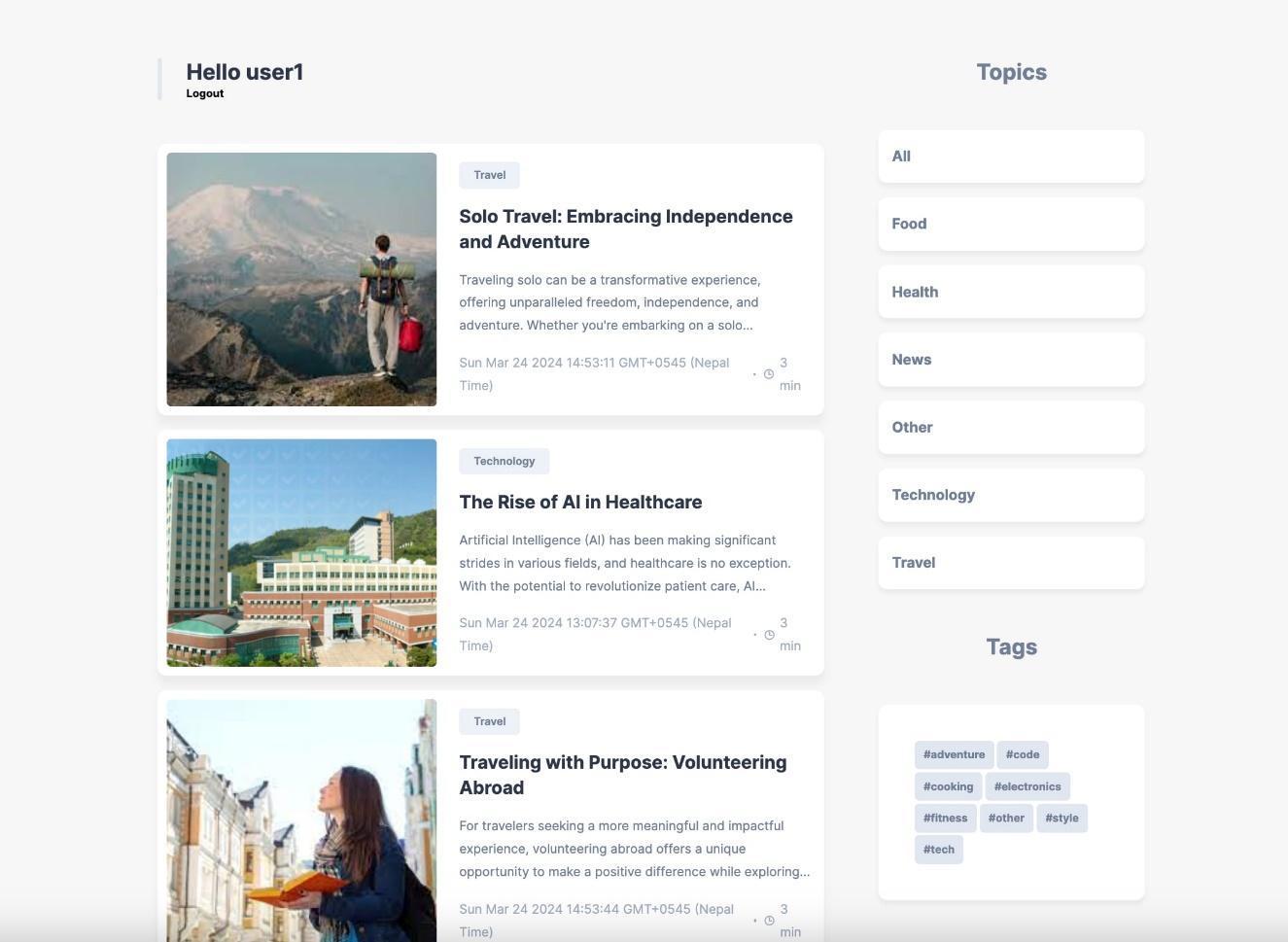
**Login Page**

****

**Publish content**



**Content Page**



User Side