Development of a CRUD Web Application

Nguyen Hoang Lan

University of Greenwich

Email: lannhgcd220043@fpt.edu.vn

Contents

Figures Table:

1. Introduction:

In this report, I present the development process of a web-based CRUD system aimed at facilitating student interactions for coursework assistance. The system provides a platform for students to post questions, share insights, and engage with peers in a collaborative learning environment. The report outlines the design choices, technologies utilized, legal considerations, and testing procedures undertaken during the development process.

2. System Development:

Design of pages:

- Home:

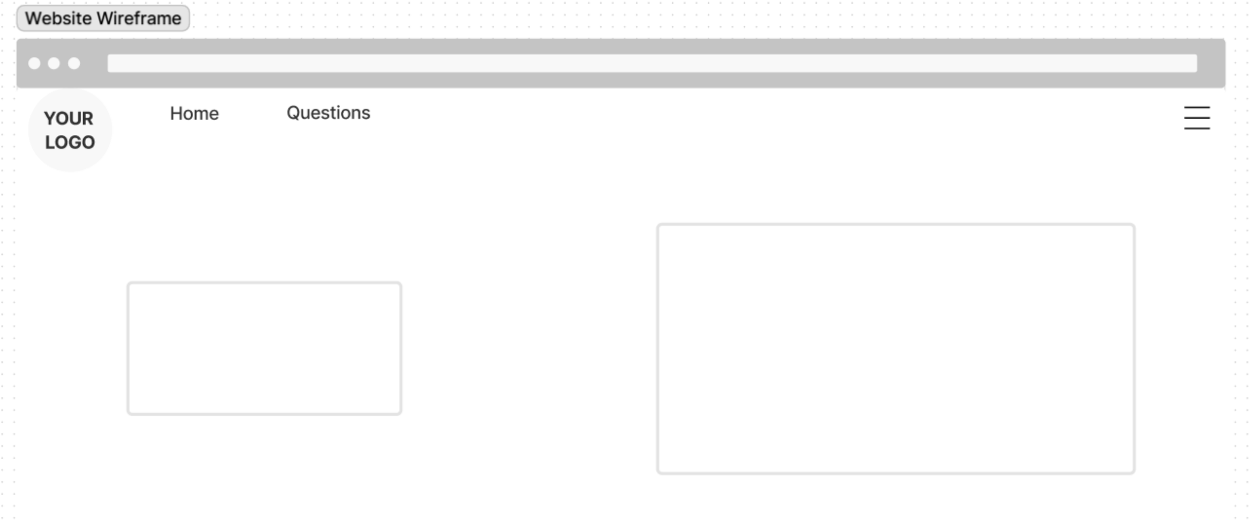


Figure 1: Home wireframe

- Questions:

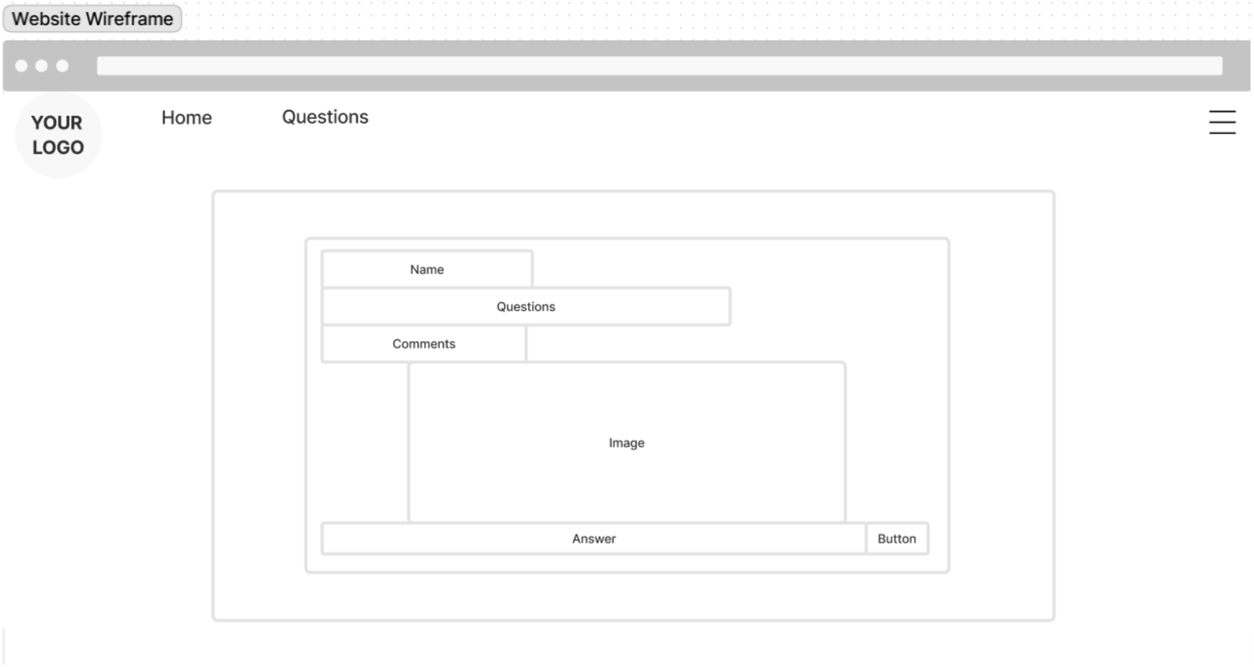


Figure 2: Questions page wireframe

- Detail question page:

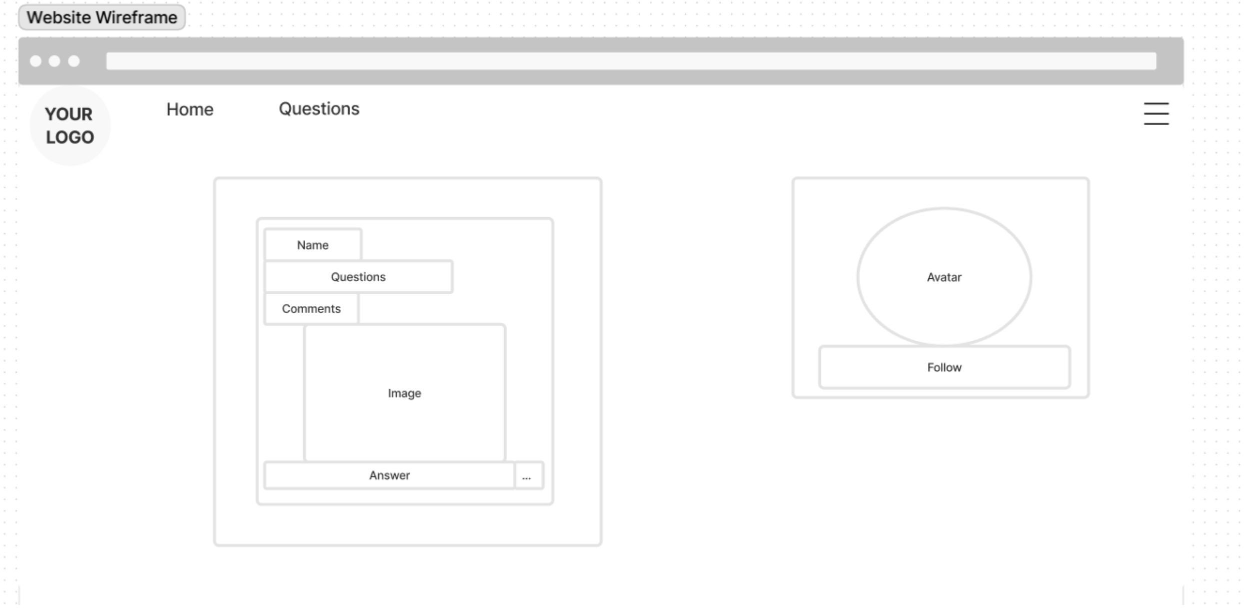
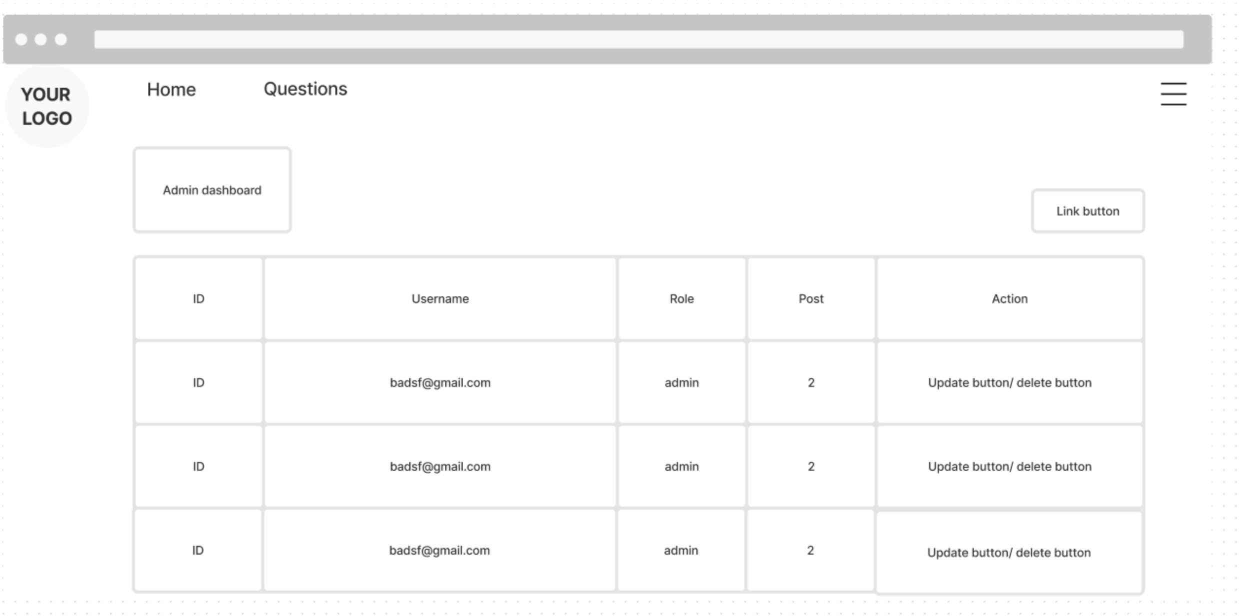


Figure 3: Detail wireframe

- Admin dashboard:



ID	Username	Role	Post	Action
ID	badsf@gmail.com	admin	2	Update button/ delete button
ID	badsf@gmail.com	admin	2	Update button/ delete button
ID	badsf@gmail.com	admin	2	Update button/ delete button

Figure 4: Admin wireframe

- Login:

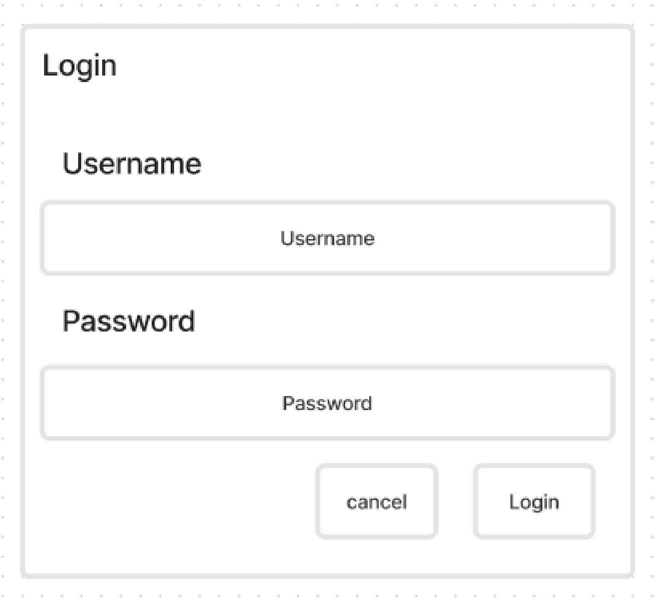


Figure 5: Login wireframe

- Register:

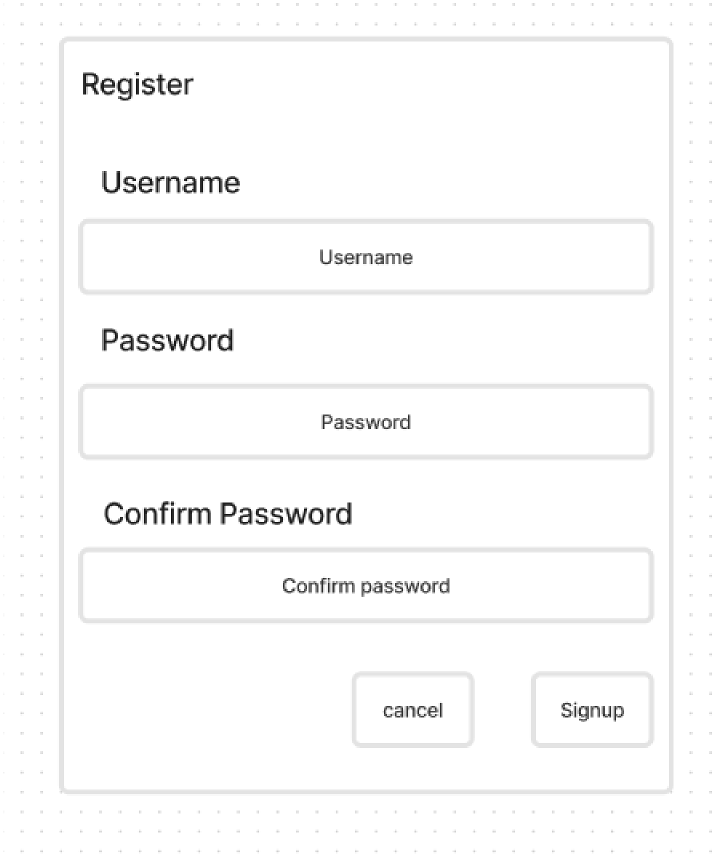


Figure 6: Register wireframe

- Email form:

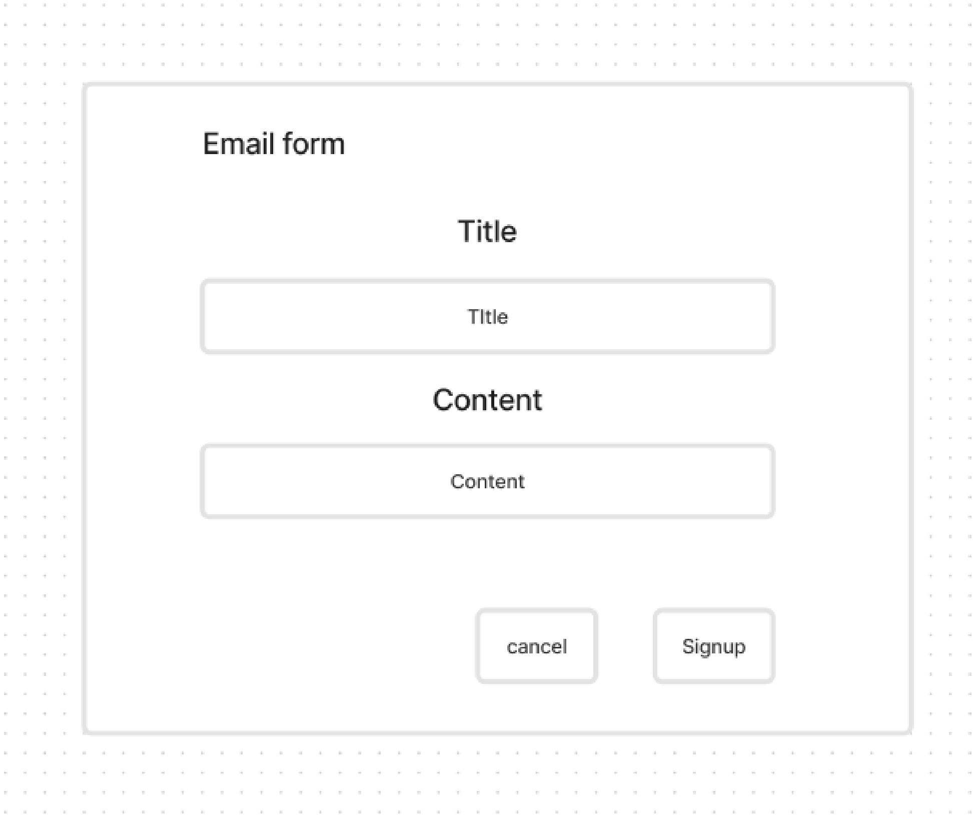
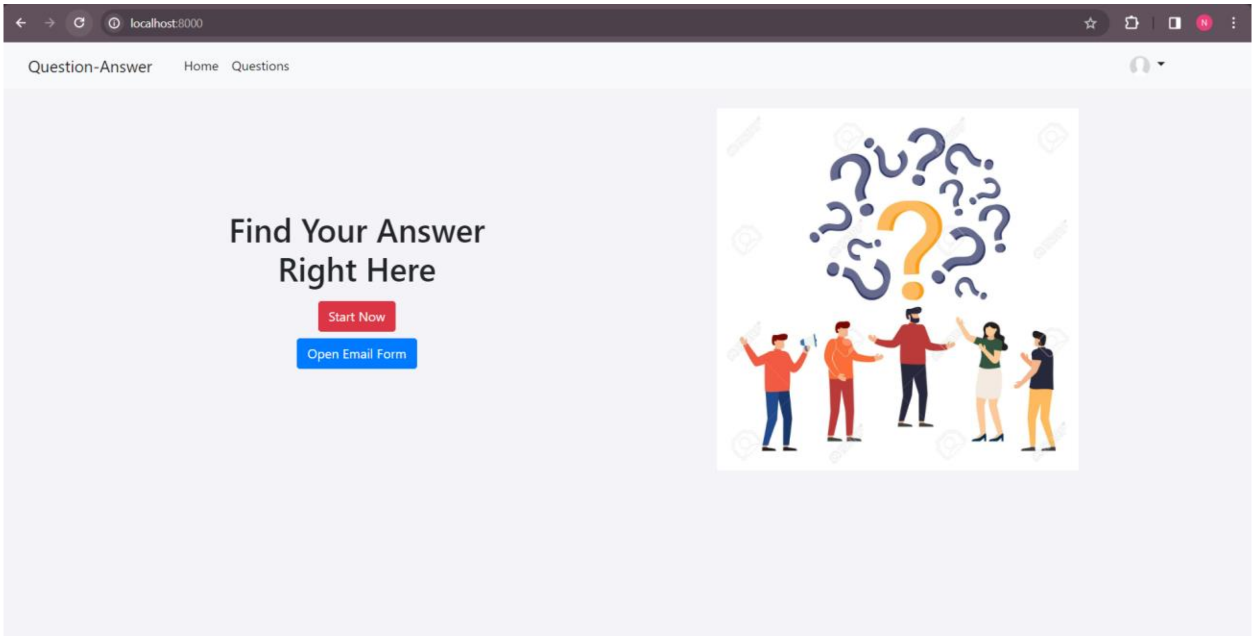


Figure 7: Sending email wireframe

Evidence of Pages:

The user interface has been crafted for effortless navigation and user-friendliness, embracing responsive design principles with Bootstrap for seamless accessibility across various devices. Pages encompass:

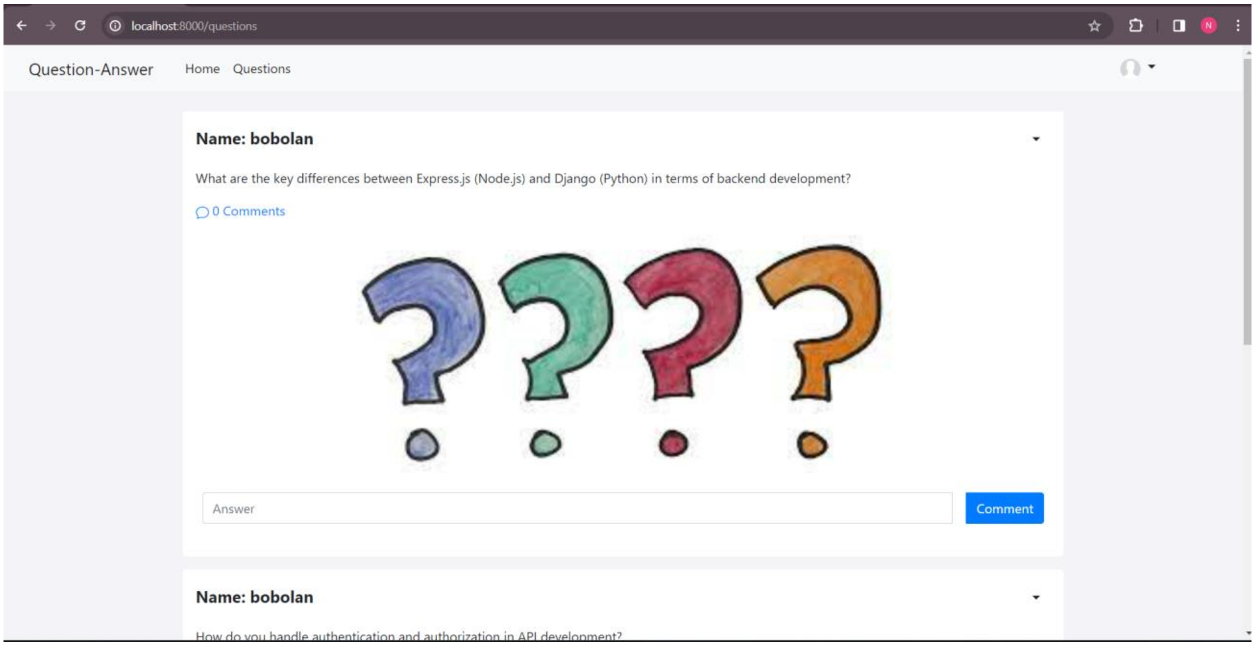
- Home: Featuring a welcoming quote and two interactive buttons. The first directs users to create a question page, while the second facilitates contacting the admin via email.



Find Your Answer Right Here

Figure 8: Home page

- Questions: Presents a comprehensive list of user queries/posts, each comprising the user's name, the question itself, the number of comments (linked for direct access), an input field for submitting comments, and options for post management such as updating or deleting.



Name: bobolan

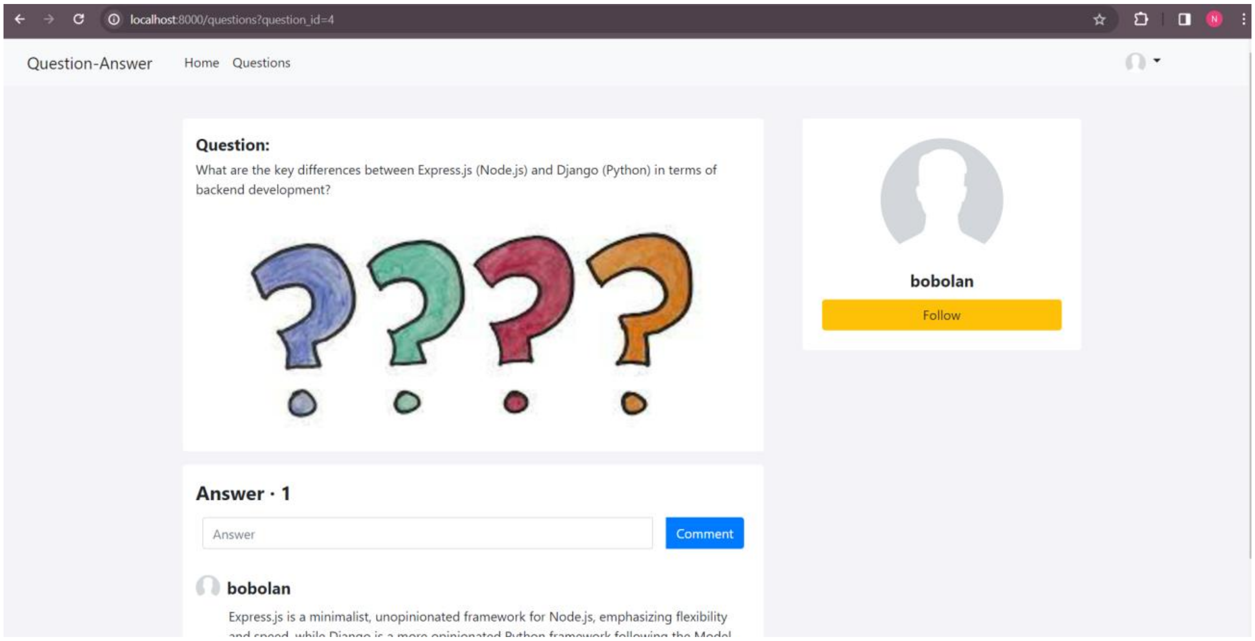
What are the key differences between Express.js (Node.js) and Django (Python) in terms of backend development?

Name: bobolan

How do you handle authentication and authorization in API development?

Figure 9: Questions page

- Individual Question Page: Provides details of a specific question including the user's name, the question, a comment section, and all associated comments.



Question:

What are the key differences between Express.js (Node.js) and Django (Python) in terms of backend development?

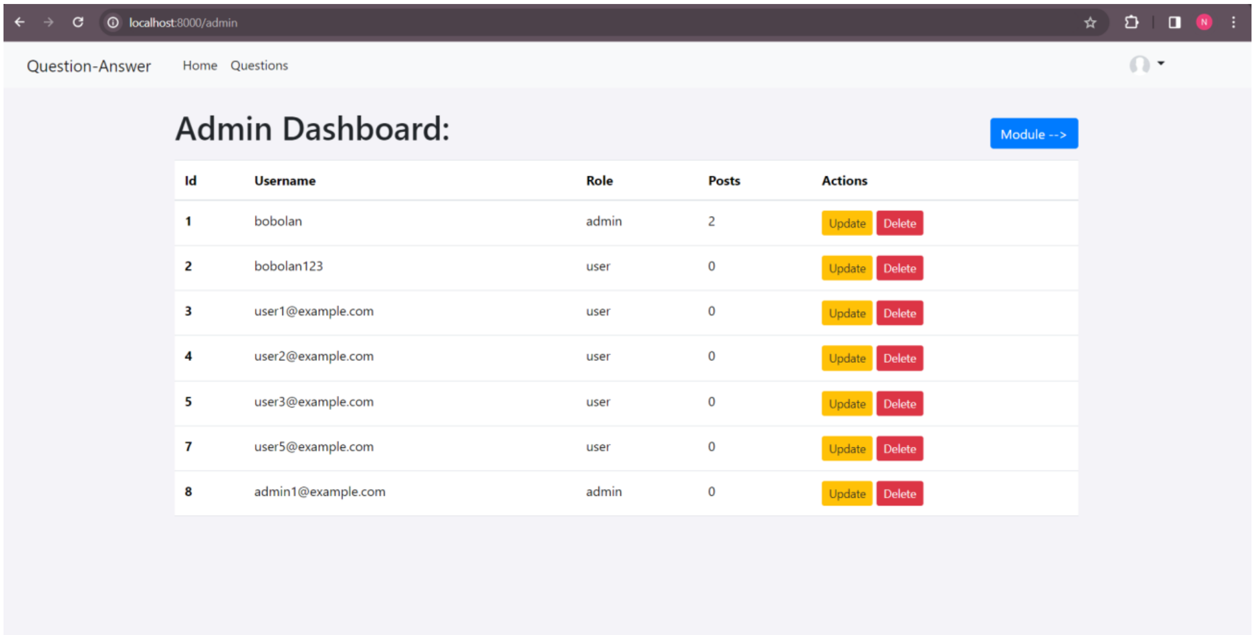
Answer · 1

bobolan

Express.js is a minimalist, unopinionated framework for Node.js, emphasizing flexibility and speed, while Django is a more opinionated Python framework following the Model

Figure 10: Detail page

- Admin Panel: Reserved for administrative tasks, including user management via a table display. Additionally, it offers a shortcut to the module management page.

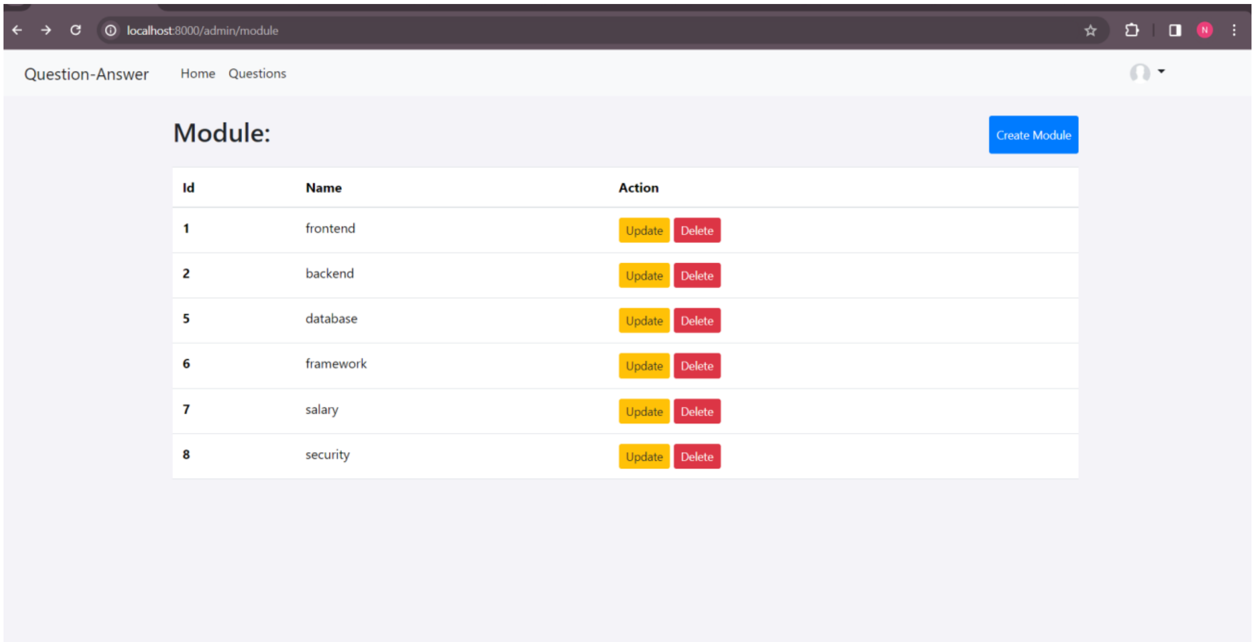


Admin Dashboard:

Id	Username	Role	Posts	Actions
1	bobolan	admin	2	Update Delete
2	bobolan123	user	0	Update Delete
3	user1@example.com	user	0	Update Delete
4	user2@example.com	user	0	Update Delete
5	user3@example.com	user	0	Update Delete
7	user5@example.com	user	0	Update Delete
8	admin1@example.com	admin	0	Update Delete

Figure 11: Admin dashboard page

- Module Management: Displays a table of modules and provides administrative functionalities for creating, reading, updating, and deleting modules.

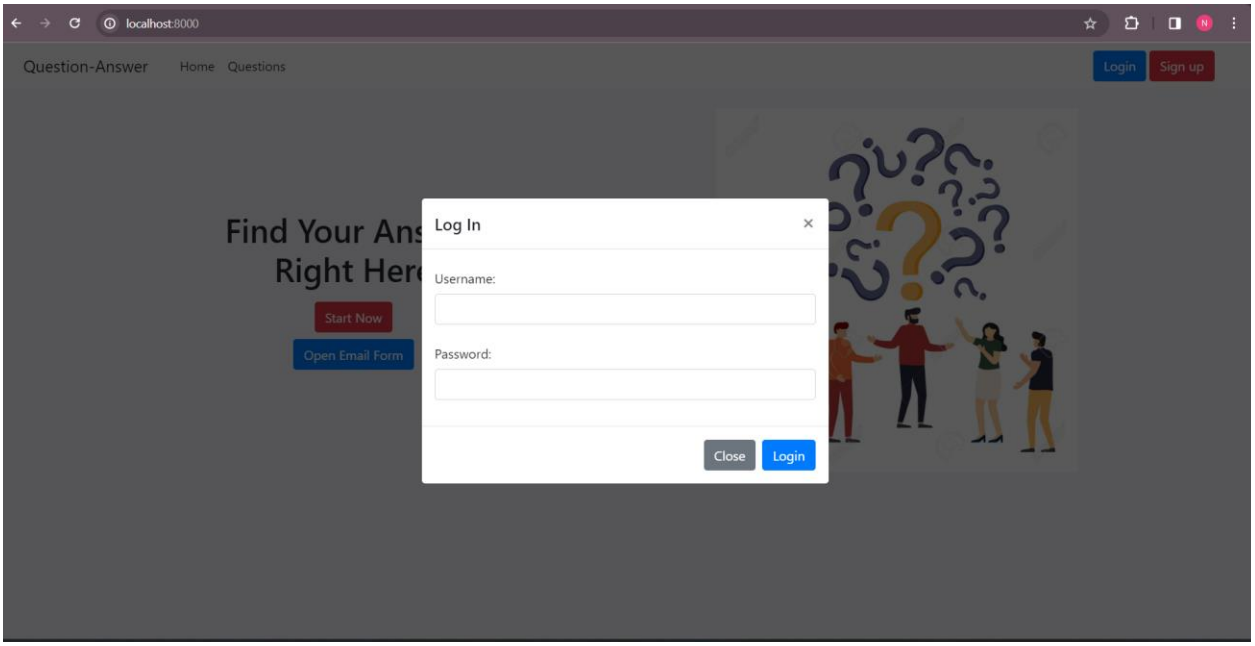


Module:

Id	Name	Action
1	frontend	Update Delete
2	backend	Update Delete
5	database	Update Delete
6	framework	Update Delete
7	salary	Update Delete
8	security	Update Delete

Figure 12: Module page

- Modal Buttons:
  - Login: Input fields for username and password, with a submit button.



Log In

Username:

Password:

Figure 13: Login model

  - Signup: Input fields for username, password, and password confirmation, along with a signup button.

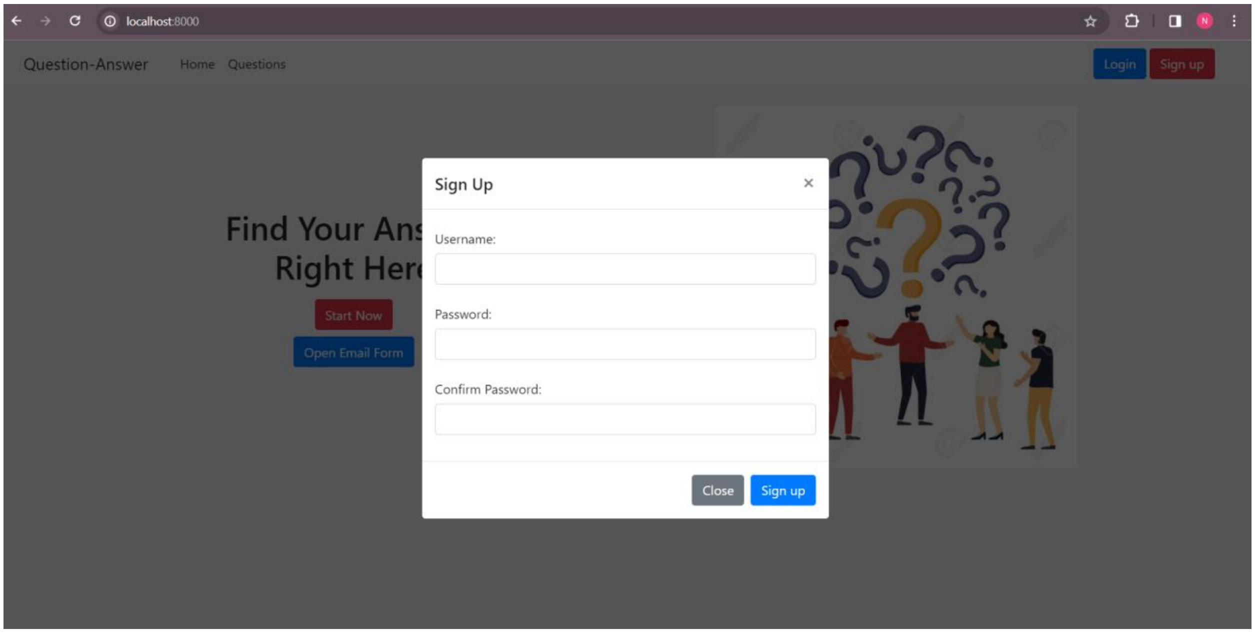


Figure 14: Register model

- ○ Open Email Form: Allows composing an email with a title and content.

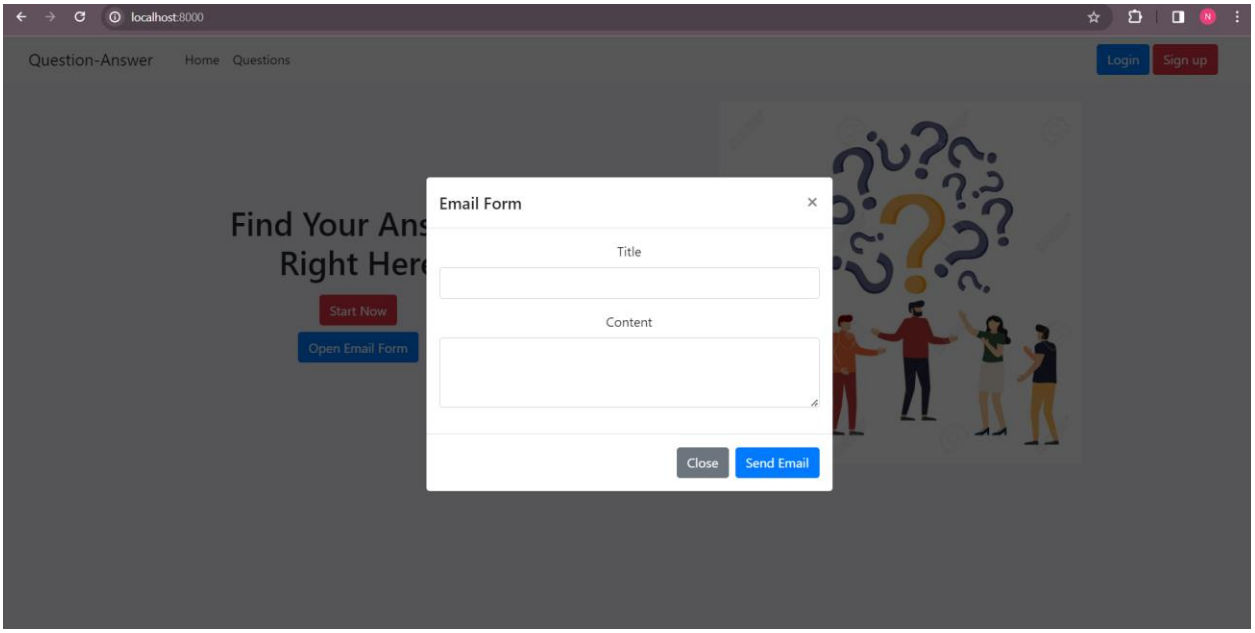
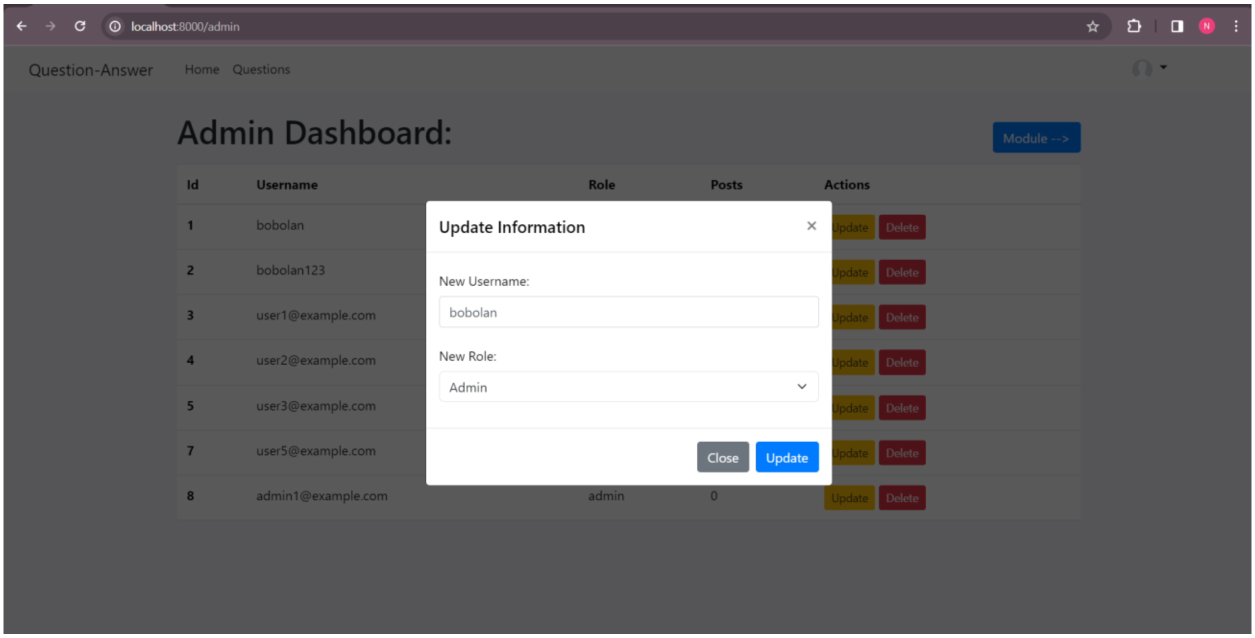


Figure 15: Email model

- ○ Update User: Enables modifying the username, role, and provides options to close or update.



Admin Dashboard:

Id	Username	Role	Posts	Actions
1	bobolan			Update Delete
2	bobolan123			Update Delete
3	user1@example.com			Update Delete
4	user2@example.com			Update Delete
5	user3@example.com			Update Delete
7	user5@example.com			Update Delete
8	admin1@example.com	admin	0	Update Delete

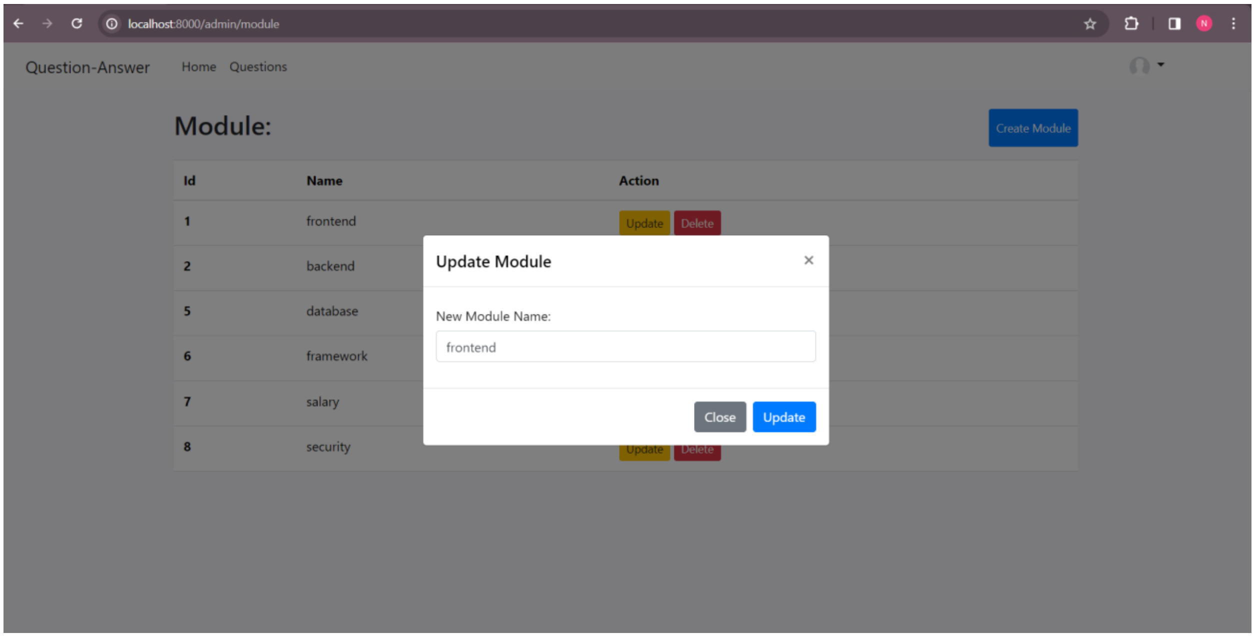
Update Information

New Username: bobolan

New Role: Admin

Figure 16: Update user model

○ Update Module: Allows updating the module name. Navigation Structure: A navigation bar has been implemented featuring the website's name "Question-Answer," along with links to the home, questions, and, for non-logged-in users, login and signup buttons. For authenticated users, a dropdown menu offers access to the admin panel if the user's role is admin, along with a logout option.



Module:

Id	Name	Action
1	frontend	Update Delete
2	backend	
5	database	
6	framework	
7	salary	
8	security	Update Delete

Update Module

New Module Name: frontend

Figure 17: Update module model

-

Data Diagram:

The database schema includes tables for users, questions, modules, answers and emails.

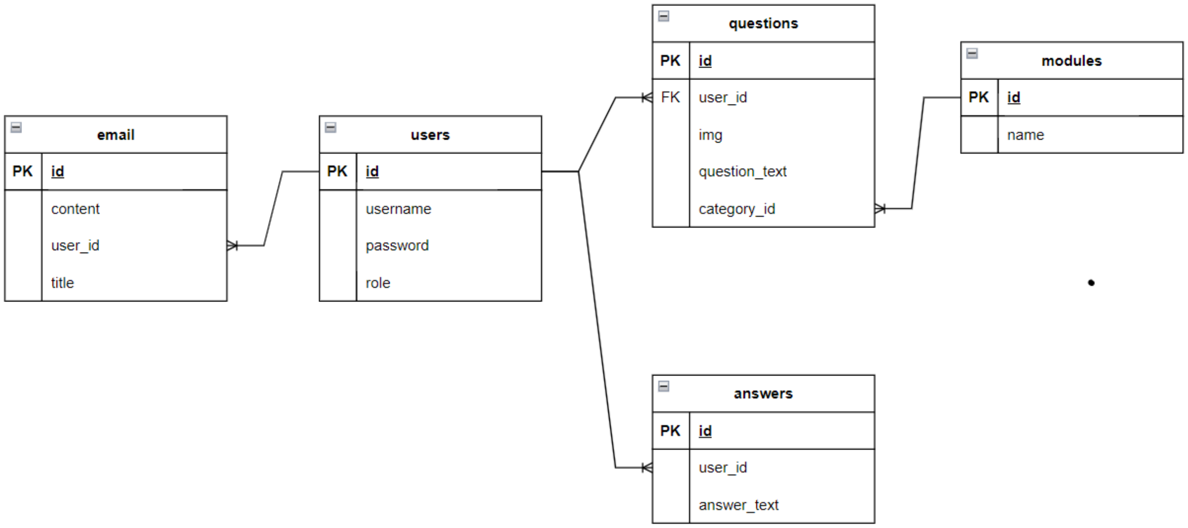


Figure 18: ERD

3. Technologies Used:

In this question-answer website, I applied several technologies to create the website which matches the security, functionality, user-friendly and accessibility to many devices. The primary technologies used include:

1. **HTML**: HTML is the backbone for building the whole website. Using tag to create a dynamic and interactive user interface.

2. **PHP PDO (PHP Data Objects)**: PHP PDO was chosen as the interface for accessing and manipulating the MySQL database.

3. **MySQL**: The relational database used by Xampp to store and manage the application's data was MySQL. Its powerful capabilities and PHP PDO compatibility made data retrieval and storing effective.

4. **CSS Frameworks (Bootstrap):** I use Bootstrap to make sure the design is responsive and to speed up the styling process. I was able to create a layout that was both aesthetically pleasing and responsive across a wide range of devices and screen sizes by using established CSS classes and components.

5. **PHPMailer:** PHPMailer provided a convenient and reliable way to send email messages from the application with simplified configuration, security and compatibility.

4. Legal, Social, and Ethical Considerations:

1. **Accessibility Standards**: Legal accessibility guidelines were followed in the development and implementation of the application. This promotes inclusion and equitable access to information by guaranteeing that people with impairments may use the application.
2. **Data Protection Regulations:** Regulations pertaining to data protection were carefully reviewed and put into practice. Steps were taken to guarantee that the program handles personal data collected in a transparent and safe manner.
3. **Personal Data Storage**: Sensitive information, like authorization and passwords, are protected with suitable access controls and encryption when storing personal data within the program. When required, user consent is sought, and users are informed in full of data processing procedures.
4. **User Awareness**: Users are made aware of how the application gathers, stores, and uses their personal information. To guarantee openness and informed consent, users are given clear and simple terms of service and privacy policies.

5. Overview of the System:

The system has an easy-to-use interface that makes it easier to connect with important features like asking inquiries, managing user accounts, and getting in touch with the administrator. Carefully chosen annotations on screenshots demonstrate how the system complies with functional requirements for module assignment and CRUD (Create, Read, Update, Delete) activities.

CRUD Questions: Users can effortlessly create and update new posts.

1. **Managing User Accounts**: The system provides an easy-to-use method for managing user accounts, giving administrators the resources they need to effectively supervise user accounts. The administrative interface is demonstrated through annotated screenshots, which offer choices for modifying or eliminating user accounts as needed, as well as clear insight into user accounts. Robust control over user administration within the system is ensured by this functionality.
2. **Contacting the Administrator**: Through a specific contact form, users can get in touch with the administrator directly, providing a direct channel of communication for questions or help. The contact form's user-friendly interface, which includes fields for name, email address, subject, and message, is demonstrated in annotated screenshots. Users receive confirmation that their message was forwarded to the administrator successfully after submitting it.

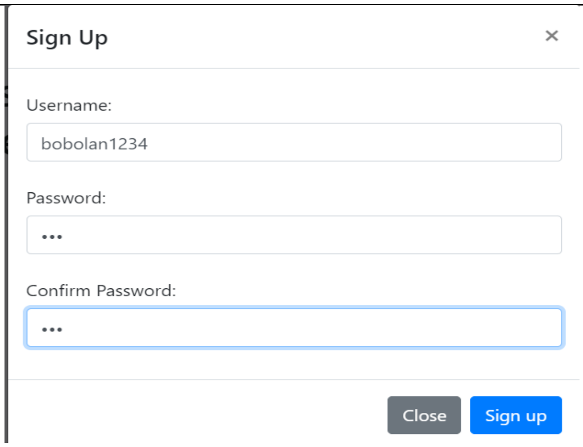
3. **Module Assignment**: The technology makes it easy for queries and postings to be assigned to modules, which improves the platform's organization and classification. The module assignment feature is shown in annotated screenshots, giving users a simplified way to choose and apply modules to their posts. This function makes sure that material is managed within the system with efficiency and clarity.

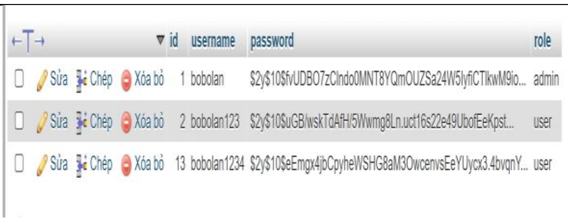
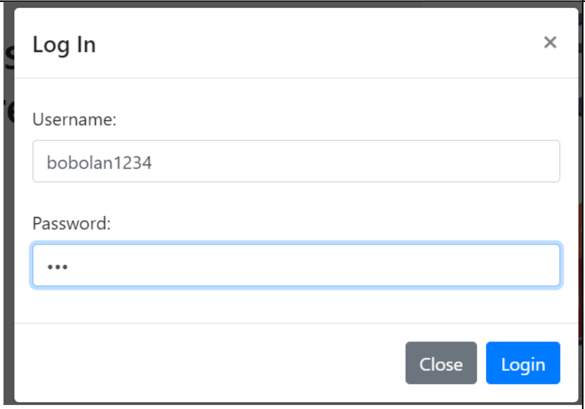
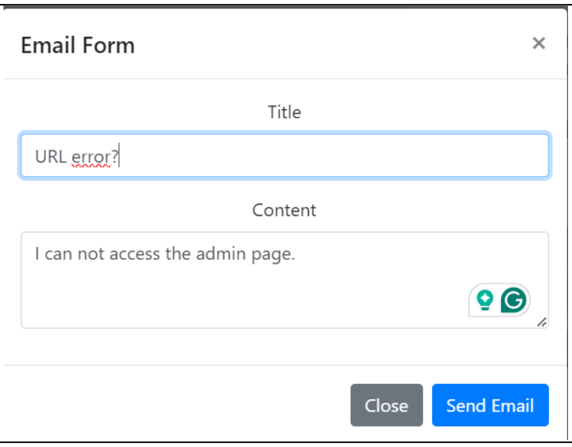
4. **Login**: An easy-to-use login screen with visible input areas for the password and username is shown to users. Screenshots with annotations highlight the user-friendly layout and display the login form and submit button. After entering their login credentials, individuals can verify their identification in the system and safely access their accounts.

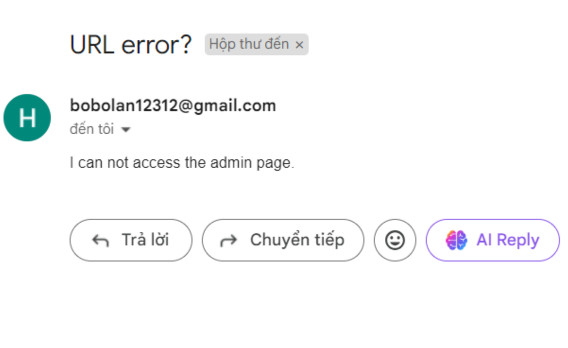
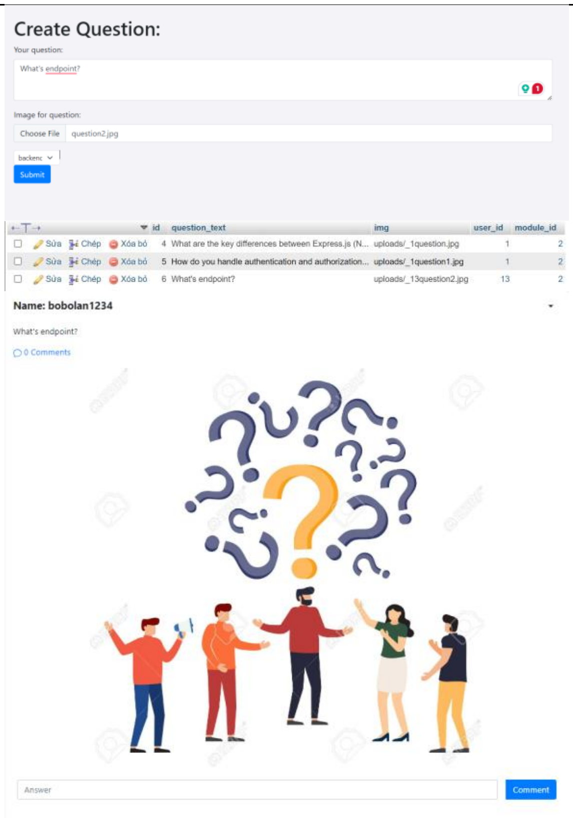
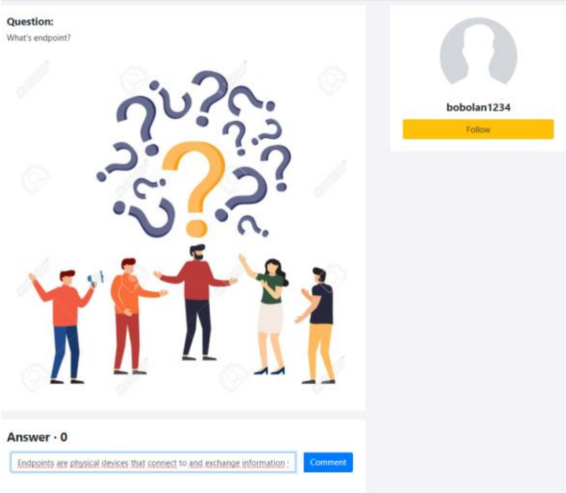
5. **Signup**: The system's shortened signup process makes it simple for new users to register an account. Screenshots with annotations show the signup interface, which has fields for entering login credentials such username, email address, and password. Users can also re-enter their password for verification in a confirmation field. Users can start the process of creating an account by clicking on the conspicuous signup button. Users are automatically led to the login page to access their newly formed account after successfully submitting the signup form and receiving confirmation that their account has been created.

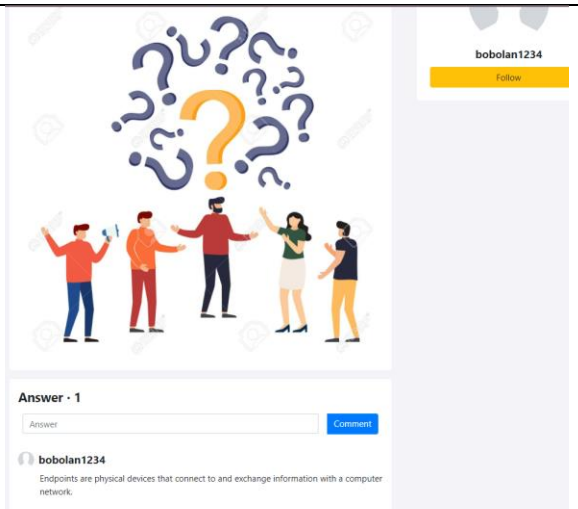
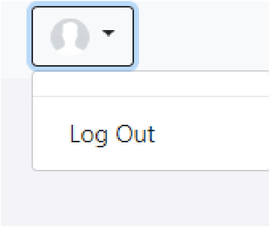
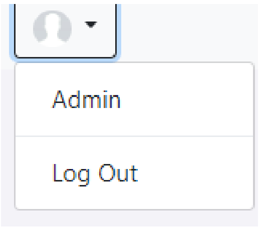
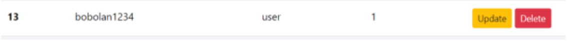
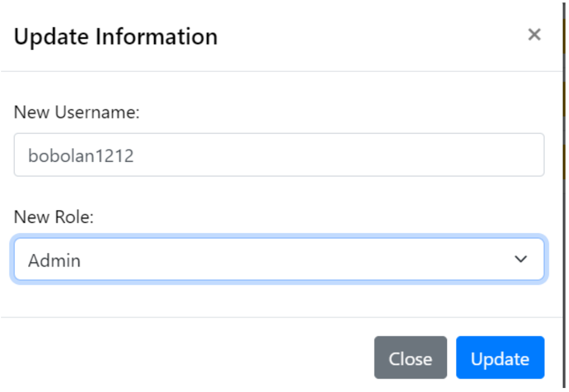
6. Testing Schedule and Evaluation:

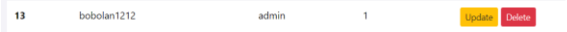
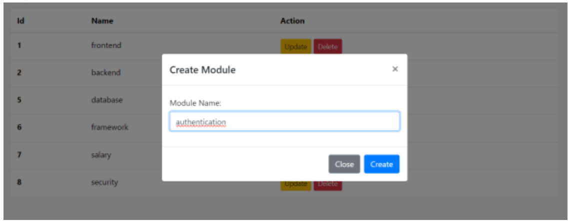
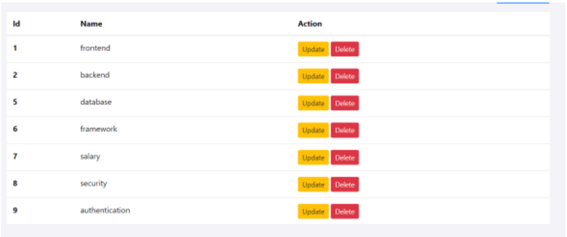
To guarantee the web application's dependability and functionality, a systematic testing plan was put in place. Input validation, CRUD operations, user authentication, and error handling were all tested. After the system was evaluated, some small faults were found and fixed quickly to increase usability.

Case	Description	Expected result	Test result	Evidence	Conclusion
Sign in	User will full fill username, password, confirm pass word to create new account. Username must not be taken and confirm password have to be the same with password.	Inform result to user, store new user account in table user.	Success to create new user.	 Sign Up Username: bobolan1234 Password: ••• Confirm Password: ••• 	Pass

				 id | username | password | role □ Sửa Chép Xóa bỏ | 1 | bobolan | $2y$10$fvUDBO7zClndo0MNT8YQmOUZSa24W5lyfiCTlkwM9io... | admin □ Sửa Chép Xóa bỏ | 2 | bobolan123 | $2y$10$uGB/wskTdAfH/5Wwmg8Ln.uct16s22e49UbofEeKpst... | user □ Sửa Chép Xóa bỏ | 13 | bobolan1234 | $2y$10$eEmgx4jbCpyheWSHG8aM3OwcenvsEeYUycx3.4bvqnY... | user	
Login	User needs to provide their username and password to access their account. The entered username must exist in the system, and the provided password must match the one associated with that username. Inform the user about the login result.	The user successfully logs in by providing a valid username and password, matching the credentials stored in the system.	The login attempt is successful, and the user gains access to their account.	 Log In Username: bobolan1234 Password: ••• Then it will redirect to home page.	Pass
Compose an email to the administrator	Fill the title and content that user want to send to admin through admin's email by default	Admin will get the email with title and content that user sent	Email sent to the admin successfully.	 Email Form Title: URL error? Content: I can not access the admin page. 	Pass

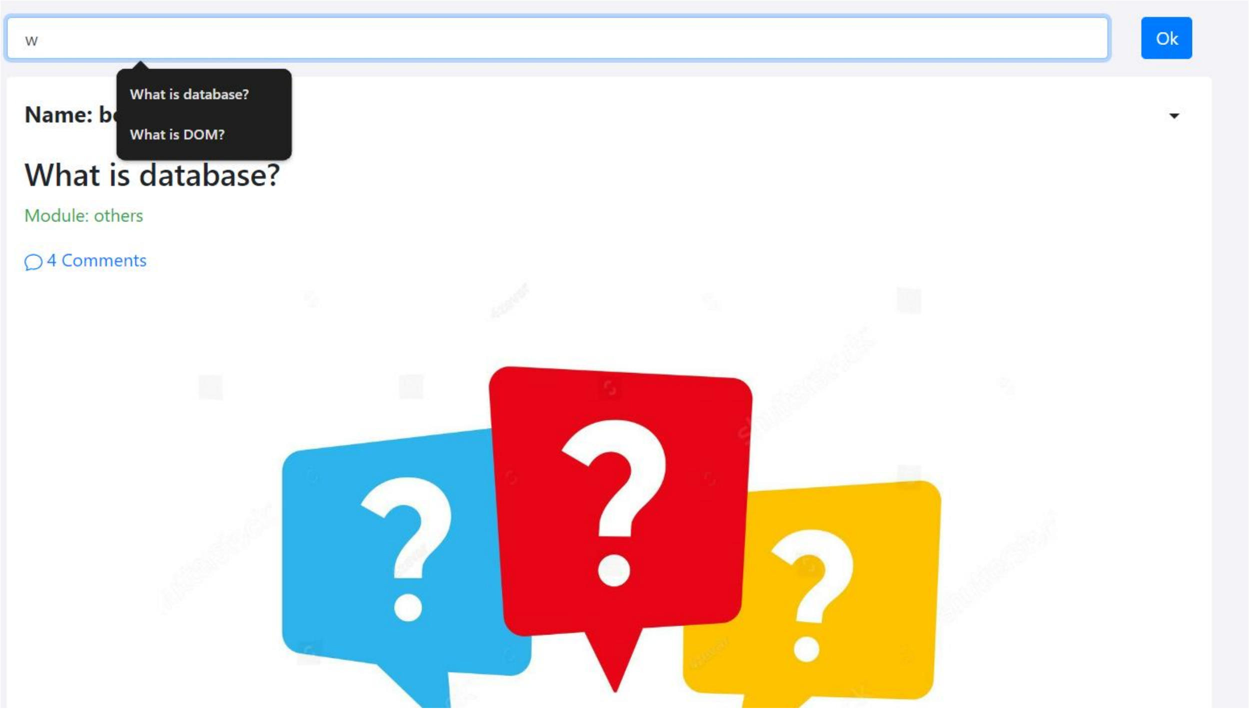
				 URL error? Hộp thư đến × bobolan12312@gmail.com đến tôi I can not access the admin page. 	
Create new post	Craft the question in the text area, optionally add an image, and select a module. Witness the magic as your input is securely saved in the questions table, seamlessly presented on the Questions page.	Experience the satisfaction of successfully saving your data in the questions table, elegantly displayed for easy access.	Validate the triumph as data impeccably stored in the questions table reflects seamlessly on the Questions page.	 Create Question: Your question: What's endpoint? Image for question: Choose File question2.jpg backenc id question_text img user_id module_id Sửa Chép Xóa bỏ 4 What are the key differences between Express.js (N... uploads/_1question.jpg 1 2 Sửa Chép Xóa bỏ 5 How do you handle authentication and authorization... uploads/_1question1.jpg 1 2 Sửa Chép Xóa bỏ 6 What's endpoint? uploads/_13question2.jpg 13 2 Name: bobolan1234 What's endpoint? 0 Comments Answer 	Pass
Comment post	Type in input tag comment and submit comment.	It will increase amounts of answers for that post. In addition, store data and display the answer in question detail page.	It stores and displays the comments successfully.	 Question: What's endpoint? bobolan1234 Answer · 0 Endpoints are physical devices that connect to and exchange information 	

				 bobolan1234 Follow Answer · 1 Answer Comment bobolan1234 Endpoints are physical devices that connect to and exchange information with a computer network.	
Display admin button in profile.	When user login the website, they won't see the admin button. On the other hand, only admin's account is authorized to use it.	Show the admin button when an administrator is logged in; otherwise, prompt for reauthorization.	The navigation bar correctly displays the admin button and appropriately disables it based on the account's role.	User's account:  Admin's account:  	Pass
Update user's information	Administration have a authorization to modify user's information.	User's information would be updated after changes.	It stores new user's data and display correctly.	Old data:  13 bobolan1234 user 1 Update new data:  Update Information New Username: bobolan1212 New Role: Admin New data:	

				 13 bobolan1212 admin 1 Update Delete	
Create new module	Fill the new module name in model module then generating the new one.	Create new module and save it in module database. In addition, ensuring the data will immediately show in the table.	After creating the new module, it stored and display in the table new row of the table module instantly.	Updated new module:  Id Name Action 1 frontend Update Delete 2 backend 5 database 6 framework 7 salary 8 security Update Delete Create Module × Module Name: authentication Close Create After create successfully:  Id Name Action 1 frontend Update Delete 2 backend Update Delete 5 database Update Delete 6 framework Update Delete 7 salary Update Delete 8 security Update Delete 9 authentication Update Delete	Pass

7. Additional feature:

- **Searching**: On the Questions page, users can easily find specific questions by typing keywords into the search bar. The system suggests question titles matching the entered keywords from the database. Once users find the desired question, they can click "Submit" to proceed. If the query is valid, users are redirected to the detailed page of that question. Otherwise, an error message is displayed, guiding users to refine their search terms.



w

Ok

What is database?

What is DOM?

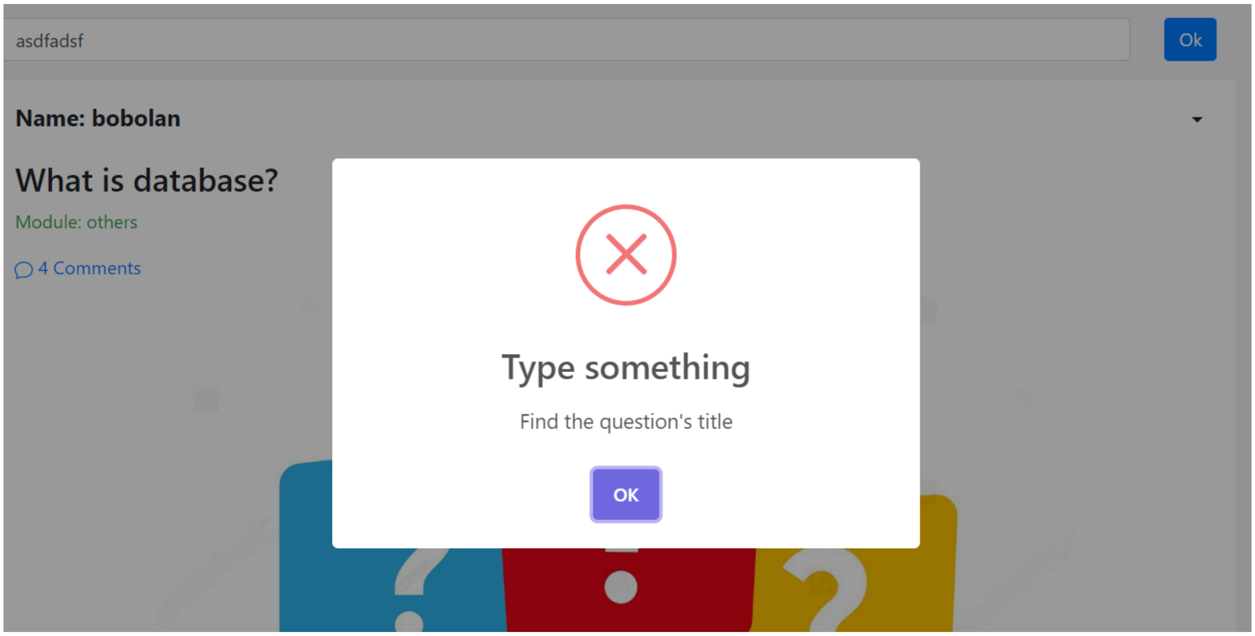
Name: b

What is database?

Module: others

4 Comments

Figure 19: Searching



asdfadsf

Name: bobolan

What is database?

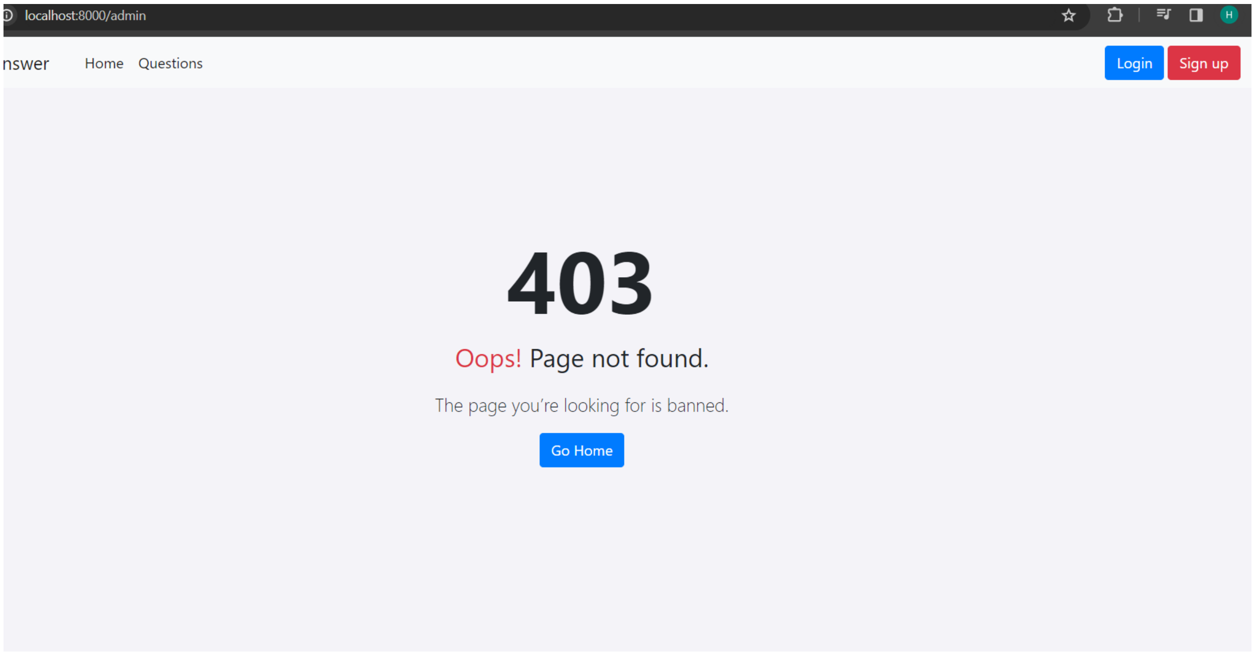
Module: others

Type something

Find the question's title

Figure 20: Error alert for searching

- Authorization: The system checks the user's role and login status before granting access to the admin page. If the user's role is not designated as admin or if they are not logged in, they are redirected to a 403 error page, indicating that access is forbidden.



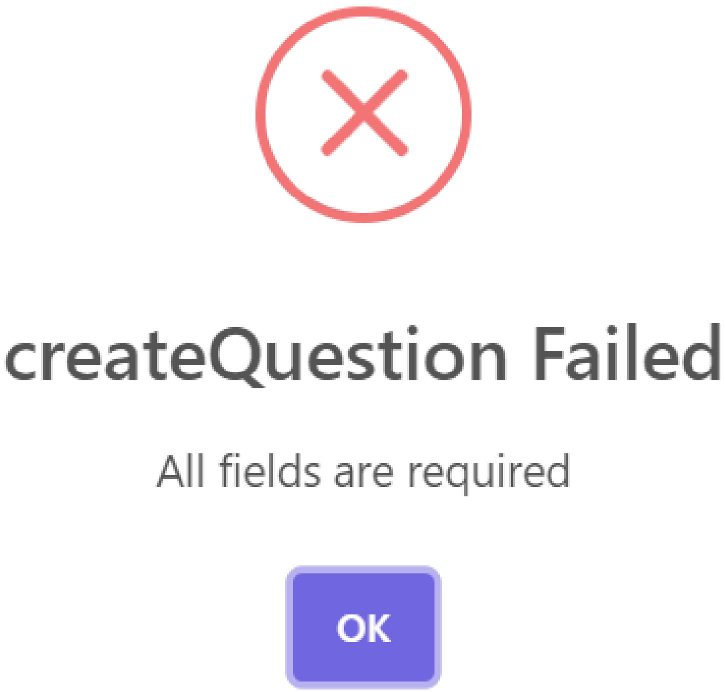
403

Oops! Page not found.

The page you're looking for is banned.

Figure 21: 404 page

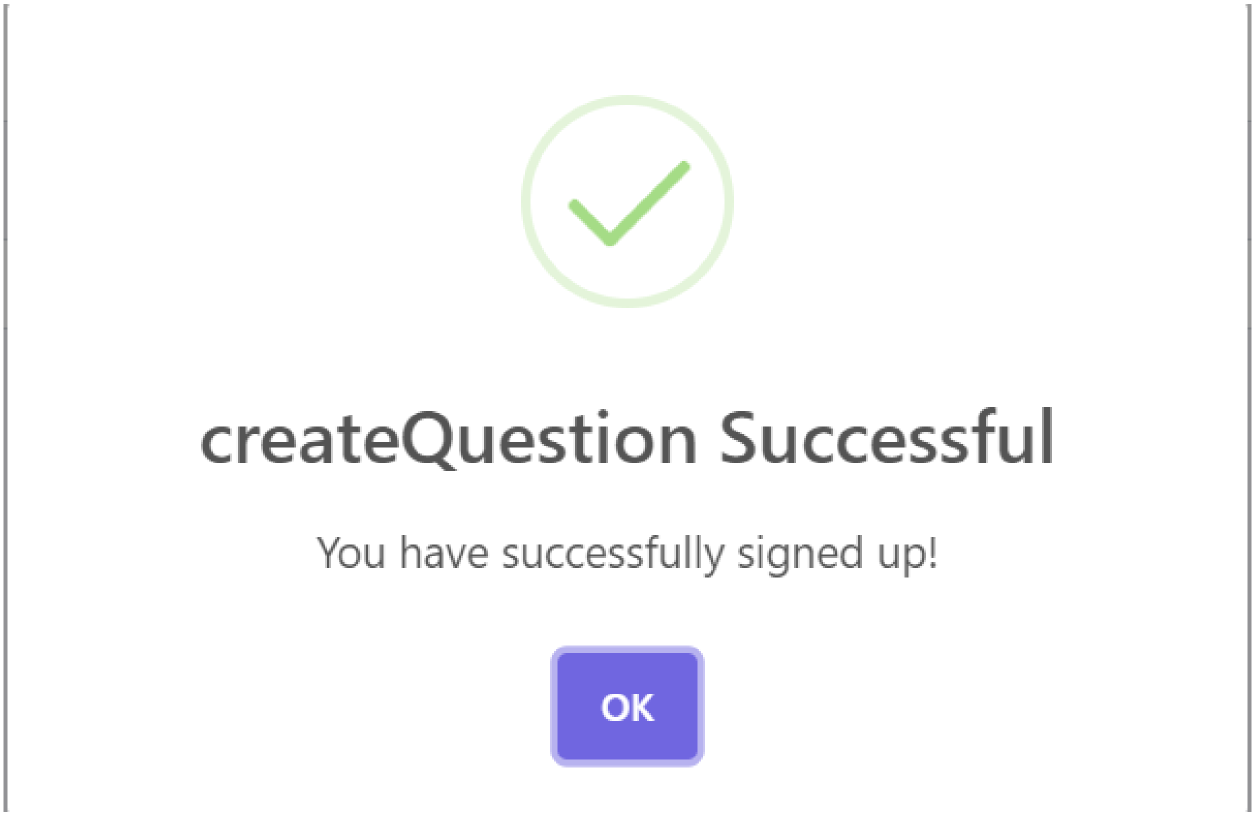
- **Notification Enhancement**: Implemented the Sweet Alert JavaScript library to enhance user notifications with elegant and customizable dialog boxes and alerts, enhancing the user experience with visually appealing feedback.
-



createQuestion Failed

All fields are required

Figure 22: Error alert

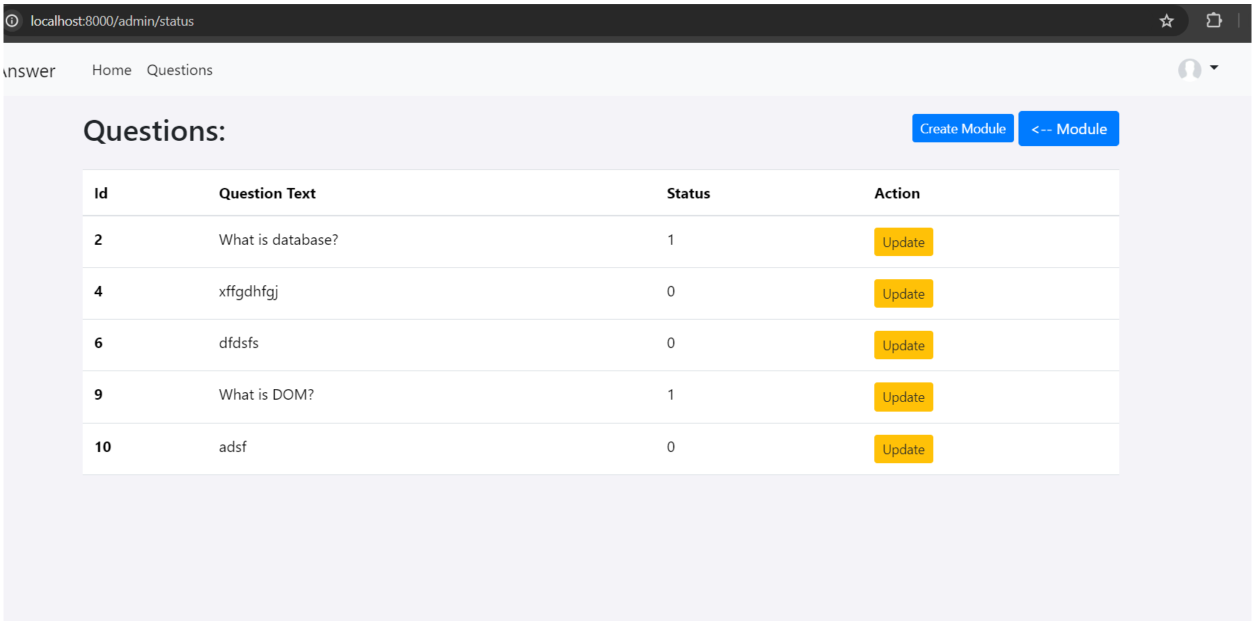


createQuestion Successful

You have successfully signed up!

Figure 23: Success alert

- Determine whether to display the status: Administrators have the authority to alter the status of questions. If a question is approved, it will be displayed; otherwise, it will remain pending and not be shown on questions page.



Questions:

Id	Question Text	Status	Action
2	What is database?	1	Update
4	xffgdhfgj	0	Update
6	dfdsfs	0	Update
9	What is DOM?	1	Update
10	adsf	0	Update

Figure 24: Status page

8. Conclusion and Future Recommendations:

To sum up, the CRUD web application development process has been successful in fulfilling the requirements mentioned in the coursework. The program has an easy-to-use interface, strong CRUD capability, safe user authentication, and efficient error-handling systems. We have successfully addressed the needs of the consumers by providing a dependable and effective solution that was carefully planned, implemented, and tested.

Future Recommendations:

Here are some suggestions for future development to improve the program even further and the user experience in general:

1. **Implement Additional Features**: To match user satisfaction and engagement, think about adding more features. Features like user profiles, alerts for new comments or posts, a search function, and social sharing options might be included in this.

2. **Refine User Interface**: To further improve the user interface, get input from users and make design iterations. Make sure the program is user-friendly and straightforward for all users by paying attention to usability and accessibility factors. It is necessary to add the footer, which has multiple side sections of information.

3. **Enhance Security Measures**: To protect user data and stop illegal access, security measures should be continuously reviewed and improved. This could entail putting policies in place like encryption for sensitive data, two-factor authentication, and frequent security assessments.

4. **Optimize Performance**: To guarantee a quick and responsive user experience, assess the application's performance and pinpoint areas that need optimization. This could entail caching solutions, reducing website load times, and streamlining database queries.

5. **Conduct User Testing**: Hold user testing sessions on a regular basis to get input and understanding from actual users. Utilize this input to pinpoint areas of difficulty, usability problems, and improvement needs. Then, prioritize changes according to the requirements and preferences of the user population.

6. **Enhance database:** Save other details like the user's location, gender, email address, and so forth. To make the authorization more precise, create extra tables, such as individual role tables.

7. **Login section:** Integrating more login option for example Gmail, Facebook and Github.

We may further improve the CRUD web application and make sure it remains relevant and successful in fulfilling the changing needs of customers by putting these future ideas into practice. Our goal is to provide an outstanding user experience and keep the program a useful tool for its users by means of constant innovation and enhancement.

9. References:

PHPMailer (n.d.) PHPMailer/PHPMailer: The Classic email sending library for php, *GitHub*, [online] Available at: https://github.com/PHPMailer/PHPMailer (Accessed 27 February 2024).