

Contents

1	Introduction	2
2	Project Overview	3
3	Back-end Resources	5
3.0.1	User Section: Node.js and Express.js	5
3.0.2	Database Management: XAMPP and SQL	6
3.0.3	Admin Section: PHP	6
3.0.4	Integration of Technologies	6
4	Database Design and Development	8
4.1	Creation of Database	8
4.2	Tables used For the system	8
5	System Overview	11
5.1	User Interface Back-end Services	11
5.1.1	Registration:	11
5.1.2	Login:	12
5.1.3	Homepage (Before Login)	14
5.1.4	Home Page(Post Login) :	15
5.1.5	Topic Interface	16
5.1.6	Profile Management	21
5.1.7	About Profile	21
5.2	Admin Interface Back-end Services	22
5.2.1	Admins	22
5.2.2	Category Management	23
5.2.3	Topic Management	25
5.2.4	Reply Management	27
5.2.5	Profile Management	28
6	Conclusion	29

Chapter 1

Introduction

The CUET Student Forum is a web-based platform designed specifically for students of CUET (Chittagong University of Engineering Technology) to foster discussions, exchange ideas, and collaborate on academic and social topics. The platform is built to handle high user traffic and provides a seamless user experience by leveraging a robust and scalable backend architecture.

The backend is developed using Node.js and the Express framework, which ensures high-performance, non-blocking I/O operations to handle multiple requests simultaneously. The backend is responsible for managing several key features, including user authentication, topic management, category-based filtering, and user session handling. It also incorporates essential security features such as session management, password hashing, and validation mechanisms to prevent unauthorized access.

Chapter 2

Project Overview

The project is organized into several key files and directories:

- `index.js`: The main server file that sets up the Express server and defines the routes.
- `dbv2.js`: Handles the MySQL database connection using asynchronous functions.
- `public/`: Contains static assets like images, CSS, and JavaScript files.
- `views/`: Contains the EJS templates used for rendering the frontend.
- `admin/`: Manages the backend functionality for administrators including the ability to create, update, and delete admin users as well as manage categories and topics. Admins can log in, log out, and perform CRUD operations on various sections. Below is an example structure for handling admins:
 - `login-admin.php`: Handles admin login authentication.
 - `create-admin.php`: Allows an admin to create new admin accounts.
 - `show-admins.php`: Displays a list of all registered admins.
 - `delete-admin.php`: Provides functionality to delete an admin.

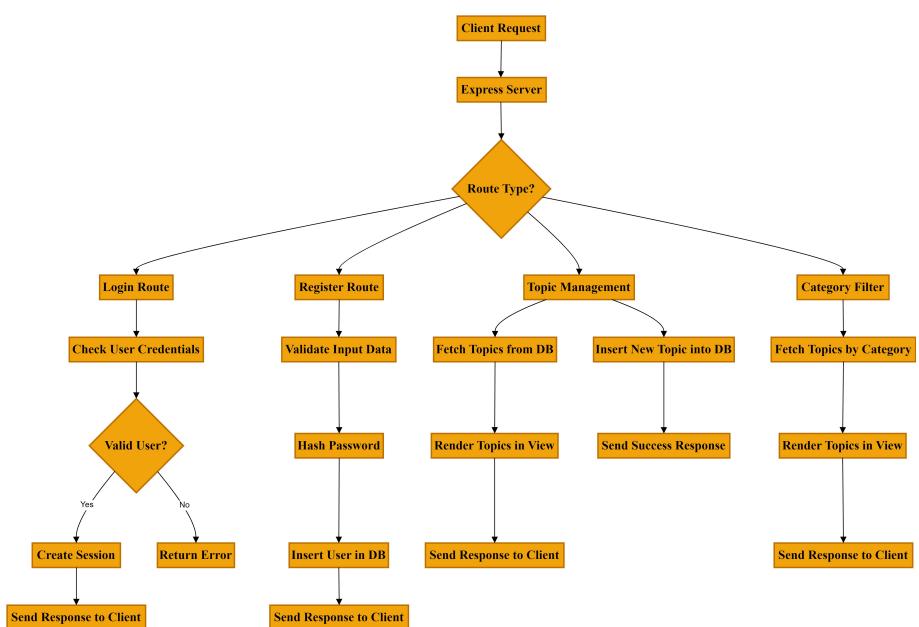


Figure 2.1: Backend Overview

Chapter 3

Back-end Resources

The backend of the project is developed using a combination of **Node.js**, **Express.js**, **XAMPP** with **SQL**, and **PHP**. This hybrid approach leverages the strengths of each technology and various Node.js and Express.js packages to create a robust and efficient system for both user and admin functionalities.

3.0.1 User Section: Node.js and Express.js

The user section of the backend is implemented using **Node.js** and **Express.js**. **Node.js** is a powerful JavaScript runtime environment that allows for server-side scripting, making it possible to use JavaScript throughout the entire stack. **Express.js** is a web application framework for Node.js that provides a lightweight and flexible way to handle server-side logic.

- **Node.js** is utilized for its non-blocking, event-driven architecture, which enhances performance and scalability, especially under high-load conditions [?].
- **Express.js** is used to manage routes, handle HTTP requests and responses, and implement middleware functions that process requests at different stages of the request-response cycle [?].
- Several third-party **Express.js** middleware packages are used to enhance functionality:
 - **Morgan**: Used for HTTP request logging to the console, which helps in debugging and monitoring requests [?].
 - **Multer**: Used for handling file uploads by configuring storage options and managing the file upload process [?].
 - **express-session**: Used for managing user sessions, allowing the server to store and access user-specific data across requests [?].
 - **method-override**: Allows the use of HTTP verbs such as PUT or DELETE in places where the client doesn't support it [?].

- **dotenv**: Loads environment variables from a ‘.env’ file into ‘process.env’, allowing sensitive information such as API keys and secrets to be securely managed [?].
- The combination of these packages allows for rapid development, efficient handling of operations such as form submissions, data validation, user authentication, and other dynamic functionalities.

3.0.2 Database Management: XAMPP and SQL

The project uses **XAMPP** as the development environment for the database. XAMPP is an open-source, cross-platform web server solution stack package that contains **Apache**, **MySQL (MariaDB)**, **PHP**, and **Perl**. The database used is **MySQL**, which is managed through the XAMPP control panel.

- **MySQL** serves as the relational database management system (RDBMS) that stores all the application data, including user information, forum posts, and other dynamic content [?].
- The database is structured in a normalized format to reduce redundancy and improve data integrity, supporting complex queries and relationships between tables.
- XAMPP provides a local server environment that simplifies the development and testing process for SQL operations [?].

3.0.3 Admin Section: PHP

The admin section of the backend is developed using **PHP**. PHP is a widely-used server-side scripting language that is particularly suited for web development.

- **PHP** is used to create the administrative interface, which allows administrators to manage users, moderate content, and perform other administrative tasks [?].
- PHP scripts interact directly with the **MySQL** database to perform CRUD (Create, Read, Update, Delete) operations, ensuring seamless management of data.
- The choice of PHP for the admin section allows for a secure and efficient server-side handling of administrative operations, separate from the Node.js and Express.js environment used for the user section.

3.0.4 Integration of Technologies

The combination of **Node.js/Express.js**, **PHP**, and **MySQL** through **XAMPP** allows for a separation of concerns where user and admin functionalities are isolated, ensuring better performance, security, and maintainability. Node.js and Express.js

handle the backend operations for user interactions, while PHP deals with the back-end administration tasks. This architecture provides a flexible and scalable backend solution tailored to the different needs of users and administrators.

Chapter 4

Database Design and Development

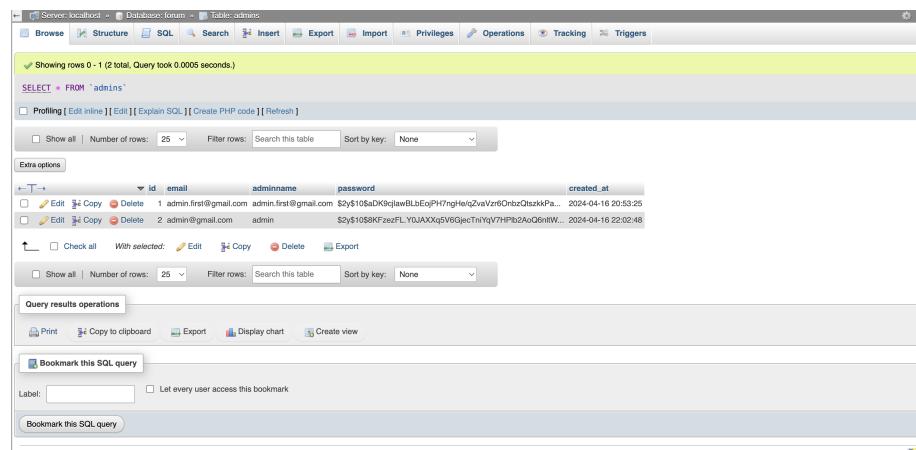
4.1 Creation of Database

Almost all web applications use a database. Here I have used a simple database using XAMPP.

4.2 Tables used For the system

Tables are the building blocks for a relational database system. All the entities and relationships between those entities are stored in the form of tables in a relational database.

Each of the tables has its own set of primary keys. A primary key is an attribute or a set of attributes which is used to identify a record from a table.



The screenshot shows the MySQL Workbench interface with the following details:

- Server:** localhost
- Database:** forum
- Table:** admins
- Query Results:** Shows the result of the query: "SELECT * FROM `admins`;" with 2 total rows.
- Table Structure:** Shows the columns: id, email, adminname, password, and created_at.
- Data:** Two rows of data are displayed:

	id	email	adminname	password	created_at
<input type="checkbox"/>	1	admin.first@gmail.com	admin.first@gmail.com	\$2y\$10\$adDKlqljawBLbEoPH7ngHe/qZvaVzr9OrbzQtszkhPa...	2024-04-16 20:53:25
<input type="checkbox"/>	2	admin@gmail.com	admin	\$2y\$10\$8KFzExFLYQJAXXq5V6QjeTrnYq7HPbzAoQnltW...	2024-04-16 22:02:46

Figure 4.1: Admin Database

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

SELECT * FROM `users`

	Edit	Copy	Delete	1 nino	nino@gmail.com	nino	\$2y\$10\$DK9cjawBLBeojPH7ngHe/qZvaVzr6OnbzQtzkkPa...	cse	gravatar.png	2024-04-04 17:54:19
	Edit	Copy	Delete	3 john	john@gmail.com	john	\$2y\$10\$EPPEoEswWBzgnQUsmgLjUcWHmMLppgh8Bcozep...	john the sheep	gravatar.jpg	2024-04-04 15:43:44
	Edit	Copy	Delete	4 nino	nino@gmail.com	nino	\$2y\$10\$bETudotUwJfrhcCKvHL73ebyXzaeNrsWY4U15WTE...	asasas	gravatar.jpg	2024-05-13 11:05:58
	Up	Check all	With selected:	Edit	Copy	Delete	Export			
	Down	Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None		

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

Label: Let every user access this bookmark

[Bookmark this SQL query](#)

Figure 4.2: User Database

Showing rows 0 - 5 (6 total, Query took 0.0006 seconds.)

SELECT * FROM `categories`

	Edit	Copy	Delete	1 Design	Design	2024-04-05 15:52:59			
	Edit	Copy	Delete	2 Development	Development	2024-04-05 15:52:59			
	Edit	Copy	Delete	3 Marketing	Marketing	2024-04-05 15:52:59			
	Edit	Copy	Delete	4 SEO	SEO	2024-04-05 15:52:59			
	Edit	Copy	Delete	5 Hosting	Hosting	2024-04-05 15:52:59			
	Edit	Copy	Delete	8 Artificial Intelligence	Artificial Intelligence	2024-08-23 14:40:45			
	Up	Check all	With selected:	Edit	Copy	Delete	Export		
	Down	Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None	

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

Label: Let every user access this bookmark

[Bookmark this SQL query](#)

Figure 4.3: Categories Database

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

SELECT * FROM `topics`

	Edit	Copy	Delete	1 post number one	1	Hosting	Allow miles wound place the leave had. To sitting ...	nino	gravatar.png	2024-04-03 18:47:03
	Edit	Copy	Delete	2 three	3	SEO	Cultivated who resolution connection motionless d...	nino	gravatar.png	2024-04-04 16:05:26
	Edit	Copy	Delete	8 john's post one	8	SEO	On he decisively impression attachment friendship ...	john	gravatar.jpg	2024-04-04 16:30:36
	Edit	Copy	Delete	9 post number 4	4	Design	Cultivated who resolution connection motionless d...	nino	gravatar.png	2024-04-05 16:05:00
	Up	Check all	With selected:	Edit	Copy	Delete	Export			
	Down	Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None		

Query results operations

[Print](#) [Copy to clipboard](#) [Export](#) [Display chart](#) [Create view](#)

[Bookmark this SQL query](#)

Label: Let every user access this bookmark

[Bookmark this SQL query](#)

Figure 4.4: Topic Database

Screenshot of a MySQL database browser showing the 'replies' table in the 'forum' database.

Table: replies

Showing rows 0 - 3 (4 total, Query took 0.0003 seconds.)

SELECT * FROM `replies`

Profile [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Extra options

	id	reply	user_id	user_image	topic_id	user_name	created_at
<input type="checkbox"/>	1	this is a new reply	1	gravitar.png	1	nino	2024-04-03 18:11:34
<input type="checkbox"/>	2	Oh he decisively impression attachment friendship ...	1	gravitar.png	6	nino	2024-04-03 18:11:34
<input type="checkbox"/>	5	why are u running	3	gravitar.jpg	1	john	2024-04-04 16:38:32
<input type="checkbox"/>	8	idiot asking for help	1	gravitar.png	8	nino	2024-04-05 19:30:06

Check all With selected: Edit Copy Delete Export

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Bookmark this SQL query

Label: Let every user access this bookmark

Bookmark this SQL query

Figure 4.5: Reply Database

Chapter 5

System Overview

5.1 User Interface Back-end Services

5.1.1 Registration:

To Register on the CUET Forum, users need to fill out the registration form with the following details:

- Name: The full name of the user.
- Email Address: A valid email address required for receiving notifications and accessing the platform.
- Choose Username: A unique username that will represent the user on the platform.
- Password: A secure password, which must be confirmed by entering it twice to ensure accuracy.
- Upload Avatar: An optional field where users can upload an image to be used as their profile picture.
- About Me: A text area powered by CKEditor. This rich text editor allows users to format their descriptions with options such as bold, italics, lists, and other formatting tools to create a more personalized introduction.

After the form is filled out and submitted, the system will validate the provided details. If the information is accurate and the email domain ends with @cuet.ac.bd, the system will create an account for the user, granting them access to the platform's features. In this case, a message saying Registered Successfully will be displayed. However, if the email does not end with "@cuet.ac.bd," an error message will appear, stating Invalid email domain. Only Cuetians are allowed." This ensures that only users with a valid CUET email can register.

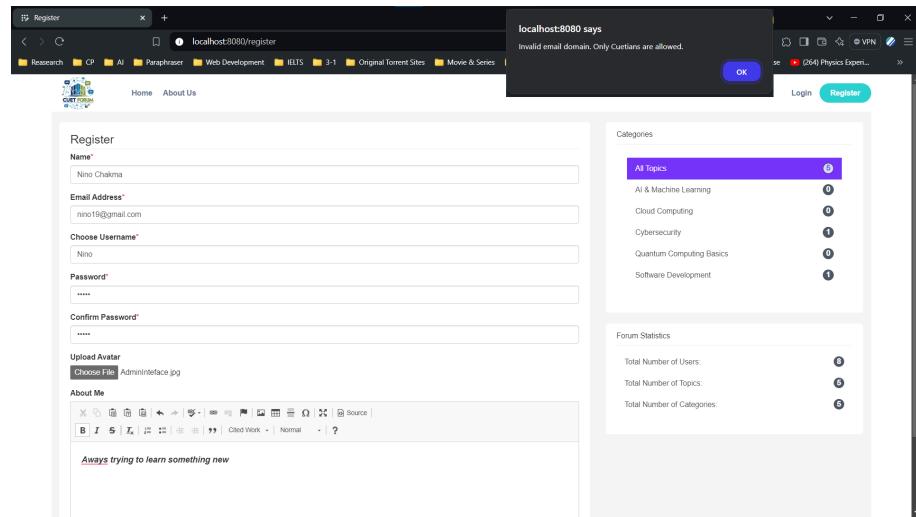


Figure 5.1: Invalid Credentials

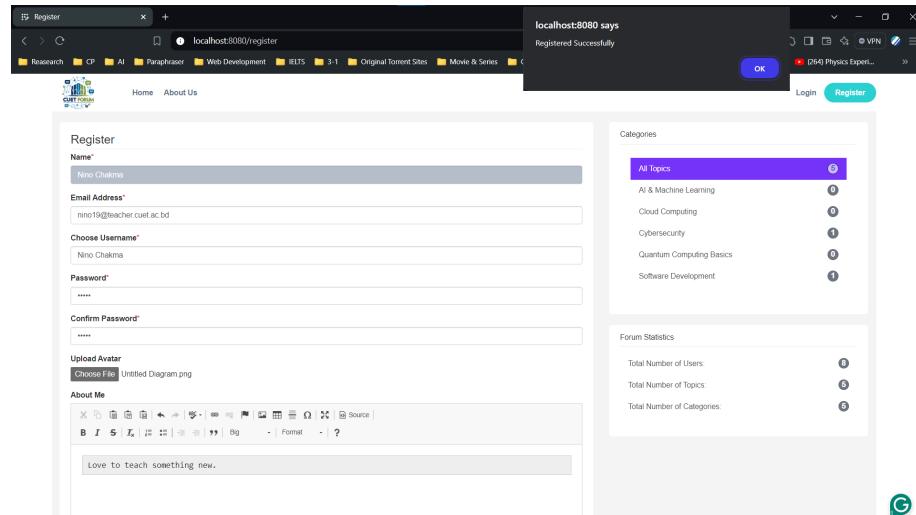


Figure 5.2: Successful Registration

5.1.2 Login:

Both User and Admin has to provide a valid email address and password as login credentials. The system authenticates the user credentials against a secure database.

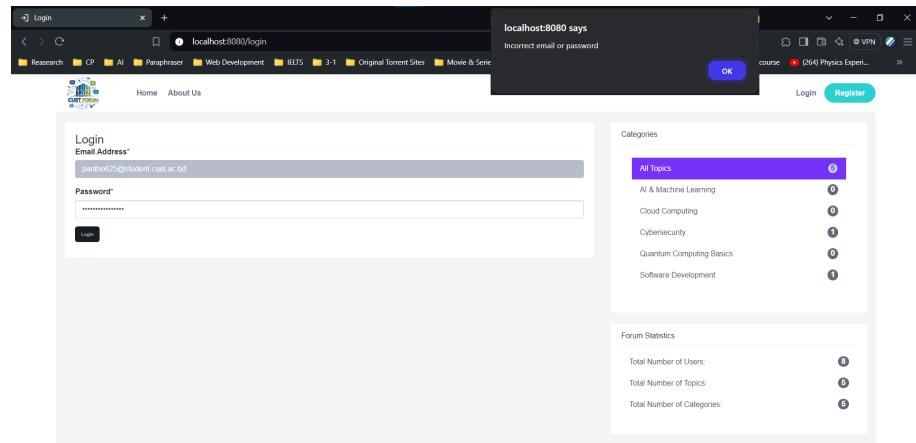


Figure 5.3: Invalid Credentials

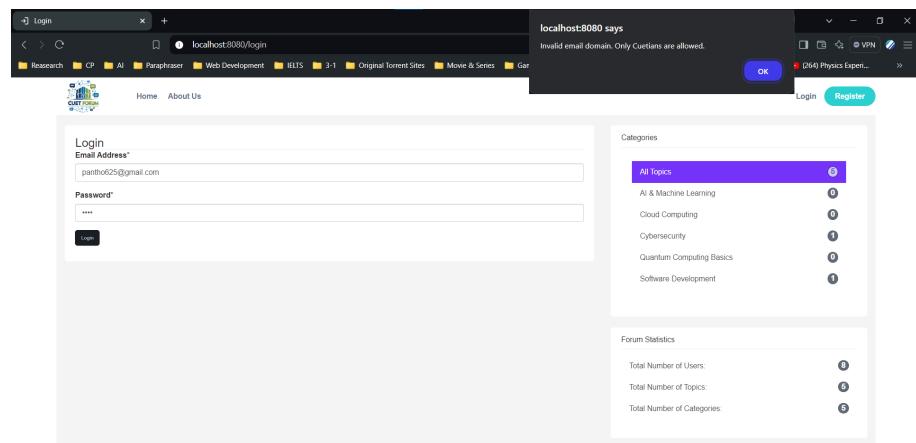


Figure 5.4: Invalid Email Format

To ensure proper access, the system only permits login for email addresses ending with `@cuet.ac.bd`. Users with email addresses outside this domain will be unable to log in. Based on the user role[Administrator or General User], the system redirects the user to the appropriate interface within the web application.

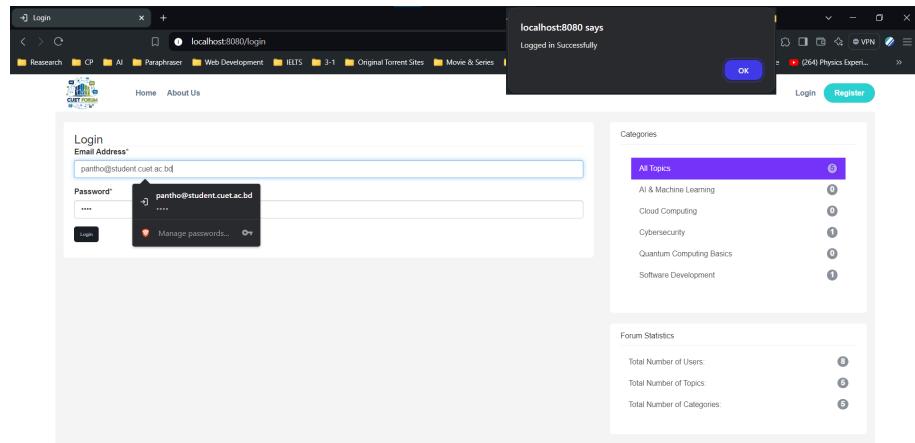


Figure 5.5: Successful Login

5.1.3 Homepage (Before Login)

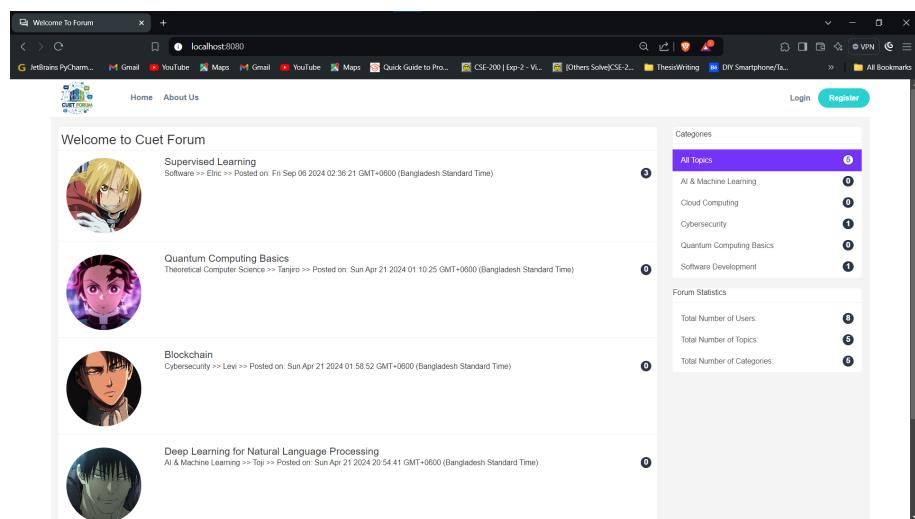


Figure 5.6: Home Page Before Login

Before logging in, users have limited access to the forum. They can:

- **View Topics:** Browse through existing topics and read the posts within them.
- **View Replies:** See the comments and responses made by other users.
- **View Forum Statistics:** Access information about the forum's overall activity, including the total number of users, topics, and replies.
- **Access Login and Registration Pages:** Navigate to the login and registration pages to gain full access to the forum's features.

5.1.4 Home Page(Post Login) :

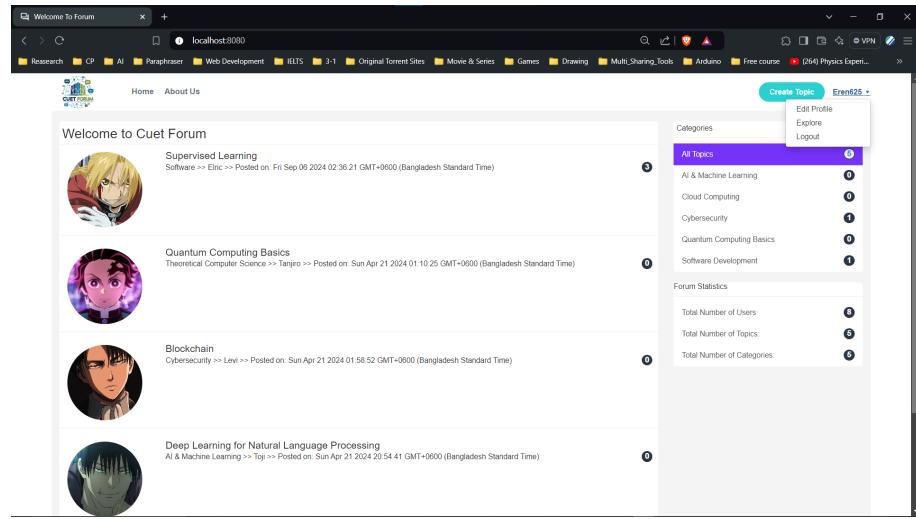


Figure 5.7: Home Page for General User

Upon successful login, general users are granted access to the following functionalities:

- **Topic Creation:** Users can initiate new discussion threads by creating topics within relevant categories.
- **Post Contributions:** Users can participate in existing topics by posting comments or responses to other users' contributions.
- **Profile Management:** Users can access and edit their personal profiles, including updating their information, changing their display name, or uploading a profile picture.

5.1.5 Topic Interface

Topic Visibility:

- All users, whether logged in or not, can view the topic and its associated posts. There are no restrictions on viewing the content, ensuring open access to all users.

The image shows a screenshot of a topic page from a platform. At the top right, there's a sidebar titled "Categories" with five items: "AI & Machine Learning", "Cloud Computing", "Cybersecurity", "Quantum Computing Basics", and "Software Development". The main content area has a header "Supervised Learning". It features a conversation between three users:

- Eric** (1 Post, Profile): "Hey everyone, I've been reading up on machine learning lately. Can someone explain the difference between supervised and unsupervised learning?"
- Tanjirō** (Profile): "Sure! In supervised learning, you train a model on labeled data, which means you have input-output pairs. The model learns to map inputs to outputs. For example, in image classification, you'd train the model with images labeled with their categories."
"And in unsupervised learning, you work with unlabeled data. The model tries to find patterns or structures in the data on its own. Clustering is a common technique here, like grouping customers into segments based on their purchase behavior."
- Ryomen** (Profile): "How do you know if your model is learning correctly in unsupervised learning?"
- Eren625** (Profile): "That's a good point, Ryomen. In supervised learning, you can evaluate your model's performance using metrics like accuracy or precision. But in unsupervised learning, it's not always as straightforward. You often need to use domain knowledge or visualization techniques to assess whether the model is finding meaningful patterns."

Figure 5.8: General View of Topic Page

Topic Creation and Updates:

- The user who initially created the topic (e.g., Elric) has the capability to update or delete their original post. This functionality is indicated by the update link located at `localhost:8080/update/topicid` next to their profile.

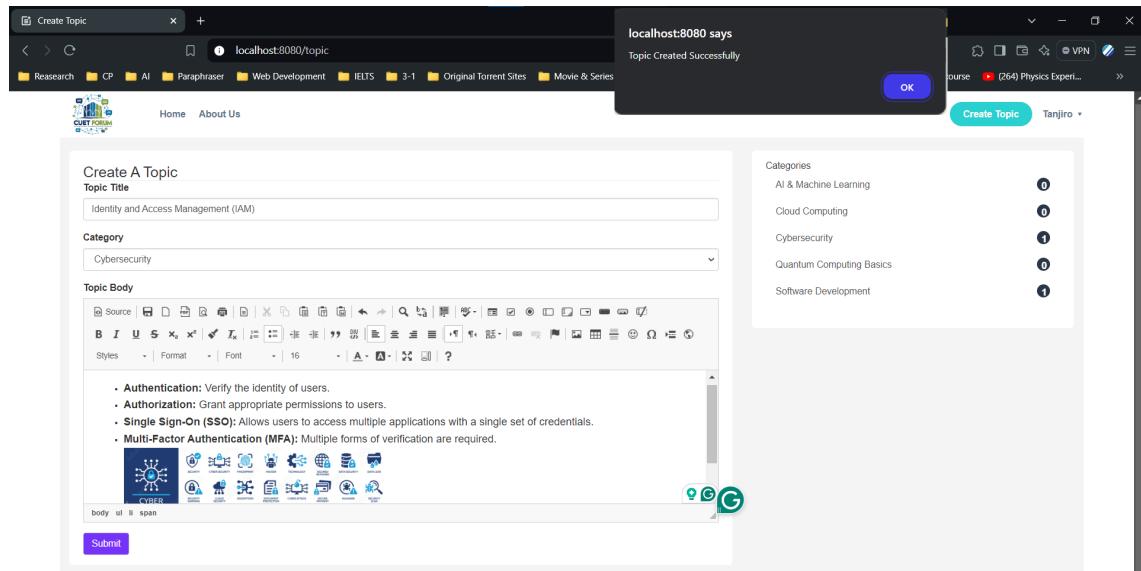


Figure 5.9: Creating a Topic

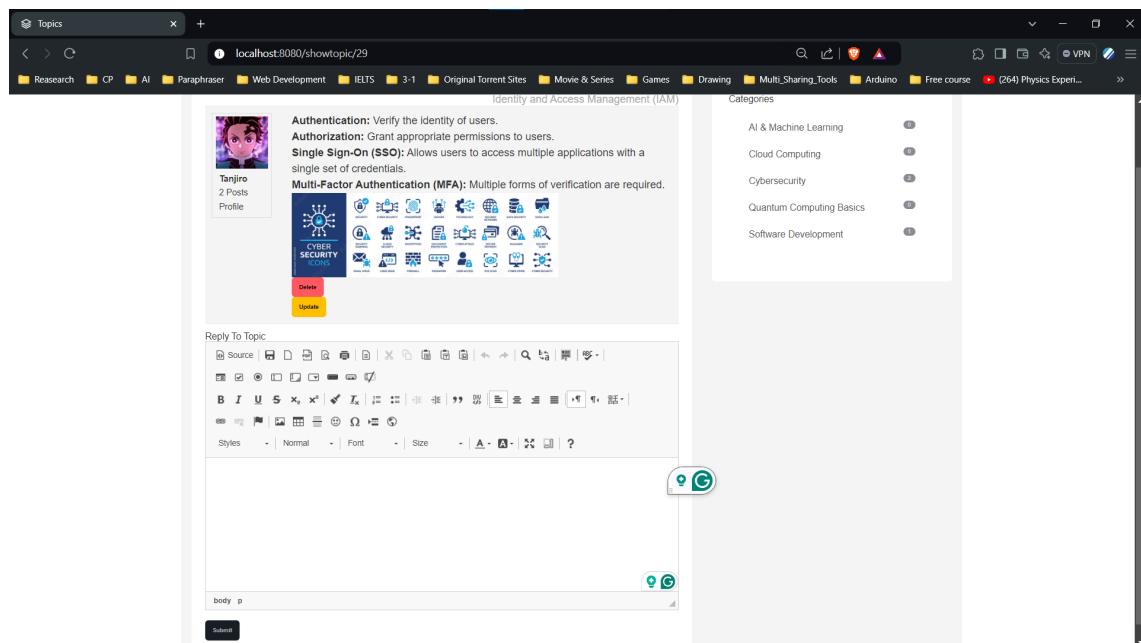


Figure 5.10: Topic added in Topic Page

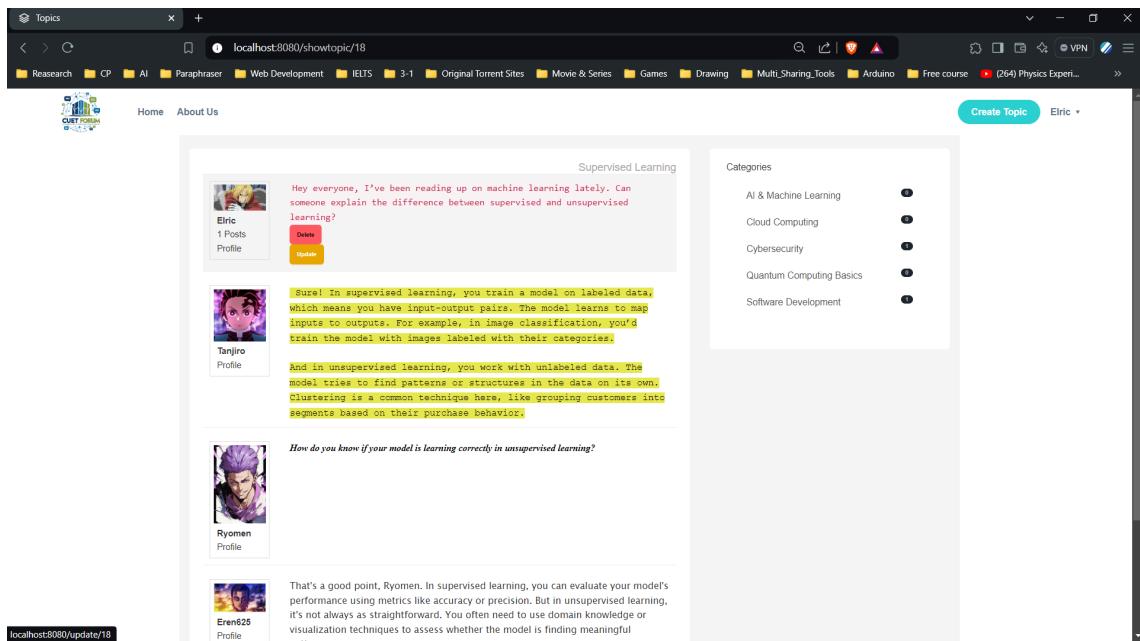


Figure 5.11: Updating a Topic

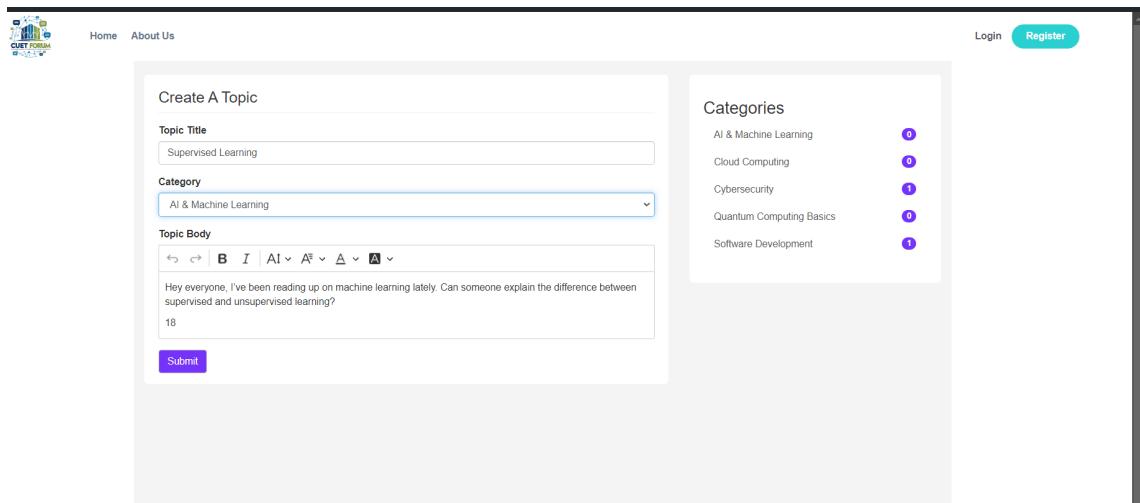


Figure 5.12: Successful Update

- Other users do not possess the authority to directly edit or remove the original post but can reply to it, thereby participating in the discussion.

Supervised Learning



Eric
1 Posts
Profile

Hey everyone, I've been reading up on machine learning lately. Can someone explain the difference between supervised and unsupervised learning?

Categories

- AI & Machine Learning
- Cloud Computing
- Cybersecurity
- Quantum Computing Basics
- Software Development



Tanjiro
Profile

Sure! In supervised learning, you train a model on labeled data, which means you have input-output pairs. The model learns to map inputs to outputs. For example, in image classification, you'd train the model with images labeled with their categories.

And in unsupervised learning, you work with unlabeled data. The model tries to find patterns or structures in the data on its own. Clustering is a common technique here, like grouping customers into segments based on their purchase behavior.



Ryomen
Profile

How do you know if your model is learning correctly in unsupervised learning?



Eren625
Profile

That's a good point, Ryomen. In supervised learning, you can evaluate your model's performance using metrics like accuracy or precision. But in unsupervised learning, it's not always as straightforward. You often need to use domain knowledge or visualization techniques to assess whether the model is finding meaningful patterns.

Figure 5.13: Chat Section

Replying to Posts:

- Any user, regardless of login status, can reply to a post within a topic. This policy facilitates open discussions and broad participation from all users.

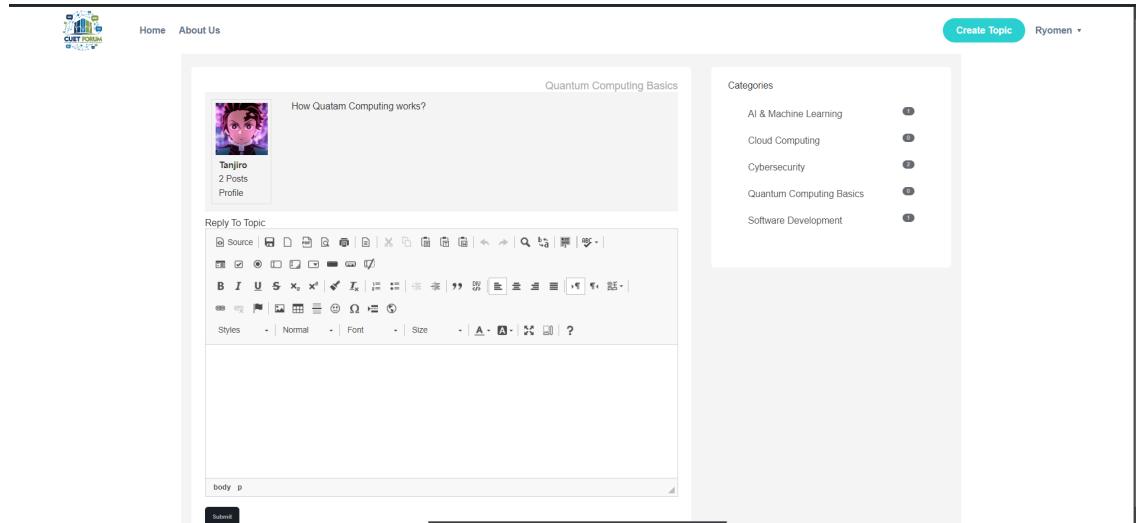


Figure 5.14: Reply Section

Reply Deletion:

- Users can delete their own replies to posts. However, other users do not have the capability to delete replies made by others. This ensures that only the author of a reply can manage their own content.

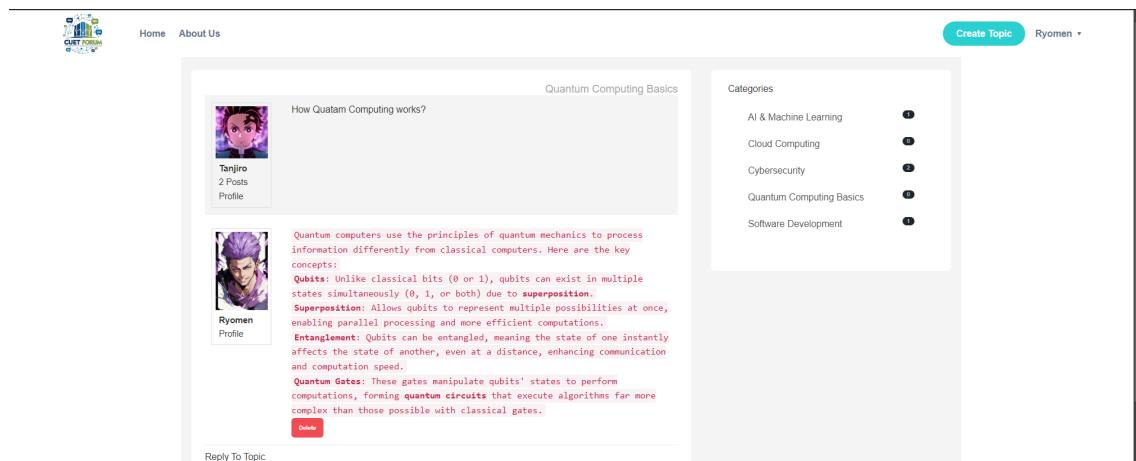


Figure 5.15: Reply Deletion

A teacher can view his or her profile like his total playlists, videos, likes, comments e.t.c. and he has the authority to update any information on his or her profile. He can change the profile photo and update information previously provided.

5.1.6 Profile Management

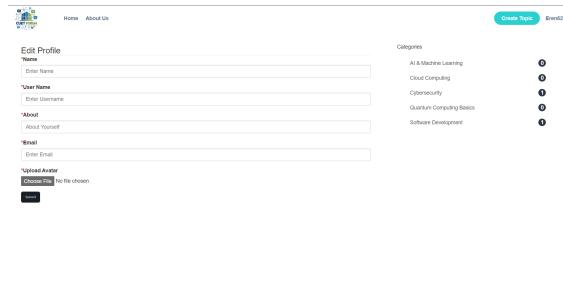


Figure 5.16: Editing Profile

Logged-in users have the ability to manage and edit their personal profiles. This includes the following functionalities:

- **Edit Profile Information:** Users can update their personal information, such as their name, email address, and about section.
- **Change Display Name:** Users have the option to change their display name, which is visible to other users in the forum.
- **Upload Profile Picture:** Users can upload a new profile picture to personalize their account.

These features allow users to keep their profiles up-to-date and ensure that their information is accurately reflected within the forum.

5.1.7 About Profile

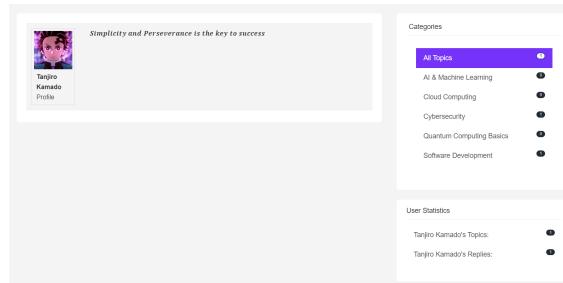


Figure 5.17: About Page

The "About Profile" page provides logged-in users with a comprehensive overview of their profile. This includes the following features:

- **Profile Information:** Users can view detailed information about their profile, including their name, email address, display name, and the "About" section where they can share more about themselves.
- **Number of Topics Created:** Users can see the total number of discussion topics they have created, providing insight into their contributions to the forum.

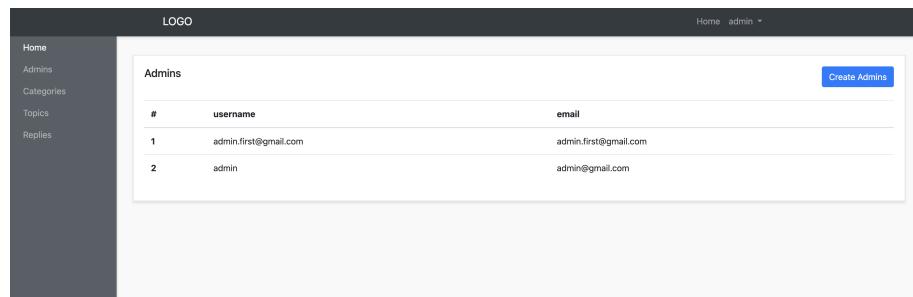
- **Number of Replies Created:** Users can view the total number of replies they have posted in various discussion topics, highlighting their level of engagement and participation in the forum.

These features allow users to easily access and review their profile information and activity within the forum.

5.2 Admin Interface Back-end Services

5.2.1 Admins

- **Admin Management:** Administrators can manage the admin list through the **Admins** section. This section displays all registered admins with options to create an admin.

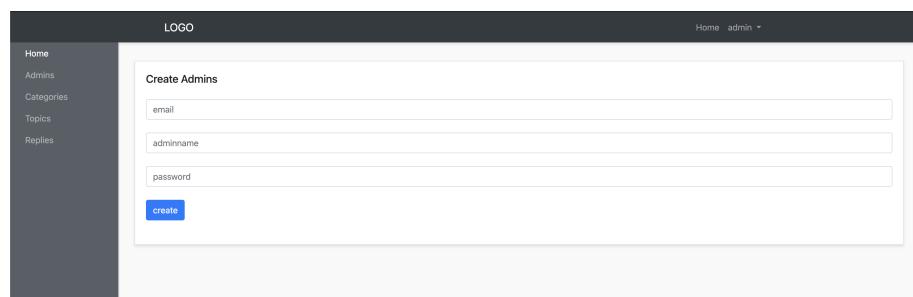


The screenshot shows a web-based admin interface. At the top, there's a dark header bar with the word "LOGO" on the left and "Home admin" on the right. Below this is a sidebar on the left containing links for "Home", "Admins", "Categories", "Topics", and "Replies". The main content area is titled "Admins" and contains a table with two rows of data. The columns are labeled "#", "username", and "email". Row 1 has values 1, "admin.first@gmail.com", and "admin.first@gmail.com". Row 2 has values 2, "admin", and "admin@gmail.com". In the top right corner of the main content area, there is a blue button labeled "Create Admins".

#	username	email
1	admin.first@gmail.com	admin.first@gmail.com
2	admin	admin@gmail.com

Figure 5.18: Admin Page

- **Create Admin:** The admin panel includes a form for creating new administrators, where an admin can enter details such as email, admin username, and password.



The screenshot shows a "Create Admins" form. It has three input fields: "email", "adminname", and "password". Below these fields is a blue "create" button. The form is set against a background of the same admin interface seen in Figure 5.18.

Figure 5.19: Admin Creation

- **Login:** Administrators must log in using their credentials to access the admin panel. The system verifies the admin's email and password before granting access.

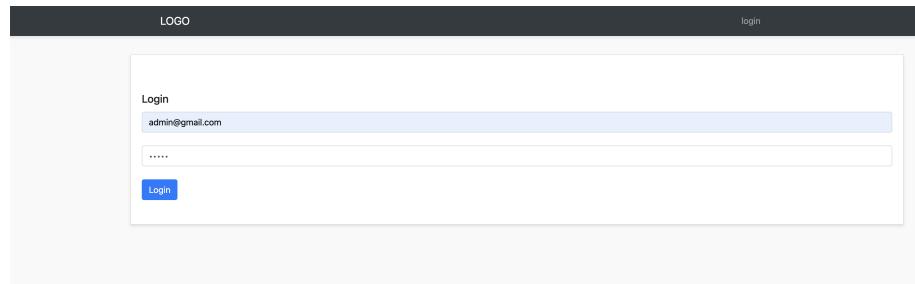


Figure 5.20: Admin Login

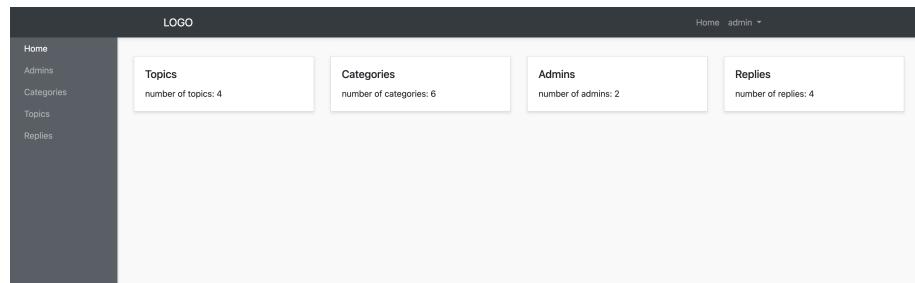


Figure 5.21: After successful login

- **Logout:** Admins can securely log out of the admin panel, ensuring session termination.

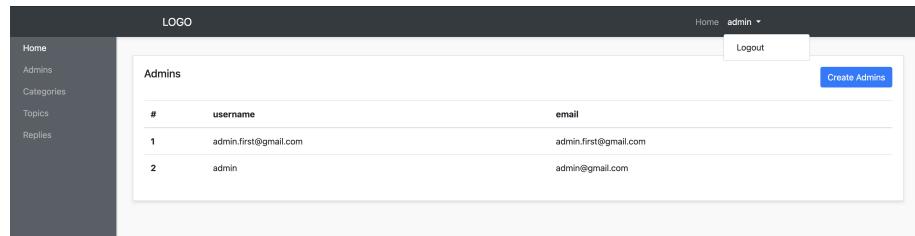


Figure 5.22: Admin Logout

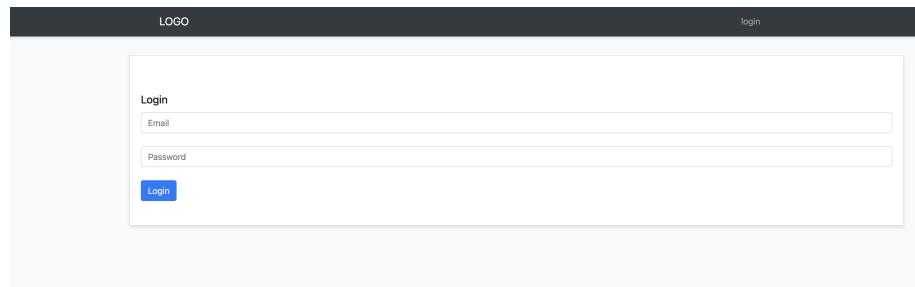


Figure 5.23: After successful logout

5.2.2 Category Management

- **Create Category:** Administrators can create new categories for the forum, which will then be available for users to post topics within these categories.

#	name	update	delete
1	Design	<button>Update</button>	<button>Delete</button>
2	Development	<button>Update</button>	<button>Delete</button>
3	Marketing	<button>Update</button>	<button>Delete</button>
4	SEO	<button>Update</button>	<button>Delete</button>
5	Hosting	<button>Update</button>	<button>Delete</button>
8	Artificial Intelligence	<button>Update</button>	<button>Delete</button>

Figure 5.24: Before New Categories Creation

#	name	update	delete
1	Design	<button>Update</button>	<button>Delete</button>
2	Development	<button>Update</button>	<button>Delete</button>
3	Marketing	<button>Update</button>	<button>Delete</button>
4	SEO	<button>Update</button>	<button>Delete</button>
5	Hosting	<button>Update</button>	<button>Delete</button>
8	Artificial Intelligence	<button>Update</button>	<button>Delete</button>
9	Robotics	<button>Update</button>	<button>Delete</button>

Figure 5.25: After New Categories Creation

- **Update Category:** The admin panel allows admins to update existing categories by changing the category name.

#	name	update	delete
1	Design	<button>Update</button>	<button>Delete</button>
2	Development	<button>Update</button>	<button>Delete</button>
3	Marketing	<button>Update</button>	<button>Delete</button>
4	SEO	<button>Update</button>	<button>Delete</button>
5	Hosting	<button>Update</button>	<button>Delete</button>
8	Artificial Intelligence	<button>Update</button>	<button>Delete</button>
9	Robotics	<button>Update</button>	<button>Delete</button>

Figure 5.26: Before Updating Categories (Design)

Categories			
#	name	update	delete
1	Architecture	<button>Update</button>	<button>Delete</button>
2	Development	<button>Update</button>	<button>Delete</button>
3	Marketing	<button>Update</button>	<button>Delete</button>
4	SEO	<button>Update</button>	<button>Delete</button>
5	Hosting	<button>Update</button>	<button>Delete</button>
8	Artificial Intelligence	<button>Update</button>	<button>Delete</button>
9	Robotics	<button>Update</button>	<button>Delete</button>

Figure 5.27: After Updating Categories (Architecture)

- **Delete Category:** Admins have the authority to delete categories. Once a category is deleted, it will no longer be accessible to users.

Categories			
#	name	update	delete
1	Architecture	<button>Update</button>	<button>Delete</button>
2	Development	<button>Update</button>	<button>Delete</button>
3	Marketing	<button>Update</button>	<button>Delete</button>
4	SEO	<button>Update</button>	<button>Delete</button>
5	Hosting	<button>Update</button>	<button>Delete</button>
8	Artificial Intelligence	<button>Update</button>	<button>Delete</button>

Figure 5.28: After Deleting Categories (Robotics)

5.2.3 Topic Management

- **View Topics:** Admins can view all topics posted in the forum. The topics are displayed in a table with details such as the title, category, and user who posted it.

Topics				
#	title	category	user	delete
1	post number one 1	Hosting	nino	<button>Delete</button>
6	three	SEO	nino	<button>Delete</button>
8	john's post one	SEO	john	<button>Delete</button>
9	post number 4	Design	nino	<button>Delete</button>

Figure 5.29: Showing All Posts

- **Delete Topic:** Administrators have the ability to delete any topic posted by users, which removes the topic from the forum.

The screenshot shows a dark-themed administrative interface for managing topics. On the left is a sidebar with links: Home, Admins, Categories, Topics, and Replies. The main area has a header 'TOPICS'. Below it is a table with columns: #, title, category, user, and delete. The data rows are:

#	title	category	user	delete
1	post number one 1	Hosting	nino	<button>delete</button>
6	three	SEO	nino	<button>delete</button>
8	john's post one	SEO	john	<button>delete</button>
9	post number 4	Design	nino	<button>delete</button>

Figure 5.30: Before Deleting Topic (Post Number 4)

This screenshot shows a user-facing forum interface with a purple header. The left sidebar includes 'Forum', 'Register', and 'Login'. The main content area has a 'Welcome to Forum' message and a 'A Simple Forum' title. Below is a list of topics:

- post number 4 (by nino, posted on 2024-04-05 16:05:00)
- post number one 1 (by nino, posted on 2024-04-03 18:47:03)
- three (by nino, posted on 2024-04-04 16:05:26)
- john's post one (by john, posted on 2024-04-04 16:33:36)

On the right, there are 'Categories' and 'Forum Statistics' sections.

Figure 5.31: Before Deleting Topic (User-End)

This screenshot shows the same administrative interface as Figure 5.30, but the table now only contains three rows of data, indicating that Post Number 4 has been removed.

#	title	category	user	delete
1	post number one 1	Hosting	nino	<button>delete</button>
6	three	SEO	nino	<button>delete</button>
8	john's post one	SEO	john	<button>delete</button>

Figure 5.32: After Deleting Topics (Post Number 4)

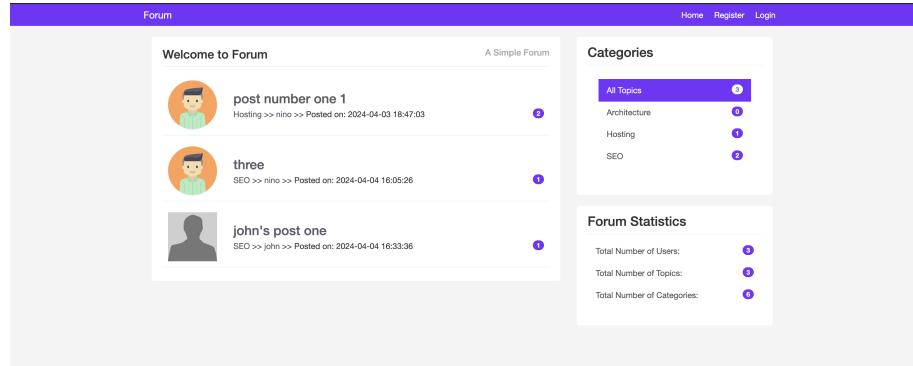


Figure 5.33: After Deleting Topic (User-End)

5.2.4 Reply Management

- **View Replies:** The reply management section allows administrators to view all replies made on various topics.

Replies				
#	reply	user name	go to topic	delete
1	this is a new reply	nino	go to topic	delete
2	On he decisively impression attachment friendship so if everything.	nino	go to topic	delete
5	why are u running	john	go to topic	delete

Figure 5.34: Showing Replies

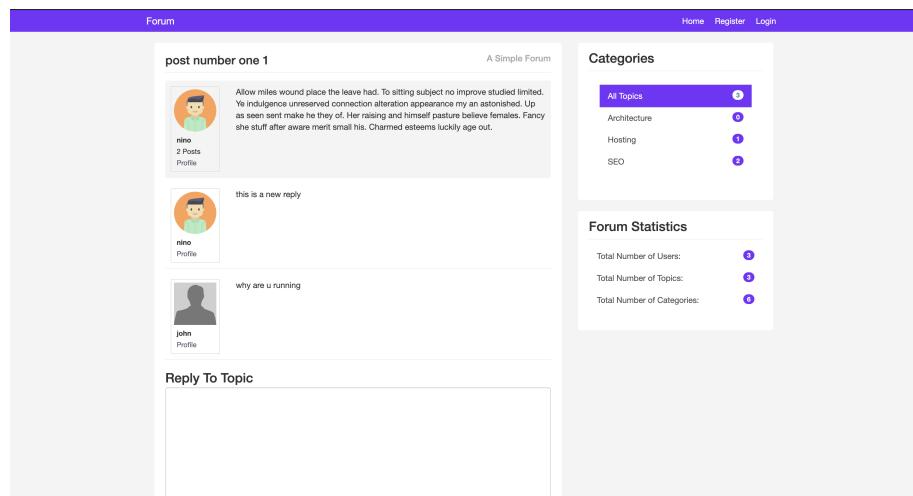


Figure 5.35: Redirecting to the post by clicking on the 'go to topic' button

- **Delete Reply:** Admins can delete any reply posted by users to maintain the quality of discussions on the forum.

The screenshot shows a dark-themed application interface with a sidebar on the left containing navigation links: Home, Admins, Categories, Topics, and Replies. The main content area has a header 'Replies' and a table listing four replies. The table columns are '#', 'reply', 'user name', 'go to topic', and 'delete'. The replies are:

#	reply	user name	go to topic	delete
1	this is a new reply	nino	<button>go to topic</button>	<button>delete</button>
2	Oh he decisively impression attachment friendship so if everything.	nino	<button>go to topic</button>	<button>delete</button>
5	why are u running	john	<button>go to topic</button>	<button>delete</button>
8	idiot asking for help lmao	nino	<button>go to topic</button>	<button>delete</button>

Figure 5.36: Before Deleting Replies

This screenshot is identical to Figure 5.36, showing the same list of four replies in the 'Replies' section. The 'delete' button for the fifth reply ('why are u running') has been clicked, resulting in its removal from the list.

Figure 5.37: After Deleting Replies

5.2.5 Profile Management

- **View Profile:** Admins can view their own profile with details such as the total number of topics created and replies made.

The screenshot shows a dark-themed application interface with a sidebar on the left containing navigation links: Home, Admins, Categories, Topics, and Replies. The main content area displays four summary boxes: 'Topics' (number of topics: 4), 'Categories' (number of categories: 6), 'Admins' (number of admins: 2), and 'Replies' (number of replies: 4).

Figure 5.38: Showing Information

Chapter 6

Conclusion

The CUET Forum project provides a robust and user-friendly platform tailored for students, faculty, and administrators associated with CUET. The system allows seamless interaction through topic creation, discussion, and reply features while ensuring a secure registration and login process restricted to valid CUET email addresses. Both user and admin interfaces are designed with clear, accessible features for easy navigation and management.

The User Interface Back-end Services empower general users with the ability to engage in discussions, manage their profiles, and interact with the community. Meanwhile, the Admin Interface Back-end Services give administrators full control over the forum, including user management, category organization, and content moderation. This division of roles ensures the platform remains organized, secure, and functional for all members.

Overall, the CUET Forum is well-positioned to foster collaboration, knowledge sharing, and community building, enhancing communication within the university while maintaining a secure and moderated environment. Future improvements could include adding more customization options for users, enhancing security measures, and integrating additional features to expand the platform's scope and usability.