SUNBEAM INFOTECH PRIVATE LIMITED

A

MAJOR PROJECT REPORT

ON

“THE RECYCLE-X”

PG-DMC

MOBILE COMPUTING

SUBMITTED BY

BHOSALE DHANASHRI SAMBHAJI

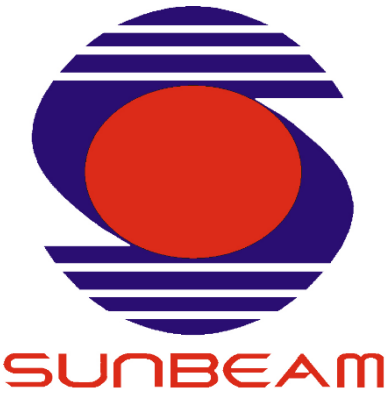
Under the Guidance of

Dr. Mrs. Manjusha Nikam

Year

2024-2025

CERTIFICATE



This is to certify that the project report entitled

"THE RECYCLE_X"

Submitted To

SUNBEAM INFOTECH PRIVATE LIMITED, CENTRE FOR DEVLOPMENT OF ADVANCE COMPUTING, PUNE.

has been completed under my guidance and supervision. To the best of my knowledge and belief, the matter presented in this project report is original and has not been submitted elsewhere for any other purpose.

Submitted by

BHOSALE DHANASHRI SAMBHAJI

Project Guide

&

Program Co-Ordinator

Dr. Mr. Yogesh Kolhe

Co-Ordinator (DMC), Sunbeam, CDAC

External Examiner(s) **Sign**

1. …… ……

2. …… ……

ACKNOWLEGEMENT

It is our foremost duty to express our deep sense of gratitude and respect to the guide **Dr. Mr. Yogesh Kolhe** for his uplifting tendency and inspiring us for taking up this project work successful.

We are also grateful to **Dr. Mr. Yogesh Kolhe** (Co-Ordinator of Mobile Computing) for providing all necessary facilities to carry out the project work and whose encouraging part has been a perpetual source of information.

We are highly indebted to **Prof. (Dr.) Nilesh Ghule** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We also thank all staff members of our department for their timely help and encouragement, which help us in completing of our project work.

We are indebted to the library personnel's for offering all the help in completing the project work. Last but not only the least we are thankful to our colleagues and those helped us directly or indirectly throughout this project work.

ABSTRACT

Introducing "Recycle X," an innovative mobile and web-based application dedicated to revolutionizing waste management through structured digital solutions. This comprehensive system bridges the gap between suppliers, consumers, and administrators, facilitating efficient waste collection, processing, and resale of recycled materials. Built using React, React Native, Node.js, Express.js, and MySQL, Recycle X ensures a seamless user experience across multiple platforms.

The Recycle X platform encompasses three primary roles:

- **Admin:** Manages system operations, monitors transactions, and ensures compliance.
- **Consumer:** Engages with waste management services, schedules pickups, and receives authentication through email OTP verification using the Node mailer library.
- **Supplier:** Collects and processes dry waste, metals, and other recyclable materials before supplying them for further processing.

The project follows the MVC architecture for backend development, ensuring maintainability and scalability. Additionally, file logging mechanisms have been implemented to track system activities efficiently. For secure authentication, MySQL is used with indexing and stored procedures to optimize database performance.

Key features of Recycle X include:

- **Multi-platform access:** React Native for mobile applications and React.js for web applications.
- **Secure consumer authentication:** Email OTP verification via Node mailer.
- **Expo for mobile application deployment and testing.**
- **Efficient waste processing and categorization.**
- **Sustainable waste management through a structured digital workflow.**

By integrating advanced technologies with sustainable practices, Recycle X aims to enhance waste management operations, promote environmental consciousness, and streamline interactions between stakeholders. Through its user-friendly interface and robust functionalities, Recycle X envisions a cleaner and more efficient waste recycling ecosystem.

Keywords: Recycle X, Waste Management, Sustainable Practices, Digital Platform, Consumer Authentication.

INDEX

1.2 Overview

In today's world, effective waste management has become a crucial challenge due to the increasing amount of waste generated by households, industries, and commercial establishments. Traditional waste management systems often struggle with inefficiencies, lack of proper segregation, and minimal incentives for recycling. Addressing these concerns, **Recycle X** emerges as an innovative waste management solution aimed at revolutionizing sustainable environmental practices through a comprehensive digital platform.

Recycle X is designed as a cutting-edge mobile and web application that seamlessly connects **consumers, suppliers, and administrators** within a unified ecosystem. The project's core objective is to optimize waste collection, processing, and repurposing through modern technological interventions. By leveraging digital platforms, automation, and secure authentication, Recycle X aims to bring efficiency, transparency, and ease to the waste recycling process.

This platform is equipped with features that streamline operations, including real-time tracking of waste collection, structured waste segregation methods, efficient supplier-consumer interactions, and an intuitive product-selling interface for recycled materials. The application integrates innovative solutions such as **automated OTP-based authentication using Node mailer**, secure transaction handling, and user activity logging to enhance overall system security and performance. Furthermore, **MySQL is utilized for structured data storage, implementing indexing and stored procedures** to improve database efficiency.

By providing a structured and efficient approach to waste management, **Recycle X** is committed to promoting sustainability, raising awareness about recycling, and reducing environmental impact. The project's digital transformation of waste handling serves as a benchmark for future waste management innovations, ultimately contributing to a cleaner and more environmentally conscious society.

1.2 Choice of Topic with Reasoning/Need for the Project

The Growing Waste Management Crisis

The rapid growth of urbanization, industrialization, and population has led to a significant increase in waste production. Current waste disposal methods are inefficient,

contributing to serious environmental and health hazards. The need for a more **structured, technology-driven waste management system** has never been more pressing.

Key challenges in existing waste management systems include:

Lack of structured waste collection and processing mechanisms.

Inefficient recycling practices leading to increased landfill waste.

Limited awareness and participation in sustainable waste management initiatives.

Absence of a digitalized, real-time tracking system for waste collection.

Unstructured supplier-consumer interaction in the recycling industry.

Minimal incentives for businesses and individuals to participate in recycling programs.

Recycle X addresses these challenges by providing an **integrated, smart waste management solution** that enhances the recycling process, optimizes waste collection, and fosters a collaborative environment among consumers, suppliers, and administrators.

1.2.2 Need for a Technological Solution

The role of technology in waste management cannot be overlooked. The use of **mobile applications, cloud computing, machine learning, and IoT-based tracking** can significantly improve the efficiency of waste collection and processing. Recycle X leverages technology to offer:

Real-time waste tracking and scheduling: GPS-enabled monitoring for better logistics.

Secure authentication: OTP verification using Nodemailer for user registration and transactions.

Automated waste segregation: Advanced database techniques like indexing and stored procedures for efficient data handling.

User-friendly mobile and web interfaces: Providing seamless access to waste collection, tracking, and processing functionalities.

By incorporating these modern technological solutions, **Recycle X** enhances accessibility, security, and operational efficiency in waste management, making it a vital tool for **sustainable environmental practices**.

1.3 Problem Statement

The existing waste management system lacks efficiency, transparency, and structured recycling mechanisms. **Recycle X** aims to develop an integrated mobile and web application that facilitates efficient waste management through a seamless interaction platform for consumers, suppliers, and administrators. The primary goal is to enhance recycling processes, optimize waste collection, and promote environmental sustainability through digital transformation.

Key aspects addressed by Recycle X:

- A structured approach to waste collection and tracking.
- An intuitive digital marketplace for recycled products.
- Secure user authentication and real-time data logging.
- A technology-driven system for efficient waste processing.
- A collaborative platform encouraging active participation in recycling initiatives.

1.4 Objectives

The **Recycle X** project is designed to achieve the following key objectives:

- **Develop a user-friendly mobile and web application** for waste management.
- **Enable an efficient waste collection and processing system** for consumers and suppliers.
- **Implement a secure authentication mechanism** (OTP-based verification using Nodemailer).
- **Introduce real-time waste tracking and logistics optimization.**
- **Promote environmental awareness** and encourage sustainable recycling practices.
- **Facilitate seamless communication and transactions** between stakeholders.
- **Optimize database performance** using MySQL indexing and stored procedures.
- **Ensure a scalable and secure architecture** to support future expansions.
- **Enhance user experience** with intuitive UI/UX design principles.

By achieving these objectives, **Recycle X** sets a benchmark for an efficient and sustainable waste management system, transforming the way waste is handled at both individual and industrial levels.

2. LITERATURE REVIEW AND STUDY

2.1 Waste Management Technologies

Waste management has been a critical global challenge, with researchers and technologists continuously exploring new solutions to improve efficiency, sustainability, and environmental impact. Several studies have examined the evolution of waste management techniques, analyzing existing waste disposal methods, recycling strategies, and the implementation of modern technologies in the field.

Analysis of Existing Waste Management Platforms

Numerous waste management platforms have emerged worldwide, aiming to enhance the efficiency of waste collection, sorting, and processing. Traditional waste management methods often suffer from inefficiencies due to manual sorting, improper disposal techniques, and inadequate tracking mechanisms. Several case studies highlight the effectiveness of digital platforms in improving waste management outcomes. For instance, studies on smart waste management systems indicate that integrating technology into the process leads to optimized collection routes, reduced costs, and higher recycling rates.

Review of Technological Interventions in Recycling Processes

Modern recycling processes leverage innovative technologies such as automated sorting, advanced material recovery, and bio-waste conversion techniques. Research highlights the adoption of mechanical and chemical recycling, where AI-driven sorting mechanisms significantly improve material recovery efficiency. Several pilot projects have demonstrated the benefits of robotics in sorting facilities, reducing contamination in recycling streams and increasing processing efficiency.

Study of User Engagement Strategies in Environmental Initiatives

User participation is a crucial aspect of successful waste management initiatives. Studies suggest that mobile applications and gamification techniques significantly increase public involvement in waste segregation and recycling programs. Research also highlights the role of government policies and incentive-based models in encouraging citizens to adopt

sustainable waste disposal practices. Many studies emphasize the need for awareness campaigns and educational initiatives to foster community participation.

Exploration of AI and IoT Applications in Waste Sorting and Management

Artificial Intelligence (AI) and the Internet of Things (IoT) have revolutionized waste management by providing real-time data analytics and automation. AI-powered waste recognition systems enable efficient sorting of recyclable materials, reducing human intervention and errors. IoT-based smart bins equipped with sensors help monitor waste levels, optimize collection schedules, and minimize overflow issues. Case studies from smart cities worldwide demonstrate the impact of AI and IoT in improving waste collection efficiency and reducing operational costs.

2.2 Digital Platform Considerations

The development of digital platforms for waste management involves several critical considerations, including system architecture, user experience, security mechanisms, and performance optimization. A comprehensive review of these aspects helps in designing an efficient and scalable solution.

Examination of Mobile and Web Application Architectures

Modern waste management applications rely on robust and scalable architectures to handle multiple user roles, data processing, and real-time interactions. Research indicates that microservices-based architectures offer significant advantages in terms of scalability and maintainability. The choice between monolithic and distributed architectures has been widely studied, with findings suggesting that a cloud-native approach ensures better performance and reliability.

User Experience Design in Service-Oriented Platforms

User experience (UX) plays a crucial role in the adoption and effectiveness of digital waste management platforms. Studies emphasize the importance of intuitive design, accessibility features, and streamlined workflows in enhancing user engagement. Research in human-computer interaction (HCI) highlights the role of visual elements, notifications, and

personalized recommendations in encouraging users to participate actively in waste management initiatives.

Security and Authentication Mechanisms in Multi-User Systems

With increasing concerns about data privacy and security, implementing robust authentication mechanisms is essential for digital waste management platforms. Literature on cybersecurity highlights best practices, such as multi-factor authentication (MFA), role-based access control (RBAC), and end-to-end encryption. Case studies on security breaches in digital waste management systems underscore the need for proactive security measures to protect user data and prevent unauthorized access.

Performance Optimization Techniques for Large-Scale Applications

Scalability and performance optimization are critical for handling large volumes of data in real-time waste management applications. Research on cloud computing and distributed systems suggests that load balancing, caching mechanisms, and asynchronous processing significantly enhance system performance. Several studies explore the impact of edge computing in waste management, where processing data closer to the source reduces latency and improves real-time decision-making capabilities.

2.3 Global Waste Management Policies and Trends

A review of international waste management policies provides valuable insights into the regulatory frameworks and best practices adopted across different regions. Various studies compare waste management regulations in developed and developing countries, highlighting the effectiveness of policy-driven initiatives in improving recycling rates and reducing landfill dependency.

Government Policies and Regulations

Governments worldwide have implemented various policies to promote sustainable waste management. Research on extended producer responsibility (EPR) programs demonstrates their effectiveness in reducing packaging waste and encouraging manufacturers to adopt eco-

friendly materials. Studies also explore the impact of landfill taxes, plastic bans, and incentives for circular economy initiatives on waste reduction.

Sustainable Development Goals (SDGs) and Waste Management

The United Nations' Sustainable Development Goals (SDGs) emphasize the importance of responsible consumption and production patterns. Literature on SDG implementation highlights the role of technology in achieving waste management targets. Several reports examine how digital platforms contribute to SDG 12 (Responsible Consumption and Production) by promoting recycling, reducing waste generation, and improving resource efficiency.

Emerging Trends in Circular Economy and Zero-Waste Strategies

The concept of a circular economy has gained traction in recent years, with researchers exploring innovative ways to minimize waste and maximize resource recovery. Studies highlight the success of zero-waste initiatives in cities worldwide, where community-driven approaches and policy interventions lead to significant waste reduction. Research on material recovery facilities (MRFs) and closed-loop recycling systems demonstrates the economic and environmental benefits of transitioning towards a circular economy.

2.4 Case Studies and Success Stories

Smart Waste Management in Singapore

Singapore has implemented an advanced waste management system leveraging AI, IoT, and data analytics. Research on the country's initiatives highlights the effectiveness of automated waste collection, smart bins, and digital tracking systems in achieving high recycling rates and reducing landfill waste.

Sweden's Waste-to-Energy Model

Sweden is known for its successful waste-to-energy (WTE) initiatives, where non-recyclable waste is converted into energy for heating and electricity. Case studies demonstrate the

environmental and economic benefits of WTE plants, showcasing Sweden's leadership in sustainable waste management.

Community-Driven Recycling Programs in India

Several grassroots initiatives in India focus on community participation in waste segregation and recycling. Studies on informal waste collection systems highlight the role of waste pickers and micro-entrepreneurs in managing urban waste effectively. Digital platforms connecting waste generators with recyclers have shown promising results in improving recycling rates and reducing landfill waste.

3. SOFTWARE REQUIREMENT SPECIFICATION

3.1 Software Requirements

3.1.1 Front-End Frameworks

Mobile Application:

- React Native: Enables cross-platform mobile development with a native-like experience.
- Expo Framework: Facilitates the development and deployment of React Native applications.

Web Application:

- React.js: Provides a component-based architecture for building user interfaces.
- Next.js: Enhances React applications with server-side rendering for improved SEO and performance.
- Tailwind CSS: Utility-first CSS framework for responsive and scalable UI design.

3.1.2 User Interface Design

- Adherence to responsive design principles for optimal usability across devices.
- Mobile-first approach to prioritize user experience on smaller screens.
- Intuitive user experience (UX) ensuring seamless navigation and interaction.
- Support for dark mode and accessibility features, including ARIA attributes for enhanced inclusivity.
- Component-based UI architecture using Chakra UI for consistency and efficiency.

3.1.3 Backend and Database

Backend Framework:

- Node.js: Lightweight and scalable JavaScript runtime for server-side development.
- Express.js: Minimalist backend framework facilitating RESTful API development.
- MVC Architecture: Ensures separation of concerns for maintainability and scalability.
- RESTful API Design: Provides a structured and standardized way to handle client-server communication.

Database:

- MySQL: Relational database management system (RDBMS) optimized for structured data storage.
- Indexing and Stored Procedures: Improve query performance and maintain data integrity.
- Normalization: Reduces redundancy and optimizes data organization for efficiency.

Authentication & Security:

- Nodemailer: Facilitates OTP-based authentication for secure user verification.
- Secure User Registration & Login: Enforces data encryption and validation mechanisms.
- JWT (JSON Web Token): Implements token-based authentication for session management.
- Role-Based Access Control (RBAC): Restricts access based on user roles and permissions.
- CORS Implementation: Ensures secure cross-origin resource sharing.

3.1.4 Additional Technologies

- File Logging: Backend tracking mechanism for debugging and auditing.
- WebSockets: Enables real-time communication for live updates and notifications.
- Cloud Storage: Securely manages documents and media files.
- Payment Gateway Integration: Supports online transactions using platforms like Stripe or PayPal.
- AI-Powered Recommendation Engine: Personalizes user experiences based on data analytics.
- Push Notification Service: Enhances user engagement with timely alerts.
- Geolocation Services: Enables real-time tracking and mapping functionalities.
- SMS & Email Notifications: Implements communication mechanisms for user updates.

3.2 System Architecture

User Roles:

- **Consumer:** Requests waste collection, views history, and interacts with the platform.
- **Supplier:** Collects and supplies waste materials for processing.
- **Administrator:** Manages users, processes, and system analytics.

Key Features:

- Waste collection scheduling with automated reminders.
- Real-time tracking for waste collection and processing.
- Waste processing management system for streamlined operations.
- Integrated product selling interface for recycled goods.
- Push notifications and alerts for important updates.
- Comprehensive analytics and reporting dashboard.
- Multi-language support for global accessibility.
- User feedback and review system for quality assurance.

3.3 Hardware Requirements

Mobile Devices:

- Minimum Android version: 8.0 (Oreo) or higher.

- Minimum 4GB RAM for smooth app performance.

Web Platform:

- Modern browsers with full HTML5 and JavaScript support.
- Minimum screen resolution: 1366x768 for optimal UI experience.

Server Requirements:

- Minimum 4-core processor for handling concurrent requests.
- 16GB RAM to support multiple user sessions effectively.
- 512GB SSD storage for fast data access and reliability.
- Cloud-based hosting solutions (AWS, Firebase, or DigitalOcean) for scalability and availability.
- Load balancer and caching mechanisms (Redis) for optimized performance.

This document outlines a robust software requirement specification ensuring scalability, security, and usability in the waste management platform.

3 SYSTEM DESIGN

4.1 System Architecture

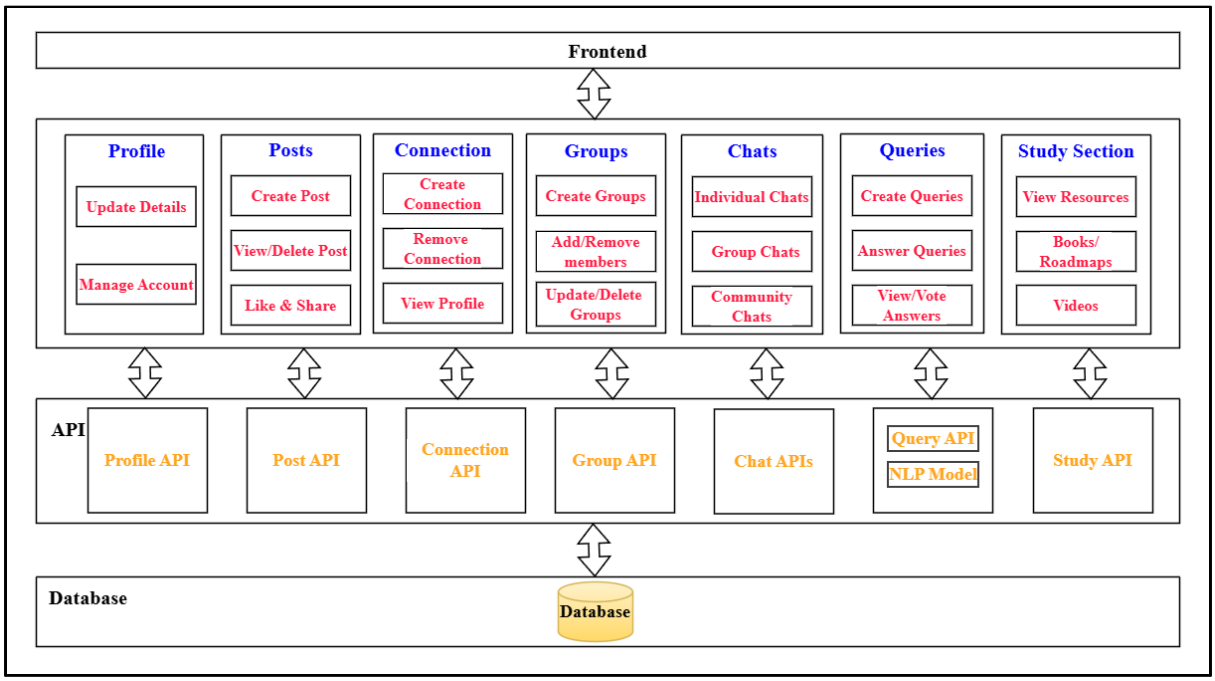


Fig: 4.1: System Block Diagram

Modules:

- **Study Section:**

The Study Section module is designed to offer college students a comprehensive platform for their academic needs. This module serves as a centralized resource hub, providing students with easy access to a wealth of study materials, exam papers, research papers, and other relevant educational resources.

One of the primary advantages of this module is its user-friendly interface, ensuring that students can quickly find the resources they require for their studies. With a robust MySQL database at its core, the module efficiently stores and manages a diverse range of educational materials.

To access these resources, students interact with a backend API built in Node.js. When students use the Next.js and Tailwind CSS frontend, API calls are made to retrieve the

requested data. This seamless integration allows for rapid and convenient resource retrieval.

- **Query Section:**

The Query Section module provides students with a platform for seeking answers and sharing knowledge. Within this module, students can pose questions to their peers across the platform. If a satisfactory answer is provided, the query is resolved. However, if the answer remains elusive, students have the option to turn to an AI-powered solution integrated into the platform, possibly utilizing prebuilt AI tools.

Behind the scenes, the Next.js and Tailwind CSS frontend interacts with a Node.js backend through API calls, ensuring a smooth user experience. Furthermore, students are encouraged to actively contribute their expertise, fostering a collaborative environment that enriches the collective knowledge of the platform's user base. This module enhances students' learning experience by providing reliable access to information and promoting peer-to-peer interaction.

- **Posts Section:**

The Posts Section module allows students to share their achievements, which can serve as a source of inspiration for fellow students. Students can post about their internships, job opportunities, or any noteworthy accomplishments. When a student accesses the website's Next.js and Tailwind CSS frontend, it interacts with a Node.js backend through API calls to retrieve these inspiring stories.

This module not only provides a platform for celebrating successes but also encourages students to contribute their achievements, fostering a positive and motivating atmosphere within the community. By showcasing the real-world opportunities and accomplishments of their peers, it empowers students to aim high and work towards their own goals.

- **Groups Section:**

  The Groups Section module empowers students to create and join groups, fostering collaborative learning environments for both academic and extracurricular activities. Within this module, students can form study groups, clubs, or communities of interest. Whether it's group projects, study sessions, or shared hobbies, this section provides a platform for students to connect and collaborate.

  To access this feature, students utilize the user-friendly Next.js and Tailwind CSS frontend, which communicates seamlessly with the Node.js backend through API calls. This ensures a smooth and responsive experience for users as they create, discover, and engage with groups.

  Additionally, students are encouraged to actively contribute by initiating new groups or participating in existing ones. By facilitating connections and knowledge-sharing, this module enhances the overall learning experience and sense of community among students. It promotes teamwork, shared interests, and mutual support, making it a valuable asset within the platform.

- **Connection Section**

  The Connection Section module is designed to facilitate meaningful connections among students, allowing them to share knowledge and expand their network with like-minded peers. Within this module, students can connect with others, fostering a community where knowledge and experiences can be freely shared. It serves as a platform for students to engage in discussions, seek advice, or simply connect with individuals who share their interests.

To access this feature, students use the intuitive Next.js and Tailwind CSS frontend, which seamlessly communicates with the Node.js backend through API calls. This ensures a user-friendly and responsive experience for students as they establish and nurture connections.

4.2 Methodology/Algorithm

Problem: Traditional search engines rely on keyword matching and ranking algorithms to retrieve relevant documents. This approach can be effective for simple queries, but it often falls short for more complex or nuanced queries. Large language models (LLMs) can be used to improve the semantic understanding of search queries and documents, leading to more accurate and relevant retrieval results. However, integrating LLMs with Elasticsearch can be challenging, as it requires careful consideration of the different components and their interactions.

Proposed Solution:

The following is a proposed solution for integrating LLMs with Elasticsearch in a comprehensive and effective manner:

3.1.2 Tokenization and Text Preprocessing:

The first step is to tokenize and preprocess the input text, both for the user query and the Elasticsearch documents. This involves splitting the text into individual tokens, removing punctuation and special characters, and converting the text to lowercase.

3.1.3 Semantic Understanding:

Next, we need to obtain semantic representations of the user query and the Elasticsearch documents. This can be done using a pre-trained LLM, such as BERT. The LLM will produce contextualized embeddings for each token, which capture the meaning of the token in the context of the surrounding text.

3.1.4 Integration with Elasticsearch:

To integrate the LLM with Elasticsearch, we can use a scoring function that combines the contextualized embeddings of the query and the documents. For example, we can use the cosine similarity between the two embeddings to measure their relevance.

3.1.5 Query Processing Pipeline:

The following is a high-level overview of the query processing pipeline:

- Tokenize and preprocess the user query.
- Obtain the contextualized embedding of the user query using the LLM.
- Retrieve relevant documents from Elasticsearch using the scoring function.
- Rank the documents in descending order of scores.
- Return the ranked documents to the user.
- Ranking and Retrieval

3.1.6 User Feedback Loop:

It is important to incorporate user feedback into the system to improve its performance over time. This can be done by collecting user feedback on the search results and using it to update the LLM model and the ranking strategies.

3.1.7 Natural Language Generation (Optional):

The system can also generate human-friendly responses to user queries using a pre-trained NLG model. This can be useful for providing summaries of the search results or answering complex questions in a comprehensive and informative way.

3.1.8 Performance Monitoring and Optimization:

It is important to monitor the performance of the system and identify areas for improvement. This can be done by tracking metrics such as precision, recall, and F1 score.

Formulas:

- **BM25 score:**

$$BM25(Q, D) = \sum_{i} IDF(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{\text{avgdl}})} \quad [1]$$

where:

- Q is the query
- D is the document
- q_i is a query term
- f(q_i, D) is the term frequency of q_i in D
- IDF(q_i) is the inverse document frequency of q_i
- k_1 and b are hyperparameters that control the importance of term frequency and document length in the ranking score
- **Cosine similarity:**

$$\text{cosine\_similarity}(E(Q), E(D)) = \frac{E(Q) \cdot E(D)}{\|E(Q)\| \cdot \|E(D)\|} \quad [1]$$

where:

E(Q) is the embedding of the query

E(D) is the embedding of the document

Improvements over traditional search systems:

The enhanced LLM offers several improvements over traditional search systems:

- It can better understand the semantic meaning of queries and documents, leading to more relevant search results.
- It can generate human-friendly responses, such as summaries and answers to questions, making it easier for users to find the information they need.
- It can be integrated with Elasticsearch to provide a scalable and efficient search solution.

Example:

Consider the following query:

What is the capital of France?

A traditional search system might return a list of documents that contain the words "capital" and "France". However, the enhanced LLM would be able to understand that the user is asking a question and would generate the following response:

The capital of France is Paris.

The enhanced LLM can also be used to generate summaries of retrieved documents. For example, if the user clicks on one of the documents returned in the search results, the

enhanced LLM can generate a summary of the document that highlights the most important information.

4.3 Data Flow Diagram

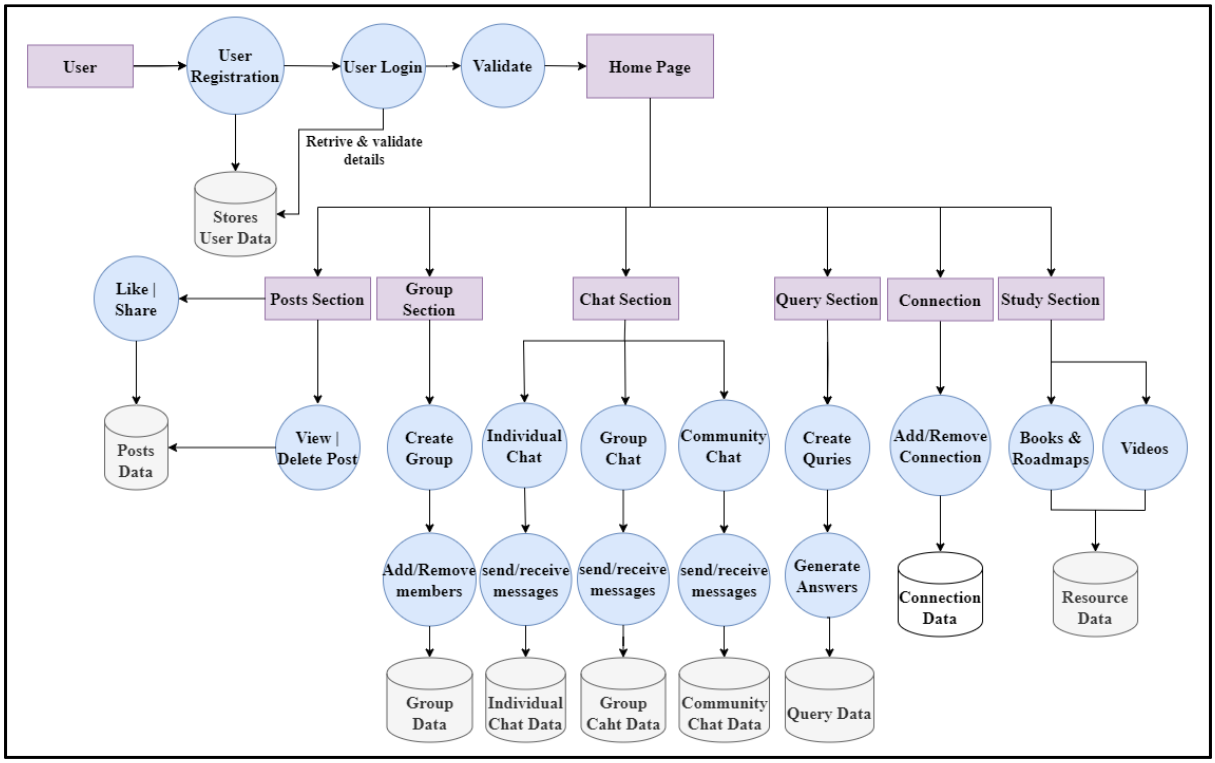


Fig. 4.2 Data Flow Diagram

In Figure 4.2, the data flow diagram for the "Campus Bridge" project shows how the data flows between different components of the system. The main external entity in the system is User. The User can register for an account, login to their account, and interact with the system through the Home page. The Home page is divided into four sections: Posts, Groups, Queries, and Study. Each section has its own database, which stores the data for that section. Users can interact with the Posts section to create, update, delete, like, and share posts. The Posts section database stores the data for all posts on the system, including the post's content, author, and creation date. Users can interact with the Groups section to join and leave groups. The Groups section database stores the data for all groups on the system, including the group's name, description, and members. Users can interact with the Queries section to create and generate queries. The Queries section database stores the data for all queries on the system, including the query's content, author, and creation date. Users can interact with the Study section to view and download resources. The Study section database stores the data for all resources on the system, including the resource's title, author, and publication date.

4.4 Use Case Diagram

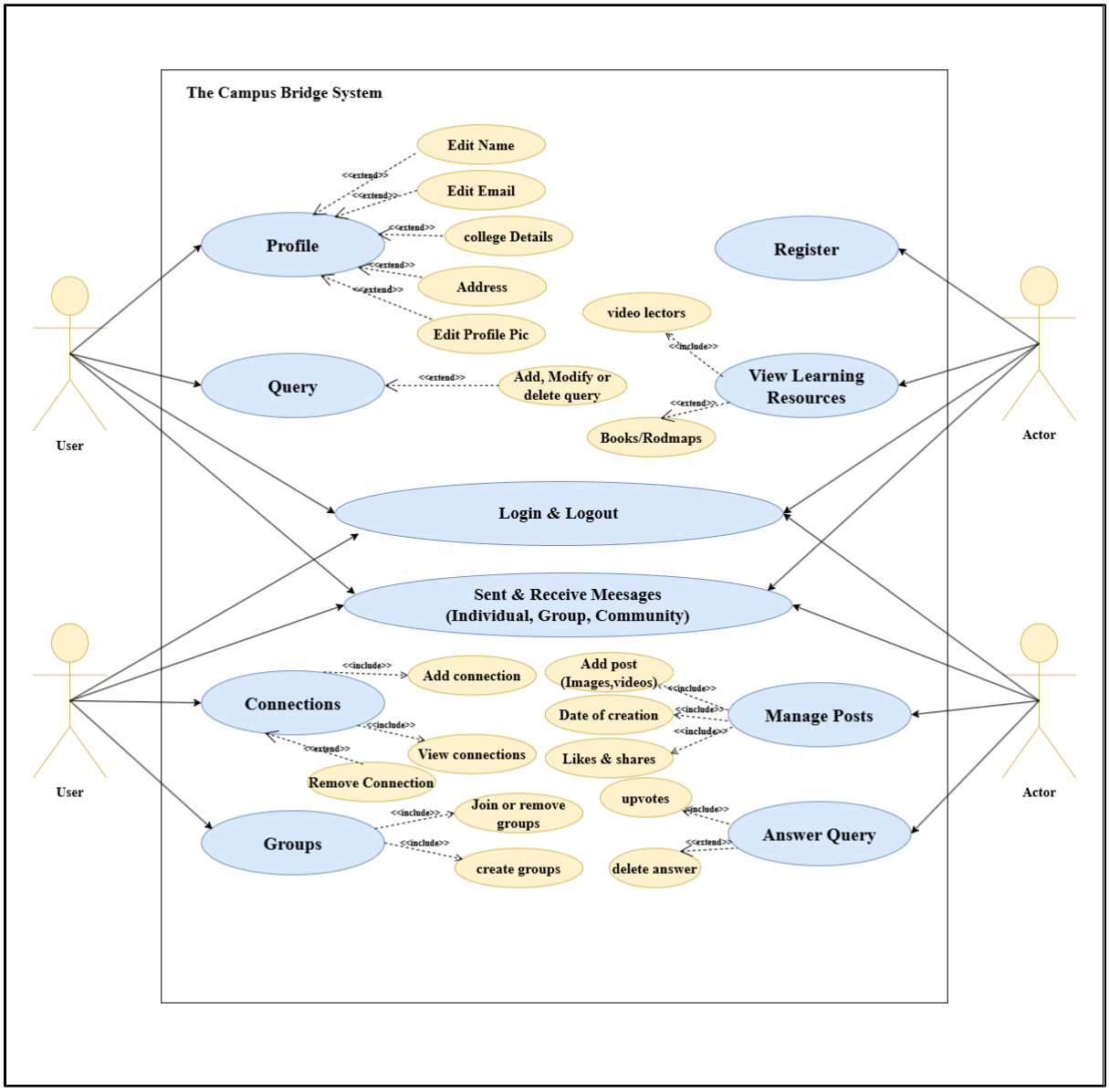


Fig. 4.3 Use Case Diagram

The use case diagram shows how user will interacts with the system. In the above use case diagram, there are multiple use case where user can interact to perform specific task like user can login to the system, sent messages etc.

4.5 Activity Diagram

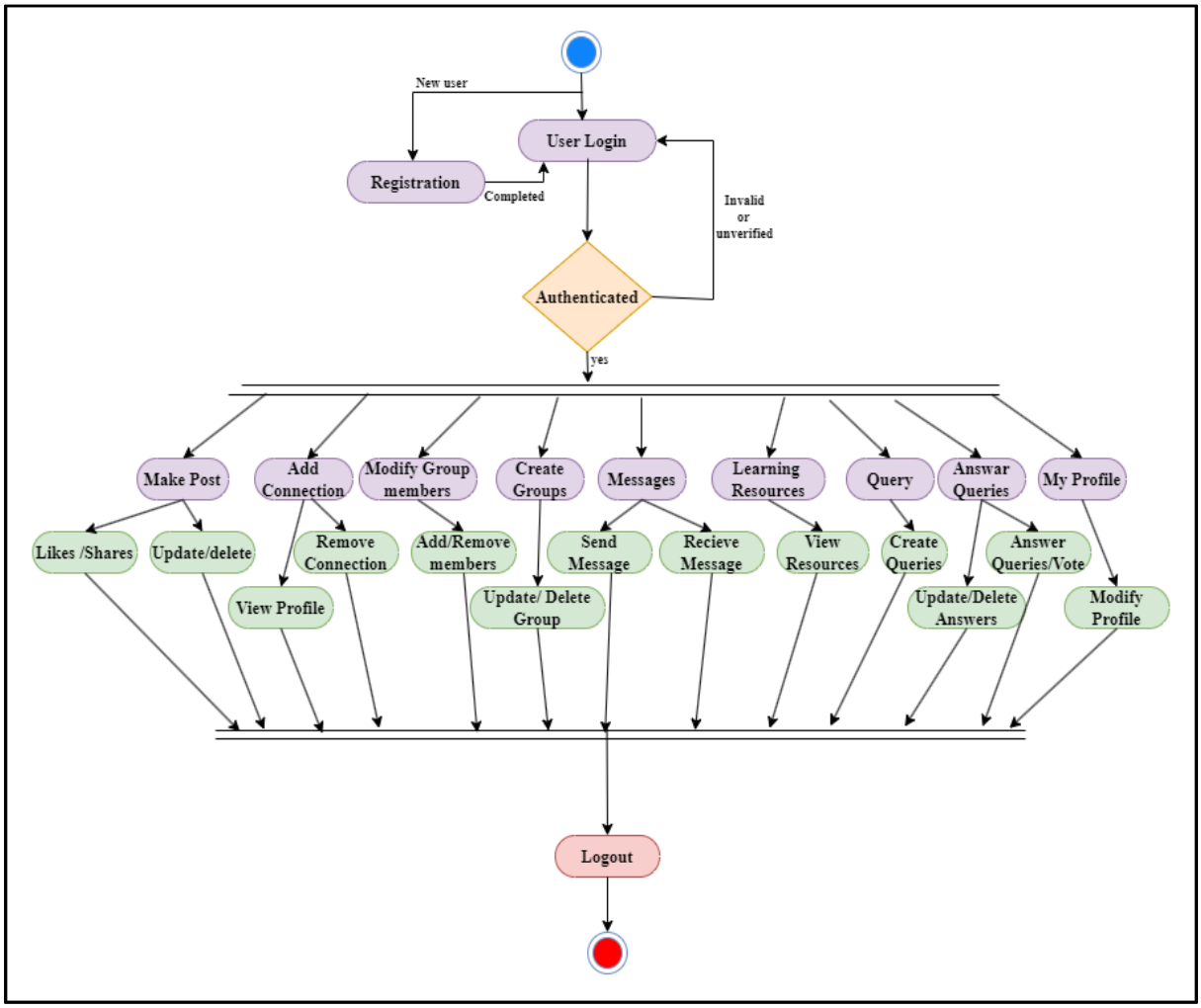


Fig. 4.4 Activity Diagram

In Figure 4.4 Activity Diagram of Campus Bridge Project, the main activities are User Activities and System Activities. User Activities include Register, Login, View Posts, Create Post, Update Post, Delete Post, Like Post, Share Post, Join Group, Leave Group, Create Query, View Query History, View Study Resources, Download Study Resources. System Activities include Validate User, Retrieve Posts, Save Post, Update Post, Delete Post, Update Likes, Update Shares, Add User to Group, Remove User from Group, Retrieve Query, Retrieve Study Resources. The activity diagram shows how the user activities and system activities interact with each other to achieve the overall goal of the system. For example, when a user creates a new post, the system activity "Create Post" is triggered to store the post in the database. When a user views a post, the system activity "Retrieve Post" is triggered to retrieve the post from the database.

4.6 Class Diagram

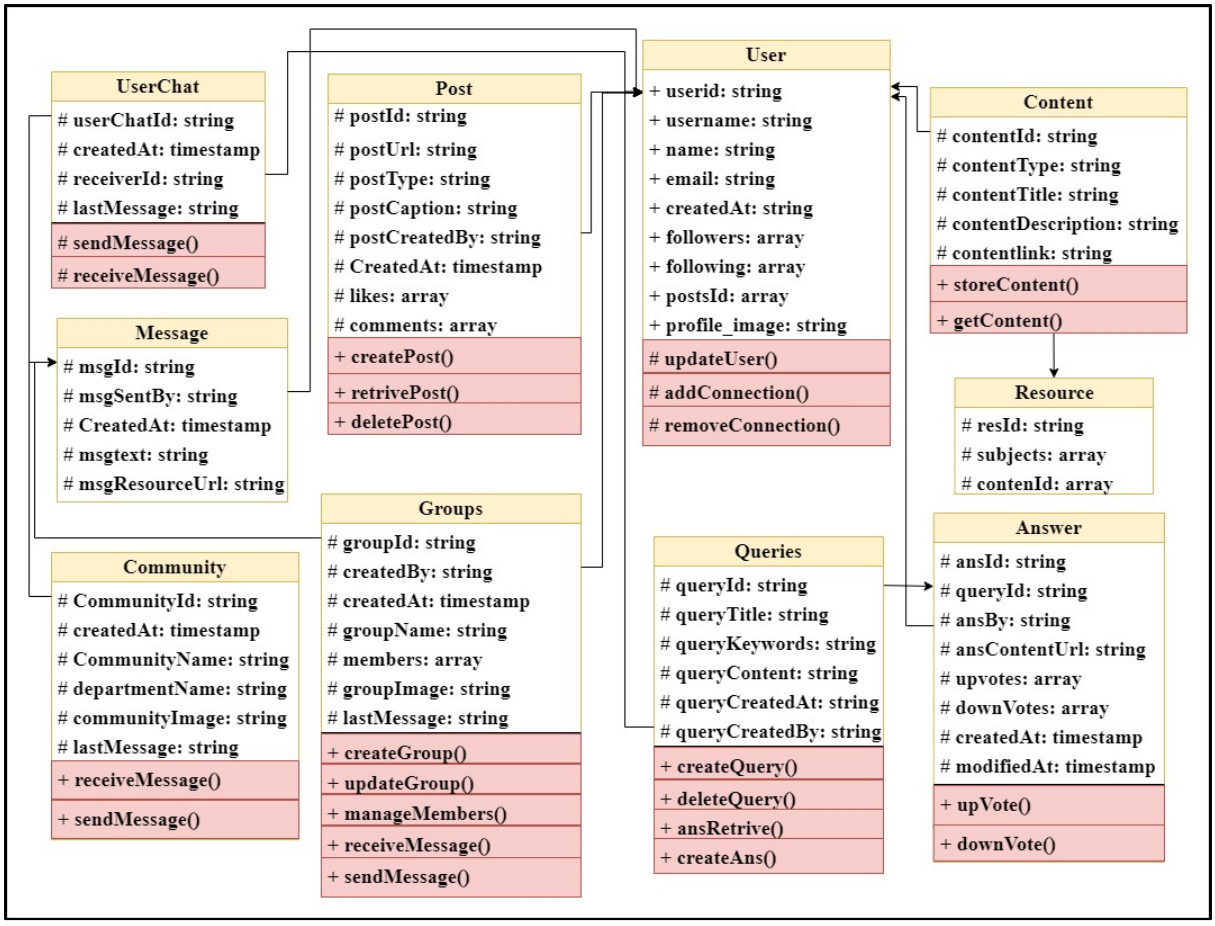


Fig. 4.5 Class Diagram

In Fig. 4.5 Class Diagram shows how different classes joined and what attributes are stored in particular class. For example Post contains postId, postural, postType, postCaption, postCreatedBy, CreatedAt, likes, commnets attributes along with its datatype and access specifier; It also shows methods in Post that are createPost(), retrievePost(), deletePost().

4.7 Sequence Diagram

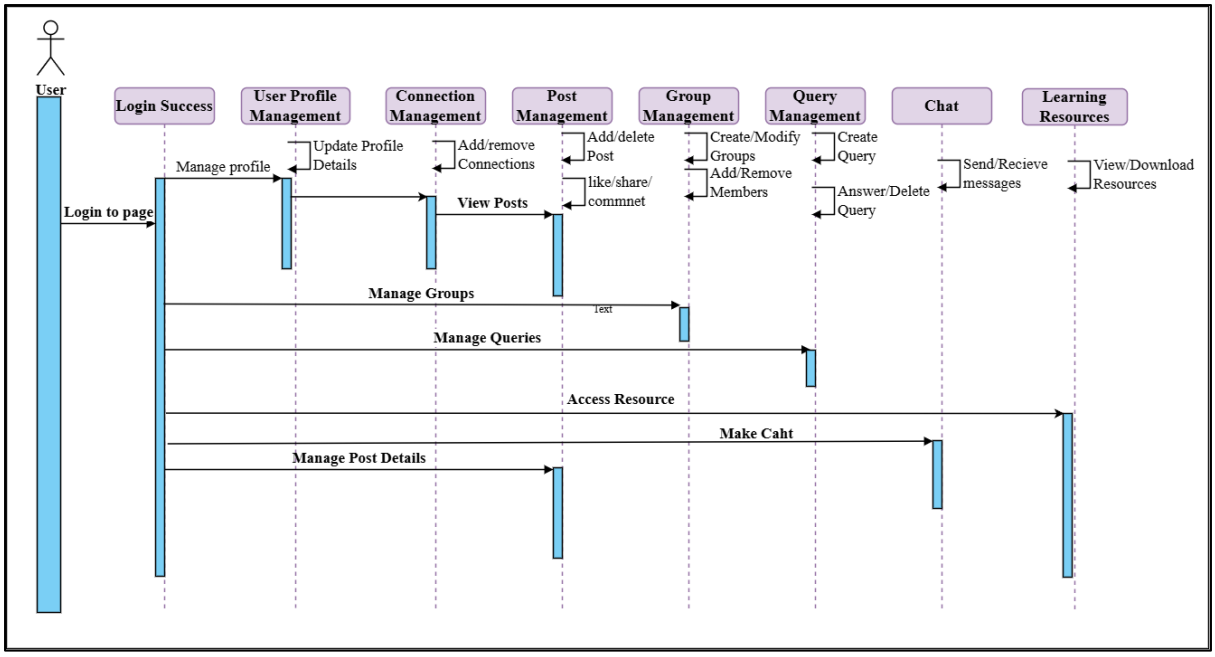


Fig. 4.6 Sequence Diagram

In Figure 4.7.1 Sequence Diagram of Campus Bridge, the main objects are User, Post, Group, Query, and Resource. The sequence diagram shows how these objects interact with each other to achieve the overall goal of the system. The sequence diagram starts with the User object sending a login request to the System object. The System object validates the User object's login credentials and then creates a session for the User object. The User object can interact with the other objects in the system. The User object can also send a request to the Group object to join a group. The Group object will be add the User object to the group's member list and return a success response to the User object. The User object can also send a request to the Query object to create a new query. The User object can also interact with the Resource object to view or download resources. The Resource object will retrieve the requested resource from the database and send it to the User object. The sequence diagram shows the sequence of events that occur when a user object interacts with the Campus Bridge system.

4.8 Flow Chart

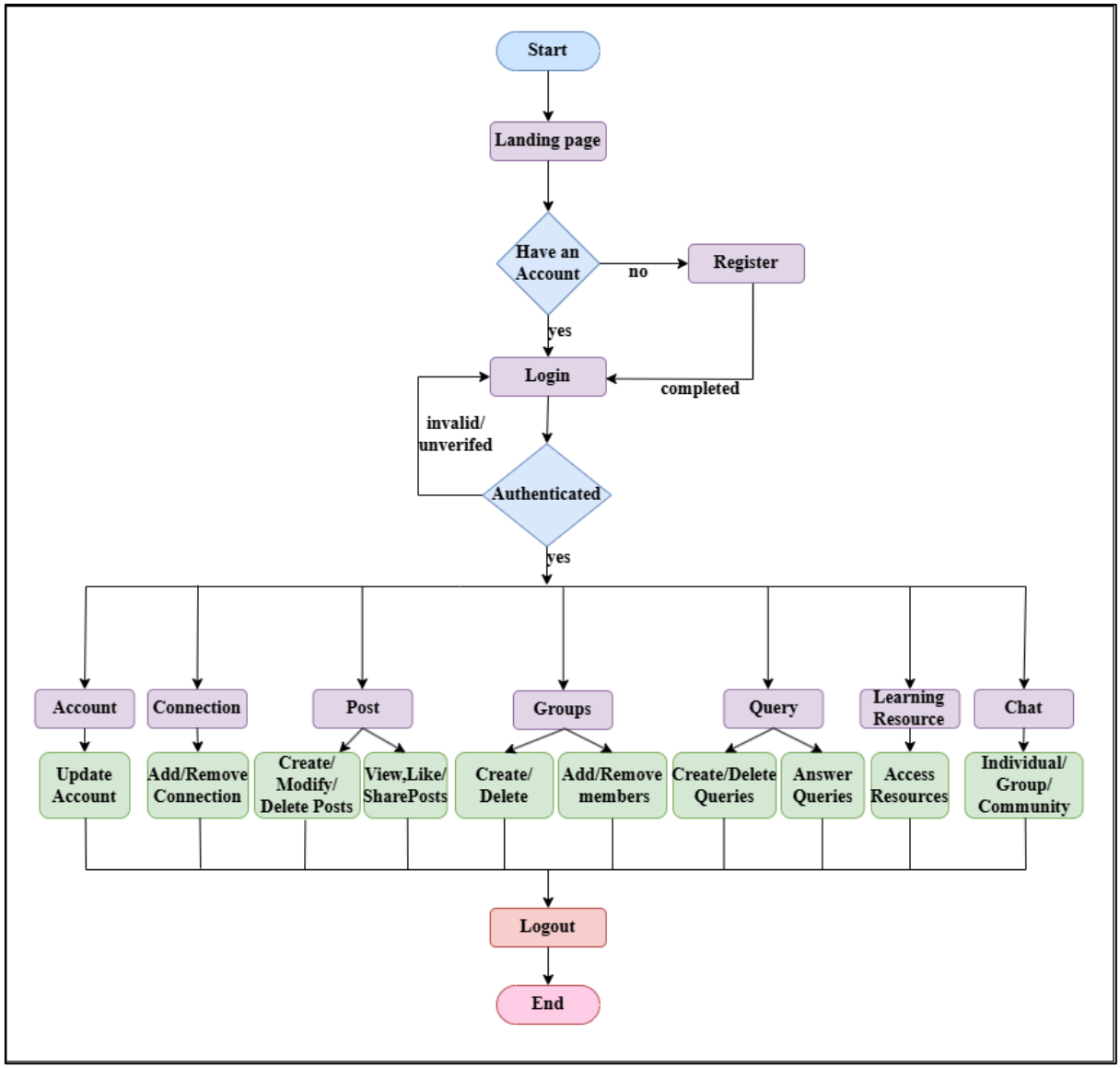


Fig. 4.7 Flow Chart

In Figure 4.8.1 Flow Chart of Creating a New Account in Campus Bridge Project, the user initiates the process of creating a new account by navigating to the Campus Bridge website and clicking on the "Create Account" button. This typically takes the user to a page where they can enter their name, email address, and password. Once the user has entered their information, they must click on the "Create Account" button to submit it to the system. The system then validates the user's input to ensure that it is valid. This includes checking to make

sure that the user's email address is in a valid format and that their password is strong enough. If the user's input is valid, the system will create a new account for them.

Once the account has been created, the system will automatically log the user in and take them to the home page. The home page is typically where users can access all of the features of the Campus Bridge system. The flow chart provides a step-by-step overview of the process of creating a new account in the Campus Bridge system. This information can be used to design and implement the system, as well as to document the system's process for creating new accounts. Additionally, the flow chart can be used to troubleshoot any problems that may occur during the account creation process**.**

5. CODING TECHNIQUES AND IMPLEMENTATION DETAILS

5.1 Sample codes:

Main.jsx: This file serves as the entry point of the application where react is initialized and the root component is rendered into the DOM

```
import React, { StrictMode } from "react";
import ReactDOM from "react-dom/client";
import App from "./App.jsx";
import { ChakraProvider } from "@chakra-ui/react";
import { extendTheme } from "@chakra-ui/react";
import { mode } from "@chakra-ui/theme-tools";
import { BrowserRouter } from "react-router-dom";
import { AuthContextProvider } from "./context/AuthContext";
import { ChatContextProvider } from "./context/ChatContext";
import { ChatContextGroup } from "./context/ChatContextGroup";
import { ChatContextCommunity } from "./context/ChatContextCommunity";
const styles = {
  global: (props) => ({
    body: {
      bg: mode("#0A0C10", "#0A0C10")(props),
      color: "#FFFFFF",
    },
  }),
};
const config = {  initialColorMode: "dark",  useSystemColorMode: false};
const theme = extendTheme({ config, styles});
ReactDOM.createRoot(document.getElementById("root")).render(
  <AuthContextProvider>
  <ChatContextProvider>
    <ChatContextCommunity>
    <ChatContextGroup>
    <StrictMode>
      <BrowserRouter>
        <ChakraProvider theme={theme}>
          <App />
        </ChakraProvider>
      </BrowserRouter>
    </StrictMode>
    </ChatContextGroup>
    </ChatContextCommunity>
    </ChatContextProvider>
  </AuthContextProvider>
);
```

- **Import Statements**: The code begins with importing necessary modules and components from React, ReactDOM, Chakra UI, React Router, and custom context providers.
- **Global Styles and Theme**: The **styles** object defines global styles for the application using Chakra UI's **mode** function to set background color based on the color mode (light or dark). These styles are then incorporated into the theme using the **extendTheme** function, along with configuration options such as initial color mode and system color mode.
- **Root Element Rendering**: The **ReactDOM.createRoot** function is used to create a root instance for rendering React elements. It targets the HTML element with the id "root" in the document and renders the application inside it.
- **Context Providers**: It wraps the **<App />** component with multiple context providers:
    - **AuthContextProvider**: Manages authentication-related state and functions.
    - **ChatContextProvider**: Manages chat-related state and functions.
    - **ChatContextCommunity**: Provides context for community-related chat.
    - **ChatContextGroup**: Provides context for group chat functionality.
- **StrictMode**: The **<StrictMode>** component is a tool for highlighting potential problems in an application. It checks for deprecated lifecycle methods and other potential issues.
- **BrowserRouter and ChakraProvider**: Inside the **<StrictMode>**, the **<BrowserRouter>** component from React Router is used to enable routing in the application. The **<ChakraProvider>** component from Chakra UI is used to provide the application with Chakra UI's theme and styling.
- **App Component**: The **<App />** component is rendered inside the **<BrowserRouter>** and **<ChakraProvider>**. This is likely the main component of the application, which contains the routing logic and other components.

This code sets up a React application with authentication and chat functionality, styled using Chakra UI, and utilizes React Router for navigation. The use of context providers suggests a modular approach to managing state and functionality across different parts of the application.

App.jsx: This file sets up the routing logic for the application, ensuring that different components are rendered based on the URL path and the user's authentication status

```
import { Navigate, Route, Routes } from "react-router-dom";
import Landing_page from "./pages/HomePage/landing_page";
import HomePage from "./pages/HomePage/HomePage";
import QueryPage from "./pages/QueryPage/QueryPage";
import AnswerQuery from "./components/Query/Answer/AnswerQuery";
import ChatPage from "./pages/ChatPage/ChatPage";
import StudyPage from "./pages/StudyPage/StudyPage";
import AuthPage from "./pages/AuthPage/AuthPage";
import PageLayout from "./Layouts/PageLayout/PageLayout";
import ProfilePage from "./pages/ProfilePage/ProfilePage";
import SharePost from "./components/FeedPosts/SharePost";
import { useAuthState } from "react-firebase-hooks/auth";
import { auth } from "./firebase/firebase";
function App() {
 const [authUser] = useAuthState(auth);
 return (
    <PageLayout>
        <Routes>
            <Route path='/' element={authUser ? <HomePage /> : <Navigate to='/main' />} />
            <Route path='/main' element={!authUser ? <Landing_page /> : <Navigate to='/'  />} />
            <Route path='chat/:username' element={authUser ? <ChatPage /> : <Navigate to='/main' />} />
            <Route path='/auth/:form' element={!authUser ? <AuthPage /> : <Navigate to='/' />} />
            <Route path='/:username' element={authUser ? <ProfilePage /> : <Navigate to='/main' /> } />
            <Route path='posts/:postId' element={authUser ? <SharePost /> : <Navigate to='/main' /> } />
            <Route path='/query' element={authUser ? <QueryPage /> : <Navigate to='/main' /> } />
            <Route path='/query/:queryId' element={authUser ? <AnswerQuery /> : <Navigate to='/main' /> } />
            <Route path='/study' element={authUser ? <StudyPage /> : <Navigate to='/main' /> } />
        </Routes>
    </PageLayout>
    );
}
export default App;
```

The Routes component from React Router is used to define the routes of the application. Inside it, multiple Route components are used to specify different paths and their corresponding components to be rendered. If the user is authenticated, the corresponding component is rendered; otherwise, the user is redirected to the main landing page or authentication page.

The PageLayout component is wrapped around the Routes, indicating that it serves as a layout component for the entire application. This layout component likely includes common elements such as headers, footers, or navigation bars that are shared across multiple pages.

Upload to Firebase Firestore Database and Storage:

```
import { ref, uploadString, getDownloadURL } from "firebase/storage";
import { doc, setDoc, collection,Timestamp } from "firebase/firestore";
import { storage } from "../firebase/firebase";
try {
  const queryCollectionRef = collection(firestore, "query");
  const newQueryRef = doc(queryCollectionRef);
  const storageReff = ref(storage, `query/${newQueryRef.id}.txt`);

  await uploadString(storageReff, inputs.content, 'raw');
  const downloadUrl = await getDownloadURL(storageReff);

  const userDoc = {
    queryTitle: inputs.title,
    queryKeywords: inputs.keywords,
    queryContent: downloadUrl,
    queryId: newQueryRef.id,
    uid: authUser.uid,
    createdAt:   Timestamp.now(),
  };
  await setDoc(newQueryRef, userDoc);
  showToast("Success", "Query created successfully", "success");
} catch (error) {
  showToast("Error", "Failed to create query", "error");
}
```

Import the necessary Firebase modules and initialize the Firebase app with your Firebase config. Then, get a Firestore instance using getFirestore().Next, define the collection where to add or update the document using collection(). Also define the document data as an object with name, email, and age fields.

Then, define the document reference using doc(), passing in the collection and the ID of the document to add or update. Note that if the document ID does not exist in the collection, it will be created.

Finally, use the setDoc() method to add or update the document in the collection. This method takes two arguments: the document reference and the document data. If the document already exists, setDoc() will update its contents with the new data. If the document does not exist, setDoc() will create it with the specified data.

Note that the setDoc() method is an asynchronous operation, so need to use await to wait for it to complete before logging a success message

Delete from Firebase Firestore Database and Storage:

```
try {
  const queryCollectionRef = collection(firestore, "query");
  const q = query(queryCollectionRef, where("uid", "==", authUser.uid), where("queryId", "==", queryId)
  const querySnapshot = await getDocs(q);
  if (!querySnapshot.empty) {
    const answerCollectionRef = collection(firestore, "answer");
    const answerQuery = query(answerCollectionRef, where("queryId", "==", queryId));
    const answerSnapshot = await getDocs(answerQuery);
    const folderRef = ref(storage, `answer/${queryId}/`);
    try {
      const files = await listAll(folderRef);
      await Promise.all(files.items.map(async (fileRef) => {
        await deleteObject(fileRef);
      }));
    } catch (error) {
      console.error(`Error deleting folder`, error);
    }
    answerSnapshot.forEach(async (answerDoc) => {
      await deleteDoc(doc(answerCollectionRef, answerDoc.id));
    });

    const docToDelete = querySnapshot.docs[0];
    await deleteDoc(doc(queryCollectionRef, docToDelete.id));

    const storageRef = ref(storage, `query/${queryId}.txt`);
    await deleteObject(storageRef);

    showToast("Success", "Query deleted successfully", "success");
  } else {
    showToast("Error", "Query not found", "error");
  }
} catch (error) {
  showToast("Error", "Failed to delete query", "error");
}
```

The deleteDoc() and deleteObject() methods are used to delete document from Firestore and files from storage respectively. Note that the deleteDoc() and deleteObject() method is an asynchronous operation, so we need to use await to wait for it to complete before logging a success message.

5.2User Interfaces

5.2.1 Home Page

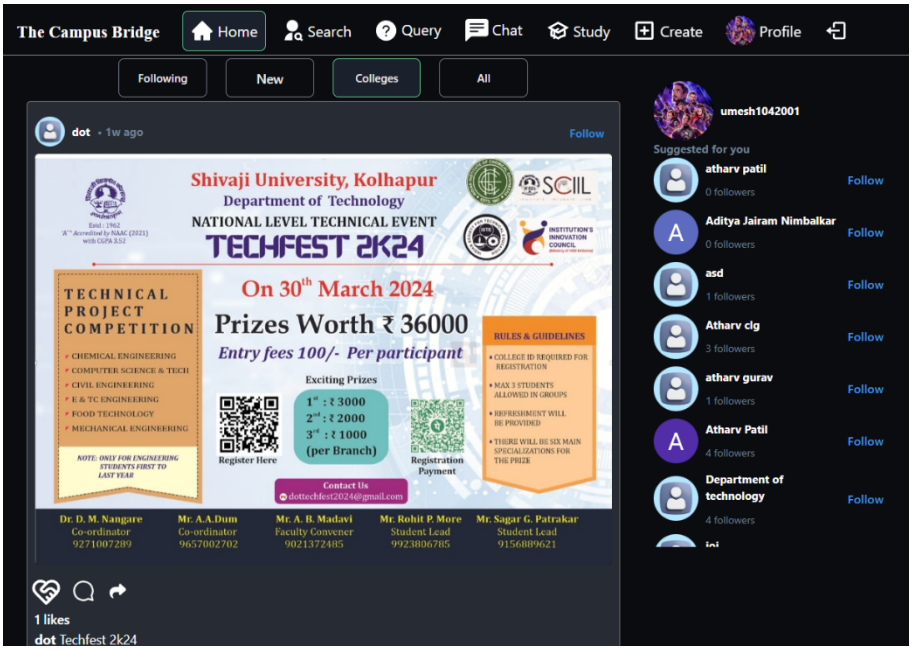


Fig. 5.1 Home page

It is the Home page which consists navigation bar for changing the sections, It shows the post by different categories like following, new, colleges and all. On the right side shows suggestions to connect according to the course, college, city. etc.

5.2.2 Query

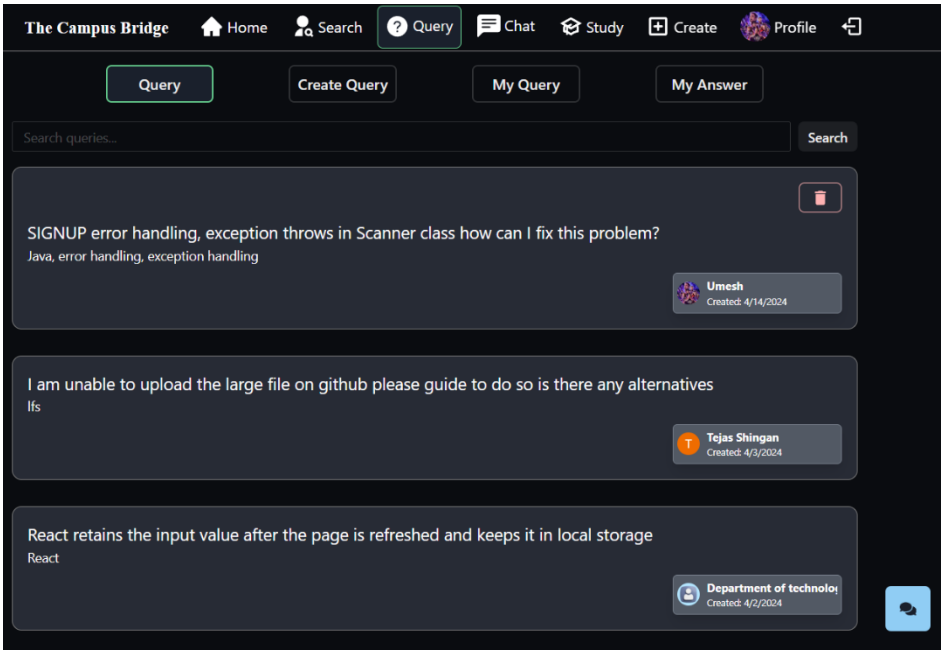


Fig. 5.2 View and Search Queries

It is the Query page for view and search queries, and choose the query to answer it or to check the submitted answers, also can vote the answers.

5.2.3 Chat

Individual Chat

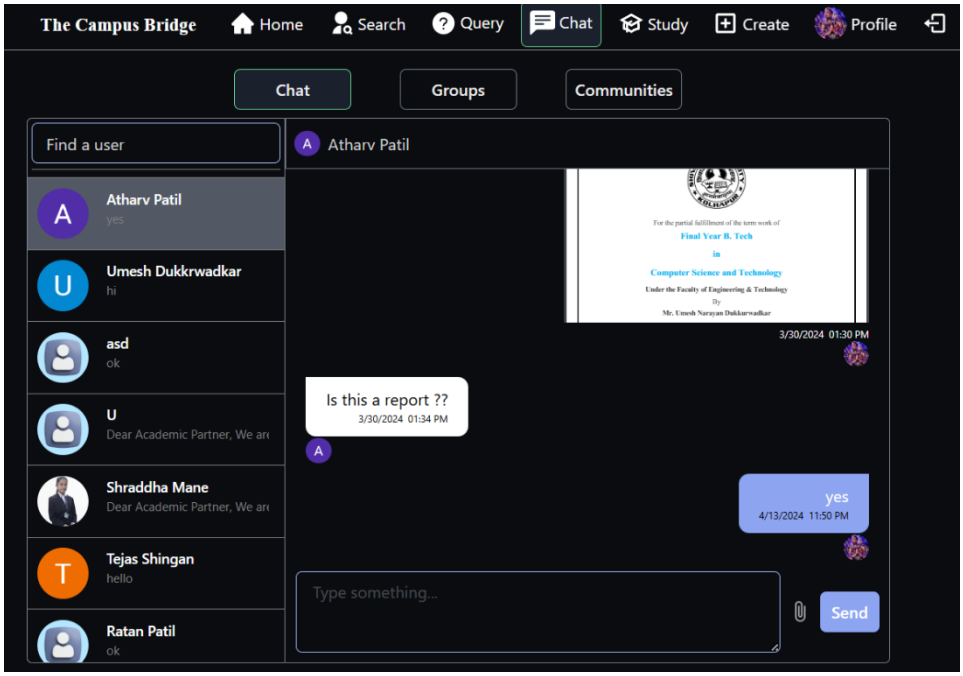


Fig. 5.3 Individual Chat

This page provides functionality for communicating with other student. It allows student to student interactive communication. It helps to share texts, Images, Videos and pdfs.

5.2.4 Study Section

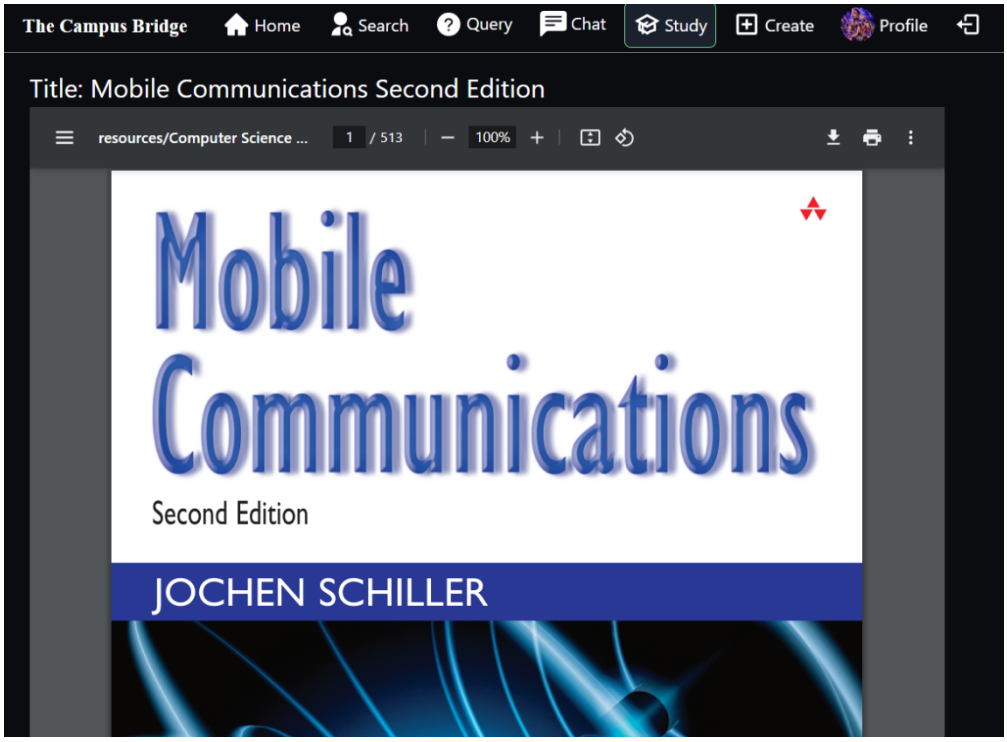


Fig. 5.4 View PDF Resources

This page provides functionality to view specific subjects book or roadmap in pdf format and also provides ability to download it.

5.2.5 View Profile

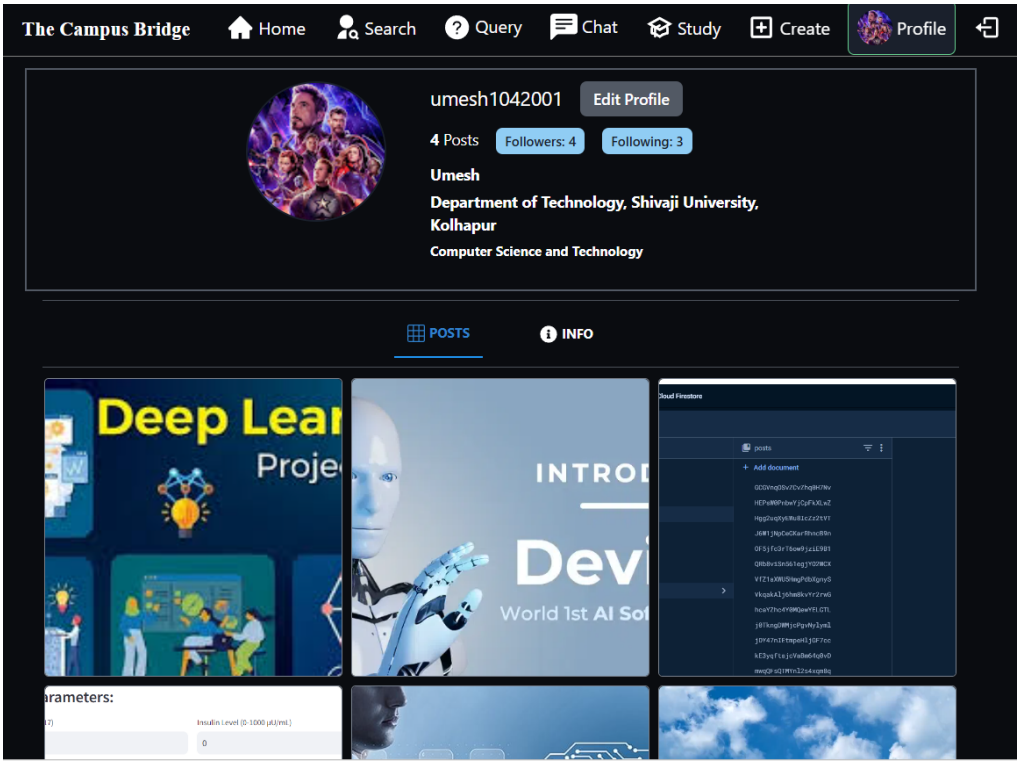


Fig. 5.5 View Profile

This page provides allows to view profile, it consists the following and followers names of specific students, posts and also personal Information.

5.2.6 Evaluation and Interpretation:

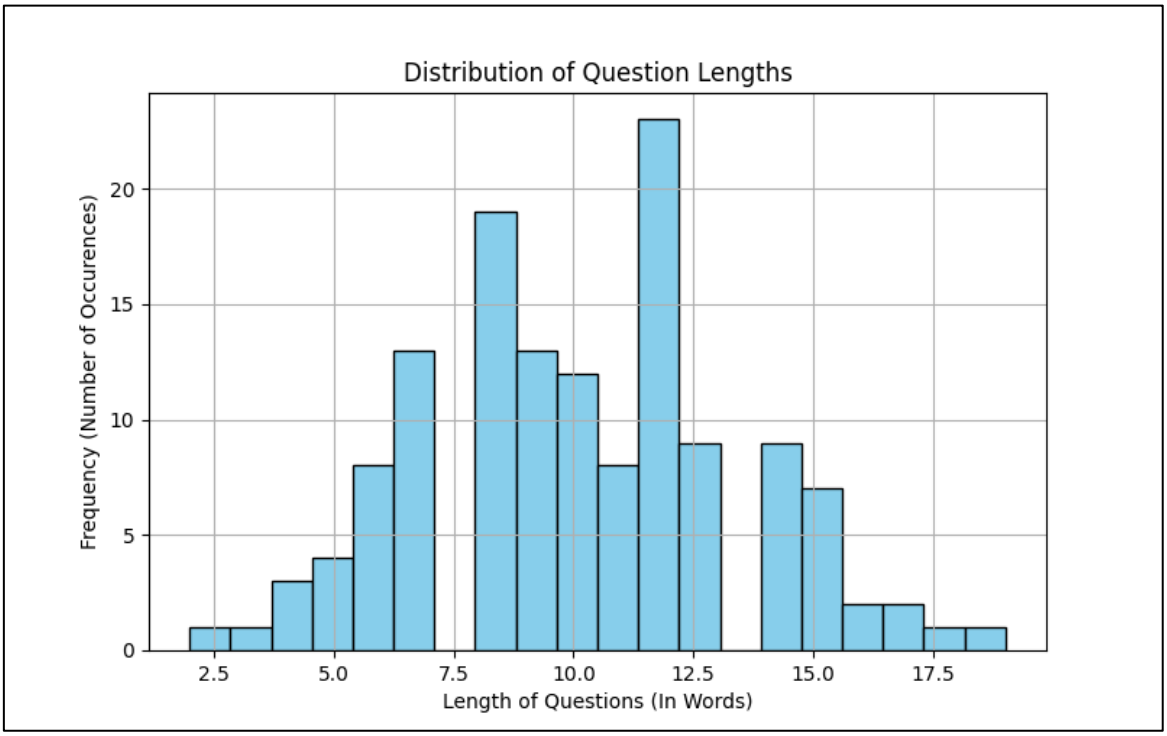


Fig. 5.6 Question Length

As you see in the Figure 4.2 the histogram provides insights into the distribution of question lengths within your dataset. By observing the shape of the histogram, you can understand the typical length range of questions. This understanding is crucial for designing NLP models or algorithms, as it helps in setting appropriate parameters or designing models that can handle questions of varying lengths effectively.

The results of the cosine similarity analysis were evaluated through a combination of quantitative and qualitative approaches. Descriptive statistics, such as mean, median, and standard deviation, were calculated to provide an overview of the similarity distribution. Additionally, the research team conducted thematic analysis on the high- similarity content to gain deeper insights into the nature and quality of the knowledge-sharing and collaborative activities within the platform.

5.3Testing

5.3.1 TESTING METHODOLOGIES

Black box Testing

It is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application. Usually Test Engineers are involved in the black box testing.

White box Testing

It is the testing process in which tester can perform testing on an application with having internal structural knowledge. Usually, The Developers are involved in white box testing.

Example for GUI Test cases

Table 5.2 GUI Test Cases

T.C. No	Description	Expected value	Actual value	Result
1	Checking whether all the components are properly arranged or not	The GUI must contain all the components properly arranged	Arranged properly	Pass
2	Checking the alignment of components placed	The alignment should be in proper way	Alignment should be correct	Pass

Positive Test Cases:

- The positive flow of the functionality must be considered.
- Valid inputs must be used for testing.
- Must have the positive perception to verify whether the requirements are justified.
- Verify that the system maintains data integrity and consistency when processing valid inputs.
- Ensure that positive test cases cover all user roles and permissions, verifying that each user type can successfully perform their intended actions.
- Ensure that the system provides appropriate feedback to the user for successful operations, such as success messages or confirmation dialogs.

Table 5.1 Positive Test Cases

Sr. No.	Test Case	Expected Result	Actual Result	Status
1	Check User is logged in or not	If user is logged in and authenticated redirect to home page else redirect to landing page.	User is redirecting properly on the basis of login status	Pass
2	Registration: Successful registration	User account created successfully, redirected to the home page	User account created successfully, redirected to the home page	Pass
3	Registration: Existing email	Error message displayed indicating the email is already in use	Error message displayed indicating the email is already in use	Pass
4	Login: Successful login	User logged in successfully, redirected to the home page	User logged in successfully, redirected to the home page	Pass
5	Login: Invalid credentials	Error message displayed indicating invalid credentials	Error message displayed indicating invalid credentials	Pass
6	When creator profile is available	Avatar, username, and time ago displayed	[Avatar], [Username], [Time Ago] displayed	Pass
7	When creator profile is not available	Skeleton avatar and username displayed	Skeleton avatar and username displayed	Pass
8	When follow status is 'Following'	Follow button displays 'Unfollow'	Follow button displays 'Unfollow'	Pass
9	When follow status is 'Not Following'	Follow button displays 'Follow'	Follow button displays 'Follow'	Pass
10	When follow status is being updated	Follow button displays loading indicator	Follow button displays loading indicator	Pass
11	Posts: Display posts	Posts are displayed on	Posts are displayed on	Pass

		the home page	the home page	

6. APPLICATIONS

- **Professional Networking:**

Recycle X offers a dedicated professional networking platform for users within the waste management sector. Consumers, suppliers, and admins can create personalized profiles, showcasing their expertise and career aspirations in sustainable waste management. By making posts, answering queries, and interacting with others in the system, users can build a professional presence. This feature not only helps individuals showcase their academic achievements but also allows them to connect with others in the recycling and environmental sectors, facilitating career growth and knowledge sharing. The platform offers opportunities to discuss industry trends, advancements, and best practices in waste management, enabling users to expand their professional network.

- **Community Engagement:**

The community engagement feature of Recycle X provides an inclusive, collaborative space for users to interact, share ideas, and contribute to meaningful discussions about sustainability, recycling processes, and waste management practices. Through this section, users can participate in forums that cover a wide range of topics, from reducing waste to implementing effective recycling systems. Recycle X breaks down geographical barriers, enabling individuals from different parts of the world to collaborate and learn from each other, regardless of location. This diverse community encourages interdisciplinary discussions and enhances the overall user experience, promoting a sense of collective responsibility for the environment and sustainability.

- **Messaging:**

The messaging feature of Recycle X facilitates effective communication between users, enabling them to share knowledge, experiences, and advice with others. There are three main communication channels:

1. **Individual Chats:** These allow users to communicate directly with one another, fostering personalized connections and one-on-one knowledge exchange.

2. **Group Chats:** Users can create or join groups for more focused discussions on specific topics related to waste management, recycling techniques, or industry challenges. Group chats allow for collaborative problem-solving and brainstorming sessions.
3. **Community Chats:** This feature enables users to engage in larger conversations with a wide audience, offering the opportunity to seek advice, share insights, and discuss the latest trends in sustainability. Whether it's discussing new recycling methods or sharing success stories, community chats act as an open forum for global knowledge exchange.

- **Query Section:**

The Query section in Recycle X serves as a valuable tool for users to get expert advice or share their own knowledge on various topics related to waste management. Users can post questions on topics they need guidance on, ranging from sustainable recycling methods to waste processing technologies. Other users, with expertise in these areas, can contribute by providing detailed answers and voting on the most helpful responses. This collaborative approach ensures that the platform becomes a hub of knowledge, where users can both ask for and provide assistance on important environmental issues. For quick answers, a **Chatbot**, powered by **Natural Language Processing (NLP)**, is available to offer immediate responses to commonly asked queries. This chatbot can answer questions quickly, making the platform more accessible and user-friendly for all.

- **Resource Hub:**

The Resource Hub of Recycle X acts as an extensive repository of educational materials and tools focused on waste management, recycling, and sustainability. The hub contains textbooks, instructional videos, industry reports, and articles on best practices in waste management. Students, professionals, and organizations can access these resources to improve their knowledge base and understanding of the latest developments in the industry. Additionally, the platform offers **personalized roadmaps** that guide users in planning their waste management efforts, from initial waste collection to final recycling. This allows individuals and businesses to chart a clear path towards more sustainable operations, empowering them to take charge of their environmental impact and manage waste more efficiently.

7. CONCLUSION

Recycle X is an innovative waste management platform that integrates **React**, **React Native**, **Node.js**, and **MySQL** to provide an efficient, scalable solution. Using **MVC architecture**, the system ensures a clean and organized structure for easy maintenance and development. **Email OTP authentication** secures the platform, while **backend logging** ensures transparency and allows for easy monitoring of activities.With **MySQL** for data storage, the system handles user and waste management data efficiently using **indexing** and **stored procedures** for optimized queries. The platform supports three main roles: **Admin**, **Consumer**, and **Supplier**, each contributing to the seamless flow of waste collection, recycling, and material distribution.Recycle X's main goals are to promote **sustainable waste management**, increase **recycling awareness**, and contribute positively to the environment, making it a vital tool in addressing waste management challenges.

8. FUTURE WORK

In the upcoming future, we can focus enhancing the platform's AI capabilities in the Query Section as a priority. This involves the refinement of AI-powered solutions to provide more accurate responses to student queries, offering an even more valuable learning resource. We can explore partnerships with educational institutions to offer the platform as a supplementary resource for their students. This can make platform more scalable. Create a system for collecting feedback from students and using it to make continuous improvements. This could include surveys, feedback forms, or a dedicated feedback section on the platform.

9. REFERENCES

[1] Gastrointestinal Endosc. *Application of optical character recognition with natural language processing for large-scale quality metric data extraction in colonoscopy reports.* giejournal 2021

[2] Gordon Drummond. *Success in Online Education Creating a Roadmap for Student Success.* United States Distance Learning Association 2008

[3] L. Szabó, A. Csepregi and, N. Obermayer-Kovács Z. Gaál. *Exploring the role of social media in knowledge sharing.* Ejkm. 2015

[4] Jane E. Brindley, Lisa M. Blaschke, Christine Walti. *Creating Effective Collaborative Learning Groups in an Online Environment* International Review of Research in Open and Distance Learning. 2009

[5] Jane E. Brindley, Lisa M. Blaschke and Christine Walti *Creating Effective Collaborative Learning Groups in an Online Environment* (Athabasca University Press) [June 2009]

[6] Janet Macdonald. *Assessing online collaborative learning: process and product.* ScienceDirect. 2003

[7] Kiran Khatter, Diksha Khurana, Aditya Koli, Sukhdev Singh. *Natural language processing: state of the art, current trends and challenges. Multimed Tools.* SpringerLink 2022

[8] Mundschenk, Nancy A.; Fuchs, Wendy W. *Professional Learning Communities: An Effective Mechanism for the Successful Implementation and Sustainability of Response to Intervention*. SRATE Journal 2016

[9] Prabu David and Jung-Hyun Kim *Mobile phone distraction while studying.* new media & society, 2014: 1-19

[10] Raymond Chiong, Jelena Jovanovic. *Collaborative Learning in Online Study Groups: An Evolutionary Game Theory Perspective.* Journal of Information Technology Education: Research, 2012

[11] Waleed Mugahe, AL-Rahmi, Mohd Shahizan Othman. *Evaluating student's satisfaction of using social media through collaborative learning in higher education.* International Journal of Advances in Engineering & Technology, 2013.

[12] Yunis Ali Ahmed Mohammad Nazir Ahmad b, Norasnita Ahmad, Nor Hidayati Zakaria. *social media for knowledge-sharing.* Telematics and Informatics, 2019.

10. Appendix A

- **API**: Application Programming Interface
- **App**: Application
- **BERT**: Bidirectional Encoder Representations from Transformers
- **BSTM**: Bayesian Sentence-based Topic Model
- **CORS**: Cross-Origin Resource Sharing
- **CPU**: Central Processing Unit
- **CSS**: Cascading Style Sheets
- **DOM**: Document Object Model
- **EGT**: Evolutionary Game Theory
- **FGB**: Factorization with Given Bases
- **GUI**: Graphical User Interface
- **HTML**: Hypertext Markup Language
- **HTTP**: Hypertext Transfer Protocol
- **ID**: Identifier
- **JSON**: JavaScript Object Notation
- **LLMs**: Large Language Models
- **MPIL**: Mobile Phone Interference in Life
- **NLP**: Natural Language Processing
- **NLG**: Natural Language Generation
- **PDFs**: Portable Document Format
- **PLCs**: Professional Learning Communities
- **RAM**: Random Access Memory
- **RtI**: Response to Intervention
- **SDKs**: Software Development Kits
- **SEO**: Search Engine Optimization
- **SQL**: Structured Query Language
- **TAOS**: Topic Aspect-Oriented Summarization
- **UI**: User Interface
- **URL**: Uniform Resource Locator
- **UTM**: University Technology Malaysia
- **Web**: World Wide Web