



JoLaTM

Discussion forums ReImagined, ReDiscussed



Final Report

UBC 360: Web Programming

Professor Ifeoma Adaji

Team Members

AbdulAziz Almutlaq - 79960175

Onwuvuche Jordan Chukwuka - 61007530

Implementation From a System Perspective

Outline of Implemented Features

A. Non Logged In Users

- Users are able to view posts directly on the main page.
- Users can retrieve their passwords via email by selecting the "Get Help Sign in" option from the login page.
- Users will have access to the same styling as a registered user.
- Users have the ability to search for posts, threads, and comments by clicking on any of the left-hand side menu bar options, and then using the search bar.
- Users can create an account by clicking on the "Sign-in" button located on the top-right of the main page's navigation bar.

B. Logged In Users

- Users can create threads with custom background images and profile pictures.
- Users can create posts in any thread with options to include body text, images, or YouTube URLs.
- Users can vote on both comments and posts made by other users.
- Users can add comments to any post within a thread.
- Users have the ability to hide or delete their own posts, as well as delete comments from their posts.
- Users can update their profile pictures, usernames, passwords, and email addresses.
- Users can use the search bar, available on the left-hand side menu, to search for posts, threads, and comments.
- Users can log in by simply entering their email and password.
- Users have the ability to view every comment they have made.
- The session is stored to maintain user state, including staying logged in between page navigations.
- Users receive different alerts:
 - The Admin [Username] has deleted your thread [Thread].
 - The Admin [Username] has deleted your post in thread [Thread].
 - [Username] liked your post in thread [Thread].
 - [Username] disliked your post in thread [Thread].
 - [Username] has posted in your thread [Thread].
 - [Username] replied to your comment in thread [Thread].

C. The Admin

- The Admin has the ability to search for specific users by entering their usernames on the dedicated tab.

- Admins can perform actions such as blocking or unblocking users and granting them admin privileges.
- The admins can hide, restore, or delete threads.
- Admins have the ability to delete any comments or posts as appropriated by Community Guidelines.

Description of JavaScript and PHP Project Files

A. PHP

- **Account-settings:** component in View. The file utilizes the UsersClass class to present up-to-date user information, and dynamically updates this information using Javascript.
- **Account:** component in View. This file utilizes the UsersClass class to showcase up-to-date information about the user and their respective threads.
- **Admin-threads:** component in View. This file utilizes the AdminClass class to exhibit all the threads present on the website. It incorporates Ajax to enable the dynamic deletion/hiding/restoration of threads on the website. Access to this page is restricted to Administrators only.
- **Admin-users:** component in View. This file uses the AdminClass class to show a list of all registered users on the website. It employs Ajax to allow for dynamic deletion, user promotion to admin, or unblocking of users. Access to this page is limited to administrators only.
- **Admin:** component in View. This file utilizes the AdminClass class to exhibit website statistics and trending threads. Access to this page and its actions is limited to administrators.
- **Error:** component in View. This file works in conjunction with Router.class.php to display an error message when a page is not found on the website.
- **Index:** component in View. This file manages the rendering of Views, titles, and security checks to ensure that access to certain pages is only granted to authorized users, admins, or guests.
- **Login:** component in View. This page is restricted from access if the user is already authorized to the system. Users cannot log in if their email is unconfirmed or if their account is disabled. The page employs Ajax to validate user input with the server and establish the current session with the correct user credentials.
- **Notifications:** component in View. This file utilizes the NotificationsClass class to exhibit the activity of other users towards the content created by the current user on the website. The content is exclusively visible to the current user.
- **Post-create:** component in View. This file utilizes the PostHelperClass and PostsClass classes to facilitate the creation of posts for their corresponding threads. It leverages Ajax and Javascript to interact with databases and perform

security checks to ensure information is securely stored in the database or file storage systems such as images.

- **Post:** component in View. This file uses the PostsClass class to display the details of a single post, along with its respective information in the thread. It utilizes Ajax and Javascript to ensure the security of comments posted in the post and thread. This page is only available for logged-in users, and the post won't be displayed if it has been deleted by the owner or an administrator. Users are not allowed to post comments or vote if the post is marked as "hidden."
- **Register-confirm:** component in View. This page is exclusive to users who possess a unique link containing a "token" attribute in the URL. Users need to enter a randomly generated code on this page to confirm their account registration. During the registration process, the token is randomly generated and saved in the database. The token length fluctuates to protect against brute-force attacks. The security code to access the page is sent to the user's email, along with a secure link to validate the account registration.
- **Register:** component in View. This page enables the user to register a new account. It is only available to users who are not logged in. The page uses Ajax and Javascript to validate user input, ensuring that the fields are not empty and that the username and email are unique and meet the required password format. Once validated, the information is stored in the database, and a confirmation email is sent to the user's email address containing a token and confirmation code. The user must enter this code on the website to confirm the account registration.
- **Restore-confirm:** component in View. This page allows users to reset their account password by entering new and confirmed passwords. Access to this page is restricted if the URL attribute is missing a randomly generated "token" or if the token has expired. The token is generated randomly with different symbols and lengths when the user initiates the "password recovery" process on the website's restore page.
- **Restore:** component in View. The page allows users to initiate the password recovery process by entering their email address. Access to this page is restricted to users who are not currently logged in. If the email address entered by the user is not found in the database or is not verified, the user will not be able to proceed with the password recovery process. Otherwise, a secure link will be sent to the user's verified email address to proceed to the next step of the password recovery process.
- **Search:** component in View. This page is designed to enable users, whether authorized or not, to search for information on the website, including comments, posts, and threads. The server will sanitize the search query to protect against potential attacks before sending it to the Model for execution. Ajax and

Javascript are utilized to dynamically validate, and update page content based on the search query and parameters.

- **Thread-create:** component in View. The creation of a thread is facilitated by the ThreadHelperClass and ThreadsClass classes, which use Ajax and Javascript to communicate with databases and perform security checks before storing information in database or file storage systems such as images.
- **Thread:** component in View. Authorized users can access this page to view details about threads, including associated posts and comments. Users can take various actions on this page, such as joining or leaving a thread, creating posts, and voting for posts and comments. The page also displays a list of top users for the thread based on the number of posts they have made.
- **Main:** component in View. This page is accessible for all authorized and non-authorized users to show the trending posts and threads. This page is also used as the main page of the website.
- **ConnectDB.class:** component in Controller. This class establishes a connection to the database and returns a "Connection" object for other controllers/models to access the database. If the connection is unsuccessful, an error page will be displayed with a message that the website is not working. For security purposes, specific details regarding the error will not be displayed.
- **Environment.class:** component in Controller. This class reads the ".env" file to retrieve the database credentials, and then passes this information to the DatabaseConnector class to establish a connection with the database.
- **Navigation.class:** component in Controller. This class handles routing of all incoming GET requests to specific pages by returning the page title and corresponding file path for rendering.
- **AdminHelperClass.class:** component in Controller. This file class implements validation for user input and directs information to the appropriate helper class method based on the request type (POST/GET). Upon successful validation, the helper class methods communicate with the AdminClass Model to perform database operations such as updating, deleting, or creating information. After the operation is complete, the server returns a JSON object with either information or error to the Ajax method.
- **CommentHelperClass.class:** component in Controller. This file class implements validation of user input to ensure its validity and prevent security risks. The helper class directs the input to the appropriate method in the CommentsClass Model, which is responsible for updating, deleting, or creating information in the database. After the operation is complete, the server returns a JSON object containing either information or an error message to the Ajax method.
- **ThreadHelperClass.class:** component in Controller. This file class performs data validation on user input like that done in the JavaScript file. It then sends the information to the appropriate method in the helper class based on whether the

request is a GET or POST. After validation is complete, the server calls the corresponding method in the ThreadsClass Model to modify, delete, or create information in the database. Finally, the server returns a JSON object with either information or error back to the Ajax method.

- **TokenHelperClass.class:** component in Controller. This file class includes validation to confirm that user input is valid, like the validation in the javascript file. Depending on whether the request is a POST or GET request, the information is sent to a specific method in the helper class. After completing the validations, the server calls the specific method in the TokensClass Model to update, delete, or create information in the database. When the operation is finished, the server sends a JSON object with information or an error back to the Ajax method.
- **UserHelperClass.class:** component in Controller. This file class includes a validation process that verifies the validity of user information passed in the request. Depending on whether the request is a POST or GET, the information is forwarded to a designated helper class method. After validation, the server calls the respective method in the UsersClass Model to create, update or delete user data in the database. Once the operation is executed, the server returns a JSON object with relevant information or an error message back to the Ajax method.
- **Controller.class:** component in Controller. This is an abstract class that enforces Models to implement specific methods such as get, post, update, and delete to interact with the database.
- **AdminClass.class:** component in Model. This controller performs insertion or deletion based on the information received from the AdminHelperClass. It also retrieves information based on method calls from the respective view in the admin pages.
- **CommentsClass.class:** component in Model. The CommentsClass class executes insertion/deletion of comments based on the information passed from the CommentHelperClass. It also executes retrieval of information based on method calls from the respective Views in the thread, comment, post, and search pages.
- **NotificationsClass.class:** component in Model. This controller processes insertion and deletion based on the user's actions on the website and information passed from various helper classes. It also retrieves information based on method calls from the notification page's respective view.
- **PostsClass.class:** component in Model. This class handles the insertion and deletion of information based on the data sent by the PostHelperClass. Additionally, it retrieves information through method calls from Views related to threads, comments, posts, and searches.
- **ThreadsClass.class:** component in Model. This controller performs database insertion/deletion based on the information received from the ThreadHelperClass. It also retrieves information based on method calls from the corresponding Views in the thread/main/admin/search pages.

- **TokensClass.class:** component in Model. This controller manages the creation and deletion of user accounts based on information passed from the UsersClass and TokenHelperClass. It also retrieves and validates information based on method calls from Views in the register-confirm and restore-confirm pages.
- **UserHelperClass.class:** component in Model. This controller facilitates insertion and deletion of data by verifying the user's credentials through various helper classes (e.g., email confirmation, admin status, etc.). Additionally, it retrieves data and performs validation operations based on the requests made by the user from Views such as account and account-settings pages.

B. JavaScript

- **Script.js:** This file manages the real-time updating of website pages through the use of JQuery and Ajax.

How The Website Works at a High Level

The codebase is segregated into client-side and server-side components.

A. Client-side

The client-side code aims to create a dynamic and responsive website. Using AJAX, we update styles and information dynamically, avoiding page refreshes while using search bars or sorting threads. Additionally, we utilize RESTful API calls to retrieve the required information from the server. This code also includes CSS style definitions and page creation for the entire website.

B. Server-side

The server side of our codebase is responsible for establishing a connection to our database and facilitating user access and modification of data. To ensure consistency in connecting to the database, we utilize a DatabaseConnector class that is called in our controllers.

Helper classes were developed to enhance security. Whenever an API call is initiated, it is first routed through a PHP helper class to verify the validity of the transmitted data. In case the data is invalid, an error is returned. However, if the data is valid, the helper class passes it to the relevant PHP controller.

Our querying is centralized in the PHP controllers. We have implemented various controllers for different functionalities: Admin, Comment, Notification, Post, Thread, Token, User.

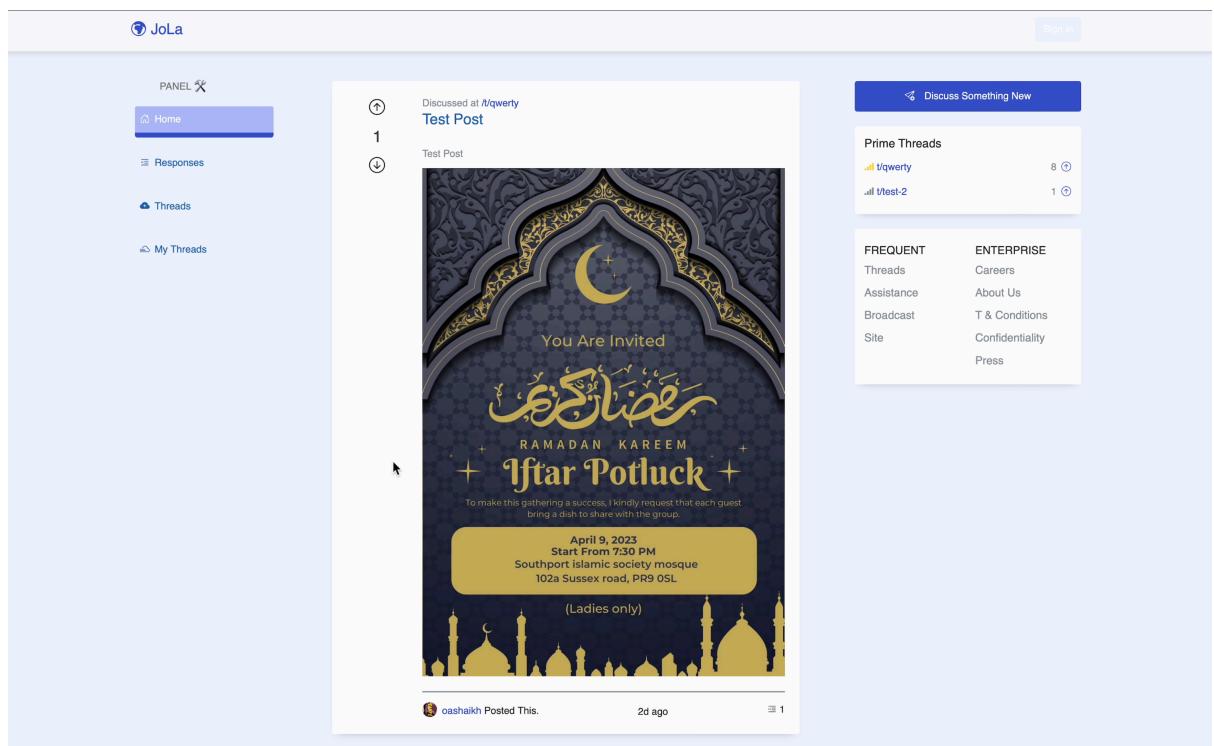
The communication with the required database models defined in our database folder is handled by each controller. Additionally, a PHP router class has been created to ensure proper navigation between pages and to set the webpage's title.

Website Limitations

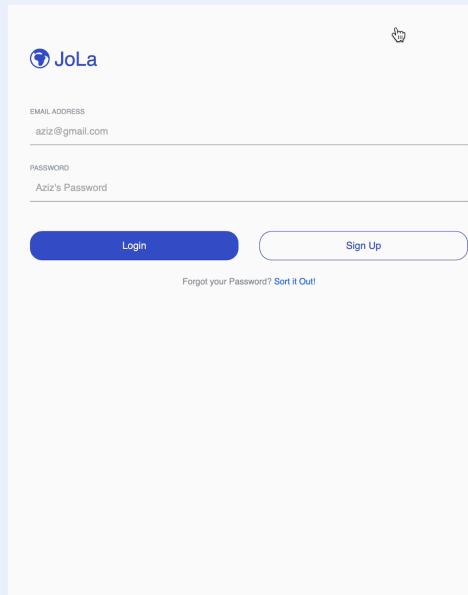
- Due to the unavailability of an SMTP server/service, the system is unable to send an automatic email containing the recovery/register link. Therefore, the administrator needs to provide the link to the user manually by accessing the database.
- Users do not have the privilege to edit their comments once they are posted.
- The system does not support replying to another user's comment. Users can only comment on a post directly.

Examples of Each Page Type

I. Home Page



II. Login Page

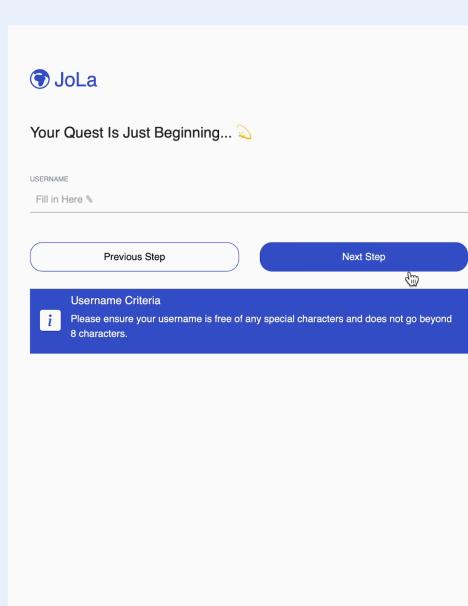


The JoLa login page features a logo with a globe icon and the text "JoLa". It has fields for "EMAIL ADDRESS" containing "aziz@gmail.com" and "PASSWORD" containing "Aziz's Password". Below the fields are two buttons: a blue "Login" button and a white "Sign Up" button. A small link at the bottom says "Forgot your Password? Sort it Out!".

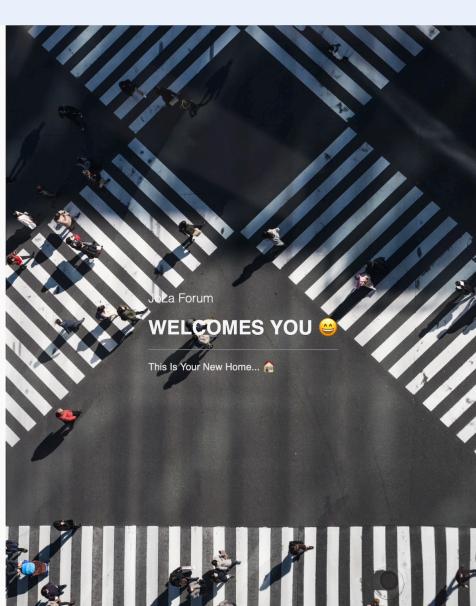


The JoLa forum welcome screen shows a dark background with blurred lights. It features the text "JoLa Forum" and "WELCOMES YOU BACK" with a smiling emoji. Below this, it says "It's Good To See You Again". The background image is a photograph of a group of people in a dark room, possibly a forum meeting.

III. Register Page



The register step 1 page has a header "JoLa" and a tagline "Your Quest Is Just Beginning... 🌐". It has a "USERNAME" field with placeholder text "Fill in Here %". Below the field are two buttons: "Previous Step" and "Next Step". A blue callout box titled "Username Criteria" contains the text: "Please ensure your username is free of any special characters and does not go beyond 8 characters." A cursor is hovering over the "Next Step" button.



The JoLa forum welcome screen is identical to the one shown in the login section, featuring the text "JoLa Forum" and "WELCOMES YOU BACK" with a smiling emoji, and the background image of people in a dark room.

 JoLa

Your Quest Is Just Beginning... 🌟

EMAIL
qwertystyle@qwertystyle.com

Previous Step Next Step

 Your Email
Our community guidelines say that your email should not be longer than 25 characters.



JoLa Forum
WELCOMES YOU 😊
This Is Your New Home... 

 JoLa

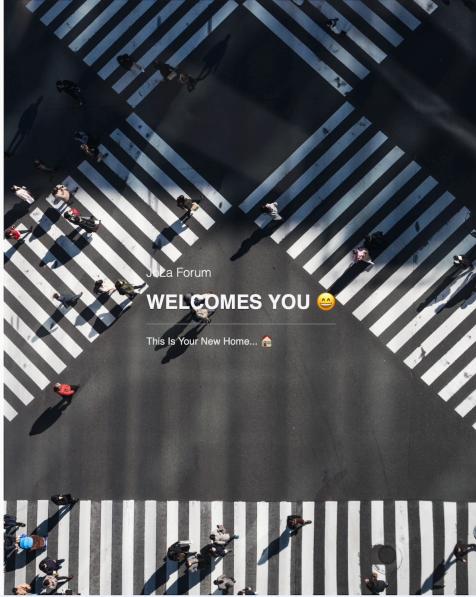
Your Quest Is Just Beginning... 🌟

PASSWORD
 Secret Password

REPEAT SECRET PASSWORD
 Repeat Secret Password

Previous Step Register

 Your Password
We recommend to create a password with minimum length of 8 characters, one uppercase letter, one special symbol.



JoLa Forum
WELCOMES YOU 😊
This Is Your New Home... 

 JoLa

Your Quest Is Just Beginning... 😊

PASSWORD
.....

REPEAT SECRET PASSWORD
.....| 

[Previous Step](#) [Register](#)

 Your Password
We recommend to create a password with minimum length of 8 characters, one uppercase letter, one special symbol.

 Oops...
Passwords don't match.



JoLa Forum

WELCOMES YOU 😊

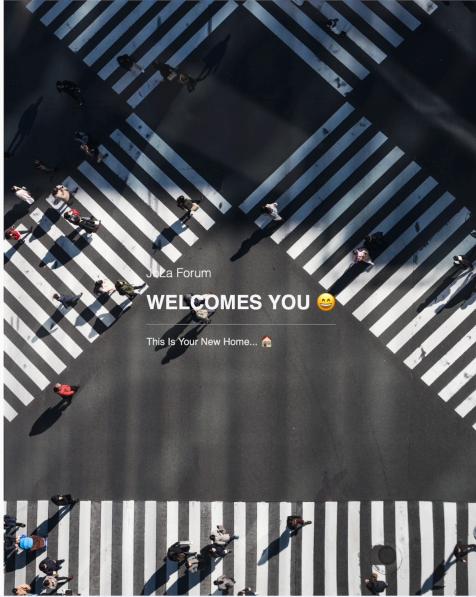
This Is Your New Home... 

 JoLa

Thanks for Joining Our Community.

We are almost done. Check your email to verify your account creation.

[TAKE ME BACK](#)

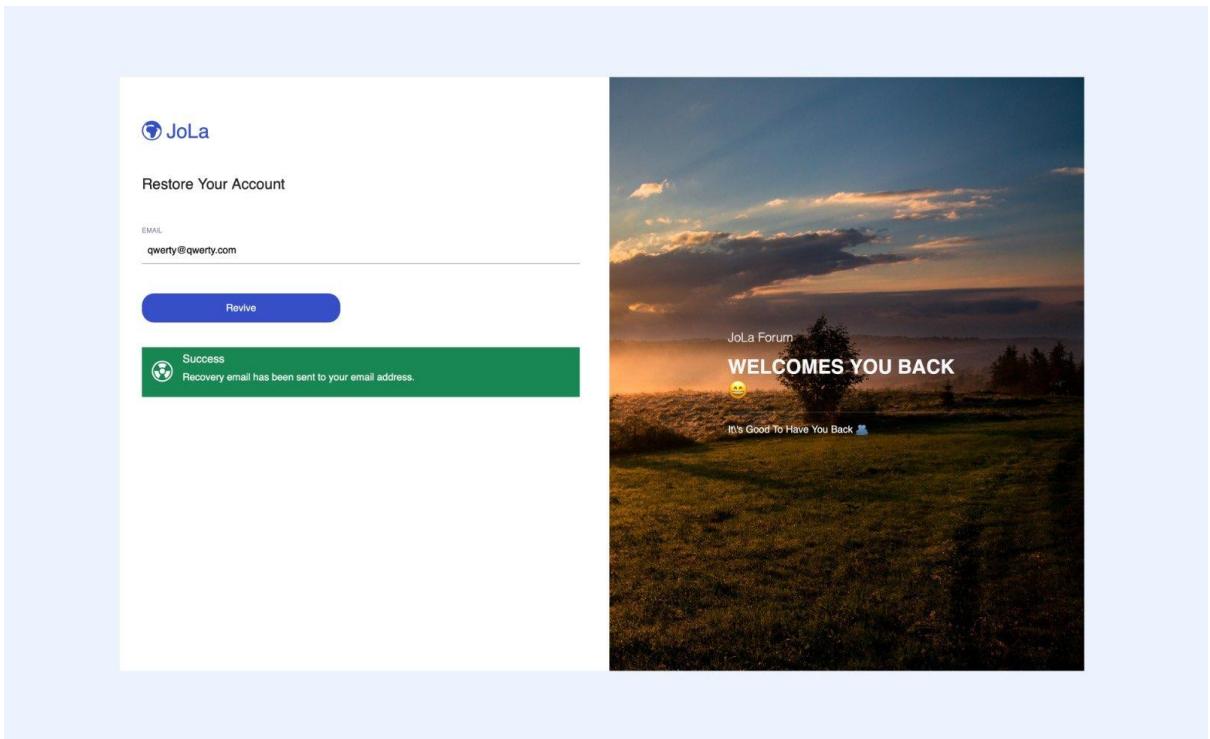


JoLa Forum

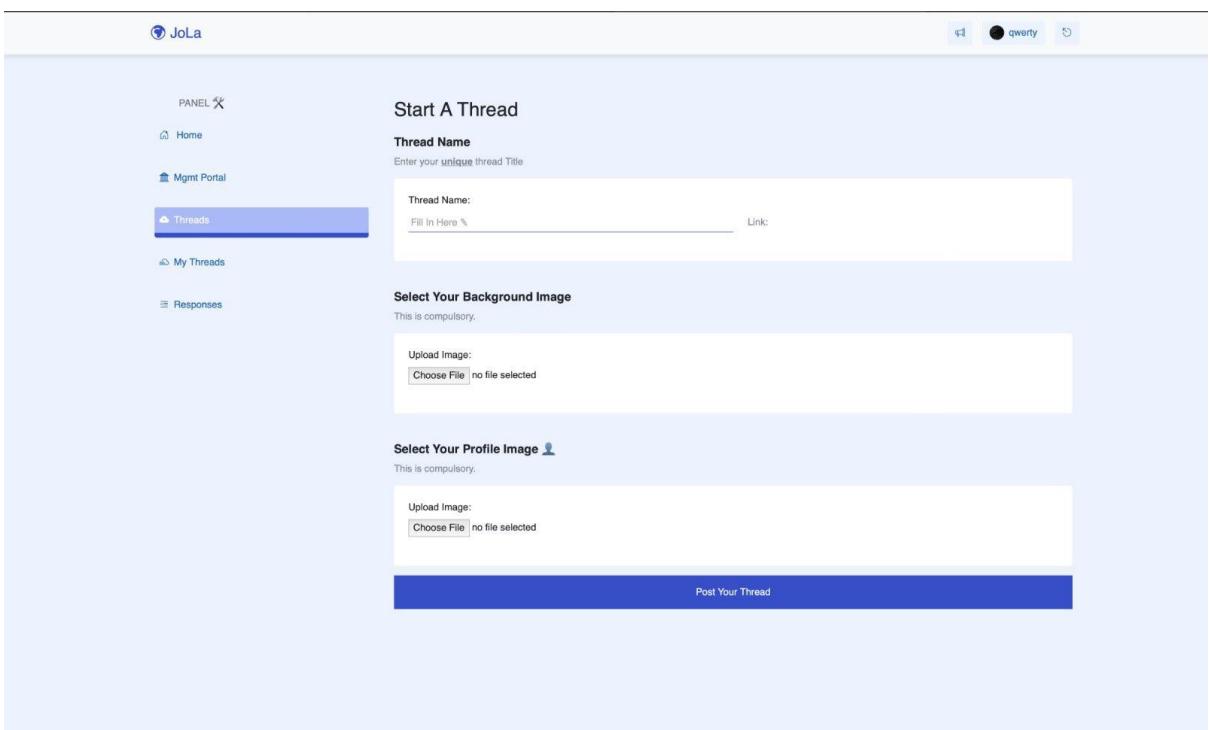
WELCOMES YOU 😊

This Is Your New Home... 

IV. Restore Account Page



V. Create Thread Page



VI. Thread Posts Page

JoLa

PERFUMES صنع في الهند

qwerty Join t/qwerty

PANEL X

Home Threads My Threads Responses

Leading Recent Retrieve a Thread 😊

Test Post

1 Test Post

④ You Are Invited

رمضان كريم RAMADAN KAREEM Iftar Potluck

PANEL X

Home Threads My Threads Responses

Leading Recent Retrieve a Thread 😊

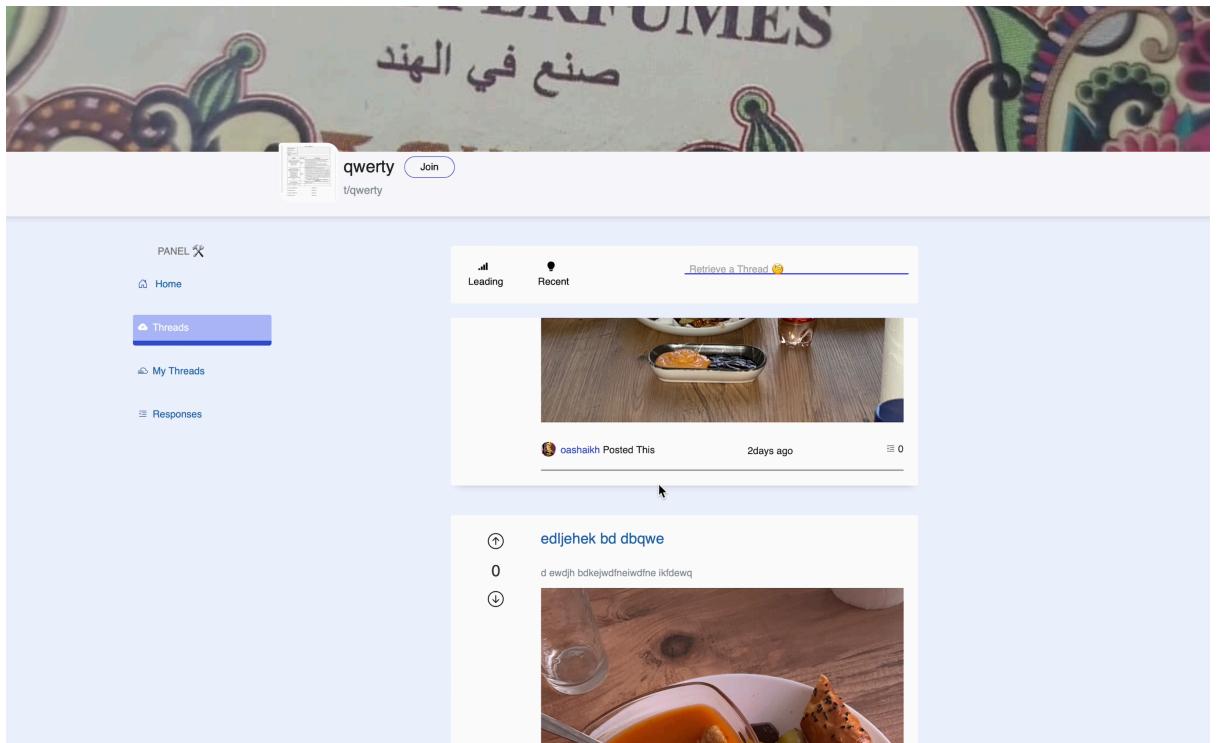
oashaikh Posted This 2days ago 1

0 qazwsx replies 12hrs ago

④ jhkbjfe def efeiun ewliefhbefbewnfilefbfie

qazwsx

jhkbjfe def efeiun ewliefhbefbewnfilefbfie



qwerty t/qwerty

PANEL X

Home Threads My Threads Responses

Leading Recent Retrieve a Thread 😊

oashaikh Posted This 2days ago 1

Hello 1mins ago

Remove

d wgduuwubdkquw

wdjnjww dlkjwdlwdwidqdhnwdiuw we d dwiqwd

qwerty Posted This 0mins ago 0

Hide Remove

صنعت في الهند

qwerty Join

PANEL X

Home Threads My Threads Responses

oashaikh Posted This 18mins ago 0

TTTTtttkuhdejdhejdhelwjkf

dwedbwibdwijkndkwydn تتصدر يصطفى انتير نمسا مسني

qwerty Posted This 0mins ago 0

Discuss Something New

Prime Profiles

oashaikh 8 ①

qwerty 1 ①

FREQUENT ENTERPRISE

Threads Careers

VII. Create Post Page

JoLa
qwerty

PANEL X

- [Home](#)
- [Threads](#)
- [Responses](#)
- [My Threads](#)

Post Something 🌐

Post Title 📝

Enter your unique Title

Post Title:
Your Title Goes Here. ↗

Say Something 🎤
What's On Your Mind?

Post Body:
Exercise your freedom to speak 💬

Uploading Any Image? 📸
Make your Post More Engaging With an Image

Upload image file:
 Choose File | no file selected

How About a Video? 🎥
Enhance Post Visuals With a \$currentPost['media_link'] Video.

YouTube URL

JoLa
qwerty

PANEL X

- [Home](#)
- [Threads](#)
- [Responses](#)
- [My Threads](#)

Post Title:
TTTTtttuhdejdhejdhejdhewjkf

Say Something 🎤
What's On Your Mind?

Post Body:
...eadWlbdwjkndkwjd...
نحشل بصمت و صوت بغير صوت في الظلام

Uploading Any Image? 📸
Make your Post More Engaging With an Image

Upload image file:
 Choose File | no file selected

How About a Video? 🎥
Enhance Post Visuals With a \$currentPost['media_link'] Video.

YouTube URL

YouTube Links Only

Create Post

VIII. Posts Page

JoLa

qwertystories

qwertystories

PANEL ☰

Home

Threads

My Threads

Responses

qwertystories

Join

t/qwertystories

edjehet bd dbqwe

d ewdjh bdkejwdfneiwdfne ikldewq

0

↑

↓



qwertystories

Join

t/qwertystories

Discuss Something New

Prime Profiles

 oashaikh 8 ⓘ

FREQUENT

Threads

Assistance

Broadcast

Site

ENTERPRISE

Careers

About Us

T & Conditions

Confidentiality

Press

Discuss Something New

Prime Profiles

 oashaikh 8 ⓘ

FREQUENT

Threads

Assistance

Broadcast

Site

ENTERPRISE

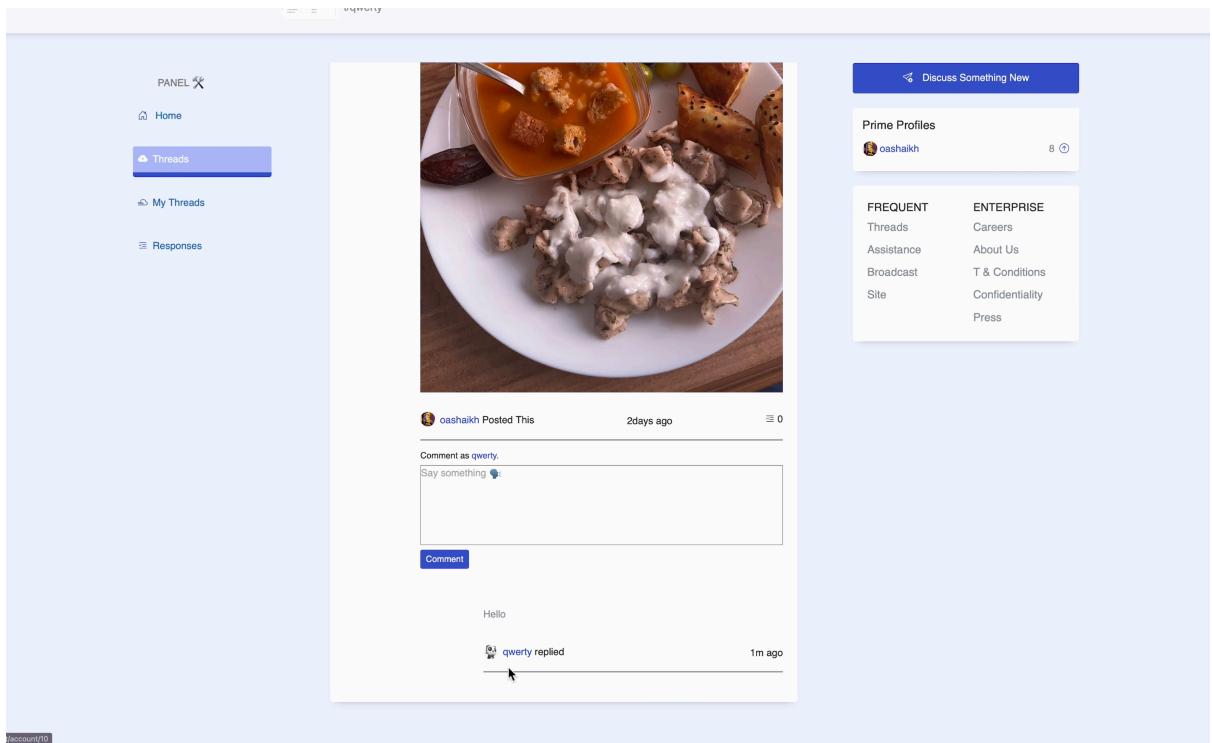
Careers

About Us

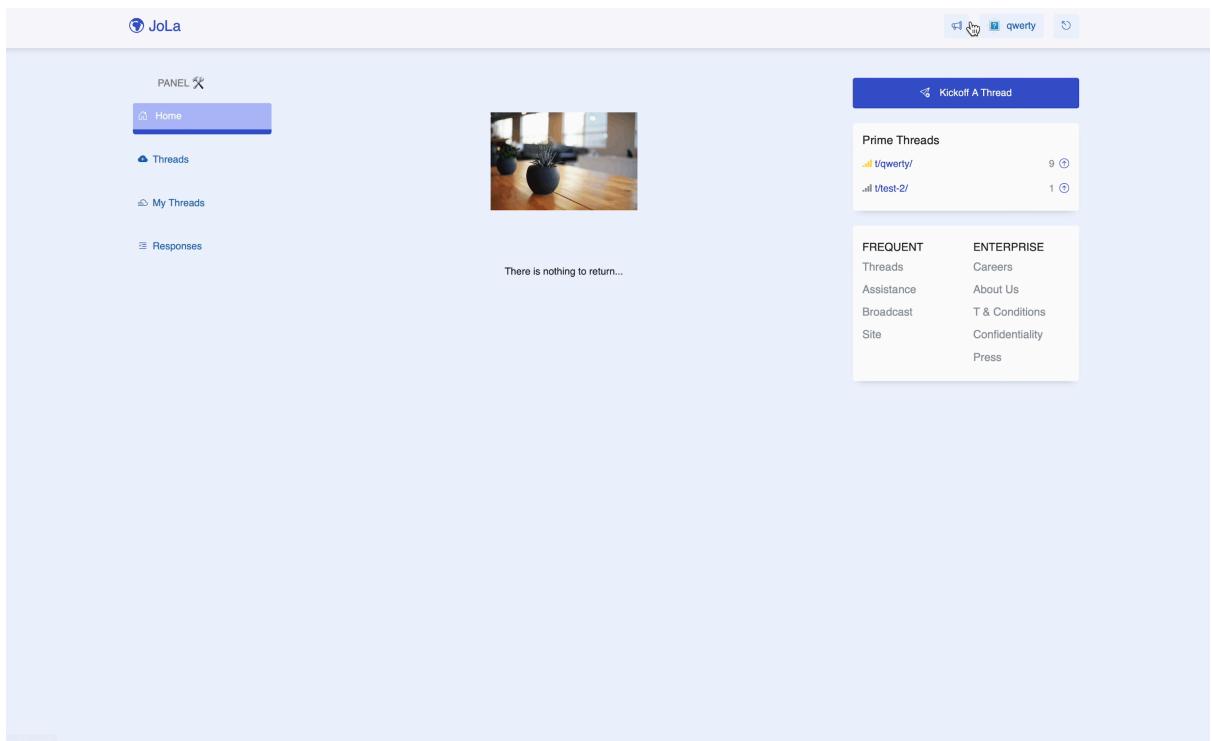
T & Conditions

Confidentiality

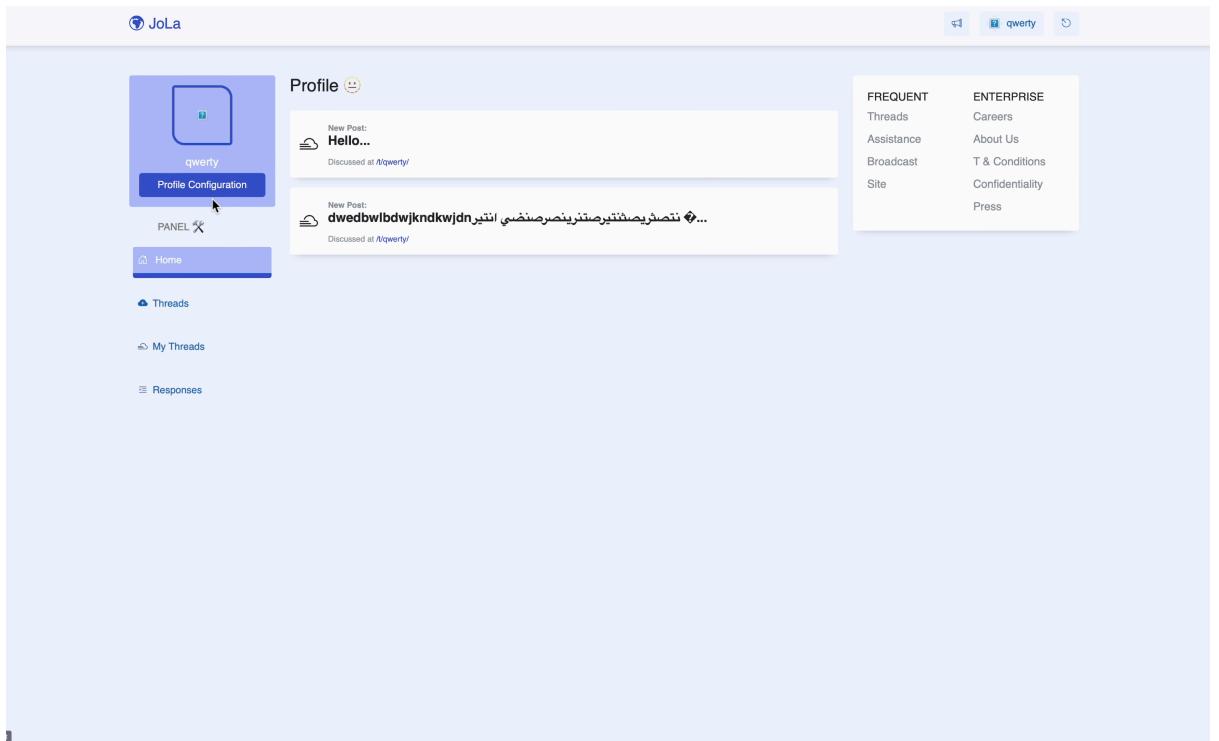
Press



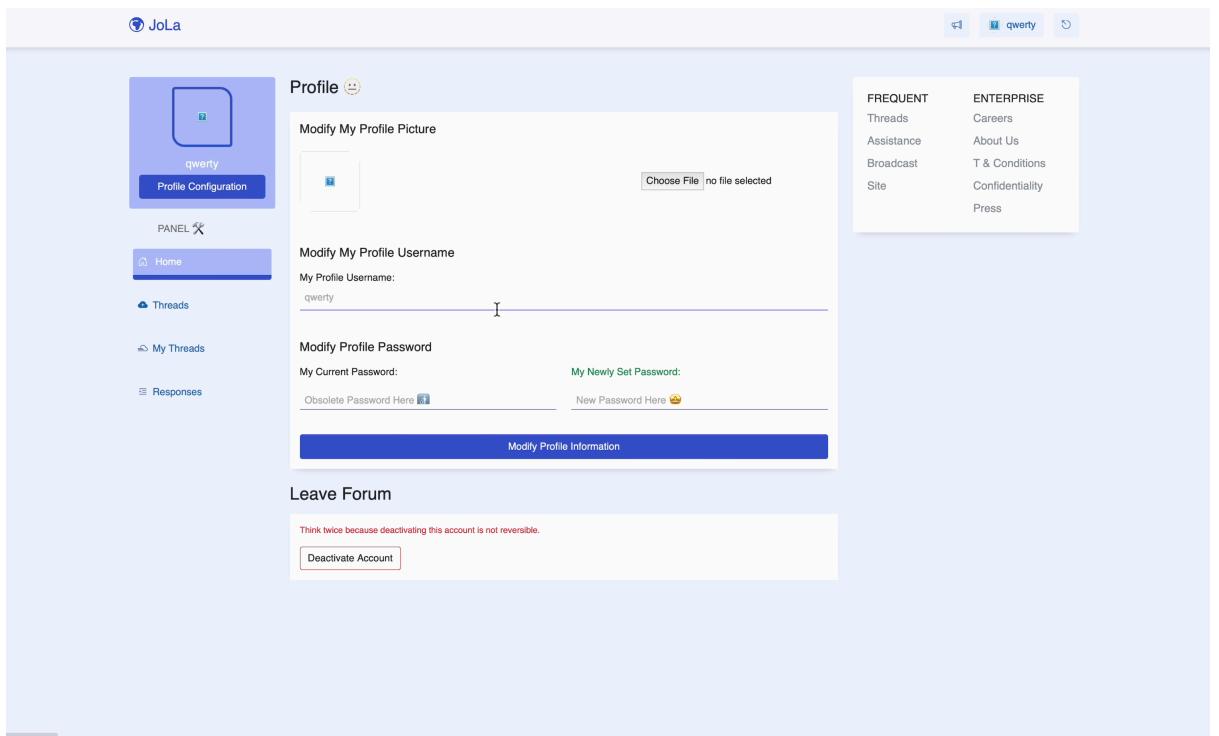
IX. User Notifications Page

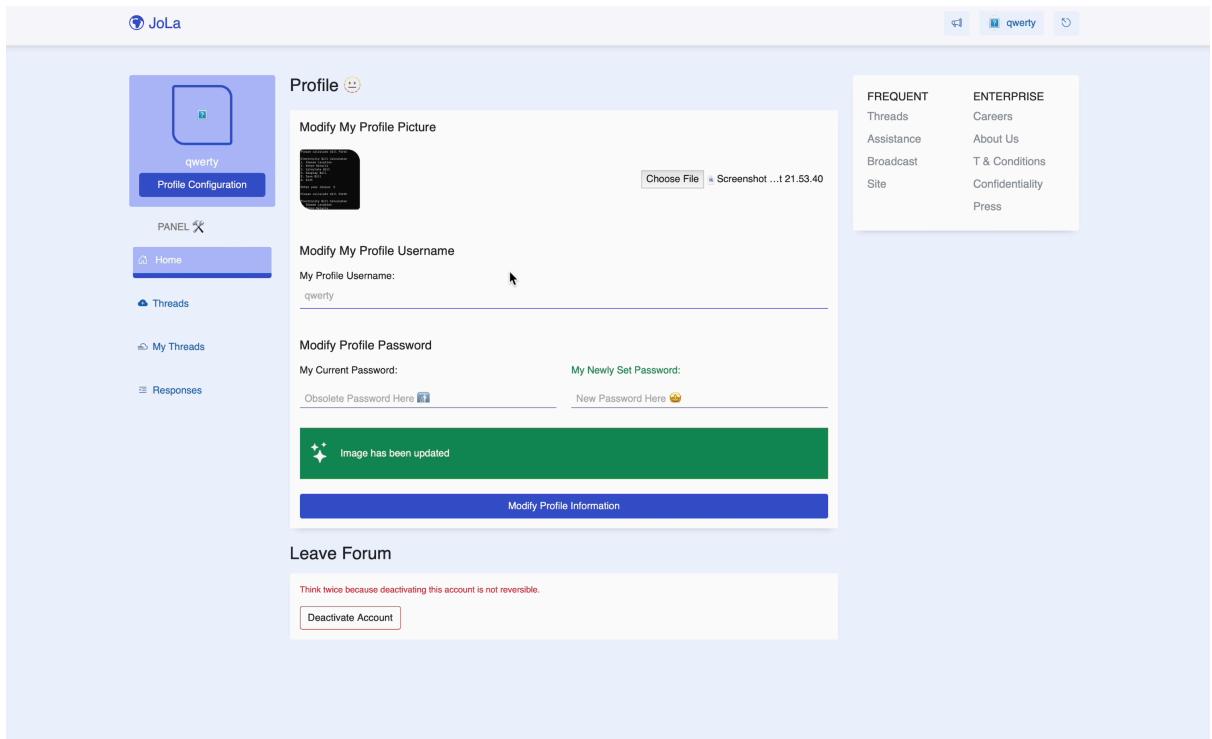


X. Account Page

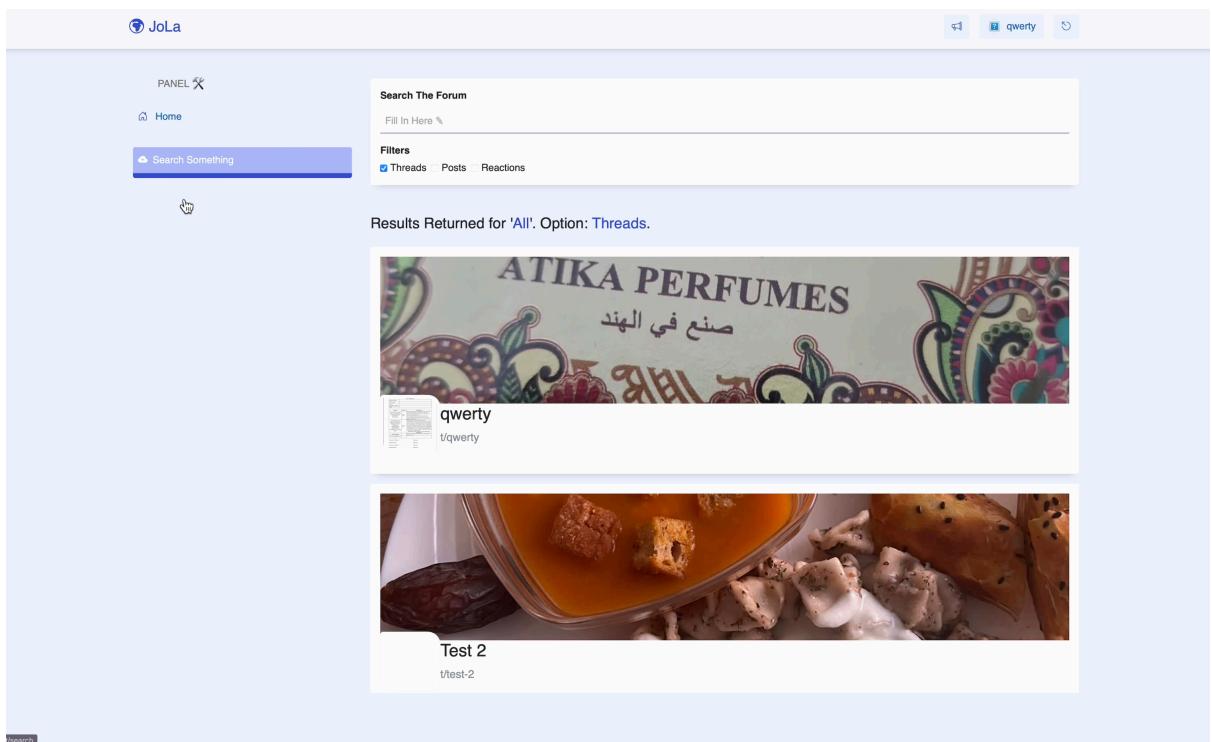


XI. Settings Page





XII. Search Page



XIII. Admin Main Page

PANEL ☰

- Home
- Mgmt Portal**
- Users
- Threads


Signed up Users
3


Threads
2

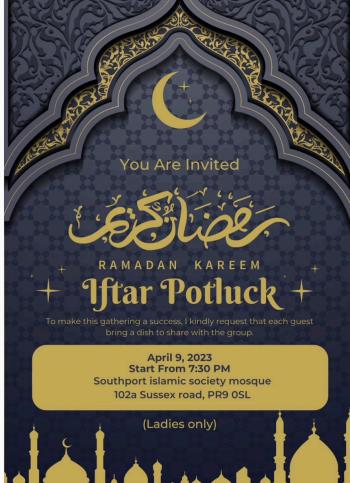

Discussions
10


Reviews
3

What's Making Rounds 

Discussed at [@qwerty](#)
Test Post

1
↑ Test Post
↓



You Are Invited
RAMADAN KAREEM
Iftar Potluck
 To make this gathering a success, I kindly request that each guest bring a dish to share with the group.
 April 9, 2023
 Start From 7:30 PM
 Southport Islamic society mosque
 102a Sussex road, PR9 0SL
 (Ladies only)

XIV. Admin Users Page

PANEL ☰

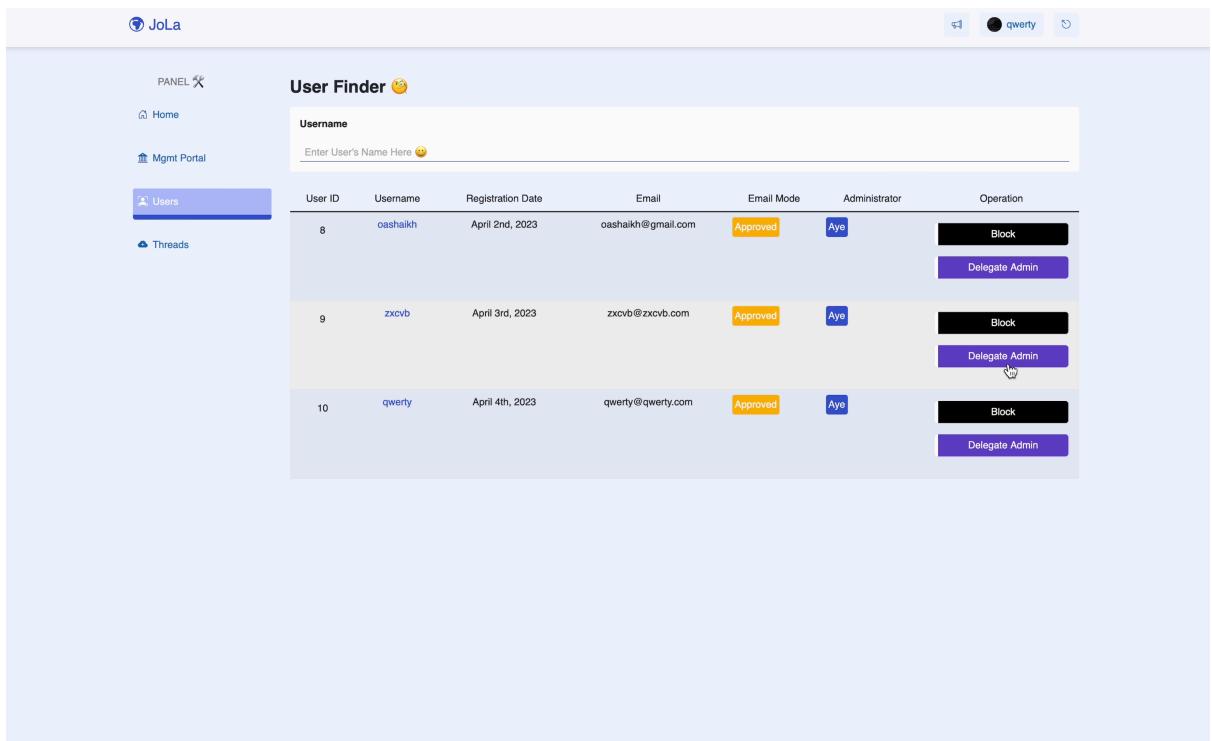
- Home
- Mgmt Portal
- Users**
- Threads

User Finder 😊

Username

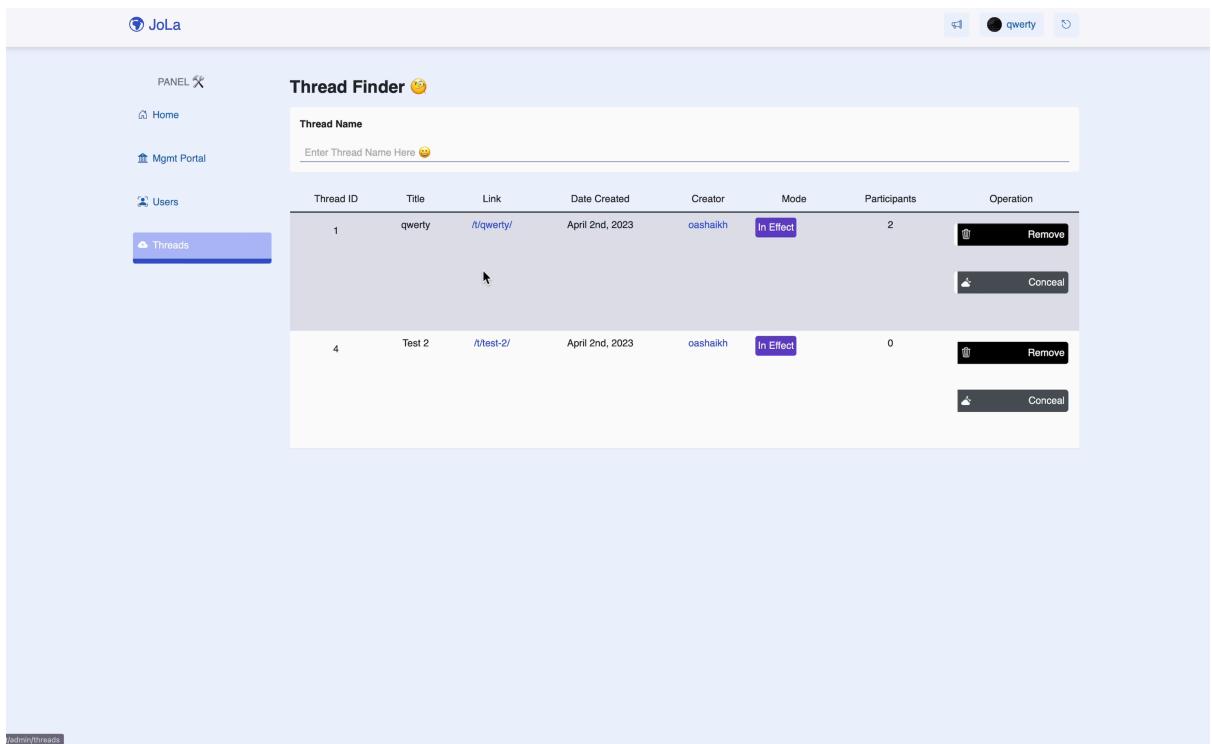
Enter User's Name Here 😊

User ID	Username	Registration Date	Email	Email Mode	Administrator	Operation
8	oashaikh	April 2nd, 2023	oashaikh@gmail.com	Approved	Aye	Block Delegate Admin
9	zxcvb	April 3rd, 2023	zxcvb@zxcvb.com	Approved	Nay	Block Promote Admin
10	qwerty	April 4th, 2023	qwerty@qwerty.com	Approved	Aye	Block Delegate Admin

A screenshot of the JoLa Admin Panel showing the 'User Finder' section. The sidebar on the left includes links for Home, Mgmt Portal, Users (selected), and Threads. The main area has a title 'User Finder 😊' and a search bar 'Enter User's Name Here 😊'. A table lists users with columns: User ID, Username, Registration Date, Email, Email Mode, Administrator, and Operation. The table shows three users: oashaikh (User ID 8), zxvcvb (User ID 9), and qwertry (User ID 10). Each user row has a 'Block' button (disabled for oashaikh) and a 'Delegate Admin' button.

User ID	Username	Registration Date	Email	Email Mode	Administrator	Operation
8	oashaikh	April 2nd, 2023	oashaikh@gmail.com	Approved	Aye	Block Delegate Admin
9	zxvcvb	April 3rd, 2023	zxvcvb@zxvcvb.com	Approved	Aye	Block Delegate Admin
10	qwertry	April 4th, 2023	qwertry@qwertry.com	Approved	Aye	Block Delegate Admin

XV. Admin Threads Page

A screenshot of the JoLa Admin Panel showing the 'Thread Finder' section. The sidebar on the left includes links for Home, Mgmt Portal, Users (selected), and Threads (selected). The main area has a title 'Thread Finder 😊' and a search bar 'Enter Thread Name Here 😊'. A table lists threads with columns: Thread ID, Title, Link, Date Created, Creator, Mode, Participants, and Operation. The table shows two threads: 'qwertry' (Thread ID 1) and 'Test 2' (Thread ID 4). Each thread row has a 'Remove' button and a 'Conceal' button.

Thread ID	Title	Link	Date Created	Creator	Mode	Participants	Operation
1	qwertry	/t/qwertry/	April 2nd, 2023	oashaikh	In Effect	2	Remove Conceal
4	Test 2	/t/test-2/	April 2nd, 2023	oashaikh	In Effect	0	Remove Conceal

PANEL X

JoLa

qwerty

Thread Finder 😊

Thread Name
Enter Thread Name Here 😊

Thread ID	Title	Link	Date Created	Creator	Mode	Participants	Operation
1	qwerty	/t/qwerty/	April 2nd, 2023	oashaikh	Concealed	2	Recover
4	Test 2	/t/test-2/	April 2nd, 2023	oashaikh	In Effect	0	Remove

Threads

PANEL X

JoLa

qwerty

Thread Finder 😊

Thread Name
Enter Thread Name Here 😊

Thread ID	Title	Link	Date Created	Creator	Mode	Participants	Operation
1	qwerty	/t/qwerty/	April 2nd, 2023	oashaikh	Removed	2	Recover
4	Test 2	/t/test-2/	April 2nd, 2023	oashaikh	In Effect	0	Remove

Threads